Journal of Neuroscience Methods 144 (2005) 19–24

JOURNAL OF NEUROSCIENCE METHODS

www.elsevier.com/locate/jneumeth

Unsupervised clustering algorithm for *N*-dimensional data

Erwin B. Montgomery Jr.[a,b,*], He Huang[a], Amir Assadi[c,d]

[a] *Department of Neurology, National Primate Research Center, University of Wisconsin-Madison, H6/538 CSC, 600 Highland Ave., Madison, WI 53792, USA*

[b] *Department of Neurology, National Primate Research Center, University of Wisconsin-Madison, Building 1 Room 104, 1233 Capital Court, Madison, WI 53715-1299, USA*

[c] *Department of Mathematics, University of Wisconsin-Madison, Madison, WI 53715-1299, USA*

[d] *Wisconsin Genome Center, University of Wisconsin-Madison, Madison, WI 53792, USA*

Received 1 April 2004; received in revised form 2 August 2004; accepted 11 October 2004

Abstract

Cluster analysis is an important tool for classifying data. Established techniques include *k*-means and *k*-median cluster analysis. However, these methods require the user to provide a priori estimations of the number of clusters and their approximate location in the parameter space. Often these estimations can be made based on some prior understanding about the nature of the data. Alternatively, the user makes these estimations based on visualization of the data. However, the latter is problematic in data sets with large numbers of dimensions. Presented here is an algorithm that can automatically provide these estimates without human intervention based on the inherent structure of the data set. The number of dimensions does not limit it.

Keywords: Cluster analysis; Multi-dimensional data analysis; Classification algorithms

1. Introduction

Statistical clustering algorithms are an important technique for analyzing complex multi-dimensional data, particularly for determining whether data are organized in classes by their distribution in some parameter space (Theodorakis and Koutroumbas, 2003). For example, clustering algorithms are used in neuronal extracellular action potential discrimination from microelectrode recordings (Offline Sorter, Plexon Inc. Dallas, TX). The waveforms of extracellular action potentials can be characterized as a set of amplitude at specific time points in the waveform. Different waveforms associated with extracellular action potentials originating from different neurons will have different sets of amplitudes. Cluster analyses are then applied to isolate different groups of waveforms with similar amplitude profiles into different clusters. There are innumerable examples of cluster algorithms ranging from analysis of gene expression (Datta and Datta, 2003) to dietary eating habits (Newby and Tucker, 2004).

Currently available clustering algorithms, such as *k*-mean or *k*-median, require a priori estimates as to the number of clusters contained in the data and locations of these clusters in the parameter space (Stata Reference Manual, 1985). For many current techniques, the initial estimates of the number of clusters are important. These algorithms utilize a portioning procedure in which the parameter space is divided depending on the number of clusters anticipated. Estimates with too few clusters lead to solutions that may not converge and estimating too many leads to arbitrary splitting of the data that reduces the generalizability of the results (Lange et al., 2004). Initial estimates of the number of clusters are affected by scale (Guedalia et al., 1999). Various attempts to estimate the number of clusters include various methods of thresholding the data (Guedalia et al., 1999). However, these algorithms require certain a priori assumptions about the data and require parameters external to the data and thus, risk

[*] Corresponding author. Tel.: +1 608 263 5430; fax: +1 608 263 1420. *E-mail address:* montgomery@neurology.wisc.edu (E.B. Montgomery Jr.).

0165-0270/$ – see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.jneumeth.2004.10.015

arbitrariness. Considerable efforts have been made to assess the validity of initial estimates of the number of clusters (Lange et al., 2004).

Often a priori estimates of the number and locations of clusters are based on some a priori intuitions or human inspection of the data. This limits the number of dimensions in the data set and in the case of visualization, typically to three dimensions that allow for visualization in a Cartesian coordinate space. A new method has been developed for cluster analysis that does not require a priori estimates and therefore, can run without supervision. In addition, this new method is not limited in the number of dimensions used to characterize the data.

2. Materials and methods

The clustering method automatically estimates the number and approximates the locations of the centroids of clusters iteratively, with minimum supervision and without a priori estimations of the number of locations of putative clusters. These estimates are then relayed to a variation of standard *k*-means clustering algorithms. There are three general phases. The first is an estimation of the neighborhoods among the data points. These neighbors provide an initial sample of data points within a neighborhood to determine the statistical characteristics of the neighborhood. Data points are related to neighborhoods in distance measured as *z* scores (statistical distance) based on the statistical characteristics of the neighborhood. The second phase statistically validates inclusion or exclusion of additional data points into a specific neighborhood, thus expanding the neighborhood to the entire cluster. The third phase analyzes the clusters and their centroids to determine, which are redundant and then combines those that are. The third phase then reassigns data points to the final set of clusters based on which centroid is closest in statistical distance to the data point. This phase is a reiterative process that continues until the centroids are no longer modified either in number or location.

The first phase establishes neighborhoods among the data in a hierarchical manner starting with the data point that has the most number of neighbors within a defined multidimensional radius. The assumption is that the data point closest to the centroid of a cluster will have the most neighbors. The Euclidean distances between all possible pairs of data points in the multi-dimensional space are determined as:

$$\text{Distance}_{N_1-N_2} = ((D_{1,N_1} - D_{1,N_2})^2 + (D_{2,N_1} - D_{2,N_2})^2 + (D_{3,N_1} - D_{3,N_2})^2 + \ldots)^{1/2}$$

where $D_1, D_2, D_3, \ldots$, are the dimensions or number of variables. Note that distances merely represent an interval scale related to some parameter by which the observations are measured.

The purpose of the determining neighborhoods is to allow sampling enough data points within a cluster to determine the statistical characteristics of the cluster. Thus, it is sufficient to sample only a fraction of the entire cluster. This allows the use of small radii that reduces the probability of including multiple clusters into a neighborhood. The following process is applied to all data points. Each data point serves as an index to its own neighborhood. Thus, each data point has a neighborhood of surrounding data points. The initial radius around the index data point is set to zero and then increased incrementally. This process is repeated for each data point until one of the data points has a neighborhood with 10% of the other data points. The remainder of the other data points will have fewer neighbors. The data points are then sorted based on the number of neighbors.

The second phase is a reiterative process applied to each data point acting as an index beginning with the data point with the most number of neighbors and processing through to the data point with the least (Fig. 1A). The process uses the distribution of neighbor data points surrounding the index data point to determine the statistical characteristics of the neighborhood. Then, the closest data point outside the neighborhood is analyzed to see, if it falls within the statistical distribution of the neighborhood (Fig. 1B). If so, the data point is added to the neighborhood, and the centroid of the new cluster is calculated. The next closest data point outside the new cluster is analyzed to determine whether it is within the statistical distribution of the new cluster. If the next data point is within the statistical distribution of the new cluster, that data point is incorporated into the new cluster, a new centroid is determined, and the data point is unavailable for subsequent clusters. This process is repeated until the analysis demonstrates that the next data point is not within the statistical distribution of the cluster. Then, the process begins with the next index data point based on the number of neighbors.

The method to determine whether the next point is a member of the cluster uses the distribution of the data points already within the cluster. A straight line is constructed connecting the centroid of the initial cluster to the next data point as shown in Fig. 1B to the next data point. The spatial distribution of the data points within the cluster are collapsed or projected onto the line connecting the next data point, and the centroid of the cluster as shown in Fig. 1C. The projections onto the connecting line form a distribution of locations on the connecting line as shown in Fig. 1D. The spatial location on the connecting line for the next data point can be converted to a *z* score based on the distribution from the projections of the data points in the initial cluster onto the connecting line. Thus, the Euclidean distance is converted to a statistical distance. If the statistical distance of next data point on the connecting line has a *z* score less than 1.96, then the next data point is considered to be within the initial cluster, and it is added as a member of the initial cluster, and the centroid is recalculated, and the next nearest data point analyzed (Fig. 1E).

A line connecting the centroid and the next closest data point is used to project the spatial distribution of the data

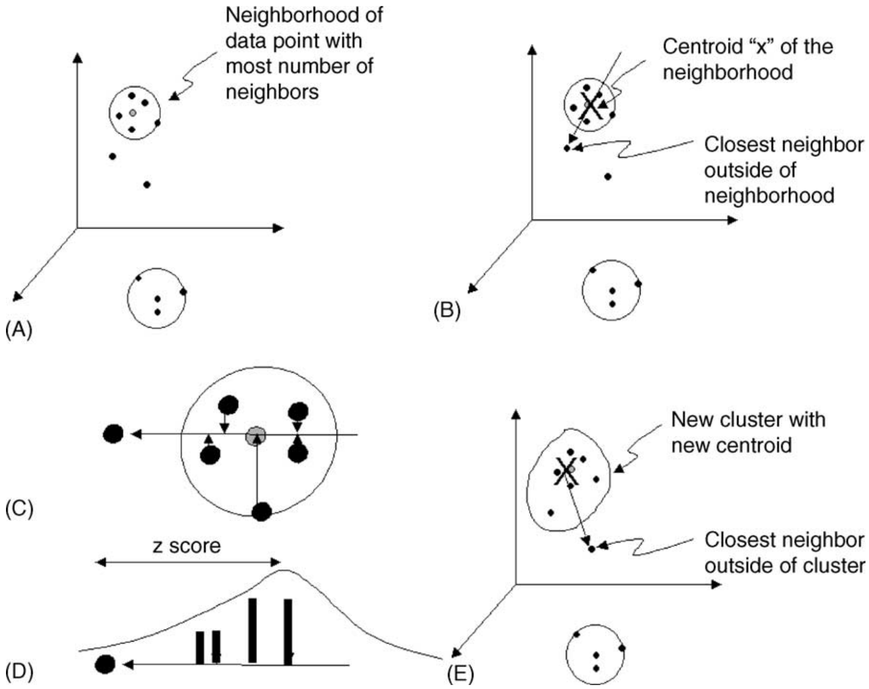
Neighborhood of data point with most number of neighbors
(A)
Centroid "x" of the neighborhood
Closest neighbor outside of neighborhood
(B)
(C)
z score
(D)
New cluster with new centroid
Closest neighbor outside of cluster
(E)

Fig. 1. (A) Distribution of hypothetical data points. An initial cluster is created by creating a radius around a point and counting the number of other data points within the radius. See text for a description of how the radius is determined. The centroid of the neighborhood with the greatest number of neighbors is identified and its centroid "X" determined. The next nearest data point outside the initial neighborhood based on Euclidean distance is identified. A straight line is constructed connecting the centroid of the initial neighborhood to the next data point as shown in B. The next step is to determine whether the next data point lies within the statistical distribution of data points in the initial neighborhood. The statistical distribution is determined by projecting the data points within the initial neighborhood onto the line connecting the centroid to the next data point (C). The projections onto the connecting line form a distribution of locations on the connecting line as shown in C and D. The spatial location on the connecting line for the next data point can be converted to a *z* score based on the distribution from the projections of the data points in the initial neighborhood onto the connecting line. This converts Euclidean distance to a statistical distance. If the next data point has a *z* score less than 1.96, then the next data point is considered to be within the initial neighborhood, and it is added as a member of the cluster. A new centroid is recalculated, and the process is repeated for the next nearest data point.

points within the cluster because it cannot be assumed that the spatial distribution of the cluster will be symmetric. Some models implicitly assume a spherical or symmetric distribution of data (Lange et al., 2004)). This implementation is different from most others using *k*-mean or *k*-median algorithms, where the mean and standard deviations (S.D.) of the distances are used irrespective of the direction in the multi-dimensional space.

The method requires aligning one of the coordinate axes to the line connecting the centroid to the next data point. The origins of the Cartesian coordinates are translated such that the origin of the coordinate system is moved to the centroid of the cluster under analysis. Axis rotations using the Graham–Schmidt algorithm align the centroid and the next data point line on the same single axis (Fig. 2), and the other axes are rotated orthogonally to the axis that now is the line between the centroid and the next data point.

The second phase creates a set of centroids base on the first phase of the algorithm that defines neighborhoods around each data point. It is possible that the data point with the greatest number of neighbors may be in the same cluster as the data point with the next greatest number of neighbors and therefore, the clusters are redundant. The third phase then analyzes the spatial distribution of the centroids using the same algorithm as described in the second phase applied to data points to eliminate redundant clusters.

Again, it is necessary to establish a neighborhood around each centroid in order to determine the statistical characteristics of the centroid's cluster. Initially, the three nearest data points to each centroid based on Euclidean distance are selected as the centroid's neighborhood and used to determine the statistical characteristics. The statistical distance of all the remaining data points, relative to the statistical characteristics of the cluster, is determined as described above and illustrated in Figs. 1 and 2. Each data point is then assigned to the cluster, whose centroid has the shortest statistical distance. The data points were assigned to the cluster and a new centroid is determined. This process is repeated until there is no change in the number and locations of the centroids.

3. Results

We tested the algorithm using a simulated set of five clusters in a five-dimensional space. Five centroids were created

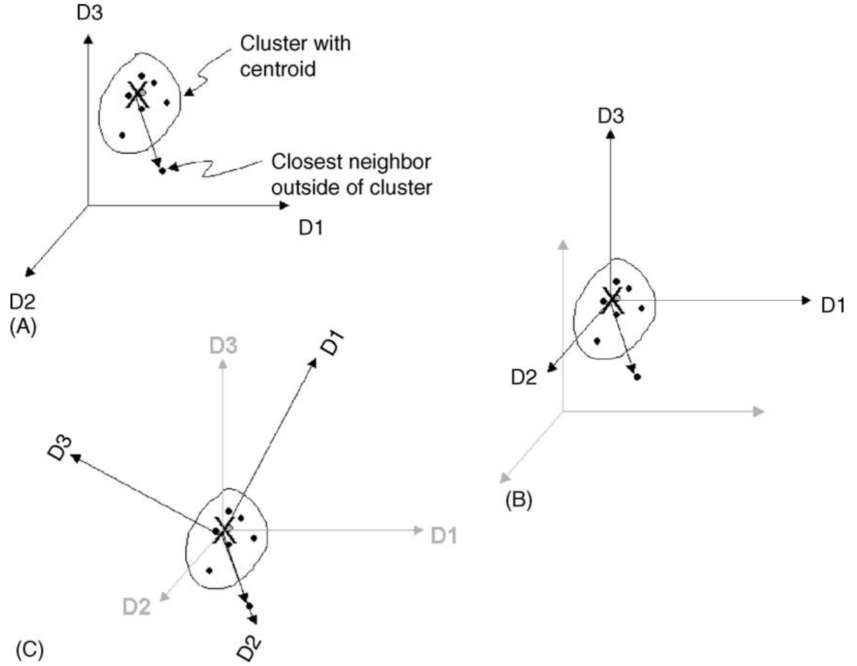

Fig. 2. Schematic representation of the methods used to determine the distance of the next data point from the clusters. The centroid of the cluster is determined (A). The coordinate system is then translated so that the origin of the translated coordinate system is now located at the centroid of the cluster (B). The translation is accomplished by subtracting the original coordinates of the centroid from each point by the corresponding coordinate in the original axes. The new values of the coordinates represent their position in the translated coordinate system. The translated coordinate system is then rotated such that the next data point lies on one of the axes (C). This is done such that one of the axes in the translated coordinate system falls on the connecting line between the centroid of the cluster and the next data point. The axis containing the largest value coordinate is the axis that will be aligned with the connecting line, for example, D_2. The rotation is performed using the Grahm–Schmidt algorithm such that the other coordinates on the other axes, for example, D_1 and D_3, become 0. Thus, the coordinates for the centroid become (0, 0, 0, . . .), while the coordinates for the next data point become (x, 0, 0, . . .), where x is the value of the coordinate on the translated and rotated axis and equals the Pythagorean distance of the next data point and the centroid of the cluster. The analysis continues with projecting all the data points of the cluster onto the new axis that lies on the connecting line.

with a distances between them of two units along each dimension. Data points then were created around these centroids. The density of data points reduced in a Gaussian manner as distance from the centroid increased (Fig. 3). The density distribution is determined by the S.D. Thus, the centroids served as the mean for a Gaussian random number generator using a specified S.D. of 0.1–2 units in 0.1 increments. This was repeated for each dimension; thus, each data point was associated with a value in each of the five dimensions. An example of the first two clusters is shown in Fig. 3. This figure shows the distribution of data points for two clusters. The horizontal axis could represent the distance along any dimension. The vertical axis represents the number of data points are interval distances along the dimension. In this case, the centroid distance of the first cluster is one unit and the centroid distance of the second cluster is three or two units from the centroid of the first cluster. The S.D. of the distribution of the data points is 0.8. While there is considerable overlap in the distributions, two distinct peaks can be appreciated.

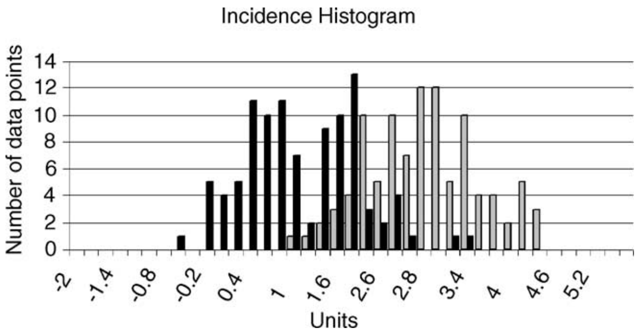

Fig. 3. Demonstration of the distribution of simulated data points along a single dimension. The horizontal axis represents the value of each data point in a parameter corresponding to one dimension. The vertical axis represents the number of data points associated with the interval value of the data point. Two clusters were created. The first had a mean parameter value of 1 and the other had a mean parameter value of 3. The S.D. of the data points in each cluster is 0.8.

Fig. 4 shows the distribution of data points created in five dimensions. The distribution is shown as a series of two-dimensional graphs for all non-repeating pairs of dimensions. Also, the distribution is shown for three of the five dimensions to help demonstrate the multi-dimensional distributions. The algorithm was applied for cluster centers separated by two units for a series of S.D. in the same units.

Success was assessed when the algorithm correctly detected the five simulated clusters. Fig. 5 shows the number of clusters detected for different standard deviations. As can be seen, the unsupervised algorithm detected the five clusters

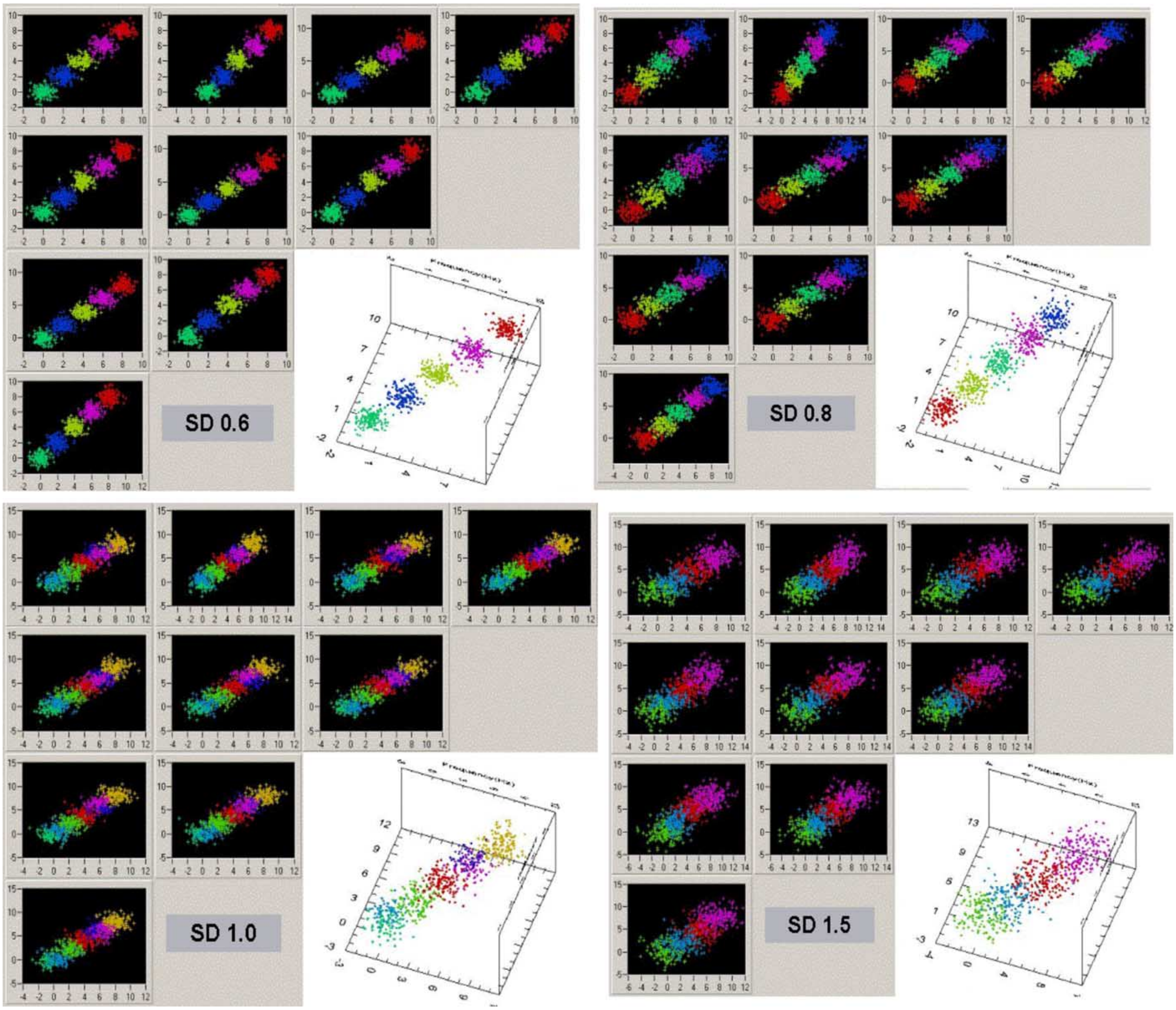

Fig. 4. Graphs showing the distributions of the simulated data points derived from a Gaussian random number generator utilizing a mean and a S.D. as described in the text. Representative graphs are shown for S.D. of 0.6, 0.8, 1.0 and 1.5. The two-dimensional graphs show the distribution of points for all non-repeating pairs of dimensions. For example, the top right figure in each set is the distribution in the first vs. second dimension, the next figure to the right is the distribution in the first and third dimension, etc. Because all five dimensions cannot be visualized simultaneously, a three-dimensional graph is shown in the lower left corner of each panel showing the distributions in three of the five dimensions in this case, first vs. second vs. third dimensions. This was done to provide a more intuitive sense of the higher dimensions. The data points are color coded for the clusters identified by the unsupervised *n*-dimensional algorithm.

until a S.D. of 0.9 was reached. The algorithm then overestimated the number of clusters, when the S.D. was 0.9–1.3. This most likely resulted because the overlap in the distributions between centriods appeared as independent clusters as shown in the multi-modal distribution shown in Figs. 3 and 5. Consider the example of two clusters whose centroids are separated by two units and whose distribution of data points are characterized by a S.D. of 1, the overlap between the two distributions will be 38% of each distribution.

With S.D. from 1.4 to 2, the algorithm underestimated the clusters. This is most likely due to the fact that the distributions of the data points overlapped to such an extent that the distribution had a reduced number of modes as shown in Fig. 6.

4. Discussion

The algorithm described here can determine the number and locations of clusters in complex multi-dimensional data without a priori estimations of the number of clusters or the locations of their centroids. The algorithm was effective up

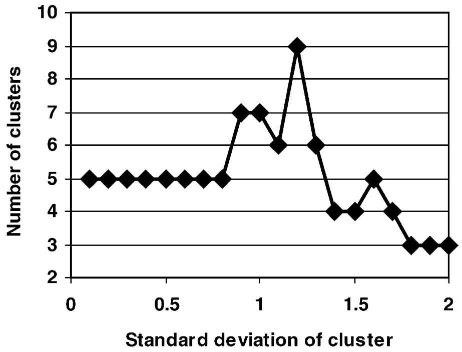

Fig. 5. The number of clusters detected in the simulated data set as a function of the S.D. used by the Gaussian random number generator used to create the five-dimensional data set. The algorithm correctly identified the five clusters up to a S.D. of 0.9. When the S.D. increased to 1.4, the algorithm underestimated the number of clusters. Note that the centers of adjacent simulated clusters were 2 units.

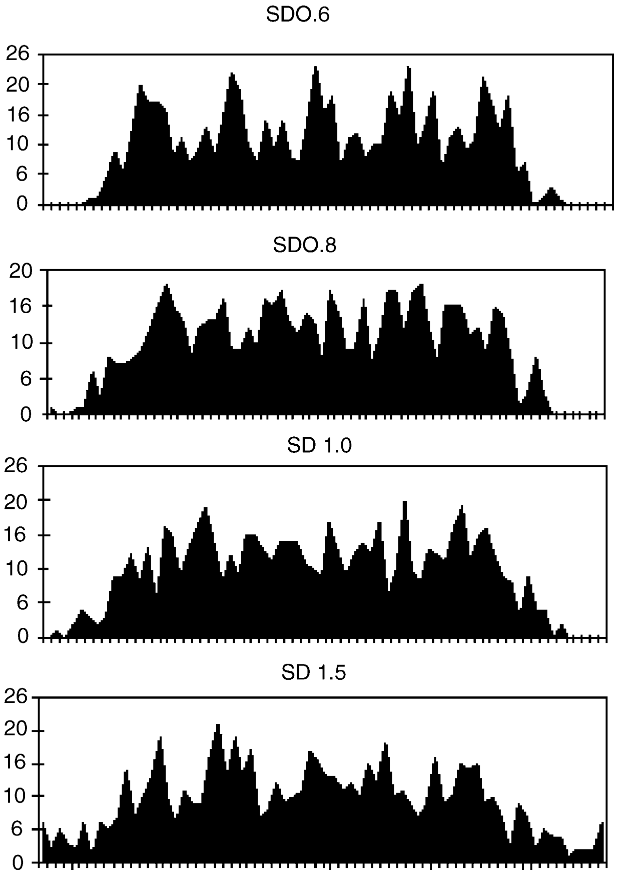

Fig. 6. An interval histogram showing the number of data points distributed along a representative dimension (axis) of the simulated data for different S.D. Note that with a S.D. of 0.6, five clusters are discernable. At a S.D. of 0.8, the clustering algorithm was still able to identify the five clusters (see also Fig. 3), while these would be difficult to visually identify. At a S.D. of 1.0, it is difficult to distinguish five peaks or clusters and the algorithm interpreted the distribution as showing seven clusters. At a S.D. of 1.5, the overlap now combines difficult clusters and four broad peaks are describable and where identified as four clusters.

to a S.D. of 0.8, which means that clusters could be separated even when there was a calculated 21.2% overlap between the clusters as can be seen in Fig. 4. With further overlap, the periphery of the clusters are additive and thus, constitute their own cluster. From a strictly empiric view, without a priori knowledge of how the clusters were constructed, these new clusters created by the overlap are, arguably, legitimate clusters in their own right. When the overlap reaches 47.8% associated with a S.D. of 1.4, the algorithm fails to detect some clusters. Thus, the algorithm lends itself to analyzing data of large dimensions. Further, it can operate unsupervised, which facilitates automated data analysis of very large data sets and very large dimensions that would not be feasible for human operators.

The algorithm differs from previous methods of cluster analysis without the necessity of a priori estimates of the number and locations of clusters. Further, the algorithm differs by not using the assumption that the multidimensional spatial distribution is symmetric. Instead, the distributions of data points within clusters vary by the relationship between the centroid and the data point being analyzed.

Acknowledgements

This work was supported by an American Parkinson Disease Association Advanced Center for Research grant and in part, by NIH grant number 5P51 RR000167 to the National Primate Research Center, University of Wisconsin-Madison and the American Parkinson Disease Association.

References

Theodorakis S, Koutroumbas K. Pattern recognition. 2nd ed. Amsterdam: Academic Press; 2003.

Datta S, Datta S. Comparisons and validation of statistical clustering techniques for microarray gene expression data. Bioinformatics 2003;19:459–66.

Newby PK, Tucker KL. Empirically derived eating patterns using factor or cluster analysis: a review. Nutr Rev 2004;62:177–203.

Stata Reference Manual. Stata Statistical Software Release 7.0. StataCorp: Colleage Station, 1985;1:224–235.

Lange T, Roth V, Braun ML, Buhmann JM. Stability-based validation of clustering solutions. Neural Comput 2004;16:1299–323.

Guedalia ID, London M, Werman M. An on-line agglomerative clustering method for nonstationary data. Neural Comput 1999;11:521–40.