Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Pattern Recognition Letters 24 (2003) 1563–1569

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Visual cluster validity for prototype generator clustering models

Richard J. Hathaway [a,*,1], James C. Bezdek [b]

[a] Department of Mathematics and Computer Science, Georgia Southern University, P.O. Box 8093, Statesboro, GA 30460-8093, USA
[b] Department of Computer Science, University of W. Florida, Pensacola, FL 32514, USA

Received 29 August 2002; received in revised form 9 November 2002

Abstract

Conventional cluster validity techniques usually represent all the validity information available about a particular clustering by a single number. The display method introduced here uses images generated from the results of any prototype generator clustering algorithm to do cluster validation.

Keywords: Cluster validity; Data visualization; Dissimilarity data; Prototype generator clustering; Visual cluster validity

1. Introduction

Clustering attempts to partition a dataset into self-similar groups (clusters). In this study we consider clustering methods based on the hard (Tou and Gonzalez, 1974), fuzzy (Bezdek, 1981) and possibilistic (Krishnapuram and Keller, 1993) c-means algorithms. Those methods and their various generalizations simultaneously attempt to partition the data and describe the geometric structure of the clusters using prototypical cluster shapes such as: volumetric clouds; hyper-spherical shells; hyper-dimensional lines, planes or regression models; etc. We refer to this large class of methodologies as *prototype generator* clustering methods.

An essential fact about prototype generator clustering methods is that they *always* produce clusters (after all, that is their job, and they do it), even if the number of clusters assumed is "incorrect", or the prototypes are inconsistent with the geometry of the clusters, or worst of all, there really are not any clusters in the data, even though every algorithm will find some. This disturbing property justifies the study of *cluster validity* techniques, which attempt to assess the "correctness" of a particular set of clusters in a given dataset.

Cluster validity is a widely studied problem. Nearly 20 years ago, Hubert and Arabie (1985) asserted that a comprehensive review of the literature would require a monograph—now, it would

* Corresponding author. Tel.: +1-912-681-5619; fax: +1-912-681-0654.
E-mail addresses: hathaway@gsaix2.cc.gasou.edu (R.J. Hathaway), jbezdek@uwf.edu (J.C. Bezdek).
[1] Partially supported by a GSU faculty research stipend.

0167-8655/03/$ - see front matter
doi:10.1016/S0167-8655(02)00395-1

require a polygraph! The vast majority of validation methods attempt to assess the degree of validity with a scalar measure of "partition quality", or "natural validity", and so on. One problem inherent with this approach is that representing the correctness of a particular cluster analysis by a single real number invariably loses much information. We take the opposite tack, proposing here a visual display of the fit to the n data by the clustering model via an $n \times n$ intensity image that uses *all* of the information produced by the clustering method. This method essentially follows the SHADE approach introduced in (Ling, 1973), adapting it for use with all prototype generator clustering methods. Our visual cluster validity (VCV) approach retains and organizes the information that is lost through the massive aggregation of information by scalar validity measures.

Section 2 gives a brief description of prototype generator clustering methods. Section 3 describes the VCV approach and describes its connections to its nearest relatives. Three numerical examples using two prototype generator clustering methods are given in Section 4. Section 5 contains a short discussion and conclusions.

2. Prototype clustering methods

Most prototype generator clustering methods partition a dataset $X = \{x_1, \ldots, x_n\} \subset R^s$ by (approximately) minimizing a member of the family of functionals

$$J_m(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} U_{ik}^m d(v_i, x_k)^2 + P_m(U), \qquad (1)$$

where $n = |X|$, $m \in [1, +\infty)$ is a user-defined fuzzification constant, c is the number of clusters assumed, U is a $c \times n$ matrix of memberships (U_{ik} = degree of association of x_k with cluster i), $V = \{v_1, \ldots, v_c\}$ is a set of prototype parameters, $d(v_i, x_k)$ measures the distance between x_k and prototype i, and $P_m(U)$ is a penalty term (possibilistic only).

The hard and fuzzy approaches (where $P_m(U) \equiv 0$) are discussed in (Bezdek, 1981), and the possibilistic approach is introduced in (Krishnapuram and Keller, 1993). A more recent discussion of this general approach and some of its most important instances are given in (Bezdek et al., 1999).

The three most important points about prototype generator clustering are that: (1) prototype generator clustering has established itself as a useful, and therefore widely used, clustering methodology in both pattern recognition and control; (2) no completely satisfactory method exists for determining the validity of clusters produced by *any* clustering algorithm (see, for example, the critique of Bezdek et al. (1997)); they compare 23 scalar indices of cluster validity and conclude that *none* of them are exceptionally reliable across a wide range of datasets); and (3) prototype generator clustering terminates with a set of datum to prototype distances $\{d(v_i, x_k)\}$. The distances $\{d(v_i, x_k)\}$ are an important key to our new VCV method, as they can be used to calculate pairwise dissimilarities between each pair of data points, and these "distances" are the basis for VCV display.

3. Validity displays

The earliest published reference we can find to a visual display technique similar to our VCV method is Ling (1973). Ling's approach, known as SHADE, gives a display of clusters having the structure of volumetric clouds. (An important point is that SHADE seems appropriate *only* for volumetric cloud clusters.) SHADE relies on the output of a clustering technique (such as an hierarchical technique) to produce a "cluster ordering" of the data. Then SHADE approximates an intensity image representation of the clusters using a crude (well, it *seems* crude in 2002; but in 1973, this was the method du jour for making images!) fifteen level halftone scheme created by overstriking standard printed characters. The halftone (intensity) level for each pair of data is based on the Euclidean distance between the points, where dark corresponds to near and light to distant. SHADE displays only the lower triangular part of the complete square display.

Closely related to SHADE, but presented more in the spirit of *finding* clusters (i.e., as a visual

clustering algorithm) rather than *displaying* clusters found with an outsourced algorithm is the "graphical method of *shading*" described on p. 577 of Johnson and Wichern (1992). This method begins with a matrix of inter-datum Euclidean distances $R = [r_{ij}]$:

$$R_{ij} = |x_i - x_j|_E = \sqrt{\sum_{k=1}^{s} (x_{ik} - x_{jk})^2}. \quad (2)$$

Johnson and Wichern give their method as a four step procedure: (i) arrange the inter-data distances from (2) into several classes of 15 or fewer, based on their magnitudes; (ii) replace all distances in each class by a common symbol with a certain shade of gray; (iii) reorganize the distance matrix R so that items with common symbols appear in contiguous locations along the main diagonal (darker symbols correspond to smaller distances); (iv) extract groups of similar items correspond to patches of dark shadings.

Recently, the method of shading was updated by Bezdek and Hathaway (2002) as a procedure called VAT, which stands for *Visual Assessment of* (*Cluster*) *Tendency*. VAT uses a digital intensity image to represent R rather than a halftone scheme with only 15 shades. Additionally, VAT uses an efficient reorganization scheme (corresponding to (iii) above) based on a modification of Prim's algorithm for computing minimal spanning trees. In this case, dark blocks along the diagonal of the digital representation of the reorganized distance matrix can be used to identify volumetric cloud clusters. We illustrate VAT with a simple example. Fig. 1(a) is a scatter plot of a 2-D dataset consisting of two volumetric cloud clusters.

Let R and R^* denote the original and reorganized distance matrices, and $I(R)$ and $I(R^*)$ be their corresponding VAT images. $I(R)$ and $I(R^*)$ derived from the data in Fig. 1(a) are shown in Fig. 1(b) and (c). The reorganized representation $I(R^*)$ clearly indicates two clusters in the data. Of course, the power of this technique lies in its applicability to cases where the data occur in dimensions greater than three, so that a scatterplot revealing the cluster structure is not possible.

How is our new VCV approach related to SHADE and VAT? VCV is similar to VAT in that a digital image (rather than a halftone scheme) is used to display the various inter-datum distances. VCV is similar to SHADE in that it is a tool to display (rather than find) the clusters produced by an outsourced clustering algorithm. In particular, the outsourced algorithm assumed for VCV is any member of the prototype generator clustering family described in Section 2. VCV specification requires details about two issues: "cluster ordering" and inter-datum distances.

Reordering of the data for VCV display is done in two steps: (s1) the clusters themselves are (possibly) reordered; and then (s2) the data in each cluster are reordered. *For VCV display*, *we define the distance between clusters as the Euclidean distance between the parameters defining the cluster prototypes*. Cluster reordering is done by (arbitrarily) keeping the original first cluster as the first reordered cluster, and then (possibly) reordering the remaining clusters so that (new) cluster $i + 1$ is the "nearest" of the remaining clusters to (newly indexed) cluster i. For example, (new) cluster 2 is picked from old clusters $\{2, 3, \ldots, c\}$ by finding the

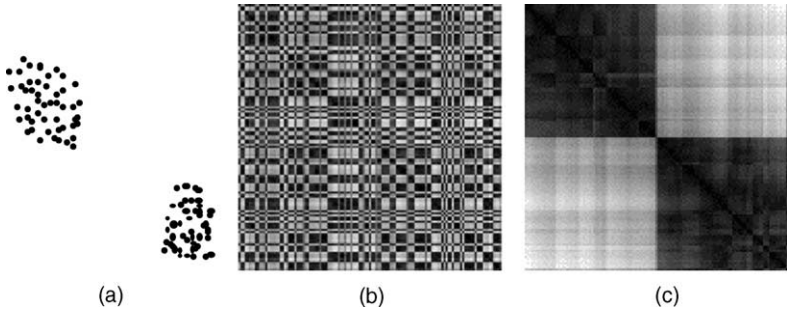

Fig. 1. (a) Data for VAT, (b) $I(R)$ and (c) $I(R^*)$.

old cluster nearest to cluster 1; then (new) cluster 3 is picked from old clusters $\{3,4,\ldots,c\}$ by finding the old cluster nearest to (new) cluster 2; and so on. In our implementation, we calculated inter-cluster distances to be the Euclidean distances between the parameters defining the cluster prototypes. (For example, the parameters for FCM consist of the cluster mean vectors.) This "greedy" reordering prevents a cluster from appearing more than once as a diagonal element in the visual display.

After the clusters are reordered, the ordering of the data in each cluster is considered. Recall that the prototype generator clustering family in Section 2 includes hard, fuzzy and possibilistic methods. If the algorithm used generates a hard partitioning, then no further reordering of the data in each cluster is done. If the clustering is a fuzzy or possibilistic (i.e., soft) partitioning, then each datum is assigned to the (crisp) cluster corresponding to its largest membership value. In other words, we assign datum x_k to cluster i if the terminal $U_{ik} \geqslant \{U_{1k},\ldots,U_{ck}\}$. A tie-breaking strategy can be used when this does not uniquely specify the assignment. (This procedure finds the crisp labeling U_{mm} of X, commonly called the maximum membership *hardening* of the soft partition U). Finally, for the fuzzy and possibilistic schemes, the data in each (crisp) cluster of U_{mm} are reordered in accordance with decreasing membership values in U. In other words, the (reordered) first datum in a cluster has the highest membership value for that cluster; the (reordered) second datum has the second highest membership value for that cluster; etc. This intra-cluster reordering roughly orders "nearby" data points so that they are close to each other. This imparts a smoother appearance to the video displays.

The second important issue in VCV is the measurement of inter-datum distances. The use of pairwise Euclidean distances only makes sense if the clusters are extremely well separated or if they consist of volumetric clouds. We need a measure of dissimilarity R_{jk} between data x_j and x_k that is small if they both fit well into the same cluster and that is large otherwise. We want to generate this dissimilarity measure in a computationally efficient manner, ideally using the $[d_{ik} = d(v_i, x_k)]$ values produced in the process of clustering via a member

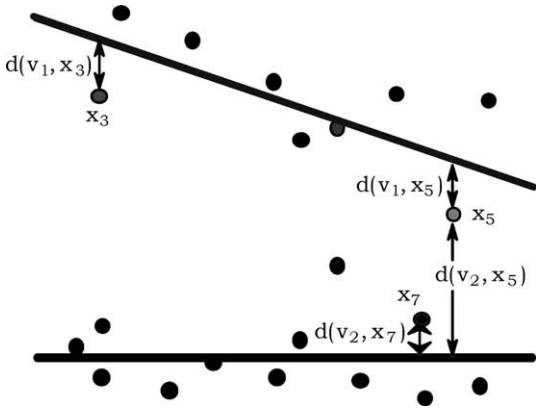

Fig. 2. Dissimilarity calculations via (3).

of (1). We define a pairwise dissimilarity R^*_{ik} for VCV as:

$$R^*_{ik} = \min_{1 \leqslant j \leqslant c} \{d_{ji} + d_{jk}\} \tag{3}$$

This choice gives a measure of dissimilarity that is symmetric and satisfies the triangle inequality, but it is not a metric since generally $R^*_{jj} > 0$, and $R^*_{jk} = 0$ does not, in general, imply that $x_j = x_k$.

Example calculations using (3) are shown in Fig. 2, where the distances $[d(v_i, x_k)]$ are measured vertically from the datum to the line prototypes. Note that this approach (correctly) indicates less dissimilarity between x_3 and x_5 (which are both close to prototype v_1), than between x_5 and x_7, even though the Euclidean distance between x_5 and x_7 is smaller than the distance between x_3 and x_5. Thus $R^*_{35} = d(v_1,x_3) + d(v_1,x_5) < R^*_{57} = d(v_2,x_5) + d(v_2,x_7)$.

After the matrix R^* of reordered pairwise dissimilarity values is found, the information is displayed as an intensity image $I(R^*)$, where small dissimilarities are represented by dark shades and large dissimilarities are represented by light shades. Roughly speaking, (darkly shaded) diagonal blocks in the $I(R^*)$ correspond to clusters in the data.

4. Numerical examples

Fig. 3(a) and (b) are sets of $c = 3$ visually apparent clusters of $n = 150$ points each in $s = 2$ dimensions. We test the VCV approach to dis-

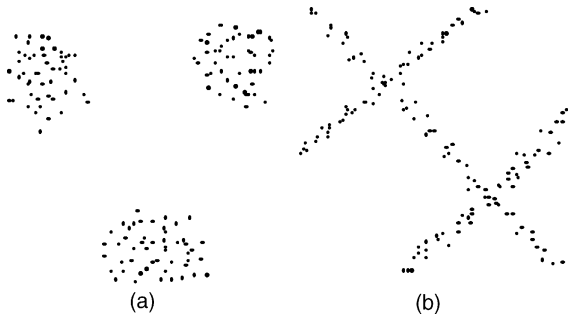

Fig. 3. (a) Three clouds data and (b) three lines data.

covering what seems to be true (i.e., that $c = 3$) for these two data sets by applying VCV to clustering outputs gotten for several values of c, and examining the images $\{I(R^*)\}$ associated with each test.

The *fuzzy c-means* (FCM, Bezdek, 1981) algorithm was used on the cloud data; and the *fuzzy c-regression models* (FCRM, Hathaway and Bezdek, 1993) algorithm was used on the linear data. In all experiments, the norm in (1) was Euclidean, the fuzzification parameter was set at $m = 2$, and iteration was terminated when the (absolute) maximum of change of each membership value $U_{ik} \leqslant 0.0001$. For $c = 2$, 3, 4 and 10, the algorithms were initialized by assigning $150/c$ data to each cluster (for $c = 4$, two of the four clusters had 37 points, and two had 38). The datum to prototype distances needed to construct the reordering in (3) were: for FCM, $d_{ik} = ((v_{i1} - x_{k1})^2 + (v_{i2} - x_{k2})^2)^{0.5}$; and for FCRM, $d_{ik} = |x_{k2} - (v_{i1}x_{k1} + v_{i2})|$. The distances in R^* were linearly scaled so that the minimum distance corresponded to black and the maximum distance corresponded to white in $I(R^*)$.

Fig. 4(a)–(d) give the results for the cloud data and consist of intensity image representations of the reordered pairwise dissimilarities calculated using (3). Notice that $c = 2$ fails to produce a clear diagonal structure, while the images for $c = 3$, 4, and 10 give a clear indication of three clusters. A very important feature of this approach is that the actual number of clusters can be indicated even when a larger number of clusters assumed c is used. This suggests that the correct value of c can sometimes be found by running the algorithm with

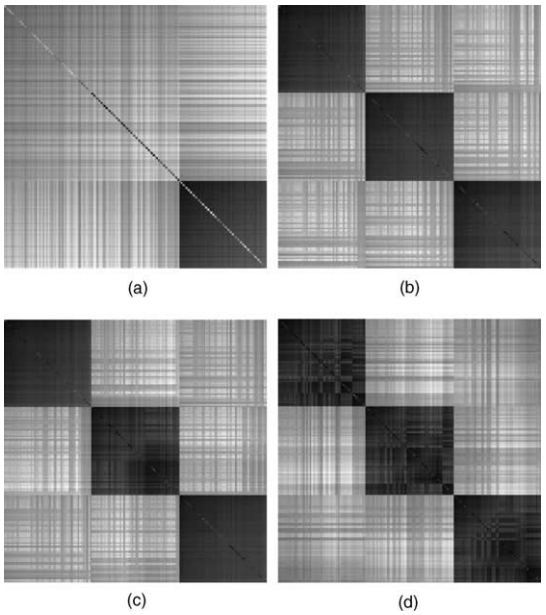

Fig. 4. (a) Three clouds, FCM @ $c = 2$, (b) three clouds, FCM @ $c = 3$, (c) three clouds, FCM @ $c = 4$ and (d) three clouds, FCM @ $c = 10$.

a large value of c, and then ascertaining its correct value from the visual evidence in the VCV image.

Fig. 5(a)–(d) give the VCV results for the linear data. Three diagonal blocks dominate in this case also, although the images are not nearly as clean as those for the cloud data. The reason for this is that FCRM has difficulty in making strong membership assignments in the overlap regions, where some data fit more than one of the linear prototypes extremely well. Severe overlap is indicated in the images by dark pixels in off-diagonal blocks. In this case there appears to be some deterioration of the clear indication of three clusters as c gets large (i.e., at $c = 10$).

As a final example, we demonstrate the VCV approach using the famous Iris data. Iris is a set of $n = 150$ points in $s = 4$ dimensions with three physical subspecies of Iris plants each represented by 50 vectors. Iris was collected by Anderson (1935), first published and used in a computational setting by Fisher (1936), and subsequently used (and misused) by just about every cluster analyzer on the planet. We used the data as listed in (Fisher, 1936). The fuzzy c-means algorithm was used with

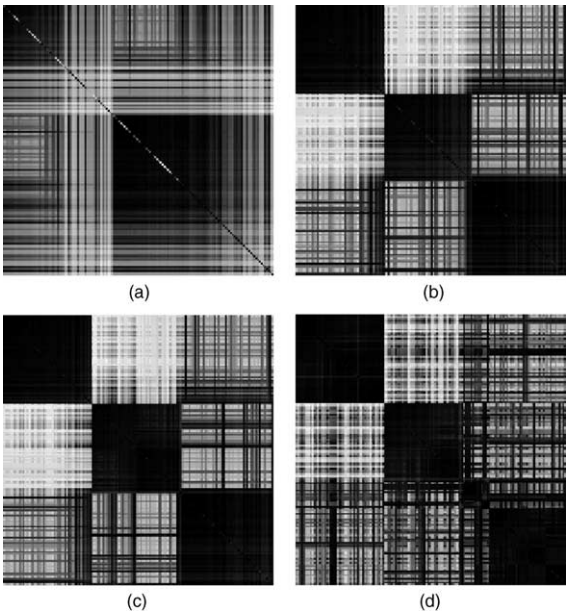

Fig. 5. (a) Three lines, FCRM @ $c = 2$, (b) three lines, FCRM @ $c = 3$, (c) three lines, FCRM @ $c = 4$ and (d) three lines, FCRM @ $c = 10$.

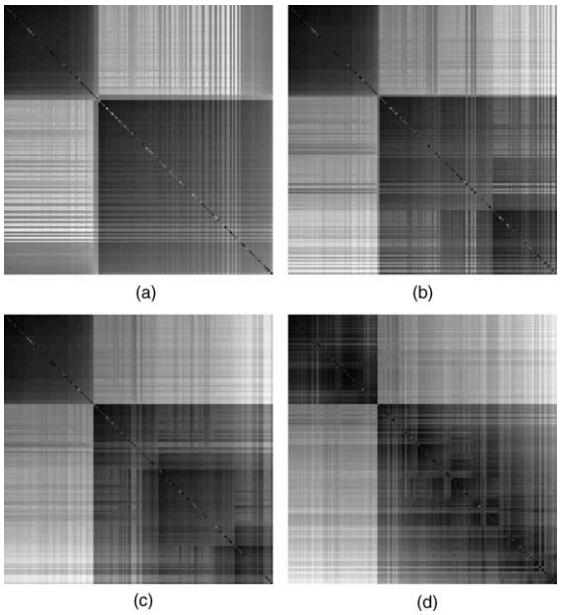

Fig. 6. (a) Iris, FCM @ $c = 2$, (b) Iris, FCM @ $c = 3$, (c) Iris, FCM @ $c = 4$ and (d) Iris, FCM @ $c = 10$.

parameters as given for the three clouds examples to cluster Iris at $c = 2$, 3, 4 and 10. The results are given in Fig. 6. While the Iris data has three different *physically* labeled classes, the images clearly indicate two clusters. Is this a reasonable result? We think so. It is clear from visual examination of sections of the 4D data in two dimensions that there are but two geometric clusters in Iris. For computational evidence, see Bezdek and Pal (1998), who tested a total of 21 validity indexes, for $c = 2, \ldots, 10$, on the Iris data. In their experiment, two indices indicated three clusters, two indices indicated four clusters, and 17 indices indicated two clusters. Note that the use of these indices requires multiple runs of a prototype clustering algorithm for $c = 2, 3, \ldots,$ while VCV requires only a single run of the prototype clustering algorithm for a single overestimate of the true number of clusters. Virtually indistinguishable images were obtained for the Iris data using the possibilistic c-means algorithm, where the values for the PCM parameters $\eta_1, \ldots, \eta_c$ were calculated using the FCM output in Eq. (9) of Krishnapuram and Keller (1993).

5. Discussion and conclusions

We introduced a new visual display approach to cluster validity for prototype generator clustering algorithms which we call the VCV method. Our new approach is similar to the original SHADE display of Ling (1973), but uses an inter-datum dissimilarity that gives (relatively) large values for data belonging to different clusters. A reordered matrix of dissimilarities is displayed as an intensity image, and the number of dark diagonal blocks in the image presumably indicates the "actual" number of clusters in the data. The reordering is based on dissimilarities which are available directly from any prototype generator clustering algorithm, and therefore, the VCV method does not have a high (additional) computational cost. Experiments using cloud and linear data indicate that the method is a promising addition to current cluster validity methodology. Much more testing, on datasets in high dimensions, is, of course, both necessary and desirable before a large amount of confidence can be placed in the VCV method. Since scalar measures of cluster validity are famously unreliable, it will be interesting to see

whether this visual method is more reliable across a wide range of data types and parameter values than the standard validity functional approach.

References

Anderson, E., 1935. The Irises of the Gaspe peninsula. Bull. Am. Iris Soc. 59, 2–5.

Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York.

Bezdek, J.C., Hathaway, R.J., 2002. VAT: A tool for visual assessment of (cluster) tendency. In: Proc. IJCNN 2002. IEEE Press, Piscataway, NJ, pp. 2225–2230.

Bezdek, J.C., Pal, N.R., 1998. Some new indexes of cluster validity. IEEE Trans. SMC, Part B Cybernet. 28, 301–315.

Bezdek, J.C., Keller, J.M., Krishnapuram, R., Pal, N.R., 1999. Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer, Norwell, MA.

Bezdek, J.C., Li, W.Q., Attikiouzel, Y.A., Windham, M.P., 1997. A geometric approach to cluster validity for normal mixtures. Soft Comput. 1, 166–179.

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

Hathaway, R.J., Bezdek, J.C., 1993. Switching regression models and fuzzy clustering. IEEE Trans. Fuzzy Syst. 1, 195–204.

Hubert, L.J., Arabie, P., 1985. Comparing partitions. J. Classification 2, 193–218.

Johnson, R.A., Wichern, D.A., 1992. Applied Multivariate Statistical Analysis, 3rd edition Prentice-Hall, Englewood Cliffs, NJ.

Krishnapuram, R., Keller, J.M., 1993. A possibilistic approach to clustering. IEEE Trans. Fuzzy Syst. 1, 98–110.

Ling, R., 1973. A computer generated aid for cluster analysis. Commun. ACM 16, 355–361.

Tou, J., Gonzalez, R., 1974. Pattern Recognition Principles. Addison-Wesley, Reading, MA.