TECHNISCHE
UNIVERSITÄT
WIEN
Vienna | Austria

DISSERTATION

Robust Statistical Grouping Methods for High-dimensional Data

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften

unter der Leitung von
Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser,
Institut für Stochastik und Wirtschaftsmathematik (E105)

und
Univ.Prof. Dipl.-Ing. Dr. Christian Breiteneder,
Institut für Softwaretechnik und interaktive Systemes (E188)

eingereicht an der Technischen Universität Wien an der Fakultät für Mathematik and Geoinformation

von

Mgr. Šárka Brodinová

Matrikelnummer 1428975

Diese Dissertation haben begutachtet:
Agustín Mayo-Iscar. University of Valladolid, Spain.
Neyko Neykov. Bulgarian Academy of Sciences, Bulgaria.

Agustín Mayo-Iscar

Peter Filzmoser

Wien, 5. Oktober 2017

Šárka Brodinová

Erklärung zur Verfassung der Arbeit

Mgr. Šárka Brodinová
Spengergasse 17/22
1050 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 5. Oktober 2017

Šárka Brodinová

Acknowledgements

I would like to extend my sincerest thanks and appreciation to my supervisor Prof. Peter Filzmoser for his support and encouragement as well as for the inspiring discussions and ideas during my PhD studies. He gave me the chance to study in Vienna and continuously guided me from the beginning to the successful end of this thesis, which I am deeply grateful for. Furthermore, I want to express my thanks to Prof. Christian Breiteneder and Maia Zaharieva for their professional advice on my research and for their expertise at writing scientific papers. I also appreciate the substantial help of the co-author of my publications, Thomas Ortner. I would like to thank all my colleagues I have had the chance to meet during my studies and those who shared the office with me - they all enriched my PhD study. Last but not least, I warmly thank my family, friends, and Christoph Grubner – they all believed in me and mentally supported me during the "less shiny" moments of my studies.

Abstract

Nowadays, recent advances in modern techniques have resulted in data collections that are huge in both size and dimension. Such a trend emerges new challenges for statistical learning procedures designed to extract key information from a large amount of data. This thesis particularly addresses current challenges of unsupervised learning in the sense of data clustering and presents procedures that follow new trends in data clustering. Identifying a group structure of any real-world data becomes nowadays problematic due to several aspects. Firstly, it is well known that standard clustering procedures, e.g. k-means, are usually efficient on clean data, i.e. data without outliers, but the performance of such methods is highly affected by the presence of outliers deviating from the true underlying group structure. Hence, there is a need for a clustering method which is more robust against outliers. Furthermore, in some application domains, e.g. media domain, outliers as observations of high interest commonly form groups of very small sizes. In this context, not only the identification of such observations but also their group structure is required. Secondly, data clustering gets more difficult in high-dimensional space where the standard dissimilarity measures commonly fail. In order to overcome such limitation, dimension reduction or variable selection techniques are usually employed during data clustering. Finally, most existing clustering method commonly assume either specific group characteristics, e.g. group sizes, or even required for the number of clusters. Such assumptions might, however, be difficult to fulfill in case of real-world data. Although the main goal of this thesis is data clustering, the introduced clustering procedures additionally aim at outlier detection. For this reason, a discussion of identifying outliers in the context of a simple group structure is elaborated as well. The development of all introduced procedures is motivated by real application scenarios and the advantages of the methods are demonstrated on real-world data examples.

Kurzfassung

Die neuesten Fortschritte in modernen Techniken resultieren in riesigen Datensammlungen in Bezug auf Größe als auch Dimension. Dieser Trend lässt neue Herausforderungen für statistische Lernverfahren entstehen, welche für die Gewinnung von Schlüsselinformation einer großen Menge von Daten verantwortlich sind. Die vorliegende Dissertation thematisiert die aktuellen Anforderungen an unüberwachtes Lernen bezüglich Daten-Clustering und zeigt Verfahren auf, welche neue Trends in Daten-Clustering verfolgt werden. Die Identifikation der Gruppen-Struktur beliebiger realer Daten erscheint aufgrund verschiedener Ursachen als sehr schwierig. Zum einen ist es bekannt, dass übliche Clustering-Verfahren, wie zum Beispiel k-means, normalerweise für saubere Daten die keine Ausreißer besitzen, eine gute Effizienz aufweisen. Die Performance dieser Methoden hängt jedoch maßgeblich von der Präsenz von Ausreißern ab, die unterschiedlich zur wahren vorhandenen Gruppen-Struktur vorliegen. Daraus folgt, dass man eine Clustering-Methode benötigt, die unempfindlicher gegenüber Ausreißern ist. Darüber hinaus können Ausreißer, die als Beobachtungen von höchstem Interesse gelten, in gewissen Anwendungsbereichen wie etwa Audio- oder Video-Medien, nur sehr kleine Gruppen bilden. Folglich benötigt man nicht nur die Identifikation dieser Untersuchungen, sondern auch Kenntnis über deren Gruppenstruktur. Zum anderen wird Daten-Clustering zunehmend schwieriger wenn hochdimensionale Daten vorliegen, weil allgemein gewöhnliche Unähnlichkeits-Messungen fehlschlagen. Um diese Einschränkungen umgehen zu können werden während dem Daten-Clustering Methoden wie Dimensionsreduzierung oder Variablen-Selektion angewendet. Letztendlich nehmen die meisten der existierenden Clustering-Methoden im Allgemeinen an, dass eine spezielle Gruppen-Charakteristik, wie z.B. Gruppengröße oder Anzahl der Cluster, vorliegt. Solche Annahmen sind jedoch schwer zu erfüllen, wenn man mit realen Daten arbeitet. Obwohl das Hauptziel der vorliegenden Arbeit auf Daten-Clustering gerichtet ist, versuchen die vorgestellten Clustering-Verfahren zusätzlich die Detektion von Ausreißern zu verwirklichen. Aus diesem Grund wird die Identifikation von Ausreißern an der zugrundeliegenden Gruppenstruktur ebenfalls besprochen. Die Entwicklung aller vorgestellten Methoden wird durch Anwendungsszenarios motiviert und deren Vorteil anhand von realen Daten demonstriert.

Contents

CHAPTER 1

Introduction

This chapter focuses on different challenges when clustering real data, and provides an overview of recent developments. Since robustness to outlying observations is the main focus of this thesis, a brief overview of outlier detection methods is provided as well.

1.1 Cluster analysis

Cluster analysis is one of the most important analysis tools which allows revealing the underlying data structure in an unsupervised fashion, where – in contrast to classification tasks – no prior information is available in terms of class labels. In general, clustering aims at grouping observed data of similar characteristics into meaningful clusters. This technique has been used in various application domains. The clustering method can, for example, help to segment customers with similar interests into groups and to group images capturing similar landscape. The definition of a similarity measure highly depends on the type of method employed. For example, distance-based methods determine similarity by employing a traditional distance measure, such as the Euclidean distance. In contrast, model-based approaches assign observations into the same group if they originate from the same distribution with a high likelihood.

1.1.1 Challenges in data clustering

Despite the numerous proposed approaches and the long history of data clustering, there are still challenges that need to be taken into account when developing new clustering procedures. Most problems are caused by recent technical developments, but some challenges were already recognized decades ago (Ertöz et al., 2003). This section discusses several challenges for clustering and the following open questions: 1) how to select the number of clusters or any parameters in general; 2) how to discover clusters of various characteristics, e.g. sizes or densities; 3) how to handle data of high-dimensions; and 4) how to cope with data outliers.

The problem of determining the number of clusters is typical for most clustering algorithms (Vineet et al., 2009). While in the case of two- and three- dimensional data sets, visualization might provide valuable insights into the group structure, this can be particularly difficult in the case of high-dimensional data sets. Figure 1.1 shows three data sets with different numbers of groups; these data are publicly available in R packages: `ruspini`, `Iris`, and `DutchUtility`. While `ruspini` is a two-dimensional data set and is therefore visualized in the original data space, the next two data sets are transformed into a two-dimensional data space using principal component analysis (PCA) with the first two principal components (PC), since the dimensionality is higher (i.e. 3 and 240, respectively). Clearly, the visualization of `ruspini` indicates the presence of four groups. Although the two-dimensional representation of `Iris` shows some grouping, the number of clusters might be wrongly selected as two. Note that the colors represent the true clusters, i.e. the species of the iris flowers, but this kind of information is not known in an unsupervised setting. The limitation of this exploratory technique is more evident for `DutchUtility`, where it is impossible to see any group structure. In such situations, either a different visualization technique or internal validity indices (Sugar and James, 2003) might be more appropriate.

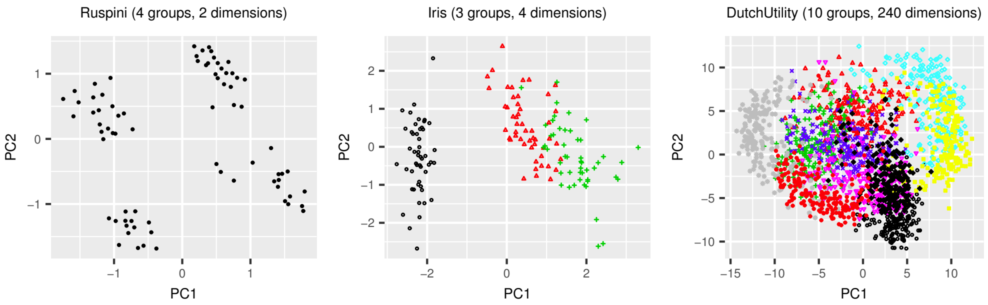

Figure 1.1: Visualization of three real-world data sets in the two-dimensional data space

Discovering groups of various characteristics (i.e. sizes, shapes, and densities) poses several additional challenges. Without proper visualization (as demonstrated previously) or pre-knowledge about the nature of the underlying groups, it is hard to state any reliable assumptions about the cluster characteristics. However, some clustering methods are built upon such assumptions. For example, model-based clustering methods usually assume that the clusters are a mixture of Gaussian densities (Bock, 1996; Fraley and Raftery, 2000). As a consequence, the detected clusters are often of convex (or elliptical) shapes. Nevertheless, model-based approaches exhibit the advantage of estimating a proper model in the sense of optimal number of clusters as well as optimal distribution parameters (Galimberti et al., 2017) by means of employing a criterion such as the BIC (Bayesian Information Criterion). In contrast to model-based clustering approaches, density-based methods are designed to detect clusters of arbitrary shapes, densities, and possibly varying sizes due to nonparametric estimation of the densities, often by using the concept of the nearest neighbors (Kriegel et al., 2011). In general, clusters are defined as regions of high density separated by areas of considerably lower densities. Although density-based methods usually do not assume any specific cluster characteristics, the estimation of density commonly depends on at least one predefined parameter (Kriegel et al., 2011), e.g. the number of nearest neighbors (Gan et al., 2007). Nevertheless, these methods demonstrate some robustness to imbalanced data where the groups are of considerably different sizes. Detecting groups of very small sizes might, however, still be problematic (He and Ma, 2013).

Data clustering in high-dimensional space has recently received a lot of attention due to the technical developments that allow measuring of an enormous number of features (variables) describing the observations (Gan et al., 2007). As far as such collected data sets are concerned, many of the extracted features are typically irrelevant for the purpose of clustering (Kriegel et al., 2009b; Sim et al., 2013). The increasing proportion of irrelevant (noise) variables decreases the effectiveness of the distance measure employed in conventional clustering methods. As the distances between the observations become more and more similar in higher dimensions, it is hard to determine whether or not the observations are actually similar in terms of the relevant (non-noise) features. A natural solution is to employ either dimension reduction (e.g. PCA) or variable selection techniques aiming at removing less important variables (Parsons et al., 2004). However, if different subsets of variables are relevant for different groups, global dimension reduction or variable selection techniques are usually not useful, as they only compute one global subspace in which data clustering is subsequently employed (Zimek, 2008). In such situations, local techniques are more appropriate to be incorporated, leading to the so-called subspace and projected clustering methods (see e.g. Aggarwal and Reddy, 2013; Sim et al., 2013; Kriegel et al., 2009b). Despite the fact that these approaches are exclusively designed to discover clusters in various data subspaces, their performance typically depends on tuning parameters (Sim et al., 2013).

The task of revealing group structure in data containing outliers poses different challenges. The presence of data outliers completely violates the general assumption of clustering methods, namely that data naturally form groups of similar observations. Outliers are observations which typically differ from the other observations, and can therefore easily destroy a clustering performance. In order to cope with this, it might be logical to apply the outlier detection procedure to exclude deviating observations, and to continue to proceed with cluster analysis. However, coping with outliers in such a way might be complicated due to the parameter specification, which is commonly required by most existing clustering (e.g. the number of clusters) as well as by outlier detection methods (Aggarwal, 2013). Alternatively, density-based clustering methods can be useful, since they commonly consider outliers (or noise) to be observations located in regions of considerably lower densities. Another way of dealing with outliers is to exclude a predefined proportion of deviating observations while applying a clustering method. The idea of excluding observations which usually do not fit to an assumed model refers to the so-called trimming-based clustering approaches. In some application domains, outliers might be of a high interest due to their atypical content, and additional information about the degree of outlyingness (deviation) can help with better understanding of the behavior of the data. However, such information can usually be provided neither by density- nor by trimming-based approaches. Therefore, Campello et al. (2015) have recently introduced a hierarchical clustering method which directly incorporates the measure of outlyingness through data clustering.

Obviously, there is no such technique that would be suitable for all kinds of applications, as different algorithms address different aspects using various concepts. In the following section, several existing methods are discussed. Generally, the selected methods cover four types of clustering approaches.

- **Distance-based clustering algorithms** assign nearby (similar) observations to one cluster, whereas distinct (dissimilar) observations are kept in separate clusters. These methods are very popular due to their simplicity, their performance is, however, commonly destroyed, when the cluster algorithm cannot cope with the various characteristics of the underlying groups, outliers and noise variables. For such situations, several methods have been modified, as presented in Section 1.1.3.
- **Model-based clustering methods** assume that data arise from a process which can be described by a finite mixture of parametric distributions. Gaussian distributions are often assumed, resulting in the detection of elliptically shaped clusters. Some modifications for both contaminated and high-dimensional data have been introduced in the literature (see Section 1.1.4).
- **Density-based clustering approaches** define clusters as regions of high density separated by areas of considerably lower densities. The densities are determined in non-parametric and often local fashion. Hence, these methods can handle arbitrarily shaped clusters and are able to deal with contaminated data. Performance of several methods is demonstrated in Section 1.1.5.
- **Subspace and projected clustering procedures** are designed to reveal clusters in high-dimensional data, within which the clusters are located in different data subspaces. While subspace clustering methods seek for all clusters in all subspaces, projected approaches attempt to find such a projection in which a cluster can be detected (Aggarwal and Reddy, 2013). The general idea of several existing methods is provided in Section 1.1.6.

It should be noted that many further methods have been developed using different concepts. For example, spectral clustering employs a graph-based data representation, which is subsequently partitioned into clusters. Such methods have been successfully applied in image processing (Shi and Malik, 2000), text mining (Dhillon, 2001), and many other applications (Aggarwal and Reddy, 2013). Another type of methods typically applied in biology is bi-clustering, often employed to analyze gene expression data. Bi-clustering approaches aim at clustering both genes (observations) and various biological conditions (variables) simultaneously (Madeira and Oliveira, 2004; Aggarwal and Reddy, 2013).

1.1.2 Real high-dimensional data example

In the following sections, a real data example is taken to demonstrate the performance of some existing clustering methods. A brief data description is provided here; a detailed description is available in Chapter 4. The data set consists of 180 archaeological glass vessels (Janssens et al., 1998), each of which is described by a 1920 dimensional extracted variable vector. The following calibration technique recovered four main groups by means of chemical concentrations (Lemberge et al., 2000) as shown in Figure 1.2. However, during the extraction, a different tool was installed, leading to the presence of two sodic subgroups; see Figure 1.2. Taking this fact into account, the number of groups can be considered to be five. It should be noted that no other information (such as the number of relevant variables or outliers) about the data is available. Therefore, the performance of the selected methods will be evaluated in terms of clustering solution achieved on on the original data set with 1920 variables. To assess the quality of the cluster solutions, a comparison with the grouping information shown in the two-dimensional visualization from Figure 1.2 is made. Note that most employed methods require parameter specifications. For this reason, information about the underlying group structure is used to optimize the parameters.

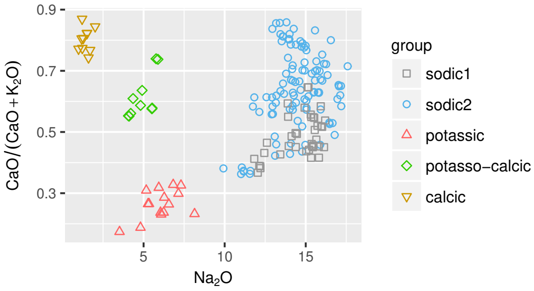

Figure 1.2: Group structure of glass vessels based on chemical concentrations (Lemberge et al., 2000).

1.1.3 Distance-based clustering algorithms

Due to their simplicity, distance-based methods are widely used and very popular. They often employ the Euclidean distance as similarity measure. Generally, such methods can be categorized into two classes: hierarchical and partitioning methods. While hierarchical clustering methods successively search for nested clusters resulting in a cluster hierarchy (a tree-like structure), partitioning algorithms find clusters simultaneously with the use of partitioning representatives, i.e. cluster centers.

Hierarchical clustering methods

The most frequently used hierarchical algorithms build clusters in an agglomerative fashion – starting with each observation representing its own cluster, and merging the most similar clusters until all observations form one cluster. Various popular criteria can be employed for merging, e.g. single-linkage, complete-linkage. While single-linkage is able to detect non-elliptical and elongated clusters, complete-linkage tends to generate spherical clusters. Another measure is considered in Ward's hierarchical clustering Murtagh and Legendre (2014): it merges those two clusters which contribute the most to the increase of the within-cluster sum of squares. In contrast, CURE (Guha et al., 1998) starts with clusters represented by well-scattered observations and continues merging them using the single-linkage approach. Alternatively, Chameleon (Karypis et al., 1999) uses a graph-based representation of data and is considered one of the best hierarchical methods in terms of discovering clusters of arbitrary shapes (Aggarwal and Reddy, 2013).

Sparse hierarchical clustering has recently been introduced by Witten and Tibshirani (2010) in order to cope with a large proportion of noise variables affecting the efficiency of a distance measure. The idea behind this approach is to first find the dissimilarity matrix calculated with respect to the variables which are relevant for data clustering, and to subsequently apply a standard merging technique to build a cluster hierarchy. Formally, let $\mathbf{X} = (\mathbf{x_1}, \dots, \mathbf{x_n})^\top$ be an $n \times p$ data matrix, where $\mathbf{x_i} = (x_{i1}, \dots, x_{ip})^\top$ for $i = 1, \dots, n$, and let $\mathbf{D} \in \mathbb{R}^{n^2 \times p}$ denote a distance matrix consisting of the distances between all observations with respect to each variable, then the approach searches for a variable weight vector $\mathbf{w} = \{w_j \geq 0, j = 1, \dots, p\}$ and a dissimilarity matrix $\mathbf{U} \in \mathbb{R}^{n \times n}$ such that

$$\mathbf{u}^\top \mathbf{D} \mathbf{w} \to \max_{\mathbf{w},\mathbf{u}}, \tag{1.1}$$

subject to $||\mathbf{u}||^2 \leq 1, ||\mathbf{w}||^2 \leq 1, ||\mathbf{w}||_1 \leq l$ for $\mathbf{w} = \{w_j \geq 0, \forall j\}$ and $l \in (1, \sqrt{p}]$, where $\mathbf{u} \in \mathbb{R}^{n^2}$ can be rewritten as the $n \times n$ matrix $\mathbf{U}$. Such a matrix represents a new dissimilarity measure employed in hierarchical clustering. The tuning parameter l controls the L_1 bound on $\mathbf{w}$, i.e. the degree of sparsity. The lower the value of l, the higher the degree of sparsity and thus the higher the number of zero elements in $\mathbf{w}$ (i.e. the lower the number of variables involved in the calculation of $\mathbf{U}$). To achieve the optimal degree of sparsity, Witten and Tibshirani (2010) also present an approach for estimating the optimal tuning parameter.

Figure 1.3 shows the results of sparse hierarchical clustering using single- and complete-linkage for the glass vessels data. Both methods are implemented in the R package `sparcl` (Witten and Tibshirani, 2013). The procedure for selecting the sparsity parameter l (Witten and Tibshirani, 2010) results in 85 variables of non-zero weights, i.e. informative variables. The obtained clustering hierarchy is cut in such a way to produce 5 clusters. The resulting clustering assignments are distinguished by different colors and symbols. Figure 1.3 clearly indicates that regardless of the merging techniques employed, sparse hierarchical clustering completely fails in discovering the groups of glass vessels.

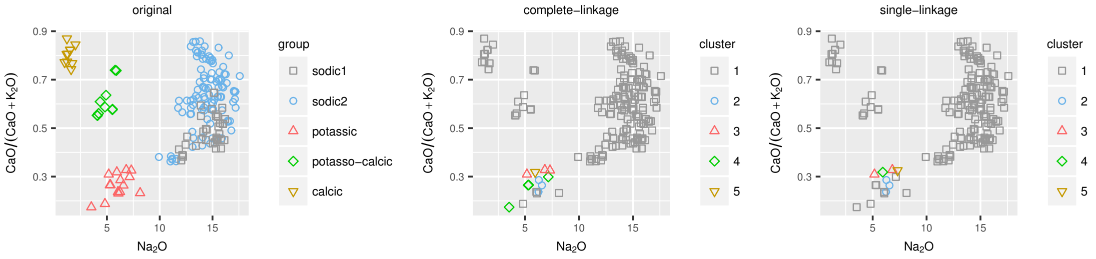

Figure 1.3: The true (original) group structure of the glass vessels (left) and the resulting cluster memberships achieved by sparse hierarchical clustering with complete-linkage (middle) and single-linkage (right).

Partitioning clustering methods

The popular k-means partitioning approach iteratively assigns each observation to the closest cluster center calculated as the arithmetic mean of the observations in one cluster. The Euclidean distance is employed to assign observations to their respective cluster center. There are some variations of k-means, mainly employing different partitioning representatives. For example, k-medians and k-medoids (also known as PAM, see Kaufman and Rousseeuw, 2005) use the median and an original observation of a cluster, respectively. The fuzzy C-means approach (Bezdek, 2013) is another popular k-means clustering method resulting in cluster memberships ranging between 0 and 1, where 1 indicates the strongest assignment of an observation to a respective cluster center.

Sparse k-means (Witten and Tibshirani, 2010) was introduced to tackle with a large number of irrelevant variables. The method searches for such a variable weight vector $\mathbf{w}$ and partitioning that renders the maximal weighted between-cluster sum of squares. As in sparse hierarchical clustering, the L_1 bound is imposed on the vector $\mathbf{w}$, controlled by the sparsity parameter l (see Chapter 4 for more details).

Trimmed k-means (Cuesta-Albertos et al., 1997) is a robust extension of k-means designed to enhance the performance of k-means on contaminated data. During the iterative procedure of k-means, $[n\alpha]$ observations with the largest distance to their respective clusters are trimmed. The remaining untrimmed observations are subsequently used to calculate new cluster centers as the arithmetic mean. The trimming assures that the calculation of new centers is not affected by trimmed observations, which are expected to be outliers. A detailed description of trimmed k-means as well as a sparse version thereof are provided in Chapter 4.

Both methods are, once again, applied to the glass vessels data set. The R implementation of sparse k-means is available in `sparcl` (Witten and Tibshirani, 2013). The procedure for the estimation of l results in no sparsity for the variable weight vector. Trimmed k-means can be computed with the R code from the package `RSKC` (Kondo et al., 2016). The trimming parameter $\alpha = 0.10$ has been selected. Figure 1.4 displays the final cluster memberships and the actual group assignments. The methods seem to be capable of discovering the two sodic subgroups, although the first subgroup appears to be divided into two clusters. While sparse k-means has difficulties to differentiate between the calcic and potasso-calcic groups, trimmed k-means cannot separate potassic glass from potasso-calcic glass samples. Therefore, a combination of both could possibly lead to a more reliable solution.

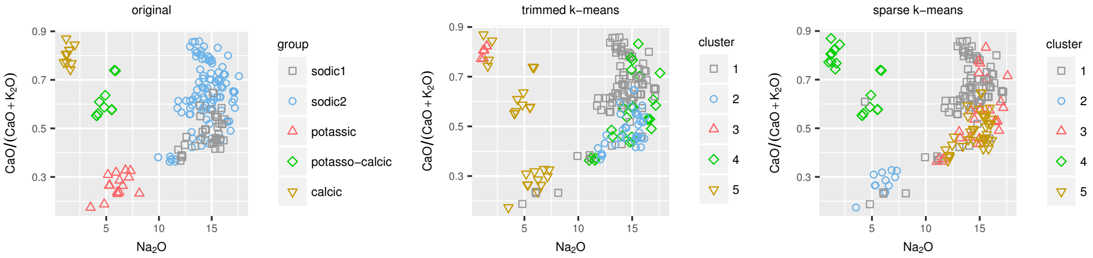

Figure 1.4: The underlying (original) group structure of the glass vessels (left), and the resulting cluster memberships achieved by trimmed (middle) and sparse (right) extensions of k-means.

1.1.4 Model-based clustering methods

The model-based approaches view observations as independent multivariate observations generated from a mixture model with a finite number of components. Usually, a mixture of Gaussian distributions is assumed (one component distribution for each cluster). Hence, the task of searching for clusters transforms into the task of estimating the Gaussian parameters, i.e. the means and the covariance matrices. Various cluster characteristics (shape, volume, and orientation) can be controlled by imposing constraints (parametrization) on the covariances, e.g. through the eigenvalue decomposition (Banfield and Raftery, 1993; Celeux and Govaert, 1995; Fraley and Raftery, 2000). Usually, the parameters are estimated via maximum likelihood using the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), and the optimal number of clusters with the optimal parametrization is commonly selected based on the Bayesian information criterion (BIC).

Robust model-based approaches

In general, there are two commonly employed strategies to robustify model-based approaches (Bock, 2002). The first approach adds an additional mixture of components into a model in order to model outliers (or noise). The second concept aims at excluding a certain proportion (α) of the most outlying observations. While in the first case, outliers are included in a model by fitting them with, for example, the Poisson (Banfield and Raftery, 1993) or the t-distribution (McLachlan and Peel, 2004), the second approach completely removes outliers from a model (Peel and McLachlan, 2000).

TCLUST (García-Escudero et al., 2008), which falls into the second category, searches for k clusters by maximizing

$$\prod_{r=1}^{k} \prod_{i \in K_r \cap L} \pi_r f(\mathbf{x}_i; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r), \tag{1.2}$$

where $f(\cdot; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$ denotes the density function of the p-variate normal distribution with mean $\boldsymbol{\mu}_r$ and covariance matrix $\boldsymbol{\Sigma}_r$, considered for each cluster. The clusters are represented by the sets $K_r, r = 1, \ldots, k$, and π_r corresponds to the cluster weights (prior probabilities) allowing for different cluster sizes. The set L contains the indices of $[n(1-\alpha)]$ (untrimmed) observations generated by a probability density function $f(\cdot)$. The remaining (trimmed) observations are supposed to be generated by a different probability density function and are thus discarded. By imposing a restriction on the eigenvalues-ratio of the covariances, the method can handle different scattering of the groups and deviations from spherically shaped groups. Both parameters k and α can be optimized using maximum likelihood curves developed by García-Escudero et al. (2011).

Further types of restrictions in this context are equal scatter matrices (Gallegos and Ritter, 2009), or constraining the determinants of the cluster scatter matrices (Gallegos, 2002). Other related algorithms can be found in the literature (see e.g. Neykov et al., 2007; García-Escudero et al., 2013).

Model-based methods in the context of high-dimensional data

It is well-known that the presence of noise variables degrades the efficiency of (not only) model-based clustering. In order to overcome this, a variable selection is commonly employed to detect a small proportion of variables that are relevant to discover the underlying group structure as well as to provide a better interpretation. Two possibilities of variable selections can be incorporated (Bouveyron and Brunet-Saumard, 2014). The variables are selected either by using the BIC or by penalizing the objective function leading to sparsity in the variables.

Variable selection based on BIC was employed by Raftery and Dean (2006) who introduced a greedy search algorithm, which compares two nested models on the full data set. A variable is included in the set of potential clustering variables if it contributes to the improvements in clustering, measured by the BIC. In contrast, noise variables are excluded if the clustering solution is not enhanced, considering all selected variables. At each stage of the search, an optimal model is chosen by means of the selected variables, the number of groups, and the parametrization of the cluster covariances.

Penalization of the objective function was incorporated by Bouveyron and Brunet-Saumard (2014) in the discriminative latent mixture model (Bouveyron and Brunet, 2012), which aims at finding the latent subspace that best discriminates the groups, using Fisher's criterion (Fisher, 1936). The subspace is, however, spanned by latent variables which are a linear combination of the original variables. This makes the interpretation of the results difficult. In order to enhance the interpretation, Bouveyron and Brunet-Saumard (2014) have proposed three procedures incorporating the L_1 penalty.

Futher approaches for dealing with high-dimensional data are based on performing subspace extraction. An overview of such methods can be found in Bouveyron et al. (2007) or Bouveyron and Brunet (2012). Bouveyron et al. (2007) also presents a method which combines the idea of subspace clustering and parsimonious modeling. The introduced method estimates a specific subspace for each group modeled by a Gaussian probability density function.

Figure 1.5 shows the performance of three model-based methods for the glass vessels data: Mclust, TCLUST, and variable selection for Mclust using BIC. The R implementation of Mclust can be found in the package `mclust` (Fraley et al., 2012). The optimal model is selected using the BIC. The R code for TCLUST is available in the package `tclust` (Fritz et al., 2012). The clustering solution shown here corresponds to the setting $\alpha = 0.10$. Finally, variable selection for Mclust using the BIC is available in the package `clustvarsel` (Scrucca and Raftery, 2014). The method selects 8 variables to be relevant for model-based clustering.

All the methods are applied with 5 clusters. The Mclust approach seems to have difficulties to detect the calcic and potasso-calcic groups as two separated clusters. Therefore, the largest sodic group (named as sodic2) is divided into two clusters as opposed to the smaller sodic group which appears to be well recovered. In contrast, TCLUST cannot detect any glass vessels group. The performance of the method indicates a rather random assignment. Moreover, TCLUST does not consider a group assignment of outliers (cluster with 0 assignment). Providing similarities of outliers with respect to clusters can be beneficial for better understanding of the data structure. It should also be noted that TCLUST was not able to provide any results based on the original data, and therefore PCA was applied with two principal components which explain about 90% of the variability in order to decrease the dimensionality. Regarding the performance of the variable selection procedure, Figure 1.5 clearly indicates that the procedure does not improve the clustering solution obtained by Mclust, as it might be expected.

1.1.5 Density-based clustering approaches

In contrast to model-based clustering, density-based procedures do not assume any parametric probability densities to group the observations. In fact, clusters are defined as areas of high densities separated from the regions of considerably low densities. The notation of density region is determined by local density of each observation commonly using nearest neighbor density estimation. Such methods can handle clusters of various characteristics and do not require pre-specification of the number of clusters.

DBSCAN (Ester et al., 1996) determines density of an observation by counting the number of its closest observations within a pre-specified radius. Observations with higher density than the pre-specified threshold are called core observations. In contrast, the observations with lower density are considered non-core observations. Such observations are excluded from data clustering and are often classified as noise or outliers. The remaining (core) observations are used to build clusters in such a way that the cluster is

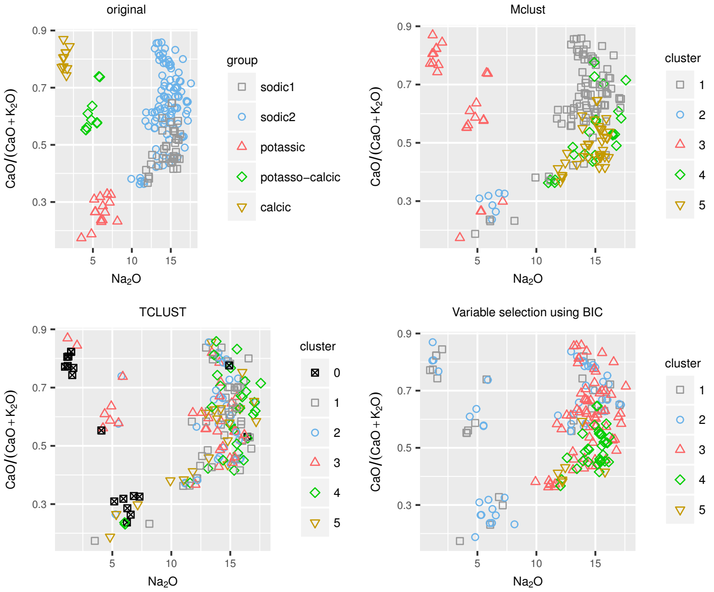

Figure 1.5: The original (true) group structure of the glass vessels data and the clustering solution obtained by three model-based methods.

formed by the core observations which are neighbors of each other. In order to employ DBSCAN, two parameters need to be specified in advance: radius, and threshold. They need to be pre-specified in order to find core observations. The method seems to be very efficient when it comes to discovering clusters of various sizes and shapes. Different cluster densities may, however, lead to poor clustering solutions (Ertoz et al., 2002).

SNN (Ertoz et al., 2002) was designed to overcome the DBSCAN limitation for situations when the clusters are of different densities. Instead of using nearest neighbors, the method employs sharing nearest neighbors. Hence, the density measure is given by the number of neighbors shared by two observations. The way of discovering both outliers and clusters is conducted in the same fashion like in case of DBSCAN.

A further approach that extends DBSCAN has been developed by Ankerst et al. (1999). The proposed OPTICS algorithm aims to successively build clusters based on reachability distances of observations. Therefore, it can be seen as a combination of hierarchical- and density-based clustering. Recently, Campello et al. (2015) have introduced HDBSCAN as an extension of OPTICS. The method directly incorporates the outlyingness measures for the purpose of outlier detection. Furthermore, in contrast to previously described approaches, the method only requires the number of nearest neighbors as input parameter. The interested readers are referred to the comprehensive overview by Aggarwal and Reddy (2013) for alternative approaches.

Figure 1.6 visualizes the resulting cluster membership obtained by the DBSCAN and SNN approaches. Both methods are implemented in R package `dbscan`. The required parameters are optimized in terms of misclassification rate (see Section 1.1.7) using the true group labels. As the methods do not require the number of clusters, the number of resulting clusters differs. While DBSCAN produced 2 groups and recovered the largest sodic group well, SNN was able to detect both sodic groups sufficiently. The smaller three glass vessels groups are mostly assigned to one cluster.

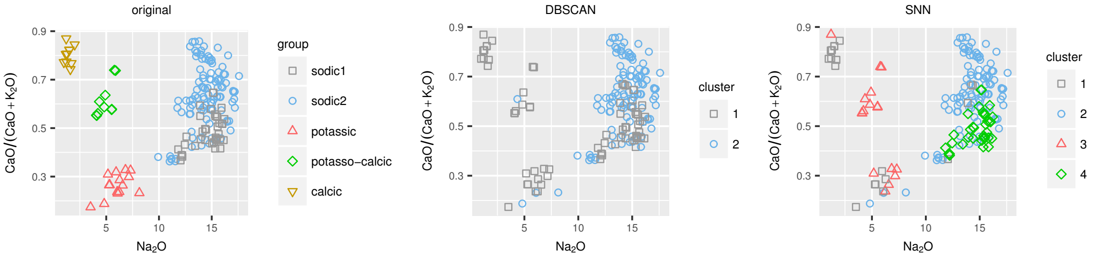

Figure 1.6: The original (true) group structure of the glass vessels data and the clustering solution obtained by two density-based approaches.

1.1.6 Subspace and projected clustering procedures

Clustering high-dimensional data is addressed by subspace and projected clustering procedures. Although some modifications of distance-based or model-based clustering could fall into the same category, this section intends to provide the basic ideas of data-mining techniques.

Subspace clustering techniques

The goal of subspace clustering techniques is to find all possible clusters in all subspaces. As a consequence, an observation can be assigned to more clusters and clusters can also have an overlapping subset of variables.

The very first approach of this kind was devolved by Agrawal et al. (1998). The introduced CLIQUE method combines density-based and grid-based concepts. The term grid-based refers to a technique which partitions each attribute into several equi-width units. Such units that contain more than a pre-specified fraction of observations are subsequently used to form clusters. The method defines a cluster as a set of connected dense units. Instead of counting the number of points, entropy-based measures can be employed to identify dense units, as proposed by (Cheng et al., 1999) in the ENCLUS approach. Despite the fact that there are further modifications of CLIQUE and different subspace techniques for mining clustering in high-dimensional data, the benefits of these methods are highly questionable. As the subspace techniques result in a large set of both clusters and their respective variables, obtaining such information might be redundant rather than useful in practice (Aggarwal and Reddy, 2013). Moreover, the performance of these techniques is often highly sensitive to the specifications of parameters (Moise et al., 2009).

Projected clustering techniques

In contrast to subspace clustering approaches, projected clustering techniques allow assigning an observation to one cluster only. In addition, the methods often result in a non-overlapping subset of variables. Usually, such methods require parameters which are often difficult to set and which are highly sensitive to their respective choices (Parsons et al., 2004; Moise et al., 2009).

One of the first approaches was introduced by Aggarwal et al. (1999). The presented PROCLUS approach randomly selects the pre-specified number of potential cluster centers. Subsequently, k observations (representing medoids) are sampled. For each medoid a relevant subset of variables is determined. In each selected variable, the distances of the respective medoid have the smallest standard deviation of distances of the considered medoid to its nearest neighbors. Such procedure is iteratively repeated until the clustering quality of the solution remains unchanged. To assess the quality, the average distance between observations and their nearest medoid is employed. The method additionally detects outliers as observations distant from their respective medoids. Instead of searching for clusters in axis parallel subspaces, ORCLUS (Aggarwal and Yu, 2000) seeks for clusters in non-axis parallel subspaces. Further existing approaches can be found in the comprehensive overviews by Parsons et al. (2004); Moise et al. (2009); Kriegel et al. (2009b).

Figure 1.7 shows 5 clusters obtained by PROCLUS. The R implementation is available in `subspace` (Hassani and Hansen, 2015). The dimensionality of a subset was selected as 500, leading to the lowest misclassification rate. Although the calcic group seems to be well recovered, the other two small glass vessels groups are once again difficult to identify. As the method is forced to detect 5 clusters, the largest sodic group is divided into two clusters. Nevertheless, the smallest sodic group can be sufficiently identified. In addition, the method detects several outliers (denoted as the cluster with 0 membership), like for the TClust approach. Considering the performance of the previously employed methods, it seems that small groups are extremely difficult to find.

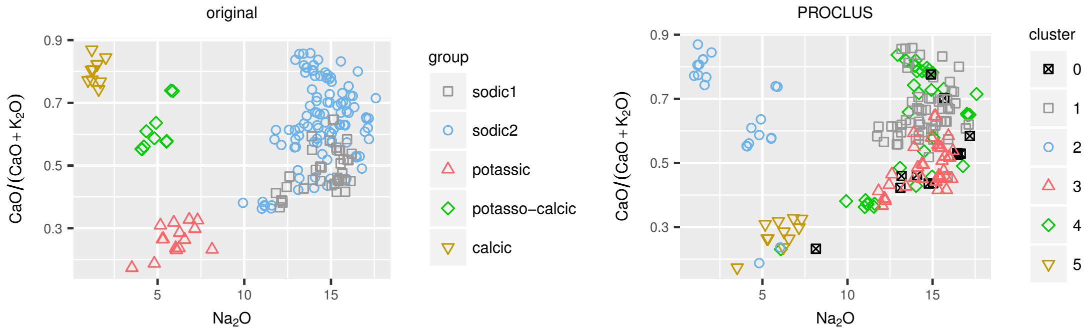

Figure 1.7: The original (true) group structure of the glass vessels data and the clustering solution obtained by the PROCLUS algorithm.

1.1.7 Evaluation measures

Just like in the case of any other statistical task (e.g. classification, outlier detection), it is also necessary to evaluate the quality of the obtained solution in data clustering. In the case of clustering, the quality of a result is assessed using either external or internal clustering evaluation indices, depending on whether or not the true group labels are known (Aggarwal and Reddy, 2013). While external indices measure to what degree the true group labels match the discovered cluster labels, internal indices validate to what extent the observations from one cluster are close to each other and distant from other clusters.

External evaluation indices

There are numerous external validity indices. Widely used indices are, for example, Purity (P) (Zhao and Karypis, 2002), F-measure (F), V-measure (Rosenberg and Hirschberg, 2007) incorporating homogeneity (H) and completeness (C) scores, and the Rand index (RI) (Hubert and Arabie, 1985). Purity measures to what degree the detected clusters consist of observations from a single group (i.e. homogeneity of a clustering solution). However, it does not reflect whether all observations from one group are assigned to a single cluster (i.e. completeness of a clustering solution). Therefore, the purity of the detected clusters gets higher values with an increasing number of clusters. In particular, the purity is the highest (i.e. 1) when each observation builds its own cluster. In contrast, the F-measure is independent of the number of clusters. It combines the concepts of precision and recall from Information Retrieval: precision measuring how homogeneous the clusters are and recall evaluating completeness of the clustering solution. Similarly, the entropy-based V-measure does not depend on the number of clusters, since it represents the harmonic mean of homogeneity and completeness scores. Finally, the Rand index is based on counting pairs of observations which agree or disagree in terms of true group labels and discovered cluster labels. All indices range between zero and one. Low value corresponds to a poor clustering solution, while large value indicates a clustering solution of high quality. Alternately, the misclassification rate (CER) can be used to evaluate the quality of a clustering solution. The measure is defined as 1-RI, therefore low values are preferable, indicating high performance.

de Souto et al. (2012) investigated several measures in the context of imbalanced groups, i.e. the group sizes are significantly different, and found out that most indices do not account for the so called class (group) size imbalance effect, i.e. when an evaluation index should reflect whether or not an object that belongs to a large (or small) group is assigned to a small (or large) cluster in an imbalanced data setting. Hence, de Souto et al. (2012) introduced a modification of the RI. However, the authors suggest excluding clusters of size one from the calculation of the RI, which can be an inappropriate solution, especially if the underlying data set contains groups of size one. Similarly, Moreno and Dias (2015) presented a modification of the B-cubed index (Bagga and Baldwin, 1998) for an imbalanced data set. Nevertheless, the adapted B-cubed index might not be suitable for highly imbalanced scenarios, since it does not even consider clusters of size two for clustering evaluation. In addition, the proposed modifications evaluate a clustering solution as a whole. In contrast, it would be more interesting to use such an index that can reflect the ability of a clustering method to correctly detect small groups. In some applications, e.g. in the media domain, groups of smaller sizes are often of high interest, as it is hard to find them. Additionally, such small groups often contain observations typically different from those located in large groups with common content. One possibility to perform such an evaluation is to employ a modification of the F-measure.

Let $\{K_r | r = 1, \ldots, k\}$ be a set containing the indices of observations assigned to the r^{th} cluster by a clustering method, where k denotes the number of clusters, and $\{G_t | t = 1, \ldots, g\}$ the set consisting of indices of observations from the t^{th} true underlying group (also called class), where g corresponds to the true number of groups. We denote n_{tr} as the number of common observations in a group G_t and a cluster K_r, and $|K_r|$ ($|G_t|$) as the number of observations in a cluster K_r (in a group G_t). The original F-measure is defined as:

$$F = \sum_{t=1}^{g} \frac{|G_t|}{n} \max_r \{F(G_t, K_r)\}, \quad F(G_t, K_r) = \frac{2\, Re(G_t, K_r)\, Pr(G_t, K_r)}{Re(G_t, K_r) + Pr(G_t, K_r)} \tag{1.3}$$

where $Re(G_t, K_r) = n_{tr}/|G_t|$ corresponds to the recall, measuring the proportion of all observations from one group assigned to a single cluster (similar to completeness) and $Pr(G_t, K_r) = n_{tr}/|K_r|$ corresponds to the precision, assessing to what degree a cluster contains observations from a single group (similar to homogeneity and purity). The weighted F-measure, wF, weights $F(G_t, K_r)$ according to the size of the true groups (instead of taking the maxima) in order to account for varying size distribution:

$$wF = \sum_{t=1}^{g} \frac{|G_t|}{n} \sum_{r=1}^{k} w_r F(G_t, K_r), \qquad w_r = \frac{n_{tr}}{|G_t|}. \tag{1.4}$$

The measure particularly allows evaluation of the ability of a clustering algorithm to detect small and big groups separately. Let T^s (T^b) be the index set of (true) groups being of smaller (bigger) sizes and R^s (R^b) the index set of detected clusters containing observations from smaller (bigger) groups. The weighted F-measure with respect to small groups is defined as:

$$wF^s = \sum_{t \in T^s} \frac{|G_t|}{n} \sum_{r \in R^s} w_r F(G_t, K_r), \qquad w_r = \frac{n_{tr}}{|G_t|}, \quad t \in T^s, r \in R^s. \tag{1.5}$$

With respect to bigger groups, the weighted F-measure is calculated in the same way, employing the index sets T^b and R^b. The same procedure can easily be applied in the case of both precision and recall. For example, the weighted precision measuring to what degree the small clusters contain observations from small groups is defined as:

$$wPr^s = \sum_{t \in T^s} \frac{|G_t|}{n} \sum_{r \in R^s} w_r Pr(G_t, K_r), \qquad w_r = \frac{n_{tr}}{|G_t|}, \quad t \in T^s, r \in R^s. \tag{1.6}$$

Similarly, the weighted recall assessing to what degree all observations from one small group are assigned to a single cluster is defined as:

$$wRe^s = \sum_{t \in T^s} \frac{|G_t|}{n} \sum_{r \in R^s} w_r Re(G_t, K_r), \qquad w_r = \frac{n_{tr}}{|G_t|}, \quad t \in T^s, r \in R^s. \tag{1.7}$$

The weighted measures range between zero and one with higher values indicating a good clustering result and lower values corresponding to a poor clustering solution.

Internal evaluation indices

Internal validity indices play an essential role in data clustering, as they evaluate the quality of clustering methods when there is no prior knowledge available. Hence, they are often employed to estimate optimal parameters, e.g. the number of clusters. Usually, a clustering approach is performed for a predefined range of a parameter. Then, the optimal choice of the parameter is selected based on minimizing or maximizing the predefined criterion. The general aim of the indices is to evaluate the resulting clusters in terms of two measures: compactness and separation, often based on pairwise distances or density-based measures. While compactness measures how close (related) the observations in a cluster are, separation evaluates how far the observations are from two different clusters. For example, the Silhouette index (Rousseeuw, 1987), as one of the most popular indices (Wiwie et al., 2015), determines the compactness of a cluster using the average within-cluster distances, whereas the separation of two clusters is based on the average between-cluster distances. In contrast, the Davies Bouldin (Davies and Bouldin, 1979) index employs the average distances between observations and their respective cluster centers.

Although a large number of further indices has been developed, the most appropriate index has not been introduced yet (Aggarwal and Reddy, 2013). Nevertheless, several comparisons of internal indices have been conducted. Such comparisons can be very helpful for selecting a proper index in practice. For example, Stegmayer et al. (2012) and Wiwie et al. (2015) compared the performance of common indices in the context of biomedical data, whereas Aggarwal and Reddy (2013) investigated 12 widely used indices on synthetic data considered with arbitrary shaped clusters or clusters of different densities and sizes.

1.2 Outlier detection in high-dimensional data space

Outlier detection is another essential unsupervised statistical and data mining technique that aims at identifying such observations that considerably deviate from the remaining data. These atypical observations – often called outliers – are very likely to appear in any real data. Depending on the situation, outliers can either represent unwanted errors which can be harmful for data clustering (see Section 1.1) as well as for other data analyses, or they can be a valuable source of information due to their atypical behavior.

To some extent, identification of outliers is related to data clustering. On the one hand, an outlier can be seen as a cluster of its own, several outliers can even form clusters of very small sizes. Therefore, many concepts employed in data clustering can also be found in outlier analysis. On the other hand, outlier detection can be recast to the task of grouping data into two groups: outliers and non-outliers. While outliers represent a class of observations with atypical behavior, the non-outliers form a class of observations with content which is typical for the analyzed data.

Outlier detection gets considerably difficult in a high-dimensional space of possible low sample size, since data become sparse and the traditional distance measures lose their ability to distinguish between outliers and non-outliers. Hence, assuming that there are not many noise variables, PCA-based outlier detection techniques may improve the effectiveness of distance measures. In the case of a large proportion of noise variables, some of the last principal components may already explain the variability of the noise part. This could lead to unreliable outlier detection results. Therefore, if a large subset of irrelevant variables for revealing outliers is present, a better approach is to employ subspace-based outlier detection which can exclude these noise variables. Moreover, in the presence of multiple group structure, most PCA-based approaches are expected to fail, as they often assume a single group structure. In the following section, the description of several methods is provided.

1.2.1 Detection of outliers for a simple data structure using PCA

PCA is a well-studied and widely employed approach for dimensionality reduction (see Jolliffe, 2002). The column-centered data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ is transformed into a subspace defining a new coordinate system $\mathbf{T} = \mathbf{XP} + \mathbf{E}$, with the $p \times k$ loading matrix $\mathbf{P}$, the $n \times k$ scores matrix $\mathbf{T}$, and the error matrix $\mathbf{E}$. The j-th column of $\mathbf{P}$ is chosen in such a way that the variance λ_j of the corresponding j-th column of $\mathbf{T}$ is maximized, subject to orthogonality to the previous columns of $\mathbf{P}$. The dimensionality of the newly constructed subspace is determined by a pre-specified number $k \leq \min\{n, p\}$ of PCs, i.e the first k score vectors.

ROBPCA

Hubert et al. (2005) introduced the robust PCA method ROBPCA and two different distances for measuring the outlyingness of observations. While the score distance SD measures outlyigness of an observation with respect to the center of the data in the robustly estimated PC space, the orthogonal distance OD represents the distance of the observations to the constructed PC space. The two distances are formally defined as

$$SD_i^{(k)} = \sqrt{\sum_{j=1}^{k} \frac{t_{ij}^2}{\lambda_j}}, \quad OD_i^{(k)} = ||\mathbf{x}_i - \mathbf{P}\mathbf{t}_i||, \tag{1.8}$$

for $i = 1, \ldots, n$, where $\mathbf{t}_i = (t_{i1}, \ldots, t_{ik})^\top$ are the score vectors in the PC space. In order to decide whether or not an observation is an outlier, two corresponding thresholds are derived using the χ^2 distribution and the normal distribution for SD and OD respectively (see Chapter 2 for more details). If either threshold is exceeded, the respective observation is classified as an outlier.

PCOut

Filzmoser et al. (2008) have developed a method for outlier identification in high-dimensions, called PCOut. The method robustly estimates PCs which are then weighted with respect to their kurtosis in order to calculate the Euclidean norms of observations in the weighted PC subspace. The norms represent Mahalanobis distances in the original data space and are subsequently transformed into the weights reflecting how much an observation is a location outlier. The weights for identifying scatter outliers are derived in the same way but without taking the kurtosis into account. Filzmoser et al. (2008) finally combine both weights in order to classify the observations with low weight (smaller than 0.25) as an outlier.

Illustrative data example

To illustrate both described methods, a simple data set of 200 observations following a multivariate normal distribution is simulated. The data set is described by 20 informative variables to which 30 noise variables are added. Additionally, 10% of the observations is contaminated by outliers corresponding to replacing them by some uniformly distributed values. Figure 1.8 visualizes the data structure using 2 PCs and shows outliers and non-outliers in different colors and symbols. In addition, the results of the two PCA-based methods are displayed as well. While the result of ROPBCA is visualized by means of score and orthogonal distances with their respective thresholds (two dashed lines), the result obtained by PCOUT is represented in terms of the final weights and the cut-off value 0.25. Although the group of outliers seems to be distinct from the non-outliers, both methods misclassify several non-outliers as outliers. Nevertheless, all outliers are easily detectable for both approaches even if the proportion of noise variables is quite high. Chapter 2 provides a comprehensive comparison of further existing robust PCA methods for the purpose of outlier detection.

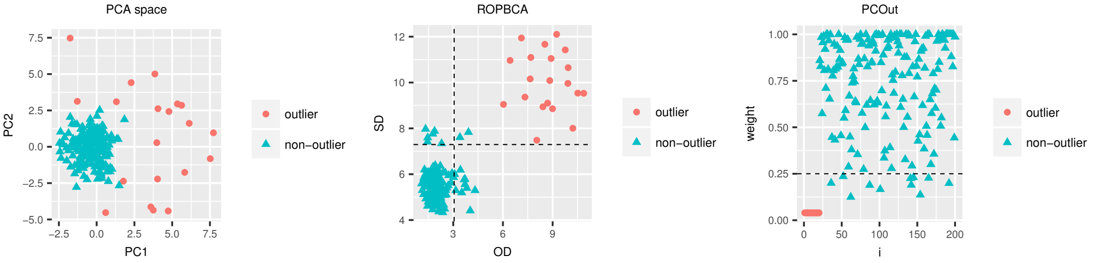

Figure 1.8: Illustrative data example displayed in PC space, and results from outlier detection by ROBPCA and PCOUT. The lines correspond to the cut-off values for classifying observations as outliers.

1.2.2 Data mining outlier detection for multiple group data structure

Data mining techniques are often designed to detect outliers in a data set exhibiting multiple group structure. Furthermore, they commonly result in an outlying score and do not provide any threshold resulting in binary labels: outliers and non-outliers. The approach discussed in this section assesses the outlyingness of observations with respect to their locality and therefore such techniques are based on the nearest neighbors.

LOF

Although the LOF (Local Outlier Factor) (Breunig et al., 2000) determines the degree of outlyingness of an observation with respect to its neighborhood in all dimensions, Zimek et al. (2012) demonstrated that LOF achieves promising results if the number of informative variables in the high-dimensional data space is not too low. Since Chapter 3 and 4 introduce clustering approaches incorporating LOF, the method is described in more detail.

The general idea of LOF is to compare the density of each observation with the densities of their respective nearest neighbors. If the difference between the densities is small, the observation is located in a homogeneous region (i.e. in a cluster) and its corresponding factor is approximately one. In contrast, big difference between the densities indicates that the observation is distant from its neighbors and, therefore, its factor gets considerably higher. Formally, Breunig et al. (2000) first determined the q-neighborhood of each observation $\mathbf{x}_i \in \mathbb{R}^p, i = 1, \ldots, n$, which is defined as

$$N_q(\mathbf{x}_i) = \{\mathbf{x} \in \mathbf{X} | d(\mathbf{x}_i, \mathbf{x}) \leq d_q(\mathbf{x}_i)\}, \tag{1.9}$$

where $d_q(\mathbf{x}_i) = d(\mathbf{x}_i, \mathbf{x}_q^i)$ denotes the Euclidean distance of $\mathbf{x}_i$ to its q^{th} nearest neighbor, $\mathbf{x}_q^i$, and is called *q-distance* of $\mathbf{x}_i$. Let $|N_q(\mathbf{x}_i)|$ be the number of observations contained in $N_q(\mathbf{x}_i)$. Note that this number can be more than q in case of ties. Next, the reachability distance of $\mathbf{x}_i$ with respect to its neighbor $\mathbf{x}$, $d_{reach_q}(\mathbf{x}_i, \mathbf{x})$, determines at which distance $\mathbf{x}_i$ is reachable from its neighboring observations $\mathbf{x}$ and is defined as:

$$d_{reach_q}(\mathbf{x}_i, \mathbf{x}) = \max\{d_q(\mathbf{x}), d(\mathbf{x}_i, \mathbf{x})\}. \tag{1.10}$$

Figure 1.9 visualizes the concept of reachability distance of $\mathbf{x}_i$ with respect to one of its neighbors $\mathbf{x}$ for $q = 3$ for two different situations (Breunig et al., 2000). The three closest neighbors of $\mathbf{x}_i$ are located in the circle around $\mathbf{x}_i$, including $\mathbf{x}$ as well. The radius of the circle around $\mathbf{x}$ corresponds to the $d_3(\mathbf{x})$ and the arrow indicates $d_{reach3}(\mathbf{x}_i, \mathbf{x})$. In the first scenario, Figure 1.9 (left), the observation $\mathbf{x}_i$ is far from its neighbor $\mathbf{x}$. Therefore, the reachability distance of $\mathbf{x}_i$ from its neighbor $\mathbf{x}$ is set to the true distance between them. In contrast, in the second scenario, if the observation $\mathbf{x}_i$ is close enough to $\mathbf{x}$, the reachability distance of $\mathbf{x}_i$ from $\mathbf{x}$ is set to the q-distance of $\mathbf{x}$, $d_3(\mathbf{x})$, see Figure 1.9 (right). In the next step, the local (reachability) density of an object $\mathbf{x}_i$ is calculated,

$$lrd_q(\mathbf{x}_i) = \frac{|N_q(\mathbf{x}_i)|}{\sum_{\mathbf{x} \in N_q(\mathbf{x}_i)} d_{reach_q}(\mathbf{x}_i, \mathbf{x})}. \tag{1.11}$$

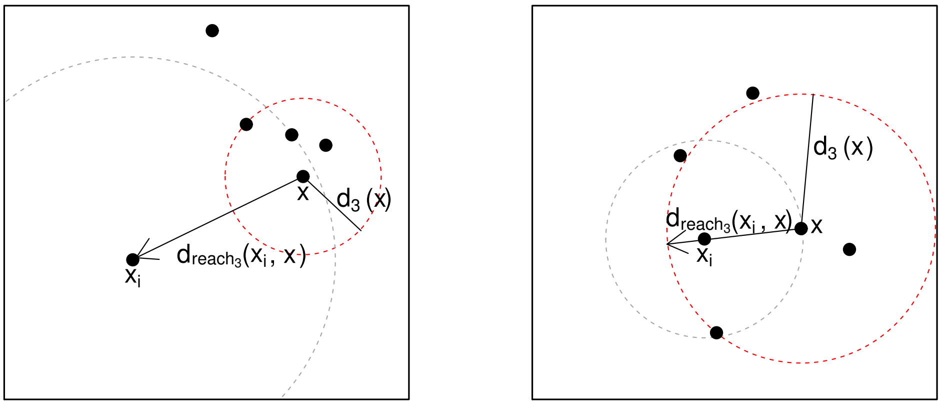

Figure 1.9: Two different situations for the reachability distance of $\mathbf{x}_i$ w.r.t one of its neighbors $\mathbf{x}$ for q=3. The reachability distance of $\mathbf{x}_i$ w.r.t. $\mathbf{x}$ is represented by an arrow, the radius of the red circle around $\mathbf{x}$ corresponds to the q-distances of $\mathbf{x}$, $d_3(\mathbf{x})$, and the three closest neighbors of $\mathbf{x}_i$ are located in the gray circle around $\mathbf{x}_i$. The reachability distance of $\mathbf{x}_i$ w.r.t. $\mathbf{x}$ is set to the true distance between them (left). The reachability distance of $\mathbf{x}_i$ w.r.t. $\mathbf{x}$ is set to $d_3(\mathbf{x})$ (right).

The larger the distance between $\mathbf{x}_i$ and its neighbors, the lower the local density of $\mathbf{x}_i$. Finally, the LOF score (Breunig et al., 2000) is computed as the ratio of local densities of the q neighbors and the local density of $\mathbf{x}_i$ normalized by the q-neighborhood size,

$$LOF_q(\mathbf{x}_i) = \frac{1}{N_q(\mathbf{x}_i)} \sum_{\mathbf{x} \in N_q(\mathbf{x}_i)} \frac{lrd_q(\mathbf{x})}{lrd_q(\mathbf{x}_i)}. \tag{1.12}$$

If $\mathbf{x}_i$ belongs to a cluster, $LOF(\mathbf{x}_i) \approx 1$ since $\mathbf{x}_i$ has a similar local density as the local density of its neighbors. In contrast, if $\mathbf{x}_i$ deviates from a cluster, i.e. if $\mathbf{x}_i$ is a local outlier, $LOF(\mathbf{x}_i) >> 1$, since the local density of $\mathbf{x}_i$ is much lower than the local densities of its q neighboring observations (Breunig et al., 2000).

As the LOF is the first local approach for outlier detection, it has inspired many researchers to extend or modify the method (Campos et al., 2016). For example, LOCI (Papadimitriou et al., 2003) defines the neighborhood based on the observations located within an ϵ range, so called ϵ-neighborhood. Since the method considers various values of ϵ, it is less sensitive to input parameters. In contrast, INFLO (Jin et al., 2006) employs the q reverse nearest neighbors. Other local approaches can be found in the study by Schubert et al. (2014).

Subspace Outlier Detection (SOD)

SOD (Kriegel et al., 2009a) aims at identifying outliers in axis-parallel subspaces of high-dimensional data. In general, the approach evaluates to what extent each observation fits its relevant subspaces spanned by its h shared nearest neighbors selected from q nearest neighbors, thus $h \geq q$. For each observation, a relevant subspace is defined by the variables which exhibit low variance, taking h neighbors into account. Consequently, the subspace outlier degree of an observation is defined as the Euclidean distance of the observation to the reference subspace normalized by the dimensionality of the selected subspace. While a value close to 0 indicates that an observation is a non-outlier, much higher values (up to 1) suggest that observations are outliers. Another approach called feature bagging (Lazarevic and Kumar, 2005) randomly selects a predefined number of variables to create a subspace in which LOF (or an alternative outlier detection procedure) is subsequently applied. This is done several times and finally, the resulting values of LOF are combined. Further approaches operating the axis-parallel subspaces can be found in (Aggarwal, 2013).

Correlation Outlier Probability (COP)

In contrast to identifying outliers in axis-parallel subspaces, COP (Kriegel et al., 2012) detects outliers in arbitrarily oriented subspaces. The approach appears to be the first method considering local correlations of variables by constructing the PC space in a local fashion. For each observation, the q nearest neighbors are determined in order to estimate the local covariance structure for the following construction of PC components. Only the last components are used to calculate the probability of an observation to be an outlier with respect to a constructed hyperplane. A value close to 1 indicates that an observation is an outlier, in contrast to a value close to 0 suggesting a non-outlier. It should be noted that the input parameter q needs to be 3 times larger than the dimensionality of the data set. Hence, the method cannot be applied on flat data (i.e $n << p$). A comprehensive discussion on outlier detection in generalized subspaces is provided by Aggarwal (2013).

Illustrative data example

To show the performance of the outlier detection algorithms LOF, SOD, and COP, one additional group is added to the data example considered in Section 1.2.1. The added group has a different location, as visualized in the PC space in Figure 1.10. The results of LOF are obtained using the R implementation in `dbscan`(Hahsler and Piekenbrock, 2017), considering $q = 10$. The R code for SOD is available in `HighDimOut` (Fan, 2015), and the provided solution is achieved with $q = 50$ and $h = 40$. The method COP is implemented in ELKI, freely available at `http://elki.dbs.ifi.lmu.de/`, and the result corresponds to the setting $q = 300$.

While most outliers receive considerably higher score than non-outliers (i.e. observations forming groups) within SOD and LOF, this is not the case for the scores obtained by COP. It seems that some non-outliers could be wrongly considered to be outliers, indicated by their high COP scores. Although LOF takes all variables into account, the method assigns the highest scores to all outliers. Of course, the results might slightly change with different choices of the parameters. Nevertheless, the parameters are supposed to be set in such a way that the differences between outliers and non-outliers are most evident.

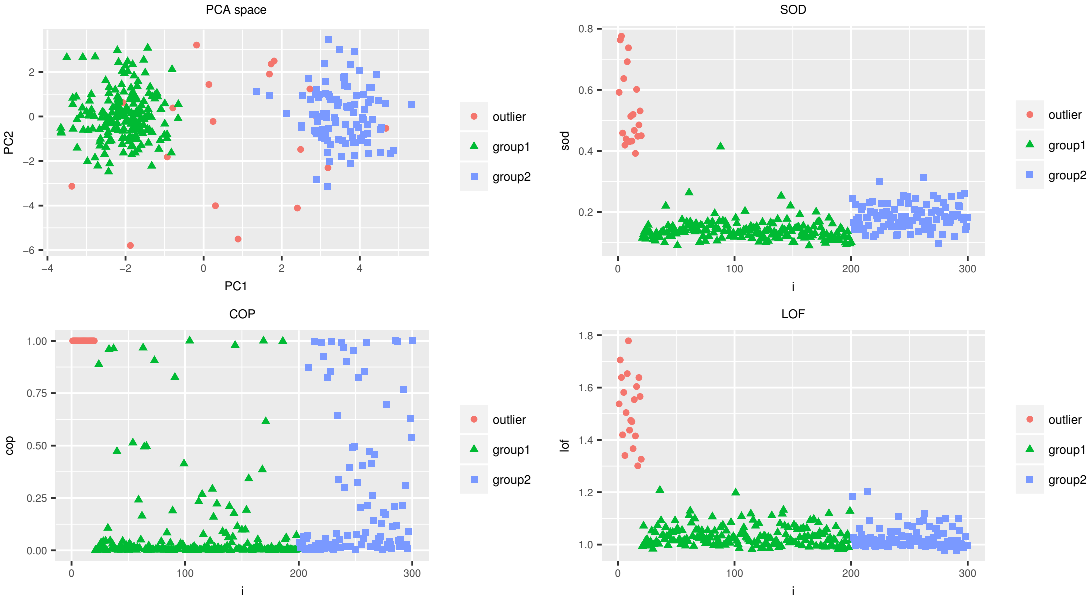

Figure 1.10: Illustrative date example containing 2 groups as well as outliers displayed in PC space, and the achieved outlier scores by LOF, SOD, and COP. The true group and outlier membership is represented by different colors and symbols.

1.3 Outline of the thesis

This thesis introduces new robust clustering methods that are compared to existing clustering approaches. In addition, several existing robust PCA-based outlier detection methods are evaluated in the thesis. The introductory chapter provides an overview of existing techniques for data clustering that attempt to address various clustering challenges. The second chapter aims at investigating the performance of existing PCA methods used to detect unusual sounds in a large audio collection. The third chapter presents a clustering approach developed to discover highly imbalanced groups in high-dimensional data. The last chapter introduces a robust and sparse k-means algorithm which tries to detect clusters, outliers, and informative variables simultaneously in high-dimensional data. All developed procedures and graphics were created with the software R (R Core Team, 2016).

Chapter 2 provides a profound evaluation of various PCA algorithms for detecting unusual instances in high-dimensional audio data. The supervised framework for PCA-based outlier detection is introduced for the following evaluation with respect to various proportions of outliers, the presence of subgroups, and the amount of available information in the form of true class labels.

Brodinová Š, Ortner T, Filzmoser P, Zaharieva M, Breiteneder C (2016). Evaluation of robust PCA for supervised audio outlier detection. In: A. Colubi, A. Blanco, C. Gatu, and G. Gonzalez-Rodriguez, editors, *Proceeding of the 22nd International Conference on Computational Statistics (COMPSTAT)*, pp. 183-194.

Chapter 3 introduces a clustering approach designed to discover highly imbalanced media groups in high-dimensional data with the special focus on mining very small groups containing potentially interesting information. The introduced procedure incorporates an existing conventional clustering approach in order to find a large number of homogeneous initial clusters, which are then successively merged into a smaller number of final clusters. To merge a pair of clusters, LOF is employed for evaluating whether two close initial clusters share the same local densities and thus need to be merged. A thorough empirical study demonstrates the advantage of identifying the group structure over existing clustering methods.

Brodinová Š, Zaharieva M, Filzmoser P, Ortner T, Breiteneder C (2017). Clustering of imbalanced high-dimensional media data. *Advances in Data Analysis and Classification.* To appear.

Chapter 4 presents a k-means based algorithm developed to identify the group structure in high-dimensional contaminated data, possibly containing a large number of noise variables. The concept of assigning weights to observations and variables is incorporated during k-means clustering. While observation weights are used to downweight the effect of outlying observations, the variables weights are employed to reflect their contribution to group separation. The procedure for estimating the parameters is eventually introduced and tested on a variety of simulated data sets. Comparisons with existing k-means based approaches show a great ability to discover clusters in real high-dimensional data.

Brodinová Š, Filzmoser P, Ortner T, Breiteneder C, Zaharieva M (2017). Robust and sparse k-means clustering for high-dimensional data. Submitted for publication.

Chapter 5 presents functionality of two R packages: `IClust` and `wrsk`. The packages implement robust clustering methods presented in Chapter 3 and 4.

Unpublished.

CHAPTER 2

Evaluation of robust PCA for supervised audio outlier detection

Abstract: Outliers often reveal crucial information about the underlying data such as the presence of unusual observations that require for in-depth analysis. The detection of outliers is especially challenging in real-world application scenarios dealing with high-dimensional and flat data bearing different subpopulations of potentially varying data distributions. In the context of high-dimensional data, PCA-based methods are commonly applied in order to reduce dimensionality and to reveal outliers. Thus, a thorough empirical evaluation of various PCA-based methods for the detection of outliers in a challenging audio data set is provided. The various experimental data settings are motivated by the requirements of real-world scenarios, such as varying number of outliers, available training data, and data characteristics in terms of potential subpopulations.

Key words: Outlier detection; Robust PCA; Audio data; Experiments

Co-authors: Thomas Ortner, Peter Filzmoser, Maia Zaharieva, Christian Breitender

2.1 Introduction

Outlier identification is an essential data mining task. Outliers do not only contaminate distributions and, thus, estimations based on the distributions, moreover, they often are the prime focus of attention. In many fields outliers carry significant, even crucial information for applications such as fraud detection, surveillance, and medical imaging. In this paper, we employ outlier detection in an automated highlight detection application for audio data. This is a first step towards the identification of key scenes in videos, where the audio is a fundamental component.

Outlier detection gets considerably more difficult in a high-dimensional space or when there are less observations than variables available (flat data). In a high-dimensional space, data becomes sparse and distances between observations differ very little. The situation becomes even more complex when groups of outliers are present due to the emerging masking effect (Becker and Gather, 1999). To justify the application of distance-based similarity measures in such a situation, the reduction of dimensionality is an inevitable course of action. A well-established approach for this purpose is the use of principal component analysis (PCA), which transforms the original variables to a smaller set of uncorrelated variables keeping as much of the total variance as possible (Jolliffe, 2002). This step removes the curse of high dimensionality for this subspace. Nevertheless, it has been shown, that even though in theory distance functions loose their meaningfulness in high dimensionality, the orthogonal complement of the principal component (PC) space might still hold crucial differences in the distance and, thus, important information for outlier detection (Zimek et al., 2012).

The focus of this paper is the thorough empirical comparison of PCA-based methods for high-dimensional and flat data, that are suitable for outlier detection in audio data. We compare classical PCA with its robust versions in terms of sensitivity regarding changes in the setup such as the percentage of outliers and the size or the distribution of the data sets. A crucial aspect in this context is the proper choice of number of components used for the construction of the PC space. We propose to manually label a small number of observations and to use those labels to estimate the best possible number of PCs without any prior knowledge of the data structure. This concept creates a reasonable situation for real-world applications. Thus, an estimation for the optimal number of components is performed throughout all the experiments including an analysis regarding the number of pre-labeled observations itself. Furthermore, we outline an approach for the optimization of critical values used for outlier detection by employing the additional information from the labeled observations, which can greatly increase the robustness of the outlier detection towards the number of chosen components.

2.2 Related work

Several authors perform simulation studies to explore the performance of the classical and various robust PCA-based methods in different scenarios in the context of outlier detection, such as varying degree of data contamination, data dimensionality, and missing data, e.g. (Pascoal et al., 2010)(Sapra, 2010)(Serneels and Verdonck, 2008)(Xu et al., 2013). For example, Pascoal et al. (2010) compare the classical PCA approach (Jolliffe, 2002) with five robust methods: spherical PCA (Locantore et al., 1999), two projections pursuit techniques (Croux et al., 2007)(Croux and Ruiz-Gazen, 2005), and the ROBPCA approach (Hubert et al., 2005) in different contamination schemes. The results show that ROBPCA outperforms the compared methods in terms of estimated recall. Similarly, Sapra (2010) shows that a robust PCA approach based on projection pursuit (Filzmoser et al., 2006) outperforms the classical PCA even for data sets with more variables than observations. In a recent simulation study, Xu et al. (2013) show that for the generated data settings the performance of ROBPCA and techniques based on projection pursuit degrades substantially in terms of expressed variance as the dimensionality of the data increases. However, the authors only consider the first few principal components and focus on a data setting where the observations and the variables are of the same magnitude. Usually, simulation studies are performed for very specific data settings, e.g. all observations/variables follow a predefined distribution. However, real data have more complex data structures than synthetic data and, thus, outlier detection on real data is even more challenging. Current evaluations on real data sets are often limited by the number of available data. As a result, a thorough investigation of different outlier detection methods for various data settings is barely feasible. For example, Sapra (2010) performs an evaluation on a small set of financial data with 120 observations. Hubert et al. (2005) report evaluations on three low-sampled real data sets with varying dimensionality. While evaluations on multiple data sets provide an estimation of the robustness of the investigated approaches, no general conclusions about the sensitivity to specific data aspects can be made. Experiments with larger real data sets are commonly tailored to the evaluation of the performance of outlier detection methods for a particular data without any variation of the experimental settings, e.g. (Filzmoser et al., 2008)(Shyu et al., 2003). In contrast, we employ a large real data set in the simulation of different experimental settings and perform a thorough evaluation of the sensitivity of the explored approaches with respect to varying data aspects.

2.3 Evaluation setup

2.3.1 Compared approaches

In general, algorithms for estimating the PC space are based on either eigenvector decomposition of the empirical covariance matrix, singular value decomposition (SVD) of the (mean-centered) data matrix, or on projection-pursuit (PP) technique. We compare several approaches including both classically and robustly estimated PCs which are suitable for high-dimensional flat data. PCA-based outlier detection can be employed using two different distances for each observation derived from the PC space Hubert et al. (2005): score distance, SD, and orthogonal distance, OD:

$$SD_i^{(k)} = \sqrt{\sum_{j=1}^{k} \frac{t_{ij}^2}{\lambda_j}}, \quad OD_i^{(k)} = ||\mathbf{x}_i - \mathbf{P}\mathbf{t}_i||, \quad i = 1, \dots, n, \tag{2.1}$$

where $\mathbf{t}_i = (t_{i1}, \dots, t_{ik})^\top$ are the score vectors in the PC space, λ_j denotes the variance of the corresponding j^{th} column of the score matrix $\mathbf{T} = \{t_{ij}, j = 1, \dots, k\}$, $\mathbf{P}$ represents the

$p \times k$ loading matrix, $\mathbf{x}_i$ is the ith observation of the data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, and the index k refers to the number of PCs. While SD represents the distance of observations in the estimated subspace to the center of data, OD measures the distance of the observations to the subspace. Two thresholds are used to detect outliers. For the SD, the 97.5% quantile of the χ^2 distribution with k degrees of freedom, i.e. $c_{SD}^{(k)} = (\chi^2_{k,0.975})^{1/2}$, and for the OD, 97.5% quantile of the standard normal distribution, $c_{OD}^{(k)} = (\hat{\mu} + \hat{\sigma} z_{0.975})^{3/2}$, can be taken as the critical values. The estimation of $\hat{\mu}$ (resp. $\hat{\sigma}$) can be obtained using the median (resp. MAD) of the values of $OD_i^{2/3}$ (see (Hubert et al., 2005) for more details). If either threshold is exceeded, the respective observation is classified as an outlier.

clPCA: Classical (non-robust) PCA (Jolliffe, 2002) for flat data is performed by means of SVD which is directly related to eigenvalue decomposition of the classical empirical covariance Wall et al. (2003). The columns of the loading matrix $\mathbf{P}$ are the right singular vectors and the variance λ_j corresponding to the j-th singular value. However, the classical covariance is sensitive to outliers (Hubert et al., 2005) and the resulting PCs do not describe the true data structure.

OGK PCA (Maronna and Zamar, 2002) is a PCA-based approach using robust covariance matrix estimation. The method starts by robustly scaling the data, $\mathbf{Y} = \mathbf{X}\mathbf{D}^{-1}$, where $\mathbf{D} = diag\{\hat{\sigma}(X_1) \dots \hat{\sigma}(X_p)\}$ is the robustly estimated univariate dispersion of each column X_j of the data matrix $\mathbf{X}$, and $\hat{\sigma}$ is computed by using τ-estimation of univariate dispersion. Next, the Gnanadesikan-Kettenring estimator (Gnanadesikan and Kattenring, 1972) is computed for all variable pairs of $\mathbf{Y}$ resulting in a robust correlation matrix, $\mathbf{U}$, where $U_{jk} = cov(Y_j, Y_k)$, $j, k = 1, \dots, p$. The eigenvector decomposition of the correlation matrix $\mathbf{U} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^\top$ allows for the projection of the data onto the directions of the eigenvectors, $\mathbf{Z} = \mathbf{Y}\mathbf{E}$. Finally, the covariance matrix is transformed back to the original space, $\mathbf{S_X} = \mathbf{D}\mathbf{E}\mathbf{L}\mathbf{E}^\top\mathbf{D}^\top$, where $\mathbf{L} = diag\{\hat{\sigma}(Z_1) \dots \hat{\sigma}(Z_p)\}$ and $\mathbf{DE}$ is the loading matrix of p orthogonal eigenvectors of dimension k and corresponds to the direction of the principal components.

GRID PCA (Croux et al., 2007) is a robust PCA approach using the GRID search algorithm. It employs the PP method to project the data on a direction which maximizes the robust variance of the projected data (Li and Chen, 1985). GRID first sorts the variables in decreasing order according to the robust dispersion. The first projection direction is found in the plane spanned by the first two sorted variables and it passes through the robust center and a grid point. The remaining variables successively enter the search plane to obtain the first optimal direction. The algorithm searches the subsequent directions in a similar way by imposing orthogonality until there is no improvement in maximizing the robust variance.

ROBPCA Hubert et al. (2005) combines robust PP techniques (Li and Chen, 1985) with robust covariance estimation. First, the data space is reduced to an affine subspace using a SVD (Hubert et al., 2002). In the next step the least outlying observations are identified using the univariate Minimum Covariance Determinant (MCD) location and scale estimator (Rousseeuw, 1984). The covariance matrix, $\mathbf{S}_0$, of the least outlying points is subsequently used to select a number of components k and to project the data on the subspace determined by the first k eigenvectors of $\mathbf{S}_0$. The FAST-MCD algorithm (Rousseew and van Driessen, 1999) is employed to obtain a robust scatter matrix, $\mathbf{S} = \mathbf{PLP}^\top$, where $\mathbf{P}$ is the loading matrix of p orthogonal eigenvectors of dimension k and $\mathbf{L}$ the diagonal matrix of k eigenvalues.

PCOut (Filzmoser et al., 2008) is a method already comprising an outlier detection algorithm, in contrast to the previously described approaches. First, the observations being far away from the center of the main body of the data are identified, i.e. *location* outliers. Then, the detection of *scatter* outliers generated from a model with the same location as the main data but with a different covariance structure is conducted. Outlier detection is performed in the subspace using the robustly scaled PCs which contribute to about 99% of the total variance.

2.3.2 Performance measures

We evaluate the performance of the compared approaches in terms of true positive rate, TPR, and false positive rate, FPR:

$$TPR = TP/(TP + FN), \qquad FPR = FP/(FP + TN), \tag{2.2}$$

where TP denotes *true positives* (correctly identified outliers), FN indicates *false negatives* (outliers declared as normal observations), FP corresponds to *false positives* (normal observations declared as outliers), and TN refers to *true negatives* (correctly identified normal observations). Additionally, we calculate the area under the Receiver Operating Characteristics (ROC) curve (AUC) representing the trade-off between TPR and FPR by a single value. Figure 2.1 illustrates the construction of a ROC curve for an example evaluation A. The estimation of the corresponding AUC of A is obtained in such way that the area is divided into regular shapes and summed up which results in $AUC = 1/2\ (1 + TPR - FPR)$. Note that when the algorithm does not detect any outlier (i.e. both TPR and FPR are zero) the AUC according to the above formula is equal to 0.5. Although the two extreme scenarios (no outlier detected and random prediction) are not identical, they are both not desired output in terms of effectiveness of outlier detection approaches. It should also be noted that the number of regular observations is much higher than the number of outliers. Thus, the defined AUC measure is much more sensitive towards changes in the total number of positively identified than towards negatively identified outliers. While this looks disproportional at first, the focus of the performed evaluations is the successful detection of outliers. Therefore, in this concept the high sensitivity towards single changes in TP is a welcome side effect.

2.3.3 Data set

We employ a high-dimensional, real-world audio data set of approximately $8,700$ observations to construct different experimental settings, i.e. flat data, varying number of outliers, varying size of available training data, etc. The data set covers the three

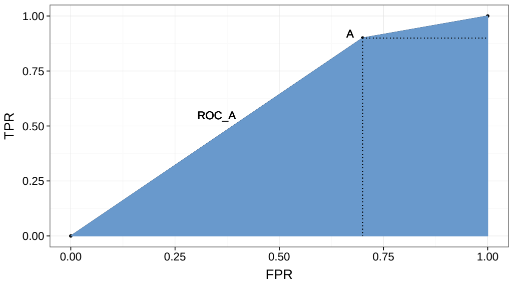

Figure 2.1: ROC curve construction.

fundamental audio types: music, speech, and environmental sounds. Each observation is represented by a set of 50 (partially) multi-dimensional features, i.e. each feature consists of one or more variables, resulting in a feature vector of 679 dimensions in total. Features were selected in order to capture a wide range of audio properties and to represent the particular qualities of the three audio types equally well. The feature set comprises features that operate in the temporal and frequency domains, e.g. features for zero crossings, amplitude or brightness, features from the MPEG7 standard, perceptional features, and various cepstral coefficients.

The observations are approximately equally distributed across the three audio types. However, the underlying data structures are strongly varying due to present subpopulations of different sizes, e.g. different genres in the music samples and different voices in the speech samples. When constructing the data sets for the experiments and for the performance evaluation of the employed approaches for outlier detection, we exploit the available labels, e.g. we define TV speech data, the largest subpopulation, as main group and select observations from environmental sounds as "outliers". This is a very challenging approach: While, usually, speech and music recordings can be easily separated by the employment of suitable features, this does not hold for environmental sounds. Environmental sounds cover a wide range of noises that sometimes have great similarities with speech data, sometimes with music and often they are just different.

The outlier detection approaches based on the two distance measures (OD and SD) employ three data sets: training, validation, and test set. The PC space spanned by k components is constructed with the observations coming from the training set. Additionally, we calculate the two critical values for the orthogonal distance, c^k_{OD}, and for the score distance, c^k_{SD}. These measures are exclusively derived from loadings and scores of the training data. Next, the observations from the validation set are projected onto the constructed PC space spanned by k PCs. An observation having an orthogonal or score distance larger than the respective critical value is declared as an outlier. This procedure is conducted with varying number of components k to select the optimal number of components, k_{opt}, in terms of maximizing AUC. The use of validation set in this context prevents potential overfitting of the estimated parameter, k_{opt}, to the characteristics of the training data. Finally, we perform an evaluation on the test data with respect to the optimal number of components k_{opt} from the validation set and the PC space spanned by k_{opt} determined by observations from the training set.

We rescale the data to make variables comparable using the mean and standard deviation of the variables in the training set. The reason for applying a non-robust scaling is the presence of many variables which are almost constant but a small proportion of values has huge deviations. The robust MAD for such variables would be very small and this would artificially increase the whole data range during the scaling. As a consequence, many of the regular observations would be made indistinguishable from real outliers. The assignment of the observations to training, validation, and test sets is done randomly and all evaluations are based on 100 replications. Since we have a larger pool of available data, independent training and validation data were constructed repeatedly. We think this is preferable over cross-validation, which would typically be used in situations where independent validation data are not available.

2.4 Experimental results

In this section we present the results of the performed experiments. We explore the sensitivity of the investigated approaches with respect to the percentage of outliers, size of training and validation sets, and data characteristics. We report results in terms of AUC, TPR, FPR, number of PCs, and the corresponding standard errors (SE) over the 100 randomly initialized replications for each experiment.

2.4.1 Sensitivity to the percentage of outliers

For this evaluation we consider TV recordings (the biggest speech subgroup) as regular observations and we randomly select observations from both environmental and music samples as outliers. We split the data equally into training, validation, and test sets, corresponding to approximately 360 regular observations per set.

In a first experiment, we calculate the PC space using only the regular observations from the training set and we consider different percentage of outliers for the validation and test sets: 2%, 5%, and 10% of the main observations (see Table 2.1). The results show that clPCA performs similar or better than the robust PCA methods, while PCOut is capable of finding only approximately half of the outliers (indicated by the low TPR). Although the performance of clPCA and its robust counterparts degrades slightly by decreasing the percentage of outliers, ROBPCA does not indicate such dependency. SE remains at a very low level during the experiments for all methods.

In a second experiment, we consider that the training set is not free of outliers in order to explore their impact on the constructed PC space. The results show that the robust PCA methods clearly outperform clPCA. PCOut performs as poorly as in the first experiment. While the number of outliers does not show any clear dependency on the resulting AUC, this is not the case for the number of PCs. GRID PCA reduces the number of selected PCs with decreasing contamination in contrast to the remaining methods. ROBPCA tends to select a considerably lower number of PCs than its counterparts. The achieved results in terms of AUC suggest that the use of robust PCA methods is recommended when there is no guarantee that the training set is free of outliers. In a real-world scenario this can not always be satisfied. Therefore, we take this into account and all further experiments consider training set containing outliers.

Table 2.1: Evaluation results for different percentage of outliers (%).

%	Method	Pure training set				Training set with outliers			
		AUC ($\mathbf{SE_{AUC}}$)	**k** ($\mathbf{SE_k}$)	**TPR** ($\mathbf{SE_{TPR}}$)	**FPR** ($\mathbf{SE_{FPR}}$)	**AUC** ($\mathbf{SE_{AUC}}$)	**k** ($\mathbf{SE_k}$)	**TPR** ($\mathbf{SE_{TPR}}$)	**FPR** ($\mathbf{SE_{FPR}}$)
10	clPCA	0.948 (0.002)	122 (1)	0.953 (0.005)	0.058 (0.003)	0.531 (0.005)	152 (16)	0.067 (0.011)	0.004 (0.001)
	GRID PCA	0.943 (0.002)	144 (2)	0.943 (0.004)	0.056 (0.002)	0.921 (0.003)	140 (1)	0.896 (0.006)	0.053 (0.001)
	ROBPCA	0.907 (0.002)	57 (2)	0.918 (0.007)	0.103 (0.004)	0.891 (0.004)	75 (3)	0.887 (0.007)	0.106 (0.005)
	OGK PCA	0.936 (0.002)	191 (4)	0.946 (0.005)	0.074 (0.003)	0.929 (0.003)	281 (4)	0.926 (0.006)	0.068 (0.002)
	PCOut	0.602 (0.006)	- (-)	0.516 (0.060)	0.311 (0.002)	0.624 (0.004)	- (-)	0.351 (0.007)	0.103 (0.002)
5	clPCA	0.946 (0.003)	136 (2)	0.949 (0.007)	0.057 (0.003)	0.557 (0.007)	205 (15)	0.124 (0.015)	0.009 (0.002)
	GRID PCA	0.942 (0.003)	163 (2)	0.937 (0.007)	0.054 (0.002)	0.933 (0.003)	152 (2)	0.923 (0.007)	0.057 (0.002)
	ROBPCA	0.916 (0.003)	51 (2)	0.929 (0.006)	0.097 (0.004)	0.918 (0.004)	54 (3)	0.928 (0.008)	0.092 (0.004)
	OGK PCA	0.931 (0.003)	168 (5)	0.931 (0.008)	0.070 (0.003)	0.925 (0.003)	203 (6)	0.918 (0.008)	0.067 (0.004)
	PCOut	0.609 (0.007)	- (-)	0.538 (0.016)	0.320 (0.006)	0.621 (0.006)	- (-)	0.359 (0.012)	0.118 (0.005)
2	clPCA	0.912 (0.007)	164 (4)	0.865 (0.016)	0.040 (0.003)	0.565 (0.009)	173 (15)	0.141 (0.019)	0.011 (0.002)
	GRID PCA	0.937 (0.007)	190 (4)	0.860 (0.015)	0.039 (0.003)	0.914 (0.006)	174 (4)	0.872 (0.013)	0.044 (0.002)
	ROBPCA	0.913 (0.005)	39 (3)	0.911 (0.010)	0.085 (0.005)	0.918 (0.005)	39 (3)	0.916 (0.011)	0.081 (0.005)
	OGK PCA	0.905 (0.007)	129 (6)	0.862 (0.015)	0.052 (0.004)	0.908 (0.007)	139 (7)	0.865 (0.016)	0.050 (0.004)
	PCOut	0.604 (0.009)	- (-)	0.531 (0.021)	0.323 (0.006)	0.615 (0.009)	- (-)	0.420 (0.022)	0.191 (0.011)

2.4.2 Sensitivity to the size of training, validation, and test sets

In this experiment, we investigate whether varying the size of training, validation, and test sets considerably influences the performance of the compared approaches. Again, we consider the biggest speech subgroup as main observations and we add 5% from the instances from music and environmental sounds as outliers. We divide the data into training, validation, and test sets according to different partitions ranging from 0.33/0.33/0.33 to 0.05/0.05/0.90 corresponding to the size of sets from 378/378/380 to 57/57/1022 observations. Note that the percentage of outliers is the same in each data set.

Figure 2.2 (left) shows the results of the evaluation in terms of AUC and number of PCs necessary to distinguish outliers from main observations. clPCA fails since the training set contains outliers. The performance of the robust PCA methods decreases with the reduction of the size of training and validation sets. GRID PCA achieves a high AUC and outperforms the remaining methods even if the size of the available training set is reduced to 170 instances. AUC falls rapidly when considering smaller data size. In contrast, ROBPCA yields still a reasonable AUC in the most extreme setting (57 observations). For a more detailed investigation, we visualize the distribution of the resulting AUC during the 100 replications for each method.

Figure 2.3 illustrates that PCOut and clPCA perform similar in each situation since the distribution of observed intervals is almost identical. This does not hold for the other three methods. The proportion of AUC ranging from 0.9 to 1 representing the results of ROBPCA decreases with the size reduction of training and validation sets. Considering

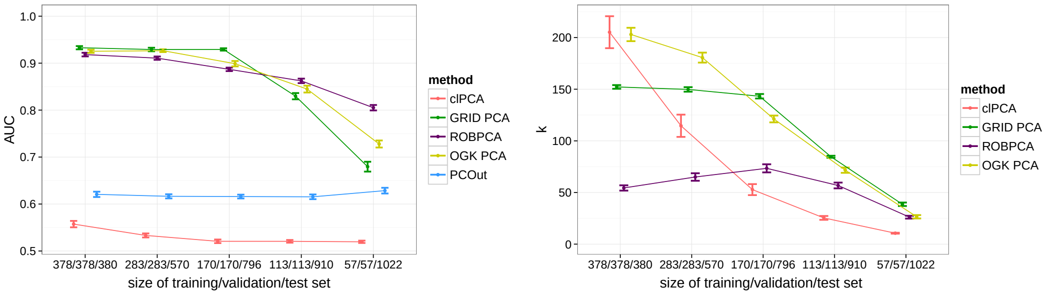

Figure 2.2: Evaluation results for varying size of training, validation, and test sets in terms of AUC (left) and the number of PCs (right).

the performance of OGK PCA, we observe that the largest proportion of AUC between 0.9 and 1 is attained when the sample size of training set is 283, and subsequently reduced size to 57 instances causes that the majority of AUC achieves the values between 0.5 and 0.8. The results of GRID PCA reveal very large proportion of AUC from the interval $(0.9, 1]$ in the first three situations. However, when the size of training and validation sets is reduced from 113 to 57, almost half of the AUC values are in the interval $(0.7, 0.4]$. Figure 2.2 (right) shows that the number of PCs selected by ROBPCA is independent from the size of the sets and it tends to choose fewer PCs while the number of components in case of the other methods is affected by decreasing the number of observations in the training and validation sets. This is given by the method itself but also by the size of the employed training set. Moreover, the number of PCs selected by clPCA deviates considerably during the replications in the first three situations. In contrast, GRID PCA indicates small SE of the selected numbers of components.

2.4.3 Sensitivity to the size of the validation set

Our last experiment employed a training set containing outliers to construct the PC space and calculate two critical values. That means, the available information about labels is required only for the validation set to select the optimal number of PCs. Additionally, the results from the experiment indicated that some of the compared approaches perform well even if the size of validation set is reduced to 170 or 57 instances. These findings motivated us to explore how many observations in the validation set need to be labeled to achieve satisfying results. We fix training and test sets to the same size, 378 observations, and vary the number of observation in the validation set from 21 up to 378 instances. We simulate the biggest speech subgroup as the main observations and we add 5% from the other two audio groups as outliers. PCOut is not included to this experiment since it does not use a validation set.

Figure 2.4 (left) shows that both GRID PCA and OGK PCA are sensitive to the size of the validation set. Additionally, the AUC deviates considerably with decreasing size of

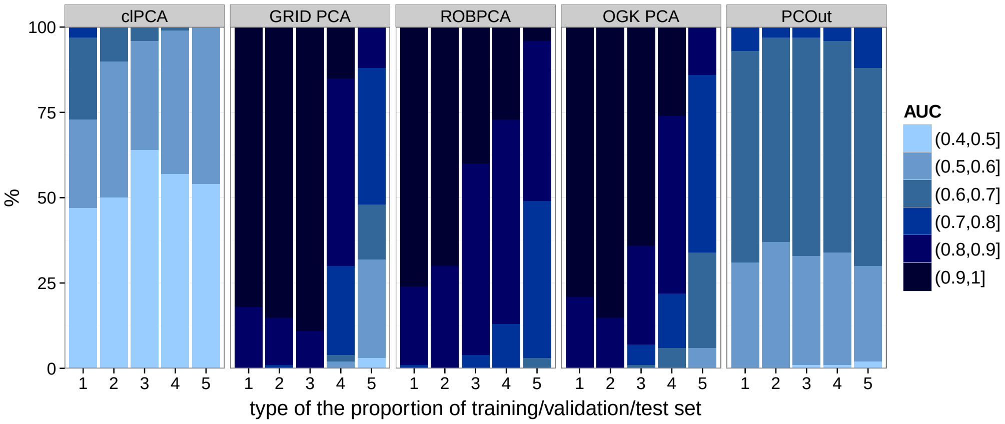

Figure 2.3: Detailed investigation of the resulting AUC during the replications for different partitions of training, validation, and test set (training/validation/test set) corresponding to the following size of sets: **1**: 378/378/380, **2**: 283/283/570, **3**: 170/170/796, **4**: 113/113/910, and **5**: 57/57/1022 observations.

the validation set. In contrast, ROBPCA performs well independently from the number of instances in the validation set and achieves a high AUC even if the size of the validation set is small in comparison to the training and test set. In general, the number of PCs (see Figure 2.4 (right)) decreases with reducing the size of validation set and deviates during the replications. ROBPCA indicates both small SE and slight decline in the selected number of PCs.

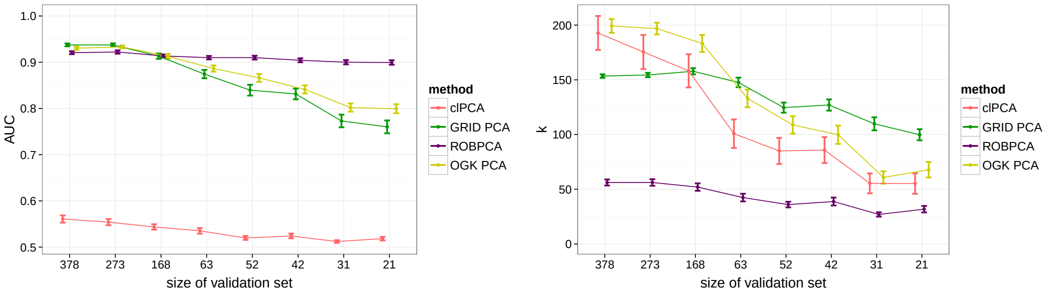

Figure 2.4: Evaluation results for varying sizes of the validation set in terms of AUC (left) and the number of PCs (right). Main observations are randomly selected from the speech sample.

To stress our conclusion that available labeled validation data set can be small to achieve reasonable results, we change the main group to music and perform the same experiment. The size of training and test set is fixed to 168 instances. Figure 2.5 indicates very similar performance and ROBPCA outperforms the remaining methods in all investigated situations.

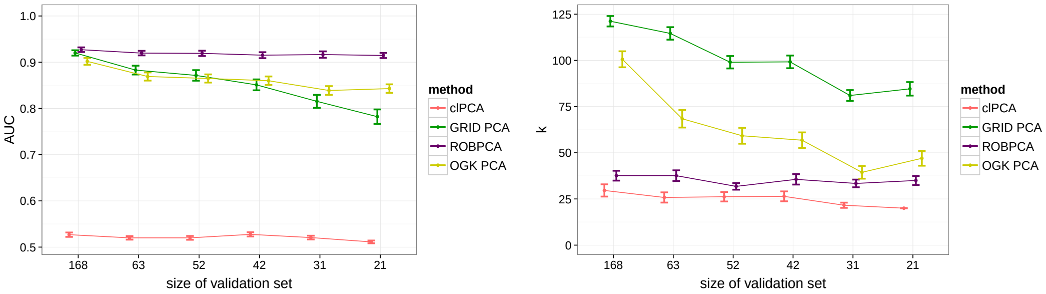

Figure 2.5: Evaluation results for varying sizes of the validation set in terms of AUC (left) and the number of PCs (right). The speech data are considered as main observations.

2.4.4 Sensitivity to the data characteristics

In this experiment we explore the sensitivity of the compared approaches to the underlying data characteristics with respect to varying data structures given by the different subpopulations in the audio dataset. We simulate the main observations consisting of three randomly selected audio subgroups with different sample size and the percentage of outliers is fixed to 5% of the corresponding main observations. We investigate the case of one majority subgroup present in the main observations and, in a next step, several subgroups.

Figure 2.6 shows that the performance is slightly better when a single majority group is considered. Although ROBPCA and GRID PCA achieve a higher AUC, the results indicate that these two methods face difficulties in coping with multi-group data structures. clPca completely fails with AUC of 0.5. Additionally, the values for the the SE of AUC are considerably higher than in the previous experiments. Overall, there is no clear dependency between the number of PCs and the different multi-group data structure.

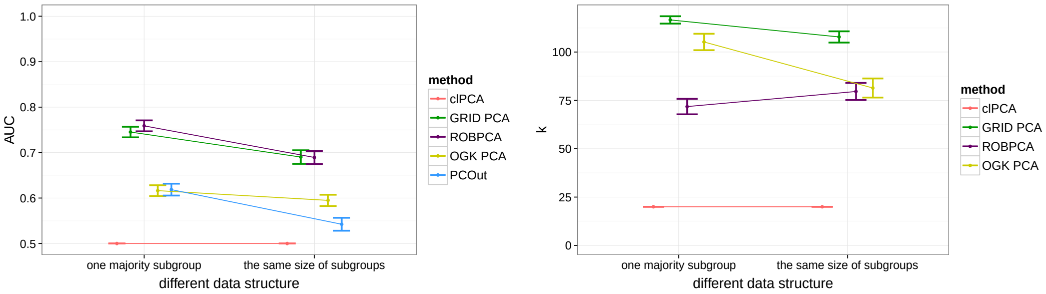

Figure 2.6: Evaluation results for different data structures in terms of AUC (left) and the number of PCs (right).

2.5 Discussion on critical values

The critical values are given by the quantiles of a χ^2 distribution for the SDs and the quantiles of the unknown distribution for the ODs which can be estimated by a robust Wilson-Hilferty approximation. Both critical values are based on the assumption of multivariate, normally distributed main observations. Those critical values are always an approximation since the distribution itself is estimated from the given observations. The central χ^2 distribution of the SDs and the non-central χ^2 distribution of ODs get distorted if the assumptions of normality are violated. In our experiments, we clearly observed data structures, which do not follow a normal distribution. We partly absorb this effect by using robust estimations. Therefore, the majority of observations can be properly modeled based on a normal distribution. To cope with the distorted distributions of the distances in addition to using robust estimations, we suggest to take advantage of the availability of validation data and to adjust the critical values. For this purpose, we can maximize the AUC performance for each fixed number of components, varying the critical values for SDs and ODs. Note, that the only meaningful critical values are the distances given by pre-labeled outliers. All other possible values will increase the FPR, without affecting the TPR. Thus, the necessary computational effort is very acceptable.

The main benefit of this procedure is the resulting robustness towards the number of chosen PCs. Figure 2.7 shows this effect for ROBPCA for one example of speech main observations with 5% outliers. The experiment indicates that performing the outlier detection for a low number of PCs is sufficient. Thus, even though the adjustment needs computation time, the total computational effort decreases, since it is no longer necessary to calculate a whole range of different numbers of PCs. At the same time, the risk of choosing an inappropriate number of PCs vanishes with increasing number of observations. It can be easily shown that the adjustment will asymptotically always perform at least as good as the theoretical critical values with increasing numbers of validation observations. If the theoretical assumptions of multivariate normal distribution holds where observations with large Mahalanobis distance are classified as outliers, the adjustment converges to the provided theoretical critical value due to the law of large numbers. For any non-normal distribution it converges to the respective true critical value and, therefore, it outperforms the theoretical critical values, derived from false assumptions. However, for large number of observations, especially outlying observations, the adjustment converges. Thus, the method should only be used to analyze setups where enough outlying observations allow for a proper estimation of the ROC curve.

2.6 Conclusion

In this paper we compared different PCA-based algorithms for outlier detection in the context of a high-dimensional audio data set. Since the classical PCA (Jolliffe, 2002) is sensitive to the presence of outliers in the training data, we employed several, well-established robust PCA methods, such as GRID PCA (Croux et al., 2007), ROBPCA (Hubert et al., 2005), OGK (Maronna and Zamar, 2002), and PCOUT (Filzmoser et al., 2008), to better

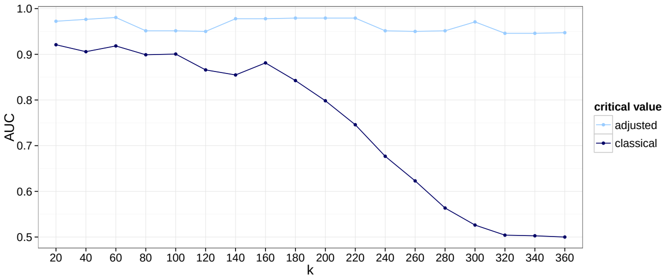

Figure 2.7: Comparison of AUC values depending on the number of components. While the quality of the classification for the classical critical values is highly depending on the chosen number of components, the adjusted critical values remains at an almost constant level.

reveal outlying samples. We performed a thorough investigation of the sensitivity of the employed approaches with respect to different data properties, percentage of outliers, and size of the available training data. In all of those settings, ROBPCA performed at the same level as the GRID and OGK algorithms. However, ROBPCA showed much lower sensitivity towards changes in the number of available training and validation observations. The reason for this property is the fewer necessary number of PCs to properly model the data structure. If the number of available observations is too low to create the necessary PCA space or to properly evaluate the used PCA space, the quality of the outcome decreases. We therefore recommend the usage of ROBPCA for outlier detection in similar setups where few pre-labeled observations allow for the individual estimation of a proper number of PCs. Further utilization of pre-labeled observations is possible by adjusting the critical values for outlier detection if the observations do not follow a normal distribution. In such a situation, if the number of observations, especially outliers, is big enough, the adjustment can significantly improve the quality of the proposed procedures, providing a more robust set of critical values, which are able to cope with skewed distributions.

Acknowledgments

This work has been partly funded by the Vienna Science and Technology Fund (WWTF) through project ICT12-010 and by the K-project DEXHELPP through COMET - Competence Centers for Excellent Technologies, supported by BMVIT, BMWFW and the province Vienna. The COMET program is administrated by FFG.

CHAPTER 3

Clustering of imbalanced high-dimensional media data

Abstract: Media content in large repositories usually exhibits multiple groups of strongly varying sizes. Media of potential interest often form notably smaller groups. Such media groups differ so much from the remaining data that it may be worthy to look at them in more detail. In contrast, media with popular content appear in larger groups. Identifying groups of varying sizes is addressed by clustering of imbalanced data. Clustering highly imbalanced media groups is additionally challenged by the high dimensionality of the underlying features. In this paper, we present the Imbalanced Clustering (IClust) algorithm designed to reveal group structures in high-dimensional media data. IClust employs an existing clustering method in order to find an initial set of a large number of potentially highly pure clusters which are then successively merged. The main advantage of IClust is that the number of clusters does not have to be prespecified and that no specific assumptions about the cluster or data characteristics need to be made. Experiments on real-world media data demonstrate that in comparison to existing methods, IClust is able to better identify media groups, especially groups of small sizes.

Key words: Clustering; Imbalanced data; High-dimensional data; Media data; LOF

Co-authors: Maia Zaharieva, Peter Filzmoser, Thomas Ortner, Christian Breiteneder

3.1 Introduction

Nowadays, large media repositories, such as YouTube[1] and Vimeo[2], are facing continuous additions of new, unknown material. Media, such as videos, sounds, and images, are straightforward to capture and share due to recent developments and advances of existing technologies. The processing and analysis of large amounts of unknown material is very challenging when trying to understand media content and when there is no prior knowledge available. Additionally, media of potential interest can easily get lost in the mass of available data. In this context, interestingness is a data-driven concept describing content which is that much different from the remaining data and, therefore, it is potentially worthy to look at in more detail. As any other data collection, media data commonly exhibits multiple groups. The underlying groups of media have strongly varying sizes, and therefore such a data set is considered as highly imbalanced. While larger groups represent a common type of media (e.g. video recordings of a popular music band), very small groups (even a size of one) commonly indicate atypical content and, thus, potential interesting material (e.g. video recordings of a non-famous street musician). The task of identifying small groups is additionally challenged by the characteristics of multimedia content. Media data are commonly represented by means of high-dimensional features. Conventional clustering methods based on traditional model assumptions usually fail in such a situation (Kriegel et al., 2009b). Therefore, the focus of this paper is on developing a cluster algorithm which addresses two core challenges: 1) clustering in a high-dimensional data space and 2) clustering imbalanced data with special attention on mining small groups.

Detecting clusters in high-dimensional space is commonly addressed by subspace or projected clustering algorithms which search for clusters in a subset of dimensions. Therefore, such methods are suitable for high-dimensional data where there is a large proportion of noise variables. However, these methods usually require for a parameter specification (Parsons et al., 2004) which may be problematic, especially for media data where no prior knowledge is available. On the contrary, if the number of noise variables is not too high, model-based clustering methods (e.g. Fraley and Raftery, 2000) or density-based algorithms (Kriegel et al., 2011), could still achieve promising results. Nevertheless, density-based approaches might be more appropriate for media data. Such methods commonly do not rely on any prior knowledge (e.g. number of clusters, shapes of clusters, and distribution of clustered points), which might be very beneficial when clustering media data.

Clustering imbalanced data, where group sizes are very different, causes additional challenges. Even though the research area of imbalanced clustering is not recent, there are still open issues which need to be addressed in the development of new methods (Krawczyk, 2016). A very first problem addressed by Krawczyk (2016), which usually occurs in centroid-based methods, is the so-called uniform effect. This means that a clustering

---

[1] http://www.youtube.com

[2] http://www.vimeo.com

algorithm generates clusters of similar sizes. Some observations from larger groups are mixed with those from smaller groups. In order to prevent the effect, Krawczyk (2016) proposed a hybridization of centroid-based and density-based methods. Another proposal can be found in literature (e.g. Wang and Chen, 2014; Qian and Saligrama, 2014). However, most approaches assume prior knowledge of the number of clusters in order to handle varying levels of imbalanced data. Finally, Krawczyk (2016) pointed out on the potential of discovering very small groups which could be useful for further analysis. Indeed, media collection is a good example of imbalanced data where small groups are of potential interest.

In this paper, we propose the Imbalanced Clustering (IClust) algorithm, an approach which is able to identify data groups of potentially strongly varying sizes. The procedure first employs an existing method which is forced to produce a large initial set of potentially pure clusters. Subsequently, clusters are successively merged using two merging conditions based on the outlier detection method - Local Outlier Factor (LOF) (Breunig et al., 2000). The algorithm stops when two merging conditions are not satisfied and, therefore, the final number of clusters does not need to be pre-specified. This is an advantage over most existing methods. Moreover, the proposed approach detects clusters of strongly varying sizes without any specific assumptions about the cluster or data characteristics, which is very important for clustering media data. Eventually, we employ a modification of the standard F-measure (Larsen and Aone, 1999) in order to better assess the ability of a clustering approach to find small and, thus, potentially interesting groups.

The remainder of this paper is organized as follows. Section 3.2 motivates the design of the proposed algorithm and describes it in detail. In Section 3.3, we select optimal parameters for the proposed algorithm, and results on real-world media data sets are presented in Section 3.4. Section 3.5 concludes the paper.

3.2 Proposed clustering algorithm

The idea for our algorithm originates in the need to efficiently detect small groups in media data, containing potentially interesting information. Small groups can easily get lost in a large media collection exhibiting groups of strongly varying sizes. In order to identify highly imbalanced groups, we employ the Local Outlier Factor (LOF) (Breunig et al., 2000), which was originally designed to reveal outliers deviating from clusters.

3.2.1 Background on Local Outlier Factor (LOF)

Preliminary experiments indicated that LOF is a highly effective approach for the identification of very small groups in media data as outliers in comparison to other existing approaches. In general, LOF determines the degree of outlyingness of an observation. The degree reflects to which extent an observation is isolated from its predefined number of the nearest observations. Let $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top \in \mathbb{R}^p$ be a data set of n observations from the Euclidean space of p dimensions. The LOF score for each observation $\mathbf{x}_i, i = 1, \ldots, n$, is defined according to Breunig et al. (2000) as:

$$LOF_q(\mathbf{x}_i) = \frac{1}{N_q(\mathbf{x}_i)} \sum_{\mathbf{x} \in N_q(\mathbf{x}_i)} \frac{lrd_q(\mathbf{x})}{lrd_q(\mathbf{x}_i)}, \tag{3.1}$$

where $N_q(\mathbf{x}_i)$ denotes the local neighborhood for $\mathbf{x}_i$ defined by its q nearest observations, i.e. neighbors, and $lrd_q(\mathbf{x}_i)$ corresponds to so-called local (reachability) density of $\mathbf{x}_i$. The local density of $\mathbf{x}_i$ reflects how distant $\mathbf{x}_i$ is with respect to its q nearest neighbors on average, taking into account the distances of its neighbors to their nearest observations, see Breunig et al. (2000) for more details. In general, if $\mathbf{x}_i$ belongs to a cluster, i.e. $\mathbf{x}_i$ is surrounded by or close enough to its neighbors $\mathbf{x} \in N_q(\mathbf{x}_i)$, the local density of $\mathbf{x}_i$ is similar to the local densities of its neighbors. As a consequence, $LOF_q(\mathbf{x}_i)$ achieves a value of approximately 1. In contrast, an observation $\mathbf{x}_i$ deviating from a cluster has considerably different local densities than its neighbors since $\mathbf{x}_i$ is highly isolated from its neighbors. Therefore, such an observation receives $LOF_q(\mathbf{x}_i) >> 1$ and can thus be declared as outlier.

In addition to the ability of LOF to detect outliers based on large LOF scores, the LOF approach exhibits several properties which might be very useful in clustering imbalanced data. First, LOF considers both distances between observations and local densities of observations, and, therefore, it is suitable for the contaminated data with the clusters of varying sizes and densities (Hasan et al., 2009). Second, the decision of declaring a point as an outlier seems to be insensitive to the choice of the predefined number of neighbors necessary for calculating LOF scores (Hasan et al., 2009). Next, LOF does not rely on any specific assumptions on the cluster characteristics (Breunig et al., 2000), which is particularly important for high-dimensional media data where such assumptions could not be verified. Finally, Zimek et al. (2012) demonstrated that LOF achieves promising results if the number of informative variables in the high-dimensional data is not too low. We expect that media data also contain many informative variables, while, for example, for gene expression data this might not necessarily be the case.

Despite the mentioned advantages of LOF and the fact that the method is capable of detecting very small groups in media data as outliers, we still need to adjust the usage of LOF in order to recover a whole group structure in imbalanced media data.

3.2.2 Naive approaches

A naive idea would be to remove the detected outliers from the data, and to use existing cluster methods for clustering the larger groups. However, this leads to difficulties since the resulting group sizes might still be very different and because many clustering algorithms assume a certain shape of the clusters. Another idea would be a recursive identification of the clusters, by starting to build the first cluster in the most dense and compact region. In the following, LOF can be used to decide which points are still members of this cluster, and which point is too far away (outlier) in order to form a new cluster. However, this decision can become unreliable as illustrated in Figure 3.1. Suppose that there are two groups C_1 and C_2 with different sizes and densities (left picture). Assume that the first cluster K has been constructed and it needs to be decided whether or not observation $\mathbf{x}$ still belongs to cluster K (middle picture). If the neighborhood size used to compute the LOF score is not small enough, the point $\mathbf{x}$ would be assigned to cluster K because of the different point densities of the underlying clusters (right picture). Using a very small neighborhood size instead would again be unreliable because the decision would be based on too little information. Therefore, for being able to identify groups of very small sizes, even of size one, we need to modify the concept, which leads to the proposed IClust approach.

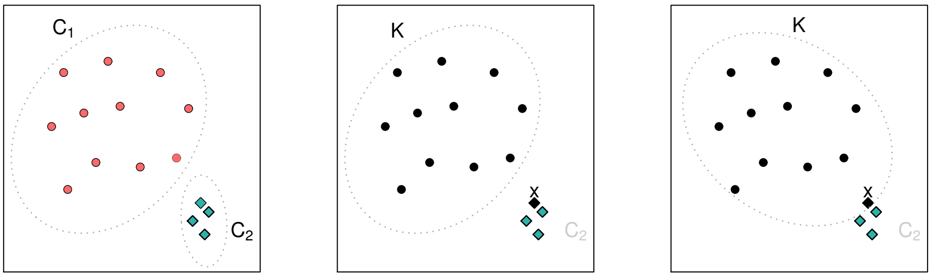

Figure 3.1: Example of a data set with two groups of different sizes and densities, C_1 and C_2 (left); the correctly identified group C_1 as a cluster K with its next closest point $\mathbf{x}$ from group C_2 displayed in black color (middle); a wrong assignment of $\mathbf{x}$ to K due to a LOF-based decision (right).

3.2.3 The IClust algorithm

The proposed algorithm is conducted in two steps. In the first step of IClust, we identify a large number of initial clusters by employing an existing clustering method. We suggest to take a simple existing method which allows to control the number of clusters, e.g. k-means. The large number of initial clusters leads to potentially highly pure regions of comparable densities of clustered points. In the second step, we merge clusters by employing the LOF approach. In each merging step it is investigated if two closest clusters share the same local densities; if this is the case, the clusters are merged. This avoids the wrong assignment as indicated in Figure 3.1. The algorithm iteratively tries to merge the next two closest clusters until there are no clusters to be merged. We give a detailed description of each step in the following sections.

3.2.4 Identifying the initial set of clusters

In the first step, the data set is split into an initial set of clusters by applying an existing clustering algorithm to subdivide the underlying data set into a large number of clusters. Although such a partitioning leads to over-clustering of the data, it allows for the detection of (highly) pure clusters. We propose to use such a clustering algorithm that is less computationally demanding and requires for the number of clusters only in order to enable to control the number of initial clusters and not to be influenced by a wrong choice of parameters which is usually data-dependent, e.g. k-means.

The number of initial clusters k_{init} needs to be set large enough (larger than the true underlying number of clusters) in order to increase the probability of obtaining highly pure clusters. However, the value of k_{init} should not be too large to have a sufficient number of observations, i.e. information, in most clusters for the merging procedure. There are several possibilities for the determination of the number of initial clusters. For example, Bloisi and Iocchi (2008) suggest to partition the data into $n/4$ clusters. Owsiński and Mejza (2007) recommend the number to be set to $n^{1/2}$. In Section 3.3 we investigate different selection strategies for k_{init} and their influence on the clustering solution.

We will experiment with several well-known clustering methods to identify the optimal choice for a starting clustering algorithm in Section 3.3. We consider two partitioning methods, *k-means* (Hartigan and Wong, 1979)[3] and *Partitioning Around Medoids* (Kaufman and Rousseeuw, 2005)[4], two hierarchical methods, *complete linkage* and *Ward's method* (Murtagh and Legendre, 2014)[5], and the model-based clustering method *Mclust*. All methods have certain drawbacks in terms of generating specifically shaped clusters and they suffer from the so-called uniform effect. By incorporating these methods in the first step we can enhance their performance in a highly imbalanced scenario. A large number of clusters generated by these methods avoids the uniform effect. Furthermore, over-clustering can prevent from being affected by the assumption about the shapes of clusters.

To illustrate both aspects, we apply k-means on an imbalanced data set. We consider a simple 2D data set with three groups of different sizes as shown in Figure 3.2 (left). Applying k-means with the true number of clusters, i.e. $k = 3$, results in the wrong assignment of observations from the large group to the smaller groups, see Figure 3.2 (middle), because k-means tends to produce spherically shaped clusters. In contrast, k-means with a larger number of clusters, e.g. $k = 6$, results in a solution as shown in Figure 3.2 (right), where the smaller groups are correctly detected and the large group is split into four small but pure clusters. Therefore, in the next step, we aim at merging small clusters that likely belong to the same group while keeping well-isolated and, thus, potentially correctly detected small groups.

3.2.5 Merging procedure

The aim of the second step is to iteratively merge clusters that are close to each other and share the same local densities. The underlying assumption is that such clusters

---

[3] k-means is implemented in R package `stats` (R Core Team, 2016).

[4] PAM is implemented in the R package `cluster` (Maechler et al., 2015).

[5] Complete linkage and Ward's method are implemented in the R package `cluster` (Maechler et al., 2015).

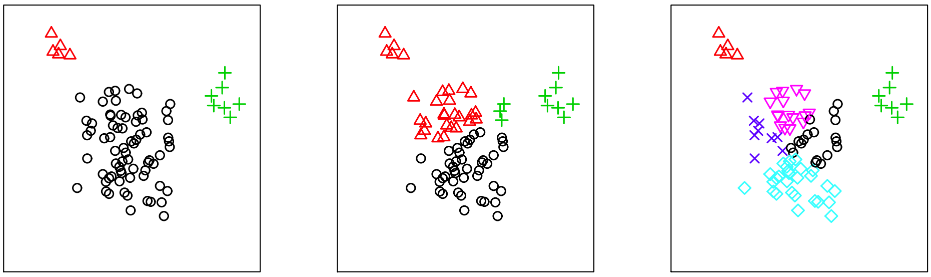

Figure 3.2: The effect of k-means applied on a 2D imbalanced data set. Imbalanced data set with three groups (left), k-means with three clusters (middle) and k-means with six clusters (right).

contain observations from the same group. In order to consider both distances and local densities, we propose to investigate if a point from one cluster can be considered as part of a second cluster and vice versa by employing the LOF. The purpose of investigating two clusters twice is to avoid that a cluster of low density is merged with a cluster of high density, as discussed at the beginning of this section. Therefore, we introduce the two merging conditions which need to be satisfied to merge the two closest clusters.

Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ be a data set of n observations and $\{K_r | r = 1, \ldots, k_{init}\}$ the initial set of clusters, where $K_r = \{\mathbf{x}_{i_r} | i_r \in I_r\}$ contains observations from the index set $I_r = \{1_r, 2_r, \ldots, |K_r|_r\}$. The merging procedure starts by finding the two closest clusters, K_l and K_m, based on the minimum distance between each pair of observations coming from different clusters (single linkage approach). In addition, the two closest points, $\mathbf{x}_o \in K_l$ and $\mathbf{x}_p \in K_m$, are identified such that

$$d(\mathbf{x}_o, \mathbf{x}_p) = \min_{\mathbf{x}_{i_l} \in K_l, \mathbf{x}_{i_m} \in K_m} d(\mathbf{x}_{i_l}, \mathbf{x}_{i_m}). \tag{3.2}$$

Subsequently, we investigate whether or not the two clusters should be merged.

For illustration, we consider the simple example in Figure 3.3 (left) showing two clusters K_l and K_m with the corresponding closest points $\mathbf{x}_o \in K_l$ and $\mathbf{x}_p \in K_m$. Figure 3.3 (middle) shows the idea of investigating whether or not $\mathbf{x}_p$ can be part of K_l considering that the neighborhood is defined by three closest neighbors, denoted as $q = 3$. The plot particularly indicates that $\mathbf{x}_p$ is close to its three neighbors from K_l, which are located in the circle around $\mathbf{x}_p$. In addition, it seems that the observations located in the neighborhood (displayed as circles) form a compact region of similar local densities. As a result, the LOF score of $\mathbf{x}_p$ should be approximately 1. In such a case, we conclude that $\mathbf{x}_p$ can be considered as part of the second cluster K_l.

Formally, we calculate the LOF score for observations from K_l and the point $\mathbf{x}_p$ according to (Breunig et al., 2000) for a predefined range of the number of nearest neighbors q,

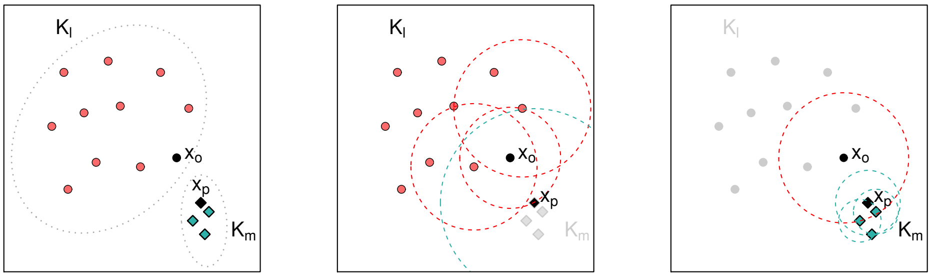

Figure 3.3: Illustration of the proposed merging procedure for two clusters K_l and K_m with the corresponding closest points $\mathbf{x}_o \in K_l$, $\mathbf{x}_p \in K_m$ (left). The neighborhood of $\mathbf{x}_p$ with its three nearest neighbors is displayed as circles. $\mathbf{x}_p$ is close to its three neighbors (from K_l) located in the circle around $\mathbf{x}_p$ (middle). $\mathbf{x}_o$ is highly isolated from its three neighbors (from K_m) located in the circle around $\mathbf{x}_o$ (right).

determined by the maximal number of nearest neighbors q_{max}:

$$LOF_q(\mathbf{x}_i), \quad i \in \{I_l \cup p\}, \quad q = 1, 2, \ldots, q_{max}, \tag{3.3}$$

This results in several values of the LOF score for each observation depending on the range of q. The reason for considering different choices of q is to obtain more information about the local densities with respect to various sizes of the neighborhood. Subsequently, we calculate a representative value, $lof(\mathbf{x}_i)$:

$$lof(\mathbf{x}_i) = \frac{1}{|q|} \sum_q LOF_q(\mathbf{x}_i) \quad i \in \{I_l \cup p\}, \quad q = 1, 2, \ldots, q_{max}, \tag{3.4}$$

where $|q|$ denotes the number of different choices of q. We provide an empirical study on the proper choice of q in Section 3.3. For now, suppose that q is given. The value of $lof(\mathbf{x}_i)$ describes the average similarity between the local density of an observation $\mathbf{x}_i$ and the local densities of its neighbors. In addition, $lof(\mathbf{x}_i)$ has similar properties as the LOF score since it is a linear combination of the original scores. The higher the value of $lof(\mathbf{x}_p)$, the more different are the local densities of $\mathbf{x}_p$ with respect to neighbors from K_l, and the more likely $\mathbf{x}_p$ is an outlier with respect to K_l, i.e. $\mathbf{x}_p$ cannot be a part of K_l. In order to decide if the compared local densities are similar, i.e. $\mathbf{x}_p$ can be part of K_l, it is necessary to determine how large $lof(\mathbf{x}_p)$ still can be to consider $\mathbf{x}_p$ as part of K_l . The most convenient option would be to decide on the basis of the resulting LOF scores. Therefore, we estimate a critical value cv^p from the values of $LOF_q(\mathbf{x}_i)$, where $i \in \{I_l \cup p\}$ and $q = 1, 2, \ldots, q_{max}$. There are several possibilities for the determination of the critical value, such as using the arithmetic mean and standard deviation or robust versions thereof. The optimal strategy for the estimation of the critical value is presented in Section 3.3. For now, we assume that cv^p is given and we test if *the first merging condition*, $lof(\mathbf{x}_p) < cv^p$, holds. If the first condition is fulfilled, we consider the compared local densities similar and apply the same comparison on $\mathbf{x}_o$ with respect to K_m, i.e. we investigate if $\mathbf{x}_o$ can be considered as part of the opposite cluster K_m.

Figure 3.3 (right) shows that $\mathbf{x}_o$ is considerably isolated from its three closest neighbors from K_m and that the observations do not build any compact region. In such a case we can conclude that there are huge differences in the local densities. Therefore, $LOF_q(\mathbf{x}_o) \gg 1$ which indicates that $\mathbf{x}_o$ can not be considered as a part of the second cluster K_m. Formally, we calculate the LOF score for the observations from K_m and the point $\mathbf{x}_o$ for the predefined range $q = 1, 2, \ldots, q_{max}$. The critical value cv^o is estimated in the same way as cv^p. Subsequently, we test if *the second merging condition*, $lof(\mathbf{x}_o) < cv^o$, is satisfied. If this condition is not fulfilled, the two clusters, K_m and K_l, are not merged and the next two closest clusters are investigated. The merging procedure stops if the two conditions are not satisfied for any pair of clusters.

The proposed IClust algorithm exhibits several advantages. First, the number of final clusters does not need to be pre-specified due to the employed merging procedure. Second, the proposed procedure makes no assumptions about the cluster characteristics. The local densities are estimated in a non-parametric way following the definition of LOF. Finally, IClust detects clusters of partly strongly varying sizes. By using an existing clustering algorithm with a large number of clusters, we avoid the so-called uniform effect. For illustration, we consider the 2D imbalanced data set and apply k-means with $k = 6$ to generate an initial set of clusters as in our first example in Figure 3.2 (right). Figure 3.4 shows each merging of the two next closest clusters. The final result indicates that both smaller and larger groups are correctly detected.

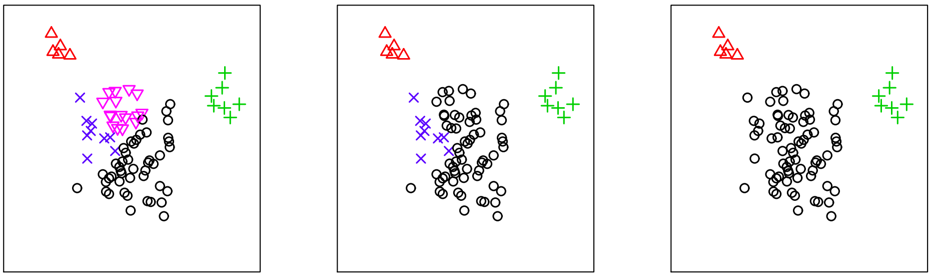

Figure 3.4: The merging procedure successively merges clusters that are close to each other and share the same distribution. The final solution indicates that the detected clusters correspond to the actual group structure.

3.3 Selection of parameters

In this section, we investigate different strategies for the parameter selection. The IClust algorithm requires for four input parameters: 1) the critical value, cv^o (cv^p), 2) the range of the nearest neighbors, q, 3) the number of initial clusters, k_{init}, and 4) the starting clustering algorithm. While the first two parameters are employed in the merging procedure, the last two parameters are used to partition the data set into an initial set of clusters. Since two parameters depend on each other, we always fix one parameter to investigate the second one and vice versa. The optimal parameter setting is chosen based on thorough empirical experiments employing the audio data set.

The audio data set consists of 4780 observations which is a collection of 12 different audio sounds. Each observation is represented by a feature vector of 679 dimensions. The extracted features capture a wide range of audio properties and operate in the temporal and frequency domains, i.e. the data include features such as zero crossings, amplitude, brightness, features from the MPEG7 standard, perceptional features, and various cepstral coefficients. In our experiments we randomly sample observations from the original groups to create imbalanced data sets. The variables of each constructed data set are normalized to mean 0 and standard deviation 1.

3.3.1 Critical value

The first experiment investigates different strategies for the estimation of the critical values, cv^o and cv^p, employed in the merging procedure. The critical values determine whether or not two clusters should be merged. Both critical values are estimated in the same way, therefore, let cv be a general estimation of the critical value. The value of cv is supposed to be automatically derived from the LOF scores calculated for the observations from two clusters. We consider several possibilities for the estimation of cv including arithmetic mean, *mean*(), empirical standard deviation, *std*(), and robust versions, such as the median, *median*() and the median absolute deviation, *mad*():

$$cv_1 = \underset{q,i}{\text{median}}\,(LOF_q(\mathbf{x}_i)) + 2\,\underset{q,i}{\text{mad}}\,(LOF_q(\mathbf{x}_i)) \tag{3.5}$$

$$cv_2 = \underset{q,i}{\text{mean}}\,(LOF_q(\mathbf{x}_i)) + 2\,\underset{q,i}{\text{std}}\,(LOF_q(\mathbf{x}_i)) \tag{3.6}$$

$$cv_3 = \underset{i}{\text{median}}\,(lof(\mathbf{x}_i)) + 2\,\underset{i}{\text{mad}}\,(lof(\mathbf{x}_i)) \tag{3.7}$$

$$cv_4 = \underset{i}{\text{mean}}\,(lof(\mathbf{x}_i)) + 2\,\underset{i}{\text{std}}\,(lof(\mathbf{x}_i)), \tag{3.8}$$

where i is either from the index set $\{I_l \cup p\}$ for the first merging condition or from $\{I_m \cup o\}$ for the second merging condition. The range for the number of nearest neighbors determined by q_{max} is fixed to $min(|\{I_l \cup p\}| - 1, 5)$ and $min(|\{I_m \cup o\}| - 1, 5)$, respectively.

Since cv determines whether or not two clusters should be merged, we investigate two situations. We first simulate the situation when two clusters should not be merged, i.e. the underlying observations are from two different groups. The second situation considers two clusters containing observations from the same group and, therefore, the two clusters are supposed to be merged. For this experiment we employ observations from the audio data. We randomly sample two clusters either from two different audio groups or from the same audio group. The sampled clusters are of varying sizes of $30, 25, 20, 15, 10, 5, 3, 1$. We investigate each possible pairwise combination thereof and perform 10 replications for each combination. The percentage of correct decisions is considered as a performance indicator for the different strategies for the estimation of the critical value, *cv*.

The results of this experiment are presented in Figure 3.5. For all strategies, the percentage of correct decisions is higher when the two investigated clusters are sampled from the same group, see the right (white) boxplot for each strategy, in comparison to the situation when the clusters are sampled from different groups, see the left (gray) boxplot for each strategy. In general, the robustly estimated critical values, i.e. cv_1 and cv_3, outperform their standard counterparts. Since there is no clear difference between the two robust strategies, we select cv_1 as the estimation of the critical values for all our following experiments.

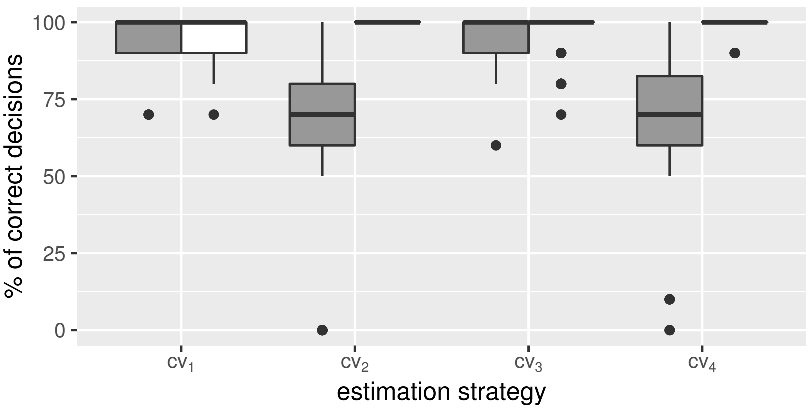

Figure 3.5: Comparison of different strategies for the estimation of the critical value *cv*. For each *cv* two boxplots are displayed. The left (gray) boxplot represents the results when two clusters are sampled from two different groups and the right (white) boxplot when the two clusters are sampled from the same group.

3.3.2 Number of nearest neighbors

The aim of the second experiment is to determine the optimal range for the number of nearest neighbors, $q = 1, 2, \dots, q_{max}$, considered in the merging procedure. We investigate three options for the maximal number of nearest neighbors, $q_{max} = 5, 10, 15$. We employ the same clusters as in the previous experiment and the percentage of correct decisions as an indicator for the optimal choice of q_{max}.

Figure 3.6 summarizes the results of the experiment. In general, all choices of q_{max} lead to a high percentage of correct decisions and no clear difference can be observed. As a result, we choose $q_{max} = 5$ for all our following experiments for computational reasons.

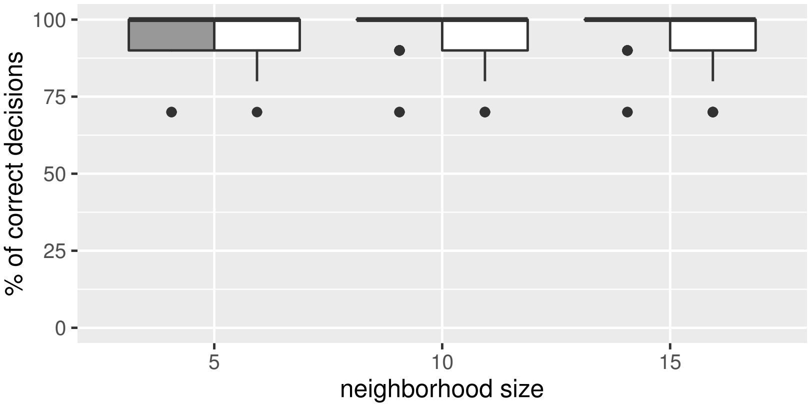

Figure 3.6: Comparison of different maximal numbers of nearest neighbors, q_{max}. Two boxplots are shown for each q_{max}. The left (gray) boxplot represents the results when two clusters are sampled from two different groups and the right (white) boxplot when the two clusters are sampled from the same group.

3.3.3 Number of initial clusters

The goal of this experiment is to identify the best strategy for the selection of the number of initial clusters, k_{init}, employed in the first step of the proposed IClust algorithm. The parameter is used to partition a given data set into a number of potentially highly pure clusters which are successively merged in the second step of IClust. In general, k_{init} is supposed to be larger than the actual number of groups in the data set. The actual number of groups is usually not known in advance and the only available information about the data set is the sample size, n. Therefore, we determine k_{init} as a function of n. One possible approach is to set k_{init} to be linear dependent on n, e.g. $k_{init} = n/4$. However, if the size of a data set is very large, the parameter k_{init} will get considerably high which will notably increase the computational effort of the merging procedure. In addition, a large k_{init} value leads to a small size of the initial clusters which might affect the efficiency of the merging procedure.

We investigate various options for k_{init} to be non-linearly dependent on n, $k_{init} = 5\log(n), 10\log(n), 15\log(n)$. This experiment is again based on 10 replications of an imbalanced data set sampled from audio data. In each replication, we randomly select 7 audio groups with 3 bigger groups of the size: $100, 75, 50$, and 4 smaller groups of the size: $4, 3, 2, 1$, resulting in 235 observations in total. We used Ward's hierarchical clustering algorithm to obtain the initial set of clusters.

In order to assess the influence of the different settings for k_{init} on the final clustering solution, we select several well-known evaluation measures: Purity (Zhao and Karypis, 2002), F-measure combining the concepts of precision and recall, and V-measure (Rosenberg and Hirschberg, 2007) incorporating homogeneity and completeness scores. Such measures evaluate a clustering solution as a whole and do not reflect the ability of a clustering method to correctly detect small groups. For this reason, we also employ weighted measures which can assess the performance of a clustering method in terms of detecting small and big groups separately. Table 3.1 contains all employed measures. All measures range between zero and one with higher values indicating a good clustering result and lower values corresponding to a poor clustering solution. Additionally, we provide two reference values representing potential extremes. The first value corresponds to a clustering solution when all detected clusters are of size one, while the second value corresponds to a clustering solution when all observations are assigned to a single cluster. We report the results before and after applying the merging procedure to demonstrate the performance of the IClust approach in comparison to the initial clustering solution.

Table 3.1: Overview of employed evaluation measures with corresponding abbreviations (abbr).

evaluation measure	abbr	weighted evaluation measure	abbr
purity	P	F-measure - big groups	wF^b
F-measure	F	precision - big groups	wPr^b
V-measure	V	recall - big groups	wRe^b
homogeneity	H	F-measure - small groups	wF^s
completeness	C	precision - small groups	wPr^s
		recall - small groups	wRe^s

Figure 3.7 depicts the results of the experiments using the conventional clustering evaluation measures. In general, IClust always improves the initial clustering solution indicated by the notable raise of the F-measure (F) and the V-measure (V) which is directly influenced by the completeness (C) of the corresponding clustering solution. Additionally, the scores for P and H show that the purity and homogeneity of the initial clusters are comparable to those of the final clusters. In general, a low number of initial clusters, $k_{init} = 5\log(n)$, leads to a high completeness (C) but also to a lower homogeneity (H) of the final clusters, i.e. final clusters partly consists of observations from different groups. On the opposite, a higher number of initial clusters, $k_{init} = 15\log(n)$, results in purer clusters (see P and H). Additionally, V, C, and F are only slightly lower than for $k_{init} = 10\log(n)$. This may indicate that $k_{init} = 10\log(n)$ is the proper choice. To further explore the influence of the different settings for k_{init}, we additionally consider the weighted clustering evaluation measures.

Figure 3.8 summarizes the results of the weighted evaluation measures. The high scores for wPr^b indicate highly pure clusters containing observations from bigger groups independently of k_{init}. However, the corresponding recall (wRe^b) decreases with an increasing number of initial clusters which reveals that bigger groups are represented by several clusters in the final clustering solution. This indicates that there are not enough observations in most initial clusters leading to difficulties for a proper merging. As a result, a high number of initial cluster, $k_{init} = 15\log(n)$, results in a lower F-measure

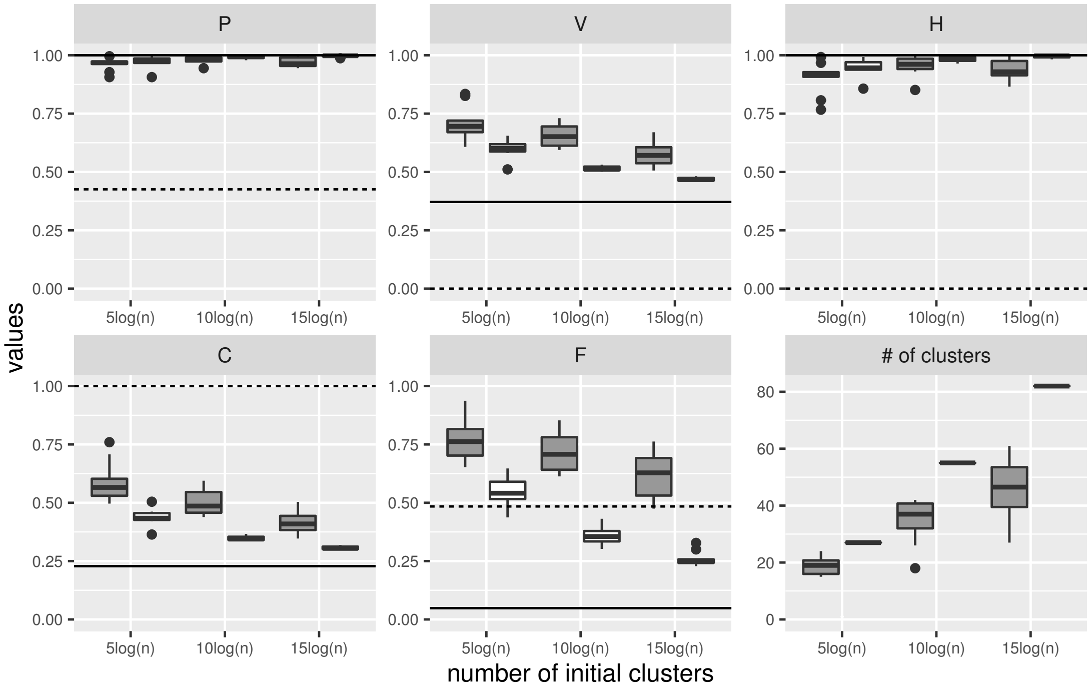

Figure 3.7: Comparison of the clustering results for different numbers of initial clusters, k_{init}, using purity (P), V-measure (V), homogeneity (H), completeness (C), F-measure (F), and number (#) of clusters. For each k_{init} two boxplots are displayed. The results of the final clusterings correspond to the left (gray) boxplot and the results of the initial clusterings to the right (white) boxplot. The lines indicate two extreme clustering solutions. Solid lines: all clusters are of size one. Dashed lines: all observations are assigned to a single cluster.

(wF^b) in comparison to the extreme situation all observations build a single final cluster. With respect to small clusters, with an increasing number of initial clusters the precision, wPr^s, increases while the recall, wRe^s, decreases slightly. A too low number of initial clusters, such as $k_{init} = 5\log(n)$, results in a poor clustering solution indicated by the lower median of wF^s in comparison to the potential extreme situation with all final clusters of size one. Therefore, we set $k_{init} = 10\log(n)$ for all our following experiments.

3.3.4 Initial clustering algorithm

The last experiment focuses on the selection of the starting clustering algorithm employed in the first step of IClust to partition the provided data set into a number of initial clusters. For this evaluation we consider the following clustering methods (see Section 3.2.4): k-means (KM), Partitioning Around Medoids (PAM), Mclust (MC), and a hierarchical clustering with Ward's criterion (W) and complete linkage (CL). The experiment is again

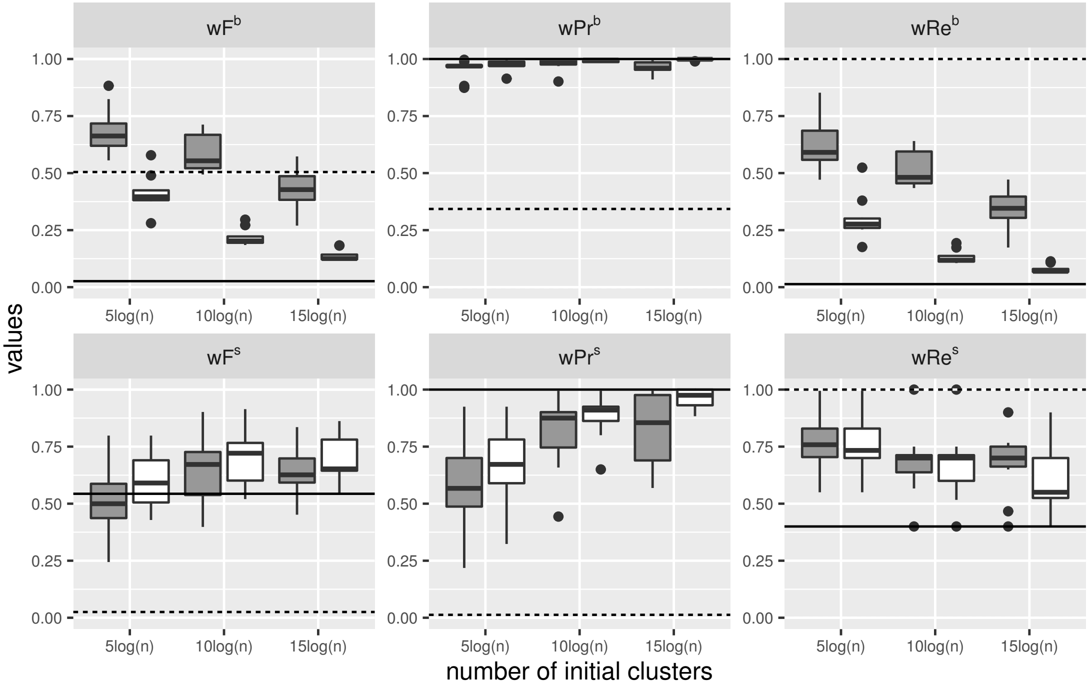

Figure 3.8: Comparison of the clustering solutions for different numbers of initial clusters, k_{init}, using the weighted measures for F-measure (wF), precision (wPr), and recall (wRe) with respect to small (^{s}) and big (^{b}) clusters. For each k_{init} two boxplots are displayed. The results of the final clusterings correspond to the left (gray) boxplot and the results of the initial clusterings to the right (white) boxplot. The lines indicate two extreme clustering solutions. Solid lines: all clusters are of size one. Dashed lines: all observations are assigned to a single cluster.

based on 10 replications of an imbalanced data set randomly sampled from the audio data. The employed data sets are identical to the data sets in the previous experiment.

Figure 3.9 presents the results using the conventional clustering evaluation measures. In general, the proposed IClust algorithm improves the initial clustering solution independently of the employed starting clustering method. This seems to confirm our assumption that IClust can enhance the performance of methods suffering from the uniform effect. The high scores for purity (P) and homogeneity (H) indicate the IClust results in highly pure clusters. Although the F measure indicates slightly better clustering results for PAM than for its counterparts, overall, the results do not show any clear differences across the employed starting clustering algorithms.

Figure 3.10 summarizes the results using the weighted clustering evaluation measures. The high precision scores (wPr^b) indicate that clusters containing observations from bigger groups are highly pure. However, the lower recall (wRe^b) and, in following, the

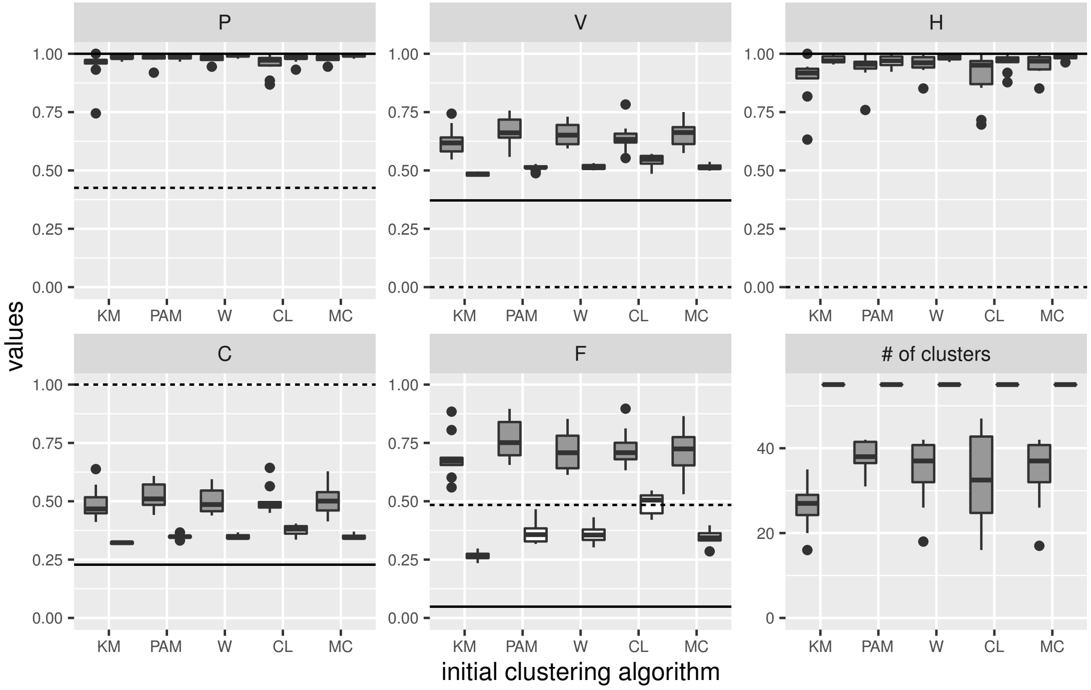

Figure 3.9: Comparison of the performance of different initial clustering algorithms, k-means (KM), PAM, Mclust (MC), and a hierarchical clustering with Ward's criterion (W) and complete linkage (CL) using the conventional clustering evaluation measures: purity (P), V-measure (V), homogeneity (H), completeness (C), F-measure (F), and number (#) of clusters. Two boxplots are shown for each algorithm. The results of the final clusterings correspond to the left (gray) boxplot and the results of the initial clusterings to the right (white) boxplot. The lines indicate two extreme clustering solutions. Solid lines: all clusters are of size one. Dashed lines: all observations are assigned to a single cluster.

lower F-measure (wF^b) show that bigger groups are commonly partitioned into multiple clusters. The PAM clustering method slightly outperforms the other employed methods in terms of wF^b. However, wF^s indicates that PAM cannot reveal smaller groups since the median is below the value representing the extreme clustering result with all final clusters of size one. Similarly, complete linkage (CL) and k-means (KM) achieve a wF^s close to the extreme clustering solution. Both methods generate clusters containing observations from more than a single group. Therefore, the precision with respect to smaller groups (wPr^s) achieves low scores and directly influences the F-measure (wF^s). Ward's method (W) and Mclust (M) better facilitate the detection of small groups in terms of wF^s. For all our following experiments we select Ward's method (W) as the starting clustering approach since it is less computationally demanding than the Mclust (MC) algorithm.

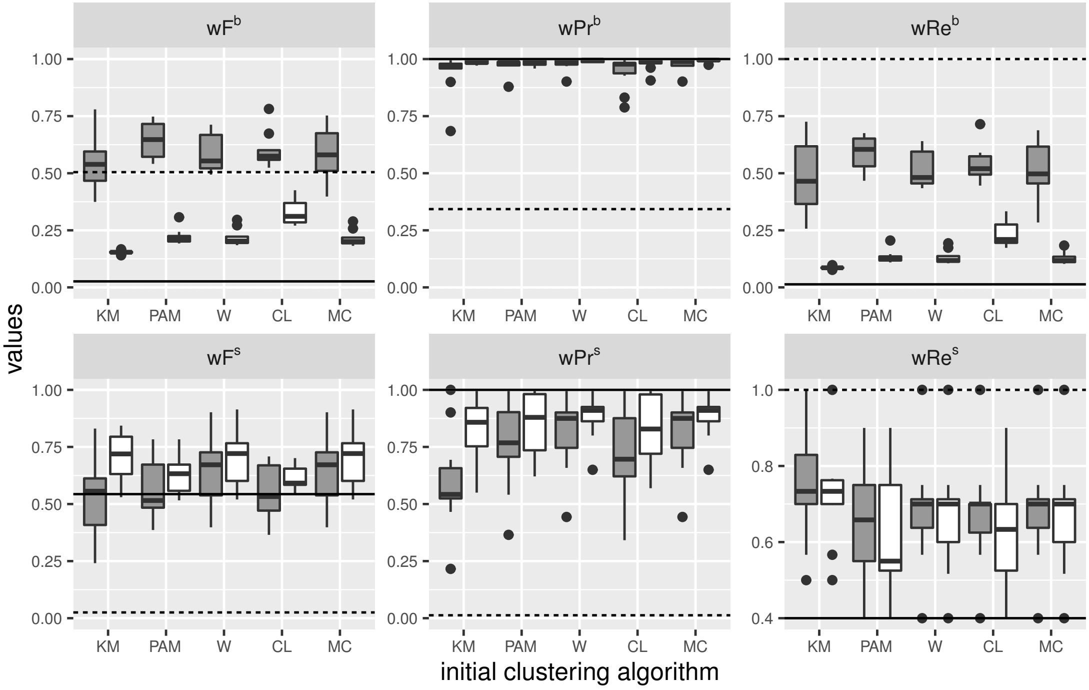

Figure 3.10: Comparison of the performance of different initial clustering algorithms, k-means (KM), PAM, Mclust (MC), and a hierarchical clustering with Ward's criterion (W) and complete linkage (CL) respectively using the weighted measures for F-measure (wF), precision (wPr), and recall (wRe) with respect to small (^{s}) and big (^{b}) clusters. Two boxplots are shown for each algorithm. The results of the final clusterings correspond to he left (gray) boxplot and the results of the initial clusterings to the right (white) boxplot. The lines indicate two extreme clustering solutions. Solid lines: all clusters are of size one. Dashed lines: all observations are assigned to a single cluster.

3.4 Experimental comparison

In this section, we compare the proposed IClust approach to several clustering methods on four real-media data sets. In our comparison, we considered several existing, well-established clustering methods: Affinity Propagation (AP) (Frey and Dueck, 2007), Mclust (MC) (Fraley and Raftery, 2000), and x-means(XM) (Ishioka, 2000). We restrict the selection of compared methods to parameter-free approaches to ensure that a clustering solution is not affected by a wrong choice of parameters since a parameter setting commonly depends on the nature of the underlying data. The empirical simulation study in Section 3.3 indicated that the Ward's algorithm (W) seems to be an optimal initial clustering approach. Therefore, we also include the method in our final comparison. The number of clusters for the W method is taken as the true number of groups (WT) and it is also estimated using three clustering indices[6]: *Davies-Bouldin* (WDB), *gap statistic* (WG), and *Silhouette* (WS) (Rousseeuw, 1987). In order to have a fair comparison, the upper boundary of a predefined range of the number of clusters is set to the same number of initial clusters employed in IClust, i.e. $10\log(n)$. Note that the same upper boundary is considered for Mclust (MC) which estimates the optimal number of clusters based on the largest BIC value. IClust is employed with the previously determined settings, $cv = cv_1$, $q = 1, 2, \ldots, 5$, $k_{init} = 10\log(n)$, and Ward's hierarchical clustering as the initial clustering approach.

In addition to audio data, we employ three media data sets publicly available in UCI machine learning repository[7], see Table 3.2. For each media set we randomly select observations from the original groups to construct a similar imbalanced data sets. The ratios between group sizes are kept the same among the constructed datasets, but the sizes of the groups are different. The idea behind this setup is to see whether or not the compared methods are affected by different amounts of information, i.e. the number of observations, in the groups. All experiments are based on 10 replications. Since similarities among the considered evaluation measures were observed, we report the results using five clustering evaluation measures: V-measure (V), homogeneity (H), completeness (C), as well as the weighted F-measures with respect to big and small clusters (wF^b and wF^s).

Table 3.2: Overview of the employed real-world data sets in terms of number of observations (n), dimensionality (p), and number (#) of groups.

	n	×	p	×	#groups	group size min	group size max
Audio	4780	×	679	×	12	102	2164
Human Activity Recognition	10299	×	561	×	6	1406	1944
Pen-Based Recognition	10992	×	16	×	10	1055	1144
Statlog Landsat Satellite	6435	×	36	×	6	626	1533

3.4.1 Comparison on the audio data

We first construct 10 imbalanced data sets from the high-dimensional audio data. Each setting includes 10 groups with 3 bigger groups of the sizes: $100, 75, 50$, and 7 smaller groups of the sizes: $4, 3, 3, 2, 2, 1, 1$, resulting in 241 observations in total.

Figure 3.11 indicates poor performance of some centroid-based approaches, such as x-means (XM) and the Ward's method with Silhouette Width (WS), in clustering highly imbalanced media (audio) groups. The methods detect a lower number of clusters than the actual number of groups (cp. WT approach) leading to low homogeneity (H). High

---

[6] All clustering indices are implemented in the R package `clusterSim` (Walesiak and Dudek, 2015).

[7] UCI Machine Learning Repository: `http://archive.ics.uci.edu/ml/`

values wF^b indicate that the methods can reasonably reveal bigger groups but low wF^s show difficulties regarding the detection of smaller groups. Similarly, the Mclust (MC) cannot reveal small audio groups indicated by the lowest wF^s. In contrast, high wF^b indicate appropriate handling of bigger groups. The performance of Ward's method with the true number of groups (WT) seems to be also affected by the presence of strongly varying group sizes. Even though the method still generates homogeneous clusters (see H), slightly low wF^s indicate difficulties in identifying very small groups. Although WDB achieves the highest homogeneity (H) and weighted F-measure with respect to smaller groups (wF^s), the underlying clustering solutions are suboptimal due to the high number of final clusters leading to the lowest completeness (C) and consequently low V-measure (V). Surprisingly, Ward's method with the Gap statistic (WG) appears to reasonably detect both smaller and larger audio groups (see wF^s and wF^b).

The proposed IClust approach outperforms WG in terms of revealing smaller groups (see wF^s). In addition, IClust is capable of finding bigger groups as well (see wF^b) in comparison to methods achieving high homogeneity (H), such as WG and Affinity Propagation (AP).

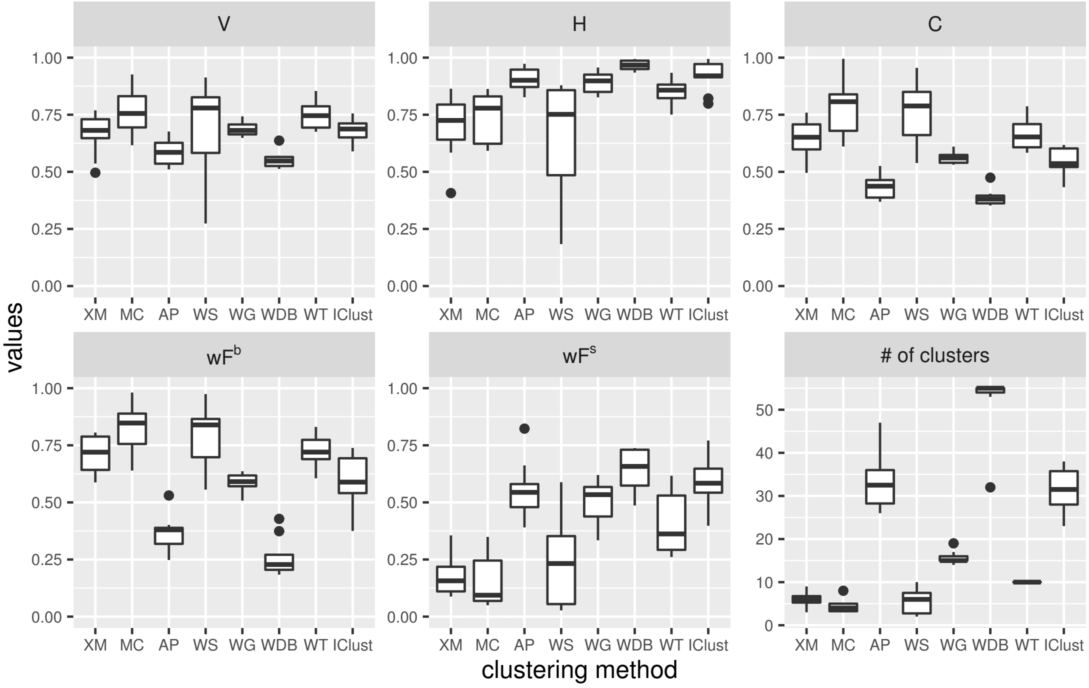

Figure 3.11: Clustering results on the audio data set in terms of V-measure (V), homogeneity (H), completeness (C), the weighted F-measures wF^b and wF^s with respect to big and small groups respectively, and the number (#) of detected clusters.

3.4.2 Comparison on the pen-based recognition data

The second experiment employs 10 imbalanced data sets constructed from the pen-based recognition data. Each setting contains 10 groups with 3 bigger groups of the sizes: $1000, 750, 500$, and 7 smaller groups of the sizes: $40, 30, 30, 20, 20, 10, 10$, resulting in a total sample size of 2380.

Figure 3.12 shows that Ward's method with the Silhouette Width (WS) and the Davis-Bouldin index (WDB) seem to have troubles to reveal small groups indicated by low wF^s. Moreover, the methods produced a considerably lower number of clusters leading to low homogeneity (H) scores. Although the model-based MC and x-means (XM) generate to some extent homogeneous clusters (see H), low wF^b as well as low wF^s demonstrate that the methods completely fail in detecting the considered media groups.

As expected, the proposed IClust method appears to identify both smaller and bigger groups indicated by high wF^s and wF^b. Although IClust produces a larger number of clusters than the true number of groups leading to low completeness (C), the methods outperform the remaining approach (i.e. WT, AP and WG) in terms of the V-measure (V).

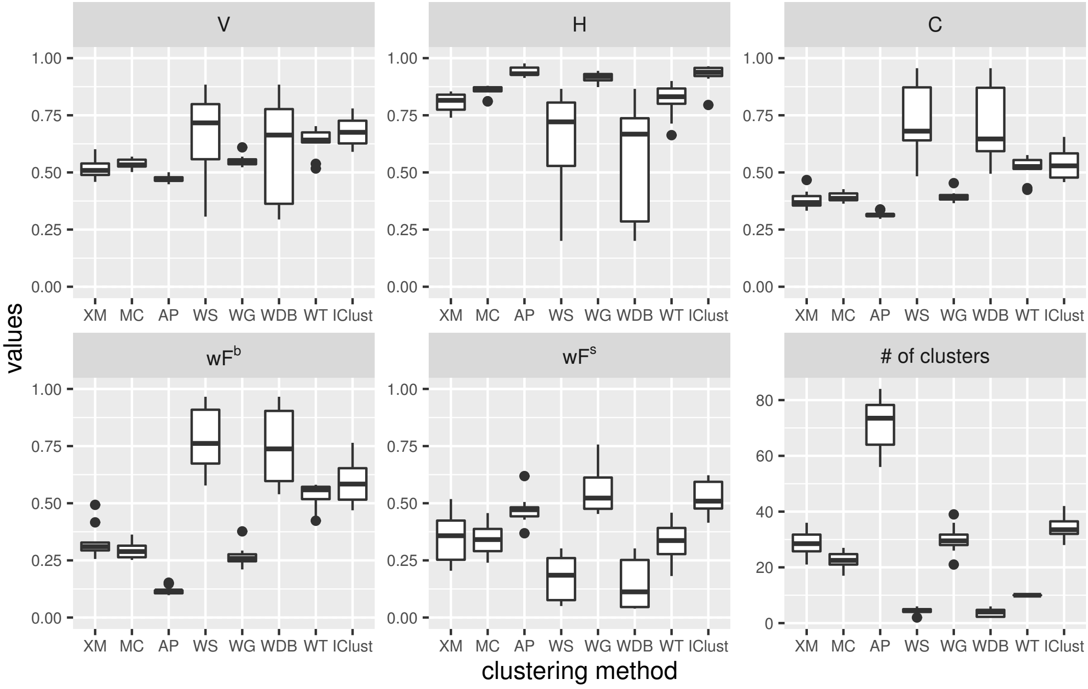

Figure 3.12: Clustering results on the pen-based recognition data set in terms of V-measure (V), homogeneity (H), completeness (C), the weighted F-measures wF^b and wF^s with respect to big and small groups respectively, and the number (#) of detected clusters.

3.4.3 Comparison on the human activity recognition data

The next comparison is applied on 10 imbalanced data sets randomly sampled from the human activity recognition data. Each setting consists of 6 groups with 3 bigger groups of the sizes: $200, 150, 100$, and 3 smaller groups of the sizes: $8, 6, 4$, resulting in a total sample size of 468.

Figure 3.13 shows that Ward's method with Silhouette Width (WS) and Davies-Bouldin (WDB) index have again difficulties in clustering imbalanced media groups as in the previous experiment. Similarly, the performance of x-means (XM) and the model-based MC appears to be violated by strongly varying group sizes indicated by low homogeneity (H). The performance of Ward's method (WT) seems to be also affected like in case of audio data indicated by low wF^s.

The proposed IClust algorithm is slightly worse than the best performing methods, such as AP and WG, in terms of generating homogeneous clusters (see H). The methods also detect more clusters than the true number of groups (cp. WT approach) leading to low completeness (C) and the V-measure (V). All three methods demonstrate the best performance regarding the detection of small groups (see wF^s).

3.4.4 Comparison on the satellite data

The last experiment compares the employed methods on 10 imbalanced data sets randomly generated from the satellite data. For each setting we sample 6 groups with 3 bigger groups of the sizes: $300, 225, 150$, and 3 smaller groups of the sizes: $12, 9, 3$, resulting in 669 observations in total.

Figure 3.14 shows that identifying media groups is again challenging for WS, WDB and MC indicated by low homogeneity (H). The detection of small groups is more problematic than revealing larger groups (see wF^s versus wF^b) for these methods. Surprisingly, x-means (XM) generates highly homogeneous clusters. However, this is caused by over-clustering the considered media data sets leading to the lowest completeness (C) and thus low V-measure (V). The low homogeneity (H) scores for WT demonstrate that a prior knowledge about the actual number of groups does not necessarily lead to a correct clustering solution in a highly imbalanced scenario.

The proposed IClust algorithm outperforms Affinity Propagation (AP) and the Ward's method with the Gap statistic (WG) with respect to handling bigger clusters in terms of wF^b. Regarding the detection of smaller groups, IClust demonstrates comparable performance (see wF^s).

3.5 Discussion and conclusions

We summarize our main findings based on the quality of the compared methods in terms of the weighted F-measures (wF^b and wF^s). Unlike conventional clustering evaluation

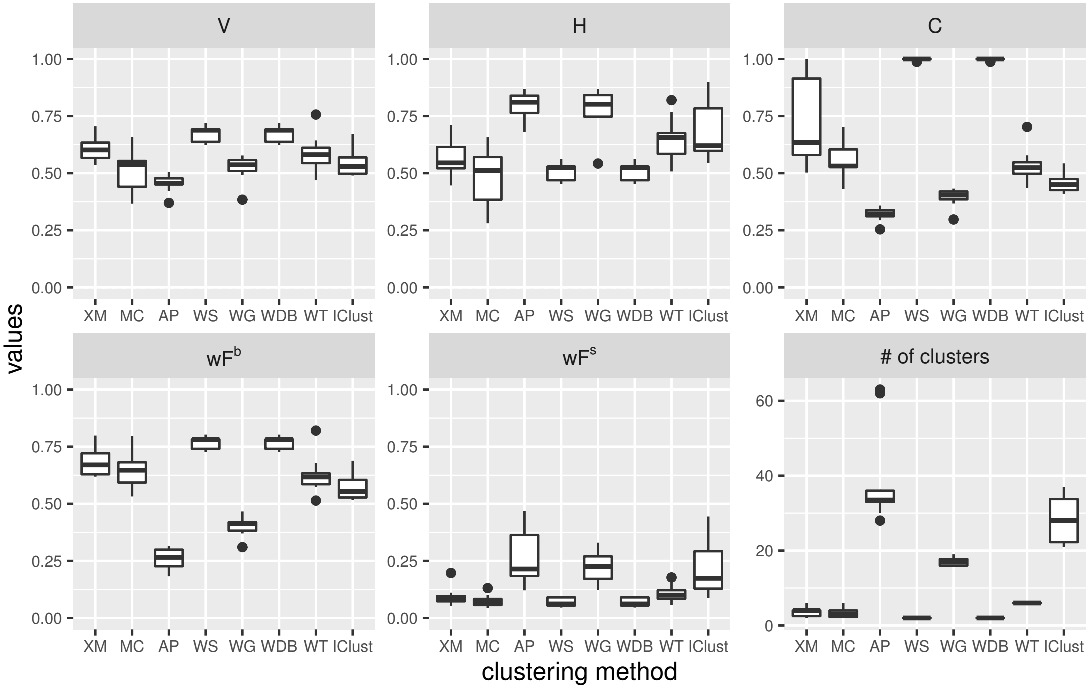

Figure 3.13: Clustering results on the human activity recognition data set in terms of V-measure (V), homogeneity (H), completeness (C), the weighted measures wF^b and wF^s with respect to big and small groups respectively, and the number (#) of detected clusters.

measures, the weighted F-measures allow to inspect the ability of clustering methods to identify big and small groups.

Figure 3.15 shows the obtained results in terms of the median performance of wF^b and wF^s achieved during the previously performed experiments. Overall, there is no clear dependence between the performance of the methods and the sizes of employed data sets. However, the results indicate a dependency between identifying bigger and smaller groups. The methods capable of identifying big groups, such as WS and WDB, show considerably weaker ability to detect small groups. This supports the fact that centroid-based methods as well as validity indexes might not be suitable for clustering imbalanced high-dimensional data. Similarly, the model-based MC demonstrates poor performance in terms of finding smaller groups. On the opposite, Affinity Propagation (AP), which does not assume any specific cluster characteristics, appears to be among the best performing methods in terms of revealing small groups (see dark shades of gray in Figure 3.15, right). However, the method turns out to poorly detect bigger groups (see light shades of gray in Figure 3.15, left).

In contrast to all investigated method, the proposed IClust approach shows the best

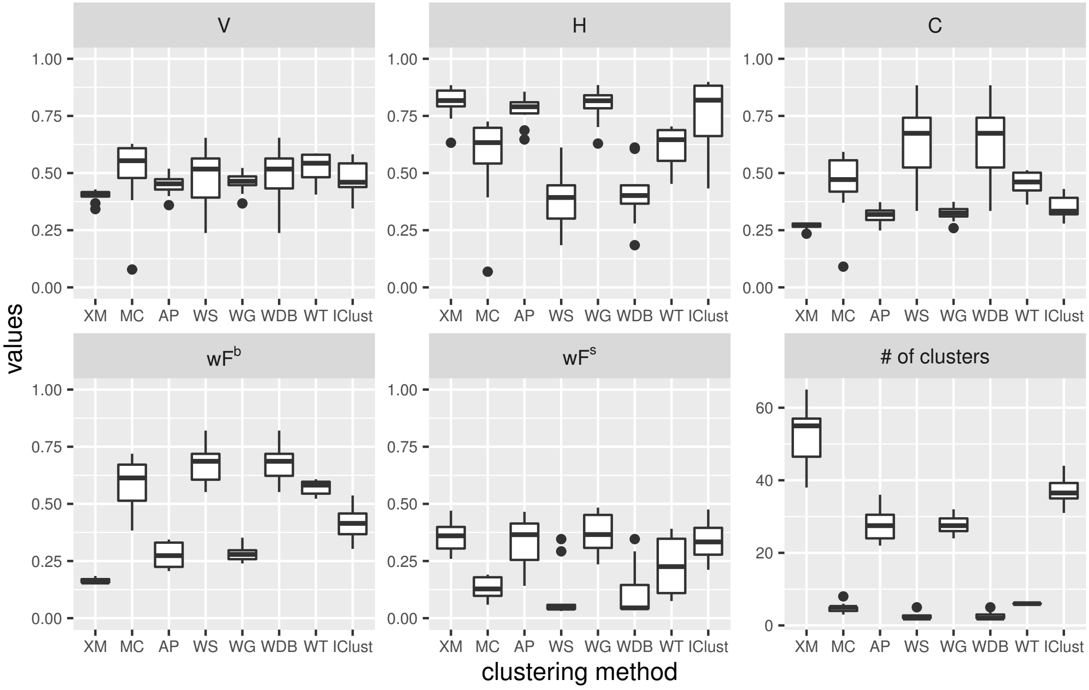

Figure 3.14: Clustering results on the satellite data set in terms of V-measure (V), homogeneity (H), completeness (C), the weighted measures wF^b and wF^s with respect to big and small groups respectively, and the number (#) of detected clusters.

performance in terms of finding small groups and, in addition, the method can still reasonably identify bigger groups (see dark shades of gray in Figure 3.15, left).

Although any real-world data set exhibits a multiple group structure, it may be difficult to determine, whether or not the groups are of different sizes if there is no prior knowledge available. This leads to the question if the proposed IClust algorithm can identify groups of approximately the same sizes (balanced data setting). For this reason, we perform an additional experiment on the balanced pen-based recognition data set which was used to evaluate various clustering methods for high-dimensional data (Müller et al., 2009). Müller et al. (2009) considered 10 groups of similar sizes resulting in a total size of 7494. Table 3.3 presents the performance of the compared methods in terms of the conventional evaluation measures. Although IClust shows slightly worse performance than the best performing XM, MC, AP, and WG in terms of purity (P) and homogeneity (H), the proposed method partitions the data set into a lower number of clusters. This is indicated by higher completeness (C) as well as higher V-measure (V). In addition, the F-measure suggests that IClust may also be used to reveal a group structure in a balanced data setting.

The proposed method also has limitations. First, IClust takes four input parameters.

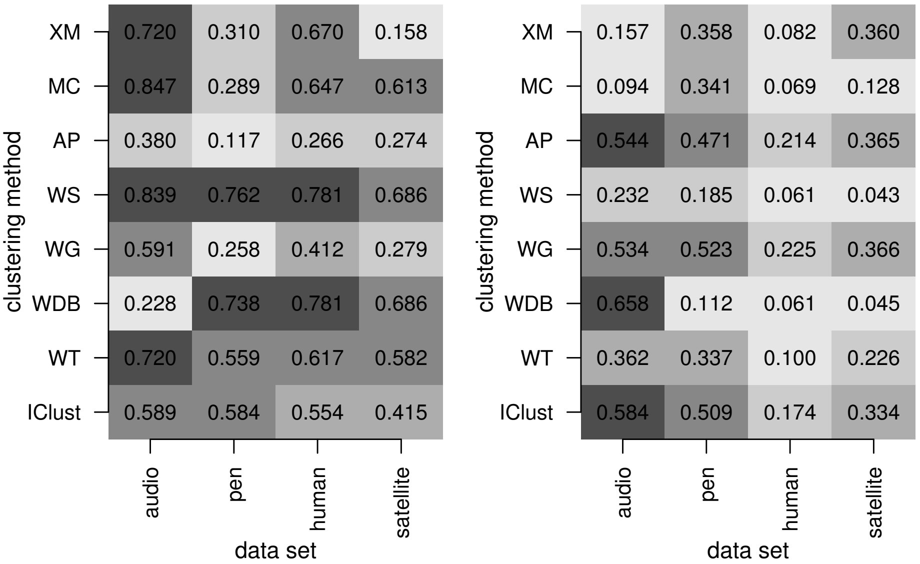

Figure 3.15: Summary of all employed experiments in terms of the weighted F-measures wF^b (left) and wF^s (right) with respect to big and small groups. Dark shades of gray indicate good performance and light shades of gray indicate poor performance.

Table 3.3: Clustering quality of the compared methods on the balanced pen-based recognition data set with respect to purity (P), V-measure (V), homogeneity (H), completeness (C), F-measure (F), and the number (#) of clusters.

	P	V	H	C	F	# of clusters
XM	0.916	0.653	0.881	0.518	0.496	79
MC	0.937	0.670	0.930	0.523	0.428	72
AP	0.981	0.615	0.971	0.450	0.218	165
WS	0.795	0.767	0.774	0.760	0.757	12
WG	0.970	0.700	0.959	0.551	0.422	67
WDB	0.626	0.693	0.635	0.763	0.663	8
WT	0.750	0.759	0.739	0.780	0.736	10
IClust	0.823	0.769	0.862	0.693	0.673	35

Although we provided a thorough empirical study to select optimal parameters, the parameter setting may be tuned for other data sets. Second, IClust tends to generate a larger number of clusters than the actual number of groups in the data. This might be due to the estimation of critical values employed in the merging procedure. A possible solution for improvement could be either to adjust the critical values to the size of the clusters which are merged during the procedure or to incorporate different robust counterparts to arithmetic mean and standard deviation. Despite the mentioned limitations, the proposed IClust algorithm exhibits also some advantages over existing methods. IClust does not require a pre-specification of the number of final clusters. This algorithm also does not assume any specific cluster and data characteristics. Moreover, the experiments demonstrated that the choice of parameters seems to be reasonable in both imbalanced and balanced scenarios. This indicates that IClust is a useful clustering method for media data, and it is a promising method also for other application domains. The R implementation of the algorithm is also freely available at `https://github.com/brodsa/IClust`.

Acknowledgments

This work has been partly funded by the Vienna Science and Technology Fund (WWTF) through project ICT12-010 and by the K-project DEXHELPP through COMET - Competence Centers for Excellent Technologies, supported by BMVIT, BMWFW and the province Vienna. The COMET program is administrated by FFG.

CHAPTER 4

Robust and sparse k-means clustering for high-dimensional data

Abstract: In real-world application scenarios, the identification of groups poses a significant challenge due to possibly occurring outliers and existing noise variables. Therefore, there is a need for a clustering method which is capable of revealing the group structure in data containing both outliers and noise variables without any pre-knowledge. In this paper, we propose a k-means-based algorithm incorporating a weighting function which leads to an automatic weight assignment for each observation. In order to cope with noise variables, a lasso-type penalty is used in an objective function adjusted by observation weights. We finally introduce a framework for selecting both the number of clusters and variables based on a modified gap statistic. The conducted experiments on simulated and real-world data demonstrate the advantage of the method to identify groups, outliers, and informative variables simultaneously.

Key words: Clusters; Outliers; Noise variables; High-dimensions; Gap statistic

Co-authors: Peter Filzmoser, Thomas Ortner, Christian Breiteneder, Maia Zaharieva

4.1 Introduction

The identification of groups in real-world high-dimensional datasets reveals challenges due to several aspects: 1) the presence of outliers; 2) the presence of noise variables; 3) the selection of proper parameters for the clustering procedure, e.g. the number of clusters. Whereas we have found a lot of work addressing the three aspects separately, a much smaller number of studies is available in case all three aspects are treated simultaneously. Indeed, in any large and high-dimensional complex dataset, not only outliers but also noise variables are very likely to appear. Hence, a clustering method needs to be designed in such a way that both aspects are taken into account, no matter if outliers are considered as highly interesting observations due to their typically different content or just as noise. The data complexity in terms of the number of groups and the proportion of outliers as well as the number of noise variables very much depends on the dataset itself. Therefore, a clustering procedure should ideally be data-independent. In other words, no information about the data complexity should be assumed. The goal of this paper is to introduce a clustering method designed for such an application scenario.

Considering the task of revealing the group structure in contaminated data, i.e. data with outliers, a natural step is to first apply an outlier detection procedure to exclude deviating observations for the following cluster analysis. However, coping with outliers in such a way might be complicated due to the parameter specification, which is commonly required by most existing clustering (e.g. the number of clusters) as well as outlier detection methods (Aggarwal, 2013). A better alternative is to use a clustering method which directly incorporates a measure of outlyingness through data clustering in order to reveal clusters and outliers simultaneously as proposed by Campello et al. (2015). Another possibility to deal with outliers in the context of clustering is to exclude a certain proportion of deviating observations while applying a clustering method. The idea of excluding observations, which usually do not fit to an assumed model, lead to so-called trimming-based clustering approaches. An overview of such methods can be found in the review by García-Escudero et al. (2010). In order to apply a trimming concept, not only the number of clusters but also the trimming level, i.e. the proportion of observations supposed to be discarded, need to be specified in advance. Although García-Escudero et al. (2011) introduce a diagnostic plot for selecting both parameters using classification trimmed likelihood curves, the procedure depends on the choice of a data-dependent parameter that controls the way how potential outliers should be handled. Determining such a parameter might however be again difficult for real-world data.

The problem of data clustering in the presence of noise variables is usually addressed by sparse- and variable selection-based clustering approaches. The methods generally aim at removing noise variables that can easily mask a group structure (Gordon, 1999). An overview of such methods can be found in the study by Galimberti et al. (2017) with a special focus on model-based clustering. Although the number of clusters in model-based clustering is commonly estimated based on the Bayesian information criterion, the methods usually assume that the size of a group is typically larger than the dimensionality of the data space where a group is located. Therefore, such approaches might have troubles to sufficiently discover groups when a large number of noise variables exist, which may lead to high-dimensional low sample size groups. A suitable method for such a situation is introduced by Witten and Tibshirani (2010). The method imposes a lasso-type penalty on incorporated variable weights in the objective function of k-means leading to the sparse k-means algorithm. In order to apply the sparse k-means, the number of clusters needs to be determined in advance, which is hardly possible for most real-world application scenarios.

The task of identifying groups becomes even more problematic when both outliers and noise variables are present. For this situation, Kondo et al. (2012) introduce the robust and sparse k-means (RSKC) that robustifies the sparse k-means by Witten and Tibshirani (2010) by incorporating a trimming concept. However, the approach assumes prior knowledge about the number of clusters, the degree of sparsity, and the trimming level in order to correctly detect clusters. Furthermore, the method has been tested only in terms of clustering and no evaluation has been performed regarding the detection of outliers. Such observations may additionally provide useful information about the analyzed datasets since they usually differ from the main group structure.

In contrast to RSKC, we introduce a robust and sparse k-means-based procedure that is capable of finding the true underlying structure in very complex data, i.e. data containing clusters, outliers, and noise variables simultaneously. The presented k-means-based algorithm incorporates a weighting function employing a measure of outlyingness in order to automatically assign a weight to each observation. While a high weight indicates that an observation is part of a cluster, a low weight refers to a potential outlier. The advantage of using a weighting function is that we do not have to pre-specify any trimming level as for trimming-based approaches. To exclude noise variables, we use a lasso-type penalty imposed on the variable weights in an objective function adjusted by observation weights. In order to correctly detect groups, we eventually propose a framework aiming at the determination of the optimal parameters, such as the degree of sparsity and the number of clusters.

The rest of this paper is organized as follows. Section 4.2 briefly reviews k-means-based clustering approaches and motivates the proposed clustering procedure which is described in detail in Section 4.3. The parameter selection is presented in Section 4.4 and thoroughly tested on simulated data sets in Section 4.6. We compare the proposed method with other k-means-based clustering methods on a real-world dataset in Section 4.7. Section 4.8 concludes the paper.

4.2 k-means-based algorithms

Despite the large number of developed clustering procedures, k-means remains one of the most popular and simplest partition algorithms (Jain, 2010). Given a data matrix $\mathbf{X} = \{x_{ij}\}, i = 1, \ldots, n$, $j = 1, \ldots, p$, with n observations described by p variables, the task of finding k clusters based on k-means was originally established using the within-cluster sum of squares W^k for the given number of clusters k as

$$W_j^k = \sum_{r=1}^{k} \sum_{i \in K_r} (x_{ij} - m_{jr})^2, \quad W^k = \sum_{j=1}^{p} W_j^k \to \min_{K_1,\dots,K_k}, \tag{4.1}$$

where W_j^k corresponds to the within-cluster sum of squares in the j^{th} variable and the set K_r contains the indices of the observations assigned to the r^{th} cluster, for $r = 1, \dots, k$. Note that such an optimization problem can also be reformulated with respect to the between-cluster sum of squares B^k (Witten and Tibshirani, 2010) as

$$B_j^k = \sum_{i=1}^{n} (x_{ij} - m_j)^2 - \sum_{r=1}^{k} \sum_{i \in K_r} (x_{ij} - m_{jr})^2, \quad B^k = \sum_{j=1}^{p} B_j^k \to \max_{K_1,\dots,K_k}, \tag{4.2}$$

where B_j^k denotes B^k in the j^{th} variable, m_j is the j^{th} coordinate of the overall data center, and m_{jr} denotes the center of the r^{th} cluster in the j^{th} variable.

Although k-means is very popular, it has several disadvantages that need to be taken into account when developing a clustering procedure. The first drawback of k-means is the random initialization of cluster centers, which may lead to non-optimal solutions. This can be overcome by using an appropriate initialization method; an overview of such approaches can be found in a study by Celebi et al. (2013). For our method, we incorporate the ROBIN (ROBust INitialization) approach by Mohammad et al. (2009). The method is able to find optimal centers in a small number of runs unlike the original k-means. ROBIN seeks for initial centers that are located in the most dense region and are simultaneously far away from each other in order to avoid the selection of outliers as initial centers. In order to identify the observations in highly dense regions, ROBIN uses LOF (Local Outlier Factor) proposed by Breunig et al. (2000). LOF was primarily introduced to measure a degree of outlyingness of observations in complex data where observations tend to form groups. The method compares local densities of observations with the local densities of their q nearest neighbors using various ratios of the Euclidean distances. The resulting outlyingness, $lof_q(\mathbf{x}_i)$, of an observation $\mathbf{x}_i$ close to 1 indicates that $\mathbf{x}_i$ is potentially part of a cluster and, therefore, a candidate for an initial cluster center, as proposed by ROBIN. In contrast, $lof_q(\mathbf{x}_i) \gg 1$ suggests that $\mathbf{x}_i$ is a possible outlier and thus $\mathbf{x}_i$ should not be considered as an initial center.

The second limitation of k-means is the employed sample mean that suffers from a lack of robustness. As a result, k-means is also not resistant against outliers and even a single deviating observation can affect the final clustering solution (García-Escudero and Gordaliza, 1999). In order to robustify k-means, Cuesta-Albertos et al. (1997) proposed a trimmed version defined as

$$^tB_j^k = \sum_{i \in L} (x_{ij} - m_j)^2 - \sum_{r=1}^{k} \sum_{i \in K_r \cap L} (x_{ij} - m_{jr})^2, \quad ^tB^k = \sum_{j=1}^{p} {}^tB_j^k \to \max_{K_1,\dots,K_k,L}, \tag{4.3}$$

where $^tB^k = \sum_{j=1}^{p} {}^tB_j^k$ represents the between-cluster sum of squares calculated on the untrimmed observations, L denotes the set containing indices of $[n(1-\alpha)]$ (untrimmed) observations that have the smallest distance to their closest cluster center, and α is the trimming level. Such a robustification excludes the α fraction of observations, i.e. potential outliers, for calculating the cluster centers in order to achieve an accurate clustering solution if α is chosen correctly according to the true outlier proportion. Determining α may however be problematic for real-world data. In order to avoid the parameter-dependent robust k-means, we propose to incorporate a measurement of outlyingness which leads to a clear decision on determining outliers. Such a concept was introduced by Filzmoser et al. (2008) in case of a one group data structure resulting in observation weights. The weights reflect how much an observation is outlying on the $[0, 1]$-scale with a low weight indicating a potential outlier. We incorporate the concept of observation weights in k-means in order to robustify the method in such a way that no parameter pre-specification is required.

The last disadvantage of k-means occurs when a group structure is detectable only in a small subset of variables. In order to find such variables, Witten and Tibshirani (2010) introduced a framework for sparse k-means based on a lasso-type penalty leading to the problem of maximizing the weighted B^k for a given k and a sparsity parameter l as

$$B^{lk} = \sum_{j=1}^{p} w_j B_j^k \to \max_{K_1,\ldots,K_k,\mathbf{w}}, \tag{4.4}$$

subject to $||\mathbf{w}||^2 \leq 1, ||\mathbf{w}||_1 \leq l$ for $\mathbf{w} = \{w_j \geq 0\} \, \forall j$ and $l \in (1, \sqrt{p}]$, which can be solved in an iterative way as proposed by Witten and Tibshirani (2010). The parameter l controls the degree of sparsity in the variable weight vector, i.e. the values of w_j. The more important (informative) the j^{th} variable, the higher the value of w_j. Our method also uses a lasso-type penalty in the objective function, but the value of B^{lk} is additionally adjusted by observation weights in order to achieve robustness. Although, the proposed method is similar to RSKC by Kondo et al. (2012), our procedure can be seen as a better alternative since no trimming level is required. In addition, we aim at analyzing the data structure more thoroughly, i.e. discovering clusters, outliers, and informative variables simultaneously.

4.3 Proposed algorithm

The introduced method is an iterative three-step approach. In the first step, k-means employing a weighting function is applied on the data space spanned by the variables with some contribution to a cluster separation (i.e. with the variables having $w_j > 0$, see Equation (4.4)). The incorporated weighting function robustifies k-means and results in observation weights reflecting the outlyingness. The second step aims at updating the variable weights with respect to both clusters and observation weights from the first step. The two steps are iteratively repeated until the variable weights stabilize. In the third step, the observations are clustered with respect to the identified informative variables and the observations with small weights are classified as outliers. The detailed description of the algorithm is given in the following subsections.

4.3.1 Step 1: Downweighting outlying observations

The aim of the first step is to robustify k-means by incorporating a weighing function in order to downweight the influence of potential outliers. Assuming that the number of clusters k is known, we apply ROBIN with $q = 10$ (Mohammad et al., 2009) on weighted data, ${}_w\mathbf{X} = \{{}_w\mathbf{x}_i\} = \{w_j\, x_{ij}\}, \forall i, \forall j$, where $\mathbf{w} = \{w_j = 1/\sqrt{p}\}, \forall j$, in order to find the first k cluster centers. Note that these initial values for w_j are considered only in the first iteration as recommended by Witten and Tibshirani (2010), but in the next iteration w_j will be already different and will better reflect the contribution to a cluster separation.

After applying ROBIN, each observation is assigned to its closest cluster center leading to the corresponding cluster membership $K_1, \dots, K_k$. We then propose to apply a weighting function on the detected clusters to reveal outliers. The weighting function should be a monotonic decreasing function using an outlyingness measure as an argument in order to obtain observation weights that range between 0 and 1, with a low weight indicating a potential outlier. Hence, it is essential to choose both a suitable outlyingness measure and an appropriate weighting function.

A naive approach is to calculate the Euclidean distance of an observation to its closest cluster center. However, using the Euclidean distance provides the information about how far an observation is from its closest center rather than how much an observation deviates or to what degree it is an outlier. In fact, such information can be easily obtained by applying LOF on each detected cluster as $lof({}_w\mathbf{x}_i) := lof_q({}_w\mathbf{x}_i), i \in K_r$, $\forall r$, with $q = 10$ as recommended by Breunig et al. (2000); Mohammad et al. (2009).

The LOF scores are then standardized as

$$lof_i^* = \frac{lof({}_w\mathbf{x}_i) - \operatorname{mean}(lof({}_w\mathbf{x}_i), i \in K_r)}{\operatorname{sd}(lof({}_w\mathbf{x}_i), i \in K_r)} \tag{4.5}$$

to be suitable for the weighting function with the mentioned properties. Preliminary studies indicated good empirical results when the observation weights, denoted as $v_i^{(1)}$, were obtained using the translated bi-weight function (Rocke, 1996) as follows

$$v_i^{(1)} = \begin{cases} 0, & lof_i^* \geq c \\ \left(1 - \left(\frac{lof_i^* - M}{c - M}\right)^2\right)^2, & M < lof_i^* < c, \\ 1, & lof_i^* \leq M \end{cases} \tag{4.6}$$

where $i \in K_r, \forall r$, $M = \operatorname{med}(lof_i^*, i \in K_r) + \operatorname{MAD}(lof_i^*, i \in K_r)$, and $c = 2$. The obtained weights correspond to the measure of outlyingness values in $[0, 1]$. While a value close to 1 indicates that an observation is part of a cluster, $v_i^{(1)} \approx 0$ suggests that $\mathbf{x}_i$ is an outlier with respect to the detected cluster. The weights based on LOF better express the degree of deviation than e.g. using a simple Euclidean distance of an observation to the closest cluster as in RSKC. In addition, the weights should be more robust against elliptically-shaped clusters due to the properties of LOF; see Breunig et al. (2000). If the shape of a cluster is slightly elliptical, RSKC might exclude observations which are further away from the cluster center but still part of a cluster. After assigning weights to observations from each detected cluster according to Equation (4.5) and (4.6), we plug the weights $v_i^{(1)}$ into the weighted between-cluster sum of squares for a given $\mathbf{w}$, and optimize the cluster assignment as

$$
\begin{aligned}
{}^{v^{(1)}}B_j^k &= \sum_{i=1}^{n} v_i^{(1)}\Big(x_{ij} - \frac{1}{\sum_{i=1}^{n} v_i^{(1)}} \sum_{i=1}^{n} v_i^{(1)} x_{ij}\Big)^2 \\
&- \sum_{r=1}^{k} \sum_{i \in K_r} v_i^{(1)}\Big(x_{ij} - \frac{1}{\sum_{i \in K_r} v_i^{(1)}} \sum_{i \in K_r} v_i^{(1)} x_{ij}\Big)^2
\end{aligned}
\tag{4.7}
$$

$$
\sum_{j=1}^{p} w_j \, {}^{v^{(1)}}B_j^k \to \max_{K_1,\dots,K_k} . \tag{4.8}
$$

in order to robustify k-means. We can clearly see from Equation (4.7) that if an observation is a potential outlier, i.e $v_i^{(1)} \approx 0$, the distance of such an observation to its closest cluster center is downweighted by the corresponding value of $v_i^{(1)}$. In contrast, an observation with $v_i^{(1)} \approx 1$ highly contributes to the maximization. The observation weights are also used to determine the next cluster centers, i.e. $\frac{1}{\sum_{i \in K_r} v_i^{(1)}} \sum_{i \in K_r} v_i^{(1)} x_{ij}, \forall r$, in a robust way by using the weighted mean of observations in each coordinate. The cluster centers with the corresponding cluster assignment are iteratively updated until a local optimum is reached during a certain number of iterations in the sense of maximization of Equation (4.8). In our experiments the method is allowed to search for the local optimum during 15 iterations, but also a higher number can be considered. Note that the local optimum is achieved on the weighted data, i.e. in a data space spanned by the variable vector with $w_j > 0$ adjusted by the values of w_j.

We illustrate the efficiency of the weighting function on an example dataset that consists of three groups with the same sizes of 40 observations. The group structure is described by 50 variables leading to high-dimensional low sample size groups. We add 750 noise variables and contaminate 10% of the observations from each group in the informative variables and in 75 noise variables; a detailed description of the data setup is provided in Section 4.6 and corresponds to the first simulation study. Figure 4.1 visualizes the generated dataset in the space spanned by the first two principal components; the group membership and outliers are differentiated by colors and symbols. The final weights, obtained during two iterations given the initial cluster centers by ROBIN, are shown in Figure 4.2 in decreasing order to visualize the shape of the weighting function. Importantly, the observation weights are calculated in the data space defined by 50 informative variables. In other words, we now assume that $\mathbf{w}$ is known beforehand in order to demonstrate the concept of the weighting function. We can see in Figure 4.2 that all observations from group 3 are correctly assigned to cluster 1 because no observations from group 3 are visible in the following plots. The plot particularly indicates that the weighting function works properly since all non-outliers obtain a weight around 1. In contrast, outliers placed in informative variables receive a weight around 0 and can thus be easily identified. A similar conclusion can be made in case of the other two clusters; see Figure 4.2 and Figure 4.2. The plots may suggest that the non-outliers with a weight smaller than 1 could be located on the edge of a cluster or slightly further from the other clustered observations. However, we cannot reveal outliers placed in noise variables as indicated by their weights equal to 1, since the noise variables are not involved in the clustering due to their zero weights.

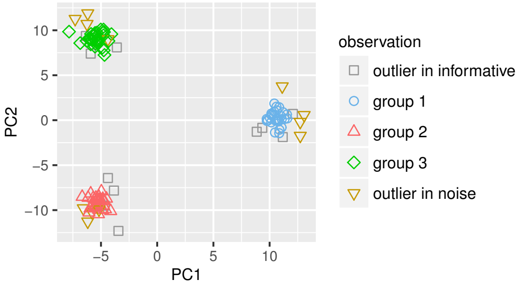

Figure 4.1: The generated dataset shown in the principal component space. The observations from 3 groups, outliers placed in informative and noise variables are displayed in different colors and symbols.

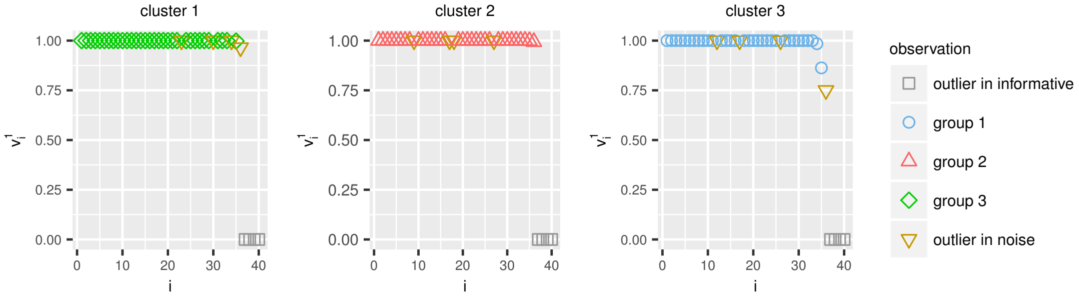

Figure 4.2: Illustration of incorporating the weighting function in k-means in order to reveal outliers as observations with low $v_i^{(1)}$ and to detect 3 clusters on the weighted data.

In order to identify outliers in noise variables as well, we additionally apply the proposed weighting function on unweighted data clusters, consisting of the data matrices $\mathbf{X}_r = \{x_{ij}\}, i \in K_r, \forall j = 1, r = 1, \ldots, k$, leading to the second observation weights $v_i^{(2)}$. Figure 4.3 shows the second resulting observation weights obtained on the data example shown in Figure 4.1. The three plots clearly indicate that all outliers placed in noise variables receive considerably lower weights in contrast to both non-outliers and outliers present in informative variables.

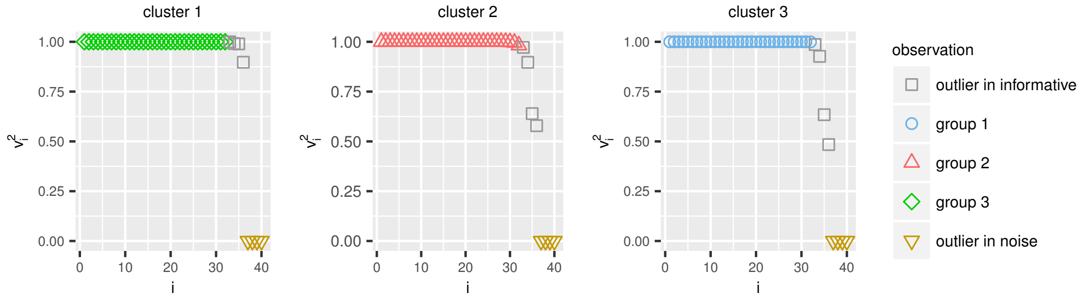

Figure 4.3: Illustration of applying the weighting function on the 3 unweighted data clusters in order to reveal outliers in noise variables as observations with low $v_i^{(2)}$.

As a consequence of applying the weighting function for the second time, each observation has two weights, $v_i^{(1)}$ and $v_i^{(2)}$, which are finally combined in a single weight

$$v_i = \min\{v_i^{(1)}, v_i^{(2)}\}. \tag{4.9}$$

Determining v_i in this way ensures that all outliers receive low weights and that we can easily identify whether or not an observation is an outlier as indicated by zero weights for all outliers in Figure 4.4.

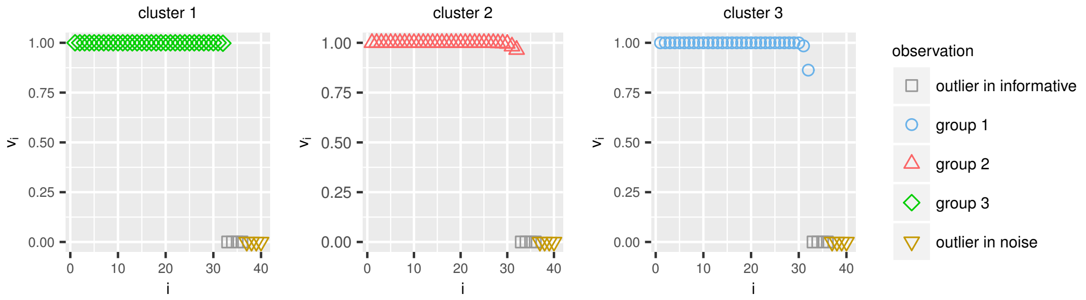

Figure 4.4: Illustration of combining the two observations weights leading to the final observation weights v_i calculated on 3 clusters.

4.3.2 Step 2: Variable selection

The purpose of the second step is to update w_j according to the maximization of Equation(4.4) for a given sparsity parameter l and the observation weights v_i. Incorporating v_i assures that the variable selection is not affected by outliers. Indeed, the presence of an outlier apparent even in one variable can considerably increase the between-cluster sum of squares. As a result, such a variable receives a high weight although the variable does not contribute to the cluster separation but rather to the separation between an outlier and a cluster (Kondo et al., 2012).

Therefore, for the obtained cluster assignment $K_1, \ldots, K_k$, the observation weights v_i from the first step, and for a given l, we update the weights w_j according to

$$ {}^{v}B^{kl} = \sum_{j=1}^{p} w_j \, {}^{v}B_j^k \rightarrow \max_{||\mathbf{w}||^2 \leq 1, ||\mathbf{w}||_1 \leq l}, \tag{4.10} $$

where ${}^{v}B_j^k$ corresponds to Equation (4.7) with v_i instead. In order to optimize Equation (4.10) with respect to $\mathbf{w}$ for a given tuning parameter l, we follow the procedure suggested by Witten and Tibshirani (2010). Whereas small l leads to high sparsity, i.e. $w_j = 0$ for most variables, a high value of l results in almost no sparsity corresponding to $w_j > 0$ for most variables. High w_j suggests that the j^{th} variable is informative and, thus, it contributes to the maximization of Equation (4.10). In contrast, $w_j = 0$ indicates that the j^{th} variable is not informative for the cluster separation and it is thus excluded in Equation (4.10).

Once the variable weights $\mathbf{w}$ are updated, the first iteration is completed and the algorithm continues with the first step with respect to updated weights w_j. This means that the ROBIN approach is again applied on ${}_{w}\mathbf{X} = \{{}_{w}\mathbf{x}_i\}$ with updated $\mathbf{w}$ in order to find the next cluster centers. The reason for the re-initialization is that ROBIN is not primarily designed to deal with a large number of noise variables. Therefore, the selection of the first cluster centers is very likely to be affected by noise variables due to $\mathbf{w} = \{w_j = 1/\sqrt{p}\}, \forall j$ in the first step. After obtaining the next centers, the method continues as described. The two steps of the proposed approach are iteratively repeated until convergence for w_j is reached according to the stopping criterion (Witten and Tibshirani, 2010).

4.3.3 Step 3: Detection of groups and outliers

The last step aims at determining the cluster membership $K_1, \ldots, K_k$ by assigning observations to their closest cluster center in the data space spanned by variables with $w_j > 0$, adjusted by their corresponding final weights. We estimate the final observation weights v_i as described in Section 4.3.1 in order to classify observations with low weights as outliers. This classification can be made based on visualization of the resulting observation weights against the corresponding observation index, as shown in Figure 4.4, and the following search for a cut-off value which clearly separates low weights from high weights. Nevertheless, we recommend to use $v_i < 0.5$ for the identification of outliers as we observed good empirical results for such a choice.

4.4 Selection of parameters

We have so far assumed pre-knowledge about the number of clusters k as well as the tuning parameter l determining the variable weights w_j. Such information is usually not available beforehand for most real-world data and, therefore, there is a need for a systematic way of estimating both parameters. The problem of selecting the optimal k has been widely studied for data where the assumption is that all variables are involved in data clustering; an overview of such procedures can be found in the studies by Sugar and James (2003) and Xu and Wunsch (2005). However, we have not found much work dedicated to the optimization of k in case that the sizes of groups are much lower than the dimensionality of the data space describing the group structure and at the same time the group structure is hidden in a large number of noise variables.

We discuss the effect when k is optimized with and without taking the contribution of variables into account using the gap statistic (Tibshirani et al., 2001). The gap statistic, Gap_k, is calculated for a clustering solution obtained by a clustering algorithm, e.g. k-means, for a given k and can be formulated as

$$Gap_k = \sum_{j=1}^{p} w_j \Big(\frac{1}{A} \sum_{a=1}^{A} \log({}_aW_j^k) - \log(W_j^k) \Big), \tag{4.11}$$

where $w_j = 1, \forall j$ since all variables are assumed to contribute equally, ${}_aW^k = \sum_j {}_aW_j^k$ corresponds to $W^k = \sum_j W_j^k$ calculated on the clustering solution obtained on the dataset with independently permuted observations in each variable (Witten and Tibshirani, 2010), and A represents the number of permuted datasets. In our experiments we consider $A = 10$. Gap_k is generally calculated for a clustering solution with varying k and the optimal number of clusters is chosen as the smallest k for which $Gap_k \geq Gap_{k+1} - se_{k+1}$ is fulfilled (Tibshirani et al., 2001), where se_k denotes the standard error of $\log({}_aW^k)$. From Equation (4.11) it is obvious that Gap_k does not only depend on k but also on $\mathbf{w}$ representing the contribution of each variable. Since all variables are assumed to be informative, Gap_k might be considerably affected if a dataset contains a large number of noise variables. Moreover, the presence of deviating observations can lead to an unreliable decision on k as well.

Figure 4.5 demonstrates the effect of noise variables and outliers on the choice of k based on the gap statistic. We consider the same data example as in Section 4.3.3 and apply k-means with ROBIN initialization for the numbers of clusters $k = 2, \ldots, 6$. The gap statistic is calculated for each clustering solution in order to select the optimal k as described above. Figure 4.5 (left) shows the values of Gap_k with the corresponding standard errors calculated on the data example with both outliers and noise variables. As expected, the presence of both disturbing factors leads to a wrong choice of the optimal k corresponding to 5 clusters. Moreover, even if only the 50 informative variables are taken into account, the choice of k is also influenced by outliers as illustrated in Figure 4.5 (middle) resulting in $k = 4$. In contrast, Figure 4.5 (right) shows Gap_k when downweighting outlying observations and noise variables leading to a correct decision, i.e $k = 3$.

The example indicates that both disturbing factors have to be considered when selecting an optimal k for k-means. In the proposed k-means-based clustering approach, we directly downweight the effect of outliers by observation weights v_i. However, the impact of noise

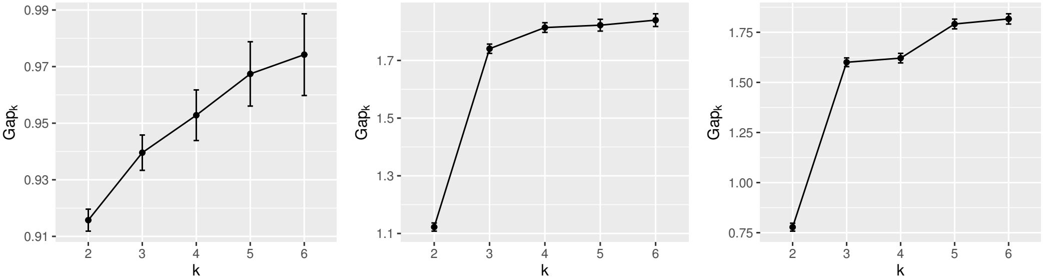

Figure 4.5: The effect of noise variables and outliers when estimating the optimal number of clusters. The values of Gap_k applied on a data set with both noise variables and outliers (left); the resulting Gap_k from a dataset where the effect of noise variables is eliminated (middle); the obtained Gap_k when both noise variables and outliers are neglected (right).

variables, which is reflected by their corresponding variable weights w_j, can be neglected only if the sparsity parameter l, see Equation (4.10), is correctly selected. In order to optimize l, Witten and Tibshirani (2010) introduced the gap statistic Gap_l, which is defined for given k as

$$Gap_l = \log(B^{lk}) - \frac{1}{A}\sum_{a=1}^{A}\log({}_aB^{lk}), \tag{4.12}$$

where ${}_aB^{lk}$ denotes the weighted between-cluster sum of squares calculated, compare Equation (4.4), with respect to a clustering solution obtained on a permuted dataset. Obviously, the calculation of Gap_l is impossible if the number of clusters k is unknown which is often the case for real data. Moreover, the presence of outliers might also influence the correct estimation of l. Therefore, we propose to adjust Gap_l by observation weights v_i in order to downweight the influence of outliers leading to the modified gap statistic ${}^vGap_{lk}$ calculated as

$${}^vGap_{lk} = \log({}^vB^{lk}) - \frac{1}{A}\sum_{a=1}^{A}\log({}^v_aB^{lk}), \tag{4.13}$$

where ${}^v_aB^{lk}$ represents ${}^vB^{lk}$ obtained on a permuted dataset. We calculate ${}^vGap_{lk}$ for a clustering solution not only with various l but also various k in order to first optimize the degree of sparsity l for each k. The value of ${}^vGap_{l^*k}$ for the optimal parameter l^* is compared with the largest ${}^vGap_{lk}$ such that ${}^vGap_{l^*k} \geq {}^vGap_{lk} - se_{lk}$, where se_k refers to the standard error of $\log({}^v_aB^{lk})$. The optimization of l leads to k values of ${}^vGap_{l^*k}$ for which the largest value corresponds to an optimal k.

Figure 4.6 (left) depicts the gap statistic for both tuning parameters when applying the proposed method on the data example in Section 4.3.1 with $k = 2, \ldots, 6$. The value of l starts at 1.1 and increases in steps of 0.5 to such a value that leads to no sparsity in the variable weights, i.e $w_j \neq 0, \forall j$. We show the optimal l for each k by larger symbols in Figure 4.6 (left). As expected, the optimal degree of sparsity l differs almost for all k. We select the optimal parameter setting which leads to the largest ${}^{v}Gap_{l^*k}$ resulting in $k = 3$ and $l = 6.6$. Indeed, such choices correspond to the correct number of clusters as well as appropriate values of w_j leading to non-zero weights for all 50 informative variables as shown in Figure 4.6 (right). Considering higher values of l, more and more noise variables obtain non-zero weights. The plot additionally illustrates that a smaller choice of l, e.g. $l = 4.1$, results in an incorrect number of clusters when following the rule for optimizing k based on Gap_k according to Tibshirani et al. (2001). This supports the fact that both parameters need to be optimized at the same time in order to correctly identify groups.

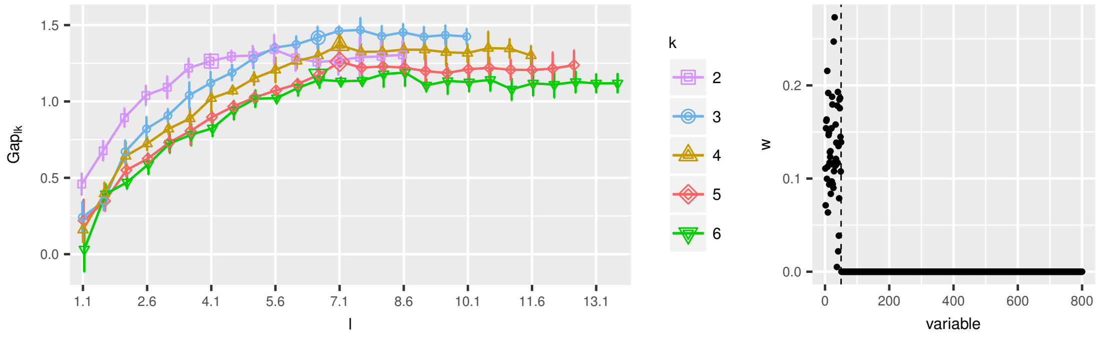

Figure 4.6: Selection of the tuning parameter l and the number of clusters k based on ${}^{v}Gap_{lk}$ (left), and the variable weights corresponding to the optimal $l = 6.6$ and $k = 3$ (right).

4.5 Evaluation setup

We evaluate the performance of the proposed method in terms of the clustering solution, outlier detection, and the identification of informative variables. The clustering solution is evaluated based on the Classification Error Rate (CER), also used by Witten and Tibshirani (2010). CER compares the true group membership with the resulting cluster membership. While CER=0 refers to the best cluster solution, CER=1 corresponds to the poorest performance. In order to evaluate the ability of our method to detect outliers, we report the mean value of observation weights v_i separately for the true non-outliers, i.e $\bar{v}^{nonout}$, and outliers, denoted as $\bar{v}^{out}$. The weights for outliers are supposed to be considerably lower than the weights for non-outliers. Since we recommend to use the final weights v_i for classifying outliers as the observations with $v_i < 0.5$, we calculate True Positive and False Positive Rates (TPR and FPR) ranging between 0 and 1. TPR is defined as the proportion of the number of correctly identified outliers and the actual number of outliers present in a given dataset. High TPR indicates a good ability to identify outliers while low TPR demonstrates poor performance. FPR is calculated as the ratio between the number of non-outliers wrongly declared as outliers and the number of the actual non-outliers in an analyzed dataset. Hence, low values of FPR are preferable over high values. The performance regarding the variable selection is evaluated by comparing the mean value of w_j for informative variables, $\bar{w}^{inf}$, with the mean value of w_j that are different from zero, denoted as $\bar{w}^{non0}$. The higher and more similar the values, the better the ability to correctly select informative variables. We provide a similar evaluation for noise variables and calculate the mean of their weights, $\bar{w}^{noise}$, which is supposed to be close to zero.

Since the clustering procedure employs k-means, we compare the method with several existing k-means-based clustering algorithms, such as k-means (K)[1], trimmed k-means (TKC)[1] by Cuesta-Albertos et al. (1997), and sparse k-means (SKC)[2] by Witten and Tibshirani (2010). The proposed weighted robust and sparse k-means (WRSK) is also compared with trimmed and sparse k-means (RSKC)[2] by Kondo et al. (2012). Although our algorithm is designed in a similar way as RSKC, we avoid to pre-specify the trimming level by incorporating the proposed weighted function. Since no procedure for selecting the optimal k and l has been presented by Kondo et al. (2012) for RSKC in case that no information about data is available, we employ the modified gap statistic considering zero weights for trimmed observations and weights equal to one for untrimmed observations. Note that all trimming-based algorithms require for the pre-specification of a trimming level α, therefore, when applying these methods we consider α as the true percentage of outliers present in a simulated dataset and $\alpha = 0.10$ for real-world data as recommended by Kondo et al. (2012) being a suitable choice for most cases.

4.6 Simulation study

In this section, we explore the ability of the proposed clustering method to correctly reveal the complex data structure in three simulation studies. We first show the efficiency of the gap statistic to properly select l and k. Then, we test the method on the datasets containing various percentages of outliers. Finally, the proposed method is compared with several existing k-means-based approaches.

We now describe the general setting of the simulated datasets considered in the three studies. Each dataset consists of n observations described by the informative as well as uninformative part in terms of the group separation. The observations in the informative part form g groups of sizes $n_t, t = 1, \ldots, g$. The groups are described by p_{inf} variables and are generated following a Gaussian model with parameters $\boldsymbol{\mu}_t \in \mathbb{R}^{p_{inf}}$ and $\boldsymbol{\Sigma}_t \in \mathbb{R}^{p_{inf} \times p_{inf}}$. The elements of the mean vector $\boldsymbol{\mu}_t = (\mu_{t1}, \ldots, \mu_{tp_{inf}})$ are constructed as

$$\mu_{tj} = \begin{cases} \mu, & j = a_z, \\ 0, & \text{else} \end{cases} \tag{4.14}$$

---

[1] We employed the code implemented in the R package `RSKC` (Kondo et al., 2016).

[2] The used code for sparse k-means as available in the R package `sparcl` (Witten and Tibshirani, 2013)

where μ is randomly chosen from the uniform distribution in $[-6,-3] \cup [3,6]$, i.e $U[-6,-3] \cup U[3,6]$. a_z represents the arithmetic sequence defined as $a_{z+1} = a_z + g, a_1 = t$ meaning that the first nonzero element of $\boldsymbol{\mu}_t$ is placed on the t^{th} position and the following nonzero elements, i.e. μ, are always on the position increased by g with respect to the previous index of the nonzero element. Considering, for example, 4 groups of 10 dimensions, the mean vectors of the first two clusters are constructed as $\boldsymbol{\mu}_1 = (\mu, 0, 0, 0, \mu, 0, 0, 0, \mu, 0, 0)$ and $\boldsymbol{\mu}_2 = (0, \mu, 0, 0, 0, \mu, 0, 0, 0, \mu, 0)$. The covariance matrix $\boldsymbol{\Sigma}_t$ is generated according to Campello et al. (2015) as

$$\boldsymbol{\Sigma}_t = \mathbf{Q} \begin{pmatrix} 1 & \rho_t & \dots & \rho_t \\ \rho_t & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \rho_t \\ \rho_t & \dots & \rho_t & 1 \end{pmatrix} \mathbf{Q}^\top, \tag{4.15}$$

where $\mathbf{Q}$ denotes a random rotation matrix satisfying $\mathbf{Q}^\top = \mathbf{Q}^{-1}$ and the off-diagonal elements ρ_t are random numbers from $U[0.1, 0.9]$. To the informative part, we also add p_{noise} noise variables that follow univariate standard normal distributions leading to a total dimensionality of $p = p_{inf} + p_{noise}$.

Such an obtained dataset is finally contaminated by replacing a certain percentage of observations in each group by outliers. We create two types of outliers in the informative variables. While uniformly distributed outliers are generated as random values from $U[-12, 6] \cup U[6, 12]$, the scattered outliers follow a Gaussian model with the same location as a group, i.e. $\boldsymbol{\mu}_t$, but a different covariance structure $\sigma \mathbf{I} \in \mathbb{R}^{p_{inf} \times p_{inf}}$. The parameter σ is randomly generated from an uniform distribution in $[3, 10]$. We also replace a certain proportion of observations from each group in the noise variables by uniformly distributed outliers, according to $U[-12, 6] \cup U[6, 12]$. Note that the observations contaminated in the informative variables differ from those in the noise variables. Furthermore, we always replace (contaminate) the first observations from each group in the informative variables, whereas observations in the noise variables are randomly selected for the following contamination.

4.6.1 Simulation 1: Selection of parameters

n the first study, we investigate the ability of the modified gap statistic to correctly select the number of clusters k and the sparsity parameter l when applying the introduced algorithm. We consider 100 datasets of 800 dimensions in which the first 50 variables describe the group structure. In order to explore the performance of the gap statistic, 3 situations with different numbers of groups are considered, i.e. $g = 3, 4, 5$. The sizes of the observations in the groups are randomly selected, ranging from 50 to 150. The contamination strategy corresponds to replacing the first 10% of observations from each group in all informative variables by scattered outliers. In contrast, the uniformly distributed outliers are placed in 75 randomly selected noise variables.

The proposed method is applied with $k = 2, \dots, 7$ and various l going from 1.1 up to $\sqrt{p}$ in steps of 0.5, in order to calculate the gap statistic and to select optimal parameters. The results are evaluated in terms of the estimated number of clusters and the evaluation measures described in Section 4.5. It should be noted that CER is calculated with respect to the group membership before contamination. Since each group is contaminated by scatter outliers, such outliers have the same location as a group and, therefore, they should be assigned to the corresponding group.

Figure 4.7 summarizes the resulting optimal k selected by the gap statistic as histograms, for the three different numbers of underlying groups ($g = 3, 4, 5$). While the gap statistic works perfectly in case of 3 groups, its performance gets slightly worse for a higher number of groups. Nevertheless, the last two histograms clearly indicate that the optimal k is correctly chosen in most cases.

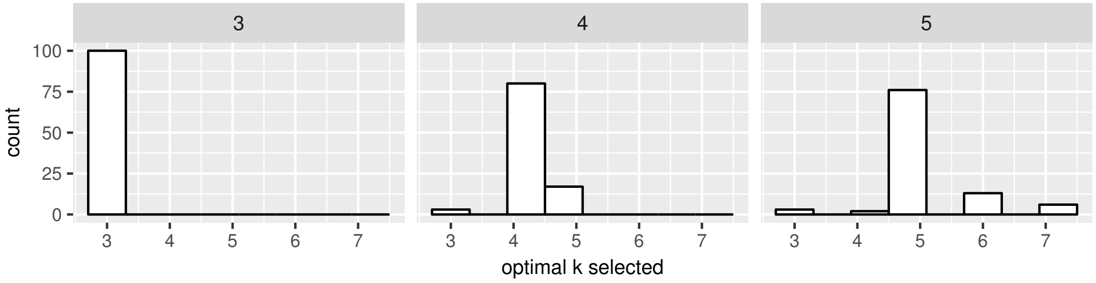

Figure 4.7: Evaluation of the results in terms of the optimal k selected by the gap statistic, for different numbers of groups, i.e. $g = 3, 4, 5$. The reported values are based on 100 simulated datasets for each g.

Figure 4.8 summarizes the results based on the evaluation measures. In general, there is no clear dependence between the considered numbers of groups and the resulting values of evaluation measures. Overall, low CER indicate that the proposed procedure can correctly identify the group structure. In addition, high as well as similar values of $\bar{w}^{inf}$ and $\bar{w}^{non0}$ demonstrate the appropriate selection of l. Hence, it seems that most of the informative variables can be correctly identified. The high performance of variable selection is also supported by zero values of $\bar{w}^{noise}$ suggesting that the method is able to discard all noise variables. We also evaluate the method regarding the detection of outliers. We can see that outliers receive on average considerably low weights in contrast to non-outliers; compare $\bar{v}^{out}$ and $\bar{v}^{nonout}$. Therefore, classifying the observations with $v_i > 0.5$ as outliers seems to be a reasonable choice. Indeed, such a cut-off value leads to a great ability to identify outliers indicated by high TPR as well as low FPR. Considering the values of the evaluation measures, we can conclude that the method as well as the parameter selection work efficiently.

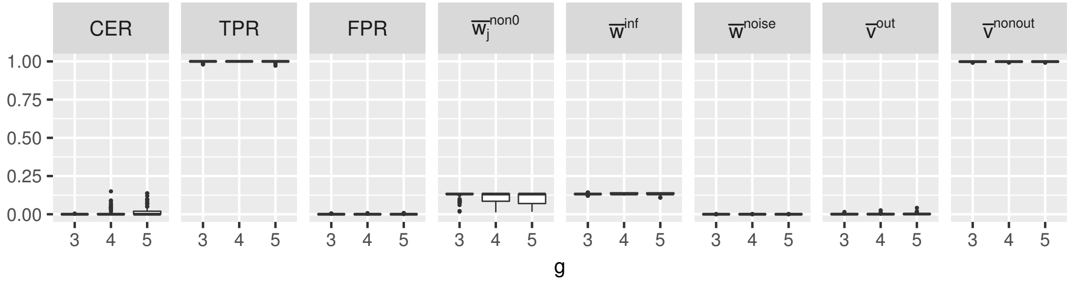

Figure 4.8: Evaluation of the results based on the optimal parameter selection determined by the modified gap statistic, for different numbers of groups, i.e. g=3, 4, 5. The reported values of the evaluation measures and the selected k represent all 100 simulated datasets.

4.6.2 Simulation 2: Resistance against outliers

The second simulation study aims at investigating how resistant the proposed method as well as the modified gap statistic are against various proportions of outliers. For this study, we generate 100 datasets that consist of 3 groups of different sizes ranging between 50 and 150 (randomly selected). The data space is defined by 170 informative variables and 830 noise variables, leading to 1000 dimensions in total. Overall, we consider 8 contamination strategies in terms of different percentages of outliers. The datasets in the first strategy are free of outliers. In contrast, the second strategy considers 5% of scatter outliers in all informative variables and no outliers in noise variables. The datasets in the remaining strategies are contaminated with 10%, 15%, 20%, 30%, and 40% scatter outliers, respectively, in the informative variables. In addition, the proportion of outliers in the 83 (10%) noise variables is always kept as 10%.

Again, the proposed algorithm is applied with the different numbers of clusters ($k = 2, 3, 4, 5, 6$) and various l. Subsequently, the gap statistic is employed to estimate the optimal parameter settings. The performance is finally evaluated by the measures described in Section 4.5 as well as the selected k. As in the previous study, we calculate CER by taking the true group membership before contamination into account.

Figure 4.9 shows the optimal number of clusters estimated by the proposed gap statistic for each contamination strategy. The histograms clearly indicate that the gap statistic allows to correctly select the number of clusters, i.e. $k = 3$, even if data sets contain 40% outliers in total. Although the selection of k seems to be affected by the highest level of contamination corresponding to 50% outliers, such a large proportion of outliers is however very extreme and unrealistic in practice.

Figure 4.10 summarizes the performance in terms of evaluation measures and demonstrates a great ability to discover the group structure independently of the number of outliers, reflected by low CER. The low CER can also be observed in case of the highest contamination. This might indicate that even if the gap statistic estimates a higher

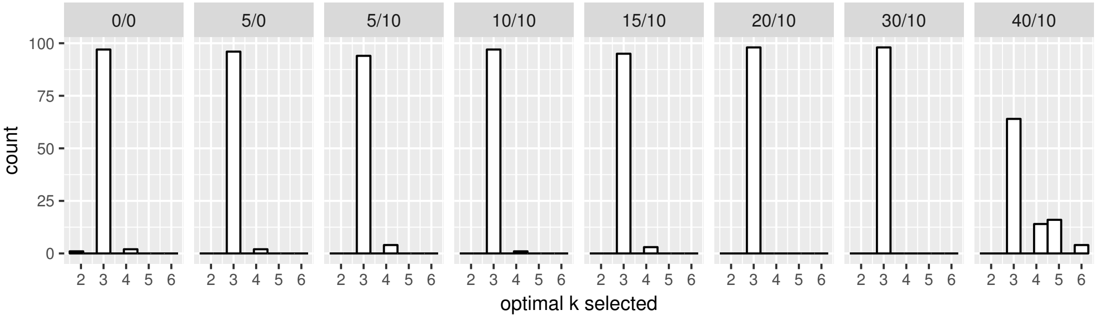

Figure 4.9: Evaluation of the ability to correctly estimate k, considering different percentages of outliers in informative and in noise variables (x/x). The reported values of evaluation measures represent all 100 simulated datasets.

number of clusters than the actual number of groups (see Figure 4.9), the detected clusters seem to be to some extent still homogeneous. The great performance of the gap statistic is additionally supported by similar values of $\bar{w}^{non0}$ and $\bar{w}^{inf}$, indicating highly efficient variable selection. Furthermore, zero values of $\bar{w}^{noise}$ imply that most noise variables are discarded for data clustering. Therefore, we can assume that the sparsity parameter l is appropriately estimated. Regarding the detection of outliers, the method can identify most outliers indicated by TPR around 1. However, TPR is slightly below 1 for the extremely contaminated data sets (i.e. 40/10). Such low TPR can be a consequence of high observation weights for outliers, reflected by higher $\bar{v}^{out}$. This might indicate that the weights of some outliers are similar to the weights of non-outliers, or the cut-off value 0.5 needs to be increased in order to achieve perfect outlier detection for a large contamination level. Although the method misclassifies around 10% of normal observations in case of no contamination (0/0), it is able to correctly classify almost all non-outliers in contaminated datasets indicated by zero FPR. Based on the overall evaluation, the method demonstrates a great ability to identify a complex data structure in contaminated datasets.

4.6.3 Simulation 3: Comparison

In the last study, we compared the proposed weighted robust and sparse k-means (WRSK) algorithm with other k-means-based approaches, such as k-means (KC), trimmed k-means (TKC), sparse k-means (SKC) and its trimmed version (RSKC) on 30 simulated datasets. Each dataset is represented by 4 groups of various sizes ranging between 15 and 150. The generated observations are described by 4000 variables. Since the additional goal is to investigate the influence of different proportions of informative variables, three settings are considered, such as a percentage of 1%, 2%, and 5% of informative variables. Moreover, 20% of the observations are replaced by uniformly distributed outliers in the first 20% of the informative variables, and 10% of other observations are contaminated

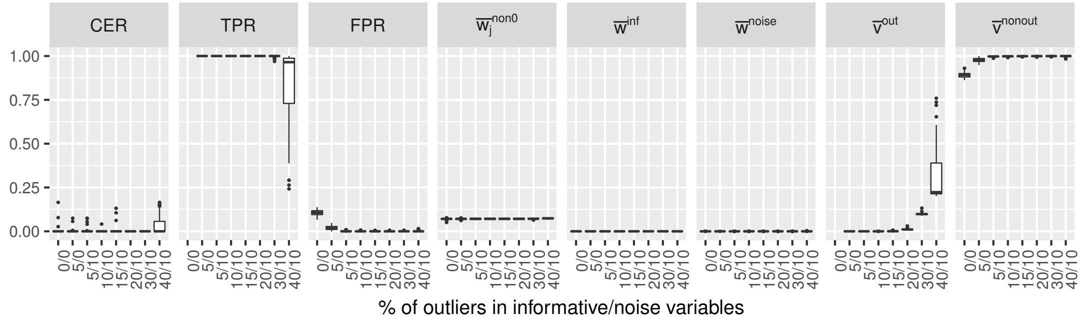

Figure 4.10: Evaluation of the results considering different percentages of outliers in informative and in noise variables (x/x). The reported values of evaluation measures represent all 100 simulated datasets.

in 20% of randomly selected noise variables.

When applying the methods on the generated datasets, we assume prior knowledge of the number of clusters and optimize l in case of sparse-based algorithms. The trimming level for both TKC and RSKC corresponds to the total percentage of outliers, i.e. $\alpha = 0.30$. We evaluate the clustering solution by CER, and if appropriate, the performance regarding the outlier detection by TPR and FPR. Note that CER is again calculated with respect to the true group memberships before contamination. Since the outliers are placed only in the subset of informative variables, there is still some information about the group separation in non-contaminated variables.

Figure 4.11 summarizes the result based on 30 simulations. In general, in comparison to the remaining k-means-based methods, both the proposed method and RSKC seem to be resistant against the different percentages of informative variables. The clustering performance of KC, TKC, and SKC increases with an increasing proportion of informative variables, indicated by decreasing CER. In addition, CER shows that the proposed method outperforms the remaining methods in terms of identifying the underlying group structure reflected by the lowest CER for most simulated datasets. Although lower TPR demonstrate that our WRSK is not capable of identifying all outliers in comparison to the trimmed-based methods, the proposed method misclassifies fewer non-outliers indicated by the lowest FPR. Considering the performance, it seems that our method is able to sufficiently identify the group structure even if a large amount of noise variables is present in a data set.

4.7 Analyzing the group structure of glass vessels

The proposed algorithm is particularly useful in the situation where a large number of variables is present as in the case of archaeological glass vessels from the 16^{th} and 17^{th} centuries, which were excavated in Antwerp being one of the most important historical

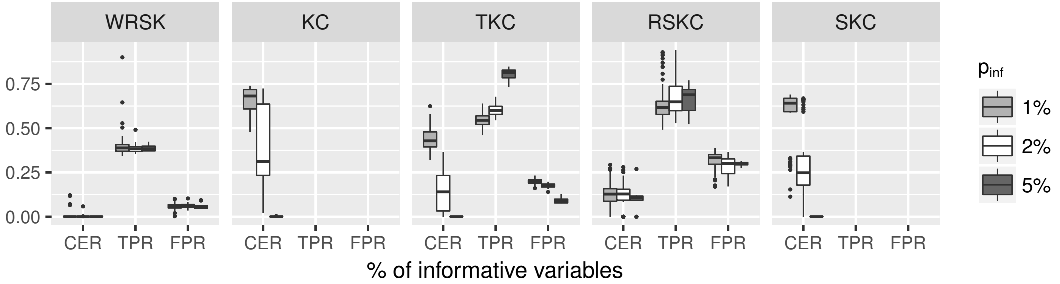

Figure 4.11: Evaluation of various k-means based clustering methods, considering various proportions of informative variables (p_{inf}). The reported values of the evaluation measures correspond to the 30 simulated datasets.

centers of both glass manufacturing and trade. In 1997, chemical analysis was conducted in order to get better insight into the glass collection, including also the possible origin of the various glass samples. For this reason, the glass vessels were analyzed by an electron-probe X-ray micro-analysis (EXPMA) to measure spectra at different energy levels (Janssens et al., 1998). Consequently, traditional calibration methods were applied on spectra to extract major chemical elements resulting in the separation of four glass vessels groups, i.e. sodic, potasso-calcic, calcic, and potassic. The connection between element concentrations of glass vessels and their origin was discussed by Janssens et al. (1998). Lemberge et al. (2000) used their findings on an extended dataset consisting of 180 glass samples described by 1920 variables (different energy levels) in order to predict the same concentrations of the major elements as Janssens et al. (1998), using partial least squares. In this paper we employ the extended dataset consisting of 4 groups as well, as shown in Figure 4.12. The plot additionally shows that the largest group (sodic) is split into two subgroups that are not clearly separated in the two-dimensional space of chemical concentrations. The two subgroups are caused by the installation of different detector efficiencies in the EXPMA. Detecting the subgroup of glass samples analyzed after the installation has been investigated e.g. by Serneels et al. (2005) and Filzmoser et al. (2008).

Our focus is to detect an entire group structure, i.e. 5 groups, which might be hidden in the high-dimensional data space. Note that there is no pre-knowledge about the informative variables, neither of outliers in each group. In addition, the group membership based on the chemical concentrations does not necessarily have to reflect the group structure based on the origin of the glass samples. However, there exist some assumptions about the connection between the chemical elements - the glass manufacturing process - and the origin (Janssens et al., 1998). Therefore, we evaluate the performance in terms of CER with respect to the group membership shown in Figure 4.12, and the cluster membership obtained by k-means-based algorithms with $k = 5$. Although it is not sure whether or not the dataset contains outliers, we set the trimming level to 0.10 for the

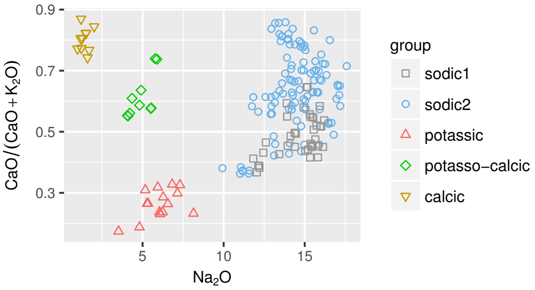

Figure 4.12: Group membership of analyzed glass vessels, based on element concentrations (Lemberge et al., 2000).

Table 4.1: Evaluation of the clustering performance of k-means-based clustering methods.

method	WRSK	KC	TKC	RSKC	SKC
CER	0.039	0.183	0.166	0.191	0.167

trimming-based methods as suggested by Kondo et al. (2012). The optimal sparsity parameter for RSKC and WRSKC is selected from 1.5 to $\sqrt{p}$ in steps of 0.1 based on the gap statistic described in Section 4.4. The evaluation of the resulting clustering solution is presented in Table 4.1, which clearly shows that WRSK outperforms the remaining methods indicated by the lowest CER. Incorporating the trimming concept or sparsity seem to improve the performance of k-means (KC) as demonstrated by slightly larger CER for TKC or SKC. RSKC shows the worst performance. The reason might be that either important variables have been excluded, or that wrong observations have been trimmed, or a combination of both.

We also examine the final variable weights obtained by the sparse k-means-based algorithms. Figure 4.13 shows the final weights for each sparse method. The resulting values of the weights demonstrate that SKC completely fails in terms of achieving sparsity in the variable weight vector, as $w_j > 0$ for almost all variables. Nevertheless, there are several variables that receive a higher weight than in case of RSKC; see two peeks highlighted by dashed lines. This may indicate that there could be useful information about the group separation in the last energy levels of the measured spectra. A very similar conclusion can be made for the weights obtained by the proposed WRSK. In addition, WRSK results in a slightly sparse variable weight vector and at the same time can appropriately identify 5 groups as indicated by the lowest CER.

In order to investigate the final variable weights obtained by the proposed method in more detail, we examine how the centers of the detected clusters are distinguishable at each energy level of the spectra, i.e. for each variable. For this reason, we calculate the cluster centers as a weighted mean of the observations in each variable with the corresponding observation weights and the identified cluster membership. The resulting centers are

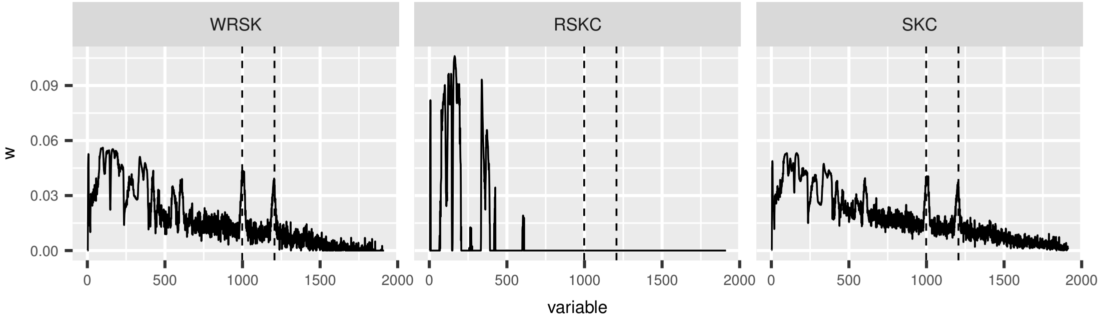

Figure 4.13: The final variable weights obtained by sparse *k*-means-based clustering methods.

displayed as spectra in Figure 4.14 (left) and are distinguished by different colors based on the final cluster membership visualized in Figure 4.14 (right). Figure 4.14 (left) particularly indicates that the centers appear to be well separated already at the low energy levels, i.e. in the first part of the variable vector. Furthermore, the center of cluster 3 appears to be well separated from other centers in the higher energy levels, highlighted by two dashed lines. In fact, the proposed WRSK is capable of identifying this informative part of the spectra; see Figure 4.13. Although the proposed method does not lead to high sparsity in the variable weight vector, the final cluster membership visualized in Figure 4.14 (right) indicates a great ability of WRSK to correctly identify informative variables since all 5 glass vessels groups are well recovered with only 5 misclassified observations. Whereas the misclassified calcic glass sample has an observation weight equal to 1, the remaining four misclassified potassic glass samples obtain weights considerably smaller than 1, i.e. $0.06, 0.00, 0.60, 0.76$. This might indicate that although these observations are originally from the potassic group, their chemical structure seems to be different from the remaining observations of that group.

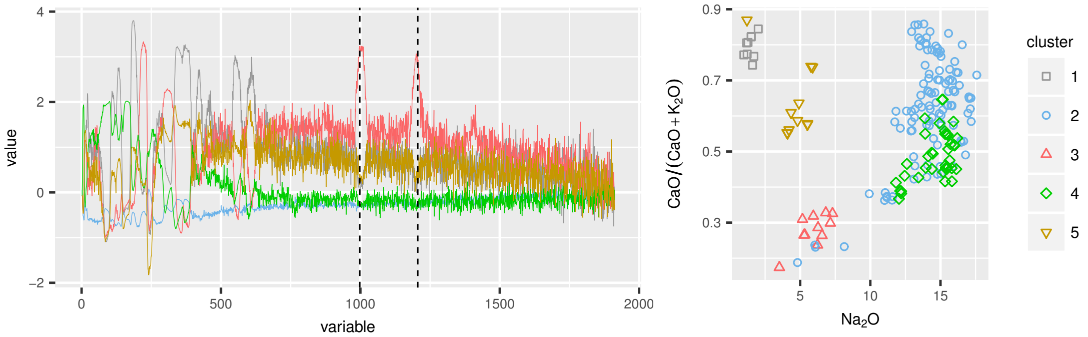

Figure 4.14: Cluster centers calculated for each variable with respect to the final cluster membership obtained by WRSK (left) and the corresponding cluster membership (right).

4.8 Conclusion

We propose a k-means-based clustering procedure that endeavors to simultaneously detect groups, outliers, and informative variables in high-dimensional data. The motivation behind our method is to improve the performance of the popular k-means method for real-world data that possibly contain both outliers and noise variables. (Kondo et al., 2012) have addressed both issues in the robust (trimmed) and sparse k-means procedure, but our method goes even further. Firstly, our method aims to identify clusters, outliers, and noise variables at the same time. Secondly, the proposed procedure is designed in such a way that the required parameters are automatically estimated and, therefore, no pre-knowledge about the data is required. By incorporating the weighting function in k-means, each observation automatically receives a weight reflecting the degree of outlyingness based on which the outliers are identified. In order to correctly detect the informative variables, we employ a sparsity concept adjusted by observation weights. The proposed modified gap statistic is employed to optimize both the sparsity parameter and the number of clusters.

The introduced method together with the modified gap statistic has thoroughly been tested on a variety of simulated data sets as well as on a high-dimensional real data set. The conducted experiments indicated a great ability of the proposed procedure to discover the group structure. The presented clustering algorithm as well as the data generating processes are implemented in the R package `wrsk`, freely available at `https://github.com/brodsa/wrsk`.

Future research includes extending the analysis of a data structure to identify the variables which are responsible for outliers. Such an idea is closely related to cell-wise outlier detection by Rousseeuw and Bossche (2016) for the situation of a single group data structure. A similar concept was introduced by Farcomeni (2014) in the context of clustering. The aim was to demonstrate that cell-wise contamination does not affect the introduced approach. However, the method has been tested in terms of clustering only, and no investigation has been conducted with respect to cell-wise outlier detection. Considering that outliers are commonly highly interesting observations due to their typically different content, it is even more important to find out which variables are behind this unusual behavior.

Acknowledgments

This work has been partly funded by the Vienna Science and Technology Fund (WWTF) through project ICT12-010 and by the K-project DEXHELPP through COMET - Competence Centers for Excellent Technologies, supported by BMVIT, BMWFW and the province Vienna. The COMET program is administrated by FFG.

CHAPTER 5

R implementation

This chapter briefly describes the implementation of two clustering methods: IClust and WRSK, presented in Chapter 3 and Chapter 4, respectively. Both methods are implemented in two separated packages created with `devtools` and `roxygen2` in R - an open source statistical software. The `IClust` package provides a function for data clustering when the true underlying groups are of highly different sizes. The `wrsk` package includes a k-means-based clustering algorithm which is robust against outliers and noise variables. This package also provides a function to generate simulated data. Both packages are freely available at `https://github.com/brodsa`. In order to install them, the `devtools` package must be loaded. This enables installing the packages from GitHub using `install_github("brodsa/IClust")` or `install_github("brodsa/wrsk")`.

5.1 IClust package

The `IClust` package includes an implementation of IClust (Imbalanced Clustering) designed to reveal groups of highly imbalanced sizes. The method is performed in two steps. While the first step uses an existing clustering method to produce a large number initial clusters, the second step employs the Local Outlier Factor (Kriegel et al., 2009b) in order to assess whether or not two close clusters should be merged. The merging procedure is repeated until no clusters can be merged. For further details, the readers are referred to the help file and to Chapter 3.

In this section, a brief introduction of the available functions in the package is provided, see also the help file for more details. For simplicity, an illustrative data example is taken. The data set is also included in the package as `ExampleData` and it consists of 4 groups of highly varying sizes: $300, 30, 10, 3$. To visualize the data set, a PCA representation is considered due to the high dimensionality of the data set, corresponding to 300 variables in total. The following code produces Figure 5.1.

```
library(IClust)
library(ggplot)

data(ExampleData)
dim(data)
# [1] 343 300

## vizualisation
pca <- prcomp(data,center=FALSE) # pca representation
pca_data <- data.frame(as.data.frame(data %*%
    pca$rotation[,1:2]),group=as.factor(label))
ggplot(pca_data,aes(x=PC1,y=PC2,color=group,shape=group))+
  geom_point()
```

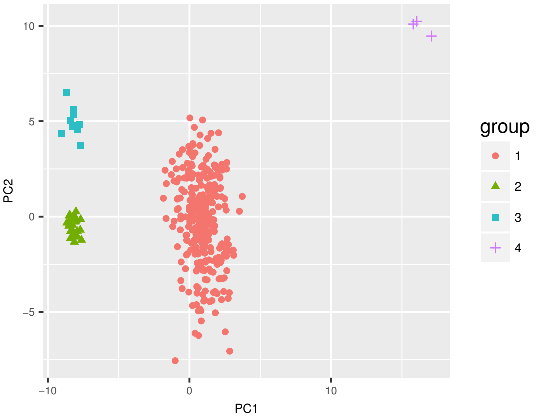

Figure 5.1: Visualization of the illustrative data consisting of 4 groups. The group membership of each observation is distinguished by different color and symbol.

The function `IClust`, applied on the data with standardized values, does not require to specify any parameters as they are set to the optimal values (see Section 3.3). The clustering result obtained by executing `IClust(data)` is shown in Figure 5.2 including the results obtained by k-means and Ward's hierarchical clustering in order to demonstrate the advantage of IClust. Figure 5.2 clearly indicates that IClust outperforms the remaining methods in terms of identifying both small and lager groups. Note that the other two methods are advised to produced 4 clusters, i.e. the actual number of groups.

Although the parameters are set to the optimal values based on the experimental investigation in Section 3.3, all considered parameter choices are included in the implementation as well. Table 5.1 presents the tuning parameters of the `IClust` function. The number of initial clusters and the initial clustering algorithm are needed for the first step to partition the data into a large number of small, potentially homogeneous, clusters. The critical value and the number of the nearest neighbors are used in the second step in order to merge two close clusters to get a final clustering solution.

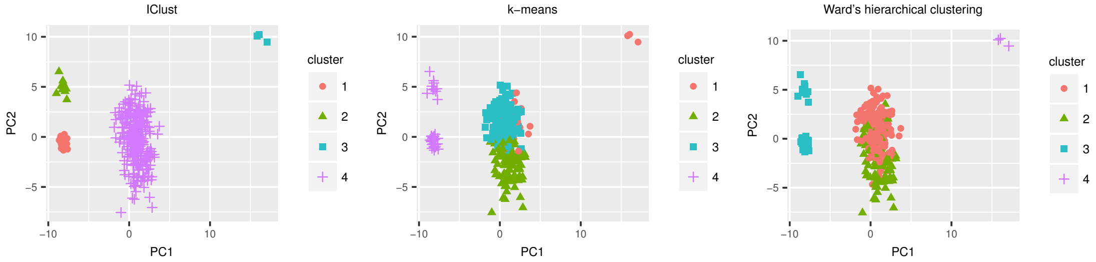

Figure 5.2: Clustering results obtained by the methods IClust, k-means, and Ward's hierarchical clustering. The cluster assignment of each observation is visualized by different color and symbol.

Table 5.1: Arguments of the function `IClust` with the optimal (default) setting.

function argument	description	default setting
`k.init`	number of initial clusters	`10*log(nrow(data))`
`method`	initial clustering method	`"ward"`
`cv`	critical value	cv_1, see Section 3.3
`q.max`	maximal number of neighbors	5

5.2 wrsk package

The `wrsk` package provides the function `wrsk` for k-means-based clustering in case of contaminated and high-dimensional data in which a group data structure is described by a small subset of clustering variables. The package additionally contains the function `SimData` for generating a synthetic data set. This section shortly describes the usage of the functions of the `wrsk` package. More details about the method and the data generating process are provided in Chapter 4 and the help files of `wrsk`.

5.2.1 SimData function

The `SimData` function generates a synthetic data set for the purpose of data clustering. The generated groups are described by the informative variables following Gaussian distributions. Besides informative variables, noise variables are simulated, following univariate standard normal distributions. Optionally, a data contamination can be performed by replacing a certain proportion of observations from each group with outliers. The outliers are either random uniformly distributed values or scatter outliers.

For instance, the following code produces 3 groups that are described by 50 informative variables and 750 noise variables. Each group consists of 40 observations from which 10% are contaminated by scatter outliers in 30 informative variables and by random values in 75 noise variables.

```
library(wrsk)

d <- SimData(size_grp=c(40,40,40),p_inf=50,p_out_inf=30,
        pct_out=0.10, scatter_out=TRUE,
        p_noise=750,p_out_noise=75,noise_pct_out=0.1)

## dimensionality
dim(d$x)
# [1] 120 800

## group and outlier membership (outlier:0)
table(d$lb)
# 0  1  2  3
# 24 32 32 32
```

5.2.2 wrsk function

The `wrsk` function is the implementation of WRSK (Weighted Robust and Sparse k-means) that aims at identifying groups, outliers, and noise variables simultaneously. The implemented method incorporates a concept of assigning weights to both observations and variables, resulting in downweighting the effect of outlying observations and noise variables, respectively.

The code below leads to the clustering solution obtained on the synthetic data set described in the previous section. The two function arguments must be specified: the number of clusters (`k=3`) and the sparsity parameter controlling the contribution of variables for data clustering (`s=7`). The result represented by the confusion matrix indicates an excellent performance of IClust.

```
res <- wrsk(data=scale(d$x), k=3, s=7)

## confusion matrix representig the results (0: outliers)
table(groups=d$lb,cluster=res$outcluster)
#       cluster
# groups  0  1  2  3
#      0 24  0  0  0
#      1  2  0  0 30
#      2  0 32  0  0
#      3  0  0 32  0
```

The function returns the values of observation and variable weights, which are accessible by executing `res$obsweights` and `res$varweights`, respectively. The resulting weights are displayed in Figure 5.3. Different colors and symbols are used to distinguish the true type of a variable (i.e. informative or noise) and whether or not an observation is a true outlier. Figure 5.3 clearly shows that all informative variables received a non-zero weight. Such variables are involved in data clustering and the remaining are excluded. Regarding the observation weights, Figure 5.3 indicates that the weights for all outliers are lower than for most non-outliers. The influence of these outliers is downweighted during k-means clustering. Note that observations with a weight lower than 0.5 are declared as outliers as suggested in Section 4.2.

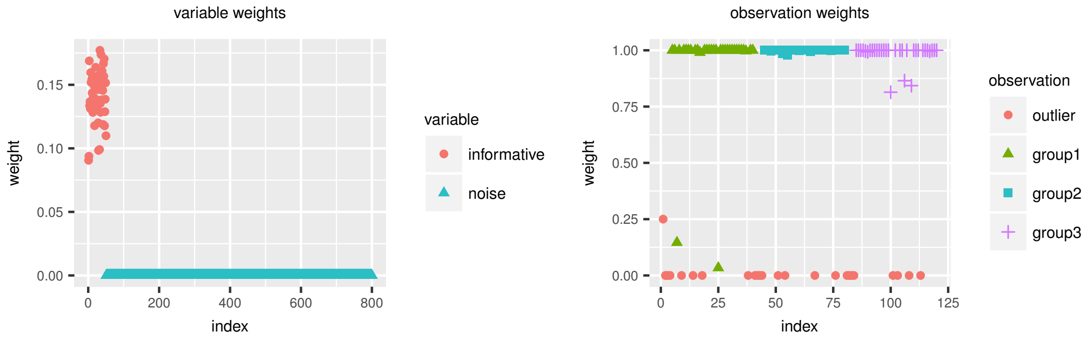

Figure 5.3: Both observation and variable weights obtained by IClust. The actual two types of variables and observations (outliers, non-outliers being part of a group) are differentiated by different colors and symbols.

5.2.3 wrskGap

The `wrsk` package additionally provides the function `wrskGap` for estimating the optimal number of clusters as well as the optimal degree of sparsity. The function compares the quality of a clustering solution with the expected quality obtained on permuted data and uses an one-standard error rule to estimate the parameters, see Section 4.4 for further details. Table 5.2 lists all arguments of `wrskGap`.

Table 5.2: Arguments of the function `wrskGap`.

function argument	description
`data`	data matrix of standardized values
`K`	number of clusters
`S`	numeric vector containing the choices of the sparsity parameter
`n.permute`	number of permuted data sets
`cores`	number of cores used for parallel computing

The following code estimates the optimal sparsity parameter, leading to the clustering solution represented by a confusion table. The estimated parameter `s=6.6` results in the same variables with non-zero weights as in case of considering $s = 7$. The clustering solution and the detected outliers also correspond to the results shown previously. In order to optimize the number of clusters, the readers are referred to `help(wrskGap)`.

```
## applying the estimating procedure
S_val <- seq(1.1,sqrt(ncol(d$x)),0.5)
k3 <- wrskGap(data=scale(d$x),K=3,S=S_val,npermut=10,cores=3)

## selecting the optimal sparsity parameter
id.max.k3 <- which.max(k3$gap)
id.opt.k3 <- which(k3$gap[1:id.max.k3]> (k3$gap[id.max.k3] -
   k3$se[id.max.k3]))[1]
S_val[id.opt.k3]
# [1] 6.6

## clustering solution (0:outliers)
table(d$lb,k3$resFinal[[id.opt.k3]]$outclusters)
#     0  1  2  3
#  0 24  0  0  0
#  1  2  0 30  0
#  2  0  0  0 32
#  3  0 32  0  0
```

Bibliography

C. C. Aggarwal. *Outlier analysis*. Springer, 2nd edition, 2013.

C. C. Aggarwal and C. K. Reddy. *Data clustering: algorithms and applications*. CRC press, 2013.

C. C. Aggarwal and P. S. Yu. Finding generalized projected clusters in high dimensional spaces. In *ACM International Conference on Management of Data*, pages 70–81, 2000.

C. C. Aggarwal, J. L. Wolf, P. S. Yu, C. Procopiuc, and J. S. Park. Fast algorithms for projected clustering. In *ACM SIGMoD Record*, pages 61–72, 1999.

R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *ACM International Conference on Management of Data*, pages 94–105, 1998.

M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *ACM International Conference on Management of Data*, pages 49–60, 1999.

A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Annual Meeting of the Association for Computational Linguistics and International Conference on Computational Linguistics (COLING-ACL)*, pages 79–85, 1998.

J. D. Banfield and A. E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 1(3):803–821, 1993.

C. Becker and U. Gather. The masking breakdown point of multivariate outliers. *Journal of the American Statistical Association*, 94:947–955, 1999.

J. C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.

D. D. Bloisi and L. Iocchi. Rek-Means: A k-means based clustering algorithm. In *International Conference on Computer Vision Systems*, pages 109–118, 2008.

H. H. Bock. Probabilistic models in cluster analysis. *Computational Statistics & Data Analysis*, 23(1):5–28, 1996.

H.-H. Bock. Clustering methods: From classical models to new approaches. *Statistics in Transition*, 5(5):725–758, 2002.

C. Bouveyron and C. Brunet. Simultaneous model-based clustering and visualization in the fisher discriminative subspace. *Statistics and Computing*, 22(1):301–324, 2012.

C. Bouveyron and C. Brunet-Saumard. Discriminative variable selection for clustering with the sparse fisher-em algorithm. *Computational Statistics*, 29(3-4):489–513, 2014.

C. Bouveyron, S. Girard, and C. Schmid. High-dimensional data clustering. *Computational Statistics & Data Analysis*, 52(1):502–519, 2007.

M. M. Breunig, H. Kriegel, R. T. Ng, and J. Sander. LOF: identifying density-based local outliers. In *ACM International Conference on Management of Data*, pages 93–104, 2000.

R. J. Campello, D. Moulavi, A. Zimek, and J. Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data*, 10(1):5, 2015.

G. O. Campos, A. Zimek, J. Sander, R. J. G. B. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30(4):891–927, 2016.

M. E. Celebi, H. A. Kingravi, and P. A. Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Systems with Applications*, 40 (1):200–210, 2013.

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

C.-H. Cheng, A. W. Fu, and Y. Zhang. Entropy-based subspace clustering for mining numerical data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 84–93, 1999.

C. Croux and A. Ruiz-Gazen. High breakdown estimators for principal components: the projection-pursuit approach revisited. *Journal of Multivariate Analysis*, 95(1):206–226, 2005.

C. Croux, P. Filzmoser, and M. Oliveira. Algorithms for projection-pursuit robust principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 87 (2):218–225, 2007.

J. Cuesta-Albertos, A. Gordaliza, and C. Matrán. Trimmed k-means: An attempt to robustify quantizers. *The Annals of Statistics*, 25(2):553–576, 1997.

D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

M. C. P. de Souto, A. L. V. Coelho, K. Faceli, T. C. Sakata, V. Bonadia, and I. G. Costa. A comparison of external clustering evaluation indices in the context of imbalanced data sets. In *Brazilian Symposium on Neural Networks*, pages 49–54, 2012.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 39:1–38, 1977.

I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 269–274, 2001.

L. Ertoz, M. Steinbach, and V. Kumar. A new shared nearest neighbor clustering algorithm and its applications. In *Workshop on Clustering High Dimensional Data and its Applications at SIAM International Conference on Data Mining*, pages 105–115, 2002.

L. Ertöz, M. Steinbach, and V. Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *SIAM International Conference on Data Mining*, pages 47–58, 2003.

M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

C. Fan. *HighDimOut: Outlier Detection Algorithms for High-Dimensional Data*, 2015. R package version 1.0.0.

A. Farcomeni. Snipping for robust k-means clustering under component-wise contamination. *Statistics and Computing*, 24(6):907–919, 2014.

P. Filzmoser, S. Serneels, C. Croux, and P. Van Espen. Robust multivariate methods: The projection pursuit approach. In *From Data and Information Analysis to Knowledge Engineering*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 270–277. Springer, 2006.

P. Filzmoser, R. Maronna, and M. Werner. Outlier identification in high dimensions. *Computational Statistics and Data Analysis*, 52(3):1694–1711, 2008.

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

C. Fraley and A. E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2000.

C. Fraley, A. E. Raftery, T. B. Murphy, and L. Scrucca. mclust version 4 for R: Normal mixture modeling for model-based clustering, classification, and density estimation. Technical Report 597, University of Washington, 2012.

B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

H. Fritz, L. A. García-Escudero, and A. Mayo-Iscar. tclust: An R package for a trimming approach to cluster analysis. *Journal of Statistical Software*, 47(12):1–26, 2012.

G. Galimberti, A. Manisi, and G. Soffritti. Modelling the role of variables in model-based cluster analysis. *Statistics and Computing*, pages 1–25, 2017.

M. T. Gallegos. Maximum likelihood clustering with outliers. In *Classification, Clustering, and Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization*, pages 247–255, 2002.

M. T. Gallegos and G. Ritter. Trimming algorithms for clustering contaminated grouped data and their robustness. *Advances in Data Analysis and Classification*, 3(2):135–167, 2009.

G. Gan, C. Ma, and J. Wu. *Data clustering: theory, algorithms, and applications.* SIAM, 2007.

L. A. García-Escudero and A. Gordaliza. Robustness properties of k-means and trimmed k-means. *Journal of the American Statistical Association*, 94(447):956–969, 1999.

L. A. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar. A general trimming approach to robust cluster analysis. *The Annals of Statistics*, 36(3):1324–1345, 2008.

L. A. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar. A review of robust clustering methods. *Advances in Data Analysis and Classification*, 4(2-3):89–109, 2010.

L. A. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar. Exploring the number of groups in robust model-based clustering. *Statistics and Computing*, 21(4): 585–599, 2011.

L. A. García-Escudero, A. Gordaliza, C. Matrán, and A. Mayo-Iscar. Grouping around different dimensional affine subspaces. In *Statistical Models for Data Analysis*, pages 131–139. Springer, 2013.

R. Gnanadesikan and J. R. Kattenring. Robust estimates, residuals, and outliert detection with multiresponce data. *Biometrics*, 28(1):81–124, 1972.

A. D. Gordon. *Classification.* Chapman and Hall, 2nd edition, 1999.

S. Guha, R. Rastogi, and K. Shim. CURE: an efficient clustering algorithm for large databases. In *ACM SIGMOD International Conference on Management of Data*, pages 73–84, 1998.

M. Hahsler and M. Piekenbrock. *dbscan: Density Based Clustering of Applications with Noise (DBSCAN) and Related Algorithms*, 2017. R package version 1.1-1.

J. A. Hartigan and M. A. Wong. A K-means clustering algorithm. *Applied Statistics*, 28: 100–108, 1979.

M. A. Hasan, V. Chaoji, S. Salem, and M. J. Zaki. Robust partitional clustering by outlier and density insensitive seeding. *Pattern Recognition Letters*, 30(11):994 – 1002, 2009.

M. Hassani and M. Hansen. *subspace: Interface to OpenSubspace*, 2015. R package version 1.0.4.

H. He and Y. Ma. *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley-IEEE Press, Hoboken, New Jersey, 1st edition, 2013.

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

M. Hubert, P. Rousseeuw, and S. Verboven. A fast method for robust principal components with applications to chemometrics. *Chemometrhics and Intelligent Laboratory Systems*, 60(1):101–111, 2002.

M. Hubert, P. Rousseeuw, and K. Vanden Branden. ROBPCA: A new approach to robust principal component analysis. *Technometrics*, 47(1):64–79, 2005.

T. Ishioka. Extended *k*-means with an efficient estimation of the number of clusters. In *International Conference on Intelligent Data Engineering and Automated Learning. Data Mining, Financial Engineering, and Intelligent Agents*, pages 17–22, 2000.

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

K. H. Janssens, I. Deraedt, O. Schalm, and J. Veeckman. *Composition of 15–17th Century Archaeological Glass Vessels Excavated in Antwerp, Belgium*, volume 15, pages 253–267. Springer Vienna, 1998.

W. Jin, A. K. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. In *Advances in Knowledge Discovery and Data Mining, Pacific-Asia Conference*, pages 577–593, 2006.

I. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, New York, USA, 2002.

G. Karypis, E.-H. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999.

L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2nd edition, 2005.

Y. Kondo, M. Salibian-Barrera, and R. Zamar. A robust and sparse k-means clustering algorithm. *arXiv preprint arXiv:1201.6082*, 2012.

Y. Kondo, M. Salibian-Barrera, and R. Zamar. RSKC: An R package for a robust and sparse k-means clustering algorithm. *Journal of Statistical Software*, 72:1–26, 2016.

B. Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 21(9):1–12, 2016.

H. Kriegel, P. Kröger, E. Schubert, and A. Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. *Advances in knowledge discovery and data mining*, pages 831–838, 2009a.

H. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1–58, 2009b.

H. Kriegel, P. Kröger, J. Sander, and A. Zimek. Density-based clustering. *WIREs Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.

H. Kriegel, P. Kroger, E. Schubert, and A. Zimek. Outlier detection in arbitrarily oriented subspaces. In *International Conference on Data Mining*, pages 379–388, 2012.

B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 16–22, 1999.

A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 157–166, 2005.

P. Lemberge, I. De Raedt, K. H. Janssens, F. Wei, and P. J. Van Espen. Quantitative analysis of 16–17th century archaeological glass vessels using PLS regression of EPXMA and μ-XRF data. *Journal of Chemometrics*, 14(5-6):751–763, 2000.

G. Li and Z. Chen. Projection-pursuit approach to robust dispersion matrices and principal components: Primary theory and monte carlo. *Journal of the American Statistical Association*, 80(391):759–766, 1985.

N. Locantore, J. Marron, D. Simpson, N. Tripoli, J. Zhang, K. Cohen, G. Boente, R. Fraiman, B. Brumback, C. Croux, et al. Robust principal component analysis for functional data. *Test*, 8(1):1–73, 1999.

S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1): 24–45, 2004.

M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik. *cluster: Cluster Analysis Basics and Extensions*, 2015. R package version 2.0.3.

R. Maronna and R. Zamar. Robust estimates of location and dispersion for high-dimensional data sets. *Technometrics*, 44(4):307–317, 2002.

G. McLachlan and D. Peel. *Finite mixture models*. John Wiley & Sons, 2004.

A. H. Mohammad, C. Vineet, S. Saeed, and J. Z. Mohammed. Robust partitional clustering by outlier and density insensitive seeding. *Pattern Recognition Letters*, 30 (11):994 – 1002, 2009.

G. Moise, A. Zimek, P. Kröger, H.-P. Kriegel, and J. Sander. Subspace and projected clustering: experimental evaluation and analysis. *Knowledge and Information Systems*, 21(3):299–326, 2009.

J. G. Moreno and G. Dias. Adapted B-CUBED metrics to unbalanced datasets. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 911–914, 2015.

E. Müller, S. Günnemann, I. Assent, and T. Seidl. Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the VLDB Endowment (PVLDB)*, 2(1):1270–1281, 2009.

F. Murtagh and P. Legendre. Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *Journal of Classification*, 31(3):274–295, 2014.

N. Neykov, P. Filzmoser, R. Dimova, and P. Neytchev. Robust fitting of mixtures using the trimmed likelihood estimator. *Computational Statistics & Data Analysis*, 52(1): 299–308, 2007.

J. W. Owsiński and M. T. Mejza. A hybrid clustering method for general purpose and pattern recognition. In *International Multiconference on Computer Science and Information Technology*, pages 121–126, 2007.

S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos. LOCI: Fast outlier detection using the local correlation integral. In *International Conference on Data Engineering*, pages 315–326, 2003.

L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explorations Newsletter*, 6(1):90–105, 2004.

C. Pascoal, M. Oliveira, A. Pacheco, and R. Valadas. Detection of outliers using robust principal component analysis: A simulation study. In *Combining Soft Computing and Statistical Methods in Data Analysis*, pages 499–507. Springer, 2010.

D. Peel and G. J. McLachlan. Robust mixture modelling using the t distribution. *Statistics and computing*, 10(4):339–348, 2000.

J. Qian and V. Saligrama. Spectral clustering with imbalanced data. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3057–3061, 2014.

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

A. E. Raftery and N. Dean. Variable selection for model-based clustering. *Journal of the American Statistical Association*, 101(473):168–178, 2006.

D. M. Rocke. Robustness properties of S-estimators of multivariate location and shape in high dimension. *The Annals of Statistics*, 24(3):1327–1345, 1996.

A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 410–420, 2007.

P. J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20(Supplement C):53–65, 1987.

P. J. Rousseeuw and W. V. d. Bossche. Detecting deviating data cells. *arXiv preprint arXiv:1601.07251*, 2016.

P. Rousseew and K. van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

S. K. Sapra. Robust vs. classical principal component analysis in the presence of outliers. *Applied Economics Letters*, 17(6):519–523, 2010.

E. Schubert, A. Zimek, and H.-P. Kriegel. Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining and Knowledge Discovery*, 28(1):190–237, Jan 2014.

L. Scrucca and A. E. Raftery. clustvarsel: A package implementing variable selection for model-based clustering in R. *arXiv prepring arXiv:1411.0606*, 2014.

S. Serneels and T. Verdonck. Principal component analysis for data containing outliers and missing elements. *Computational Statistics & Data Analysis*, 52(3):1712–1727, 2008.

S. Serneels, C. Croux, P. Filzmoser, and P. J. Van Espen. Partial robust M-regression. *Chemometrics and Intelligent Laboratory Systems*, 79(1):55–64, 2005.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, DTIC Document, 2003.

K. Sim, V. Gopalkrishnan, A. Zimek, and G. Cong. A survey on enhanced subspace clustering. *Data Mining and Knowledge Discovery*, 26(2):332–397, 2013.

G. Stegmayer, D. H. Milone, L. Kamenetzky, M. G. Lopez, and F. Carrari. A biologically inspired validity measure for comparison of clustering methods over metabolic data sets. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(3): 706–716, 2012.

C. A. Sugar and G. M. James. Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 98(463):750–763, 2003.

R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of The Royal Statistical Society Series B-statistical Methodology*, 63(2):411–423, 2001.

C. Vineet, M. A. Hasan, S. Salem, and M. J. Zaki. Sparcl: an effective and efficient algorithm for mining arbitrary shape-based clusters. *Knowledge and Information Systems*, 21(2):201–229, 2009.

M. Walesiak and A. Dudek. *clusterSim: Searching for Optimal Clustering Procedure for a Data Set*, 2015. R package version 0.44-2.

M. E. Wall, A. Rechtsteiner, and L. M. Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003.

Y. Wang and L. Chen. Multi-exemplar based clustering for imbalanced data. In *International Conference on Control Automation Robotics & Vision*, pages 1068–1073, 2014.

D. M. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010.

D. M. Witten and R. Tibshirani. *sparcl: Perform sparse hierarchical clustering and sparse k-means clustering*, 2013. R package version 1.0.3.

C. Wiwie, J. Baumbach, and R. Röttger. Comparing the performance of biomedical clustering methods. *Nature methods*, 12(11):1033, 2015.

H. Xu, C. Caramanis, and S. Mannor. Outlier-robust pca: The high-dimensional case. *IEEE Transactions on Information Theory*, 59(1):546–572, 2013.

R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

Y. Zhao and G. Karypis. Criterion functions for document clustering: Experiments and analysis. Technical report, University of Minnesota, 2002.

A. Zimek. Correlation clustering. unpublished PhD thesis, June 2008. URL `http://nbn-resolving.de/urn:nbn:de:bvb:19-87361`.

A. Zimek, E. Schubert, and H. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5):363–387, 2012.

Index

Curriculum Vitae

Personal Data

Name: Šárka Brodinová
Date of birth: 01.10.1988
Place of birth: Ústí nad Orlicí, Czech Republic
Nationality: Czech

Education

since 2014	PhD in Technical Mathematics, Vienna University of Technology, Austria
2012 – 2014	Master in Applications of Mathematics in Economy, Palacký University Olomouc, Czech Republic
2009 – 2012	Bachelor in Mathematics-Economics of Banking Systems, Palacký University Olomouc, Czech Republic

Work Experience

since Oct 2016	Junior Researcher, K-Project DEXHELPP, Vienna University of Technology, Vienna, Austria
Jan 2014 – Sep 2016	Junior Researcher, FAMOUS Vienna University of Technology, Vienna, Austria

Awards

Oct 2015	Co-winner of the 2015 YSM Data Analysis Competition hosted by UNIDO: Industrial development in least developed countries.
Sep 2015	Book-price for the presentation at International Workshop on Simulations 2015, Vienna, Austria.