An efficient new way to represent multi-dimensional data

PHILIPPE KENT
Department of Mathematics, Ecole Polytechnique Fédérale, Lausanne

A new method for graphically representing mult-dimensional points in two dimensions is presented. The representation is moderately robust in tolerating a certain amount of noise before being obscured. The method is particularly suitable for presenting the results from cluster analysis based on a minimum spanning tree. Information not readily perceived in conventional arrangements is made apparent. The method is non-iterative.

1. INTRODUCTION

In an experimental situation one is often confronted with multi-dimensional data, measurements of a system taken at various times for example. The coordinates of a point represent the different measurements taken at one time in this case. Certain patterns present in the measurements would then show up in the form of clusters of points in p-space (where p is the dimensionality of the data).

Innumerable calculations can be performed with multi-dimensional data. SPSS,[1] BMDP,[2] and CLUSTAN[3] are a few of the better-known statistical program packages which include many such routines. Hartigan[4] and Gnanadesikan[5] provide texts of suitable methods. Refs 6–8 provide examples of lesser-known (and lesser-used) methods.

Much less numerous are methods which display such patterns. These include simple projections from multi-dimensional space to a two-dimensional space formed by pairs of existing coordinates, discriminant coordinates (canonical variates), or, most commonly, the largest eigenvectors. Pairs of other combinations of coordinates could also be used. One must bear in mind, however, that increasing the complexity of the coordinates chosen, while improving the image obtained, renders that image's interpretation in terms of the original coordinates more difficult. Non-linear mapping[9] is a good example. This iterative process involves attempting to plot the multi-dimensional points in two dimensions while preserving as much as possible the distance relationships between points. The final result will often show what structure is present in the data, but not how to quantify it. Andrews presents an interesting method based on representing a multi-dimensional point by a two-dimensional curve.[10] A collection of points in p-space ($p > 2$) is plotted as a collection of curves in 2-space. A useful projection vector can often be graphically determined from the results. The method is useful for relatively small collections of points of up to 4–5 dimensions, the graph usually becoming too cluttered when higher dimensions are present.

If cluster analysis has been performed, two further methods are commonly used to output information from the analysis. (Everitt[11] provides a good summary of the techniques of cluster analysis.) The first consists in furnishing numbers characterising each cluster's mean, dispersion, eigenvectors, distance from other clusters, etc. The other widely used method of representing the result of a cluster analysis is the dendrogram. This is a diagram showing the relationship (usually a distance) between the different points as these points are accumulated into a cluster, or discarded from a cluster, depending on whether the clustering was agglomerative or divisive. The dendrogram allows the observer to visualise the hierarchy of formation of clusters.

The method presented in section 2 does not replace existing methods of displaying data, but adds to them by providing a method which allows the user to obtain some further information not apparent in the conventional representations. To demonstrate the pitfalls of depending only on information from eigenvectors, for example, consider Fig. 1 as an analogy of p-space in two

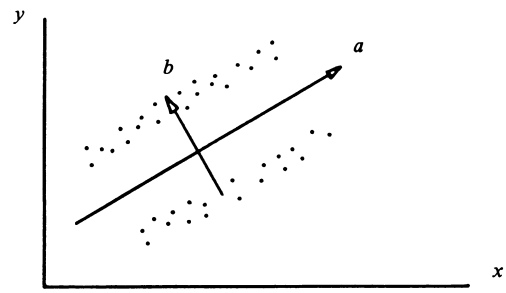

Figure 1.

dimensions. Projection on an original coordinate (x or y) (whether standardised or not), or on the larger eigenvector (a) is of no help in showing the real situation. In this case, projection on to the smaller eigenvector (b) is useful. But a useful axis for projection is not always so easily found; it need not necessarily be an eigenvector, nor indeed exist at all. A dendrogram would indicate the presence of two distinct clusters in the case shown in Fig. 1, but the dendrogram will not provide any information as to the arrangement in space of the clusters.

2. THE METHOD

2.1 Generalities

All the points present in p-space can be connected by a 'minimum spanning tree'. Indeed, this can be used as the basis for a computationally efficient cluster analysis, the single linkage method. The efficiency derives from the fact that each of the possible distances between the points is required only once. (The single-linkage method of cluster analysis is also the only one to satisfy all the conditions set by Jardine & Sibson.)[12] The minimum spanning tree allows various characteristic parameters such as 'linearity' or 'inconsistency' to be obtained as well as the usual projections and dendrogram (see Ref. 13 for example). The minimum spanning tree has been used to construct

non-linear projections.[14] Drawing the minimum spanning tree on conventional projections provides the observer with an indication of how much overlap is present in the projection. The indication is of most value when it is seen that no overlap is present.

(A tree is a connected graph with no closed paths. A spanning tree is a tree containing every node present. A minimum spanning tree (MST) is a spanning tree for which the sum of the weights of the edges (= distance between the two nodes concerned in our case) is minimum. The minimum spanning tree is unique if no edge of equal weight is present. It will usually still be unique even if equal edges are present as long as these edges do not have a common node.)

Fig. 2*a* shows a minimum spanning tree over the points indicated, where the dashed line represents a link which would be cut by a clustering algorithm.

The method presented here uses the 'trunk' of the minimum spanning tree as one of the 'coordinates' of a projection of p-space to 2-space, and one of the original coordinates as the other coordinate, producing a plot relating an original coordinate to a position 'along the tree'. The inverted commas are used because the (minimum spanning) tree does not usually consist of only one branch, namely the 'trunk'. The problem is then to cut the branches and present them in an order which will be of visual use to the observer. It is believed that by finding the most populous branch, using it as the 'trunk', and inserting the remaining branches at their branch points with the 'trunk', a reasonable 'tree coordinate' is produced. One can then proceed similarly with each of the original coordinates in turn, using each time the same 'tree coordinate'. If these two-dimensional representations are now placed side by side (with the 'tree' coordinates parallel to each other), an overall view of the p-space is obtained. Fig. 2*b* shows the result of this process when applied to the points represented in Fig. 2*a*.

Considering Fig. 2*b*, one sees that the cluster represented by the black points is of a 'chain' type, covering all values of X and high values of Y, whereas the cluster of white points can be seen to be at middle values of X and low values of Y and compact in both directions X and Y.

2.2 Details

The distance measure used in the examples presented here is the euclidean metric. Another distance measure could be chosen.

The minimum spanning tree is computed according to Prim's algorithm.[15]

(*a*) Begin with an arbitrary point.

(*b*) Connect the point from (*a*) to its nearest neighbour. The tree now contains two points.

(*c*) Connect to the tree the point which is the nearest of the nearest neighbours of the points of the tree but which is not yet in the tree.

(*d*) Repeat step (*c*) until all points have been connected to the tree.

The advantage of this algorithm is that it requires each of the $n(n-1)/2$ distances only once (where n is the number of points). As there will more often than not be many more points than dimensions, it will be advantageous to keep the n by p matrix of points rather than half of the n by n matrix of distances (where p is the number of dimensions).

Clusters are formed, essentially as a by-product, by severing edges which are larger than the average edge, starting from the largest edge. The lower limit (the average edge length) is arbitrary, based on the assumption that edges smaller than the average should not be cut. The dashed lines in Fig. 2 show such an edge. Again, any other clustering method which provides each point with a cluster number could be chosen.

The 'trunk' of the tree is found by searching for the most populous path (i.e. the longest in number of points, not in distance) through the tree. Printing is in the order defined by the above path under the constraint that points of a same cluster be printed contiguously. Branches containing other points are inserted just before the point

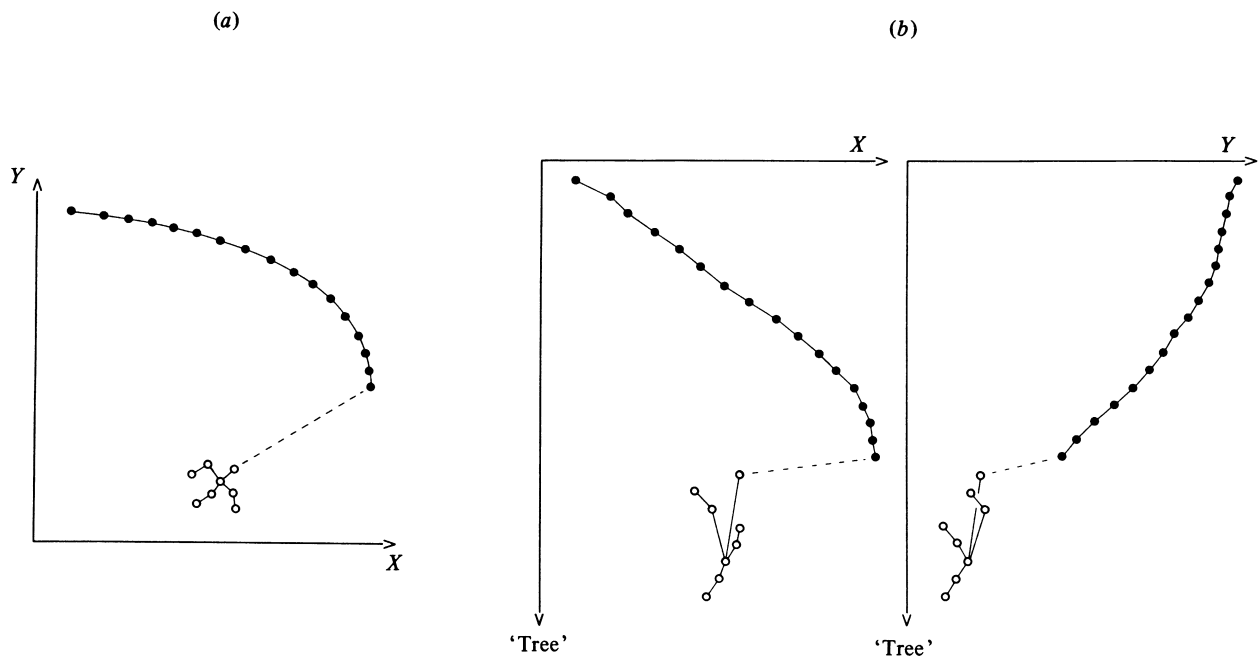

Figure 2.

of branching with the trunk and in an order similar to the one defined for the trunk.

On the computer output, each of the original points is represented by one output line. An output line will represent only one point. Duplicate points will produce duplicate output lines. Output lines are printed on one or more pages of a computer line-printer output which can then be placed side by side for viewing; this disposition yields one continuous line for each point. Blank output lines are inserted between points belonging to different clusters to facilitate interpretation.

3. SAMPLE RESULTS

Fig. 3*a* shows the conventional projections of a set of random points generated along a 3-dimensional spiral. Fig. 3*b* shows the same situation when a certain amount of noise has been added. Note that in 3*b* all useful information has apparently been lost through the projections. Figs. 4*a* and 4*b* show the graphical output of the method described in section 2, for the cases shown in 3*a* and 3*b* respectively. The remarkable feature here is that 4*b* is easily recognised as representing the same phenomenom as 4*a*, less information having been lost in passing from three dimensions to the representation 4*b* than in the transition to the conventional projections, 3*b*.

The conventional projections of an example of 'sausage' clusters are shown in Fig. 5. Fig. 6 shows the output using the method developed here. The two main clusters are clearly visible on the strips corresponding to the variables. The common element between the two clusters is seen to be the increasing (or decreasing) value of the first variable. The strips also show the difference between the two clusters to be the shift to lower values of the second and third variables in the second cluster as compared to the first one. A projection using eigenvectors would show both of the above conclusions only if one vector is such that it explains the largest intra-cluster variance, and the other such that it explains the largest inter-cluster variance.

As a final example, the data from Fisher were chosen.[16] This set of measurements of four variables on each of 150 flowers has become something of a *de facto* standard benchmark in the multidimensional literature. It is considered that *Iris setosa* is easily distinguished from the other two species, *Iris versicolor* and *Iris virginica*; but that these last two are not easily shown to be distinguishable. Fig. 7 shows the output concerning this data set. One will notice on Fig. 7 the presence of three, perhaps four main clusters (clustering via a more subtle procedure depending on edge inconsistencies was used in this case).[13] Petal lengths and widths can easily be seen to

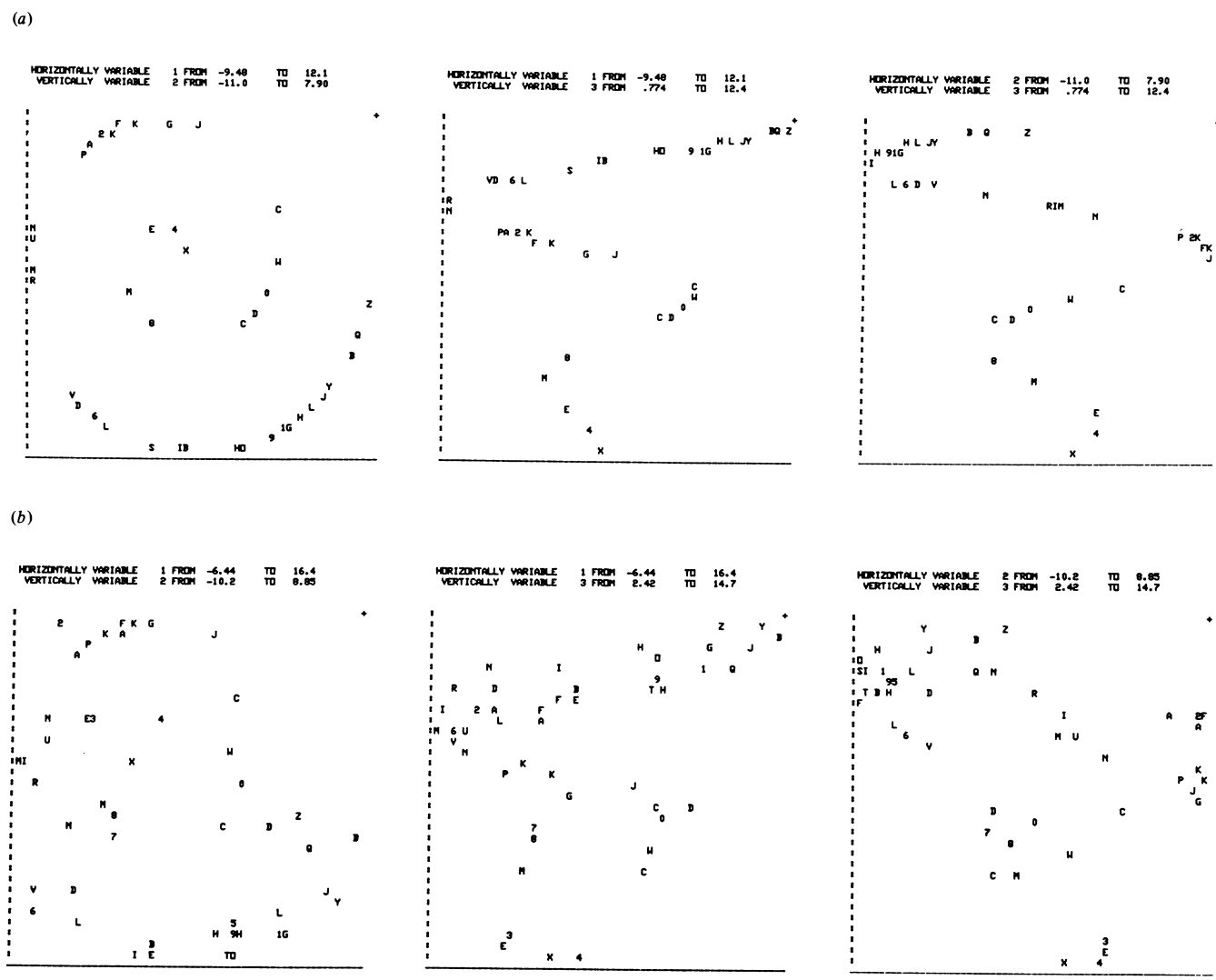

Figure 3.

(a)

```
                GRAPHS FOR VARIABLES
                  MIN      1      MAX      MIN      2      MAX      MIN      3      MAX
POINT NO         -9.48           12.1      -11.0           7.90      .774           12.4
```

(b)

```
                GRAPHS FOR VARIABLES
                  MIN      1      MAX      MIN      2      MAX      MIN      3      MAX
POINT NO         -6.44           16.4      -10.2           8.85      2.42           14.7
```

Figure 4.

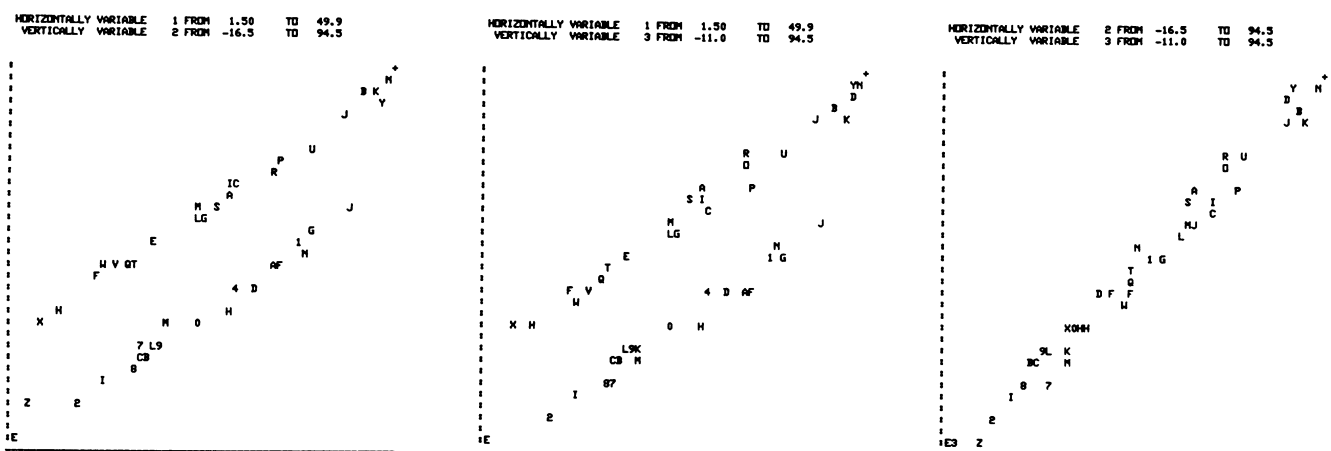

Figure 5.

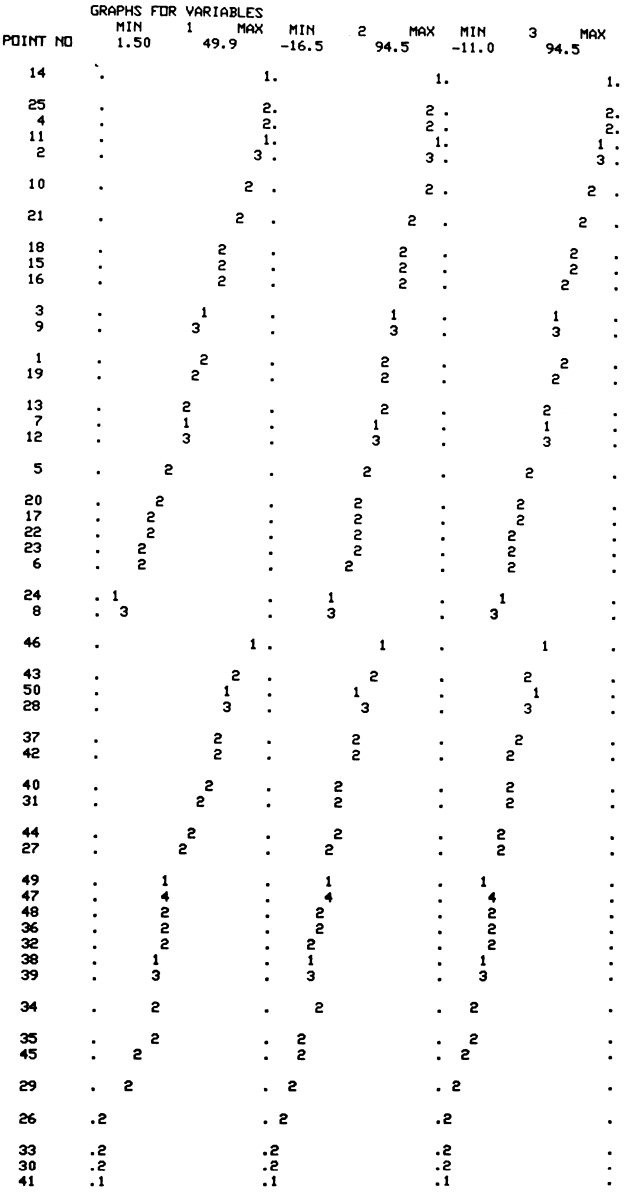

Figure 6.

characterise three clusters. (It may be useful to view the graph at a low angle in this respect.) One can further see that the clusters are relatively compact in the direction of petal length, slightly less so in the directions of petal width and sepal length, and definitely more diffuse in the direction of sepal width. *Iris setosa* is seen to be the most easily distinguishable species.

The points numbered 1–50 correspond to *Iris setosa*. All have been printed contiguously. Points 51–100 correspond to *Iris versicolor*, and points 101–150 to *Iris virginica*. Five cases of *Iris versicolor* are found in the virginica cluster.

Figs. 8 and 9 show what some other representations produce with the iris data. Fig. 8 is the familiar principal-components picture. The *Iris setosa* are indeed seen quite separate from the rest, but without some study of the eigenvector composition the reason for this separation is not apparent. The two other species do not appear to be separable.

Fig. 9 was produced through a non-linear mapping technique.[9] Here one clearly separable and two touching clusters are visible. (Remember though that one does not necessarily have the information shown by representing various species with different symbols.) The visual impression corresponds to that obtained with the method described in section 2, but, the projection being non-linear, interpretation in terms of sepal or petal length or width is hindered.

4. TECHNICAL DETAILS

The computer program used to produce the examples shown is available from the author. The program consists of approximately 260 lines of ANSI-conforming FORTRAN 77 written with an emphasis on efficient memory utilisation as well as portability, and 150 comment lines. A version in BASIC for a Hewlett-Packard 9831A also exists.

Execution time is mainly constrained by the computation of the minimum spanning tree, in particular the calculation of the distance between two points. The execution time will depend approximately on the square

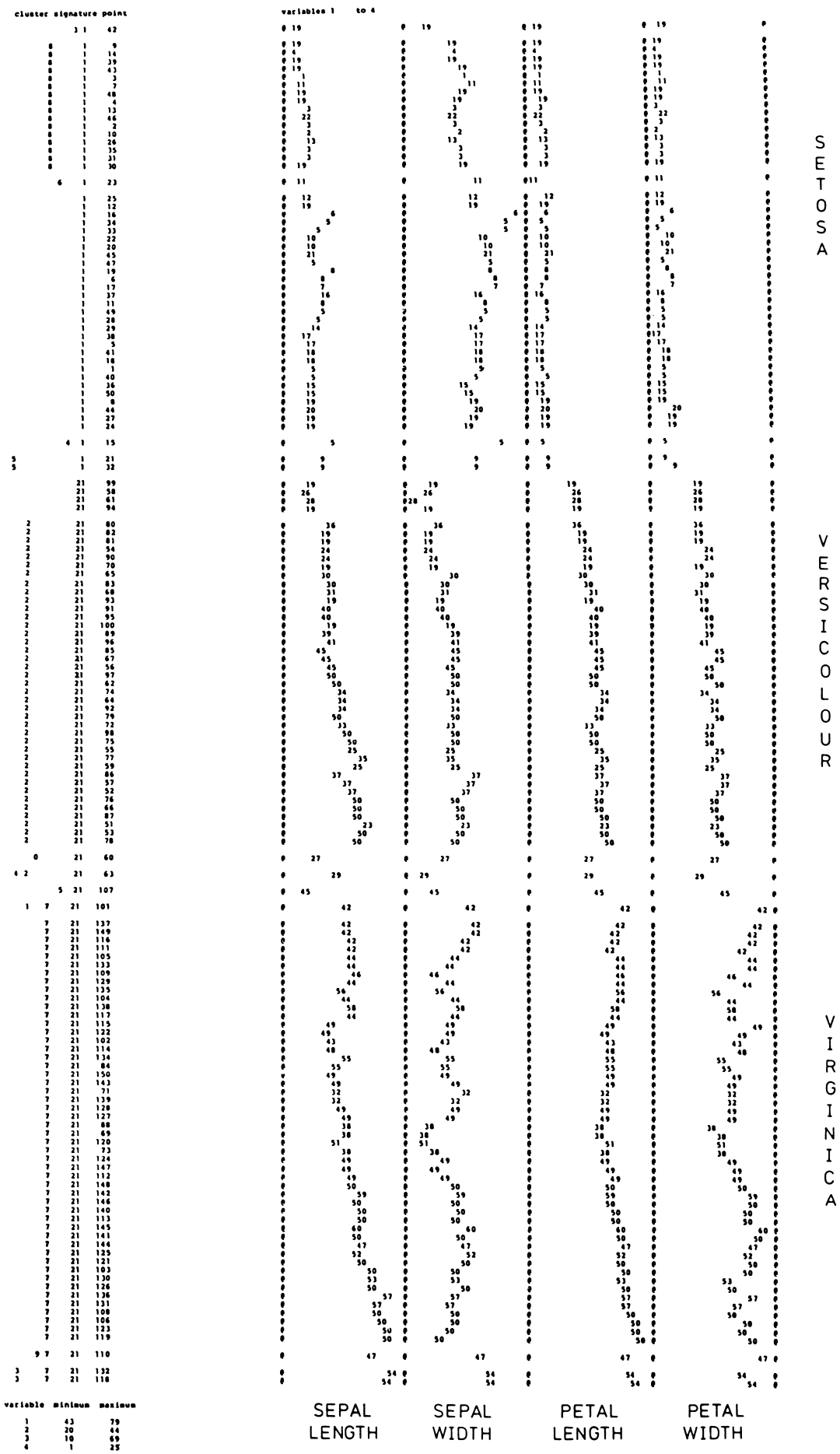

Figure 7.

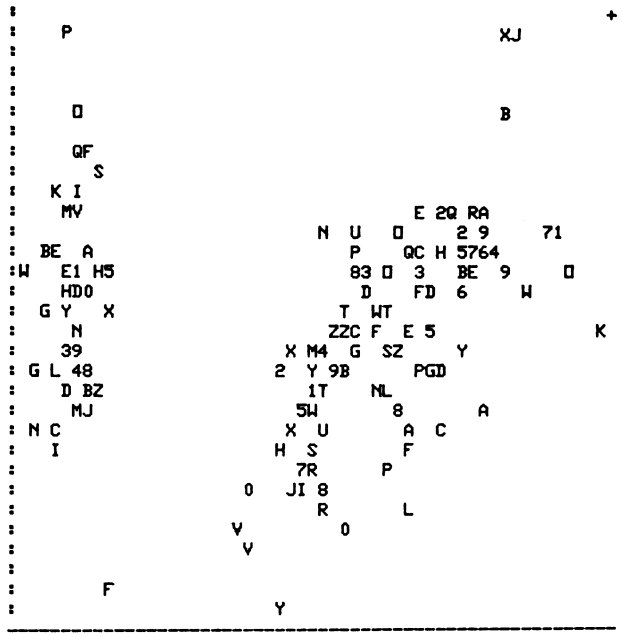

Figure 8.

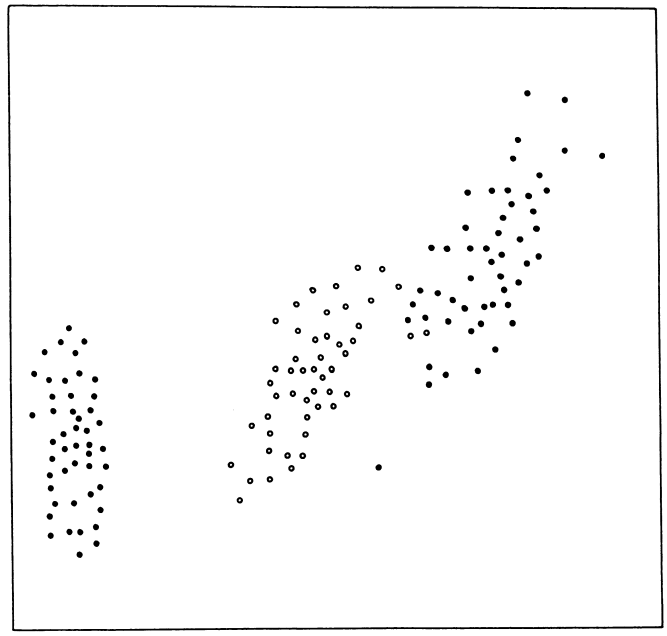

Figure 9.

of the number of points times the number of dimensions. The timings for Fisher's iris data (150 points, 4 dimensions) using the FORTRAN 77 program running on a CDC Cyber 73 (elementary multiplication time = 5.7 μs) are: 608 ms of CPU time for data input, 1680 ms for minimum spanning tree construction, 447 ms for clustering, 135 ms for determination of printing order, 803 ms for printing out.

The requirements of data storage within the program (in a blank COMMON block) are proportional to n (points) times $p+2$ (p dimensions).

5. CONCLUSIONS

A method for representing multi-dimensional data has been described. The results are easily interpreted in terms of the original coordinates (see Fig. 7). The method will tolerate a certain amount of noise in the data before the representation is obscured (see Figs. 3 and 4). The method is particularly suitable for presenting the results of a cluster analysis; indeed, a cluster analysis is provided almost 'free'. Ribbon- or sheet-like clusters are acceptable (see Figs. 3 and 4 again). The emphasis here has been on the graphical output.

Input to a program is simple: the number of points, the number of dimensions, the data format and the data proper. Clusters derived from an external clustering procedure can easily be provided for. The 'tree' order used is useful even if the strips for each coordinate are not output, provided that an informative label for each point takes their place.

The method is computationally efficient: several hundred points of 10–20 dimensions can be accommodated in a small mini-computer. A line printer is all that is needed for output.

Remarks of a referee were helpful in clarifying the presentation of this work.

REFERENCES

1. Nie, Hull, Jenkins, Steinbrenner and Bent, *SSPS*, McGraw-Hill, Maidenhead (1975).
2. W. J. Dixon and M. B. Brown (eds), *BMDP-79*, University of California Press, Berkeley (1979).
3. D. Wishart, *CLUSTAN user manual*, Program Library Unit, Edinburgh University (1978).
4. J. A. Hartigan, *Clustering Algorithms*, Wiley, Chichester (1975).
5. R. Gnanadesikan, *Methods for Statistical Data Analysis of Multivariate Observations*, Wiley, Chichester (1977).
6. W. J. Borucki, D. H. Card and G. C. Lyle, A method of using cluster analysis to study statistical dependence in multivariate data, *IEEE Transactions on Computers* **24**, 1183–1191 (1975).
7. D. H. Schartzmann and J. J. Vidal, An algorithm for determining the topological dimensionality of point clusters, *IEEE Transactions on Computers* **24**, 1175–1182 (1975).
8. G. V. Trunk, Statistical estimation of the intrinsic dimensionality of a noisy signal collection, *IEEE Transactions on Computers* **25**, 165–171 (1976).
9. J. W. Sammon Jr, A nonlinear mapping for data structure analysis, *IEEE Transactions on Computers* **18**, 401–409 (1969).
10. D. F. Andrews, Plots of high-dimensional data, *Biometrics* **28**, 125–136 (1972).
11. Brian Everitt, *Cluster Analysis*, Heinemann Educational Books, London (1974).
12. N. Jardine and R. Sibson, The construction of hierarchic and non-hierarchic classifications, *The Computer Journal* **11**, 177–184 (1968).
13. C. T. Zahn, Graph-theoretical methods for detecting and describing Gestalt clusters, *IEEE Transactions on Computers* **20**, 68–86 (1971).
14. R. C. T. Lee, J. R. Slagle and H. Blum, Triangulation method for the sequential mapping of points from N-space to two-space, *IEEE Transactions on Computers* **26**, 288–292 (1977).
15. R. C. Prim, Shortest connection networks and some generalizations, *Bell System Technical Journal* **36**, 1389–1401 (1957).
16. R. A. Fisher, The use of multiple measurements in taxonomic problems, *Annals of Eugenics* **7**, 179–188 (1936).