Improved strategies for distance based clustering of objects on subsets of attributes in high-dimensional data

Kampert, M.M.D.

Improved Strategies for Distance Based Clustering of Objects on Subsets of Attributes in High-Dimensional Data

Proefschrift
ter verkrijging van
de graad van Doctor aan de Universiteit Leiden
op gezag van Rector Magnificus prof. mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op Woensdag 3 Juli 2019
klokke 16:15 uur

door

Maarten Marinus Dion Kampert
geboren te Bergen op Zoom
in 1985

Promotor:

Prof. dr. Jacqueline J. Meulman (Universiteit Leiden)

Samenstelling van de promotiecommissie:

Prof. dr. Bart de Smit (Universiteit Leiden, voorzitter)

Prof. dr. Aad J.W. van der Vaart (Universiteit Leiden, secretaris)

Prof. Jerome H. Friedman, Ph.D.(Stanford University)

Prof. dr. Mia Hubert (KU Leuven)

Prof. dr. Mark de Rooij (Universiteit Leiden)

Prof. dr. Richard D. Gill (Universiteit Leiden)

Contents

Chapter 1

Introduction to Clustering Objects on Subsets of Attributes

In line with Tukey (1977), data analysis can be broadly classified into two major types:

(i) exploratory or descriptive, meaning that the investigator does not have pre-specified models or hypotheses but wants to understand the general characteristics or structure of the high-dimensional data, and

(ii) confirmatory or inferential, meaning that the investigator wants to confirm the validity of a hypothesis / model or a set of assumptions given the available data.

The research that underlies this monograph mainly addresses methods of the exploratory type and is motivated by Clustering Objects on Subsets of Attributes (COSA), a framework that is presented in Friedman and Meulman (2004). The COSA framework is proposed for the clustering of objects in multi-attribute data. In particular, COSA focusses on clustering structures where the relevant attribute subsets for each individual cluster are allowed to be unique, or to partially (or even completely) overlap with those of other clusters. COSA was inspired by the analysis of high-dimensional data, recently subsumed under the name 'Omics', where the number of attributes P are much larger compared to the number of objects N.

COSA is embedded in a context that can be summarized into the following topics: cluster analysis, regularized attribute weighting, and distances. Each of these topics will be briefly introduced in the following sections, so that the relevant background information is explained for the chapters that follow in this monograph. Moreover, we also use the introduction to demarcate what will not be part of our focus. The introduction ends with a reader's guide and the research problems that are addressed in the following chapters.

1.1 Cluster Analysis

We build the introduction on the following perspective on the practice of cluster analysis from Kettenring (2006):

> 'The desire to organize data into homogeneous groups is common and natural. The results can provide either immediate insights or a foundation upon which to construct other analyses. This is what cluster analysis is all about – finding useful groupings that are tightly knit (in a statistical sense) and distinct (preferably) from each other.'

In accordance with Cormack (1971), we are interested in the exploration of data that may contain natural clusters that possess intuitive qualities of 'internal cohesion' and 'external isolation'. This search for these natural groupings is conducted in absence of the groups labels for each object, which explains why it is often called *unsupervised learning* (Duda et al., 2001; Hastie, Tibshirani, & Friedman (2009).

Jain (2010) describes three different purposes of cluster analysis:

1. Finding an underlying clustering structure: to gain insight into data, generate hypotheses, detect anomalies, and identify salient features.
2. Natural classification: to identify the degree of similarity among forms or organisms (e.g., phylogenetic relationship).
3. Compression: as a method for organizing the data and summarizing it through cluster prototypes.

In this monograph, the first purpose will be of main importance, the second purpose is of lesser importance, and the third purpose is only of importance if it serves the first two purposes.

There are many books on the sole subject of cluster analysis; e.g., we consulted Hartigan (1975), Jain and Dubes (1988), Kaufman and Rousseeuw (1990), Arabie, Hubert, and De Soete (1996), and Aggarwal and Reddy (2013). When we searched for 'cluster analysis' via Google Scholar (2018), it gave about 19,800 results for the year 2018, about 79,400 results since 2014, and about 2,000,000 results when all restrictions on the time-period were removed. In other words, there is a vast literature in numerous scientific fields, indicating many different clustering problems, as well as thousands of different algorithmic implementations of cluster analysis.

1.1.1 Definitions of a Clustering Problem

Preferably, the choice of a cluster analysis method is motivated by a definition for the underlying clustering structure that is sought for. As soon as a definition of a cluster is formalized, then the clustering problem at hand can be operationalized into an optimization problem based on a (dis)similarity measure and an objective function. It may sound simple, but the formalization of the definition of a cluster by itself, is already one of the hardest problems. The main reason is that a general definition of a cluster,

> 'a group of homogeneous objects that are more similar to each other than to those of other groups',

is too vague. Although some research has been conducted into the appropriateness of many formalizations of cluster definitions (e.g., see Milligan 1996), a definite answer is far from agreed upon.

Even more so, for some specific combinations of cluster formalizations, it may be impossible to perform a cluster analysis (Fisher & Van Ness, 1971; Kleinberg, 2003). Fisher & Van Ness (1971) show that there exists no cluster analysis for which all the following properties can be satisfied at the same time:

- *convexity*: a cluster's convex hull does not intersect any of the other clusters;
- *points or cluster proportion*: the cluster boundaries do not alter when any of the objects or objects are duplicated a random number of times;
- *cluster ommission*: a cluster will always be found, even when other existing clusters are removed from the data;
- *monotonicity*: the clustering results should not change when a monotone transformation is applied to the elements of the similarity matrix.

Similarly, Kleinberg (2003) provided a proof for what has been referred to as the 'Impossibility Theorem for Clustering', which states that it is impossible to perform a cluster analysis that satisfies the following three properties:

- *scale invariance*: the cluster analysis method should account for the same results after an arbitrary scaling of the distance measure;
- *consistency*: a shrinkage of the within-cluster distances between objects and an expansion of the of the between-cluster distances should not change the identified cluster structure;
- *richness*: the cluster analysis method is able to achieve all possible partitions on the data.

1.1.2 Towards a Criterion instead of a Data Model

Despite the fact that some combinations of simple formalizations of the cluster definition cannot lead to a complete and satisfactory cluster analysis method, there are still many implicitly and explicitly formalized cluster definitions that possess many useful properties that can lead to meaningful cluster analysis methods. The most fundamental formalizations of the cluster definition are based on mixture(s) of multivariate probability distributions as an underlying model for the data, such that a likelihood can be formulated. Having defined a likelihood, the parameters for the underlying mixture of multivariate probability distributions can be optimized within the framework of *Bayesian statistics*, or *frequentist statistics*, which does not require any prior beliefs with respect to the set of the parameters. For some examples we refer to Fraley

and Raftery (2002), Hoff (2006), Wang and Zu (2008), Bouveyron and Brunet (2013), Celeux et al. (2014), and McLachlan (2017).

A common belief, however, is that data models are an oversimplification of the real underlying generative mechanism for the data, and most of the algorithms that follow from these cluster definitions are often too computationally expensive (Breiman, 2001; Friedman & Meulman, 2004; Harpaz & Haralick, 2007; Steinley & Brusco, 2008; Wiwie, Baumbach, & Röttger, 2015). In agreement with Breiman (2001), we do not necessary seek to identify an underlying data model from which the clustering structure follows. Instead, the emphasis in this monograph is on the algorithmic processing of high-dimensional data for the extraction of useful information, referred to as a focus of Data Science in Efron and Hastie (2016). In Breiman (2001), this particular focus could have been described as working with an 'algorithmic' model where the data mechanism is treated as unknown. Still, we rely on the formalization of a cluster definition into a criterion that needs to be optimized. Examples of such cluster analysis methods that are relevant for this monograph are Jing and Huang (2007), Steinley and Brusco (2008), Witten and Tibshirani (2010), and Arias-Castro and Pu (2017). Note however, that these type of clustering methods do not conflict with the idea that a data set can be seen as of the sampling result of a generating (data) model. The generating mechanism is simply assumed unknown, and therefore the clustering methods circumvent the choice for a specific mixture of multivariate parametric probability models.

1.1.3 Partitioning and Hierarchical Clustering

At least implicitly, many of the clustering structure definitions can be derived from the definition of a clustering algorithm. The clustering algorithm is the set of rules that is used to find a (local) optimum of the criterion for a clustering structure that may underlie a specific dataset $\mathbf{X}$ of size $N \times P$, which consists of N objects with each P attribute values; or for a clustering structure that may underlie a specific distance matrix $\mathbf{\Delta}$ for the objects of size $N \times N$. It is noteworthy that a distance matrix most often is derived from $\mathbf{X}$, see Section 1.3. Depending on the criterion, clustering algorithms lend themselves to be broadly divided into two kinds of clustering problems: partitioning and hierarchical clustering.

Partitioning

The purpose of a partitional cluster analysis method, is to find a partition of the data into L mutually exclusive clusters, defined as the set of clusters

$$\mathcal{C} = \{C_l \,|\, l = 1, \ldots, L\}\,, \tag{1.1}$$

where C_l is a subset of the set of N objects, represented by their indices $\{1, \ldots, N\}$. The partition $\mathcal{C}$ exhausts the set of N objects. Let Γ be the set of all possible partitions for $\mathcal{C}$, then based on the matrix $\mathbf{X}$, or the distance matrix $\mathbf{\Delta}$, the 'best' estimate for $\mathcal{C}$ is found by optimization of a specific criterion $Q_L(\mathcal{C})$. The general formulation (cf.

Van Os, 2001) of this optimization problem for partitional clustering is

$$\begin{aligned} &\operatorname*{opt}_{\mathcal{C}} Q_L(\mathcal{C} \,|\, \mathbf{X} \cup \mathbf{\Delta}), \\ &\text{subject to } \begin{cases} C_l \in \mathcal{C} \in \Gamma, \\ C_l \neq \emptyset, \\ C_l \cap C_{l'} = \emptyset, \\ \bigcup_{l=1}^{L} C_l = \{1, \ldots, N\}. \end{cases} \end{aligned} \tag{1.2}$$

Note the number of partitions in Γ is a Stirling number of the Second kind. Thus, computing $Q_L(\mathcal{C} \,|\, \mathbf{X})$ over all partitions is computationally intensive for a large N. Therefore, often an iterative heuristic algorithm is used to minimize $Q_L(\mathcal{C} \,|\, \mathbf{X})$, which (most likely) ends with a solution for $\mathcal{C}$ that results in a local optimum.

To our knowledge, the most popular and widely used algorithm for such a partitional clustering problems is referred to as K-means (Jain, 2010). The K-means algorithm has been (re-)discovered in multiple scientific fields by, e.g., Steinhaus (1956), Lloyd (proposed in 1957, published in 1982), Ball and Hall (1965), and MacQueen (1967). In K-means, where according to our notation $K = L$, we define $Q_C(\mathcal{C} \,|\, \mathbf{X})$ to be

$$Q_L(\mathcal{C}, \mathbf{M} \,|\, \mathbf{X}) = \sum_{l=1}^{L} \sum_{i=1}^{N} c_{il} \sum_{k=1}^{P} (x_{ik} - \mu_{kl})^2, \tag{1.3}$$

where $c_{il} = 1$ when object $i \in C_l$, or zero otherwise; and $\mathbf{M}$ is the matrix of size $P \times L$ in which each μ_{kl} is collected. Here, μ_{kl} is defined as a prototypical value on attribute k within cluster C_l. Here, the (local) optimum for $\mathcal{C}$ (and $\mathbf{M}$) is found when $Q_L(\mathcal{C}, \mathbf{M} \,|\, \mathbf{X})$ is minimized for $\mathcal{C}$ and $\mathbf{M}$ with the use of an alternating least squares algorithm.

Many other algorithms that are based on the partitional clustering problem or comparable problems, can be formulated as an extension or enhancement of the K-means criterion. Iconic examples are

- K-medoids clustering, where the distances between objects and a prototype object (medoid) within a cluster are minimized (Kaufman & Rousseeuw, 1987);
- fuzzy C-means, which is proposed by Dunn (1973) and later improved by Bezdek (1981), where the regularity conditions of mutually exclusive cluster is relaxed such that objects are allowed to be a member of multiple clusters through 'soft' assignment by changing the domain for c_{il}, i.e., requiring

$$0 < c_{il} < 1, \text{ subject to } \sum_{l=1}^{K} c_{il} = 1; \tag{1.4}$$

- a K-means algorithm by De Soete & Carroll (1994), which is based on a reduced space spanned by the $k = 1, \ldots P$ attributes.

A larger list of examples is available in Bock (2007).

Hierarchical Clustering

Hierarchical clustering is performed on the $N \times N$ distance matrix $\mathbf{\Delta}$. For the general formulation of hierarchical clustering we let $\mathcal{H} = \{\mathcal{C}_m \,|\, m = 1 \ldots N\}$ be a set of nested partitions of the objects. By merging two subsets of $\mathcal{C}_{m+1}$, we form $\mathcal{C}_m$, and thus a cluster hierarchy, explaining the name 'hierarchical clustering'. The general formulation (cf. Van Os, 2001) for the hierarchical clustering problem is

$$\begin{aligned} &\underset{\mathcal{H}}{\text{opt}}\, Q_H(\mathcal{H} \,|\, \mathbf{\Delta}), \\ &\quad \text{subject to} \\ &\quad \begin{cases} \mathcal{C}_m \text{ is a partition of } \{1, \ldots N\} \\ C_l \subseteq C_{l'} \text{ or } C_l \cap C_{l'} = \emptyset, \text{ for all } C_l \in \mathcal{C}_m \text{ and } C_{l'} \in \mathcal{C}_{m+1}. \end{cases} \end{aligned} \tag{1.5}$$

The number of all possible hierarchical structures for $\mathcal{H}$ is the product of N different Stirling numbers of the second kind. In other words, compared to the partitional clustering problems, it is even harder to find a solution for $\mathcal{H}$ that is an optimum for operationalizations of $Q_H(\mathcal{C} \,|\, \mathbf{\Delta})$.

Often an incremental sum of costs is minimized, i.e., the incremental cost that comes from forming partition $\mathcal{C}_{m+1}$ from $\mathcal{C}_m$. Such an idea perfectly lends itself for recursive hierarchical clustering algorithms that either find nested clusters in an agglomerative (bottom-up) mode, or in a divisive (top-down) mode. In the agglomerative mode every data point will start as a cluster on its own (a *singleton*) and at every transition from $\mathcal{C}_{m+1}$ to $\mathcal{C}_m$, the least dissimilar pair of clusters is merged, until a cluster hierarchy $\mathcal{H}$ is formed. In the divisive mode all data points start as one cluster, and then by recursion each cluster is divided into two smaller clusters.

Since the agglomerative mode is most well-known and will take part in this monograph, we will expand on its algorithmic steps. In the first step there are N singleton clusters, i.e., each object is a cluster on its own. Then, the strategy is to merge the two singleton clusters l' and l'' that are least dissimilar. Here, the initial distance $d_{l'l''}$ between to singleton clusters is exactly the same as the distance between two individual objects.

Having merged clusters l' and l'', new distances should be computed between the merged cluster that consists of $C_{l'} \cup C_{l''}$, on the one hand, and each remaining cluster C_l, on the other hand. The most well-known definitions of such (updated) distances are formulated via the general Lance-Williams (1967) update formula:

$$d^2_{l,l' \cup l''} = \alpha_1 d^2_{ll'} + \alpha_2 d^2_{ll''} + \beta d^2_{ll'} + \gamma |d^2_{ll'} - d^2_{ll''}|, \tag{1.6}$$

where $|x|$ indicates the absolute value of x, and where the parameters α_1, α_2, β and γ can take different values as specified in Table 1.1. This particular distance, $d^2_{l,l' \cup l''}$ from (1.6), is also referred to as a linkage function that represents the incremental costs between merging cluster C_l with the already merged cluster $C_{l'} \cup C_{l''}$. The point is that at each step a cluster merger is found for which the incremental costs are minimal. It is of interest to note that in the original study by Lance and Williams (1967), each $d_{l,l' \cup l''}$ was not squared, which is necessary for the original Ward linkage function (Wishart, 1969; Murtagh & Legendre, 2014).

The four most well-known linkage functions can all be obtained via the Lance-Williams update formula, and are referred to as the single-link (Florek et al., 1951a; Florek et al. 1951b; McQuitty, 1957; Sneath, 1957); average-link (Sokal & Michener, 1958); complete-link(Sorensen, 1948); and Ward-link functions (Ward, 1963). For each of these four linkage functions we have described in Table 1.1 the definitions of the parameters in equation (1.6). In this monograph we will show hierarchical clustering results for the average-link function and the Ward-link function. Compared to parti-

Table 1.1: The Lance-Williams parameter definitions for the Single, Average, Complete and Ward linkage functions.

link:	α_1	α_2	β	γ
Single	$1/2$	$1/2$	0	$-1/2$
Average	$\frac{N_{l'}}{(N_{l'}+N_{l''})}$	$\frac{N_{l'}}{(N_{l'}+N_{l''})}$	0	0
Complete	$1/2$	$1/2$	0	$1/2$
Ward	$\frac{N_l+N_{l'}}{N_l+N_{l'}+N_{l''}}$	$\frac{N_l+N_{l''}}{N_l+N_{l'}+N_{l''}}$	$-\frac{N_l}{N_l+N_{l'}+N_{l''}}$	0

tioning algorithms, hierarchical clustering is especially preferred in situations where there may be no underlying "true number of L clusters. Moreover, hierarchical clustering results lend themselves for easy visualization by means of a dendrogram. Such a dendrogram offers the domain expert of the data the opportunity to search for a partition of the objects by cutting the dendrogram at certain heights.

1.2 Clustering in High-Dimensional Data Settings

Due to advances in technology and the collection of larger amounts of data, we do not only see a rise in the numerous applications for cluster analysis, but often also face 'the curse of dimensionality'. The more attributes there are present on which a cluster structure is not distinguished, the more difficult it may become to recover the cluster structure. Among others, this has been demonstrated already by Milligan (1980); DeSarbo and Mahajan (1984); and DeSarbo, Carroll, Clark, and Green (1984). Since traditional clustering algorithms consider all of the attributes of a data set as equally important, more advanced algorithms are needed to separate the irrelevant from the relevant attributes to recover the cluster structure. For these advanced algorithms the search space of possible solutions is increased since it combines the search for a clustering structure with the search for the relevant attributes.

The need to distinguish signal attributes from noise attributes becomes even larger in high-dimensional data settings. In these settings the number of attributes P is much higher than the number of objects N (Parsons, Haque & Liu, 2004; Jain, 2010; Kriegel, Kröger, & Zimek, 2009). Under the assumption that the fraction of signal attributes versus noise attributes becomes smaller for larger P, it is common that all objects in the data would become nearly equidistant from each other, and hence, the underlying cluster structure is even more difficult to recover.

1.2.1 Weighting of the Attributes

Cluster algorithms for high-dimensional data cannot do without a strategy where the signal is separated from the noise. A common, but crude, strategy that has been applied as a solution was to first conduct a principal component analysis (PCA), and to perform a clustering on the first principal components or attribute combinations. Such a procedure, referred to as tandem-clustering has been criticized in e.g., De Soete and Carroll (1994). Although the first few principal components represent the most information for the data set as a whole, these principal components do not necessarily represent the most information for the underlying cluster structure. There are clustering algorithms, however, that apply PCA in such a way that they seek a lower dimensional column space for the attributes that is maximally informative about a detected clustering structure; Van Buuren and Heiser (1989), De Soete and Carroll (1994), Lee et al. (2010), and Jin and Wang (2016). Nevertheless, these algorithms will not be part of the focus of this monograph; these particular clustering structures are difficult to interpret since they are based on one (lower dimensional) projection of the column space.

Better suited for interpretation are the clustering algorithms that have a strategy to mitigate or remove the irrelevant attributes in the data set via attribute weights or a simple selection of a subset of attributes. Thus, each to be identified cluster has the same weighting for the attributes, or the same number of selected attributes. An example of a clustering algorithm where the attributes are weighted can be found in Witten and Tibshirani (2010), and Arias-Castro & Pu (2017) is an example where the attributes are selected.

The main challenge of these type of clustering methods for high-dimensional data is to prevent overfitting. Most often, overfitting is prevented by exchanging some of the variance with bias (Hastie, Tibshirani & Friedman, 2009) such that the solution space for the selection or weighting of the attributes is restricted. For a review on how to deal with the weighting or selection of attributes, see Saeys, Inza, and Larrañaga (2007).

1.2.2 Subspace Clustering

Up until now we discussed attribute weighting and selection in the dataset as a whole, in this monograph we will concentrate on a specific clustering method for high-dimensional data that finds for each cluster its own unique attribute weights or subset of attributes. Such clustering methods are able to uncover clusters that exist in multiple, possibly overlapping subspaces of the attributes. In Parsons et al. (2004) these are referred to as subspace clustering or projected clustering methods, and defined as techniques that

> "seek to find clusters in a dataset by selecting the most relevant dimensions for each cluster separately."

In a systematic review of the intrinsic methodological differences for clustering problems, Kriegel et al. (2009) distinguish subspace and projected clustering from

correlational clustering, and from *biclustering* approaches. In subspace clustering, and its synonym *projected clustering*, the clusters are sought for in *axis-parallel* subspaces, i.e., the clusters are seen as objects that are close to each other based on a cluster-specific (weighted) sum of the attributes. In *correlational clustering* methods, however, clusters are sought for in any of the arbitrarily oriented subspaces. Here, each cluster can have its own non-linear mappings of the attribute space. Biclustering methods are a hybrid mix of subspace clustering and correlational clustering methods. Each cluster is a realization of a restricted axis-parallel subspace or restricted affine subspace of the attributes (for examples see Van Mechelen, Bock and De Boeck (2004), or Oghabian et al. (2004)). Of these three types of clustering methods, the results from subspace clustering and projected clustering are directly based on the original attributes, and therefore the easiest to interpret, which is an important aspect for exploratory data analysis.

Although often interchangeably used in this monograph, Kriegel et al. (2009) also narrow the definition of subspace clustering to be able to separate it from projected clustering. In the projected clustering algorithms each object is assigned to exactly one subspace cluster, while subspace clustering algorithms are more general, and aim to find all clusters in all (axis-parallel) subspaces. By exploring all subspaces, an object can belong to multiple clusters from different subspaces. Moreover, a subspace of a cluster is referred to as 'soft' when all attributes in the data are used, but just differently weighted. A last type of subspace algorithms described in Kriegel et al. (2009), are the 'hybrid' subspace algorithms. For these algorithms it is neither necessary that all objects are assigned to a cluster, nor all (axis-parallel) subspaces need to be explored, e.g., hybrid soft subspace algorithms may bear a close relationship to exploratory projection pursuit (Friedman & Tukey, 1974; Friedman, 1985). Projection pursuit seeks to find lower dimensional linear projections of the data that reveal structure, e.g., clustering structure(s) (cf. Kruskal, 1972).

Finding the Subspace

For subspace clustering algorithms, the attribute subspace is usually found based on the clustering structure of the objects, and vice versa. The way the algorithms find the attribute subspaces relevant for clustering is often divided into two approaches: bottom-up and a top-down. In the bottom-up approach, first an attempt is made to find the subspaces, and then cluster members are found for each of the subspaces. Often, for each attribute a histogram is created. Then, from the bins of the attribute histograms that are above a given threshold, the next attribute is found on which (most of) these objects also have a high density such that a two-dimensional subspace is created. This process is repeated until there are no 'dense' subspaces left. When the subspaces are formed, then the adjacent dense objects are combined to form a cluster, often leading to overlapping clusters (Parsons et al., 2004; Kriegel et al. 2009).

In the top-down approach a start is made with an initial clustering structure obtained from the full attribute space where each attribute has equal weight. Then, for each cluster or group of objects an attribute weight is obtained for each attribute. Based on these updated attribute weights, an updated clustering structure is obtained,

and so forth. The top-down approach is often applied for partitional clustering (Parsons et al., 2004; Deng et al, 2016).

Subspace Clustering in this Monograph

The research in this monograph is motivated by the *Clustering Objects on Subsets of Attributes* approach, referred to as COSA (Friedman & Meulman, 2004). According to the strict framework of Kriegel et al. (2009), COSA is a hybrid soft subspace algorithm. In this monograph the focus is on subspace clustering methods, where the subspaces are assumed to be axis-parallel. Moreover we restrict ourselves to the top-down approach for these algorithms.

1.3 Distances and COSA[1]

Whether or not a clustering algorithm can be rooted back to partitional clustering or hierarchical clustering: there is always a notion of (dis)similarity between the objects, or a notion of (dis)similarity between an object and the representative attribute values of a cluster. A notion of a distance (dissimilarity), or proximity (affinity), can be used for almost any data set. Aggarwal (2013) even describes that 'the problem of clustering can be reduced to the problem of finding a distance function for that data type' at hand. Not surprisingly, representative distance functions for high-dimensional data have become a solid field of research (Aggarwal, 2003; Pekalska & Duin, 2005; France, Carroll, & Xiong, 2012; Wang & Sun, 2014).

In this monograph the notions distance and dissimilarity are used interchangeably to facilitate reading. However, there is a formal distinction between the two. While a distance, denoted by D_{ij}, satisfies

$$\begin{aligned} \textit{non-negativity}&: D_{ij} \geq 0, \\ \textit{reflexivity}&: \text{ if } x_{ik} = x_{jk} \text{ for all } k, \text{ then } D_{ij} = 0, \\ \textit{symmetry}&: D_{ij} = D_{ji}, \\ \textit{triangluar inequality}&: D_{ij} \leq D_{ih} + D_{hj}, \end{aligned}$$

a dissimilarity does not need to comply with the triangular inequality condition. Thus, a distance is a dissimilarity, while the converse need not hold. Whenever the triangular inequality is violated by what we call a distance, it will be pointed out.

1.3.1 About Distance Functions

The most classical distance functions for data sets are those that fall in the framework of the Minkowski distance. To define this distance, let x_{ik} denote the value of object i on attribute k, and define the attribute distance d_{ijk} between a pair of objects i and j as follows:

$$d_{ijk} = |x_{ik} - x_{jk}|, \tag{1.7}$$

[1] A large part of this section is from Kampert, Meulman, and Friedman (2017).

the absolute difference between object i and j on attribute k. Then, the definition of the ℓ_p Minkowski distance is

$$D_{ij}^{(\ell_p)} = \left(\sum_{k=1}^{P} d_{ijk}^{p} \right)^{\frac{1}{p}}. \tag{1.8}$$

When $p = 1$ we obtain the Manhattan distance, with $p = 2$ we obtain the Euclidean distance.

A common notion is that these specific distance functions, where all attributes have equal weight, cannot represent the clustering structure in high-dimensional data settings. Assuming that that the percentage of signal in these data sets becomes smaller when the number of attributes, denoted by P, becomes larger, we have for each object i:

$$\lim_{P \to \infty} \frac{\max_j(D_{ij}) - \min_j(D_{ij})}{\min_j(D_{ij})} \to 0, \tag{1.9}$$

i.e., an increasing dimensionality of the data set in P, renders the distance function less meaningful since it becomes more difficult to distinguish object i with its farthest and its closest neighbor. Thus, distance functions that equally weight the attributes, are not able able to reveal an underlying (clustering) structure in the data (Beyer et al. 1999; Hinneburg et al. 2000; Aggarwal et al. 2001).

Clustering algorithms that work with distances that incorporate attribute weighting are more likely to succeed in finding the cluster structure in the data. Even for the smaller multivariate data sets where the number of attributes (P) is smaller than the number of objects (N) the notion of attribute weights in distance functions received attention, e.g., Sebestyen (1962), Gower (1971), De Soete, De Sarbo and Carroll (1985), De Soete (1988). For these specific examples, however, the estimation procedure for the weights of the attributes heavily relies on the assumption that $N > P$, while in high-dimensional data sets $P \gg N$, rendering degenerate solutions for the attribute weights.

Using the Manhattan distance as an example, a distance function that allows for attribute weighting would have the form

$$D_{ij}[\mathbf{w}] \quad = \quad \sum_{k=1}^{P} w_k d_{ijk}, \tag{1.10}$$

where $w_k \in \mathbb{R}_{\geq 0}$ is the weight for attribute k, but is subject to a number of restrictions to prevent degenerate distance representations. While these restrictions bound the variance of the distance to a 'proper' and low level, it also causes an increase in the bias of the distance, the bias-variance trade-off (e.g., see Hastie et al. 2009). The aim is to find restrictions that represent an optimal balance in the exchange between lower variance and higher bias. An example of such a weighted distance can be found in Witten and Tibshirani (2010), and would be coping well on a data set as displayed in Figure 1.1.

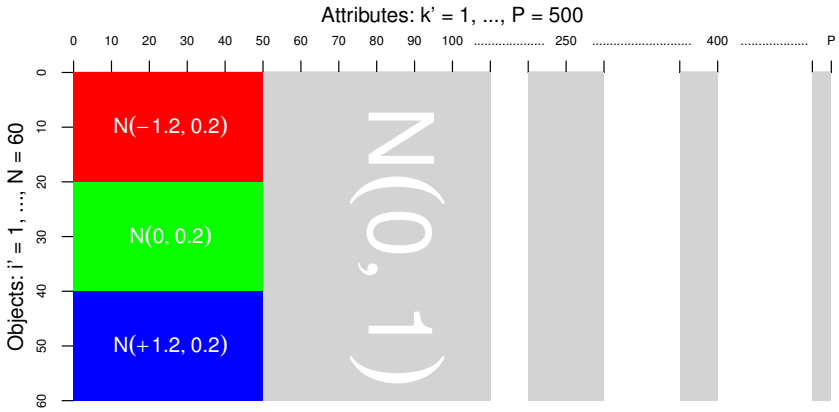

Figure 1.1: A Monte Carlo data set $\mathbf{X}$ *with 60 objects (vertical) and 500 attributes (horizontal, not all of them are shown due to* $P \gg N$*). There are three groups of 20-objects each (red, green, and blue) clustering on 50 attributes. Note that* i *and* k *are ordered into* i' *and* k'*, respectively, to show the cluster blocks. After generating a data set from this model, each attribute is standardized to have zero mean and unit variance.*

Up until now we showed a distance function for an example as displayed in Figure 1.1, where only one subset of attributes is important for all groups of objects. In this particular example, all objects are assumed to belong to a cluster each. In particular, there are no objects in the data that do not belong to any of the clusters. This is a very particular structure, and it is unlikely to be present in many high dimensional settings, but it is assumed in most clustering approaches.

1.3.2 COSA Distances

In many data sets, one can hope to find one or more clusters of objects, while the remainder of the objects are not close to any of the other objects. Moreover, it could very well be true that one cluster of objects is present in one subset of attributes, while another cluster is present in another subset of attributes. In this case, the subsets of attributes are different for each cluster of objects, and therefore the distance functions in equations (1.8) and (1.10), cannot be a good representation.

In general, the subsets on which objects cluster may be overlapping or partially overlapping, but they may also be disjoint. An example is shown in Figure 1.2; the display shows a typical structure in which the groups of objects cluster on their own subset of attributes. The first group (with objects 1-15) clusters on the attributes 1-30, and the second group (with objects 16-30) clusters on attributes 16-45. So the

two groups are similar with respect to attributes 16-30, and different with respect to attributes 1-15 and 31-45, respectively. The two subsets of attributes, 1-30 and 16-45, are partially overlapping. The remaining 70 objects in the data form an nonclusterable background (noise), and the remaining 955 attributes do not contain any clusters at all.

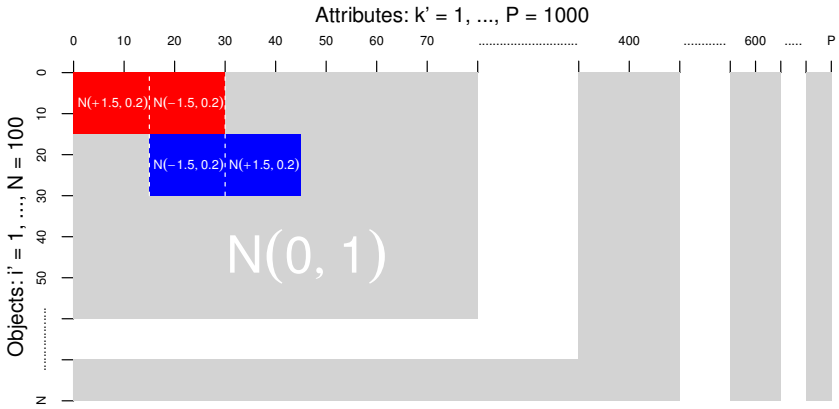

Figure 1.2: A Monte Carlo model for 100 objects with 1,000 attributes (not all are shown due to $P \gg N$). There are two small 15-object groups (red and blue), clustering each on 30 attributes out of 1000 attributes, with partial overlap, and nested within an unclustered background of 70 objects (gray). After generating a data set from this model, each attribute is scaled to unit variance with zero-mean.

In the rest of this monograph we will refer to the specific data model from Figure 1.2 as the prototype model, since it is this situation for which COSA (Friedman & Meulman, 2004) was designed. For datasets from the prototype model, a representative distance function should allow for attribute weights that are cluster specific. A COSA distance can be defined as

$$D_{ij}(\mathbf{v}_{ij}) = \sum_{k=1}^{P} v_{ijk} d_{ijk}, \tag{1.11}$$

where $v_{ijk} \in \mathbb{R}_{\geq 0}$, is an attribute weight for the object pair $\{i, j\}$ and can take into account the subspace of the cluster to which object i belongs and the subspace of the cluster to which object j belongs. As is the case with attribute weighted distance in equation (1.10), the solution space of the attribute weights for each v_{ijk} in equation (1.11) is regularized such that overfitting is prevented with the COSA distance. Note that with the use of the object pair specific attribute weights $\{v_{ijk}\}$, the COSA

distance does not need to satisfy the triangular inequality, even when the attribute distances do satisfy the triangular inequality. More details about the COSA distances will be given in Chapter 2.

1.3.3 Visualizing COSA Distances

As is shown in Friedman and Meulman (2004) and in Kampert, Meulman and Friedman (2017), the COSA distance can reveal clusters that are formed in attribute subspaces of high dimensional datasets, and represent them in easily interpretable and meaningful ways. In this monograph, most attention will be given to a visualization of lower-dimensional mappings of the distances by using hierarchical clustering or multidimensional scaling analysis (MDS). Both visualization types will be described in this subsection, and are based on the COSA distance of an example data set from the COSA prototype model.

The Dendrogram

A dendrogram that is obtained in hierarchical clustering is a fast way to display a possible clustering structure that is contained in the COSA distances. In Figure 1.3 we depict the dendrogram for average linkage hierarchical clustering on the COSA distances from the generated data from the prototype COSA model in Figure 1.2. Note that the grouping structure in the data set is revealed. There are two groups (each with 15 objects) and a large remaining group for which the objects are not similar to each other. The colors of the data points are according to the cluster structure colors from the prototype COSA model.

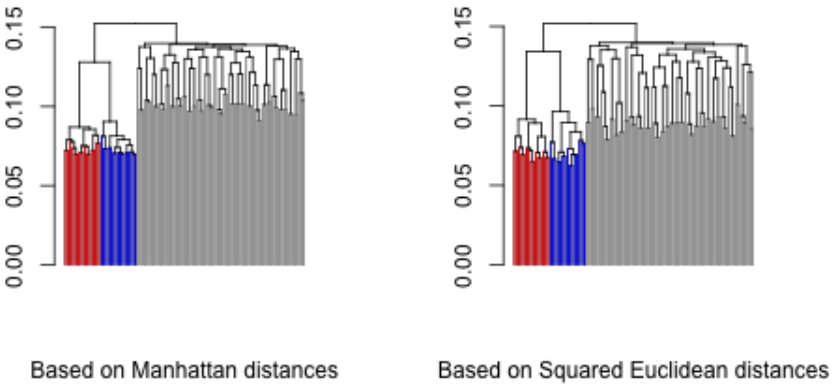

Figure 1.3: Average linkage dendrogram of the COSA distances (left panel) based on Manhattan attribute distances, and the average linkage dendrogram of the COSA distances based on squared Euclidean attribute distances (right panel).

Although the structure in the data is not particularly complex, the clustering structure would not have been revealed by ordinary types of distances, as is shown in

Figure 1.4. Applying hierarchical clustering to the Manhattan distance, or Squared Euclidean distance without attribute weighting, did not reveal the clustering structure. The same applies to the results obtained when these two distance measures were computed on weighted attributes, where the set of attribute weights would be the same for each distance (cf. Witten & Tibshirani, 2010). Although not shown, similar findings in this subsection would have been obtained for complete and single linkage hierarchical clustering.

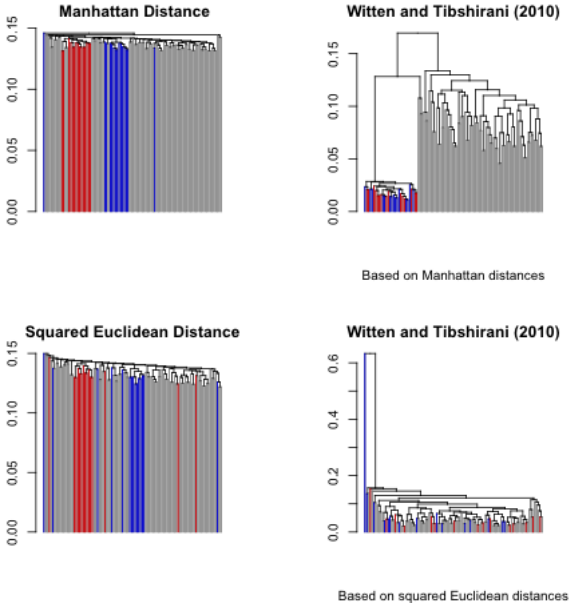

Figure 1.4: Dendrograms obtained from hierarchical clustering for four different distance matrices derived from the simulated data from the protoype model. The results of the Manhattan distances are shown in the first row, the results of the squared Euclidean distances are shown in the second row. The results in the first column represent the distance functions where all attribute have equal weights, in the second column the attributes are weighted in accordance with Witten and Tibshirani (2010).

Multidimensional Scaling

Apart from dendrogams, we can also use the COSA distances to display the objects in a low-dimensional space that is obtained by multidimensional scaling (MDS). This is done preferably by using an algorithm that minimizes a least squares loss function, usually called STRESS, defined on dissimilarities and Euclidean distances. This loss function (in its raw, squared, form) is written as:

$$\text{STRESS}(\mathbf{Z}) = ||\mathbf{\Delta} - D(\mathbf{Z})||^2 , \tag{1.12}$$

where $||\cdot||^2$ denotes the squared Euclidean norm. Here, $\mathbf{\Delta}$ is the $N \times N$ COSA dissimilarity matrix with elements $D_{ij}(\mathbf{v}_{ij})$ and $D(\mathbf{Z})$ is the Euclidean distance matrix derived from the $N \times p$ configuration matrix $\mathbf{Z}$ that contains coordinates for the objects in a $p-$dimensional representation space. An example of an algorithm that minimizes such a metric least squares loss function is the so-called SMACOF algorithm. The original SMACOF(Scaling by Maximizing a Convex Function) algorithm is described in De Leeuw and Heiser (1982). Later, the meaning of the acronym was changed to Scaling by Majorizing a Complicated Function in Heiser (1995).

The Classical Scaling approach, which is also known as Principal Coordinate Analysis, or Torgerson-Gower scaling (Young & Householder, 1938; Torgerson, 1952; Gower, 1966), uses an eigen value decomposition, and minimizes a loss function (called STRAIN in Meulman, 1986) that is defined on scalar products ($\mathbf{ZZ}'$) and not on distances $D(\mathbf{Z})$:

$$\text{STRAIN}(\mathbf{Z}) = \left|\left|\left(-\frac{1}{2}\mathbf{J}\mathbf{\Delta}^2\mathbf{J}\right) - \mathbf{ZZ}'\right|\right|^2, \tag{1.13}$$

where $\mathbf{J} = \mathbf{I} - N^{-1}\mathbf{11}'$, a centering operator that is applied to squared dissimilarities in $\mathbf{\Delta}^2$, $\mathbf{I}$ is the $N \times N$ identity matrix, and $\mathbf{1}$ is a vector with 1's.

The drawback of minimizing the STRAIN loss function is that the resulting configuration $\mathbf{Z}$ is obtained by a projection of the objects into a low-dimensional space. Due to this projection, objects having dissimilarities that are large in the data, may be displayed close together in the representation space, giving a false impression of similarity. By contrast, a least squares metric MDS approach (such as SMACOF) gives a nonlinear mapping instead of a linear projection, and will usually preserve large distances in low-dimensional space. See Meulman (1986, 1992) for more details.

In Figure 1.5 and Figure 1.6, the objects are displayed in the two-dimensional space of the least squares MDS solution and the classical MDS solution, respectively.

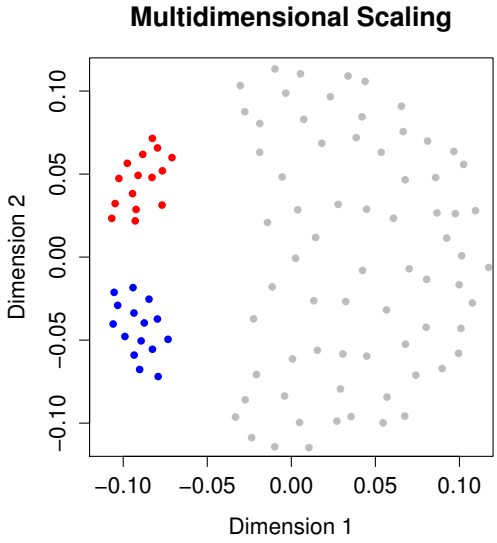

Figure 1.5: Metric least squares multidimensional scaling solution of the configuration

Figure 1.5 shows the metric least squares MDS configuration for the two groups of objects (in red and blue), while the gray objects show a typical representation of a high-dimensional cloud of points with equal distances, nonlinearly mapped into two-dimensional space. In Figure 1.6 we observe that the large cloud of gray points, representing objects that are not similar to any of the other objects, seem to form a cluster as well, while they are the noise objects in Figure 1.2. Their closeness is due to the linear projection characteristic for the classical MDS approach. Therefore, the representation given in Figure 1.5, is to be preferred since it shows that the noise objects are not closely related.

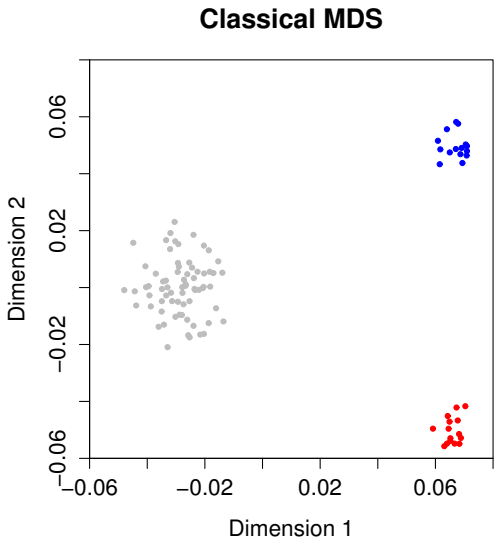

Figure 1.6: Classical multidimensional scaling solution of the configuration

.

1.3.4 Visualization of the Attribute Weights

After having found clusters of objects based on the COSA distances, it is also possible to explore the subsets of attributes that are important for different clusters. If the dispersion of the data in an attribute is small for a particular group of objects, than the attribute is important for that particular group. Suppose that all attribute distances are measured on the same scale, or are normalized to have the same scale on each attribute, then the importance of attribute k in cluster C_l is defined as

$$I_{kl} = \left(\frac{1}{N_l^2} \sum_{i,j \in C_l} c_{il} c_{jl} d_{ijk} + \epsilon \right)^{-1}, \tag{1.14}$$

where

$$N_l = \sum_{i=1}^{N} c_{il}, \tag{1.15}$$

and ϵ^{-1} represents the maximum obtainable importance value. While the importance value is favoring attributes with small within-group variability, it does not pay attention to separation between groups.

The attribute importance values of a cluster can be visualized and compared to attribute importance values that could have been expected based on chance only for random groups of objects of a similar size. In other words: to see whether the value

of a particular attribute importance is higher than could be expected by chance, a simple resampling method can be used. See Figure 1.7 for the visualization of the 50 highest attribute importance values for each cluster separately, as well as the remaining objects.

In Figure 1.7 the black line indicates the attribute importance of each attribute in each cluster. The green lines are the attribute importance lines for groups of the same size, consisting or randomly sampled objects from the data. The red line is the average of the green lines. The larger the difference between the black line and the red line, the more evidence that the attribute importance values are not just based on chance. Note the sudden drop of the black attribute importance line after 30 attributes. This effect is in line with the simulated data, in which each group is clustered on 30 attributes only. Moreover, notice that there are no attributes that can be considered important for the remaining objects.

In addition to the attribute importance values, we can look at the boxplots of the within-cluster attribute weights of the object pairs. For each group we show a boxplot of the attribute weights for each attribute $k' \in \{1, \ldots, 50\}$. The first 45 of these 50 attributes are the attributes that contribute to the clustering as is shown in COSA prototype model from Figure 1.2. It is clear that the COSA attribute weights display the same structure as was found for the attribute importances: group 1 has large weights for attributes 1-30, group 2 has large weights for attributes 16-45, group 1 and 2 have large weights on the overlapping attributes 16-30, and all weights for the remaining objects are small. To conclude: COSA clearly separates the signal from the noise in the data.

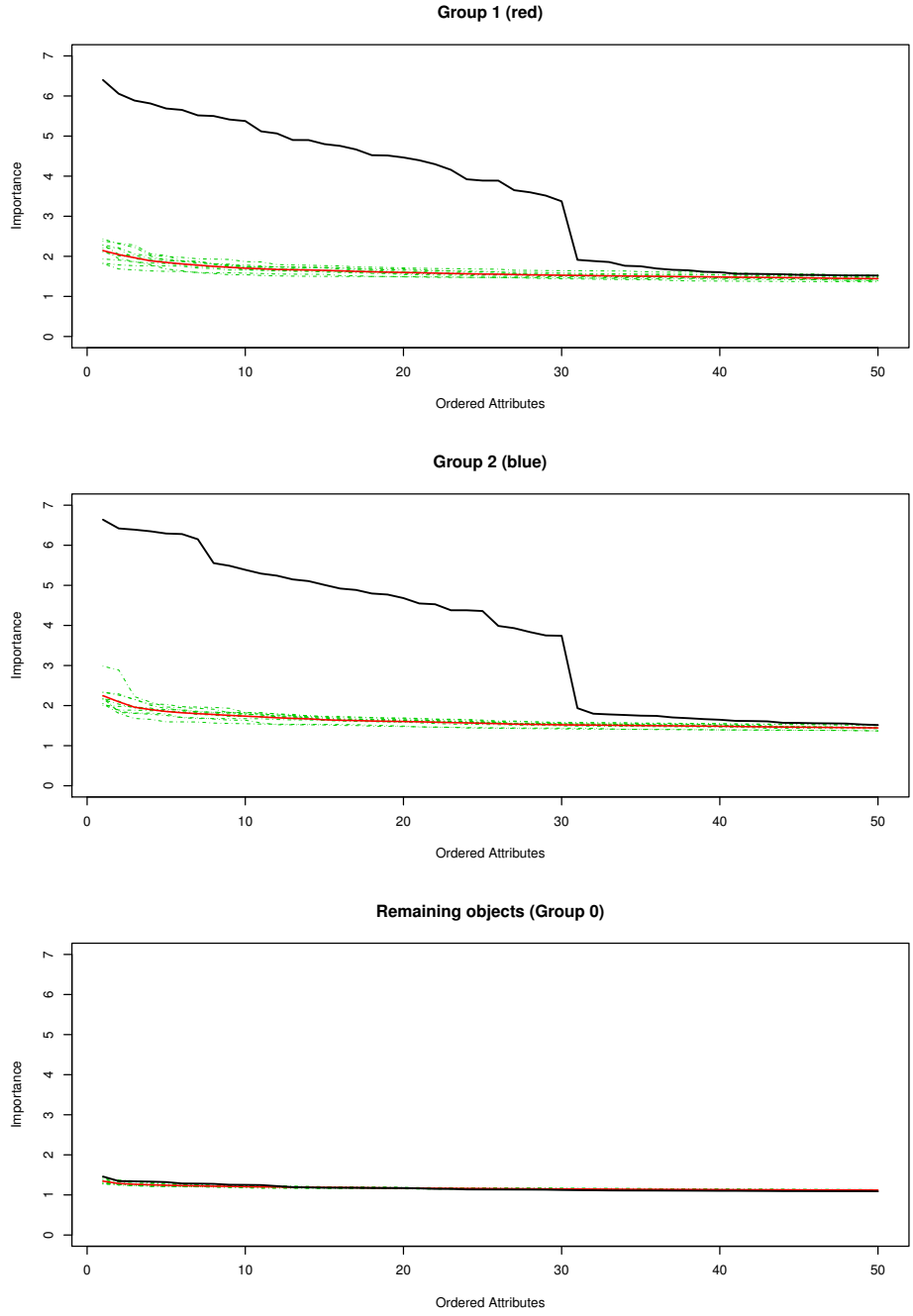

Figure 1.7: Display of the attribute importances of group 1, group 2 and the remaining objects. Here, ϵ is set equal to 0.05, such that maximum attribute importance value would have been 20.

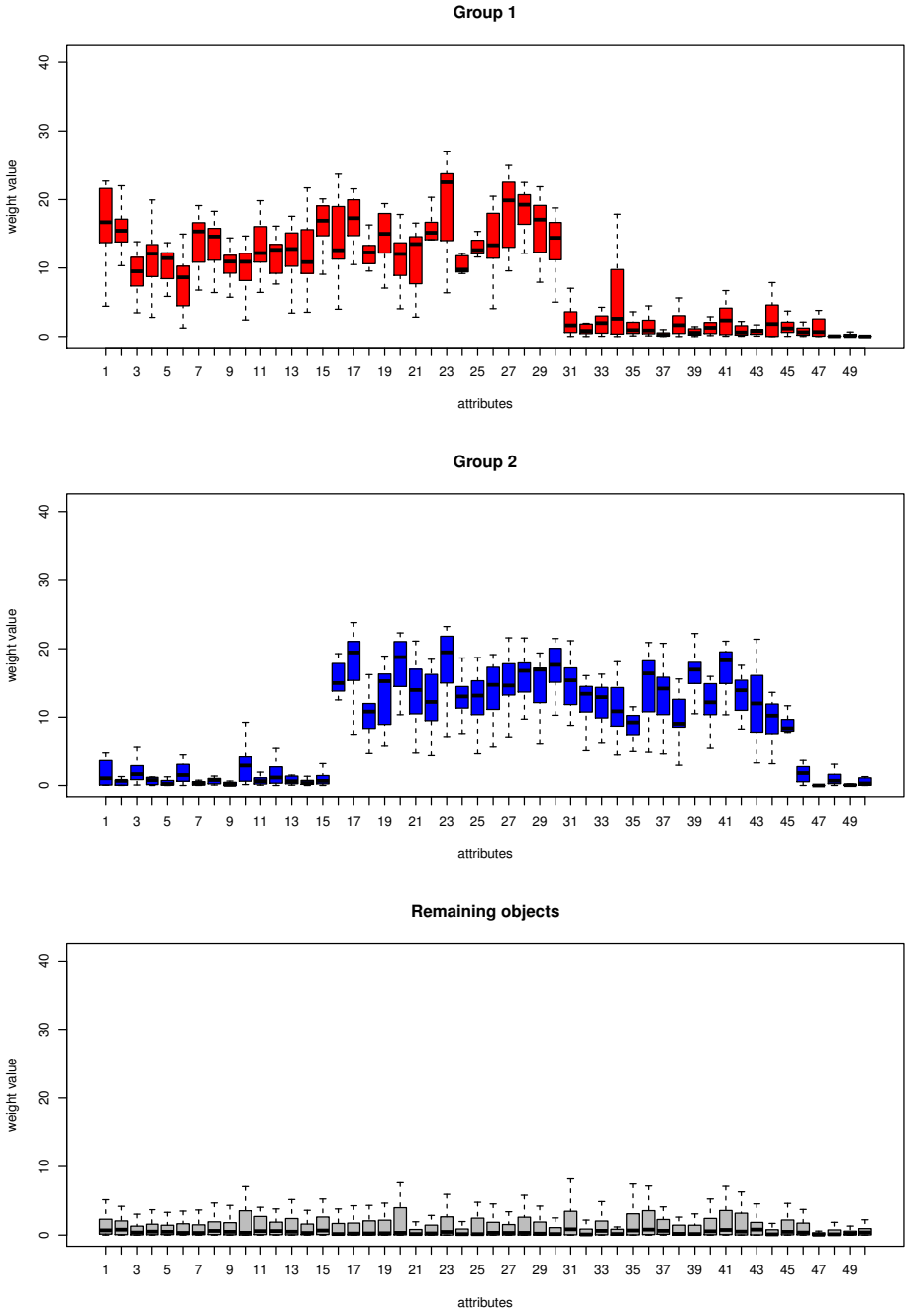

Figure 1.8: Boxplots of the weights of attributes $k' = \{1, \ldots 50\}$ for group 1, group 2 and the remaining objects.

1.4 Guide for this Monograph

The main product of the COSA framework are distances for the objects in a high-dimensional data set that contains a clustering structure, where each cluster can have its own subspace of attributes. We described the background knowledge that is needed for the introduction of the COSA framework in the first two sections. In the previous section we have demonstrated the use of the COSA distances for obtaining clusters of objects on subsets of attributes. In this section we specify the aim of this monograph and the outline of its chapters.

1.4.1 Aims

Although the COSA framework proposed in Friedman & Meulman (2004) has 504 citations on Google Scholar to the date of December 18, 2018, in most of these publications COSA was not applied for the purpose of exploratory analysis to form new hypotheses based on empirical data. We only know of Meulman (2003), Nason (2004), Damian et al. (2006), Lubke & McArtur (2014), Sánchez-Blanco et al. (2018), where the exploratory usefulness of COSA was demonstrated. More often, when COSA was applied, it served the purpose of a comparison with another proposed clustering method (Jing and Huang, 2007; Steinley & Brusco, 2008; Witten & Tibshirani, 2010). A reason that COSA's usefulness for exploratory analysis only got little attention may be related to the fact that it is being perceived as complex, and that the choices for its tuning parameters seem to be difficult (Jolliffe & Philipp, 2010; Witten & Tibshirani, 2010).

Here, it may not have been instrumental that Friedman & Meulman (2004) proposed two COSA algorithms: one general algorithm of a theoretical nature that got motivated in detail, but did not got implemented; and a COSA K-nearest neighbors algorithm that did get implemented, but was provided with only little motivation. This monograph builds upon the latter algorithm. To mitigate the criticism on the complexity and user-unfriendliness of COSA, the twofold purpose of this monograph is plain and simple: study the behavior of COSA to demonstrate its usefulness and where there are opportunities, improve COSA.

1.4.2 Outline of the Chapters

We start with a recapitulation of COSA K-nearest neighbors algorithm (COSA-KNN) in Chapter 2. Chapter 3 is about two improvements. First, it shows that median-based estimates for the attribute weights are more successful than the mean-based estimates in COSA-KNN. Second, the choice of two tuning parameters, K and λ, crucial in COSA is addressed. K denotes the size of the neighborhoods for each object in a cluster, and λ denotes the value that regulates the influence of the attribute weights. We propose and illustrate a strategy to choose optimal values for the tuning parameters, the so-called Gap statistic (based on Tibshirani et al., 2001).

Chapter 4 provides a series of improvements for COSA:

i. a different initialization of the attribute weights;

ii. allowing for zero-valued attribute weights;

iii. a COSA distance that better separates pairs of objects in different clusters;

iv. a reformulation of the COSA-KNN criterion such that the tuning parameter K becomes redundant;

We propose that λ can also be used to regulate size of the neighborhood, which renders the tuning parameter K superfluous. Moreover, it creates the possibility to find a different neighborhood size for each object. This approach and its associated algorithm will be called COSA-λNN. Examples will show the successfulness of the improvements.

In Chapter 5 we introduce a partitioning algorithm especially suitable to represent L clusters from a COSA distance matrix, referred to as MVPIN. At a first examination of its effectiveness, MVPIN seems to produce promising results in combination with distances from COSA-KNN, and especially from COSA-λNN. We compare COSA in combination with MVPIN to other state-of-the-art L clustering algorithms in Chapter 6, and find that it is a compelling option to find L clusters of objects in high-dimensional data. Chapter 6 also provides a strategy for a COSA-based cluster validation. Last, Chapter 7 provides a general discussion of the monograph.

Chapter 2

A Recapitulation of COSA

The purpose of this chapter is to restate the conceptual framework that underlies the algorithm for clustering objects on subsets of attributes (COSA). Since the algorithm uses a K nearest neighbors (KNN) method, we will refer to this algorithm as COSA-KNN. The properties and technical details of COSA-KNN received little attention in Kampert, Meulman, and Friedman (2017), and was referred to the original COSA paper by Friedman and Meulman (2004). However, the paper by Friedman and Meulman (2004) mainly provides the theoretical framework for another algorithm of which COSA-KNN is an approximation. Furthermore, the many extra details and options described in the COSA paper may obscure the main properties of COSA-KNN.

The main purpose of COSA-KNN is to produce a distance matrix that can serve as input for proximity analysis methods. This distance matrix contains the distances between N objects. A distance between object i and j, denoted by D_{ij}, is constructed from a weighted sum of P attribute distances. These P attribute distances, denoted by $\{d_{ijk}\}_{k=1}^{P}$, are in turn constructed from measurements collected in a $N \times P$ data set. The data set is assumed to have an underlying clustering structure in which the objects are clustered on cluster-specific subsets of attributes. To ensure this clustering is represented in the distance matrix, the attribute distances are weighted. The weighting procedure ensures that the distance matrix consists of small distances for object pairs within a cluster, and larger distances for the between-cluster object pairs. We will refer to this property as 'majorizing'.

The 'majorizing' distances are obtained from an iterative algorithm. We start with an initial set of attribute weights from which a first distance matrix can be derived. Then, based on a K nearest neighbors method, new attribute weights are calculated, from which in turn a new distance matrix is derived, and so on. This iterative procedure is continued until a particular convergence criterion is reached.

The chapter is organized as follows. In Section 2.1, we introduce a definition of the COSA distance. We present the underlying COSA framework and the motivation for the KNN method in Section 2.2. Section 2.3 is about obtaining the attribute weights, given the K nearest neighbors. Section 2.4 is about obtaining the K nearest neighbors given the attribute weights. Section 2.5 contains a review of the COSA-

KNN algorithm. Section 2.6 ends the chapter with a discussion.

2.1 Introduction to the COSA Distance

Visual representations of distances are very popular in data analysis methods. These distance-based methods are advantageous for discovery, identification and recognition of an underlying clustering structure in the data. Nowadays, there are many data settings in which the the number of attributes is much larger than the number of objects, i.e., $P \gg N$. These high-dimensional data settings have been the motivation for the COSA approach. Often, many ordinary distance measures, e.g. the squared Euclidean distance or the Manhattan distance, cannot capture the underlying clustering structure, because they are composed from equally weighted attribute distances and there may be many attributes that are irrelevant for the clustering structure. A COSA distance is also composed from the attributes, but it allows for weighting of each attribute based on a strategy devised to capture the underlying clustering structure.

2.1.1 Attribute Distance d_{ijk}

Consider a data set, represented by $\mathbf{X} \in \mathbb{R}^{N \times P}$, a matrix of size $N \times P$. Each row $\mathbf{x}_i$ of $\mathbf{X}$ records the measurements of P attributes corresponding to an object i. Then, for each object pair $\{i, j\}$ we can construct an attribute distance for attribute k. An attribute distance is of the form

$$d_{ijk} = |x_{ik} - x_{jk}| / s_k, \tag{2.1}$$

where s_k can represent any measure of dispersion. For example, this dispersion is often the standard deviation of the values for attribute k. However, in this monograph we define s_k as the interquartile range of attribute k divided by 1.35. With this definition of s_k, the attribute distance is the the absolute difference between object i and j on a robustly standardized attribute k.

2.1.2 Attribute Weights

Suppose there is an underlying clustering structure in a high-dimensional data set $\mathbf{X}$, with $P \gg N$. For such data, it is very unlikely that all the attributes contribute to an underlying clustering structure. Objects from a cluster might be preferentially close on some attributes and far from others. Here, (squared) Euclidean or Manhattan distances most likely conceal an existing clustering structure since each attribute is equally weighted. A distance that incorporates variable weighting is much more likely to succeed in revealing the clustering structure in the data (Sebestyen, 1962; Gower, 1971; DeSarbo et al., 1984; De Soete et al., 1985; Saeys et al., 2007; Kriegel et al., 2009; Jain, 2010; Deng, et. al, 2016; McLachlan et al., 2017). We define

$$D_{ij}[\mathbf{w}] = \sum_{k=1}^{P} w_k d_{ijk}, \qquad \text{where } \sum_{k=1}^{P} w_k = 1, \text{ and } w_k \geq 0. \tag{2.2}$$

The weight vector $\mathbf{w} = \{w_k\}_{k=1}^{P} \in \mathbb{R}^P$ in equation (2.2) can capture a clustering structure when high values of the weights are assigned to attributes important for the clustering, and low or 0 values for the weights for those attributes that are not important for the clustering.

However, it could very well be true that one cluster of objects is present in one subset of attributes, while another cluster is present in another subsets of attributes. Denote by $\mathcal{G}$ the clustering structure that consists of clusters $\{\mathcal{G}_l\}_{l=1}^{M}$. Each cluster of objects, $\mathcal{G}_l$, could have its own weight vector $\mathbf{w}_l^* = \{w_{kl}^*\}_{k=1}^{P}$, where $\sum_{k=1}^{P} w_{kl}^* = 1$, and $w_{kl}^* \geq 0$. This normalized vector of attribute weights $\mathbf{w}_l^* = \{w_{kl}^*\}_{k=1}^{P}$ indicates the importance of each attributes $k \in \{1 \dots P\}$ for the clustering within cluster $\mathcal{G}_l$. Thus, these attribute weights allow for clustering of objects on their own subsets of attributes.

2.1.3 Constructing a COSA Distance

With the cluster-specific attribute weights $\mathbf{w}_l^*$, we can define a within-cluster COSA distance as

$$D_{ij}[\mathbf{w}_l^*] = \sum_{k=1}^{P} w_{kl}^* d_{ijk}, \tag{2.3}$$

for $l = 1, \dots, M$. We can also link the attribute weights of cluster $\mathcal{G}_l$ to the objects in that cluster by the following rule:

$$i \in \mathcal{G}_l \Rightarrow \mathbf{w}_i = \mathbf{w}_l^*. \tag{2.4}$$

Collecting the vectors $\mathbf{w}_i$ in the rows of the $N \times P$ matrix $\mathbf{W}$, we now have a basis for defining the i-weighted within-cluster COSA distance

$$D_{ij}[\mathbf{w}_i] = \sum_{k=1}^{P} w_{ik} d_{ijk}. \tag{2.5}$$

Within-cluster COSA distances are not allowed to violate a particular 'within-cluster' assumption. Based on the within-cluster distance definition with attribute weights of cluster $\mathcal{G}_l$, the average of the distances *within* the cluster $\mathcal{G}_l$, should be smaller than the average of the distances *between* the objects from cluster $\mathcal{G}_l$, and the objects of any other cluster $\mathcal{G}_{l'}$. Thus, for each combination of l and l', this is formalized as

$$\frac{1}{N_l^2} \sum_{i \in \mathcal{G}_l} \sum_{j \in \mathcal{G}_l} D_{ij}[\mathbf{w}_l^*] < \frac{1}{N_l N_{l'}} \sum_{i \in \mathcal{G}_l} \sum_{j \in \mathcal{G}_{l'}} D_{ij}[\mathbf{w}_l^*], \tag{2.6}$$

where $l \neq l'$, and N_l is the number of objects in cluster l.

When a pair of objects $\{i, j\} \in \mathcal{G}_l$, it follows from equation (2.4) that

$$D_{ij}[\mathbf{w}_i] = D_{ij}[\mathbf{w}_l^*] = D_{ji}[\mathbf{w}_j], \tag{2.7}$$

where

$$D_{ji}[\mathbf{w}_j] = \sum_{k=1}^{P} w_{jk} d_{jik}, \tag{2.8}$$

is the distance between object j and i, which is symmetric. However, for objects pairs that do not belong to the same cluster, the symmetry can be absent, i.e.,

$$D_{ij}[\mathbf{w}_i] \neq D_{ji}[\mathbf{w}_j]. \tag{2.9}$$

To be able to create a symmetric $N \times N$ matrix of COSA distances, we need a definition for a 'full' COSA distance that can cope with objects within the clusters, as well as objects from different clusters. Any definition is allowed, as long as it satisfies two conditions. First, the definition should assure that when objects i and j are in the same cluster, the COSA distance is equal to the within-cluster COSA distance in (2.5). Second, a COSA distance should satisfy the 'cluster happy' assumption. The cluster happy assumption states that that the average of the distances *within* a cluster $\mathcal{G}_l$, should be smaller than the average of the distances *between* the objects from that cluster $\mathcal{G}_l$, and the objects of any other cluster $\mathcal{G}_{l'}$. When we denote the COSA distance with $D_{ij}[\mathbf{W}]$, where $\mathbf{W}$ is the $N \times P$ attribute weights matrix, then the cluster happy assumption is satisfied when

$$\frac{1}{N_l^2} \sum_{i \in \mathcal{G}_l} \sum_{j \in \mathcal{G}_l} D_{ij}[\mathbf{W}] < \frac{1}{N_l N_{l'}} \sum_{i \in \mathcal{G}_l} \sum_{j \in \mathcal{G}_{l'}} D_{ij}[\mathbf{W}], \tag{2.10}$$

for all clusters $\mathcal{G}_l$. This specific inequality is an objective that drives many cluster analysis methods.

COSA distances typically satisfy the cluster happy assumption since they can be described as a 'majorizing' distance, which is any distance that is equal to a within-cluster COSA distance when objects $\{i, j\} \in \mathcal{G}_l$, and is larger otherwise. In this monograph the first example of such a symmetric COSA distance is

$$D_{ij}[\mathbf{W}] = \sum_{k=1}^{P} v_{ijk} d_{ijk}, \tag{2.11}$$

where v_{ijk} is a pairwise attribute weight that facilitates the majorizing property, here defined as

$$v_{ijk} = \max(w_{ik}, w_{jk}). \tag{2.12}$$

For the COSA distance in equation (2.11) we make the dependence on the weight matrix $\mathbf{W}$ explicit on the left-hand side, while we leave it implicit in the notation for v_{ijk} on the right-hand side.

The definition of the COSA distance (2.11) ensures that the distance within-clusters is based only on one subset of attributes, and the distance between clusters is based on both subsets of attributes that characterizes the two clusters. For each attribute k the largest attribute weight is selected from both subsets, explaining its majorizing property. Given that i and j are in the same cluster $\mathcal{G}_l$, we have

$$w_{ik} = w_{kl}^*, \text{ and } w_{jk} = w_{kl}^*, \text{ for all } k, \tag{2.13}$$

from which it follows that $D_{ij}[\mathbf{W}]$ reduces to the within-cluster COSA distance (2.5), i.e.,

$$D_{ij}[\mathbf{W}] = D_{ij}[\mathbf{w}_i]. \tag{2.14}$$

When the objects i and j belong to two different clusters l and l', respectively, we have

$$w_{ik} = w^*_{kl} \text{ and } w_{jk} = w^*_{kl'}, \text{ where } w^*_{kl} \neq w^*_{kl'}, \tag{2.15}$$

and due to the majorizing pairwise weights (2.12), the majorizing property is shown,

$$D_{ij}[\mathbf{w}_i] \leq D_{ij}[\mathbf{W}] = D_{ji}[\mathbf{W}] \geq D_{ji}[\mathbf{w}_j]. \tag{2.16}$$

While the majorizing property of $D_{ij}[\mathbf{W}]$ assures that the cluster happy assumption in (2.10) is satisfied, it also may introduce violations of triangular inequality, which can be illustrated with a simple example. Suppose we have three objects $\{h, i\} \in \mathcal{G}_l$, and $j \in \mathcal{G}_{l'}$ from two different clusters, and $P = 2$ attributes. Let the attribute distances be distances in the strict sense, and also let the following be true:

$$d_{ij1} + d_{ij2} > d_{hi1} + d_{hj1} + d_{hj2}. \tag{2.17}$$

An example to obtain the inequality in (2.17), is by ordering the objects as

$$x_{ik} < x_{hk} < x_{jk}, \tag{2.18}$$

for both attributes, that may lead to the results

$$\begin{aligned} &d_{ij1} = d_{hi1} + d_{hj1}, \text{ and} \\ &d_{ij2} > d_{hj2}. \end{aligned} \tag{2.19}$$

Now, suppose that the attribute weights of objects h and i are the opposite of the weight attribute values for the cluster of object j, i.e.,

$$\begin{aligned} &w_{h1} = w_{i1} = 1, \text{ and } w_{j1} = 0 \\ &w_{h2} = w_{i2} = 0, \text{ and } w_{j2} = 1 \end{aligned} \tag{2.20}$$

Then, it follows that

$$D_{ij}[\mathbf{W}] > D_{hi}[\mathbf{W}] + D_{hj}[\mathbf{W}], \tag{2.21}$$

a violation of triangle inequality.

2.2 Towards a criterion for COSA-KNN

To obtain the 'majorizing' COSA distances we need to know the values of the attribute weights $\mathbf{W}$, and in turn, we need the clustering structure $\mathcal{G}$. Obtaining both $\mathbf{W}$ and $\mathcal{G}$ is based on a criterion that relies on the 'cluster happy' assumption from (2.6) and (2.10). For a specific data set $\mathbf{X}$, we could obtain the solutions for $\mathbf{W}$ and $\mathcal{G}$ when we minimize a criterion in which the average distance of pairs of objects within a cluster is used, i.e., a criterion defined as

$$Q^{(1)}(\mathcal{G}, \mathbf{W}) = \left\{ \sum_{l=1}^{M} \frac{1}{N_l^2} \sum_{i \in \mathcal{G}_l} \sum_{j \in \mathcal{G}_l} D_{ij}[\mathbf{W}] \right\}. \tag{2.22}$$

The joint solution for $\mathbf{W}$ and $\mathcal{G}$ would be

$$\left\{\widehat{\mathcal{G}}, \widehat{\mathbf{W}}\right\} = \underset{\{\mathbf{W},\mathcal{G}\}}{\operatorname{argmin}} Q^{(1)}(\mathbf{W}, \mathcal{G}). \tag{2.23}$$

For the solution in equation (2.23) to be well-defined, we need restrictions on the attribute weights and assumptions for the clustering structure $\mathcal{G}$. In accordance with Friedman and Meulman (2004) we assume mutually exclusive clusters and that each object is assigned to a cluster, which follows from (2.4). Moreover, to be able to obtain the attribute weights for each cluster, we need to know how many clusters there are. In many clustering procedures the number of clusters has to be specified beforehand. We consider this as a major weakness, since often the number of clusters is unknown. Therefore, COSA proposes a procedure that avoids the prespecification of the number of clusters.

2.2.1 A K Nearest Neighbors (KNN) Circumvention

Instead of prespecifying the number of clusters M, Friedman and Meulman (2004) propose a strategy that involves a K nearest neighbors (KNN) method. This method is best explained as follows. For each object i we construct a set of its K closest objects, denoted as the neighborhood KNN(i). The measure of closeness is based on the COSA distance $D_{ij}[\mathbf{W}]$. Hence,

$$\text{KNN}(i) = \{j \mid D_{ij}[\mathbf{W}] \leq d_i(K)\}, \tag{2.24}$$

where $d_i(K)$ is the Kth order statistic of $\{D_{ij}[\mathbf{W}]\}_{j=1}^{N}$ sorted in ascending values.

Suppose we have the COSA distances for a data set $\mathbf{X}$ and the 'true' attribute weights $\mathbf{W}$, and we choose a value for K that is smaller than N. Then, among the K objects that are most similar to i, there will be an overrepresentation of objects that belong to the same cluster as compared to all the N objects. Thus,

$$\frac{1}{K} \sum_{j \in \text{KNN}(i)} I\{j \in \mathcal{G}_l\} > \frac{1}{N} \sum_{j=1}^{N} I\{j \in \mathcal{G}_l\}, \qquad \text{given } i \in \mathcal{G}_l, \tag{2.25}$$

in which I is the indicator function that returns 1 if its argument is true, and 0 if false. The more pronounced the clustering, the stronger this inequality becomes. Therefore, to the extent that inequality (2.25) holds, the average distance within KNN(i) reflects the average distance within the cluster to which object i belongs ($i \in \mathcal{G}_l$). This is expressed as

$$\frac{1}{N_l^2} \sum_{i \in \mathcal{G}_l} \sum_{j \in \mathcal{G}_l} D_{ij}[\mathbf{W}] \simeq \frac{1}{K^2} \sum_{j \in \text{KNN}(i)} \sum_{j' \in \text{KNN}(i)} D_{jj'}[\mathbf{w}_i], \tag{2.26}$$

where, by extension of (2.5),

$$D_{jj'}[\mathbf{w}_i] = \sum_{k=1}^{P} w_{ik} d_{jj'k}. \tag{2.27}$$

Thus, with KNN we can obtain an approximation of the criterion proposed in (2.22). The result is

$$Q^{(2)}(\mathbf{W}) = \left\{ \sum_{i=1}^{N} \frac{1}{K^2} \sum_{j \in \mathrm{KNN}(i)} \sum_{j' \in \mathrm{KNN}(i)} D_{jj'}[\mathbf{w}_i] \right\}. \tag{2.28}$$

Given any matrix of attribute weights, we can find the neighborhoods of each object i from (2.24). However, to obtain a good estimate of the attribute weights, in turn, we need to know the clustering structure or an approximation, e.g., equation (2.26).

Let $\mathbf{C}$ be the $N \times N$ indicator matrix in which the k nearest neighbors for each object i ($1 \leq i \leq N$) are collected. Each element in $\mathbf{C}$, denoted by c_{ij}, has value 1 when $j \in \mathrm{KNN}(i)$ and 0 otherwise. Thus,

$$K = \sum_{j=1}^{N} c_{ij}. \tag{2.29}$$

Then, with the use of $\mathbf{C}$ we can now express our criterion in a more transparent way. The result is

$$Q^{(3)}(\mathbf{C}, \mathbf{W}) = \left\{ \sum_{i=1}^{N} \frac{1}{K^2} \sum_{j=1}^{N} \sum_{j'=1}^{N} c_{ij} c_{ij'} D_{jj'}[\mathbf{w}_i] \right\}. \tag{2.30}$$

2.2.2 A Further Approximation with KNN

We have proposed to approximate the average distance within a cluster $\mathcal{G}_l$ by the average distance within the neighborhood for an object that belongs to the same cluster. In turn, Friedman and Meulman (2004, p. 825) propose, in the interest of reduced computation, to approximate the average distance within the neighborhood of each object by

$$\frac{2}{K} \sum_{j=1}^{N} c_{ij} D_{ij}[\mathbf{w}_i] \simeq \frac{1}{K^2} \sum_{j=1}^{N} \sum_{j'=1}^{N} c_{ij} c_{ij'} D_{jj'}[\mathbf{w}_i]. \tag{2.31}$$

Thus, within the neighborhood of object i, we now only sum over the distances that involve object i. This average distance approximation is based on what we will refer to as the *weak Huygens' principle* (Bavaud, 2002), implying that the sum of our distances on the right-hand side in (2.31) can be expressed as

$$D_{jj'}[\mathbf{w}_i] = \sum_{k=1}^{P} w_{ik} d_{jj'k} = \sum_{k=1}^{P} w_{ik} \frac{|x_{jk} - x_{j'k}|}{s_k}, \tag{2.32}$$

which can be interpreted as a specific minimized measure of statistical dispersion, i.e.,

$$\frac{1}{K^2} \sum_{j=1}^{N} \sum_{j'=1}^{N} c_{ij} c_{ij'} \sum_{k=1}^{P} w_{ik} \frac{|x_{jk} - x_{j'k}|}{s_k} = \frac{2}{K} \sum_{k=1}^{P} \min_{y_{ik}} \sum_{j=1}^{N} c_{ij} w_{ik} \frac{|x_{jk} - y_{ik}|}{s_k}, \tag{2.33}$$

where the corresponding minimizing central tendency measure is denoted as

$$y_{ik}^{*} = \underset{y_{ik}}{\operatorname{argmin}} \frac{1}{K} \sum_{j=1}^{N} c_{ij} w_{ik} \frac{|x_{jk} - y_{ik}|}{s_k}. \tag{2.34}$$

Hence, our average distance within the neighborhood of object i can also be interpreted as a statistical dispersion measure. It is the sum of the absolute differences of an object on attribute k with the median of attribute k.

Instead of taking the median of each attribute k, Friedman and Meulman (2004) propose to use the attribute values of object i. The idea is that $x_{ik} \simeq y_{ik}^{*}$ for all i, k such that

$$\frac{2}{K} \sum_{j=1}^{N} c_{ij} D_{ij}[\mathbf{w}_i] \simeq \frac{2}{K} \sum_{j=1}^{N} c_{ij} \sum_{k=1}^{P} w_{ik} \frac{|x_{jk} - y_{ik}^{*}|}{s_k}. \tag{2.35}$$

While the criterion from equation (2.30) involves all pairs of neighbors within the neighborhood of each object i, we now only use those distances that involve object i. Thus, we can obtain an approximation of the optimal attribute weights and neighborhoods by minimizing the criterion

$$Q^{(4)}(\mathbf{C}, \mathbf{W}) = \left\{ \sum_{i=1}^{N} \frac{1}{K} \sum_{j=1}^{N} c_{ij} D_{ij}[\mathbf{w}_i] \right\}, \tag{2.36}$$

for $\mathbf{C}$ and $\mathbf{W}$. Last, note that changing $\frac{2}{K}$ to $\frac{1}{K}$ as a constant does not affect the solution for the optimal attribute weights.

2.2.3 A Restriction on the Attribute Weights

The solution values for the attribute weights that minimize the criterion in (2.36) is an undesired solution. For each neighborhood we would obtain an attribute weight of $w_{ik} = 1$ for attribute k that has the smallest sum of within-neighborhood distances, and for all other attributes $k' \neq k$ an attribute weight of $w_{k'l}^{*} = 0$. Note that such an undesired solution also produces the maximal possible dispersion of the attribute weight values for each object i, because the attribute weights have the restrictions

$$w_{ik} \geq 0 \text{ and } \sum_{k=1}^{P} w_{ik} = 1. \tag{2.37}$$

COSA avoids clustering on only one attribute by restricting the sum of the dispersions of the attribute weights for each object i. In particular, the measure for the dispersion that Friedman and Meulman (2004) introduce is the negative entropy of the attribute weights for object i, denoted as

$$e(\mathbf{w}_i) = \sum_{k=1}^{P} w_{ik} \log(w_{ik}). \tag{2.38}$$

This negative entropy reaches its minimum when all attribute weights are equal for an object i, and it has an asymptotic maximum of 0. The negative entropy reaches its maximum when $w_{ik} \to 1$ and, hence, all other attributes $k' \neq k$ have $w_{ik'} \to 0$. Note, the negative entropy is not defined for attribute weights equal to 0, which leads to a new restriction of the attribute weights defined as

$$\sum_{k=1}^{P} w_{ik} = 1, \text{ where } 0 < w_{ik} < 1. \tag{2.39}$$

Let h be a pre-specified upper-bound on the sum of the negative entropies of the attribute weight values, and assign it a value lower than 0. Then, the criterion can be expressed as

$$\begin{aligned} Q^{(5)}(\mathbf{C}, \mathbf{W}) \quad = \quad & \left\{ \sum_{i=1}^{N} \frac{1}{K} \sum_{j=1}^{N} c_{ij} D_{ij}[\mathbf{w}_i] \right\}, \\ & \text{subject to } \sum_{i=1}^{N} e(\mathbf{w}_i) \leq h. \end{aligned} \tag{2.40}$$

Friedman and Meulman (2004) express this criterion into its unconstrained Lagrangian form, i.e.,

$$Q^{(5)}(\mathbf{C}, \mathbf{W}) = \sum_{i=1}^{N} \left\{ \frac{1}{K} \sum_{j=1}^{N} c_{ij} D_{ij}[\mathbf{w}_i] + \lambda \left(e(\mathbf{w}_i) + \log(P) \right) \right\}, \tag{2.41}$$

where λ can be interpreted as Lagrange multiplier associated with the negative entropy constraint in (2.40). Note that we have added the natural logarithm of P to avoid the criterion from obtaining a negative value (that is useful for transformations of the criterion function, as we will see in later chapters of this monograph).

In COSA we do not set a value for h in equation (2.40), but set a value for λ (> 0) in equation (2.41). The higher the value of λ, the more the attribute weights are forced to be equal to each other, and the lower the variance of each average distance within each cluster, but the higher the bias of each average distance within each cluster. Thus, λ can be regarded as a penalty parameter controlling the bias-variance trade-off, by analogy with more general density estimation procedures.

2.2.4 A Final Criterion and Towards an Algorithm

With criterion $Q^{(5)}$ from equation (2.41) we have reached the endpoint of our series of approximate criteria. We will refer to criterion $Q^{(5)}$ as the COSA-KNN criterion $Q(\mathbf{C}, \mathbf{W})$ and drop the superscript numbering from now on. Thus, the COSA-KNN criterion is defined as

$$Q(\mathbf{C}, \mathbf{W}) = \sum_{i=1}^{N} \left\{ \frac{1}{K} \sum_{j=1}^{N} c_{ij} D_{ij}[\mathbf{w}_i] + \lambda \left(e(\mathbf{w}_i) + \log(P) \right) \right\}. \tag{2.42}$$

To obtain the 'cluster happy' distances, $D_{ij}[\mathbf{W}]$ in (2.11), we minimize $Q(\mathbf{C}, \mathbf{W})$ with respect to $\mathbf{C}$ and $\mathbf{W}$ jointly. Although this is a well-defined minimization problem, there is no straightforward solution because of its combinatorial nature. Therefore, it is common to find the optimal attribute weights that minimize the criterion (2.42). First, we set values for λ, and K, and initialize the attribute weights $\mathbf{W}$. Then, given initial values of the attribute weights $\mathbf{W}$, we can calculate our first distance matrix from which we obtain the sets of nearest neighbors for each object i (see equation 2.24), collected in our first estimate of $\mathbf{C}$. In this step we minimize the COSA-KNN criterion for $\mathbf{C}$, by keeping $\mathbf{W}$ fixed. We denote this conditional criterion as $Q(\mathbf{C} \,|\, \mathbf{W})$. Next, the conditional COSA-KNN criterion $Q(\mathbf{W} \,|\, \mathbf{C})$ is minimized for $\mathbf{W}$, while keeping the first solution for $\mathbf{C}$ fixed (see Section 2.3). These new estimates of the optimal attribute weights, in turn, are used to calculate the new distances, and so on. This iterative procedure is continued until the attribute weights stabilize.

When formulated into pseudo-code, the above described procedure can be displayed as the algorithm:

```
0 :  Set λ; K
1 :  Initialize W
2 :  loop {
3 :     Ĉ ← argmin_C Q(C | W)
4 :     Ŵ ← argmin_W Q(W | C)
5 :  } until W stabilizes
6 :  Output: D_ij[W].
```

Thus, the algorithm alternates between finding the 'neighborhoods given the attribute weights', denoted as $\operatorname{argmin}_{\mathbf{C}} Q(\mathbf{C} \,|\, \mathbf{W})$, and finding 'the attribute weights given the neighborhoods, denoted as $\min_{\mathbf{W}} Q(\mathbf{W} \,|\, \mathbf{C})$.

2.3 Weights given the Neighborhoods

In this section we describe how we can find the attribute weights $\mathbf{W}$ given that the optimal neighborhoods denoted in $\mathbf{C}$ are known, denoted as

$$\widehat{\mathbf{W}} = \operatorname*{argmin}_{\mathbf{W}} Q(\mathbf{W} \,|\, \mathbf{C}).$$

2.3.1 A Closed Form Solution for W

Although a closed-form solution for the minimizing attribute weights does not exist for criterion $Q(\mathbf{C}, \mathbf{W})$ (2.42), a closed-form solution does exist for criterion $Q(\mathbf{W} \,|\, \mathbf{C})$. This solution is

$$\widehat{w}_{ik} = \exp\left(-\frac{\sum_{j=1}^{N} c_{ij} d_{ijk}}{K\lambda}\right) \Bigg/ \sum_{k'=1}^{P} \exp\left(-\frac{\sum_{j=1}^{N} c_{ij} d_{ijk'}}{K\lambda}\right). \tag{2.43}$$

In (2.43), the average attribute distance within a neighborhood is first compared with the penalty parameter λ, and then taken as the exponent of the natural number e. The denominator ensures that the attribute weights are normalized such that they sum up to 1. For the deriviations that result into this solution (2.43) we refer to the Appendix in Section 2.7.1.

When we interpret the average distance for attribute k as a measure of statistical dispersion of attribute k for the neighborhood of object i, denoted as

$$S_{ik} = \frac{\sum_{j=1}^{N} c_{ij} d_{ijk}}{K}, \tag{2.44}$$

then the solution for the weights can be expressed as

$$\widehat{w}_{ik} = \exp\left(-\frac{S_{ik}}{\lambda}\right) \Bigg/ \sum_{k'=1}^{P} \exp\left(-\frac{S_{ik'}}{\lambda}\right). \tag{2.45}$$

This notation involving S_{ik} shows that λ can be seen as a scale parameter. Those attributes that have a low dispersion in the neighborhood of object i, as compared to λ, receive a high attribute weight, and vice-versa. Note the consistency with the description of the effect of λ in Section 2.2.3. When $\lambda \to \infty$, then the attribute weights will become equal to one another. A value for $\lambda \to 0$ results in clustering on one attribute only. The definition in (2.44) also discloses its role in the bias-variance trade-off: as the value of λ is reduced, fewer objects have influence on the estimated weights, thereby increasing the variance of the attribute weight estimates and reducing the power of the algorithm.

2.3.2 An Additional Interpretation of W

Here, the optimal attribute weight, $\widehat{w}_{ik}$ can also be interpreted as weight for S_{ik} in a *softmax* function. The *softmax* function is a generalization of the sigmoïd function that can be seen as an 'activation' function that frequently occurs in neural network models, multinomial logistic models, or kernel methods (Bishop & Christopher, 2006, p. 198; Hastie, Tibshirani, & Friedman, 2009, p. 393). Its name can be explained from the fact that the softmax weighted sum over k for negative signed S_{ik} results into a λ-smoothed version of the maximum defined as

$$-S_{ik_i^*} = \max\left\{-S_{ik}\right\}_i^N, \tag{2.46}$$

from which follows that k_i^* is the attribute for which the dispersion is the lowest in the neighborhood of object i. Thus,

$$S_{ik_i^*} = \min\left\{S_{ik}\right\}_i^N. \tag{2.47}$$

When $\lambda \to 0$ the softmax function, or the optimal attribute weights, direct us to the attribute k_i^* with minimal dispersion in the neighborhood for object i, denoted as

$$\lim_{\lambda \to 0} \sum_{k=1}^{P} \widehat{w}_{ik} S_{ik} = S_{ik_i^*}. \tag{2.48}$$

On the other hand, when $\lambda \to \infty$, all weights become uniform, and we obtain the arithmetic mean over k of each S_{ik}. Thus, the optimal attribute weights from (2.43) are λ-smoothers of the attribute distances. The optimal distance for $D_{ij}[\mathbf{w}_i]$ is smoothed away from the attribute distance $d_{ijk_i^*}$ for which the dispersion in the neighborhood of object i is smallest, towards the arithmetic mean of attribute distances of object i and j.

2.4 Finding Neighborhoods while W is Fixed

In the previous section we have shown that given the neighborhoods, there is a closed form solution for the optimal attribute weights. It would seem that with the result from equation (2.43) we can find the true minimum of $Q(\mathbf{C}, \mathbf{W})$ by a 'simple' complete enumeration search over all possible combinations of the N neighborhoods. However, when using a complete and 'naive' search, the number of combinations for $\mathbf{C}$ that need to be evaluated is

$$|\mathcal{C}| = \binom{N-1}{K}^N, \tag{2.49}$$

a number too large for most N and K. For example, with $N = 30$ and $K = 5$, there are already 1.73603e+152 possibilities. Thus, this would be an impractical exhaustive search, which explains the need for the employment of an iterative algorithm, of which a heuristic was given in Section 2.2.4.

Given a preliminary set of attribute weights $\mathbf{W}$, we can derive a distance matrix. These distances, in turn, are the basis with which we find the K nearest neighbors for each object i (2.24). With this K nearest neighbors method we minimize the conditional COSA-KNN criterion for $\mathbf{C}$ given the attribute weights $\mathbf{W}$, i.e.,

$$\widehat{\mathbf{C}} = \underset{\mathbf{C}}{\operatorname{argmin}}\, Q(\mathbf{C} \,|\, \mathbf{W}). \tag{2.50}$$

When we set all attribute weights $\mathbf{W}$ equal to $\{1/P\}$ at the initialization step, the corresponding $\mathbf{C}$ is probably too far away from its actual solution. Because of the numerous distinctly suboptimal local solutions in $Q(\mathbf{C}, \mathbf{W})$, it may be an overoptimistc idea that we converge to a global optimal solution for $\mathbf{W}$ by alternating minimization of $Q(\mathbf{C} \,|\, \mathbf{W})$ and $Q(\mathbf{W} \,|\, \mathbf{C})$. To avoid convergence towards suboptimal local solutions, Friedman and Meulman (2004) propose to combine the K nearest neighbors search strategy with another type of distance function: one that renders the initialization of the attribute weight values to $\mathbf{W} = \{1/P\}$, to be a better starting-point.

2.4.1 Finding C, by Replacing $D_{ij}[\mathbf{w}_i]$ with $D_{ij}[\mathbf{W}]$

Since the attribute weights $\mathbf{W}$ are given, the solution for $\mathbf{C}$ remains the same when we do not take into account the negative entropy restriction on the attribute weights,

i.e.,

$$\begin{aligned}\widehat{\mathbf{C}} &= \underset{\mathbf{C}}{\operatorname{argmin}}\, Q(\mathbf{C}\,|\,\mathbf{W}) \\ &= \underset{\mathbf{C}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \frac{1}{K} \sum_{j=1}^{N} c_{ij} D_{ij}[\mathbf{w}_i] + \lambda \sum_{i=1}^{N} e(\mathbf{w}_i) \right\} \\ &= \underset{\mathbf{C}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{c_{ij}}{K} D_{ij}[\mathbf{w}_i] \right\}. \end{aligned} \tag{2.51}$$

In (2.51) we see that $c_{ij} = 1$ for those objects j that belong to the K nearest neighbors of i based on $D_{ij}[\mathbf{w}_i]$ (2.27), and $c_{ij} = 0$ otherwise. Note however, in (2.24) we perform the K nearest neighbors method on $D_{ij}[\mathbf{W}]$ (instead of $D_{ij}[\mathbf{w}_i]$). In the first iteration of the algorithm it would not lead to any different results for $\widehat{\mathbf{C}}$, since we initialize each attribute weight as $w_{ik} = \{1/P\}$, leading to $D_{ij}[\mathbf{W}] = D_{ij}[\mathbf{w}_i]$. However, in the next iterations, the optimal attribute weight vectors will start to be different for each object, i.e., $\mathbf{w}_i \neq \mathbf{w}_j$, and hence produce different results for $\widehat{\mathbf{C}}$.

If we would work with K nearest neighbors on $D_{ij}[\mathbf{w}_i]$ for each object i, then the estimated neighborhoods will converge very fast into a suboptimal local minimum of $Q(\mathbf{C}\,|\,\mathbf{W})$ for $\mathbf{C}$. The reason originates from the asymmetric property

$$D_{ij}[\mathbf{w}_i] = \sum_{k=1}^{P} w_{ik} d_{ijk} \neq \sum_{k=1}^{P} w_{jk} d_{jik} = D_{ji}[\mathbf{w}_j], \qquad \text{where } i \neq j, \tag{2.52}$$

allowing for a situation in which object j ends up in the neighborhood of i because $D_{ij}[\mathbf{w}_i]$ is small, even though $D_{ji}[\mathbf{w}_j]$ may be very large. Vice versa, object i could not end up in the neighborhood of object j because $D_{ji}[\mathbf{w}_j]$ is too large, although $D_{ij}[\mathbf{w}_i]$ is small. Thus, this asymmetric property forces the algorithm to stick to the first estimate of the neighborhoods in $\mathbf{C}$, even though these neighborhoods were calculated based on a suboptimal estimate of the attribute weights $\mathbf{W}$.

Friedman and Meulman (2004) propose to perform KNN on $D_{ij}[\mathbf{W}]$ instead of $D_{ij}[\mathbf{w}_i]$, i.e.,

$$\widehat{\mathbf{C}} \simeq \underset{\mathbf{C}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{c_{ij}}{K} D_{ij}[\mathbf{W}] \right\}, \tag{2.53}$$

subject to the restriction

$$\sum_{j=1}^{P} c_{ij} = K, \qquad \text{for each } i.$$

We will review this strategy in the next section. At this point, we explain that estimating $\mathbf{C}$ from the K nearest neighbors on $D_{ij}[\mathbf{W}]$ instead of $D_{ij}[\mathbf{w}_i]$ is a first search

strategy with which the algorithm becomes less prone to get stuck in suboptimal local minima. When performing K nearest neighbors on the distance $D_{ij}[\mathbf{W}]$ the algorithm also takes into account the estimates of the 'between-cluster' distances, which nudges the estimates of the neighborhoods, such that it is easier to break out of local minima for $\mathbf{C}$ in $Q(\mathbf{C} \mid \mathbf{W})$. Since $D_{ij}[\mathbf{W}]$ is symmetric, and

$$D_{ij}[\mathbf{W}] \geq \max\left(D_{ij}[\mathbf{w}_i], D_{ji}[\mathbf{w}_j]\right), \tag{2.54}$$

it is much more likely that when object i is in the neighborhood of j, vice versa also holds. Thus, by replacing $D_{ij}[\mathbf{w}_i]$ with $D_{ij}[\mathbf{W}]$ to find the neighborhoods, we now wish to minimize the approximate criterion

$$\tilde{Q}(\mathbf{C} \mid \mathbf{W}) = \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{c_{ij}}{K} \sum_{k=1}^{P} v_{ijk} d_{ijk}, \tag{2.55}$$

for $\mathbf{C}$ given $\mathbf{W}$. Here the pairwise attribute weights v_{ijk} for object i and j are defined as before in (2.12).

Although we now have an approximate criterion $\tilde{Q}(\mathbf{C} \mid \mathbf{W})$ that is less prone to end up in local minima compared to $Q(\mathbf{C} \mid \mathbf{W})$, the algorithm will still (most likely) not succeed in finding a good final solution for $\mathbf{C}$, due to the initialization of $w_{ik} = 1/P$ for each object i and attribute k. With $P \gg N$ and many attributes non-relevant for the clustering, our initialization for $\mathbf{W}$ will be very far from the optimal solution for the attribute weights in the highly non-convex COSA-KNN criterion $Q(\mathbf{C}, \mathbf{W})$ (2.42). This poses a problem for the approximate criterion $\tilde{Q}(\mathbf{C} \mid \mathbf{W})$ since it is strongly influenced by the d_{ijk} that have large values. The large values for the attribute distances will mask the underlying clustering structure, and hence, complicates the algorithmic process of finding the actual optimal neighborhoods in $\mathbf{C}$, especially for attribute weight values far from the global optimal solution.

2.4.2 Inverse Exponential Distance

Friedman and Meulman (2004) propose to perform KNN on a distance that is based on a weighted inverse exponential mean function in which the influence of the larger attribute distances is diminished compared to the smaller attribute distances. We refer to this function as the *inverse exponential distance*, and it is defined as

$$D_{ij}^{(\eta)}[\mathbf{W}] = -\eta v_{ij\cdot} \log\left\{ v_{ij\cdot}^{-1} \sum_{k=1}^{P} v_{ijk} \exp\left(-\frac{d_{ijk}}{\eta}\right)\right\}, \tag{2.56}$$

where

$$v_{ij\cdot} = \sum_{k=1}^{P} v_{ijk}, \tag{2.57}$$

the sum of the pairwise attribute weights over their attributes, So, the inverse exponential distance in (2.56) involves the natural logarithm of a weighted average of the

inverse η-exponentially transformed attribute distances. Since (2.56) involves the majorizing pairwise attribute weights (v_{ijk} from (2.12)), it can also violate the triangular inequality assumption.

For now, interpret η as a rate parameter of the inverse exponential function of the attribute distance, with a pre-set value larger than 0, $\eta > 0$. The inverse exponential function over the attribute distance outputs an attribute similarity, defined as $\exp\left(-\frac{d_{ijk}}{\eta}\right)$. The smaller η the faster the inverse exponential function approaches 0 for larger values of d_{ijk}, as is shown in Figure 2.1a. Thus, the influence of large d_{ijk} becomes η-exponentially less important in the weighted sum over the attributes. By taking the logarithm over the sum of the attribute similarities and multiplying this logarithm with $-\eta$, it becomes the weighted inverse exponential mean of the attribute distances for object pair i and j. Note that the larger attribute distancs have a much smaller role in $D_{ij}^{(\eta)}[\mathbf{W}]$ (2.56), compared to the COSA distance $D_{ij}[\mathbf{W}]$ (2.11). This can be seen in Figure 2.1 for values of η lower than ∞, and larger than 0.

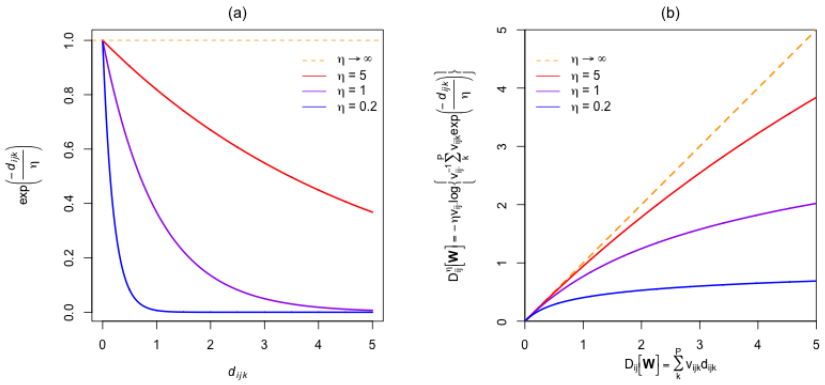

Figure 2.1: The left panel (a) *depicts the inverse exponential function on the attribute distance d_{ijk}, transforming it into a proximity where the role of a larger value for d_{ijk} becomes less important, controlled by a rate/scale parameter η. The right panel* (b) *shows the transformation of inverse exponential distance $D_{ij}^{(\eta)}[\mathbf{W}]$ (vertical axis) into $D_{ij}[\mathbf{W}]$ (horizontal axis) when $\eta \to \infty$ and each $v_{ijk} = 1/P$.*

2.4.3 Homotopy between $D_{ij}^{(\eta)}[\mathbf{W}]$ and $D_{ij}[\mathbf{W}]$

Although the inverse exponential distance may seem very different from the distance $D_{ij}[\mathbf{W}]$ (2.11), we see in Figure 2.1 that both distances become asymptotically equal for $\eta \to \infty$, i.e.,

$$\lim_{\eta \to \infty} D_{ij}^{(\eta)}[\mathbf{W}] = D_{ij}[\mathbf{W}]. \tag{2.58}$$

Thus, there is a homotopy between $D_{ij}[\mathbf{W}]$ and $D_{ij}^{(\eta)}[\mathbf{W}]$. $D_{ij}^{(\eta)}[\mathbf{W}]$ gradually transforms into $D_{ij}[\mathbf{W}]$, when the homotopy parameter η approaches ∞. For a direct derivation of (2.58) (see the Appendix in Section 2.7.2).

A simpler way to explain equation (2.58), is to re-express the weighted inverse exponential distance as a minimization problem (*cf.* Friedman and Meulman, 2004). To do so, we first introduce a general type of pairwise attribute weights denoted by t_{ijk}, collected in the vector $\mathbf{t_{ij}}$. When we assume

$$\sum_{k=1}^{P} t_{ijk} = v_{ij\bullet} \text{ and } t_{ijk} > 0,$$

then it can be shown that the inverse exponential distance for COSA-KNN, $D_{ij}^{(\eta)}[\mathbf{W}]$ in (2.56), is the result of the minimization problem

$$\min_{\mathbf{t}_{ij}} \left\{ \sum_{k=1}^{P} t_{ijk} d_{ijk} + \eta \sum_{k=1}^{P} t_{ijk} \log\left(\frac{t_{ijk}}{v_{ijk}}\right) \right\}, \tag{2.59}$$

which is explained in more detail in the Appendix in section 2.7.3.

The minimization problem in equation (2.59) discloses a role for η as a penalty parameter. In the minimization over $\mathbf{t}_{ij}$, the parameter η regulates the influence of the *Kullback-Leibler divergence* (Kullback & Leibler, 1951), defined as

$$D_{KL}(\mathbf{t}_{ij} \| \mathbf{v}_{ij}) = \sum_{k=1}^{P} t_{ijk} \log\left(\frac{t_{ijk}}{v_{ijk}}\right). \tag{2.60}$$

In particular, η regulates how much t_{ijk} is allowed to diverge from v_{ijk} in (2.59), while minimizing $\sum_{k=1}^{P} t_{ijk} d_{ijk}$. Let $\widehat{t}_{ijk}$ denote the solution for t_{ijk} in (2.59). Then, when $\eta = 0$, the Kullback-Leibler divergence term drops out of the minimization problem, resulting in a maximum value for t_{ijk} that corresponds to attribute k of the smallest attribute distance, and a zero weight for the remaining attribute distances, $\widehat{t}_{ijk} = 0$. When $v_{ijk} = 1/P$ and η is small, then the smaller attribute distances will receive higher values for their corresponding $\widehat{t}_{ijk}$'s, which explains why the influence of the larger attribute distances in $D_{ij}^{(\eta)}[\mathbf{W}]$ is mitigated.

When $\eta \to \infty$, the Kullback-Leibler divergence in (2.60) obtains an absolute role in the minimization over t_{ijk} in equation (2.59), the value of each d_{ijk} will have no role. The solution for each t_{ijk} will equal v_{ijk}, since that is the only result with which we achieve the minimum of zero for the Kullback-Leibler divergence. Thus, with $\eta \to \infty$

we have $D_{ij}^{(\eta)}[\mathbf{W}] \to D_{ij}[\mathbf{W}]$, shown in the following series of steps:

$$\begin{aligned}
\lim_{\eta\to\infty} D_{ij}^{(\eta)}[\mathbf{W}] &= \lim_{\eta\to\infty} -\eta v_{ij\bullet} \log\left\{ v_{ij\bullet}^{-1} \sum_{k=1}^{P} v_{ijk} \exp\left(-\frac{d_{ijk}}{\eta}\right)\right\} \\
&= \lim_{\eta\to\infty} \min_{\mathbf{t}_{ij}} \left\{ \sum_{k=1}^{P} t_{ijk} d_{ijk} + \eta \sum_{k=1}^{P} t_{ijk} \log\left(\frac{t_{ijk}}{v_{ijk}}\right)\right\} \\
&= \lim_{\eta\to\infty} \sum_{k=1}^{P} v_{ijk} d_{ijk} + \eta \sum_{k=1}^{P} v_{ijk} \log\left(\frac{v_{ijk}}{v_{ijk}}\right) \\
&= \sum_{k=1}^{P} v_{ijk} d_{ijk} \\
&= D_{ij}[\mathbf{W}]. \qquad (2.61)
\end{aligned}$$

The above equations explain the right panel (b) in Figure 2.1. It shows that the inverse exponential distance $D_{ij}^{(\eta)}[\mathbf{W}]$ can be seen as an homotopic approximation of $D_{ij}[\mathbf{W}]$.

2.4.4 Homotopy between $\tilde{Q}^{(\eta)}(\mathbf{C}\,|\,\mathbf{W})$ and $\tilde{Q}(\mathbf{C}\,|\,\mathbf{W})$

In Section 2.4.1, we obtained a solution for $\mathbf{C}$, by minimizing the approximate criterion $\tilde{Q}(\mathbf{C}\,|\,\mathbf{W})$ in (2.55), based on $D_{ij}[\mathbf{W}]$. Similarly, we can obtain a homotopic approximate solution for $\mathbf{C}$ by performing KNN on the homotopic distance $D_{ij}^{(\eta)}[\mathbf{W}]$ (2.59). Such an homotopic approximate solution for $\mathbf{C}$ minimizes the homotopic approximate criterion defined as

$$\begin{aligned}
\tilde{Q}^{(\eta)}(\mathbf{C}\,|\,\mathbf{W}) &= \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{c_{ij}}{K} D_{ij}^{(\eta)}[\mathbf{W}] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{N} -\eta \frac{c_{ij} v_{ij\bullet}}{K} \log\left\{ v_{ij\bullet}^{-1} \sum_{k=1}^{P} v_{ijk} \exp\left(-\frac{d_{ijk}}{\eta}\right)\right\}. \qquad (2.62)
\end{aligned}$$

Working with $\tilde{Q}(\mathbf{C}\,|\,\mathbf{W})$ (2.55), and equal attribute weights in $\mathbf{W}$, introduces the problem where an underlying clustering structure cannot be recovered due to large values for d_{ijk}. Large values for d_{ijk}/η, will have less influence on the criterion $\tilde{Q}^{(\eta)}(\mathbf{C}\,|\,\mathbf{W})$ compared to $\tilde{Q}(\mathbf{C}\,|\,\mathbf{W})$. Thus, by using $\tilde{Q}^{(\eta)}(\mathbf{C}\,|\,\mathbf{W})$, we can avoid those suboptimal local minima in which we could have got stuck due to too large attribute distances. Hence, when we set η to be (not too) small and minimize $\tilde{Q}^{(\eta)}(\mathbf{C}\,|\,\mathbf{W})$ over $\mathbf{C}$, we can use uniformly initialized attribute weights for $\mathbf{W}$, due to the stronger influence of the smaller attribute distances.

Using the criterion $\tilde{Q}^{(\eta)}(\mathbf{C}\,|\,\mathbf{W})$ comes with a side-effect. Since the criterion favors low attribute distances and mitigates the influence of the larger attribute distances, an opposite effect could also occur. For a too low η it could happen that for each object the wrong neighbors are selected based on some of the attribute distances that,

based on chance, were smaller than slightly larger, but relevant, attribute distances. Still, $\tilde{Q}^{(\eta)}(\mathbf{C} \mid \mathbf{W})$ is useful for COSA-KNN, it provides a good starting point for the estimation of $\mathbf{C}$ given an initialization where $v_{ijk} = 1/P$. In the following two subsections, we will give more details on the homotopy parameter η, and the homotopy strategy implemented in the algorithm for COSA-KNN.

2.4.5 Comparing η with λ

There is a strong similarity between the role of η in $\tilde{Q}^{(\eta)}(\mathbf{C} \mid \mathbf{W})$ and that of λ in the COSA-KNN criterion $Q(\mathbf{W} \mid \mathbf{C})$ in (2.42). The parameter η regulates the strength with which the Kullback-Leibler divergence term in equation (2.59) is penalized. Similarly, λ is the penalty parameter that regulates the sum of each negative entropy $e(\mathbf{w}_i)$ in (2.38). The higher λ, the more the attribute weights $\widehat{\mathbf{W}}$ are encouraged to resemble $1/P$, and the lower the negative entropy. In the same way, the higher η, the more each attribute weight $\widehat{t}_{ijk}$ is encouraged to resemble v_{ijk}, and the lower the Kullback-Leibler divergence term in $D_{ij}[\mathbf{W}]^{(\eta)}$ (2.59).

The lower λ, the less strong the encouragement for equal attribute weights in $\widehat{\mathbf{W}}$, which may lead to an asymptotic degenerative solution where only the attribute(s) with the smallest distances obtain a maximum attribute weight and all others obtain a zero solution. Similarly, when $\eta \to 0$, each $\widehat{t}_{ijk}$ moves away from v_{ijk} towards an asymptotic solution where $\widehat{t}_{ijk}$ gets the maximal weight for the smallest attribute distance, and for all other attributes the value is zero, and end up with a degenerative solution.

Because η and λ have a similar regularization effect, the optimal solutions for the attribute weights t_{ijk} and w_{ik} are similar too. The closed form solution for the optimal attribute weight w_{ik} was given in equation (2.43) as

$$\widehat{w}_{ik} = \exp\left(-\frac{\sum_{j=1}^{N} c_{ij} d_{ijk}}{K\lambda}\right) \Bigg/ \sum_{k'=1}^{P} \exp\left(-\frac{\sum_{j=1}^{N} c_{ij} d_{ijk'}}{K\lambda}\right).$$

The attribute weight $\widehat{w}_{ik}$ obtains a high value when the sum of the distances of object i with its nearest neighbors is small for that specific attribute k as compared to the other attributes. There is a closed-form solution for the optimal t_{ijk} in the minimization problem (2.59) as well (for the derivation, see Section 2.7.3). The result is

$$\widehat{t}_{ijk} = v_{ij\cdot} \frac{v_{ijk} \exp\left(-\frac{d_{ijk}}{\eta}\right)}{\sum_{k'=1}^{P} v_{ijk'} \exp\left(-\frac{d_{ijk'}}{\eta}\right)}. \tag{2.63}$$

The attribute weight $\widehat{t}_{ijk}$ is small when d_{ijk} is small (compared to the other attribute distances of object i with object j). In addition, from (2.63) we can see a homotopy

between $\widehat{t}_{ijk}$ and v_{ijk} for when that when $\eta \to \infty$, i.e.,

$$\begin{aligned}\lim_{\eta\to\infty} \widehat{t}_{ijk} &= \lim_{\frac{1}{\eta}\to 0} \widehat{t}_{ijk} \\ &= v_{ij\bullet}\frac{v_{ijk}\exp\left(-0d_{ijk}\right)}{\sum_{k'=1}^{P} v_{ijk}\exp\left(-0d_{ijk'}\right)} \\ &= v_{ij\bullet}\frac{v_{ijk}}{v_{ij\bullet}} \\ &= v_{ijk} \quad . \end{aligned} \tag{2.64}$$

Equation (2.64) is another way of demonstrating how $\widehat{t}_{ijk} \to v_{ijk}$ for $\eta \to \infty$. When $\eta \to 0$, the $\widehat{t}_{ijk}$ diverges away from v_{ijk} and towards 0 for most $\widehat{t}_{ijk}$'s, except the $\widehat{t}_{ijk'}$ for the attribute k' for which d_{ijk} is smallest. Note, the solution for each t_{ijk} is based on the attribute distances of one object pair $\{i, j\}$ only, while the solution for w_{ik} is based on the attribute distances of K object pairs. Given that $K > 1$, the solution for w_{ik} is more robust than the solution for t_{ijk}. Thus, some prudence is advised to not set η to be too small, otherwise we may favor too much the confounding attribute distances that got small by chance.

2.4.6 Homotopy Strategy that Avoids Local Minima

When we use the KNN strategy in the first iteration on $D_{ij}^{(\eta)}[\mathbf{W}]$, with $\eta = \lambda$ and $\mathbf{W} = \{1/P\}$, we obtain a good starting estimate for the solution for $\mathbf{C}$. Each $\widehat{t}_{ijk}$ is being pulled towards (or away from) $v_{ijk} = 1/P$ with a similar regularization strength as $\widehat{w}_{ik}$ is encouraged to resemble $1/P$. However, while the solution for t_{ijk} (2.63) is based on the attribute distance of one object pair only, the solution for v_{ijk} is based on K distances for object i ($\widehat{w}_{ik}$), and K distances for object j ($\widehat{w}_{jk}$).

The attribute weight $\widehat{t}_{ijk}$ is very sensitive to attribute distances that end up being small just by chance (sampling fluctuations). Still, given that in the first iteration each attribute weight w_{ik} is initialized with a value of $1/P$, it is more probable that the estimate for $\mathbf{C}$ is closer to the optimal solution when KNN is performed on $D_{ij}^{(\eta)}[\mathbf{W}]$, than when performed on $D_{ij}[\mathbf{W}]$. Given that each v_{ijk} has the starting value $1/P$, finding $\mathbf{C}$ based on $D_{ij}[\mathbf{W}]$ is more difficult due to the high influence of the large attribute distances. Hence, with this strategy in which we use $D_{ij}^{(\eta)}[\mathbf{W}]$ in the first iteration, we obtain an estimate for $\mathbf{C}$ which is closer to the optimal solution, and therefore avoids suboptimal local minima.

When, in the next iteration, our estimated solution for $\mathbf{W}$ gets closer to the (global) optimal solution in $Q(\mathbf{C}, \mathbf{W})$, then v_{ijk} is not uniform anymore. The solutions for both t_{ijk} and v_{ijk} receive high values for the smaller attribute distances and low values for the larger attribute distances; therefore the value of the Kullback-Leibler divergence, $D_{KL}(\mathbf{t}_{ij} \| \mathbf{v}_{ij})$ in (2.60), becomes smaller compared to the previous iteration. Thus, when we do *not* increase η with every new estimate for v_{ijk}, we can end up in a situation that allows for minimized t_{ijk}'s that might get (too) close to the degenerative solution where $\widehat{t}_{ijk} \to 0$.

Summarizing, when we perform the KNN method on $D_{ij}^{(\eta)}[\mathbf{W}]$ when η is too low, we might actually obtain a solution which gets us further away from the optimal solution for $\mathbf{C}$ due to attribute distances that got based on chance. With each change of our estimate of $\mathbf{W}$ in the direction of the optimal solution, we need to increase η correspondingly. However, when we increase η too much, and our newly found $\widehat{W}$ did not get 'close enough' to the optimal solution, we might again end up with an estimate for $\mathbf{C}$ that gets us further away from the optimal solution. With a too high value for η, we have the large attribute distances that may mask the underlying clustering structure. Hence, for an optimal *homotopy relaxation path* a careful consideration should be made on how η should be increased with each iteration.

Friedman and Meulman (2004) implemented a linear relaxation path of $\tilde{Q}^{(\eta)}(\mathbf{C} \mid \mathbf{W})$ to $\tilde{Q}(\mathbf{C} \mid \mathbf{W})$. At the start the η parameter is set equal to λ. Then, with each iteration η is increased with λ multiplied by a relaxation parameter α. According to Friedman and Meulman (2004) "there is as yet no theory to suggest appropriate values of α in particular applications", but based on (not-shown) empirical evidence they advised to set the relaxation to be fairly small, $\alpha \lesssim 0.1$, to avoid that the "algorithm causes to converge towards an inferior local minimum in spite of its potentially good starting point". This algorithm for COSA-KNN is called COSA algorithm 2 in Friedman and Meulman (2004, p. 825), we will refer to it as the COSA-KNN algorithm, which in pseudo-code is

COSA-KNN

0 : Set: $\lambda; K; \alpha \lesssim 0.1$
1 : Initialize: $\eta = \lambda; \mathbf{W} = \{1/P\} \in \mathbb{R}^{N \times P}$;
2 : loop {
3 : Compute distances $D_{ij}^{(\eta)}[\mathbf{W}]$ (2.56)
4 : $\mathbf{C} \leftarrow \text{KNN}\left(D_{ij}^{(\eta)}[\mathbf{W}]\right)$ (2.24)
5 : Compute attribute weights $\mathbf{W}$ (2.43)
6 : Increase $\eta : \eta + \alpha * \lambda$
7 : } until $\mathbf{W}$ stabilizes
8 : Output: $D_{ij}[\mathbf{W}]$.

2.4.7 A Critical Note on COSA-KNN (2017)

Kampert et al. (2017) describe an updated implementation of COSA-KNN, compared to Friedman and Meulman (2003), in which inner-loops have been added. The

resulting algorithm can be described as

COSA-KNN 2017
0 : Set: $\lambda; K; \alpha \lesssim 0.1$
1 : Initialize: $\eta = \lambda; \mathbf{W} = \{1/P\} \in \mathbb{R}^{N \times P}$;
2 : Outer Loop {
3 : Inner Loop {
a Compute distances $D_{ij}^{\eta}[\mathbf{W}]$
b Compute $\mathbf{C}$ from $\left\{KNN(i)_{i=1}^{N}\right\}$
c Compute weights $\mathbf{W}$
} Until convergence.
4 : Increase $\eta : \eta + \alpha * \lambda$
5 : } Until $\mathbf{W}$ stabilizes
6 : Output: $\{D_{ij}^{\eta}[\mathbf{W}],$ and $\mathbf{W}\}$

An explanation for the use of these inner and outer-loops received little attention in Kampert, Meulman, and Friedman (2017). Moreover, given the information we have described about the homotopy parameter (η) and its relaxation path, it might actually be debatable to apply such inner-loops. As is explained in Section 2.4.6, when $\widehat{\mathbf{W}}$ is updated towards the optimal solution, and η is not increased, it is likely that we move away again from the optimal solution for $\mathbf{W}$, causing the algorithm to be prone to producing an unwanted suboptimal local minimum for $Q(\mathbf{C}, \mathbf{W})$.

2.5 The Tuning of COSA-KNN

Throughout this chapter we did not yet discuss what values we set for λ and K. Friedman and Meulman (2004) advised to set the tuning parameters as $K = \sqrt{N}$, $\lambda = 0.2$, and a relaxation rate of $\alpha = 0.1$. These settings are based on empirical evidence in Friedman and Meulman (2004) and Kampert et al. (2017). However, in the same studies, it also claimed that in the presence of a more subtle structure in the data, the results can be fairly sensitive to choices for the tuning parameter values. In this section we provide a brief review of the advised 'default' values for λ and K in COSA-KNN.

2.5.1 The Tuning Parameter K

The value to which we can set the tuning parameter K, the number of objects in each neighborhood for each object i, received little attention. Friedman and Meulman (2004, p. 826) remark the following strategy:

> *"The size K that is chosen for the nearest neighborhoods is not critical and results are fairly stable over a wide range of values. It should be sufficiently large to provide stable estimates of S_{ik} (2.44) but not too much larger than the size of the cluster containing the ith object. Setting $K = \sqrt{N}$ is a*

reasonable choice, although some experimentation may be desirable after reviewing the sizes of the clusters uncovered."

Here we put a critical note that the circumvention of specifying the number of clusters by choosing the tuning parameter K, comes with a cost. For example, when we assume to find unequal cluster sizes we will not have an optimal choice for K. Furthermore, for clusters consisting of around $N/2$ objects, the algorithm becomes more prone too getting stuck in inferior local minima. On the one hand, a too small value for K, might render the estimates for w_{ik} to be unstable, but, on the other hand, with $K \to N/2$ we achieve the maximum for

$$|\mathcal{C}| = \binom{N-1}{K}^N,$$

the cardinality of the number of all combinations of possible neighborhoods (2.49), providing more inferior local minima for the algorithm to (get stuck in).

2.5.2 The Scale Parameter λ

About the scale parameter λ, it was said in Friedman and Meulman (2004, p. 824):

"Since an optimal value of λ is situation dependent and there is as yet no theory to suggest good values, the only recourse is to experiment with several values and to examine the results."

In the presence of sharp clustering on small subsets attributes, Friedman and Meulman (2004) remarked that the algorithm is not highly sensitive to values in the range $0.1 \leq \lambda \leq 0.4$. Note that a definition of a sharp clustering structure was not given. Nowadays, a sharp clustering structure can also be presented in a data set of $P = 200,000$ attributes and $N = 150$ objects. Depending on the optimal value for K, for such a data set the algorithm will most likely show different results in the range of $0.1 \leq \lambda \leq 0.4$. In general it can be noted that the larger P, the larger the solution space for the attribute weights, and the more differences there are that can be expected for the values of λ.

2.5.3 Relating K and λ

It is important to realize that there is an interplay between K and λ to be taken into account. The tuning parameters meet each other in the definition of the attribute weights (2.43). We defined $S_{ik} = K^{-1} \sum_{j=1}^{N} c_{ij} d_{ijk}$, and expressed the attribute weight as

$$\widehat{w}_{ik} = \exp\left(-\frac{S_{ik}}{\lambda}\right) \Bigg/ \sum_{k'=1}^{P} \exp\left(-\frac{S_{ik'}}{\lambda}\right).$$

Given a fixed value for λ, the higher we set K, the larger each S_{ik}, and the smaller the difference between the minimum and the maximum over k of S_{ik} $(1 \leq k \leq P)$,

which will result in more equal weights. A similar effect was described for λ when we interpreted the attribute weights w_{ik} as a softmax function for S_{ik}. Thus, the value for K and that of λ can facilitate or mitigate each other in steering towards equal attribute weights.

2.6 Discussion

COSA-KNN is a pioneering algorithm, because it was ahead of its time when it got developed. To our knowledge there is yet no other algorithm that is designed to output a cluster-happy distance matrix that can represent clustering of objects on different or overlapping subsets of attributes in high-dimensional data settings. COSA-KNN circumvents the need to specify the expected number of clusters in the data by use of a K nearest neighbors method, and by implementing a λ-regulated negative entropy restriction on the attribute weighting, COSA-KNN manages to weigh the P attributes, where $P > N$. Moreover, with a relaxation parameter α, a linear relaxation path is set on the homotopy parameter in an approximate criterion for COSA-KNN such that the algorithm elegantly avoids inferior local minima.

Since Friedman and Meulman (2004) remark that they are not aware of a theory with which we can provide optimal values for the tuning parameters K, λ, and α, their properties need more investigation. First, it might be interesting to apply the Gap statistic (Tibshirani, Walther, & Hastie, 2001) to COSA-KNN to find optimal values of the tuning parameters K and λ. In Witten and Tibshirani (2010) the Gap statistic showed promising results in finding the optimal value of a tuning parameter that is comparable to λ. Second, the constant value for α in the homotopy relaxation might also need some further development. For now, α provides a linear relaxation path for the homotopic transition of $\tilde{Q}^{(\eta)}(\mathbf{C} \,|\, \mathbf{W})$ to $\tilde{Q}(\mathbf{C} \,|\, \mathbf{W})$ during the iterations of COSA-KNN. Since the homotopy parameter is sensitive for too low and too high values of α, it might be interesting to explore opportunities where α is expressed as a function of the Kullback-Leibler divergence between the updated attribute weights and the attribute weights of the previous iteration. To conclude, when we obtain a better understanding of the properties of these tuning parameters, we also might be able to relax the restrictions for the attribute weights to $w_{ik} \geq 0$ and $\sum_{k=1}^{P} w_{ik} = 1$, i.e., allowing for zero-value attribute weights in data sets where $P \gg N$.

2.7 Appendix

2.7.1 Optimal Solution for the Attribute Weights in $Q(\mathbf{W} \mid \mathbf{C})$

Assuming that the optimal $\mathbf{C}$ is known, the COSA-KNN criterion becomes an an objective function of $\mathbf{W}$ only:

$$Q(\mathbf{W}) = \sum_{i=1}^{N} \left\{ \sum_{k=1}^{P} w_{ik} S_{ik} + \lambda \sum_{k=1}^{P} w_{ik} \log(w_{ik}) \right\}. \tag{2.65}$$

To incorporate the constraint on the weights, $\sum_{k=1}^{P} w_{kl} = 1$, we introduce the Lagrangian multiplier δ. The result is

$$Q(\mathbf{W}, \delta) = \sum_{i=1}^{N} \left\{ \sum_{k=1}^{P} w_{ik} S_{ik} + \lambda \sum_{k=1}^{P} w_{ik} \log(w_{ik}) - \delta \left(\sum_{k=1}^{P} w_{ik} - 1 \right) \right\}. \tag{2.66}$$

Minimizing this unconstrained criterion with respect to $\mathbf{W}$ is a sum of N separate convex optimization problems. Hence, we can find the optimal vlaue for attribute weight w_{ik}, of each object i and attribute k, by minimizing the sub-criterion

$$Q_i(\mathbf{w}_i, \delta) = \sum_{k=1}^{P} w_{ik} S_{ik} + \lambda \sum_{k=1}^{P} w_{ik} \log(w_{ik}) - \delta \left(\sum_{k=1}^{P} w_{ik} - 1 \right), \tag{2.67}$$

with respect to w_{ik}.

The partial derivative of $Q_i(\mathbf{w}_i, \delta)$ with respect to w_{ik} equals

$$\begin{aligned} \frac{\partial Q_i(\mathbf{w}_i, \delta)}{\partial w_{ik}} &= S_{ik} + \frac{\partial}{w_{ik}} \lambda w_{ik} \log(w_{ik}) - \delta \\ &= S_{ik} + \lambda w_{ik} \cdot \frac{\partial}{w_{ik}} [\log(w_{ik})] + \lambda \log(w_{ik}) \cdot \frac{\partial}{w_{ik}} [w_{ik}] - \delta \\ &= S_{ik} + \lambda + \lambda \log(w_{ik}) - \delta, \end{aligned} \tag{2.68}$$

and the partial derivative of $Q_i(\mathbf{w}_i, \delta)$ with respect to δ equals

$$\frac{\partial Q_i(\mathbf{w}_i, \delta)}{\partial \delta} = \sum_{k=1}^{P} w_{ik} - 1. \tag{2.69}$$

Setting (2.68) equal to zero, and solving for w_{ik}, gives us

$$\widehat{w}_{ik} = \exp\left(\frac{-S_{ik}}{\lambda} \right) \exp\left(\frac{\delta - \lambda}{\lambda} \right). \tag{2.70}$$

Substituting $\widehat{w}_{ik}$ from (2.70) with w_{ik} into $\frac{\partial Q_i(\mathbf{w}_i, \delta)}{\partial \delta}$ (2.69), we have

$$\sum_{k=1}^{P} \widehat{w}_{ik} = \exp\left(\frac{\delta - \lambda}{\lambda} \right) \sum_{k=1}^{P} \exp\left(\frac{-S_{ik}}{\lambda} \right) = 1.$$

Therefore,

$$\exp\left(\frac{\delta-\lambda}{\lambda}\right) = \frac{1}{\sum_{k=1}^{P}\exp\left(\frac{-S_{ik}}{\lambda}\right)}.$$

Inserting the latter result for $\exp\left(\frac{\delta-\lambda}{\lambda}\right)$ into (2.70), we obtain the solution for the optimal w_{ik} as formulated in (2.43):

$$\widehat{w}_{ik} = \exp\left(-\frac{S_{ik}}{\lambda}\right) \Bigg/ \sum_{k'=1}^{P}\exp\left(-\frac{S_{ik'}}{\lambda}\right),$$

where $S_{ik} = \frac{\sum_{j=1}^{N} c_{ij}d_{ijk}}{K}$.

2.7.2 Homotopy between $D_{ij}^{(\eta)}[\mathbf{W}]$ and $D_{ij}[\mathbf{W}]$

The distance $D_{ij}^{(\eta)}[\mathbf{W}]$, expressed as

$$D_{ij}^{(\eta)}[\mathbf{W}] \quad = \quad -\eta v_{ij\bullet}\log\left\{v_{ij\bullet}^{-1}\sum_{k=1}^{P} v_{ijk}\exp\left(-\frac{d_{ijk}}{\eta}\right)\right\},$$

transitions into the distance $D_{ij}[\mathbf{W}]$, expressed as

$$D_{ij}[\mathbf{W}] = \sum_{k=1}^{P} v_{ijk}d_{ijk},$$

with $\eta \to \infty$.

Explanation by Taylor Expansion Theory

Let

$$f(\theta) = \frac{1}{\theta}g(\theta), \tag{2.71}$$

be an infinitely differentiable function, and suppose we can think of the Taylor series

$$\begin{aligned} f(\theta) &= \sum_{n=0}^{\infty}\frac{f^{(n)}(a)}{n!}(\theta-a)^n \\ &= \left(\frac{1}{\theta}g(a)\right) + \left(\frac{1}{\theta}g'(a) - \frac{1}{\theta^2}g(a)\right)(\theta-a) + \ldots, \end{aligned} \tag{2.72}$$

as its 'infinite order' Taylor polynomial of f at a, where $f^{(n)}$ is the nth derivative of f with respect to θ.

Then, when we substitute η for $1/\theta$ and express the homotopic distance $D_{ij}^{(\eta)}[\mathbf{W}]$ into the functions $f(\theta)$ and $g(\theta)$, we obtain

$$\begin{aligned} D_{ij}^{(\eta)}[\mathbf{W}] = D_{ij}^{(1/\theta)}[\mathbf{W}] &= f(\theta) = \frac{1}{\theta} g(\theta), \\ \text{where } g(\theta) &= -v_{ij\bullet} \log \left\{ v_{ij\bullet}^{-1} \sum_{k=1}^{P} v_{ijk} \exp\left(-\theta d_{ijk}\right) \right\}. \end{aligned}$$

We are interested in the expression of $D_{ij}^{(1/\theta)}[\mathbf{W}]$ when $\theta \to 0$. Hence, we set $a = 0$ in the Taylor expansion, where the first and second term will be sufficient, since $a = 0$ is extremely close to $\theta \to 0$. We obtain

$$\begin{aligned} \lim_{\theta \to 0} D_{ij}^{(1/\theta)}[\mathbf{W}] &\simeq \left(\frac{1}{\theta} g(0) \right) + \left(\frac{1}{\theta} g'(0) - \frac{1}{\theta^2} g(0) \right) (\theta - 0) \\ &= g'(0). \end{aligned} \tag{2.73}$$

The derivative of g with respect to θ is

$$\frac{\partial g(\theta)}{\partial \theta} = g'(\theta) = \frac{v_{ij\bullet} \sum_{k=1}^{P} v_{ijk} d_{ijk} \exp(-\theta d_{ijk})}{\sum_{k=1}^{P} v_{ijk} \exp(-\theta d_{ijk})}. \tag{2.74}$$

When we plug this result into equation (2.73), we obtain with the following series of steps, the desired transition:

$$\begin{aligned} \lim_{\eta \to \infty} D_{ij}^{(\eta)}[\mathbf{W}] &= \lim_{\theta \to 0} D_{ij}^{(1/\theta)}[\mathbf{W}] \\ &\simeq \frac{v_{ij\bullet} \sum_{k=1}^{P} \max(w_{ik}, w_{jk}) d_{ijk} \exp(-0 d_{ijk})}{\sum_{k=1}^{P} v_{ijk} \exp(-0 d_{ijk})} \\ &= \frac{v_{ij\bullet} \sum_{k=1}^{P} v_{ijk} d_{ijk}}{v_{ij\bullet}} \\ &= \sum_{k=1}^{P} v_{ijk} d_{ijk} \\ &= D_{ij}[\mathbf{W}]. \end{aligned} \tag{2.75}$$

2.7.3 Derivation for the Inverse Exponential Distance

In this subsection we show the derivation that explains that the inverse exponential distance from equation (2.59) can be interpreted as the result of a minimization problem over attribute weights $\{t_{ijk}\}$:

$$\begin{aligned} D_{ij}^{(\eta)}[\mathbf{W}] &= \min_{\mathbf{t}_{ij}} \left\{ \sum_{k=1}^{P} t_{ijk} d_{ijk} + \eta \sum_{k=1}^{P} t_{ijk} \log\left(\frac{t_{ijk}}{v_{ijk}}\right) \right\}, \\ &= -\eta v_{ij\bullet} \log\left\{ v_{ij\bullet}^{-1} \sum_{k=1}^{P} v_{ijk} \exp\left(-\frac{d_{ijk}}{\eta}\right) \right\}, \end{aligned} \tag{2.76}$$

with the restrictions $t_{ijk} > 0$, $v_{ijk} > 0$, and

$$\sum_{k=1}^{P} t_{ijk} = v_{ij\bullet}, \qquad \text{where} \qquad v_{ij\bullet} = \sum_{k=1}^{P} v_{ijk}.$$

We will first find a solution for t_{ijk} in the minimization problem of the first line in (2.76). After we have found the solution, we show the derivation with which we can can obtain the second line in (2.76).

The Solution for t_{ijk}

When we use the Lagrangian multiplier δ to express the minimization problem from the first line in (2.76), its unconstrained form, we obtain

$$D_{ij}^{(\eta)}[\mathbf{W}] = \min_{\mathbf{t}_{ij}} \left\{ \sum_{k=1}^{P} t_{ijk} d_{ijk} + \eta \sum_{k=1}^{P} t_{ijk} \log\left(\frac{t_{ijk}}{v_{ijk}}\right) - \delta \left(\sum_{k=1}^{P} t_{ijk} - v_{ij\bullet} \right) \right\}.$$

The partial derivatives with respect to v_{ijk} and δ are

$$\frac{\partial D_{ij}^{(\eta)}[\mathbf{W}]}{\partial t_{ijk}} = d_{ijk} + \eta \left[\log(t_{ijk}) - \log(v_{ijk})\right] + \eta - \delta, \text{ and} \tag{2.77}$$

$$\frac{\partial D_{ij}^{(\eta)}[\mathbf{W}]}{\partial \delta} = \sum_{k=1}^{P} t_{ijk} - v_{ij\bullet}. \tag{2.78}$$

By setting both partial derivatives equal to zero, we can find the global minimum of $D_{ij}^{(\eta)}[\mathbf{W}]$ with respect to t_{ijk}. Setting (2.77) equal to zero, and solving for t_{ijk} gives

$$\widehat{t}_{ijk} = \exp\left(\frac{\delta - \eta}{\eta}\right) v_{ijk} \exp\left(\frac{-d_{ijk}}{\eta}\right). \tag{2.79}$$

Substituting the result for t_{ijk} into (2.78) gives

$$\frac{\partial D_{ij}^{(\eta)}[\mathbf{W}]}{\partial \delta} = \sum_{k=1}^{P} \exp\left(\frac{\delta-\eta}{\eta}\right) v_{ijk} \exp\left(\frac{-d_{ijk}}{\eta}\right) - v_{ij\cdot}. \tag{2.80}$$

When we set (2.80) equal to zero and solve for $\exp\left(\frac{\delta-\eta}{\eta}\right)$, we obtain

$$\exp\left(\frac{\delta-\eta}{\eta}\right) = v_{ij\cdot} \frac{1}{\sum_{k=1}^{P} v_{ijk} \exp\left(\frac{-d_{ijk}}{\eta}\right)}. \tag{2.81}$$

When substituting the result in (2.81) back in (2.79) we arrive at the optimal solution for t_{ijk}. The result is

$$\widehat{t}_{ijk} \quad = \quad v_{ij\cdot} \frac{v_{ijk} \exp\left(\frac{-d_{ijk}}{\eta}\right)}{\sum_{k'=1}^{P} v_{ijk'} \exp\left(\frac{-d_{ijk'}}{\eta}\right)}. \tag{2.82}$$

The Inverse Exponential Distances

Now that we have the solution for the minimizing object-pair's specific attribute weights $\{t_{ijk}\}$, we can rewrite the first line in (2.76) into the second line with the use of the definition for the optimal t_{ijk} in (2.82). Denote the numerator in equation (2.82) to be the scalar

$$\tilde{t}_{ijk} = v_{ijk} \exp\left(\frac{-d_{ijk}}{\eta}\right), \tag{2.83}$$

and the denominator as

$$\tilde{t}_{ij\cdot} = \sum_{k=1}^{P} \tilde{t}_{ijk} = \sum_{k}^{P} v_{ijk} \exp\left(\frac{-d_{ijk}}{\eta}\right), \tag{2.84}$$

such that

$$\widehat{t}_{ijk} = v_{ij\cdot} \frac{\tilde{t}_{ijk}}{\tilde{t}_{ij\cdot}}.$$

Then, we can use the following series of derivations to come to an intermediate expression of the inverse exponential distance:

$$\begin{aligned} D_{ij}^{(\eta)}[\mathbf{W}] &= \sum_{k=1}^{P} \widehat{t}_{ijk} d_{ijk} + \eta \sum_{k=1}^{P} \widehat{t}_{ijk} \log\left(\frac{\widehat{t}_{ijk}}{v_{ijk}}\right) \\ &= \sum_{k=1}^{P} \widehat{t}_{ijk} \left(d_{ijk} + \eta \widehat{t}_{ijk} \left\{\log\left(\widehat{t}_{ijk}\right) - \log\left(v_{ijk}\right)\right\}\right) \\ &= \sum_{k=1}^{P} \left(v_{ij\cdot} \frac{\tilde{t}_{ijk}}{\tilde{t}_{ij\cdot}}\right) \left(d_{ijk} + \eta \left\{\log\left(v_{ij\cdot} \frac{\tilde{t}_{ijk}}{\tilde{t}_{ij\cdot}}\right) - \log\left(v_{ijk}\right)\right\}\right). \end{aligned} \tag{2.85}$$

When we move $v_{ijk\bullet}$ and $\tilde{t}_{ij\bullet}$ outside the summation, and simplify the term within the brackets using (2.83) and (2.84), we obtain:

$$\begin{aligned} D_{ij}^{(\eta)}[\mathbf{W}] &= \frac{v_{ij\bullet}}{\tilde{t}_{ij\bullet}} \sum_{k=1}^{P} \tilde{t}_{ijk} \left(d_{ijk} + \eta \left\{ \log(\tilde{t}_{ijk}) - \log\left(\frac{\tilde{t}_{ij\bullet}}{v_{ij\bullet}}\right) - \log\left(v_{ijk}\right) \right\} \right) \\ &= \frac{v_{ij\bullet}}{\tilde{t}_{ij\bullet}} \sum_{k=1}^{P} \tilde{t}_{ijk} \left(d_{ijk} + \eta \left\{ \left(-\frac{d_{ijk}}{\eta} \right) - \log\left(\frac{\tilde{t}_{ij\bullet}}{v_{ij\bullet}}\right) \right\} \right) \\ &= \frac{v_{ij\bullet}}{\tilde{t}_{ij\bullet}} \sum_{k=1}^{P} \tilde{t}_{ijk} \left(-\eta \log\left\{ \frac{\tilde{t}_{ij\bullet}}{v_{ij\bullet}} \right\} \right) \end{aligned} \tag{2.86}$$

The intermediate result (2.86) expresses $D_{ij}^{(\eta)}[\mathbf{W}]$ only in the terms of η, $v_{ij\bullet}$, $\tilde{t}_{ijk}$, and $\tilde{t}_{ij\bullet}$. What remains is to further simplify the equation such that $\tilde{t}_{ij\bullet}$ occurs only once and $\tilde{t}_{ijk}$ drops out. The derivation is

$$\begin{aligned} D_{ij}^{(\eta)}[\mathbf{W}] &= \frac{v_{ij\bullet}}{\tilde{t}_{ij\bullet}} \sum_{k=1}^{P} \tilde{t}_{ijk} \left(-\eta \log\left\{ \frac{\tilde{t}_{ij\bullet}}{v_{ij\bullet}} \right\} \right) \\ &= -\eta v_{ij\bullet} \log\left\{ \frac{\tilde{t}_{ij\bullet}}{v_{ij\bullet}} \right\} \frac{1}{\tilde{t}_{ij\bullet}} \sum_{k=1}^{P} \tilde{t}_{ijk\bullet} \\ &= -\eta v_{ij\bullet} \log\left\{ v_{ij\bullet}^{-1} \tilde{t}_{ij\bullet} \right\}. \end{aligned}$$

Finally, when we substitute the definition for $\tilde{t}_{ij\bullet}$(2.84) back into the equation, we arrive at the second line in (2.76):

$$\begin{aligned} D_{ij}^{(\eta)}[\mathbf{W}] &= -\eta v_{ij\bullet} \log\left\{ v_{ij\bullet}^{-1} \tilde{t}_{ij\bullet} \right\} \\ &= -\eta v_{ij\bullet} \log\left\{ v_{ij\bullet}^{-1} \sum_{k=1}^{P} v_{ijk} \exp\left(-\frac{d_{ijk}}{\eta} \right) \right\}. \end{aligned}$$

Chapter 3

A Robust and Tuned COSA-KNN

The implementation of the COSA-KNN algorithm might not have enough power with its default values of the tuning parameters λ and K, being $\lambda = 0.2$ and $K = \sqrt{N}$. These values may be too far off from the optimal tuning parameter values that can retrieve an underlying the clustering structure. In such a situation, we need a search strategy that can provide good candidate values for the tuning parameters.

To increase the number of good candidate values, we propose in Section 3.1 to apply a robust version of COSA-KNN on a data set. In the algorithm of this robust version of COSA-KNN, a criterion is minimized that is based on median-based attribute dispersions, instead of mean-based attribute dispersions. The result is that in the robust version of COSA-KNN the attribute weights are also computed based on median-based attribute dispersions, instead of mean-based attribute dispersions, rendering the selection of the subset of attributes to be less sensitive to the noise in the data.

We propose a criterion-based search strategy that can find successful values of the tuning parameters. We describe in Section 3.2 the behavior of both the original COSA-KNN criterion, and the criterion for the robust COSA-KNN, as a function of the tuning parameters. Compared to the original criterion, the robust criterion values show a systematic 'zigzag' behavior between the odd and even values for K. This behavior diminishes for larger value of K, but is amplified by the value for λ. The higher λ, the more pronounced the peculiarity.

In section 3.3 we describe the criterion-based search strategy: the Gap statistic procedure, developed by Tibshirani, Walther, and Hastie (2001). The Gap statistic procedure is a permutation approach in which we select that particular combination of values for the tuning parameters that provides the largest gap between the value of the robust criterion obtained for a particular data set, and (an approximation of) the expected value obtained for a null-reference model, i.e. a comparable data set with no clustering structure. Two versions of the Gap statistic procedure will be described: the

original procedure where the natural logarithm is taken over the robust COSA-KNN criterion of each data set, and the version where the criterion is not transformed. Since the Gap statistic was originally developed to choose the number of clusters in K-means type algorithms, it is not self-evident that it would work for λ and K in COSA-KNN.

We will combine the Gap statistic procedure with the robust version of COSA-KNN and a pre-specified grid of values for λ and K. In Section 3.4, we apply the Gap statistic to a real data example, and show that with the resulting optimal tuning parameters a sharper representation of the clustering structure is obtained, i.e., the structure exhibits a clearer separation between the different clusters. In this data example, the consequences of the zigzag peculiarity of the robust criterion will also be discussed. Section 3.4.2 shows the effectiveness of the Gap statistic on a simulated data example that consists of a subtle clustering structure that would not have been found with the default settings of (robust) COSA-KNN. Section 3.4.3 shows the satisfactory behavior of the Gap statistic procedure and COSA-KNN, when applied on a data example that represents noise only. The discussion of the chapter is provided in Section 3.5.

3.1 Robust versus Non-Robust Attribute Dispersions

The goal of COSA-KNN is to obtain distances between objects that can represent a hard-to-detect clustering structure present in a data set ($\mathbf{X}$) of N objects by P attributes. The clustering structure is hard to detect when many of these P attributes do not contribute to the clustering, and the small number of attributes that do contribute to the clustering, may be cluster-specific, i.e., different for each cluster. In Chapter 1 we described a typical data model of such a clustering structure, referred to as the prototype model. To facilitate reading, we also display this specific prototype model in Figure 3.1.

The COSA-KNN distances are defined as

$$D_{ij}[\mathbf{W}] = \sum_{k=1}^{P} \max(w_{ik}, w_{jk}) d_{ijk}, \tag{3.1}$$

where each d_{ijk} is defined as

$$d_{ijk} = \frac{|x_{ik} - x_{jk}|}{IQR(\{x_{ik}\}_{i=1}^{N})/1.35} \quad , \tag{3.2}$$

and the attribute weights are defined as

$$w_{ik} = \frac{\exp\left(-\frac{S_{ik}}{\lambda}\right)}{\sum_{k=1}^{P} \exp\left(-\frac{S_{ik}}{\lambda}\right)} \quad , \tag{3.3}$$

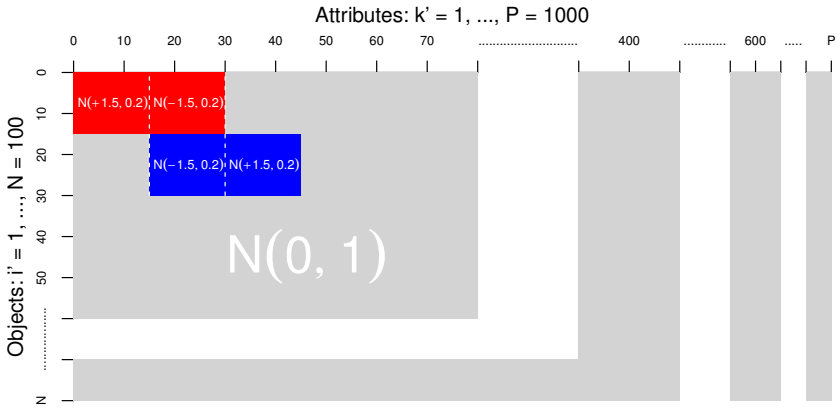

Figure 3.1: A Monte Carlo model for 100 objects with 1,000 attributes (not all are shown due to $P \gg N$). There are two small 15-object groups (red and blue), clustering each on 30 attributes out of 1000 attributes, with partial overlap, and nested within an unclustered background of 70 objects (grey). After generating a data set from this model, each attribute is standardized to have zero mean and unit variance.

where S_{ik} is the arithmetic mean of the distances on attribute k, between object i and its K nearest neighbors, i.e.,

$$S_{ik} = \sum_{j=1}^{N} \frac{c_{ij} d_{ijk}}{K}, \tag{3.4}$$

with c_{ij} equal to 1 when j belongs to the K nearest neighbors of i, and 0 otherwise. Although it is clear that S_{ik} is influenced by K in equation (3.4), the value for S_{ik} is also implicitly influenced by the value of λ, since the K nearest neighbors for object i are found based on the COSA distance $D_{ij}[\mathbf{W}]$ from an earlier iteration in the algorithm COSA-KNN, and $D_{ij}[\mathbf{W}]$ in (3.7) consists of the attribute weights that involve the value of λ.

Instead of a mean of the distances on attribute k, we can also use the median. Using medians as an alternative measure of central tendency eliminates effects of so-called outliers, making the overall procedure more robust as compared to the use of the mean-based version. In this chapter we will refer to median-based attribute dispersion as the robust attribute dispersions, and to the mean-based attribute dispersions as non-robust attribute dispersions. Thus, instead of the non-robust attribute dispersion, we can compute the robust attribute dispersion as

$$\widetilde{S}_{ik} = \text{median}\left(\{d_{ijk}\}_{j \in KNN(i)}\right), \tag{3.5}$$

such that the definition of a median-based (robust) attribute weight becomes

$$\widetilde{w}_{ik} = \frac{\exp\left(-\frac{\widetilde{S}_{ik}}{\lambda}\right)}{\sum_{k=1}^{P}\exp\left(-\frac{\widetilde{S}_{ik}}{\lambda}\right)} \quad , \tag{3.6}$$

which leads to the median-based (robust) COSA distance, i.e.,

$$D_{ij}[\widetilde{\mathbf{W}}] = \sum_{k=1}^{P}\max(\widetilde{w}_{ik},\widetilde{w}_{jk})d_{ijk}. \tag{3.7}$$

Although Friedman and Meulman (2004) advised to use this robust version for the attribute dispersion, no explanation, nor any example of comparison was given to demonstrate its effectiveness.

3.1.1 Demonstrating a Prototype Data Example

When we apply COSA-KNN to a data set generated from the prototype model (Figure 3.1), then the results are less dependent on K and λ when the robust attribute dispersions (equation 3.5) are used, compared to the use of the mean-based attribute dispersions. We depicted in Figure 3.2 an example of dendrograms of both versions of the COSA distances for three different combinations of values for λ and K. For this specific data example, we define a dendrogram to be successful when we can cut it into 70 singletons and two groups of each 15 objects.

Although the successful results of the non-robust COSA-KNN can show a sharper representation of the clustering structure, i.e., more distinct clusters, the robust version of COSA-KNN has more combinations for the values of K and λ that lead to successfull results. In Figure 3.3 we show that for combinations of $1 \leq K \leq 50$ and $0.01 \leq \lambda \leq 1$, the true clustering was revealed successfully for 69 percent of the times in a dendrogram of the distances for the robust version of COSA-KNN, while for the non-robust version only 31 percent of the same combinations for K and λ produced successful dendrograms. Thus, when using the non-robust attribute dispersion in equation (3.4), it is more likely to miss the true clustering structure in the data with COSA-KNN.

Another important difference between both versions of the algorithm occurs in the first iterations. Empirical findings, displayed in Figure 3.4, show that in general the robust version of COSA-KNN needs fewer iterations to retrieve a good clustering structure. Both the robust and non-robust version of the attribute weights are computed based on neighborhoods. However, at the starting iterations we see that the neighborhoods of the clustered objects consist of a large number of 'incorrect' neighbors, where incorrect neighbors are defined as objects that are from a different cluster. We conjecture that the robust version of COSA-KNN is less affected by these incorrect neighbors, when compared to the non-robust version of COSA-KNN.

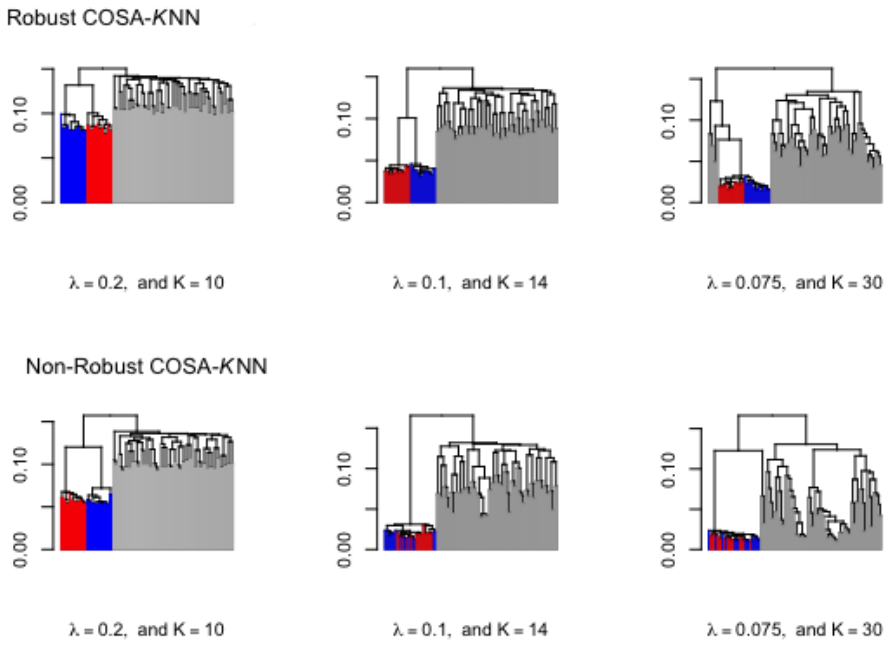

Figure 3.2: Average linkage dendrograms. The dendrograms in the first row are visualizations of the COSA-KNN distances using robust attribute dispersions. The dendrograms in the second row use non-robust attribute dispersions.

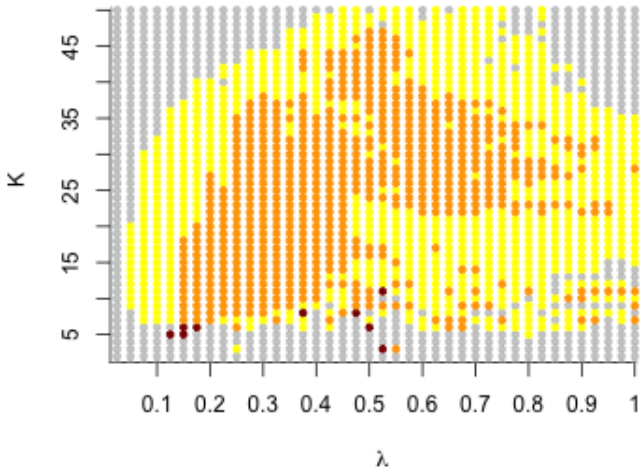

Figure 3.3: The grid of K (vertical axis) versus λ (horizontal axis) combinations for the robust and non-robust COSA-KNN: the grey points indicate unsuccessful dendrograms for both the robust and the non-robust versions, the yellow points indicate successful dendrograms for only the robust versions, dark red points indicate successful dendrograms for only the non-robust version, and the orange points indicate the successful dendrograms for both versions.

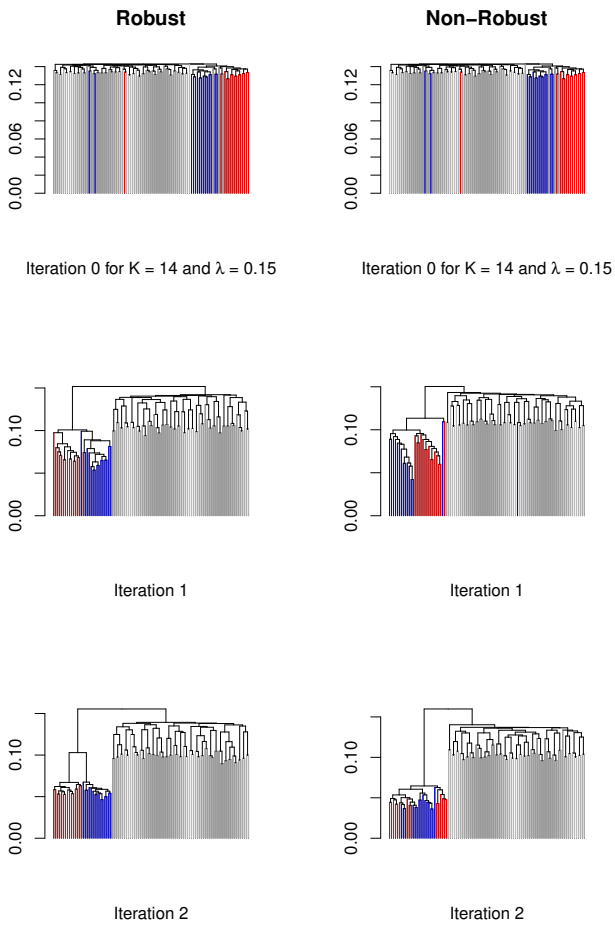

Figure 3.4: The dendrograms of the robust COSA-KNN distances converge faster towards a correct representation of the clustering structure than the non-robust COSA-KNN distances for $K = 14$ *and* $\lambda = 0.15$ *at iterations* 0, 1, *and* 2.

Attribute Distance Distributions

When an incorrect neighbor has a different clustering mechanism on the signal attributes in the particular neighborhood of an object, then it likely enlarges the dispersion for each of these signal attributes on which the incorrect neighbor clusters differently. Suppose we use the prototype model from Figure 3.1, then an attribute distance from equation (3.2) is folded-normally distributed (Leone, Nelson, & Not-

tingham, 1961):

$$d_{ijk} \sim \left\{ \frac{1}{\sqrt{2\pi\sigma_{ijk}^2}} e^{-\frac{(d_{ijk}-\mu_{ijk})^2}{2\sigma_{ijk}^2}} + \frac{1}{\sqrt{2\pi\sigma_{ijk}^2}} e^{-\frac{(d_{ijk}+\mu_{ijk})^2}{2\sigma_{ijk}^2}} \right\}. \tag{3.8}$$

Here, μ_{ijk} and σ_{ijk}^2 are the mean and variance of the normal distribution that is to be folded. When objects i and j belong to the same cluster, then for a signal attribute from an unique subset we have $\mu_{ijk} = 0$ and $\sigma_{ijk}^2 \approx (2 * 0.2^2)/(0.15 \cdot 0.2^2 + 0.85 \cdot 1^2))$. For the same attribute, the distance between a red cluster object and a noise object, we have $\mu_{ijk} = 1.5/(0.15 \cdot 0.2^2 + 0.85 \cdot 1^2)$ and $\sigma_{ijk}^2 = (1^2 + 0.2^2)/(0.15 \cdot 0.2^2 + 0.85 \cdot 1^2)$, resulting in a distribution with a higher mean and larger variance, as is depicted in Figure 3.5. Indeed, from Figure 3.5 we can also obtain that the attribute distance between an object from the red cluster with a noise-object (wrong neighbor) is most likely to result in the largest (outlying) attribute distance. While the non-robust attribute dispersion, S_{ik} in (3.4), is sensitive to these larger attribute distances, the robust attribute dispersion, $\widetilde{S}_{ik}$ in (3.5), is (to some extent) protected (S_{ik} in (3.4)) due to the properties of the median.

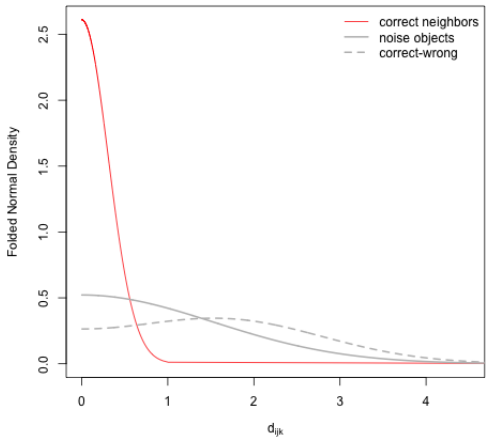

Figure 3.5: The theoretical probability density functions of the attribute distances between correct neighbors (red), correct and incorrect neighbors (dashed grey), and between noise objects (solid grey) for a signal attribute from a unique subset.

The Behavior of $\widetilde{S}_{ik}$ and S_{ik} in COSA-KNN

Shortly after the first iteration, the robust version of COSA-KNN often already performs better than the non-robust version. Both versions of the algorithm start with

equal attribute weights, and hence the same COSA distances. We will refer to these attribute weights and COSA distances as the results of 'iteration 0'. The dendrograms in the top panels of Figure 3.4 visualize these distances. In these identical dendrograms at iteration 0, the objects from the same cluster (red or blue color) seem closer to each other than objects that do not belong to the same cluster. However, the complete clustering structure from this specific data set is not yet revealed. Based on these distances, we find a first estimate of the neighborhoods of each the objects and compute the attribute weights.

From the results of iteration 0 it is very likely that each object will have a neighborhood that does not yet contain the 'true' nearest neighbors. Therefore, the effect of outliers on either the robust and non-robust attribute dispersions already becomes visible for the different COSA-KNN versions in iteration 1. Figure 3.6 displays this difference at iteration 1 for the attribute dispersions of 15 red cluster objects from the data that was generated from the prototype model (Figure 3.1), i.e. $i = i' \in \{1, \ldots 15\}$. For these objects from the red cluster are known to have their own unique subset of attributes for $k = k' \in \{1, \ldots 15\}$ and a shared subset of attributes with the objects from the blue cluster for $k = k' \in \{16, \ldots, 30\}$, all other attributes are considered as noise. Figure 3.6 depicts for each attribute the average over the 15 objects for either versions of the attribute weights (top panel), as well as the two versions of the attribute weights (lower panel). Figure 3.6 depicts higher averages for the dispersions of the unique subset of attributes, than for the shared subset attributes. Since the generating process of attribute values in the shared subset is the same for objects in the red and the blue cluster, the attribute dispersions can only be distorted by the remaining noise objects. However, the averages of the attribute dispersions on the unique subset, can be distorted by the objects from the blue cluster as well. The more objects there are to distort the neighborhood, the more the non-robust version of the attribute dispersions (and weights) are affected. The averages of the dispersions (and the weights) for the unique subset of attribute are higher (and lower), than those of the robust versions. Since it is to be expected that neighborhoods of the objects are more distorted in the beginning iterations, we see that the robust COSA-KNN converges faster and produces more often successful distances, when compared to the non-robust version.

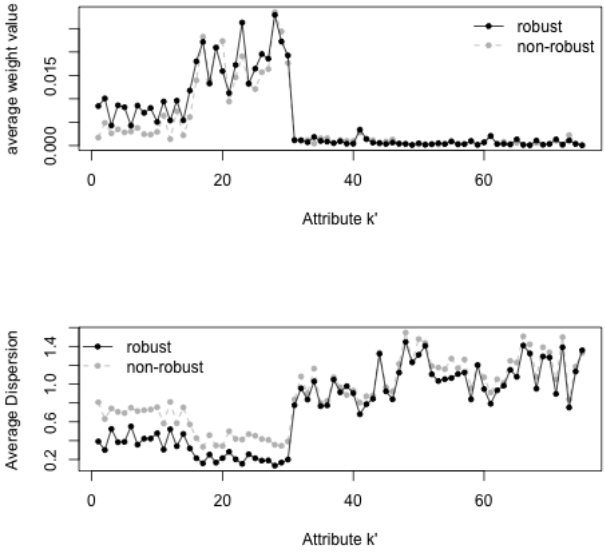

Figure 3.6: The top panel depicts the robust (black) and non-robust (grey) average attribute weights of the red cluster objects for attribute $k' = 1 \ldots 75$. The lower panel depicts the corresponding robust and non-robust average attribute dispersions, the average is taken over the neighborhoods from each object in the red cluster.

3.1.2 Data Example 2: Empowered Results

The robust version of COSA-KNN may be the only option in situations where the clustering structure is more subtle. Figure 3.7 depicts an example of a generating model for such a data set, referred to as the subtle structure model. The clustering mechanism in this model is very similar to that of the prototype model. While N remains 100 objects, the number of attributes is set to $P = 10,000$ (instead of a $1,000$). We still have two clusters, but this time each cluster consists of 20 objects (instead of 15). The sizes of the subsets of signal attributes remain 30 attributes, and the overlap remains 15 attributes. The process for data generation remains the same to that of the prototype model for the noise attribute values, as well as for the signal attribute values in the unique and shared subsets. After the data is generated, each attribute is standardized to have a zero mean and unit variance.

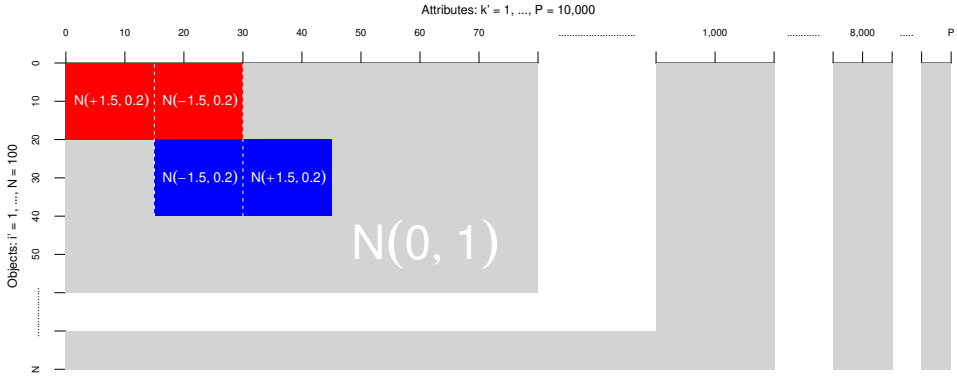

Figure 3.7: A subtle structure data model for dataset $\mathbf{X}$ *with 100 objects and 10,000 attributes (of which only 60 are shown). There are two groups of 20-objects (red and blue) each clustering on a subset of 30 attributes. After generating a data set from this model, each attribute is standardized to have zero mean and unit variance.*

Independent of the choice for λ or K, the non-robust version of COSA-KNN is not able to pick up the clustering structure in the domain $1 \leq K \leq 50$ with any of the values for

$$\lambda \in \{0.01, 0.05, 0.075, 0.1, 0.125, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, ..., 1\}.$$

However, provided we choose a suitable combination of K and λ, the robust version can result in distances that yield a successful dendrogam, as is shown in Figure 3.8. While both versions of the COSA-KNN manage to distinguish the noise objects from the clustered objects, only the robust COSA-KNN version manages to distinguish the two clusters.

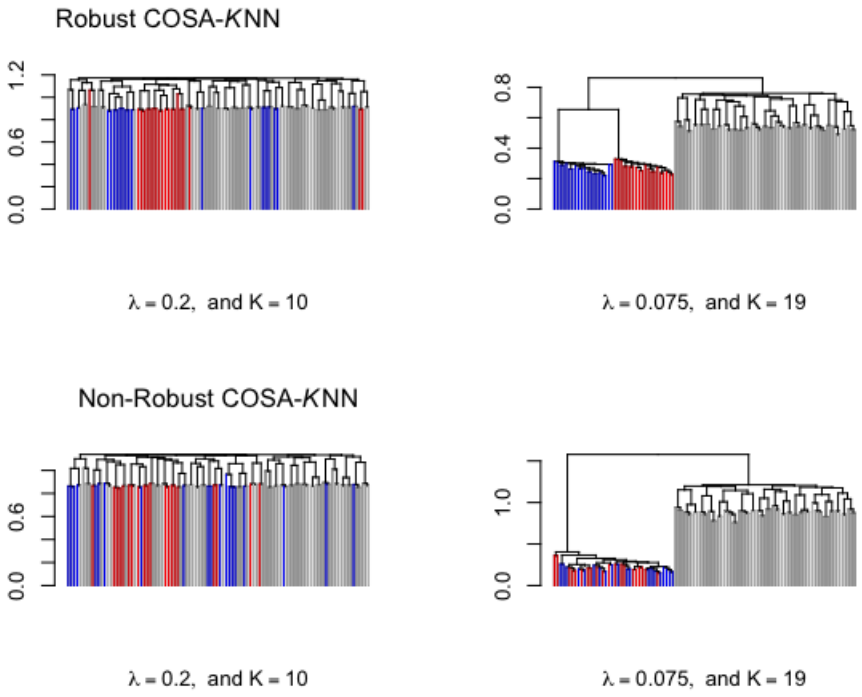

Figure 3.8: Average linkage dendrograms of COSA-KNN distances for different values for λ and K. The top-left and top-right dendrograms represent the results of the robust version, the bottom dendrograms represents the results of the non-robust version.

A reason why the non-robust COSA-*K*NN cannot retrieve the clustering structure is its sensitivity to nestedness of the clustering. Because the non-robust version is slower in finding the clustering structure in the first iterations, the chances are also higher that objects from different clusters with overlapping subsets of attributes, have the tendency to stay longer in each others neighborhoods during the iterations; see Figure 3.9. Since the attribute dispersion from equation (3.4) is based on the mean of all the attribute distances within the neighborhood, Figure 3.6 depicts that the dispersions become small (and the weights become high) for the shared subset of attributes. Since the mean-based attribute dispersions cannot be 'robustly' corrected for incorrect neighbors, the cluster-unique attribute dispersions within the neighborhood of an object remain overestimated. This overestimation is not only due to the distortion of the noise objects, but also due the overrepresentation of the objects from another 'neighbor cluster', rendering attribute weights with low values for the cluster-unique subset.

When there are too many attributes in the data set (or too many objects), this

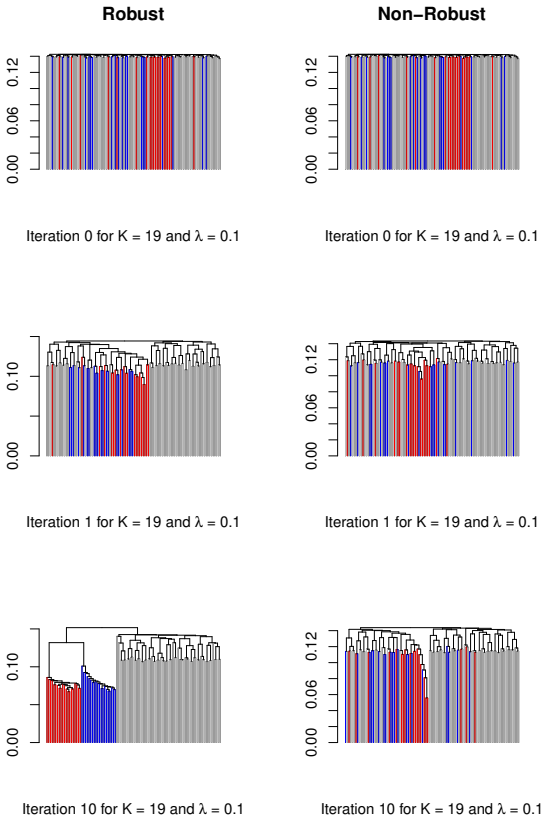

Figure 3.9: Dendrograms for the robust distances converge faster towards a correct representation, as compared to the non-robust COSA-KNN distances for $K = 19$ *and* $\lambda = 0.10$.

particular distortion process may even affect the median-based attribute dispersions. As can be seen in Figure 3.9, for the specific data set from the subtle structure model, neither non-robust version of COSA-KNN manages to retrieve the clustering structure for the usual values of $\lambda = 0.2$ and $K = \sqrt{N}$. Thus, a strategy is needed to find optimal values for λ and K.

3.2 The Criterion as a Function of λ and K

To select the values of λ and K, we propose to apply a permutation approach that is based on the criterion of COSA-KNN, and is referred to as the Gap statistic (Tibshirani et al., 2001). For a successful application of this approach, a good understanding

of the properties of the COSA-KNN criterion regarding its tuning parameters is required. The definition of the original COSA-KNN criterion is

$$Q(\lambda, K) = \sum_{i=1}^{N} \left\{ \sum_{k=1}^{P} w_{ik} S_{ik} + \lambda \left(\sum_{k=1}^{P} w_{ik} \log(w_{ik}) + \log(P) \right) \right\}. \tag{3.9}$$

In this criterion the value for each d_{ijk} is fixed, such that the values for each w_{ik} in (3.3) and each S_{ik} in (3.4), follow from the values for λ and K. Since $Q(\lambda, K)$ is based on the non-robust attribute weights, it is referred to as the non-robust criterion. Figure 3.10 displays the behavior of the values of this non-robust criterion for different values of λ and K, given the generated data from the prototype model.

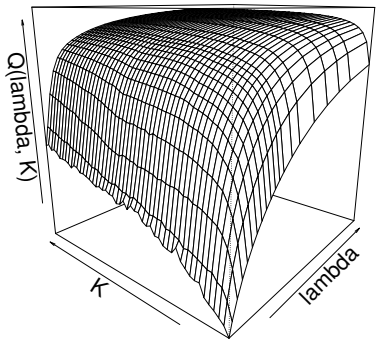

Figure 3.10: The non-robust COSA-KNN criterion values on a propotype data set sampled from the COSA model with $N = 100$ objects and $P = 1000$ attributes. We ran COSA-KNN on a grid with $K \in \{1, 2, 3, ..., 50\}$ and $\lambda \in \{0.01, 0, 025, 0.05, 0.075, ..., 0.3, 0.35, 0.4, 0.5, ..., 1.0\}$. The arrows at the axes in the plot indicate the direction in which either the value of λ or the value of K increases.

Figure (3.10) displays that *in general* higher values of either λ or K result in an increased value for the criterion, which is a concave function. While keeping K fixed, a higher value for λ renders the values of the attribute weights within each neighborhood to become more equal, which results in a stronger influence of the larger attribute dispersions, resulting in an increase in the criterion value (the reverse is true for the smaller attribute dispersions). However, changing the value for λ may also render the attribute weights to indicate different subsets of attributes, resulting in different COSA distances from which different neighborhoods are extracted. Thus, the monotone pattern in the relationship between λ and $Q(\lambda \mid K)$ can be interrupted.

While keeping the value for λ fixed, a similar monotone pattern exists between the criterion and K. With a larger K, each object i will obtain more nearest neighbors. Since the distance between object i and the K^{th} nearest neighbor is lower than (or equal to) the distance between object i and the $(K+1)^{th}$ nearest neighbor, we may expect that the weighted sum of the attribute dispersions also increases for $K + 1$ in

the criterion in (3.9). Note, however, when we add the $(K+1)^{th}$ neighbor objects to each corresponding neighborhood, the distribution of the attribute dispersions within each neighborhood may change, which may change the distribution of the attribute weights. When this change is the difference between detecting a clustering structure or not, then it may happen that the criterion actually drops for a larger value of K. Random fluctuations can also contribute to violations of monotonicity at a fixed low value for λ and the lower values of K. The lower the value of λ, the more the attribute weights diverge from equal weights, and the more sensitive they are for random fluctuations of the attribute dispersions.

3.2.1 The Robust Criterion and K: a ZigZag Tendency

The robust version of COSA-KNN contains the robust attribute dispersions, $\widetilde{S}_{ik}$ in (3.5) and the robust attribute weights, $\widetilde{w}_{ik}$ in (3.6). The criterion is defined as

$$\widetilde{Q}(\lambda, K) = \sum_{i=1}^{N} \left\{ \sum_{k=1}^{P} \widetilde{w}_{ik} \widetilde{S}_{ik} + \lambda \left(\sum_{k=1}^{P} \log(\widetilde{w}_{ik}) \widetilde{w}_{ik} + \lambda \log(P) \right) \right\}. \tag{3.10}$$

The properties that were described for the non-robust criterion in equation (3.9), can also hold *in general* for this robust criterion, but only as long as the values for K in the criterion are only *even*, or only *odd*.

The robust criterion has a peculiar behavior over the value of K. Since the peculiar behavior is directly related to the definition of the robust attribute dispersion in (3.5), it is useful to first clarify the direct relationship between $\widetilde{S}_{ik}$ and $\widetilde{Q}(\lambda, K)$, without the involvement of the robust attribute weights $\widetilde{w}_{ik}$ in (3.6).

Based on the definition of equation (3.6), we can rewrite the robust criterion in equation (3.10) as,

$$\widetilde{Q}(\lambda, K) = \sum_{i=1}^{N} -\lambda \log \left\{ \frac{1}{P} \sum_{k=1}^{P} \exp\left(\frac{-\widetilde{S}_{ik}}{\lambda} \right) \right\}. \tag{3.11}$$

This rewritten formulation is a sum of N 'softmin' attribute dispersions, that can be continuously deformed over λ to a sum of mean attribute dispersions, or to a sum of the minimum attribute dispersions. For each neighborhood, the definition of the softmin over P attribute dispersions is

$$\widetilde{Q}(\lambda, K)_i = -\lambda \log \left\{ \frac{1}{P} \sum_{k=1}^{P} \exp\left(\frac{-\widetilde{S}_{ik}}{\lambda} \right) \right\}, \tag{3.12}$$

For $\lambda \to 0$, we can obtain the minimum over the P attribute dispersions, i.e.,

$$\lim_{\lambda \to 0} \widetilde{Q}(\lambda, K)_i = \min_k \left(\widetilde{S}_{ik} \right). \tag{3.13}$$

For $\lambda \to \infty$, we obtain the arithmetic mean of the attribute dispersions, i.e.,

$$\lim_{\lambda \to \infty} \widetilde{Q}(\lambda, K)_i = \frac{1}{P} \sum_{k=1}^{P} \widetilde{S}_{ik}. \tag{3.14}$$

The higher λ, the stronger the influence of the large attribute dispersions (and vice versa). Therefore, the higher λ, the higher the value of the criterion. (Note that one can arrive at equations (3.11), (3.13), and (3.14), with derivations that are similar to those used in Sections 2.7.2 and 2.7.3 from the Appendix of Chapter 2)

In general, for higher values of K, we can also expect larger attribute dispersions, and thus a larger value of the robust criterion. However, as is displayed in Figure 3.11, the behavior of the criterion over the values of K shows a particular zigzag behavior between the odd and even values for K. This particular zigzag pattern seems to be stronger present in the lower values of K, and is amplified for larger values of λ.

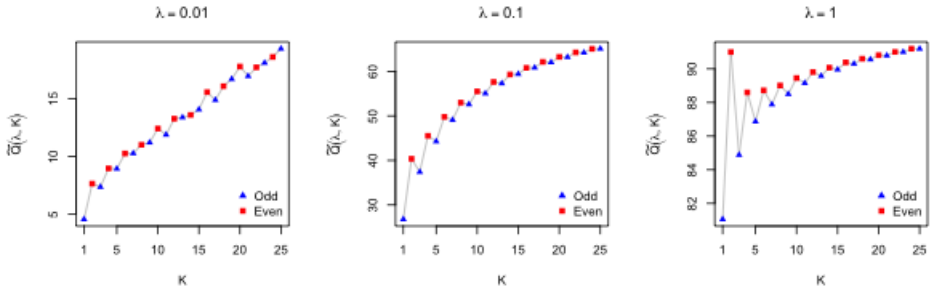

Figure 3.11: The behavior of $\widetilde{Q}(\lambda, K)$ as a function of K for λ equal to 0.01, 0.1, and 1. The blue triangles represent odd values for K, the red squares represent the even values for K. These results were obtained from a data set of $N = 100$ objects by $P = 1000$ attributes data that consisted of noise only: i.i.d. $\sim N(0, 1)$. After generation of the data, the attributes were standardized to have zero-mean and unit-variance.

While keeping λ fixed, we not only see that the criterion value is lower for an odd value for K, compared to $K + 1$, but the criterion value for the odd value for K is also lower than, or approximately equal to, the criterion value based on $K - 1$. This zigzag pattern can be formulated as

$$\widetilde{Q}(\lambda, K-1) \gtrapprox \widetilde{Q}(\lambda, K) < \widetilde{Q}(\lambda, K+1) \gtrapprox \widetilde{Q}(\lambda, K+2) < \ldots, \text{ for odd } K. \quad (3.15)$$

Thus, for an odd number of neighbors K, the sum of the (inverse-exponential) means of the robust attribute dispersions is lower than the sum for $K+1$ neighbors, but also lower than, or approximately equal to, when based on $K - 1$ neighbors.

From the perspective of a truncated mean (or trimmed mean), we can explain the difference between an *odd* value of K, and the even value $K - 1$, for a median-based attribute dispersion. We have seen in Section 3.1.1 (Figure 3.6) that the mean-based (non-robust) attribute dispersion is more sensitive to the higher attribute distances than the median-based (robust) attribute dispersion. This robustness property can be generalized to attribute dispersions that are defined by the truncated mean of the attribute distances, where first an equal amount at the high and low end of the attribute distances is discarded, and then the mean computed over the remaining

attribute distances. The higher the percentage of attribute distances that are discarded, the more robust the truncated mean is to outlying large attribute distances. Note that the attribute distance is bounded by zero, and that there are less (if any) small outlying distances to be expected.

The median-based attribute dispersion, $\widetilde{S}_{ik}$ in (3.5), is a special case of the truncated mean-based attribute dispersion. In particular, for K is odd, the attribute dispersion $\widetilde{S}_{ik}$ is the fully truncated mean of the attribute distances, i.e. only the middle attribute distance remains. For K is even, $\widetilde{S}_{ik}$ is the truncated mean where $K/2-1$ of the smallest attribute distances, and $K/2-1$ of the largest attribute distances are discarded. While in general for values of K, a higher proportion of the attribute distances is discarded, the proportion for an even number of neighbors, defined as $K+1$, is always smaller than the discarded proportion of attributes for odd K, resulting in another zigzag pattern, i.e.,

$$\frac{(K-1)-2}{(K-1)} < \frac{K-1}{K} > \frac{(K+1)-2}{(K+1)} < \frac{(K+2)-1}{(K+2)} > \ldots, \text{ for odd } K. \quad (3.16)$$

Although for large values of K these inequalities may be negligible, the smaller the value of K the larger these inequalities become. For odd K, the proportion of extreme attribute distances that is discarded, is larger than the proportion that is discarded for the $K-1$ and $K+1$ even numbers of neighbors. It is more likely that the robust attribute dispersion for odd K is smaller, than the robust attribute dispersion based on $K-1$, or $K+1$, explaining the zigzag pattern as displayed in Figure 3.11.

What is left unexplained is that the zig-zag pattern is moderated by the values of λ. The higher the values of λ, the stronger the influence of the larger attribute dispersions, thus the larger the difference between the softmin attribute dispersions for odd and even K. Therefore we see for the highest value of λ, in the right panel of Figure 3.11, that the zigzag pattern is most strongly pronounced. Moreover, for $K=2$ we even see that the value of the criterion is even higher than a whole range of both even and odd values values for $K \in \{1, 3, \ldots, 23\}$. This high value of the criterion for $K=2$ should be related to the standard error of the attribute dispersions.

Since we cannot rely on the central limit theorem for particularly the region of low values for K, we conjecture that the attribute dispersions within each neighborhood have a distribution with positive skew. Then, attribute dispersions that have larger standard errors, are also more likely to have larger values. Since, for higher values of λ the softmin over the robust attribute dispersions starts to resembles the mean, a stronger influence is to be expected from the larger attribute dispersions, amplified by the larger standard errors.

Even though the standard error of the attribute dispersions for an odd K is larger (based on one attribute distance), than even $K+1$ (based on two attribute distances), the larger standard errors can only exercise a strong influence when the robust attribute dispersion is also expected to be large. Since the robust dispersions for even K have the tendency to be larger, as formulated in equation (3.15), we can expect that large standard errors have stronger influence on the robust attribute dispersions for even K. Thus, in the region of high values for λ, and for lower values of K, it is likely to obtain a higher value of the criterion (the sum of softmin attribute dispersions)

for an even value of K, e.g. $K = 2$, as compared to its higher values of K, which is ascribed to the amplifying influence of the standard error.

However, as can be seen in the left panel of Figure 3.11, the opposite effect is likely to occur for the value of the criterion in the region of the lower values of λ. For a low value of λ, the zigzag pattern over K appears to be less pronounced, and less consistent. The reason is that in the region of lower value of K, we see that a lower K and a lower λ, causes *smaller* standard errors for the softmin of the attribute dispersions.

The influence of the larger attribute dispersions is mitigated by lower values of λ in the softmin estimate, resulting in a less pronounced zigzag pattern is less pronounced. Although counter-intuitively we would expect a larger standard error for a minimum, than for the mean of the attribute dispersions, this is not necessarily the case in the region of the lower values of K. In fact, the opposite may hold true. Due to the lower bound of 0, and the positive skew of the attribute dispersions for lower K, there is not much more variation left for the minimum of the robust attribute dispersions. However, the larger K, the more variation there is possible for the minimum of the attribute dispersions, therefore we see that the zigzag pattern can break for the higher values of K.

In this section we have demonstrated the behavior of the criterion for noise-only data. When the data would consist of a clustering structure, then there are more 'outlying' attribute dispersions that are based on neighborhoods that consist of incorrect and correct neighbors (see Figure 3.5 for the example distribution of an attribute distance between an object with an incorrect neighbor). These outlying robust attribute dispersions also amplify the zigzag pattern in similar vein as is done by higher values of λ. In conclusion, the COSA-KNN criterion that is based on the robust attribute dispersions shows a zigzag pattern as a function of K, which is amplified by λ or by a clustering structure, but, when plotting the criterion as function of only even or odd K, the behavior of the robust criterion in equation (3.10) is more similar to that of the non-robust criterion in equation (3.9).

3.3 Tuning λ and K with the Gap statistic

The robust version of COSA-KNN showed to be faster (in Section 3.1.1) and better (in Section 3.1.2) in capturing the clustering structure, as compared to its non-robust version. However, we have seen that in the presence of a subtle structure in the data, a method for choosing the values of λ and K is required to obtain a good performance of COSA-KNN. In this section we will work with robust version of COSA-KNN, and propose to find the values of the tuning parameters via a procedure that is based on the criterion: the Gap statistic (Tibshirani et al., 2001). The Gap statistic was originally developed for selecting the number of clusters in standard K-means clustering algorithms. However, the procedure also got successfully implemented in Witten and Tibshirani (2010), and Arias-Castro and Pu (2017) for closely related tuning-problems.

The idea of the Gap statistic is to compare the value of $\log(\widetilde{Q}(\lambda, K))$ on observed

data with the expected value of the criterion for an appropriate null reference model (Tibshirani et al., 2001). Define the Gap statistic as

$$\text{Gap}_{\log}(\lambda, K) = E_{\widetilde{Q}^\circ}\left[\log\left(\widetilde{Q}^\circ(\lambda, K)\right)\right] - \log\left(\widetilde{Q}(\lambda, K)\right), \tag{3.17}$$

where $E_{\widetilde{Q}^\circ}\left[\log\left(\widetilde{Q}^\circ(\lambda, K)\right)\right]$ denotes the expectation of the criterion for data sets that originate from an appropriate null reference model. We estimate the expectation by taking the average from B copies of $\log\left(\widetilde{Q}^\circ(\lambda, K)\right)$, where each copy is computed based on a Monte Carlo sample drawn from the reference distribution.

We draw samples from the reference distribution for $\log\left(\widetilde{Q}^\circ(\lambda, K)\right)$ by computing the COSA-KNN criterion on B permuted data sets $\mathbf{X}_1^\circ, \ldots, \mathbf{X}_B^\circ$. Each permuted data set is generated by independent permutations of the observations within each attribute. This renders correlated attributes in the original data to become uncorrelated in the permuted data sets. Thus, the Gap statistic quantifies the strength of the clustering that is obtained on the real data set, compared to the clustering result that is obtained from data sets that do not contain any clustering structure. Eventually, we select the values of λ and K for which the Gap is largest, or the smallest Gap that is within one standard error (1SE) of the largest Gap, and corresponds to a higher value for λ and K.

3.3.1 1SE Rule and the Simpler COSA-KNN Model

Although the one standard error (1SE) rule was originally proposed for a different tuning parameter problem (Breiman, Friedman, Stone, & Olshen, 1984), its purpose remains similar in the Gap statistic procedure. With the 1SE rule we wish to choose tuning parameter values of a simpler model that would still be comparable to the optimal model.

We define the COSA-KNN model to become simpler for higher values of λ or larger values of K, which can be conceptually explained as follows. For higher values of λ, the more difficult it will become to find unique subsets of attributes for each cluster, thus the fewer 'degrees of freedom' for the candidate solutions of the subsets of attributes. Similarly, for larger values of K, the COSA distances will have more difficulty (less 'degrees of freedom') in capturing a clustering structure of smaller clusters. In the most extreme case where $\lambda \to \infty$ and $K \to N - 1$, the COSA distances will just become a 'simple' sum over the attribute distances, e.g., the ordinary Manhattan distances. Thus, we define the COSA model to be more 'complex', since it has more degrees of freedom for smaller λ and smaller K to represent the more more complex clustering structures – smaller clusters with each their own unique subset of important attributes.

Let λ^* and K^* be the values for λ and K that correspond with the largest Gap statistic, then, we select the smallest Gap statistic that corresponds with $\lambda \geq \lambda^*$ and $K \geq K^*$, and is within the range of 1SE of the largest Gap statistic. Note that in general for these higher values of λ and K, the standard error of the criterion will be lower, and therefore the variance of the Gap statistic will also be lower (since these

are simpler COSA-KNN models). Thus, another way to look at the 1SE rule is that it acknowledges the error with which the maximum Gap statistic itself is estimated.

3.3.2 With or Without the Natural Logarithm

The original version of the Gap statistic, in (3.17), is based on the natural logarithm. Mohajer, Englmeier, and Schmid (2011) corroborated the findings by Dudoit and Fridlyand (2002) that this particular use of the logarithm renders the procedure to prefer more complex models (by overestimating the number of clusters), compared to the Gap statistic that is not based on the natural logarithm, i.e.

$$\text{Gap}(\lambda, K) = E_{\widetilde{Q}^\circ}\left[\widetilde{Q}^\circ(\lambda, K)\right] - \widetilde{Q}(\lambda, K). \tag{3.18}$$

Moreover, Mohajer et al.(2011) demonstrated a proof, for criteria that have a monotone relationship with their tuning parameters, that there are situations in which it becomes impossible to find the optimal Gap statistic in the original procedure. Their proof also shows that whenever the original $\text{Gap}_{\log}(\lambda, K)$ results in a solution, this solution will always be a possible solution with $\text{Gap}(\lambda, K)$ as well, but the reverse is not necessarily true. However, it is not clear whether the proof of Mohajer et al. (2011) also holds for $\widetilde{Q}^\circ(\lambda, K)$.

The motivation Tibshirani et al. (2001) provide for taking the logarithm, seems solely based on interpretation reasons from likelihood theory. For those cases where the Gap statistic is applied on results from a K-means clustering algorithm, the Gap statistics behaves as a likelihood-ratio statistic based on mixture models (e.g. Scott and Symons, 1971). However, this is not an advantage we know how to exploit for COSA, nor does the logarithm provides us with computational advantages. Nevertheless, we will compare both versions of the Gap statistic: with and without using the logarithm in equations (3.17) and (3.18), respectively.

3.3.3 The Algorithm for the Gap statistic procedure

For computing the Gap statistic with COSA-KNN, the following steps are required:

1. Compute the criterion obtained by performing COSA-KNN on the data $\mathbf{X}$ for each candidate combination of the tuning parameter values K and λ.

2. Obtain permuted datasets $\mathbf{X}_1^\circ, \ldots, \mathbf{X}_B^\circ$ by independently permuting the observations within each attribute.

   (a) For $b = 1, 2, \ldots, B$, compute $\log\left(\widetilde{Q}_b^\circ(\lambda, K)\right)$, the criterion obtained by performing COSA-KNN with candidate tuning parameter values λ and K on the data $\mathbf{X}_b^\circ$.

   (b) Compute

$$\text{Gap}_{\log}(\lambda, K) = \frac{1}{B}\sum_{b=1}^{B} \log\left(\widetilde{Q}_b^\circ(\lambda, K)\right) - \log\left(\widetilde{Q}(\lambda, K)\right), \tag{3.19}$$

or

$$\text{Gap}(\lambda, K) = \frac{1}{B}\sum_{b=1}^{B} \widetilde{Q}_b^{\circ}(\lambda, K) - \widetilde{Q}(\lambda, K). \tag{3.20}$$

3. Choose λ^* and K^* corresponding to the largest value of $\text{Gap}(\lambda, K)$. Then, choose the simplest COSA-KNN model (smallest standard error) that is within range of one standard error of the value of $\text{Gap}(\lambda^*, K^*)$. We assure in the computation of the standard error that it additionally takes into account the simulation error, i.e.

$$\text{se}_{\log(\widetilde{Q}^{\circ}(\lambda,K))} = \sqrt{\left(1 + \frac{1}{B}\right) \text{VAR}_b\left(\log\left(\widetilde{Q}_b^{\circ}(\lambda, K)\right)\right)}, \tag{3.21}$$

for $\text{Gap}_{\log}(\lambda, K)$, or as

$$\text{se}_{\widetilde{Q}^{\circ}(\lambda,K)} = \sqrt{\left(1 + \frac{1}{B}\right) \text{VAR}_b\left(\widetilde{Q}_b^{\circ}(\lambda, K)\right)}, \tag{3.22}$$

for $\text{Gap}(\lambda, K)$.

In the text that will follow, the 'Gap statistic' may have two meanings. Either the Gap statistic refers to the one that is computed in equation (3.20), or it will be clear from the context that we may refer to the Gap statistic in general, i.e., both versions of the Gap statistic. Moreover, we refer to the Gap statistic as computed in equation (3.19), as the $\text{Gap}_{\log}$ statistic.

3.4 Applications of the Gap Statistic

In this section we will apply the Gap Statistic procedure to three different data sets to demonstrate its use and effectiveness. Since we apply the Gap statistic to the robust criterion of COSA-KNN, we need to take into account the zigzag pattern over the odd and even values of K. Remember that the standard errors are higher for attribute dispersions of even K, compared to attribute dispersions on odd K. In a data set with a clustering structure we can even expect this inequality to become larger due to larger possible attribute distances in neighborhoods consisting of a mix of correct and incorrect neighbors. Assuming that the observed data contains a detectable clustering structure, the zigzag pattern is expected to be more strongly present in the criterion $\widetilde{Q}(\lambda, K)$, than in $\widetilde{Q}^{\circ}(\lambda, K)$. Therefore the zigzag pattern propogates in the Gap statistics as well, where larger Gap statistics are to be expected for even K, compared to odd K. More about the zigzag behavior will follow in a demonstration of the use of the Gap statistic on a real data example.

In Section 3.4.2 we give an example that shows how the Gap statistic procedure contributes in revealing a subtle clustering structure that would not have been revealed using COSA-KNN with default values for λ and K. That the Gap statistic does not steer COSA-KNN to find a clustering at all costs, is shown with a noise-only data example in Section 3.4.3. We will also describe and compare the results for both versions Gap and $\text{Gap}_{\log}$ of the Gap statistic procedure.

3.4.1 ApoE3 Data Example: Mind the Gap ZigZag

To show the zigzag pattern with the Gap statistic procedure, we apply the robust version of COSA-KNN together with the Gap statistic on a small metabolomics data set from the study by Damian et al. (2007) about a Apolipoprotein variant 'E3'. Therefore, we refer to this data set as ApoE3. The ApoE3 data set consists of $P = 1550$ LC-MS (liquid chromatography-mass spectrometry) measurements of plasma lipids on $N = 38$ mice, of which 18 were transgenic, and 20 were wild type mice.

Since the zigzag pattern in the robust criterion of COSA-KNN is more strongly visible for small K, we can expect to also see zigzag behavior over odd and even values for K for the Gap Statistics. Still, even with the zigzag pattern we can see that the Gap statistic contributes towards a better separation of the clusters, although COSA-KNN also captures the clustering structure in this data set when the default tuning parameter values would have been used. Figure 3.12 depicts the default COSA-KNN results; using the default values of the tuning parameters we obtain a good separation of the wild type mice from the transgenic mice. Both the multidimensional scaling and the hierarchical clustering visualization show that the wild type mice are more homogeneous as a group than the transgenic mice.

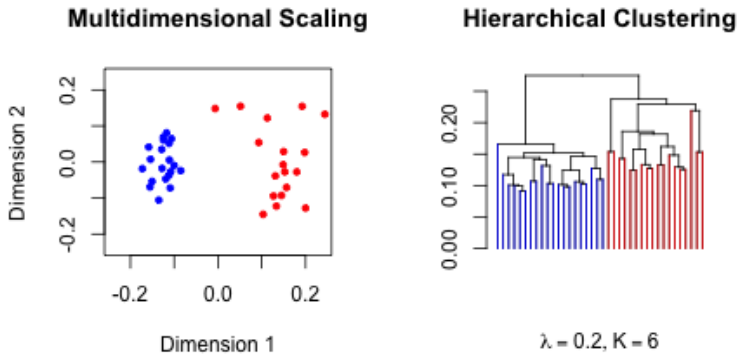

Figure 3.12: The visualization by nultidimensional scaling (left panel) and an average linkage dendrogram (right panel) of the COSA-KNN distances for the default tuning parameters $\lambda = 0.2$ and $K = 6$. The red objects are the transgenic mice and the blue objects are the wild type mice.

Step 1: Set the Grid of Tuning Parameter Combinations

We start by computing the (robust) criterion of COSA-KNN over a grid of λ and K. Since the data set is small and not much computing time is required, we will use a dense grid consisting of all combinations for

$$\lambda \in \{0.01, 0.025, 0.05, 0.075, \ldots, 0.3, 0.35, 0.4\}, \text{ and}$$

$K \in \{1, 2, \ldots, 24\}$.

A visualization of the criterion $\widetilde{Q}(\lambda, K)$, as well as the $\log \widetilde{Q}(\lambda, K)$, is shown in Figure 3.13. As expected, the value of the criterion is systematically higher for even K than for odd K.

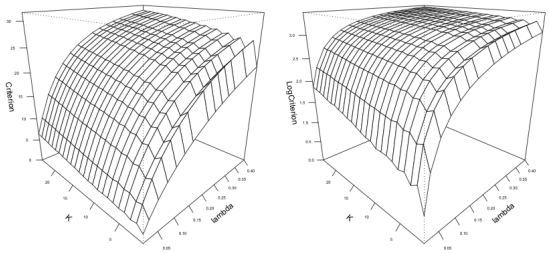

Figure 3.13: The robust criterion $\widetilde{Q}$ (left panel) and the logarithm of the robust criterion $\log \widetilde{Q}$ (right panel) depicted as a function λ and K.

Step 2: Compute the Gap Statistics

In the second step of the Gap statistic procedure we compute the estimates of the expectation of $\widetilde{Q}^{\circ}$ and $\log \widetilde{Q}^{\circ}$ for each combination of λ and K based on the B Monte Carlo reference data sets. The results can be found in Figure 3.14. Compared to the value of the criterion for the data, the expected value of the criterion of the null reference model, $E_{\widetilde{Q}_b^{\circ}}\left[\widetilde{Q}_b^{\circ}\right]$, is higher for most combinations of λ and K, and it shows a slightly stronger 'zigzag' tendency between even and odd values for K. A similar pattern is found for the logarithm of the value of the criterion for the data set and that of the expected value of the log criterion of the reference model.

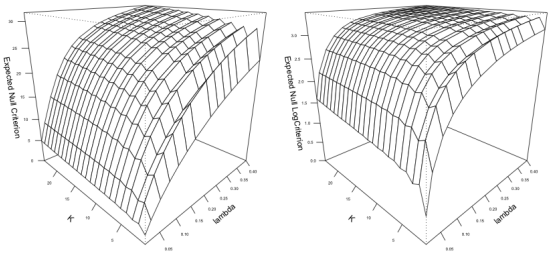

Figure 3.14: The computed expectation of the robust criterion $\widetilde{Q}^{\circ}$ (left panel), and its natural logarithm $\log \widetilde{Q}^{\circ}$ (right panel), depicted as a function λ and K.

Since the zigzag tendency is more strongly pronounced in the values for the null reference criterion, both $\text{Gap}(\lambda, K)$ and $\text{Gap}_{\log}(\lambda, K)$ have higher values for even K compared to odd K, as can be seen in Figure 3.15. When we visualize the Gap statistics as a function of odd or even values of K only, then the zigzag pattern disappears, as can be seen in Figure 3.16. Apart from the zigzag pattern, the Gap statistic seems to be unimodal for λ while keeping K fixed, and unimodal for most K while keeping λ fixed.

Our findings for the optimal tuning parameters with Gap and $\text{Gap}_{\log}$ are consistent with the results from Mohajer et. al (2011). We see that the $\text{Gap}_{\log}$ prefers the slightly more complex COSA-KNN models, i.e. lower values of λ and smaller numbers of K receive high values for $\text{Gap}_{\log}$, as compared to the values of Gap. Also, the maximum value for $\text{Gap}_{\log}$ is for a lower λ and smaller K, than the maximum value of Gap.

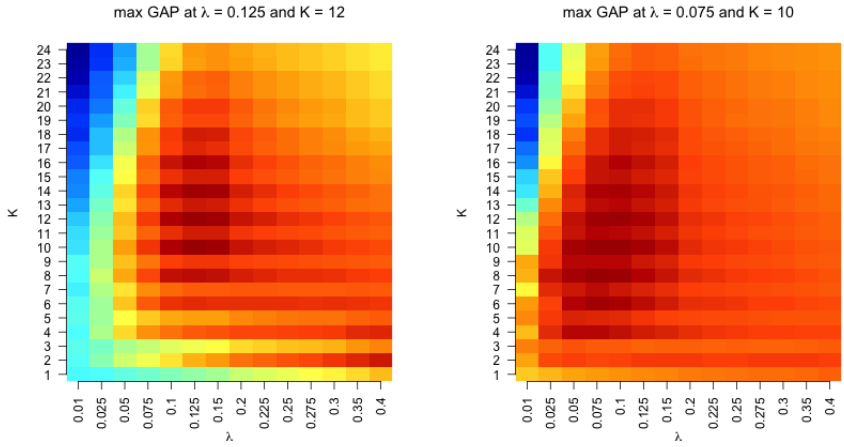

Figure 3.15: A heat map of the Gap statistics $Gap(\lambda, K)$ (left) and $Gap_{\log}(\lambda, K)$ (right) for the ApoE3 data. A dark red color corresponds to a high value of the Gap statistic, whereas the dark blue color corresponds to a low value of the Gap statistic.

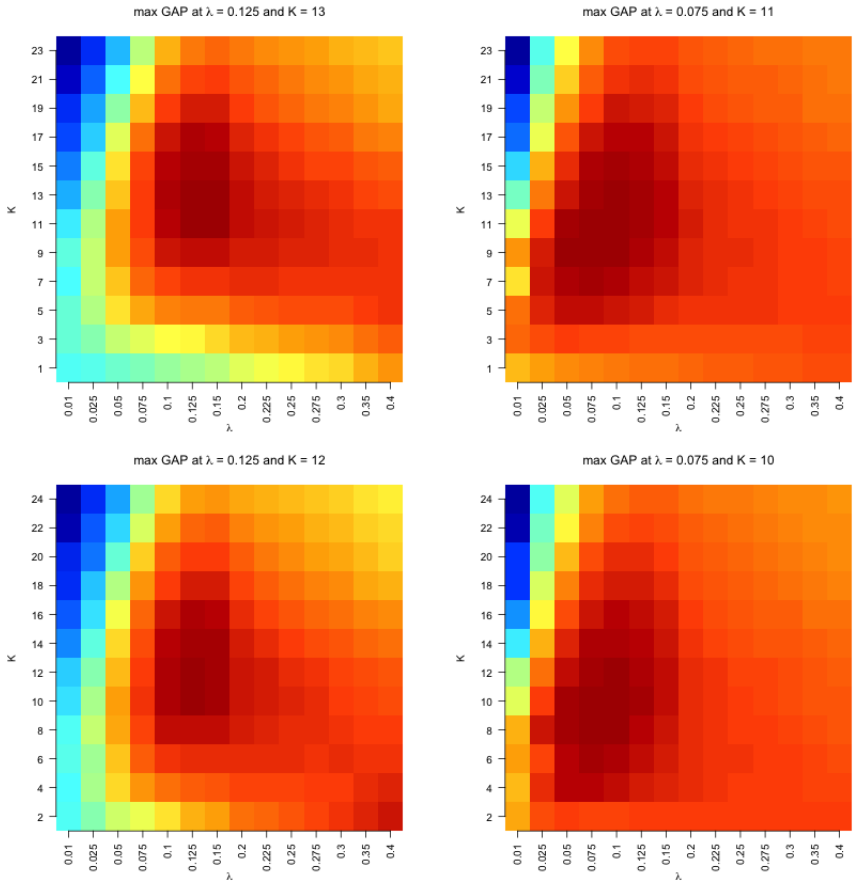

Figure 3.16: A heat map of the Gap statistics $Gap(\lambda, K)$ (left) and $Gap_{\log}(\lambda, K)$ (right) for the ApoE3 data. The top panels depict the Gap statistics for only odd values for K, the bottom panels depict only even values for K. A dark red color corresponds to a high value of the Gap statistic, whereas the dark blue color corresponds to a low value of the Gap statistic.

Step 3: Choose the Optimal Tuning Parameter Values

In the Tables 3.1 and 3.2, the eight highest Gap statistics for Gap and $\text{Gap}_{\log}$, respectively, are in ascending order. First, we will describe the results of the eight highest values for the Gap statistics.

Table 3.1 shows that there are only even numbers for K selected as good candidates in the region of $10 \leq K < 16$. The maximum value for the Gap statistic, $\text{Gap}(K = 12, \lambda = 0.125) = 2.6091$, occurs at $K^* = 12$ and $\lambda^* = 0.125$. Using Breiman's one standard error rule, we should select the values for K and λ of the simplest COSA-KNN model with a Gap statistic value higher than 2.5357. From the simpler models we can either select $\lambda = 0.150$ with $K = 10$, since it has the lowest standard error, or the model where $\lambda = 0.150$ and $K = 12$, since it is the model with the lowest standard error that has higher values on both the tuning parameters. Keeping in mind that the standard errors are also estimations, and that in general higher values for K and λ correspond with lower standard errors, we select the values $\lambda = 0.150$ and $K = 12$ to represent the simplest model within the range of one standard error.

Table 3.1: The eight highest Gap statistics for the ApoE3 data in ascending order: $Gap(\lambda^*, K^*) - se_{\widetilde{Q}^\circ(\lambda^*, K^*)} = 2.5357$. The red Gap statistic value corresponds to λ^* and K^*; the orange Gap statistic value corresponds to the chosen simplest model that is within the range of one standard error.

λ	K	$Gap(\lambda, K)$	$\widetilde{Q}(\lambda, K)$	$E_b\widetilde{Q}_b^\circ(\lambda, K)$	$\text{se}_{\widetilde{Q}^\circ(\lambda,K)}$
0.125	16	2.4760	20.6368	23.1129	0.0792
0.100	12	2.4801	17.6645	20.1446	0.0842
0.150	14	2.5114	21.8847	24.3961	0.0618
0.150	10	2.5384	20.8790	23.4174	0.0625
0.150	**12**	**2.5444**	21.4148	23.9592	0.0650
0.125	15	2.5480	20.2153	22.7634	0.0707
0.125	10	2.5932	19.2494	21.8427	0.0770
0.125.	**12**	**2.6091**	19.7452	22.3543	0.0735

In Figure 3.17 we depict the COSA-KNN results for the tuning parameters that rendered the maximum Gap statistic, as well as the results for the values of λ and K, based on the one standard error rule. Compared to the results obtained from the default tuning parameter settings of COSA-KNN, the clusters are now better separated, mainly due to a more homogeneous wild type mouse group (depicted in blue).

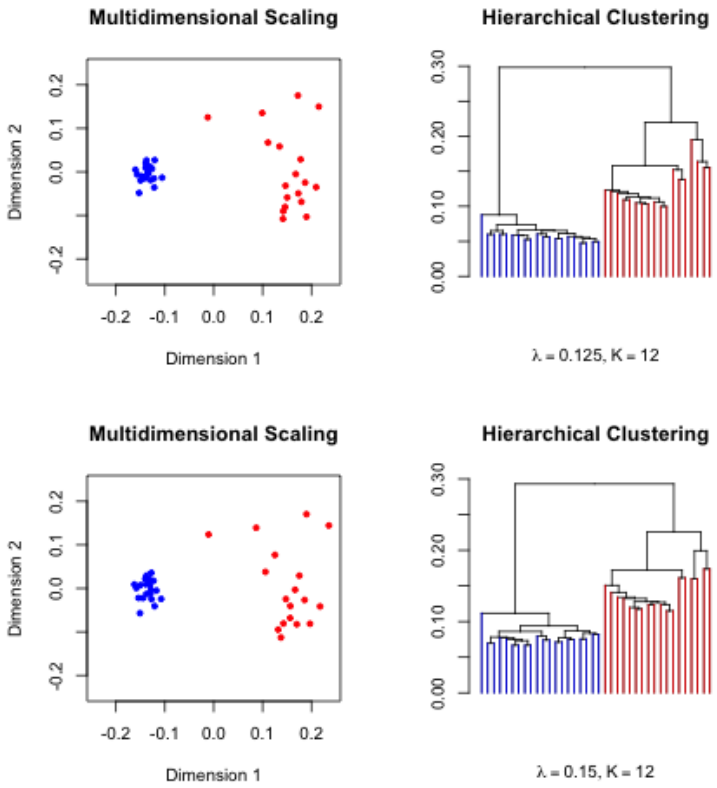

Figure 3.17: The multidimensional scaling and hierarchical clustering visualizations of the COSA-KNN results for λ^* and K^* (in the top row), and the values for λ and K of the simplest model within the range of one standard error (bottom row).

Compared to the Gap statistics, the values for $\text{Gap}_{\log}$ steer towards candidate values of $8 \leq K \leq 12$ and $0.075 \leq \lambda \leq 0.100$; more complex COSA-$K$NN models. In Figure 3.18 we depict the COSA-KNN results for λ^* and K^* based on the $\text{Gap}_{\log}$ values, as well as the results for the candidate values for λ and K based on the one standard error rule. In these visualizations, the wild type mice are more homogeneous than was shown in Figure 3.12. Instead of being a large group of dispersed objects, the transgenic mice group seems to get separated into subgroups, or at least one core subgroup. Next to these results that we demonstrated, similar results were obtained when the Gap statistic procedure was applied on a grid of only odd values for K (Appendix 3.6.1).

Table 3.2: Eight highest Gap statistics of the log criteria of the ApoE3 data in ascending order: $Gap_{\log}(\lambda^*, K^*) - se_{\log \widetilde{Q}^\circ(\lambda^*, K^*)} = 0.1301$. *The red Gap statistic value corresponds to* λ^* *and* K^**; the orange Gap statistic value corresponds to the chosen simplest model that is within the range of one standard error.*

λ	K	$\text{Gap}_{\log}(\lambda, K)$	$\log \widetilde{Q}(\lambda, K)$	$E_b \log \widetilde{Q}_b^\circ(\lambda, K)$	$\text{se}_{\log \widetilde{Q}^\circ(\lambda, K)}$
0.050	10	0.1280	2.4226	2.5505	0.0074
0.075	6	0.1304	2.6141	2.7445	0.0055
0.075	12	0.1306	2.7109	2.8415	0.0054
0.100	**12**	**0.1314**	2.8716	3.0029	0.0042
0.100	12	0.1331	2.8154	2.9486	0.0045
0.100	10	0.1338	2.8452	2.9791	0.0038
0.075	8	0.1345	2.6526	2.7871	0.0051
0.075	**10**	**0.1357**	2.6822	2.8179	0.0055

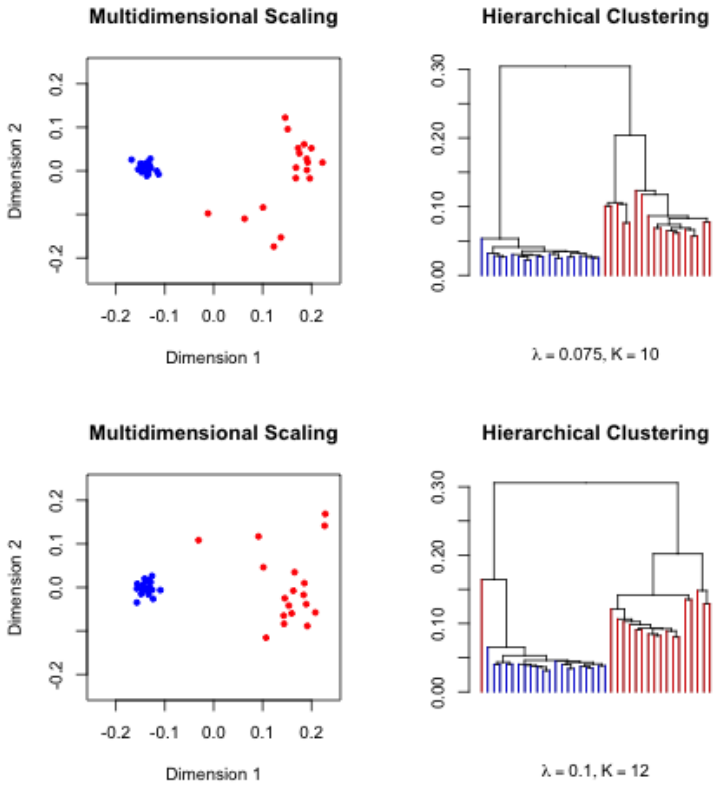

Figure 3.18: The multidimensional scaling and hierarchical clustering visualizations of the COSA-KNN results based on $Gap_{\log}$ *for* λ^* *and* K^* *(in the top row), and the values for* λ *and* K *of the simplest model that is within the range of one standard error of the value for* λ^* *and* K^* *(bottom row).*

Conclusion Gap Statistic on the ApoE3 Data

The application of the Gap statistic procedure on COSA-KNN results of the ApoE3 data, reveals a sharper clustering structure than the structure would have revealed with the usual choices of $\lambda = 0.2$ and $K = \sqrt{N}$. This holds for all four solutions based on either the maximum or the one standard error rule for Gap, and for Gap_{log}. There are small differences between the solutions. The Gap_{log} statistics seem to prefer COSA-KNN models for higher λ, and higher K; the more complex models with higher standard errors. Since we don't know the full truth for this data set, it remains inconclusive whether these models are better or worse than the ones proposed by the Gap statistic where the use of the logarithm is omitted.

Because we are working with the robust version of the COSA-KNN, the highest Gap statistic values will occur for even K for this data example. When taking into account the Gap statistic for only odd K, the highest Gap statistics occur in the similar regions for K. Not being aware of this artifact could result in specifying a grid for K and λ which makes it more prone to end up in a local maximum for the Gap statistic. Especially in the case of small N data sets that results in a grid with small values for K, and complex dependency structures in the data. E.g., for this data set, a grid for $K \in \{2, 13, 24\}$ results in $K = 2$. Since it is difficult to determine whether N is large enough, and moreover, since cluster sizes for this particular unsupervised setting are unknown, we would advise to only work with a grid where the candidate values for K are only even, or odd.

3.4.2 The Gap statistic on Subtle Structure Data

To demonstrate the power of the Gap statistic procedure, we show how it can steer COSA-KNN towards good candidate values of the tuning parameters for data generated from subtle structure model in Section 3.1.2 (see Figure 3.7). As was shown already, when we visualize the distances from the default settings of COSA-KNN, i.e. $\lambda = 0.2$ and $K = \sqrt{N} = 10$, we cannot retrieve the grouping structure by hierarchical clustering (with average, single, complete and ward linkage), nor after visualizations of results obtained by Multidimensional Scaling. See Figure 3.19 for the two types of visualizations of robust COSA-KNN results for the usual tuning parameter values.

Since we know the clustering, we could argue that the objects in the MDS visualization (in red and blue) are closer to each other than the noise objects (grey). However, without the coloring, this would not have been clear.

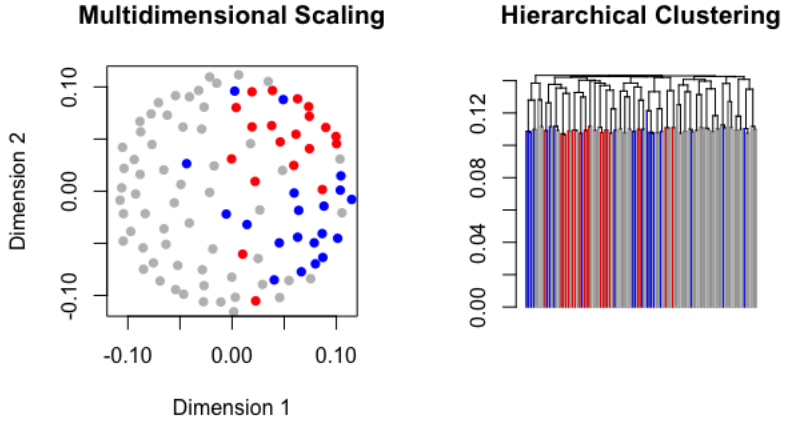

Figure 3.19: A visualization of the multidimensional scaling configuration (left panel) and the average linkage dendrogram of COSA-KNN distances with default parameter settings $\lambda = 0.2$ *and* $K = 10$.

Applying the Gap Statistic Procedure

To find the subtle structure in the data set, we compute the COSA-KNN criterion value, and $B =$ null reference criteria for each combination of

$$\lambda \in \{0.01, 0.05, .075, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40\}, \text{ and}$$

$$K \in \{6, 8, 12, 18, 26, 36, 48\}.$$

In Figure 3.20 we show the heatmap of the resulting Gap statistic values for Gap and $\text{Gap}_{\log}$, and in the Tables 3.3 and 3.4, the relevant information of the six highest Gap and $\text{Gap}_{\log}$ statistic values is given. The largest value for the Gap statistic is obtained for $\lambda^* = 0.075$ and $K^* = 18$. There is only one simpler COSA-KNN model within the range of one standard error ($\text{se}_{\widetilde{Q}^\circ(\lambda^*, K^*)} = 0.1713$); the model with $\lambda = 0.075$ and $K = 26$ at a Gap of 2.3429, see Table 3.3. The highest value for the $\text{Gap}_{\log}$ statistic is found for the combination $\lambda^* = 0.05$ and $K^* = 18$, with no simpler model in the range of one standard error.

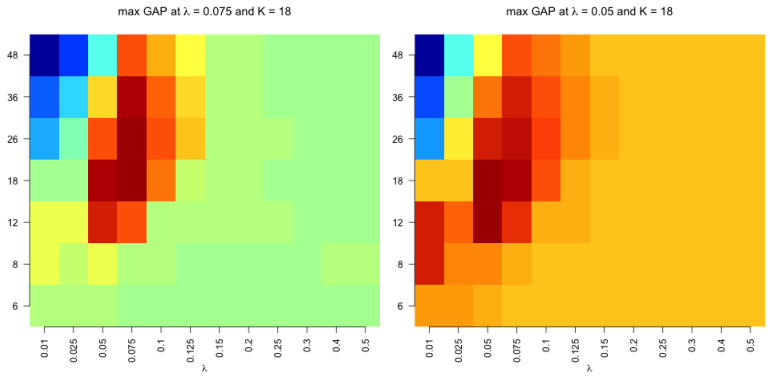

Figure 3.20: A heatmap of the Gap values (left) and the $Gap_{\log}$ values for the Example 2 data as a function of the grid for λ and K.

Table 3.3: Top 6 highest Gap statistics in ascending order: $Gap(\lambda^*, K^*) - se_{\widetilde{Q}^\circ(\lambda^*,K^*)} = 2.2144$. The red Gap statistic value corresponds to λ^* and K^*; the orange Gap statistic value corresponds to the chosen simplest model that is within the range of one standard error.

λ	K	$\mathrm{Gap}(\lambda, K)$	$\widetilde{Q}(\lambda, K)$	$E_b\widetilde{Q}_b^\circ(\lambda, K)$	$\mathrm{se}_{\widetilde{Q}^\circ(\lambda,K)}$
0.100	26	1.6792	66.8569	68.5361	0.0386
0.050	12	1.9360	43.0336	44.9696	0.1848
0.075	36	2.1513	64.3225	66.4738	0.0823
0.050	18	2.2413	46.7598	49.0010	0.1633
0.075	**26**	**2.3429**	61.0269	63.3699	0.1095
0.075	**18**	**2.3858**	57.2769	59.6627	0.1713

Table 3.4: Top 6 highest Gap_{log} statistics in ascending order: $Gap_{\log}(\lambda^*, K^*) - se_{\log\widetilde{Q}^\circ(\lambda^*,K^*)} = 0.0435$. The red Gap_{log} statistic value corresponds to λ^* and K^*.

λ	K	$\mathrm{Gap}_{\log}(\lambda, K)$	$\log\widetilde{Q}(\lambda, K)$	$E_b \log\widetilde{Q}_b^\circ(\lambda, K)$	$\mathrm{se}_{\log\widetilde{Q}^\circ(\lambda,K)}$
0.010	8	0.0330	2.4676	2.5006	0.0190
0.010	12	0.0332	2.5307	2.5638	0.0199
0.075	26	0.0377	4.1113	4.1490	0.0017
0.075	18	0.0408	4.0479	4.0887	0.0029
0.050	12	0.0440	3.7620	3.8060	0.0041
0.050	**18**	**0.0468**	3.8450	3.8918	0.0033

The Results

From Figure 3.21 we can see that successful COSA-KNN results can be obtained for both combinations that are proposed with the Gap statistic; the values of λ^* and K^* that correspond to the maximal Gap statistic, and the one standard error rule values for λ and K. Hardly any difference can be seen between the dendrograms or multidimensional scaling configurations for $\{\lambda = 0.075, K = 18\}$ and $\{\lambda = 0.075, K = 26\}$, which shows that with the one standard we still obtain successful results that can be interpreted as more stable (due to the lower standard error of the criterion).

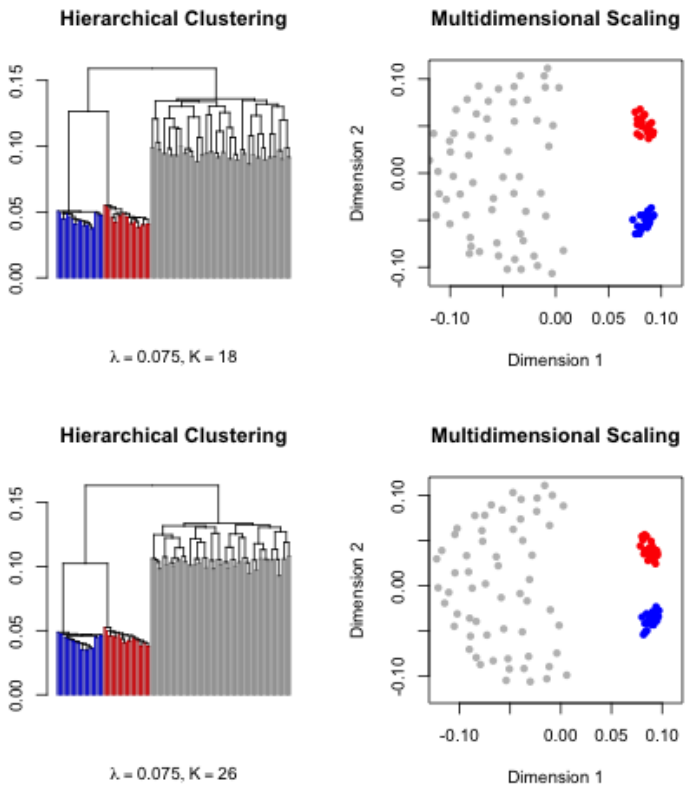

Figure 3.21: The Multidimensional Scaling and Hierarchical Clustering visualizations of the COSA-KNN results based on Gap for K^* and λ^* (in the top row), and the values for λ and K of the simplest model that is within the range of one standard error of the value for λ^* and K^* (bottom row).

The COSA-KNN results for the values of $\{\lambda = 0.050,\ K = 18\}$, suggested by the Gap_{log} procedure, show less desirable results, as can be seen in Figure 3.22. Although we still would be able to cut the dendrogam into the two groups and 60 singleton noise objects, the clustering structure is not optimally revealed, some of the noise objects

seem to be close to the clustered objects. While the results may be worse compared to the suggested values of the tuning parameters based on the Gap statistic that omits the logarithm, the results are still better when compared to those obtained with the usual tuning parameter values (Figure 3.19)

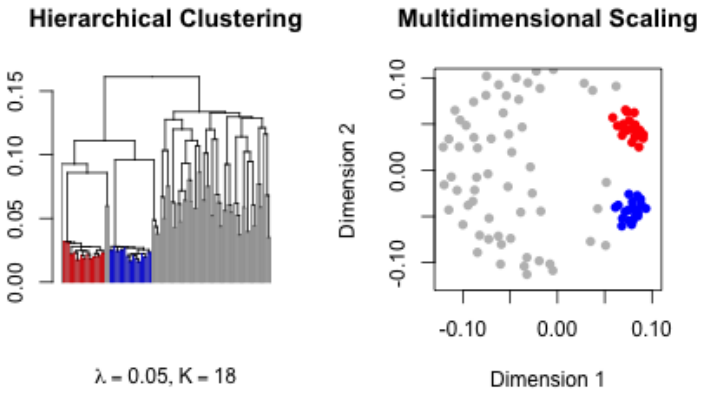

Figure 3.22: Visualization of the COSA distances with optimized tuning parameters $\lambda = 0.05$ *and* $K = 18$*, selected by the* Gap_{log}*-statistic procedure.*

Conclusion Gap Statistic on the Subtle Structure Data

When we apply the Gap statistic procedure to the COSA-KNN criterion for a data set that is generated from the subtle structure model, we can find the values for λ and K with which we can reveal the clustering structure. The results are better for the optimized values of the tuning parameters based on the Gap statistic, than for those based on the Gap_{log}. In accordance with the findings of Mohajer et al. (2011), we see that with the Gap_{log} statistic a lower value of λ (and K) is preferred, that in general corresponds with higher standard errors of the criterion. Thus, a more complex model is preferred with the Gap_{log} statistic compared to the Gap statistic.

3.4.3 The Gap Statistic and Noise Only Data

The Gap statistic renders COSA-KNN to be more powerful by steering towards good candidate values for λ and K. One could argue that the use of Gap statistic, it also renders COSA-KNN to be more prone to a Type-I error, i.e. showing a clustering structure that is supported on only random fluctuations. In this section we will use an example to demonstrate the behavior of the Gap statistic in combination with COSA-KNN on a data set that is supposed to be noise only.

To obtain a data set that is not supposed to contain any systematic clustering structure, we generate a data set of$N = 100$ by $P = 10,000$ i.i.d. standard normal

values. On this data set we will apply COSA-KNN in combination with the Gap statistic, using the exact same settings as in section **??**. Thus, we apply the Gap statistics to the robust COSA-KNN criterion for all tuning parameter combinations on the grid

$$\lambda \in \{0.01, 0.05, .075, 0.10, 0.15, 0.20, 0.25, 0.30, 0.40\}, \text{ and}$$

$$K \in \{6, 8, 12, 18, 26, 36, 48\}.$$

While the criterion values values for λ and K seem fine in the left and middle panel of Figure 3.23, the surface of the Gap statistic values for $\text{Gap}(\lambda, K)$ show more irregular behavior, compared to what we have seen so far. There does not seem to be a reasonable hill climbing path towards a (local) maximum value for the Gap statistic values. The same holds for the $\text{Gap}_{\log}$ statistics, of which the heatmap is visualized in the right panel of Figure 3.24. This may be a first indication that we are dealing with a data set with no clustering structure. Similarly, from the heatmaps in Figure 3.24, no clear area can be obtained that indicates a 'sweet heat spot'.

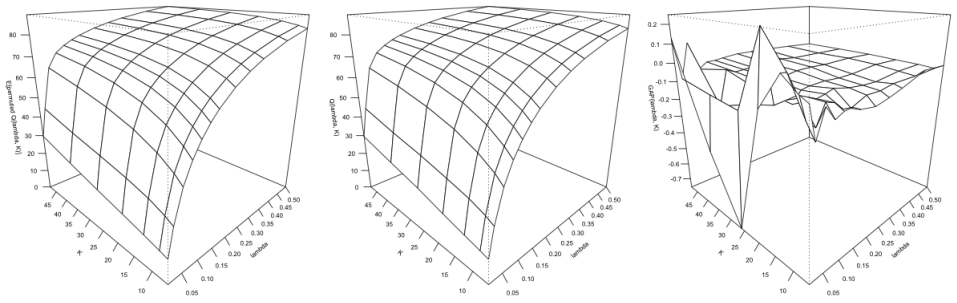

Figure 3.23: Surface of the estimated expectation of the log-criterion of the reference model (left) and the criterion (middle) of COSA-KNN for λ and K. The right panel shows the Gap statistic.

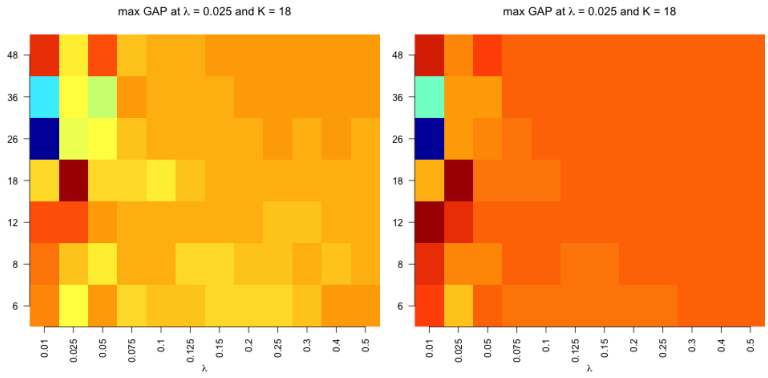

Figure 3.24: A heatmap of the Gap values (left) and the $Gap_{\log}$ values for the 'Noise Only' data as a function of the grid for λ and K.

A second indication of no clustering structure is when all Gap statistics have low (or even negative) values that are accompanied by large standard errors. In Table 3.5 we see for our specific data set that the values of the Gap statistics are lower when compared to a data that does contain a clustering structure, and has similar dimensions (e.g. the data in the previous section). Moreover, most of the highest Gap statistics have values that are within the range of one standard error from 0, indicating that the COSA-KNN with the optimized values for λ and K performs equally when compared to noise, which actually should be the case for our simulated noise data set.

Table 3.5: Top 8 highest Gap statistics in ascending order. According to the one standard error rule, we should choose the values for λ and K of the simplest possible COSA-KNN model with a Gap statistic value higher than $Gap(\lambda^*, K^*) - se_{\widetilde{Q}^\circ(\lambda^*,K^*)} = 0.040$. The red Gap statistic value corresponds to λ^* K^*; the orange Gap statistic value corresponds to the chosen simplest model that is within the range of one standard error.

λ	K	$\text{Gap}(\lambda, K)$	$\widetilde{Q}(\lambda, K)$	$E_b\widetilde{Q}_b^\circ(\lambda, K)$	$\text{se}_{\widetilde{Q}^\circ(\lambda,K)}$
0.050	12	-0.001	45.611	45.610	0.091
0.010	6	0.021	11.487	11.508	0.095
0.010	8	0.039	12.468	12.508	0.167
0.025	12	0.096	28.750	28.846	0.208
0.050	**48**	**0.097**	63.466	63.563	0.197
0.010	12	0.099	14.196	14.295	0.222
0.010	48	0.128	29.499	29.626	0.366
0.025	**18**	**0.247**	31.581	31.828	0.207

The optimal tuning parameters for λ and K are the same for both versions of the Gap statistic procedures (only the Gap statistic without the logarithm is shown here in Table 3.5 and Figure 3.25). Figure 3.25 depicts the visualizations belonging to the Gap optimized COSA-KNN results. Although, in general the dendrograms have

a stronger tendency to show a clustering structure, these structures seem to be very instable since the dendrogram for $\lambda = 0.025$ with $K = 18$ is very different from the results obtained with the 1 S.E. rule where $\lambda = 0.05$ with $K = 18$. Moreover, the visualizations of the multidimensional scaling configurations do not seem to hint at a clustering structure.

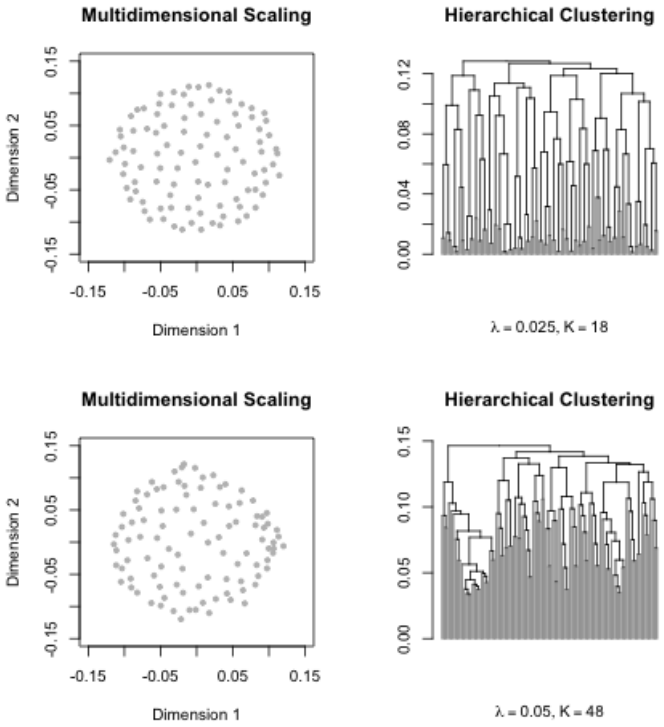

Figure 3.25: The multidimensional scaling and hierarchical clustering visualizations of the COSA-KNN results for 'noise only' data based on Gap for K^* and λ^* (in the top row), and the values for λ and K of the simplest model that is within the range of one standard error of the value for K^* and λ^* (bottom row).

Conclusion of the Use of the Gap Statistic on Noise Only

We did not observe for our noise data example that the Gap statistic could steer COSA-*K*NN towards committing a Type-I error. Moreover, there are three indications by which we could foresee such an error. First, a resulting clustering structure is very likely to be based on sampling fluctuations only if the surface of the Gap statistic values is very irregular. Second, a large standard error for each Gap statistic also indicates that the Gap may just got larger than 0 based on chance. These are two strong indications from which we can derive whether we should trust our Gap

statistic results for COSA-KNN. Last, instead of using the visualizations based on average linkage dendrograms, the visualizations of the COSA-KNN results by multi-dimensional scaling seem to provide more stable results.

3.5 Discussion

When a clustering structure is present in the data, the robust version of COSA-KNN is more successful than the non-robust version of COSA-KNN. Not only is the robust version faster in revealing the clustering structure, we have also seen an example where only the robust version can reveal the clustering structure. The number of successful candidate combinations of the tuning parameter values of λ and K, is enlarged for robust COSA-KNN.

To automatically find successful candidate combinations of the tuning parameter, we demonstrated that the application of the Gap statistic on the criterion of robust COSA-KNN showed successful results. While the non-robust criterion seems to be a concave smooth surface over a grid of values for both λ and K, the robust version of the criterion shows a zigzag pattern over the odd and even values for K. This particular zigzag pattern is also propogated in the Gap statistic values, leading towards the preference of even values for K, over odd values for K. We suggest to use a grid with preferably only even values for K.

The Gap statistic values can be computed over the robust COSA-KNN criterion directly, or on the natural logarithm of each criterion. Although for these two different procedures we did not find differences that would have resulted in different clustering structures, we did replicate earlier findings that the $\text{Gap}_{\log}$ statistic procedure has the tendency to prefer the tuning parameter values corresponding to more complex models, i.e. lower values for λ and lower values of K. Although it remains inconclusive which of the two versions of the Gap statistic procedure is better, we prefer the version that applies Occam's Razor. Thus, we will be using the Gap statistic procedure that omits the logarithm in the succeeding chapters of this monograph for COSA.

While keeping K fixed, the Gap statistic seems to be very close to a unimodal function for the values of λ. Thus, for each K we may find candidates for the optimal λ using a golden search algorithm (e.g., Brent, 1971) as was also proposed in Arias-Castro and Pu (2017) for the sparsity parameter of their algorithm. However, this extension deserves more investigation.

3.6 Appendix

3.6.1 Considering Only Odd Values K for the ApoE3 Data

Table 3.6: The five highest Gap_{log} statistics for odd K only and of the log criteria, of the ApoE3 data in ascending order: $Gap_{\log}(\lambda^*, K^*) - se_{\log \widetilde{Q}^\circ(\lambda^*, K^*)} = 0.1152$. The red Gap_{log} statistic value corresponds to λ^* and K^*; the orange Gap statistic value corresponds to the chosen simplest model that is within the range of one standard error.

λ	K	$\mathrm{Gap}_{\log}(\lambda, K)$	$\log \widetilde{Q}(\lambda, K)$	$E_b \log \widetilde{Q}_b^\circ(\lambda, K)$	$\mathrm{se}_{\log \widetilde{Q}^\circ(\lambda,K)}$
0.050	9	0.1175	2.3919	2.5095	0.0067
0.075	9	0.1178	2.6517	2.7694	0.0047
0.100	**13**	**0.1178**	2.8744	2.9922	0.0037
0.100	11	0.1186	2.8466	2.9651	0.0037
0.075	**11**	**0.1200**	2.6855	2.8055	0.0048

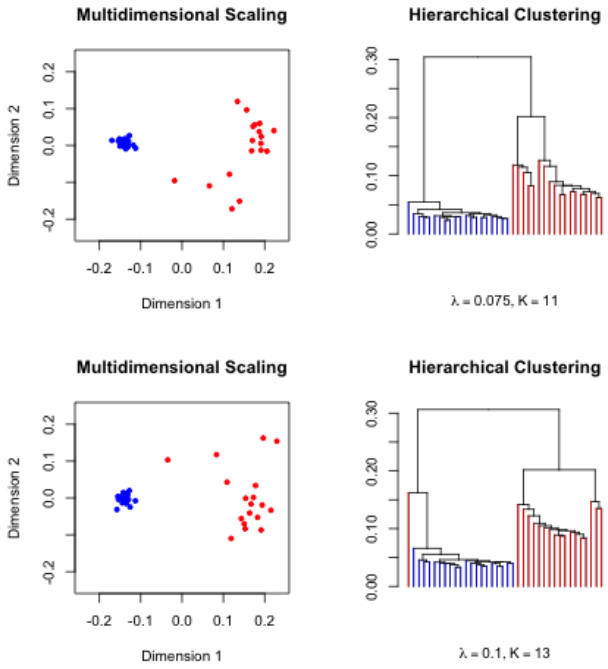

Figure 3.26: A Visualization of the COSA-KNN results that are optimized based on the $Gap_{\log}$ statistic procedure. The top panels shows the multidimensional scaling configuration and the average linkage dendrogram for the tuning parameters values that correspond to the maximum $Gap_{\log}$ statistic, the bottom panels show the visualizations for the simplest model in the range of one standard error.

Table 3.7: The seven highest Gap statistics, for odd K only, of the ApoE3 data in ascending order: $Gap(\lambda^*, K^*) - se_{\widetilde{Q}^\circ(\lambda^*,K^*)} = 2.2590$. The red Gap statistic value corresponds to λ^* and K^*; the orange Gap statistic value corresponds to the chosen simplest model that is within the range of one standard error.

λ	K	$\text{Gap}(\lambda, K)$	$\widetilde{Q}(\lambda, K)$	$E_b\widetilde{Q}_b^\circ(\lambda, K)$	$\text{se}_{\widetilde{Q}^\circ(\lambda,K)}$
0.125	17	2.2211	20.7195	22.9406	0.0701
0.150	**15**	**2.2654**	21.9253	24.1907	0.0625
0.125	15	2.2848	20.2745	22.5593	0.0673
0.150	11	2.2903	20.8549	23.1451	0.0597
0.150	13	2.3032	21.4353	23.7385	0.0611
0.125	11	2.3123	19.2336	21.5459	0.0704
0.125	**13**	**2.3322**	19.7785	22.1107	0.0732

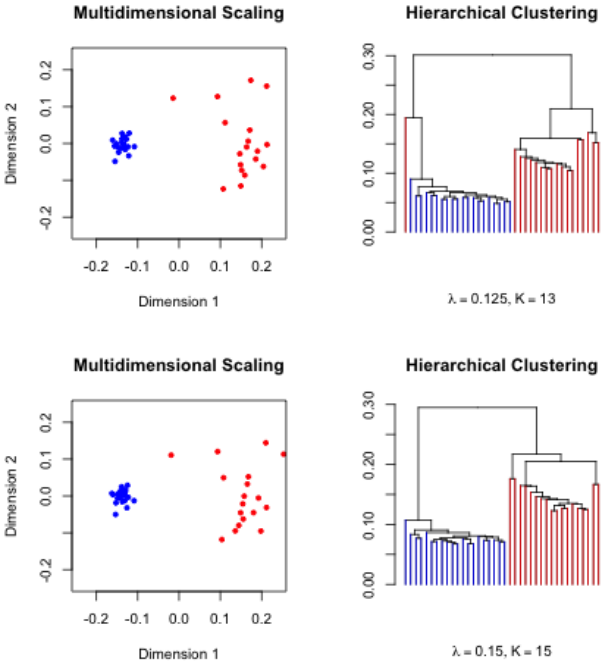

Figure 3.27: A Visualization of the COSA-KNN results that are optimized based on the Gap statistic procedure. The top panels shows the multidimensional scaling configuration and the average linkage dendrogram for the tuning parameters values that correspond to the maximum Gap statistic, the bottom panels show the visualizations for the simplest model in the range of one standard error.

3.6.2 Top 20 Gap Statistics for the ApoE3 Data

Table 3.8: The 20 highest Gap statistics for K on the criterion of the ApoE3 data in ascending order: $GAP(\lambda^*, K^*) - se_{\widetilde{Q}^\circ(\lambda^*,K^*)} = 2.5357$. Note that the first odd value for K is the 18^{th} highest Gap statistic.

λ	K	$\text{Gap}(\lambda, K)$	$\widetilde{Q}(\lambda, K)$	$E_b\widetilde{Q}_b^\circ(\lambda, K)$	$\text{se}_{\widetilde{Q}^\circ(\lambda,K)}$.
0.1250	11	2.3123	19.2336	21.5459	0.0704
0.4000	2	2.3270	26.2375	28.5646	0.0562
0.1250	13	2.3322	19.7785	22.1107	0.0732
0.2000	12	2.3402	23.8172	26.1574	0.0532
0.1000	16	2.3432	18.5401	20.8833	0.0807
0.2000	10	2.3579	23.2857	25.6436	0.0557
0.1500	8	2.3696	20.3386	22.7082	0.0578
0.1000	8	2.3785	16.7005	19.0790	0.0850
0.1250	8	2.4038	18.7388	21.1426	0.0667
0.1000	14	2.4183	18.1167	20.5350	0.0869
0.1500	16	2.4302	22.3189	24.7491	0.0691
0.1000	10	2.4639	17.2057	19.6696	0.0752
0.1250	16	2.4760	20.6368	23.1129	0.0792
0.1000	12	2.4801	17.6645	20.1446	0.0842
0.1500	14	2.5114	21.8847	24.3961	0.0618
0.1500	10	2.5384	20.8790	23.4174	0.0625
0.150	**12**	**2.5444**	21.4148	23.9592	0.0650
0.125	14	2.5480	20.2153	22.7634	0.0707
0.125	10	2.5932	19.2494	21.8427	0.0770
0.125	**12**	**2.6091**	19.7452	22.3543	0.0735

3.6.3 Top 20 $\text{Gap}_{\log}$ Statistics for the ApoE3 Data

Table 3.9: The 20 highest Gap_{log} statistics for K based on the log criteria of the ApoE3 data in ascending order: $GAP_{\log}(\lambda^*, K^*) - se_{\log \widetilde{Q}^{\circ}(\lambda^*, K^*)} = 0.1301$. Note that the first odd value for K is the 17^{th} highest Gap_{log} statistic.

λ	K	$\text{Gap}_{\log}(\lambda, K)$	$\log \widetilde{Q}(\lambda, K)$	$E_b \log \widetilde{Q}_b^{\circ}(\lambda, K)$	$\text{se}_{\log \widetilde{Q}^{\circ}(\lambda, K)}$
0.100	11	0.1186	2.8466	2.9651	0.0037
0.125	14	0.1187	3.0064	3.1251	0.0031
0.100	16	0.1190	2.9199	3.0389	0.0039
0.075	11	0.1200	2.6855	2.8055	0.0048
0.050	6	0.1203	2.3585	2.4788	0.0064
0.125	8	0.1207	2.9306	3.0513	0.0032
0.075	14	0.1214	2.7401	2.8615	0.0053
0.125	12	0.1241	2.9829	3.1070	0.0033
0.100	14	0.1253	2.8968	3.0221	0.0042
0.100	6	0.1254	2.7759	2.9013	0.0039
0.125	10	0.1264	2.9575	3.0839	0.0035
0.050	8	0.1268	2.3943	2.5211	0.0069
0.050	10	0.1280	2.4226	2.5505	0.0074
0.075	6	0.1304	2.6141	2.7445	0.0055
0.075	12	0.1306	2.7109	2.8415	0.0054
0.100	**12**	**0.1314**	2.8716	3.0029	0.0042
0.100	8	0.1331	2.8154	2.9486	0.0045
0.100	10	0.1338	2.8452	2.9791	0.0038
0.075	8	0.1345	2.6526	2.7871	0.0051
0.075	**10**	**0.1357**	2.6822	2.8179	0.0055

3.6.4 $\text{Gap}_{\log}$ for the Noise Only Data Set

Table 3.10: Top 8 highest Gap statistics in ascending order. According to the one standard error rule, we should choose the values for λ and K of the simplest possible COSA-KNN model with a Gap statistic value higher than $GAP_{\log}(\lambda^*, K^*) - se_{\log \widetilde{Q}^\circ(\lambda^*, K^*)} = 0.0013$. The red Gap statistic value corresponds to λ^* K^*; the orange Gap statistic value corresponds to the chosen simplest model that is within the range of one standard error.

K	λ	$GAP_{\log}(\lambda, K)$	$\log \widetilde{Q}(\lambda, K)$	$E_b \log \widetilde{Q}_b^\circ(\lambda, K)$	$\text{se}_{\log \widetilde{Q}^\circ(\lambda, K)}$
12	0.050	-0.0000	3.8201	3.8201	0.0020
48	**0.050**	**0.0015**	4.1505	4.1520	0.0031
6	0.010	0.0018	2.4412	2.4430	0.0083
8	0.010	0.0031	2.5232	2.5263	0.0133
12	0.025	0.0033	3.3586	3.3619	0.0072
48	0.010	0.0043	3.3843	3.3886	0.0124
12	0.010	0.0068	2.6530	2.6598	0.0155
18	**0.025**	**0.0078**	3.4526	3.4603	0.0065

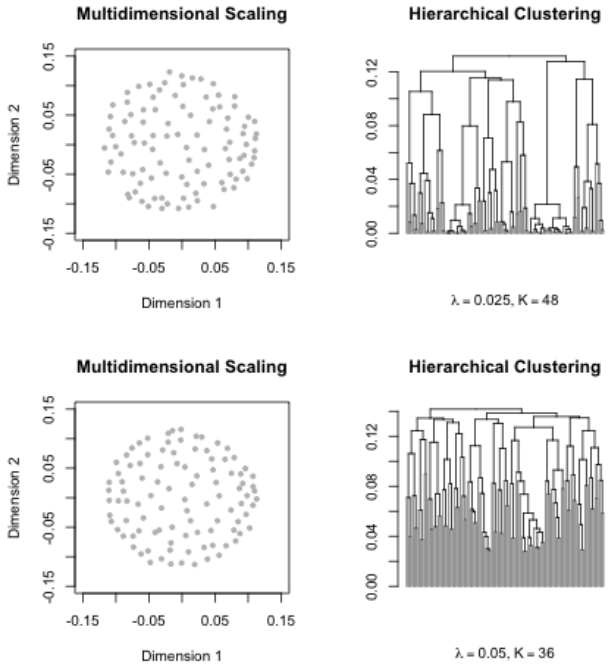

Figure 3.28: The multidimensional scaling and hierarchical clustering visualizations of the COSA-KNN results for 'noise only' data based on $Gap_{\log}$ for K^* and λ^* (in the top row), and the values for λ and K of the simplest model that is within the range of one standard error of the value for K^* and λ^* (bottom row).

Chapter 4

COSA Nearest Neighbors: COSA-KNN and COSA-λNN

In this chapter we will propose further improvements for COSA-KNN. With a change in notation, we can generalize the clustering problem from Chapter 2, into settings where the assumption of mutually exclusive clusters does not need to hold. Furthermore, with this changed notation we will introduce a COSA distance that is better in separating pairs of objects from different clusters. This new non-strict COSA-distance will render COSA-KNN more powerful. Lastly, we propose a λ-tuned strategy in which attribute weights are allowed to become equal to zero, and where the non-zero attribute weights are the result of a *Kullback-Leibler divergence* regularization.

Despite these improvements, a disadvantage remains: in COSA-KNN each object is assigned an equal number of K nearest neighbors. Of the K nearest neighbors belonging to an object in a small cluster, we may have some neighbors that are 'living far apart'. Similarly, for objects belonging to the large clusters, we may have more objects 'living close together' than the K positions available. For these situations, we may expect better results if COSA would allow for a varying number of nearest neighbors per object. This is accomplished in COSA-λNN. Instead of a fixed K, COSA-λNN sets a restriction on the space of the neighborhood around each object. The space of the neighborhood is bounded by a pre-set λ. Only those objects that live within the space of the neighborhood are entitled to be called a nearest neighbor.

To summarize, the chapter starts with the introduction of the improvements for COSA-KNN. When all these improvements are implemented in COSA-KNN, we will refer to its algorithm as COSA-$K\text{NN}_1$. What follows is a comparison between COSA-$K\text{NN}_1$ and the robust version of COSA-KNN from Chapter 3, referred to as COSA-$K\text{NN}_0$. The comparison consists of a simulation study and an application to a real data set. Then, in Section 4.3 we will introduce COSA-λNN, which will be compared with both COSA-$K\text{NN}_1$ and COSA-$K\text{NN}_0$.

4.1 Improvements of COSA-KNN

4.1.1 A Notation for Overlapping Clusters

It would be a misunderstanding to think that COSA-KNN only copes with data where we assume that the underlying clustering structure consists of mutually exclusive clusters. The distances of COSA-KNN can also very well represent a distance of a structure of overlapping clusters. However, the notation with which COSA-KNN was described thus far, does not allow for such a clustering structure.

From rule (2.5) of Chapter 2 we obtain that the attribute weights for each object i, denoted by $\mathbf{w}_i$, represents the attribute weight vector of the cluster that object i belongs to. However, when we wish to allow for overlapping clusters, such that object i could belong to multiple clusters, then the interpretation of $\mathbf{w}_i$ becomes more complicated. The object's attribute weights vector becomes a mixture of multiple weight vectors of multiple clusters.

To allow for a clustering structure where each object can belong to multiple clusters, we simplify the notation by associating the attribute weights vectors solely with a group of objects C_l, and not with an object on its own. Instead of M mutually exclusive true clusters in $\mathcal{G}$, we now define a grouping structure $\mathcal{C}$ that consists of L groups that may overlap, denoted by $\{C_l\}_{l=1}^{L}$. Each group C_l has its own set of attribute weights. While in Chapter 2 the attribute weights vector for a (mutual-exclusive) true cluster G_l was denoted by $\mathbf{w}_l^*$, from here on we will denote the attribute vector by $\mathbf{w}_l$ for a group of objects that can represent a true cluster, and have an overlap with other clusters. In the columns of the P times L matrix $\mathbf{W}$ we collect each attribute weights vector $\mathbf{w}_l$. An element of $\mathbf{W}$ is denoted by w_{kl}, and represents the weight for attribute k $(1 \leq k \leq P)$ for group C_l $(1 \leq l \leq L)$. A similar change is proposed for the attribute dispersions. We will denote by S_{kl} the average distance on attribute k within the l^{th} cluster of objects. The reader who is familiar with algorithm 1 for COSA in Friedman and Meulman (2004) will see that we are using the notation of that algorithm. Note, however, that the notation here allows for more flexibility, e.g., overlapping clusters.

In previous chapters, we used the $N \times N$ indicator matrix $\mathbf{C}$. For each object i (in the rows) it was indicated which objects j (in the columns) belonged to the group of the K nearest neighbors of object i. From now on, we will change the definition of this indicator matrix as follows: the indicator matrix $\mathbf{C}$ is of size $N \times L$ with elements $\{c_{il}\}$. Here, $c_{il} = 1$ indicates that object i belongs to the l^{th} cluster of objects, wge holds for $c_{il} = 0$.

4.1.2 COSA-KNN Reparametrized

In COSA-KNN, the clustering structure is approximated by $\mathcal{C}$, consisting of $L = N$ groups of objects, i.e. the number of groups is equal to the number of objects in the data. Particularly for COSA-KNN, each group of objects, $C_l \in \mathcal{C}$, corresponds to a neighborhood of K objects and these neighborhoods are indexed by $l\,(1 \leq l \leq L = N)$. For each unique object $j\,(1 \leq j \leq N)$ there is a unique neighborhood C_l for which

the distance to all other objects $i \in C_l$ is smaller than, or equal to, any other object $i \notin C_l$. Thus, C_l can represent a true cluster, but it can also be a group of noise objects.

With the reparametrized $N \times L$ matrix for $\mathbf{C}$, the non-robust version of the COSA-KNN criterion is

$$Q(\mathbf{W}, \mathbf{C}) = \sum_{l=1}^{L} \left\{ \frac{1}{K} \sum_{i=1}^{N} c_{il} D_{ij_l}[\mathbf{w}_l] + \lambda \left(\sum_{k=1}^{P} w_{kl} \log(w_{kl}) + \log(P) \right) \right\}, \tag{4.1}$$

where the within-cluster distance between object i and j in group C_l is

$$D_{ij}[\mathbf{w}_l] = \sum_{k=1}^{P} w_{kl} d_{ijk}, \tag{4.2}$$

and j_l is defined as the object j for which the average distance to all other objects $i \in \mathcal{C}_l$ is smaller than the average distance of object j to the objects in any other group l', $i \in \mathcal{C}_{l'}$, i.e.,

$$\left\{ K^{-1} \sum_{i=1}^{N} c_{il} D^{(\eta)}_{ij_l}[\mathbf{W}] \right\} < \left\{ K^{-1} \sum_{i=1}^{N} c_{il'} D^{(\eta)}_{ij_l}[\mathbf{W}] \right\}, \tag{4.3}$$

given $l \neq l'$, where

$$D^{(\eta)}_{ij}[\mathbf{W}] = -\eta v_{ij\bullet} \log \left\{ v^{-1}_{ij\bullet} \sum_{k=1}^{P} v_{ijk} \exp\left(-\frac{d_{ijk}}{\eta} \right) \right\}. \tag{4.4}$$

This specific non-strict distance for COSA in equation (4.4) is based on the inverse exponential COSA distance, defined in (2.44) of Chapter 2. However, in this reparametrization, v_{ijk} is defined as

$$v_{ijk} := \max\left\{ w_{kl_i}, w_{kl_j} \right\}, \tag{4.5}$$

the maximum taken over the weight of attribute k in the neighborhood l_i for object i, and the weight for attribute k in the neighborhood l_j of object j. The neighborhood l_j is defined as

$$l_j = \underset{l}{\operatorname{argmin}}\, K^{-1} \left\{ \sum_{i=1}^{N} c_{il} D^{(\eta)}_{ij}[\mathbf{W}] \right\}, \tag{4.6}$$

corresponding to the unique neighborhood C_{l_j} for which the average distance of object j to all other objects $i \in C_{l_j}$ is smaller than the average distance of object j to the objects of any other group with $l \neq l'$.

In Chapter 2 we have described the algorithm that is used as an heuristic to find the estimates for the optimal $\mathbf{C}$ and $\mathbf{W}$, by minimizing the COSA-KNN criterion in (4.1). Given an estimate for $\mathbf{W}$, we find the minimum for $\mathbf{C}$ by applying a K nearest

neighbors method on the inverse exponential distance. Then, given $\mathbf{C}$, the optimal solution for each w_{kl} is achieved when $\mathbf{W}$ is the minimum of $Q(\mathbf{W}, \mathbf{C})$, this is when

$$\widehat{w}_{kl} = \exp\left(-\frac{S_{kl}}{\lambda}\right) \Bigg/ \sum_{k'=1}^{P} \exp\left(-\frac{S_{k'l}}{\lambda}\right), \tag{4.7}$$

where

$$S_{kl} = \frac{\sum_{i=1}^{N} c_{il} d_{ij_l k}}{K}, \tag{4.8}$$

the mean-based attribute dispersion for attribute k in the neighborhood C_l. In the previous chapter we have seen that the COSA-KNN is improved when the median-based attribute dispersion is used. In this chapter we will propose two robust versions of the COSA-KNN algorithm, where the definition of the attribute dispersion is based on the median. Thus we re-define the attribute dispersion as

$$S_{kl} = \text{median}\left(\{d_{ij_l k}\}_{i \in C_l}\right), \tag{4.9}$$

for the COSA-KNN algorithms.

Setting $L = N$, the algorithm for COSA-KNN remains the same at this point, and is called COSA-$K\text{NN}_0$, the benchmark COSA algorithm. The result is

COSA-KNN$_0$
0 : Set: λ; K; $\alpha \lesssim 0.1$; $L = N$
1 : Initialize: $\eta = \lambda; \mathbf{W} = \{1/P\} \in \mathbb{R}^{P \times L}$;
2 : loop {
3 : Compute distances $D_{ij}^{(\eta)}[\mathbf{W}]$ as in (4.4)
4 : $\widehat{\mathbf{C}} \leftarrow$ K nearest neighbors on $D_{ij}^{(\eta)}[\mathbf{W}]$
5 : $\widehat{\mathbf{W}} \leftarrow$ Compute attribute weights $\mathbf{W}$ as in (4.7), using (4.9)
6 : Increase $\eta : \eta + \alpha * \lambda$
7 : } until $\mathbf{W}$ stabilizes
8 : Output: $D_{ij}[\mathbf{W}]$.

4.1.3 Improved Attribute Weights

The main motivation of COSA-KNN was to consider the analysis of high-dimensional data, as obtained in genomics, proteomics, and metabolomics. As compared to 2004, the typical kind of data sets that come from those areas of research have nowadays many more attributes, i.e. the P / N ratio is much larger. In particular, there are more attributes in the data set that do not play any role in any of the groups, rendering the signal-to-noise ratio to become smaller. Here, the behavior of COSA-KNN may be suboptimal. So far, COSA-KNN drives the attribute weights to have a small negative entropy, i.e., staying close to equal-valued attribute weights. Moreover, up till now, COSA-KNN does not allow for zero-valued attribute weights. We propose two

improvements for the attribute weights. First, we generalize the criterion of COSA-KNN. Instead of penalizing the divergence from equal attribute weights, we can also penalize the divergence from another type of pre-specified attribute weights. The second improvement is a modification of this generalized criterion, such that COSA-KNN assures that a pre-specified proportion of the attributes within each group, receives a zero-value weight.

Regularization of the Kullback-Leibler divergence

We can rewrite the criterion in equation (4.1) as

$$Q(\mathbf{W},\mathbf{C}) = \sum_{l=1}^{L}\left\{\frac{1}{K}\sum_{i=1}^{N} c_{il} D_{ij_l}[\mathbf{w}_l] + \lambda \sum_{k=1}^{P} w_{kl}\log\left(\frac{w_{kl}}{1/P}\right)\right\}, \tag{4.10}$$

from which we can obtain that the criterion of COSA-KNN consists of a regularized *Kullback-Leibler divergence* term (Kullback & Leibler, 1951) for each neighborhood, the divergence between $\mathbf{w}_l$ and a set of P uniform pre-specified attribute weights with the value $1/P$.

We can generalize the criterion by replacing each $1\,/\,P$ with a pre-specified attribute weight u_{kl}, for which it holds that for each group C_l the attribute weights $\{u_{kl}\}_{k=1}^{P}$ belong to the probability simplex, i.e. $\sum_{k=1}^{P} u_{kl} = 1$. This gives

$$Q(\mathbf{W},\mathbf{C}) = \sum_{l=1}^{L}\left\{\frac{1}{K}\sum_{i=1}^{N} c_{il} D_{ij_l}[\mathbf{w}_l] + \lambda D_{KL}(\mathbf{w}_l|\mathbf{u}_l)\right\}, \tag{4.11}$$

where the Kullback-Leibler divergence is

$$D_{KL}(\mathbf{w}_l \,|\, \mathbf{u}_l) = \sum_{k=1}^{P} w_{kl}\log\left(\frac{w_{kl}}{u_{kl}}\right). \tag{4.12}$$

All pre-specifications for u_{kl} that are different from $u_{kl} = 1/P$ for all l and k, will result in more selective attribute weights for $\mathbf{w}_l$. Given $\mathbf{C}$, the optimal solution for the attribute weight w_{kl} becomes

$$\widehat{w}_{kl} = \frac{u_{kl}\exp\left(-S_{kl}/\lambda\right)}{\sum_{k'=1}^{P} u_{k'l}\exp\left(-S_{k'l}/\lambda\right)}. \tag{4.13}$$

By introducing u_{kl}, we can add prior knowledge about the attribute weights in the criterion (4.11). For example, when certain attributes for a certain group of objects are expected to be important for the clustering, then we can give these attributes higher pre-specified weight values, compared to the other attributes.

Prior information can also be incorporated based on information in the data. When the attributes are normalized to represent the same scale, our empirical evidence so far suggests to use

$$u_{kl} = \frac{\exp(-s_k/\lambda)}{\sum_{k'}^{P}\exp(-s_{k'}/\lambda)}, \tag{4.14}$$

where the lowercase s_k is the interquartile range of attribute k on all observations divided by 1.35. In this particular case, the pre-specified attribute weights are the same for each group C_l. Based on the concept that the interquartile range on a normalized attribute is smaller when clustering is present, given that clustering occurs within the range, lower values of the pre-specified attribute weights are to be expected for the high interquartile attribute ranges, and vice versa.

Zero-Valued Attribute Weights

To allow for zero-valued attribute weights we modify the pre-specified attribute weights into

$$u_{kl} = \frac{a_{kl} \exp(-s_k/\lambda)}{\sum_{k'}^{P} a_{k'l} \exp(-s_{k'}/\lambda)}, \tag{4.15}$$

where a_{kl} is the result of an indicator function (attribute activation function), which indicates whether attribute k obtains a non-zero weight in group C_l, or not:

$$a_{kl} = \begin{cases} 1 & \text{if } S_{kl} \leq \widetilde{S}_l(P_\lambda); \\ 0 & \text{if } S_{kl} > \widetilde{S}_l(P_\lambda). \end{cases} \tag{4.16}$$

Here, $\widetilde{S}_l(P_\lambda)$ is the value of the P_λ^{th} smallest attribute dispersion in group C_l, where

$$P_\lambda = \text{ceiling}\left(P \times \min(1, \lambda)\right). \tag{4.17}$$

Here, P_λ is a λ related number of attributes. Knowing P_λ, we obtain $\widehat{S}_l(P_\lambda)$ as the statistic of order P_λ in the set $\{S_{kl}\}_k^P$, where the elements are sorted by descending value. Thus, we assign a zero-value attribute weight, $w_{kl} = 0$, when the average distance on that attribute within group C_l belongs to the higher averages, i.e. $S_{kl} > \widehat{S}_l(P_\lambda)$.

4.1.4 Stronger Between-Cluster COSA Distances

To find the grouping structure $\mathbf{C}$, we use the K nearest neighbors method on COSA distances. So far, this was the inverse exponential distance $D_{ij}^{(\eta)}[\mathbf{W}]$ in (4.4) (and from (2.56) in Chapter 2). Was in Chapter 2 that when $\eta \to \infty$, this particular distance evolves into the 'majorizing' COSA distance $D_{ij}[\mathbf{W}]$ in equation (2.11), i.e. a weighted sum of the attribute distances:

$$D_{ij}[\mathbf{W}] = \sum_{k=1}^{P} v_{ijk} d_{ijk}, \tag{4.18}$$

where now, the definition of v_{ijk} is the maximum of w_{kl_i} and w_{kl_j}, as provided in equation (4.5). Although the maximum assures that between-cluster COSA distances will be larger than within-cluster COSA distances, it does not strongly take into account the differences in the attribute weights of the two clusters. Whether the

attribute weights are $\{w_{kl_i} = 0.1, w_{kl_j} = 0.9\}$ or $\{w_{kl_i} = 0.8, w_{kl_j} = 0.9\}$, the influence of attribute k in the separation of objects i and j remains the same.

We can modify the definition of v_{ijk}, as long as the between-cluster COSA distance obeys the following necessary condition: when $w_{kl_i} = w_{kl_j}$ for each attribute k, we have

$$D_{ij}[\mathbf{W}] = v_{ijk}d_{ijk} = \sum_{k=1}^{P} w_{kl_i}d_{ijk} = \sum_{k=1}^{P} w_{kl_j}d_{ijk}. \tag{4.19}$$

We propose the following modification of the definition for v_{ijk}. Let $|w_{kl_i} - w_{kl_j}|$ be the absolute difference of the weights in groups C_{l_i} and C_{l_j} for attribute k, and let $1 - |w_{kl_i} - w_{kl_j}|)$ be the overlap. Then, when we add the λ^{-1}-scaled *odds* of the difference versus the overlap of the two attribute weights to the prior definition of v_{ij} in equation (4.5), we obtain

$$v_{ijk} := \max(w_{kl_i}, w_{kl_j}) + \lambda^{-1}\frac{|w_{kl_i} - w_{kl_j}|}{(1 - |w_{kl_i} - w_{kl_j}|)}. \tag{4.20}$$

Given that the different clusters each have a different subset of attributes, the λ-scaled odds term in equation (4.20), assures a stronger separation between the objects from different clusters in the COSA distances. The more different the subsets of attributes, or the smaller λ, the larger the between-cluster distance.

4.1.5 The Algorithm of COSA-KNN$_1$

At this point we have all the ingredients to write down the improved COSA-KNN algorithm, called COSA-KNN$_1$. Similar to the COSA algorithms from the previous chapters, all these improvements have a corresponding approximate criterion that needs to be minimized over $\mathbf{C}$ for a fixed $\mathbf{W}$ (see Chapter 2). The approximate criterion is defined as

$$\tilde{Q}^{(\eta)}(\mathbf{C}, \mathbf{W}) = \sum_{l=1}^{L}\left\{\frac{1}{K}\sum_{i=1}^{N} c_{il}D_{ij_l}^{(\eta)}[\mathbf{W}]\right\}. \tag{4.21}$$

We sketch the algorithm as follows:

COSA-KNN$_1$

0 : Set: λ; K; $L = N$; $\{u_{kl}\}$; $P_\lambda = P \times \min(1, \lambda)$
1 : Initialize: $\eta = \lambda$; $w_{kl} = u_{kl}$
2 : loop {
3 : Compute distances $D_{ij}^{(\eta)}[\mathbf{W}]$ as in (4.4), using (4.20)
4 : $\widehat{\mathbf{C}} \leftarrow$ K nearest neighbors on $D_{ij}^{(\eta)}[\mathbf{W}]$
5 : $\widehat{\mathbf{W}} \leftarrow$ Compute attribute weights for $\mathbf{W}$ as in (4.13), using (4.15)
6 : Increase $\eta : \eta + \alpha * \lambda$
7 : } until $\mathbf{W}$ stabilizes
8 : Output: $D_{ij}[\mathbf{W}]$.

Thus, COSA-KNN$_1$ incorporates the sparser attribute weights and the new COSA distances based on the pairwise weights in equation (4.20). Note that the homotopy path over η consists of two steps. We have a start value of $\eta = \lambda$, and have set $\alpha = \infty$ in this chapter, instead of $\alpha = 0.1$ (the value used in the previous chapters). Using $\alpha = 0.1$ or $\alpha = \infty$ would not have changed any of the conclusions in this chapter. Thus, we use the inverse exponential distance in (4.4) as a 'start' distance and use the distance in equation (4.19) for the rest of the iterations in the algorithm. Last, we say that the solution for W stabilizes when the average change in the attribute weights is smaller than 10^{-5}.

4.2 Comparing COSA-KNN$_1$ to COSA-KNN$_0$

While using the median-based (robust) attribute disperions (see Chapter 3), we compare COSA-KNN$_1$ with COSA-KNN$_0$ on two simulated data sets and a real data set. The first simulated data set is generated from the COSA prototype model, whose structure is summarized in Chapter 1 (Figure 1.2), and Chapter 3 (Figure 3.1). The other simulated data set and the real data also have been used in Witten and Tibshirani (2010) to show that COSA-KNN$_0$ was not able to cope with such data. Therefore, the second simulated data set is generated from a model that we call a 'weak spot model', since it shows the weak spot of the original COSA. The weak spot model was already displayed in Chapter 1 (Figure 1.1.), but with smaller variances, and it was shown for a different purpose. In this Chapter we describe the weak spot model in Section 4.2.2. The real data set contains gene expression in breast cancer tumors cells, obtained from a study by Perou et. al (2001). Except for the analysis of the real data set, the tuning parameters λ and K were set equal to 0.2 and $\sqrt{N}$ for the old and improved COSA-KNN. For the real data set the only difference was that λ was set equal to the value that made sure 496 attributes were selected out of 1753 for each neighborhood (*cf.* Witten and Tibshirani, 2010). In COSA-KNN$_1$ we used the pre-specified attribute weights in (4.15).

4.2.1 COSA's Prototype Data

In Figure 4.1 we display the COSA distances for both COSA-KNN$_0$ and COSA-KNN$_1$ in average linkage dendrograms. It is clear that the clustering is sharper for the COSA-KNN$_1$; the distances between the objects from different clusters are large, and the distances within the clusters are small. The distances between the noise objects have a higher variability, but there is no spurious clustering visible.

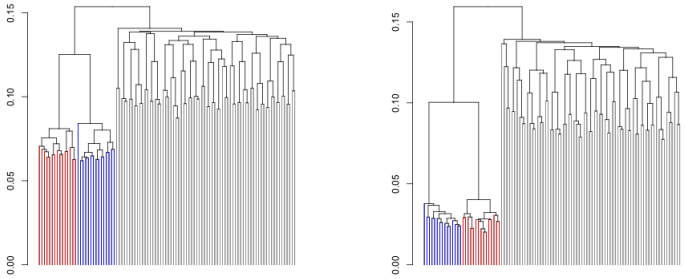

Figure 4.1: Average linkage dendrograms for the distances of COSA-KNN_0 (left panel) and COSA-KNN_1 (right panel).

The two-dimensional multidimensional scaling (MDS) solution of both sets of COSA distances is depicted in Figure 4.2. The two clusters (the red cluster and the blue cluster) are more homogeneous for COSA-KNN$_1$, while no clear differences are visible between COSA-KNN$_0$ and COSA-KNN$_1$ for the noise objects (in grey).

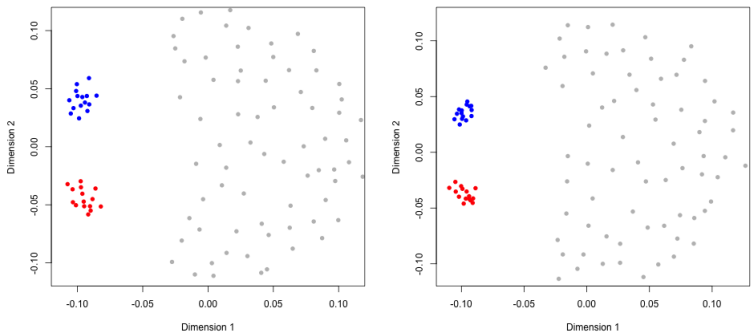

Figure 4.2: Two-dimensional MDS solution for two sets of COSA distances. COSA-KNN_0 on the left, and COSA-KNN_1 on the right.

Of the three improvements that define COSA-KNN$_1$, compared to COSA-KNN$_0$, it is the improved between-cluster COSA distance, based on the pairwise attribute weights in equation (4.20), that mainly explains the sharper clustering results of COSA-KNN$_1$. The zero-value attribute weights play a less prominent, but a facilitating role. With the zero-value attribute weights, it is easier for the attribute weights of two clusters to become disjoint, rendering a larger between-cluster distance due to the 'odds' term in equation (4.20). For this specific simulation example no differences were found between the use of the pre-specified attribute weights defined in equation

(4.15), or for $u_{kl} = 1/P$.

4.2.2 COSA's Weak Spot

To obtain a good understanding of COSA, it useful to know the type of data that COSA was *not* developed for. As was already found by Peter Hoff in the discussion of the paper by Friedman and Meulman (2004), COSA mainly detects differences in the variances on attributes among groups, but not in the means. The clustering found in COSA is predominantly due to the objects having measurements at attributes that are tightly concentrated around common values, and less due to the differences with these common values from objects of another group.

In Figure 4.3 we depict a model for a data set on which COSA-KNN$_0$ would not be able find the clustering. We will call this model the 'weak spot' model. Based on this model we generate data where the attributes only show differences in the means for each group, but not a difference in the variances. The data set has dimensionality $N = 60$ by $P = 1000$, and consists of three clusters of each 20 objects, clustering on the first 50 attributes coming from an i.i.d. normal distribution with standard deviation 1 and the means set at $\mu_{C_1} = -1.5$, $\mu_{C_2} = 0$, $\mu_{C_3} = +1.5$. All remaining data values are i.i.d. realizations from a standard normal distribution.

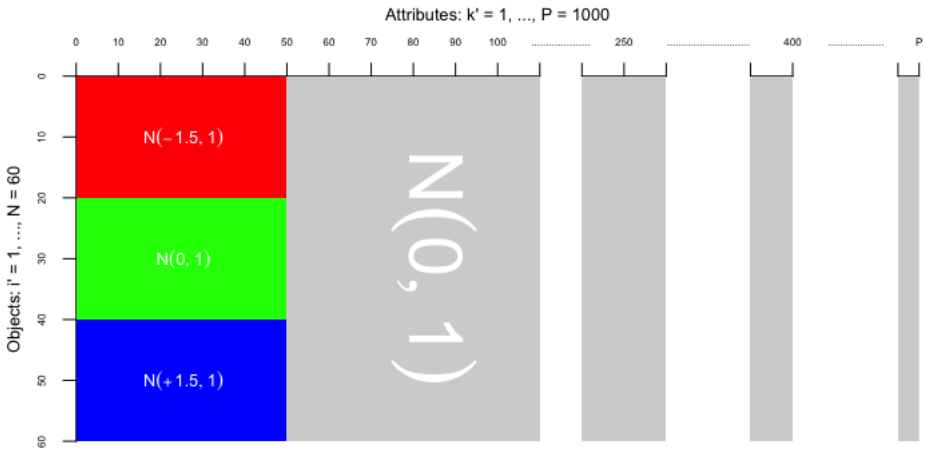

Figure 4.3: Data model with which data can be generated that emphasize COSA's weak spot. After generating a data set from this model, each attribute is standardized to have zero mean and unit variance.

The two version of COSA have been applied with $\lambda = 0.2$ and $K = \text{ceiling}(\sqrt{N})$. The resulting complete linkage dendograms and MDS representations can be found in

Figures 4.4 and 4.5, respectively. The dendrograms show that the clustering structure is not captured in the first set of COSA distances, but is captured with the distances from COSA-KNN$_1$ (right panel).

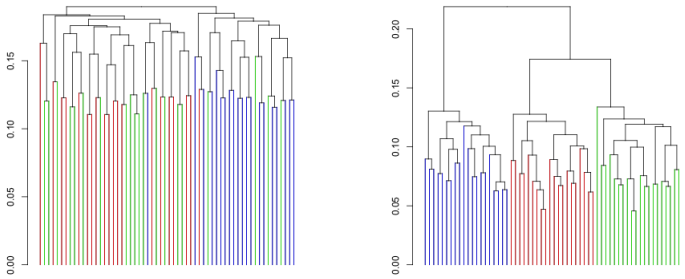

Figure 4.4: Complete linkage dendrograms for two sets of COSA distances. The COSA-KNN_0 distances are on the left, the COSA-KNN_1 distances are on the right.

Comparable results are obtained in Figure 4.5. While the color labels in the Figure hint that the two-dimensional MDS solution for the distances of COSA-KNN$_0$ almost recovers the clustering structure, a good separation of the clusters is obtained with the distances of COSA-KNN$_1$ (right panel).

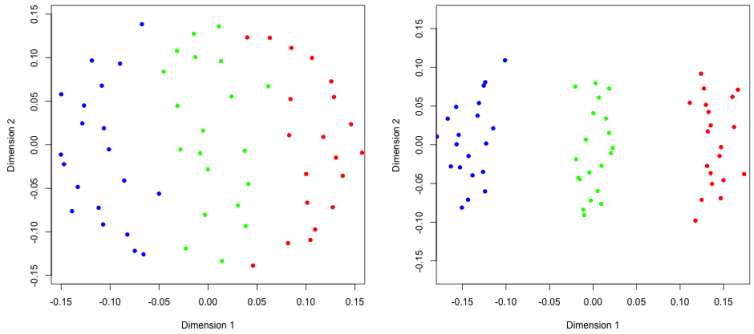

Figure 4.5: Two-dimensional MDS solutions for two sets of COSA distances. The COSA-KNN_0 distances are on the left, the COSA-KNN_1 distances are on the right.

Thus, compared to COSA-KNN$_0$, COSA-KNN$_1$ is not only able to result in sharper clustering, it is also better in detecting mean differences on attributes between groups. This improvement is ascribed to the definition of the pre-specified attribute weights, $\{u_{kl}\}_{k=1}^{P}$ in equation (4.15), for each cluster. Due to the multi-modal distribution of the attributes that are important for the clustering, a lower inter-quartile

range is obtained on these attributes, rendering higher values for the pre-specified attribute weights.

4.2.3 Breast Cancer Data

In Perou et al. (2000), 65 surgical specimens of human breast tumors were profiled on gene expression microarrays. Some of the samples were taken from the same tumor before and after chemotherapy. In the study 62 of the 65 samples were classified into four categories: "basal-like" (eight observations, displayed by red), "Erb-B2" (seven observations, in green), "normal breast-like" (blue, eleven observations), or "ER+" (orange, 46 observations). The three remaining observations did not belong to any of the four classes. These classes were found when 65 samples were hierarchically clustered using 496 out of the available 1753 genes. When all 1753 genes were used, these clusters did not reveal themselves.

With only a few misclassifications, Witten and Tibshirani (2010) did manage to almost perfectly recover the four classes on the 62 samples and 1753 genes with their type of distances: a weighted sum of squared euclidean attribute distances. The attribute weights in this particular sum are restricted by an ℓ_2 and $\ell_1$norm, and are the same for all clusters. Witten and Tibshirani (2010) also showed that COSA was not successful on this specific data set. Figures 4.6 and 4.7 show that, indeed, the dendrograms and the two-dimensional MDS solutions of COSA-KNN$_0$ are not successful. Although suboptimal, the results for COSA-KNN$_1$ are much better.

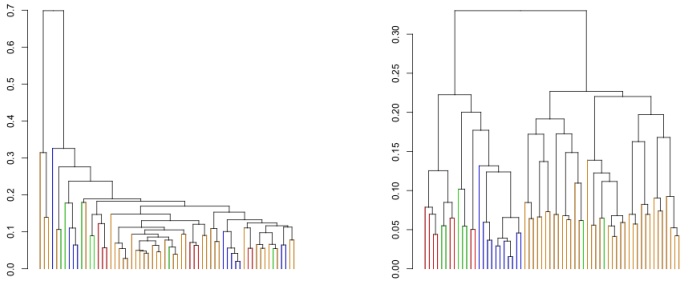

Figure 4.6: Complete linkage dendrograms for two sets of COSA distances on the Breast Cancer data. The COSA-KNN_0 distances are on the left, the COSA-KNN_1 distances are on the right. Here, the color labels have the following representation; the basal-like tumors are displayed in red, the Erb-B2 tumors in green, the normal breast-like tumors in blue, the ER+ tumors are displayed in orange.

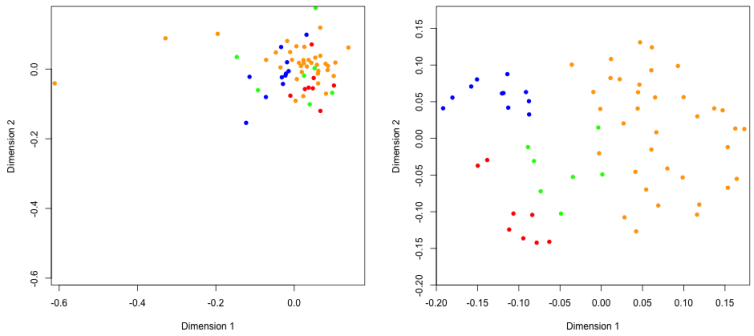

Figure 4.7: Two-dimensional MDS solutions for the two sets of COSA distances on the Breast cancer data: COSA-KNN_0 (left) vs. COSA-KNN_1 (right). The basal-like tumors are displayed in red, the Erb-B2 tumors in green, the normal breast-like tumors in blue, the ER+ tumors are displayed in orange.

4.3 COSA-λNN

In this section, we propose to circumvent choosing the value of the tuning parameter K, by letting λ, instead of K, indirectly control the size of each group of objects (C_l). The algorithm that incorporates this modification is called COSA-λNN. A motivation for this improvement will follow from a description of a general form of the COSA criterion for COSA Nearest Neighbors algorithms, and the (mean-based) attribute dispersions.

4.3.1 Motitavion based on the Criterion

So far, all versions of the COSA-KNN algorithms fit in the format of the COSA criterion defined as

$$Q(\mathbf{W},\mathbf{C}) = \sum_{l=1}^{L}\sum_{k=1}^{P}\begin{cases} w_{kl}\left\{S_{kl} + \lambda \log\left(\frac{w_{kl}}{u_{kl}}\right)\right\}, & \text{if } u_{kl} \neq 0; \\ 0, & \text{if } u_{kl} = 0. \end{cases} \tag{4.22}$$

As was shown in the previous chapter, Section 3.2.1, when we minimize $Q(\mathbf{W},\mathbf{C})$ in equation (4.22) over the attribute weights ($\mathbf{W}$), the criterion has a simpler form, i.e.,

$$Q(\mathbf{C}) = \sum_{l=1}^{L} -\lambda \log\left\{\sum_{k=1}^{P} u_{kl} \exp\left(-\frac{S_{kl}}{\lambda}\right)\right\}. \tag{4.23}$$

When each S_{kl} is defined as in equation (4.9), and each u_{kl} as in equation (4.15), then $Q(\mathbf{W},\mathbf{C})$ and $Q(\mathbf{C})$ represent the criteria for COSA-KNN$_1$. For $Q(\mathbf{W},\mathbf{C})$ and $Q(\mathbf{C})$ to be the criteria of COSA-KNN$_0$ the value for each u_{kl} is set equal to $1/P$. For the non-robust version of the COSA-KNN criterion (see Chapter 3), the definition

of each attribute dispersion S_{kl} should be the average of the K attribute distances within the neighborhood C_l; also, each $u_{kl} = 1/P$.

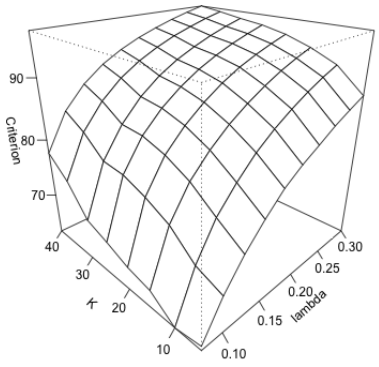

Figure 4.8: Criteron values for λ and K of the non-robust COSA-KNN, when applied to a data set that is generated from the prototype model.

All versions of COSA-KNN have two tuning parameters, λ and K. In general, for a larger K, the COSA-KNN criterion obtains a higher value. For an example see Figure 4.8. In the two previous chapter we also described how this behavior can be modified by λ: it could be that a certain larger value for K assures that for a certain fixed value of λ the attribute weights can be defined more sharply, such that the value of the criterion becomes lower. However, when the value for K becomes too large, the role of λ in the attribute selection becomes smaller, since for $K \to N$ we have $S_{kl} = S_{kl'}$ due to the standardized attribute distances (see Chapter 2, equation (2.1)), and hence, all neighborhoods will have the same attribute weights. Therefore, according to Friedman and Meulman (2004), the value for K should not be larger than the cluster sizes in the data, but should also not be too small, to avoid unstable attribute dispersions. However, for some situations it will be difficult to choose a K when the data consists of clusters varying highly in size.

4.3.2 COSA-λNN: Different Sizes for C_l

In COSA-λNN, the size of each neighborhood may be different, while in COSA-KNN the size for each neighborhood of objects is set equal to K. We will denote the size of each C_l in COSA-λNN by N_l. To obtain the value of each N_l, we exploit the behavior of the attribute dispersions, as was explained for the COSA-KNN criterion in the previous section. We have seen that for $Kl \to N$, all attribute dispersions will become equal, and we fail to detect a clustering structure since we cannot distinguish the signal

attributes from the noise. Relying on the assumption that of the P attributes, there are many that just represent noise values for a cluster, it could be useful to detect a sharp distinction between large values of the attribute dispersions (mostly noise), and small values of the attribute dispersion (mostly signal).

In COSA-λNN, the idea is to set the value of each N_l to create a sharp distinction between noise and signal attribute dispersions. In particular, we base the size of each neighborhood on a type of variance of $\{S_{kl}\}$ in neighborhood C_{l_j} and the Kullback-Leibler divergence ($D_{KL}(\mathbf{w}_l \,|\, \mathbf{u}_l)$ in equation 4.12). Compared to the median-based attribute dispersions, the value of the variance of the mean-based attribute dispersions in C_l is more sensitive to different values of N_l. Similarly, compared to median-based attribute dispersions, the value of $D_{KL}(\mathbf{w}_l \,|\, \mathbf{u}_l)$ is more sensitive to different values of N_l when based on the mean-based attribute dispersions. Therefore, in COSA-λNN, the definition of the attribute dispersion is based on the mean, i.e.

$$S_{kl} = \frac{\sum_{i=1}^{N} c_{il} d_{ij_l k}}{N_l}. \tag{4.24}$$

Here, $c_{il} = 1$ for the objects i that belong to the N_l nearest neighbors of j_l, and $c_{il} = 0$ otherwise. Thus,

$$N_l = \sum_{i=1}^{N} c_{il}. \tag{4.25}$$

Remember that j_l in equation (4.3) denotes the object j for which all objects in the neighborhood C_l are the nearest N_l neighbors. The nearest neighbors are found based on $D_{ij}^{(\eta)}[\mathbf{W}]$ in equation (4.4), in which equation (4.20) is used for the definition of v_{ijk}.

We find N_l for each C_l, by adding the next closest object until we reach a 'certain' (sub)optimal value for a particular ratio between $D_{KL}(\widehat{\mathbf{w}}_l|\mathbf{u}_l)$ and the weighted variance of $\{S_{kl}/\lambda\}$. Here, the variance of the attribute dispersions, $\{S_{kl}\}_{k=1}^{P}$, in neighborhood $C_l$is weighted by $\widehat{\mathbf{w}}_l$, i.e.

$$\mathrm{VAR}\,(S_{kl}) = \sum_{k=1}^{P} \widehat{w}_{kl} S_{kl}^2 - \left(\sum_{k=1}^{P} \widehat{w}_{kl} S_{kl} \right)^2 \tag{4.26}$$

Since both $D_{KL}(\widehat{\mathbf{w}}_l|\mathbf{u}_l)$ and $\mathrm{VAR}\,(S_{kl})$ are functions of λ, the sizes of the neighborhoods will also be a function of λ – hence the name: COSA-λNN.

4.3.3 Selection of N_l based on $D_{KL}(\widehat{\mathbf{w}}_l|\mathbf{u}_l)$ and VAR (S_{kl})

COSA-λNN exploits two relations to obtain the size of each neighborhood. The relation between $D_{KL}(\mathbf{w}_l|\mathbf{u}_l)$ in (4.12) and N_l, as well as the relation between $\mathrm{VAR}\,(S_{kl})$ in (4.26) and N_l. Within a neighborhood of objects that cluster on the same subset of attributes, we expect a relatively small subset of attributes each with a low S_{kl}, and a high attribute weight $\widehat{w}_{kl}$. For the remaining attributes we expect large attribute dispersions, and low attribute weights. In particular, the higher the number of objects

that cluster on the same subset of attributes, the sharper this distinction. Thus, as long as we increase N_l with objects that cluster on the same subset of the attributes, we obtain more or less a higher variance of the weighted attribute dispersions.

At a certain point, this particular variance reaches either an inflection point (from concave to convex) or a local maximum for N_l. When approaching such a point for $\mathrm{VAR}(S_{kl})$, each next nearest object that is being added to the neighborhood does not give a large increase in the variance anymore. For example, it could be that for a neighborhood the subsets of attributes that contribute to the clustering have been discovered. Then, adding an extra object that clusters on about the same subset of attributes does not contribute much to a sharper distinction between the relevant and the non-relevant subset of attributes. By increasing N_l after such an inflection point has been reached, we are probably adding objects that (also) cluster on other non-overlapping subsets of attributes. Therefore, the variance of S_{kl} will decrease, since the attribute dispersions will become more similar on more of the attributes for k. Note that the attribute dispersions that are based on larger N_l, will become more similar to each other since each d_{ijk} is scaled for each attribute (see Chapter 2, equation (2.1)).

As is depicted for an example in Figure 4.9, it may happen that $\mathrm{VAR}(S_{kl})$ approaches an inflection point first, and that the variance starts to increase again for larger N_l. An explanation is that after such an inflection point, we may be adding objects to the neighborhood that do not cluster on the same subset of attributes but on a larger overlapping subset of attributes.

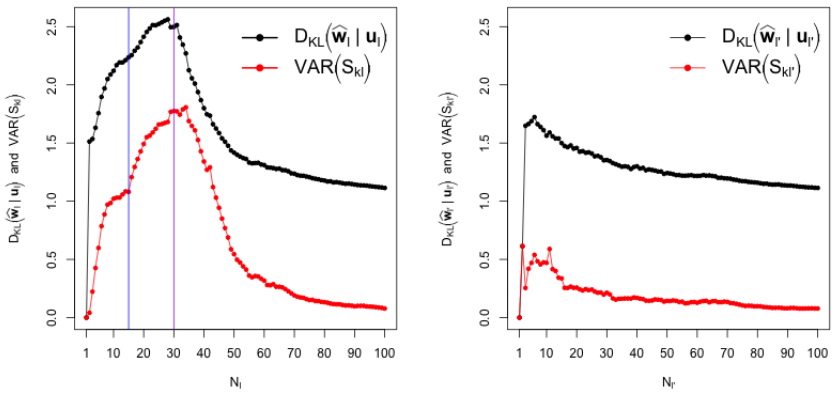

Figure 4.9: $D_{KL}(\widehat{\mathbf{w}}_l|\mathbf{u}_l)$ and $VAR\,(S_{kl})$ plotted against N_l for the neighborhood of an object j_l (left panel) that belongs to the 'blue' cluster of the prototype model. The blue vertical line is at $N_l = 15$ and the purple vertical line is at $N_l = 30$. In the right panel a similar plot is shown for a noise object $j_{l'}$ of neighborhood l' from the same data set of the prototype COSA model. The nearest neighbors were based on the COSA-λNN distances that resulted from COSA-λNN with $\lambda = 0.2$.

Figure 4.9 shows the relation between $\text{VAR}\,(S_{kl})$ and N_l for a neighborhood l around an object i_l that belongs to the blue cluster of the prototype COSA model (left panel), and for a neighborhood l' around a noise object $i_{l'}$ (right panel). These two objects come from the same data set that was generated from prototype model in Section 4.2.1. For an increasing number of nearest neighbors of j_l, we see an increase in $\text{VAR}\,(S_{kl})$ until we reach a point of diminishing increases around $N_l = 15$ (the blue vertical bar). Then, we reach a second point of diminishing increases for $\text{VAR}\,(S_{kl})$ at $N_l = 30$. At this point the neighborhoods consist of the blue and red clusters of the prototype model. After this point, we see that the variance of the attribute dispersions decreases rapidly. Figure 4.9 also shows the relation between $D_{KL}(\widehat{\mathbf{w}}_l|\mathbf{u}_l)$, in (4.12), and N_l. Although smoother, its pattern is quite similar to that of $\text{VAR}\,(S_{kl})$.

In the right panel of Figure 4.9 we show a similar plot, but then for one of the noise objects ('grey'). Note that for each $N_{l'}$ in the neighborhood of this noise object, both the Kullback-Leibler divergence and the variance of the attribute dispersions are smaller. Moreover, the $D_{KL}(\widehat{w}_{l'}|\mathbf{u}_{l'})$ starts decreasing already after a much smaller value of N_l'. The same holds for $\text{VAR}\{S_{kl}\}$, however it shows a less stable pattern. Thus both $D_{KL}(\widehat{\mathbf{w}}_l|\mathbf{u}_l)$ and $\text{VAR}\,(S_{kl})$ give us information about the size of the neighborhood for each object.

The ratio of $\text{VAR}\,(S_{kl})$ to $D_{KL}(\widehat{\mathbf{w}}_l|\mathbf{u}_l)$ is even more powerful in revealing the

inflection point of interest. We denote this ratio by $\lambda^*_{N_l}$, and it is defined as

$$\lambda^*_{N_l} = \sqrt{\frac{\text{VAR}\,(S_{kl})}{D_{KL}(\widehat{\mathbf{w}}_l|\mathbf{u}_l)}} \quad . \tag{4.27}$$

Compared to the left panel of Figure 4.9, in Figure 4.10 the stationarity point for the signal object in the region of $N_l \in \{8, 9, \ldots, 15\}$ is better visible. For the noise object (in grey), the resulting line Figure 4.10 consists of more fluctuations when compared with the results in the right panel of Figure 4.9.

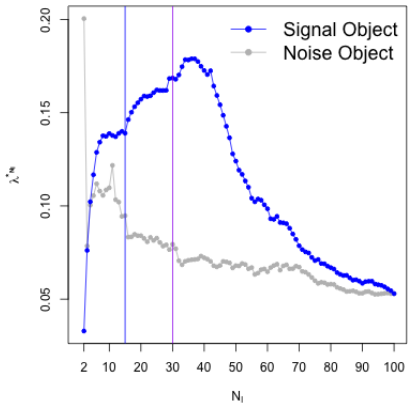

Figure 4.10: $\lambda^*_{N_l}$ plotted against N_l for the neighborhood of an object belonging to the cluster and for $N_{l'}$ of the neighborhood of a noise object.

Based on the value of $\lambda^*_{N_l}$, COSA-λNN finds each N_l. The value for N_l is defined as the first neighborhood size for which

$$\lambda^*_{N_l} \geq \lambda^*_{N_l+1}. \tag{4.28}$$

For each neighborhood l, COSA-λNN starts with $N_l = 2$ and computes $\lambda^*_{N_l}$. Then, COSA-λNN increases N_l until it observes (4.28). Thus,

$$\mathcal{C}_l = i \in \lambda NN(j) = \{i \,|\, \lambda^*_{N_l} \geq \lambda^*_{N_l+1}\}. \tag{4.29}$$

With Figures 4.9 and 4.10 we have illustrated the conceptual idea how COSA-λNN finds N_l using $\lambda^*_{N_l}$. Note that the definition for $\lambda^*_{N_l}$ is not just arbitrary. This definition in equation (4.28) can also be seen as a solution for the minimization problem of a particular constrained version of the COSA criterion (Appendix Section 4.6.1), i.e., it is a solution that satisfies the Complementary Slackness condition of the Karush-Kuhn-Tucker (KKT) conditions (see, e.g., Boyd & Vandenberghe 2004).

4.3.4 The COSA-λNN Algorithm

We sketch the algorithm COSA-λNN as follows:

COSA-λNN

0 : Set: λ; $L = N$; $\{u_{kl}\}$; $P_\lambda = P \times \min(1, \lambda)$
1 : Initialize: $\eta = \lambda$; $w_{kl} = u_{kl}$
2 : loop {
3 : Compute distances $D_{ij}^{(\eta)}[\mathbf{W}]$ as in (4.4), based on (4.20)
4 : For each l_j, set $N_{l_j} = 1$
5 : For each l_j, loop {
6 : $N_{l_j} = N_{l_j} + 1$
7 : $\{\widehat{c}_{il_j}\}_{i=1}^{N} \leftarrow$ N_l nearest neighbors on $D_{ij}^{(\eta)}[\mathbf{W}]$
8 : $\widehat{\mathbf{w}}_{l_j} \leftarrow$ Compute attribute weights for $\mathbf{w}_{l_j}$ as in (4.13)
9 : } until each N_l satisfies condition (4.28)
10 : Increase $\eta : \eta + \alpha * \lambda$
11 : } until $\mathbf{W}$ stabilizes
12 : Output: $D_{ij}[\mathbf{W}]$.

4.4 Comparing COSA-λNN with COSA-KNN

In this section we will add the results of the COSA-λNN to the results of COSA-KNN$_0$ and COSA-KNN$_1$ obtained in Section 4.2. In particular, the same data set has been analyzed with COSA-λNN: the data sets generated from the prototype model, the data set generated from the weak spot model (Figure 4.3), and the breast cancer data (Perou et. al., 2001).

4.4.1 Simulated Data

For the data set that was generated from the COSA prototype model, it is difficult to say whether the results obtained with COSA-λNN show an improvement over COSA-KNN$_1$. Figure 4.11 shows that the distances between the clusters and the group of noise objects remain the same. There is a noteworthy difference between the dendrogram of the COSA-λNN (right panel) and that of COSA-KNN$_1$ (middle panel). For the results of COSA-λNN the distances between the noise object pairs are more variable. These varying distances may have the disadvantage that some noise object pairs, trio's, or even some quartets may appear closer to each other.

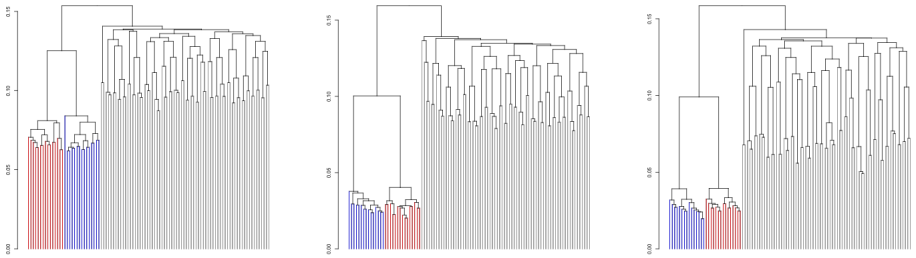

Figure 4.11: The resulting dendrograms of the distances from the Old COSA-KNN_0 (left), COSA-KNN_1 (middle), COSA-λNN (right) obtained from a data set of the prototype model.

Figure 4.12 shows the Multidimensional Scaling (MDS) results for COSA-λNN in the right panel. Here, the MDS solution of the COSA-λNN results in a sharp and similar clustering structure as that of the COSA-$K\text{NN}_1$ results. Note that the distances between the noise objects do not appear to be smaller, as was the case in the dendrograms. Consistent with the dendrogram for COSA-λNN, though less clearly visible, there are fewer noise objects that are closer-by to the clustered objects (red and green) as compared to the MDS solution for COSA-KNN.

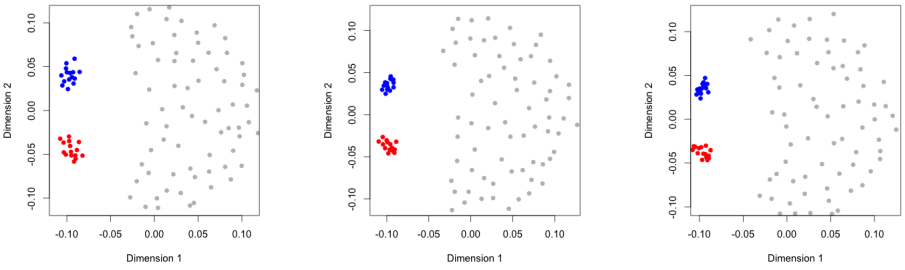

Figure 4.12: The resulting MDS solutions of the distances from the Old COSA-KNN (left), COSA-KNN_1 (middle), COSA-λNN (right) obtained from a data set of the prototype model.

Very similar dendrograms and MDS solutions are obtained from the results of COSA-λNN and the results of COSA-$K\text{NN}_1$ on the data that was generated from the COSA weak spot model, see Figure 4.13. It may seem that the dendrograms indicate that the cophenetic distance between the blue cluster and the green cluster is larger for the COSA-KNN distances, when compared to those obtained by COSA-λNN distances. However, these results do not show in the MDS solutions.

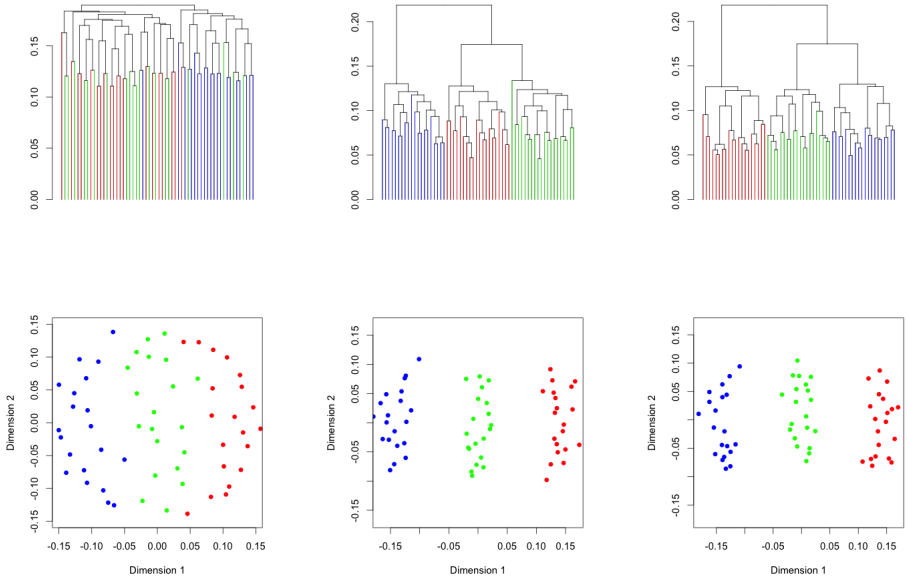

Figure 4.13: The average linkage dendrograms and MDS solutions of the distances from the results of COSA-KNN_0 (left), COSA-KNN_1 (middle), COSA-λNN (right) that were obtained from a data set generated from the weak spot model.

4.4.2 Breast Cancer Data

Although COSA does not 'perfectly' replicate the clustering structure in the breast cancer data set, the visualizations of the COSA-λNN results capture the structure better than the COSA-KNN results, see Figure 4.14. The dendrogram of the COSA-λNN distances seem to show the fewest missclassifications.

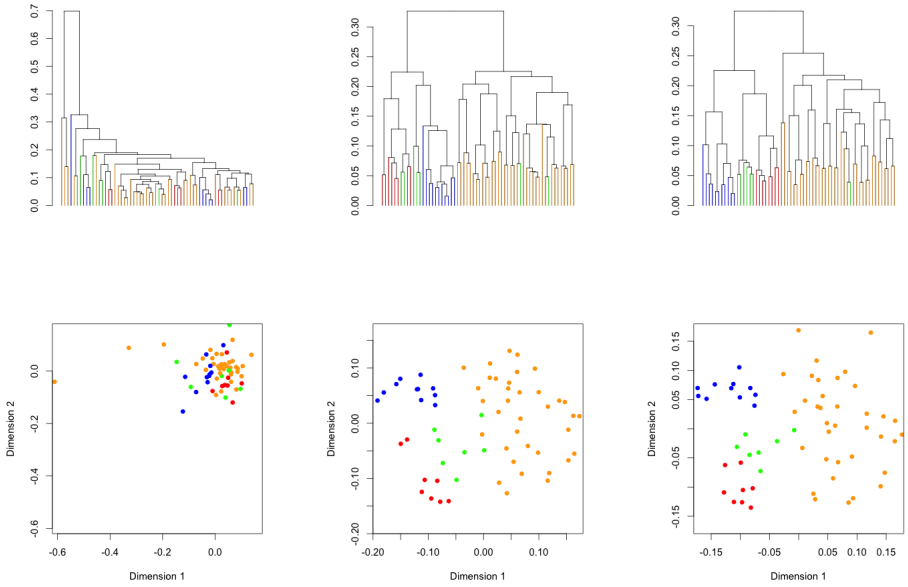

Figure 4.14: The complete linkage dendrograms and MDS solutions of the distances from the results of COSA-KNN_0 (left), COSA-KNN_1 (middle), and COSA-λNN (right) that were obtained from the Breast cancer data set.

4.5 Discussion

In this chapter we introduced a new notation and proposed to implement prior attribute weights, sparser attribute weights and a new COSA distance. We have seen that these implementations render COSA-KNN to reveal a sharper clustering both for simulated data examples and a real data set. Without these implementations COSA-KNN would not be able to separate clusters that mainly differ in their mean on attributes where the within-cluster variances are equal to those of the noise objects.

Furthermore, we have also shown that we can do without the tuning parameter K. Instead of fixing the size of each neighborhood to K, we proposed COSA-λNN to restrict the neighborhood space by $\lambda^*_{N_l}$, whose value is found based on λ. We have shown that COSA-λNN does equally well as the improved version of COSA-KNN, if not better. Instead of setting the size of each neighborhood to a fixed number K, COSA-λNN restricts the neighborhood space by $\lambda^*_{N_l}$, which allows for neighborhoods of different sizes. This is something that is shown to be useful for the Breast cancer data, where the ground truth clusters where of unequal sizes. The advantage of this improvement is not only that each neighborhood can be of a different size, but also comes with a large reduction in computing costs since the value for K does not need to be tuned anymore. While COSA-KNN needs tuning for all combinations of K and

λ, with COSA-λNN only the value for λ needs to be tuned.

Although the scale parameter λ, as well as the homotopy parameter η have default values that are set at $\lambda = \eta = 0.2$, successful results will be shown in the rest of this monograph where the value λ in COSA-λNN is tuned with the Gap statistic procedure (Tibshirani et al., 2001). Concerning the relaxation of the η parameter, so far, a good choice of its value remains dependent on empirical experience. However, in this chapter and the rest of the Monograph we have shortened the homotopy path for η, in all versions of COSA-KNN and COSA-λNN. While at the start we remain with the initialization of $\eta = \lambda$, after our first estimate of $\mathbf{C}$, we directly set $\eta = \infty$. A theoretical framework for the relaxation of η may need further investigation.

What we did not show is whether the COSA-$K\text{NN}_1$ and COSA-λNN are more prone to Type-I errors. Empirical evidence so far suggests that this is not the case. For example, even after the application of the Gap statistic procedure, no clustering structures were found in a data set of 100 objects, and $P = 1000$ or $P = 10,000$ i.i.d. attribute values from a standard normal distribution. In addition, the standard error en the non-smooth fluctuations of the Gap statistics were also a good indication that the solutions obtained with COSA-$K\text{NN}_1$ and COSA-λNN may have been based on noise-only data examples.

4.6 Appendix

4.6.1 Minimizing the A Constrained COSA Criterion

Suppose we have a constrained COSA criterion we wish to minimize.

$$\widehat{\mathbf{W}} = \underset{\mathbf{W}\in\mathcal{W}}{\operatorname{argmin}} \left\{ \sum_{l=1}^{L} \sum_{k=1}^{P} w_{kl} S_{kl} \right\}, \tag{4.30}$$

where for each column l of $\mathbf{W}$ in the space $\mathcal{W}$ we have

$$\sum_{k=1}^{P} w_{kl} \log\left(\frac{w_{kl}}{u_{kl}}\right) \leq \frac{\mathrm{VAR}(S_{kl})}{(\lambda_l^*)^2},$$

and

$$\sum_{k=1}^{P} w_{kl} - 1 = 0, \text{ for all } l.$$

Here, each u_{kl}, S_{kl} and λ_l^* is pre-specified. The values for each u_{kl} are in the real interval $[0, 1]$, and have the property:

$$\sum_{k=1}^{P} u_{kl} = 1. \tag{4.31}$$

The value for each λ_l^* is greater than 0. The value for $\mathrm{VAR}(S_{kl})$ follows from u_{kl}, S_{kl} and λ_l^* . Its definition is

$$\sum_{k=1}^{P} w_{kl}^* S_{kl}^2 - \left(\sum_{k=1}^{P} w_{kl}^* S_{kl} \right)^2, \tag{4.32}$$

where

$$w_{kl}^* = \frac{u_{kl} \exp\left(\frac{-S_{kl}}{\lambda_l^*}\right)}{\sum_{k'=1}^{P} u_{k'l} \exp\left(\frac{-S_{k'l}}{\lambda_l^*}\right)}. \tag{4.33}$$

The KKT Conditions

The solution for $\mathbf{W}$ can be obtained by the Karush-Kuhn-Tucker conditions, these are

(C1) Stationarity:

$$\begin{aligned} 0 \in \partial_{\mathbf{w}} & \sum_{l=1}^{L}\sum_{k=1}^{P} a_{kl} w_{kl} S_{kl} \\ & + \partial_{\mathbf{w}} \lambda_l \left\{ \sum_{l=1}^{L}\sum_{k=1}^{P} w_{kl} \log\left(\frac{w_{kl}}{u_{kl}}\right) - \frac{\mathrm{VAR}(S_{kl})}{(\lambda_l^*)^2} \right\} \\ & + \partial_{\mathbf{w}} \sum_{l=1}^{L} \delta_l \left(\sum_{k=1}^{P} w_{kl} - 1 \right); \end{aligned} \tag{4.34}$$

(C2) Complementary slackness: for each l, we have

$$\lambda_l \left\{ \sum_{k=1}^{P} w_{kl} \log\left(\frac{w_{kl}}{u_{kl}}\right) - \frac{\mathrm{VAR}(S_{kl})}{(\lambda_l^*)^2} \right\} = 0; \tag{4.35}$$

(C3) Primal feasibility:

$$\sum_{l=1}^{L}\sum_{k=1}^{P} w_{kl} \log\left(\frac{w_{kl}}{u_{kl}}\right) - \frac{\mathrm{VAR}(S_{kl})}{(\lambda_l^*)^2} \leq 0, \tag{4.36}$$

and

$$\sum_{k=1}^{P} w_{kl} - 1 = 0, \text{ for all } l. \tag{4.37}$$

(C4) Dual feasibility:

$$\lambda_l \geq 0, \text{ for all } l. \tag{4.38}$$

Stationarity and primal solution

Let us start with the stationarity condition (C1). Since the right-hand side of (4.34) is a sum over L neighborhoods, we can rewrite the derivative with respect to $\mathbf{w}_l$ for each neighborhood separately. Each such function for l can be written as

$$0 \in \partial_{\mathbf{w}_l} \left\{ \sum_{k=1}^{P} \left\{ w_{kl} S_{kl} + \lambda_l \log\left(\frac{w_{kl}}{u_{kl}}\right) \right\} - \delta_l \left(\sum_{k=1}^{P} w_{kl} - 1 \right) \right\}. \tag{4.39}$$

We have now a partial derivative from which we can obtain a primal solution for each w_{kl}, given as

$$\widehat{w}_{kl} = \frac{u_{kl} \exp\left(\frac{-S_{kl}}{\lambda_l}\right)}{\sum_{k'=1}^{P} u_{k'l} \exp\left(\frac{-S_{k'l}}{\lambda_l}\right)} \quad . \tag{4.40}$$

Note that the derivations for the solution $\widehat{w}_{kl}$ are similar to those given in the Appendix of Chapter 2 (Section 2.7.1). However, there is a difference. Here, λ_l is not specified beforehand, it needs to be found such that the solution for $\widehat{w}_{kl}$ obeys the primal feasibility condition.

(C2) Complementary slackness and dual solution

Here, the complementary slackness condition represents the solution we need to search for, if we wish to obtain *strong duality*. After we have obtained the solution (4.40) that solves the primal problem, we are left with the Lagrange dual function for each l:

$$\begin{aligned} g(\lambda_l) &= \sum_{k=1}^{P} \widehat{w}_{kl} S_{kl} \; + \lambda_l \left(\sum_{k=1}^{P} \widehat{w}_{kl} \log \left(\frac{\widehat{w}_{kl}}{u_{kl}} \right) - \frac{\mathrm{VAR}(S_{kl})}{(\lambda_l^*)^2} \right) \\ &= -\lambda_l \log \left\{ \sum_{k=1}^{P} u_{kl} \exp \left(\frac{-S_{kl}}{\lambda_l} \right) \right\} - \lambda_l \frac{\mathrm{VAR}(S_{kl})}{(\lambda_l^*)^2}. \end{aligned} \tag{4.41}$$

Here, $g(\lambda_l)$ is a first lower bound for our restricted minimization problem in (4.30) for any $\lambda_l > 0$ and δ_l.

The highest lower bound is given when we can find the value for λ_l that solves $\frac{\partial g(\lambda_l)}{\partial \lambda_l} = 0$, where

$$\begin{aligned} \frac{\partial g(\lambda_l)}{\partial \lambda_l} &= \sum_{k=1}^{P} \frac{u_{kl} \exp \left(\frac{-S_{kl}}{\lambda_l} \right)}{\sum_{k'=1}^{P} u_{k'l} \exp \left(-\frac{S_{k'l}}{\lambda_l} \right)} \left(-\frac{S_{kl}}{\lambda_l} \right) \\ &\quad - \log \left\{ \sum_{k=1}^{P} u_{kl} \exp \left(-\frac{S_{kl}}{\lambda_l} \right) \right\} - \frac{\mathrm{VAR}(S_{kl})}{(\lambda_l^*)^2} \\ &= \sum_{k=1}^{P} \widehat{w}_{kl} \log \left(\frac{\widehat{w}_{kl}}{u_{kl}} \right) - \frac{\mathrm{VAR}(S_{kl})}{(\lambda_l^*)^2}. \end{aligned} \tag{4.42}$$

However, setting (4.42) equal to zero and solving for λ_l is quite difficult. In accordance with the complementary slackness condition, we need to find the value for λ_l by numerical optimization such that

$$\sum_{k=1}^{P} \widehat{w}_{kl} \log \left(\frac{\widehat{w}_{kl}}{u_{kl}} \right) = \frac{\mathrm{VAR}(S_{kl})}{(\lambda_l^*)^2}.$$

4.6.2 When $\lambda^*_{N_l}$ and λ_l Satisfy Complementary Slackness

Suppose $\lambda^*_l = \lambda_l$, and we have

$$\sum_{k=1}^{P} \widehat{w}_{kl} \log\left(\frac{\widehat{w}_{kl}}{u_{kl}}\right) - \frac{\mathrm{VAR}(S_{kl})}{(\lambda^*_l)^2} = 0, \tag{4.43}$$

then, we have

$$\lambda^*_l = \sqrt{\frac{\sum_k^P \widehat{w}_{kl} S^2_{kl} - \left(\sum_k^P \widehat{w}_{kl} S_{kl}\right)^2}{\sum_{k=1}^P \widehat{w}_{kl} \log\left(\frac{\widehat{w}_{kl}}{u_{kl}}\right)}} \quad . \tag{4.44}$$

Noteworthy is that the condition in equation (4.43) can be achieved for $\lambda^*_{N_l} = \lambda_l$, when

$$\frac{\partial}{\partial \lambda_l}\left\{\lambda_l \sum_{k=1}^{P} \widehat{w}_{kl} \log\left(\frac{\widehat{w}_{kl}}{u_{kl}}\right)\right\} = \sum_{k=1}^{P} \widehat{w}_{kl} \log\left(\frac{\widehat{w}_{kl}}{u_{kl}}\right) - \frac{\mathrm{VAR}(S_{kl})}{(\lambda^*_l)^2} = 0. \tag{4.45}$$

When the Kullback-Leibler divergence is multiplied with λ_l, it becomes a concave function for the value of λ_l, as can be seen in the right panel of Figure 4.15. The maximum of this function is achieved when Complementary Slackness holds given that $\lambda^*_{N_l} = \lambda_l$ (equation 4.43).

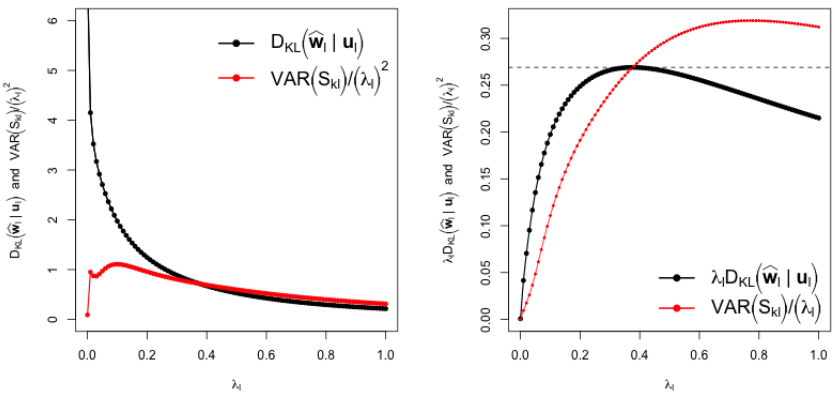

Figure 4.15: In the left panel the Kullback-Leibler divergence (black) and the variance of the dispersions (red) are plotted against λ_l. In the right panel the KL-divergence, as well as the variance term, is multiplied by λ_l.

Derivation

Given that u_{kl} is independent from λ_l, then, the value of λ_l that obeys (4.45), also maximizes

$$\lambda_l \sum_{k=1}^{P} \widehat{w}_{kl} \log\left(\frac{\widehat{w}_{kl}}{u_{kl}}\right). \tag{4.46}$$

Since the Kullback-Leibler divergence is concave when *multiplied* by λ_l, we can find the maximum when we solve

$$\frac{\partial}{\partial \lambda_l}\left\{\lambda_l \sum_{l=1}^{L}\sum_{k=1}^{P} \widehat{w}_{kl} \log\left(\frac{\widehat{w}_{kl}}{u_{kl}}\right)\right\} = 0, \tag{4.47}$$

with respect to λ_l. We can also express this equation as,

$$\frac{\partial}{\partial \lambda_l}\left(-\lambda_l \log\left\{\sum_{k=1}^{P} u_{kl} \exp\left(-\frac{S_{kl}}{\lambda_l}\right)\right\}\right) - \frac{\partial}{\partial \lambda_l}\sum_{k=1}^{P} \widehat{w}_{kl} S_{kl} = 0. \tag{4.48}$$

The derivative w.r.t. λ_l for the first term, the λ_l-inverse exponential mean of the attribute dispersions, is as follows:

$$\frac{\partial}{\partial \lambda_l} - \lambda_l \log\left(\sum_{k=1}^{P} u_k \exp\left(-\frac{S_k}{\lambda_l}\right)\right) = \sum_{k=1}^{P} \widehat{w}_k \log\left(\frac{\widehat{w}_k}{u_k}\right). \tag{4.49}$$

The second part, the derivative of the weighted mean of the attribute dispersions, will be shown based on more elaborate derivations. First we can decompose the weighted mean, defined as

$$\widehat{w}_k S_k = \frac{u_k S_k \exp\left(\frac{-S_k}{\lambda_l}\right)}{\sum_{k'=1}^{P} u_{k'} \exp\left(\frac{-S_{k'}}{\lambda_l}\right)}, \tag{4.50}$$

into the functions $g(\lambda_l)$ and $h(\lambda_l)$, where

$$g(\lambda_l) = u_k S_k \exp\left(\frac{-S_k}{\lambda_l}\right), \tag{4.51}$$

with

$$g'(\lambda_l) = \frac{u_k S_k^2 \exp\left(\frac{-S_k}{\lambda_l}\right)}{\lambda_l^2}, \tag{4.52}$$

and

$$h(\lambda_l) = \sum_{k'=1}^{P} u_{k'} \exp\left(\frac{-S_{k'}}{\lambda_l}\right), \tag{4.53}$$

with

$$h'(\lambda_l) = \frac{1}{\lambda_l^2}\sum_{k'=1}^{P} u_{k'} S_{k'} \exp\left(\frac{-S_{k'}}{\lambda_l}\right). \tag{4.54}$$

Based on the quotient rule we know that

$$\frac{\partial}{\partial\lambda_l}\widehat{w}_k S_k = \frac{g'(\lambda_l)h(\lambda_l) - g(\lambda_l)h'(\lambda_l)}{\left[h(\lambda_l)\right]^2}. \tag{4.55}$$

Thus,

$$\frac{\partial}{\partial\lambda_l}\sum_{k=1}^{P}\widehat{w}_k S_k = \sum_{k=1}^{P}\frac{\partial}{\partial\lambda_l}\widehat{w}_k S_k = \frac{1}{\lambda_l^2}\left\{\sum\widehat{w}_k S_k^2 - \left(\sum\widehat{w}_k S_k\right)^2\right\}. \tag{4.56}$$

When we plug this into (4.56) and (4.49) back into the equation of (4.47), we obtain

$$\frac{\partial}{\partial\lambda_l}\lambda_l D_{KL} = \sum_{k=1}^{P}\widehat{w}_k\log\left(\frac{\widehat{w}_k}{u_k}\right) - \frac{1}{\lambda_l^2}\left\{\sum\widehat{w}_k S_k^2 - \left(\sum\widehat{w}_k S_k\right)^2\right\}. \tag{4.57}$$

Thus, given that u_{kl} is independent from λ_l, the value for λ_l that maximizes

$$\lambda_l\sum_{k=1}^{P}\widehat{w}_k\log\left(\frac{\widehat{w}_k}{u_k}\right) \tag{4.58}$$

is found when

$$\lambda_l = \sqrt{\frac{\sum_{k=1}^{P}\widehat{w}_{kl}S_{kl}^2 - \left(\sum_{k=1}^{P}\widehat{w}_{kl}S_{kl}\right)^2}{\sum_{k=1}^{P}\widehat{w}_{kl}\log\left(\frac{\widehat{w}_{kl}}{u_{kl}}\right)}}\quad. \tag{4.59}$$

Chapter 5

Obtaining L Groups from COSA Distances

Although the main output of COSA are distances that we have labeled "cluster-happy", it is not straightforward to extract L groups from a COSA distance matrix. COSA on its own is not equipped to be compared with L-groups clustering algorithms. COSA does not provide the option of choosing L, instead, it circumvents the pre-specification of the number of clusters by working with N neighborhoods. Thus, an extra step is required when the main objective is to extract L clusters from the COSA distances.

The chapter outline is as follows. In the first section of this chapter we will discuss two popular methods that have been applied already by others to extract L clusters from the COSA-KNN distances. The first method is referred to as Partitioning Around Medoids (PAM; Kaufman & Rousseeuw, 1987), and we will show that it does not provide optimal results when it is applied. The second method is hierarchical clustering that is followed by cutting a dendrogram. We will show that cutting a dendrogram may actually be a good choice for COSA, but only if Ward's minimum variance method is used. However, Ward's method is not sensitive enough to detect small homogeneous groups that have overlapping subsets of attributes, and are in the presence of a large heterogeneous or residual group of noise objects. In these latter data settings, Ward's method splits the large heterogeneous cluster into smaller clusters, and merges the smaller homogeneous clusters, as will be demonstrated in Section 5.2.

To overcome this disadvantage of Ward's method, we present in Section 5.3 a new algorithm that uses a Minimum Variance strategy to Partition In Neighborhoods: MVPIN. This new algorithm applies a restricted form of Ward's method on the distances such that it is able to detect small homogeneous clusters that are similar to each other, in the presence of large heterogeneous clusters. In MVPIN, the merging of clusters by Ward's Minimum Variance method is restricted by the popular Jarvis and Patrick (1973) shared nearest neighbors similarity; that is, clusters are only allowed

the merge if a similarity measure based on the *shared nearest neighbors* between the two clusters is the highest among the similarities that pass a certain threshold.

The performance of MVPIN is demonstrated in Sections 5.4 and 5.5 on a synthetic data set and 11 benchmark data sets. In Section 5.4 the results for MVPIN are compared to Ward's method and to PAM when L, the number of clusters, is specified beforehand. Because the number of clusters is usually not known beforehand, we also show in Section 5.5 the performance of MVPIN when the number of clusters is decided by the Gap statistic procedure. Section 5.6 provides the conclusion and discussion of the chapter.

5.1 Obtaining L Groups from Distances

Among the most popular methods to cluster a distance matrix into L groups are medoid based methods such as Partitioning Around Medoids (PAM; Kaufman & Rousseeuw (1987) and hierarchical clustering methods (Wiwie, Baumbach & Röttger, 2015; de Souto et al. 2008; Kaufman & Rousseeuw, 1990; Jain & Dubes, 1988). To our knowledge, PAM and hierarchical clustering are the only two methods that have been used to cluster COSA distances into L groups: this was done in a study by Jing et al. (2007) and a study by Witten and Tibshirani (2010). Since the main purposes of these studies were not related to COSA, little attention was given to what method should have been used to extract L groups from the COSA distances. From Jing et al. (2007) one can only retrieve that a hierarchical clustering algorithm was used, but what type of hierarchical clustering and what selection procedure to extract L groups remains unclear. The procedures that were used in the study by Witten and Tibshirani (2010), however, can be replicated. These were

1. applying partitioning around medoids (PAM) to the COSA distances;
2. cutting an average linkage dendrogram for the COSA distances at a given height that leads to L groups.

These two choices to extract L groups from the COSA distances are suboptimal. Even for what we call a 'perfect' distance matrix, these choices can easily result in assigning the wrong group labels, as will be shown in the next sections.

5.1.1 Partitioning Around Medoids

The objective of PAM (Kaufman & Rousseeuw, 1987) is to minimize the distances between the medoid and the objects within each cluster. Here, a medoid within a cluster is the object that is designated to be most representative for the cluster. PAM can be applied directly on the COSA distances. Figure 5.1 displays the average linkage dendrogram and the MDS configuration of the COSA-λNN distances, obtained for a simulated data set from the COSA prototype model where there are two small clusters and one large remaining group. Thus, $L = 3$. The colors in Figure 5.1 represent the cluster labels that are the result of PAM when applied directly on the COSA distances.

As can be seen, PAM is able to find the two small homogeneous groups that cluster on overlapping subsets of attributes. However, some of the noise objects are also assigned to one of the two small clusters (colored red instead of grey). Apparently, the COSA distance between each of these wrongly classified noise objects and the medoid of the red cluster, is smaller than the distance between these 'red' noise objects and the medoid of the 'grey' noise objects. This undesirable result is inherent to the objective of PAM, i.e. minimizing the sum over the distances of each object with the medoid of the cluster it belongs to.

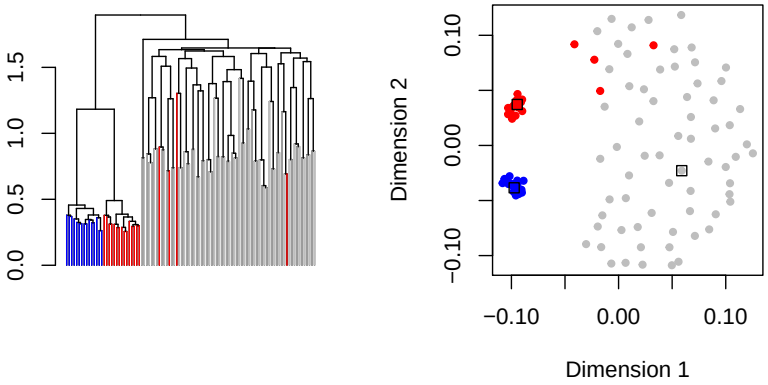

Figure 5.1: An average linkage dendrogram (left panel) and the MDS configuration (right panel) of the COSA distances. The colors of the objects are the labels that are the results of applying PAM to the COSA distances directly. The medoid of each PAM cluster is indicated with a square.

Although close, PAM does not result in an optimal partition for clusters when applied to a distance matrix that represents clusters of different densities. This effect can be easily detected when we compare the resulting clusters from PAM on their 'thightness' and 'separability', as can be done by the silhouette (Rousseeuw, 1987). For every object i we can compute its silhouette, silh(i), by combining two concepts. The first concept is the average distance of object i with all other objects within the cluster of object i, defined as

$$\text{AVW}_i = \sum_{j \neq i} c_{jl^*} D_{ij}[\mathbf{W}] \Big/ \sum_{j \neq i} c_{jl^*} \,, \tag{5.1}$$

where l^* is the cluster that object i is assigned to. The second concept is the minimum average distance of object i to the objects of any other cluster than l^*, defined as

$$\text{MINAVB}_i = \min_{l \,|\, l \neq l^*} \left\{ \sum_{j \neq i} c_{jl} D_{ij}[\mathbf{W}] \Big/ \sum_{j \neq i} c_{jl} \right\}. \tag{5.2}$$

Then, the silhouette of an object is expressed as

$$\text{silh}(i) = \frac{\text{MINAVB}_i - \text{AVW}_i}{\max\{\text{AVW}_i, \text{MINAVB}_i\}}. \tag{5.3}$$

The silhouette of object i is an indication of how well object i matches its cluster. The closer the value is to 1, the better object i matches its cluster. When the silhouette of object i becomes negative (with a minimum of -1), it is an indication that it would have been more natural to assign object i to its closest neighboring cluster. If an object is on its own a cluster of size 1, then its silhouette is set equal to 0, giving it a 'neutral' value.

When we use the graphical display of the silhouette (Rouseeuw, 1987), we can see four 'red' clustered objects of which three have obtained a negative silhouette and one has an 'outlying' low positive silhouettte, see Figure 5.2. These are the four wrongly clustered objects. Having many positive silhouettes within a cluster shows

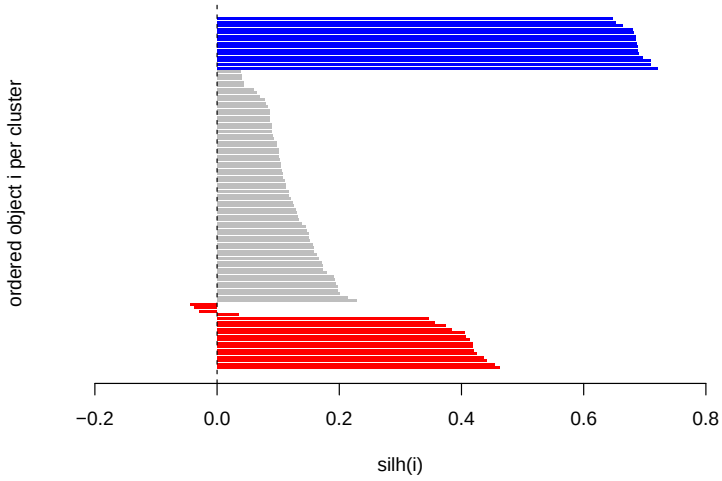

Figure 5.2: Graphical Display of silhouettes of the PAM clustering solution, each colored horizontal bar respresents an object. The length of the bar represents the object's silhouette.

that the cluster is well separable, and the higher the values of the positive silhouettes, the thighter the clustering. Rousseeuw (1987) proposed to use the average of the silhouettes within a cluster as a measure of validation of that cluster, and suggests that the average over all the silhouettes can be used as a measure of validation to evaluate the strength of the resulting partition of the set of objects, and therefore may serve for the selection of the optimal number of clusters in a data set. Although for low-dimensional data settings ($N < P$), in an extensive study about cluster validation measures by Arbelaitz et al. (2013) it is shown that the silhouette (still) outperforms more recent cluster measures for choosing the number of clusters.

A reason that PAM does not show an optimal performance on the COSA distances is that PAM is sensitive to violations of the triangular inequality. In a strict sense,

the COSA distances are non-metric dissimilarities, i.e., the distances may violate the triangular inequality. The higher the differences in the subsets of attribute weights in COSA, the more likely it is that the COSA distances will violate the triangle inequality property (see Chapter 2). Suppose two objects, o_1 and o_2, are dissimilar to each other, but similar to a third object o_3. Then, it may still be possible that when o_3 is a medoid, that o_1 and o_2 end up in the same cluster through a violation of the triangular inequality. The path from o_1 to o_2 via the candidate medoid o_3 can be shorter than the direct path from o_1 to o_2:

$$D_{12} > D_{13} + D_{23}. \tag{5.4}$$

Such triangular violations can complicate the discovery of homogeneous clusters in PAM. For example, when the medoid 3 is being updated to medoid 2, suddenly the same group would become more heterogeneous. The larger the number of distances that violate the triangular inequality, the more the medoids based algorithms are affected, and the more likely it becomes that the underlying clustering structure cannot be revealed. For a further explanation, see Baraty, Simovici, and Zara (2011). Thus, misrepresentations of within-cluster homogeneity can obscure the partitioning in the data.

We can avoid the violations of the triangle inequality by applying PAM directly to the Euclidean distances of a best-fitting MDS configuration of $N - 1$ dimensions. In De Leeuw and Groenen (1997) it is shown that the MDS configuration in $N - 1$ dimensions results in metric Euclidean distances for which the global minimum of the STRESS function is achieved (for the STRESS function, see Chapter 1, page 16, equation (1.12)). However, applying PAM to the best fitting Euclidean distances, or any of the Euclidean distances that correspond to lower-dimensional configurations, results in more misclassified objects for the particular data sets that are generated from the prototype model. Although the minimized STRESS function gives a configuration matrix of which the Euclidean distances between the objects with large COSA distances remain relatively large (as compared to classical scaling solutions), it may come at the cost of the distances between the objects of the two smaller clusters. Thus, fixing the triangular inequality violations with the use of MDS does not provide better results for PAM.

The worst performance of PAM in combination with an MDS configuration is when default settings of the implementations of `smacof()` and `pam()` are applied in `R`. Extracting the configuration matrix $\mathbf{X}$ in `R` using the default options in `smacof()`, and then feeding this configuration matrix into `pam()` with its default options, gives results as shown in Figure 5.3. Note that this is an easily made, but inconsistent combination of choices. When the input of `pam()` is a configuration matrix, it computes the Manhattan distances in $\mathbf{X}$, while the specific MDS configuration matrix is actually supposed to contain Euclidean distances. The results of using this detrimental strategy are shown in Figure 5.3. While in Figure 5.1 the two smaller clusters were detected, now the two smaller clusters are interpreted as one red cluster, and the noise objects are split into two groups. These results are consistent with findings in Van der Laan et al. (2003); PAM may experience difficulties in recognizing relatively small clusters.

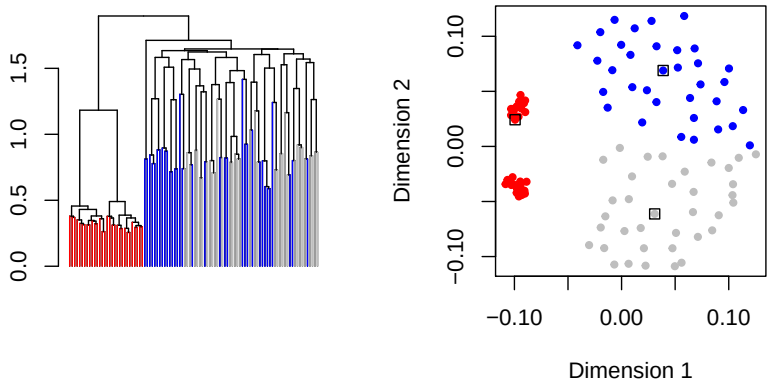

Figure 5.3: The resulting L groups of PAM on a two-dimensional MDS configuration of the COSA distances, shown in an average linkage dendrogram (left), and in the MDS configuration (right), in which the medoids are indicated as points within a square.

5.1.2 Cutting the Dendrogram

Another common approach to extract L clusters from a distance matrix is by cutting a dendrogram. Here, we will show with an example why cutting an average linkage dendrogram of a perfect COSA distance matrix is suboptimal. Since COSA can reveal clusters with different densities, it is common to see in average linkage dendrograms that all members in a group will meet each other at different heights. However, one of the standard rules in cutting a dendrogram, is to cut at one specific height only (Jain & Dubes 1988, Kaufman & Rousseeuw, 1990).

Suppose we wish to cut the average linkage dendrogram from the default COSA-λNN results on a simulated data set from the COSA prototype model, displayed in the left panel of Figure 5.4. The ground truth seems to be perfectly revealed: a clustering structure of two homogeneous separate groups that seem to nest together into one bigger group (due to sharing of a subset of attributes), and a remainder heterogeneous group, containing the noise objects (in grey). Although from a pure COSA perspective one could argue that the results of one cut only will be good enough, e.g., a cut that results in two groups of 15 objects and 70 singleton objects, it is not possible to extract the $L = 3$ groups in just one cut. Note that this may be undesirable when the noise objects could be interpreted as a heterogeneous group with disturbances in the (gen)omics system (Van Wieringen & Van der Vaart, 2015).

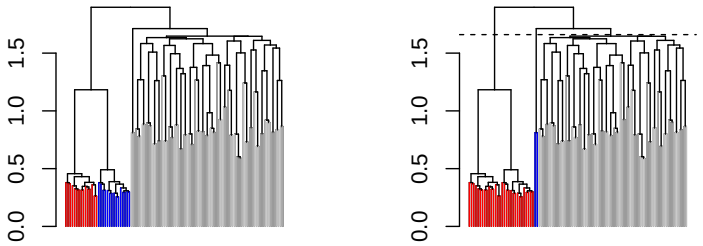

Figure 5.4: The average linkage dendrograms of the COSA-λNN distance obtained from the data set that was generated from the prototype mode. In the left panel each object is labeled with the color of its cluster group according to the truth structure, in the right panel the objects are colored according to their group label obtained by cutting the dendrogram.

If the only purpose of performing COSA would be to fit a dendrogram on the COSA distances that needs to be cut in L groups, than better results can be expected from a density based linkage strategy. While linking an out-of-cluster object to an already large group may not have a large impact on the average distance, it will have a large impact on the density of the distances within that group. Thus, linking according to a strategy of a minimum variance increase, such as Ward's minimum variance method (Ward, 1963; Wishart, 1969), would provide better results for the sole purpose of finding $L = 3$ groups; the two small homogeneous clusters and the residual group of noise objects (see Figure 5.5). This is a finding in line with Jain and

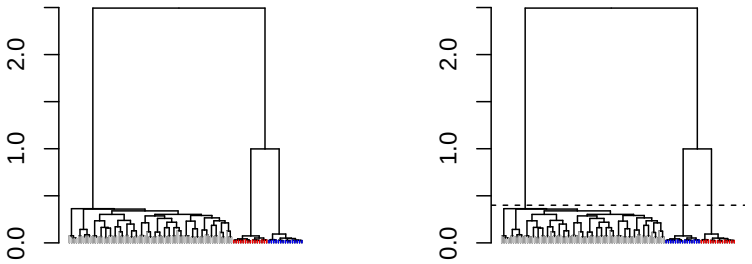

Figure 5.5: The Ward's method dendrogram for the COSA distances of the prototype model data set with each object labelled with a color of its cluster group according to the ground truth (on the left), and the same dendrogram with each of the objects colored according to the group label obtained from cutting the dendrogram (on the right).

Dubes (1988), in which an overview of several comparative studies was presented with the conclusion that for the purpose of dendrogram cutting at a specific height to find L groups, Ward's method outperforms other hierarchical clustering methods (among which single, average, and complete linkage). For the purpose of plotting the COSA distances, however, the dendrograms based on Ward's minimum variance method are suboptimal. The distances between the noise objects are supposed to be much larger than the distances of the objects within the homogeneous clusters.

5.2 A Closer Look at Ward's Method

Originally, Ward's method has been proposed as an agglomerative procedure for forming hierarchical groups of mutually exclusive clusters for rating values in a set of N objects and P attributes. Here, the N objects are progressively fused into hierarchical clusters by applying an algorithm to minimize a certain objective function that can "be any functional relation that an investigator selects to reflect the relative desirability of groupings" (Ward, 1963). As an example for an objective function that would represent loss of information, Ward (1963) used the 'error sum of squares'. Applying the algorithm of Ward's method based on the error sum of squares loss function is also referred to as Ward's minimum variance method.

Using Ward's method to find L groups "probably will yield a good solution, although it may be one that does not optimize the objective function for the specified number of groups" (Ward, 1963), since the solution is restricted to a hierarchical grouping only. Thus, even though the title of the study by Ward (1963) is 'Hierarchical grouping to optimize an objective function', the method does not globally optimize the criterion for given L. In Ward's method the rate of change of the criterion, e.g., as in (5.15), is minimized with each union of two clusters starting from $L = N$ singleton clusters towards L equal to the desired number of clusters. At each step, the unification of the two 'intermediate' selected clusters produces the least impairment of the optimal value of the criterion.

5.2.1 The Distance-based Version of Ward's Method

Wishart (1969) generalized Ward's minimum variance method by rewriting it in terms of the distance-based update formula of the Lance-Williams family of clustering algorithms (Lance & Williams, 1967). The Lance-Williams updating formula describes how to compute distances between clusters that are formed by merging two existing clusters, and the remaining clusters. As we have seen in Chapter 2, using *Huygens' principle*, the error sum of squares within clusters can be rewritten as the sum of squared Euclidean distances. Using the squared Euclidean distances, Wishart (1969) derived the Lance-Williams update formula for Ward's method. Here, we will use the specific update formula on the squared COSA distances in equation (5.7).

Let each object initially be regarded as a singleton cluster, and the 'Ward' distance between two singleton clusters l and l' be defined as the COSA distance of object l and l', i.e.

$$\delta(l, l') = D_{ll'}[\mathbf{W}], \tag{5.5}$$

where, initially, the cardinality of cluster l, and l', are

$$N_l = N_{l'} = 1. \tag{5.6}$$

Thus, at the start, the between-clusters distance matrix Δ would be of size $N \times N$, where each element denoted by $\delta(l, l')$, is a COSA distance. Having initialized the Ward distances, we can recursively reduce the number of clusters by one. We find those two (still singleton) clusters for which $\delta(l, l')$ is smallest, and then fuse l and

l' into a new cluster of two objects. The number of clusters is now reduced by one. Then, to compute Δ again, which is now reduced to an $(N-1)\times(N-1)$ matrix, the Ward distances between the new cluster and all other clusters are updated via:

$$\delta^2\left(l\cup l', l''\right) = \frac{N_l + N_{l''}}{N_l + N_{l'} + N_{l''}}\delta^2\left(l, l''\right) + \frac{N_{l'} + N_{l''}}{N_l + N_{l'} + N_{l''}}\delta^2\left(l', l''\right) - \frac{N_{l''}}{N_l + N_{l'} + N_{l''}}\delta^2\left(l, l'\right), \tag{5.7}$$

where $l \neq l'$, $l' \neq l''$, $l'' \neq l'''$. In the subsequent steps the update formula from equation (5.7) is recursively applied to those two clusters l and l' that have the smallest Ward between-cluster distance until L clusters are reached and Δ is of size $L \times L$.

5.2.2 A Drawback of Ward's method

Although we have seen in the previous sections that for $L = 3$, the cluster labels obtained with Ward's method (Figure 5.5) are prefered over those obtained with PAM (Figure 5.1), or cutting the average linkage dendrogram (Figure 5.4), Ward's method has a disadvantage that needs to be taken into account when applied to COSA distances. Suppose we have a clustering structure that consists of a large heterogeneous group of noise objects and two small homogeneous groups that cluster exactly the same way on a large overlapping subset of attributes, but they also have a small unique subset of attributes each. In such a situation the update formula from equation (5.7) would assure a minimum increase at a step where the two homogeneous clusters are merged, instead of merging subclusters or objects from the large heterogeneous cluster.

Suppose we modify the COSA prototype model (from Figure 1.2 in Chapter 1) into a data model with an extra large subset of overlapping attributes, as can be obtained in Figure 5.6. Instead of having an overlapping subset of 15 attributes, we now have an overlapping subset of 27 attributes and two unique subsets, consisting of only 8 attributes each. From a data set generated from the 'extra overlap' model, compared to a data set from the COSA prototype model, we see in Figure 5.7 that Ward's method breaks down for $L = 3$ clusters (also does PAM, and cutting the average linkage dendrogram).

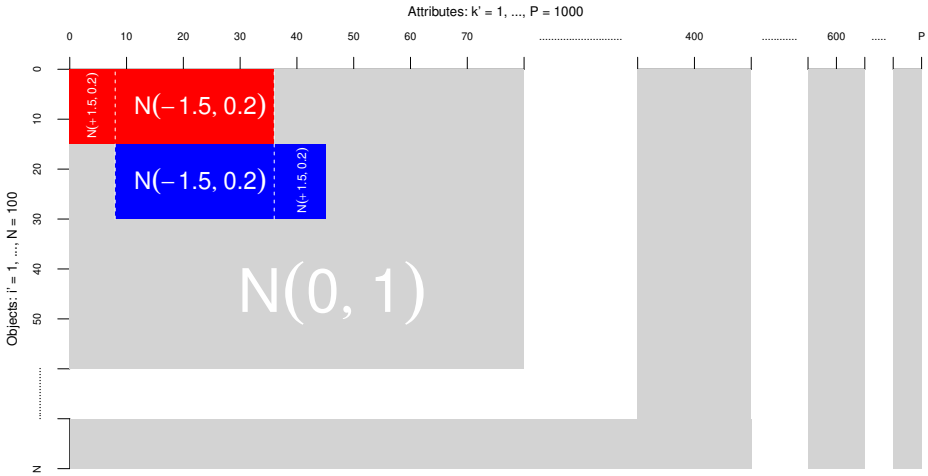

Figure 5.6: Extra Overlapping Attributes model for a data set with 100 objects and 1,000 attributes (of which only a limited number is shown). There are two groups of 15-objects (red and blue) each clustering on a subset of 35 attributes. On an overlapping subset of 27 attributes the clustering is exactly the same for the two groups. After generating a data set from this model, each attribute is scaled to unit variance and zero-mean.

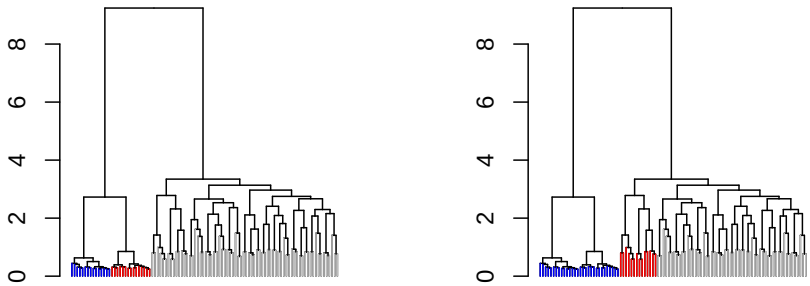

Figure 5.7: The Ward's method dendrograms fitted on the COSA distances of the data set from the 'extra overlap' model. In the left panel each object is colored according to the model in Figure 5.6, in the right panel each object is colored according to the group label obtained from cutting the dendrogram into three groups.

5.3 MVPIN

To obtain L groups of objects from COSA distances there are two properties that could be exploited. The first property is that the within cluster distances should be small. With respect to this first property we have seen that Ward's method performs well, by only merging clusters as long as the increment of the sum of the squared COSA distances is the smallest. The second property, that the within-cluster distances for COSA are based on neighborhoods of overlapping nearest neighbors, is not (yet) exploited. To be able to recover the small but stable homogeneous groups with a largely similar clustering pattern, we will also exploit this latter property.

We propose a new algorithm that can be implemented on a distance matrix, called MVPIN. While using Ward's Minimum Variance method, the algorithm Partitions In shared Neighborhoods. In MVPIN we combine Ward's method based on a modified version of the Jarvis and Patrick (1973) similarity measure that is based on nearest neighbors. In essence, MVPIN is an agglomerative clustering algorithm that applies Ward's minimum variance method as a linkage strategy on the COSA distances; however, the agglomeration process is steered differently. This different steering process renders MVPIN to be better able to recover unequally sized clusters, as long the number of shared nearest neighbors within each cluster is high.

5.3.1 Shared Nearest Neighbors

Instead of merging two clusters l and l' for which the Ward between-cluster distance is smallest, we pose an extra condition that the shared number of nearest neighbors between l and l should be the highest. Here, the shared number of nearest neighbors is based on a similarity measure between two objects as explained in Jarvis and Patrick (1973). Let K be the number of objects in a neighborhood, i.e. the neighborhood size. Then, the definition of the shared nearest neighbors similarity measure between objects i and j is

$$\widetilde{n}_{ij}(K) = \begin{cases} 1 + |\mathrm{KNN}(i) \cap \mathrm{KNN}(j)|, & \text{for } i \in \mathrm{KNN}(j) \text{ and } j \in \mathrm{KNN}(i) \\ 0, \text{ otherwise.} \end{cases} \tag{5.8}$$

Thus, $\widetilde{n}_{ij}(K)$ can be seen as the intersection of two neighborhoods of i and j given that i and j are in each others neighborhood. The neighborhoods $\mathrm{KNN}(i)$ and $\mathrm{KNN}(j)$ are each based on the COSA distances, as was defined in equation 2.22 of Chapter 2 (p. 30). According to Jarvis and Patrick (1973), the definition of the shared nearest neighbors similarity between two clusters l and l' is

$$\widetilde{N}_{ll'}(K) = \max_{\{i,j\}} \{\widetilde{n}_{ij}(K) \,|\, i \in C_l, j \in C_{l'}\}\,. \tag{5.9}$$

Note that the similarity in equation (5.9) can be seen as the inverse distance of the single linkage function between two clusters l and l'. Instead of linking the two clusters with the minimum cluster distance, we can link the two clusters with the maximum cluster similarity.

To assure the shared nearest neighbors similarity contributes to recovering a stable clustering structure in the data, a good value for K needs to be specified. The larger the value for K, the more object pairs will have a high number of shared nearest neighbors. Thus, by increasing K, eventually the between-cluster objects will have a similar number of shared nearest neighbors as within-cluster objects. Similarly, when the value for K is set to be too small, then, depending on the size of the cluster, the within-cluster objects may end up with too few shared nearest neighbors, or perhaps even none.

The idea of the similarity measure in MVPIN is that the value for K is chosen such that $K+1$ equals the size of the smallest 'clearly' detectable cluster in the data. Since the clustering structure in the data is unknown, thus we need a strategy to approximate K. We propose a strategy that approximates the value of K by searching for the smallest homogeneous group of objects that may be indicative for the smallest detectable cluster. Suppose that for each value of K the average number of shared nearest neighbors is expressed as a proportion of K, i.e.

$$p_K = \frac{1}{NK} \sum_{i=1}^{N} \sum_{j \in \mathrm{KNN}_{(i)}} \widetilde{n}_{ij}(K), \tag{5.10}$$

which is referred to as the proportion of shared nearest neighbors out of K nearest neighbors. Then, the idea is that the smallest detectable cluster in the data has size $K+1$, where the value for K corresponds to the first local maximum for p_K, where $p_{K-1} < p_K > p_{K+1}$. For an example, see Figure 5.8, where the input consisted of the COSA distances for a data set that was generated from the COSA model with extra overlapping attributes (Figure 5.6). In Figure 5.8 the value for K that corresponds to the first local maximum is equal to $K = 14$. Not surprisingly, $K+1$ corresponds to the size of the two smallest clusters, i.e. $\min_l(N_{\mathcal{C}_l}) = 15$. Here, the first local maximum exactly represents the total number of nearest neighbors that have the same cluster membership for an object of one of the two smallest clusters.

When the first local maximum at p_K is higher than the straight line between p_1 and p_{N-1}, then we set $\widehat{K}$ equal to the value for K that corresponds to this first local maximum. If, however, the first local maximum for K does not have a higher proportion of shared nearest neighbors than 'expected' with a linear increase from $K = 1$ to $K = N - 1$, then we set $\widehat{K}$ equal to $N - 1$, such that MVPIN reduces Ward's method (as will become clear in the next paragraph). Empirical evidence so far suggests that when the value for $\widehat{K}$ corresponds with a first local maximum for p_K that is lower than the linear 'expectation', then MVPIN becomes sensitive for small homogeneous clusters that are due to noise only.

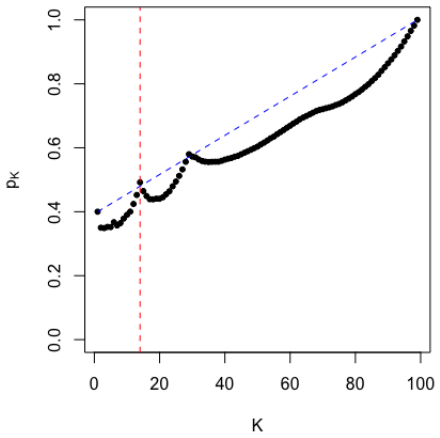

Figure 5.8: The proportion of the average of the shared nearest neighbors out of K, for $K \in \{1, \dots N-1\}$. The vertical red line indicates $\widehat{K}$, the first robust local maximum for p_K. The blue straight line connects $p_{K=1}$ with $p_{K=N-1}$

5.3.2 The MVPIN Update Formula

MVPIN is an algorithm based on the update formula for Ward's method on the COSA distances, however it is steered by the single linkage function $\widetilde{N}_{ll'}(K)$ from equation (5.9) based on the shared neighbors similarity $\widetilde{n}_{ij}(K)$ from equation (5.8). MVPIN only considers a fusion for clusters l^* and l'^* if

$$\{l^*, l'^*\} \in \underset{\{l,l'\}}{\operatorname{argmax}} \left(\widetilde{N}_{ll'}(K) \right), \tag{5.11}$$

where $l \neq l'$, and therefore $l^* \neq l'^*$. Thus, the update formula now becomes

$$\begin{aligned} \delta^2 \left(l^* \cup l'^*, l''^* \right) = & \frac{N_{l^*} + N_{l''^*}}{N_l + N_{l'^*} + N_{l''^*}} \delta^2 \left(l^*, l''^* \right) + \frac{N_{l'^*} + N_{l''^*}}{N_l + N_{l'^*} + N_{l''^*}} \delta^2 \left(l'^*, l''^* \right) \\ & - \frac{N_{l''^*}}{N_l + N_{l'^*} + N_{l''^*}} \delta^2 \left(l^*, l'^* \right), \end{aligned} \tag{5.12}$$

where $l^* \neq l'^*$, $l'^* \neq l''^*$, $l''^* \neq l'''^*$.

The first fusion step of the two singleton clusters l^* and l'^* are now based upon the minimized COSA distance for the maximized shared nearest neighbors similarity, i.e.

$$\min_{\{l^*,l'^*\}} \delta(l^*, l'^*) = \min_{\{l^*,l'^*\}} \left\{ D_{l^* l'^*}[\mathbf{W}] \;\middle|\; \{l^*, l'^*\} \in \underset{\{l,l'\}}{\operatorname{argmax}} \left(\widetilde{N}_{ll'}(K) \right) \right\}. \tag{5.13}$$

Note that MVPIN reduces to Ward's method when $\widehat{K} = N - 1$ since for all object pairs we will see that $\widetilde{n}_{ij}(\widehat{K}) = N - 1$.

The search for L clusters with MVPIN starts with finding $\widehat{K}$. Having set $\widehat{K}$, we compute the shared nearest neighbors similarity measures, $\{\widetilde{n}_{ij}\}$. Then, we apply the update formula in equation (5.12) to the COSA distances and the shared nearest neighbors similarity measures until the number of L clusters is reached. Thus, the algorithm can be described as

MVPIN
0 : Set: L;
1 : Compute: $\widehat{K}$, $\widetilde{n}_{ij}(\widehat{K}) \forall \{i, j\}$;
2 : Initialize: Δ, where each $\delta(i, j) = D_{ij}[\mathbf{W}]$;
3 : Loop {
Reduce size of Δ using equation (5.12) based on $\widetilde{N}_{ll'}(\widehat{K})$ (5.9)
Update $\mathbf{C}$
If (size of Δ is $L \times L$){Go to 4}
3 : }
4 : Output: $\mathbf{C}$.

(5.14)

5.4 MVPIN in Action

5.4.1 Results on Simulated Data

In Figure 5.9 three steps of the algorithm in (5.14) are shown for the synthetic data set of the COSA model with extra overlapping attributes from Figure 5.6. Objects that are merged into a group have been connected here with black lines. The connections are shown in a two-dimensional MDS configuration of the COSA-λNN distances for the data set that was also used in Figure 5.8. Thus, for this particular data set we have a neighborhood size of $\widehat{K} = 14$.

In the first (left) panel of 5.9, it shown which (sets of) objects are allowed to be linked with the update formula (5.12), based on the restriction in (5.11). Between the objects within the small homogeneous objects, edges are drawn indicating that these objects are allowed to be connected under the restriction. Then after some iterations, we see in the second (middle) panel in Figure 5.9 the intermediate results for MVPIN with a (maximum) shared nearest neighbors similarity of 9 or higher. Due to the shared nearest neighbors restriction in (5.11), we see that the small homogeneous clusters cannot be connected to each other, and that the noise objects start to merge into groups of objects, even though the noise objects have larger distances, than the distances between the two homogeneous clusters. The last panel (right) in Figure 5.9 displays the final results of MVPIN for $L = 3$. Already at a maximum similarity of five shared nearest neighbors, MVPIN detects two small homogeneous clusters and a large residual group of noise objects.

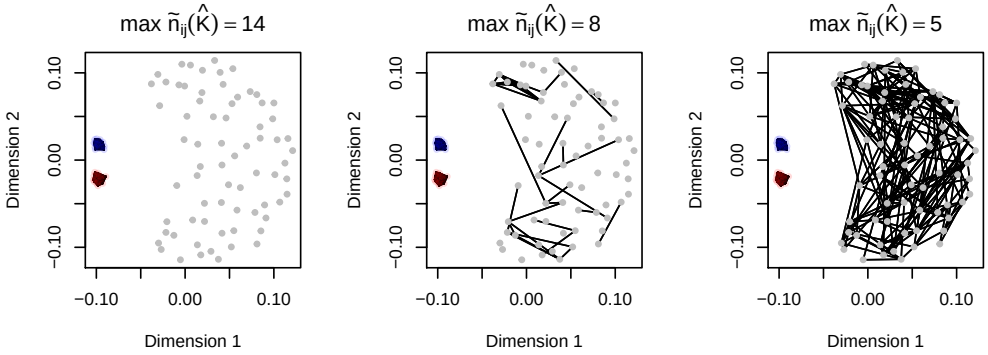

Figure 5.9: Searching for $L = 3$ when applying MVPIN to the COSA distances for an optimal value of K equal to 14, $\widehat{K} = 14$. It is of interest to note that the graphical display is shown on a two-dimensional MDS configuration of the COSA distances.

Repeating this particular simulation experiment on 100 simulated data sets, generated from the extra overlap model in Figure 5.6, resulted into a complete recovery of the clustering structure for 28 (out of 100) times for MVPIN, Ward's method was able to do so only for only 4 (out of the 100) data sets, and PAM only once. The clustering structure was never recovered by cutting an average linkage dendrogram at the specific height that would result into a partition of $L = 3$ groups.

5.4.2 Comparing MVPIN, WARD and PAM

Except the data from the extra overlapping attributes model, on all the earlier data examples no differences occurred between the performance of Ward's method and that of MVPIN on the optimized COSA distances for either COSA-KNN or COSA-λNN. For COSA distances of the simulated data sets from the previous chapters, MVPIN and Ward's method were able to recover the clustering structure perfectly, and performed equally well (or even better) than PAM, and cutting the average linkage dendrogram.

Real Data Sets

Results have been obtained for Ward and MVPIN applied to the optimized COSA distances for 12 oncological benchmark data sets. Apart from the ApoE3 data (Chapter 3) and the Breast cancer data (Chapter 4), the remaining oncological benchmark data sets have not been described yet. Note that the details for the data sets will be described in Chapter 6. For now, Table 5.1 shows the details of the 'true' clustering structure that was supposed to be recovered from the data sets.

Table 5.1: The benchmark data sets consist of a metabolomics data set (nr. 0) and microarray gene expression data sets (nrs. 1 - 10).

#	Data Name	L	$N(N_1 + \ldots + N_L)$	P	Source
0	ApoE3 Mice	2	38 (18 + 20)	1,550	Damien et al. (2004)
1	Brain	5	42 (10 + 10 + 10 + 4 + 8)	5,597	Pomeroy (02)
2	Breast	2	276 (183 + 93)	22,215	Wang et al. (05)
3	Colon	2	62 (22 + 40)	2,000	Alon et al. (99)
4	Leukemia	2	72 (47 + 25)	3,571	Golub et al. (99)
5	Lung1	2	181 (150 + 31)	12,533	Gordon et al. (02)
6	Lung2	2	203 (139 + 64)	12,600	Bhattacharjee et al. (01)
7	Lymphoma (DLBCL)	3	62 (42 + 9 + 11)	4,026	Alizadeh et al. (00)
8	Prostate	2	102 (50 + 52)	6,033	Singh et al. (02)
9	SRBCT	4	63 (23 + 8 + 12 + 20)	2,308	Kahn (01)
10	SuCancer	2	174 (83 + 91)	7,909	Su et al. (01)
11	SuCancer	2	174 (83 + 91)	7,909	Su et al. (01)
12	X-Perou	4	62 (8 + 7 + 11 + 36)	1,753	Perou et al.(00)

Rand Index Results

Instead of visualizing the assigned clustering structure in a dendrogram or an MDS configuration for each of the 12 benchmark data sets, we will use the Rand Index (Rand 1971) to show the performance of recovering of the clustering structure. A Rand Index (RI) of 1 indicates perfect retrieval of the ground truth clustering structure, and a Rand Index of 0 is an indication of anti-clustering, i.e., all pairs of objects that should belong to the same cluster are in different clusters and vice versa. The Rand Index results of the clustering algorithms for the 12 benchmark data sets are shown in Figure 5.14.

From the results on the benchmark data sets, there is no clear winner among MVPIN, Ward and PAM. The merit of MVPIN is mainly visible for the DLBCL data set with a positive difference of 0.246 on the Rand Index for cluster recovery. For all other data sets we see that MVPIN, Ward's method, and PAM have a very similar performance. It is of interest to note that PAM performs best with a Rand index difference 0.047 for the Colon data set, and worst for the DLBCL and Lung1 data sets.

We are aware that it is often argued that there are other indices in cluster validation research that make better use of the [0,1] interval to measure the similarity between clustering structures than the Rand Index (Romano et. al, 2016; Steinley, 2004), e.g. the adjusted Rand index (ARI; Hubert & Arabie, 1985), Normalized Mutual Information (NMI; Strehl & Gosh, 2002), and the adjusted Mutual Information (AMI; Vin, Eps, & Bailey, 2010). However, using either of the other measures (ARI, NMI, AMI) resulted in the same conclusions for the comparison of MVPIN, WARD and PAM. The reason we used the Rand Index will become clear in Chapter 6, where we compare the current chapter's results of MVPIN with those from other studies where the Rand Index was used (Arias-Castro & Pu, 2017; Deng et al., 2011).

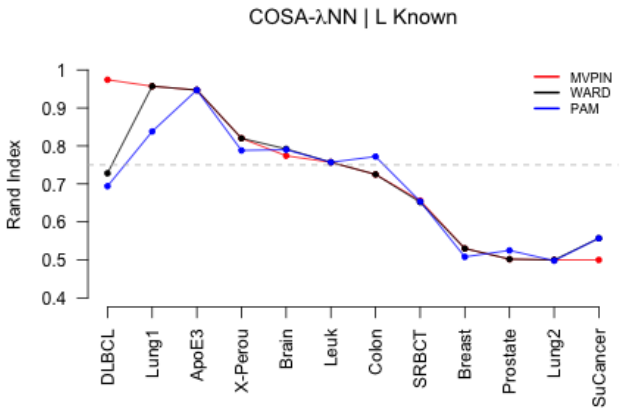

Figure 5.10: The Rand Index computed over the resulting partitions from MVPIN (red), Ward (black) and PAM (blue). The algorithms have been applied to the optimized COSA-λNN distances for each data set. The horizontal dashed grey line indicates a Rand Index of 0.75, and the data sets are ordered corresponding to the results of MVPIN.

5.5 Gap statistic: A strategy for when L is unknown

So far, we assumed that L, the optimal number of clusters in the data, was known. In practice this is seldomly the case. As will be shortly discussed in Chapter 6, strategies employed to choose the optimal value of L are elaborately explored, but remains an unsolved ongoing topic of research (Hancer et al., 2017; Arbelaitz et al. 2013; Lee & Olafsson, 2013; Tibshirani & Walther, 2005; Dudoit & Fridlyand, 2002). In this Section, we will consider L as a tuning parameter that needs to be optimized for PAM, Ward's methods and MVPIN. For PAM we will follow the example set in Kaufman and Rousseeuw (1990) to select the value of L that corresponds with the highest average silhouette over the objects, i.e. the average based on equation (5.3). For Ward's method and MVPIN we will rely on an adjusted version of the Gap statistic procedure (Tibshirani et al., 2001), described in Chapter 2.

Ward's method is motivated by the minimization of a criterion, i.e. any objective function that can evaluate the loss of information. By replacing the squared Euclidean distances with the squared COSA distance, the criterion of Ward's minimum variance method, referred to as the objective function in Ward (1963, p. 237), or Wishart (1969, p. 166) becomes

$$Q_{Ward}(L) = \frac{1}{2}\sum_{l=1}^{L}\sum_{i=1}^{N}\sum_{j=1}^{N} c_{il}c_{jl}D_{ij}^{2}[\mathbf{W}], \tag{5.15}$$

where $c_{il} = 1$ when $i \in C_l$. We find that due to the left-skewed density of the COSA distances, this criterion has a polynomial growth rate for decreasing L. For Ward's method we will apply the Gap statistic procedure to the criterion from equation (5.15) to determine the number of clusters. Our findings so far show that the Gap statistic

procedure applied to the criterion directly would always result in favoring the smallest number of groups, while the GAP statistic procedure applied to the logged criterion has the tendency to favor too large L. Best results were obtained when the GAP statistic was applied to the square root of the criterion, i.e.

$$\text{GAP}_{\surd} = E_{Q^\circ}\left[\sqrt{Q^\circ_{Ward}(L)}\right] - \sqrt{Q_{Ward}(L)}. \tag{5.16}$$

Here, E_{Q° denotes the expectation of the Ward criterion of COSA distances that were optimized for comparable data sets in which only noise was present. Last, to make sure all COSA distances of the permuted data sets as well as the real data set were comparable, we normalized the COSA distances to have sum of squares equal to N.

MVPIN is also an approximate heuristic directed by the rate of change of the criterion in(5.15), although with a restriction on the maximum number of shared nearest neighbors, resulting from (5.11). Therefore, the criterion for MVPIN is defined as the sum of rates of changes:

$$Q_{MVPIN}\left(L \,|\, \widehat{K}\right) = \frac{1}{N-L}\sum_{m=N}^{L+1} \min\left\{\delta^2\left(l^* \cup l'^*\,,\, l''^* \,|\, \Delta_{m\times m}\right)\right\}. \tag{5.17}$$

The criterion in (5.17), is the average of the minimum distances obtained from each of the $N-L$ restricted Ward distance matrices $(\{\Delta_{m\times m}\}_{m=N}^{L})$ that are recursively reduced from size $N\times N$ to $L\times L$, via the update formula in equation (5.12). Note that the procedure to obtain the estimate $\widehat{K}$ is based on equation (5.10) described in Section 5.3.1.

The Gap statistic procedure applied to the criterion for MVPIN, showed consistent results for the Gap, $\text{Gap}_{\log}$, and $\text{Gap}_{\surd}$, and for both the COSA-λNN or COSA-KNN distances applied to all data examples in this chapter. See Figure 5.11 for a visualization of the gap statistics procedure for MVPIN applied to the COSA distances of the simulation example from the extra overlapping attributes model. Applying the Gap statistic to the COSA distances of the simulated data example from Figure 5.6, and to the COSA distances of $B = 25$ permuted versions of this data set, resulted in the correct selection of L, i.e. $L = 3$ clusters.

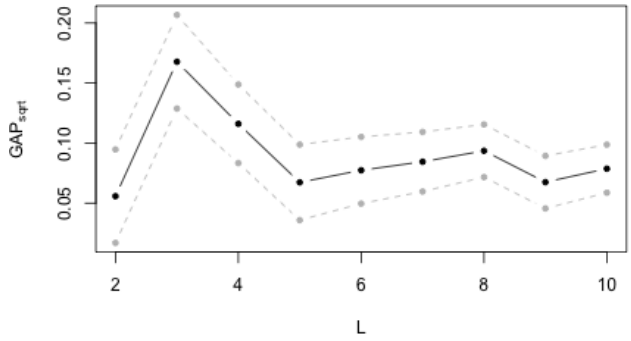

Figure 5.11: For MVPIN we obtain $L = 3$ for the value with the maximum gap statistic. The dashed grey lines are the 1 standard deviation upper and lower bounds of the Gap statistic.

5.5.1 Benchmark Data Sets Results

When applying the Gap statistic procedure in MVPIN and Ward, and the average silhouette for PAM, to determine the number of clusters L, the performances of the three methods on the COSA-λNN distances, are less similar. As can be seen in Figure 5.12, MVPIN is five (out of twelve) times a clear winner in the recovery of the ground truth clustering structure. For the results we used $B = 25$ permuted 'noise' data sets over which the average criterion could be computed to obtain an estimate of the expected criterion under noise only, i.e. $E_{Q^\circ}\left[\sqrt{Q^\circ_{Ward}(L)}\right]$.

Noteworthy is the bad performance of Ward's method on the Lung1 data set. Here, the max Gap statistic for the criterion for Ward's method occurs at $L = 5$ clusters, where the true clustering structure has $L = 2$. Apart from the Lung1 data set, we see that the performance of Ward's method and PAM on the COSA-λNN distances is very similar with a slight advantage for PAM.

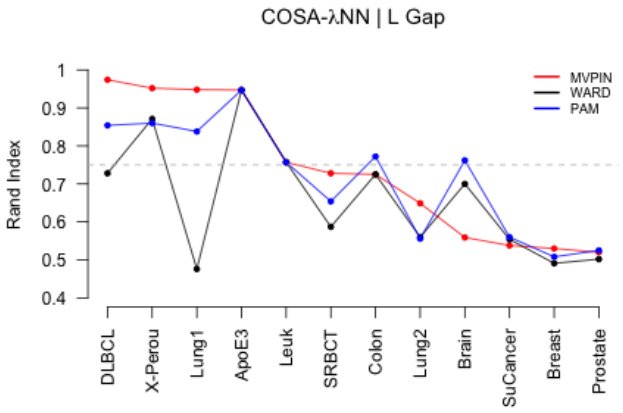

Figure 5.12: The Rand Index computed over the resulting partition from the gap-statistic optimized versions of MVPIN (red), Ward (black) and PAM (blue). The horizontal dashed grey line indicates a Rand Index of 0.75, and the data sets are ordered corresponding to the results of MVPIN, optimized for L.

5.6 Conclusion and Discussion

MVPIN is a clustering algorithm designed for using COSA distances to partition objects in a number of L groups. Compared to competing clustering algorithms such as PAM, hierarchical clustering with average linkage or Ward's method, we find that MVPIN is better at detecting small homogeneous clusters that are similar to each other, because of a similar clustering pattern on an overlapping subset of overlapping attributes. In a comparative study we have shown in a first examination that when the number of clusters is unknown, MVPIN in combination with the Gap statistic shows better results than those obtained by PAM and Ward's method on the (optimized) COSA-λNN distances for the 12 benchmark data sets. Assuming L to be known beforehand, the performances of the clustering algorithms are more similar to one another. Although not shown, similar, but weaker conclusions could be made for MVPIN, PAM and Ward's method on the optimized COSA-KNN distances (Section 5.7.1).

As of yet, we have only demonstrated our empirical experience of a first examination of MVPIN. Although we have provided an advice and a conceptual explanation for finding a good value of K, more research is needed to see under what conditions this advice holds or could break down. For future research it could be of interest to see how MVPIN would perform when applied directly to the neighborhoods that have been used for the computation of the attribute dispersions in either COSA-KNN or COSA-λNN. Moreover, it may be interesting to search for a data-driven lower threshold of the shared number of nearest neighbors from which the algorithmic steps in MVPIN perhaps already sooner could switch to the unrestricted versions of Ward's method.

There are more algorithms that can find a partition for a set of objects while using a distance matrix as input. For future research it could be interesting to compare the performance of MVPIN with the clustering algorithms PAMSIL (Van der Laan et al., 2003) and MiniDisconnect (Lee & Olafsson , 2011). We may think that PAMSIL could be a good competitor since it was designed to find small clusters in biological contexts. However, PAMSIL does not take into account density properties of the small clusters, e.g. a within-cluster shared nearest neighbors index, making it more sensitive to small groups of objects that are simply clusters on sampling fluctuations. MiniDisconnect, however, focuses on minimizing a concept of disconnectivity between clusters that is based on a mutual shared nearest neighbors measure as defined in Chidanda Gowda and Krishna (1978). Moreover, MiniDisconnect assumes L to be unknown, and learns by itself what the optimal value of L needs to be.

We did experiment with the implicit medoid based algorithm 'fast search and find of density peaks' by Laio and Rodriguez (2014). The algorithm did not seem to be capable to deal with the non-metric COSA dissimilarities, and showed a performance that was similar to that of cutting an average linkage dendrogram at a certain height to obtain L groups. Moreover, the 'cut-off' distance tuning parameter in the fast search for density peaks algorithm does not allow for a simple tuning strategy.

5.7 Appendix

5.7.1 MVPIN Results on the COSA-KNN Distances

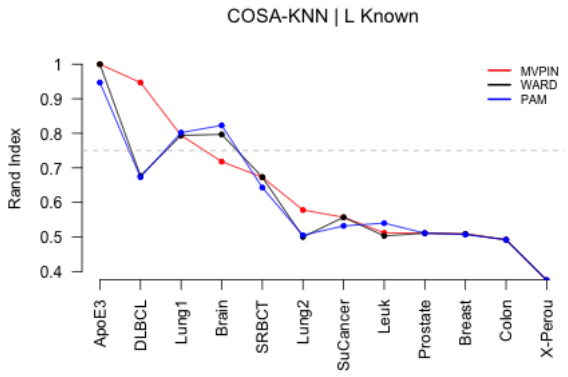

Figure 5.13: The Rand Index computed over the resulting partitions from MVPIN (red), Ward (black) and PAM (blue) applied to the optimized COSA-KNN distances for each data set. The dashed grey line indicates a Rand Index of 0.75.

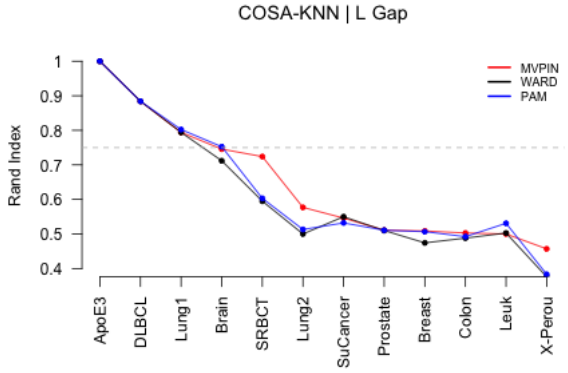

Figure 5.14: The Rand Index computed over the resulting partitions from MVPIN (red), Ward (black) and PAM (blue), with a Gap optimized L, applied to the optimized COSA-KNN distances. The dashed grey line indicates a Rand index of 0.75.

Chapter 6

L-Groups Clustering of Omics Benchmark Data on Regularized Weighted Attributes

As some of the current 'state-of-the-art' L-groups clustering algorithms that also include a form of regularized weighting of the attributes, COSA is largely motivated for omics data. In this chapter we will focus on such L-groups clustering algorithms that were inspired by, or compared themselves with, COSA (Friedman & Meulman, 2004). Note that both COSA-KNN and COSA-λNN, together called COSA-NN, cannot directly be compared with L-groups clustering algorithms. However, in the previous chapter we have seen that with the additional use of MVPIN we can provide a comparison of the COSA-NN algorithms with L-groups clustering algorithms.

The chapter is outlined as follows. Section 6.1 provides a description of L-groups clustering algorithms that implement regularized attribute weighting. We will compare the results of these algorithms with the results of MVPIN on the original COSA-KNN (Chapter 3) and COSA-λNN (Chapter 4) distances in Section 6.2. The comparison will be based on 11 omics benchmark data sets of which 10 were used in earlier studies (e.g. Arias-Castro & Pu, 2017; Jin & Wan, 2017). In addition to a replication of results from these earlier studies, the comparison is extended with new L-groups clustering algorithms that take into account regularized weighting of the attributes. While in this first comparison the number of clusters L was pre-set towards a supposedly 'true' number of classes in the data, we compare the algorithms again in Section 6.3, but this time letting L to be tuned by the Gap statistic of Tibshirani, Walther, and Hastie (2001).

In Section 6.4 a closer look is provided on the two better performing clustering algorithms: Sparse Alternate Sum clustering (SAS; Arias-Castro & Pu, 2017) and

COSA-λNN in combination with MVPIN. We show the two-dimensional multidimensional scaling configurations of the optimized distances of the 'best' benchmark data set for COSA-λNN, as well as the best data set for SAS. By best data set, we mean that the algorithm performed best in recovering the underlying 'true' clustering structure. The results will show that SAS fails to recover the small and homogeneous clusters of COSA-λNN's best data set. For the best data set of SAS, COSA-λNN seems to detect another unknown underlying clustering structure that does not fully represent the truth.

Even though the first part of the chapter is about partitioning the set of objects into L groups, it is not conform the main goal of the COSA framework. The main goal of COSA is to come up with cluster happy distances that may even reveal possibly nested or overlapping grouping structures. These grouping structures do not necessarily exhaust all objects. Already in Chapter 2 we argued that in most graphical representations of distances it is not desired nor necessary to assume a known number of clusters, or to look for a specific partition of the objects. The purpose of COSA is rather explorative: showing the type of clustering structure the data exhibits, from which, after interpretation together with a domain expert, a selection of clusters could be explored depending on the context of the problem.

Instead of framing COSA-λNN and the other algorithms as competitors of each other, we can also use them as each other's complement to validate a detected clustering. In Section, 6.5 we will show how the COSA framework can be used for validation purposes of an assumed known clustering structure. For cluster validation purposes it was proposed in Friedman and Meulman (2004) to work with a strategy based on attribute importance weights. For these attribute importance weights a specific maximum importance parameter needs to be specified beforehand. We will show that when using the optimized cluster-specific attribute weights, instead, the same results can be obtained without the need to specify an extra parameter beforehand.

6.1 L-Groups Clustering on Regularized Weighted Attributes

The original COSA paper by Friedman and Meulman (2004) first describes an algorithm that has the purpose of finding a partition of L groups out of N objects that cluster on subsets of the P attributes. This algorithm is referred to as the 'COSA wrapper algorithm' since it 'uses a conventional iterative clustering method as a primitive' to extend any clustering algorithm on a distance matrix towards clustering of objects on subsets of attributes. Apart from any of the input parameters that are needed in the primitive algorithm, and replacing K, the size of each neighborhood, by L, the number of clusters, this wrapper algorithm has for the rest the same input parameters as the 2004 COSA-KNN algorithm.

Since Friedman and Meulman (2004) used the COSA wrapper algorithm mainly as an introduction for the COSA-KNN algorithm, they never put the COSA wrapper algorithm into action. To our knowledge, only Jin and Wang (2016) and Steinley and Brusco (2008) implemented a specification of the COSA wrapper algorithm for the

purpose of comparison. While in Jin and Wang (2016) no details were given about the parameter settings and about the primitive clustering algorithm being used, Steinley and Brusco (2008) provided all the information needed to be able to replicate (and reproduce) their results. However, the study by Steinley and Brusco (2008) did not apply to the high-dimensional data settings, but to data settings where $P < N$.

The focus of this chapter, however, will be on the COSA nearest neighbors algorithms COSA-KNN and COSA-λNN, together referred to as the COSA-NN algorithms. As we have seen in the previous chapter, the original COSA-KNN has been compared with other L-groups clustering algorithms in the study by Jing, Ng, and Huang (2007) and Witten and Tibshirani (2010). The studies either applied hierarchical clustering or PAM (Kaufman & Rousseeuw, 1987) on default, and not even optimized, COSA distances to find a partition of L groups on the set of N objects. As we have seen in the previous chapter, these clustering algorithms can turn out to be suboptimal.

The two studies by Jing, et al. (2007) and Witten and Tibshirani (2010) together form our starting point for the selection of L-groups clustering algorithms that apply regularized weigthing to the attributes for the following reasons. The main purpose of the study by Jing et al. (2007) was to present their algorithm referred to as Entropy Weighted K-means (EWKM). Nowadays, we see that EWKM has become a default benchmark algorithm in the area that they themselves refer to as subspace clustering (Deng et al. 2016; Chen et al. 2012). Here, subspace clustering is a cluster analysis that has the objective to find L clusters of objects that have their own subsets of regularized attribute weights, i.e. COSA. We see that Enhanced Soft Subspace Clustering (ESSC; Deng et al. 2010) is proposed as an improvement of EWKM in the pattern recognition literature of subspace clustering (Deng et al. 2016).

The purpose of the study by Witten and Tibshirani (2010) was to present the Sparse Clustering framework, abbreviated as SPARCL. SPARCL consists of an algorithm that implements sparse weighting of the attributes for K-means, and an algorithm that implements sparse weighting of the attributes for hierarchical clustering. As EWKM, SPARCL was included in a benchmark study of clustering algorithms for high-dimensional data settings (e.g., Arias-Castro & Pu, 2017; Jin & Wang 2016). From the results of these studies Sparse Alternate Sum clustering (SAS) by Arias-Castro and Pu (2017) is shown to be a significant improvement of the Sparse K-means algorithm (SPARCL). As compared to COSA, EWKM and ESSC, however, both SPARCL and SAS do not allow for clustering of objects on different subsets of attributes.

We will give a brief description of the four recently introduced L-groups clustering algorithms. To smoothen the introduction to these algorithms, the order will be SAS, SPARCL, EWKM, and ESSC, an order based on their technical properties.

Sparse Alternate Sum Clustering (SAS)

The Sparse Alternate Sum (SAS) algorithm for clustering is presented as a framework that can wrapped around any standard procedure for L-groups clustering, e.g. PAM or K-means. As is the case with COSA, SAS is an algorithm built around minimizing

a loss function that is based on the average within-cluster distance. However, the difference with COSA is that all the clusters obtained by SAS are based on the same subset of attributes. Moreover, the attributes are not weighted. Whereas in COSA $a_{kl} = 1$ when the attribute weight for attribute k is non-zero in the attributes subset of cluster l, in SAS we only have attribute activation: either the attribute is active $a_{kl} = 1$, or not $a_{kl} = 0$, regardless of clusters, i.e. $a_{kl'} = a_{kl}$ for two different clusters l and l'. Therefore, we have dropped the second subscript l in the description of the objective criterion of SAS:

$$Q_{\text{SAS}}(\mathbf{C}, \mathbf{a}) = \sum_{l=1}^{L} \sum_{i=1}^{N} c_{il} \sum_{k=1}^{P} a_k \left(\frac{x_{ik} - \mu_{kl}}{s_k} \right)^2 . \tag{6.1}$$

Here μ_{kl} is the within-cluster mean of cluster l on attribute k, and the activation set $\{a_k\}_{k=1}^{P}$, is restricted by the tuning parameter value P_a, the number of active attributes,

$$\sum_{k=1}^{P} a_k = P_a. \tag{6.2}$$

The objective of SAS is to minimize this criterion that represents the normalized within-cluster sum of squares, or, from *Huygens' principle*, the squared Euclidean distances.

The algorithm for SAS that heuristically minimizes $Q_{\text{SAS}}(\mathbf{C}, \mathbf{a})$ (6.1) is open source software and available at `https://github.com/victorpu/SAS_Hill_Climb`. For the algorithm to minimize $Q_{\text{SAS}}(\mathbf{C}, \mathbf{a})$ (6.1), the number of clusters L, and the number of active attributes P_a need to be specified beforehand. Then, a first estimate of the subset of attributes, indicated by the activation vector $\mathbf{a}_{P\times 1}$, will come from performing ordinary K-means (Hartigan & Wong, 1979) on the data set. The estimated subset of attributes are those P_a attributes for which the within-cluster distances are the smallest. Thus, as long as there is (an estimate of) an obtained clustering structure, denoted by $\mathbf{C}$, one can find the minimizing attributes with the activation vector $\mathbf{a}$. Having obtained a solution for $\mathbf{a}$ one can perform K-means clustering, using only the attributes that each have an $a_k = 1$, to obtain an update for $\mathbf{C}$. This iteration process is completed when convergence within a maximum number of iterations is achieved. In summary, the SAS algorithm alternates between the steps:

1: Keeping $\mathbf{a}$ fixed, compute $\mathbf{C}$ using K-means.

2: Keeping $\mathbf{C}$ fixed, compute $\mathbf{a}$, the first P_a attributes for which the average within-cluster distance are smallest.

Arias-Castro and Pu (2017) propose to determine the value of the tuning parameter P_a by the Gap statistic procedure (Tibshirani et al., 2001). This way, SAS is similar to the Forward Selection Algorithm as presented in Fowlkes, Gnanadesikan, and Kettenring (1988), where the selection of the subset of variables decided based on a comparison with Gaussian noise data that does not contain a grouping structure.

Sparse K-means (SPARCL)

SPARCL's K-means algorithm is based on attribute weights that are regularized by controlling the absolute sum (ℓ_1-norm) and the sum of squares (ℓ_2-norm) of the attribute weight values. Still, as is the case with SAS, the attribute weights are the same for all clusters and its default criterion can be expressed as the sum of within-cluster squared Euclidean distances. The default criterion to be minimized by SPARCL is defined as

$$Q_{\text{SPARCL1}}\left(\mathbf{C}, \mathbf{w}\right) = \sum_{l=1}^{L} \sum_{i=1}^{N} c_{il} \sum_{k=1}^{P} w_k \left(\frac{x_{ik} - \mu_{kl}}{s_k} \right)^2, \tag{6.3}$$

over any clustering C_l and any weights $w_1, \ldots, w_P \geq 0$ with constraints,

$$||\mathbf{w}||_1 \leq P_w, \qquad ||\mathbf{w}||_2 \leq 1. \tag{6.4}$$

The ℓ_1 constraint on $\mathbf{w}$ results in sparsity for small values of the tuning parameter P_w, conform the typical behavior of any of the generalizations of the Lasso (Hastie, Tibshirani, & Wainwright, 2015). The tuning parameter P_w can be determined by the Gap statistic. Note the necessity of the ℓ_2 constraint. If this constraint would not have been present, all the weight would be put on only those attributes with the smallest average within-cluster sum of squares.

Apart from pre-setting L and P_w, the attribute weights are initialized to $w_k = 1/\sqrt{P}$ for each attribute k, before the algorithm starts. Holding the attribute weights fixed, a solution for the clustering structure is the output of K-means applied to the data set, where the attributes are multiplied with their corresponding attribute weight. Similarly, holding the solution for $\mathbf{C}$ fixed, there is a closed form solution for the attribute weights. Thus, similar to SAS, the algorithm of SPARCL proposes an alternating minimization strategy until convergence. The open source software of SPARLC is available as an `R` package at the *CRAN* repositories. In this chapter we used version 1.0.3.

Entropy Weighted K-means (EWKM)

Whereas in SAS and SPARCL the minimization of the average within-cluster distance automatically coincides with maximizing the average between-clusters distance, this is not the case when each cluster will have a unique subset of attribute weights. The Entropy Weighted K-means (EWKM) algorithm by Jing et al. (2007) assumes each cluster to have a unique subset of attribute weights, as in COSA. The criterion of EWKM can be rewritten as a special case of the COSA criterion, and is here defined as

$$Q_{EWKM}\left(\mathbf{C}, \mathbf{W}\right) = \sum_{l=1}^{L} \left\{ \sum_{i=1}^{N} c_{il} \sum_{k=1}^{P} w_{kl} \left(\frac{x_{ik} - \mu_{kl}}{s_k} \right)^2 + \lambda \sum_{k=1}^{P} w_{kl} \log(w_{kl}) \right\}, \tag{6.5}$$

where

$$\begin{aligned} &\lambda > 0, \\ &0 < w_{kl} < 1, \\ &\sum_{k=1}^{P} w_{kl} = 1. \end{aligned} \tag{6.6}$$

To apply the EWKM algorithm, a value for L and λ should be chosen. In Jing et al. (2007) it is reported that the value for λ shows robust clustering results for values in-between 0.3 and 7. The attribute weights are initialized to $w_{kl} = 1/P$ and cluster-centers are initialized at random. Then, an alternating minimization strategy of three steps is repeated until convergence:

1: Given fixed cluster centers (μ_{kl}) and attribute weights $\mathbf{W}$, the minimizing partition matrix $\mathbf{C}$ is computed.

2: Given a fixed partitioning matrix $\mathbf{C}$, the minimizing cluster centers are computed.

3: Given fixed cluster centers and a fixed $\mathbf{C}$, the minimizing attribute weights $\mathbf{W}$ are computed.

A downside of EWKM is its initialization sensitivity. Depending on the initialization, it may happen that the EWKM solution results in fewer than L clusters. For example, it can easily happen that a centroid is initialized with a value to which none of the objects will be assigned to. Therefore it is common to restart the algorithm a maximum number of times to assure the result is a set of L clusters. Even when you restart EWKM until L clusters are obtained, such that there are objects assigned to each of the L centroids, the cluster solution can be very different for any other initialization. Thus, EWKM is very prone to local minima. Therefore it is also very common to see that the algorithm is rerun a number of times to obtain an average result for the 'ideal' tuning parameter λ. For example, in Deng et al. (2016) the performance of EWKM is measured based on 10 reruns with 10 restarts each. In this chapter we use the available open source software of EWKM of the `R` package `wskm` version 1.4.28 (Williams et al., 2015) from the CRAN repository.

Enhanced Soft Subspace Clustering (ESSC)

The Enhanced Soft Subspace Clustering (ESSC) algorithm by Deng et al. (2010, 2011) is proposed as an improvement of EWKM. The improvement is that ESSC not only minimizes the within-cluster compactness, but also maximizes the between-cluster separation. Moreover, it replaces the crisp clustering by K-means a fuzzy $C-$means algorithm (Dunn, 1973; Bezdek, 1981). While at first c_{il} could only be equal to 0 or 1, in the ESSC framework it can have any real value in-between (and including) 0 and

1. The criterion that the ESSC algorithm minimizes is

$$
\begin{aligned}
Q_{ESSC}\left(\mathbf{C},\mathbf{W}\right) = & \sum_{l=1}^{L}\sum_{i=1}^{N} c_{il}^{m} \sum_{k=1}^{P} w_{kl} \left(\frac{x_{ik}-\mu_{kl}}{s_k}\right)^2 \\
& + \lambda \sum_{l=1}^{L}\sum_{k=1}^{P} w_{kl}\log(w_{kl}) \\
& - \gamma \sum_{l=1}^{L}\left(\sum_{i=1}^{N} c_{il}^{m}\right) \sum_{k=1}^{P} w_{kl} \left(\frac{\mu_{kl}-\mu_{k0}}{s_k}\right)^2,
\end{aligned} \tag{6.7}
$$

where

$$
\begin{aligned}
& \lambda > 0, \\
& \min_i \left\{ \sum_{k=1}^{P} w_{kl} \left(\frac{x_{ik}-\mu_{kl}}{s_k}\right)^2 \Big/ \sum_{k=1}^{P} w_{kl} \left(\frac{\mu_{kl}-\mu_{k0}}{s_k}\right)^2 \right\} \geq \gamma > 0, \\
& 0 < w_{kl} < 1, \sum_{k=1}^{P} w_{kl} = 1, \\
& 0 \leq c_{il} \leq 1, \sum_{l=1}^{L} c_{il} = 1, \\
& m = \frac{\min\left(N, P-1\right)}{\min\left(N, P-1\right)-2}, \\
& \mu_{k0} = N^{-1} \sum_{i=1}^{N} \mathbf{x}_{ik}.
\end{aligned} \tag{6.8}
$$

Thus, the criterion contains three terms: the weighted within-cluster compactness, the entropy of weights, and the weighted between-cluster separation. The first and second terms are directly inherited from the criterion of EWKM, except that the hard crisp partition matrix now has become fuzzy. To maximize the average between-clusters distance, the extra tuning parameter $\gamma(\gamma > 0)$ is used to control the influence of the between-cluster separation.

ESSC is presented as a leading performer in the pattern recognition area of (soft) subspace clustering. However, it comes with the disadvantage of tuning parameters, λ and γ, for which no strategy or theory is presented. Furthermore, ESSC is even more sensitive than EWKM to the initialization of the attribute weights and local minima. Therefore, it is common to see ESSC being applied with at least the same number of restarts and reruns as EWKM. To our knowledge, there is no free (or commercial) software available. Moreover, following all the information in Deng et al. (2010, 2011, or 2016), we did not manage to replicate any trustworthy results when we created our own versions of `C` code that could be called in `R`. Therefore, we were forced to exclude ESSC in the further part of this chapter.

6.2 Comparison of the Algorithms on Omics Data

Except ESSC, we will compare the described algorithms with MVPIN applied to COSA-KNN and COSA-λNN, using 11 (gen)omics benchmark data sets, of which details are displayed in Table 6.1. The first data set is the metabolomics data that was used Chapter 3. The other 10 data sets are gene expression data sets that originate from oncology studies and were used by Arias-Castro and Pu (2017) and Jin and Wang (2016) to evaluate the SAS clustering, and the Influential Features PCA (IF-PCA) clustering procedure respectively. The ApoE3 data set comes with the `rCOSA` package, all other data sets can still be found online at `www.stat.cmu.edu/~jiashun/Research/software/GenomicsData`.

Except for the ApoE3 data, the data sets have been used to compare various L-groups clustering procedures that are not included in this chapter. We have excluded K-means (Hartigan & Wong, 1979), K-means++ (Arthur and Vassilvitskii, 2007), and hierarchical clustering on Euclidean Distance with average linkage since, since they are not able to weight the attributes, and therefore show no competitive advantage over attribute weighting methods (Jin & Wang, 2016). Other clustering procedures that were able to take into account attribute weighting were also excluded, a.o., Regularized K-means (Sun et al., 2012), and Spectral Gem (Lee et al., 2009). All these procedures were outperformed by either SAS or SPARCL on the benchmark data sets in Arias-Castro and Pu (2017).

There are two regularized attribute weighting L-groups clustering procedures that remain competitive though, these are IF-PCA (Jian & Wang, 2016) and adaptively hierarchically penalized Gaussian-mixture-model (AHPGMM) based clustering (Wang & Zhu, 2008). In Arias-Castro and Pu (2017) IF-PCA clustering procedure performed well on the (gen)omics data sets, and showed different behaviour than SAS or SPARCL. In particular, IF-PCA shows better performance than SAS on three of the data sets, and only on one data set it performed clearly worse. Still, we excluded IF-PCA in this chapter for two reasons. IF-PCA has no free available code that can be used on a free platform (e.g. code is available in Matlab only), and the IF-PCA results (in combination with K-means) are more difficult to interpret with respect to the attribute weights when compared to COSA. The AHPGMM has been excluded from the comparison since its performance and behavior was very similar to that of SAS and SPARCL. Moreover, we stated in Chapter 1 that model based clustering would not be the focus of this monograph.

6.2.1 Description of the Omics Data

The data sets from Table 6.1 have been 'cleaned' and analyzed in previous studies: the ApoE3 mice was cleaned and in Damain et al. (2004); the Brain, Colon, Leukemia, Lymphoma, Prostate and SRBCT gene expression data sets in Dettling (2004) and Dudoit and Fridlyand (2002); the Lung1 gene expression data set comes directly from Gordon et al. (2002); and the remaining Breast, Lung2, SuCancer data sets were analyzed in Yousefi et al. (2010) and grouped into two classes. The Breast2 data set did not have an original true clustering structure, and there were originally 5 groups

in the Lung2 data set, and 11 groups in the SuCancer data set. Since the original labels are not publicly available, we will work with the labels provided by Yousefi et al. (2010). Thus, the data sets we are using have been pre-processed already: non-variant attributes and attributes with too many missing values have been discarded from the data sets, and attributes with only few missing values got imputed with their mean value or nearest neighbor value.

Table 6.1: The metabolomics and the microarray gene expression data sets.

#	Data Name	L	$N(N_1 + \ldots + N_l)$	P	Source
0	ApoE3 Mice	2	38 (18 + 20)	1,550	Damien et al. (2004)
1	Brain	5	42 (10 + 10 + 10 + 4 + 8)	5,597	Pomeroy (02)
2	Breast	2	276 (183 + 93)	22,215	Wang et al. (05)
3	Colon	2	62 (22 + 40)	2,000	Alon et al. (99)
4	Leukemia	2	72 (47 + 25)	3,571	Golub et al. (99)
5	Lung1	2	181 (150 + 31)	12,533	Gordon et al. (02)
6	Lung2	2	203 (139 + 64)	12,600	Bhattacharjee et al. (01)
7	Lymphoma (DLBCL)	3	62 (42 + 9 + 11)	4,026	Alizadeh et al. (00)
8	Prostate	2	102 (50 + 52)	6,033	Singh et al. (02)
9	SRBCT	4	63 (23 + 8 + 12 + 20)	2,308	Kahn (01)
10	SuCancer	2	174 (83 + 91)	7,909	Su et al (01)

6.2.2 Previous Results for the Omics Data Sets

Before we start with the comparison, we first describe relevant results of SAS and SPARCL that have already been reported. In Arias-Castro and Pu (2017) SAS and SPARCL have been compared using the 10 (gen)omic benchmark data sets. From the rejoinder of the discussion about IF-PCA in Jin and Wang (2016), we also have results for SPARCL. In both studies the classification error rate (CE) was used to report how well each algorithm could retrieve the 'true' clustering structure from Table 6.1. Instead of the classification error, we will show the Rand Index (Rand, 1971), which is 1 - CE. Thus a Rand Index (RI) of 1 indicates perfect retrieval of the true clustering structure, and a Rand Index of 0 is an indication of anti-clustering. The relevant results of Arias-Castro and Pu (2017) are displayed in Figure 6.1, and the relevant results of Jin and Wang (2016) in Figure 6.2.

In these figures we use the following abbreviations:

`SASdgs` : SAS where P_a is tuned with the Gap statistic using the default grid search;

`SASgss` : SAS where P_a is tuned with the Gap statistic using golden section search, for a description, see Section 6.2.3.

`SPARCL` : Sparse K-means with all its default settings in the `R` package `sparcl`.

Most necessary settings and details can be obtained from the paper by Arias-Castro and Pu (2017) code to replicate the results of Figure 6.1. The only details missing

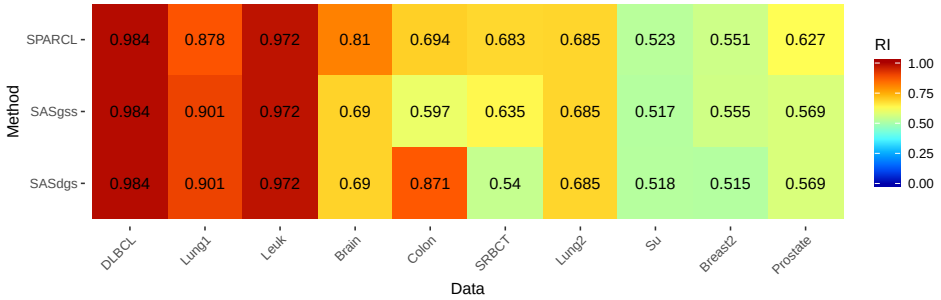

Figure 6.1: Rand Index results for SAS and SPARCL on the 10 data sets, retrieved from Arias-Castro and Pu (2017). Here, the order of the data sets is determined by the results from Section 6.2.4.

were the version number of the `sparcl` package that was used in `R`, and the state of the random number generator to obtain pseudo random cluster centroids.

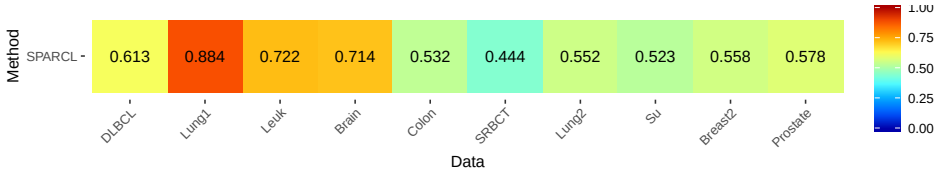

Figure 6.2: Rand index results for SPARCL, retrieved from the rejoinder of Jin and Wang (2016).

The results in Figure 6.1 indicate that both SAS algorithms and `SPARCL` seem to perform 'equally well' on the data sets. However, we see that the performance of `SPARCL` is worse in Jin and Wang (2016), see Figure 6.2. The reasons for this difference remains unknown since we do not know what version of SPARCL is used in either studies, and we do not know what parameter settings have been used for SPARCL in Jin and Wang (2016). Moreover, as we will see later, even though the code for the application of SPARCL in Arias-Castro and Pu (2017) is publicly available, we could not replicate the results of SPARCL (and SAS).

6.2.3 A Scheme for an Extended Replication

We will replicate the analyses shown in Figure 6.1 and Figure 6.2, and extend the design with three algorithms, and one extra data set. The extra data set is the ApoE3 data set (see Table 6.1).The extra algorithms are EWKM (see Section 6.1), and COSA-KNN (Chapter 3) and COSA-λNN (Chapter 4). We stay as close as possible to the reported parameter settings or the advised default parameter settings for each algorithm. A description of the settings, especially regarding the tuning parameters, is presented here:

1. `SASdgs`.The source code of SAS indicates by default a grid of $N_{grd} = 50$ integers for the number of attribute to select, i.e. P_a (see equation 6.2). However, for the benchmark data sets in Arias-Castro and Pu (2017), the grid size is

$$N_{grd} = 10 \times \lfloor P/100 \rfloor.$$

Here, the $\lfloor \cdot \rfloor$ function rounds its input to the closest lowest integer. E.g., for the benchmark data set with the fewest attributes, the ApoE3 data, the grid consists of $10 \times \lfloor 1550/100 \rfloor = 150$ integer values that P_a could take. Each value in the grid is part of the following series:

$$P_a \in \left\{ \left\lfloor P - \frac{(N_{grd}-1)P}{N_{grd}} \right\rfloor, \ldots, \left\lfloor P - \frac{2P}{N_{grd}} \right\rfloor, \left\lfloor P - \frac{P}{N_{grd}} \right\rfloor \right\}.$$

Thus, for the ApoE3 data set the grid of values is $\{31, \ldots, 1488, 1519\}$. The optimal value for P_a is the value that corresponds with the maximal Gap statistic based on $B = 25$ permuted data sets. The K-means algorithm that is used in SAS, is set to a maximum of 20 iterations (for each run) and only 1 random set of start values for the centroids.

2. `SASgss`. Instead of a full grid search strategy, the algorithm `SASgss` implements a golden section search strategy (Brent, 1973) to find the maximal Gap statistic that corresponds to the optimal value for P_a. `SASgss`, is faster than the full search strategy presented in `SASdgs`. In the golden search strategy the lower and upper bound for the optimal value for P_a are set as follows:

$$P - (1-\varphi)(P-1) \leq P_a \leq 1 + (1-\varphi)(P-1),$$

where φ is the golden ratio, i.e. $(1+\sqrt{5})/2$. This procedure, however, assumes a concave criterion for SAS. When there is an overlapping number of relevant attributes for each cluster, and some attributes are not shared by all clusters, it is easily shown with an example that the criterion of SAS can become multimodal as a function of P_a.

Apart from the lower and upper bound for the optimal P_a all other settings are the same as those used in `SASdgs`. Note however, while in the original comparative study by Arias-Castro and Pu (2017) 50 permuted data sets were used for `SASggs`, we will use 25 permuted data sets.

3. `SPARCL`. The tuning parameter in SPARCL is P_w, see equation (6.4), and its optimal value will also be determined based on the maximal Gap statistic over a grid search with the 25 permuted data sets. In SPARCL the default size of the grid is equal to $N_{grd} = 10$ and its values are exponentially distributed starting from 1.2 until $0.9\sqrt{P}$. Expressed in `R` code this would be

```
exp(seq(log(1.2), log(sqrt(P) * 0.9), len = 10))
```

In SPARCL the K-means algorithm is used to obtain clusters. Different from SAS, however, the maximum number of iterations is set equal to 6, and 20 random sets of centroids are used as starting values.

4. `EWKM`. The grid used for the tuning parameter λ in EWKM is

$$\lambda \in \{0.1, 0.2, \ldots, 1.4, 1.5, 2, 3, \ldots, 6, 7, 10, 50, 100, 1000\}.$$

These grid values are the union of the values used for λ in Jing et al. (2006), and those values used in Deng et al. (2016) for EWKM. Since EWKM is highly sensitive for local minima, we re-ran EWKM 10 times, each with a maximum of 10 restarts, for every value of λ.

Instead of selecting the ideal value for λ based on the highest average Rand Index, as was done in Deng et al. (2016), we also choose λ based on the maximum value of the Gap statistic. We select our tuning parameter based on the largest Gap between the average criterion of the 10 re-runs of EWKM on the data, on the one hand, and the average null criterion based on the 25 permuted data sets, on the other hand. Note that we did not perform 10 re-runs of EWKM on each permuted data set, but we did allow a maximum of 10 restarts.

5. `COSA-`K`NN`. For COSA-KNN we use the exact same settings as in Chapter 3. The grid for λ is

$$\lambda \in \{0.01, 0.025, 0.05, .075, 0.10, 0.125, 0.15, 0.20, 0.25, 0.30, 0.40\},$$

and combined with each K in the set

$$K \in \{6 + 2 \times 0,\ 6 + 2 \times 1, 6 + 2 \times 2,\ \ldots, 6 + 2(N_{grdK} - 1)\}.$$

The set of values for K is of size N_{grdK}. For COSA-KNN the value of $N_{grdK} = \lfloor N/4 \rfloor - 4$, where N ($N > 20$) is the number of objects in the data set.

$$\min_{N_{grdK}} (6 + 2(N_{grdK} - 1) - flr(N/2)) > 0. \tag{6.9}$$

As for the previous algorithms, the Gap statistic is used on 25 permuted data sets to choose λ and K.

6. `COSA-`λ`NN` . For COSA-λNN we use the same settings as in Chapter 4. The choice for a grid of λ values, however, was not yet described. As is the case with `SASdgs` and `SPARCL`, the values and the size of the grid for λ are dependent on P. The size of the grid is

$$N_{grd} = 10 \times flr(P/1000),$$

and its logged values are uniformly spaced over the interval $\log(0.05) \leq \log(\lambda) \leq \log(0.6)$. To choose for the optimal λ, the maximum value for the Gap statistic is computed over 25 permuted data sets.

To make a fair comparison, we made sure that all the algorithms were applied to exactly the same 25 permuted data sets for the Gap statistic strategy. Furthermore, all scripts with which the exact same results of this chapter can be reproduced are online at `https://tinyurl.com/MonographCOSA-Chapter6`.

6.2.4 Results of the Extended Replication

To replicate the results from Figure 6.1 and Figure 6.2, we pre-specified L, the number of clusters, to be equal to the supposedly true number of clusters, as shown in Table 6.1. The results of our replication and its extension are given in Figure 6.3. Except for the Colon data set with `SASdgs` (where in the original studies the authors used a finer grid for P_a), and the state of random number generator in `R`, we used the exact same `R` code as was used in Arias-Castro and Pu (2017) for the SAS and SPARCL algorithms.

However, when we compare our results in Figure 6.3 with those of Arias-Castro and Pu (2017) in Figure 6.1, we can see that our results show lower Rand indices (a mean difference of = 0.070), i.e. the SAS and SPARCL algorithms performed worse on most of the data sets. Two clear exceptions for `SASdgs` are the Brain and SRBCT data, with a change from 0.690 to 0.787, and 0.54 to 0.65, respectively. The general trend, however, of the replicated results for SAS and SPARCL remain the same. Except for the Colon data, the data sets from which the true clustering structure could have been retrieved with Rand Index values > 0.8 were the Brain, Leukemia, Lung1, and DLBCL data sets. When we compare our Rand Index values with those in Figure 6.2 from Jin and Wang (2016), we see that our results for `SPARCL` are better (mean difference = 0.079).

We extended the comparative study with the ApoE3 benchmark data set, and additional 'state-of-the-art' algorithms. Regarding the values for the Rand Indices of at least 0.7 as satisfactory, the ApoE3 data set does not seem to get clustered well by `SPARCL`, or by `EWKM`. However, each algorithm did show to be competitive on at least one data set, i.e. each algorithm belongs in the top 3 of at least one data set with a Rand Index ≥ 0.7. Moreover, except for `EWKM`, each algorithm is the winner on at least one of the data sets.

Overall, the algorithms `COSA-`λ`NN` and `SASdgs` together show the best performance. `COSA-`λ`NN` obtained the highest average Rand Index ($\overline{RI} = 0.716$), with `SASdgs` being a very close runner-up ($\overline{RI} = 0.699$). It is noteworthy that `EWKM` only obtained 'considerable' results on the DLBCL ad SRBCT data. In general, `COSA-`λ`NN` shows a better performance than `COSA-`K`NN` (Mean difference = 0.052). Nevertheless, only `COSA-`K`NN` perfectly retrieves the clustering structure from the ApoE3 data.

For the benchmark data, the best clustering results were obtained on the ApoE3 Brain, Colon, Leuk, Lung1, DLBCL and SRBCT data sets with each at least one algorithm having a Rand index performance > 0.7. We refer to these data sets as the 'clusterable' data sets. On the Breast2, Lung2, Prostate, and SuCancer data, none of the algorithms seemed to be able to retrieve any of the clustering structures, having Rand indices of $RI < 0.7$.

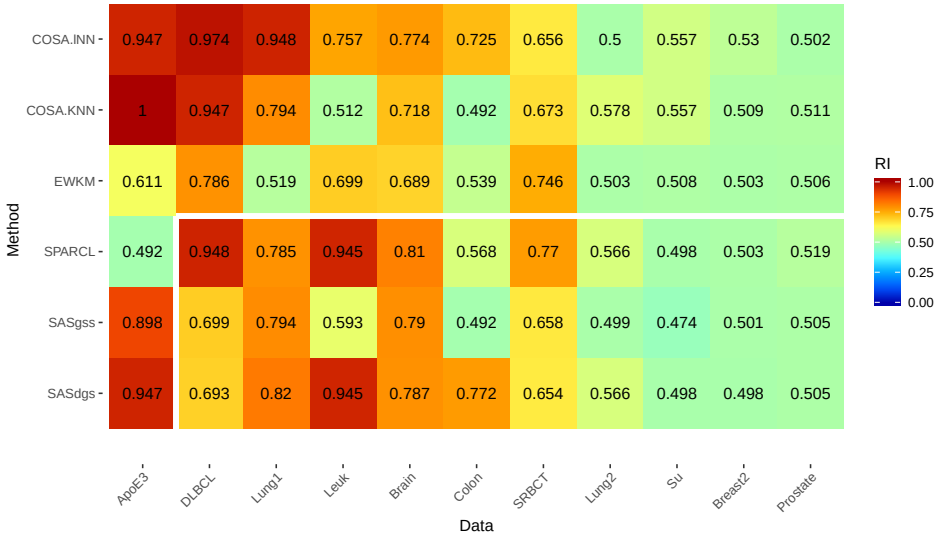

Figure 6.3: The separated lower right rectangular shows results that should be compared with those from the by Arias-Castro and Pu (2017) given in Figure 6.1. We have extended the study with the ApoE3 data (the most left column), and the `COSA-λNN`*,* `COSA-KNN`*,* `EWKM` *algorithms.*

6.3 L is rarely known

A common limitation of the results is that L, the number of clusters, was just pre-set beforehand and thus assumed known. This limitation did not receive any attention in the previous studies of the clustering algorithms. It is more informative to compare the algorithms under the assumption that L is unknown. To do so, we need a strategy to determine the number of clusters obtained by each algorithm. There are many cluster validation measures that can guide the researcher to determine the number of clusters; the silhouette width (Rousseeuw, 1987) is just one example among many. For overviews on this ongoing topic of cluster validation research, see Dudoit and Fridlyand (2002), Tibshirani and Walther (2005), Arbelaitz et al. (2013), Lee and Olafsson (2013), and Hancer et al. (2017). However, apart from Tibshirani and Guenther (2005), these overviews do not pay attention to either regularized attribute weighting clustering algorithms, or high-dimensional data settings. In this section we will fill this gap.

To select L for each algorithm we will apply the Gap statistic to the criterion as a function of L for each of the L groups clustering algorithms. For SAS, SPARCL, and EWKM this is a natural choice since the Gap statistic performs well on algorithms that have objective functions that have the tendency to find convex-like shaped clusters in the data (Lee & Olaffson, 2013). However, it may get outperformed for specific high-dimensional data settings by the prediction strength method in Tibshirani and Guenther (2005). This latter 'better' approach, however, introduces a new problem of setting a certain threshold for prediction strength, and it comes with extra com-

putational costs due to a to be specified number of repetitions of the cross-validation procedure. Since we perform MVPIN on the COSA-λNN and COSA-KNN distances, we apply the Gap statistic procedure to the objective of MVPIN (see Chapter 5, equation 5.17).

6.3.1 Results when assuming L unknown

We applied the Gap statistic for all the algorithms on a grid $L \in \{2, 3, \ldots, 10\}$, while using the same 25 permuted data sets we used for selecting the optimal tuning parameters for the attribute weighting. Letting the number of clusters L to be estimated by the Gap statistic resulted in the Rand indices presented in Figure 6.4.

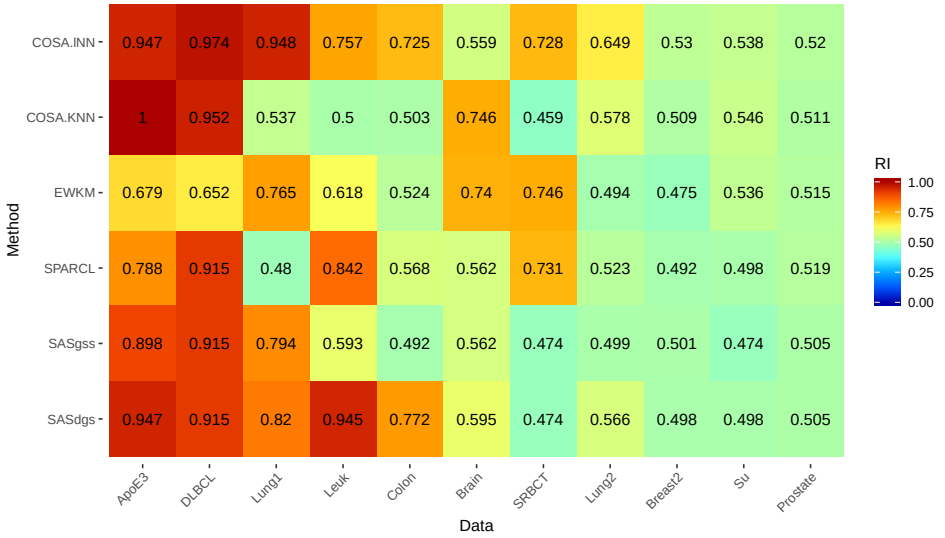

Figure 6.4: A comparison of the algorithms assuming L is unknown; for each algorithm the number of clusters is optimized by the Gap statistic.

Overall, all Rand Indices became a little lower (mean difference = 0.017) when compared to the situation in Figure 6.3, where L was pre-specified equal to the number of clusters. The largest changes in the Rand Index were observed for those data sets with more than two clusters. From the current results, `COSA-λNN` and `SASdgs` remain the best performers for the data sets. While for `SASdgs` the Rand Indices became slightly lower on average (mean = 0.685, and was 0.699), the Rand Indices of `COSA-λNN` remained on average about the same (mean = 0.716).

With the Gap statistic procedure applied to the value of the criterion of the two SAS algorithms, only two clusters were consistently selected (see Table 6.2). While for the DLCBL data a selection of a lower number of clusters results in a higher value of the Rand Index (from 0.693 to 0.915), for the Brain and SRBCT data sets, these are lower Rand Index values (0.787 to 0.595, from 0.654 to 0.474, respectively). The

performance of `SASdgs`, as well as `COSA-`λ`NN`, on the DLBCL data will be discussed in further detail in Section 6.4.

Table 6.2: The estimated number of clusters based on the Gap statistic $\widehat{L}$, and the supposedly true number of clusters L in brackets behind the name of each data set. The estimate $\widehat{L}$ in red indicates a better Rand Index value than obtained with L, in blue the estimate for $\widehat{L}$ indicates a lower Rand Index value than than obtained with L.

Data	L	SASdgs	SASggs	SPARCL	EWKM	COSA-KNN	COSA-λNN
ApoE3	2	2	2	**3**	**5**	2	2
Brain	5	**2**	**2**	**2**	**6**	**6**	**3**
Breast	2	2	2	**3**	**6**	2	2
Colon	2	2	2	2	**6**	5	2
Leukemia	2	2	2	**3**	**6**	**3**	2
Lung1	2	2	2	**6**	**5**	6	**3**
Lung2	2	2	2	**3**	**5**	2	**3**
DLBCL	3	**2**	**2**	**2**	**6**	**4**	3
Prostate	2	2	2	2	**4**	2	**4**
SRBCT	4	**2**	**2**	**3**	6	**2**	**6**
SuCancer	2	2	2	2	**6**	**3**	**3**

For `SPARCL` the Gap statistic procedure 'worsened' the results (mean difference = -0.044) on the clusterable data sets. Large decreases in the Rand Indices occurred for the Brain (0.81 to 0.562), Leukemia (0.945 to 0.842), the Lung1 (0.785 to 0.48), and DLBCL (0.945 to 0.915) data sets. However, the Rand Indices improved for the ApoE3 data (0.492 to 0.788). For `EWKM` the application of the Gap statistic procedure resulted in slightly better Rand Index values (mean difference = 0.012). Compared to the results in Figure 6.3, the results are worse for `COSA-`K`NN` (mean difference = -0.041).

6.3.2 Discussion of the Results So Far

Although COSA-KNN and COSA-λNN are not designed to find L groups automatically, in combination with MVPIN they can be used for this purpose. In general, COSA-λNN performs better than COSA-KNN when used in combination with MVPIN. Moreover, when assuming L is unknown, the application of COSA-λNN with MVPIN gives the arguably best clustering performance. In general, both COSA-λNN and SAS with the default grid search, provided the best clustering results on the benchmark data. However, it remains difficult to claim that both COSA-λNN and SAS outperform the other algorithms. For every data set, there is an alternative with a comparable performance, if not better.

When we optimize the pre-set number L with the Gap statistic procedure, the performances of the algorithms on the data sets change. Whereas estimating the 'wrong' number of clusters led on average towards a better recovery of the clustering structure for EWKM, COSA-KNN, and COSA-λNN, it did not lead to a better average performance for SPARCL or the SAS algorithms.

The replicated results for SAS and SPARCL are different from, and worse than,

those in the original study (Arias-Castro & Pu, 2017). A first explanation for the lower Rand Index values of SAS (and SPARCL) could be that we have hit a bad starting state of the (pseudo) random generator. The 'seed' we used for our random number generator represented the date related to the start of the comparative study, i.e. `set.seed(20180830)`. Re-running the algorithm on the DLBCL data set did not result in better performances (results not shown here). Note, however, the results of SPARCL are better than those presented in Jin and Wang (2016).

We conjecture that the differences in the results may be due to the volatile sensitivity to random starts of SAS and SPARCL algorithms. The regularized attribute weighting K-means (or C-means) type algorithms that perform attribute weighting for high-dimensional data settings are more prone to local minima than the original K-means algorithm. In the subspace clustering literature, this problem is even larger since every cluster has its own subset, therefore EWKM automatically comes with the 'advice' to re-run the algorithm a multiple (of ten) times on each data set.

Consistent with the original studies by Arias-Castro and Pu (2017) and Jin and Wang (2016), is that none of the algorithms recovers the clustering structure of the Breast2, Lung2, Prostate and SuCancer data sets. Noteworthy is that even for supervised learning algorithms, where the group-label was used as an outcome variable, these datasets were difficult (Dettling, 2004; Yousefi et al., 2010). Another reason may be that the division into two groups of the Breast2, Lung2, and SuCancer by Yousefi et al. (2010), might not have been a good or the only representation of the original grouping structure.

A first strong point of this extension of the original study is that we compared SAS and SPARCL with the 'subspace clustering' algorithm EWKM. Moreover, this comparative study also gives a better understanding of the average performance of EWKM with a selection strategy for the value of tuning parameter, which was independent of the clustering structure. In Deng et al. (2013; 2016) the tuning parameters were 'ideally' adjusted towards the highest average value of the Rand Index with respect to the clustering structure. This strategy cannot be implemented in practice, since the cluster-labels are not known. In our study, EWKM does not belong to the best performers when comparing their average Rand Index values with the (single) Rand Index values of the other algorithms.

The second strong point of this comparative study is that we have compared the algorithms while not considering L, the number of clusters, to be known. However, this strong point coincides with a limitation: the ongoing debate on how to decide for L clusters (Hancer et al., 2017). When we use the Gap statistic to select the number of clusters in combination with SAS, we always seem to determine two clusters. When the attribute weights are the same for all clusters, it seems that the more strict the attribute weighting, the more the Gap statistic procedure points towards a smaller estimated number of clusters. This is an hypothesis that seems to be consistent with findings in Kou (2014), and may be resolved by applying the robust-GUD statistic to choose the number of clusters.

6.4 A Closer Look at SAS and COSA-λNN

To obtain a better understanding about the performance of SAS with the default grid search strategy and COSA-λNN in combination with MVPIN, we will take a closer look at the DLCBL (Alizadeh et al., 2000) data set that gave the 'best' results for COSA-λNN, and the Leukemia data set (Golub et al., 1999) that gave the 'best' results for SAS. In this section we will see that the graphical display of the MDS configuration of the distances can contribute largely to the understanding of the detected clustering structure by the algorithms, compared to having only group labels and attribute weights from the L clustering algorithm.

6.4.1 DLBCL Data: Issue of Equal Variance Clusters

Instead of L-groups clustering algorithms, we see that hierarchical clustering dendrograms or multidimensional scaling (MDS) of distances is widely used on gene expression profiles to discover subclasses of disease, e.g. see all studies of the benchmark data sets. Tibshirani et. al (2005) argue that "clustering DNA microarray data is considered a hard problem not only because of the large dimension of the data, but also because there may be no underlying 'true' number of clusters; the expression levels of some genes may not vary consistently with other genes, and clusters may have varying width". MDS and the use of hierarchical clustering lend themselves for a graphical display to obtain a notion of the differences between clusters and also the spectrum of the number of clusters that could have been selected.

The above description is applicable to the Diffuse large B-cell lymphoma (DLBCL) data. In Alizadeh et al. (2000) the group of DLBCL (42 objects) could be distinguished from the chronic lymphocytic leukemia (CLL; 12 objects) and follicular lymphoma (FL; 9 objects) groups. The latter two groups again, could be distinguished from each other via a certain gene-expression profile. Compared to the truth where $L = 3$, Figure 6.5 shows that for SAS the best Rand Index on the DLCBL data is obtained for $L = 2$, while we show in Figure 6.6 COSA-λNN with MVPIN obtained the highest value of the Rand Index for $L = 4$.

Conform most K-means clustering algorithms, SAS has the tendency to select an attribute set that coincides with a partitioning of the data where the clusters are close to equal variance clusters. This tendency complicates the recovery of nested clustering structure when comparing different values for L. For $L = 2$, SAS recognizes the large remainder group of diffuse large B-cell lymphoma, and merges the two smaller groups. For $L = 3$, SAS seems to divide the large remainder group of diffuse large B-cell lymphoma into two groups, instead of recognizing the CLL and FL groups (see Figure 6.5). Moreover, since the two latter groups are small, merging the two groups together does not have such a large impact on the Rand Index, which explains the high Rand Index value of 0.91.

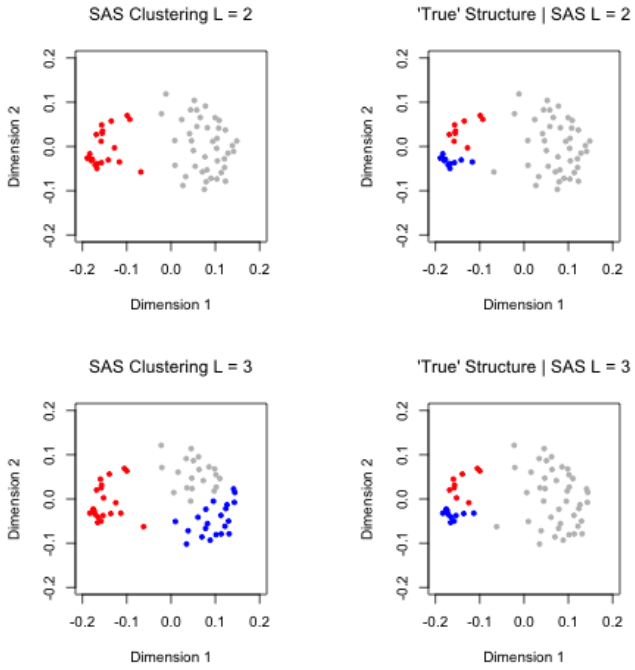

Figure 6.5: SAS selected 634 (out of $P = 4026$) attributes with the default grid search strategy for $L = 3$, and 694 attributes for $L = 2$.

Since for every different value of L exactly the same COSA-λNN distances are the input for MVPIN, it is easier to recover a nested clustering structure, if present in the data. Moreover, as was seen in the previous chapter, COSA-λNN in combination with MVPIN is less enforced to search for clusters of equal variance, e.g., for the DLBCL data example we can see in Figure 6.6 that with $L = 4$, there is one tumor cell selected to be a cluster on its own (in purple), which is consistent with the original study in Alizadeh et al. (2000).

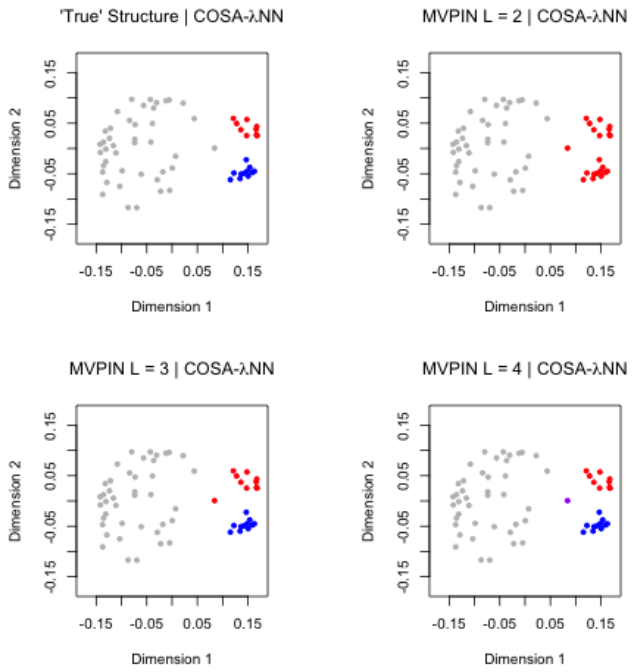

Figure 6.6: The MDS configuration of the optimized COSA-NN results where the value for λ was 0.213 (corresponding to 857 non-zero attribute weights in each cluster). In the left upper panel the true clustering is shown in the MDS configuration, the right upper panel shows the MVPIN solution on the COSA distances for $L = 2$, the left bottom panel for $L = 3$, and the right bottom panel for $L = 4$.

6.4.2 Leukemia Data with 3,571 or 7,129 genes?

The largest difference between SAS and COSA-λNN on the clusterable data sets is obtained on the Leukemia data set. The Leukemia data set consists of 47 patients with acute lymphoblastic leukemia (ALL) and 25 patients with acute myeloid leukemia (AML). When regarding the Rand Index as the indicator of performance, SAS seems to be a clear winner on the Leukemia data as compared to COSA with MVPIN (see the two upper panels of Figure 6.7). However, when we take a closer look at the original study of the Leukemia data set in Golub et al. (1999), we find that in the original study the raw data consisted of samples that were 'measured' on the expression of more than 3,571 genes (Affymetrix probes). The selection process that was used to downsize the data towards 3,571 genes can be found in Dudoit and Fridlyand (2002).

The complete source of all the gene expression consists of 7,129 genes is available in the `R` package `golubEsets` at the Bioconductor repository (Golub, 2018). When this data set is used, suddenly we see that COSA-λNN in combination with MVPIN does better than optimized SAS, as can be seen from the lower two panels of 6.7.

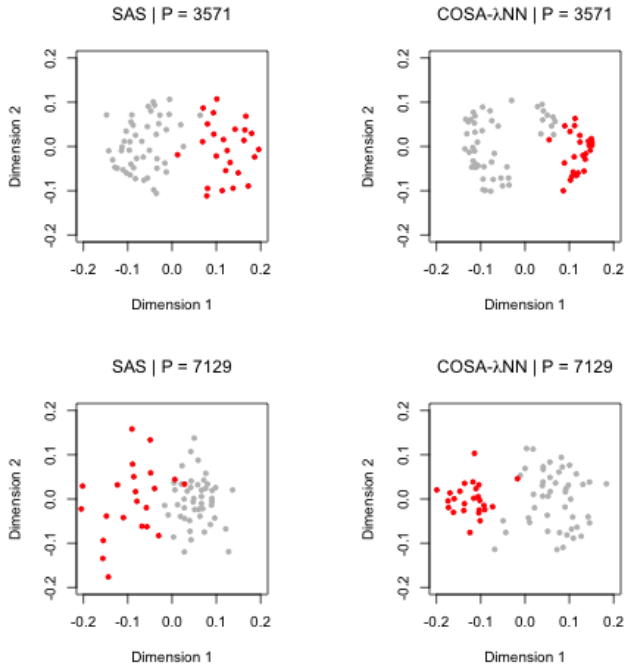

Figure 6.7: The MDS configuration of the SAS results (left panel) and COSA-NN results (right panel) on the Leukemia data sets. The red points represent the 25 AML patients, and the grey points represent the 42 ALL patients.

Note that in Deng et al. (2011) this larger data set also has been used. Their reported (highest) average Rand Index values are 0.6152 and 0.625, for EWKM and ESSC, respectively. On this larger data set SAS obtained a Rand Index value of 0.893 and COSA-λNN in combination with MVPIN resulted in the highest Rand Index value of 0.972.

6.5 Validating Clusters in the COSA Framework

For seven out of the eleven benchmark data sets, we managed to detect a similar grouping structure with Rand Index values higher than 0.7, on at least one of the clustering algorithms. Thus, one can say that we implicitly validated a grouping structure that was assumed to be there in the data sets. In this section we will show that the COSA framework can also be used to test for validation of an assumed known or found clustering structure ($\mathcal{C}$) in the data. We will first explain the technical details of our validation procedure, with the SuCancer and Breast2 data as examples, where the clustering structure was not recovered by any of the algorithms. Finally, our validation results of all the benchmark data sets will be shown.

6.5.1 Optimized attribute weights for validation

We have seen in the previous chapters that when the clustering structure is assumed to be known (each c_{il} is known), there is a closed form expression for the attribute weights, i.e.

$$\widehat{w}_{kl} = \frac{u_{kl} \exp\left(-\frac{S_{kl}}{\lambda}\right)}{\sum_{k'=1}^{P} u_{k'l} \exp\left(-\frac{S_{k'l}}{\lambda}\right)}, \tag{6.10}$$

where

$$S_{kl} = \frac{\sum_{i=1}^{N} c_{il} d_{ij_i^* k}}{N_l}. \tag{6.11}$$

Equation (4.13) is the minimizing solution for each attribute weight in the COSA criterion,

$$Q(\mathbf{W}, \lambda) = \sum_{l=1}^{L} \sum_{k=1}^{P} w_{kl} S_{kl} + \lambda \sum_{l=1}^{L} \sum_{k=1}^{P} w_{kl} \log\left(\frac{w_{kl}}{u_{kl}}\right). \tag{6.12}$$

Here, the left term consists of the sum of the weighted attribute dispersions and the right term consists of the regularized sum of the Kullback-Leibler divergences. While at first both terms increase for higher values of λ, we see that the regularized sum of the L Kullback-Leibler divergences is the first one to achieve its maximum, and eventually its role in the criterion fades out for higher values of λ. At this certain moment, any increase of the COSA criterion is solely ascribed to an increase of the L sums of weighted attribute dispersions, as can be seen in Figure 6.8.

Given a grouping structure, and its corresponding set of attribute dispersions $\{S_{kl}\}$, the λ parameter regularizes and scales the sum of the divergence between the attribute weights and the pre-set attribute weights (each u_{kl}). Suppose we set each $u_{kl} = 1/P$, and find the value for λ at which the regularized sum of divergences achieve its maximum, i.e.

$$\widehat{\lambda} = \underset{\lambda}{\operatorname{argmax}} \sum_{l=1}^{L} \left\{ \lambda \sum_{k=1}^{P} \widehat{w}_{kl} \log\left(\frac{\widehat{w}_{kl}}{u_{kl}}\right) \right\}. \tag{6.13}$$

Then, $\widehat{\lambda}$ is the value for λ for which the regularization was the 'weakest'. Here, the departure from uniform attribute weights (given $u_{kl} = 1/P$), is the highest, which we interpreted as the strongest possible 'selective strength'.

Thus, attribute weights that are computed based on $\lambda = \widehat{\lambda}$, result in the least restricted regularized sum of L Kullback-Leibler divergences between $\mathbf{w}_l$ and $\mathbf{u}_l$. Therefore we see the attributes weights based on $\widehat{\lambda}$ as an optimum in the weighting of the attributes, and as a certain dual solution for the attribute weights for which strong duality holds (Boyd & Vandenberghe, 2004; also, see the Appendix of Chapter 4).

Since the COSA criterion can be reformulated as L independent sums, it follows that we can also find a separate optimal regularizing values for each cluster l, instead of one value for λ. Let

$$\Lambda = \{\lambda_1, \ldots, \lambda_l, \ldots, \lambda_L\}, \tag{6.14}$$

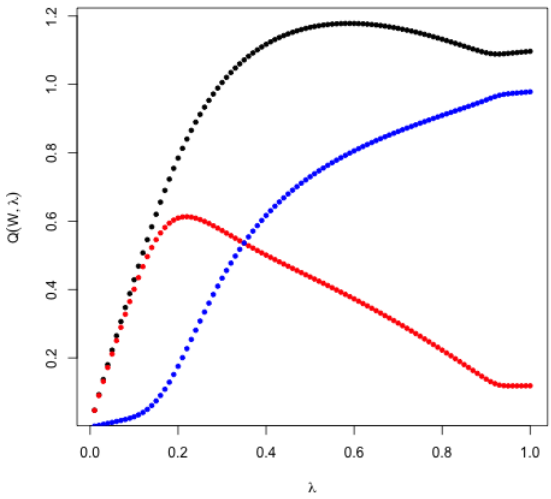

Figure 6.8: The decompostion of the criterion for COSA-λNN applied to the SuCancer data set. The red line is the sum of the L regularized Kullback-Leibler Divergence terms (right term in equation (4.11)*), the blue line is the sum of the L weighted attribute dispersions term (left term in equation* (4.11)*), and in black the COSA criterion shown.*

Then, the criterion becomes

$$Q(\mathbf{W}, \Lambda) = \sum_{l=1}^{L} \sum_{k=1}^{P} \widehat{w}_{kl} S_{kl} + \lambda_l \sum_{l=1}^{L} \sum_{k=1}^{P} \widehat{w}_{kl} \log \left(\frac{\widehat{w}_{kl}}{u_{kl}} \right). \tag{6.15}$$

When we maximize the regularized Kullback-Leibler divergence term over λ_l for each l, than we can compute cluster-specific attribute weights based on $\widehat{\lambda}_l$, where

$$\widehat{\lambda}_l = \underset{\lambda_l}{\operatorname{argmax}} \left\{ \lambda_l \sum_{k=1}^{P} \widehat{w}_{kl} \log \left(\frac{\widehat{w}_{kl}}{u_{kl}} \right) \right\}. \tag{6.16}$$

With the cluster-specific optimal attribute weights,

$$\widehat{w}^*_{kl} = \frac{u_{kl} \exp\left(-\frac{S_{kl}}{\widehat{\lambda}_l} \right)}{\sum_{k'=1}^{P} u_{k'l} \exp\left(-\frac{S_{k'l}}{\widehat{\lambda}_l} \right)}, \tag{6.17}$$

we can compute COSA distances, and visualize them in a dendrogram or MDS configuration. Moreover, we can use the cluster maximized regularized Kullback-Leiber divergences as a criterion

$$Q^{\lambda KL_l} = \max_{\lambda_l} \left\{ \lambda_l \sum_{k=1}^{P} \widehat{w}^*_{kl} \log \left(\frac{\widehat{w}^*_{kl}}{u_{kl}} \right) \right\}, \tag{6.18}$$

on which we can validate a clustering structure. A cluster could be referred to as validated when its criterion, $Q^{\lambda KL_l}$, is lower than a certain proportion, denoted by α, of a number of B criteria, of which each criterion represents a random cluster of objects that has exactly the same size as cluster l. A similar omnibus validation 'test' can be applied to the whole clustering structure by computing

$$p_{vld} = \frac{1}{B}\sum_{b=1}^{B} I\left\{\sum_{l=1}^{L} Q^{\lambda KL_l} \leq \sum_{l=1}^{L} Q_b^{\lambda KL_l^\circ}\right\}, \tag{6.19}$$

where I is the indicator function that returns 1 when the input condition holds true, and where each $Q_b^{\lambda KL_l^\circ}$ is the maximized regularized Kullback-Leibler divergence for a randomly drawn cluster from the data that has similar size as cluster l.

Validating the Two Clusters in Su Cancer data

Perhaps one of the reasons why the two clusters in SuCancer could not be detected is that the 174 samples originally represent 11 different types tumor cells, but their labels were merged into two clusters in Yousefi et al., 2010. The first group consists of the tumor cells from the Bladder/ureter (8 samples), Breast (26 samples), Colorectal (23 samples), and Prostate (26 samples). The second group consists of cells from the Ovary (27 samples), Gastroesophagus (12 samples), Kidney (11 samples), Liver (7 samples), Pancreas (6 samples), Lung Adeno (6 samples) and Lung Squamous (14 samples).

When we compute the two cluster-specific sets of optimized attribute weights for the purpose of validating the two large clusters, then an MDS configuration of the COSA distances that are based on the optimized attribute weights is able to show two clearly separable clusters in Figure 6.9. On the right panel of the same figure the empirical estimate of the sampling distribution function is shown for $\{Q_b^{\lambda KL_l^\circ}\}_{b=1}^{B}$ with $B = 200$. It clearly shows that the maximized sum of the two Kullback-Leibler divergence terms, equation (6.18), is much lower than each of the 200 randomly maximized $Q_b^{\lambda KL_l^\circ}$. The fusion that was performed in Yousefi et al. (2010) may have created two groups of which the heterogeneity within each group is larger than 200 randomly chosen similar size clusters.

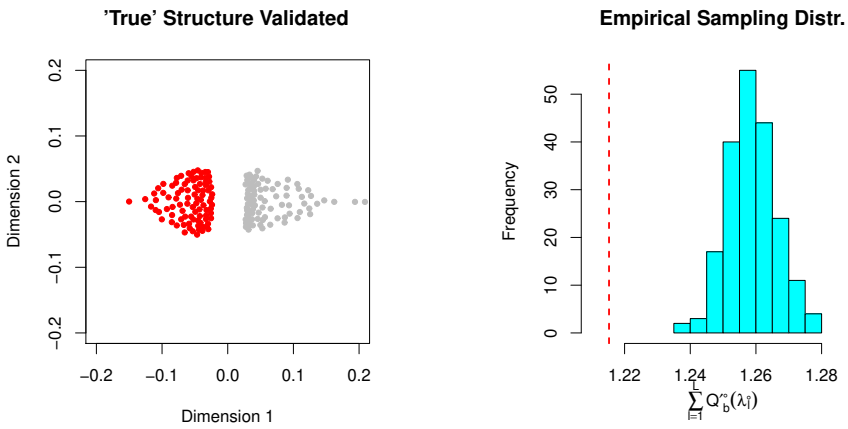

Figure 6.9: The left panel is the MDS configuration shown for the COSA distances based on the optimized attribute weights for the validation of the two clusters in the SuCancer data. The two groups are colored in red and grey. The right panel shows a histogram of an empirical estimate of the sampling distribution of $\sum_{l=1}^{L} Q_b^{\lambda KL_l^\circ}$ *for* $B = 200$ *samples, the vertical red line shows the location of* $\sum_{l=1}^{L} Q^{\lambda KL_l}$ *for the SuCancer data*

Validating the Two Clusters in the Breast Cancer Data

For the clustering structure posed on the Breast cancer data set we obtained low Rand Index values. Similar low results were found in other studies (Arias-Castro & Pu, 2017; Jin & Wang, 2016; Youssefi et al. 2010). A reason may be that in the original study of the Breast Cancer Data by Wang et al. (2005), there were no cluster labels applied to the Lymph-node-negative Breast Cancer Tumor samples on which the 22,215 RNA transcripts were collected. In Yousefi et al. (2010) the data were split into two groups, one group of 93 tumor cells where a relapse of distant metastases occured within 5 years (Gr 1), and a group of 183 (Gr 2) tumor cells where metastases occured after 5 years.

Even though the Breast Cancer data seems to have a 'clusterable' structure, as a whole, it can be validated with $p_{vld} = 0$ for $B = 200$ random similar sized partitions. However, when we take a look at each of the proportions, for each cluster separately, we see $p_{vld} = 1$ for the 183 metastates tumor cells, and $p_{vld} = 0$ for the group of 93 tumor cells. Apparently, all $B = 200$ randomly selected groups (each of 183 objects), have a stronger optimized attribute weighting than the original group of 183 tumor cells, see Figure 6.10.

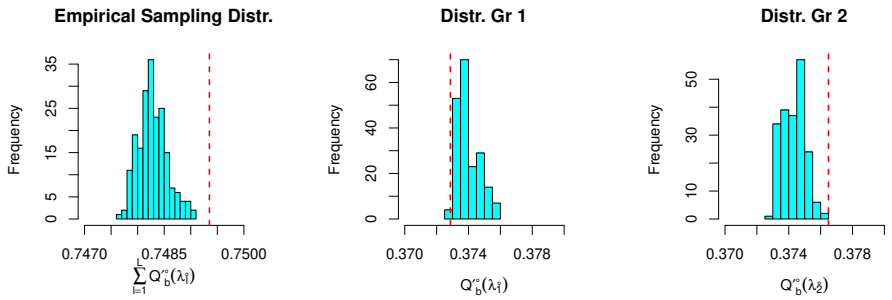

Figure 6.10: The left panel shows a histogram of an estimate of the sampling distribution of $\sum_{l=1}^{L} Q_b^{\lambda KL_l^\circ}$ *for* $B = 200$ *samples, the location for the posed grouping structure inthe Breast data. The middle and the right panels show the estimated sampling distributions for* $B = 200$ *samples of each cluster separately.*

Attribute Importance Weights

Another extra validation step for a cluster that was already shown in the introduction of this monograph (and in Friedman & Meulman, 2004) was to verify whether a relevant subset of attributes can be identified based on an attribute importance measure. The attribute importance was defined as

$$I_{kl} = (S_{kl} + \epsilon)^{-1}. \tag{6.20}$$

In the source code of the COSA implementations, ϵ was set by default to 0.05, to give a limit of 20 on the maximum attribute importance value.

When we look at the first 100 attributes (out of the 22,215 attributes) that are ordered on importance, we see in Figure 6.11 that hardly any of the attributes have an importance that is higher than any of the corresponding ordered importance weights for similar sized random (noise) groups in the data.

Instead of using the attribute importance weights, we propose to use the optimized attribute weights for validation. Now that there is a strategy to optimize λ given a known clustering structure, a similar visualization can be made for the optimized attribute attribute weights for validation, see Figure 6.12. By setting the ordered optimized attribute weights against their counterparts of only 'noise' clusters, we see that the optimized attributes of the second cluster are higher than those expected on average.

While the order of the attributes is exactly the same for optimized attribute weights, as the attribute importance, there is a subtle difference. The x^{th} highest attribute importance only depends on the x^{th} ordered attribute, and is set against the importance of an attribute dispersion of the same order, but from a noise cluster. The optimized attribute weight also takes into account the values of the attribute dispersions of all other attributes within that same 'noise' cluster. Moreover, for the

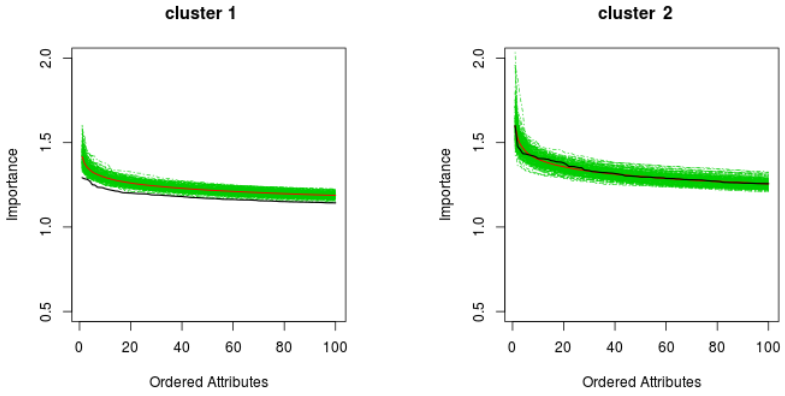

Figure 6.11: The first 100 ordered attribute importance weights for the two posed clusters in the Breast data. The results for cluster 1 are shown in the left panel and the results for cluster 2 in the right panel. The black line represents the 100 ordered attribute importances of the true cluster, each of the $B = 50$ green lines represents 100 ordered attribute importances obtained for a similar size random noise cluster, and the red line is the average of the green lines.

optimized attribute weights no pre-set tuning of ϵ is required to limit the maximum attribute importance.

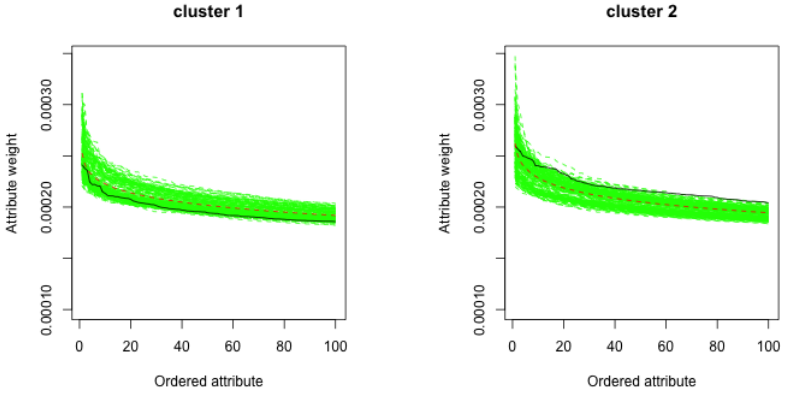

Figure 6.12: The first 100 ordered attribute importance weights for each true clustering structure in the Breast data. The results for Cluster 1 are shown in the left panel and the results for cluster 2 in the right panel. The black line represent the 100 ordered attribute importances of the true cluster, each of the $B = 50$ green lines represents 100 ordered attribute importances of a similar size random noise cluster, and the red line is the average of the green lines.

6.5.2 Validation Results for the Benchmark Data Sets

For all datasets we computed the validation p-values (p_{vld}) for each true clustering structure, and for each cluster separately. The results are shown in Table 6.3. Except for the Brain, Breast2 and SuCancerdata sets, all remaning data sets showed a grouping structure where the maximum regularized sum of Kullback-Leibler divergences between the optimized attribute weights and equal attribute weights was higher than for any of the $B = 200$ noise data sets.

Table 6.3: The validation results of the true structure of the datasets for $B = 200$. Of the true group columns in this table Gr1 represents the largest cluster, Gr2 represents the second largest cluster, etc. The proportions that are larger than 0.05, are colored blue.

Data / p_{vld}	Omnibus	Gr1	Gr2	Gr3	Gr4	Gr5
ApoE3	0.000	0.000	0.000			
Brain	0.090	0.580	0.200	0.070	0.285	0.430
Breast	0.005	0.995	0.000			
Colon	0.000	0.215	0.000			
Leukemia	0.000	0.000	0.005			
Lung1	0.000	0.888	0.000			
Lung2	0.000	0.330	0.000			
DLBCL	0.000	0.035	0.890	0.110		
Prostate	0.000	0.445	0.000			
SRBCT	0.000	0.000	0.000	0.085	0.075	
SuCancer	1.000	1.000	0.960			

6.6 Discussion

COSA-KNN and COSA-λNN are designed for the purpose of exploration of the data, and thus do not come with the possibility of specifying a beforehand known number of clusters to be found in the data. However, when we apply the MVPIN algorithm to the COSA distances, COSA-λNN in combination with MVPIN seems to perform at least equally well as, if not better than, competitive state-of-the art L groups clustering algorithms in the task of retrieving a true clustering from eleven famous (gen)omics benchmark data sets.

To our knowledge, this is the first time that EWKM algorithm is compared with the L groups clustering algorithms SAS and SPARCL. While EWKM allows for clusters that have their own subset of attribute weights, SAS and SPARCL only allow for one set of attribute weights for all clusters. In our comparison SAS and SPARCL have a better performance than EWKM. Since EWKM allows for unique subsets of attribute weights, the number of local suboptima increases, and hence the sensitivity to starting values, rendering a loss in power. Compared to SAS, SPARCL, and EWKM, the results of the COSA-NN algorithms remain stable since it does not rely on random starting values.

Despite the 'overperformance bias' warning in Yousefi et al. (2011), it is still common to see that the performance of the algorithm is shown when its parameter settings are optimized (e.g. optimized for the Rand Index value) for a known posed clustering structure. For examples we refer to almost all the studies that involve EWKM or ESSC. Note that we have reported the performance of SAS, SPARCL, and COSA's and the average performance of EWKM after the parameter settings have been optimized, independent of the truth.

We argue that the overperformance bias also holds in those cases where the choice for the number of clusters is specified equal to the true number of clusters. In explorative research, the number of clusters is not known beforehand. Therefore we applied the Gap statistic to tune the number of clusters. For our benchmark data sets we have seen that specifying L based on the Gap statistic, instead of the truth, leads towards slightly worse results. An interesting finding that remains to be explained is that for all the benchmark data the Gap statistic directed SAS towards two groups, but directed the other algorithms towards a higher number of groups.

The validation of clusters and the number of clusters is a vast and ongoing area of research. In this chapter we showed how the COSA framework can also be used for cluster-validation. Given a clustering structure, we can compute an optimal COSA distance and optimal cluster attribute weights based on a self-learned cluster-specific λ_l. Then, the self-learned cluster attribute weights can be used together with attribute importance measures to filter for those attributes on which a cluster is homogeneous (the attribute on which the cluster has a low variance). For now, the validation of the clusters based on the COSA framework was shown from the perspective of permutation tests. Its properties, however, need more investigation. Moreover, to obtain a notion of stability and credibility of the 'validated' clusters or the standard errors of the attribute weights, a sub-sampling or Jackknife procedure (Efron, 1982) would be of interest to implement.

One can also run COSA with the value of λ adjusted towards a beforehand 'assumed' partitioning from a K-means or C-means based method to see whether the clustering structure could also be retrieved when there is a stronger emphasis on cluster cohesion as compared to cluster separability. While COSA focusses on the cluster homogeneous attributes, the K-means and C-means type of algorithms emphasize one set of attributes on which the clusters differ in location, and these methods focus on maximizing the separation between clusters.

All analyses and results in this chapter where based on eleven benchmark data sets. While finalizing this chapter's results the authors became aware of a new algorithm called "Robust continuous clustering" (Shah & Koltun, 2017) which was applied to an even larger set of 35 publicly available benchmark data sets from de Souto et al. (2008). In future work it would be of interest to add non-overlapping data sets and compare the performances of this chapter's algorithms with those in de Souto et al. (2008) and Shah and Koltun (2017).

Chapter 7

General Discussion

The main objective of the COSA framework is to produce distances that can capture an underlying clustering structure from high-dimensional data, where each cluster can have its own important attributes. Since its first formulation by Friedman and Meulman (2004), this monograph is the first extensive study on the properties of COSA. In Chapter 1, the background information for COSA is described. Chapter 2 provides a recapitulation of the COSA-K Nearest Neighbors (COSA-KNN) algorithm. Chapter 3 gives an explanation why median-based attribute weights are more robust than mean-based attribute weights, and in the same chapter a strategy is presented for choosing the tuning parameter values in COSA-KNN of λ and K. In Chapter 4, we reformulate COSA in such a way that only the tuning parameter λ remains necessary, referred to as COSA-λNN. Moreover, we show that with a different initialization of the attribute weights, and with a COSA distance that better separates pairs of objects in different clusters, COSA can become more powerful. To derive L clusters from the distances obtained by either COSA-λNN, or COSA-KNN, we propose in Chapter 5 a partitioning algorithm, referred to as MVPIN. In a first examination of its effectiveness, MVPIN produces promising results in combination with COSA-KNN, and especially with COSA-λNN. We compared COSA with MVPIN to other state-of-the-art L clustering algorithms in Chapter 6. We showed that COSA-λNN, but also in combination with MVPIN, is a compelling option for the clustering of high-dimensional data.

7.1 Limitations

This monograph shows that COSA has good potential for real world application. However, the many compelling examples of applications of COSA in this monograph should be considered as demonstrations. Although all the examples do provide useful insights in the behavior of the original COSA algorithm and its improvements, we should address certain limitations in more detail. These are limitations concerning the simulation studies; the computational costs of COSA, theoretical and technical

details of COSA, and missing data.

7.1.1 The Simulation Examples

The variety of models that have been used to generate high-dimensional data for COSA has been small. The two typical models that have been used as starting points were presented in the Chapter 1 of this monograph: the COSA prototype model, and the COSA weak spot model. The motivation for these two generative models has been to support the general conclusion that COSA is especially powerful in identifying clusters that have their own subset of attributes with locally low dispersion as compared to the dispersion computed for these attributes over all the objects. Therefore, it is of importance that all attributes have a similar scale. Moreover, based on these two models, it is also shown that COSA is less strong in finding clusters that have a large within-cluster variance on their own subset of attribute, when compared to the between-cluster variance on these attributes.

A further limitation is the absence of a description of the 'breakdown' points for COSA. It would be interesting to have a well informed overview of the sensitivity of COSA to changes in each within-cluster attribute dispersion, or the cluster sizes, the number of irrelevant attributes, and the number of masking objects. Moreover, the breakdown points that result from such a sensitivity analysis, would be especially valuable to know about if they could indicate when to use COSA versus other algorithms. For example, both the simple approach to sparse clustering (SAS), and the fully improved version of COSA, perform equally well on data from the prototype model as well as the weak spot model. However, empirical evidence so far suggests that when the variance between the cluster attribute means becomes smaller in the weak spot model, SAS will outperform COSA. Similarly, when in the prototype model the number of overlapping attributes becomes larger, or, when the number of masking objects becomes larger, COSA will outperform SAS.

Apart from the two generative models used as starting point, it would have been interesting to see how COSA's performance would have been effected when other families of probability distributions (e.g., mixtures of gamma distributions), are used for generative models. In Steinley and Brusco (2008), a modified version of COSA still had a competitive performance on other normal-mixture model based clustering algorithms. However, these were results based in lower-dimensional data settings ($N > P$), and data in which the underlying clusters were not allowed to have their own unique subspace of attributes. Still, Steinley and Brusco (2008) presented a well-designed comprehensive Monte Carlo study to compare a number of clustering algorithms.

To our knowledge there is, as of yet, no comprehensive (Monte Carlo) study available where distance functions for high-dimensional data are compared. There are some comprehensive studies for distance functions such as France, Carroll, and Hiong (2012), Pekalska and Duin (2005), and Aggarwal (2001). None of these studies, however, provide a comparative study between distance functions that apply a weighting strategy to the attributes in high-dimensional data.

7.1.2 Computational Costs

In a prescription by Kriegel et al. (2016) it is stated that

> 'Any paper proposing a new algorithm should come with an evaluation of efficiency and scalability (particularly when we are designing methods for "big data")'.

So far we did not provide any information on the computational costs of COSA-KNN, COSA-λNN, and any of the other algorithms. In the same study by Kriegel et al. (2016), rules of thumb are given that show that the comparison of computational costs of the algorithms (or implementations) is far from trivial.

However, the computational complexity of the original COSA-KNN algorithm is easily determined. In each iteration a distance matrix is computed on N objects and P attributes which results in $P \times N(N-1)/2$ operations, then to obtain the $N \times P$ attribute weights, we need to find the K nearest neighbors for each object i, by sorting the distances for each object using a sort method. The worst case time complexity of the sort method is $\mathcal{O}(N\log(N))$. Then, for each iteration in COSA-KNN, the worst case computing time for COSA-KNN is

$$\mathcal{O}\left(PKN^2\log(N)\right). \tag{7.1}$$

The computational complexity of COSA-λNN is more difficult to formulate. COSA-λNN has an extra merge sorting algorithm for the attributes with complexity $\mathcal{O}(P\log(P))$. Moreover, instead of having neighborhoods each being of size K, COSA-λNN allows the sizes of the neighborhoods to be different for each object. We denote the size of each neighborhood by the function of λ, $N_i(\lambda)$, and is defined as

$$N_i(\lambda) = \left|\lambda NN(i)\right|, \tag{7.2}$$

the number of the λ driven nearest neighbors of object i, as defined in equation (4.29) from Chapter 4. Having described the parameters, the COSA-λNN worst case complexity is

$$\mathcal{O}\left(P\log(P)\left[\sum_{i=1}^{N} N_i(\lambda)\right]N\log(N)\right). \tag{7.3}$$

Whether COSA-λNN or COSA-KNN has higher computational costs is dependent on the noise and clustering structure in the data. When

$$K > \left(\log(P)\sum_{i=1}^{N} N_i(\lambda)\right)/N, \tag{7.4}$$

then COSA-KNN will have higher computational costs. This particular setting occurs in a situation when data set consists of a very large proportion of noise objects, each living in small neighborhoods. However, more important is that that the cost for optimizing the tuning parameters for COSA-λNN is (much) lower than for COSA-KNN, since the optimization is over a one-dimensional (versus a two-dimensional) grid of the tuning parameter values.

It may be helpful to report that the computing time (wall clock time) for COSA-λNN and COSA-KNN was fairly equal to each other on all the data examples we used in this monograph. Another remark is that the COSA algorithms were slower than all other algorithms we ran in this monograph. However, we remark that COSA gives ample opportunity for parallelization. Moreover, the implementations of the algorithms are based on a mixture of `Fortran`, `C++`, and `R` code, each having their own compiling perks and quirks. To stay in line with at least some of the recommendations in Kriegel et al. (2016): we did use realistic data sets (Chapter 6), and all code that has been used in this monograph is published online, see `https://www.tinyurl.com/MonographCOSA`.

7.1.3 Optima and Convergence

So far, the convergence properties of COSA have not been extensively discussed in this monograph. Neither proof, nor empirical support has been given to show that the COSA algorithms converge. Empirical evidence so far suggests that the solution for the attribute weights in COSA most likely converges. For some empirical data examples, for example the COSA weak spot model, we see that COSA ends in an oscillation between two solutions for $\mathbf{W}$. This 'back and forth' process between two solutions typically occurs when local minima for $Q(\mathbf{W} \mid \mathbf{C})$ and $\widetilde{Q}(\mathbf{C} \mid \mathbf{W})$ are equally attractive, but not compatible. When this occurs, we advise to use the solution for the attribute weights that has the minimum value for $Q(\mathbf{W} \mid \mathbf{C})$. This oscillating process between the two solutions can be referred to as a second order stationary process, and loosely speaking, could be seen as convergence.

For small data sets we could find the global minimum. Consider a data set of $N = 20$ objects, on which we wish to run COSA-KNN with $K = 10$. Then, it is still feasible to find the global minimum for the leading criterion $Q(W \mid C)$.Out of the

$$\binom{N}{K} = 184,756$$

neighborhoods of size K, there are 'only'

$$\binom{N-1}{K} = 92,378$$

neighborhoods for each object for which we need to find the minimum within-neighborhood sum of the distances ($D_{ij}[\mathbf{w}]$). For COSA-λNN a comparable, but feasible, strategy can be created to find the global minimum for $Q(W \mid C)$.

When investigating the convergence properties, the homotopy strategy in COSA also needs to be considered. As was described in Chapter 2, the COSA algorithms have an homotopy strategy implemented to avoid convergence to suboptimal local minima. Whereas in COSA-KNN the suboptimal local minima are avoided due to a linear path for the homotopy parameter η, the COSA-λNN only applies the homotopy strategy at the start of the first iteration and then iterates without a homotopy path, i.e. $\eta = \lambda$ at the start, but after the first itertation η is set equal to ∞. For the

examples in this monograph, however, we would not have obtained differences in the interpretation of the COSA-λNN results, compared to the situation where the linear path was used.

To show empirically that suboptimal local minima are avoided with the homotopy strategy, it could be informative to plot the values for each iteration of the within-neighborhood criterion $Q(\mathbf{W}|\mathbf{C})$, the criterion with the COSA distances $\widetilde{Q}(\mathbf{C}|\mathbf{W})$, and the iteration number. The comparison of a plot for the results of COSA with the (linear) homotopy strategy, on the one hand, and for the results where no homotopy strategy was used, on the other hand, could lead to insights for new homotopy strategies.

It may be easy to come up with a homotopy strategy that would improve the path with which suboptimal local minima are avoided. The homotopy parameter η regularizes the Kullback-Leibler divergence between each of the t_{ijk}'s and v_{ijk}'s, as given in the second term of the inverse exponential distance:

$$D_{ij}^{\eta}[\mathbf{W}] = \min_{\mathbf{t}_{ij}} \left\{ \sum_{k=1}^{P} t_{ijk} d_{ijk} + \eta \sum_{k=1}^{P} t_{ijk} \log \left(\frac{t_{ijk}}{v_{ijk}} \right) \right\} \tag{7.5}$$

(equation 2.47 from Chapter 2). An idea to improve the path over which the distance evolves is to assure that in iteration b with homotopy parameter η_b, the regularized Kullback-Leibler divergence is never larger than the divergence term for any of the object pairs in the next iteration, $b+1$. In this way it can be assured that the role of the solution for the attribute weights ($\mathbf{v}_{ij}$) becomes more important with each iteration. However, so far, such strategies result in considerably higher computational costs.

Another open avenue that could be explored is to avoid local minima by introducing fuzzy membership for the clusters or neighborhoods. Whereas in COSA the membership of object i to cluster (or neighborhood) l could only be true ($c_{il} = 1$) or false ($c_{il} = 0$), a fuzzy version of the criterion, where $0 \leq c_{il} \leq 1$ with

$$\sum_{l=1}^{L} c_{il} = 1, \tag{7.6}$$

could smooth away some of the local optima in the criterion. For a comparable strategy and its properties, see Heiser and Groenen (1997). Thus, apart from smoothing the distances between the objects with a homotopy strategy, the attribute weights could also be smoothed across the objects with the use of fuzzy cluster memberships.

7.1.4 Missing Data

A problem of practical importance is how well COSA deals with missing data. In Chapter 4 we did apply COSA to a data set that contained missing values, but how COSA copes with incomplete data has not been discussed. When either object i or object j have a missing value on attribute k, then the attribute weight v_{ijk} is modified

with the following rule:

$$v_{ijk} \leftarrow I(x_{ik} \neq \text{missing})I(x_{jk} \neq \text{missing})v_{ijk}. \tag{7.7}$$

For each object i' that has missing values on attributes, the attribute weights on the non-missing values are re-normalized as

$$w_{kl^*_{i'}} \leftarrow w_{kl^*_{i'}} \Big/ \sum_{k=1}^{P} I(x_{i'k} \neq \text{missing}) w_{kl^*_{i'}} \ . \tag{7.8}$$

Thus, for those object pairs where at least one of the objects has a missing value on attribute k, we set v_{ijk} equal to a value of zero, and the attribute weights for the non-missing attributes $w_{kl^*_{i'}}$ are renormalized to sum to 1 for each object i'. If for two objects there are no overlapping non-missing attribute values, then these two objects are assigned an infinite COSA distance.

When applying COSA to data with missing values, one should be aware of resulting effects from the above strategy. Suppose objects i and j have no missing attributes, have exactly the same clustering pattern, and let object i' be a version of object i where at least one of the attribute values x_{ijk} is missing. Then, with COSA's strategy on dealing with missing values we could have the following inequality property for the attribute weights:

$$\sum_{k=1}^{P} v_{i'jk} > \sum_{k=1}^{P} v_{ijk}. \tag{7.9}$$

Due to the re-normalization of the attribute weights, this inequality property (7.9) (7.9) would strongly hold when object i has missing values on the attributes that would have received attribute weights above average. The consequence of this property is that the COSA distance between object i' and j becomes larger than it should have been, risking that objects i' and j will not end up in the same cluster.

The reverse effect occurs when object i' has missing values on the attributes that would have received weight values below average, when the attributes are not important for the clustering. Then, due to the re-normalization of the attribute weights we obtain

$$\sum_{k=1}^{P} v_{i'jk} < \sum_{k=1}^{P} v_{ijk}. \tag{7.10}$$

This inequality property (7.10) results in a smaller COSA distance for objects i' and j, and is less problematic.

In Chapter 4 we applied COSA-KNN and COSA-λNN on a benchmark gene expression data set for breast cancer tumors (Perou, 2000) that contained missing values, and COSA seems to cope well. A very likely reason is that there were missing values on those attributes that were irrelevant for the clustering of the involved objects, while these objects did have values on the attributes important for clustering. For such a specific distribution of missing values, re-normalization of the attribute weights will not have a detrimental effect on the COSA distances.

The renormalization strategy in equation (7.8) can have harmful effects for object pairs where all attributes have equal importance. Let us create the following three scenarios. In the first scenario, scenario **i.**, we have a complete data set of $N = 100$ objects by $P = 1,000$ non-missing attributes that consists of noise values only (i.i.d $\sim N(0, 1)$). In the second and third scenario we have for 20 out of the $N = 100$ objects that either have **ii.** 200 missing values, completely at random, out of the $P = 1,000$ attributes for each object; or **iii.** 200 missing values on exactly the same attributes for each object. Figure 7.1 displays the typical average linkage dendrogams for our COSA-KNN distances.

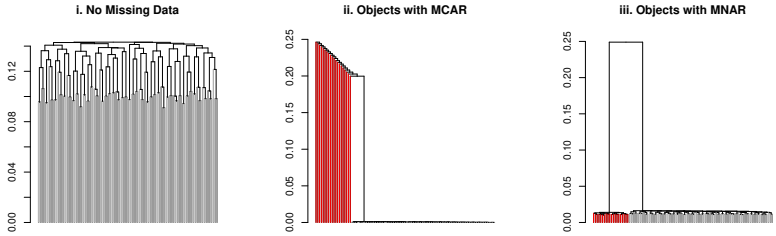

Figure 7.1: Results of COSA-KNN distances of 'noise only' data, displayed in average linkage dendrograms. The left dendrogram gives the results based on a data set with no missing values, the middle dendrogram is based on the data set with 20 objects (in red) that have missing values completely at random (MCAR) on 200 attributes (out of the 1,000 attributes), the right dendrogram is based on a data set where 20 objects (in red) have 200 missing not at random (MNAR) values on exactly the same attributes. The COSA distances are normalized to have a sum of squares equal to N.

In the three scenario's all objects should be approximately equidistant to each other. However, as we can see in Figure 7.1, for scenario's **ii.** and **iii.** the distances seem to be systematically different for the objects with missing values (colored red) and the objects that have no missing values (in black). In scenario **ii** we see that the objects with no missing values are somehow very similar to each other, and the objects with missing values are very distant to the other objects. Moreover, note that the objects with missing values have the largest distances between each other. In scenario **iii.** all objects that have the exact same MNAR pattern for the attribute values seem to be equidistant to each other. Similarly, all objects without missing attribute values also seem to be equidistant to each other. While the MCAR scenario in practice can easily be detected and used as an advice to discard the objects that have missings, the MNAR may lead to misleading clustering conclusions. However, also the MNAR disturbance can be easily detected when re-running COSA on the attributes that do not contain any missing values. Apart from the need of being aware of such results, it may be of interest to further study the behavior of COSA to recognize results that are systematically influenced by missing values. Still, it is a strength that COSA can cope with missing values. Especially since the typical K-means, or fuzzy C-means,

algorithms are not able to cope with missing values at all.

7.2 Future Avenues

In the previous section we stipulated how some limitations of this monograph could be further studied as research problems in future investigations. In this section we give a small overview on topics that we did not deal with in this monograph, but are still deemed noteworthy in relation to the COSA framework. Some of these topics have already been proposed or studied in Friedman and Meulman (2004), Kampert, Meulman and Friedman (2017). Others are open avenues for further study that have not (yet) received any attention at all.

7.2.1 Different regularization strategies for the attributes

While the COSA algorithms uses Kullback-Leibler divergence regularization, we conjecture that a family of COSA algorithms with closed-form solutions for the attribute weights can be created from other classes of divergences as well. A canonical example of a regularization based on the Bregman divergence (Bregman, 1967) is COSA criterion where, instead of the Kullback-Leibler divergence, the Squared Euclidean distance between the attribute weights in $\mathbf{W}$ and the initial attribute weights $\{\mathbf{u}_l\}_{l=1}^{L}$ is regularized. Especially in the light of the new COSA-λNN algorithm, where the number of zero-value attribute weights is also steered by λ, these different regularization forms could be interesting directions for future research.

Another interesting direction is to simplify the COSA framework. Instead of attribute weights, a hard crisp subset of equally weighted attributes may lead to better and simpler results. The number of attributes that are selected for each neighborhood could be selected based on the largest gap between the sum of the within-neighborhood attribute dispersions and a reference sum of attribute dispersion from random neighborhoods. Note that this strategy could also be seen as a modification of SAS towards a COSA approach, resulting in different COSA distances.

7.2.2 COSA Distances

We have defined a new COSA distance in Chapter 4 to create a stronger separation between objects from different clusters. We have seen that v_{ijk} could be defined differently from being the maximum of $w_{kl_i^*}$ and $w_{kl_j^*}$. So far, we have only one restriction for any definition of a COSA distance, i.e. the COSA distance should reduce to a COSA within-cluster distance when $v_{ijk} = w_{kl_i^*} = w_{kl_j^*}$ for all atributes k, expressed as

$$D[\mathbf{W}] = D[\mathbf{w}_i] = D[\mathbf{w}_j]. \tag{7.11}$$

Thus, we could study new COSA distances with the purpose to incorporate between-cluster distances based on the centroids of clusters, i.e. the within-cluster average value for each attribute. Let $\overline{d}_{kl_{ij}}$ be the distance between the average value on attribute k

for cluster l_i, on the one hand, and the average value on attribute k for cluster l_j, on the other hand. Then, we could consider a following definition for v_{ijk}:

$$v_{ijk} = \frac{w_{kl_i^*} + w_{kl_j^*}}{2} + \left(\frac{|w_{kl_i^*} - w_{kl_j^*}|}{1 - |w_{kl_i^*} - w_{kl_j^*}|} \right) \left(\frac{1 + \overline{d}_{kl_{ij}}}{\eta} \right). \tag{7.12}$$

Here, η is the homotopy parameter, and with this definition of v_{ijk}, we obtain one of the many possible definitions in the family of COSA distances. Note that when $w_{kl_i^*} = w_{kl_j^*}$ holds for all k, then we also have $v_{ijk} = w_{kl_i^*} = w_{kl_j^*}$ for each k.

So far, we have shown examples that merely indicate that there is a wide open world to explore for COSA distances. Another direction could be to create proper COSA distances that have a perfect fit with an MDS configuration that is constrained on finding clusters, or even ultrametric COSA distances from which a dendrogram could be formed directly.

7.2.3 Targeting and the Attribute Distances[1]

Not only the COSA distances, but also the attribute distances have potential for further research. In this monograph the COSA clustering could be on any possible joint values on subsets of attributes. However, in Friedman and Meulman (2004) it was also possible to look for clusters that group on particular values. This was referred to as 'targeted clustering' and can actually also be seen as a first step that may bring us closer towards distances based on composite kernel spaces (Wange et al., 2016). Examples of the usefulness of targeting can be found in Friedman and Meulman (2004), as well as in Kampert, Meulman and Friedman (2017); here we just give a short explanation on what we mean by targeted attribute distances.

Suppose that objects only cluster on particular values, say y_k, which are possibly different for each attribute k. Here, the $\{y_k\}$ are chosen to be of special interest; and can be used to reduce the search space of the solutions for the clustering structure; when chosen correctly, these targets render it more likely to recover clusters. Examples are groups of consumers (objects) that spend relatively large amounts on products (attributes), while we wish to ignore consumers who spend relatively small or average amounts (or the other way around). If we focus on one particular value, we call this single targeting. We modify the original distance between objects i and j on attribute k, $d_{ijk} = d(x_{ik}, x_{jk})$, into targeted distances, and require objects i and j to be close to each other *and* to the particular target.

The so-called single target distance is defined as:

$$d_{ijk}(t_k) = \max[d_k(x_{ik}, y_k), d_k(x_{jk}, y_k)], \tag{7.13}$$

where y_k is the target value, e.g., a *high* or *low* or even *average* value. This distance is small only if both objects i and j are close to the target value y_k on attribute k. In addition to single targeting, we can also focus on two different targets, e.g. being naturally either high or low values. An example is in microarray data, where we could

[1] A large part of this subsection is from Kampert, Meulman, and Friedman (2017).

search for clusters of samples with either high or low (but not moderate) expression levels on subsets of genes (attributes). In dual targeting, we define two targets y_{1k} and y_{2k}, and we use the dual target distance

$$d_{ijk}(y_{1k}, y_{2k}) = \min[d_{ijk}(y_{1k}), d_{ijk}(y_{2k})] \tag{7.14}$$

on selected attributes x_k, where $d_{ijk}(\cdot)$ is the corresponding single target distance (7.13). This dual target distance is small whenever x_{ik} and x_{jk} are either both close to y_{1k} *or* both close to y_{2k}. Thus, in gene expression and consumer spending examples, one might set y_{1k} and y_{2k} to values near the maximum and minimum data values of the attributes, respectively, and we will cause COSA to seek clusters based on extreme attribute values, ignoring clusters with (uninteresting) moderate attribute values.

7.2.4 Mixed Types of Attributes

In this monograph we only used 'numeric' attributes. Although, no specific advice was given on the choice for the specific attribute distance functions, COSA was originally designed for mixed type of attributes e.g. numerical and categorical (Friedman & Meulman, 2004), and this is (still) a desired and ongoing topic of research, e.g. see Grané and Romera (2016), Van de Velden et al. (2018); for an overview see Foss et al. (2018).

7.2.5 Different Objectives for COSA

Future avenues for different objectives with COSA may also be considered when relating COSA to techniques as Points of View Analysis (PVA; Tucker & Messick, 1963, Meulman & Verboon, 1993) and the Self-Organizing Map (SOM; Kohonen, 1980). In this last subsection of the monograph we will restate aspects of COSA such that it becomes relatable to techniques as PVA and SOM. We conjecture that these relatable techniques provide useful insights regarding the properties and further improvement of the COSA approach.

Points of View Analysis

When we consider the neighborhood of an object i to be a cluster l_i, then the COSA within-cluster distance is based the attribute weights of the neighborhood of object i, and is defined as

$$D[\mathbf{w}_{l_i}]_{ij} = \sum_{k=1}^{P} w_{kl_i} d_{ijk}, \tag{7.15}$$

which is a proper metric distance when each attribute distance also satisfies the metric properties. Here, we argue that this specifc COSA within-cluster distance could also be interpreted as a distance from the viewpoint of object i.

Similarly, $D[\mathbf{w}_{l_j}]_{ji}$ can be interpreted as a distance from the viewpoint of object j, and, when $\mathbf{w}_{l_i} \neq= \mathbf{w}_{l_j}$, it is most likely that

$$D[\mathbf{w}_{l_i}]_{ij} \neq D[\mathbf{w}_{l_j}]_{ji}, \tag{7.16}$$

meaning that the viewpoint of the distances from object j is different than that from object i. Suppose the objective would be to find for each of the N possible COSA within-cluster distance matrices a MDS representation for the objects, then we have for each object a visualization of each object's viewpoint. Similarly, viewpoints could be formed, based on the attribute weights from equation (6.17) in Chapter 6, for each COSA-validated cluster. Since each cluster is based on its own unique partitioning of the attributes, these viewpoints are closely related and linked to the objective in the so-called Point of View Analysis (Tucker & Messick, 1963, Meulman & Verboon, 1993).

Self-Organizing Maps

COSA is also relatable to Kohonen's Self-Organizing Maps (SOM), an artifical neural network technique based on competitive learning with which the data are non linearly projected onto a lower-dimensional display (Kohonen, 2001). Consider each object i_l from neighborhood l_i as a neuron that receives input information from the neighboring objects. Comparable with SOM, we find that the neighboring neurons for i_l will gradually specialize to represent similar inputs. In other words, when objects i and j live in closely the same neighborhoods, the attribute weights $\mathbf{w}_{l_j}$ and $\mathbf{w}_{l_j}$, for neurons l_i and l_j, respectively, will become more similar over time, i.e., during the iterations in the COSA algorithm. Although the attribute weights and the objects for each neighborhood become updated with each iteration, in COSA the attribute values of the neuron remain the same, while in SOM the attribute values of the neuron are 'smoothed' based on the nearest neighbor objects, e.g.,

$$x_{l_i k} = \sum_{j}^{N} c_{jl_i} x_{jk} \Big/ \sum_{j}^{N} c_{jl_i} \ . \tag{7.17}$$

Information Retrieval

When relating COSA to a special case of SOM, as was done in the previous section, the possibilities of COSA and information retrieval may also become apparent. E.g., an estimate for the missing value x_{ik} could be retrieved based on equation (7.17), as long as the nearest neighbors do not have missing values on attribute k. In Gabrielsson and Gabrielson (2008) and Purbey et al. (2014), SOM is being used for the purpose of information retrieval in recommender systems.

When we know the behavior of COSA in the presence of missing data, in greater detail, the COSA approach could contribute to research involved with information retrieval. Suppose that object i does not have a value on attribute k, but each of its nearest neighbors ($c_{jl_i} = 1$), has a non-missing value on the specific attribute. Then,

we may be able to compute the attribute weights w_{kl_i} when the attribute dispersion is computed as

$$S_{kl_i} = \frac{1}{N_{l_i}^2} \sum_{j=1}^{N} \sum_{j'=1}^{N} c_{jl_i} c_{j'l_i} d_{jj'k}. \tag{7.18}$$

Here, the definition of the attribute dispersion S_{kl_i} does not involve object i, itself. Thus, even though we do not know the value of object i on attribute k, we can compute to what extent attribute k is important for the clustering of object i, which is valuable information for e.g., recommender systems and imputation strategies for missing data.

References

Aggarwal, C., & Reddy, C. (2013). *Data clustering: Algorithms and applications*. CRC Press.

Aggarwal, C. C. (2003). Towards systematic design of distance functions for data mining applications. In *Proceedings of the ninth acm sigkdd international conference on knowledge discovery and data mining* (pp. 9–18). New York, NY, USA: ACM. doi: 10.1145/956750.956756

Aitchison, J. (1986). *The statistical analysis of compositional data*. London: Chapman and Hall.

Alizadeh, A. A., Eisen, M. B., Eisen, M. B., Davis, R. E., Davis, R. E., Ma, C., ... Staudt, L. M. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, *403*(6769), 503–11. doi: 10.1038/35000501

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, *96*(12), 6745–6750. doi: 10.1073/pnas.96.12.6745

Amorim, R. C. (2015). Feature relevance in ward's hierarchical clustering using the l_p norm. *Journal of Classification*, *32*, 46–62.

Andrews, J. L., & McNicholas, P. D. (2014). Variable selection for clustering and classification. *Journal of Classification*, *31*(2), 136–153.

Arabie, P., Hubert, L., & De Soete, G. (1996). *Clustering and classification*. World Scientific Publishing Company.

Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. *Pattern Recognition*, *46*(1), 243–256. doi: 10.1016/j.patcog.2012.07.021

Arias-Castro, E., & Pu, X. (2017). A simple approach to sparse clustering. *Computational Statistics and Data Analysis*, *105*, 217–228. doi: 10.1016/j.csda.2016.08.003

Arthur, D., & Vassilvitskii, S. (2007). k-means++: the advantages of careful seeding. In *Proceedings of the 18th annual acm-siam symposium on discrete algorithms.* Society for Industrial and Applied Mathematics.

Ball, G., & Hall, D. (1965). *ISODATA, a novel method of data anlysis and pattern classification* (Tech. Rep.). Stanford, CA: Stanford Research Institute.

Baraty, S., Simovici, D., & Zara, C. (2011). The impact of triangular inequality violations on medoid-based clustering. In M. Kryszkiewicz, H. Rybinski, A. Skowron, & Z. Raś (Eds.), *Foundations of intelligent systems. ismis 2011. lecture notes in computer science* (Vol. 6804, pp. 285–316). Amsterdam: Springer, Berlin, Heidelberg.

Bezdek, J. (1981). *Pattern recognition with fuzzy objective function algorithms.* Plenum Press.

Bhattacharjee, A., Richards, W. G., Staunton, J., Li, C., Monti, S., Vasa, P., ... Meyerson, M. (2001). Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences*, *98*(24), 13790–13795. doi: 10.1073/pnas.191502998

Bock, H.-H. (2007). Clustering methods: A history of k-means algorithms. In P. Brito, G. Cucumel, P. Bertrand, & F. de Carvalho (Eds.), *Selected contributions in data analysis and classification* (pp. 161–172). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-73560-1_15

Bouveyron, C., & Brunet, C. (2013). Model-based clustering of high-dimensional data: A review. *Computational Statistcs and Data Analysis*, *22*(71), 52-78.

Bregman, L. (1967). The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, *7*(3), 200 - 217. doi: https://doi.org/10.1016/0041-5553(67)90040-7

Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, *16*(3), 199–231. doi: 10.1214/ss/1009213726

Breiman, L., Friedman, J., Stone, C., & Olshen, R. (1984). *Classification and regression trees.* Taylor & Francis.

Brent, R. (1971). *Algorithms for finding zeros and extrema of functions without calculating derivatives.* Department of Computer Science, Stanford University.

Celeux M. L.; Maugis-Rabusseau, C.; Raftery, A. E., G. M.-M. (2014). Comparing Model Selection and Regularization Approaches to Variable Selection in Model-Based Clustering. *Journées Société Fran caise de Statistique*, *155*(2), 57–71.

Chen, X., Ye, Y., Xu, X., & Huang, J. Z. (2012). A feature group weighting method for subspace clustering of high-dimensional data. *Pattern Recognition*, *45*(1), 434–446. doi: 10.1016/j.patcog.2011.06.004

Chidananda Gowda, K., & Krishna, G. (1978). Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, *10*(2), 105–112. doi: 10.1016/0031-3203(78)90018-3

Commandeur, J. J. F., Groenen, P. J. F., & Meulman, J. J. (1999). A Distance-based Variety of Nonlinear Multivariate Data Analysis, Including Weights for Object and Variables. *Psychometrika*, *64*(2), 169–186.

Cormack, R. M. (1971). A Review of Classification. *Journal of the Royal Statistical Society. Series A (General)*, *134*(3), 321–367.

Critchley, F., & Heiser, W. (1988). Hierarchical trees can be perfectly scaled in one dimension. *Journal of Classification*, *5*(1), 5–20. doi: 10.1007/BF01901668

Damian, D., Oresics, M., Verheij, E., Meulman, J. J., Friedman, J., Adourian, A., ... van der Greef, J. (2007). Applications of a new subspace clustering algorithm (cosa) in medical systems biology. *Metabolomics*, *3*(1), 69–77.

De Sarbo, G., & Mahajan, V. (1984). Constrained classification: The use of a priori information in cluster analysis. *Psychometrika*, *49*, 187–215.

De Soete, G., & Carroll, J. D. (1994). K-means clustering in a low-dimensional Euclidean space. In E. Diday (Ed.), *New approaches in classification and data analysis* (pp. 212–219). Springer-Verlag. doi: 10.1007/978-3-642-51175-2_24

De Leeuw, J., & Groenen, P. (1997). Inverse multidimensional scaling. *Journal of Classification*, *14*, 3-21. doi: 10.1007/s003579900001

De Leeuw, J., & Heiser, W. J. (1982). Theory of multidimensional scaling. In P. Krishnaiah & L. Kanal (Eds.), *Handbook of statistics* (Vol. 2, pp. 285–316). Amsterdam: The Netherlands: North-Holland.

Deng, Z., Choi, K. S., Chung, F. L., & Wang, S. (2010). Enhanced soft subspace clustering integrating within-cluster and between-cluster information. *Pattern Recognition*, *43*(3), 767–781. doi: 10.1016/j.patcog.2009.09.010

Deng, Z., Choi, K. S., Chung, F. L., & Wang, S. (2011). EEW-SC: Enhanced entropy-weighting subspace clustering for high dimensional gene expression data clustering analysis. *Applied Soft Computing Journal*, *11*(8), 4798–4806. doi: 10.1016/j.asoc.2011.07.002

Deng, Z., Choi, K. S., Jiang, Y., Wang, J., & Wang, S. (2016). A survey on soft subspace clustering. *Information Sciences*, *348*, 84–106. doi: 10.1016/j.ins.2016.01.101

De Sarbo, W., Carroll, J., Clarck, L., & Green, P. (1984). Synthesized clustering: a method for amalgamating clustering bases with differential weighting of variables. *Psychometrika*, *49*, 57–78.

De Soete, G. (1985). Ovwtre: a program for optimal variable weighting for ultrametric and additive tree fitting. *Journal of Classification*, *5*, 101–104.

De Soete, G., De Sarbo, W., & Carroll, J. (1985). Optimal variable weighting for hierarchical clustering: analternating least-squares algorithm. *Journal of Classification*, *2*, 173–192.

de Souto, M. C., Costa, I. G., de Araujo, D. S., Ludermir, T. B., & Schliep, A. (2008). Clustering cancer gene expression data: A comparative study. *BMC Bioinformatics*, *9*, 1–14. doi: 10.1186/1471-2105-9-497

Dettling, M. (2004). BagBoosting for tumor classification with gene expression data. *Bioinformatics*, *20*(18), 3583–3593. doi: 10.1093/bioinformatics/bth447

Duda, R., Hart, P., & Stork, D. (2001). *Pattern classification.* Wiley.

Dudoit, S., & Fridlyand, J. (2002). A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biologyiology*, *3*(7), research0036.1–0036.21. doi: 10.1186/gb-2002-3-7-research0036

Dunn, J. C. (1973). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, *3*(3), 32–57. doi: 10.1080/01969727308546046

Efron, B. (1982). *The jackknife, the bootstrap, and other resampling plans.* Society for Industrial and Applied Mathematics.

Efron, B., & Hastie, T. (2016). *Computer age statistical inference: Algorithms, evidence, and data science.* Cambridge University Press.

Fisher, L., & van Ness, J. W. (1971). Admissible clustering procedures. *Biometrika*, *58*(1), 91–104. doi: 10.1093/biomet/58.1.91

Florek, K., Lukaszewicz, J., Perkal, J., & Zubrzycki, S. (1951a). Sur la Liaison et la Division des Points dun Ensemble Fini. *Colloquium Mathematicae*, *2*, 282–285.

Florek, K., Lukaszewicz, J., Perkal, J., & Zubrzycki, S. (1951b). Taksonomia Wroclawska. *Przeglad Antropol.*, *17*, 193–211.

Foss, A. H., & Markatou, M. (2018). Clustering mixed-type data in r and hadoop. *Journal of Statistical Software*, *83*(13). doi: 10.18637/jss.v083.i13

Fraley, C., & Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, *97*(458), 611–631. doi: 10.1198/016214502760047131

France, S. L., Douglas Carroll, J., & Xiong, H. (2012). Distance metrics for high dimensional nearest neighborhood recovery: Compression and normalization. *Information Sciences*, *184*(1), 92–110. doi: 10.1016/j.ins.2011.07.048

Friedman, J. H. (1987). Exploratory Projection Pursuit. *Journal of the American Statistical Association*, *82*(397), 249–266. doi: 10.1007/978-3-642-45103-4_1

Friedman, J. H., & Meulman, J. J. (2004). Clustering objects on subsets of attributes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *66*(part 4), 815–849.

Friedman, J. H., & Tukey, J. W. (1974). A projection prusuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, *C-23*(9), 881-889.

Gabrielsson, S., & Gabrielsson, S. (2006). *The use of self-organizing maps in recommender systems.* (unpublished thesis)

Golub, T. (2017). golubesets: exprsets for golub leukemia data [Computer software manual]. (R package version 1.20.0)

Golub, T. R., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., ... Lander, E. S. (1999). Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, *286*(5439), 531–537. doi: 10.1126/science.286.5439.531

Gordon, G. J., Jensen, R. V., Hsiao, L. L., Gullans, S. R., Blumenstock, J. E., Ramaswamy, S., ... Bueno, R. (2002). Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research*, *62*(17), 4963–4967. doi: 10.1158/0008-5472.can-04-2818

Gower, J. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, *27*, 857-871. doi: 10.2307/2528823

Gower, J. C. (1966). Some distance properties of latent roots and vector methods used in multivariate analysis. *Biometrika*, *53*, 325–338.

Grané, A., & Romera, R. (2018). On Visualizing Mixed-Type Data: A Joint Metric Approach to Profile Construction and Outlier Detection. *Sociological Methods and Research*, *47*(2), 207–239. doi: 10.1177/0049124115621334

Hancer, E., & Karaboga, D. (2017). A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number. *Swarm and Evolutionary Computation*, *32*, 49–67. doi: 10.1016/j.swevo.2016.06.004

Harpaz, R., & Haralick, R. (2007). Linear manifold correlation clustering. *International Journal of Information Technology and Intelligent Computing*, *2*(2).

Hartigan, J. (1975). *Clustering algorithms.* Books on Demand.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *28*(1), 100–108.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *Elements of Statistical Learning: Data Mining, Inference and Prediction (Second Edition)*. New York: Springer-Verlag.

Heiser, W. J. (1995). Convergent computation by iterative majorization: Theory and applications in multidimensional data analysis. In W. Krzanowski (Ed.), *Recent advances in descriptive multivariate analysis* (pp. 157–189). Oxford University Press.

Hoff, P. D. (2006). Model-based subspace clustering. *Bayesian Analysis*, *1*(2), 321–344. doi: 10.1214/06-BA111

Hubert, L., & Arabie, P. (1985). Comparing Partitions. *Journal of Classification*, *2*, 193–218. doi: 10.1007/BF01908075

Jain, A. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, *31*(8), 651–666.

Jain, A., & Dubes, R. (1988). *Algorithms for clustering data*. Prentice Hall PTR.

Jardine, N., & Sibson, R. (1968). The construction of hierarchic and non-hierarchic classifications. *The Computer Journal. Section A / Section B*, *11*.

Jin, J., & Wang, W. (2016). Influential features PCA for high dimensional clustering. *Annals of Statistics*, *44*(6), 2323–2359. doi: 10.1214/15-AOS1423

Jing, L., Ng, M. K., & Huang, J. Z. (2007). An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data. *IEEE Transactions on Knowledge and Data Engineering*, *19*(8), 1026–1041.

Jolliffe, I. T., & Philipp, A. (2010). Some recent developments in cluster analysis. *Physics and Chemistry of the Earth*, *35*(9-12), 309–315. doi: 10.1016/j.pce.2009.07.014

Kampert, M., Meulman, J., & Friedman, J. (2017). rcosa: A software package for clustering objects on subsets of attributes. *Journal of Classification*, 514–547. doi: 10.1007/s00357-017-9240-z

Kaufman, L., & Rousseeuw, P. (1990). *Finding groups in data: An introduction to cluster analysis*. Wiley.

Kaufmann, L., & Rousseeuw, P. (1987). Clustering by means of medoids. *Data Analysis based on the L1-Norm and Related Methods*, 405-416.

Kettenring, J. R. (2006). The practice of cluster analysis. *Journal of Classification*, *23*(1), 3–30. doi: 10.1007/s00357-006-0002-6

Khan, J., Wei, J. S., Ringnér, M., Saal, L. H., Ladanyi, M., Westermann, F., ... Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, *7*(6), 673–679. doi: 10.1038/89044

Kleinberg, J. M. (2003). An impossibility theorem for clustering. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems 15* (pp. 463–470). MIT Press.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*(1), 59–69. doi: 10.1007/BF00337288

Kohonen, T. (2001). *Self organizing maps*. Springer Verlag.

Kohonen, T., Huang, T., & Schroeder, M. (2001). *Self-organizing maps*. Springer Berlin Heidelberg.

Kou, J. (2014). Estimating the number of clusters via the GUD statistic. *Journal of Computational and Graphical Statistics*, *23*(2), 403–417. doi: 10.1080/10618600.2013.778778

Kriegel, H.-P., Kröger, P., & Zimek, A. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transacionts on Knowledge Discovery from Data*, *3*, 1–58.

Kruskal, J. (1972). Linear transformation of multivariate data to reveal clustering. In R. Shepard, A. Romney, & S. Nerlove (Eds.), *Multidimensional scaling. theory and applications in the behavioral sciences.* (Vol. 1, pp. 179–191). Amsterdam: Seminar Press, New York.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86. doi: 10.1214/aoms/1177729694

Laio, A., & Rodriguez, A. (2014). Clustering by fast search and find of density peaks. *Science*, *344*(6191), 1492–1496. doi: 10.1126/science.1242072

Lance, G. N., & Williams, W. T. (1967). A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems. *The Computer Journal*, *9*(4), 373–380. doi: 10.1093/comjnl/9.4.373

Lee, A., Luca, D., Klei, L., Devlin, B., & Roeder, K. (2009). Discovering genetic ancestry using spectral graph theory. *Genetic epidemiology*, *34*, 51-9. doi: 10.1002/gepi.20434

Lee, A. B., Luca, D., & Roeder, K. (2010). A spectral graph approach to discovering genetic ancestry. *The Annals of Applied Statistics*, *4*(1), 179–202. doi: 10.1214/09-AOAS281

Lee, J. S., & Olafsson, S. (2011). Data clustering by minimizing disconnectivity. *Information Sciences*, *181*(4), 732–746. doi: 10.1016/j.ins.2010.10.028

Lee, J. S., & Olafsson, S. (2013). A meta-learning approach for determining the number of clusters with consideration of nearest neighbors. *Information Sciences*, *232*, 208–224. doi: 10.1016/j.ins.2012.12.033

Leone, F. C., Nelson, L. S., & Nottingham, R. B. (1961). The folded normal distribution. *Technometrics*, *3*(4), 543-550. doi: 10.1080/00401706.1961.10489974

Lloyd, S. P. (1982). Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129–137. Originally as an unpublished Bell laboratories Technical Note (1957). doi: 10.1109/TIT.1982.1056489

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth berkeley symposium on mathematical statistics and probability, volume 1: Statistics* (pp. 281–297). Berkeley, Calif.: University of California Press.

Martins, A. F. T., & Astudillo, R. F. (2016). *From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification.* doi: 10.1109/72.279181

McLachlan, G., Bean, R., & Ng, S. (2017). Clustering. In *Bioinformatics vol ii: Structure, function, and applications* (pp. 345–362). Humana Press.

McQuitty, L. L. (1957). Elementary linkage analysis for isolating orthogonal and oblique types and typal relevancies. *Educational and Psychological Measurement*, *17*, 207–229.

Meulman, J. J. (1986). *A Distance Approach To Nonlinear Multivariate Analysis.* Leiden: DSWO Press.

Meulman, J. J. (1992). The Integration of Multidimensional Scaling and Multivariate Analysis with Optimal Transformations. *Psychometrika*, *57*, 539–565.

Meulman, J. J. (2003). Prediction and classification in nonlinear data analysis: something old, something new, something borrowed, something blue j. *Psychometrika*, *68*(4), 493–517.

Meulman, J. J., & Heiser, W. J. (2010). *IBM SPSS Categories 19.* Chicago: IBM-SPSS Inc.

Meulman, J. J., & Verboon, P. (1993). Points of View Analysis Revisited: Fitting Multidimensional Structures to Optimal Distance Components with Cluster Restrictions on the Variables. *Psychometrika*, *58*(1), 7–35.

Milligan, G. (1996). Clustering validation: Results and implications for applied analyses. In P. Arabie, L. Hubert, & G. De Soete (Eds.), *Clustering and classification* (pp. 345–375). Oxford: World Scientific Publishing Company.

Milligan, G. W. (1980). An examination of the effect of six types of error perturbation on fifteen clustering algorithms. *Psychometrika*, *45*, 325–342.

Milligan, G. W. (1989). A validation study of a variable weighting algorithm for cluster analysis. *Journal of Classification*, *6*(1), 53–71. doi: 10.1007/BF01908588

Mohajer, M., Englmeier, K.-H., & Schmid, V. (2011). A comparison of gap statistic definitions with and without logarithm function. *Computing Research Repository - CORR*, 1–20.

Murtagh, F., & Legendre, P. (2014). Ward's Hierarchical Clustering Method: Which Algorithms Implement Ward's Criterion? *Journal of Classification*, 274–295. doi: 10.1007/s00357-014-9161-z

Oghabian, A., Kilpinen, S., Hautaniemi, S., & Czeizler, E. (2014). Biclustering methods: Biological relevance and application in gene expression analysis. *PLoS ONE*, *9*(3). doi: 10.1371/journal.pone.0090801

Parsons, L., Haque, E., & Liu, H. (2004). Subspace clustering for high dimensional data: A Review. *ACM SIGKDD Explorations Newsletter*, *6*(1), 90–105. doi: 10.1145/1007730.1007731

Patrick, E. A. (1973). Clustering Using a Similarity Measure Based on Shared Near Neighbors. *IEEE Transactions on Computers*, *C-22*(11), 1025–1034. doi: 10.1109/T-C.1973.223640

Pekalska, E., & Duin, R. (2005). *The dissimilarity representation for pattern recognition: Foundations and applications*. World Scientific.

Perou, C. M., Sørlie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S., Rees, C. A., ... Botstein, D. (2000). Molecular portraits of human breast tumours. *Nature*, *406*, 747.

Pinsker, M. (1964). *Information and information stability of random variables and processes*. Holden-Day.

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., Mclaughlin, M. E., ... Ii, T. R. G. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, *415*(January), 436–442.

Purbey, N., Pawde, K., Gangan, S., & Karani, R. (2014). Using Self-Organizing Maps for Recommender Systems. *International Journal of Soft Computing and Engineering*, *4*(5), 47–50.

R Core Team. (2014). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria. Retrieved from `http://www.R-project.org/`

Romano, S., Vinh, N. X., Bailey, J., & Verspoor, K. (2016). Adjusting for Chance Clustering Comparison Measures. *Journal of Machine Learning Research*, *17*, 1–32. doi: https://dx.doi.org/10.1615%2FJWomenMinorScienEng.2012002908

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, *23*(19), 2507–2517. doi: 10.1093/bioinformatics/btm344

Scott, A., & Symons, M. (1971). Clustering methods based on likelihood ratio criteria. *Biometrics*, *27*, 387–389. doi: 10.2307/2529003

Sebestyen, G. (1962). *Decision-making processes in pattern recognition.* Macmillan.

Shah, S. A., & Koltun, V. (2017). Robust continuous clustering. *Proceedings of the National Academy of Sciences*, *114*(37), 9814–9819. doi: 10.1073/pnas.1700770114

Singleton, R. C. (1969). Algorithm 347: an efficient algorithm for sorting with minimal storage [M1]. *Commun. ACM*, *12*(3), 185–186. doi: 10.1145/362875.362901

Sneath, P. H. A. (1957). The application of computers to taxonomy. *Journal of General Microbiology*, *17*, 201–226.

Sokal, R. R., & Michener, C. D. (1957). A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, *38*, 1409–1438.

Sorensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biologiske Skrifter*, *5*, 1–34.

Steinhaus, H. (1956). Sur la division des corps materiels en parties. *Bulletin of the Polish Academy of Sciences*, *4*(3), 801–804. doi: citeulike-article-id:3738255

Steinley, D. (2004). Properties of the Hubert-Arabie adjusted Rand index. *Psychological Methods*, *9*(3), 386–396. doi: 10.1037/1082-989X.9.3.386

Steinley, D., & Brusco, M. J. (2008). Selection of variables in cluster analysis: An empirical comparison of eight procedures. *Psychometrika*, *73*(1), 125–144. doi: 10.1007/s11336-007-9019-y

Strehl, A., & Ghosh, J. (2002). Cluster ensembles - A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, *3*(3), 583–617. doi: 10.1162/153244303321897735

Su, A. I., Welsh, J. B., Sapinoso, L. M., Kern, S. G., Dimitrov, P., Lapp, H., ... Hampton, G. M. (2001). Molecular Classification of Human Carcinomas by Use of Gene Expression Signatures. *Cancer Research*, *61*(858), 7388–7393. doi: doi: 10.1063/1.4772594

Sun, W., Wang, J., & Fang, Y. (2012). Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electronic Journal of Statistics*, *6*. doi: 10.1214/12-EJS668

Tibshirani, R., & Walther, G. (2005). Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, *14*(3), 511–528. doi: 10.1198/106186005X59243

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*, 411–423.

Tolosana-Delgado, R., Otero, N., & Pawlowsky-Glahn, V. (2005). Some Basic Concepts of Compositional Geometry. *Mathematical Geology*, *37*(7), 673–680. doi: 10.1007/s11004-005-7374-8

Topsøe, F. (2000). Some inequalities for information divergence and related measures of discrimination. *IEEE Transactions on Information Theory*, *46*(4), 1602–1609. doi: 10.4134/JKMS.2008.45.1.197

Topsøe, F. (2001). Bounds for Entropy and Divergence for Distributions over a Two-Element Set. *Journal of Inequalities in Pure and Applied Mathematics*, *2*(2), 1–13.

Torgerson, W. (1952). Multidimensional scaling: I. theory and method. *Psychometrika*, *17*, 713–726.

Tucker, L. R., & Messick, S. (1963). An individual differences model for multidimensional scaling. *Psychometrika*, *28*(4), 333–367. doi: 10.1007/BF02289557

Tukey, J. (1977). *Exploratory data analysis.* Addison-Wesley Publishing Company.

Van Mechelen, I., Bock, H. H., & De Boeck, P. (2004). Two-mode clustering methods: A structured overview. *Statistical Methods in Medical Research*, *13*(5), 363–394. doi: 10.1191/0962280204sm373ra

van Buuren, S., & Heiser, W. J. (1989). Clustering n objects into k groups under optimal scaling of variables. *Psychometrika*, *54*, 699-706. doi: 10.1007/BF02296404

van der Laan, M. J., & Pollard, K. S. (2003). A New Partitioning Around Medoids Algorithm. *Journal of Statistical Computation and Simulation*, *73*(8), 575–584.

van de Velden, M., Iodice D'Enza, A., & Markos, A. (2018). Distance-based clustering of mixed data. *Wiley Interdisciplinary Reviews: Computational Statistics*(November). doi: 10.1002/wics.1456

van Os, B. (2001). *Dynamic programming for partitioning in multivariate data analysis.* Leiden: Department of Data Theory, Leiden University.

van Wieringen, W. N., & van der Vaart, A. W. (2015). Transcriptomic Heterogeneity in Cancer as a Consequence of Dysregulation of the GeneCGene Interaction Network. *Bulletin of Mathematical Biology*, *77*(9), 1768–1786. doi: 10.1007/s11538-015-0103-7

Wang, F., & Sun, J. (2014). Survey on distance metric learning and dimensionality reduction in data mining. *Data Mining and Knowledge Discovery*, *29*(2), 534–564. doi: 10.1007/s10618-014-0356-z

Wang, J., Deng, Z., Choi, K. S., Jiang, Y., Luo, X., Chung, F. L., & Wang, S. (2016). Distance metric learning for soft subspace clustering in composite kernel space. *Pattern Recognition*, *52*, 113–134. doi: 10.1016/j.patcog.2015.10.018

Wang, S., & Zhu, J. (2008). Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics*, *64*(2), 440–448. doi: 10.1111/j.1541-0420.2007.00922.x

Wang, Y., Klijn, J. G., Zhang, Y., Sieuwerts, A. M., Look, M. P., Yang, F., ... Foekens, J. A. (2005). Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet*, *365*(9460), 671–679. doi: 10.1016/S0140-6736(05)70933-8

Ward Jr, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*(301), 236–244.

Williams, G., Huang, J. Z., Chen, X., Wang, Q., & Xiao, L. (2014). wskm: Weighted k-means clustering [Computer software manual]. Retrieved from `http://CRAN.R-project.org/package=wskm` (R package version 1.4.19)

Wishart, D. (1969). Note : An Algorithm for Hierarchical Classifications. *Biometrics*, *25*(1), 165–170. doi: 10.2307/2528688

Witten, D. M., & Tibshirani, R. (2010). A framework for feature selection in clustering. *Journal of the American Statistical Association*, *105*(490), 713–726. doi: 10.1198/jasa.2010.tm09415

Wiwie, C., Baumbach, J., & Röttger, R. (2015). Comparing the performance of biomedical clustering methods. *Nature Methods*, *12*(11), 1033–1038. doi: 10.1038/nmeth.3583

Young, F., & Householder, A. (1938). Discussion of a set of points in terms of their mutual distances. *Psychometrika*, *3*, 19–22.

Yousefi, M. R., Hua, J., Sima, C., & Dougherty, E. R. (2010). Reporting bias when using real data sets to analyze classification performance. *Bioinformatics*, *26*(1), 68–76. doi: 10.1093/bioinformatics/btp605

Index

Summary

The research in this monograph is focused on clustering of objects in high-dimensional data, given the assumption that the objects do not cluster on all the attributes, or not even on a single subset of attributes, but often on different subsets of attributes in the data. With the objective to reveal such a clustering structure, Friedman and Meulman (2004) proposed a framework and a specific algorithm, called COSA. Instead of producing clusters directly, COSA gives a representative distance matrix that can subsequently be analyzed by a variety of distance-based analysis methods, such as hierarchical clustering or multidimensional scaling. In this monograph we study the behavior of COSA to demonstrate its usefulness, but also a number of weaknesses, and we propose improvements to address the latter.

Chapter 1

The important notions of the context in which COSA is embedded are cluster analysis, regularized attribute weighting in high-dimensional data settings, and distances. To gain insight into data, and to be able to generate hypotheses, or detect anomalies and salient features one can perform a cluster analysis. Cluster analysis is the search for natural groupings in a data set of objects that are measured on attributes, while conducted in absence of the group labels of each object. When successful, these natural groupings are 'tightly' knit and preferably distinct from each other on the attributes.

The cluster analysis should involve an attribute weighting strategy, since in high-dimensional data settings it is assumed that there are many attributes irrelevant for the clustering of a group. Possibly, each cluster could have its own subset of relevant attributes. The main challenge for these attribute weighting strategies in cluster analysis is to produce stable results. By limiting the solution space for the values of the attribute weights, stability can be increased

For any type of cluster analysis, there is always a notion of (dis)similarity between the objects. In COSA, the representative distance function has the ability to reveal clusters that are formed in attribute subspaces of high dimensional datasets. Multi-dimensional scaling analysis (MDS) configurations, as well as the dendrogram that results from hierarchical clustering, are fruitful visualizations of the COSA distances with which these particular clustering structures can be revealed.

Chapter 2

The COSA algorithm that is the starting point in this monograph is the Friedman and Meulman algorithm that is based on a K nearest neighbors strategy, and will be referred to as COSA-KNN. The main purpose of COSA-KNN is to produce a matrix that contains the distances between N objects. A distance between object i and j, denoted by D_{ij}, is constructed from a weighted sum of P attribute distances. These P attribute distances, denoted by $\{d_{ijk}\}_{k=1}^{P}$, are in turn constructed from measurements collected in a $N \times P$ data set. The data set is assumed to have an underlying clustering structure in which the objects are clustered on cluster-specific subsets of attributes. However, COSA-KNN circumvents the need to specify the expected number of clusters in the data by the use of a K nearest neighbors method.

The attribute distances are weighted with a λ-regulated constraint on the negative entropy of the attribute weights. This particular constraint ensures that COSA-KNN can produce sensible solutions for the attribute weights of each cluster in high-dimensional data settings where $P \gg N$. The definition of the COSA distance matrix itself ensures that the distances for the object pairs within a cluster will always be smaller than the distances for the between-cluster object pairs. We will refer to this property of the distances as 'majorizing'.

The 'majorizing' distances are obtained from an iterative algorithm that heuristically minimizes a the COSA-KNN criterion. We start with an initial set of attribute weights from which a first distance matrix can be derived. Then, based on a K nearest neighbors method, new attribute weights are calculated, from which in turn a new distance matrix is derived, and so on. This iterative procedure is continued until a particular convergence criterion is reached. Within this iterative process COSA-KNN uses a strategy based on the homotopy between the criterion and an approximate criterion for COSA-KNN such that the algorithm can elegantly avoid inferior local minima.

Chapter 3

The choice of the values for two tuning parameters in COSA is crucial to detect a subtle clustering structure in the data. These two tuning parameter are λ and K. The definition of λ is related to the range of the values that defines a grouping of objects on an attribute, and it directly determines the weights of the attributes. The definition of K is the the size of the neighborhood for each object in a cluster.

The number of successful values of the tuning parameters can be increased when we apply a robust version of COSA-KNN. In the robust version of COSA-KNN, we use median-based estimates for the attribute weights, compared to the originally proposed mean-based estimates in COSA-KNN. The robust version of COSA-KNN is faster and more successful in revealing clustering structures in the data.

To automatically find successful combinations for the values of the tuning parameters, the so-called Gap statistic procedure is used (based on Tibshirani et al., 2001). The Gap statistic procedure is a permutation approach in which we select that particular combination of tuning parameters that provides the largest gap between

the value of the robust COSA-KNN criterion obtained for a particular data set, and (an approximation of) the expected value obtained for a null-reference model, i.e. a comparable data set with no clustering structure.

While the non-robust COSA-KNN criterion seems to be a concave smooth surface over a grid of values for both λ and K, the robust version of the criterion shows a zigzag pattern over the odd and even values for K. This particular zigzag pattern is also propogated in the Gap statistic values, leading towards the preference of even values for K, over odd values for K. Since the estimates based on even values for K are in general a little bit more stable, we suggest to use a grid with preferably only even values for K.

Chapter 4

We make COSA-KNN more powerful by implementing four improvements:

i. by a change of notation in the definition, we can generalize the clustering problem from Chapter 2, to settings where the assumption of mutually exclusive clusters does not need to hold;

ii. instead of letting λ regulate the negative entropy of the attribute weights for each neighborhood, we let λ regulate the Kullback-Leibler divergence between the attribute weights and a pre-specified set of attribute weights. This way, we can incorporate pre-specificied attribute weights that indicate the importance of each attribute in the clustering;

iii. we reformulate the criterion to allow for zero-valued attribute weights, also driven by λ;

iv. we change the COSA distance such that it is more succesful in separating pairs of objects from different clusters.

While COSA-KNN would not be able to detect clusters that mainly differ in their mean on attributes where the within-cluster variances are equal to those of the noise objects, these improvements make COSA-KNN to do so. Moreover, they make COSA-KNN more flexible and more powerful.

In COSA-KNN each object is assigned the same number of K nearest neighbors. Instead of setting the size of each neighborhood to a fixed value for K, we can let the size of each neighborhood be driven by the tuning parameter λ, making K superfluous. This approach and its associated algorithm will be called COSA-λNN.

COSA-λNN performs equally well as the improved version of COSA-KNN, if not better. The advantage of COSA-λNN over COSA-KNN is not only that each neighborhood can be of a different size, but it also comes with a large reduction in computing costs since the value for K does not need to be tuned anymore. While COSA-KNN needs tuning for all value combinations of λ and K, in COSA-λNN only the value for λ needs to be tuned.

Chapter 5

Although the main output of COSA is the distance matrix that we have labeled as "cluster-happy", it is not straightforward to extract L groups from this matrix. COSA by itself is not equipped to be compared directly with L-groups clustering algorithms. To extract L clusters from the COSA distances we propose to apply a new algorithm called MVPIN, since it uses a Minimum Variance strategy to Partition In Neighborhoods.

MVPIN applies a restricted form of Ward's method; the restriction is based on the popular Jarvis and Patrick (1973) shared nearest neighbors similarity. While using Ward's Minimum Variance method, clusters are only allowed to merge if the similarity measure based on the *shared nearest neighbors* between the two clusters is the highest among the similarities that pass a certain threshold.

When compared to competing clustering algorithms such as Partitioning Around Medoids (PAM; Kaufman & Rousseeuw, 1987), hierarchical clustering with average linkage, or Ward's method, a first examination shows that MVPIN is more successful at detecting the clustering structure for prototypical COSA data. In particular, MVPIN performs well in detecting small homogeneous clusters that are similar to each other, because of a similar clustering pattern on an overlapping subset of overlapping attributes. In combination with the Gap statistic procedure, MVPIN shows better overall results than those obtained by PAM and Ward's method on the (optimized) COSA-λNN distances for 12 benchmark data sets.

Chapter 6

Like some of the current 'state-of-the-art' L-groups clustering algorithms that include a form of regularized weighting of the attributes, COSA is largely motivated by the analysis of omics data. State-of-the-art L-groups clustering algorithms that were either inspired by, or compared themselves with COSA (Friedman & Meulman, 2004) are Entropy Weighted K-means clustering (EWKM; Jing, Ng, & Huang, 2007), Sparse clustering (SPARCL; Witten & Tibshirani, 2010), Simple Approach to Sparse clustering (SAS; Arias-Castro & Pu, 2017). When we replicate the results of these algorithms, and compare them with the results of MVPIN applied to the original COSA-KNN distances (Chapter 3) and COSA-λNN distances (Chapter 4), we find that COSA-λNN and SAS are the overall winners. The comparison is based on the analysis of 11 omics benchmark data sets.

Instead of framing COSA-λNN and the other algorithms as competitors of each other, we can also use them for validation of each others results. Given a clustering structure, we can compute an optimal COSA distance and optimal cluster attribute weights based on a self-learned cluster-specific λ_l. Then, the self-learned cluster attribute weights can be used together with attribute importance measures to filter for those attributes on which a cluster is homogeneous (the attribute on which the cluster has a low variance). The validation of the clusters based on the COSA framework can be formulated as a permutation test.

Chapter 7

This monograph ends with a discussion chapter in which a number of topics are being reviewed. These are the limitations in the monograph related to the choice of the simulation examples, the study of the computational costs of COSA-KNN and COSA-λNN, convergence properties, and how missing data are dealt with within the COSA framework. Furthermore, various aspects of possible future improvements and extensions are discussed. For possible improvements, we discuss a different regularization strategy for the weighting of the attributes, the COSA distances, and the distances at the attribute level. We also discuss how the COSA framework can be extended to a framework for Information Retrievial, Self-Organizing Maps (Kohonen, 1980) or Point of View Analysis (Tucker & Messick, 1963; Meulman & Verboon, 1993). These shortcomings, improvements, and extensions together serve as a plea for further study of COSA.

Summary in Dutch (Samenvatting)

Dit proefschrift richt zich op het clusteren van objecten in hoogdimensionale data, waarbij de aanname geldt dat de objecten niet op alle attributen clusteren, zelfs niet op een enkele subset van attributen, maar vaak op verschillende subsets van attributen. Met het doel om een dergelijke clusterstructuur te kunnen vinden, stelden Friedman en Meulman (2004) een raamwerk voor met een specifiek algoritme COSA genaamd. In plaats van rechtstreeks clusters te produceren, geeft COSA een representatieve afstandsmatrix die vervolgens kan worden geanalyseerd met behulp van een groot aantal methoden voor afstandsanalyse, zoals hiërarchische clustering of multidimensional scaling analyse. Het doel van dit proefschrift is om het gedrag van COSA te bestuderen zodat we zowel het nut als de zwakke plekken kunnen aantonen en verbeteringen kunnen voorstellen die de zwakke punten aanpakken.

Hoofdstuk 1

Belangrijke begrippen voor COSA zijn cluster analyse, gereguleerde weging van de attributen in hoogdimensionale data, en afstanden. Een cluster analyse kan worden uitgevoerd om inzicht te krijgen in data en om hypothesen te kunnen genereren, of om anomalien en opvallende attributen te detecteren. Cluster analyse is het zoeken naar natuurlijke groeperingen in een dataset van objecten die worden gemeten op attributen, terwijl de groepenlabels van de objecten ontbreken. Binnen deze natuurlijke groepen zijn de objecten homogeen, en zijn de groepen bij voorkeur erg verschillend van elkaar wat betreft de waarden op de attributen.

Omdat we kunnen aannemen dat er veel irrelevante attributen zijn die niet bijdragen aan de clustering van een groep in hoogdimensionale data, dient de cluster analyse een strategie te hanteren waarmee de attributen gewogen kunnen worden. Met name zou elk cluster een eigen subset van relevante attributen kunnen hebben. De belangrijkste uitdaging voor dit soort strategieën voor het wegen van attributen is ervoor te zorgen dat de cluster analyse stabiele resultaten produceert. Door de oplossingsruimte voor de waarden van de gewichten voor ieder attribuut te beperken, kan een dergelijke stabiliteit worden vergroot.

Voor elk type cluster analyse is er altijd een notie van (dis)similariteit tussen de ob-

jecten. In COSA zijn dit de representatieve afstanden. De COSA afstanden bieden de mogelijkheid om clusters te onthullen die zijn gevormd in bepaalde subsets van de attributen in hoogdimensionale data. Multidimensional scaling analyse (MDS), evenals het dendrogram dat resulteert uit hiërarchische clustering, zijn bruikbare technieken waarmee deze specifieke clusterstructuren kunnen worden onthuld in een visualisatie van de COSA-afstanden.

Hoofdstuk 2

Het COSA-algoritme dat centraal staat in dit proefschrift afgeleid van het algoritme uit Friedman en Meulman (2004) dat gebruik maakt van een K naaste buren strategie, en hier COSA-KNN genoemd wordt. Het belangrijkste doel van COSA-KNN is om een afstandsmatrix te produceren die de afstanden tussen N objecten bevat. Een afstand tussen object i en j, aangeduid met D_{ij}, en is opgebouwd uit een gewogen som van P afstanden, één afstand voor elk attribuut: de attribuutafstand. Deze P attribuutafstanden, aangeduid met $\{d_{ijk}\}_{k=1}^{P}$, worden op hun beurt weer opgebouwd uit metingen verzameld in een $N \times P$ dataset. Er wordt aangenomen dat de dataset een onderliggende clusterstructuur heeft waarin de objecten zijn geclusterd op clusterspecifieke subsets van de attributen. Een voordeel hierbij is dat het niet noodzakelijk is om in COSA-KNN het verwachte aantal clusters op te geven; dit wordt namelijk slim omzeild door gebruik te maken van de K naaste buren strategie.

De attribuutafstanden worden gewogen op basis van een λ-gereguleerde beperking op de *negatieve entropie* van de attribuutgewichten. Deze specifieke beperking zorgt ervoor dat COSA-KNN bruikbare oplossingen kan bieden voor de attribuutgewichten van elk cluster in de hoogdimensionale data waarbij P veel groter is dan N. De definitie van de afstandsmatrix zelf zorgt ervoor dat de afstanden voor de objectparen binnen een cluster altijd kleiner zijn dan de afstanden voor de objectparen in verschillende clusters. We zullen deze eigenschap van de afstanden aanduiden als 'majorizing '.

De 'majorizing'-afstanden worden verkregen uit een iteratief algoritme dat op heuristische wijze het COSA-KNN-criterium minimaliseert. We beginnen met een eerste set attribuutgewichten waaruit een eerste afstandsmatrix kan worden afgeleid. Vervolgens worden er op basis van de K naaste buren methode nieuwe attribuutgewichten berekend, waaruit weer een nieuwe afstandsmatrix kan worden afgeleid, enzovoort. Deze iteratieve procedure wordt voortgezet totdat een bepaald convergentiecriterium is bereikt. Binnen dit iteratieve proces gebruikt COSA-KNN een strategie dat gebruikt maakt van de homotopie tussen het criterium van COSA-KNN en een 'criterium bij benadering', zodat het algoritme op elegante wijze ongewenste lokale minima kan vermijden.

Hoofdstuk 3

De keuze van de waarden voor twee instellingsparameters in COSA is cruciaal om een subtiele clusterstructuur in de data te kunnen detecteren. Deze twee paramteters zijn λ en K. De definitie van λ is het bereik van de waarden die een groep van objecten op een attribuut kenmerken, en het heeft rechtstreeks invloed op de waarden van de

attribuutgewichten. De definitie van K is de grootte van het aantal naaste buren (de buurt) voor elk object in een cluster.

Het aantal succesvolle waarden van de parameters kan worden vergroot wanneer we een robuuste versie van COSA-KNN toepassen op een dataset. In de robuuste versie van COSA-KNN gebruiken we op de mediaan gebaseerde schattingen voor de attribuutgewichten, waar in de oorspronkelijke versie van COSA-KNN attribuutgewichten gebaseerd zijn op het gemiddelde. De robuuste versie van COSA-KNN is sneller en beter in het ontdekken van clusterstructuren in de data.

Om succesvolle combinaties van de waarden van de parameters automatisch te vinden, wordt de zogenaamde Gap-statistiek procedure gebruikt (gebaseerd op Tibshirani et al., 2001). De Gap-statistiek procedure is een permutatiemethode waarin we een specifieke combinatie van parameters selecteren die de grootste kloof (= Gap) biedt tussen enerzijds de waarde van het robuuste COSA-KNN criterium dat behoort bij de dataset, en anderzijd de benadering van de verwachte waarde die hoort bij een vergelijkbare dataset zonder clusterstructuur.

Hoewel het niet-robuuste COSA-KNN criterium een concaaf vlak oppervlak lijkt te geven voor een selectie van waarden voor zowel λ als K, toont de robuuste versie van het criterium een zigzagpatroon over de oneven en even waarden voor K. Dit specifieke zigzagpatroon is ook zichtbaar in de Gap-statistiek waarden, wat leidt tot de voorkeur voor even waarden voor K boven oneven waarden voor K. Omdat de schattingen op basis van even waarden voor K over het algemeen stabieler zijn, raden we aan om alleen met even waarden voor K te werken.

Hoofdstuk 4

We verhogen de kracht van COSA-KNN door deze vier verbeteringen in te voeren:

i. met een verandering in de notatie in de definitie van het cluster probleem voor COSA uit hoofdstuk 2, kunnen we het probleem ook generalizeren naar situaties waarbij de aanname van elkaar uitsluitende clusters niet nodig is;

ii. in plaats van λ de negatieve entropie van de attribuutgewichten voor elke buurt te laten reguleren, kunnen we λ de Kullback-Leibler divergentie tussen de attribuutgewichten en een vooraf gespecificeerde set attribuutgewichten laten reguleren. Op deze manier kunnen we vooraf gespecificeerde attribuutgewichten op laten nemen in de analyse die het belang van elk attribuut in de clustering aangeven;

iii. we herformuleren het criterium om ervoor te zorgen dat er ook attribuutgewichten een nulwaarde zullen krijgen, aangedreven door λ;

iv. we passen de COSA-afstand aan zodat deze beter is in het scheiden van objectenparen in verschillende clusters.

De originele versie van COSA-KNN is niet goed in staat om clusters te detecteren die hoofdzakelijk verschillen in hun gemiddelde op attributen, en binnen-clustervarianties hebben die gelijk zijn aan die van de ruisobjecten. Dit soort clusterstructuren kunnen nu ook door COSA-KNN gevonden worden wanneer deze verbeteringen

zijn doorgevoerd in COSA-KNN. Oftewel, deze verbeteringen maken COSA-KNN flexibeler en krachtiger in meer situaties.

In COSA-KNN krijgt elk object hetzelfde aantal K naaste buren toegewezen. In plaats van de grootte van elke buurt in te stellen op een vaste waarde voor K, kunnen we de grootte van elke buurt ook laten bepalen door de parameter λ. Op deze manier maken we K dus overbodig. Deze benadering en het bijbehorende algoritme noemen we COSA-λNN.

De prestaties van COSA-λNN zijn even goed als de verbeterde versie van COSA-KNN, zo niet beter. Het voordeel van COSA-λNN ten opzichte van COSA-KNN is niet alleen dat elke buurt een andere grootte kan hebben, het gaat ook gepaard met minder rekentijd, aangezien we niet meer een succesvolle waarde voor K hoeven te zoeken. Terwijl we naar een succesvolle waardecombinaties zoeken van λ en K in COSA-KNN, hoeft dit in COSA-λNN alleen nog maar voor λ te gebeuren.

Hoofdstuk 5

Alhoewel de belangrijkste output van COSA een "cluster-happy" afstandsmatrix is, is het niet vanzelfsprekend hoe L-groepen uit deze matrix te extraheren. COSA op zichzelf kan niet automatisch worden vergeleken met cluster algoritmes die ten doel hebben om L groepen te vinden. Om L clusters uit de COSA-afstandsmatrix te extraheren, stellen we een nieuw algoritme voor, genaamd MVPIN (Minimum Variance strategy to Partition In Neighborhoods).

MVPIN is een beperktere vorm van Ward's methode. De beperking is gebaseerd op de populaire Jarvis en Patrick (1973) similariteitsmaat, het aantal overeenkomstige naaste buren. Bij het uitvoeren van Ward's minimale variantie methoden voegen we alleen twee cluster samen wanneer de similariteitsmaat voor deze twee clusters het hoogste is en ook hoger is dan een bepaalde vastgestelde drempel.

Vergeleken met concurrerende cluster algoritmes zoals Partitioning Around Medoids (PAM; Kaufman & Rousseeuw, 1987) of hiërarchische clustering gebaseerd op 'average linkage' of Ward's methode, laat een eerste onderzoek zien dat MVPIN beter presteert in het detecteren van de clusterstructuur voor prototypische datasets voor COSA. De prestaties van MVPIN zijn met name goed wat betreft het detecteren van kleine homogene clusters die erg vergelijkbaar zijn met elkaar omdat ze een groot overlappend clusteringspatroon hebben op een bepaalde subset van attributen. In combinatie met de Gap-statistiekprocedure laat MVPIN over het algemeen betere resultaten zien dan die van PAM en Ward's methode op de (geoptimaliseerde) afstanden van COSA-λNN toegepast op 12 benchmark datasets.

Hoofdstuk 6

COSA is grotendeels gemotiveerd om toegepast te worden op omics-data, net als een aantal van de huidige geavanceerde L-groepen cluster algoritmes die een vorm van gereguleerde weging van de attributen bevatten. Dit soort L-groepen cluster algoritmes die ofwel geïnspireerd door COSA, dan wel vergeleken met de oorspronkelijke COSA, dit zijn, o.a., Entropy Weighted K-means clustering (EWKM; Jing, Ng, &

Huang, 2007), Sparse clustering (SPARCL; Witten & Tibshirani, 2010), en Simple Approach to Sparse clustering (SAS; Arias-Castro & Pu, 2017). Wanneer we de resultaten van deze algoritmes repliceren en vervolgens vergelijken met de resultaten van MVPIN dat is toegepast op de oorspronkelijke COSA- KNN afstanden (Hoofdstuk 3) en de COSA-λNN afstanden (Hoofdstuk 4), dan zien we dat COSA-λNN en SAS over het algemeen het beste presteren. De vergelijking is gebaseerd op 11 omics benchmark-datasets.

In plaats van COSA-λNN en de andere algoritmes als concurrenten van elkaar te framen, kunnen we ze ook gebruiken om elkaars resultaten te valideren. Gegeven een bekend veronderstelde clusterstructuur, kunnen we een optimale COSA-afstand en optimale cluster attribuutgewichten berekenen op basis van een zelf-geleerde cluster-specifieke parameter λ_l. Vervolgens kunnen de zelf-geleerde cluster-attribuutgewichten samen met attribuut-belangrijkheidsmaten worden gebruikt om te filteren op de attributen waarop een cluster homogeen is (het attribuut waarop het cluster een lage variantie heeft). De validatie van de clusters op basis van het COSA-raamwerk kan als een permutatietest gebruikt worden.

Hoofdstuk 7

Dit proefschrift wordt afgesloten met een discussiehoofdstuk waarin een aantal onderwerpen worden besproken. Hieronder vallen mogelijk tekortkomingen met betrekking tot de keuze van de simulatievoorbeelden, de studie van de rekenkosten voor COSA-KNN en COSA-λNN, de studie naar convergentie-eigenschappen, en hoe ontbrekende data binnen het COSA-raamwerk kunnen worden behandeld. Wat er verder nog wordt besproken zijn mogelijke toekomstige verbeteringen en uitbreidingen. Onder de mogelijke verbeteringen wordt er gekeken naar een andere regularisatiestrategie voor de attribuutgewichten, de COSA-afstanden en de afstanden op het niveau van de attributen. We bespreken ook hoe het COSA-raamwerk kan worden uitgebreid tot een raamwerk voor het schatten van ontbrekende data, Self-Organizing Maps (Kohonen, 1980), en Point of View Analysis (Tucker & Messick, 1963; Meulman & Verboon, 1993). Deze tekortkomingen, verbeteringen en uitbreidingen tezamen zijn een pleidooi voor verdere studie van COSA.

Acknowledgements

I would like to thank...

Prof. Jacqueline Meulman for the tremendous number of opportunities, your inspiration, and many insights.

Prof. Willem Heiser for your indispensable help at the end of this PhD project.

Prof. Aad van der Vaart, for being available for many unannounced questions, and Prof. Jerome Friedman for your words: 'the kid is on to something'.

Vincent Buurman and Gino Kpogbezan, my "brothers from other mothers".

my office mates for the many eye-openers and fun! In particular, Harald van Mil, Sanne Willems, Rianne de Heide, Konstantin Eckle, and Stéphanie van der Pas.

"the group": Dino Festi, Pınar Kılıçer, Martin Djukanovic, and Abtien Javanpeykar.

all other colleagues and friends that I have acquired in Statistics. In particular, Wilco Emons and Marcel van Assen for introducing me to inferential statistics.

the supporting staff of the Math Institute, Snellius, and Science Education and Student Affairs.

,y friends and family. In particular,

Bestuur Groen (Claudia and Bert, Leon and Niki, Lonneke, and Maikel),

Johan and Evi,

Age, Regina, and Vé,

Anne, Sara, Pap, Mam,

Iuliana.

Curriculum Vitae

Maarten Kampert was born on October 11, 1985 in Bergen op Zoom, the Netherlands. Between 2003 and 2005 he studied classical guitar as a student of Hein Sanderink at the Conservatory in Tilburg. In 2005 he was one of the two Dutch representatives in the European Youth Guitar Ensemble.

In 2008, he received a Bachelor in Psychology (cum laude) from Tilburg University. During his study in Tilburg, he took on many active roles, among which: member of the examinations appeal board, chairman of the Study Association of Psychology (2007 - 2008), chairman of the faculty council (2008-2010), and data analysis freelancer with his own company Analutika (founded in 2008).

In 2008, he was accepted in the Research Master programme in the Social and Behavioural Sciences at Tilburg University. However, he left this programme in 2009, to join the first cohort of students of the Master programme "Statistical Science for the Life & Behavioural Sciences" in Leiden. In 2011, Maarten was the first student to have completed the Statistical Science programme in Leiden.

During his period in Leiden, the author co-founded the Young Statisticians Division of the Netherlands Society for Statistics (2011), and became an executive board member of the Society for Statistics in the Netherlands (2011 - 2018).

After his Masters, he spent a year on a research project in the Behavior Genetics department of VU Amsterdam, under the supervision of Prof. Dorret Boomsma and Jouke-Jan Hottenga. In 2012 he started his PhD project under the supervision of Prof. Jacqueline Meulman, and was appointed as lecturer with coordinating tasks in the Master Statistical Science (2013 - 2016). Currently, he is the lecturer of the course Statistical Computing with R (2014 - present).

In January 2019, he became assistant professor in Statistics at the Mathematical Institute in Leiden.