A COMPARATIVE STUDY OF EXISTING AND NOVEL METHODS FOR ESTIMATING THE NUMBER OF CLUSTERS IN A DATA SET

by

GRO NILSEN

THESIS

for the degree of

MASTER OF SCIENCE

(Master's degree in Modelling and Data Analysis)

Faculty of Mathematics and Natural Sciences
University of Oslo

March 2009

Acknowledgments

The work on this thesis has taken place from January 2008 to March 2009 at the University of Oslo.

First and foremost, I would like to thank my two supervisors, Ole Christian Lingjærde and Ørnulf Borgan, for invaluable guidance, feedback and discussions. Their doors have been open for me this entire period, and for that I am sincerely grateful.

I would also like to thank friends and family for their support during this year. In particular, I want to thank my friends in study room B800 for their help and encouragement, as well as many enjoyable moments that have kept my spirit up throughout the process. Last, but definitely not least, a special thanks to Hans Inge and Kira, who have put a smile on my face after long and demanding days of work.

Contents

Chapter 1

Introduction

The tendency and ability to group similar objects is a very basic and important skill for most living creatures. Even the early humans and animals must for example have realized that some items shared the property of being edible, while others were poisonous. We all tend to group objects, sometimes unconsciously, to be able to process large amounts of information. A simple example of this may be the waiter who divides the guests in his restaurant into several groups on the basis of which table they are seated at. In addition to the general, everyday classifications that we all deal with, a more scientific interest in the issue of how to organize observed data into meaningful groups was taken already in the ancient Greece. Aristotle, for example, made a classification of animals based on differences and similarities between them. Such attempts of discovering categories of animals and plants have been important throughout the history of biology, the most famous contribution perhaps made by "the father of modern taxonomy", Carl von Linné.

When the amount of observed data is relatively small and comprehensible, and the dimension is low, groups may be discovered by pure inspection and/or graphical visualization of the data. With larger, high-dimensional data sets, however, more systematic methods and algorithms are needed. This field of study is referred to as *Cluster analysis*, a term that was first used by Tryon (1939). The aim of cluster analysis is to discover distinct groups or clusters in data sets, where the objects in the same groups should be similar to each other, while at the same time dissimilar from objects in the other groups. In the last decades, cluster analysis has had widespread use, and a range of clusterings methods and computer programs have been developed. Cluster analysis is applied in a variety of fields, such as geography, chemistry, archeology, economics, marketing, biology and medical research.

One particular discipline in which cluster analysis has become very common, is genomics (the study of the genome of organisms). In later years there have been great advances in this field of study, and new biotechnological methods have made it possible to conduct high-throughput experiments of gene expression. One commonly used method is microarrays, where the expression of several thousand genes are simultaneously measured in each experiment.

This results in high-dimensional data sets. It is of interest in many such experiments, for example in cancer research, to find groups of co-regulated genes, or groups of patients that have similar genetic expression profiles and clinical outcomes. Cluster analysis is widely used for this purpose.

The challenge in cluster analysis is discover the optimal partitioning of the data, and an intrinsic part of this problem is to determine how many clusters are present. In two- or three-dimensional data sets, one may plot the data and visually assess the number of clusters. For high-dimensional data sets, such as data from microarray experiments, however, a numeric method for estimating the number of clusters is called for. It is common in genomics research to determine the number of clusters present from visual examination of a graphical representation of the clustering (the basis for such plots is explained later), in addition to a biological interpretation of the result (e.g. Alizadeh et al., 2000, Perou et al., 2000, Sørlie et al., 2003). Figure 1.1 gives an example of this. Here, the graphical representation of the clustering of a breast tumour data set studied by Sørlie et al. (2003) is shown. The different colours represent five distinct breast tumour subtypes identified by the authors in this article, based on visual inspection of the clustering result, as well as an assessment of the biological appropriateness of these clusters.

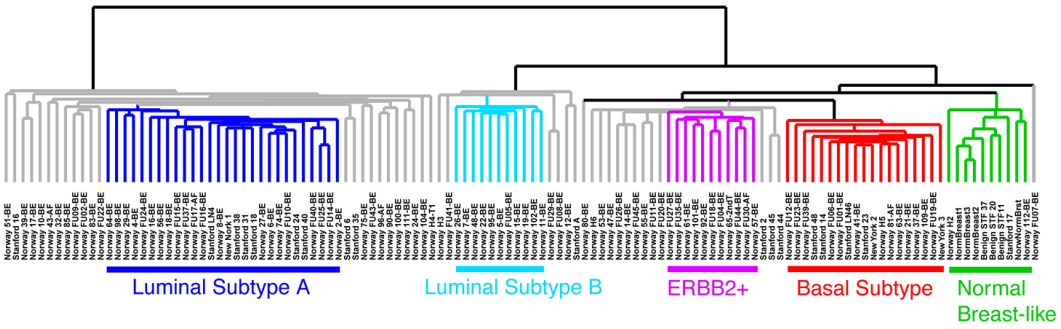


Figure 1.1: Graphical representation of the clustering of breast tumours studied by Sørlie et al. (2003). The authors identified five clusters in this data set, represented by different colours, based on visual inspection of the clustering result, and an understanding of the distinct biological properties of these clusters. Note that some tumours (coloured in grey) were not associated with any of the clusters.

The breast tumour subtypes found by Sørlie et al. (2003) have indeed been shown to have clinical relevance and have been rediscovered in other data. However, this way of identifying the number of clusters is quite subjective, and may be hard to reproduce by others. Furthermore, it is in danger of being influenced by the researchers' prior beliefs and expectations. An important aspect of cluster analysis is therefore to develop objective and automatic methods for estimation of the number of clusters in a data set. Many methods have been proposed over the years, and a goal of this thesis is to thoroughly describe some of the existing methods and introduce novel methods. Another goal is to compare the methods' effectiveness on real microarray data sets and simulated data sets, and thereby get an indication of the methods' virtues and limitations.

An introduction to cluster analysis is given in Chapter 2. Since the purpose of cluster analysis is to find groups of objects that are similar, some common distance and similarity measures are introduced first. Two approaches to clustering are thereafter presented and discussed, namely K-means clustering and hierarchical clustering. In Chapter 3, several methods for estimating the number of clusters are presented, and two novel approaches are introduced. In Chapter 4, the described methods are applied on two microarray data sets to estimate the number of clusters in each of these. One of these is the already mentioned breast tumour data in Figure 1.1. To get further insight into virtues and shortcomings of the proposed methods, their performance on simulated data from several simulation scenarios are studied in Chapter 5. Since the true number of clusters is known in these simulations, they form a basis for a comparison of the methods' effectiveness. In Chapter 6, the results are discussed and conclusions are drawn. Some suggested topics for future research are also given. Additional figures from the various simulation scenarios in Chapter 5 are given in Appendix A, while tables with detailed results from the simulations are given in Appendix B.

Chapter 2

Cluster Analysis

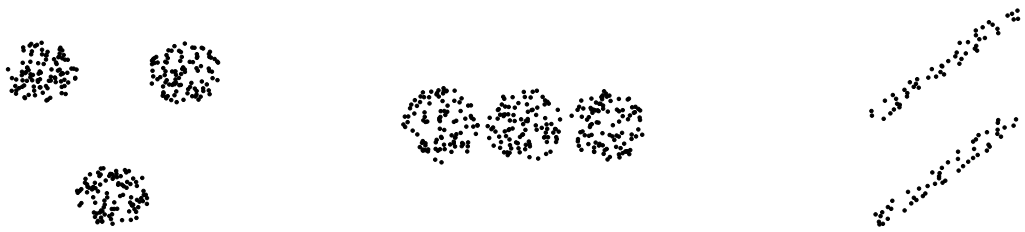
2.1 Introduction

The purpose of cluster analysis is to partition a set of individual objects (which may represent for example tumours or genes) into groups or clusters consisting of similar objects. Cluster analysis is also referred to as unsupervised learning, because the true group structure is not known in advance. This is in contrast to supervised learning methods, such as discriminant analysis, where the groups are known a priori and where the purpose is to identify which group a new object should be assigned to.

An inherent difficulty of cluster analysis is the lack of a definition of the term *cluster*. What is perceived as a cluster depends on the judgment of the investigator, and a single definition of the term is therefore not likely to be suitable for all situations. Ideally, the members of a cluster should be similar or closely related to each other in some respect, while at the same time separated from the objects in other clusters. That is, the clusters should be *internally cohesive* as well as *externally isolated* (Gordon, 1999). The degree to which observed clusters actually are characterized by such properties varies from situation to situation, however, as illustrated by Figure 2.1. The clusters in the first panel are both internally cohesive and externally isolated, while the clusters in the middle panel are very cohesive, but not very isolated. In the last panel, the clusters are isolated, but not very cohesive. Despite the lack of isolation in Figure 2.1b and the lack of cohesiveness in Figure 2.1c, most observers would probably still argue that there are three and two clusters in these respective data sets, just by inspecting the plots. When several objects are measured on much more than 2 variables the situation is complicated by the fact that it is very difficult or impossible to display the data graphically in such a way that the group structure becomes evident. The need for numerical methods to discover the clusters is therefore much greater when the dimension is high.

Several approaches to cluster analysis have been suggested and some will be presented in this chapter. However, since most methods search for clusters that are cohesive, as well as isolated, a fundamental concept in all strategies is the definition of a similarity or distance

measure. Some of the most common measures are on that account described in the next section.



(a) Cohesive and separated clusters (b) Cohesive, but not very isolated clusters (c) Isolated, but not very cohesive clusters

Figure 2.1: An illustration of clusters of varying degree of internal cohesion and external isolation.

2.2 Measures of similarity and distance

Since the purpose of cluster analysis is to identify groups of objects that are similar in some respect, a quantitative measure of proximity between the objects is required. Many clustering methods start out with an $N \times N$ matrix of pairwise proximities between the N objects to be clustered, and use this information to group proximate objects together. Proximity measures may broadly be divided into distance measures and similarity measures, and two objects are "close" if they have small distance, or large similarity, between them. Some common distance and similarity measures are presented next.

A common way of measuring the proximity between two objects, is to define a distance function. A distance function $d()$ is said to be a *metric* if it satisfies the following conditions for all $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ in $\mathbb{R}^p$:

- Triangle inequality: $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$
- Symmetry: $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$
- Non-negativity: $d(\mathbf{x}, \mathbf{y}) \geq 0$, with equality if and only if $\mathbf{x} = \mathbf{y}$.

The most common choice of distance function in cluster analysis is the *Euclidean distance*. Here, the distance between the p-dimensional vectors $\mathbf{x} = [x_1, x_2, \ldots, x_p]^T$ and $\mathbf{y} = [y_1, y_2, \ldots, y_p]^T$ is given by

$$d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_2 = \sqrt{\sum_{j=1}^{p} (x_j - y_j)^2}. \tag{2.1}$$

This distance is also referred to as the l_2-norm. In some contexts the squared Euclidean distance is also used. Another distance measure, also known as the l_1-norm, is the *Manhattan distance*

$$d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_1 = \sum_{j=1}^{p} |x_j - y_j|,$$

which measures distances between points in a "city street" grid. The *Minkowski distance*, or l_r-norm, is the general form of the two former distance measures, and is given by

$$d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_r = \left(\sum_{j=1}^{p} |x_j - y_j|^r \right)^{1/r} \quad (r \geq 1)$$

Of the similarity measures, the most widely used is *Pearson's centered correlation*, which is usually represented by the correlation coefficient ρ. This measure indicates the strength and direction of a linear relationship between two vectors. Given the vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^p$, the centered correlation of $\mathbf{x}$ and $\mathbf{y}$ is defined by

$$\rho(\mathbf{x}, \mathbf{y}) = \frac{\sum_{j=1}^{p}(x_j - \overline{x})(y_j - \overline{y})}{\sqrt{\sum_{j=1}^{p}(x_j - \overline{x})^2 \sum_{j=1}^{p}(y_j - \overline{y})^2}} = \frac{\mathbf{x}_c^T \cdot \mathbf{y}_c}{||\mathbf{x}_c||_2 ||\mathbf{y}_c||_2} \tag{2.2}$$

where $\overline{x} = \frac{1}{p}\sum_{j=1}^{p} x_j$, $\overline{y} = \frac{1}{p}\sum_{j=1}^{p} y_j$ and $\mathbf{x}_c$ and $\mathbf{y}_c$ are the centered vectors $(\mathbf{x} - \overline{x}\mathbf{1})$ and $(\mathbf{y} - \overline{y}\mathbf{1})$, respectively.

An important property of the correlation coefficient is that it can only take values in the range -1 to 1 (as is easily seen from the relationship (2.3) below). A correlation coefficient of 1 corresponds to perfect similarity in the sense that the centered vectors $\mathbf{x}_c$ and $\mathbf{y}_c$ are parallel, i.e. $\mathbf{x}_c = \alpha \mathbf{y}_c$, for some $\alpha > 0$. The correlation coefficient may easily be converted into a distance measure by taking $1 - \rho(\mathbf{x}, \mathbf{y})$, which is then bounded in [0, 2]. This distance measure is not a metric, however, because neither the triangle inequality nor the non-negativity condition ($d(\mathbf{x}, \mathbf{y}) = 0$ iff $\mathbf{x} = \mathbf{y}$) hold. Another property of the correlation coefficient is that it has a geometrical interpretation as the cosine of the angle between the two vectors. We have the relationship

$$\cos\theta = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}||_2 ||\mathbf{y}||_2} \tag{2.3}$$

where $\mathbf{x} \cdot \mathbf{y} = \sum_{j=1}^{p} x_j y_j$, and θ is the angle between the vectors $\mathbf{x}$ and $\mathbf{y}$. Substituting $\mathbf{x}_c$ for $\mathbf{x}$ and $\mathbf{y}_c$ for $\mathbf{y}$ in (2.3) we find that $\rho(\mathbf{x}, \mathbf{y}) = \cos\theta$, where θ is now the angle between the two centered vectors $\mathbf{x}_c$ and $\mathbf{y}_c$.

Variations of the $1 - \rho$ distance measure include the uncentered version where $\overline{x}$ and $\overline{y}$ are replaced by 0 in (2.2), and a version where the absolute value of ρ is applied $(1 - |\rho|)$.

The choice of a distance or similarity measure will influence the result of the clustering, and different results may emerge from using different measures. Hence, it is important to specify

an appropriate distance or similarity measure. What is appropriate depends on the application area, however, and the decision must therefore come from subject matter considerations (e.g. Hastie et al., 2001). For microarray data, a popular choice is the $1 - \rho$ distance.

Relationship between Euclidean distance and Pearson's centered correlation

Under certain assumptions, Euclidean distance can be shown to be proportional to Pearson's centered correlation. Assume two p-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$ that have been centered and normalized to unit norm. Hence,

$$\sum_{j=1}^{p} x_j = \sum_{j=1}^{p} y_j = 0$$

and

$$\sqrt{\sum_{j=1}^{p} x_j^2} = \sqrt{\sum_{j=1}^{p} y_j^2} = 1.$$

Using (2.2) we then have

$$\rho(\mathbf{x}, \mathbf{y}) = \frac{\sum_{j=1}^{p}(x_j - \overline{x})(y_j - \overline{y})}{\sqrt{\sum_{j=1}^{p}(x_j - \overline{x})^2 \sum_{j=1}^{p}(y_j - \overline{y})^2}} = \sum_{j=1}^{p} x_j y_j.$$

Further, using (2.1), the squared Euclidean distance between the vectors is

$$\begin{aligned} d_{Euc}^2(\mathbf{x}, \mathbf{y}) &= \sum_{j=1}^{p}(x_j - y_j)^2 \\ &= \sum_{j=1}^{p} x_j^2 + \sum_{j=1}^{p} y_j^2 - 2\sum_{j=1}^{p} x_j y_j \\ &= 2 - 2\sum_{j=1}^{p} x_j y_j \\ &= 2 - 2\rho(\mathbf{x}, \mathbf{y}) \end{aligned} \tag{2.4}$$

In other words, for centered and normalized vectors $\mathbf{x}$ and $\mathbf{y}$, an increase in the Euclidean distance is associated with a corresponding decrease in the correlation coefficient (apart from a multiplicative factor of 2). Hence, minimizing the Euclidean distance will be equivalent to maximizing the centered correlation.

2.3 Clustering methods

We may distinguish between at least three types of clustering methods: Combinatorial methods, methods based on mixture modeling and mode seeking methods. In combinatorial algorithms, no assumptions are made about an underlying probability model and all work is done directly

on the observed data. Both the clusters and the visualization of the results may differ from method to method. Two of the most popular combinatorial clustering approaches are K-means clustering and hierarchical clustering, and these will be presented below. (For a more thorough introduction to cluster analysis see for example Everitt et al. (2001).)

2.3.1 K-means clustering

The K-means clustering algorithm is perhaps the most popular of the so-called "optimization clustering methods". These methods aim to partition objects into a preset number of groups in such a way that some defined criterion is optimized (minimized or maximized).

Consider a $N \times p$ data matrix X consisting of measurements taken on p variables for N objects. Let C represent a partition that maps the N objects into K clusters, i.e. C is the surjective mapping:

$$C : \{1, \ldots, N\} \to \{1, \ldots, K\}.$$

In other words, C represents a way of allocating N objects to K clusters and the cluster assignment of a given object i may be written as $C(i) = k$, where $1 \leq k \leq K$. Given N and K the total number of possible such partitions is given by (e.g. Everitt et al., 2001)

$$P_C(N, K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^N.$$

Even for moderate values of N and K, the number of possible partitions will be very large. For example, $N = 50$ objects allocated to $K = 4$ groups, gives us $P_C(50, 4) = 5.3 \times 10^{28}$ possible different partitions. Hence, for most practical purposes it is not feasible to consider all possible partitions in the search for an optimal one and this problem has lead to the development of several algorithms that consider only a subset of all the possible partitions. The basic strategy of these algorithms is to specify an initial partition, and then iteratively change cluster allocations in such a way that the value of a chosen criterion is always improved upon. The algorithms differ in the choice of a criterion and in their strategies for modifying the cluster assignments at each step.

The K-means algorithm is one such method. This method tries to minimize the total within-cluster dispersion, given by

$$W(C) = \sum_{k=1}^{K} \sum_{C(i)=k} d^2(\mathbf{x}_i, \overline{\mathbf{x}}_k), \tag{2.5}$$

for a given partition, C. Here $\mathbf{x}_i = [x_{i1}, \ldots, x_{ip}]^T$ represents a vector of observed values for object i ($i \in \{1, \ldots, N\}$) and $\overline{\mathbf{x}}_k$ denotes the centroid vector of the k'th cluster. That is, $\overline{\mathbf{x}}_k$ is

the vector of means for the p variables calculated over the objects found in cluster k:

$$\overline{\mathbf{x}}_k = \left[\frac{1}{n_k} \sum_{C(i)=k} x_{i1}, \ \ldots, \ \frac{1}{n_k} \sum_{C(i)=k} x_{ip} \right]^T,$$

where n_k is the number of objects in the k'th cluster. Using Euclidean distance, (2.5) may be written as

$$W(C) = \sum_{k=1}^{K} \sum_{C(i)=k} ||\mathbf{x}_i - \overline{\mathbf{x}}_k||_2^2 = \sum_{k=1}^{K} \frac{1}{2n_k} \sum_{C(i)=k} \sum_{C(j)=k} ||\mathbf{x}_i - \mathbf{x}_j||_2^2,$$

The last equality is easily shown by adding and subtracting $\overline{\mathbf{x}}_k$ inside the norm in the right hand side.

As stated above, the goal of the K-means algorithm is to find the particular partition C^* that minimizes $W(C)$, so for Euclidean distances we have

$$C^* = \underset{C}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{C(i)=k} ||\mathbf{x}_i - \overline{\mathbf{x}}_k||_2^2.$$

To achieve this, the first step in the algorithm is to specify the number of clusters K. This choice is often made on the basis of subjective considerations, but there also exist methods that try to decide on an optimal number of clusters in the data set (next chapter). Next, one has to decide on an initialization of cluster centroids. There are several ways of doing this. One possibility is to randomly assign objects to K clusters and then calculate the centroid of each cluster. Another possibility is to directly specify an initial set of cluster centroids, for example by randomly choosing K out of the N objects as the initial centroids. Yet another way is to use the result from another clustering method to define the initial cluster centroids. After the initializations have been made, the algorithm proceeds iteratively by reassigning objects to the clusters in such a way that $W(C)$ is minimized. This is done through the following steps:

1. Given the current set of centroids $\{\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_K\}$, (2.5) is minimized by assigning the objects to the cluster with the closest centroid. That is,

$$C(i) = \underset{1 \le k \le K}{\operatorname{argmin}} ||\mathbf{x}_i - \overline{\mathbf{x}}_k||_2^2,$$

   where $C(i)$ represents the cluster that object i is allocated to. $C(i)$ will thus be the cluster k that minimizes the squared Euclidean distance between $\mathbf{x}_i$ and $\overline{\mathbf{x}}_k$. This is done for all of the N objects. If the minimum is not unique, such that the minimum value is found for more than one k, the object is arbitrarily allocated to one of the clusters in question.

2. After the reallocation, the new set of centroids $\{\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_K\}$ is computed.

3. Steps 1 and 2 are repeated until no reallocation causes $W(C)$ to decrease.

Since the value of $W(C)$ is reduced for each iteration in the algorithm, convergence towards a local minimum is assured. However, the partition C found by K-means clustering may not be the optimal (global) solution, C^*, as the solution typically depends on the preset number of clusters and the initial positioning of the centroids. In fact, an inherent problem of the algorithm is that it may return different clusters for different choices of the initial centroids. It may therefore be advisable to run the algorithm several times with different initializations, and keep the solution for which the value of $W(C)$ is minimized. Other potential problems of the K-means algorithm is that it tends to be biased toward spherical clusters, and to produce clusters that are roughly equally-sized (e.g. Everitt et al., 2001). The method is, however, computationally fast and still widely used.

2.3.2 Hierarchical clustering

Another clustering approach that is widely used is hierarchical clustering. The end result of all hierarchical clustering methods is a hierarchy with $N - 1$ levels, where N is the total number of objects. This hierarchy is often represented graphically by a tree-like structure called a *dendrogram*, and an example of such a representation was given in Figure 1.1 and is also shown in Figure 2.2. The hierarchical structure is evident in these figures, where the most similar objects are grouped together at the lower levels and the less similar objects are grouped together at the higher levels. Also, note that at the bottom of the dendrogram, before any grouping has taken place, the clusters consist only of single objects, while at the top level there is only a single cluster containing all objects. Another property of the dendrogram is that the height portrayed on the y-axis reflects the distance between the two clusters that are merged at that level.

Hierarchical clustering methods basically follow one out of two strategies; *divisive* (top-down) clustering or *agglomerative* (bottom-up) clustering. In the divisive paradigm, the methods start at the top with all the objects contained in one cluster, and then at each level recursively split an existing cluster into two new clusters. The choice of split is made so as to produce two new clusters with the largest possible between-group distance. Far more used are, however, methods following the agglomerative paradigm, and in the following, we focus on this alternative.

Agglomerative hierarchical clustering start with all objects in separate clusters at the bottom, and then proceed by merging the two closest clusters into a single cluster at the next level. This continues over $N - 1$ steps until all N objects are contained in one cluster. A definition of a distance measure between two groups of objects (clusters) is thus required. Several definitions have been suggested, and the various agglomerative clustering methods differ in how this

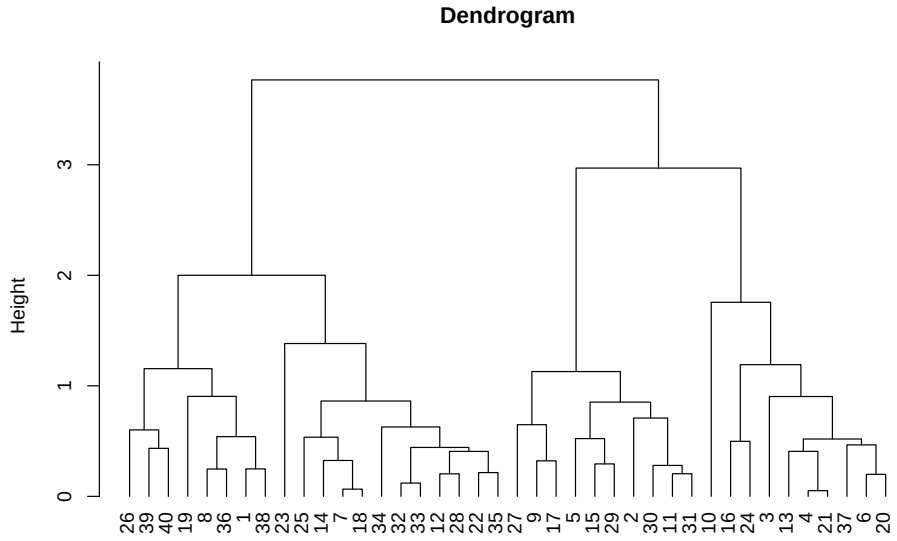


Figure 2.2: A simple illustration of a dendrogram where 40 objects (represented by the numbers 1,...,40) have been hierarchically clustered. The height at which two clusters or objects are merged reflects the distance between them.

intergroup distance or *linkage* method, is defined.

Linkage methods

Three of the most common linkage methods are single linkage (SL), complete linkage (CL) and group average linkage (GA) (Hastie et al., 2001). In *single linkage* (also called "nearest-neighbour" technique) the intergroup distance, $d(G, H)$, between two clusters denoted G and H, is taken to be that of the closest pair of objects in G and H. That is,

$$d_{SL}(G, H) = \min_{i \in G, j \in H} d(\mathbf{x}_i, \mathbf{x}_j).$$

Complete linkage (also called "furthest neighbour" technique), on the other hand, defines the intergroup distance between the clusters to be that of the furthest pair of objects;

$$d_{CL}(G, H) = \max_{i \in G, j \in H} d(\mathbf{x}_i, \mathbf{x}_j).$$

Finally, *group average linkage* takes the average distance (or dissimilarity) between the clusters to represent the intergroup distance;

$$d_{GA}(G, H) = \frac{1}{n_G n_H} \sum_{i \in G} \sum_{j \in H} d(\mathbf{x}_i, \mathbf{x}_j).$$

Here n_G and n_H are the number of objects in cluster G and H, respectively. For all three definitions, i and j represent objects found in G and H, respectively, and $d(\mathbf{x}_i, \mathbf{x}_j)$ denotes the distance between these objects (e.g. Euclidean distance or $1 - \rho$ distance). The three linkage methods will give similar results if the clusters are cohesive and well separated from each other. When this is not the case, the three linkage methods may produce very different results.

One problem with single linkage is the phenomenon known as *chaining*. Since single linkage only depends on the smallest distance between objects in two groups, it tends to combine clusters due to single objects being close to each other regardless of the distance to the other objects in that cluster. This is often seen to produce long chains of clusters. On the other hand, complete linkage represents the opposite extreme. Two clusters are only combined if all the objects in the groups are relatively close. This may produce cohesive clusters, but a problem can be that some of the objects in a cluster may be closer to objects in other clusters than to objects in their own cluster. Group average linkage tries to make a compromise between the two other methods by attempting to produce relatively cohesive clusters that are relatively far apart. Note, however, that with group average linkage the results depend upon the numerical scale used to measure the distances $d(\mathbf{x}_i, \mathbf{x}_j)$. Thus, applying a monotone strictly increasing transformation to the distance measures may change the results for group average linkage, whereas this will not be the case for single and complete linkage.

The three definitions of intergroup distance mentioned above are the ones most commonly used, but other definitions have also been suggested. These include, among others, centroid linkage, median linkage and Ward's linkage (e.g. Timm, 2002). *Centroid linkage* defines the distance between two clusters to be the distance between the centroids of the clusters. Let $\overline{\mathbf{x}}_G$ and $\overline{\mathbf{x}}_H$ represent the centroids of cluster G and H, respectively. The distance between the two clusters is taken to be

$$d_{Centroid}(G, H) = d(\overline{\mathbf{x}}_G, \overline{\mathbf{x}}_H),$$

and the two clusters whose centroids are the closest will be merged. Letting J represent the new cluster, the centroid of J is the (unweighted) average of the two merged clusters:

$$\overline{\mathbf{x}}_J = (n_G\overline{\mathbf{x}}_G + n_H\overline{\mathbf{x}}_H)/(n_G + n_H) = \frac{1}{n_J}\sum_{i \in J}\mathbf{x}_i,$$

where $n_J = n_G + n_H$.

Median linkage is similar to centroid linkage, but here the centroid of the new cluster J is redefined as a weighted average of the centroids of the two merged clusters:

$$\overline{\mathbf{x}}_J = (\overline{\mathbf{x}}_G + \overline{\mathbf{x}}_H)/2.$$

Median linkage may thus differ substantially from centroid linkage when clusters of very different size are merged.

Ward's linkage takes on a different strategy. Here the distance between two clusters is defined as the increase in the within-cluster dispersion if the two clusters were to be merged, and the method will merge clusters so as to minimize this increase. Let W_G and W_H represent the within cluster dispersion of the clusters G and H, i.e.

$$W_G = \sum_{i \in G} ||\mathbf{x}_i - \overline{\mathbf{x}}_G||_2^2,$$

and

$$W_H = \sum_{i \in H} ||\mathbf{x}_i - \overline{\mathbf{x}}_H||_2^2.$$

If we were to merge the clusters G and H to form cluster J, the within cluster dispersion of this cluster will be

$$W_J = \sum_{i \in J} ||\mathbf{x}_i - \overline{\mathbf{x}}_J||_2^2,$$

where $\overline{\mathbf{x}}_J = (n_G\overline{\mathbf{x}}_G + n_H\overline{\mathbf{x}}_H)/(n_G + n_H)$. The increase in the within cluster dispersion following this merge, and hence the distance between cluster G and H, will be the difference between the within-cluster dispersion of cluster J and the combined within-cluster dispersion of clusters G and H. Hence, for Ward's linkage we have $d(G, H) = W_J - (W_G + W_H)$, which simplifies to

$$d_{Ward}(G, H) = \frac{n_G n_H}{n_G + n_H} ||\overline{\mathbf{x}}_G - \overline{\mathbf{x}}_H||_2^2.$$

Note that this distance is proportional to the squared Euclidean distance between the centroids of the clusters to be merged. Hence Ward's linkage differs from centroid linkage in that the centroid distance is weighted by the factor $(n_G n_H)/(n_G + n_H)$. As a consequence, in a situation where centroid linkage considers the merge of two clusters G and H to be as good as a merge between G' and H', Ward's linkage would favor the merge of the smallest pair of clusters. Accordingly, when using Ward's linkage, the larger a cluster becomes, the less likely it is to merge with another large cluster until at the very end.

Determining the number of clusters

One of the advantages of hierarchical clustering over K-means clustering, is that the user is not required to determine a number of clusters K before the clustering starts. Hence the K-means algorithm may have to be run for a variety of K's to find the optimal number, whereas the hierarchical clustering only has to be performed once. If the number of different K's to be tried is very large, there may thus be a computational advantage to using hierarchical clustering. However, for a limited number of K's, the K-means algorithm is likely to be computationally faster.

The most common way to produce a chosen number of clusters K from hierarchical clustering, is to simply cut the dendrogram horizontally at a certain height. Such a cut produces

a certain number of clusters since each level of the hierarchy represents a particular grouping of the data into disjoint clusters. For example, in Figure 2.2 we may cut the dendrogram at the height 2.5 to obtain three distinct clusters. In the dendrogram of breast tumours shown in Figure 1.1, on the other hand, separate cut heights were (informally) specified to define each cluster. More formal ways of doing this are discussed in Chapter 3. Other more sophisticated methods for non-horizontal cutting of the dendrogram have also been proposed (e.g. Langfelder et al., 2008).

The use of a dendrogram in hierarchical clustering provides a visual advantage in that it gives the user the flexibility to decide upon a cut that seems to represent a natural grouping of the data into K clusters, *after* the dendrogram has been studied. However, since the choice of such a cut off may be made from visual inspection of the dendrogram, and is usually influenced by the user's expertise and theories, a disadvantage will be the lack of objectivity in this decision.

Hence, for both K-means clustering and hierarchical clustering, more objective methods for finding the number of clusters are needed. Various such methods have been proposed and some of these are described in the next chapter.

Chapter 3

Estimating the number of clusters

3.1 Introduction

A key challenge in both K-means clustering and hierarchical clustering is to determine the number of clusters, K, to split the data in. In K-means clustering, the user is required to select K before the clustering starts, while in hierarchical clustering K will be the number of clusters that appears when cutting the dendrogram at a selected level. Objective and formal methods for determining the appropriate value of K are therefore sought, and several such methods have been suggested over the years. Gordon (1999) categorizes such methods as either *local* or *global*. Local methods use some statistic to decide whether a single cluster should be subdivided, and this proceeds until the null hypothesis of a single cluster is not rejected. These methods thus use only part of the data at each stage, except at the first stage when evaluating whether the data set should be split into separate clusters at all. This implies that the local methods may only be applied to hierarchically-nested partitions. A disadvantage of local methods is that they usually require the specification of a threshold level or a significance level to determine whether the statistic is large (or small) enough to imply that a cluster should be subdivided, and the value of this will typically depend on the data set in question. Among others, Duda & Hart (1973) and Beale (1969) propose different local methods (see Gordon, 1999).

Global methods, on the other hand, evaluate some statistic over the entire data set, and try to optimize this statistic as a function of the number of clusters, K. Many global methods have been suggested. Historically, much interest has been taken in how measures of within-cluster homogeneity and between-cluster heterogeneity can be used to find the optimal number of clusters. Hence, many methods have been based on some statistic that takes one or both of these measures into account, e.g. Calinski & Harabasz (1974), Hartigan (1975), Krzanowski & Lai (1988), Kaufman & Rousseeuw (1990). A small disadvantage of many such methods is that they do not evaluate the possibility that the data should not be divided into separate clusters at all. A more recent proposal by Tibshirani et al. (2001) is also based on within-cluster homogeneity, but the additional comparison to a reference distribution allows for the testing

of there being only one cluster. In later years more interest has been directed toward methods that uses prediction error to evaluate the quality of the clustering (e.g. Dudoit & Fridlyand (2002), Tibshirani & Walther (2005), Kapp & Tibshirani (2007)).

In the following several methods for finding the number of clusters will be described. These include the Gap algorithm (Tibshirani et al., 2001), and some alternative, and novel, versions of this method, the Silhouette method (Kaufman & Rousseeuw, 1990), the Prediction strength method (Tibshirani & Walther, 2005) and the In-group proportion method (Kapp & Tibshirani, 2007). Hence, mainly newer methods are considered in this thesis. In subsequent chapters, the different methods will be evaluated using real data sets and simulations.

Since the within-cluster homogeneity is a central concept in many methods, a measure of this is described first.

3.2 Total within-cluster dispersion (W_K)

A goal in all clustering algorithms is to gather similar or close objects in the same cluster, leading the within-cluster homogeneity to be as small as possible. In contrast to Chapter 2, we now consider the allocation of the objects to be fixed (for a given number of clusters K), and the issue is to determine K. A natural measure of homogeneity within the clusters will be the dispersion of the objects within the clusters (e.g. Hastie et al., 2001). The *total within-cluster dispersion*, denoted W_K, can be defined as

$$W_K = \sum_{k=1}^{K} \frac{1}{2n_k} D_k, \tag{3.1}$$

where K is the number of clusters, n_k is the number of objects in the k'th cluster and D_k is the sum of the pairwise distances between the objects in cluster k. Using squared Euclidean distances we have

$$D_k = \sum_{i \in C_k} \sum_{j \in C_k} ||\mathbf{x}_i - \mathbf{x}_j||_2^2 = 2n_k \sum_{i \in C_k} ||\mathbf{x}_i - \overline{\mathbf{x}}_k||_2^2,$$

where the p-dimensional vector $\mathbf{x}_i$ represents a random object measured on p variables, and $\overline{\mathbf{x}}_k$ is the centroid vector of the k'th cluster. Thus, for squared Euclidean distance, we obtain

$$W_K = \sum_{k=1}^{K} \sum_{i \in C_k} ||\mathbf{x}_i - \overline{\mathbf{x}}_k||_2^2. \tag{3.2}$$

Note that this measure is identical to $W(C)$ which was defined in Section 2.3.1. In the notation above, however, the subscript K emphasizes that the measure depends upon the choice of the number of clusters, K, while the allocation, C, of objects to clusters is now considered fixed for any given K.

3.2.1 The behaviour of W_K

W_K quantifies the extent to which objects in the same clusters tend to be close to each other. One approach to the problem of determining the number of clusters may therefore be to examine how the choice of K affects W_K and then select the K that minimizes W_K. However, for many clustering methods it is often seen that W_K decreases monotonically as a function of K, and the K that minimizes W_K is therefore usually found at $K = N$. That is, W_K is minimized when each cluster consists of just one object. Intuitively, this seems reasonable since increasing the number of cluster centroids over the feature space will tend to bring the objects closer to the centroids. This will in turn decrease W_K. For hierarchical clustering we can show that this must always be the case.

Lemma 1. *For hierarchical clustering methods, W_K decreases monotonically as a function of K.*

Proof. Consider the dendrogram in Figure 2.2. Say we cut the dendrogram at a height that produces K clusters, and calculate W_K. If no merge occurs at the same height in the dendrogram, increasing K by one must involve dividing one of the original K clusters into two new clusters. If two or more splits occur at the same height, W_{K+1} is not defined, although we may break the tie by arbitrarily (or according to some rule) by selecting one cluster to divide. Since the other clusters remain the same, the only change in W_K must come from the division of one original cluster. To show that W_K must decrease (or at least stay the same) when K is increased, it is thus sufficient to show that the dispersion in the original cluster (before the split) was larger or equal to the sum of the dispersions of the two new clusters. More formally, we must show that

$$\sum_{i \in C} ||\mathbf{x}_i - \overline{\mathbf{x}}||^2 \geq \sum_{i \in C_A} ||\mathbf{x}_i - \overline{\mathbf{x}}_A||^2 + \sum_{i \in C_B} ||\mathbf{x}_i - \overline{\mathbf{x}}_B||^2, \tag{3.3}$$

where C denotes the original cluster, C_A and C_B denotes the two new clusters resulting from the division, and $\overline{\mathbf{x}}$, $\overline{\mathbf{x}}_A$ and $\overline{\mathbf{x}}_B$ represent the centroid vectors of C, C_A and C_B, respectively. Define $f_A(\mathbf{c}) = \sum_{i \in C_A} ||\mathbf{x}_i - \mathbf{c}||^2$ and $f_B(\mathbf{c}) = \sum_{i \in C_B} ||\mathbf{x}_i - \mathbf{c}||^2$, where $\mathbf{c} = [c_1, \ldots, c_p]^T$ represents some point in the feature space. We want to find the $\mathbf{c}$'s that minimize $f_A(\mathbf{c})$ and $f_B(\mathbf{c})$, respectively. These functions are strictly convex and hence have unique global minima that satisfy the stationarity conditions $\nabla f_A(\mathbf{c}) = 0$ and $\nabla f_B(\mathbf{c}) = 0$ respectively. Since $||\mathbf{x}_i - \mathbf{c}||^2 = \sum_j (x_{ij} - c_j)^2$, we have for f_A that

$$\frac{d}{dc_j} \sum_{i \in C_A} ||\mathbf{x}_i - \mathbf{c}||^2 = \frac{d}{dc_j} \sum_{i \in C_A} (x_{ij} - c_j)^2 = -2 \sum_{i \in C_A} (x_{ij} - c_j),$$

for $j = 1, \ldots, p$. Setting this equal to 0 and denoting by n_A the number of objects in cluster

A, gives us

$$\sum_{i \in C_A} x_{ij} = n_A c_j,$$

from which we find

$$\hat{c}_j = \frac{1}{n_A} \sum_{i \in C_A} x_{ij}$$

Using vector notation we get $\hat{\mathbf{c}} = \overline{\mathbf{x}}_A$. In other words, the unique global minimum for $f_A(\mathbf{c})$ is achieved when $\mathbf{c}$ is the centroid of the objects in cluster A. Similarly, we find that the minimum of $f_B(\mathbf{c})$ is found for $\mathbf{c} = \overline{\mathbf{x}}_B$. Since these are the global minima, we have that

$$\sum_{i \in C_A} ||\mathbf{x}_i - \overline{\mathbf{x}}_A||^2 \leq \sum_{i \in C_A} ||\mathbf{x}_i - \overline{\mathbf{x}}||^2,$$

and

$$\sum_{i \in C_B} ||\mathbf{x}_i - \overline{\mathbf{x}}_B||^2 \leq \sum_{i \in C_B} ||\mathbf{x}_i - \overline{\mathbf{x}}||^2,$$

Hence, by adding the left hand sides and the right hand sides of the two equations, we get (3.3) and we have shown that this relationship holds. Also, assuming that a sensible linkage method is used in the clustering procedure, we should never get $\overline{\mathbf{x}}_A = \overline{\mathbf{x}}_B = \overline{\mathbf{x}}$ for the two new clusters. This means that the inequality in (3.3) holds strictly, and it is thus shown that, in the case of hierarchical clustering, W_K is always a decreasing function of K. □

It is harder to make conclusive statements about the behaviour of W_K when K-means clustering is used. While increasing K by 1 in hierarchical clustering always means that one of the existing clusters must be divided into two new clusters, this need not be the case in K-means clustering. Hence it is not for sure that W_K will decrease monotonically as a function of K. The result of K-means clustering depends, as mentioned in Section 2.3.1, on the initial positioning of the K centroids, and this initialization will therefore also affect W_K. We could, however, devise a variant of the K-means algorithm that would consistently split one of the existing clusters into two as K is increased. This variant would then satisfy (3.3). Consider two runs of the K-means algorithm, denoted Run_K and Run_{K+1}, where the number of clusters is K and $K+1$, respectively. A possible variant of the algorithm could be to let the K cluster centroids resulting from Run_K define K of the initial centroids in Run_{K+1}. The initial position of the $(K+1)$'st centroid in Run_{K+1} could then be chosen such that a split is ensured in the cluster from Run_K with the largest dispersion. One way to do this could be to determine the object closest to the centroid of the cluster with the largest dispersion, and then let this point define the initial position of the $(K+1)$'st centroid. If the cluster with the largest dispersion in Run_K is sufficiently separated from all other clusters, this procedure will result in an original cluster being split into two clusters as K increases. Hence (3.3) will hold and W_K will thus be a decreasing function of K for this variant of K-means clustering. When the cluster with

the largest dispersion is not sufficiently separated from all other clusters, other assumptions will have to be specified in order to ensure that the procedure will lead to the division of an original cluster.

3.2.2 Using W_K to determine the optimal number of clusters

How can we use W_K to determine the number of clusters in a data set when the general tendency is for W_K to decrease monotonically as K gets larger? The solution $\hat{K} = N$ is obviously not a useful one, since this tells us nothing about the cluster structure of the data. However, if we assume that there actually exists K^* natural distinct groupings in the data set, one may expect that the curve of W_K exhibits a change in behaviour at $K = K^*$. That is, under the assumption of K^* natural clusters, we expect that W_K will decrease faster for $K < K^*$ than for $K > K^*$ (Hastie et al., 2001). This occurs because

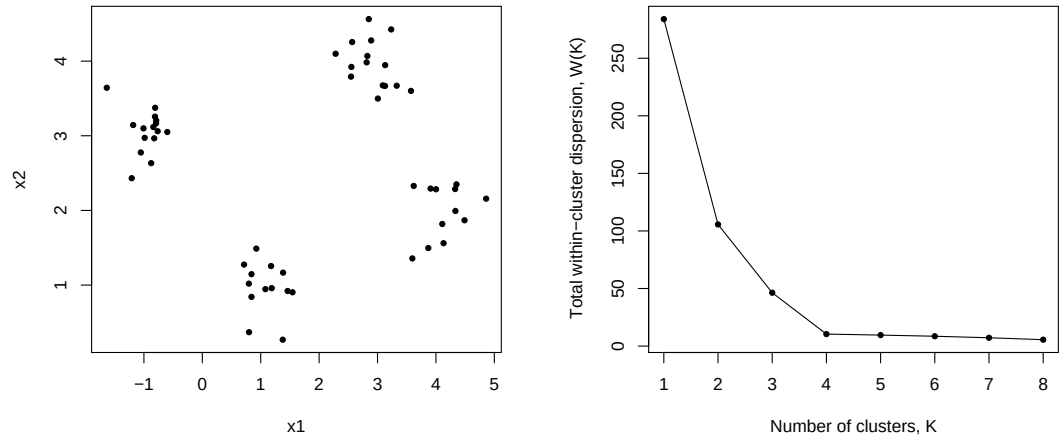
- for $K < K^*$, the true underlying groups will probably be subsets of the clusters found so far. This is because objects that belong to the same natural group will tend to be placed in the same cluster. As K increases, the natural groups will successively appear as separate clusters, and consequently make W_K decrease substantially. Hence we expect $W_{K+1} << W_K$.
- for $K > K^*$, on the other hand, increasing K will usually result in at least one of the natural groups being divided into two subgroups. Though this will still lead to a decrease in W_K, the amount it decreases by will probably be much smaller than for $K < K^*$. This effect seems logical, since splitting a natural cluster of close objects into two clusters should be expected to decrease the within-cluster dispersion by less than what splitting the union of two well-separated clusters does. Hence we expect $W_{K+1} < W_K$, but not $W_{K+1} << W_K$.

Given that the scenario above holds, one should expect to observe a sharp decrease in the successive differences of the within-cluster dispersions, $W_K - W_{K+1}$, at $K = K^*$. The estimate for the optimal number of clusters, $\hat{K}$, may thus be found by identifying a "kink" in the plot of W_K.

Figure 3.1 illustrates such a kink in the graph of W_K. The true number of clusters in the data set in the left panel is four, and from the plot we can easily see that the kink in the curve of W_K is located at $K = 4$.

3.3 Gap

One method that bases itself on the heuristic explained above, is *Gap* (Tibshirani et al., 2001). In this method, the logarithm of the total within cluster dispersion ($\log(W_K)$) is calculated for the observed data and for data generated from an adequate reference distribution. This



Data set with four clusters Graph of W_K

Figure 3.1: An illustration of how W_K may decrease as a function of K and how the kink in the curve of W_K may be used to determine the number of clusters present in a data set. The kink is located at $K = 4$, and this is in fact the natural number of clusters found in the data set in the left panel.

is done for different choices of K, and the gap between the two resulting curves is then used to determine the optimal number of clusters $\hat{K}$. This is essentially an automatic method for locating the kink mentioned earlier that takes into account how W_K is expected to decrease under a null hypothesis of no cluster structure in the data.

3.3.1 Definition and choice of reference distribution

Formally, the Gap statistic is defined as

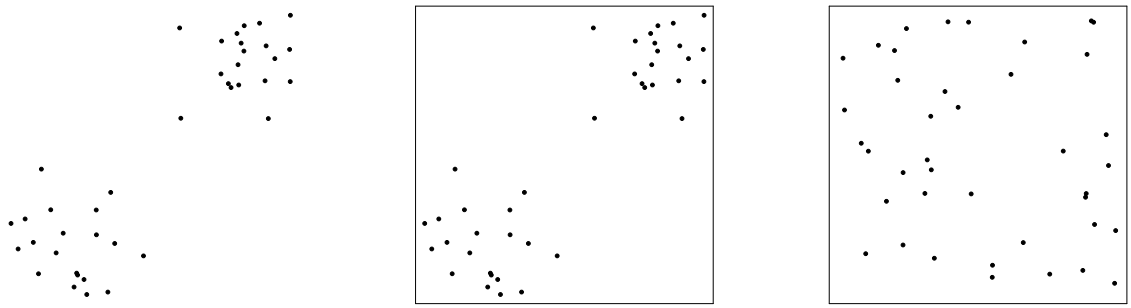
$$\text{Gap}(K) = E^*(\log(W_K)) - \log(W_K) \tag{3.4}$$

where E* denotes the expectation of $\log(W_K)$ under a sample size N from a reference distribution. The choice of an appropriate reference distribution is thus an important aspect of the Gap statistic. By using a reference distribution, this method makes it possible to test a null hypothesis of one single cluster (i.e. $K = 1$), and will only reject this null model if evidence that $K > 1$ is found. Tibshirani et al. (2001) suggest two methods for generating reference data sets. Given a $N \times p$ data matrix X, where the rows (samples) are to be clustered, these methods can be described as follows:

Range method: Generate each reference feature from a uniform distribution over a box aligned with the feature axes of the observed values, that is, within the range of the observed values for that feature. Thus, for each $j = 1, \ldots, p$, generate values from the uniform distribution $U[a_j, b_j]$, where $a_j = \min_{1 \le i \le N}(x_{ij})$ and $b_j = \max_{1 \le i \le N}(x_{ij})$. This gives the final reference set Z. Figure 3.2 illustrates the generation of reference data using the Range method (in 2

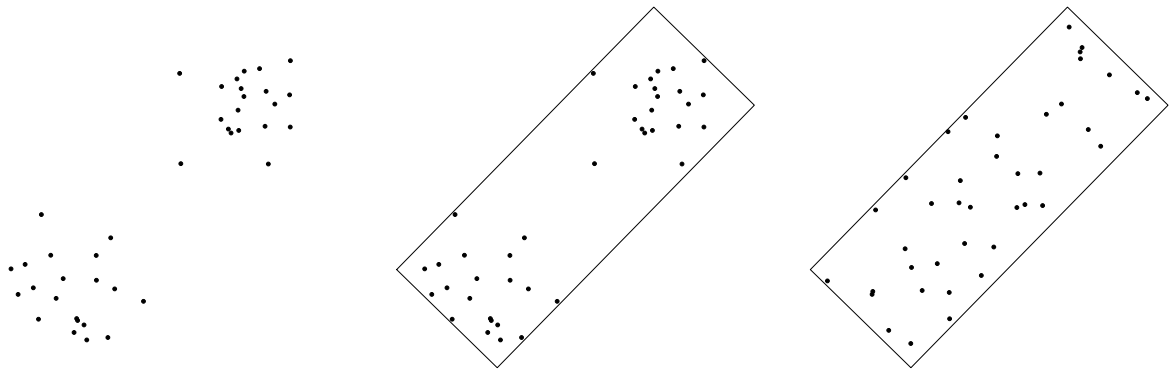
dimensions).

PC method: Generate each reference feature from a uniform distribution over a box aligned with the principal component axes of the observed data. In detail, the columns in X are first mean-centered and then the singular value decomposition $X = UDV^T$ is computed. Then the transformation $X' = XV = UD$ is made and values Z' are drawn uniformly on the ranges of the columns in X' as described in the Range method. The last step is to backtransform via $Z = Z'V^T$ and then add the earlier subtracted column mean. This gives the final reference set Z. Figure 3.3 illustrates how reference data are generated using the PC method (in 2 dimensions).



Data | Bounding box aligned with feature axes | Reference data

Figure 3.2: Illustration of the generation of reference data using the Range method.



Data | Bounding box aligned with p.c. axes | Reference data

Figure 3.3: Illustration of the generation of reference data using the PC method.

The essential difference between the methods is that the PC method takes the shape or geometry of the observed data into account, whereas the range method does not. This generally makes the PC method more accurate and hence usually preferable, though the range method is slightly simpler to implement. Unless otherwise stated, the PC method will therefore be used

in the later applications of the Gap method.

3.3.2 Using Gap to determine the number of clusters.

In order to use the Gap statistic to determine the optimal number of clusters, Gap(K) has to be calculated for different choices of K. To do this, B reference sets, denoted $Z_1, \ldots, Z_B$, are first generated using one of the methods described above. Then, for $K = 1, \ldots, K_{max}$, where K_{max} is the maximum number of clusters to be considered, the observed data are clustered into K groups, and the same is done for each reference set. The total within-cluster dispersion of the observed data (W_K) and the total within-cluster dispersion of the B reference sets ($W_K^{*1}, \ldots, W_K^{*B}$), may then be calculated. Hence, the Gap statistic given by (3.4) may be found, where $\mathrm{E}^*(\log(W_K))$ is taken to be the average value of $\log(W_K^{*b})$ over the B reference sets. As discussed in Section 3.2, the total within-cluster dispersion is expected to decrease monotonically as K increases, and this applies for the clustering of the observed data as well as the clustering of the reference data. However, since the reference data are generated from a uniform distribution, they should not exhibit any cluster structure and hence the curve of ave(W_K^{*b}) should decrease in a smooth fashion. On the other hand, the curve of W_K will (as described earlier), under the assumption of K^* natural clusters, tend to decrease faster for $K < K^*$ than for $K > K^*$ and will therefore exhibit a kink at $K = K^*$. The greatest difference or the largest gap between the curves will probably be located at this kink and the largest gap thus provides a useful criterion for determining the optimal number of clusters.

However, since W_K also continues to decrease after the kink (though at a slower rate), the gap(s) at $K > K^*$ may actually be somewhat (but not much) larger than the gap at $K = K^*$. Hence, it is not enough to just locate the largest gap, one must also check if it is "significantly" larger than the preceding gap. To do this, Tibshirani et al. (2001) suggest calculating the standard errors of the Gap statistic for each K, and then check whether the gap minus its standard error is greater than the preceding gap. If this holds, the gap is considered to be significantly larger than the preceding gap. The authors have found empirically that using one standard error works quite well in this setting. Note, however, that this rule is liberal compared to standard statistical hypothesis testing where it is more common to use at least two standard errors.

Based on the above reasoning, the following criterion is used to estimate the number of clusters in the Gap algorithm: Choose the smallest K for which Gap(K) is greater or equal to Gap($K+1$) $- s_{K+1}$, where s_{K+1} is the standard error of Gap($K+1$). Note that choosing the smallest K that satisfies the criterion, may result in disregarding the K that actually maximizes Gap(K). In some cases this ensures that $\hat{K}$ is not chosen to be too large, while in other cases it leads to a clear underestimate of the number of clusters. With these aspects in mind, one should not solely rely on the result from the automatic criterion, but also study the entire gap curve (with standard errors). This is also pointed out by the authors, especially in cases

where there are sub-clusters within larger well-separated clusters, which can lead to several local maxima in the gap curve. An example of this is given later in this chapter.

In summary, the algorithm for calculating the Gap statistic and using it to determine the number of clusters, consists of the following steps:

1. For $K = 1, \ldots, K_{max}$ (where K_{max} is the maximum number of clusters to be considered), cluster the observed data into K groups and calculate W_K.

2. Generate B reference data sets from a uniform distribution using either the range method or the PC method. For $K = 1, \ldots, K_{max}$, cluster each reference set and calculate W_K^{*b} $(b = 1, \ldots, B)$. Then compute the Gap statistic

$$\text{Gap}(K) = \frac{1}{B} \sum_{b=1}^{B} \log(W_K^{*b}) - \log(W_K) \tag{3.5}$$

3. Compute the standard deviation of $\log(W_K^{*b})$

$$\text{sd}_K = \sqrt{\frac{1}{B} \sum_{b=1}^{B} \left(\log(W_K^{*b}) - \bar{l}\right)^2},$$

where $\bar{l} = (1/B) \sum_{b=1}^{B} \log(W_K^{*b})$, and define the total standard error

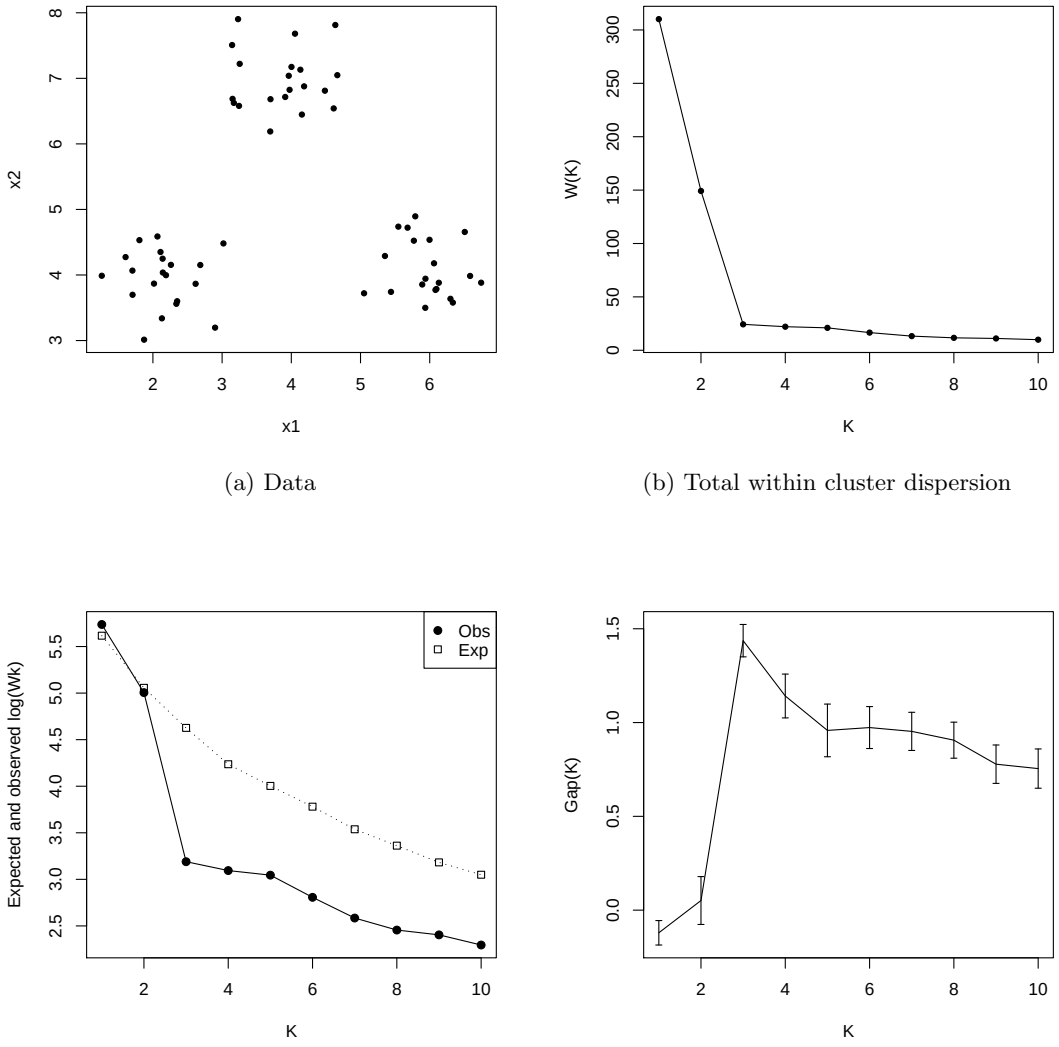
$$s_K = \sqrt{1 + 1/B}\ \text{sd}_K.$$

4. Finally, determine the optimal number of clusters, $\hat{K}$, from the *Gap criterion*: let $\hat{K}$ be the smallest K such that $\text{Gap}(K) \geq \text{Gap}(K + 1) - s_{K+1}$.

Figure 3.4 shows some results from running the Gap algorithm on a data set with three natural clusters. The top left panel shows the simulated data, while the top right panel shows W_K as a function of K. We can see that there is a noticeable kink in the curve at $K = 3$. The bottom left panel shows the functions $\log(W_K)$ and $\text{E}^*(\log(W_K))$, represented by circles and squares, respectively, while the bottom right panel shows the gap curve plotted with its ± 1 standard errors. It is quite easy to see that $K = 3$ is the smallest K that satisfies $\text{Gap}(K) \geq \text{Gap}(K + 1) - s_{K+1}$.

3.4 Alternative versions of the Gap algorithm

Though the Gap algorithm described above provides a useful criterion for determining the optimal number of clusters, alternative versions or extensions may provide further insight into the problem. One interesting extension may be to look for additional clusters within the $\hat{K}$ clusters found by the Gap algorithm. A different version may be to consider alternatives to the



(a) Data

(b) Total within cluster dispersion

(c) Observed and expected $\log(W_K)$

(d) Gap curve

Figure 3.4: Using the Gap statistic to determine the number of clusters in a data set. The top left panel shows a data set with three clusters, while the top right panel shows W_K as a decreasing function of K. Note the kink at $K = 3$. At the bottom, the left panel shows the expected and observed curves of $\log(W_K)$. The right panel shows the gap curve for $K = 1, \ldots, 10$, and it is quite easily seen that $K = 3$ is the smallest K that satisfies the Gap criterion.

way the reference data are generated and used in the original version. In the following sections both of these approaches will be discussed further.

3.4.1 Reference Gap

In the Gap method, reference data are generated using the entire data set, either by the Range method or the PC method. Both methods generate reference data sets that are not expected to exhibit any cluster structure. A relevant question to consider may be whether it is appropriate

to use such reference data for each value of K. To test a hypothesis of no cluster structure versus two clusters in the observed data, it does seem reasonable to compare the total within-cluster dispersion for two clusters of observed data to the total-within cluster dispersion for two clusters of reference data with no real cluster structure. However, it is not as intuitively clear that one should use a "no-cluster" reference distribution to test the hypotheses of there being three clusters versus two, four clusters versus three and so on. One could instead consider using different reference distributions for different values of K, where the parameters of the reference distribution are determined by the range and shape of the objects allocated to $K-1$ clusters.

On this account, I suggest an alternative version of the Gap algorithm that uses different reference distributions according to the value K. This method will be referred to as *Reference Gap*. The reference data in this version are generated by the following strategy:

- For $K = 1$: Use the PC method as suggested in Gap to generate reference data sets.
- For $K = 2, \ldots, K_{max}$: Use the $K-1$ clusters found for the previous value of K to determine the uniform distribution from which reference data should be generated. That is, for $c = 1, \ldots, K-1$, generate n_c reference points from a uniform distribution over a box aligned with the principal component axes of the observed data found in cluster c (where n_c is the number of objects in cluster c). This should result in reference data sets with $K-1$ clusters.

The reference data are then clustered into K clusters, and used to calculate $\text{Gap}(K)$ from (3.5).

Figure 3.5 illustrates the idea for a data set with three clusters. The left panels show the observed data, clustered into K clusters ($K = 2, \ldots, 5$), where the different clusters are represented by different colours. The corresponding right panels shows the reference data used in the calculation of $\text{Gap}(K)$, which have been generated over a box aligned with the principal components of the objects in the $K-1$ clusters on the left side. (The basis for reference data generated for $K = 2$, which is one single cluster, is not shown.) Note that the reference data get more and more similar to the observed data as K increases. In particular, for $K > K^*$ (where $K^* = 3$ in this case), the reference data are very similar to the observed data. Hence when the reference data are clustered into K groups, and $K > K^*$, we can expect the total within cluster dispersion of the reference data ($\text{ave}(W_K^{*b})$) to be close to the total within cluster dispersion of the observed data (W_K). In effect we also expect the calculated gaps to be small for $K > K^*$. Further, at $K = K^*$, W_K should be expected to decrease substantially because this represents the optimal clustering of the data. W_K^{*b} should also decrease when compared to W_{K-1}^{*b}, but probably by much less than W_K because the reference data are still quite dispersed compared to the observed data. Hence it seems reasonable to expect that the largest drop in the gap curve occurs at $K = K^*$. For $K < K^*$ it is difficult to predict the behaviour of the gap curve. The gaps may be quite large (especially for high-dimensional data), but the curve

will probably not have drops of similar magnitude as at $K = K^*$.

The top panels of Figure 3.6 show the expected and observed curves of $\log(W_K)$ for the original Gap method and the Reference Gap method, when applied on the data set in Figure 3.4a. Note that whereas the expected curve for the original Gap method decreases in a slow, smooth fashion, the equivalent curve for the Reference Gap version has a shape that is much more similar to the observed curve. Hence the gap between the curves is much smaller for most values of K in the Reference Gap method than the original Gap, as seen in the bottom panels where the gap curves are plotted with ± 1 standard errors for the two methods. Also note that the gap curve for the Reference Gap version has a peak and thereafter a substantial drop at $K = 3$, and that for larger number of clusters the gap is very small (and sometimes negative).

Motivated by the reasoning above, and the fact that the Reference Gap curve is very different from the original Gap curve, it does not seem advisable to use the same criterion as the one used in the original Gap method. Hence I suggest a different criterion for determining the optimal number of clusters in the Reference Gap method. First, for all K where $\text{Gap}(K) < 0$, define $\text{Gap}(K) = 0$. Negative gaps occur when the total within-cluster dispersion of clustered reference data is smaller than the observed total within-cluster dispersion, and such negative gaps may impair the criterion suggested below. Let $\text{Diff}(K)$ denote the difference between a gap and the succeeding gap, i.e. $\text{Diff}(K) = \text{Gap}(K) - \text{Gap}(K+1)$. We may say that this difference is significant if the difference is larger than 2 standard errors. Let $\mathcal{K}$ represent a set that includes all K for which $\text{Gap}(K) - 2s_K > \text{Gap}(K+1)$. Then, finally, estimate the optimal number of clusters by the K for which $\text{Diff}(K)$ is maximized, under the condition that this difference is significant:

$$\hat{K} = \begin{cases} 1 & \text{if } \mathcal{K} = \emptyset \\ \underset{K \in \mathcal{K}}{\operatorname{argmax}} \, \text{Diff}(K) & \text{otherwise} \end{cases} \tag{3.6}$$

If no value of K gives a gap that is significantly larger than the next gap, we define $\hat{K} = 1$. Note that a problem with this criterion is that it is not useful for data sets that has no real cluster structure. This is because the reference data will be similar to the observed data already at $K = 1$, hence, there is no reason to expect a particularly large drop in the gap curve from $K = 1$ to $K = 2$.

3.4.2 Recursive Gap

Solberg (2007) suggests a recursive variant of the Gap method that makes it possible to check for additional clusters within the clusters found by Gap. This version starts by running the Gap algorithm on the data set, resulting in an estimate of $\hat{K}$ clusters. Then, for each of the $\hat{K}$ clusters, the Gap algorithm is used again to look for sub-clusters and this is repeated recursively until no new sub-clusters are found. The algorithm finally returns the total number of clusters.

In summary, this version of the Gap algorithm, referred to as *Recursive Gap*, consists of

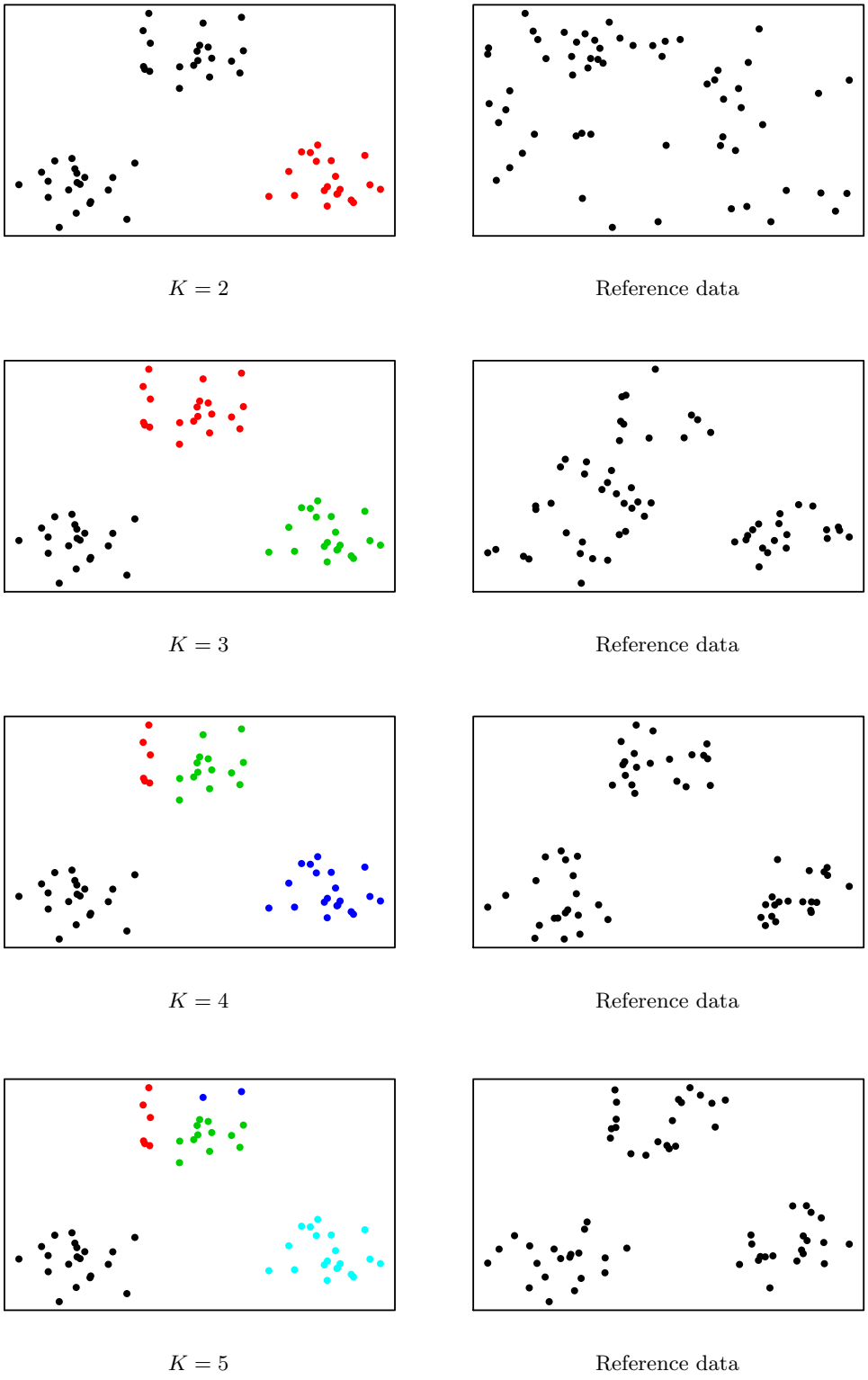
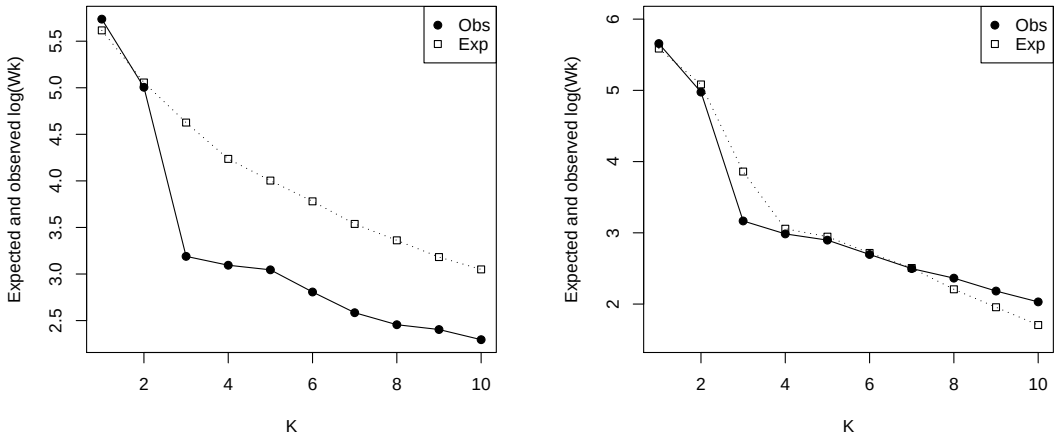
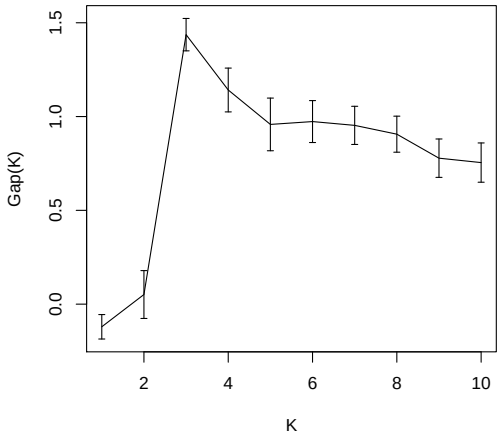


Figure 3.5: Generation of reference data in the Reference Gap method. To the left, the observed data are clustered in K clusters ($K = 2, \ldots, 5$), represented by different colours, while the corresponding right panels show the reference data generated using the $K - 1$ clusters in the left panels.

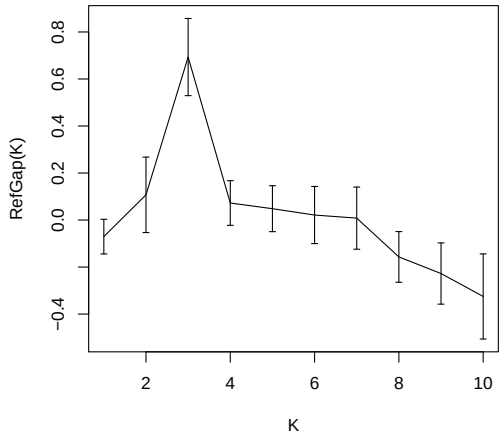


Original Gap

Reference Gap



Original Gap



Reference Gap

Figure 3.6: The two top panels show the observed and expected curves of $\log(W_K)$ when applying the original Gap method and the Reference Gap method on the data set shown in Figure 3.4a.. For the Reference Gap method, the expected curve is closer to the observed curve than what is the case for the original Gap method. The two bottom panels show the corresponding gap curves (plotted with ± 1 standard error) for the original Gap method and the Reference Gap version.

the following steps:

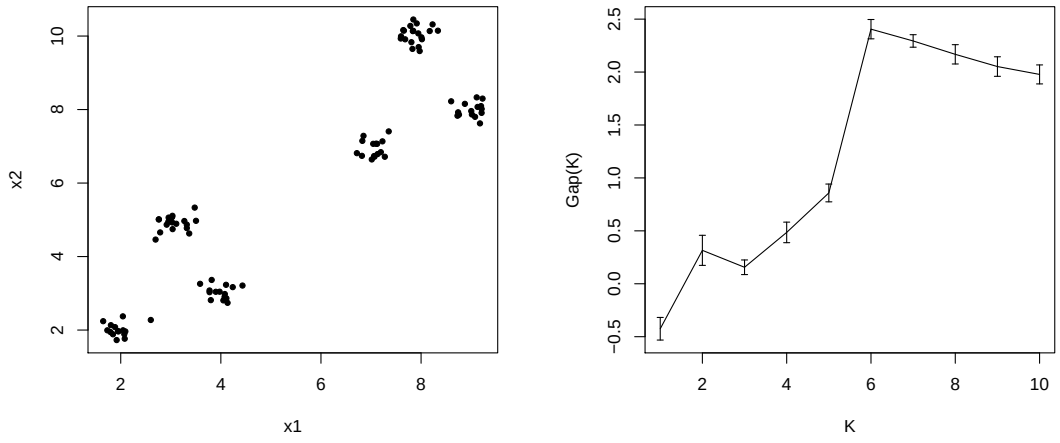
1. Run the Gap algorithm as described in Section 3.3. This results in an estimate of the optimal number of clusters for this run, $\hat{K}$.

2. If $\hat{K} > 1$, repeat steps 1 and 2 recursively for each of the $\hat{K}$ clusters. Otherwise, terminate the recursion.

3. If $\hat{K} = 1$, return the value 1. Otherwise, add the return-values from the recursive runs

and return this sum.

Recursive Gap is particularly useful when the data set consists of "clusters of clusters". Consider a simple, illustrative example such as the data set shown in the left panel of Figure 3.7. This data set consists of two main groups, each made up of three clusters. We may refer to the two main groups as "super-clusters", and the clusters found within each of these as "sub-clusters". The right panel of Figure 3.7 shows the gap curve after running the Gap algorithm on the data set. Though the largest gap is found at $K = 6$ (which is the correct solution), the Gap algorithm returns $\hat{K} = 2$ as the estimate for the optimal number of clusters. This happens because the Gap criterion is satisfied already at $K = 2$. Hence, the automatic criterion only finds the two super-clusters, and not the sub-clusters within these. Note, however, that the plot of the gap curve, reveals that the curve has a local maximum at $K = 2$ and a global maximum at $K = 6$ (for $1 \leq K \leq 10$). There is thus evidence for $\hat{K} = 6$ as well.



Data Gap curve

Figure 3.7: The left panel shows a data set consisting of two groups of clusters, where each group is made up of three clusters. The right panel shows the gap curve after running the Gap algorithm on the data set. The smallest K that satisfies the Gap criterion is $K = 2$, hence this is the estimated optimal number of clusters automatically returned by the original Gap algorithm. Notice, however, that the gap curve is maximized for $K = 6$ (the true number of clusters), and this indicates that this may be an appropriate solution as well.

Recursive Gap continues to look for additional sub-clusters within the two clusters found by Gap for the data set in Figure 3.7. Figure 3.8 shows the gap curves resulting from the recursive runs. The top panels show the gap curves from running the Gap algorithm on each of the two super-clusters (A and B) found in the first run. These curves both result in an estimate of three clusters. The panels below show the gap curves from recursive runs on the three sub-clusters found in super-clusters A and B, to the left and right, respectively. In each of these curves the Gap criterion is satisfied for $K = 1$, hence no additional sub-clusters are found. Recursive Gap therefore correctly suggests $\hat{K} = 6$ as the optimal number of clusters,

and hence performs better than Gap in this setting.

Though useful in low-dimensional situations such as the one described above, the virtue of Recursive Gap is perhaps greatest when the data set is high-dimensional (Solberg, 2007). In such cases, Gap tends to give an estimate of the number of clusters that is lower than what is optimal. This may be because the box that is used to generate the reference sets will be extremely large, and, assuming that the observed data set is made up of more than one cluster, the actual observations will occupy only small parts of the box. Hence, the Gap algorithm will search for clusters in a large space without being able to focus in on the areas where the observations are actually located. This will often lead the Gap algorithm to return too few clusters, because the Gap criterion is satisfied too early. Hence it tends be harder to choose a higher number of clusters in high-dimensional data sets. The recursive version, on the other hand, makes it possible to "zoom" in on the parts of the box where the observations are actually located, and thus makes it possible to check for more clusters than what was found by the original Gap algorithm.

When the data have been hierarchically clustered, there is another property of the recursive version that helps improve the accuracy of cluster number estimation. As described in Section 2.3.2, the most basic strategy for producing K clusters from a hierarchical clustering is to cut the dendrogram horizontally at the height that gives K clusters. In some cases, however, horizontal cuts may not give the optimal K clusters. To illustrate this problem, consider the dendrogram shown in Figure 3.9a. This dendrogram represents three clusters; one large and somewhat dispersed cluster, and two smaller, more cohesive clusters. The upper, dotted line in the dendrogram shows the horizontal cut that will produce three clusters. Notice that this cut *does not* give us the three natural clusters, but instead produces a split of the large cluster into two clusters (of which one consists of only one sample), while keeping the two smaller clusters merged. The lower, dotted line, on the other hand, shows the first horizontal cut for which the two smaller clusters are separated. This cut will, however, produce no less than eight clusters, because the large cluster is subdivided further. Hence, horizontal cutting of this dendrogram will never give us the three optimal clusters. When applying the original Gap algorithm, the K clusters for which $\text{Gap}(K)$ is calculated are the clusters coming from horizontal cuts of the dendrogram. This implies that Gap is not able to find the three true clusters in Figure 3.9. Gap instead estimates that there are only two clusters; the large cluster and the merge of the two smaller clusters.

The recursive version, on the other hand, does not depend on a horizontal cut of the dendrogram. Since Recursive Gap uses the clusters found by the original Gap method to look for additional sub-clusters, the sub-trees representing these clusters may be cut *independently* of each other. In this dendrogram, Recursive Gap finds two sub-clusters in one of the original clusters (splitting the merge of the two smaller clusters), and no sub-clusters in the large cluster. This corresponds to cutting the dendrogram as shown in Figure 3.9b, which in fact produces

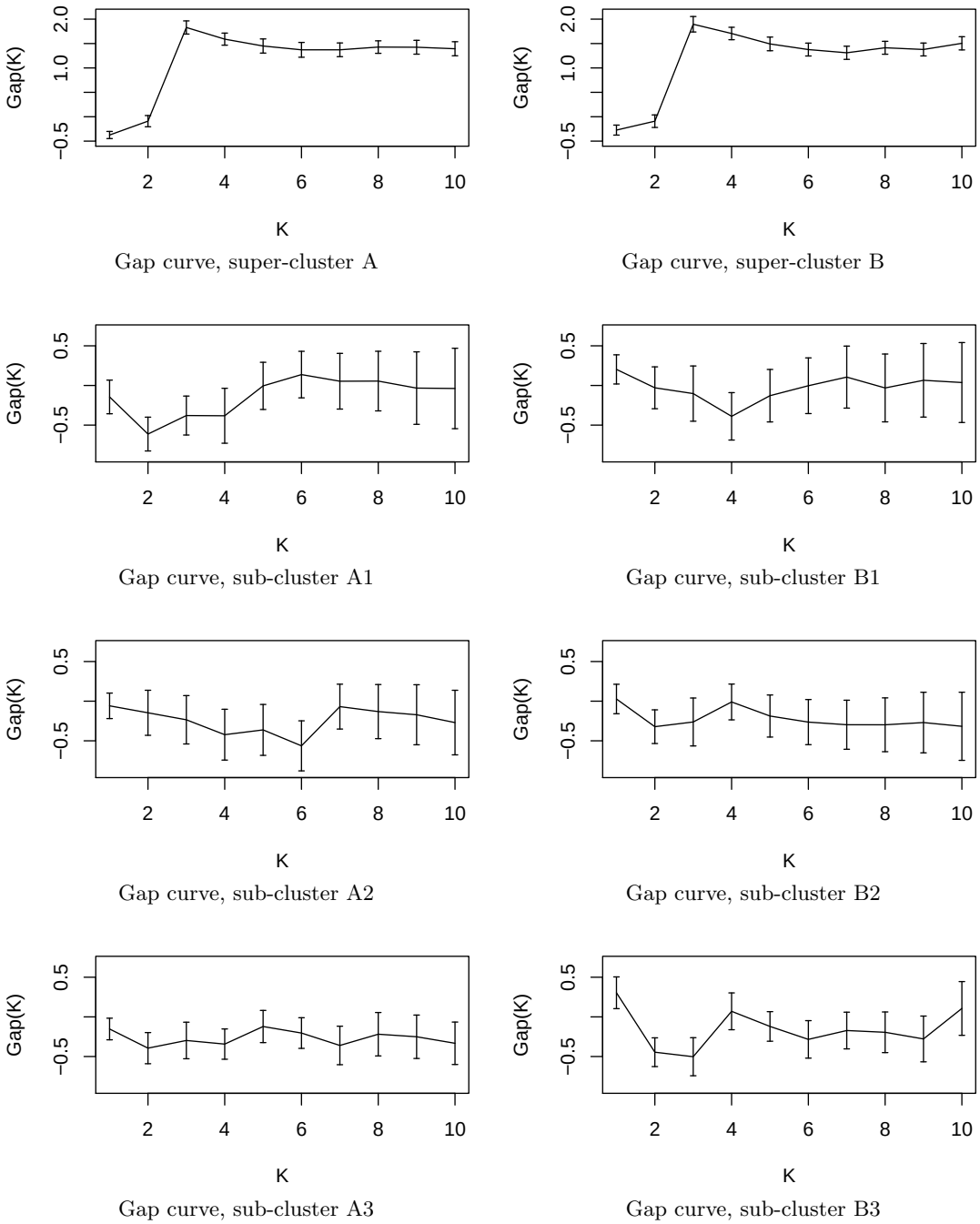
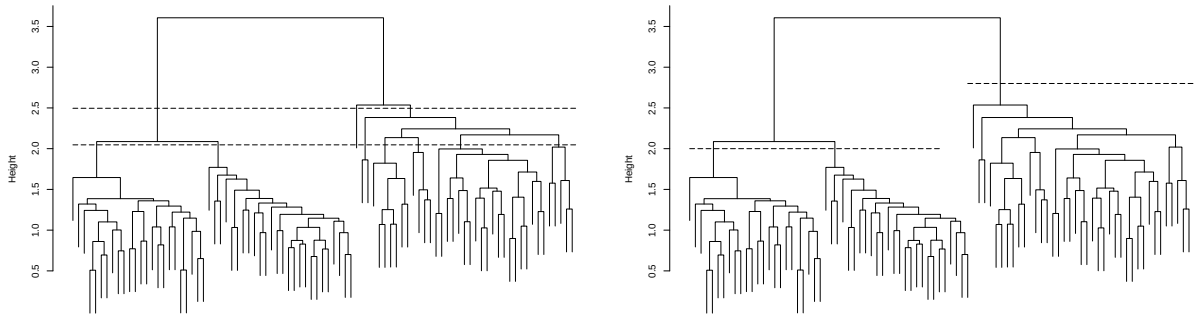


Figure 3.8: Recursive runs of the Gap algorithm on the two clusters originally found in the data set shown in Figure 3.7. The two top panels show the gap curves for the two super-clusters (denoted A and B) found in the first run, each resulting in an estimate of three sub-clusters. The subsequent panels show the gap curves for the recursive runs on the three sub-clusters found in super-clusters A and B, to the left and right, respectively. No additional sub-clusters are found in these curves, hence Recursive Gap (correctly) estimates $\hat{K} = 6$.

the three natural clusters. Hence, the recursive version has the advantages, over the original method, of the ability of "zooming in" on the observed data, as well as the ability of cutting the dendrogram in a non-horizontal way. These properties may lead to more precise estimates of the number of clusters in data sets.



(a) Horizontal cutting

(b) Recursive cutting

Figure 3.9: This figure shows the dendrogram resulting from hierarchical clustering with average linkage of a data set made up of three clusters. In the left panel, the upper, dotted line shows the height at which a horizontal cut will produce three clusters. These three clusters are, however, not the three natural clusters. The lower, dotted line shows the first horizontal cut that results in the separation of the two clusters to the left in the dendrogram. This cut also leads to further divisions of the right cluster such that a total of eight clusters are produced. The dotted lines in the right panel of the figure demonstrates how the dendrogram is cut when applying Recursive Gap. The three clusters emerging from these cuts are in fact the natural clusters in the data set.

3.4.3 Enhanced recursive algorithm (ERA)

As described above, the recursive version of the Gap algorithm tends to work better than the original version on high-dimensional data and on data consisting of "clusters of clusters". This is in part due to Recursive Gap's ability to zoom in on the observed data, and in part it's ability to cut the dendrogram in a non-horizontal way. However, Recursive Gap does not work sufficiently well in some situations. If the original Gap returns only one cluster, then Recursive Gap will necessarily also estimate that there is just one cluster, because the recursive runs are never even started. Since the recursion is stopped the minute Gap returns $\hat{K} = 1$, the method may also fail to consider sub-clusters located at lower levels.

An idea is therefore to enhance the recursive version such that it is able to handle cases where the recursion is terminated too early. This method is referred to as *ERA* (Enhanced Recursive Algorithm). As in Recursive Gap, ERA starts by running Gap on the data, resulting in an estimated number of clusters, $\hat{K}$. These $\hat{K}$ clusters may be called *significant* clusters. The Gap algorithm is then run again recursively on each of the $\hat{K}$ significant clusters. However, as opposed to Recursive Gap, the recursive runs are in this version started, and continued, even if the returned estimate is $\hat{K} = 1$. Specifically, if the Gap method estimates $\hat{K} = 1$ for a cluster, the cluster is still split tentatively into two clusters, and the Gap algorithm is run recursively

on these. If one of these *tentative* clusters is found to have significant sub-clusters, whereas the other does not, the latter is redefined as a significant cluster. This procedure is run recursively until some stopping threshold is reached. Finally, the total number of clusters to be returned by ERA will be the sum of significant clusters that do not contain any significant sub-clusters.

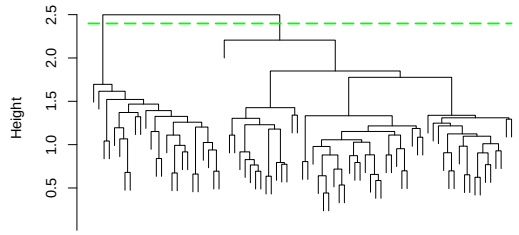
An additional requirement in this method is that all significant clusters must be made up of a minimum number of samples, n_{min} (e.g. at least 5 samples). This is reasonable, because the term cluster (or group) implies that it should consist of several objects. In many data sets one may have some samples that do not really fit in any of the clusters, and hence appear as outliers that stand out in the dendrogram. These samples are typically clustered on top of the dendrogram, or on top of one of the clusters seen in the dendrogram. Since ERA allows the splitting into two tentative clusters, and redefines one of them as significant if the other is found to have significant sub-clusters, such outliers could in many cases make the method overestimate the number of clusters. By implementing a requirement of minimum cluster size, the algorithm is able to discard outliers (such that they receive no cluster affiliation).

Figure 3.10 illustrates how ERA operates. The dendrogram in this figure comes from the hierarchical clustering with average linkage of a data set made up of four clusters. The first panel, Figure 3.10a, shows the result of the first run, where the green, dotted line represents that two significant clusters that are found. These clusters are the clusters found by Gap. The next panel, Figure 3.10b, shows the result of the first recursive run on the right cluster. The red, dotted line illustrates that no significant sub-clusters are found, and that the cluster is thus divided into two tentative sub-clusters. The third panel, Figure 3.10c, shows that no significant sub-clusters are found in the next recursive run on the right tentative sub-cluster either, as marked by the red, dotted line. Finally, Figure 3.10d shows that two significant clusters are in fact found in the next recursive run, and that the left tentative sub-cluster from Figure 3.10c is thus also defined as a significant cluster. This is marked by the green, dotted lines. The left tentative sub-cluster from Figure 3.10b is *not* taken to be a significant cluster as a consequence of the minimum cluster size requirement ($n_{min} = 5$). (The results of the other recursive runs are not shown as they did not result in the finding of any sub-clusters.) Hence ERA correctly finds a total of four clusters in this data set. This is opposed to Recursive Gap which, like Gap, only finds two clusters, since the recursive runs are stopped the minute no significant sub-clusters are found.

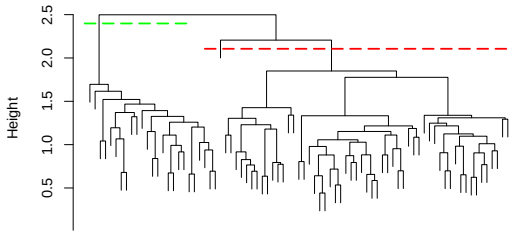
The Enhanced Recursive Algorithm (ERA) is summarized as follows:

1. Run the original Gap algorithm and get an estimate of the number clusters for this run, $\hat{K}$.

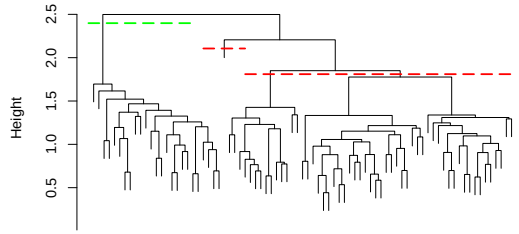
2. - If $\hat{K} > 1$, label these $\hat{K}$ clusters as *significant* on the condition that they consist of at least n_{min} samples. Then repeat steps 1 and 2 recursively for each significant cluster.



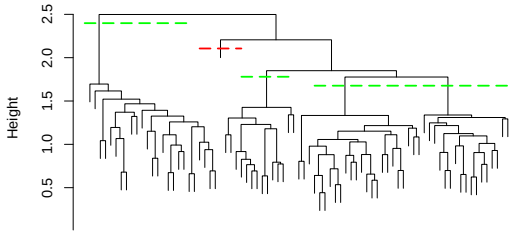
(a) Gap finds two significant clusters in the first run (marked by the green line).



(b) No significant sub-clusters are discovered in the first recursive run on the right cluster, and the cluster is hence split into two tentative sub-clusters (marked by the red line).



(c) The second recursive run does not reveal any significant sub-clusters either, and the right tentative sub-cluster is split into two new tentative sub-clusters (marked by the red line). No recursive runs are made on the left tentative sub-cluster since it consists of just one object.



(d) The next recursive run reveals two significant sub-clusters (marked by the green line). The other tentative sub-cluster from the split in Figure 3.10c is thus also defined as a significant cluster. The left tentative sub-cluster from Figure 3.10b is *not* defined as a significant cluster (as marked by the red line), due to the minimum cluster size requirement ($n_{min} = 5$).

Figure 3.10: An illustration of how ERA operates. Green lines represent the finding of significant clusters, whereas red lines represent tentative clusters. In total, four significant clusters are found by ERA, and this is in fact the correct answer for the clustered data set.

- If $\hat{K} = 1$, and some stopping threshold (described below) has not been reached, split the cluster into 2 sub-clusters. Label these clusters as *tentative*, and repeat steps 1 and 2 recursively for each of them on the condition that they consist of at least n_{min} samples. If one of the tentative clusters is found to have significant sub-clusters, whereas the other does not, the latter is re-labeled as significant (given that it is of size n_{min} or more).

3. If a cluster is significant and has no significant sub-clusters, return the value 1. If the cluster is tentative, and has no significant sub-clusters, return the value 0. Otherwise, add the return-values from the recursive runs and return this sum.

4. If no significant clusters are found throughout the recursive runs, the value 1 is returned as the final result of the method. This corresponds to no cluster structure in the data.

Note that ERA does not necessarily have to be applied in combination with the Gap algorithm. In fact, the method provides more of a general framework that may be used in

collaboration with many other methods for estimating the number of clusters.

Definition of a stopping threshold

A challenge in the algorithm described above is that one has to decide upon a stopping threshold at which the algorithm should stop looking for additional sub-clusters. For hierarchical clusterings, a suggested stopping threshold is that for a split into two tentative sub-clusters to occur, the height of the cluster considered for division must be larger than some preset threshold, h. (The height of a cluster is given on the y-axis of the dendrogram, and corresponds to the distance between the two potential sub-clusters.) The value of h could be chosen by studying the dendrogram to find a suitable height to stop at, however, this would make the method less objective and inclined to be influenced by the user's prior expectations and beliefs. Since the estimated number of clusters may very well depend on how deep into the dendrogram we are willing to search, and hence the threshold h, a better option is to run the algorithm several times with different values of h, and select the optimal threshold value, h^*, from these runs. Given H different threshold values, a possible strategy for finding h^* may be formulated as follows:

- Run the algorithm above for the H different values of the threshold h. Record the estimated number of clusters, $\hat{K}_h$, as well as the cluster membership of the samples, for each threshold.
- Calculate $\text{Gap}(\hat{K}_h)$ using (3.5). Note that since each cluster is required to consist of a minimum number of samples, some samples may not be a member of any of the $\hat{K}_h$ clusters. In such cases, these objects should not be used in the generation of reference data when calculating the gap.
- Let h^* represent the threshold h for which $\text{Gap}(\hat{K}_h)$ is maximized.
- Then repeat the algorithm R times with the threshold $h = h^*$, and finally estimate the optimal number of clusters in the data set by the most frequently estimated number of clusters over these R repetitions.

The last step in this strategy is influenced by the observation that the estimated number of clusters for a particular threshold h^* may vary over several repetitions. This is due to differences in the reference data generated in the Gap algorithm, which may, in particular at lower thresholds, sometimes lead to the finding of significant sub-clusters, while at other times not. By using the most frequently estimated number of clusters for the threshold h^* over the R repetitions, the risk of accidentally finding too many clusters is reduced. (This strategy may lead to a tie between two (or more) solutions if two (or more) solutions are reported equally frequently. In such cases a conservative rule is to take the smaller of these solutions to be the optimal number of clusters.)

The strategy above leaves the user to decide upon the number of different threshold values to apply (H). Note that the maximum number of sensible thresholds is $N-1$, since there are $N-1$ levels at which clusters merge in the dendrogram. When applying the method on simulated and real data sets in later chapters, I used the 20 % highest levels in the dendrogram, minus 1/100000, as threshold values (hence $H = 0.2 \times (N-1)$). This choice assures that every level, within the 20 % largest levels, at which a possible new significant sub-cluster may be found, is considered. At the same time the number of calculations is kept relatively low. Furthermore, it seems to work well for the situations studied in thesis.

An implementation of ERA in the software `R` is found at `http://www.ifi.uio.no/forskning/grupper/bioinf/Projects/`, along with an illustrative example of how it is used.

3.5 Silhouette

Silhouette is another method, proposed by Kaufman & Rousseeuw (1990), that is used to find the number of clusters in data sets. In this method, one basically compares the distances within a cluster to the distances to objects in other clusters. The data set is first clustered into K clusters, and then the "silhouette" of each object is calculated. For each object i, this statistic incorporates the average distance to the other objects in its own cluster ($a(i)$), and the average distance to the objects in the closest cluster besides its own ($b(i)$). More formally, if $i \in G$, then

$$a(i) = \frac{1}{n_G - 1} \sum_{j \in G} d(\mathbf{x}_i, \mathbf{x}_j)$$

and

$$b(i) = \min_{H \neq G} \frac{1}{n_H} \sum_{j \in H} d(\mathbf{x}_i, \mathbf{x}_j).$$

Here n_G and n_H are the number of objects in the clusters G and H, respectively. $a()$ is then a measure of within-cluster homogeneity, while $b()$ can be seen as a measure of between-cluster heterogeneity.

The silhouette of the object i is then defined as

$$s(i) = \frac{b(i) - a(i)}{\max\left[a(i), b(i)\right]}.$$

$s(i)$ will take a value between -1 and 1. A large silhouette value (close to 1) indicates that the object is well clustered, since this means that the distance to objects within its own cluster is small, while the distance to objects in other clusters is large. A low silhouette value (close to -1) will on the other hand indicate that an object is misclassified, since it is closer to objects in another cluster than objects in its own cluster. If $s(i)$ is close to 0, it means that $a(i)$ and $b(i)$ are almost equal. Hence it is not clear which of the clusters G and H the object should have been assigned to. Note that $a(i)$ is not defined if cluster G only contains one object. In

such cases, it is standard to set $a(i) = 0$ (Kaufman & Rousseeuw, 1990). Also note that $s(i)$ is not defined for $K = 1$, since at least one extra cluster in addition to G is required to calculate $b(i)$. Consequently, Silhouette may only be applied for $K \geq 2$, and the possibility of a single cluster is thus not considered by this method.

The average silhouette over the entire data set when the number of clusters is K will be

$$\text{Sil}(K) = \frac{1}{N} \sum_{i=1}^{N} s(i).$$

Silhouette estimates the optimal number of clusters by the K which maximizes the average silhouette, i.e.

$$\hat{K} = \underset{2 \leq K \leq K_{max}}{\text{argmax}} \ \text{Sil}(K),$$

where K_{max} is the maximum number of clusters to be considered.

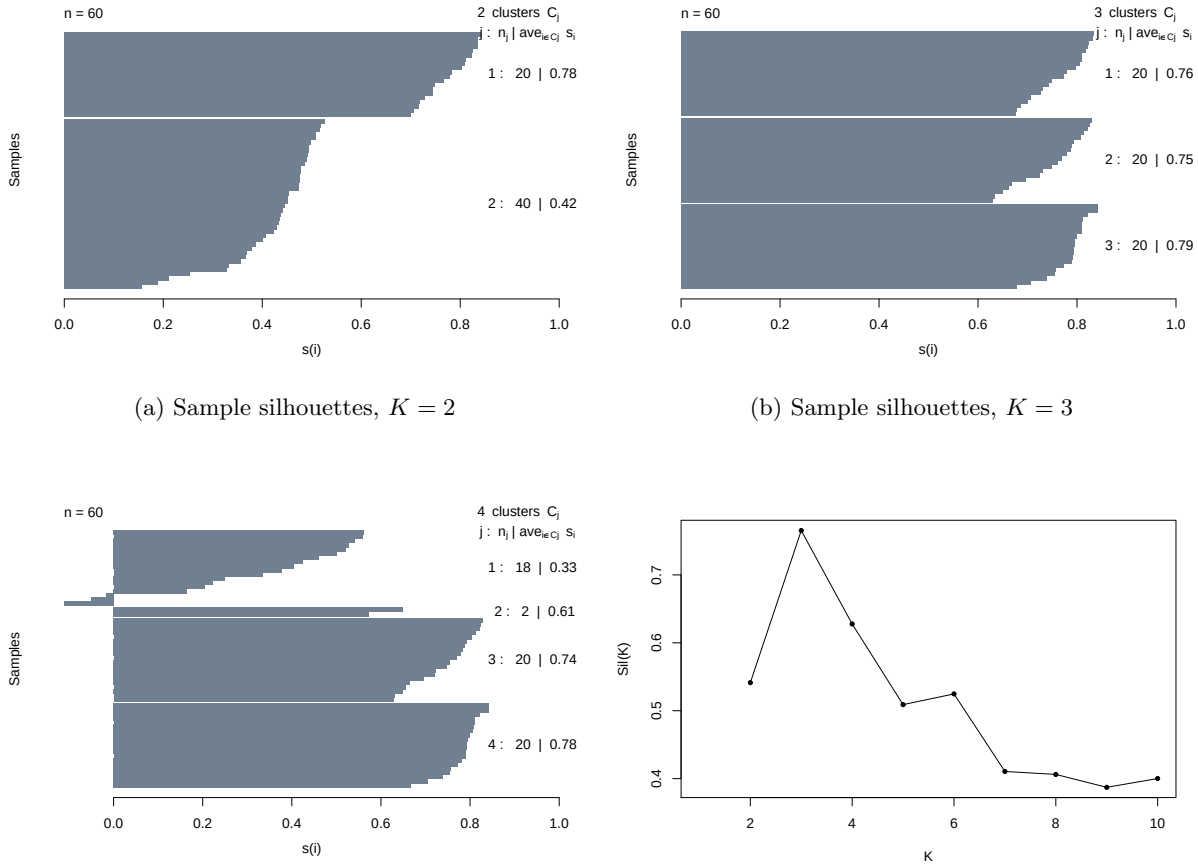
Let SC (the silhouette coefficient) represent the maximal average silhouette, i.e.

$$\text{SC} = \max_{2 \leq K \leq K_{max}} Sil(K).$$

Kaufman & Rousseeuw (1990) give the following guidelines for interpreting the value of the silhouette coefficient:

- $0.71 \leq \text{SC} \leq 1.00$: A strong cluster structure has been found.
- $0.51 \leq \text{SC} \leq 0.70$: A reasonable cluster structure has been found.
- $0.26 \leq \text{SC} \leq 0.50$: The cluster structure is weak, and may be artificial. Other methods should be applied to the data set as well.
- $\text{SC} \leq 0.25$: No substantial cluster structure has been found.

Figures 3.11a - 3.11c show silhouette plots for the data set shown in Figure 3.4a, for $K = 2$, 3 and 4, respectively. The text to the right in these panels give the number of objects in the clusters, as well as the average silhouette for each cluster. For $K = 2$, the average silhouettes of the two clusters are 0.78 and 0.42, respectively. Note that many of the samples in the largest cluster have a silhouette below 0.4. All of the objects have silhouette values of 0.6 or more for $K = 3$, and the average silhouette of the three clusters are 0.76, 0.75 and 0.79, respectively. For $K = 4$, some samples have negative silhouette values, and the average silhouette of one of the clusters is thus as low as 0.33. Figure 3.11d shows the average silhouette for the entire data set, plotted as a function of K ($2 \leq K \leq 10$). The curve is clearly maximized for $K = 3$, and this is hence the estimated number of clusters by the method. At this value we have $\text{SC} = \text{Sil}(3) = 0.77$, which corresponds to a strong cluster structure according to the guidelines above.



(a) Sample silhouettes, $K = 2$

(b) Sample silhouettes, $K = 3$

(c) Sample silhouettes, $K = 4$

(d) Average silhouette

Figure 3.11: The first three panels show sample silhouettes for the data set shown in Figure 3.4a for $K = 2$, 3 and 4, respectively. For $K = 3$, all of the objects have $s(i) > 0.6$. For $K = 2$, many of the silhouette values are below 0.4, while for $K = 4$ some of the silhouettes are even negative. The last panel shows the average silhouette over the entire data set, plotted as a function of K. The curve is clearly maximized at $K = 3$, which is thus reported as the optimal number of clusters.

The advantage of Silhouette is that it is easily understood and that it is computationally inexpensive. It may also be implemented for a variety of clustering methods. A (small) disadvantage is of course that the method does not evaluate whether it is appropriate to cluster the data at all ($K = 1$). However, it should be possible to modify the method to handle single-cluster cases as well.

3.6 Prediction strength (PS)

Tibshirani & Walther (2005) suggest using the *prediction strength* statistic to estimate the number of clusters in a data set. To calculate this measure, two independent data sets are used; one learning set and one test set. Ideally, these data sets are two independent sets drawn from the same population. If only one data set is available (which will usually be the case), r-

fold cross-validation or repeated random division of the data may be used to construct learning sets and test sets. Either way, the basic idea is to first cluster both the learning set and the test set into K clusters. Then the centroids of the clusters in the learning set are calculated, and used to classify the objects in the test set. The prediction strength statistic measures the extent to which the learning cluster centroids are able to predict the "true" clusters in the test data.

3.6.1 Definition

Formally, let the $n \times p$ data matrix X_L represent the learning data, and let X_T represent the $m \times p$ matrix of test data independent of X_L. Hence $X_L = [\mathbf{l}_1, \ldots, \mathbf{l}_n]^T$ and $X_T = [\mathbf{t}_1, \ldots, \mathbf{t}_m]^T$. Both X_L and X_T are clustered into K clusters using a chosen clustering method, resulting in K "learning clusters" and K "test clusters". The centroids of the K learning clusters are then used to predict cluster affiliation for the objects in the test set. One reasonable way of doing this is to calculate the distances between the objects in the test set and the learning cluster centroids, and classify the objects in the test set to the cluster with the nearest centroid. That is, let class(i) represent the cluster classification of object i in X_T when classified by the learning cluster centroids, and define

$$\text{class}(i) = \underset{1 \leq k \leq K}{\operatorname{argmin}}\, d(\mathbf{x}_i, \overline{\mathbf{x}}_k), \tag{3.7}$$

where $\overline{\mathbf{x}}_k$ represents the centroid vector of training cluster k.

Based on these cluster classifications, a matrix of *co-memberships* is constructed, where the ij'th element is equal to 1 if objects i and j are classified to the same cluster, and zero otherwise. That is, denoting this matrix by M, we have

$$M[i,j] = \begin{cases} 1 & \text{if class}(i) = \text{class}(j) \\ 0 & \text{otherwise} \end{cases}$$

To calculate the prediction strength statistic, let $C_{T1}, C_{T2}, \ldots, C_{TK}$ represent the group of test samples in test clusters $1, 2, \ldots, K$ for a particular choice of K. Further let $m_{T1}, m_{T2}, \ldots, m_{TK}$ denote the number of samples in the K test clusters. The "prediction strength" of the K learning cluster centroids is then defined as

$$\text{PS}(K) = \min_{1 \leq k \leq K} \frac{1}{m_{Tk}(m_{Tk}-1)} \sum_{i \neq j \in C_{Tk}} 1(M[i,j] = 1) \tag{3.8}$$

In words, for $k = 1, \ldots, K$ we calculate the proportion of test sample pairs found in test cluster C_{Tk} that also fall into the same cluster when classified by the learning cluster centroids. The prediction strength for a particular choice of K is taken to be the minimum of these proportions

over the K test clusters. Note that when $K = 1$, PS(K) will necessarily be equal to 1. Also, if a test cluster C_{Tk} only consists of one object, (3.8) is not defined.

3.6.2 Using prediction strength to find the number of clusters

By calculating PS(K) for different choices of K, we may estimate the optimal number of clusters. The intuition behind the idea is that when $K = K^*$ (where K^* is the true number of clusters), one may expect that the learning clusters and test clusters are similar, and hence that the learning cluster centroids will predict test clusters well. On the other hand, when $K > K^*$ the extra learning and test clusters will typically be less similar, and PS(K) will thus tend to be smaller. Hence PS(K) can be expected to drop when K gets larger than K^*. Based on this reasoning, Tibshirani & Walther (2005) suggest that the optimal number of clusters is estimated by the largest K such that PS(K) is greater than or equal to some preset threshold t. The authors have found that for well separated clusters a threshold in the range 0.8 - 0.9 works well.

In summary, the Prediction strength method for finding the optimal number of clusters is outlined as follows:

- For $K = 1, \ldots, K_{max}$ and $r = 1, \ldots, R$ repetitions:

  1. Randomly divide the data set into a learning set, X_L, of size n, and a test set, X_T, of size $m = N - n$. The value of n is determined by the learning set fraction (L_{frac}) and is calculated from $n = \text{round}(L_{\text{frac}} \times N)$.
  2. Cluster the learning and test data sets into K learning and test clusters.
  3. Using (3.7) (or some alternative classification scheme), classify the samples in X_T according to the learning cluster centroids.
  4. Determine the co-membership matrix, M, for the classification of the test samples, and calculate $PS_r(K)$ as defined by (3.8) for this r'th repetition.

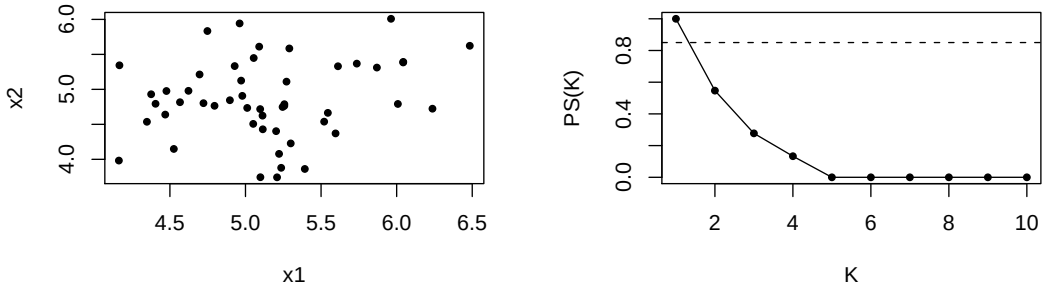
- Calculate the average prediction strength over the R repetitions:

$$\text{PS}(K) = \frac{1}{R} \sum_{r=1}^{R} \text{PS}_r(K)$$

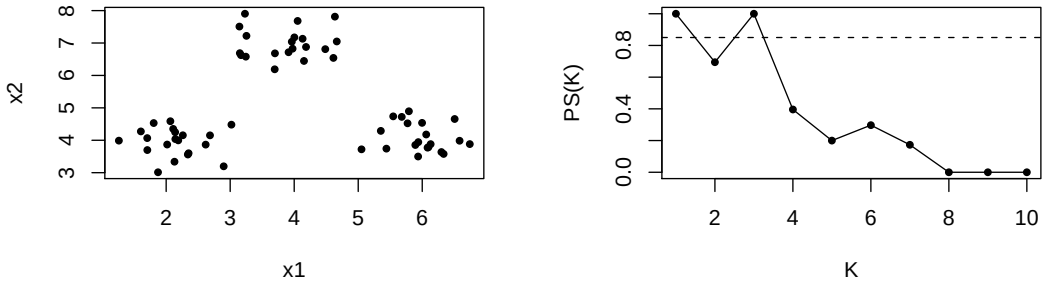
  If $PS_r(K)$ is undefined for any r, then this repetition is ignored in the computation of the average prediction strength. Finally, choose $\hat{K}$ as the largest K such that PS(K) $\geq t$, where t is some preset threshold (often set in the range $0.8 - 0.9$).

Figure 3.12 shows three examples where Prediction strength is used to determine the number of clusters. The three data sets (shown to the left) are made up of one, three and six clusters, respectively. Five random divisions of the data were used to construct learning and test sets,

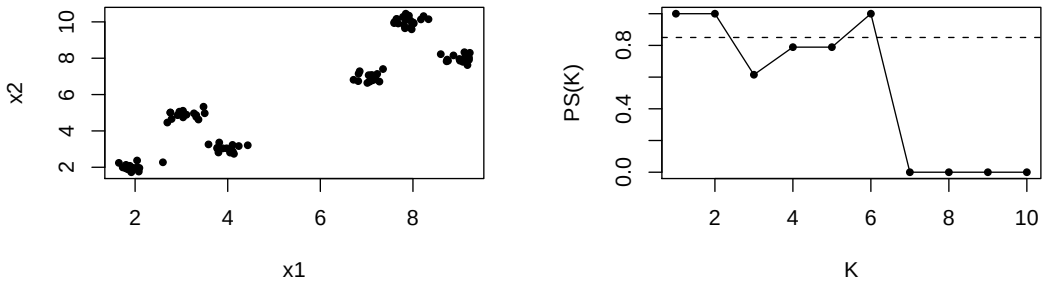
and the right panels of the figure show the average prediction strength over the five repetitions, for $K = 1, \ldots, 10$. Choosing the largest K such that $\mathrm{PS}(K)$ is above $t = 0.85$ (marked by the dotted line), results in choosing the true number of clusters in all three cases.



Data set (one cluster) Prediction strength curve



Data set (three clusters) Prediction strength curve



Data set (six clusters) Prediction strength curve

Figure 3.12: Using Prediction strength to estimate the number of clusters. The data sets are shown in the left panels, with one, three and six clusters respectively. The right panels show the prediction strength for $K = 1, \ldots, 10$, averaged over five random divisions of the data into learning and test sets. The dotted line shows the threshold at $t = 0.85$. Using the prediction strength criterion, we see that the largest K for which $\mathrm{PS}(K) \geq t$ is in fact $K = 1, 3, 6$ for the three respective data sets.

3.7 In-group proportion (IGP)

The *in-group proportion* statistic, proposed by Kapp & Tibshirani (2007), is a measure of cluster quality originally intended for cluster validation. That is, it measures the extent to which clusters are reproducible in new data sets. However, it also provides a method for estimating the number of clusters in data sets.

3.7.1 Definition

Similar to Prediction strength, the In-group proportion method assumes a $n \times p$ matrix of learning data, X_L, as well as a $m \times p$ matrix of test data, X_T, independent of X_L. The objects in the learning set are clustered into K groups, and the centroid of each cluster is calculated. Using (3.7), the objects in the test set are then classified to one of K groups based on their distance to the learning set centroids. As in the previous section, denote by class(i) the cluster classification for the i'th object in X_T

The in-group proportion pf a cluster is defined as the proportion of objects in X_T classified to the cluster k whose nearest neighbour was also classified to cluster k. Formally, denote by NN(i) the nearest neighbour of the i'th object in X_T and define

$$\mathrm{NN}(i) = \underset{j \neq i}{\operatorname{argmin}}\, d(\mathbf{x}_i, \mathbf{x}_j),$$

That is, NN(i) is the object in X_T that is the closest to the object i. Let k be the class label for all objects in X_T classified to cluster k. The in-group proportion of cluster k, IGP_k, is then defined as

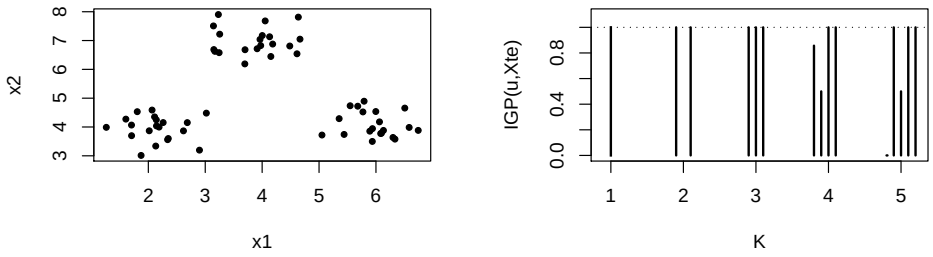
$$\mathrm{IGP}_k = \frac{\#\{i|\mathrm{class}(i) = \mathrm{class}(\mathrm{NN}(i)) = k\}}{\#\{i|\mathrm{class}(i) = k\}}. \tag{3.9}$$

A cluster of high quality (a well defined cluster) will be expected to have a large in-group proportion, that is, an IGP close to 1. Note that IGP_k is not defined if no test objects are classified to a cluster k.

Figure 3.13 shows some in-group proportions for a data set with 3 clusters. The left panel shows the data set, while the right panel shows the calculated in-group proportion of each group for $K = 1, \ldots, 5$. For $K = 1, 2, 3$ the IGP is equal to 1 for all groups. For $K = 4$, one of the groups has an IGP far below 1, and for $K = 5$ two groups have IGPs far below 1 (one of them is in fact zero). It seems that when K exceeds K^* (where $K^* = 3$ in this example), at least one of the groups' IGP drops significantly from 1.

3.7.2 Using IGP to estimate the number of clusters

As seen in the example above, we may calculate IGP_k (where $k = 1, \ldots, K$) for different values of K. Hence, given a suitable criterion, the in-group proportion may be used to find the



Data set (three clusters) IGPs for K clusters

Figure 3.13: The left panel shows a data set with three clusters. The right panel shows the in-group proportion of each of the K clusters (IGP_k), for different choices of K ($K = 1, \ldots, 5$).

number of clusters in data sets. Kapp (2007) suggests a procedure for determining the number of clusters using the in-group proportion. This criterion seems heavily influenced by the ideas presented in the Prediction strength method.

First, the overall in-group proportion for a certain choice of K is taken to be the minimum IGP_k evaluated over the K clusters. That is, define

$$\text{IGP}(K) = \min_{1 \leq k \leq K} \text{IGP}_k \tag{3.10}$$

If one of the clusters' IGP is undefined, then $\text{IGP}(K)$ is also left undefined. As in Prediction strength, Kapp (2007) suggest that we estimate the number of clusters by the largest K for which $\text{IGP}(K)$ is greater than or equal to some pre-specified threshold, t. The value of t is determined by the user's demands toward the quality of each cluster, but Kapp (2007) has found that a threshold of 0.95 often works well.

In summary, the In-group proportion method for estimating the number of clusters contains the following steps:

- For $K = 1, \ldots, K_{max}$ and $r = 1, \ldots, R$ repetitions:

  1. Randomly divide the data set into a learning set, X_L, of size n, and a test set, X_T, of size $m = N - n$ (where n is determined by the learning set fraction, L_{frac}).
  2. Cluster the learning set into K clusters and calculate the K learning cluster centroids $(\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_K)$.
  3. Classify the objects in the test set to one of the K clusters using (3.7):

  $$\text{class}(i) = \underset{1 \leq k \leq K}{\text{argmin}}\, d(\mathbf{x}_i, \overline{\mathbf{x}}_k).$$

  4. Using (3.9), calculate the in-group proportion of all clusters, IGP_k ($k = 1, \ldots, K$).

If no test objects are classified to a cluster, then the IGP of that cluster is left undefined.

5. The overall IGP for a given value of K is taken to be the minimum IGP over the K clusters:

$$\text{IGP}_r(K) = \min_{1 \le k \le K} \text{IGP}_k,$$

for this r'th repetition. If IGP_k is undefined for any of the K clusters, then $\text{IGP}_r(K)$ is also left undefined.

- Calculate the average in-group proportion over the R repetitions:

$$\text{IGP}(K) = \frac{1}{R} \sum_{r=1}^{R} \text{IGP}_r(K) \tag{3.11}$$

Any $\text{IGP}_r(K)$ that is not defined is ignored in this calculation. Finally, let $\hat{K}$ be the largest K for which $\text{IGP}(K) \geq t$, where t is some pre-specified threshold.

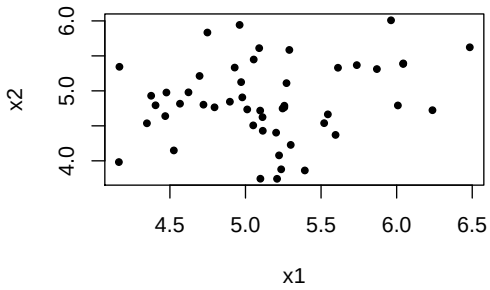
The left panels of Figure 3.14 show the same data sets as in Figure 3.12, while the right panels show the calculated $\text{IGP}(K)$ for $K = 1, \ldots, 10$, averaged over five random divisions of the data into learning and test sets. The results are quite similar to the results of the Prediction strength. Using the criterion described above, $\hat{K}$ will be 1, 3 and 6 for the three data sets respectively.

Adjustment of the definition of IGP(K)

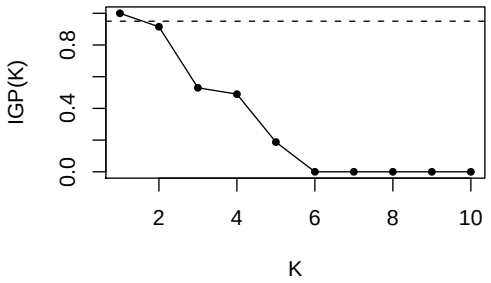
Kapp (2007) points out that when one of the clusters in the data set is small compared to the other clusters, the learning data is often clustered inaccurately (especially when using K-means clustering). In effect, the learning cluster centroids may not be suitable for the classification of the test objects. Thus, for data sets with clusters of very different size, (3.11) tends to underestimate the number of clusters. A possible remedy for this problem may be to use the entire data set (instead of dividing it into learning data and test data) to calculate the centroids, and then classify each object according to these centroids. The minimum of the resulting IGP's for a given value of K is denoted f_K. This procedure, on the other hand, tends to overestimate the number of clusters when the clusters are evenly-sized because the centroids predicts cluster affiliation too well (even for unnatural clusters). Since we usually do not know the true structure of the data, it is not clear which of the two procedures we should use.

Kapp (2007) therefore suggests a measure in between the two, where different weight is given to the two terms. That is, re-define (3.11) such that

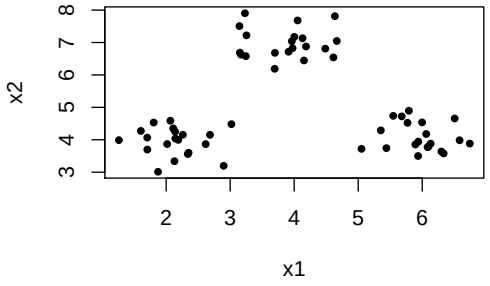
$$\text{IGP}(K) = \left(\frac{1}{R} \sum_{r=1}^{R} \text{IGP}_r(K) \right) + f_K + e_K, \tag{3.12}$$



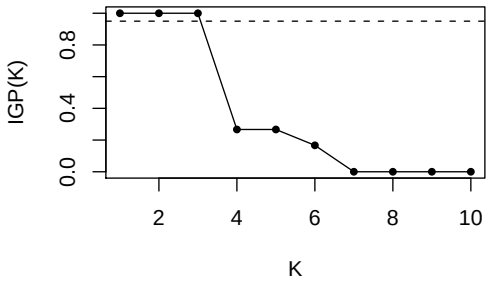
Data set (one cluster)



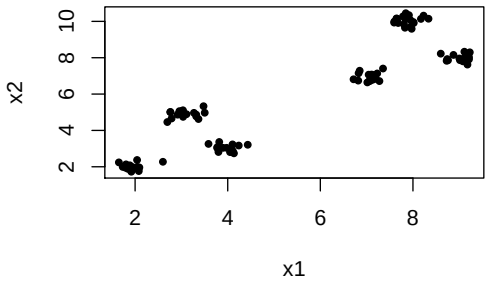
In-group proportion curve



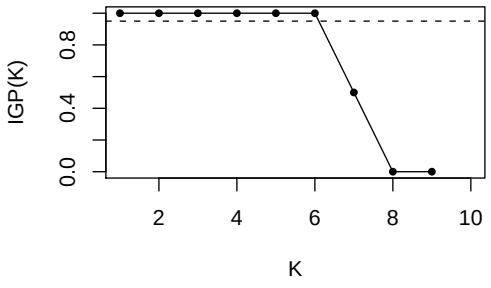
Data set (three clusters)



In-group proportion curve



Data set (six clusters)



In-group proportion curve

Figure 3.14: Using the in-group proportion to determine the optimal number of clusters. The data sets are shown on the left side, with one, three and six clusters respectively. The right side shows the curve of IGP(K) for each data set. The dotted line shows a threshold at $t = 0.95$. In these examples, the largest K for which IGP(K) is greater than $t = 0.95$, is in fact the true number of clusters in the data sets (1, 3 and 6 respectively).

where e_K is a term that should cancel out the contribution from that of the preceding terms which is harmful. One way to define e_K is to add another step to the algorithm described above, where one classifies all the objects in the entire data set using the learning set centroids and calculates the IGP of each cluster. Define e_{rK} as the minimum of these IGPs and let e_K be the average over R repetitions. That is, define $e_K = \frac{1}{R}\sum_{r=1}^{R} e_{rK}$. Using this term in 3.12 will tend to cancel out the preceding biased term, and Kapp (2007) has found that this definition of IGP(K) improves the method's performance.

Chapter 4

Microarray data

4.1 Microarray technology

Microarray technology make it possible to simultaneously study the expression of thousands of genes, and is a commonly used method in genomics research. Several types of microarrays exist, but only cDNA microarrays are (briefly) described here because both of the real data sets studied in this thesis come from this type of arrays. cDNA microarrays are microscope slides that have a large number of DNA samples (probes) printed on them in a regular pattern, hence forming an array. Gene expression can be measured by letting cDNA (complementary DNA) from a sample hybridize with the probes on the microarray slide. In a two-colour cDNA microarray experiment, mRNA is extracted from one experimental sample and one reference sample, and then reverse transcribed into cDNA. The cDNA from the experimental sample is then labeled with one dye (Cy5, red fluorescence), while the cDNA from the reference sample is labeled with a different dye (Cy3, green fluorescence). The two labeled cDNA sets are then mixed in equal quantity, and allowed to hybridize with the probes on the microarray slide. The gene expression pattern of both samples are then measured by a laser scanner that records the red and green fluorescence signals for each probe on the microarray. The logarithm of the ratio between these signals (log(Cy5/Cy3)) assess the relative expression of a gene in the two samples. Hence the data retrieved from the entire microarray will indicate the gene expression pattern of the experimental sample relative to the reference sample. Figure 4.1 shows the general outline of a cDNA microarray experiment. (See for example Xiong (2006) for more details about microarrays.)

In cancer studies, tissue samples are taken from several patients, and each sample is compared to a reference sample on a microarray slide. The results from the different microarray experiments can then be compared to identify co-expressed genes that show the same expression profiles across the samples, and/or samples that have similar gene expression patterns. Cluster analysis is often used to analyze the data from such microarray studies.

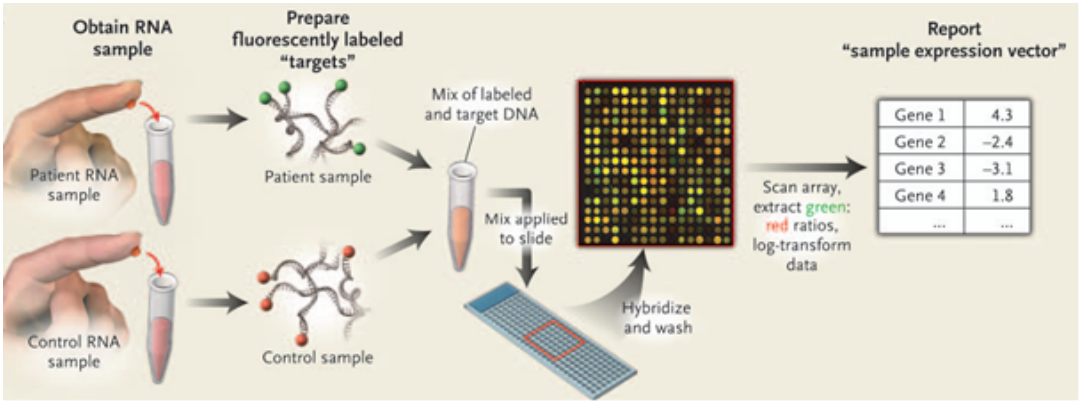


Figure 4.1: A basic outline of the steps in a cDNA microarray experiment (Quackenbush, 2006).

4.2 Identification of breast tumour subtypes

4.2.1 Background

Several studies have indicated that breast cancer tumors may be divided into distinct subtypes based on differences in gene expression patterns. Perou et al. (2000) found through hierarchical clustering of 65 breast tumor and normal breast tissue samples that the samples fell into two major categories: an "ER+ /Luminal-like" group (estrogen receptor positive) and a "ER−" group (estrogen receptor negative). A characteristic feature of the samples in the ER+ /Luminal-like group was that many of the genes expressed by breast luminal cells were relatively overexpressed. The ER− group could further be divided into three subgroups referred to as "Basal-like", "ERBB2+" and "Normal breast-like". The samples in the Basal-like group were characterized by high expression of many of the genes that are characteristic of breast basal epithelial cells, while samples in the ERBB2+ group were characterized by overexpression of the Erb-B2 oncogene. The Normal breast-like samples resembled the gene expression pattern of normal breast tissue samples (high expression of basal epithelial cell and adipose cell genes, and low expression of luminal epithelial cell genes).

Sørlie et al. (2001) hierarchically clustered 85 breast tissue samples (tumors and normal), and found further support for the subtypes of breast cancer found by Perou et al. (2000). In addition to the repeated observation of the basal, ERBB2+ and normal breast-like subtypes, they also found that the previously characterized ER+/luminal group could be further subdivided into two, possibly three subtypes: "Luminal A", "Luminal B" and "Luminal C". The samples in these subtypes all showed high expression of luminal epithelial genes, but while the Luminal A group showed a high expression of luminal epithelial genes containing ER, the Luminal B and Luminal C subtypes showed low or moderate expression of these genes. The Luminal C subtype could be further distinguished from the other luminal subtypes by high expression of a novel set of genes with unknown function.

Sørlie et al. (2003) added 38 more tumor samples to the data set used in Sørlie et al. (2001) such that their data set consisted of gene expression values for a total of 122 breast tissue samples (referred to as the "Norway/Stanford cohort"). Of the 122 tissue samples, 115 were from malign breast cancer tumors (carcinomas), 3 were from benign tumors (fibroadenomas) and 4 were from normal breast tissues. Among all the genes measured in the cDNA microarray experiments, a set of *intrinsic genes* were selected. This set included the genes that showed the most variation among the different tumors, while varying the least among successive samples taken from the same patient's tumor. The genes in this subset should thus be expected to represent intrinsic properties of the tumors as opposed to differences in the sampling. The final list consisted of 534 genes represented by 552 probes (i.e. some of the genes were represented by more than one probe). The data set consisting of the intrinsic gene expression values for the 122 samples described above will in the following be referred to as "the Sørlie data set".

Using hierarchical clustering and the intrinsic genes, Sørlie et al. (2003) found further evidence of five subtypes of breast cancer discovered earlier. Figure 4.2 shows the published dendrogram for this data set. Different colours represent the five subtypes: *Luminal A* (dark blue), *Luminal B* (light blue), *ERBB2+* (pink), *Basal* (red) and *Normal breast-like* (green). Samples with low correlation to any subtype are depicted as grey. That is, the sub-clusters shown include the core members of the subtype, which are the samples that are characterized by a high correlation with the cluster centroid. Note that separate cut-offs were applied for each cluster to signify what is a high correlation. The five subtypes identified by the cluster analysis are characterized by distinct variation in their gene expression. Further, the authors also found differences in the clinical outcomes associated with the different subtypes (worst for Basal and ERBB2+, and best for Luminal A). Hence, the subtypes have been found to represent both biological and clinical diversity among breast tumors.

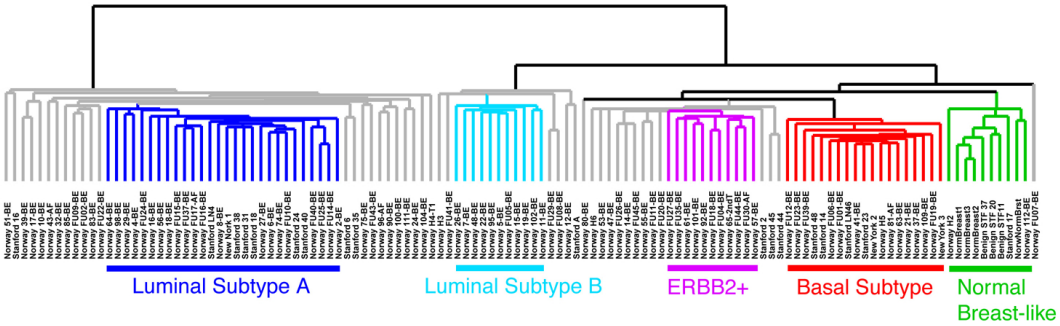


Figure 4.2: Hierarchical clustering of the Sørlie data set as published in Sørlie et al. (2003). Genes were centered, and centroid linkage with Pearson's centered correlation was used. Five breast cancer subtypes, represented by different colours, were identified.

4.2.2 Hierarchical clustering of the Sørlie data set

The Sørlie data set was downloaded from the SGBCC portal (`http://genome-www.stanford.edu/breast_cancer/`). Prior to any analysis, missing gene expression values in the data set were imputed using an average of the values of their 10 nearest neighbours (the function `impute.knn` in `R` was used). The genes were then centered, and the samples were centered and standardized. Hierarchical clustering with squared Euclidean distances and average linkage resulted in the dendrogram shown in Figure 4.3. This dendrogram differs quite a lot from the dendrogram presented in Sørlie et al. (2003) (Figure 4.2). There are a number of possible reasons for these differences:

- Sørlie et al. (2003) used Eisen's clustering algorithm with *centroid* linkage (Eisen et al., 1998), implemented in the software `Cluster` (Eisen, 1998), whereas I used the function `hclust` with *average* linkage in the software `R`. It is possible to perform centroid linkage clustering in `R` as well, but for this data set the resulting dendrogram would look very "messy". This is due to the problem of so called *inversions*, which occur when the distance from the union of two clusters, G and H, to a third cluster, J, is less than the distance from either G or H to J. `Cluster` deals with the problem of inversions by adjusting the dendrogram heights such that they increase monotonically, and hence the dendrogram in Figure 4.2 does not exhibit any inversions. The use of different linkage functions is probably the major reason that the dendrograms look so different. (See Speed (2003, chap. 4) for a brief discussion of differences between Eisen's clustering and average linkage clustering.)

- The imputation of missing gene expression values is done via a automatic procedure in `Cluster`. I used the function `impute.knn` in `R` to impute the missing values by an average of the values of their 10 nearest neighbours.

- Sørlie et al. (2003) used one minus Pearson's centered correlation as distance function, whereas I used squared Euclidean distance. However, since the samples were centered and standardized before the clustering, squared Euclidean distance and one minus the correlation distance should produce equivalent results (cf. Section 2.2). In fact, I chose to use squared Euclidean distance because of this property (and not one minus the correlation distance because my implementation of the methods for finding the number of clusters use Euclidean distance).

- In hierarchical clustering, two identical clustering results may lead to two different dendrogram representations. Whenever a merge is made between two clusters, a decision has to be made regarding which of the clusters should be placed to the left in the tree, and which should be placed to the right. This makes it important to realize that only

the order in which clusters are merged in the dendrogram, and *not* the proximity of their placement, tell us about their relative similarity.

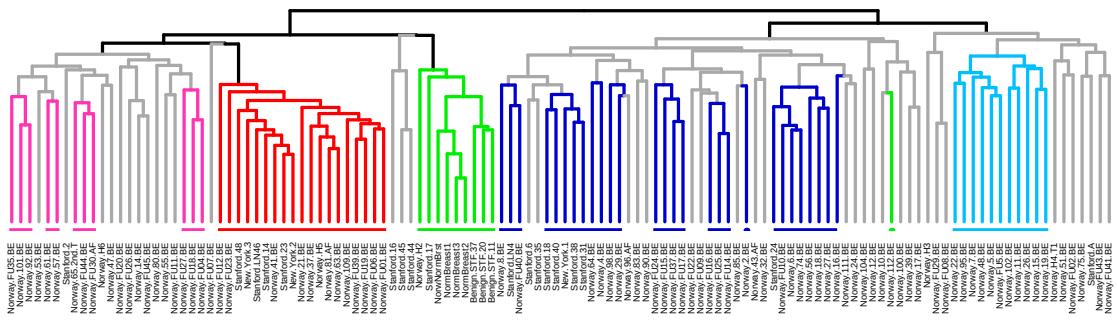


Figure 4.3: Hierarchical clustering with average linkage and squared Euclidean distance of the Sørlie data set (produced in R). The colour of the leaves show which sub-cluster the sample was associated with in the original dendrogram (Figure 4.2).

Despite two very different looking dendrograms, the clustering results that they are based on are in fact not that different. A comparison of which samples have been clustered together in the two dendrograms showed that there is in fact a great amount of agreement between the two. The colour of the leaves in Figure 4.3 represents the cluster affiliation of that sample in Figure 4.2. This shows that most of the samples that were clustered together in Figure 4.2 are also clustered together (or at least close to each other) in Figure 4.3. Note in particular that the Basal subtype (red) is completely preserved in the latter dendrogram, and that only one sample (green) is completely "misplaced". Overall it seems that five subgroups may be found in both dendrograms. (Though not shown, a comparison of the location of the samples marked as grey in Figure 4.2 also showed great correspondence between the two clustering results.)

When identifying the 5 subtypes, Sørlie et al. (2003) based their conclusions on their visual inspection of the dendrogram, as well as their biological interpretation of such groupings. The number of clusters in the data set was thus not found by objective and automatic methods. Hence, it is interesting to see what results the methods described in Chapter 3 give for the Sørlie data set, and if any of the methods are able to identify the five subgroups found by Sørlie et al. (2003).

4.2.3 Estimation of the number of clusters in the Sørlie data

The seven methods described in Chapter 3 were applied to the clustering of the Sørlie data set shown in Figure 4.3, with the parameters listed in Table 4.1. Several trials were made, because all of the methods, except Silhouette, depend on the random generation of reference data or the random splitting of the data set into learning and test sets. Hence, even though the data set is identical over the 50 trials, the Gap methods, Prediction strength and In-group

proportion may report different results in the various trials. We do expect, however, that the methods give the same result over most of the trials. Table 4.2 lists the number of times that each method estimates a given number of clusters in the Sørlie data set, over a total of 50 trials.

Table 4.1: Parameter values for the various methods when applied on the Sørlie data set and the Micma data set.

Gap, Reference Gap and Recursive Gap	
Maximum number of clusters	$K_{max} = 10$
Number of reference sets	$B = 50$
ERA	
Number of reference sets	$B = 50$
Minimum number of samples in each cluster	$n_{min} = 7$
Number of thresholds	$H = (N - 1) * 0.2$
Number of repetitions	$R = 15$
Silhouette	
Maximum number of clusters	$K_{max} = 10$
In-group proportion	
Maximum number of clusters	$K_{max} = 10$
Number of repetitions	$R = 5$
Learning set fraction	$L_{\text{frac}} = \frac{1}{2}$
Threshold	$t = 0.95$
Prediction strength	
Maximum number of clusters	$K_{max} = 10$
Number of repetitions	$R = 5$
Learning set fraction	$L_{\text{frac}} = \frac{1}{2}$
Threshold	$t = 0.85$

As the table shows, the results are quite varying. The methods that give the most consistent results compared to the findings of Sørlie et al. (2003), are Recursive Gap and ERA. Recursive Gap finds five clusters in 47 out of 50 trials, while ERA finds five clusters in all of the trials. In the trials where Recursive Gap estimates only three clusters it is because the Gap algorithm sometimes does not find any sub-clusters in the left main branch of the dendrogram in Figure 4.3. This does not affect ERA, since an tentative split is still made in the left branch. The algorithm then finds two significant sub-clusters in the left tentative cluster, and thus redefines the right tentative cluster as significant. ERA then reports that there are three sub-clusters in the left main branch. This makes the enhanced recursive version somewhat more accurate in finding the optimal number of clusters in this data set (if we assume that five clusters is in fact the optimal result).

Three of the methods, namely Gap, Reference Gap, and Silhouette, consistently report that

there are two clusters in the data set. This is not such a bad estimate, because the greatest distinction between the samples in the Sørlie data set (and other breast tumour data sets) was found between samples that have high expression of luminal epithelial genes and samples that show low expression of these genes (Sørlie et al., 2003). The difference in expression pattern for these genes is particularly great between Luminal A subtype samples and Basal subtype samples. The fact that Gap, Reference Gap and Silhouette are able to detect two clusters of Luminal-like and Basal-like subtypes is positive.

In-group proportion and Prediction strength are the least effective methods for finding the number of clusters in the Sørlie data set. As seen in Table 4.2, these methods report one cluster in 49 and 47 of the 50 trials, respectively, and are thus to a great extent unable to find any cluster structure in the data at all. This is probably because the thresholds of 0.95 (IGP) and 0.85 are too high when the methods are applied on real data sets. In fact, Kapp (2007) points out that a threshold of 0.95 is perhaps not suitable for real data sets where clusters are not very isolated nor very cohesive. However, with no suggestions of more suitable thresholds it is difficult to improve upon these results.

Table 4.2: The table shows the number of times (out of 50 trials) that the various methods estimate a given number of clusters in the Sørlie data set.

Method **K**	1	2	3	4	5
ERA	0	0	0	0	50
Recursive Gap	0	0	3	0	47
Gap	0	50	0	0	0
Reference Gap	0	50	0	0	0
Silhouette	0	50	0	0	0
In-group proportion	49	0	0	0	1
Prediction strength	47	3	0	0	0

The dendrogram in Figure 4.4 shows the five clusters (represented by different colours) found in the majority of the 50 trials by Recursive Gap and ERA. The colours of the labels represent the cluster affiliation of each sample according to the results reported in Sørlie et al. (2003), and we see that the clusters correspond well to the subtypes identified in the article. Note that the two clusters found by Gap, Reference Gap and Silhouette correspond to the union of the light blue and the dark blue clusters, and the union of the pink, red and green clusters. Also note that as long as a horizontal cut is applied in these methods they will be unable to discover the five clusters found by the recursive methods, as a horizontal cut producing 5 clusters would divide the light blue cluster in two, while keeping the pink and red clusters merged.

Figure 4.5a - 4.5e show the statistic used in the various methods plotted as a function of the number of clusters K, when applied on the Sørlie data set (one of the 50 trials). No such "result curves" are given for Recursive Gap and ERA, because several plot would be required

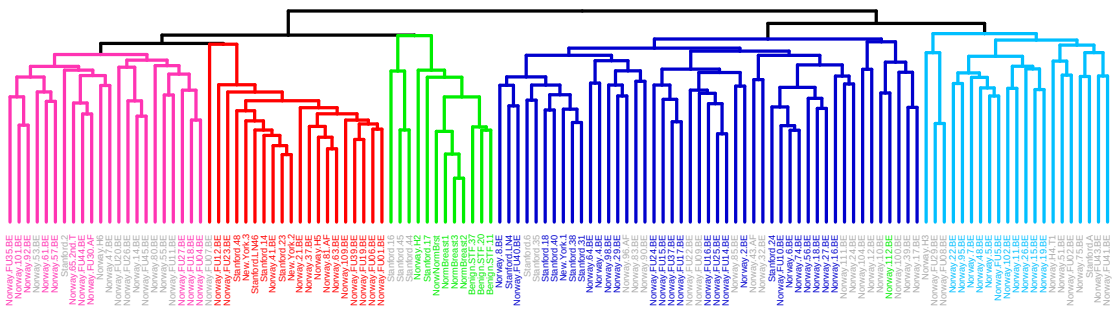


Figure 4.4: Clusters found in the Sørlie data set by ERA and Recursive Gap, represented by different colours. The colour of each label shows which sub-cluster the sample was associated with in Sørlie et al. (2003) (cf. Figure 4.2).

due to their recursive nature.

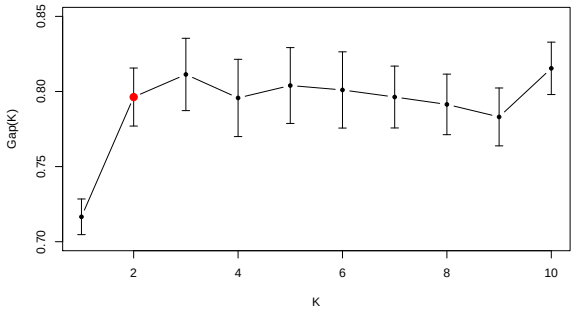
4.3 The Micma breast tumour data

4.3.1 Background

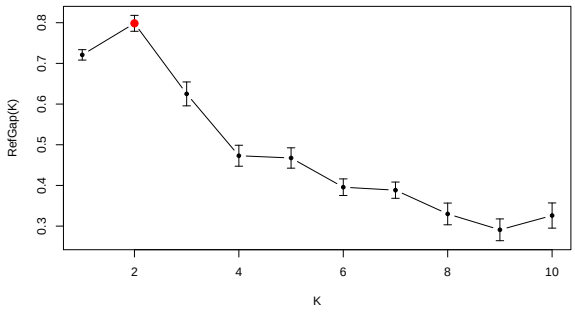
The Micma data set consists of gene expression values measured for 115 breast tumour patients using cDNA microarrays (see Naume et al., 2007). Of these, 114 of the cancer tissue samples had known "disseminated tumor cell" (DTC) status (disseminated means "spread over a large area of a body, tissue, or organ") and one patient was diagnosed with ductal carcinoma in situ (non-invasive cancer). Since the data set consists of breast tumour samples, one may expect that the five subtypes described earlier are represented in this data set as well. In fact, when classifying the samples to subtypes based one their correlation with the subgroup centroids found by Sørlie et al. (2003), Naume et al. (2007) identified all five breast tumor subtypes. In this classification only expression values for Sørlie's intrinsic gene set were used.

4.3.2 Hierarchical clustering

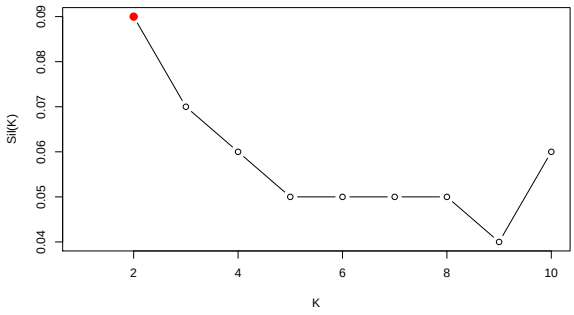
The Micma data originally consisted of gene expression values of approximately 40000 gene probes measured for 115 patients. In the hierarchical clustering, I wanted to use only the genes found in Sørlie's intrinsic gene list. The genes in the intrinsic gene list are represented by a gene symbol, a UniGeneID and a gene name. I first reduced the intrinsic gene list to a set of *unique* gene symbols (the intrinsic gene list originally consists of 552 probes and 534 unique genes, hence some genes were represented by more than one probe). The next step was to extract the intrinsic genes in the Micma data set. A problem in doing this was that gene symbols are not universal, so the gene symbols listed for the Micma data did not necessarily match the gene symbols in the intrinsic gene list even if the genes they represented were the same. To deal



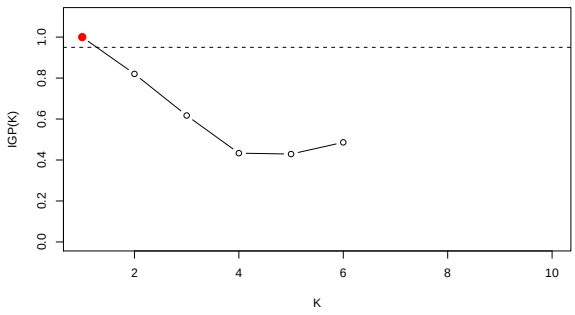
(a) Gap ($\hat{K} = 2$)



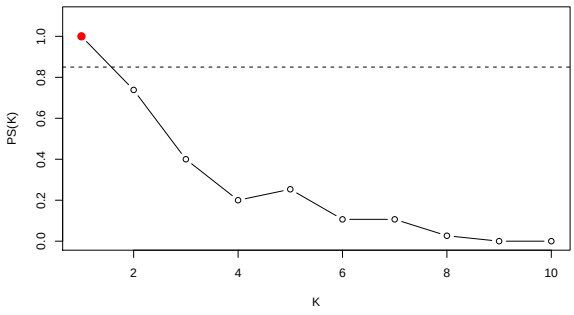
(b) Reference Gap ($\hat{K} = 2$)



(c) Silhouette ($\hat{K} = 2$)



(d) In-group proportion ($\hat{K} = 1$)



(e) Prediction strength ($\hat{K} = 1$)

Figure 4.5: The graphs show the statistic used by the respective methods plotted as a function of K when applied on the Sørlie data set (one of the 50 trials). The dotted lines in 4.5d and 4.5e show the threshold values at $t = 0.95$ and $t = 0.85$ for In-group proportion and Prediction strength, respectively.

with this, I downloaded a list of known gene symbol aliases (`http://www.genenames.org`). I then selected the gene symbols in the Micma data set that matched a gene symbol, or one of it's known aliases, in the intrinsic list. A few of the genes represented in the Micma data set did not have a gene symbol listed, and these genes were selected only if they matched a UniGeneID in the intrinsic gene list. Finally, because several probes in the Micma data set represented the same gene, the median gene expression value over the probes that represented the same gene was used.

The final reduced Micma data set consisted of gene expression values for 463 intrinsic genes measured for 115 breast cancer patients. Before clustering the samples, the genes were mean-centered and the samples were normalized (such that squared Euclidean distance and one minus Pearson's centered correlation distance would produce equivalent results). Figure 4.6 shows the dendrogram resulting from hierarchical clustering with squared Euclidean distance and average linkage of the reduced Micma data set.

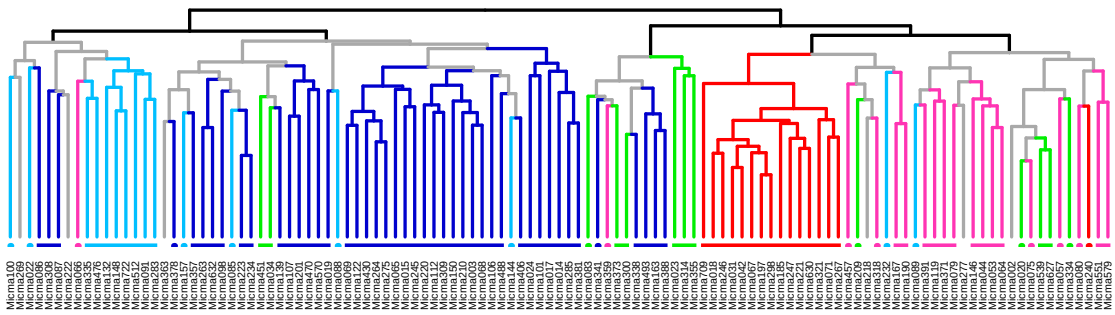


Figure 4.6: Hierarchical clustering with average linkage and squared Euclidean distance of the reduced Micma data set (produced in R). The colour of each leaf represents the subtype that the sample was classified to by Naume et al. (2007).

As mentioned, Naume et al. (2007) classified the samples to the subtype whose expression centroid it correlated the most with (the expression centroids of the different subtypes are described in supplemental table 3 in Sørlie et al. (2003)). The colour of the leaves in the dendrogram in Figure 4.6 represents the subtype that the given sample was classified to. Though the grouping of the subtypes is not as evident in this data set as it was in the Sørlie data set (cf. Figure 4.3), many of the samples that were classified to the same subtype tend to cluster together. In particular, the samples classified to the Basal subtype (red) form a very cohesive cluster (with the exception of just one sample). Many of the samples classified to the Luminal A group (dark blue), the Luminal B group (light blue) and the ERBB2+ group (pink) are also clustered quite close to each other, whereas the samples classified to the Normal breast-like group (green) are a bit more spread. Hence, in accordance with Sørlie et al. (2003), there is some evidence that five groups of breast tumours are found in this data set as well.

4.3.3 Estimation of the number of clusters

The methods described earlier were applied on the Micma data set to find the optimal number of clusters. The parameters for each method are listed in Table 4.1. Table 4.3 lists the number of times each method estimates a given number of clusters, over 50 trials. Similarly to the results for the Sørlie data set, the methods are not in agreement of what the optimal number of clusters is.

Table 4.3: The table shows the number of times (out of 50) that the various methods estimate a given number of clusters in the Micma data set.

Method **K**	1	2	3	4	5	6	7
ERA	0	0	0	0	1	49	0
Recursive Gap	0	0	0	2	1	44	3
Gap	0	49	1	0	0	0	0
Reference Gap	0	50	0	0	0	0	0
Silhouette	0	50	0	0	0	0	0
In-group proportion	50	0	0	0	0	0	0
Prediction strength	49	1	0	0	0	0	0

Recursive Gap and ERA give the results that are the most in correspondence with previous findings for breast tumour data. However, instead of five subgroups, these methods report $K = 6$ as the most frequent result. Since this is a completely new data set that is not related to the Sørlie data set, it is not clear what the optimal number of clusters is. It may be unrealistic to expect to see the same clear grouping into five clusters as found in the Sørlie data. There is also yet much to learn about breast cancer subtypes and there may well be more than five subtypes. In fact, the results of Sørlie et al. (2001) indicated that the Luminal-like group could be divided into three groups, namely Luminal A, Luminal B and Luminal C, so there are some indications that there are six subtypes of breast cancer. In the end, it is positive that Recursive Gap and ERA are able to detect more than two clusters, and six clusters is probably closer to a true answer than the results of the other methods. The dendrogram in Figure 4.7 shows the six clusters most frequently found by the Recursive Gap and ERA, represented by different colours. The colour of the labels indicate the subtype that the samples were classified to by Naume et al. (2007), and the colour coding of the clusters correspond to the subtype that the majority of samples in the cluster was classified to. Note two of the clusters have a majority of Luminal A samples (dark blue labels), these clusters are coloured purple and dark blue, respectively.

Gap, Reference Gap and Silhouette all agree that there are two clusters. As for the Sørlie data set, this is not such a bad estimate because a division into a Luminal-like group and a Basal-like group represents the clearest distinction between the tumors. The two clusters found by Gap, Reference Gap and Silhouette correspond to the union of the light blue, purple and dark blue clusters, and the union of the pink, red and green clusters, respectively, in Figure 4.7.

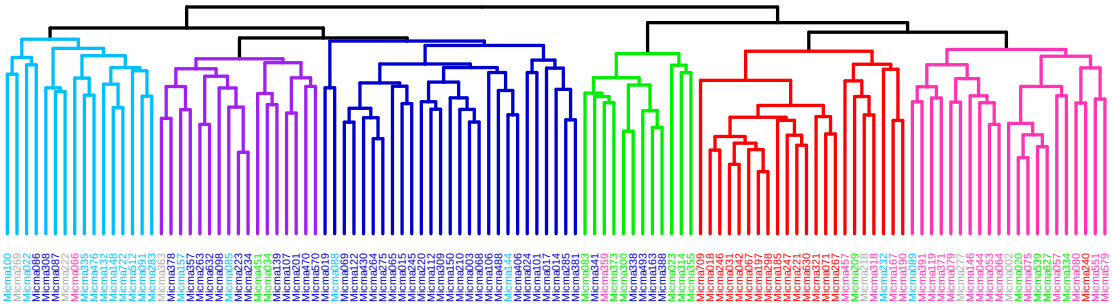
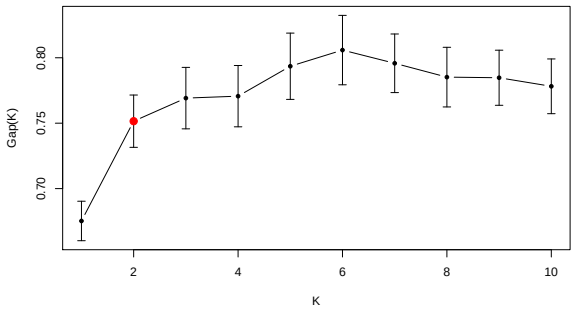


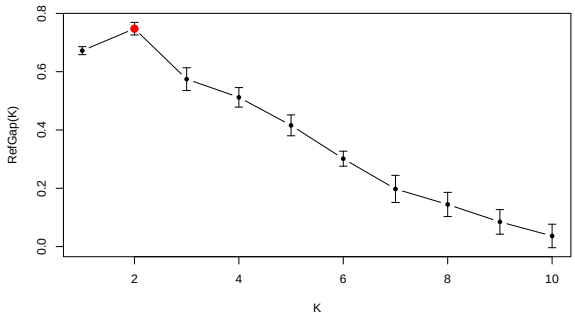
Figure 4.7: Clusters found in the Micma data set by ERA and Recursive Gap. The clusters are represented by different colours, and the colour of each label shows which subtype the sample was classified to in Naume et al. (2007) (cf. Figure 4.6)

In-group proportion and Prediction strength both estimate that there is only one cluster, i.e. no cluster structure is found in the data set. These are the weakest results, since several studies have indicated that breast cancers do fall into subgroups. However, as mentioned previously, the threshold levels applied in these methods are probably too high to be able to discover clusters that are not very well-separated.

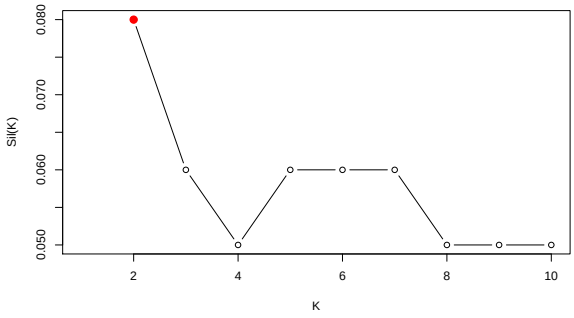
Figure 4.8a - 4.8e show the statistic used in the respective methods as functions on the number of clusters K, when applied on the Micma data set (for one of the 50 trials).



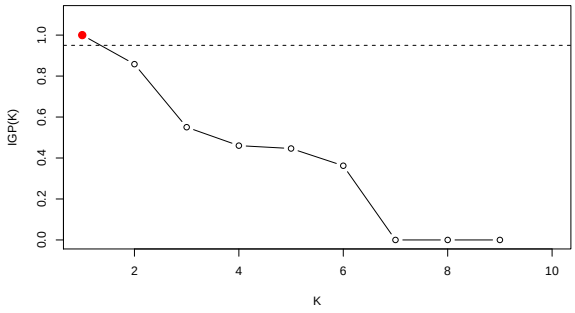
(a) Gap ($\hat{K} = 2$)



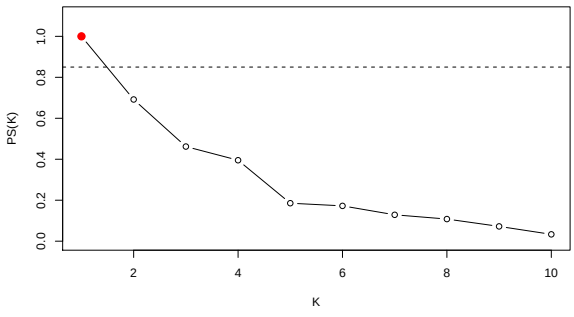
(b) Reference Gap ($\hat{K} = 2$)



(c) Silhouette ($\hat{K} = 2$)



(d) In-group proportion ($\hat{K} = 1$)



(e) Prediction strength ($\hat{K} = 1$)

Figure 4.8: The plots show the statistic used in the respective methods plotted as a functions of K, when applied on the Micma data set (one of the 50 trials). The dotted lines in 4.8d and 4.8e show the threshold values at $t = 0.95$ and $t = 0.85$ for In-group proportion and Prediction strength, respectively.

Chapter 5

Simulations

Simulations can be used to study the effectiveness and virtues, as well as the shortcomings, of the proposed methods in a controlled setting since the true answer is known. By varying relevant factors and observing the effect on a method's performance, we can get an indication of the conditions under which a method is successful or fails.

Four simulation scenarios are constructed in this chapter to test the effect of some factors on each method's performance. These factors include the relative separation of the cluster means, the distribution (heavy-tailed or not), the variance of the clusters, the number of features (the dimension) and the presence/absence of sub-clusters. Each scenario will be described in detail below, but first the general setup and some common notation is introduced.

5.1 General setup and notation

In each simulation scenario, the data sets are designed to contain a certain number of clusters, K^*, which are generated by simulating samples from K^* different multivariate normal distributions. If $\mathbf{x}_{ki}$ represents a random sample i in cluster k we thus have

$$\mathbf{x}_{ki} \sim N_p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where p is the number of features, and $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ represent the mean vector and the covariance matrix of cluster k, respectively. Each cluster k is made up of n_k samples generated in this fashion, and the total number of samples in the data set will thus be $N = \sum_{k=1}^{K^*} n_k$.

The covariance matrix, $\boldsymbol{\Sigma}_k$, is a square matrix whose diagonal elements give the variance of each feature and whose off-diagonal elements give the covariance between each pair of features. To keep the scenarios simple, no covariances are introduced between any of the features. Hence in all of the simulations the off-diagonal elements of $\boldsymbol{\Sigma}_k$ are zero. The variance of a feature j in cluster k is denoted σ_{kj}^2, and the vector $\boldsymbol{\sigma}_k^2 = \left[\sigma_{k1}^2, \ldots, \sigma_{kp}^2\right]^T$ thus defines the p diagonal elements of $\boldsymbol{\Sigma}_k$. In most of the scenarios the variances are identical for the p features and for

the K^* clusters. In these cases we write $\sigma_{kj}^2 = \sigma^2$, and hence $\mathbf{\Sigma}_k = \sigma^2 \mathbf{I}_p$ (since the covariances are zero).

The cluster means are in all of the scenarios defined such that the distances between them are given and fixed. Except for the last scenario (Scenario D), the mean vectors are constructed to be orthogonal, and can then, for the purpose of defining the distances between pairs of mean vectors, with no loss of generality be assumed to differ from each other in only a few of the features. In the later representations of the cluster means, the notation $\mathbf{0}_l$ will be used to represent a vector consisting of zeros that is of length l. Hence, a cluster mean vector written as

$$\boldsymbol{\mu}_k = \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{1}_3 \\ \mathbf{0}_4 \end{bmatrix},$$

represents the vector

$$\boldsymbol{\mu}_k = [0, 0, 0, 1, 1, 1, 0, 0, 0, 0]^T.$$

5.2 Parameter values

Each method requires the specification of one or several parameter values. The following values are applied in the simulations:

- **Gap, Reference Gap and Recursive Gap**:
  Maximum number of clusters: $K_{max} = 10$.
  Number of reference sets: $B = 50$.

- **ERA**:
  Number of reference sets: $B = 50$.
  Minimum number of samples in each cluster: $n_{min} = 5$.
  Number of repetitions: $R = 9$.
  Number of thresholds: $H = 0.2 \times (N - 1)$.

- **Silhouette**:
  Maximum number of clusters: $K_{max} = 10$.

- **In-group proportion**:
  Maximum number of clusters: $K_{max} = 10$.
  Number of repetitions: $R = 5$.
  Learning set fraction: $L_{frac} = \frac{1}{2}$.
  Threshold: $t = 0.95$.

- **Prediction strength**:
  Maximum number of clusters: $K_{max} = 10$.

Number of repetitions: $R = 5$.
Learning set fraction: $L_{frac} = \frac{1}{2}$.
Threshold: $t = 0.85$.

To a large extent, these values correspond to the the values specified by the authors in the paper where the method was initially presented.

5.3 Simulation scenarios

Four main simulation scenarios are considered to investigate the ability of the methods to find the correct number of clusters in a variety of settings, and Table 5.1 gives a brief outline of these. Scenario A represents the most standardized situation. Here, four clusters with mean vectors in equal distances from each other are generated, and all of the samples are simulated from normal distributions with variance σ^2. Scenario B is similar to Scenario A in that the four cluster means are equally distanced, but the setting is complicated by generating the samples from contaminated normal distributions. That is, the feature values for each sample are drawn with probability $1 - q$ from a normal distribution with variance σ^2, and with probability q from a normal distribution with variance $(c\sigma)^2$. In Scenario C, the samples are simulated from normal distributions with variance σ^2, but the cluster means are now defined such that the distances between them are unequal. For Scenario A, B and C we consider three versions, where the number of features (p) is 10, 100 and 1000, respectively.

In the last setting, Scenario D, sub-clusters are incorporated in the data sets. Two versions are considered here. In Scenario D1, five clusters, of which two are sub-clusters, are generated by drawing samples from normal distributions with variance σ^2. In Scenario D2, six clusters, of which three are sub-clusters, are generated from normal distributions, but the variance of two of the sub-clusters is smaller than that of the other clusters. The number of features (p) is 100 in both versions.

Across all of the scenarios, a range of variance values are applied to obtain clusters of varying cohesiveness. As the variance increases, samples from different clusters become more proximate, and clusters may even overlap. Hence by applying a range of variance values we can test each method's ability to find the correct number of clusters in settings of varying difficulty. In each scenario, a total of 100 data sets are generated for every variance value. The prime interest is in comparing the percentage of times that the methods find the correct number of clusters, and such results are listed for every scenario below. It is, however, also interesting to see whether the methods tend to overestimate or underestimate the number of clusters, and what solution they most frequently return. The full distributions of the number of clusters given by the methods in the various simulation scenarios are therefore listed in Appendix B.

Table 5.1: A brief outline of the simulation scenarios.

	K^*	Cluster mean distances	Variance	Sub-clusters	Dimension(s)
Scenario A	4	Equal	$\sigma_{kj}^2 = \sigma^2$	No	10,100,1000
Scenario B	4	Equal	$\sigma_{kj}^2 = (1 + qc^2 - q)\sigma^2$	No	10,100,1000
Scenario C	4	Unequal	$\sigma_{kj}^2 = \sigma^2$	No	10,100,1000
Scenario D1	5	Unequal	$\sigma_{kj}^2 = \sigma^2$	Yes	100
Scenario D2	6	Unequal	$\sigma_{kj}^2 = \sigma^2$ or $\sigma_{kj}^2 = 0.8\sigma^2$	Yes	100

5.3.1 Scenario A: 4 clusters with equally distanced cluster means

As mentioned earlier, this is the most standardized scenario as the distances between all pairs of cluster means are equal, and the variances are identical across all the p features and all the clusters ($\mathbf{\Sigma}_k = \sigma^2\mathbf{I}_p$). To construct four clusters we generate $n_k = 25$ samples from four different p-dimensional normal distributions, such that for the i'th sample in cluster k we have

$$\mathbf{x}_{ki} \sim N_p(\boldsymbol{\mu}_k, \sigma^2\mathbf{I}).$$

For a given number of features, p, the four cluster means are in this scenario defined as

$$\boldsymbol{\mu}_1 = \begin{bmatrix} \mathbf{1}_{\frac{p}{10}} \\ \mathbf{0}_{\frac{9p}{10}} \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} \mathbf{0}_{\frac{p}{10}} \\ \mathbf{1}_{\frac{p}{10}} \\ \mathbf{0}_{\frac{8p}{10}} \end{bmatrix}, \quad \boldsymbol{\mu}_3 = \begin{bmatrix} \mathbf{0}_{\frac{2p}{10}} \\ \mathbf{1}_{\frac{p}{10}} \\ \mathbf{0}_{\frac{7p}{10}} \end{bmatrix}, \quad \boldsymbol{\mu}_4 = \begin{bmatrix} \mathbf{0}_{\frac{3p}{10}} \\ \mathbf{1}_{\frac{p}{10}} \\ \mathbf{0}_{\frac{6p}{10}} \end{bmatrix}.$$

Hence each mean vector differs from the other mean vectors in $\frac{1}{5}$ of the p features, and the squared Euclidean distance between any pair of cluster means is $\frac{p}{5}$.

To test how each method's ability to find the correct number of clusters depends on the dimension of the data set, three versions of Scenario A are considered. In these, the number of features is set to either 10, 100 or 1000, and the three alternatives are referred to as Scenario A10, A100 and A1000, respectively. As mentioned above, the value of the variance can be varied to influence the cohesiveness of the clusters, and this gives us the possibility to study how cohesive the clusters need to be for each method to find the correct number of clusters. Since the distance between the cluster means increases as a function of the dimension (p), different values of σ^2 are used in the three versions to simulate clusters of varying degree of cohesiveness. Six values are applied in each case:

$$\sigma^2 \in \begin{cases} (0.025, 0.05, 0.075, 0.1, 0.125, 0.15) & \text{for } p = 10 \\ (0.1, 0.2, 0.3, 0.4, 0.5, 0.7) & \text{for } p = 100 \\ (0.8, 1.0, 1.2, 1.4, 1.6, 1.8) & \text{for } p = 1000 \end{cases}$$

Figure 5.1 shows the dendrograms resulting from hierarchical clustering of random data sets

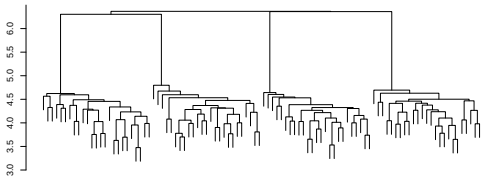
generated in Scenario A100 for the listed values of σ^2. Similar figures are found for Scenario A10 and A1000 in Appendix A (Figures A.1 and A.2). Note that the four clusters get less and less cohesive as the variance increases. The four clusters are for example easily identifiable in the dendrogram in Figure 5.1a, whereas it is harder to recognize four groups in the dendrogram in Figure 5.1f. In the latter dendrogram, several samples are also so dissimilar from the other samples that they are clustered in the top of the dendrogram. These samples may be seen as outliers, and such outliers will tend to complicate the estimation of the number of clusters for many methods (this issue will be discussed in more detail later). Below each dendrogram are the number of clusters found by the various methods. While all of the methods, except Reference Gap, are able to correctly estimate four clusters in the data sets with low to midrange values of σ^2, only ERA is successful for the two largest σ^2.

Scenario A results

For each alternative of Scenario A, and each value of σ^2, a total of 100 data sets were generated, and the various methods were applied to the data sets to estimate the number of clusters. Tables 5.2a - 5.2c list the percentage of times that each method was able to estimate the true number of clusters ($K^* = 4$). The last column of the tables gives the methods' relative ranks in the scenario, based on their average ranks over the six variance values. Note that these ranks are only indicative of each method's relative performance, as they do not give any information about the extent to which the success percentages of the methods differ. Another problem is that equal weight is given to the ranking for every value of σ^2 in the calculation of the average ranks. That a method fails in settings where the variance is low should be taken as a greater weakness of the method than if it fails for larger variances, however. Hence the actual success rates listed in the table must be considered along with the ranking of the methods.

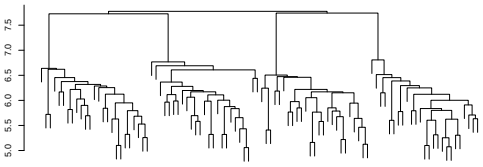
ERA receives the highest rank across all the dimensions in Scenario A, and this superiority is supported by the success percentages reported in the tables. The performance of ERA is as good as or (much) better than that of the other methods in the vast majority of the simulations. This is especially true for the 10- and 100-dimensional scenarios, where only the largest variance values make ERA's success drop to about 50 %. Notice, however, that these success rates are still far above the the other methods' success rates. In the 1000-dimensional case, ERA has a success rate of 85 % or more for $\sigma^2 \leq 1.4$. For the two largest values of σ^2 in this scenario, its performance is poor, but this is also the case for the other methods.

Silhouette indisputably gets the second highest rank in the 10-dimensional version of Scenario A, and does better, sometimes much better, than both Gap and Recursive Gap for all values of σ^2. Overall, Recursive Gap has somewhat higher success than Gap in this setting, and is ranked number 4. Gap, on the other hand, receives the second lowest ranking, which is also below Prediction strength (but by small margins). In the 100- and 1000-dimensional settings, Recursive Gap performs a bit better than both Silhouette and Gap, and hence gets ranked



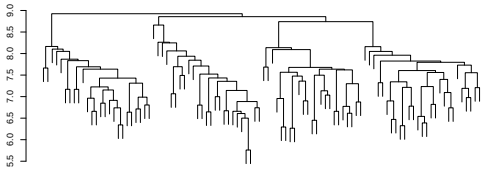
(a) $\boldsymbol{\sigma^2 = 0.1}$

Reference Gap:	2
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



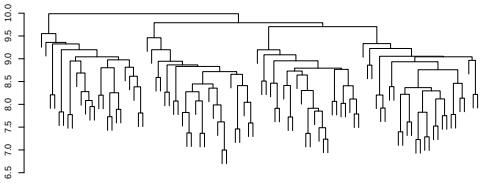
(b) $\boldsymbol{\sigma^2 = 0.2}$

Reference Gap:	2
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



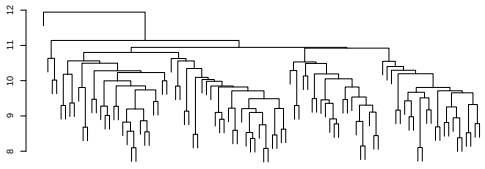
(c) $\boldsymbol{\sigma^2 = 0.3}$

Reference Gap:	3
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



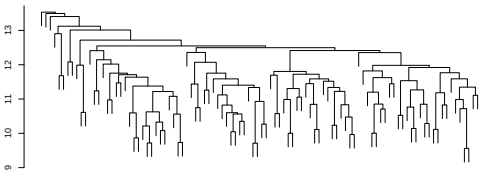
(d) $\boldsymbol{\sigma^2 = 0.4}$

Reference Gap:	3
In-group proportion:	4
Prediction strength:	1
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



(e) $\boldsymbol{\sigma^2 = 0.5}$

Reference Gap:	6
In-group proportion:	1
Prediction strength:	1
Silhouette:	2
Gap:	1
Recursive Gap:	1
ERA:	4



(f) $\boldsymbol{\sigma^2 = 0.7}$

Reference Gap:	9
In-group proportion:	1
Prediction strength:	1
Silhouette:	2
Gap:	1
Recursive Gap:	1
ERA:	4

Figure 5.1: Dendrograms resulting from hierarchical clustering of random data sets simulated for different values of σ^2 in Scenario A100 ("4 clusters with equally distanced cluster means - 100 dimensions"). Below each dendrogram are the estimated number of clusters in the data sets given by the various methods.

Table 5.2: The percentage of times that each method returned the correct number of clusters ($K^* = 4$) for different values of σ^2 in Scenario A ("4 clusters with equally distanced cluster means"). The last column of each table gives the relative ranks of the methods.

(a) Scenario A10 (10 dimensions)

Method $\boldsymbol{\sigma^2}$	**0.025**	**0.05**	**0.075**	**0.1**	**0.125**	**0.15**	**Rank**
ERA	100	100	99	94	81	51	1
Recursive Gap	97	91	60	30	9	4	4
Gap	97	90	57	19	6	0	6
Silhouette	100	100	94	59	31	11	2
In-group proportion	99	74	16	12	2	0	7
Prediction strength	100	95	28	0	0	0	5
Reference Gap	98	85	69	37	31	19	3

(b) Scenario A100 (100 dimensions)

Method $\boldsymbol{\sigma^2}$	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.7**	**Rank**
ERA	100	100	100	100	99	48	1
Recursive Gap	100	100	98	74	16	1	2
Gap	100	100	97	60	11	1	3
Silhouette	100	100	97	60	11	1	3
In-group proportion	100	100	44	35	8	1	5
Prediction strength	100	100	81	5	1	0	6
Reference Gap	94	60	56	49	24	11	7

(c) Scenario A1000 (1000 dimensions)

Method $\boldsymbol{\sigma^2}$	**0.8**	**1.0**	**1.2**	**1.4**	**1.6**	**1.8**	**Rank**
ERA	100	100	100	85	18	2	1
Recursive Gap	100	100	92	68	8	0	2
Gap	100	99	89	44	4	0	4
Silhouette	100	98	33	60	13	2	3
In-group proportion	99	72	42	50	22	10	5
Prediction strength	100	94	61	14	2	0	6
Reference Gap	45	33	25	21	29	28	6

number 2 in these scenarios. The performances of Silhouette and Gap are rather similar, in fact they actually have identical percentages of correct estimates in Scenario A100, which gives both of them rank 3. (As seen in Appendix B, however, the detailed result distributions differ). In Scenario A1000, Silhouette and Gap are ranked as number 3 and 4, respectively, though this difference in ranking is not obvious (for instance, Gap does much better than Silhouette for $\sigma^2 = 1.2$). A curiosity is that Silhouette for some reason has much higher success for $\sigma^2 = 1.4$ than for $\sigma^2 = 1.2$.

In-group proportion and Prediction strength consistently receive low ranks in Scenario A. In the 10-dimensional case, In-group proportion is in fact ranked the lowest, due to poor performance for all but the lowest value of σ^2. Prediction strength has high success for the two lowest values of σ^2, but then more or less collapses for the subsequent variances. This gives the method the fifth highest rank, barely ahead of Gap. In Scenario A100 and A1000, In-group

proportion and Prediction strength are ranked number 5 and 6, respectively. In these cases both methods do quite well for the lowest variances, with Prediction strength somewhat more successful than In-group proportion, whereas their performances drop for the midrange values of σ^2. For the three highest variances in both settings, In-group proportion has greater success than Prediction strength. In fact, Prediction strength consistently has the lowest success rates of all the methods for the three highest variances across all dimensions.

Reference Gap is the method which is the most adversely affected by high dimension in this scenario. In the 10-dimensional case, Reference Gap performs adequately, and even somewhat better than many of the other methods, including Recursive Gap and Gap. Hence it is ranked number 3 in Scenario A10. In the 100-dimensional, and especially the 1000-dimensional case, however, Reference Gap breaks down at much lower variance values than all of the other methods, and is thus (by far) the worst performer in these scenarios. Note that even though Reference Gap and Prediction strength are equally ranked in Scenario A1000, Prediction strength should still be considered the superior method of the two, since it does much better for the three lowest values of σ^2.

5.3.2 Scenario B: 4 clusters from contaminated normal distributions

The cluster means are in this scenario defined as in Scenario A, i.e.

$$\boldsymbol{\mu}_1 = \begin{bmatrix} \mathbf{1}_{\frac{p}{10}} \\ \mathbf{0}_{\frac{9p}{10}} \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} \mathbf{0}_{\frac{p}{10}} \\ \mathbf{1}_{\frac{p}{10}} \\ \mathbf{0}_{\frac{8p}{10}} \end{bmatrix}, \quad \boldsymbol{\mu}_3 = \begin{bmatrix} \mathbf{0}_{\frac{2p}{10}} \\ \mathbf{1}_{\frac{p}{10}} \\ \mathbf{0}_{\frac{7p}{10}} \end{bmatrix}, \quad \boldsymbol{\mu}_4 = \begin{bmatrix} \mathbf{0}_{\frac{3p}{10}} \\ \mathbf{1}_{\frac{p}{10}} \\ \mathbf{0}_{\frac{6p}{10}} \end{bmatrix}.$$

In the previous scenario, the values of the p features generated for each sample in cluster k were drawn from a p-dimensional normal distribution with mean vector $\boldsymbol{\mu}_k$ and variance σ^2. In microarrays experiments one sometimes observes that the data appear to come from so called contaminated normal distributions, i.e. distributions that are heavy-tailed relative to the normal distribution (e.g. Liestøl et al., 2008). In Scenario B, we therefore complicate the setting by generating the values of the p features from a mixture of normal distributions, where the variance is σ^2 with probability $1-q$ and $(c\sigma)^2$ with probability q (for some $c > 1$).

Formally, let Y represent a dummy variable that takes on the value 0 with probability $1-q$ and the value 1 with probability q, i.e.

$$\Pr(Y = 0) = 1 - q \quad \text{and} \quad \Pr(Y = 1) = q.$$

For each of the p features of a random sample $\mathbf{x}_{ki}$ we first draw the value of Y, and then

proceed to draw the actual value of the j'th feature of this sample from the distribution

$$x_{kij} \sim \begin{cases} N(\mu_{kj}, \sigma^2) & \text{if } Y_{kij} = 0 \\ N(\mu_{kj}, (c\sigma)^2) & \text{if } Y_{kij} = 1. \end{cases}$$

The value of each feature is thus drawn from a composite of two normal distributions with the same mean, but with different variances. This results in a distribution with somewhat heavier tails than the normal distribution, where the values of c and q influence how heavy the tails are.

The total variance of the j'th feature of a sample i in cluster k, denoted $\mathrm{Var}(x_{kij})$, is given by

$$\begin{aligned} \mathrm{Var}(x_{kij}) &= \mathrm{E}[\mathrm{Var}(x_{kij}|Y)] + \mathrm{Var}[\mathrm{E}(x_{kij}|Y)] \\ &= \mathrm{Var}(x_{kij}|Y=0)\Pr(Y=0) + \mathrm{Var}(x_{kij}|Y=1)\Pr(Y=1) + \mathrm{Var}(\mu_{kj}) \\ &= \sigma^2(1-q) + (c\sigma)^2 q \\ &= (1 + qc^2 - q)\sigma^2. \end{aligned} \tag{5.1}$$

Hence, since the final expression is independent of k and j, the variance is identical for all the features and for the four clusters. To create *contaminated* normal distributions, only a small fraction of the feature values should be generated from the distribution with the larger variance, and accordingly we choose $q = 0.05$. Hence a random 5 % of the p features of each sample are expected to be drawn from the distribution with variance $(c\sigma)^2$. As in Scenario A, we consider three alternatives where the number of features is 10, 100 and 1000, and refer to these as Scenario B10, B100 and B1000, respectively. In each variant, we apply $c = 2$ and $c = 3$, whereas the range of values for the total variance is defined according to the dimension:

$$\mathrm{Var}(x_{kij}) \in \begin{cases} (0.03, 0.05, 0.07) & \text{for } p = 10 \\ (0.1, 0.2, 0.3) & \text{for } p = 100 \\ (0.5, 0.75, 1.0) & \text{for } p = 1000 \end{cases}$$

Given the fixed values of c and $\mathrm{Var}(x_{kij})$, the value of σ^2 is calculated from (5.1) such that:

$$\sigma^2 = \mathrm{Var}(x_{kij})/(1 + qc^2 - q).$$

Figure 5.2 shows dendrograms resulting from hierarchical clustering of random data sets simulated from Scenario B100 with different combinations of c and $\mathrm{Var}(x_{kij})$. Similar figures are given in Appendix A for Scenario B10 and B1000 (Figures A.3 and A.4). Compared to the dendrograms in Figure 5.1 (Scenario A), the samples within each cluster are somewhat less

uniform, especially for $c = 3$. The number of clusters found by the various methods are listed below each dendrogram. For $c = 2$, all of the methods are able to find the four clusters, whereas for $c = 3$ the results depend on the total variance. That is, all the methods are successful for $\text{Var}(x_{kij}) = 0.1$, only Reference Gap fails for $\text{Var}(x_{kij}) = 0.2$, whereas only ERA gives the correct estimate for $\text{Var}(x_{kij}) = 0.3$.

Scenario B results

For each dimension, 100 data sets were generated for the different combinations of c and $\text{Var}(x_{kij})$, and the methods were applied to each simulation. Tables 5.3a - 5.3c show the percentage of times that each method returned the correct number of clusters ($K^* = 4$) in Scenario B10, B100 and B1000, respectively. The relative rank of each method, based on the average ranking over all combinations of c and $\text{Var}(x_{kij})$, is listed in the final column of each table.

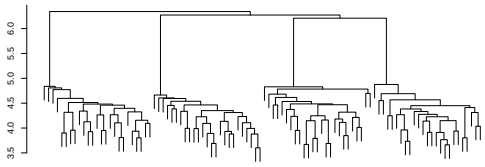
Remember that the difference between this scenario and Scenario A is that a few random features of each sample are drawn from a normal distribution with higher variance. A comparison of the results from Scenario A and B for similar values of σ^2 and $\text{Var}(x_{kij})$ indicate the effect of simulating from heavy-tailed distributions.

ERA is largely unaffected by applying a heavier-tailed distribution to generate data (at least for the values of $\text{Var}(x_{kij})$ and c tested here). In fact, ERA has close to perfect success rates for all values of $\text{Var}(x_{kij})$ and c, and is thus by far the best performing method across all dimensions.

All the other methods are to a much greater extent affected by the heavier-tailed distributions. The effect is naturally the greatest for $c = 3$ and the largest variance values. The relative performance of the methods are not very different compared to what we observed in Scenario A, however. Silhouette is still better than both Gap and Recursive Gap in the 10-dimensional case, and is ranked number 2 here. Gap receives the lowest rank in this scenario, but note that its success rates are not that different from those of its closest competitors. In the 100- and 1000-dimensional cases, Silhouette and Gap have identical success percentages, and are both ranked number 3. Recursive Gap overall does a little better, and is ranked number 2.

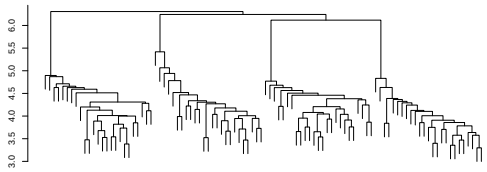
In-group proportion and Prediction strength are ranked number 5 and 6 in Scenario B. Notice, however, that In-group proportion seems to handle the effect of heavier-tailed distributions somewhat better than Prediction strength. While Prediction strength did a bit better than In-group proportion for similar variance values in Scenario A, In-group proportion generally does better than Prediction strength in Scenario B, in particular for $c = 3$ and the highest dimensions. In fact, in the 100- and 1000-dimensional cases, the success percentages of In-group proportion is not that far behind those of Silhouette and Gap.

Reference Gap struggles with the highest dimensions in this scenario as well. While it overall does a somewhat better job than the other methods, except ERA and Silhouette, in



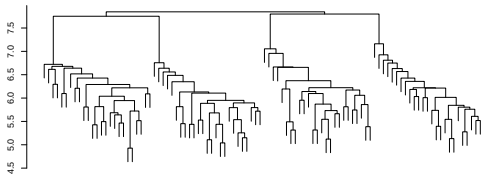
(a) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.1}, \boldsymbol{c} = \mathbf{2}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



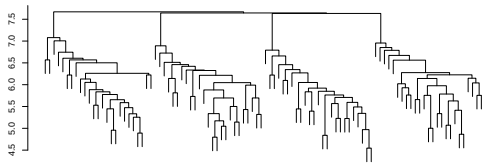
(b) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.1}, \boldsymbol{c} = \mathbf{3}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



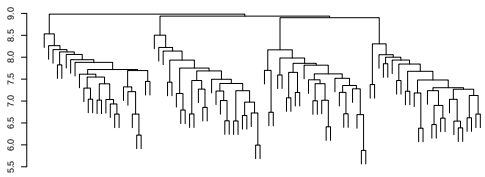
(c) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.2}, \boldsymbol{c} = \mathbf{2}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



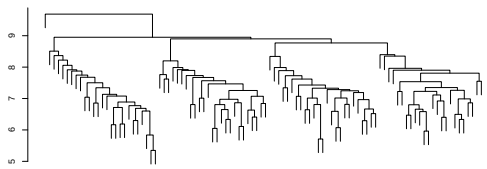
(d) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.2}, \boldsymbol{c} = \mathbf{3}$

Reference Gap:	3
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



(e) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.3}, \boldsymbol{c} = \mathbf{2}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



(f) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.3}, \boldsymbol{c} = \mathbf{3}$

Reference Gap:	5
In-group proportion:	1
Prediction strength:	1
Silhouette:	5
Gap:	1
Recursive Gap:	1
ERA:	4

Figure 5.2: Dendrograms resulting from hierarchical clustering of random data sets simulated for different values of $\mathrm{Var}(x_{kij})$ and c in Scenario B100 ("4 clusters from contaminated normal distributions - 100 dimensions"). Below each dendrogram are the estimated number of clusters in the data sets given by the various methods.

Table 5.3: The percentage of times that each method returned the correct number of clusters ($K^* = 4$) for the different values of c and $\mathrm{Var}(x_{kij})$ in Scenario B ("4 clusters from contaminated normal distributions"). The last column in each table lists the relative ranks of the methods.

(a) Scenario B10 (10 dimensions)

Method **Var(x_{kij})** **c**	**0.03 2**	**0.03 3**	**0.05 2**	**0.05 3**	**0.07 2**	**0.07 3**	**Rank**
ERA	100	100	100	100	100	99	1
Recursive Gap	95	76	81	35	48	14	4
Gap	96	74	75	31	48	11	7
Silhouette	100	89	94	44	66	14	2
In-group proportion	98	86	65	41	23	8	5
Prediction strength	100	79	82	22	24	2	5
Reference Gap	95	83	81	41	53	16	3

(b) Scenario B100 (100 dimensions)

Method **Var(x_{kij})** **c**	**0.1 2**	**0.1 3**	**0.2 2**	**0.2 3**	**0.3 2**	**0.3 3**	**Rank**
ERA	100	100	100	100	100	100	1
Recursive Gap	100	99	99	61	70	6	2
Gap	100	99	99	55	64	4	3
Silhouette	100	99	99	55	64	4	3
In-group proportion	100	99	98	54	40	4	5
Prediction strength	100	99	94	31	34	0	6
Reference Gap	94	93	68	44	47	11	7

(c) Scenario B1000 (1000 dimensions)

Method **Var(x_{kij})** **c**	**0.5 2**	**0.5 3**	**0.75 2**	**0.75 3**	**1.0 2**	**1.0 3**	**Rank**
ERA	100	100	100	100	100	97	1
Recursive Gap	100	98	100	75	90	17	2
Gap	100	97	99	66	87	8	3
Silhouette	100	97	99	66	87	8	3
In-group proportion	100	97	97	66	70	8	5
Prediction strength	100	98	96	38	50	0	6
Reference Gap	76	60	56	40	20	21	7

the 10-dimensional version, it is the worst performer in the 100-dimensional, and especially the 1000-dimensional case.

5.3.3 Scenario C: 4 clusters with unequally distanced cluster means

In Scenario A and B, the fact that the four cluster means were placed in equal distance to each other made the dendrograms look rather artificial compared to what is normally seen in the clustering of real data sets. Specifically, in Figure 5.1, we saw that none of the four clusters merged until at the very top of the dendrogram, and here they merged at about the same height because each pair of cluster means was equally distanced. In real data sets the clusters are more likely to be of varying degree of dissimilarity, and hence some clusters are closer to

each other than others. With this in mind a new scenario is constructed, where the distances between the cluster means are unequal. Hence some cluster pairs are more similar than others.

The four clusters are still generated from p-dimensional normal distributions such that for a random sample i in cluster k we have

$$\mathbf{x}_{ki} \sim N_p(\boldsymbol{\mu}_k, \sigma^2\mathbf{I}).$$

Each cluster is made up of $n_k = 25$ samples, and the variance is equal for all the features and all the clusters (hence $\boldsymbol{\Sigma}_k = \sigma^2\mathbf{I}_p$). To generate unequally distanced clusters, the cluster means are redefined to

$$\boldsymbol{\mu}_1 = \begin{bmatrix} \mathbf{2}_{\frac{p}{10}} \\ \mathbf{0}_{\frac{9p}{10}} \end{bmatrix}, \quad \boldsymbol{\mu}_2 = \begin{bmatrix} \mathbf{0}_{\frac{p}{10}} \\ \mathbf{1.6}_{\frac{p}{10}} \\ \mathbf{0}_{\frac{8p}{10}} \end{bmatrix}, \quad \boldsymbol{\mu}_3 = \begin{bmatrix} \mathbf{0}_{\frac{2p}{10}} \\ \mathbf{1.3}_{\frac{p}{10}} \\ \mathbf{0}_{\frac{7p}{10}} \end{bmatrix}, \quad \boldsymbol{\mu}_4 = \begin{bmatrix} \mathbf{0}_{\frac{3p}{10}} \\ \mathbf{1}_{\frac{p}{10}} \\ \mathbf{0}_{\frac{6p}{10}} \end{bmatrix}.$$

The squared Euclidean distance between every pair of cluster means is listed in Table 5.4. Cluster 1 is the most isolated cluster, while cluster 3 and 4 are the least separated.

As in Scenario A and B, the number of features (or dimension), p, is varied over three values: $p = 10$, $p = 100$ and $p = 1000$. These three alternatives are referred to as Scenario C10, C100 and C1000 respectively. Since the distances between the cluster means are now increased compared to that of Scenario A, new ranges of values for σ^2 are applied:

$$\sigma^2 \in \begin{cases} (0.05, 0.1, 0.125, 0.15, 0.2, 0.25) & \text{for } p = 10 \\ (0.2, 0.4, 0.5, 0.6, 0.8, 1.0) & \text{for } p = 100 \\ (1.0, 1.4, 1.6, 1.8, 2.0, 2.2) & \text{for } p = 1000 \end{cases}$$

The hierarchical clustering of data sets simulated from Scenario C100 with the listed values of σ^2 are shown in Figure 5.3. Similar figures for Scenario C10 and C1000 are given in Appendix A (Figures A.5 and A.6, respectively). In all of the dendrograms the clusters are merged at increasing heights; cluster 3 and 4 are joined first, then cluster 2, while cluster 1 is merged with the others at the final stage, because this cluster is the most separated from the others. Aas in the previous scenarios the clusters get less and less cohesive as the variance increases.

Table 5.4: Squared Euclidean distances between each pair of cluster means in Scenario C.

d^2_{Euc}	$\boldsymbol{\mu}_2$	$\boldsymbol{\mu}_3$	$\boldsymbol{\mu}_4$
$\boldsymbol{\mu}_1$	$0.656p$	$0.569p$	$0.5p$
$\boldsymbol{\mu}_2$		$0.425p$	$0.356p$
$\boldsymbol{\mu}_3$			$0.269p$

The number of clusters found by the various methods are listed below each dendrogram. All of the methods correctly estimate four clusters in the data sets with the two lowest variance values, whereas Reference Gap and Silhouette fail for the midrange values. In-group proportion, Recursive Gap and ERA are the only successful methods for $\sigma^2 = 0.8$, while ERA is the only method to succeed for $\sigma^2 = 1.0$.

Scenario C results

100 data sets were generated for each dimension and each value of σ^2. Tables 5.5a - 5.5c list the percentage of times that each method returned the true number of clusters ($K^* = 4$) for the different variance values in Scenario C10, C100 and C1000, respectively. The rank of each method is given in the final column of the tables.

Table 5.5: The percentage of times that each method returned the correct number of clusters ($K = 4$) for different values of σ^2 in Scenario C ("4 clusters with unequally distanced cluster means"). The last column of each table gives the rank of each method, based on the average ranking over the six values of σ^2.

(a) Scenario C10 (10 dimensions)

Method $\boldsymbol{\sigma^2}$	**0.05**	**0.1**	**0.125**	**0.15**	**0.2**	**0.25**	**Rank**
ERA	100	100	100	98	85	64	1
Recursive Gap	100	100	91	79	44	15	2
Gap	100	99	87	68	23	5	4
Silhouette	100	90	66	51	20	11	5
In-group proportion	99	65	30	18	8	3	7
Prediction strength	100	84	48	12	1	0	6
Reference Gap	100	91	66	55	36	28	3

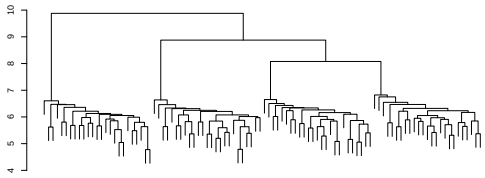
(b) Scenario C100 (100 dimensions)

Method $\boldsymbol{\sigma^2}$	**0.2**	**0.4**	**0.5**	**0.6**	**0.8**	**1.0**	**Rank**
ERA	100	100	100	97	94	70	1
Recursive Gap	100	100	97	80	33	5	2
Gap	100	100	91	65	8	0	3
Silhouette	95	22	9	12	2	0	7
In-group proportion	100	91	63	44	10	5	4
Prediction strength	100	96	72	35	0	0	5
Reference Gap	96	71	42	29	23	21	6

(c) Scenario C1000 (1000 dimensions)

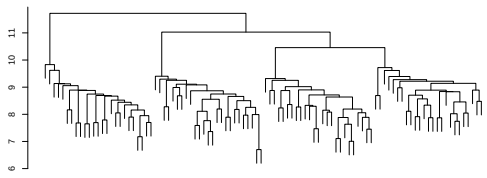
Method $\boldsymbol{\sigma^2}$	**1.0**	**1.4**	**1.6**	**1.8**	**2.0**	**2.2**	**Rank**
ERA	100	100	100	98	80	25	1
Recursive Gap	100	100	97	92	66	27	2
Gap	100	100	96	76	44	10	3
Silhouette	0	0	0	1	0	1	7
In-group proportion	100	96	84	63	58	33	3
Prediction strength	100	95	84	66	37	21	5
Reference Gap	40	53	53	39	35	19	6

The largest difference between the results in this scenario and those in Scenario A, is found



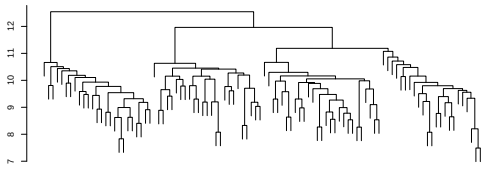
(a) $\boldsymbol{\sigma^2 = 0.2}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



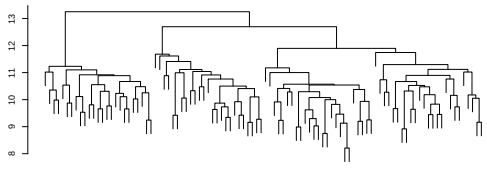
(b) $\boldsymbol{\sigma^2 = 0.4}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



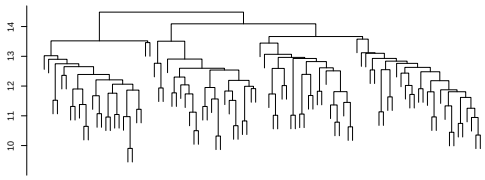
(c) $\boldsymbol{\sigma^2 = 0.5}$

Reference Gap:	3
In-group proportion:	4
Prediction strength:	4
Silhouette:	3
Gap:	4
Recursive Gap:	4
ERA:	4



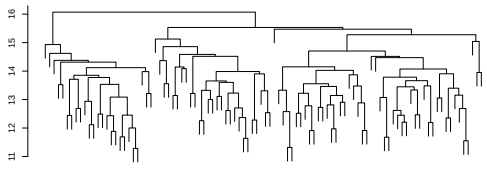
(d) $\boldsymbol{\sigma^2 = 0.6}$

Reference Gap:	3
In-group proportion:	4
Prediction strength:	4
Silhouette:	2
Gap:	4
Recursive Gap:	4
ERA:	4



(e) $\boldsymbol{\sigma^2 = 0.8}$

Reference Gap:	3
In-group proportion:	4
Prediction strength:	3
Silhouette:	2
Gap:	3
Recursive Gap:	4
ERA:	4



(f) $\boldsymbol{\sigma^2 = 1.0}$

Reference Gap:	3
In-group proportion:	3
Prediction strength:	2
Silhouette:	2
Gap:	3
Recursive Gap:	3
ERA:	4

Figure 5.3: Dendrograms resulting from hierarchical clustering of random data sets simulated for different values of σ^2 in Scenario C100 ("4 clusters with unequally distanced cluster means - 100 dimensions"). Below each dendrogram are the estimated number of clusters in the data sets given by the various methods.

for Silhouette. In fact, Silhouette stands out by its poor performance. It does a decent job in the 10-dimensional setting, where it is ranked number 5, above both In-group proportion and Prediction strength. However, in the 10-dimensional case in Scenario A, Silhouette also did quite a lot better than Gap and Recursive Gap. In the 100-dimensional and especially the 1000-dimensional setting, Silhouette performs very poorly compared to the other methods. In fact, in Scenario C100, Silhouette is only successful for the lowest variance value, while it has virtually no success in Scenario C1000. As a result, it justly receives the lowest ranking in these settings.

ERA is, as in the previous scenarios, the most successful method and is ranked number 1 for all dimensions in Scenario C. With a success rate of 100 % for the three lowest variances, and 80 % or more for all but the largest variance value across all dimensions, it does quite a lot better than the other methods. This is most pronounced for large values of σ^2 in the 10- and 100-dimensional settings, while Recursive Gap is not that far behind in the 1000-dimensional case.

Recursive Gap and Gap are the second and third best methods in Scenario C, respectively. (In the 10-dimensional setting, Gap is actually ranked behind Reference Gap, but this is only due to the results for the two highest variances. For the lower variances, Gap is the superior method of the two.) Notice that for midrange to high values of σ^2, Recursive Gap performs better than Gap, and though the same observation was made in Scenario A and B, it is somewhat more pronounced in this scenario.

In-group proportion and Prediction strength are both quite unsuccessful compared to the other methods in Scenario C10, and are hence ranked number 7 and 6, respectively. In Scenario C100 and especially in Scenario C1000, there is less difference in the performance of these methods and Gap. Combined with the fact that Silhouette and Reference Gap both do very poorly in these settings, both In-group proportion and Prediction strength improve their rankings. In fact, in the 1000-dimensional setting, In-group proportion and Gap share rank number 3. Also note that while Prediction strength tended to perform better than In-group proportion for the lowest values of σ^2 in Scenario A, this tendency is much weaker in Scenario C100 and C1000. As in the previous scenarios, Reference Gap performs much better in low dimension than in higher dimensions in Scenario C. In the 10-dimensional case, it performs quite decent, and nearly as good as Gap. In the 100- and 1000-dimensional settings, however, we once again observe that Reference Gap performs much poorer than most of the other methods for the lowest values of σ^2. Hence it is ranked number 6 in Scenario C100 and C1000, only ahead of Silhouette.

5.3.4 Scenario D: Data sets with sub-clusters

Since one of the virtues of both Recursive Gap and ERA is their ability to discover sub-clusters, a final scenario is set up to study their performance relative to the other methods in

such settings. The "sub-cluster" concept is a rather vague one, however. It refers to clusters that are very close or similar to each other compared to their distance to other clusters in the data set. They are therefore inclined to be perceived as one large (super-)cluster. Since what is regarded as sub-clusters depends on the *relative* distances between all the clusters in the data set, it should be noted that the sub-cluster concept is not a feature of the cluster itself. It is rather a property of the total set of clusters, and the presence of sub-clusters thus says something about the structure of the entire data set. What is perceived as sub-clusters is therefore affected by the entire data set that the sub-clusters are seen as a part of.

How similar the clusters must be relative to their similarity with others clusters to be called sub-clusters, is not clear. Hence, what is perceived as sub-clusters depends on the observer. In Scenario C, for example, we did not speak of sub-clusters, even though cluster 3 and 4 were (moderately) closer to each other than to the other clusters, and could hence in theory have been perceived as two sub-clusters. This was not the intention in this scenario, however. In Scenario D, the main purpose is to study the effect sub-clusters in the data sets. In this scenario, the sub-clusters are therefore constructed to be markedly more similar to each other than to the others clusters.

Two versions of Scenario D are considered. In both of these alternatives the number of features is 100, and three of the cluster means are defined as in Scenario C100:

$$\boldsymbol{\mu}_2 = \begin{bmatrix} \mathbf{0}_{10} \\ \mathbf{1.6}_{10} \\ \mathbf{0}_{80} \end{bmatrix}, \quad \boldsymbol{\mu}_3 = \begin{bmatrix} \mathbf{0}_{20} \\ \mathbf{1.3}_{10} \\ \mathbf{0}_{70} \end{bmatrix}, \quad \boldsymbol{\mu}_4 = \begin{bmatrix} \mathbf{0}_{30} \\ \mathbf{1}_{10} \\ \mathbf{0}_{60} \end{bmatrix}.$$

Remember that the last cluster mean in Scenario C100 was $\boldsymbol{\mu}_1 = \begin{bmatrix} \mathbf{2}_{10} \\ \mathbf{0}_{90} \end{bmatrix}$. To construct sub-clusters, we define sub-cluster means that are derived from $\boldsymbol{\mu}_1$. That is, in the first version, Scenario D1, the mean vectors of the two sub-clusters are defined as

$$\boldsymbol{\mu}_{1a} = \boldsymbol{\mu}_1 + \begin{bmatrix} \mathbf{1}_5 \\ \mathbf{0}_{95} \end{bmatrix} = \begin{bmatrix} \mathbf{3}_5 \\ \mathbf{2}_5 \\ \mathbf{0}_{90} \end{bmatrix} \text{ and } \quad \boldsymbol{\mu}_{1b} = \boldsymbol{\mu}_1 + \begin{bmatrix} \mathbf{0}_5 \\ \mathbf{1}_5 \\ \mathbf{0}_{90} \end{bmatrix} = \begin{bmatrix} \mathbf{2}_5 \\ \mathbf{3}_5 \\ \mathbf{0}_{90} \end{bmatrix},$$

while in the second version, Scenario D2, an additional sub-cluster is added with cluster mean

$$\boldsymbol{\mu}_{1c} = \boldsymbol{\mu}_1 + \begin{bmatrix} \mathbf{0}_{10} \\ \mathbf{1}_5 \\ \mathbf{0}_{85} \end{bmatrix} = \begin{bmatrix} \mathbf{2}_{10} \\ \mathbf{1}_5 \\ \mathbf{0}_{85} \end{bmatrix}.$$

Hence, in Scenario D1, we have a total of five clusters, which include sub-clusters 1a and 1b, as well as clusters 2, 3, and 4. In Scenario D2, the total number of clusters is six, which

include the five clusters listed for Scenario D1, plus one additional sub-cluster denoted 1c. The squared Euclidean distance between every pair of cluster means is listed in Table 5.6. Note that the sub-cluster means are much closer to each other than any other pair of cluster means.

Table 5.6: Squared Euclidean distance between each pair of cluster means in Scenario D.

d^2_{Euc}	$\boldsymbol{\mu}_{1b}$	$\boldsymbol{\mu}_{1c}$	$\boldsymbol{\mu}_2$	$\boldsymbol{\mu}_3$	$\boldsymbol{\mu}_4$
$\boldsymbol{\mu}_{1a}$	10	10	90.6	81.9	75
$\boldsymbol{\mu}_{1b}$		10	90.6	81.9	75
$\boldsymbol{\mu}_{1c}$			54.6	61.9	55
$\boldsymbol{\mu}_2$				42.5	35.6
$\boldsymbol{\mu}_3$					26.9

Another difference between the two versions relates to the variance of the sub-clusters. In Scenario D1, the variance is the same for all of the clusters, including the sub-clusters. Hence, for $k = 1a, 1b, 2, 3, 4$, samples are drawn from multivariate normal distributions such that

$$\mathbf{x}_{ki} \sim N_{100}(\boldsymbol{\mu}_k, \sigma^2\mathbf{I}). \tag{5.2}$$

In Scenario D2, on the other hand, the variance of sub-clusters 1a and 1b is defined to be 80 % that of the variance in the other clusters. Hence for $k = 1c, 2, 3, 4$, the generated as in (5.2), while for $k = 1a, 1b$ the samples are generated such that

$$\mathbf{x}_{ki} \sim N_{100}(\boldsymbol{\mu}_k, 0.8\sigma^2\mathbf{I}).$$

In both versions, the number of samples in the sub-clusters (1a,1b,1c) is 15, while the number of samples in clusters 2, 3, and 4 is 25. As in the previous scenarios, a range of values for σ^2 are applied to make the clusters less and less cohesive, and the following values are used in both version D1 and D2:

$$\sigma^2 \in (0.05, 0.1, 0.15, 0.2, 0.25, 0.3).$$

Figure 5.4 and 5.5 show the hierarchical clustering of data sets simulated from Scenario D1 and D2, respectively. The sub-clusters are located to the left in all of the dendrograms, and stand out from the other clusters by being much closer to each other. As the value of σ^2 increases, these clusters get less and less separable, and are thus more and more inclined to be regarded as one large super-cluster. In Figure 5.5, note that while all the three sub-clusters are much closer to each other than to the other clusters, sub-clusters 1a and 1b are also somewhat more cohesive than the others (because the variance is smaller). Below each dendrogram are the estimated number of clusters in the data sets by the various methods. Prediction strength, Recursive Gap and ERA stand out as the most effective methods in Figure 5.4, while Recursive

Gap and ERA are successful for the greatest share of variance values in Figure 5.5.

Scenario D results

A total of 100 data sets were generated for the different values of σ^2 in Scenario D1 and D2. Table 5.7a and 5.7b show the percentage of times that each method returned the correct number of clusters, $K^* = 5$ and $K^* = 6$, in Scenario D1 and D2, respectively. The ranks of the various methods are listed in the final columns.

Table 5.7: The percentage of times that each method returned the correct number of clusters ($K^* = 5$ in 5.7a and $K^* = 6$ in 5.7b) for different values of σ^2 in Scenario D ("Data sets with sub-clusters"). The final column of each table lists the ranking of the various methods in that scenario.

(a) Scenario D1 (same variance)

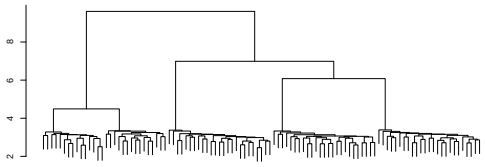
Method σ^2	**0.05**	**0.1**	**0.15**	**0.2**	**0.25**	**0.3**	**Rank**
ERA	100	100	100	95	64	22	1
Recursive Gap	100	100	100	90	53	16	2
Gap	100	90	15	1	0	0	5
Silhouette	0	0	0	0	0	0	7
In-group proportion	100	100	75	50	29	16	3
Prediction strength	100	100	87	29	12	9	4
Reference Gap	78	28	4	1	0	0	6

(b) Scenario D2 (smaller variance)

Method σ^2	**0.05**	**0.1**	**0.15**	**0.2**	**0.25**	**0.3**	**Rank**
ERA	100	100	99	91	58	33	1
Recursive Gap	100	100	93	62	24	3	2
Gap	100	55	1	0	0	0	4
Silhouette	0	0	0	0	0	0	7
In-group proportion	99	90	9	1	0	0	3
Prediction strength	100	74	0	0	0	0	5
Reference Gap	7	0	1	1	0	0	6

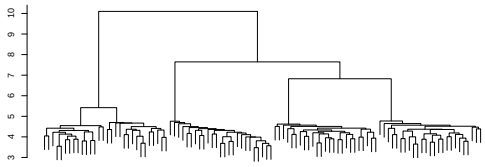
There are great differences in the performance of the various methods in these settings. ERA and Recursive Gap are by far the most effective methods in both Scenario D1 and D2, and are justly ranked number 1 and 2, respectively. In Scenario D1, these methods have quite similar success rates, with ERA a somewhat more effective for the largest values of σ^2. In Scenario D2 the methods' performances are similar for the lowest variances, whereas ERA has considerably more success for the three largest variances.

In-group proportion and Prediction strength are the third and fourth best methods, respectively, in Scenario D. These methods are quite successful for the lower values of σ^2, but are outperformed by ERA and Recursive Gap for the midrange to high variances, in particular in Scenario D2. Gap also performs well for the lowest values of σ^2, but is more or less unsuccessful for the larger variances. Note that Prediction strength is unjustly ranked behind Gap in Scenario D2, despite a significantly higher success rate for $\sigma^2 = 0.1$. This is due to Gap having



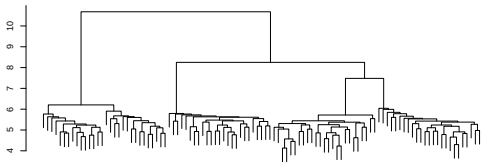
(a) $\boldsymbol{\sigma^2 = 0.05}$

Reference Gap:	2
In-group proportion:	5
Prediction strength:	5
Silhouette:	4
Gap:	5
Recursive Gap:	5
ERA:	5



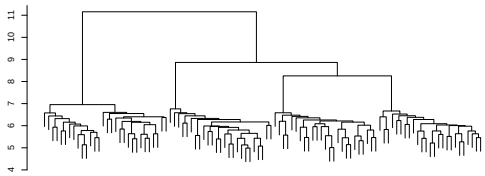
(b) $\boldsymbol{\sigma^2 = 0.1}$

Reference Gap:	4
In-group proportion:	5
Prediction strength:	5
Silhouette:	2
Gap:	5
Recursive Gap:	5
ERA:	5



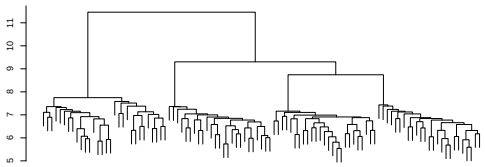
(c) $\boldsymbol{\sigma^2 = 0.15}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	5
Silhouette:	2
Gap:	4
Recursive Gap:	5
ERA:	5



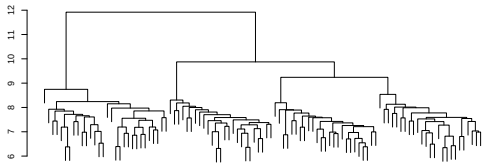
(d) $\boldsymbol{\sigma^2 = 0.2}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	5
Silhouette:	2
Gap:	4
Recursive Gap:	5
ERA:	5



(e) $\boldsymbol{\sigma^2 = 0.25}$

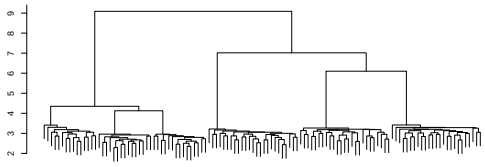
Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	2
Gap:	4
Recursive Gap:	4
ERA:	4



(f) $\boldsymbol{\sigma^2 = 0.3}$

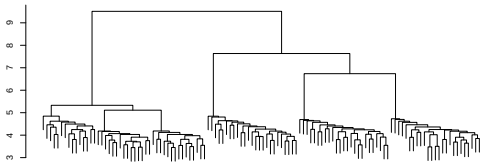
Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	2
Gap:	4
Recursive Gap:	4
ERA:	4

Figure 5.4: Dendrograms resulting from hierarchical clustering of random data sets simulated for different values of σ^2 in Scenario D1 ("Data sets with sub-clusters - same variance"). Below each dendrogram are the estimated number of clusters in the data sets given by the various methods ($K^* = 5$ is the true number of clusters).



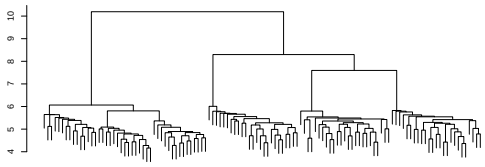
(a) $\boldsymbol{\sigma^2 = 0.05}$

Reference Gap:	2
In-group proportion:	6
Prediction strength:	6
Silhouette:	4
Gap:	6
Recursive Gap:	6
ERA:	6



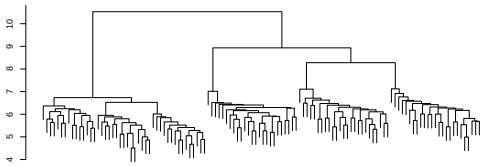
(b) $\boldsymbol{\sigma^2 = 0.1}$

Reference Gap:	2
In-group proportion:	6
Prediction strength:	6
Silhouette:	4
Gap:	6
Recursive Gap:	6
ERA:	6



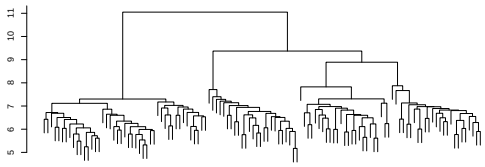
(c) $\boldsymbol{\sigma^2 = 0.15}$

Reference Gap:	2
In-group proportion:	5
Prediction strength:	4
Silhouette:	2
Gap:	4
Recursive Gap:	6
ERA:	6



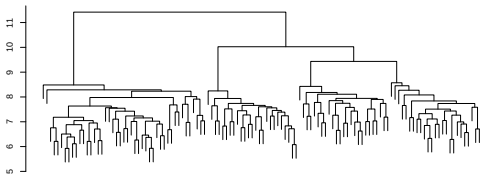
(d) $\boldsymbol{\sigma^2 = 0.2}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	2
Gap:	4
Recursive Gap:	6
ERA:	6



(e) $\boldsymbol{\sigma^2 = 0.25}$

Reference Gap:	2
In-group proportion:	4
Prediction strength:	4
Silhouette:	2
Gap:	4
Recursive Gap:	6
ERA:	6



(f) $\boldsymbol{\sigma^2 = 0.3}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	2
Gap:	4
Recursive Gap:	4
ERA:	4

Figure 5.5: Dendrograms resulting from hierarchical clustering of random data sets simulated for different values of σ^2 in Scenario D2 ("Data sets with sub-clusters - lower variance"). Below each dendrogram are the estimated number of clusters in the data sets given by the various methods ($K^* = 6$ is the correct answer).

minimally more success for $\sigma^2 = 0.15$.

Reference Gap has a poor performance in Scenario D, and is only moderately successful for the lowest value of σ^2 in version D1. By far the least successful method is, however, Silhouette, which is never able to estimate the correct number of clusters in neither Scenario D1 nor Scenario D2. Silhouette hence has trouble separating sub-clusters, even when they are quite cohesive. The method actually tends to estimate 2 clusters in many of the simulations (see Appendix A), thus it also seems that it has trouble separating the other clusters. Silhouette also had great trouble in Scenario C, and it thus appears that it does not perform well whenever some clusters are more similar than others (particularly in high dimension).

5.4 Summarized results

Overall, ERA stands out as the most effective method in the simulation scenarios considered here. ERA performs as good as or better than the other methods for nearly all the variance values in all the scenarios. In many of the simulations its success is also much higher than the success of the other methods. This is especially the case for the midrange to high variance values, and is more pronounced in the 10- and 100-dimensional settings than in the 1000-dimensional settings. ERA is particularly effective, compared to the other methods, when the clusters are generated from contaminated normal distributions (Scenario B) and when there are sub-clusters in the data sets (Scenario D).

Recursive Gap is overall the second most successful method in the simulation scenarios. It is ranked number 2 in a majority of settings, and is, save ERA, only beaten by Silhouette and Reference Gap in Scenario A10 and B10. Recursive Gap is in particular successful relative to the other methods in Scenario D, where its performance is nearly as good as ERA's performance.

Gap has average success rates, and is ranked number 3 or 4 in many of the scenarios. For the most part, Gap does somewhat better than In-group proportion and Prediction strength in Scenario A, B and C. In Scenario D, on the other hand, these methods are more successful than Gap. Hence both In-group proportion and Prediction strength seem somewhat more competent at handling sub-clusters than Gap.

Silhouette and Reference Gap stand out as the methods with the most varying performances. Silhouette performs as good as or better than Gap in Scenario A and B, but is to a great extent unsuccessful in Scenario C (100 and 1000 dimensions) and in Scenario D. This shows that Silhouette is inadequate in situations where some clusters are closer than others - especially in high dimensions. Reference Gap, on the other hand, consistently performs much better in low dimension than in high dimensions. In Scenario A, B and C, Reference Gap performs quite good in the 10-dimensional cases, but in the 100- and especially 1000-dimensional cases, its performance is poor. It is also to a great extent unsuccessful in Scenario D. Another flaw of Reference Gap is that its estimates tend to vary over a range of values for K, instead

of centering on one or two values as the other methods do. This can be seen in the detailed result tables in Appendix B. Hence Reference Gap seems somewhat random and thus less trustworthy.

5.5 Discussion

The results from the simulations above indicate that ERA is the best method - at least for the scenarios studied here. To get some insight into what the problems of the other methods may be, we look at some examples in which these methods fail in this section. Further, we discuss some features of ERA that contribute to its success in the simulation scenarios.

5.5.1 Result curves

By plotting the statistic applied by each method as a function of the number of clusters K, we can get an indication of what makes the methods fail, and by what margin. Such curves will in the following be referred to as *result curves*. Earlier in this chapter (and in Appendix A), dendrograms were plotted for the various simulation scenarios, along with the number of clusters estimated by the methods. In some cases, it turned out that one or more of the methods failed in finding the correct answer, even when the number of clusters seemed rather clear in the dendrogram. Figures 5.6 - 5.10 show result curves for some of these situations. Notice that result curves are not given for Recursive Gap and ERA. This is due to their recursive nature, which implies that result curves would have to be plotted for each recursive run.

The top left panel of Figure 5.6 shows a dendrogram for a data set simulated from Scenario A10, where $\sigma^2 = 0.05$ (cf. Figure A.1b). The other panels show result curves for the methods when applied on this data set, and the number of clusters estimated by methods are marked by a star. In this example, In-group proportion is the only method to fail, since it finds only two clusters instead of the correct number which is four. However, In-group proportion's result curve reveals that the correct solution is missed by only a small margin, since IGP(4) is just below the threshold at $t = 0.95$ (dotted line). The use of such a high threshold may to some extent explain why In-group proportion sometimes fails even when the variance is small. Also note in this example that Prediction strength barely succeeds as PS(4) is just above the threshold at $t = 0.85$.

Figure 5.7 gives an example in which Gap fails. For this data set, Gap wrongly estimates one cluster, in spite of the fact that the Gap curve is clearly maximized for $K = 4$. This happens because Gap(2) is not significantly larger than Gap(1), and $K = 1$ is then the smallest K for which $\text{Gap}(K) \geq \text{Gap}(K+1) - s_{K+1}$ (the Gap criterion). This observation is not uncommon in cases where Gap is unsuccessful. Notice that Gap is not the only method to fail in this example. In-group proportion and Prediction strength also estimate that there is only one cluster present in the data set, but as seen from the result curves, IGP(4) and PS(4) are not

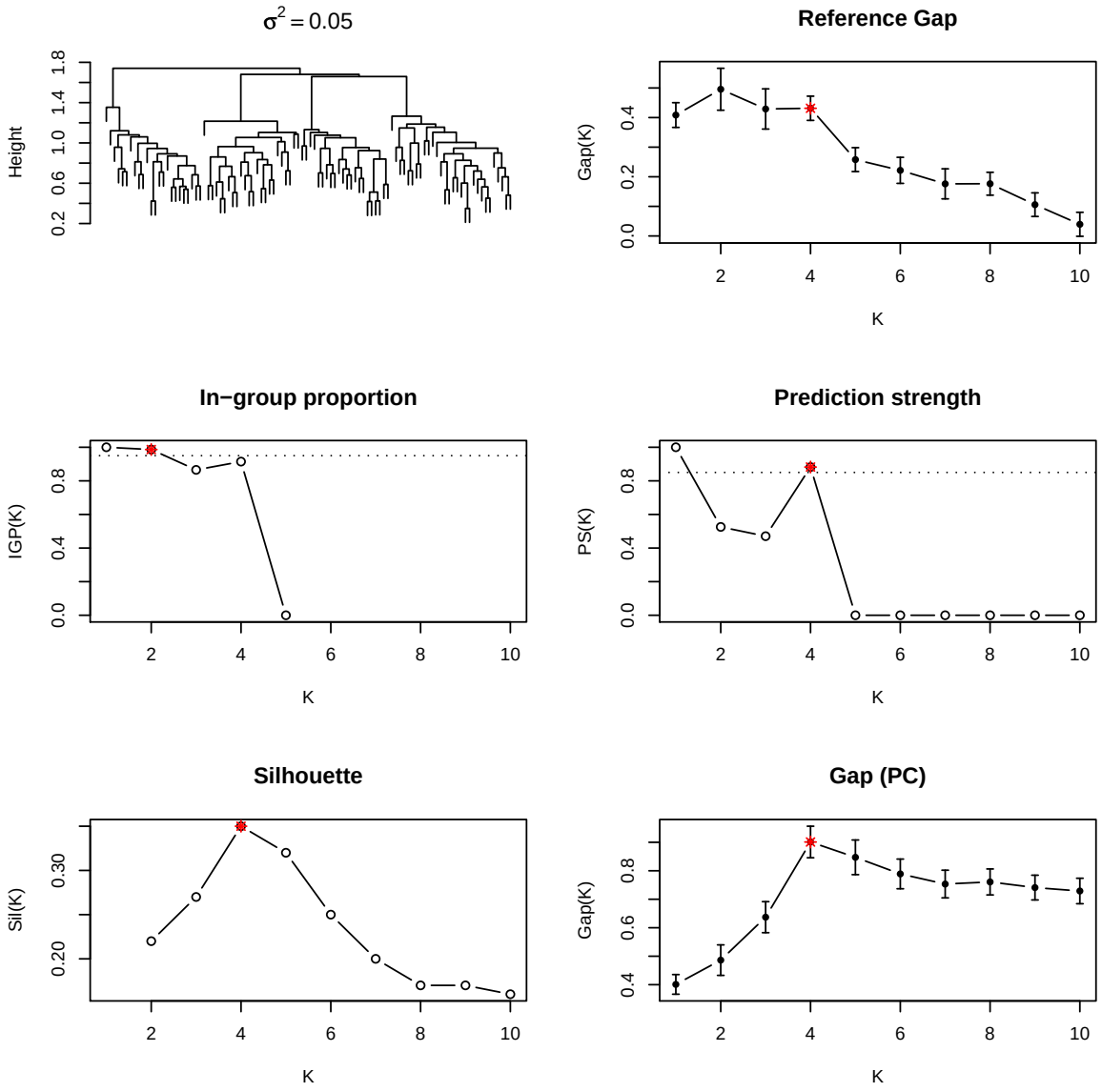


Figure 5.6: An example where In-group proportion fails. The left top panel shows the hierarchical clustering of a data set simulated from Scenario A10 with $\sigma^2 = 0.05$, in which four clusters are quite evident. The remaining panels show the result curves for Reference Gap, In-group proportion, Prediction strength, Silhouette and Gap, respectively. The estimated number of clusters are marked by a star. Note that In-group proportion misses the correct solution ($K = 4$) by a very small margin, since IGP(4) is just below the threshold marked by the dotted line ($t = 0.95$).

too far from the respective thresholds. Also note that since Gap only finds one cluster in this example, Recursive Gap is also bound to wrongly estimate one cluster (cf. Figure A.1d).

The simulations revealed that Silhouette has problems when some of the clusters are closer than others, as the situation was in Scenario C and D. In Scenario C, Silhouette particularly struggled in the 1000-dimensional case. The top left panel in Figure 5.8 shows a dendrogram from Scenario C1000, where $\sigma^2 = 1.0$. The dendrogram clearly indicates the presence of four clusters, still Silhouette, as the only method, estimates that there are only two clusters

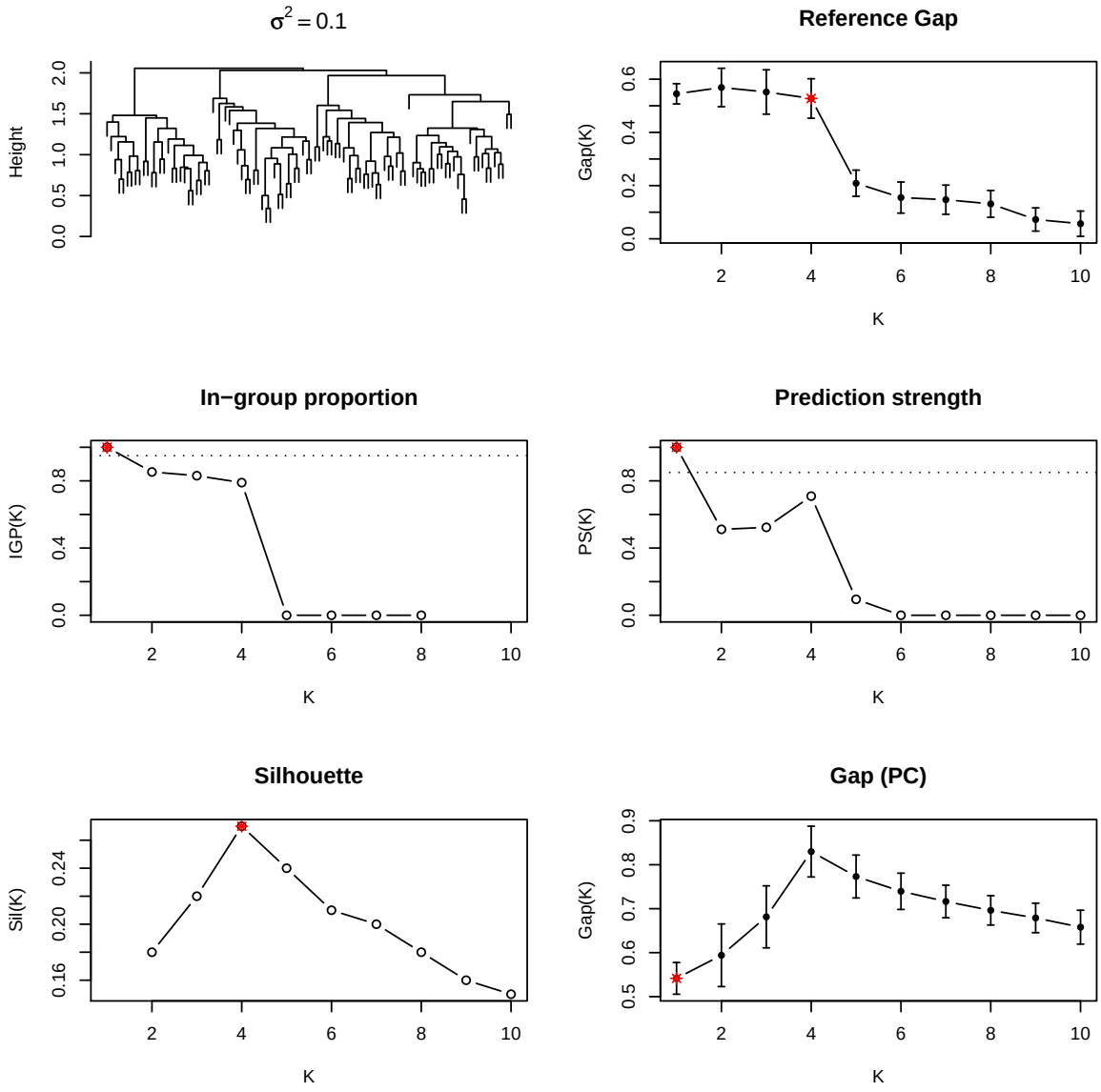


Figure 5.7: An example where Gap fails. The left top panel shows the dendrogram for a data set simulated from Scenario A10 ($\sigma^2 = 0.1$), in which four clusters are present. Result curves are given for various methods in the other panels. The result curve for Gap is maximized for $K = 4$, but, in line with the Gap criterion, Gap still returns the solution $\hat{K} = 1$ because $\text{Gap}(1) \geq \text{Gap}(2) - s_2$.

in the data set. As seen from the result curve, however, the average silhouettes at $K = 2$ and $K = 4$ barely differ. Hence Silhouette fails by a very small margin. Note that for the five smallest values of K the average silhouettes differ by less than 0.015, and that the average silhouette for all values of K are very small (below 0.1). According to the guidelines of Kaufman & Rousseeuw (1990) this corresponds to "no substantial structure" (cf. Section 3.5), and that another method may be preferable. Silhouette result curves for the other 1000-dimensional scenarios also revealed very low average silhouettes (results not shown), even when Silhouette actually estimated the correct number of clusters. In general it seems that the

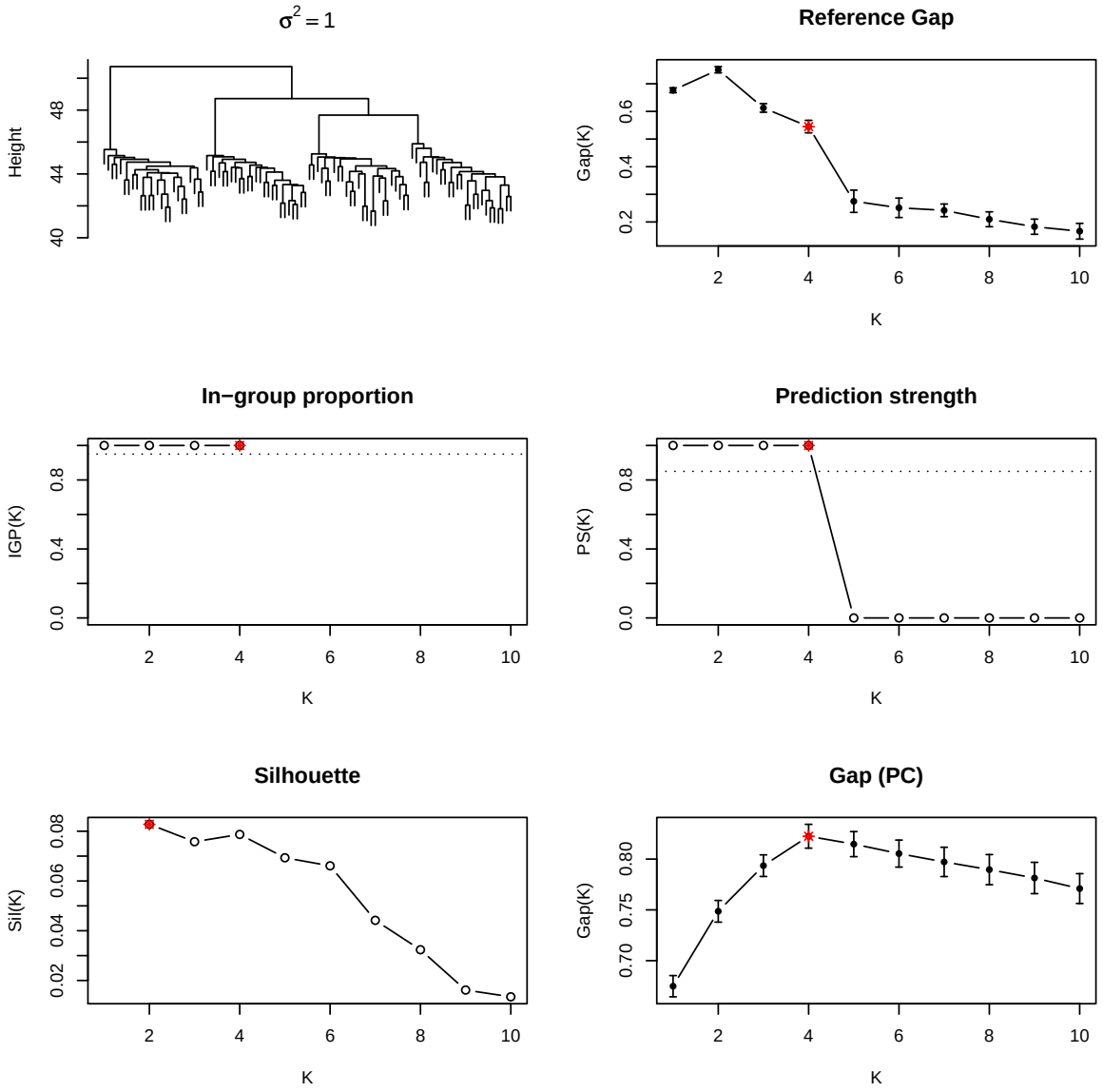


Figure 5.8: The data set clustered in the top left panel is simulated from Scenario C1000, with $\sigma^2 = 1.0$, and is made up of four clusters. Silhouette is the only unsuccessful method in this example as it only estimates two clusters. Note, however, that there is a very small difference between the average silhouettes at $K = 2$ and $K = 4$, hence Silhouette just barely fails.

average silhouettes fall as the dimension gets higher, which may indicate that Silhouette is less useful for high-dimensional data sets.

As seen earlier, the performance of Reference Gap tended to decrease substantially as the dimension got higher. Remember that Reference Gap was suggested as an alternative version of the Gap method, and that the criterion used in this method was rather intuitive and based on observations for data sets in low dimensions. Hence the fact that Reference Gap has a poor performance in high dimensions may indicate that the suggested criterion is not suitable as the dimension increases. Figure 5.9 gives an example where Reference Gap is the only method

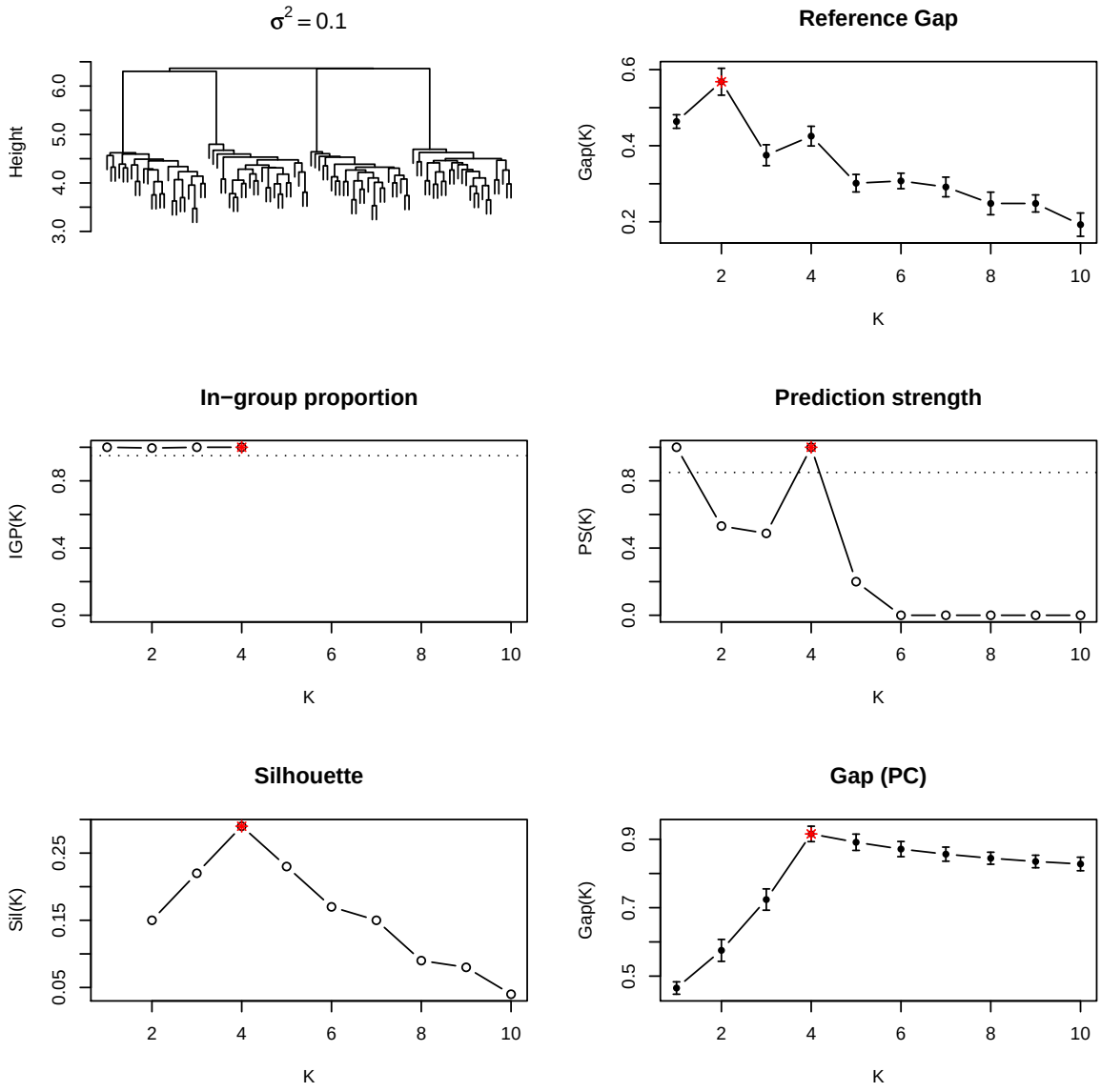


Figure 5.9: An example where Reference Gap is the only method to fail. The top left panel shows the clustering of a data set simulated from Scenario A100, with $\sigma^2 = 0.1$. The dendrogram clearly indicates the presence of four clusters in the data set, but Reference Gap returns the result $\hat{K} = 2$, because the drop in the result curve is the largest from $K = 2$ to $K = 3$.

to fail. In this example, a data set with four clusters is simulated from Scenario A100, with $\sigma^2 = 0.1$. Reference Gap fails because the drop in the Reference Gap curve is larger from $K = 2$ to $K = 3$ than from $K = 4$ to $K = 5$.

5.5.2 Horizontal cut, outliers and minimum cluster size

Two features of ERA contribute to the success of this method compared to the others methods. Firstly, ERA is not based on a horizontal cut of the dendrogram. Secondly, one of the parameters in the method is that one may demand that each cluster must contain a minimum number

of samples. These two qualities help ERA overcome some problems that other methods struggle with. Consider for example the dendrogram in Figure 5.10. A horizontal cut producing four clusters will divide the rightmost cluster into two clusters (where one of them will consist of only one sample), while keeping the two true clusters in the middle of the dendrogram merged. Hence the four true clusters may never be found by applying a static horizontal cut, and the methods whose result curves are shown in Figure 5.10, are therefore unsuccessful. ERA and Recursive Gap, on the other hand, continues to search for sub-clusters within each of the three clusters found by Gap. This feature makes them able to find the four true clusters in this example (cf. Figure A.3d).

Figure 5.11 gives another illustration of ERA's qualities. The dendrogram shown in this figure is based on the hierarchical clustering of a data set simulated from Scenario C10, with $\sigma^2 = 0.125$. Notice that, similar to the previous example, a horizontal cut at $K = 4$ will give us two true clusters, one cluster made up of only one sample, and one large cluster in which two true clusters are kept merged. Hence the methods that apply horizontal cuts of the dendrogram all fail, as seen in the result curves. A difference between this and the previous example is, however, that Recursive Gap is not able to find the true number of clusters either. This is because of the single sample located on top of the two rightmost clusters, which stops the method from finding any sub-clusters. ERA, on the other hand, discards this sample because it is required to find clusters with a minimum of 5 samples, and is thereafter able to locate the two last true clusters (cf. Figure A.5c).

A third illustration of ERA's virtues is given in Figure 5.12, where a data set is simulated from Scenario B100 with $\text{Var}(x_{kij}) = 0.3$ and $c = 3$. In the dendrogram resulting from the hierarchical clustering of this data set, one sample is so different from all the other samples that it is placed as an outlier in the top of the dendrogram. A horizontal cut producing four clusters will in this case give us one cluster made up of just one sample, two true clusters and one cluster in which two of the true clusters remain merged. The methods are differently affected by this outlier, and only ERA is able to find four clusters. In-group proportion, Prediction strength and Gap all agree that there is only one cluster, while Silhouette and Reference Gap estimates five clusters, of which one of them only consists of one sample. Note that Recursive Gap is also bound to return only one cluster since no recursive runs are started when Gap estimates only one clusters. ERA, on the other hand, is not stopped by Gap finding only one cluster, and continues to recursively search the dendrogram. Also, since ERA is required to return clusters with a minimum of 5 samples, it discards the outlier in the top of the dendrogram, and thereafter finds the four true clusters (cf. Figure 5.2f).

If many of the simulated data sets contained such outliers, it may explain why ERA is so successful compared to the other methods. In the next chapter, an effort will be made to find out to what extent the presence of such outliers has affected the simulation results.

Finally, note that none of the methods are actually specified to use a horizontal cut. Still,

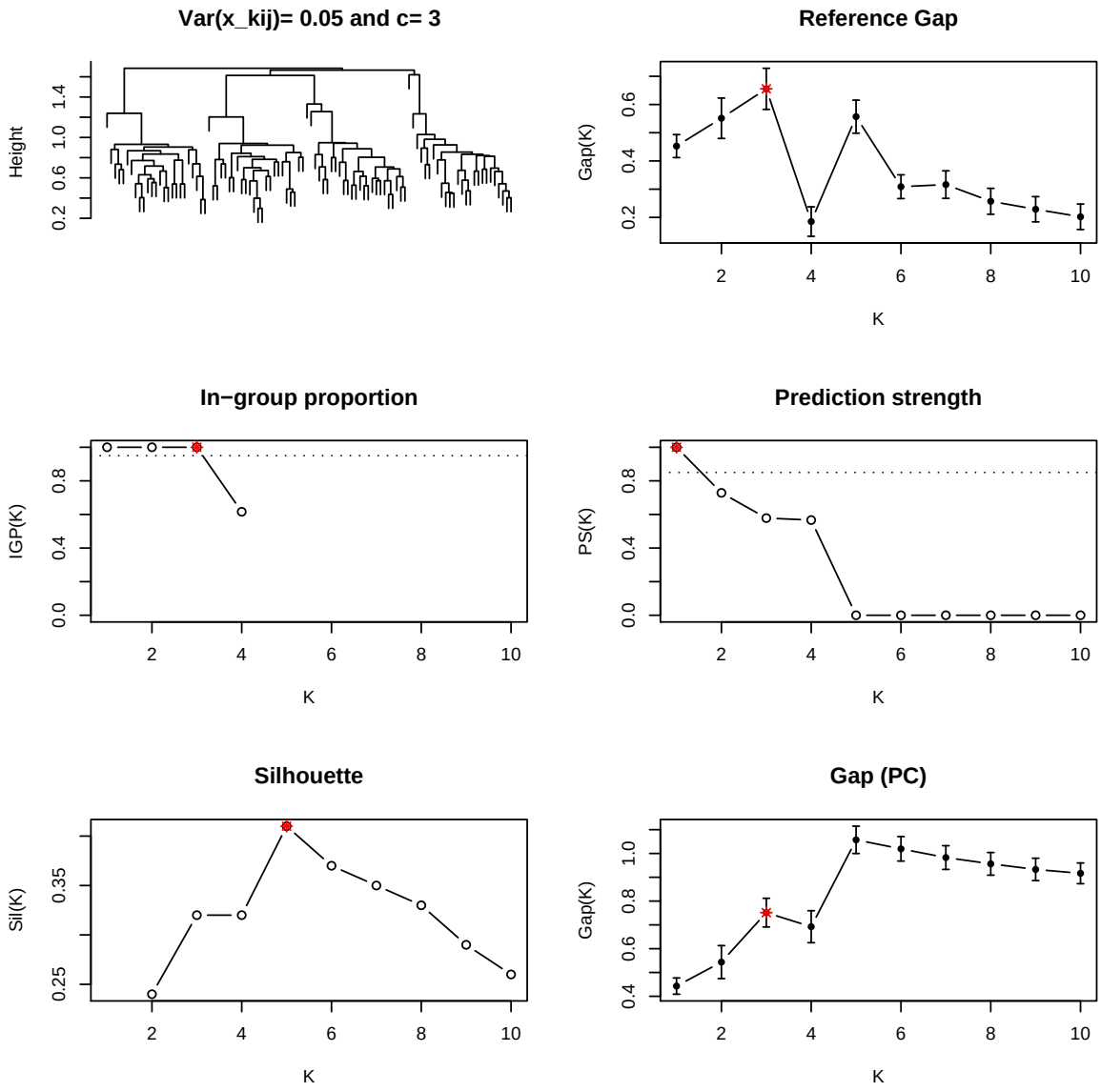


Figure 5.10: An example where a horizontal cut will never give us the four true clusters. The dendrogram in the top left panel shows the hierarchical clustering of a data set simulated from Scenario B10, with $\mathrm{Var}(x_{kij}) = 0.05$ and $c = 3$. Due to an extreme observation in the rightmost cluster, a horizontal cut at $K = 4$ will divide this cluster in two, while keeping the two clusters in the middle merged. All of the methods whose result curves are shown in this figure fail, because they all apply horizontal cuts. ERA and Recursive Gap, on the other hand, continue to search for sub-clusters within the three clusters found by Gap, and are therefore able to find all of the four true clusters.

no other suggestions for producing a given number of clusters from hierarchical clustering was made by the authors of the methods. The horizontal cut was therefore applied in the simulations since this is the most standard method. Similarly, all of the methods could in theory have been adapted to require that each cluster must consist of a minimum number of samples.

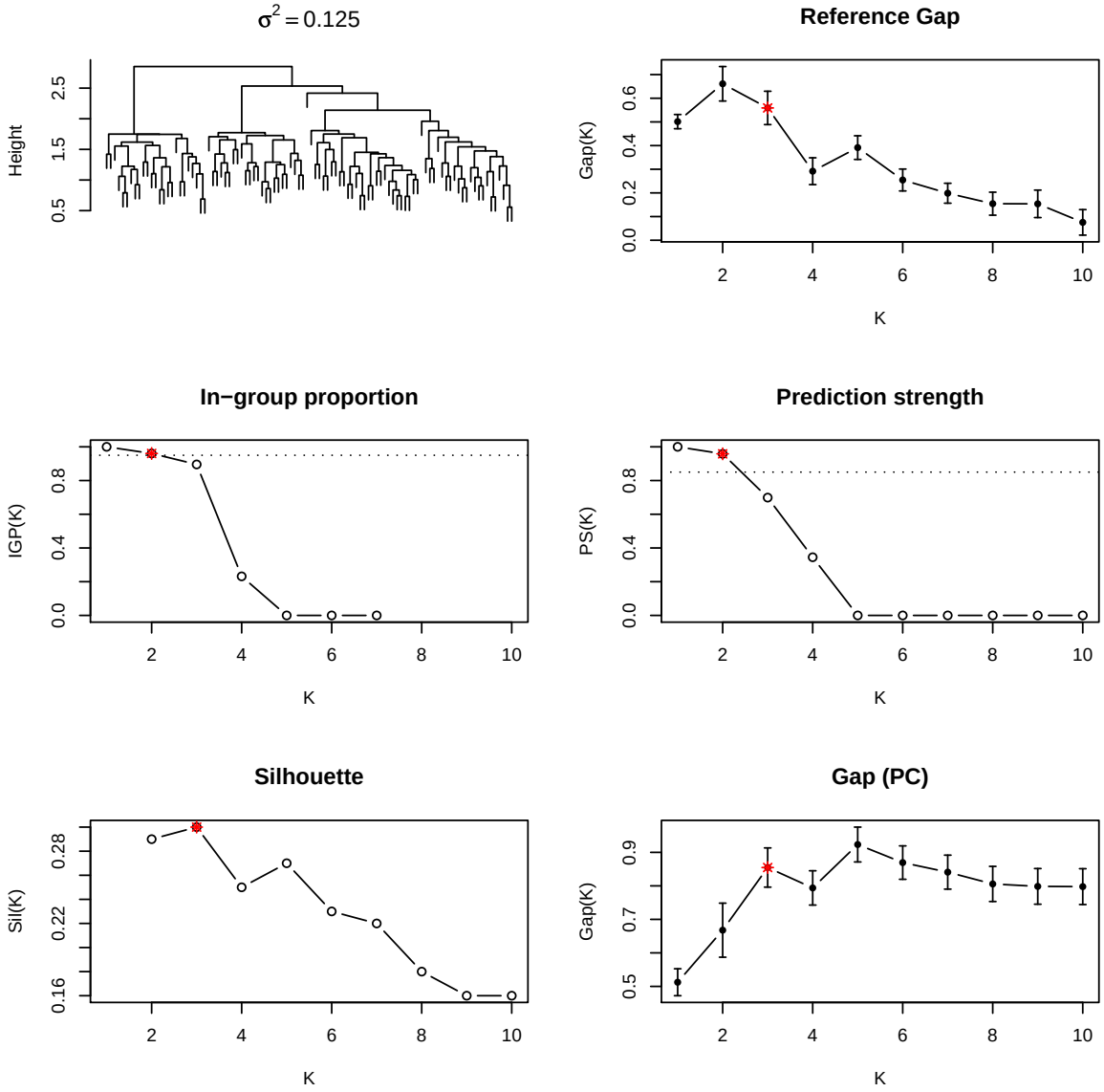


Figure 5.11: Another example that demonstrates some of ERA's virtues. The clustered data set is simulated from Scenario C10 ($\sigma^2 = 0.125$), and is made up of four clusters. A horizontal cut at $K = 4$ will not produce the four true clusters, because of one sample that is clustered on top of the two rightmost clusters. This prevents all the methods, except ERA, from finding the true number of clusters. ERA, on the other hand, recursively searches the three clusters found by Gap, discards the one unfitting sample due to the minimum cluster size requirement ($n_{min} = 5$), and hence locates the four true clusters.

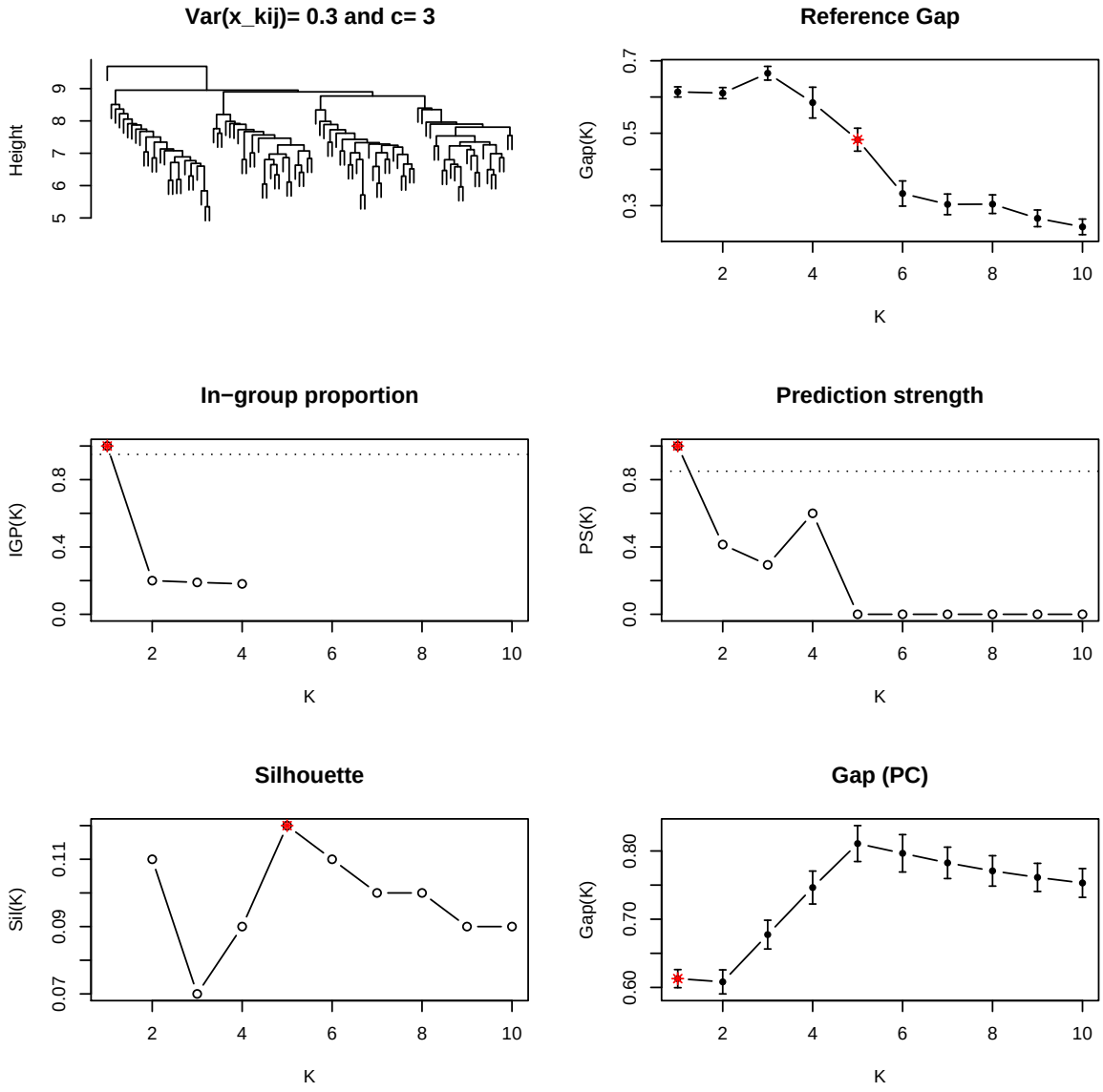


Figure 5.12: A final example where only ERA succeeds in finding the true number of clusters. The hierarchical clustering of a data set simulated from Scenario B100 with $\mathrm{Var}(x_{kij}) = 0.3$ and $c = 3$ is shown in the top left panel. The one outlier in the top of the dendrogram makes all the other methods fail. ERA, on the other hand, does not rely on a horizontal cut, and requires that all clusters must contain at least 5 samples. Therefore, the outlier is discarded and the four true clusters are thereafter found.

Chapter 6

Conclusions and discussion

6.1 Summary and conclusions

The main objective of this thesis was to describe and compare existing and novel methods for estimating the number of clusters in a data set. As described in Chapter 1, this is a highly relevant issue in cluster analysis, and is particularly needed for the high-dimensional data sets coming from microarray experiments. A basic introduction to cluster analysis was given in Chapter 2. This included the presentation of the two most common clustering methods, namely K-means clustering and hierarchical clustering. In Chapter 3, several methods for determining the number of clusters were described. Some of these are quite well-known and sometimes applied to real data sets, such as Silhouette and Gap. Others, such as In-group proportion, Prediction strength and Recursive Gap, are newer and not that familiar. In addition to these existing methods, two novel approaches were suggested. These were built upon the ideas of Gap and Recursive Gap, and were named Reference Gap and ERA, respectively.

All of the methods were applied to both real data sets (microarray data), and simulated data sets. In the microarray breast tumour data sets studied in Chapter 4, the true number of clusters are not really known. However, several studies of breast tumour data have indicated the existence of five (possibly six) subtypes of breast tumours. Hence there is some evidence that the optimal number of clusters in the Sørlie data set and the Micma data set presented in Chapter 4, is five (or possibly six). To the extent that this is true, ERA and Recursive Gap stood out as the most successful methods as they estimated five clusters in the Sørlie data set and six clusters in the Micma data set. Gap, Reference Gap and Silhouette all estimated two clusters in both data sets, which to some extent makes sense because the most significant differences in biology and clinical outcome are found between two major groups of breast tumours. The weakest performers for the microarray data sets were In-group proportion and Prediction strength, which did not find any cluster structure at all in neither of the data sets.

For the simulated data sets in Chapter 5, the novel method ERA was overall the most successful method in all of the scenarios. In fact, in many cases it performed far better than

the other methods. Recursive Gap also performed quite well, and was in the majority of the settings ranked the second best method. Gap was overall more successful than In-group proportion and Prediction strength. A finding was, however, that in the presence of sub-clusters, both of these methods, and in particular In-group proportion, did somewhat better than Gap. Another discovery was that Silhouette was very ineffective whenever some clusters were closer than others, especially in high dimension. When all of the clusters were more equally spaced, on the other hand, this method performed as well as or better than Gap. The other novel method introduced in this thesis was Reference Gap. This method is similar to Gap, but the different reference data are for different values of K. Reference Gap proved to be more successful than the original Gap for low-dimensional data sets. However, as the dimension increased, the performance of Reference Gap declined substantially. This does not necessarily mean that it is a bad idea to generate reference data in this alternative fashion. The criterion used in Reference Gap was, however, based on intuition and observations for low-dimensional data sets, and is perhaps not appropriate when the dimension is high. An enhancement of the criterion could thus maybe improve Reference Gap's performance in high dimensional data sets, and is an avenue for further work.

6.2 Discussion

The observation that ERA (and Recursive Gap) were the most successful methods when sub-clusters were introduced in Scenario D did not come as a great surprise since these methods were in part developed with such settings in mind. That ERA also excelled in the other scenarios was rather unanticipated, however. As mentioned in Chapter 5, ERA is much better equipped to handle the presence of outliers than the other methods, due to non-horizontal cutting of the dendrogram and a minimum cluster size requirement. If many of the simulated data sets contained outliers, it may explain why ERA was more successful than the other methods. To that end, a new set of simulations are designed to test the extent to which the presence of outliers may have affected the simulation results.

6.2.1 Effect of outliers in the data sets

The effect outliers have on the methods' performances was discussed in Section 5.5.2, and ERA is, as described, the only method for which the presence of outliers is not destructive. An outlier is in this respect a sample that is so different from the other samples that it does not really fit in any of the clusters, and that is hence clustered alone on top of the dendrogram. Examples of this is found in many of the dendrograms shown in Chapter 5, particularly for large variances.

It is a virtue of ERA that it is able to handle, and discard, outliers automatically. However, it would be interesting to get some idea of the extent to which ERA's success is a result of the

presence of outliers in the simulated data sets. On that account, the simulations in Scenarios A, B and C are repeated, but now with the requirement that no outliers are allowed on top of the dendrograms. Specifically, if a horizontal cut producing two clusters results in one cluster with less than five samples (the minimum cluster size requirement in ERA), the data set is rejected and a new data set is generated. This is repeated until a total of 100 accepted data sets have been produced for each setting. Scenario D is not repeated because outliers on the top of the dendrogram was not a problem there.

The methods' success percentages for these reruns of Scenario A, B and C without outliers, are listed in Table 6.1, 6.2, and 6.3. In parenthesis is the percentage increase in the number of correct estimates compared to the original results listed in Chapter 5 (Table 5.2, 5.3 and 5.5). The last column of each table shows the ranking of the methods, with the original ranking in parenthesis. Finally, the number of simulations it took to get a total of 100 approved data sets is listed in the last row of the tables. Note that mainly the largest variance values led to the rejection of some data sets.

In the rerun of Scenario A, only the three largest values of σ^2 in each dimension led to any significant changes. The performance of Silhouette is particularly improved in the 10-dimensional case, with an increase of 24, 38 and 32 % for the three largest σ^2, respectively. Reference Gap and Recursive Gap also do somewhat better for these σ^2, whereas the performances of the other methods are only moderately improved. Despite Silhouette's improvement, ERA still does quite a lot better for the two largest variances, however. In the rerun of Scenario A100, In-group proportion, Silhouette, Gap and especially Recursive Gap have increased success percentages, and the increase is particularly great for $\sigma^2 = 0.5$. Note that Recursive Gap and ERA do quite a lot better than the other methods for the larger variances, and that Recursive Gap is as effective as ERA for all but the the largest σ^2 (where ERA is almost twice as efficient). The greatest increase in success percentages in the rerun of Scenario A1000 is found for In-group proportion when $\sigma^2 = 1.6$ and $\sigma^2 = 1.8$, making In-group proportion the most successful method for these variances. The number of correct estimates is below 50 %, however, and In-group proportion's success rate is down to 47 % already at $\sigma^2 = 1.2$ (at which both Recursive Gap and ERA have perfect success rates). Silhouette and Recursive Gap also improve, and Recursive Gap is thus practically as successful as ERA in this setting. In fact, Recursive Gap is ranked above ERA, due to its minimally larger success for the two largest values of σ^2.

Scenario B (in addition to Scenario D) was the scenario in which ERA did particularly well compared to the other methods. The results from the rerun of this scenario, show that outliers on top of the dendrogram affected the results to a quite large extent. All of the methods increase their success rates when data sets with such outliers are rejected, but to a varying extent. In the rerun of Scenario B10, ERA is still the best performer overall, with almost perfect performance across all combinations of $\text{Var}(x_{kij})$ and c. The other methods follow much closer

Table 6.1: The percentage of times that each method returned the correct number of clusters ($K = 4$) for different values of σ^2 in a rerun of Scenario A ("4 clusters with equally distanced centroids"), where no outliers where allowed on the top of the dendrogram. In parenthesis is the percentage increase in number of correct estimates compared to the original results for Scenario A (Table 5.2). The last column of each table gives the ranking of the methods, with the original ranking of the method in parenthesis.

(a) Rerun of Scenario A10 (10 dimensions)

Method σ^2	**0.025**	**0.05**	**0.075**	**0.1**	**0.125**	**0.15**	**Rank**
ERA	100 (0)	100 (0)	100 (1)	96 (2)	84 (3)	70 (19)	1 (1)
Recursive Gap	97 (0)	83 (−8)	60 (0)	40 (10)	29 (20)	9 (5)	4 (4)
Gap	97 (0)	81 (−9)	55 (−2)	31 (12)	10 (4)	4 (4)	6 (6)
Silhouette	100 (0)	100 (0)	98 (4)	83 (24)	69 (38)	43 (32)	2 (2)
In-group proportion	99 (0)	67 (−7)	18 (2)	14 (2)	13 (11)	2 (1)	7 (7)
Prediction strength	100 (0)	92 (−3)	27 (−1)	2 (2)	0 (0)	0 (0)	5 (5)
Reference Gap	98 (0)	81 (−4)	76 (7)	53 (16)	50 (19)	28 (9)	3 (3)
# Simulations	100	100	102	123	201	333	

(b) Rerun of Scenario A100 (100 dimensions)

Method σ^2	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.7**	**Rank**
ERA	100 (0)	100 (0)	100 (0)	100 (0)	99 (0)	66 (18)	1 (1)
Recursive Gap	100 (0)	100 (0)	100 (2)	100 (26)	98 (82)	35 (34)	2 (2)
Gap	100 (0)	100 (0)	96 (−1)	85 (25)	64 (53)	7 (6)	4 (3)
Silhouette	100 (0)	100 (0)	96 (−1)	85 (25)	64 (53)	10 (9)	3 (3)
In-group proportion	100 (0)	100 (0)	47 (3)	47 (12)	44 (36)	0 (−1)	5 (5)
Prediction strength	100 (0)	100 (0)	75 (−6)	15 (10)	0 (−1)	0 (0)	6 (6)
Reference Gap	94 (0)	60 (0)	57 (1)	42 (−7)	35 (11)	6 (−5)	7 (7)
# Simulations	100	100	102	129	364	7401	

(c) Rerun of Scenario A1000 (1000 dimensions)

Method σ^2	**0.8**	**1.0**	**1.2**	**1.4**	**1.6**	**1.8**	**Rank**
ERA	100 (0)	100 (0)	100 (0)	92 (7)	23 (5)	1 (−1)	2 (1)
Recursive Gap	100 (0)	100 (0)	100 (8)	85 (17)	24 (16)	4 (0)	1 (2)
Gap	100 (0)	100 (1)	94 (5)	54 (10)	7 (3)	1 (1)	3 (4)
Silhouette	100 (0)	97 (−1)	30 (−3)	80 (20)	33 (20)	0 (−2)	5 (3)
In-group proportion	100 (1)	87 (15)	47 (5)	50 (0)	47 (25)	48 (38)	3 (5)
Prediction strength	100 (0)	93 (−1)	58 (−3)	23 (9)	2 (0)	0 (0)	6 (6)
Reference Gap	40 (−5)	29 (−4)	20 (−5)	16 (−5)	13 (−16)	13 (−15)	7 (6)
# Simulations	100	101	105	146	206	586	

behind than what was the case for the original results, however. This is in particular true for Silhouette, which does nearly as good as ERA up until the last combination of $\text{Var}(x_{kij})$ and c (where its success is down to 79 %). In the reruns of Scenario B100 and B1000, the success rates for In-group proportion, Silhouette and Gap all increase markedly for the three last combinations of $\text{Var}(x_{kij})$ and c. Silhouette and Gap now have success percentages of 85 % or more for all but the last combination, while In-group proportion does very well up until the last two combinations. Recursive Gap is the method that gains the most from removing data sets with outliers in Scenario B100 and B1000, however. In fact, Recursive Gap is now successful in virtually all of the simulations, and essentially performs as well as ERA.

Table 6.2: The percentage of times that each method returned the correct number of clusters ($K = 4$) for different values of σ^2 in a rerun of Scenario B ("4 clusters from contaminated normal distributions"), where no outliers are allowed on the top of the dendrogram. In parenthesis is the percentage increase in number of correct estimates compared to the original results for Scenario B (Table 5.3). The last column of each table gives the ranking of the methods, with the original ranking of the method in parenthesis.

(a) Rerun of Scenario B10 (10 dimensions)

Method / **Var(x_{kij})** / c	**0.03** **2**	**0.03** **3**	**0.05** **2**	**0.05** **3**	**0.07** **2**	**0.07** **3**	**Rank**
ERA	100(*0*)	100 (*0*)	99(−*1*)	100 (*0*)	100 (*0*)	100 (*1*)	1 (*1*)
Recursive Gap	95 (*0*)	93 (*17*)	81 (*0*)	78 (*43*)	60 (*12*)	67 (*53*)	4 (*4*)
Gap	96 (*0*)	91 (*17*)	79 (*4*)	71 (*40*)	61 (*13*)	50 (*39*)	6 (*7*)
Silhouette	100(*0*)	100(*11*)	97 (*3*)	88 (*44*)	94 (*28*)	79 (*65*)	2 (*2*)
In-group proportion	98 (*0*)	94 (*8*)	61(−*4*)	68 (*27*)	43 (*20*)	48 (*40*)	7 (*5*)
Prediction strength	100(*0*)	94 (*15*)	80(−*2*)	49 (*27*)	26 (*2*)	22 (*20*)	5 (*5*)
Reference Gap	95 (*0*)	97 (*14*)	88 (*7*)	82 (*41*)	69 (*16*)	63 (*47*)	3 (*3*)
# Simulations	100	115	106	219	131	548	

(b) Rerun of Scenario B100 (100 dimensions)

Method / **Var(x_{kij})** / c	**0.1** **2**	**0.1** **3**	**0.2** **2**	**0.2** **3**	**0.3** **2**	**0.3** **3**	**Rank**
ERA	100(*0*)	100(*0*)	100(*0*)	100 (*0*)	100 (*0*)	100 (*0*)	1 (*1*)
Recursive Gap	100(*0*)	100(*1*)	100(*1*)	100(*39*)	100(*30*)	99 (*93*)	2 (*2*)
Gap	100(*0*)	100(*1*)	100(*1*)	88 (*33*)	93 (*29*)	51 (*47*)	4 (*3*)
Silhouette	100(*0*)	100(*1*)	100(*1*)	88 (*33*)	93 (*29*)	52 (*48*)	3 (*3*)
In-group proportion	100(*0*)	100(*1*)	99 (*1*)	86 (*32*)	54 (*14*)	48 (*44*)	5 (*5*)
Prediction strength	100(*0*)	100(*1*)	96 (*2*)	48 (*17*)	48 (*14*)	8 (*8*)	6 (*6*)
Reference Gap	94 (*0*)	94 (*1*)	69 (*1*)	54 (*10*)	50 (*3*)	40 (*29*)	7 (*7*)
# Simulations	100	100	101	138	114	1706	

(c) Rerun of Scenario B1000 (1000 dimensions)

Method / **Var(x_{kij})** / c	**0.5** **2**	**0.5** **3**	**0.75** **2**	**0.75** **3**	**1.0** **2**	**1.0** **3**	**Rank**
ERA	100(*0*)	100 (*0*)	100(*0*)	100 (*0*)	100 (*0*)	100 (*3*)	1 (*1*)
Recursive Gap	100(*0*)	100 (*2*)	100(*0*)	100(*25*)	100(*10*)	100(*83*)	1 (*2*)
Gap	100(*0*)	98 (*1*)	99 (*0*)	85 (*19*)	94 (*7*)	65 (*57*)	3 (*3*)
Silhouette	100(*0*)	98 (*1*)	99 (*0*)	85 (*19*)	93 (*6*)	65 (*57*)	4 (*3*)
In-group proportion	100(*0*)	98 (*1*)	97 (*0*)	81 (*15*)	71 (*1*)	57 (*49*)	5 (*5*)
Prediction strength	100(*0*)	98 (*0*)	96 (*0*)	57 (*19*)	62 (*12*)	9 (*9*)	6 (*6*)
Reference Gap	76 (*0*)	59 (−*1*)	56 (*0*)	40 (*0*)	26 (*6*)	28 (*7*)	7 (*7*)
# Simulations	100	101	100	144	108	679	

For the most part there are only slight differences between the original results for Scenario C and the results from the rerun. In the 10-dimensional setting there are only small improvements in the performances of Silhouette, Gap and Recursive Gap for the largest variances, and ERA is still by far the better method. In 100 dimensions, only Recursive Gap has a discernible increase in success for the two largest values of σ^2. ERA still performs considerably better for these σ^2, however. In the 1000-dimensional case, outliers on top of the dendrogram seem to have influenced the original results to a very small extent, as only a few data sets were rejected

Table 6.3: The percentage of times that each method returned the correct number of clusters ($K = 4$) for different values of σ^2 in a rerun of Scenario C ("4 clusters with unequally distanced centroids"), where no outliers where allowed on the top of the dendrogram. In parenthesis is the percentage increase in number of correct estimates compared to the original results for Scenario C (Table 5.5).The last column of each table gives the ranking of the methods, with the original ranking of the method in parenthesis.

(a) Rerun of Scenario C10 (10 dimensions)

Method σ^2	**0.05**	**0.1**	**0.125**	**0.15**	**0.2**	**0.25**	**Rank**
ERA	100 (*0*)	99 (*−1*)	99 (*−1*)	97 (*−1*)	82 (*−3*)	62 (*−2*)	1 (*1*)
Recursive Gap	100 (*0*)	100 (*0*)	96 (*5*)	84 (*5*)	46 (*2*)	21 (*6*)	2 (*2*)
Gap	100 (*0*)	99 (*0*)	94 (*7*)	76 (*8*)	23 (*0*)	10 (*5*)	3 (*4*)
Silhouette	100 (*0*)	93 (*3*)	71 (*5*)	50 (*4*)	19 (*−1*)	17 (*6*)	5 (*5*)
In-group proportion	99 (*0*)	50 (*−15*)	34 (*4*)	14 (*−4*)	6 (*−2*)	2 (*−1*)	7 (*7*)
Prediction strength	100 (*0*)	81 (*−3*)	45 (*−3*)	13 (*1*)	1 (*0*)	0 (*0*)	6 (*6*)
Reference Gap	100 (*0*)	89 (*−2*)	75 (*9*)	56 (*1*)	25 (*−11*)	26 (*−2*)	4 (*3*)
# Simulations	100	101	101	106	121	157	

(b) Rerun of Scenario C100 (100 dimensions)

Method σ^2	**0.2**	**0.4**	**0.5**	**0.6**	**0.8**	**1.0**	**Rank**
ERA	100 (*0*)	100 (*0*)	100 (*0*)	100 (*3*)	92 (*−2*)	73 (*3*)	1 (*1*)
Recursive Gap	100 (*0*)	100 (*0*)	94 (*−3*)	84 (*4*)	47 (*14*)	31 (*26*)	2 (*2*)
Gap	100 (*0*)	100 (*0*)	89 (*−2*)	63 (*−2*)	14 (*6*)	2 (*2*)	3 (*3*)
Silhouette	95 (*0*)	22 (*0*)	12 (*3*)	3 (*−9*)	0 (*−2*)	0 (*0*)	7 (*7*)
In-group proportion	100 (*0*)	91 (*0*)	63 (*0*)	46 (*2*)	19 (*9*)	4 (*−1*)	4 (*4*)
Prediction strength	100 (*0*)	96 (*0*)	68 (*−4*)	33 (*−2*)	2 (*2*)	0 (*0*)	5 (*5*)
Reference Gap	96 (*0*)	71 (*0*)	30 (*−12*)	33 (*4*)	26 (*3*)	22 (*1*)	6 (*6*)
# Simulations	100	100	100	102	120	211	

(c) Rerun of Scenario C1000 (1000 dimensions)

Method σ^2	**1.0**	**1.4**	**1.6**	**1.8**	**2.0**	**2.2**	**Rank**
ERA	100 (*0*)	100 (*0*)	100 (*0*)	98 (*0*)	78 (*−2*)	31 (*6*)	1 (*1*)
Recursive Gap	100 (*0*)	100 (*0*)	98 (*1*)	93 (*1*)	64 (*−2*)	31 (*4*)	2 (*2*)
Gap	100 (*0*)	100 (*0*)	97 (*1*)	80 (*4*)	46 (*2*)	12 (*2*)	4 (*3*)
Silhouette	0 (*0*)	0 (*0*)	0 (*0*)	1 (*0*)	0 (*0*)	0 (*−1*)	7 (*7*)
In-group proportion	100 (*0*)	96 (*0*)	85 (*1*)	67 (*4*)	59 (*1*)	32 (*−1*)	3 (*3*)
Prediction strength	100 (*0*)	95 (*0*)	84 (*0*)	67 (*−1*)	32 (*−5*)	20 (*−1*)	5 (*5*)
Reference Gap	40 (*0*)	53 (*0*)	54 (*1*)	39 (*0*)	30 (*−5*)	15 (*−4*)	6 (*6*)
# Simulations	100	100	101	101	101	102	

for all values of σ^2. Hence there are only small, random changes in the results, and ERA does as well as or somewhat better than the other methods for all but one value of σ^2.

The conclusion of the reruns where only data sets without outliers on top of the dendrogram were considered, is that such outliers have affected the results in Scenario A and B to a considerable extent. The success rates of most of the methods were noticeably increased for the largest variances, and this made the success of ERA less pronounced. The relative ranking of the methods, remained more or less the same, however, and ERA still proved to be the best method overall. In fact, in several settings ERA still performed far better than most of the other methods for large values of σ^2. It could only be truly challenged by it's predecessor, Recursive

Gap, in the reruns of Scenario A1000, B100 and B1000. To that end, the recursive methods, and in particular the enhanced, novel method ERA, stand out as valuable contributions to the challenging task of estimating the number of clusters in data sets.

6.2.2 What is a cluster really?

A final issue one should keep in mind when considering the simulation results is that the concept of a cluster is not a well-defined one. That is, it is not clear how cohesive, and separated from the other samples, a set of samples have to be to be characterized as a cluster. This makes it difficult to say how many clusters there truly are in a data set, even in simulations. In the simulations described in Chapter 5, clusters were constructed by generating samples from different distributions, and samples generated from the same distribution were taken to belong to the same cluster. However, as the variance increases, samples from these different clusters get more proximate, and the clusters are more likely to overlap. Hence for the largest variances, it is hard to tell whether the data sets in fact did consist of the K^* clusters we assumed. When judging the methods' performances we can therefore only say something about their abilities to discover the assumed K^* clusters coming from K^* different distributions. Whether this in fact corresponds to the actual "true" number of clusters in the data sets remains an open discussion.

6.3 Topics for future research

The results found for real and simulated data sets in this thesis indicate that ERA provides a valuable approach to estimating the number of clusters. Further testing of ERA is, however, needed to get a deeper understanding of its strengths and weaknesses.

In the simulations described in Chapter 5, the effect of some chosen factors on the performance of the methods were studied. These factors included the relative distances between the cluster means, the cluster variance, the distribution (heavy-tailed or not), the number of features and the presence/absence of sub-clusters. The simulations were limited to include these factors due to time and space limitations, but it would be interesting to study the effect of several other factors as well. Such factors could, among others, include the number of samples in each cluster, the number of true clusters, and the shape of the clusters. Only spherical clusters simulated from normal (or contaminated normal) distributions were considered in this thesis. Other cluster shapes may make it more challenging for the methods to find the number of clusters, and it would be particularly interesting to see how ERA handles such cases. One possibility is, for example, to generate elongated clusters by introducing correlation between the features.

Another issue is that in all of the simulations and for both of the microarray data sets, only hierarchical clustering was applied. The reason for this is that hierarchical clustering

is usually the preferred method in microarray studies. In future studies it would still be interesting to compare the methods' performances when other clustering methods, such as K-means clustering, is applied. Note that for the recursive methods, a certain hierarchical arrangement of the clusters will be inferred when applying K-means clustering as well, due to the very definition of these methods. That is, since the recursive runs are made on the clusters found in the first run, the sub-clusters found in the recursive runs must necessarily be nested within these.

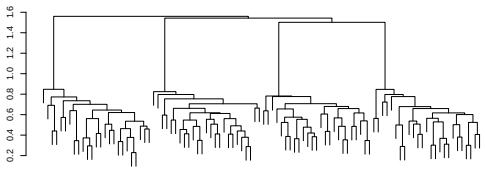
A notable aspect of ERA is that it provides a very general and thorough framework for estimating the number of clusters in hierarchical nested partitions of data sets. In this thesis, ERA was applied in combination with the Gap method, and the Gap algorithm was used to decide whether a cluster could be significantly divided into sub-clusters. However, ERA does not necessarily have to use Gap for this purpose, and it could be specified to collaborate with any of the other methods described here instead. Hence a very interesting topic for future work would be to test the performance of ERA when applied with other methods such as Silhouette, In-group proportion or Prediction strength. Also, the suggested procedure for locating an appropriate stopping threshold in ERA is rather complex and cumbersome, and also requires that the user of the method decides upon a number thresholds to test. An effort to develop a more objective and easily understood technique for selecting a stopping threshold may improve ERA further.

Finally, there is also a need for evaluation of the clusters found by the methods. This topic is referred to as cluster validation, where the intention is to get an objective evaluation of the quality of the suggested clusters. Several methods have been suggested for this purpose (see for example Everitt et al. (2001, chapter 8) for a basic overview of some of these).

Appendix A

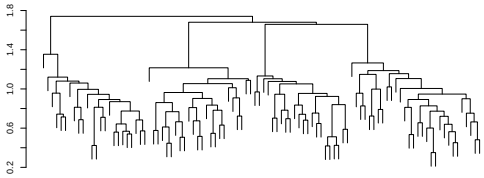
Additional dendrograms

Figures A.1 - A.6 show dendrograms resulting from hierarchical clustering of data sets generated from Scenario A10, A1000, B10, B1000, C10 and C1000 in Chapter 5, respectively.



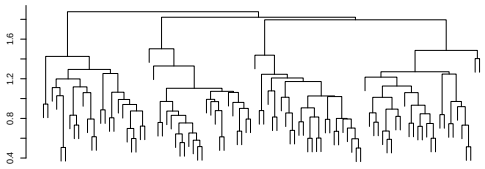
(a) $\boldsymbol{\sigma^2 = 0.025}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



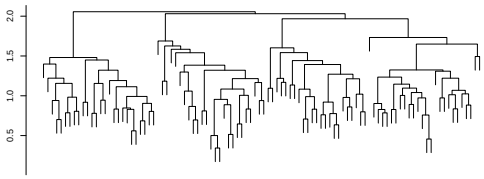
(b) $\boldsymbol{\sigma^2 = 0.05}$

Reference Gap:	4
In-group proportion:	2
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



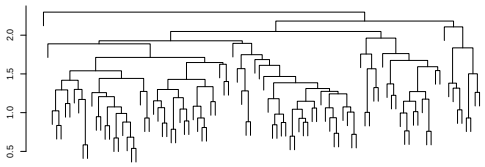
(c) $\boldsymbol{\sigma^2 = 0.075}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



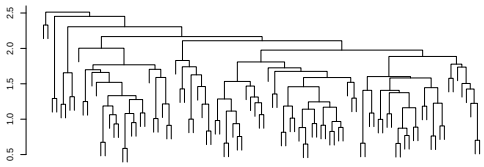
(d) $\boldsymbol{\sigma^2 = 0.1}$

Reference Gap:	4
In-group proportion:	1
Prediction strength:	1
Silhouette:	4
Gap:	1
Recursive Gap:	1
ERA:	4



(e) $\boldsymbol{\sigma^2 = 0.125}$

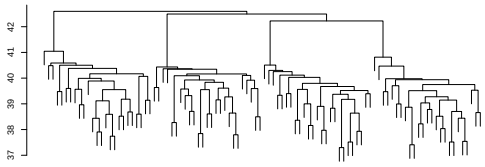
Reference Gap:	7
In-group proportion:	1
Prediction strength:	1
Silhouette:	2
Gap:	1
Recursive Gap:	1
ERA:	4



(f) $\boldsymbol{\sigma^2 = 0.15}$

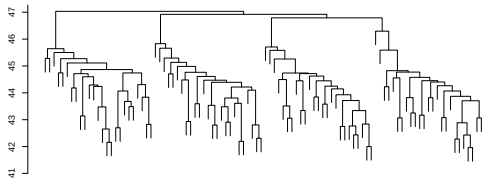
Reference Gap:	2
In-group proportion:	1
Prediction strength:	1
Silhouette:	2
Gap:	1
Recursive Gap:	1
ERA:	1

Figure A.1: Dendrograms resulting from hierarchical clustering of random data sets simulated for different values of σ^2 in Scenario A10 ("4 clusters with equally distanced cluster means - 10 dimensions"). Below each dendrogram are the estimated number of clusters in the data sets given by the various methods.



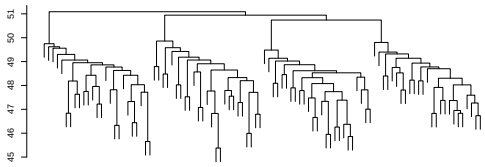
(a) $\boldsymbol{\sigma^2 = 0.8}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



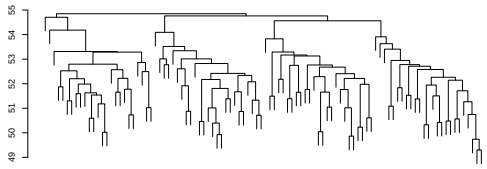
(b) $\boldsymbol{\sigma^2 = 1.0}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



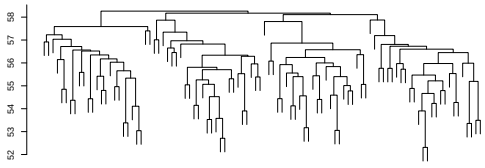
(c) $\boldsymbol{\sigma^2 = 1.2}$

Reference Gap:	2
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



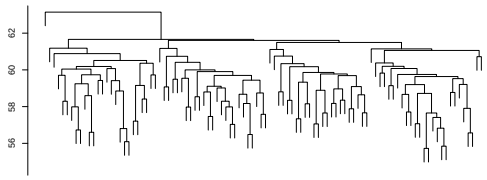
(d) $\boldsymbol{\sigma^2 = 1.4}$

Reference Gap:	3
In-group proportion:	3
Prediction strength:	1
Silhouette:	5
Gap:	3
Recursive Gap:	4
ERA:	4



(e) $\boldsymbol{\sigma^2 = 1.6}$

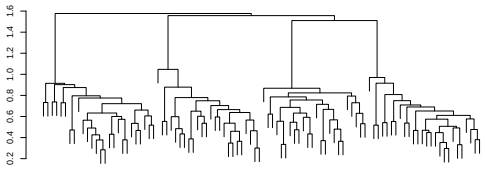
Reference Gap:	2
In-group proportion:	4
Prediction strength:	1
Silhouette:	4
Gap:	4
Recursive Gap:	3
ERA:	4



(f) $\boldsymbol{\sigma^2 = 1.8}$

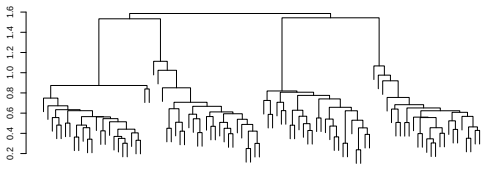
Reference Gap:	3
In-group proportion:	1
Prediction strength:	1
Silhouette:	2
Gap:	1
Recursive Gap:	1
ERA:	1

Figure A.2: Dendrograms resulting from hierarchical clustering of random data sets simulated for different values of σ^2 in Scenario A1000 ("4 clusters with equally distanced cluster means - 1000 dimensions"). Below each dendrogram are the estimated number of clusters in the data sets given by the various methods.



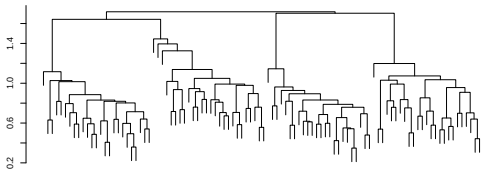
(a) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.03}, \boldsymbol{c} = \mathbf{2}$

Reference Gap:	4
In-group proportion:	3
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



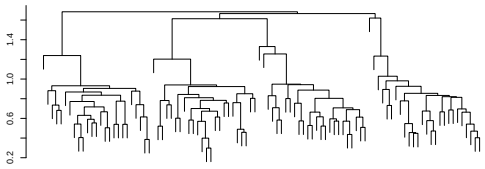
(b) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.03}, \boldsymbol{c} = \mathbf{3}$

Reference Gap:	2
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



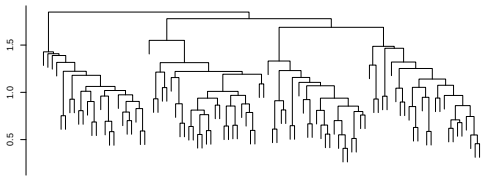
(c) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.05}, \boldsymbol{c} = \mathbf{2}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



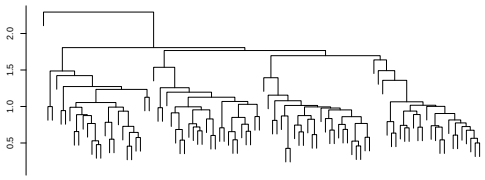
(d) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.05}, \boldsymbol{c} = \mathbf{3}$

Reference Gap:	3
In-group proportion:	3
Prediction strength:	1
Silhouette:	5
Gap:	3
Recursive Gap:	4
ERA:	4



(e) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.07}, \boldsymbol{c} = \mathbf{2}$

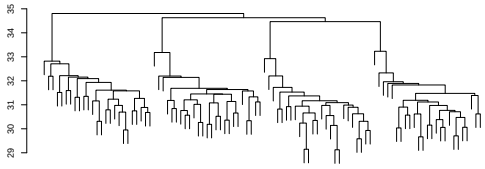
Reference Gap:	4
In-group proportion:	1
Prediction strength:	1
Silhouette:	4
Gap:	1
Recursive Gap:	1
ERA:	4



(f) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.07}, \boldsymbol{c} = \mathbf{3}$

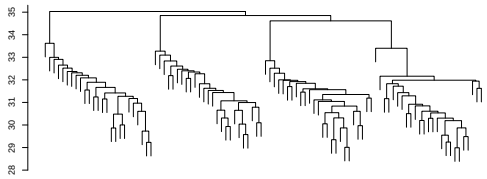
Reference Gap:	1
In-group proportion:	1
Prediction strength:	1
Silhouette:	5
Gap:	1
Recursive Gap:	1
ERA:	4

Figure A.3: Dendrograms resulting from hierarchical clustering of random data sets simulated for different values of σ^2 in Scenario B10 ("4 clusters from contaminated normal distributions - 10 dimensions"). Below each dendrogram are the estimated number of clusters in the data sets given by the various methods.



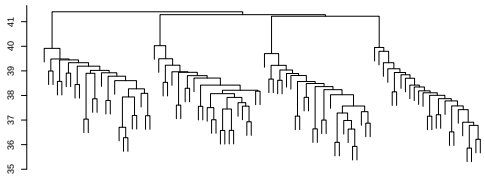
(a) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.5}, \boldsymbol{c} = \mathbf{2}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



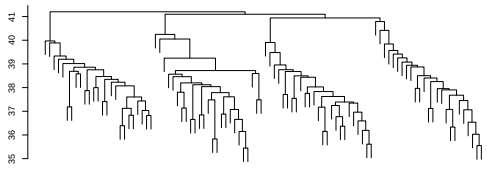
(b) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.5}, \boldsymbol{c} = \mathbf{3}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



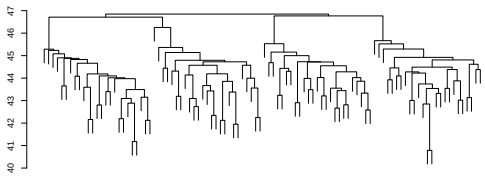
(c) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.75}, \boldsymbol{c} = \mathbf{2}$

Reference Gap:	3
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



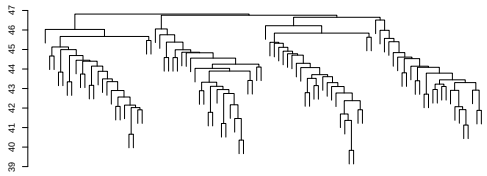
(d) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{0.75}, \boldsymbol{c} = \mathbf{3}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	1
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



(e) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{1.0}, \boldsymbol{c} = \mathbf{2}$

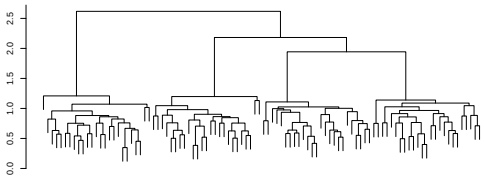
Reference Gap:	2
In-group proportion:	3
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



(f) $\mathbf{Var}(\boldsymbol{x_{kij}}) = \mathbf{1.0}, \boldsymbol{c} = \mathbf{3}$

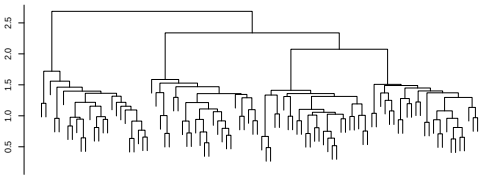
Reference Gap:	4
In-group proportion:	4
Prediction strength:	1
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4

Figure A.4: Dendrograms resulting from hierarchical clustering of random data sets simulated for different values of σ^2 in Scenario B1000 ("4 clusters from contaminated normal distributions - 1000 dimensions"). Below each dendrogram are the estimated number of clusters in the data sets given by the various methods.



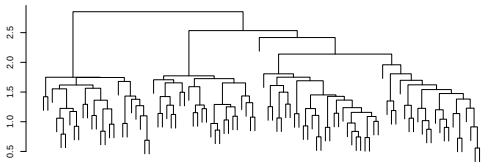
(a) $\boldsymbol{\sigma^2 = 0.05}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



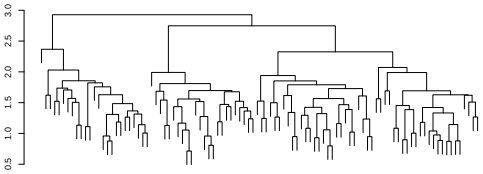
(b) $\boldsymbol{\sigma^2 = 0.1}$

Reference Gap:	4
In-group proportion:	3
Prediction strength:	4
Silhouette:	4
Gap:	4
Recursive Gap:	4
ERA:	4



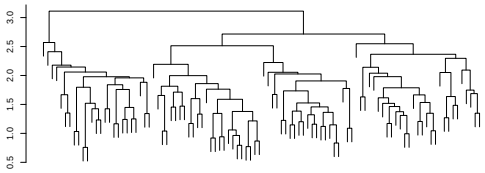
(c) $\boldsymbol{\sigma^2 = 0.125}$

Reference Gap:	3
In-group proportion:	2
Prediction strength:	2
Silhouette:	3
Gap:	3
Recursive Gap:	3
ERA:	4



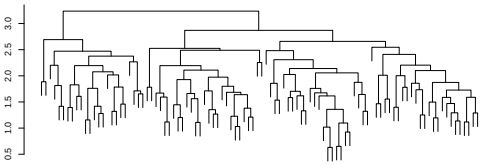
(d) $\boldsymbol{\sigma^2 = 0.15}$

Reference Gap:	3
In-group proportion:	3
Prediction strength:	4
Silhouette:	3
Gap:	3
Recursive Gap:	4
ERA:	4



(e) $\boldsymbol{\sigma^2 = 0.2}$

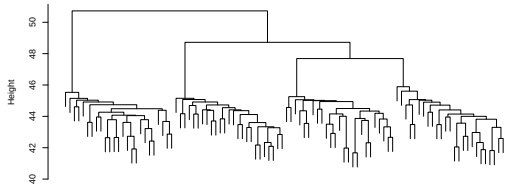
Reference Gap:	3
In-group proportion:	2
Prediction strength:	2
Silhouette:	2
Gap:	3
Recursive Gap:	4
ERA:	4



(f) $\boldsymbol{\sigma^2 = 0.25}$

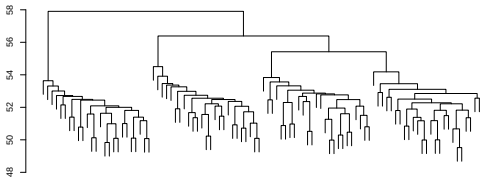
Reference Gap:	3
In-group proportion:	1
Prediction strength:	2
Silhouette:	2
Gap:	3
Recursive Gap:	4
ERA:	4

Figure A.5: Dendrograms resulting from hierarchical clustering of random data sets simulated for different values of σ^2 in Scenario C10 ("4 clusters with unequally distanced cluster means - 10 dimensions"). Below each dendrogram are the estimated number of clusters in the data sets given by the various methods.



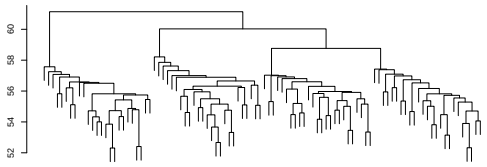
(a) $\boldsymbol{\sigma^2 = 1.0}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	2
Gap:	4
Recursive Gap:	4
ERA:	4



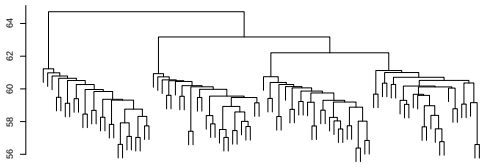
(b) $\boldsymbol{\sigma^2 = 1.4}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	2
Gap:	4
Recursive Gap:	4
ERA:	4



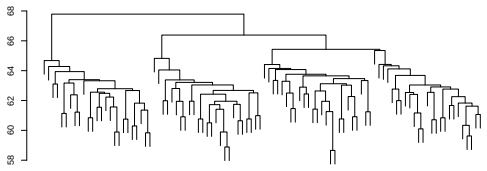
(c) $\boldsymbol{\sigma^2 = 1.6}$

Reference Gap:	3
In-group proportion:	3
Prediction strength:	4
Silhouette:	2
Gap:	4
Recursive Gap:	4
ERA:	4



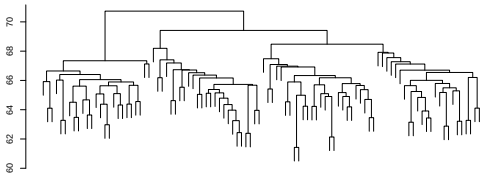
(d) $\boldsymbol{\sigma^2 = 1.8}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	4
Silhouette:	2
Gap:	4
Recursive Gap:	4
ERA:	4



(e) $\boldsymbol{\sigma^2 = 2.0}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	2
Silhouette:	2
Gap:	3
Recursive Gap:	3
ERA:	3



(f) $\boldsymbol{\sigma^2 = 2.2}$

Reference Gap:	4
In-group proportion:	4
Prediction strength:	3
Silhouette:	2
Gap:	3
Recursive Gap:	3
ERA:	3

Figure A.6: Dendrograms resulting from hierarchical clustering of random data sets simulated for different values of σ^2 in Scenario C1000 ("4 clusters with unequally distanced cluster means - 1000 dimensions"). Below each dendrogram are the estimated number of clusters in the data sets given by the various methods.

Appendix B

Detailed simulation results

Tables B.1 - B.11 give the detailed results from the simulation scenarios in Chapter 5.

Table B.1: The tables report the number of times (out of 100) that a method estimates a given number of clusters for the various values of σ^2 in Scenario A10. The number of times a method estimated the correct number of clusters ($K = 4$) is in bold.

(a) $\sigma^2 = 0.025$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	3	0	0	**97**	0	0	0	0	0	0
Gap	3	0	0	**97**	0	0	0	0	0	0
Silhouette	0	0	0	**100**	0	0	0	0	0	0
In-group proportion	0	1	0	**99**	0	0	0	0	0	0
Prediction strength	0	0	0	**100**	0	0	0	0	0	0
Reference Gap	0	2	0	**98**	0	0	0	3	0	0

(b) $\sigma^2 = 0.05$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	9	0	0	**91**	0	0	0	0	0	0
Gap	10	0	0	**90**	0	0	0	0	0	0
Silhouette	0	0	0	**100**	0	0	0	0	0	0
In-group proportion	9	6	11	**74**	0	0	0	0	0	0
Prediction strength	4	1	0	**95**	0	0	0	0	0	0
Reference Gap	1	11	0	**85**	0	0	0	3	0	0

(c) $\sigma^2 = 0.075$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	1	0	0	**99**	0	0	0	0	0	0
Recursive Gap	40	0	0	**60**	0	0	0	0	0	0
Gap	40	1	2	**57**	0	0	0	0	0	0
Silhouette	0	1	0	**94**	5	0	0	0	0	0
In-group proportion	57	17	10	**16**	0	0	0	0	0	0
Prediction strength	70	2	0	**28**	0	0	0	0	0	0
Reference Gap	3	15	8	**69**	2	1	2	0	0	0

(d) $\sigma^2 = 0.1$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	1	0	4	**94**	1	0	0	0	0	0
Recursive Gap	66	1	3	**30**	0	0	0	0	0	0
Gap	71	4	6	**19**	0	0	0	0	0	0
Silhouette	0	12	3	**59**	22	2	2	0	0	0
In-group proportion	80	4	4	**12**	0	0	0	0	0	0
Prediction strength	97	2	1	**0**	0	0	0	0	0	0
Reference Gap	9	18	23	**37**	9	2	1	1	0	0

(e) $\sigma^2 = 0.125$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	3	5	9	**81**	2	0	0	0	0	0
Recursive Gap	84	3	4	**9**	0	0	0	0	0	0
Gap	84	7	3	**6**	0	0	0	0	0	0
Silhouette	0	44	6	**31**	14	4	1	0	0	0
In-group proportion	93	2	3	**2**	0	0	0	0	0	0
Prediction strength	100	0	0	**0**	0	0	0	0	0	0
Reference Gap	8	15	25	**31**	12	7	2	0	0	0

(f) $\sigma^2 = 0.15$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	23	4	15	**51**	6	1	0	0	0	0
Recursive Gap	94	2	0	**4**	0	0	0	0	0	0
Gap	94	5	1	**0**	0	0	0	0	0	0
Silhouette	0	74	4	**11**	8	1	1	0	1	0
In-group proportion	98	2	0	**0**	0	0	0	0	0	0
Prediction strength	100	0	0	**0**	0	0	0	0	0	0
Reference Gap	9	11	12	**19**	27	10	4	5	3	0

Table B.2: The tables report the number of times (out of 100) that a method estimates a given number of clusters for the various values of σ^2 in Scenario A100. The number of times a method estimated the correct number of clusters ($K = 4$) is in bold.

(a) $\sigma^2 = 0.1$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	0	0	**100**	0	0	0	0	0	0
Silhouette	0	0	0	**100**	0	0	0	0	0	0
In-group proportion	0	0	0	**100**	0	0	0	0	0	0
Prediction strength	0	0	0	**100**	0	0	0	0	0	0
Reference Gap	0	6	0	**94**	0	0	0	0	0	0

(b) $\sigma^2 = 0.2$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	0	0	**100**	0	0	0	0	0	0
Silhouette	0	0	0	**100**	0	0	0	0	0	0
In-group proportion	0	0	0	**100**	0	0	0	0	0	0
Prediction strength	0	0	0	**100**	0	0	0	0	0	0
Reference Gap	0	17	23	**60**	0	0	0	0	0	0

(c) $\sigma^2 = 0.3$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	2	0	0	**98**	0	0	0	0	0	0
Gap	2	0	1	**97**	0	0	0	0	0	0
Silhouette	0	0	0	**97**	3	0	0	0	0	0
In-group proportion	23	25	8	**44**	0	0	0	0	0	0
Prediction strength	19	0	0	**81**	0	0	0	0	0	0
Reference Gap	0	19	23	**56**	2	0	0	0	0	0

(d) $\sigma^2 = 0.4$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	26	0	0	**74**	0	0	0	0	0	0
Gap	26	2	12	**60**	0	0	0	0	0	0
Silhouette	0	18	1	**60**	16	5	0	0	0	0
In-group proportion	45	4	16	**35**	0	0	0	0	0	0
Prediction strength	91	3	1	**5**	0	0	0	0	0	0
Reference Gap	0	11	24	**49**	12	4	0	0	0	0

(e) $\sigma^2 = 0.5$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	1	**99**	0	0	0	0	0	0
Recursive Gap	82	0	2	**16**	0	0	0	0	0	0
Gap	82	2	5	**11**	0	0	0	0	0	0
Silhouette	0	78	3	**11**	4	3	1	0	0	0
In-group proportion	85	3	4	**8**	0	0	0	0	0	0
Prediction strength	99	0	0	**1**	0	0	0	0	0	0
Reference Gap	0	4	17	**24**	21	21	4	6	3	0

(f) $\sigma^2 = 0.7$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	30	3	17	**48**	2	0	0	0	0	0
Recursive Gap	99	0	0	**1**	0	0	0	0	0	0
Gap	99	0	0	**1**	0	0	0	0	0	0
Silhouette	0	99	0	**1**	0	0	0	0	0	0
In-group proportion	99	0	0	**1**	0	0	0	0	0	0
Prediction strength	100	0	0	**0**	0	0	0	0	0	0
Reference Gap	0	3	4	**11**	6	18	18	20	20	0

Table B.3: The tables report the number of times (out of 100) that a method estimates a given number of clusters for the various values of σ^2 in Scenario A1000. The number of times a method estimated the correct number of clusters ($K = 4$) is in bold.

(a) $\sigma^2 = 0.8$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	0	0	**100**	0	0	0	0	0	0
Silhouette	0	0	0	**100**	0	0	0	0	0	0
In-group proportion	0	0	1	**99**	0	0	0	0	0	0
Prediction strength	0	0	0	**100**	0	0	0	0	0	0
Reference Gap	0	39	16	**45**	0	0	0	0	0	0

(b) $\sigma^2 = 1.0$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	1	0	**99**	0	0	0	0	0	0
Silhouette	0	0	1	**98**	1	0	0	0	0	0
In-group proportion	9	9	10	**72**	0	0	0	0	0	0
Prediction strength	6	0	0	**94**	0	0	0	0	0	0
Reference Gap	0	47	20	**33**	0	0	0	0	0	0

(c) $\sigma^2 = 1.2$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	8	0	0	**92**	0	0	0	0	0	0
Gap	8	0	3	**89**	0	0	0	0	0	0
Silhouette	0	7	59	**33**	0	1	0	0	0	0
In-group proportion	38	15	5	**42**	0	0	0	0	0	0
Prediction strength	38	1	0	**61**	0	0	0	0	0	0
Reference Gap	0	44	30	**25**	1	0	0	0	0	0

(d) $\sigma^2 = 1.4$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	1	14	**85**	0	0	0	0	0	0
Recursive Gap	23	0	9	**68**	0	0	0	0	0	0
Gap	23	6	27	**44**	0	0	0	0	0	0
Silhouette	0	25	4	**60**	11	0	0	0	0	0
In-group proportion	37	2	11	**50**	0	0	0	0	0	0
Prediction strength	86	0	0	**14**	0	0	0	0	0	0
Reference Gap	0	35	39	**21**	5	0	0	0	0	0

(e) $\sigma^2 = 1.6$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	15	15	52	**18**	0	0	0	0	0	0
Recursive Gap	68	5	19	**8**	0	0	0	0	0	0
Gap	65	14	17	**4**	0	0	0	0	0	0
Silhouette	0	81	6	**13**	0	0	0	0	0	0
In-group proportion	64	5	9	**22**	0	0	0	0	0	0
Prediction strength	94	4	0	**2**	0	0	0	0	0	0
Reference Gap	0	20	29	**29**	12	6	2	2	0	0

(f) $\sigma^2 = 1.8$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	65	20	13	**2**	0	0	0	0	0	0
Recursive Gap	90	5	5	**0**	0	0	0	0	0	0
Gap	90	9	1	**0**	0	0	0	0	0	0
Silhouette	0	86	12	**2**	0	0	0	0	0	0
In-group proportion	84	2	4	**10**	0	0	0	0	0	0
Prediction strength	97	2	1	**0**	0	0	0	0	0	0
Reference Gap	0	9	12	**28**	27	14	3	6	1	0

Table B.4: The tables report the number of times (out of 100) that a method estimates a given number of clusters for the various values of σ^2 in Scenario B10. The number of times a method estimated the correct number of clusters ($K = 4$) is in bold.

(a) $\text{Var}(x_{kij}) = 0.03$, $c = 2$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	5	0	0	**95**	0	0	0	0	0	0
Gap	4	0	0	**96**	0	0	0	0	0	0
Silhouette	0	0	0	**100**	0	0	0	0	0	0
In-group proportion	1	1	0	**98**	0	0	0	0	0	0
Prediction strength	0	0	0	**100**	0	0	0	0	0	0
Reference Gap	1	4	0	**95**	0	0	0	0	0	0

(b) $\text{Var}(x_{kij}) = 0.03$, $c = 3$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	24	0	0	**76**	0	0	0	0	0	0
Gap	25	0	1	**74**	0	0	0	0	0	0
Silhouette	0	0	0	**89**	11	0	0	0	0	0
In-group proportion	9	1	4	**86**	0	0	0	0	0	0
Prediction strength	21	0	0	**79**	0	0	0	0	0	0
Reference Gap	0	4	2	**83**	10	0	1	0	0	0

(c) $\text{Var}(x_{kij}) = 0.05$, $c = 2$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	19	0	0	**81**	0	0	0	0	0	0
Gap	24	0	1	**75**	0	0	0	0	0	0
Silhouette	0	0	0	**94**	6	0	0	0	0	0
In-group proportion	15	9	11	**65**	0	0	0	0	0	0
Prediction strength	18	0	0	**82**	0	0	0	0	0	0
Reference Gap	3	10	2	**81**	4	0	0	0	0	0

(d) $\text{Var}(x_{kij}) = 0.05$, $c = 3$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	65	0	0	**35**	0	0	0	0	0	0
Gap	64	1	4	**31**	0	0	0	0	0	0
Silhouette	0	7	0	**44**	30	14	5	0	0	0
In-group proportion	51	2	6	**41**	0	0	0	0	0	0
Prediction strength	77	1	0	**22**	0	0	0	0	0	0
Reference Gap	13	7	7	**41**	23	6	3	0	0	0

(e) $\text{Var}(x_{kij}) = 0.07$, $c = 2$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	51	0	1	**48**	0	0	0	0	0	0
Gap	50	0	2	**48**	0	0	0	0	0	0
Silhouette	0	11	0	**66**	20	3	0	0	0	0
In-group proportion	55	11	11	**23**	0	0	0	0	0	0
Prediction strength	76	0	0	**24**	0	0	0	0	0	0
Reference Gap	10	11	7	**53**	18	1	0	0	0	0

(f) $\text{Var}(x_{kij}) = 0.07$, $c = 3$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	1	0	0	**99**	0	0	0	0	0	0
Recursive Gap	86	0	0	**14**	0	0	0	0	0	0
Gap	86	0	3	**11**	0	0	0	0	0	0
Silhouette	0	50	0	**14**	13	11	9	3	0	0
In-group proportion	85	3	4	**8**	0	0	0	0	0	0
Prediction strength	98	0	0	**2**	0	0	0	0	0	0
Reference Gap	17	11	12	**16**	23	11	8	1	1	0

Table B.5: The tables report the number of times (out of 100) that a method estimates a given number of clusters for the various values of σ^2 in Scenario B100. The number of times a method estimated the correct number of clusters ($K = 4$) is in bold.

(a) $\mathrm{Var}(x_{kij}) = 0.1$, $c = 2$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	0	0	**100**	0	0	0	0	0	0
Silhouette	0	0	0	**100**	0	0	0	0	0	0
In-group proportion	0	0	0	**100**	0	0	0	0	0	0
Prediction strength	0	0	0	**100**	0	0	0	0	0	0
Reference Gap	0	6	0	**94**	0	0	0	0	0	0

(b) $\mathrm{Var}(x_{kij}) = 0.1$, $c = 3$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	1	0	0	**99**	0	0	0	0	0	0
Gap	1	0	0	**99**	0	0	0	0	0	0
Silhouette	0	0	0	**99**	1	0	0	0	0	0
In-group proportion	1	0	0	**99**	0	0	0	0	0	0
Prediction strength	1	0	0	**99**	0	0	0	0	0	0
Reference Gap	0	6	0	**93**	1	0	0	0	0	0

(c) $\mathrm{Var}(x_{kij}) = 0.2$, $c = 2$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	1	0	0	**99**	0	0	0	0	0	0
Gap	1	0	0	**99**	0	0	0	0	0	0
Silhouette	0	0	0	**99**	1	0	0	0	0	0
In-group proportion	1	0	1	**98**	0	0	0	0	0	0
Prediction strength	6	0	0	**94**	0	0	0	0	0	0
Reference Gap	0	9	22	**68**	1	0	0	0	0	0

(d) $\mathrm{Var}(x_{kij}) = 0.2$, $c = 3$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	39	0	0	**61**	0	0	0	0	0	0
Gap	39	1	5	**55**	0	0	0	0	0	0
Silhouette	0	19	0	**55**	20	6	0	0	0	0
In-group proportion	39	1	6	**54**	0	0	0	0	0	0
Prediction strength	67	1	1	**31**	0	0	0	0	0	0
Reference Gap	0	7	21	**44**	24	3	1	0	0	0

(e) $\mathrm{Var}(x_{kij}) = 0.3$, $c = 2$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	30	0	0	**70**	0	0	0	0	0	0
Gap	30	2	4	**64**	0	0	0	0	0	0
Silhouette	0	21	0	**64**	13	2	0	0	0	0
In-group proportion	35	9	16	**40**	0	0	0	0	0	0
Prediction strength	62	3	1	**34**	0	0	0	0	0	0
Reference Gap	0	11	21	**47**	21	0	0	0	0	0

(f) $\mathrm{Var}(x_{kij}) = 0.3$, $c = 3$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	94	0	0	**6**	0	0	0	0	0	0
Gap	94	1	1	**4**	0	0	0	0	0	0
Silhouette	0	91	0	**4**	3	2	0	0	0	0
In-group proportion	94	1	1	**4**	0	0	0	0	0	0
Prediction strength	100	0	0	**0**	0	0	0	0	0	0
Reference Gap	0	1	6	**11**	22	22	19	9	10	0

Table B.6: The tables report the number of times (out of 100) that a method estimates a given number of clusters for the various values of σ^2 in Scenario B1000. The number of times a method estimated the correct number of clusters ($K = 4$) is in bold.

(a) $\text{Var}(x_{kij}) = 0.5$, $c = 2$

Method K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	0	0	**100**	0	0	0	0	0	0
Silhouette	0	0	0	**100**	0	0	0	0	0	0
In-group proportion	0	0	0	**100**	0	0	0	0	0	0
Prediction strength	0	0	0	**100**	0	0	0	0	0	0
Reference Gap	0	7	17	**76**	0	0	0	0	0	0

(b) $\text{Var}(x_{kij}) = 0.5$, $c = 3$

Method K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	2	0	0	**98**	0	0	0	0	0	0
Gap	2	0	1	**97**	0	0	0	0	0	0
Silhouette	0	0	0	**97**	2	1	0	0	0	0
In-group proportion	2	0	1	**97**	0	0	0	0	0	0
Prediction strength	2	0	0	**98**	0	0	0	0	0	0
Reference Gap	0	8	30	**60**	1	1	0	0	0	0

(c) $\text{Var}(x_{kij}) = 0.75$, $c = 2$

Method K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	0	1	**99**	0	0	0	0	0	0
Silhouette	0	0	0	**99**	1	0	0	0	0	0
In-group proportion	0	2	1	**97**	0	0	0	0	0	0
Prediction strength	2	1	1	**96**	0	0	0	0	0	0
Reference Gap	0	27	16	**56**	1	0	0	0	0	0

(d) $\text{Var}(x_{kij}) = 0.75$, $c = 3$

Method K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	25	0	0	**75**	0	0	0	0	0	0
Gap	25	3	6	**66**	0	0	0	0	0	0
Silhouette	0	23	1	**66**	8	2	0	0	0	0
In-group proportion	25	3	6	**66**	0	0	0	0	0	0
Prediction strength	57	5	0	**38**	0	0	0	0	0	0
Reference Gap	0	18	26	**40**	10	4	2	0	0	0

(e) $\text{Var}(x_{kij}) = 1.0$, $c = 2$

Method K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	10	0	0	**90**	0	0	0	0	0	0
Gap	10	0	3	**87**	0	0	0	0	0	0
Silhouette	0	9	0	**87**	4	0	0	0	0	0
In-group proportion	15	4	11	**70**	0	0	0	0	0	0
Prediction strength	46	3	1	**50**	0	0	0	0	0	0
Reference Gap	0	48	29	**20**	2	1	0	0	0	0

(f) $\text{Var}(x_{kij}) = 1.0$, $c = 3$

Method K	1	2	3	**4**	5	6	7	8	9	10
ERA	3	0	0	**97**	0	0	0	0	0	0
Recursive Gap	83	0	0	**17**	0	0	0	0	0	0
Gap	83	2	7	**8**	0	0	0	0	0	0
Silhouette	0	78	0	**8**	10	2	2	0	0	0
In-group proportion	84	2	6	**8**	0	0	0	0	0	0
Prediction strength	99	0	1	**0**	0	0	0	0	0	0
Reference Gap	0	6	15	**21**	22	19	11	3	3	0

Table B.7: The tables report the number of times (out of 100) that a method estimates a given number of clusters for the various values of σ^2 in Scenario C10. The number of times a method estimated the correct number of clusters ($K = 4$) is in bold.

(a) $\sigma^2 = 0.05$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	0	0	**100**	0	0	0	0	0	0
Silhouette	0	0	0	**100**	0	0	0	0	0	0
In-group proportion	0	0	1	**99**	0	0	0	0	0	0
Prediction strength	0	0	0	**100**	0	0	0	0	0	0
Reference Gap	0	0	0	**100**	0	0	0	0	0	0

(b) $\sigma^2 = 0.1$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	0	1	**99**	0	0	0	0	0	0
Silhouette	0	3	7	**90**	0	0	0	0	0	0
In-group proportion	1	7	27	**65**	0	0	0	0	0	0
Prediction strength	0	6	10	**84**	0	0	0	0	0	0
Reference Gap	1	4	3	**91**	0	0	1	0	0	0

(c) $\sigma^2 = 0.125$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	3	2	4	**91**	0	0	0	0	0	0
Gap	3	2	8	**87**	0	0	0	0	0	0
Silhouette	0	9	20	**66**	5	0	0	0	0	0
In-group proportion	10	20	40	**30**	0	0	0	0	0	0
Prediction strength	7	20	25	**48**	0	0	0	0	0	0
Reference Gap	4	8	21	**66**	1	0	0	0	0	0

(d) $\sigma^2 = 0.15$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	2	**98**	0	0	0	0	0	0
Recursive Gap	10	5	6	**79**	0	0	0	0	0	0
Gap	11	10	11	**68**	0	0	0	0	0	0
Silhouette	0	21	18	**51**	8	2	0	0	0	0
In-group proportion	20	41	21	**18**	0	0	0	0	0	0
Prediction strength	19	54	15	**12**	0	0	0	0	0	0
Reference Gap	4	14	22	**55**	3	1	0	1	0	0

(e) $\sigma^2 = 0.2$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	1	3	11	**85**	0	0	0	0	0	0
Recursive Gap	29	16	11	**44**	0	0	0	0	0	0
Gap	29	23	25	**23**	0	0	0	0	0	0
Silhouette	0	59	12	**20**	8	1	0	0	0	0
In-group proportion	57	28	7	**8**	0	0	0	0	0	0
Prediction strength	59	35	5	**1**	0	0	0	0	0	0
Reference Gap	5	14	35	**36**	5	3	2	0	0	0

(f) $\sigma^2 = 0.25$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	4	6	24	**64**	2	0	0	0	0	0
Recursive Gap	56	18	11	**15**	0	0	0	0	0	0
Gap	55	22	18	**5**	0	0	0	0	0	0
Silhouette	0	73	9	**11**	5	2	0	0	0	0
In-group proportion	80	14	2	**3**	1	0	0	0	0	0
Prediction strength	84	15	1	**0**	0	0	0	0	0	0
Reference Gap	7	22	35	**28**	5	1	2	0	0	0

Table B.8: The tables report the number of times (out of 100) that a method estimates a given number of clusters for the various values of σ^2 in Scenario C100. The number of times a method estimated the correct number of clusters ($K = 4$) is in bold.

(a) $\sigma^2 = 0.2$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	0	0	**100**	0	0	0	0	0	0
Silhouette	0	1	4	**95**	0	0	0	0	0	0
In-group proportion	0	0	0	**100**	0	0	0	0	0	0
Prediction strength	0	0	0	**100**	0	0	0	0	0	0
Reference Gap	0	2	2	**96**	0	0	0	0	0	0

(b) $\sigma^2 = 0.4$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	0	0	**100**	0	0	0	0	0	0
Silhouette	0	71	7	**22**	0	0	0	0	0	0
In-group proportion	0	0	9	**91**	0	0	0	0	0	0
Prediction strength	0	1	3	**96**	0	0	0	0	0	0
Reference Gap	0	10	19	**71**	0	0	0	0	0	0

(c) $\sigma^2 = 0.5$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	1	1	1	**97**	0	0	0	0	0	0
Gap	1	1	7	**91**	0	0	0	0	0	0
Silhouette	0	74	17	**9**	0	0	0	0	0	0
In-group proportion	1	3	33	**63**	0	0	0	0	0	0
Prediction strength	2	6	20	**72**	0	0	0	0	0	0
Reference Gap	0	17	39	**42**	2	0	0	0	0	0

(d) $\sigma^2 = 0.6$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	3	**97**	0	0	0	0	0	0
Recursive Gap	5	6	9	**80**	0	0	0	0	0	0
Gap	5	8	22	**65**	0	0	0	0	0	0
Silhouette	0	76	10	**12**	2	0	0	0	0	0
In-group proportion	5	22	29	**44**	0	0	0	0	0	0
Prediction strength	7	27	31	**35**	0	0	0	0	0	0
Reference Gap	0	17	50	**29**	4	0	0	0	0	0

(e) $\sigma^2 = 0.8$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	6	**94**	0	0	0	0	0	0
Recursive Gap	27	28	12	**33**	0	0	0	0	0	0
Gap	27	33	32	**8**	0	0	0	0	0	0
Silhouette	0	88	7	**2**	2	0	1	0	0	0
In-group proportion	38	36	16	**10**	0	0	0	0	0	0
Prediction strength	49	39	12	**0**	0	0	0	0	0	0
Reference Gap	0	20	34	**23**	15	6	2	0	0	0

(f) $\sigma^2 = 1.0$

Method \ K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	3	26	**70**	1	0	0	0	0	0
Recursive Gap	51	35	9	**5**	0	0	0	0	0	0
Gap	51	36	13	**0**	0	0	0	0	0	0
Silhouette	0	93	5	**0**	2	0	0	0	0	0
In-group proportion	58	30	7	**5**	0	0	0	0	0	0
Prediction strength	68	30	2	**0**	0	0	0	0	0	0
Reference Gap	0	19	16	**21**	13	12	11	5	3	0

Table B.9: The tables report the number of times (out of 100) that a method estimates a given number of clusters for the various values of σ^2 in Scenario C1000. The number of times a method estimated the correct number of clusters ($K = 4$) is in bold.

(a) $\sigma^2 = 1.0$

Method / K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	0	0	**100**	0	0	0	0	0	0
Silhouette	0	100	0	**0**	0	0	0	0	0	0
In-group proportion	0	0	0	**100**	0	0	0	0	0	0
Prediction strength	0	0	0	**100**	0	0	0	0	0	0
Reference Gap	0	6	54	**40**	0	0	0	0	0	0

(b) $\sigma^2 = 1.4$

Method / K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	0	0	0	**100**	0	0	0	0	0	0
Gap	0	0	0	**100**	0	0	0	0	0	0
Silhouette	0	100	0	**0**	0	0	0	0	0	0
In-group proportion	0	0	4	**96**	0	0	0	0	0	0
Prediction strength	0	0	5	**95**	0	0	0	0	0	0
Reference Gap	0	9	38	**53**	0	0	0	0	0	0

(c) $\sigma^2 = 1.6$

Method / K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	0	**100**	0	0	0	0	0	0
Recursive Gap	1	1	1	**97**	0	0	0	0	0	0
Gap	1	1	2	**96**	0	0	0	0	0	0
Silhouette	0	99	1	**0**	0	0	0	0	0	0
In-group proportion	1	1	14	**84**	0	0	0	0	0	0
Prediction strength	1	3	12	**84**	0	0	0	0	0	0
Reference Gap	0	15	30	**53**	2	0	0	0	0	0

(d) $\sigma^2 = 1.8$

Method / K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	2	**98**	0	0	0	0	0	0
Recursive Gap	1	4	3	**92**	0	0	0	0	0	0
Gap	1	4	19	**76**	0	0	0	0	0	0
Silhouette	0	95	4	**1**	0	0	0	0	0	0
In-group proportion	1	6	30	**63**	0	0	0	0	0	0
Prediction strength	3	7	24	**66**	0	0	0	0	0	0
Reference Gap	0	21	37	**39**	2	1	0	0	0	0

(e) $\sigma^2 = 2.0$

Method / K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	0	20	**80**	0	0	0	0	0	0
Recursive Gap	1	7	26	**66**	0	0	0	0	0	0
Gap	1	7	48	**44**	0	0	0	0	0	0
Silhouette	0	99	1	**0**	0	0	0	0	0	0
In-group proportion	1	9	32	**58**	0	0	0	0	0	0
Prediction strength	5	17	41	**37**	0	0	0	0	0	0
Reference Gap	0	23	33	**35**	6	3	0	0	0	0

(f) $\sigma^2 = 2.2$

Method / K	1	2	3	**4**	5	6	7	8	9	10
ERA	0	2	73	**25**	0	0	0	0	0	0
Recursive Gap	3	15	55	**27**	0	0	0	0	0	0
Gap	3	15	72	**10**	0	0	0	0	0	0
Silhouette	0	99	0	**1**	0	0	0	0	0	0
In-group proportion	3	23	41	**33**	0	0	0	0	0	0
Prediction strength	7	36	36	**21**	0	0	0	0	0	0
Reference Gap	0	33	36	**19**	11	1	0	0	0	0

Table B.10: The tables report the number of times (out of 100) that a method estimates a given number of clusters for the various values of σ^2 in Scenario D1. The number of times a method estimated the correct number of clusters ($K = 5$) is in bold.

(a) $\sigma^2 = 0.05$

Method \ K	1	2	3	4	**5**	6	7	8	9	10
ERA	0	0	0	0	**100**	0	0	0	0	0
Recursive Gap	0	0	0	0	**100**	0	0	0	0	0
Gap	0	0	0	0	**100**	0	0	0	0	0
Silhouette	0	0	0	100	**0**	0	0	0	0	0
In-group proportion	0	0	0	0	**100**	0	0	0	0	0
Prediction strength	0	0	0	0	**100**	0	0	0	0	0
Reference Gap	0	16	0	6	**78**	0	0	0	0	0

(b) $\sigma^2 = 0.1$

Method \ K	1	2	3	4	**5**	6	7	8	9	10
ERA	0	0	0	0	**100**	0	0	0	0	0
Recursive Gap	0	0	0	0	**100**	0	0	0	0	0
Gap	0	0	0	10	**90**	0	0	0	0	0
Silhouette	0	92	0	8	**0**	0	0	0	0	0
In-group proportion	0	0	0	0	**100**	0	0	0	0	0
Prediction strength	0	0	0	0	**100**	0	0	0	0	0
Reference Gap	0	30	0	42	**28**	0	0	0	0	0

(c) $\sigma^2 = 0.15$

Method \ K	1	2	3	4	**5**	6	7	8	9	10
ERA	0	0	0	0	**100**	0	0	0	0	0
Recursive Gap	0	0	0	0	**100**	0	0	0	0	0
Gap	0	0	0	85	**15**	0	0	0	0	0
Silhouette	0	100	0	0	**0**	0	0	0	0	0
In-group proportion	0	0	0	25	**75**	0	0	0	0	0
Prediction strength	0	0	0	13	**87**	0	0	0	0	0
Reference Gap	0	37	3	56	**4**	0	0	0	0	0

(d) $\sigma^2 = 0.2$

Method \ K	1	2	3	4	**5**	6	7	8	9	10
ERA	0	0	0	5	**95**	0	0	0	0	0
Recursive Gap	0	0	0	10	**90**	0	0	0	0	0
Gap	0	0	0	99	**1**	0	0	0	0	0
Silhouette	0	100	0	0	**0**	0	0	0	0	0
In-group proportion	0	0	0	50	**50**	0	0	0	0	0
Prediction strength	0	0	0	71	**29**	0	0	0	0	0
Reference Gap	0	29	14	56	**1**	0	0	0	0	0

(e) $\sigma^2 = 0.25$

Method \ K	1	2	3	4	**5**	6	7	8	9	10
ERA	0	0	0	36	**64**	0	0	0	0	0
Recursive Gap	0	0	0	47	**53**	0	0	0	0	0
Gap	0	0	0	100	**0**	0	0	0	0	0
Silhouette	0	100	0	0	**0**	0	0	0	0	0
In-group proportion	0	0	0	71	**29**	0	0	0	0	0
Prediction strength	0	0	0	88	**12**	0	0	0	0	0
Reference Gap	0	22	13	65	**0**	0	0	0	0	0

(f) $\sigma^2 = 0.3$

Method \ K	1	2	3	4	**5**	6	7	8	9	10
ERA	0	0	0	78	**22**	0	0	0	0	0
Recursive Gap	0	0	0	84	**16**	0	0	0	0	0
Gap	0	0	0	100	**0**	0	0	0	0	0
Silhouette	0	100	0	0	**0**	0	0	0	0	0
In-group proportion	0	0	1	83	**16**	0	0	0	0	0
Prediction strength	0	0	1	90	**9**	0	0	0	0	0
Reference Gap	0	25	26	49	**0**	0	0	0	0	0

Table B.11: The tables report the number of times (out of 100) that a method estimates a given number of clusters for the various values of σ^2 in Scenario D2. The number of times a method estimated the correct number of clusters ($K = 6$) is in bold.

(a) $\sigma^2 = 0.05$

Method \ K	1	2	3	4	5	**6**	7	8	9	10
ERA	0	0	0	0	0	**100**	0	0	0	0
Recursive Gap	0	0	0	0	0	**100**	0	0	0	0
Gap	0	0	0	0	0	**100**	0	0	0	0
Silhouette	0	0	0	100	0	**0**	0	0	0	0
In-group proportion	0	0	0	0	0	**99**	1	0	0	0
Prediction strength	0	0	0	0	0	**100**	0	0	0	0
Reference Gap	0	92	0	1	0	**7**	0	0	0	0

(b) $\sigma^2 = 0.1$

Method \ K	1	2	3	4	5	**6**	7	8	9	10
ERA	0	0	0	0	0	**100**	0	0	0	0
Recursive Gap	0	0	0	0	0	**100**	0	0	0	0
Gap	0	0	0	40	5	**55**	0	0	0	0
Silhouette	0	5	0	95	0	**0**	0	0	0	0
In-group proportion	0	0	0	4	6	**90**	0	0	0	0
Prediction strength	0	0	0	8	18	**74**	0	0	0	0
Reference Gap	0	87	0	10	3	**0**	0	0	0	0

(c) $\sigma^2 = 0.15$

Method \ K	1	2	3	4	5	**6**	7	8	9	10
ERA	0	0	0	1	0	**99**	0	0	0	0
Recursive Gap	0	0	0	7	0	**93**	0	0	0	0
Gap	0	0	0	96	3	**1**	0	0	0	0
Silhouette	0	96	0	4	0	**0**	0	0	0	0
In-group proportion	0	0	0	54	37	**9**	0	0	0	0
Prediction strength	0	0	0	69	31	**0**	0	0	0	0
Reference Gap	0	70	0	18	10	**1**	0	1	0	0

(d) $\sigma^2 = 0.2$

Method \ K	1	2	3	4	5	**6**	7	8	9	10
ERA	0	0	0	9	0	**91**	0	0	0	0
Recursive Gap	0	0	0	38	0	**62**	0	0	0	0
Gap	0	0	0	100	0	**0**	0	0	0	0
Silhouette	0	100	0	0	0	**0**	0	0	0	0
In-group proportion	0	0	0	89	10	**1**	0	0	0	0
Prediction strength	0	0	0	97	3	**0**	0	0	0	0
Reference Gap	0	49	1	45	3	**1**	1	0	0	0

(e) $\sigma^2 = 0.25$

Method \ K	1	2	3	4	5	**6**	7	8	9	10
ERA	0	0	0	31	7	**58**	4	0	0	0
Recursive Gap	0	0	0	75	1	**24**	0	0	0	0
Gap	0	0	0	100	0	**0**	0	0	0	0
Silhouette	0	99	0	1	0	**0**	0	0	0	0
In-group proportion	0	0	0	98	2	**0**	0	0	0	0
Prediction strength	0	0	0	100	0	**0**	0	0	0	0
Reference Gap	0	63	3	34	0	**0**	0	0	0	0

(f) $\sigma^2 = 0.3$

Method \ K	1	2	3	4	5	**6**	7	8	9	10
ERA	0	0	0	55	12	**33**	0	0	0	0
Recursive Gap	0	0	0	94	3	**3**	0	0	0	0
Gap	0	0	0	100	0	**0**	0	0	0	0
Silhouette	0	100	0	0	0	**0**	0	0	0	0
In-group proportion	0	0	1	99	0	**0**	0	0	0	0
Prediction strength	0	0	0	100	0	**0**	0	0	0	0
Reference Gap	0	62	11	27	0	**0**	0	0	0	0

Bibliography

Alizadeh, A. A., Eisen, M., Davis, E. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X. et al. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature* **403**, 503–511.

Dudoit, S. & Fridlyand, J. (2002). A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology* **3**, 0036.1–0036.21.

Eisen, M. (1998). Cluster and Treeview manual. `http://ranalblgov/manuals/ClusterTreeViewpdf`.

Eisen, M. B., Spellman, T. B., Brown, P. O. & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *PNAS* **95**, 14863–14868.

Everitt, B. S., Landau, S. & Leese, M. (2001). *Cluster Analysis*. London: Arnold, 4th ed.

Gordon, A. D. (1999). *Classification*. Boca Raton, FL: Chapman & Hall/CRC, 2nd ed.

Hastie, T., Tibshirani, R. & Friedman, J. (2001). *The Elements of Statistical Learning: Data mining, Inference and Prediction*. New York: Springer.

Kapp, A. V. (2007). *Cluster analysis of microarray data using the in-group proportion*. Ph.D. thesis, Stanford University.

Kapp, A. V. & Tibshirani, R. (2007). Are clusters found in one dataset present in another dataset? *Biostatistics* **8**, 9–31.

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

Langfelder, P., Zhang, B. & Horvath, S. (2008). Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut library for R. *Bioinformatics* **24**, 719–720.

Liestøl, K., Baumbusch, L., Børresen-Dale, A.-L. & Lingjærde, O. C. (2008). Outliers and detection of copy number changes. *In review* .

Naume, B., Zhao, X., Synnestvedt, M., Borgen, E., Russnes, H., Lingjærde, O. C., Strømberg, M., Wiedswang, G., Kvalheim, G., Kåresen, R. et al. (2007). Presence of bone marrow micrometastasis is associated with different recurrence risk within molecular subtypes of breast cancer. *Molecular Oncology* **1**, 160–171.

Perou, C. M., Sørlie, T., Eisen, M. B., van deRijn, M., Jeffrey, S. S., Rees, C. A., Pollack, J. R., Ross, D. T., Johnsen, H., Akslen, L. A. et al. (2000). Molecular portraits of human breast tumours. *Nature* **406**, 747–752.

Quackenbush, J. (2006). Microarray analysis and tumor classification. *The New England Journal of Medicine* **354**, 2463–2472.

Solberg, E. (2007). *Klustring av mikromatrisedata: Estimering av antall klustre og identifikasjon av subtyper.* Master's thesis, University of Oslo.

Speed, T. (2003). *Statistical analysis of gene expression microarray data.* Boca Raton, FL: Chapman & Hall/ CRC.

Sørlie, T., Perou, C. M., Tibshirani, R., Aas, T., Geisler, S., Johnsen, H., Hastie, T., Eisen, M. B., van de Rijn, M., Jeffrey, S. S. et al. (2001). Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *PNAS* **98**, 10869–10874.

Sørlie, T., Tibshirani, R., Parker, J., Hastie, T., Marron, J. S., Nobel, A., Deng, S., Johnsen, H., Pesich, R., Geisler, S. et al. (2003). Repeated observation of breast tumor subtypes in independent gene expression data sets. *PNAS* **100**, 8418–8423.

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

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

Timm, N. H. (2002). *Applied Multivariate Analysis.* New York: Springer.

Tryon, R. C. (1939). *Cluster analysis.* Ann Arbor: Edwards Brothers Inc.

Xiong, J. (2006). *Essential Bioinformatics.* New York: Cambridge University Press.