ELSEVIER Computational Statistics & Data Analysis 30 (1999) 381–392

COMPUTATIONAL STATISTICS & DATA ANALYSIS

Displaying a clustering with CLUSPLOT

Greet Pison*, Anja Struyf, Peter J. Rousseeuw

Department of Mathematics and Computer Science, U.I.A., Universiteitsplein 1, B-2610 Antwerp, Belgium

Received 1 November 1997; received in revised form 1 September 1998; accepted 30 November 1998

Abstract

In a bivariate data set it is easy to represent clusters, e.g. by manually circling them or separating them by lines. But many data sets have more than two variables, or they come in the form of inter-object dissimilarities. There exist methods to partition such a data set into clusters, but the resulting partition is not visual by itself. In this paper we construct a new graphical display called CLUSPLOT, in which the objects are represented as points in a bivariate plot and the clusters as ellipses of various sizes and shapes. The algorithm is implemented as an S-PLUS function. Several options are available, e.g. labelling of objects and clusters, drawing lines connecting clusters, and the use of color. We illustrate this new tool with several examples. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Cluster analysis; Discriminant analysis; Multidimensional scaling; Principal components; Statistical software

1. Introduction

There are two main types of clustering methods. The *hierarchical* methods construct a dendrogram, which is a tree of which the leaves are the data objects and the branches can be seen as clusters. On the other hand, a *partitioning* method divides the data into k nonoverlapping clusters, so that objects of the same cluster are close to each other and objects of different clusters are dissimilar.

The output of a partitioning method is simply a list of clusters and their objects, which may be difficult to interpret. It would therefore be useful to have a graphical

* Corresponding author Tel.: +32(0)3/820.24.19; fax: +32(0) 3/820.24.21. E-mail: http://win-www.uia.ac.be/u/statis/index.html.

0167-9473/99/$ – see front matter © 1999 Elsevier Science B.V. All rights reserved.
PII: S 0167-9473(98)00102-9

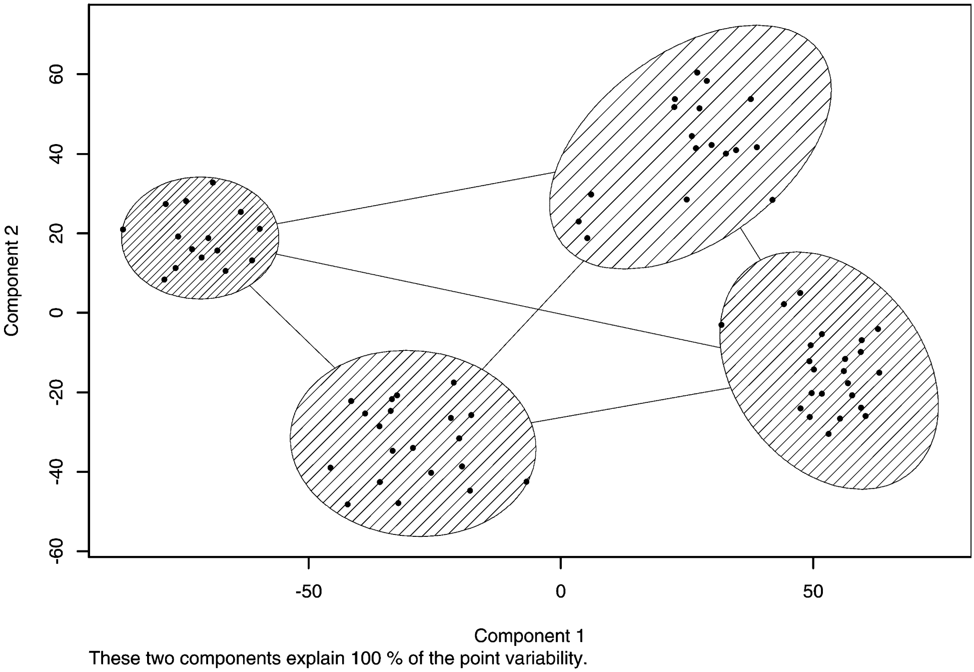

Fig. 1. Clusplot of the Ruspini data (75 points in 2 dimensions).

display which describes the objects with their interrelations, and at the same time shows the clusters. This would allow us to picture the size and shape of the clusters, as well as their relative position. Following a suggestion on p. 318 of (Kaufman and Rousseeuw, 1990, henceforth KR) we will construct such a display, called CLUSPLOT, and an algorithm for its implementation.

For instance, let us consider the bivariate data set of Ruspini (1970) which contains $n = 75$ objects, and partition it into $k = 4$ clusters. For this we have used the Partitioning Around Medoids (PAM) method of [KR], but of course also other clustering methods can be applied. Then CLUSPLOT uses the resulting partition, as well as the original data, to produce Fig. 1. The ellipses are based on the average and the covariance matrix of each cluster, and their size is such that they contain all the points of their cluster. This explains why there is always an object on the boundary of each ellipse. It is also possible to draw the *spanning ellipse* of each cluster, i.e. the smallest ellipse that covers all its objects. The spanning ellipse can be computed with the algorithm of Titterington (1976).

To get an idea of the distances between the clusters, we can draw segments of the lines between the cluster centers. In the plot, the shading intensity is proportional to the density of the cluster, i.e. its number of objects divided by the area of the ellipse.

For higher-dimensional data sets we apply a dimension reduction technique before constructing the plot, as described in Section 2. Section 3 concentrates on dissimilarity data, where we will represent the objects as bivariate points by means of multidimensional scaling. Section 4 describes the implementation of CLUSPLOT in S-Plus, with the available options. Section 5 formulates some conclusions and several proposals for further extensions.

2. Higher-dimensional data

Let us take a p-dimensional data set $X = \{(x_{i1}, x_{i2}, \ldots, x_{ip});\ i = 1, \ldots, n\}$. We can reduce the dimension of the data by principal component analysis (PCA), which yields a first component with maximal variance, then a second component with maximal variance among all components perpendicular to the first, and so on. The principal components lie in the directions of the eigenvectors of a scatter matrix, which can be the classical covariance matrix or the corresponding correlation matrix. Another possibility is to start from a robust scatter matrix (which can resist the effect of outliers), such as the minimum volume ellipsoid and the minimum covariance determinant estimators of Rousseeuw (1984). A fast algorithm for the latter was recently constructed in (Rousseeuw and Van Driessen, 1997).

After carrying out the PCA, we plot the first two principal components and list the percentage of the total variance explained by them.

The CLUSPLOT display is then the bivariate plot of the objects relative to the first two principal components, and the clusters are again represented as ellipses. Even in Fig. 1, where the data are two-dimensional from the start, CLUSPLOT has displayed the data relative to the principal components rather than to the original axes. The reason is that component 1, with the largest dispersion, is then plotted on the longest axis (i.e., the horizontal one).

Example. This example uses a data set from a Belgian factory of nuclear fuel (Rousseeuw et al., 1996). The variables are the concentration of four isotopes, in 45 batches of plutonium. In CLUSPLOT we can choose to apply principal component analysis on either the correlation matrix or the covariance matrix. Here the latter is used. The clusplots in Fig. 2 give a two-dimensional representation of the objects and the spanning ellipses of the clusters. Note that the boundary of a spanning ellipse always contains several objects. The distance between two clusters is represented as a line connecting the cluster centers. Objects belonging to different clusters are plotted with different characters. At the bottom of both plots, we see that 99.94% of the point variability is explained by the first two principal components. This plot is thus a faithful representation of the four-dimensional data. The cluster numbers appear in the plot. Fig. 2(a) shows the two clusters obtained by the k-means method for $k = 2$. We observe that cluster 1 is essentially bimodal. The k-means clustering for $k = 3$ is displayed in Fig. 2(b), in which the clusters are much more compact. For these data, the clusplots indicate that the clustering with $k = 3$ is preferable to that with $k = 2$.

Moreover, the clusplot may also point to defects of a clustering. For instance, suppose that we have applied a clustering method which has ‘misclassified’ an object

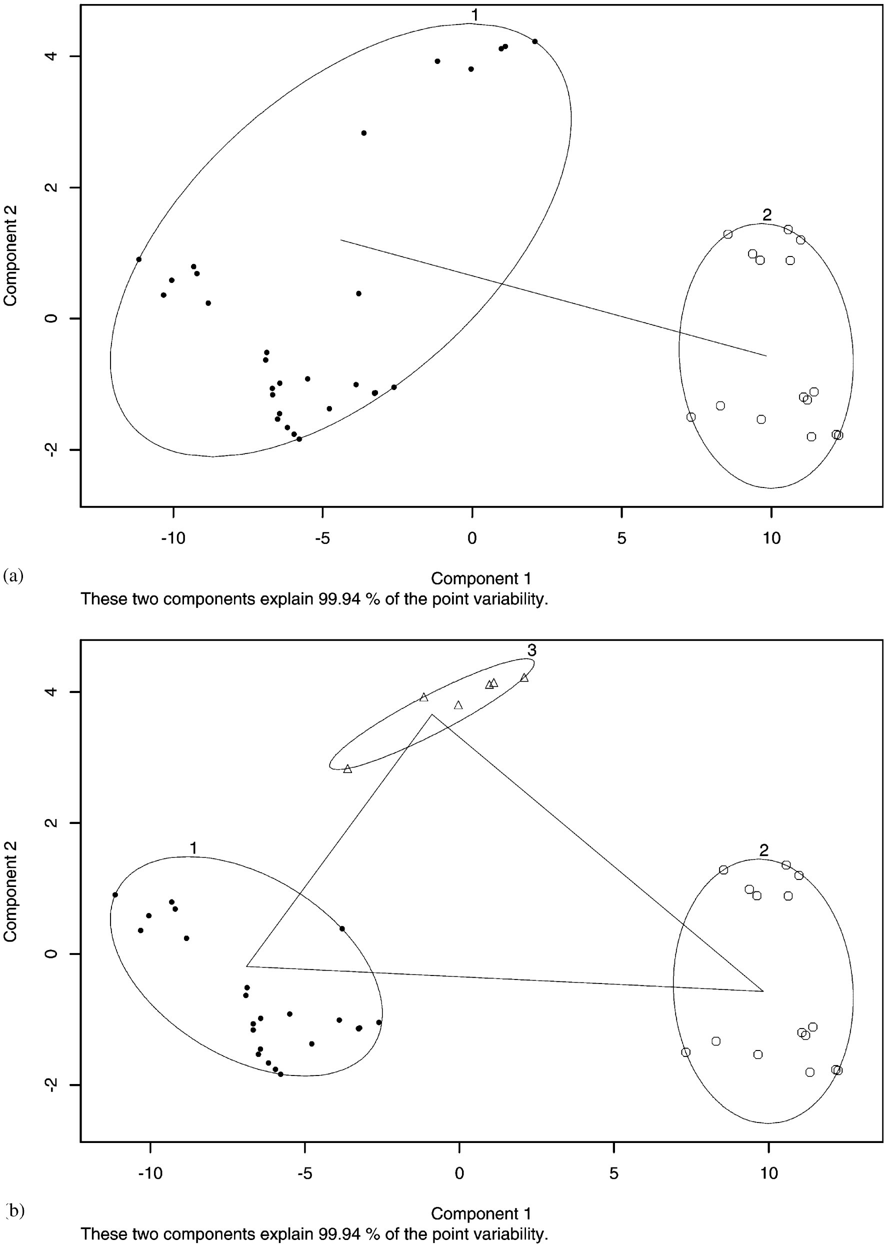

Fig. 2. Clusplot of the plutonium data (45 points in 4 dimensions) for (a) two clusters; and (b) three clusters.

of cluster 2 as belonging to cluster 1. This yields the clusplot in Fig. 3, which alerts us to the problem and indicates which object (in the lower right corner) is responsible. Therefore, the clusplot may serve as a diagnostic tool for the validity of a clustering.

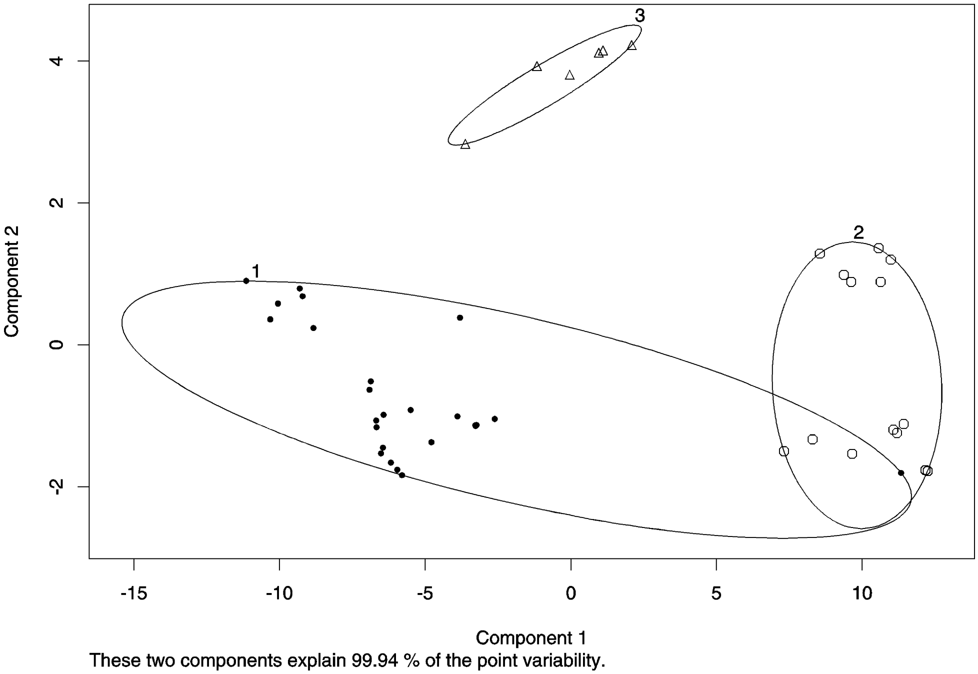

Fig. 3. Clusplot of the plutonium data with three clusters, in which one point was misclassified.

3. Dissimilarity data

When the data set consists of inter-object dissimilarities (together forming an n by n matrix D), another method will be used to obtain a two-dimensional plot of the n objects.

Dissimilarities are nonnegative numbers $d(i,j)$ that are small when i and j are 'near' to each other and that become large when i and j are very different. They have two properties:

- $d(i,i) = 0$,
- $d(i,j) = d(j,i)$.

Dissimilarities between two objects can be computed in different ways, e.g. when the data contain nominal or ordinal variables, but also subjective measures of discordance are allowed. We say that the matrix $D=(d(i,j); 1 \leq i,j \leq n)$ is metric if, in addition to the above two properties, also the triangle inequality

- $d(i,j) \leq d(i,h) + d(h,j)$

holds. In that case, the dissimilarities are called distances.

In general, a multidimensional scaling (MDS) method constructs a set of n points, characterized by their coordinates relative to some axes, such that the Euclidean distances between these n points approximate the original dissimilarities. The

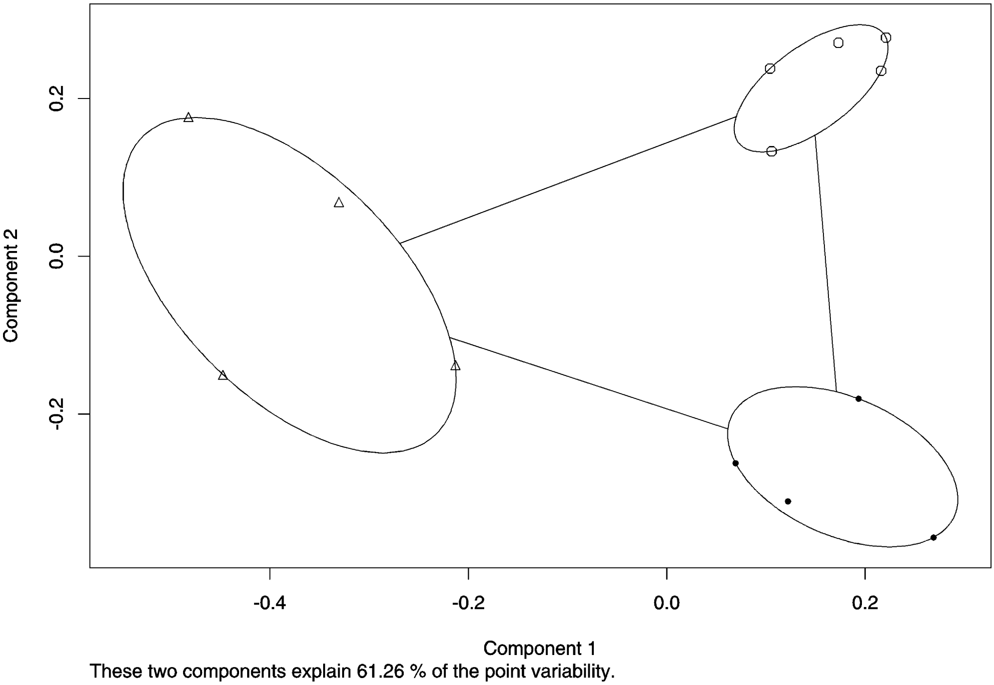

Fig. 4. Clusplot of the Harman dissimilarity data.

approximation will be best when the dissimilarity matrix D was metric already. When D is not metric, we can add a constant c to all off-diagonal entries of D so that the resulting matrix becomes metric. Also note that an MDS method yields components such that the first component explains as much variability as possible, the second component explains as much of the remaining variability as possible, and so on.

CLUSPLOT applies an MDS method to D (which may or may not be metric), and then displays the first two components. The percentage of point variability explained by these two components (relative to all components) is again listed below the plot. Then ellipses are drawn around the clusters, as in the preceding sections.

Example. Let us consider the real data set of Harman (1967) which contains dissimilarities between 13 psychological tests. Based on these dissimilarity data and the output of the clustering algorithm PAM with $k = 3$, CLUSPLOT yields Fig. 4 which clearly shows the clusters. We have chosen to plot the spanning ellipses, and to use different plotting characters for the objects of different clusters.

4. Implementation and availability

We have implemented CLUSPLOT as an S-PLUS function, taking full advantage of the powerful statistical, numerical and graphical tools available in the S-PLUS environment. In particular, we could use the intrinsic functions `princomp` for PCA and `cmdscale` for MDS. Moreover, S-PLUS has incorporated several clustering algorithms, including (since version 3.4) the functions `daisy`, `pam`, `fanny` and `clara` implemented by (Struyf et al., 1997; henceforth SHR).

The S-PLUS call to `clusplot` is of the form

```
clusplot (x, clus, diss = T, cor = T, stand = F, lines = 2,
shade = F, color = F, labels = 0, plotchar = T, span = T)
```

It is always necessary to specify the first three arguments, whereas the others may be left out, in which case the function will use their default settings. Let us look at all the arguments in turn.

- `x`: data matrix or dataframe, or dissimilarity matrix. In the latter case `x` can be a symmetric matrix or the output of the intrinsic S-PLUS functions `daisy` or `dist`. Note that the user does not have to specify n, the number of objects, because clusplot derives n from the size of `x`.
- `clus`: a vector of length n representing a clustering of `x`. For each of the n objects the vector lists the number or name of the cluster to which it has been assigned. We can obtain `clus` as the clustering component of the output of the S-PLUS functions `pam`, `fanny` or `clara` implemented and described in [SHR].
- `diss`: When `x` is a dissimilarity matrix then `diss` must have the value TRUE, else `diss` has the value FALSE.

The optional arguments are:

- `cor`: This argument is only relevant when `diss = F`, that is when `x` is a dataframe or a data matrix. The principal components may be based on the covariance matrix (`cor = F`) or on the correlation matrix (`cor = T`). The default is `cor = T`.
- `stand`: When `stand = T`, the n points in the two-dimensional plot are standardized to have zero location and unit spread. The default is `stand = F`.
- `lines`: This argument can have the values 0, 1 and 2 and determines which lines are drawn to connect the ellipses. In case the ellipses E_1 and E_2 overlap on the line through their centers m_1 and m_2, no line is plotted. Otherwise, the length of the line depends on the value of the lines argument:
  If `lines = 0` no connecting lines are drawn.
  If `lines = 1` the line segment between m_1 and m_2 is drawn.
  If `lines = 2` the line segment between the boundaries of E_1 and E_2 is drawn.
  The default is `lines = 2`.
- `shade`: The option `shade = T` specifies that ellipses are shaded in relation to their density. A cluster's density is its number of objects divided by the area of the ellipse. The default is `shade = F`.
- `color`: Ellipses can be colored with respect to their density using the option `color = T`. With increasing density, the colors are light blue, light green, red and purple. The default is `color = F`.
- `labels`: The possible values are 0, 1, 2, 3 and 4. If `labels = 0`, no labels are given. (This is the default.) With the option `labels = 1`, both the points and the ellipses can be identified by clicking on them. When `labels = 2`, the points and

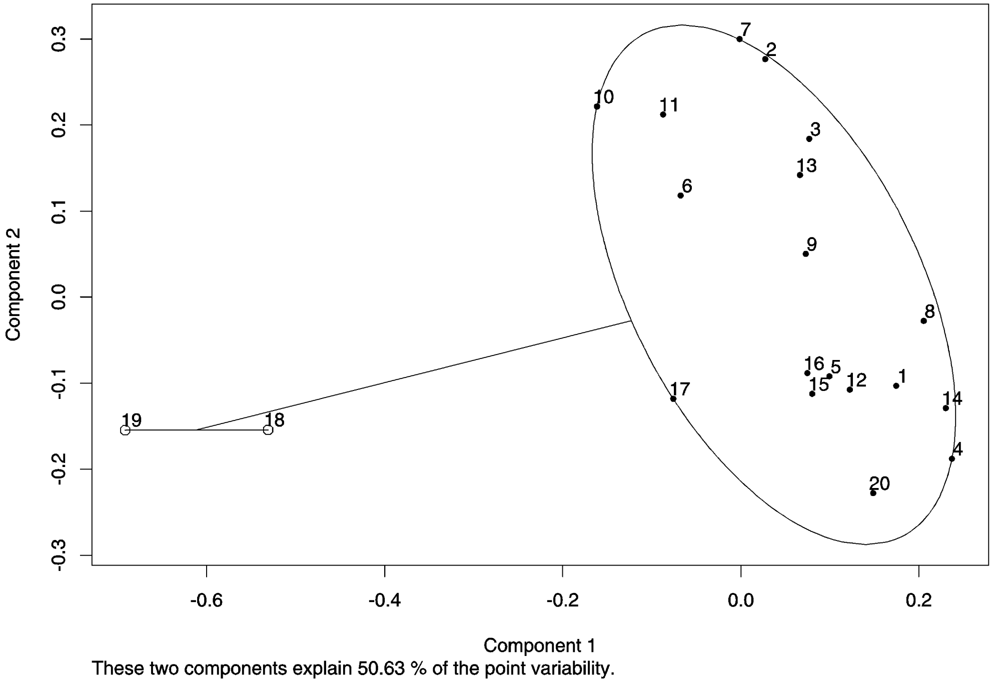

Fig. 5. Clusplot of the Abbot–Perkins dissimilarity data, with a cluster of only 2 points.

ellipses are labelled from the start. When labels = 3 only the points are labelled, and when labels = 4 only the ellipses are labelled.

- plotchar: If you want different plotting characters for objects belonging to different clusters, use the option plotchar = T (this is the default).
- span: This tells clusplot whether to draw ellipses based on the average and covariance matrix of the points in the cluster (span = F), or to draw spanning ellipses (span = T). The latter choice is the default, because it yields more concentrated ellipses.

The function clusplot also takes care of special cases. When a cluster consists of only one point, a tiny circle is drawn around it. When the objects of a cluster fall on a straight line and span = F we obtain a narrow ellipse around them, and if span = T we obtain the exact line segment.

Example. The data set of Abbott and Perkins (1978) lists dissimilarities between twenty student ratings about course organization, the quality of the course text, and the teacher's ability. The clusplot for $k = 2$ is Fig. 5. One of the clusters contains 18 items, while the other only has 2 items. We used the option span = T, hence the line segment between these two objects was drawn. If we had used the option span = F then a narrow ellipse would have appeared. The option labels = 3 reveals that the two objects are numbers 18 and 19, which refer to the utility of the course text. It appears that these items are not very related to the items representing the teacher's performance.

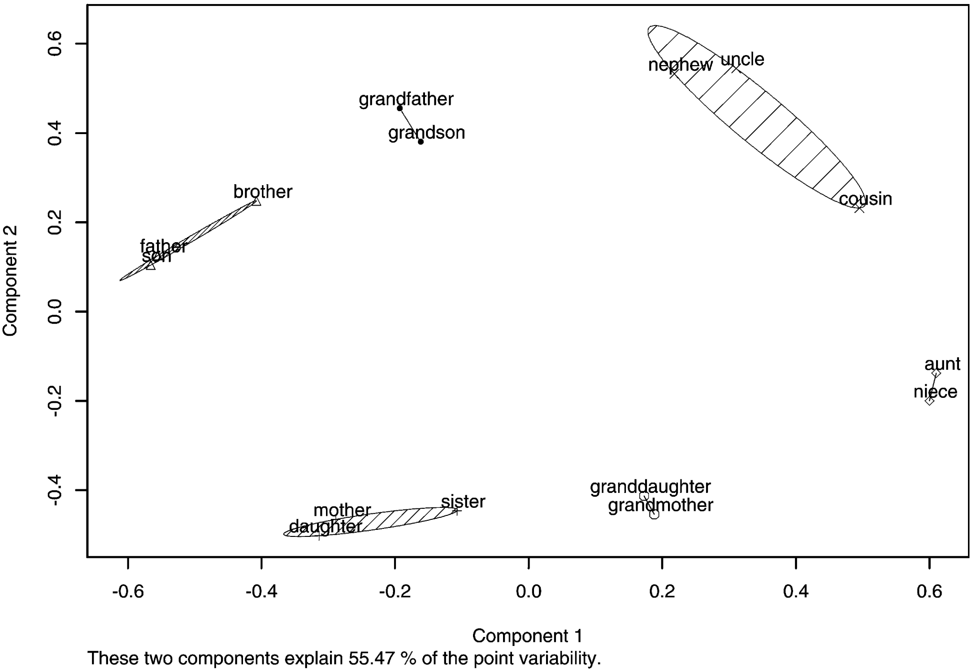

Fig. 6. Clusplot of the Rosenberg dissimilarity data, with many clusters of few objects.

Example. Let us consider an example with many clusters of few objects. The data set of (Rosenberg, 1982) is about 15 words, to be clustered on the basis of some aspect of meaning. The 15 words are grandfather, grandmother, grandson, granddaughter, brother, sister, father, mother, son, daughter, nephew, niece, uncle, aunt and cousin. With the given dissimilarity matrix and the output of pam [SHR], we obtain the clusplot in Fig. 6. Because the option span = T is used, some clusters are shown as line segments. When we use the option labels = 3 it is very clear which persons are grouped together. For instance, we see that gender has a large influence on perceived dissimilarity. The ellipses are shaded according to their density.

The S-PLUS code and helpfile of the function clusplot are available from our website

http://win-www.uia.ac.be/u/statis/index.html.

Questions or remarks about the implementation can be directed to Greet.Pison @uia.ua.ac.be and Anja.Struyf@uia.ua.ac.be. The S-PLUS code runs quite fast, and for span = F the clusplot appears instantaneously. When span = T, clusplot computes the spanning ellipses by Titterington's (1976) iterative algorithm, which takes somewhat longer because of the iteration loop. But even so, all figures in this paper took at most five seconds on our Sun SparcStation 20/514. We also considered an extreme case with 10 000 objects and 3 clusters, for which clusplot took 61 s.

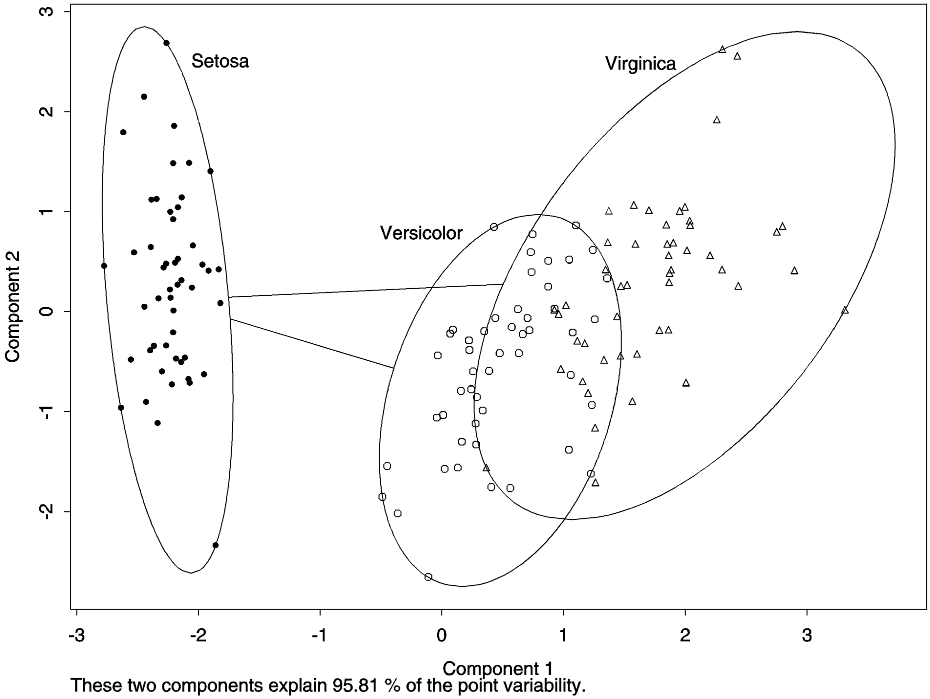

Fig. 7. Clusplot of the Iris training set.

The function `clusplot` runs about five times faster if we take out the code for the spanning ellipse and replace it by Fortran code, which is then compiled and made accessible from within S-Plus. This version of the function `clusplot`, with the accompanying subroutine SPANEL, is also available at the website.

Apart from the `clusplot` function, the website also contains little S-PLUS macros that reproduce the figures of this paper.

5. Conclusions and outlook

We have developed a new S-PLUS function `clusplot` providing a bivariate graphical representation of a clustering partition. Earlier graphical tools include the distance plot (Chen et al., 1974) and the silhouette plot (Rousseeuw, 1987). An important advantage of the `clusplot` is that it shows both the objects and the clusters. The clusplot can also be seen as a generalized and automated version of the taxometric map (Carmichael and Sneath, 1969), which showed the clusters but not the objects.

Clusplots might be extended in several ways, e.g. by replacing each ellipse by the convex hull of all points in the cluster using the algorithm of Eddy (1977), or by the bagplot (Rousseeuw and Ruts, 1997) of the cluster. Another possibility is to draw ellipses based on the Minimum Covariance Determinant estimator, which is easily obtained with the function `cov.mcd` (Rousseeuw and Van Driessen, 1997) built into S-PLUS 4.0.

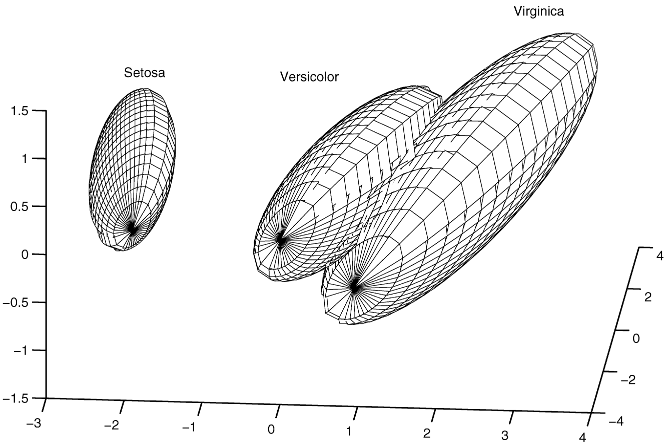

Fig. 8. Three-dimensional clusplot of the Iris training set.

A natural question is whether the clusplot can be generalized to a three-dimensional representation. In this case, the ellipses become ellipsoids. From the computational viewpoint this is not hard to do (e.g. the spanning ellipsoid is computed by the same algorithm as the spanning ellipse), and the computation time increases only slightly. But the current version of S-PLUS is not yet able to render ellipsoids, so other software is needed for producing the actual plot. We think it would be important and useful for S-PLUS to include some graphic routines as building blocks for user-defined high-level graphic functions. Indeed, many statisticians would like more degrees of freedom to develop their own plot functions in the same software they use for the computations. Therefore, we hope that S-PLUS will soon expand in this direction.

Note that the data partition displayed by `clusplot` need not be the result of a clustering algorithm, but may also be given by the user. For instance, in discriminant analysis we have a training sample where we know for each object to which one of k populations it belongs. Applying `clusplot` to the training sample can already tell us something about the relative position of the populations, and how well they are separated. Fig. 7 shows this plot for the well-known training data set of Fisher (1936), consisting of 50 flowers each of the Iris species Setosa, Versicolor, and Virginica. Four variables were measured: sepal length, sepal width, petal length and petal width. We see that the first two principal components explain 96% of the variability, and that Setosa stands apart whereas Versicolor and Virginica overlap. One might wonder whether the separation between the latter training samples would become clearer in three dimensions. Fig. 8 confirms this to be the case. The spanning ellipsoids in this clusplot were rendered by means of MATLAB.

References

Abbott, R.D., Perkins, D., 1978. Development and construct validation of a set of student rating-of-instruction items. Educational and Psychological Measurement 38, 1069–1075.

Carmichael, J.W., Sneath, P.H.A., 1969. Taxometric maps. Systematic Zoology 18, 402–415.

Chen, H., Gnanadesikan, R., Kettenring, J.R., 1974. Statistical methods for grouping corporations. Sankhyā Ser. B 36, 1–28.

Eddy, W.F., 1977. A new convex hull algorithm for planar sets. ACM Trans. Math. Software 3, 398–403.

Fisher, R.A., 1936. The use of multiple measurements in taxonomic problems. Ann. Eugenics 7, Part II, 179–188.

Harman, H.H., 1967. Modern Factor Analysis. University of Chicago Press, Chicago.

Kaufman, L., Rousseeuw, P.J., 1990. Finding Groups in Data. Wiley, New York.

Rosenberg, S., 1982. The method of sorting in multivariate research with applications selected from cognitive psychology and person perception. In: Hirschberg, N., Humphreys, L.G. (Eds.), Multivariate Applications in the Social Sciences. Erlbaum, Hillsdale, NJ, pp. 117–142.

Rousseeuw, P.J., 1984. Least median of squares regression. J. Amer. Statist. Assoc. 79, 871–880.

Rousseeuw, P.J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20, 53–65.

Rousseeuw, P.J., Kaufman, L., Trauwaert, E., 1996. Fuzzy clustering using scatter matrices. Comput. Statist. Data Anal. 135–151.

Rousseeuw, P.J., Ruts, I., 1997. The Bagplot: a bivariate box-and-whiskers plot, submitted for publication.

Rousseeuw, P.J., Van Driessen, K., 1997. A fast algorithm for the minimum covariance determinant estimator, submitted for publication.

Ruspini, E.H., 1970. Numerical methods for fuzzy clustering. Inform. Sci. 2, 319–350.

Struyf, A., Hubert, M., Rousseeuw, P.J., 1997. Integrating robust clustering techniques in S-plus. Comput. Statist. Data Anal. 26, 17–37.

Titterington, D.N., 1976. Algorithms for computing D-optimal design on finite design spaces. Proc. 1976 Conf. on Information Science and Systems. Johns Hopkins University, Baltimore, MD, pp. 213–216.