File Naming

Each file is assigned a unique name that will allow for individual researchers to compare performance on specific images.

Storage Requirements

The number of megabytes of storage for each portion of the database are now listed.

Megabytes for Each Portion of the Database

Database Component	Training	Testing
Grayscale cities	152	16
Grayscale states	83	9
Grayscale ZIP Codes	193	9
Mixed bi-tonal alphabetics and numerics	52	6
Bi-tonal numerics only	38	10

Overall, approximately 600 Mbytes of the CDROM are used. This includes the storage needed for formatting information.

Availability

The database described in this correspondence is available from the Center of Excellence for Document Analysis and Recognition (CEDAR) at the State University of New York at Buffalo.

ACKNOWLEDGMENT

Prof. S. N. Srihari, Director of CEDAR, was the Principal Investigator for the handwritten ZIP Code recognition project. Dr. J. Tan of Arthur D. Little, Inc., helped shape the composition of the database. Dr. E. Cohen assisted in the supervision of the data capture process. J. Giattino directed the formatting of the CDROM.

REFERENCES

[1] R. Bradford and T. Nartker, "Error correlation in contemporary OCR systems," in *Proc. 1st Int. Conf. Document Anal. Recogn.*, Saint-Malo, France, Sept. 30–Oct. 2, 1991, pp. 516–524.

[2] E. Cohen, J. J. Hull, and S. N. Srihari, "Understanding handwritten text in a structured environment: Determining ZIP Codes from addresses," *Int. J. Pattern Recogn. Artif. Intell.*, vol. 5, no. 1 & 2, pp. 221–264, 1991.

[3] T. K. Ho, J. J. Hull, and S. N. Srihari, "Combination of decisions by multiple classifiers," in *Structured Document Image Analysis*, H. S. Baird, H. Bunke, and K. Yamamoto, Eds. New York: Springer-Verlag, 1992, pp. 188–202.

[4] J. J. Hull, A. Commike, and T. K. Ho, "Multiple algorithms for handwritten character recognition," in *Proc. Int. Workshop Frontiers Handwriting Recogn.*, Montreal, Canada, Apr. 2–3, 1990, pp. 117–130.

[5] J. J. Hull, T. K. Ho, J. Favata, V. Govindaraju, and S. N. Srihari, "Combination of segmentation-based and wholistic handwritten work recognition algorithms," in *Proc. From Pixels to Features III: Int. Workshop Frontiers Handwriting Recogn.*, Bonas, France, Sept. 23–27, 1991, pp. 229–240.

[6] G. Nagy, "Candide's practical principles of experimental pattern recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 199–200, Mar. 1983.

[7] ——, "At the Frontiers of OCR," *Proc. IEEE*, vol. 7, pp. 1093–1100, July 1992.

[8] J. C. Simon and O. Baret, "Cursive word recognition," in *Proc. From Pixels to Features III: Int. Workshop Frontiers Handwriting Recogn.*, Bonas, France, Sept. 23–27, 1991, pp. 1–20.

[9] C. Y. Suen, presented at the Int. Workshop Frontiers Handwriting Recogn., Montreal, Canada, Apr. 2–3, 1990.

Feature-Preserving Clustering of 2-D Data for Two-Class Problems Using Analytical Formulas: An Automatic and Fast Approach

Ja-Chen Lin and Wen-Hsiang Tsai

***Abstract*—We propose in this correspondence a new method to perform two-class clustering of 2-D data in a quick and automatic way by preserving certain features of the input data. The method is analytical, deterministic, unsupervised, automatic, and noniterative. The computation time is of order n if the data size is n, and hence much faster than any other method which requires the computation of an n-by-n dissimilarity matrix. Furthermore, the proposed method does not have the trouble of guessing initial values. This new approach is thus more suitable for fast automatic hierarchical clustering or any other fields requiring fast automatic two-class clustering of 2-D data. The method can be extended to cluster data in higher dimensional space. A 3-D example is included.**

***Index Terms*— Two-class clustering, cluster representatives, feature-preserving, analytical formulas, decision boundary, automatic fast clustering, *k*-means, hierarchical methods.**

I. INTRODUCTION

Two-class clustering problems are frequently encountered in real applications. For example, block truncation coding for image compression [1], divisive clustering for hierarchical clustering [2], binary decision tree construction, etc. It is therefore desired to develop a fast automatic method that can be employed to partition an input set H of n patterns into two classes. Unfortunately, most of the clustering tools developed so far, such as the k-means method [3], the divisive method using a dissimilarity matrix [4], etc., are iterative and thus unsuitable for performing fast automatic two-class clustering.

It is desirable to avoid iterative computation by using mathematical formulas to express the decision boundary, which separates the two classes, in terms of the input patterns directly. One way of achieving this goal based on the moment-preserving principle is explained below. When the n input patterns are one-dimensional, say, forming a set $H = \{x_i\}_{i=1}^n$, the partition of H into two disjoint clusters H_A and H_B is an easy job. We may assume that every pattern in cluster H_A resembles (in some sense) a single point x_A, and similarly, every pattern in cluster H_B resembles another single point x_B. The two points x_A and x_B are called cluster representatives. Assume further that the fractions of the numbers of patterns in H_A and H_B are p_A and p_B, respectively. It is clear that

$$p_A + p_B = 1. \tag{1}$$

By preserving the first three moments, i.e., by requiring that

$$p_A x_A^k + p_B x_B^k = \overline{x^k} = \frac{1}{n}\sum_{i=1}^{n} x_i^k \quad \text{for } k = 1, 2, \text{ and } 3, \tag{2}$$

and by the natural requirement (1), we can solve Eqs. (1) and (2) to obtain the four unknowns $\{x_A, x_B, p_A, p_B\}$. The solution can be

Manuscript received March 9, 1992; revised January 5, 1993. This work was supported by the National Science Council, Republic of China, under Contract NSC 81-0408-E-009-589. Recommended for acceptance by Associate Editor R. P. W. Duin.

The authors are with the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050, Republic of China.

IEEE Log Number 9214426.

found in [5] or [6]. Having obtained x_A, x_B, p_A and pB, one may define, as Tsai did in [6], the decision boundary x to be the p_A-tile.

When the input patterns are two-dimensional data, we found that moment-preserving is no more so easy to apply to partition the given data into two clusters H_A and H_B. To see this, let $H = \{(x_i, y_i)\}_{i=1}^n$ be the given data to be partitioned, and let the fractions of the numbers of patterns in H_A and H_B again be p_A and p_B, respectively. Assume also that every pattern in cluster H_A resembles a cluster representative (x_A, y_A), and every pattern in cluster H_B resembles another cluster representative (x_B, y_B). The goal is again to obtain some formulas which can be used to compute

$$\{x_A, y_A, x_B, y_B, p_A, p_B\} \tag{3}$$

easily. To get the solutions of these six unknowns, we need five additional equations other than (1). A natural try is to construct these five equations by applying the moment-preserving principle to the first five moments, resulting in

$$p_A x_A^j y_A^k + p_B x_B^j y_B^k = \overline{x^j y^k} = \frac{1}{n}\sum_{i=1}^{n} x_i^j y_i^k \quad \text{for } j + k = 1 \text{ and } 2. \tag{4}$$

In other words, it may be tried to preserve each of the five moments $\{\bar{x}, \bar{y}, \bar{xy}, \bar{x^2}, \bar{y^2}\}$ Unfortunately, the six equations in (1) and (4) are themselves a contradictive equation set, and no set of $\{x_A, y_A, x_B, y_B, p_A, p_B\}$ can be generated. The proof is straightforward and thus omitted.

It is therefore the purpose of this study to find some other features to replace the roles of the five moments $\{\bar{x}, \bar{y}, \bar{xy}, \bar{x^2}, \bar{y^2}\}$. It is found that preserving

$$\{\bar{x}, \bar{y}, \bar{r}, \bar{\theta}, \phi\} \tag{5}$$

is a very good solution. The definition of $\bar{r}, \bar{\theta}$, and ϕ will be given in the next section.

The remainder of this papar is organized as follows. In Section II the formulas to compute $\{x_A, y_A, x_B, y_B, p_A, p_B\}$ by preserving the set of features of (5) are introduced, and the method to construct the decision boundary to separate the two classes is also discussed. In Section III, we give some experimental results. In Section IV, the method is compared with the k-means and hierarchical methods. Then we discuss in Section V the safe way to apply our method. The possibility to extend the method to higher dimensional data, and the general rule to construct a feature-preserving method are both discussed in Section VI. Finally, concluding remarks are given in Section VII.

II. Analytical Formulas for Two-Class Clustering of 2-D Data

Without the loss of generality, we assume that $(\bar{x}, \bar{y})$, which is the centroid of the given n–point system $H = \{(x_i, y_i)\}_{i=1}^n$, to be the origin (0, 0). If this is not the case, a translation by the amount of $\triangle x = \bar{x}$ and $\triangle y = \bar{y}$ should first be done, and then, after $\{x_A, y_A, x_B, y_B\}$ are generated, these four numbers should be transformed back to the old coordinates by an inverse translation with $\triangle x = -\bar{x}$ and $\triangle y = -\bar{y}$. Similarly, we also assume that a preprocessing step of rotation has been performed so that the [1] of H coincides with the y-axis. Note that the principal axis is a line going through the centroid $(0, 0)$ of H with directional angle ϕ characterized by the following two equations [7]:

$$\begin{cases} \tan 2\phi = 2\bar{xy}/(\bar{x^2} - \bar{y^2}), \\ (\bar{x^2} - \bar{y^2})\cos 2\phi + 2\bar{xy}\sin 2\phi > 0. \end{cases}$$

Let $\{(r_i, \theta_i)\}_{i=1}^n$ be the polar coordinates of the (standardized) points $\{(x_i, y_i)\}_{i=1}^n$. Similarly, let (r_A, θ_A) and (r_B, θ_B) be the polar coordinates of the cluster representative (x_A, y_A) and (x_B, y_B), respectively. The preserving of $\{\bar{x}, \bar{y}, \bar{r}, \bar{\theta}\}$, with $\bar{r}$ and $\bar{\theta}$ defined as $\bar{r} = \frac{1}{n}\sum_{i=1}^n r_i$ and $\bar{\theta} = \frac{1}{n}\sum_{i=1}^n \theta_i$, means that

$$p_A x_A + p_B x_B = \bar{x} = 0 \tag{6}$$

$$p_A y_A + p_B y_B = \bar{y} = 0 \tag{7}$$

$$p_A r_A + p_B r_B = \bar{r} \tag{8}$$

$$p_A \theta_A + p_B \theta_B = \bar{\theta} \tag{9}$$

Equations (6) and (7) imply that

$$x_A = -\frac{p_B}{p_A} x_B \tag{10}$$

$$y_A = -\frac{p_B}{p_A} y_B \tag{11}$$

As a result,

$$r_A = \sqrt{x_A^2 + y_A^2} = \frac{p_B}{p_A}\sqrt{x_B^2 + y_B^2} = \frac{p_B}{p_A} r_B \tag{12}$$

which in turn means that

$$r_B = \frac{\bar{r}}{2p_B} \tag{13}$$

because

$$\bar{r} = p_A r_A + p_B r_B = p_A\left(\frac{p_B}{p_A} r_B\right) + p_B r_B = 2p_B r_B \tag{14}$$

can be derived from (8).

On the other hand, the straight line connecting the system centroid $(\bar{x}, \bar{y}) = (0, 0)$ and (x_A, y_A) is identical to the the straight line connecting $(\bar{x}, \bar{y})$ and (x_B, y_B), for both lines go through $(0, 0)$ and have the same slope

$$s = \frac{y_A - \bar{y}}{x_A - \bar{x}} = \frac{y_A}{x_A} = \frac{-(p_B/p_A)y_B}{-(p_B/p_A)x_B} = \frac{y_B}{x_B} = \frac{y_B - \bar{y}}{x_B - \bar{x}}$$

by (10) and (11).

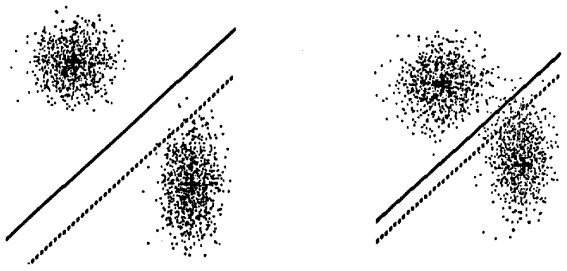
Since the three points (x_A, y_A), $(0, 0)$, (x_B, y_B) are on the same straight line, and (0,0) is between (x_A, y_A) and (x_B, y_B) by (10) and (11), we see that the polar angles θ_A and θ_B of the two points (x_A, y_A) and (x_B, y_B) must differ from each other by π. Without the loss of generality, let $\theta_B = \theta_A + \pi$. Equation (9) thus becomes $\bar{\theta} = p_A\theta_A + p_B(\theta_A + \pi) = (p_A + p_B)\theta_A + p_B\pi = \theta_A + p_B\pi$. We then have

$$p_B = (\bar{\theta} - \theta_A)/\pi. \tag{15}$$

If we can obtain the value of θ_A, then the values of all six unknowns $\{x_A, y_A, x_B, y_B, p_A, p_B\}$ can be computed easily because we can first compute p_B and p_A by (15) and

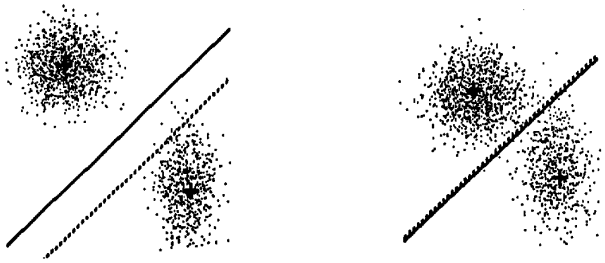
$$p_A = 1 - p_B, \tag{16}$$

respectively. Then we can evaluate r_B and r_A by the formulas $r_B = \bar{r}/(2p_B)$ and $r_A = r_B p_B/p_A$, as stated in (13) and (12). Finally, $x_A = r_A\cos\theta_A$, $y_A = r_A\sin\theta_A$, $x_B = r_B\cos\theta_B = r_B\cos(\theta_A + \pi) = -r_B\cos\theta_A$, and $y_B = r_B\sin\theta_B = r_B\sin(\theta_A + \pi) = -r_B\sin\theta_A$. Therefore, all we have to do now is to assign a suitable value to θ_A which is defined earlier to be the polar angle of (x_A, y_A), or equivalently, the directional angle of the straight line going through the three points (x_A, y_A), $(0, 0)$, and (x_B, y_B).



(a) (b)

Fig. 1. Computed pairs of cluster representatives (marked by two crosses) for some 2100-point sets in each of which the subpopulations of the two clusters are approximately 50% vs. 50%. Both the dotted line l and the solid line l' may be used as decision boundaries.



(a) (b)

Fig. 2. Same as Fig. 1, except that the data used has been replaced by some 67% vs. 33% data.

A reasonable and convenient way to assign a suitable value to θ_A is to require the preserving of the principal axis orientation, which results in $\theta_A = \phi$ with ϕ denoting the directional angle of the principal axis of the given 2-D data set $H = \{(x_i, y_i)\}_{i=1}^n$. The fact that preserving the principal axis orientation implies $\theta_A = \phi$ is proved in [8].

Having obtained the two cluster representatives (x_A, y_A) and (x_B, y_B), the decision boundary to separate the two classes can be defined to be the straight line l which is perpendicular to the line segment $\overline{AB}$ connecting $A = (x_A, y_A)$ and $B = (x_B, y_B)$ such that l splits the 2-D plane into two half planes and the half plane containing (x_A, y_A) has np_A patterns. Several examples illustrating this kind of decision boundary will be given in the next section.

However, if we want to design a classifier in a much quicker way, an alternative method is to use the straight line l' perpendicular to and bisecting the line segment $\overline{AB}$ as the decision boundary. The time needed to generate l' is much shorter than that for generating l. Of course, when we use this easy-to-obtain decision boundary l', the fractions of the numbers of patterns in clusters H_A and H_B do not necessarily agree with the values of p_A and p_B computed by (16) and (15). However, this trade–off between the reduction of the computation time and the sticking to the estimated population distribution is worthy, especially when a quick design of classifiers is the main goal. After all, the computed p_A and p_B themselves are just estimated values, instead of the exact values of the population distribution.

III. Illustrative Examples

In this section, we show the results of using our formulas to compute $\{x_A, y_A, x_B, y_B, p_A, p_B\}$ and construct both kinds of the decision boundaries l and l' for some randomly generated data set $H = \{(x_i, y_i)\}_{i=1}^n$. The results are shown in Figs. 1 and 2. We first use an algorithm for generating random numbers to create a 2-D set S_A, and use the same algorithm to create another 2-D set S_B. These two sets are then merged together to form H. The proposed approach is finally applied to H. The computed cluster representatives are marked by two crosses, and the computed decision boundaries l and l' are shown as a dotted straight line and a solid straight line, respectively, in each figure. As stated at the end of the last section, both l and l' are perpendicular to the line segment connecting the two generated crosses. The only difference is that l' bisects the line segment connecting the two generated crosses while l guarantees that the fractions of the populations on the two sides of l match the values p_A and p_B computed by (16) and (15). In both Figs. 1 and 2, H contains 2100 points. In Fig. 1, each H is designed to be the mergence of two disjoint subsets, and the number of points in each subset is approximately 1050 points. In Fig. 2, however, one of the subsets contains approximately 1400 points and the other subset contains about 700 points. It is observed that the values of p_A and p_B computed by formulas [16] and [15] do get close to the expected ratio of 50% vs. 50% in Fig. 1, and 67% vs. 33% in Fig. 2. It is also observed that the computation time taken to generate $\{x_A, y_A, x_B, y_B, p_A, p_B\}$ is only about two seconds for a 2100-point set H using an IBM PC with an 80386 processor.

IV. Comparison with K-Means and Hierarchical Methods

In this section, we compare our method with some clustering methods available in the IMSL [9] package, namely, the hierarchical agglomerative methods [10] and the k-means method. The k-means method used in IMSL is based on an algorithm written by Hartigan and Wong [11], and this program tries to reduce the total sum E of the within-cluster squares. When $k = 2$, i.e., when it is applied to the 2-class problem discussed here, E is expressed as

$$E = E(H_A, H_B) = \sum_{a \in H_A} [d(a, \bar{a})]^2 + \sum_{b \in H_B} [d(b, \bar{b})]^2 \qquad (17)$$

for any two-class partition $\{H_A, H_B\}$ of the given input set H. As usual, $\bar{a}$ and $\bar{b}$ denote the centroids of clusters H_A and H_B, respectively. In Fig. 3, we compare the clustering results of ours with those of the k-means program provided by IMSL. It can be seen that our method yields clustering results similar to those of the k-means method. The two methods give identical results for (a), (c)–(f). Only two points in (b) and one point in (g) are clustered differently. The data shown in Fig. 3 were those proposed by Nagy [12]. We use these data sets because they illustrate cluster distributions of typical clustering problems such as "neck," unequal cluster populations, etc. Similar typical clustering problems were also pointed out by Zahn [13].

The hierarchical agglomerative methods available in the IMSL package were also tested. It was observed that none of the proposed, the k-means, and the hierarchical agglomerative methods can cluster well the nonlinear data set (g) and the nonspherical data set (d) sketched in Fig. 3. When there are necks in the data, like in (b) and (c), our method cut necks, but not exactly (i.e., a little portion of the patterns is misclustered). Similar troubles also exist for the k-means method (see (c) of Fig. 3), the complete linkage method, and Ward's method. As for the single linkage method, it is terribly inapplicable to data sets with "necks" because of the chaining effect.

The computation time used is also an important index to evaluate distinct methods. In general, the time for the k-means method is about two times longer than ours, but the time needed for either of

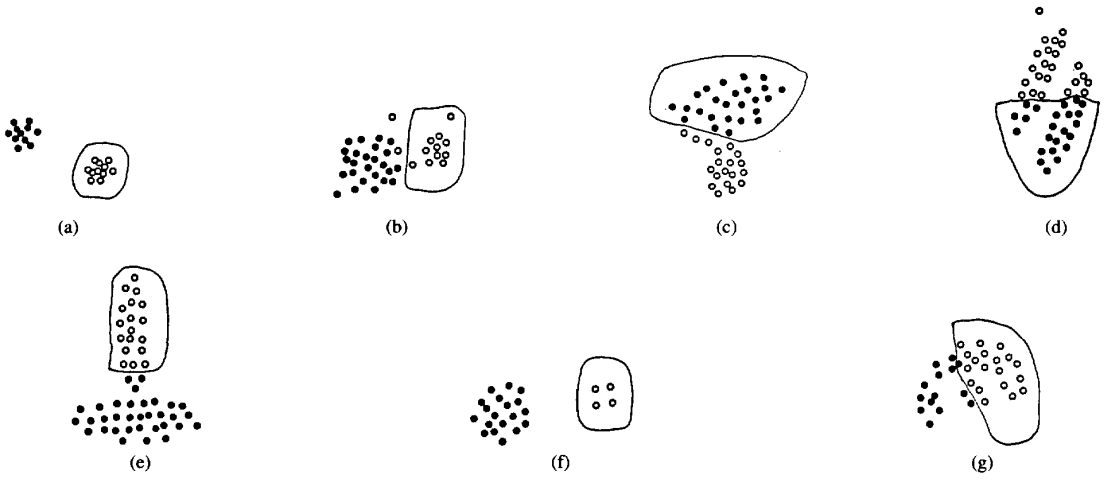


Fig. 3. Comparing the results of applying the proposed method with those of applying the k-means method. The data sets are those used by Nagy [12]. The black-dot cluster and the white-dot cluster represent the two clusters detected by our method; while the points within the closed curve and the points not enclosed by the curve form two other clusters detected by a k-means algorithm proposed by Hartigan and Wong[11], which intended to decrease the total sum of the within-cluster squares.

the two hierarchical agglomerative methods becomes several hundred times longer than ours if the data size is 1000. We have also tried a hierarchical divisive method given in [4]. The clustering results are similar to ours (when applied to Nagy's data sets), but the computation time also becomes several hundred times longer than ours when $n = 1000$ although it is faster than the hierarchical agglomerative methods. A reason to explain this fact is that our method has the work load of order n because all we need is to compute the average values $\{\bar{x}, \bar{y}, \bar{r}, \bar{\theta}, \bar{x^2}, \bar{y^2}, \bar{xy}\}$ of n patterns, where $\{\bar{x^2}, \bar{y^2}, \bar{xy}\}$ are used to obtain the principal axis. On the other hand, it is also easy to see that each iteration of the k-means method has the computation load of order n. However, each of the hierarchical methods, no matter agglomerative or divisive, has the work load of at least n^2 because of the construction of an n-by-n dissimilarity matrix [4]. Also notice that the computer storage problem for this n-by-n matrix. This makes the hierarchical methods unsuitable for personal computers when the sample size is $n = 1000$.

As for the k-means method, although the computation time and storage are of no trouble, outliers far away from the rest of the data will cause unexpected clustering results when any of them is taken as one of the initial guess points. The clustering result might be that one point forms a cluster and the other $n - 1$ points form the other. On the other hand, outliers only affect our method a little because the average functions $\bar{x}, \bar{y}, \bar{r}$, etc., will "smooth" their impact.

Therefore ,our method is quick and storage–saving, and has no worry about the choice of the initial guess, the number of iterations needed, the problem of being convergent or not, etc. As long as the quality of the clustering result is not too far away from those obtained by other methods, our method deserves a try, especially for data of large sizes. Therefore, we try to find in the next section the limitation of our method and discuss the situations in which it is safe to use our method.

V. Applying Proposed Method in a Safe Way

Since our method uses a straight line l' to split data into two clusters, we concentrate our discussion on linearly separable problems only. When the two clusters are circular-like (hollow or not), our method works well even if the two clusters touch each other (see (a)–(c) of Fig. 4 for illustrations). However, when there are well-elongated shapes in the data, our method may fail if the two clusters are too close to each other. For example, the data in (l) and (m) of Fig. 4 can be clustered well by our method (even though the two clusters touch each other), but the data in (d), (f), (h) (and maybe (j)) are not suitable to our method. Note that in (f) and (j) our cluster representatives are close to the means of the given clusters, and in (d) and (h) although our cluster representatives are far away from the means of the given (left and right) clusters, splitting the data into upper and lower halves, as shown in (d) and (h), is not worse than splitting the data into left and right halves, as in the direction shown in (e), in the sense that they give a smaller E defined in (17). In short, all the clustering results shown in Fig. 4 have small E. Moreover, the k-means method also have trouble in handling (d), (f), (h), and (j) if the policy of minimizing E is used. In fact, with the goal of minimizing E, the clustering results of the k-means method are very similar to ours for all the data sets shown in Fig. 4, with the only exception (d), of which the decision boundary generated by the k-means method is neither vertical nor horizontal, but slanted.

When the two given clusters in each of (d), (f), (h), and (j) of Fig. 4 are far away enough from each other, as sketched in (e), (g), (i) and (k), respectively, our method yields "visually" good results again. Our experience is that, if well-elongated clusters are among the data to be clustered, it is usually safe to apply our method if the two expected clusters A and B satisfy $\min\{d(a,b)|\ a \in A,\ b \in B\} \geq \frac{1}{2}\max\{\mathrm{Diam(A)}, \mathrm{Diam(B)}\}$. Here, the diameter Diam(C) of a point set C is defined as $\mathrm{Diam}(C) = \max\{d(c,c')|\ c, c' \in C\}$. Note that the inequality should be interpreted as a sufficient condition instead of a necessary condition. For example, both (l) and (m) of Fig. 4 can be partitioned well by our method although the sets are close to each other in each case.

At the end of this section, we give in Fig. 5 some examples of partitioning input data into two classes when the data are in fact formed of more than two clusters. In some cases, our method might improperly cut one of the clusters into two halves (see (b) and (d)). Since improper cutting is possible (a trouble also occurs to the k-

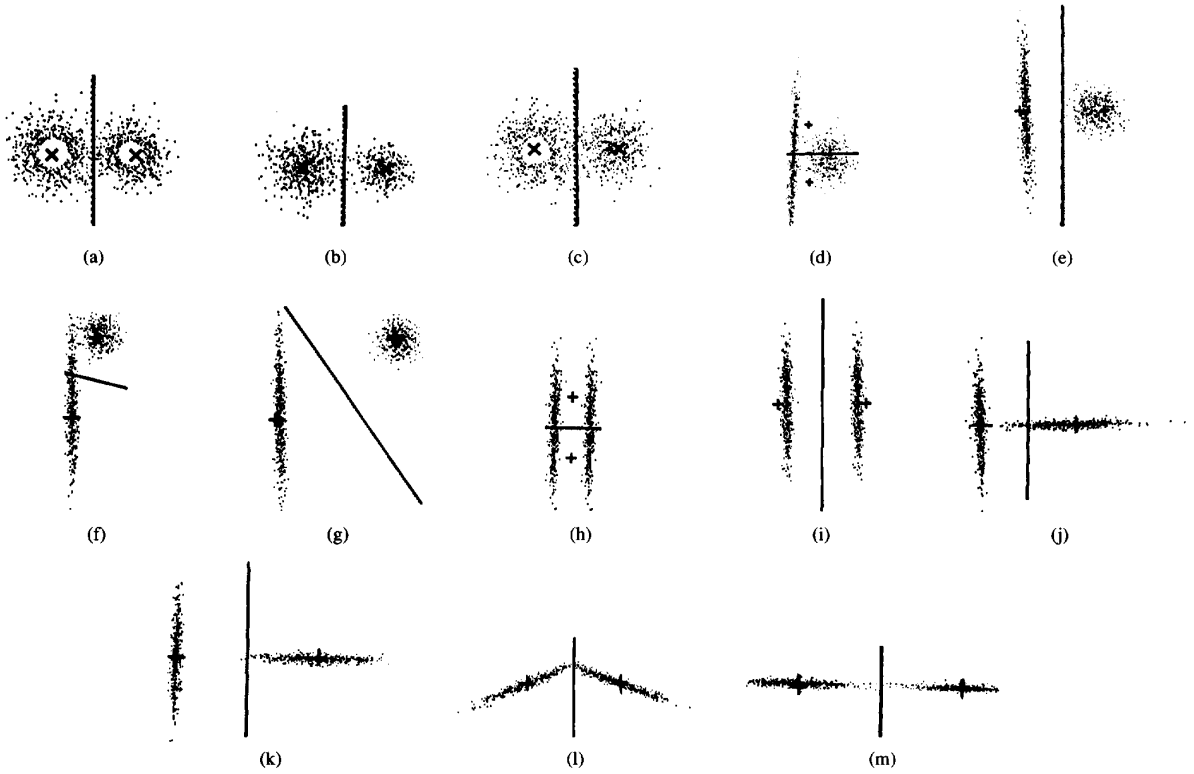


Fig. 4. Applying our procedure to some linear-separable data sets of which the two clusters are hollow circular-like vs. circular-like, circular-like vs. well-elongated, well-elongated vs. well-elongated, etc. As usual, the two crosses are the two detected cluster representatives, and the solid line is the detected decision boundary l'. Note that (d), (f), (h), and (j) are examples where applying our method is not suitable although these partitions have small total sums E of within-cluster squares. Also note that the crosses in (f) and (j) have been very close to the means of the expected clusters.

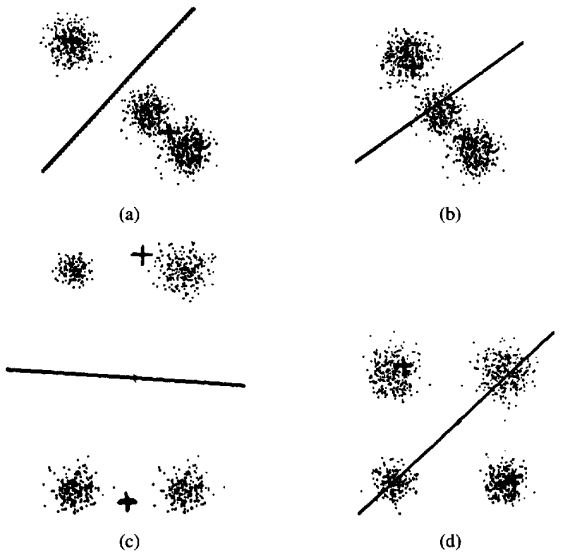


Fig. 5. Applying the proposed two-class clustering method when the input set is in fact formed of three clusters ((a) and (b)) or four clusters ((c) and (d)).

means method and some hierarchical methods), it is necessary to merge back the clusters, which are improperly cut, after applying our method repeatedly.

With the k-means (which minimizes E) and our methods both applied to the data shown in Figs. 3–5, we observed that these two methods have similar clustering results and requires approximately the same amounts of computation time. We therefore classify our method as an automatic fast clustering method with performance similar to that of the k-means method which minimizes the E. Note that the k-means method has the trouble of choosing an initial guess, however.

VI. Higher Dimensional Problems and Features to be Preserved

In this section, we discuss the possibility of extending our method to higher dimensional problems, and the general principle to choose the features to be preserved.

Without the loss of generality, we discuss the 3-D case only. However, the method discussed below can be generalized to any higher dimension. Assume that a given set $H = \{(x_i, y_i, z_i)\}_{i=1}^{n}$ is to be split into two clusters H_A and H_B with cluster representatives being (x_A, y_A, z_A) and (x_B, y_B, z_B), respectively. The goal is again to obtain some formulas to compute

$$\{x_A, y_A, z_A, x_B, y_B, z_B, p_A, p_B\} \tag{18}$$

easily. Notice that there are two more unknowns $\{z_A, z_B\}$ other than the six unknowns $\{x_A, y_A, x_B, y_B, p_A, p_B\}$ appearing in (3) for the 2-D case. In general, if the dimensionality increases from d to $d+1$,

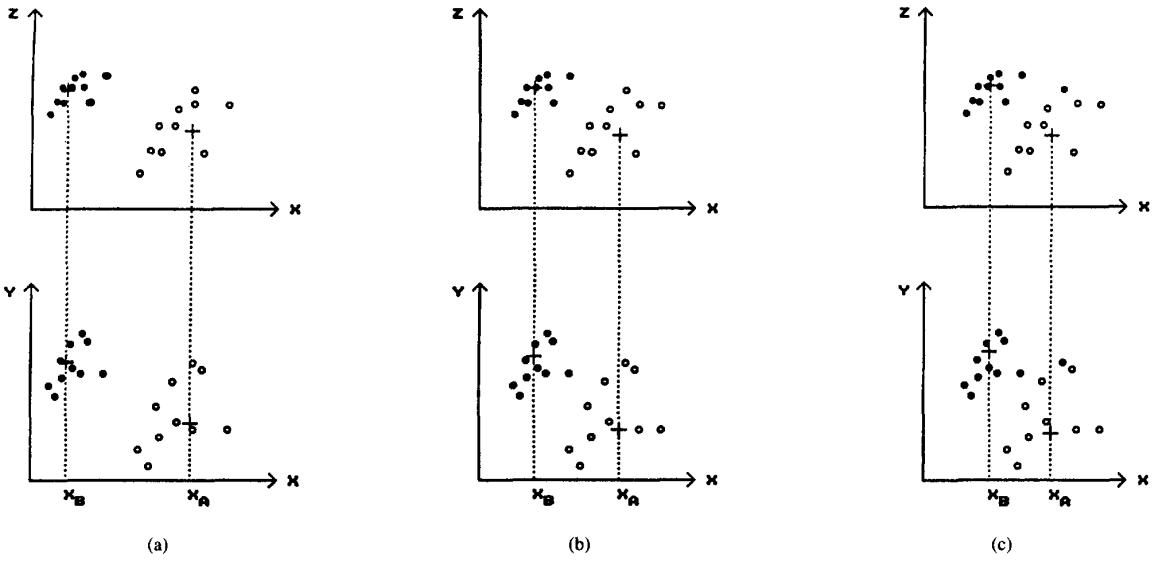


Fig. 6. An expample illustrating the 3-D clustering result using the 3-D version of the proposed method. Originally, there are two clusters, each contains ten 3-D points. Our mehtod clustered these twenty 3-D points into a white-dot cluster and a black-dot cluster, as shown in (a). To use 2-D coordinate systems to illustrate the 3-D relationship between these two clusters, we had projected the 3-D coordinates of the 20 input patterns and the detected representatives (x_A, y_A, z_A) and (x_B, y_B, z_B), marked by crosses, onto the x–y and x–z planes, respectively. When these two clusters move toward each other, as in (b), our clustering result is still perfect. When the two clusters are tied together, as in (c), our method reassign one of the white points into the black class (see the rightmost black point in (c)).

two more unknowns are created, and hence two more equations are needed. The first additional equation needed can of course be set up by preserving the mean value of the $(d+1)$th space variable, for example, by requiring $p_A z_A + p_B z_B = \bar{z}$ in the 3-D case. On the other hand, the second additional equation can be set up by preserving the directional angle component of the principal axis. For example, in the 3-D case, if the spherical coordinate system is used, and if the origin is set to be the centroid of H, then the eight equations we used to solve the eight unknowns in (18) is the equation $p_A + p_B = 1$ and the equations obtained by preserving $\{\bar{x}, \bar{y}, \bar{z}, \bar{r}, \bar{\theta}, \theta_{P.A.}, \phi_{P.A.}\}$ with "P.A." denoting "principal axis" of H and $\bar{r}$ the average of the 3-D radii. Note that the principal axis of a set in d-dim space is the eigenvector corresponding to the largest eigenvalue of the d–by–d correlation matrix. After certain derivations, the solution is $p_B = (\bar{\theta} - \theta_A)/\pi = (\bar{\theta} - \theta_{P.A.})/\pi$; $p_A = 1 - p_B$; $r_A = \bar{r}/(2p_A)$; $\theta_A = \theta_{P.A.}$; $\phi_A = \phi_{P.A.}$; $x_A = (r_A \sin\phi_{P.A.})\cos\theta_{P.A.}$; $y_A = (r_A \sin\phi_{P.A.})\sin\theta_{P.A.}$; $z_A = r_A \cos\phi_{P.A.}$; $x_B = -(p_A/p_B)x_A$; $y_B = -(p_A/p_B)y_A$; and $z_B = -(p_A/p_B)z_A$. The decision boundary is then a plane bisecting and perpendicular to the line segment connecting the two 3-D cluster representatives.

Fig. 6 gives an illustrative example showing the clustering results of the 3-D version of our method, in which three subcases are considered to see what happens when the two clusters are away, near, or mixed. Notice that in the first two subcases (a) and (b) the clustering results are perfect, while in (c) there is only one misjudgment.

We now give a short discussion about the general principle to choose the variables to be preserved. In general, when it is d-dimensional, there are $2d+2$ unknowns $\{p_A, p_B; x_A, x_B; y_A, y_B; z_A; z_B; \cdots\}$. Besides $p_A + p_B = 1$ and the preserving of the centroid $\{\bar{x}, \bar{y}, \bar{z}, \cdots\}$, we still need $d+1$ features to be preserved. It seems that a lot of features might be used, such as the squared moments $\{\overline{x^2}, \overline{y^2}, \overline{z^2}, \cdots\}$, the correlated moments $\{\overline{xy}, \overline{xz}, \overline{yz}, \cdots\}$, the absolute moments $\{\overline{|x|}, \overline{|y|}, \cdots\}$, the angles $\{\bar{\theta}, \bar{\phi}, \cdots\}$, the lengths $\{\sqrt{\overline{x^2+y^2}}, \sqrt{\overline{y^2+z^2}}, \sqrt{\overline{x^2+y^2+z^2}}, \cdots\}$, the special-property-axes such as the principal axis or any other axis that can be used to define shape orientations, and so on. The question is that certain combinations might form contradictive equation sets. To see whether a set of equations is contradictive or not, one can utilize well-known inequalities such as Schwartz's inequality, Holder's inequality, Minkowski's inequality, etc. When one has proved that the equation set has a solution and derived the formula to generate that solution, experiments using distinct types of data to test the clustering performance of this formula are important. Many combinations have been observed in this study to have poor clustering results. In general, maximizing the information contained in the $d+1$ selected quantities will usually be a way leading to success, if they do not form a contradictive equation set.

VII. Concluding Remarks

In this correspondence, we have proposed a new fast method to perform two-class clustering for 2-D data. The method is analytical, automatic, deterministic, unsupervised and noniterative. We have derived some simple analytical formulas to compute the two cluster representatives and the decision boundary by preserving the centroid, principal axis orientation, average polar radius and average polar angle of the input data set. The clustering result is satisfactory, and for an input set of several thousand points, the clustering procedure to generate the two desired cluster representatives and the fractions of the numbers of patterns in the two classes takes only a few seconds using a microcomputer. The computation speed is several hundred times faster than many hierarchical methods like the single linkage, complete linkage, Ward's methods, and so on when the number of patterns is about 1000. The clustering result is, roughly speaking, not worse than those of the hierarchical methods. Unlike hierarchical

clustering methods or any graph-theoretical method using nearest neighbors, our method does not compute pairwise distances between input patterns; therefore, a lot of time is saved. But, since we do not use "local information" such as nearest neighbors, the global information that we use, such as $\{\bar{x}, \bar{y}, \bar{r}, \bar{\theta}, \bar{\phi}\}$, can only give us a "rough" partition; it should be of no surprise if our method misassigns one or two points while some hierarchical methods do not. In a word, a clustering method using global information is fast, but the clustering result is usually not the best (see, for example, part (b) or (e) of Fig. 3). We have also compared the proposed method with the k-means method. Although the k-means method which minimizes the total sum of the within-clustersquares has similar clustering results and computation time as ours, the former has the trouble of choosing a safe initial guess (more specifically, it has the problem of avoiding taking an outlier as the initial guess of a cluster representative). Because of the weakness of the other methods mentioned above, the proposed method becomes very attractive for fast automatic hierarchical clustering or any other fields requiring fast automatic two-class clustering.

ACKNOWLEDGMENT

The authors wish to thank the referees for their valuable comments which lead to significant improvement on the writing of the correspondence. The authors also thank the editors, especially, Dr. A. K. Jain and Dr. R. P. W. Duin, for prompt processing of the correspondence.

REFERENCES

[1] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, vol. II. New York: Academic Press, 1982.

[2] D. Lecompte, L. Kaufman, and P. J. Rousseeuw, "Hierarchical cluster analysis of emotional concerns and personality characteristics in a freshman population," *Acta Psychiatrica Belgica*, vol. 86, pp. 324–333, 1986.

[3] R. J. Hathaway and J. C. Bezdek, "Recent convergence results for the fuzzy c-means clustering algorithms," *J. Classification*, vol. 2, pp. 29–39, 1988.

[4] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data*. New York: John Wiley & Sons, 1990.

[5] A. J. Tabatabai and O. R. Mitchell, "Edge location to subpixel values in digital imagery," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 188–201, Mar. 1984.

[6] W. H. Tsai, "Moment—preserving thresholding: A new approach," *Comput. Vision Graphics Image Process.*, vol. 29, pp. 377–393, 1985.

[7] J. C. Lin, "Universal principal axis : an easy-to-construct tool useful in defining shape orientations for almost every kind of shape," *Pattern Recognit.*, vol. 26, pp. 485–493, 1993.

[8] ——, "Analytical approach of clustering techniques with applications," Nat. Sci. Council Tech. Rep., No. NSC81–0408-E-009–589, National Chiao Tung Univ., Taiwan, Republic of China, 1992.

[9] IMSL, *Int. Mathematical and Statistical Library*. IMSL, Houston, TX, 1989.

[10] P. A. Devijver and J. Kittler, *Pattern Recognition: a Statistical Approach*. London: Prentice-Hall, 1982, p.406.

[11] J. A. Hartigan and M. A. Wong, "A k-means clustering algorithm," *Applied Statistics*, vol. 28, pp. 100–108, 1979.

[12] G. Nagy, "State of the art in pattern recognition," *Proc. IEEE*, vol. 56, pp. 836–862, May, 1968.

[13] C. T. Zahn, "Graph-theoretical methods for detecting and describing gestalt clusters," *IEEE Trans. Comput.*, vol. C-20, pp. 68–86, Jan. 1971.