Austrian Journal of Statistics
2024, Volume 53, 1–19.
http://www.ajs.or.at/
doi:10.17713/ajs.v53i3.1169

Bootstrapping Cluster Analysis Solutions with the R Package ClusBoot

Sugnet Gardner-Lubbe
Stellenbosch University
South Africa

Abstract

Finding true clusters in an unsupervised setting is a difficult problem. In most cases a data set can be clustered into a specific number of clusters whether this supports the underlying structure of the data or not. The package **ClusBoot** uses a bootstrap analysis of any clustering algorithm to provide its user with some measures of the stability in the clustering solution. Observations that cluster together repeatedly over many bootstrap replications can be considered similar enough to be grouped into a cluster while observations that only cluster together by chance indicates a lack of true grouping structure. The package performs the bootstrap analysis and provide the user with summary measures in the form of a bootstrap-silhouette plot and graphical visualisation to assess the stability of the clustering solution.

Keywords: bootstrap, cluster analysis, R.

1. Introduction

Cluster analysis or unsupervised learning aims to identify groups or clusters in data. There are many cluster analysis algorithms which follow some step-wise approach while other are based on some form of modelling to divide items or observations into clusters. All these methods aim to find groups of observations such that those inside a cluster are more alike than those from different clusters. In many cases the number of clusters, k, needs to be specified upfront. These methods will divide the observations into k clusters, whether or not the underlying structure of the data supports a k-cluster configuration. On the other hand, hierarchical methods produce a series of mergers (or occasionally divisions) that can be represented as n clusters, $(n-1)$ clusters, $(n-2)$ clusters, ..., 2 clusters, a single cluster, where n denotes the number of objects in the data set. Usually hierarchical clustering procedures produce a tree structure and the actual clustering into k clusters is obtained by cutting the tree at an appropriate height or distance between clusters. Finding a solution with k clusters in no way confirms a true separation in the data.

Consider a bivariate data set consisting of $n = 50$ observations from two uncorrelated standard normal random variables. There is no underlying clustering structure in this data set. Performing a simple complete linkage cluster analysis with the defaults of the R function

`hclust()` (Becker, Chambers, and Wilks 1988) on the Euclidean distances between the samples and cutting the clustering tree such that three clusters is obtained, the left hand plot in Figure 1 is produced. Repeating the process with a new random sample of $n = 50$ observations provides a completely different clustering solution as shown in the right panel, confirming the lack of any three-group structure in the data.

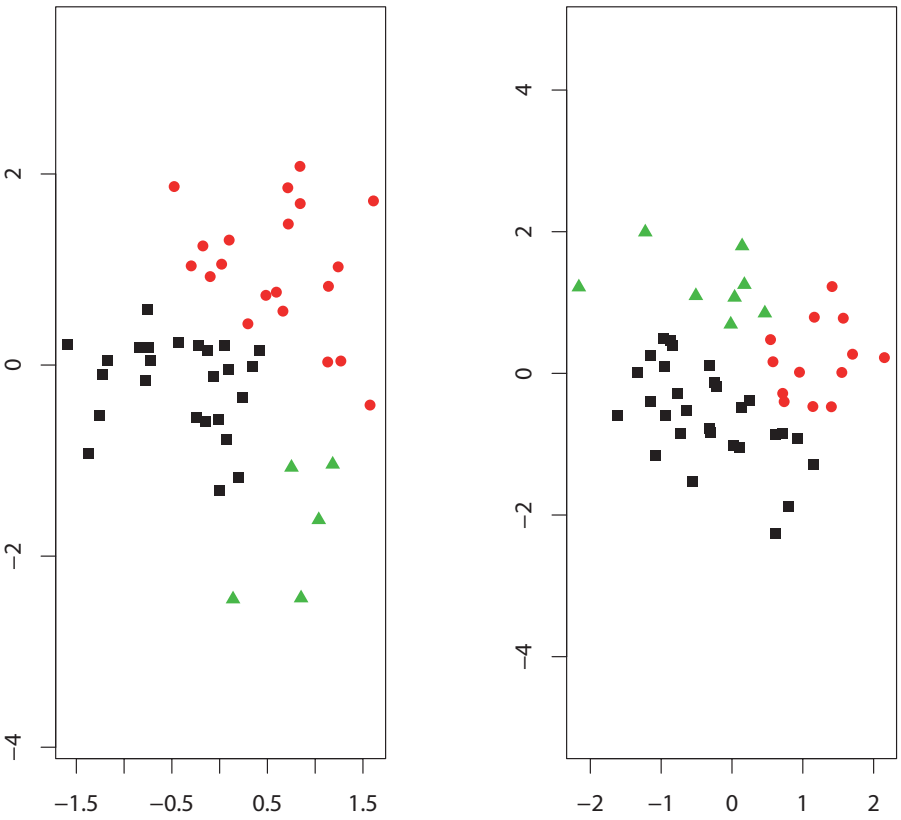

Figure 1: Three-cluster solution of complete linkage clustering on a single random standard normal bivariate data set

The problem with unsupervised learning is exactly that it is unsupervised, i.e. it is unknown whether the data has a clustering structure or how many clusters exist in the data. For a comprehensive discussion on the problem see Hennig (2015). In a more recent paper Ullmann, Hennig, and Boulesteix (2022) provide different approaches to "validity" in cluster analysis. These authors discuss cluster validity in terms of:

- Recovery of "true" clusters, which necessarily requires some form of truth to be known.
- External validation where additional information is available to assess the cluster solution.
- Internal validation calculating an index that is supposed to measure how well the clustering fits the data. Several such measures have been defined in order to evaluate the quality of a clustering solution. As an example, Rousseeuw (1987) defines the silhouette width for each observation. Averaging over the n silhouette widths provides a measure of how well the observations are clustered in their own cluster relative to the nearest alternative cluster. In addition, maximising the average silhouette width over different values of k can be used to select the number of clusters.

- Visual validation which is not the primary focus of this paper, but will be employed through multidimensional scaling plots as auxiliary confirmation of the clustering solution. The scatter plots in Figure 1 is a trivial example of this.
- Cluster stability which is the aspect we will focus on here.

In a comprehensive overview Liu, Yu, and Blair (2022) explains that cluster stability will indicate that a good clustering solution has been found since it will reproduce over an ensemble of perturbed data sets, nearly identical to the original data. Methods based on bootstrapping (see Jain and Moreau (1987), Kerr and Churchill (2001), Dudoit and Fridlyand (2003), Hennig (2007), Fang and Wang (2012) and Yu, Chapman, Florio, Eischen, Gotz, Jacob, and Blair (2019)) and cluster validation via data splitting and subsampling (see Levine and Domany (2001), Ben-Hur, Elisseeff, and Guyon (2002), Dudoit and Fridlyand (2002), Lange, Roth, Braun, and Buhmann (2004) and Tibshirani and Walther (2005)) are discussed, as well as the Loevinger method (Bertrand and Mufti 2006), using matrix manipulation to assess cluster stability (Steinley 2008) and the optimal transport alignment method of Liu, Seo, and Lin (2019). Nemec and Brinkhurst (1988) apply the bootstrap to some measure such as the similarity at merger while Pillar (1999) focus on nearest neighbour sum of squared distances. Another application of subsetting is discussed in (Ben-Hur and Guyon 2003) where cluster stability is used in combination with dimension reduction to select the optimal clustering algorithm, the normalisation and dissimilarity measure as well as the number of clusters.

Different applications of these methods have different aims or is specific to particular clustering algorithms. The bootstrap method implemented in **ClusBoot** can be applied to any clustering solution, irrespective of the method used to arrive at the clustering of n observations into k clusters. Silhouette values are computed to assess different samples' and clusters' stability and the a visualisation is suggested for visualising the tendency of the individual observations to be classified to other clusters. Aspects of the methodology is similar to that of Hennig (2007) and will be referred to in the following sections.

This paper is organised as follows. Section 2 discuss how the bootstrap methodology is applied here as well as the computation of silhouette values. In section 3 the focus is shifted to performing the stability analysis with **ClusBoot** with two examples. A simulation study in section 4 compares **ClusBoot** to some other options available in R and illustrate how the silhouette values can be used for selecting the number of clusters. The paper concludes with a discussion in section 5.

2. Boostrap analysis of clustering solutions

In this application of bootstrap analysis, consider a data set $\mathbf{X} : n \times p$ of n observations on p variables. In many clustering algorithms, the input to the cluster analysis is in the form of an $n \times n$ matrix $\mathbf{D}$, of pairwise dissimilarities between the observations. The chosen cluster analysis method is applied and a vector of cluster membership $\mathbf{a} : n \times 1$ is obtained where $a_i \in \{1, 2, ..., k\}$.

A bootstrap sample is obtained by selecting, with replacement, from the set of observations labelled $\{1, ..., n\}$ say $i_1, ..., i_n$. A bootstrap replicate $\mathbf{X}^* : n \times p$ of the rows of $\mathbf{X}$ corresponding to the labels $i_1, ..., i_n$ is obtained. Alternatively, a bootstrap replicate $\mathbf{D}^* : n \times n$ of pairwise dissimilarities for all pairs in $i_1, ..., i_n$ is constructed from the matrix $\mathbf{D}$. The same cluster analysis method is applied to $\mathbf{X}^*$ or $\mathbf{D}^*$ to obtain a vector of cluster memberships $\mathbf{a}^*$. The complete bootstrap clustering process is repeated a large number, say $B = 1000$ times, to obtain cluster membership vectors $\mathbf{a}_1^*, ..., \mathbf{a}_B^*$.

In order to condense the information contained in the B cluster membership vectors into a useful form, the proportion of times each pair of objects are clustered together is computed. Note that since the bootstrap samples are taken with replacement, the same observation might appear more than once in a bootstrap replicate, but will be clustered with itself. The

number of times each of the original observations appear across the B bootstrap replicates also differ so that the proportions are based on

$$\frac{number\ of\ times\ pair\ cluster\ to\ the\ same\ clustering\ group}{number\ of\ times\ pair\ appear\ in\ the\ same\ boostrap\ data\ set}. \tag{1}$$

Say observation i' appears twice in bootstrap sample b while observation j' appears once, the pair (i', j') will appear twice in the same bootstrap data set and since the two occurrences of i' will cluster in the same cluster, the number of times the pair (i', j') cluster to the same clustering group will either be zero or two. The proportion above is computed across all B bootstrap replicates.

These proportions can be represented in a symmetric matrix $\mathbf{P} : n \times n$ where the diagonal values are all equal to one since each observation will always cluster with itself. The off-diagonal values will range between zero and one, where values close to one represent pairs of observations that repeatedly cluster together. Observations that often cluster together show a strong cluster relationship. Observations that never, or hardly ever cluster together are very dissimilar. Since the proportions are continuous values, these provide a measure of the association between each pair of objects.

The implementation of Hennig (2007) in **fpc** is similar, computing a stability measure for each cluster based on the Jaccard coefficient, rather than the proportion of times objects cluster together. When drawing a bootstrap sample, since sampling is with replacement from the n observations, it is almost inevitable that some observations will appear more than once, while others will not appear in any particular bootstrap replicate. In **fpc** the Jaccard coefficient is based on comparing the original cluster solution to the bootstrap replicates, inadvertently "penalising" the matching cluster from the bootstrap replicate for not containing observations which does not appear in the particular bootstrap sample and therefore not in the matching cluster for that bootstrap replicate. The values **ClusBoot** compute in matrix $\mathbf{P}$ based on equation (1) compare the number of times a pair of observations cluster together with the number of times they appear together in the bootstrap samples, thus no "penalisation" is introduced if one or both of the samples do not appear in any particular bootstrap replicate.

On the other hand, Liu *et al.* (2022) points out that the repetition of observations in a bootstrap sample can cause bias, especially in centre-based methods such as k-means. If observation i' appears twice in the bootstrap sample the number of times it appear with each other sample will be doubled in the denominator of (1), thus offsetting some of the bias. In B bootstrap replicates, observation i' will appear twice in some replicates while not at all in others, averaging out over repeated clustering solutions.

Similar to the silhouette plots of Rousseeuw (1987) the bootstrap results are used to provide a bootstrap-silhouette of the clustering solution. For each cluster, the clustering tightness, compared to the "nearest" alternative cluster is computed. Individual cluster tightness is computed as

$$t_h = \frac{2}{n_h(n_h - 1)} \sum \sum_{i,j \in C_h, i<j} p_{ij} \tag{2}$$

where n_h is the number of observations clustered to cluster h (C_h), i.e. the mean of the lower (or upper) triangle of the diagonal block of the matrix $\mathbf{P}$ associated with cluster h, $h = 1, ..., k$. The "nearest" alternative cluster to cluster h is the cluster $l \neq h, l = 1, ..., k$ whose elements tend to cluster with cluster h most often, i.e. with the largest mean proportion

$$t_{hl}^{(alt)} = \frac{1}{n_h n_l} \sum_{i \in C_h} \sum_{j \in C_l} p_{ij}. \tag{3}$$

Finally, the bootstrap-silhouette value is computed as

$$s_h = t_h - max_{l \neq h} t_{hl}^{(alt)}. \tag{4}$$

Note that in the above, the silhouette values are computed per cluster which is not analogous to the original silhouettes defined by Rousseeuw (1987), which are defined for every individual point and then averaged. In a perfect situation where the observations in cluster h are tight knit and very different from all other observations, $t_h = 1$ and $t_{hl}^{(alt)} = 0$, $l = 1, ..., k$, $l \neq h$ so that a maximum value of $s_h = 1$ is obtained. At the other extreme, if the observations of cluster h cluster more often with the observations of cluster l' than with other observations in cluster h or observations in another cluster $l \neq l', l \neq h$, the value $t_{hl'}^{(alt)} > t_{hl}^{(alt)}$, $l \neq l'$ and $t_{hl'}^{(alt)} > t_h$, which will lead to a negative value for s_h. It is assumed that any "reasonably good" clustering algorithm will not lead to any negative s_h values, but theoretically negative bootstrap-silhouette values are not precluded.

In a similar manner, bootstrap-silhouette values can be computed for individual observations. In equation (2) the mean proportion within the cluster is computed. To define an individual silhouette value for sample i, the mean proportion is computed for all other observations clustering with sample i in the original cluster solution, i.e.

$$t_i = \frac{1}{n_h - 1} \sum_{j \in C_h, j \neq i} p_{ij} \tag{5}$$

where sample i appears in cluster h in the original solution and n_h is the number of observations in cluster h. The individual bootstrap-silhouette value is then defined as

$$s_i = t_i - max_{l \neq h} t_{hl}^{(alt)}. \tag{6}$$

These individual bootstrap-silhouette values can be combined to calculate an overall stability measure for the clustering solution,

$$stability = \frac{1}{n} \sum_{i=1}^{n} s_i \tag{7}$$

to allow comparison of different algorithms for the same data set or even across data sets.

Even though the matrix $\mathbf{P}$ provide valuable information on the clustering structure or lack thereof in the data set, a large matrix of proportions could be difficult to interpret at a glance. A visual representation of the bootstrap analysis of the clustering solution can be obtained by performing a multidimensional scaling analysis related to the symmetric matrix $\mathbf{P} : n \times n$. Define the dissimilarity measure as

$$d_{ij} = 1 - p_{ij} \tag{8}$$

then $d_{ii} = 0$ and the more different observations i and j, the less seldom they will cluster together, i.e. p_{ij} will be small and d_{ij} will be relatively large.

Typically, there are many small p_{ij} values which lead to many $d_{ij} > 0.9$ (say). These represent the pairs of observations not clustering together. Since the aim of the representation is to identify the forming of clusters, more emphasis is placed on the representation of small dissimilarities. Let δ_{ij} represent the Euclidean distance between observations i and j in the MDS representation, a configuration in two dimensions $\mathbf{Y} : n \times 2$ is sought to minimise the weighted stress function

$$\frac{\sum \sum_{i<j} (d_{ij} - \delta_{ij})^2}{\sum \sum_{i<j} d_{ij}^2}. \tag{9}$$

3. ClusBoot implementation

The `case.study.psychiatrist` data set (Mechelen and Boeck 1989) available in the Cluster Benchmark Data Repository of the International Federation of Classification Society's website https://ifcs.boku.ac.at/repository/data/case_study_psychiatrist/index.html is included in the package **ClusBoot**. The data set consists of observations on 30 patients on 28 variables. All variables are binary indicators except `V25` which is a 101 point rating scale.

The `clusboot()` function performs the cluster analysis on the data set as well as the bootstrap analysis and returns a clustering vector $\mathbf{a} : n \times 1$, the proportion matrix $\mathbf{P} : n \times n$ and bootstrap-silhouette values. The function accepts as input either the data set $\mathbf{X} : n \times p$ or matrix of dissimilarities $\mathbf{D} : n \times n$ which-ever is appropriate input for the clustering function. To specify the specific clustering method, a function is specified which returns as only output a clustering vector of the form $\mathbf{a} : n \times 1$. As most R functions which perform some form of cluster analysis has multiple output lists, typically a wrapper function needs to be specified. The default included in the **ClusBoot** package is the function `complete.linkage` with the structure

```
complete.linkage <- function (X, k)
{  cutree(hclust(dist(X)), k)  }
```

Optionally the number of bootstrap replications to be performed can be specified in the argument `B` and provision is made for any additional arguments to be sent to the clustering function.

```
clusboot (datmat, B = 1000, clustering.func = complete.linkage,  ...)
```

Since one of the variables in the `case.study.psychiatrist` data set is not measured on the same scale as the others, the data set is standardised to mean zero and unit standard deviation for each variable before applying complete linkage hierarchical clustering on the Euclidean distances between patients. The following code performs a six cluster analysis, including a bootstrap analysis of the clustering solution.

```
library(ClusBoot)
data(case.study.psychiatrist)
boot.out <- clusboot (scale(case.study.psychiatrist), B=1000, k=6,
    clustering.func = complete.linkage)
```

The usual clustering tree shown in Figure 2 is obtained by performing the complete linkage hierarchical clustering with the usual R functions outside the `clusboot()` function with the code

```
out.hclust <- hclust(dist(scale(case.study.psychiatrist)))
plot(out.hclust)
rect.hclust(out.hclust, k = 6, border = "red")
```

In order to assess the stability of the clustering solution in Figure 2, the bootstrap output proportion matrix is given in Figure 3. This output is obtained by calling

```
boot.proportions(boot.out,
                 col = colorRampPalette(c("white","mediumseagreen"))(101),
                 show.vals = TRUE)
```

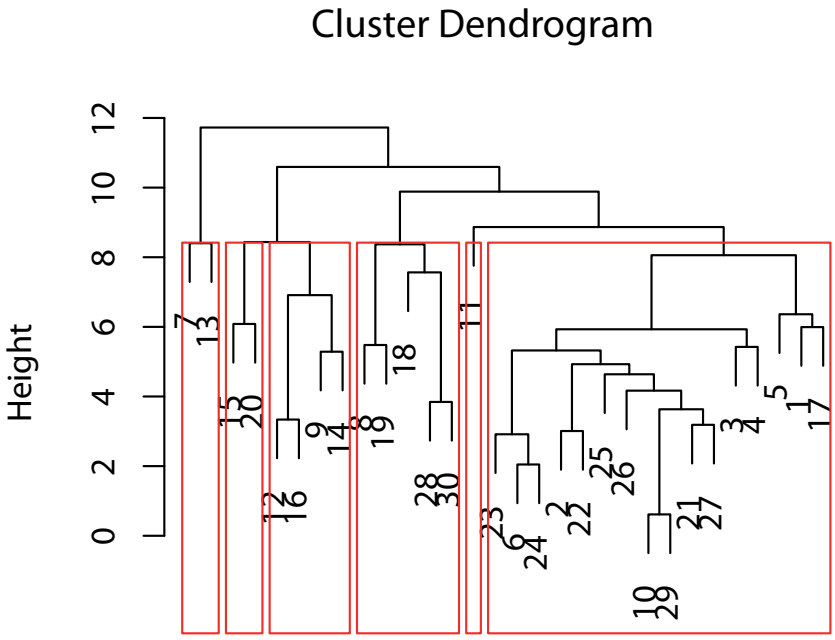

Figure 2: Complete linkage clustering tree with six clusters on scaled case study psychiatrist data set

The objects in the matrix **P** is sorted such that objects that appear in the same cluster is adjacent, while the clusters are ordered in decreasing cluster tightness (see equation (2)). The `clusboot()` function returns an object of class `clusboot` which includes a vector `sil.order` which is typically not used by the end-user, but by the function `boot.proportions()` and the silhouette plots.

Patient 11 clusters by itself. Perusal of Figure 3 reveals that over 1000 bootstrap replicates it maximally clustered 27% of the time with patient 4 from the large cluster. It never clustered with any of the other clusters except for patient 19. In addition, the cluster containing patients 8, 18, 19, 28 and 30 does not seem to form a cohesive unit. More than 40% of the time patients 28 and 30 cluster with the large cluster. Note that patients 7 and 13 are clustered together, but the bootstrap analysis suggest that they are very different. Furthermore patient 7 does not cluster with any other patient while patient 13 might be more similar to patients 1 and 17 than to patient 7.

With a call to the `boot.silhouette()` function, the silhouette values are obtained as well as a plot of these values. This is shown for the six cluster solution of the `case.study.psychiatrist` data set in Figure 4.

```
boot.silhouette (boot.out)
```

The cluster labels $\{1, 2, ..., k\}$ are used to identify the clusters. Using the clustering vector output from the `clusboot()` function, cluster 5 is easily identified as the cluster containing only patient 11 while the cluster with patients 7 and 13 are cluster 2. Since there are no off-diagonal values in the diagonal block of the matrix **P** associated with cluster 5 (patient 11), no bootstrap-silhouette value is computed. The bootstrap-silhouette plot confirms the conclusions reached by inspection of Figure 3, with low values for cluster 2 patients 7 and 13) and cluster 3 (patients 8, 18, 19, 28, 30).

Finally the `plot` method provides the MDS map of the bootstrapped clustering solution in Figure 5. The function plots the observations colour coded by cluster membership. With the

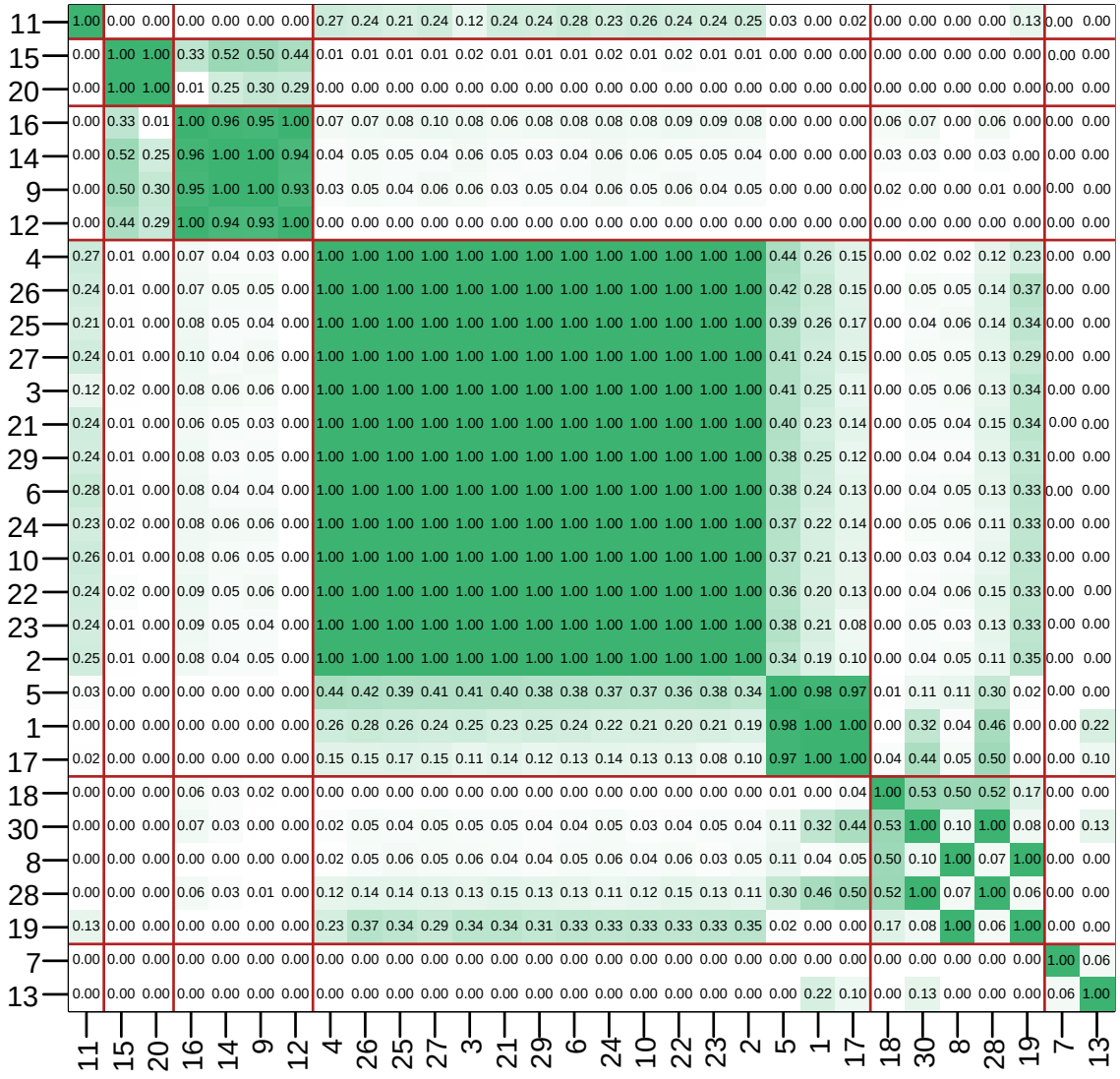

Figure 3: Proportion each pair of patients in the `case.study.psychiatrist` data set cluster together

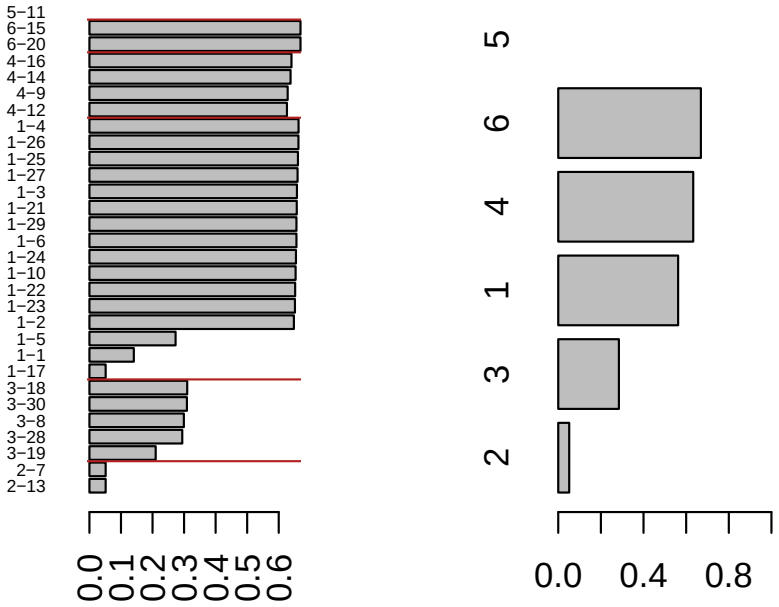

Figure 4: Bootstrap-silhouette plot of six cluster solution of the scaled `case.study.psychiatrist` data set

code below, the plot is repeated with the patient identifiers to compare with the discussion above.

```
out <- plot(boot.out)
plot (out, asp=1, type="n", xaxt="n", yaxt="n", xlab="", ylab="")
for (i in 1:nrow(out))
    text (out[i,1], out[i,2], label=rownames(boot.out$proportions)[i],
      col=boot.out$clustering[i])
legend ("left",legend=1:6,col=1:6,pch=15)
```

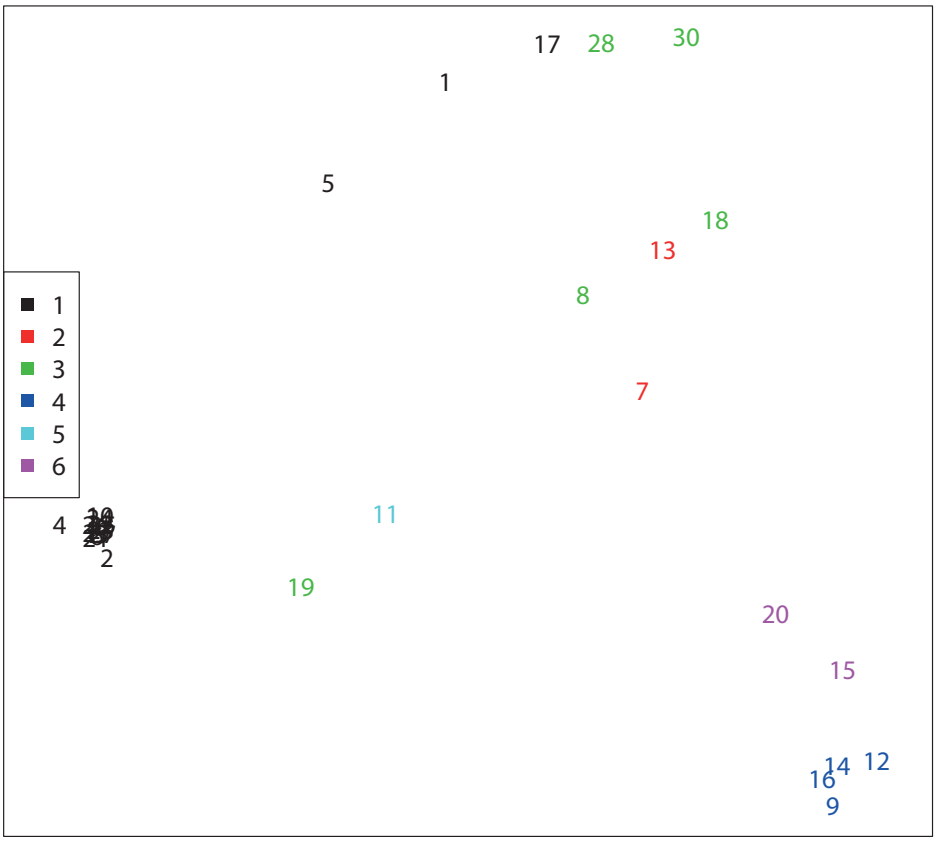

Figure 5: MDS plot of bootstrap analysis of six cluster solution of the scaled `case.study.psychiatrist` data set

Perusal of Figure 5 confirms the interpretations made based on Figure 3 and Figure 4. Patient 11 is fairly isolated, but closest to patient 19. Patients 8, 18, 19, 28 and 30 are spread widely with specifically patients 28 and 30 close to members of the large cluster. The separation between patients 1, 5 and 17, from their other cluster members are very prominent in Figure 5. None of the conclusions reached based on the bootstrap analysis contradicts the structure of the clustering tree in Figure 2, but it should be kept in mind that both the original analysis and the bootstrap application is based on complete linkage. The clustering characteristics emphasised by the specific clustering methodology will therefore carry through to the bootstrap analysis.

It is important not to confuse the MDS plot of the bootstrap analysis with an MDS plot of the data set. In Figure 5, the distances between observations are based on the output from the bootstrap replications as provided in the matrix $\mathbf{P}$. In Figure 6 a Sammon map (Venables and Ripley 2002) of the data set $\mathbf{X} : n \times p$ based on the Euclidean distances between observations

is shown. Typically one would expect grouping of the observations according to the clustering solution. Inspection of the Sammon map is the visual validation referred to in Ullmann *et al.* (2022).

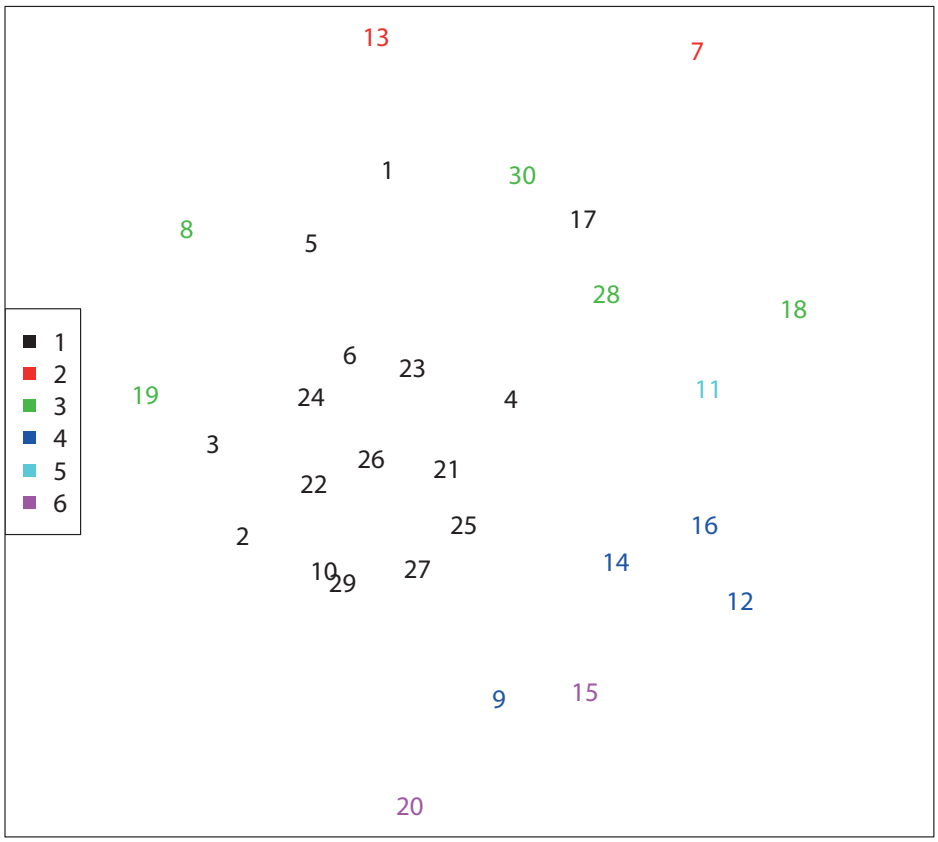

Figure 6: Sammon map of Euclidean distances of scaled `case.study.psychiatrist` data set

Although we prefer assessing cluster stability with the proportion of times pairs of objects cluster together across bootstrap replicates rather than computing the Jaccard similarity with the original clustering solution, the **fpc** package provides additional features convenient to utilise. Hennig (2007) discuss alternative resampling options which can be accessed by selecting `clustering.func = fpc.clusterboot` in the call to function `clusboot()`. The additional argument `clustermethod` now fulfills the role of `clustering.func` while `bootmethod` specifies the resampling method. The default `bootmethod = "boot"` is equivalent to the vanilla clusboot option, while other options available are `"subset", "noise", "jitter", "bojit"` which allows for selecting random subsets, replacing a certain percentage of the points by random noise, add random noise to all points or a nonparametric bootstrap followed by adding noise. For a detailed discussion on the different options and comparison of the methods, see Hennig (2007) and the helpfile of the `clusterboot()` function in package **fpc**. Using `clustering.func = fpc.clusterboot` in the call to function `clusboot()` will perform the applicable resampling, followed by the clustering of each replicate and computation of proportion of times pairs of objects cluster together as detailed above.

The **ClusBoot** package only provides one wrapper function as illustration, specifically to emphasize that the focus is on assessing cluster stability, rather than providing a series of recommended clustering functions / algorithms to evaluate. The **pfc** package provides several wrapper functions which can be accessed when using `clustering.func = fpc.clusterboot`. Some of these are illustrated in Table 1 below.

The average bootstrap-silhouette values in Table 1 should not be viewed as an indication

Table 1: Using `clustering.func = fpc.clusterboot` to access different resampling options and interface functions from **fpc**

Option	call	Average silhouette width
Complete linkage with 'vanilla' `clusboot`	`clusboot ( scale(case.study.psychiatrist[,-15]), B = 1000, k = 6, clustering.func = complete.linkage)`	0.515
Complete linkage with bootstrap resampling with fpc interface function	`clusboot( scale(case.study.psychiatrist[,-15]), B = 1000, k = 6, clustering.func = fpc.clusterboot, clustermethod = hclustCBI, method = "complete", multipleboot = TRUE)`	0.505
Complete linkage with subset resampling	`clusboot( scale(case.study.psychiatrist[,-15]), B = 1000, k = 6, clustering.func = fpc.clusterboot, bootmethod = "subset", clustermethod = hclustCBI, method = "complete", multipleboot = TRUE)`	0.508
Complete linkage with subset resampling, changing default `subtuning` parameter	`clusboot( scale(case.study.psychiatrist[,-15]), B = 1000, k = 6, clustering.func = fpc.clusterboot, bootmethod="subset", subtuning = 20, clustermethod = hclustCBI, method = "complete", multipleboot = TRUE)`	0.497
Complete linkage with noise resampling	`clusboot( scale(case.study.psychiatrist[,-15]), B = 1000, k = 6, clustering.func = fpc.clusterboot, bootmethod = "noise", clustermethod = hclustCBI, method = "complete", multipleboot = TRUE)`	0.315
Complete linkage with jitter resampling	`clusboot( scale(case.study.psychiatrist[,-15]), B = 1000, k = 6, clustering.func = fpc.clusterboot, bootmethod="jitter", clustermethod = hclustCBI, method = "complete" multipleboot = TRUE)`	0.811
Complete linkage with bootstrap resampling followed by jittering	`clusboot( scale(case.study.psychiatrist[,-15]), B = 1000, k = 6, clustering.func = fpc.clusterboot, bootmethod = "bojit", clustermethod = hclustCBI, method="complete", multipleboot = TRUE)`	0.468
K-means with bootstrap resampling with fpc interface function	`clusboot( scale(case.study.psychiatrist[,-15]), B = 1000, k=6, clustering.func = fpc.clusterboot, clustermethod = kmeansCBI, multipleboot = TRUE)`	0.237
PAM with bootstrap resampling with fpc interface function	`clusboot( scale(case.study.psychiatrist[,-15]), B = 1000, k=6, clustering.func = fpc.clusterboot, clustermethod = pamkCBI, multipleboot = TRUE)`	0.261
Spectral clustering with bootstrap resampling with fpc interface function	`clusboot( scale(case.study.psychiatrist[,-15]), B = 1000, k=6, clustering.func = fpc.clusterboot, clustermethod = speccCBI, multipleboot = TRUE)`	0.230

of how well the method fairs in general. It is simply an illustration of different resampling methods and clustering interface functions. Note that the default argument `multipleboot = FALSE` does not yield comparable results with the default option of `clusboot`. Furthermore the argument `scaling = TRUE` in the function `hclustCBI()` can affect results. Since the input data set in Table 1 is scaled it has no effect in this case.

The complete linkage hierarchical cluster in the previous section conveniently provides a clustering tree representation of the cluster analysis. A completely different way of performing cluster analysis would be for instance to use model-based clustering as provided in the R package **mclust** (Scrucca, Fop, Murphy, and Raftery 2016). Similar to the analysis of Scrucca *et al.* (2016) a three cluster analysis of the wine data set contained in the **gclus** package is performed. The data set contains 13 measurements on the chemical analysis of 178 wines from three different cultivars grown in the same region in Italy. The following code is adapted from Scrucca *et al.* (2016) to perform the cluster analysis.

```
data(wine, package = "gclus")
Class <- factor(wine$Class, levels = 1:3,
    labels = c("Barolo", "Grignolino", "Barbera"))
X <- data.matrix(wine[,-1])
mod <- Mclust(X, G=3)

----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust VVE (ellipsoidal, equal orientation) model with 3 components:

 log-likelihood   n  df       BIC       ICL
      -3015.335 178 158 -6849.391 -6850.734

Clustering table:
 1  2  3
59 69 50
```

Several models with different constraints on the population covariance matrices are fitted and based on the BIC criterion, the best-fitting three cluster model is the ellipsoidal model with equal orientation for the three clusters.

In order to apply `clusboot()` to the `mclust` clustering, a wrapper function provides the clustering function returning only the clustering vector.

```
mclust.clustering <- function (X, G, model)
    {  Mclust(X, G,  modelNames=model)$classification }
plot (clusboot(X, G=3, model="VVE", clustering.func=mclust.clustering),
      pch=3, col=c("cadetblue","darkkhaki","brown"))
```

The default option `show.silhouette = TRUE` in the call to `plot.clusboot()` uses the individual bootstrap-silhouette values to indicate the stability with which each of the samples are clustered across the bootstrap replicates. A large bootstrap-silhouette value indicates stable clustering across replicates, but typically the researcher is interested in those samples that do not cluster well. The samples sizes in Figure 7 is inversely related to the indivdual boostrap-silhouette values. The MDS plot of the $\mathbf{P}$ matrix in the left panel clearly show a substantial difference between the blue and red clusters. For the green cluster, one sample clusters with the red cluster fairly often, while the large size of the red sample that clusters

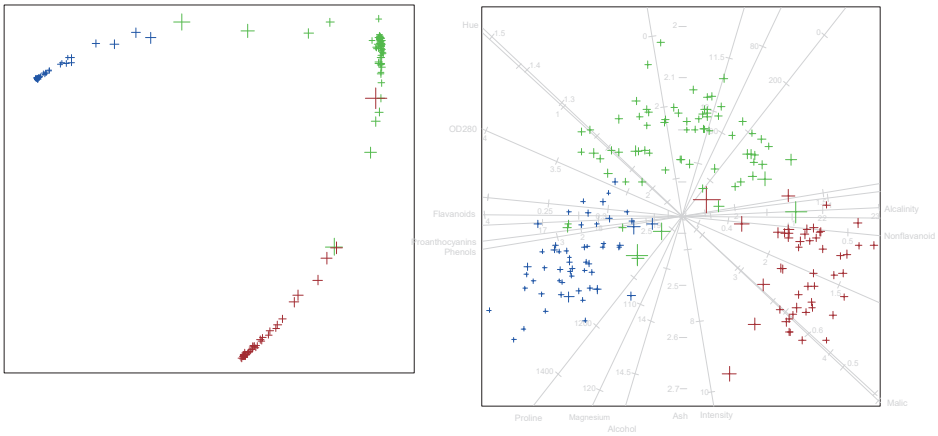

Figure 7: MDS plot of bootstrap analysis of three cluster mclust solution of the wine data set

with the green cluster indicates it clusters more with this cluster. In the right panel of Figure 7 a PCA biplot (see (Gower, Lubbe, and le Roux 2011)) is provided for visual validation of the clustering outcome. Essentially it is a classical scaling MDS of the Euclidean distances between samples, with addition of axes to provide information on the variables. From this plot it is clear that it is the samples located on the outer edges of the clusters, mainly in areas where the clusters overlap, that show the least amount of stability.

4. A closer look at bootstrap-silhouette values

In this section simulated data with "known truth" will be used to compare ClusBoot to other stability methods. The following methods available in R are compared:

- Clusterwise bootstrap assessment by Hennig (2007), implemented in **fpc**. The `bootmean` component of the `clusterboot` function provides a value for each cluster. In this simulation study the mean across clusters is used as overall stability measure.
- Bootstrap Jaccard by Yu *et al.* (2019), implemented in **bootcluster**. The function `stability` is only applicable to kmeans clustering and provide the measure in the component `overall`.
- Rand index partitioning and subsampling validation by Nieweglowski (2023), implemented in **clv**. The stability for each of B pairs of data subsets for partitioning is provided by the function `cls.stab.sim.ind` with argument `sim.ind.type = "rand"`. The mean across repetitions is used as stability measure.
- Dot product partitioning and subsampling validation by Ben-Hur and Guyon (2003), implemented in **clv**. The stability for each of B pairs of data subsets for partitioning is provided by the function `cls.stab.sim.ind` with argument `sim.ind.type = "dot"`. The mean across repetitions is used as stability measure.
- Stability-based validation by Lange *et al.* (2004), implemented in **clv**. The stability for each of B pairs of data subsets for partitioning is provided by the function `cls.stab.sim.ind` with argument `sim.ind.type = "sim"`. The mean across repetitions is used as stability measure.
- Bootstrap method in **ClusBoot** with overall stability measure `avg.sil.width`.

A total of 50 data sets are simulated with three clusters in two dimensions, and the addition of 5 noise samples.

Cluster 1: 50 samples from a $N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$.

Cluster 2: 80 samples from a $N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$.

Cluster 3: 100 samples Uniformly distributed on $\mathbf{a} \times \mathbf{b}$.

Two noise samples are generated from a $N(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$

and another three noise samples from a $N(\boldsymbol{\mu}_4, \boldsymbol{\Sigma}_4)$.

where $\boldsymbol{\mu}_1' = \begin{pmatrix} -4 & 2 \end{pmatrix}$, $\boldsymbol{\mu}_2' = \begin{pmatrix} 0 & 5 \end{pmatrix}$, $\boldsymbol{\mu}_3' = \begin{pmatrix} -1 & 2 \end{pmatrix}$, $\boldsymbol{\mu}_4' = \begin{pmatrix} 2 & 7 \end{pmatrix}$, $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_4 = 0.1\mathbf{I}_2$, $\boldsymbol{\Sigma}_2 = \begin{pmatrix} 1 & -0.4 \\ -0.4 & 0.2 \end{pmatrix}$, $\boldsymbol{\Sigma}_3 = 0.5\mathbf{I}_2$, $\mathbf{a} = \begin{pmatrix} -6 & -2 \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} 3 & 7 \end{pmatrix}$.

Five clustering solutions were obtained: hierarchical clustering with average linkage, k-means, pam and two models using **mclust**, EII (spherical, equal volume clusters) and VVV (ellipsoidal, varying volume, shape, and orientation clusters). The function `cls.stab.sim.ind` only makes provision for the non-model based methods. To assess how well the clustering solution captures the true cluster structure the adjusted rand index Hubert and Arabie (1985) was computed and is shown in the top panel of Figure 8.

It is clear that hierarchical clustering has the poorest match to the true clusters while the VVV model clustering fits the true clusters best. After performing $B = 100$ replicates for each of the stability measures, the results for the different measures are very similar, but do not fully reflect the matching of the clustering solution to the true clusters. All methods except hierarchical clustering appear to be more or less equally stable, i.e. finding the same (even sometimes partially incorrect) clusters over and over through repetitions. Interestingly, the only two methods that allow for stability testing on model based clusters, **ClusBoot** and **fpc** both suggest slighly less stable performance for model VVV than model EII.

Similar to the silhouette width of Rousseeuw (1987), maximising some measure of the quality of the clustering can be used to select the optimal number of clusters. Many of the resampling and partitioning and subsampling methods discussed in Liu *et al.* (2022) have been developed not so much for describing cluster stability, but for selecting the optimal number of clusters. Liu *et al.* (2022) apply different stability methods on the iris data to determine the optimal number of clusters and report that $k = 3$ was chosen by the methods of Dudoit and Fridlyand (2003), Yu *et al.* (2019) and Tibshirani and Walther (2005) while $k = 2$ was chosen by the methods of Fang and Wang (2012) and Liu *et al.* (2019) while Hennig (2007) chose a more complex model with $k = 5$.

In Figure 9 different clustering algorithms were applied to the iris data set and $B = 100$ bootstrap replicates performed with the `clusboot()` function to find the maximum average bootstrap-silhouette width for each method. Complete linkage hierarchical clustering selects $k = 4$ clusters, while all the other methods indicate $k = 2$ provides the most stable solution.

5. Concluding remarks

The **ClusBoot** package provides any practitioner with some approaches to evaluate the true clustering structure in their data with a bootstrap analysis. Detailed information on how each observation tends to cluster with every other observation can be found in the matrix of joint clustering proportions. In cases with a large number of observations this can be very cumbersome to interpret. The bootstrap-silhouette values provides a summary of how tight knit each cluster is, how stable each sample is clustered and an overall measure of stability. Through the similation study it was established that the variety of methods for assessing overall stability are very similar, but **ClusBoot** provides an additional MDS display of the bootstrap analysis which can identify a few individuals that might be "problem samples" in that they do not consistently cluster with the same core cluster samples. This plot, in addition to biplots which can be utilised as multivariate scatter plots, provide valuable insight in the clustering solution.

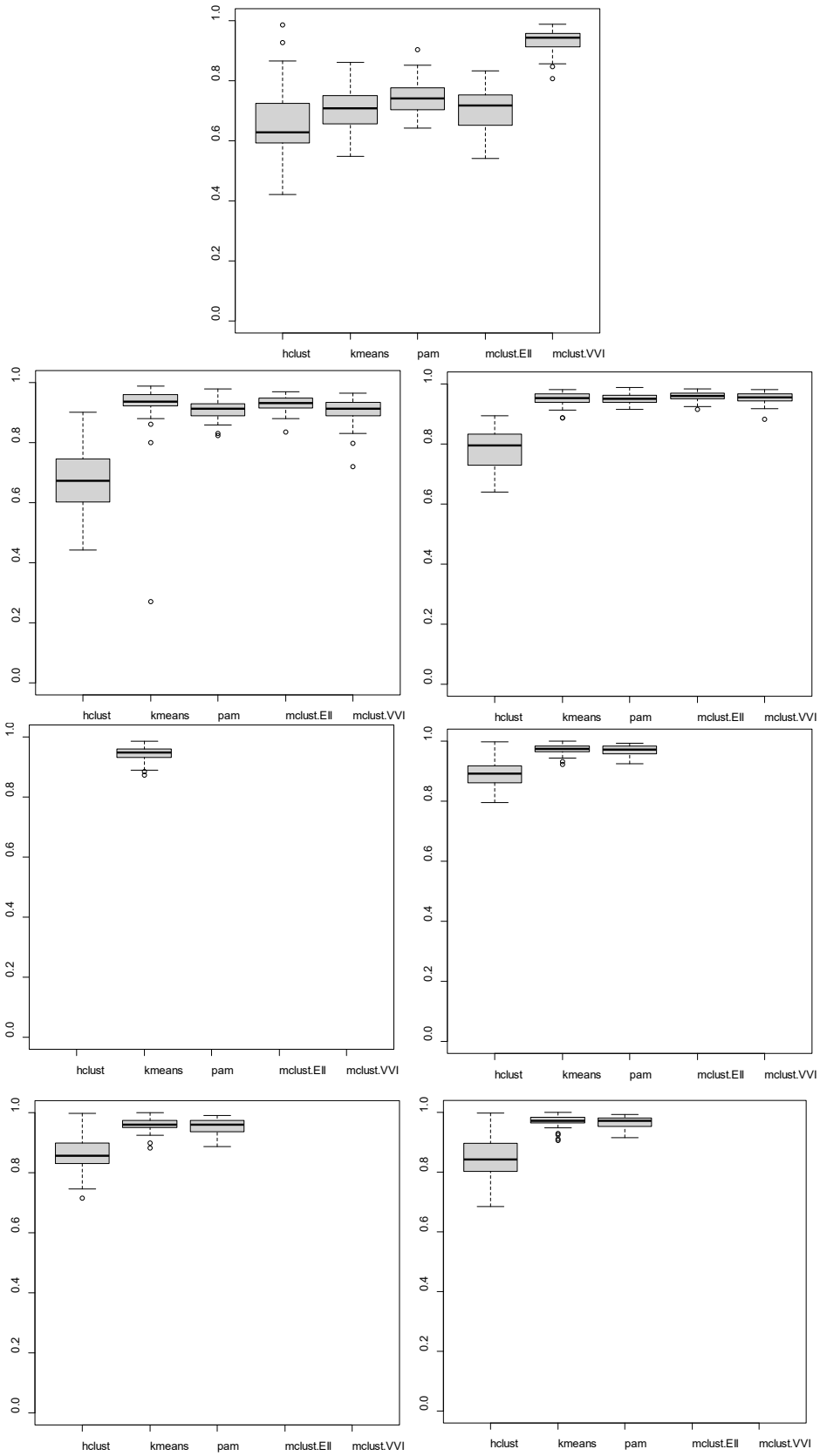

Figure 8: Simulation results. Top panel: Adjusted rand index comparing clustering solution to true cluster structure. Row 2 left: Average bootstrap-silhouette values obtained with **ClusBoot**. Row 2 right: Mean of Jaccard based stability values obtained with **fpc**. Row 3 left: Bootstrap Jaccard obtained with **bootcluster**. Row 3 right: Rand index validation based on partitioning and subsampling obtained with **clv**. Row 4 left: Dot product validation obtained with **clv**. Row 4 right: Stability-based validation obtained with **clv**.

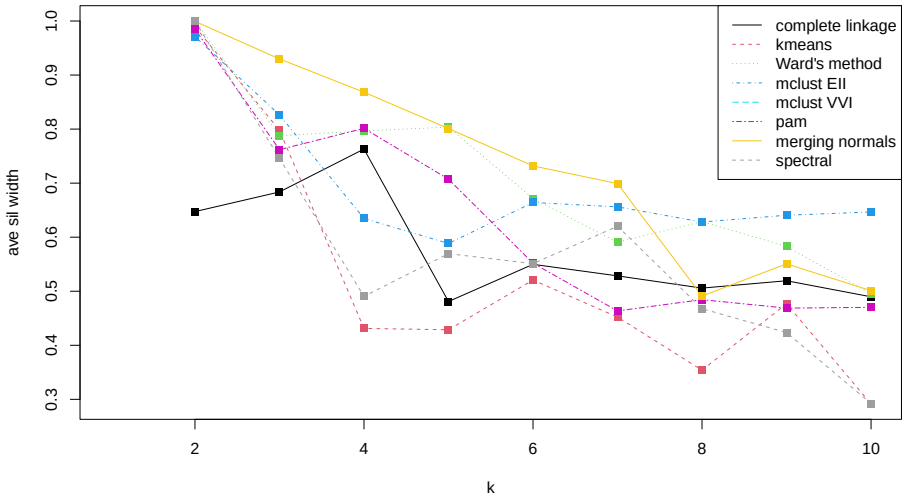

Figure 9: Maximising average bootstrap-silhouette width to select the optimal number of clusters for the iris data set

Finding true clusters in a data set is a very difficult problem. With the **ClusBoot** package the user is able to evaluate the extent to which some objects cluster very well together while there might be considerably more uncertainty associated with some of the clusters or some individuals in one or more of the clusters.

Acknowledgments

This work is based upon research supported by the National Research Foundation (NRF) of South Africa. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author and therefore the NRF does not accept any liability in regard thereof.

The author would like to thank two anonymous referees for very helpful input which greatly improved the manuscript and package functionality.

References

Becker AR, Chambers JM, Wilks AR (1988). *The New S Language*. Wadsworth & Brooks/Cole.

Ben-Hur A, Elisseeff A, Guyon I (2002). "A Stability Based Method for Discovering Structure in Clustered Data." *Pacific Symposium on Biocomputing*, **7**, 6–17. doi:10.1142/9789812799623_0002.

Ben-Hur A, Guyon I (2003). "Detecting Stable Clusters using Principal Component Analysis." *Functional Genomics: Methods and Protocols*, pp. 159–182. doi:10.1385/1-59259-364-X:159.

Bertrand P, Mufti GB (2006). "Loevinger's Measures of Rule Qaulity for Assessing Cluster Stability." *Computational Statistics and Data Analysis*, **50**(4), 992–1015. doi:10.1016/j.csda.2004.10.012.

Dudoit S, Fridlyand J (2002). "A Prediction-based Resampling Method for Estimating

the Number of Clusters in a Dataset." *Genome Biology*, **3**(7), 1–21. doi:10.1186/gb-2002-3-7-research0036.

Dudoit S, Fridlyand J (2003). "Bagging to Improve the Accuracy of a Clustering Procedure." *Bioinformatics*, **19**(9), 1090–1099. doi:10.1093/bioinformatics/btg038.

Fang Y, Wang J (2012). "Selection of the Number of Clusters via the Boostrap Method." *Computational Statistics and Data Analysis*, **56**(3), 468–477. doi:10.1016/j.csda.2011.09.003.

Gower JC, Lubbe S, le Roux NJ (2011). *Understanding Biplots*. Wiley.

Hennig C (2007). "Cluster-wise Assessment of Cluster Stability." *Computational Statistics and Data Analysis*, **52**(1), 258–271. doi:10.1016/j.csda.2006.11.025.

Hennig C (2015). "What are the True Clusters?" *Pattern Recognition Letters*, **64**, 53–62. doi:10.1016/j.patrec.2015.04.009.

Hubert L, Arabie P (1985). "Comparing Partitions." *Journal of the Classification*, **2**, 193–218.

Jain AK, Moreau JV (1987). "Bootstrap Technique in Cluster Analysis." *Pattern Recognition Letters*, **20**, 547–568.

Kerr MK, Churchill GA (2001). "Bootstrapping Cluster Analyis: Assessing the Reliability of Conclusions from Microarray Experiments." *Proceedings of the National Academy of Sciences*, **98**, 8961–8965.

Lange T, Roth V, Braun ML, Buhmann JM (2004). "Stability-based Validation of Clustering Solutions." *Neural Computation*, **16**(6), 1299–1323.

Levine E, Domany E (2001). "Resampling Method for Unsupervised Estimation of Cluster Validity." *Neural Computation*, **13**(11), 2573–2593.

Liu J, Seo B, Lin L (2019). "Optimal Transport, Mean Partition, and Uncertaining Assessment in Cluster Analysis." *Statistical Analysis and Data Mining: The ASA Data Science Journal*, **12**(5), 359–377. doi:10.1002/sam.11418.

Liu T, Yu H, Blair RH (2022). "Stability Estimation for Unsupervised Clustering: A Review." *Wiley Interdisciplinary Reviews: Computational Statistics*, **14**(6), e1575. doi:10.1002/wics.1575.

Mechelen IV, Boeck PD (1989). "Implicit Taxonomy in Psyciatric Diagnosis: A Case Study." *Journal of Social and Clinical Psychology*, **8**, 276–287. doi:10.1521/jscp.1989.8.3.276.

Nemec NFL, Brinkhurst RO (1988). "Using the Bootstrap to Assess Statistical Significance in the Cluster Analysis of Species Abundance Data." *Canadian Journal of Fisheries and Aquatic Sciences*, **45**, 965–970. doi:10.1139/f88-118.

Nieweglowski L (2023). *clv: Cluster Validation Techniques*. R package version 0.3-2.3, URL https://CRAN.R-project.org/package=clv.

Pillar VD (1999). "The Bootstrapped Ordination Re-examined." *Journal of Vegetation Science*, **10**, 895–902. doi:10.2307/3237314.

Rousseeuw PJ (1987). "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis." *Journal of Computational and Applied Mathematics*, **20**, 53–65.

Scrucca L, Fop M, Murphy TB, Raftery AE (2016). "mclust 5: Clustering, Classification and Density estimation Using Gaussian Finite Mixture Models." *The R Journal*, **8**, 205–233.

Steinley D (2008). "Stability Analysis in K-means Clustering." *British Journal of Mathematical and Statistical Psychology*, **61**(2), 255–273. doi:10.1348/000711007X184849.

Tibshirani R, Walther G (2005). "Cluster Validation by Prediction Strength." *Journal of Computational and Graphical Statistics*, **14**(3), 511–528. doi:10.1198/106186005X59243.

Ullmann T, Hennig C, Boulesteix AL (2022). "Validation of Cluster Analysis Results on Validation Data: A Systematic Framework." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **12(3)**(3), e1444. doi:10.1002/widm.1444.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4 edition. Springer.

Yu H, Chapman B, Florio AD, Eischen E, Gotz D, Jacob M, Blair RH (2019). "Bootstrapping Estimate of Stability for Clusters, Observations and Module Selection." *Computational Statistics*, **34**(1), 349–372. doi:10.1007/s00180-018-0830-y.

Affiliation:

Sugnet Gardner-Lubbe
MuViSU (Centre for Multi-dimensional Data Visualisation)
Department of Statistics and Actuarial Science
Stellenbosch University
Private Bag X1, Matieland
7602, South Africa
E-mail: slubbe@sun.ac.za

Austrian Journal of Statistics http://www.ajs.or.at/
published by the Austrian Society of Statistics http://www.osg.or.at/
Volume 53 *Submitted:* 2020-05-18
2024 *Accepted:* 2023-10-23