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

Kampert, M.M.D.

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.

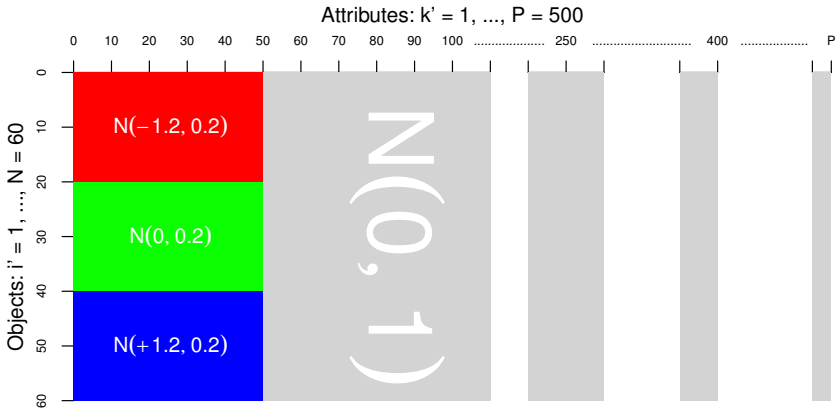

Figure 1.1: A Monte Carlo data set $\mathbf{X}$ *with 60 objects (vertical) and 500 attributes (horizontal, not all of them are shown due to* $P \gg N$*). There are three groups of 20-objects each (red, green, and blue) clustering on 50 attributes. Note that* i *and* k *are ordered into* i' *and* k'*, 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.

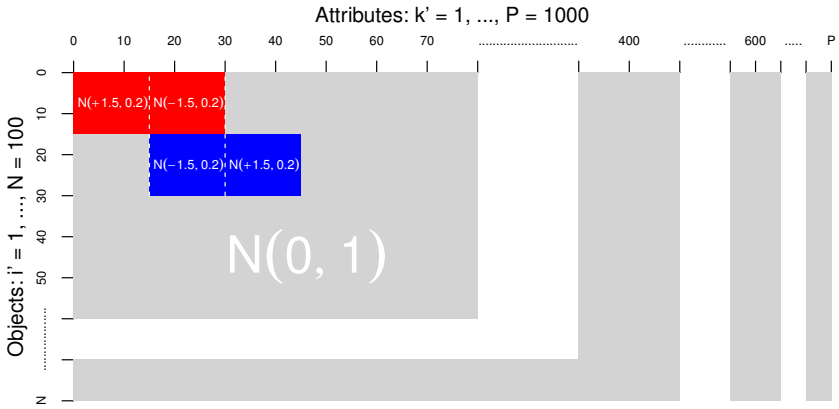

Figure 1.2: A Monte Carlo model for 100 objects with 1,000 attributes (not all are shown due to $P \gg N$). There are two small 15-object groups (red and blue), clustering each on 30 attributes out of 1000 attributes, with partial overlap, and nested within an unclustered background of 70 objects (gray). 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.

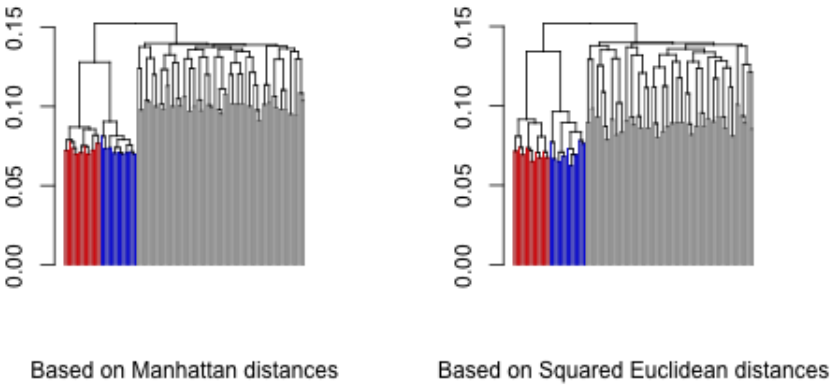

Figure 1.3: Average linkage dendrogram of the COSA distances (left panel) based on Manhattan attribute distances, and the average linkage dendrogram of the COSA distances based on squared Euclidean attribute distances (right panel).

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.

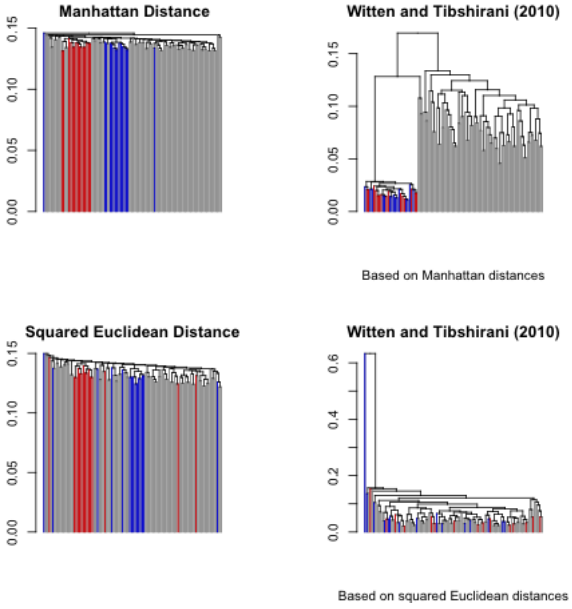

Figure 1.4: Dendrograms obtained from hierarchical clustering for four different distance matrices derived from the simulated data from the protoype model. The results of the Manhattan distances are shown in the first row, the results of the squared Euclidean distances are shown in the second row. The 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.

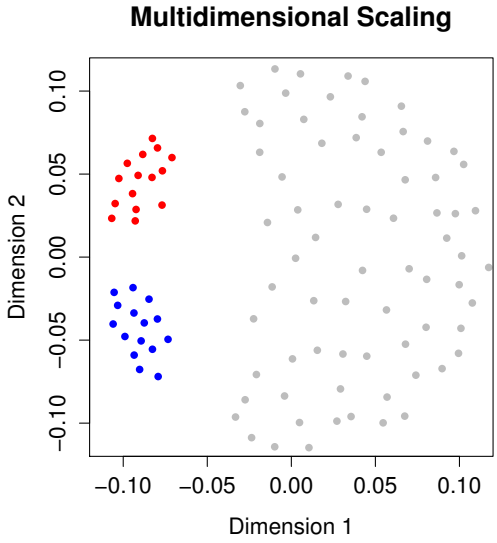

Figure 1.5: Metric least squares multidimensional scaling solution of the configuration

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.

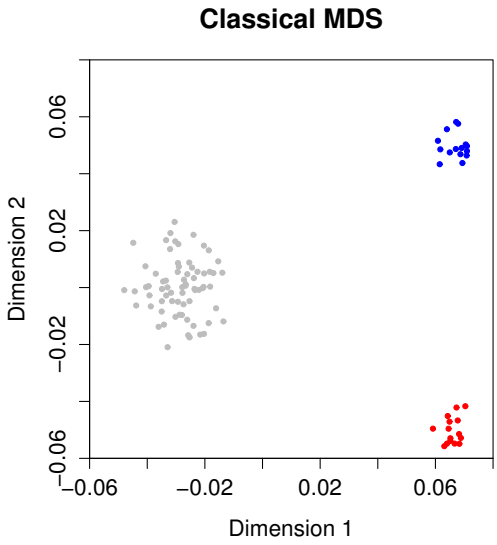

Figure 1.6: Classical multidimensional scaling solution of the configuration

.

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.

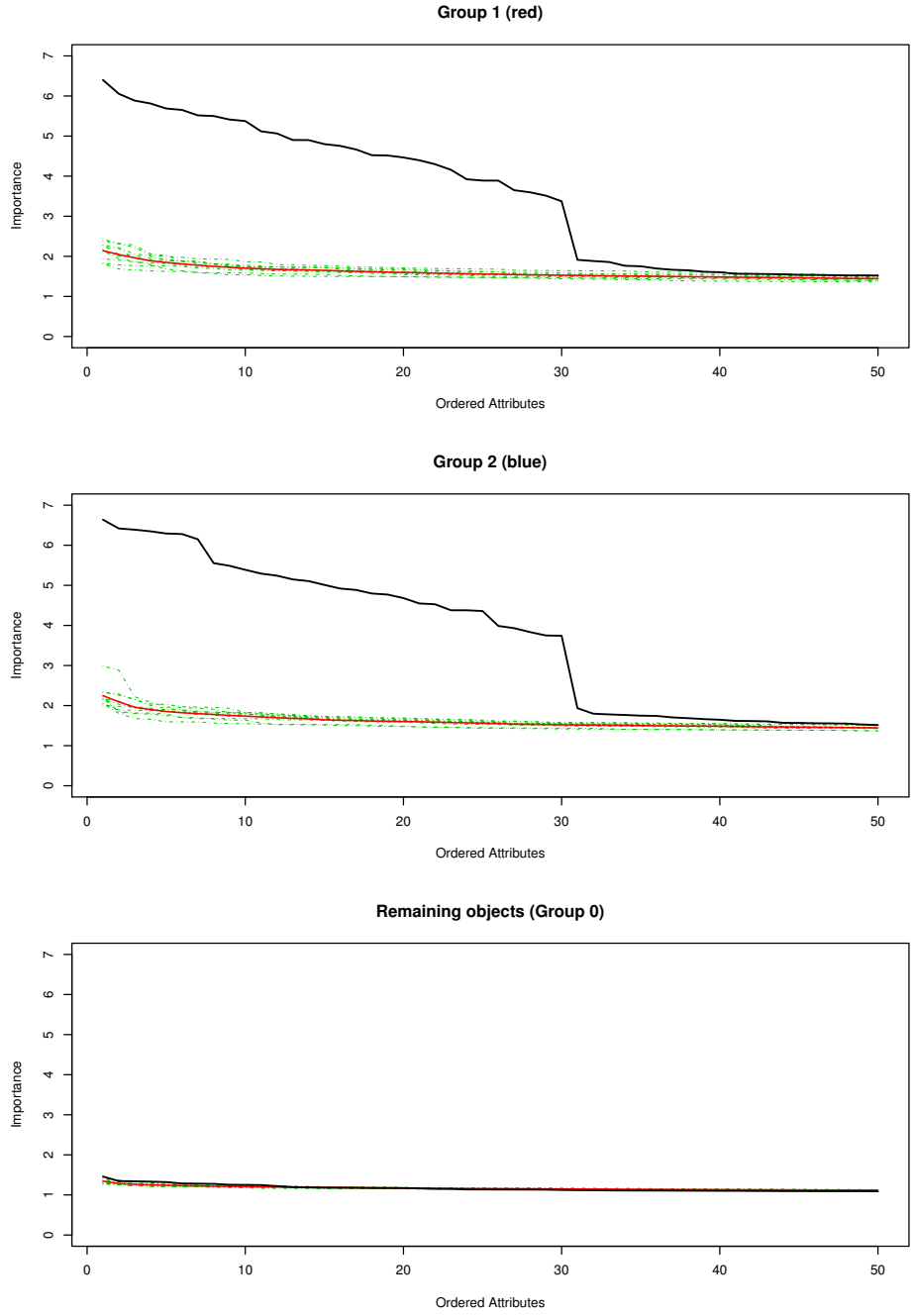

Figure 1.7: Display of the attribute importances of group 1, group 2 and the remaining objects. Here, ϵ is set equal to 0.05, such that maximum attribute importance value would have been 20.

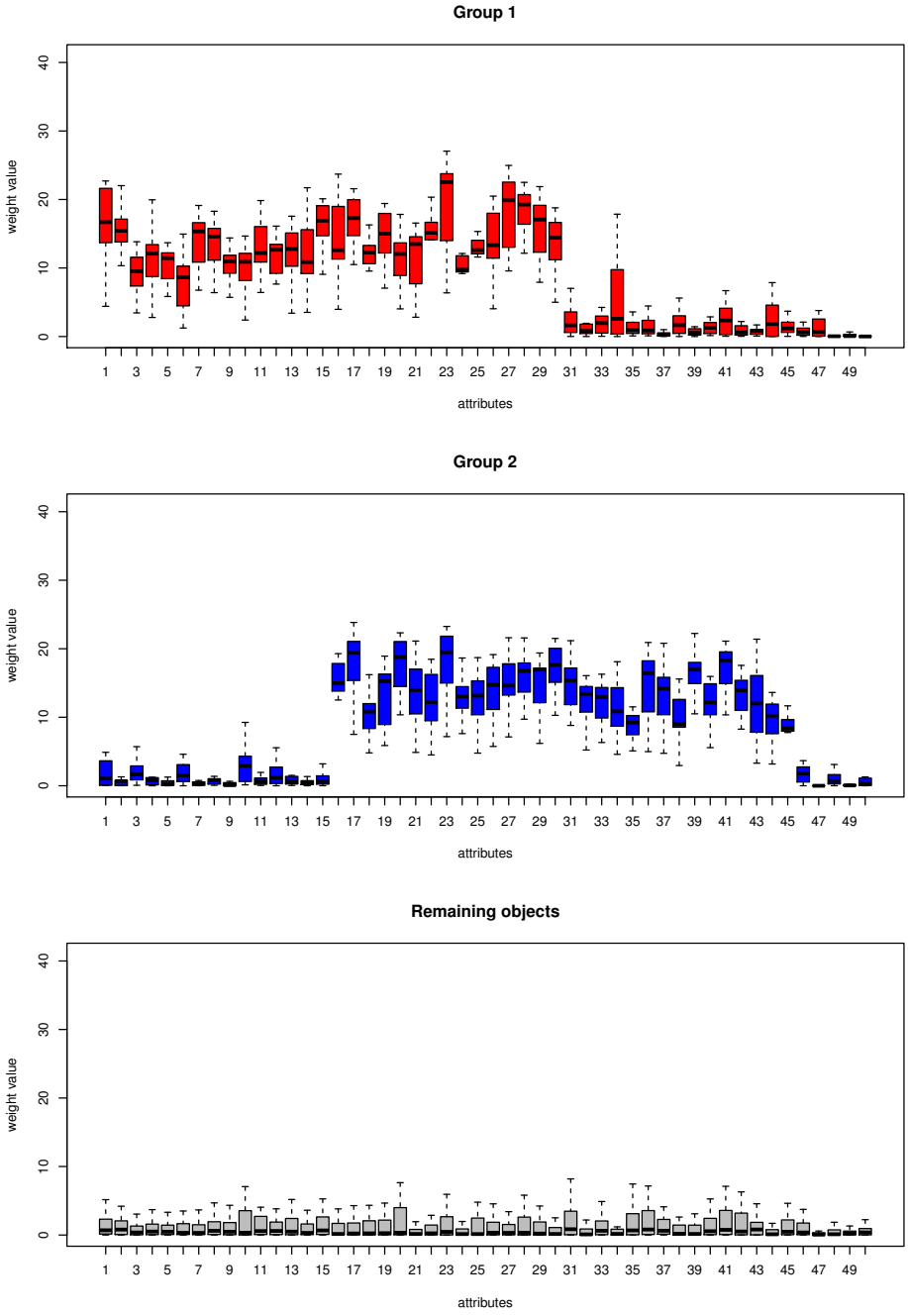

Figure 1.8: Boxplots of the weights of attributes $k' = \{1, \ldots 50\}$ for group 1, group 2 and the remaining objects.

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.