Short Papers

A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry

Mu-Chun Su, *Member*, *IEEE*, and
Chien-Hsing Chou, *Student Member*, *IEEE*

Abstract—In this paper, we propose a modified version of the K-means algorithm to cluster data. The proposed algorithm adopts a novel nonmetric distance measure based on the idea of "point symmetry." This kind of "point symmetry distance" can be applied in data clustering and human face detection. Several data sets are used to illustrate its effectiveness.

Index Terms—Data clustering, pattern recognition, k-means algorithm, face detection.

1 Introduction

Cluster analysis is one of the basic tools for exploring the underlying structure of a given data set and is being applied in a wide variety of engineering and scientific disciplines such as medicine, psychology, biology, sociology, pattern recognition, and image processing. The primary objective of cluster analysis is to partition a given data set of multidimensional vectors (patterns) into so-called homogeneous clusters such that patterns within a cluster are more similar to each other than patterns belonging to different clusters. Cluster seeking is very experiment-oriented in the sense that cluster algorithms that can deal with all situations are not yet available. Extensive and good overviews of clustering algorithms can be found in the literature [1], [2], [3], [4], [5], [6]. Perhaps the best-known and most widely used member of the family is the K-means algorithm or the Isodata algorithm [7]. Lately, neural networks, for example, competitive-learning networks [8], self-organizing feature maps [9], [10], and adaptive resonance theory (ART) networks [11], [12] also have often been used to cluster data. Each approach has its own merits and disadvantages.

While it is easy to consider the idea of a data cluster on a rather informal basis, it is very difficult to give a formal and universal definition of a cluster. Most of the conventional clustering methods assume that patterns having similar locations or constant density create a single cluster. Location or density becomes a characteristic property of a cluster. Other properties of clusters are proposed based on human perception [13] or specific tasks (e.g., shape from texture [14]). The properties of clusters have to be specified before clustering is performed, however, they are usually unknown in advance.

In order to mathematically identify clusters in a data set, it is usually necessary to first define a measure of similarity or proximity which will establish a rule for assigning patterns to the domain of a particular cluster center. As it is to be expected, the measure of similarity is problem dependent. The most popular similarity measure is the Euclidean distance. The smaller the distance, the greater the similarity. By using Euclidean distance as a measure of similarity, hyperspherical-shaped clusters of equal size are usually detected. This measure is useless or even undesirable when clusters tend to develop along principal axes. To take care of hyperellipsoidal-shaped clusters, the Mahalanobis distance from $\underline{x}$ to $\underline{m}$, $D(\underline{x},\underline{m}) = (\underline{x}-\underline{m})^T\Sigma^{-1}(\underline{x}-\underline{m})$, is one of the popular choices. The matrix Σ is the covariance matrix of a pattern population, $\underline{m}$ is the mean vector, and $\underline{x}$ represents an input pattern. One of the major difficulties associated with using the Mahalanobis distance as a similarity measure is that we have to recompute the inverse of the sample covariance matrix every time a pattern changes its cluster domain, which is computationally expensive.

In fact, not only similarity measures, but also the number of clusters which cannot always be defined a priori will influence the clustering results. One popular approach to specifying the number of clusters is to increase the number of clusters and to compute some certain performance measures in each run, until partition into an optimal number of clusters is obtained. A good overview of cluster validation is given in [1]. A new approach proposed by Su et al. [15], Su and Chang [16] is to interpret self-organizing feature maps trained by the data sets. In this paper, we focus on the selection of similarity measures. We propose a nonmetric measure based on the concept of point symmetry. We intend to trade-off flexibility in clustering data with computational complexity. The paper is organized as follows: In Section 2, we briefly present the idea of point symmetry and the proposed point symmetry distance. In Section 3, the clustering algorithm employing the point symmetry distance is discussed. Several examples are used to demonstrate the effectiveness of the new algorithm. Section 4 presents the simulation results. Finally, Section 5 concludes the paper.

2 The Point Symmetry Distance

Unless a meaningful measure of distance or proximity between pairs of objects has been established, no meaningful cluster analysis is possible. The most common proximity index is the Minkowski metric, which measures dissimilarity [1]. Given N patterns, $\underline{x}_i = (x_{i1}, \cdots, x_{in})^T$, $i = 1, 2, \cdots, N$ the Minkowski metric for measuring the dissimilarity between the *j*th and *k*th patterns is defined by

$$d(j,k) = \left(\sum_{i=1}^{n} |x_{ji} - x_{ki}|^r\right)^{1/r} \tag{1}$$

where $r \geq 1$. The Euclidean distance $(r = 2)$ is one of the most common Minkowski distance metrics. By using the Euclidean distance, the conventional K-means algorithm tends to detect hyperspherical-shaped clusters.

Since clusters can be of arbitrary shapes and sizes, the Minkowski metrics seem not a good choice for situations where no a priori information about the geometric characteristics of the data set to be clustered exists. Therefore, we have to find another more flexible measure. One of the basic features of shapes and objects is symmetry. Symmetry is considered a preattentive feature which enhances recognition and reconstruction of shapes and objects [17]. Looking around us, we get the immediate impression that almost every interesting area consists of a qualitative and generalized form of symmetry. Symmetry is such a powerful concept and its workings can be seen in many aspects of the world. For example, a square, a cube, or a four-bladed propeller can all be turned 90 degrees without apparent change; they are said to have a

- *M.-C. Su is with the Department of Computer Science and Information Engineering, National Central University, Chung-Li, Taiwan 320, R.O.C. E-mail: muchun@csie.ncu.edu.tw.*
- *C.-H. Chou is with the Department of Electrical Engineering, Tamkang University, Tamsui, Taiwan 25137, R.O.C. E-mail: ister@ms19.hinet.net.*

*Manuscript received 9 Feb. 2000; revised 13 July 2000; accepted 23 Jan. 2001.
Recommended for acceptance by P. Meer.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 111428.*

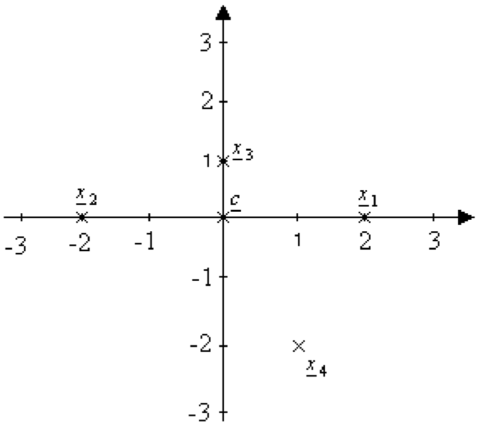
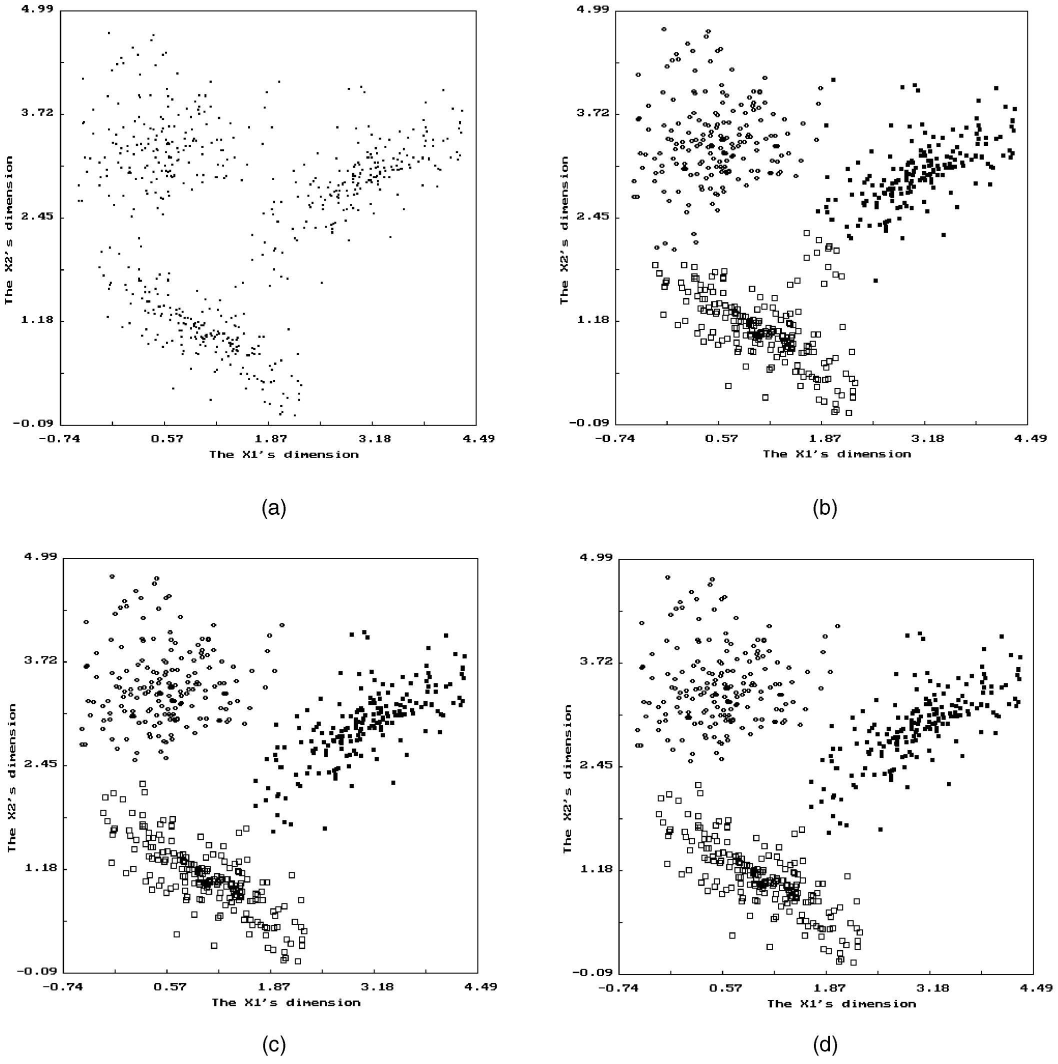


Fig. 1. An example of the point symmetry distance.

"four-fold axis of symmetry." A sphere has the highest possible symmetry; no twist or turn is detectable. The common starfish has five appropriate planes of symmetry and a five-fold rotation axis. Crystals of chemical substances show symmetry that derives from the lattice of molecules composing them. Symmetry is also an important parameter in physical and chemical processes and is an important criterion in medical diagnosis. They show how the laws of nature give symmetry to their products. However, the exact mathematical definition of symmetry [18], [19] is inadequate to describe and quantify symmetry found in the natural world or those found in the visual world.

Since symmetry is so common in the abstract and in nature, it is reasonable to assume some kinds of symmetry exit in the structures of clusters. Based on this idea, we will assign patterns to a cluster center if they present a symmetrical structure with respect to the cluster center. The immediate problem is how to find a metric to measure symmetry. A kind of symmetrical metric has been proposed by Reisfeld et al. and they used the symmetry transform as context-free attention operators [20]. For our opinions, their symmetrical metric is useful in image processing instead of in cluster analysis. Another kind of "*symmetry distance*" has been proposed by Zabrodsky et al. and their goal is to detect symmetry in a figure extracted from an image [21]. Their basic strategy is to choose the symmetry that is the "closest" to the figure measured by an appropriate metric, in which they adopted the minimum sum of the squared distances over which the vertices must be removed to impose the assumed symmetry; they call it the *symmetry distance*. It



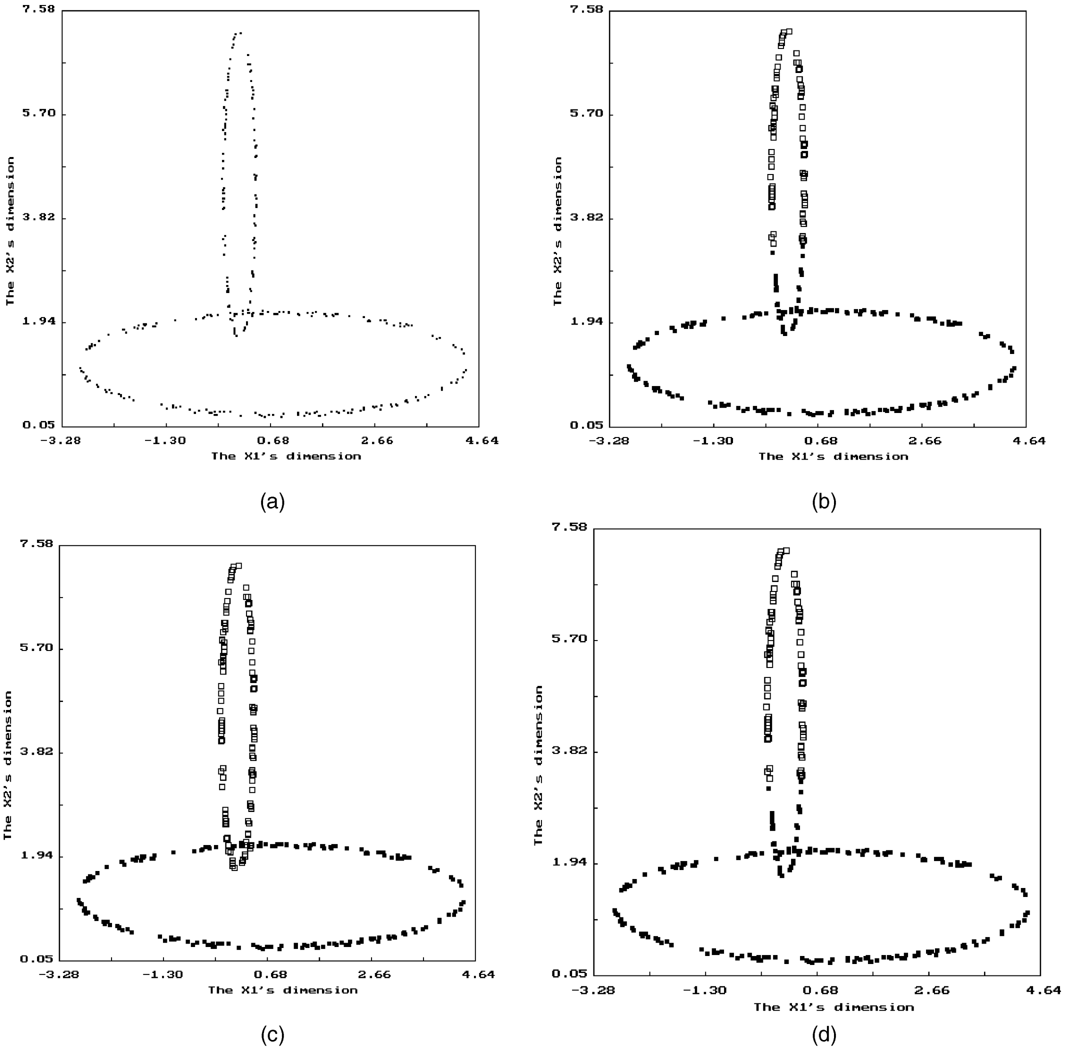
(a)

(b)

(c)

(d)

Fig. 2. (a) The data set contains of a mixture of compact spherical and ellipsoidal clusters. (b) The clustering result achieved by the K-means algorithm with the Euclidean distance. (c) The final clustering result achieved by the SBKM algorithm. (d) The clustering result achieved by the SBCL algorithm.



(a)

(b)

(c)

(d)

Fig. 3. (a) The data set contains two ellipsoidal shells. (b) The clustering result achieved by the K-means algorithm with the Euclidean distance. (c) The final clustering result achieved by the SBKM algorithm. (d) The clustering result achieved by the SBCL algorithm.

follows that we need an algorithm for efficiently imposing a given symmetry with a minimum displacement [22]. In the K-means algorithm, the cluster centroids represent the most important information. Therefore, "point symmetry" (symmetry about a point, in this case, the cluster center) is suitable to be applied in the K-means algorithm.

Based on above discussions, we propose a nonmetric distance based on the concept of point symmetry. The point symmetry distance is defined as follows: Given N patterns, $\underline{x}_i, i = 1, \cdots, N$, and a reference vector $\underline{c}$ (e.g., a cluster centroid), the "point symmetry distance" between a pattern $\underline{x}_j$ and the reference vector $\underline{c}$ is defined as

$$d_s(\underline{x}_j, \underline{c}) = \min_{\substack{i=1,\cdots,N \\ and\ i \neq j}} \frac{||(\underline{x}_j - \underline{c}) + (\underline{x}_i - \underline{c})||}{(||\underline{x}_j - \underline{c}|| + ||\underline{x}_i - \underline{c}||)}, \tag{2}$$

where the denominator term is used to normalize the point symmetry distance so as to make the point symmetry distance insensible to the Euclidean distances $||\underline{x}_j - \underline{c}||$ and $||\underline{x}_i - \underline{c}||$. If the right hand term of (2) is minimized when $\underline{x}_i = \underline{x}_{j*}$, then the pattern $\underline{x}_{j*}$ is denoted as the symmetrical pattern relative to $\underline{x}_j$ with respect to $\underline{c}$. Note that (2) is minimized when the pattern $\underline{x}_i = (2\underline{c} - \underline{x}_j)$ exists in the data set (i.e., $d_s(\underline{x}_j, \underline{c}) = 0$). The idea of the point symmetry is very simple and intuitive. It is instructed to observe the geometrical interpretation of the definition of the point symmetry distance. Fig. 1 gives the concept. For this case, we have four patterns $\underline{x}_1 = (2, 0)^T$, $\underline{x}_2 = (-2, 0)^T$, $\underline{x}_3 = (0, 1)^T$, $\underline{x}_4 = (1, -2)^T$, and one reference vector $\underline{c} = (0, 0)^T$. According to (2), we can easily compute

$$d_s(\underline{x}_1, \underline{c}) = \frac{||(\underline{x}_1 - \underline{c}) + (\underline{x}_2 - \underline{c})||}{||(\underline{x}_1 - \underline{c})|| + ||(\underline{x}_2 - \underline{c})||} = \frac{0}{2+2} = 0,$$

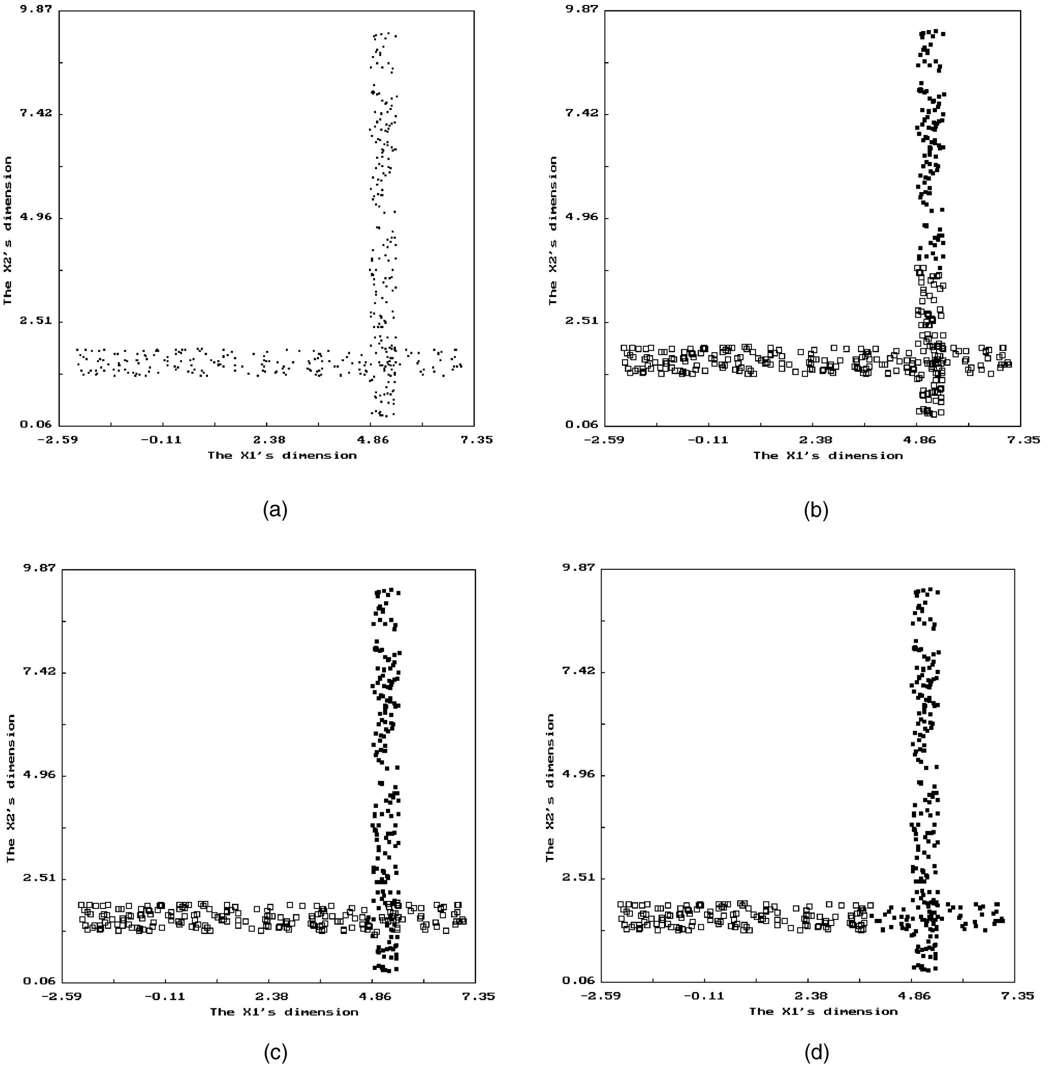
$$d_s(\underline{x}_2, \underline{c}) = \frac{||(\underline{x}_2 - \underline{c}) + (\underline{x}_1 - \underline{c})||}{||(\underline{x}_2 - \underline{c})|| + ||(\underline{x}_1 - \underline{c})||} = \frac{0}{2+2} = 0,$$

$$d_s(\underline{x}_3, \underline{c}) = \frac{||(\underline{x}_3 - \underline{c}) + (\underline{x}_4 - \underline{c})||}{||(\underline{x}_3 - \underline{c})|| + ||(\underline{x}_4 - \underline{c})||} = \frac{\sqrt{2}}{1+\sqrt{5}} = 0.437,$$

and

$$d_s(\underline{x}_4, \underline{c}) = \frac{||(\underline{x}_4 - \underline{c}) + (\underline{x}_3 - \underline{c})||}{||(\underline{x}_4 - \underline{c})|| + ||(\underline{x}_3 - \underline{c})||} = \frac{\sqrt{2}}{\sqrt{5}+1} = 0.437.$$

Understandably, the patterns $\underline{x}_1$ and $\underline{x}_2$ are the most symmetrical pair relative to the reference vector $\underline{c}$ in Fig. 1.



(a)

(b)

(c)

(d)

Fig. 4. (a) The data set contains a combination of two crossed lines. (b) The clustering result achieved by the K-means algorithm with the Euclidean distance (c) The final clustering result achieved by the SBKM algorithm. (d) The clustering result achieved by the SBCL algorithm.

3 The Proposed Clustering Algorithm

In previous work [23], we proposed a new competitive learning algorithm for training single-layer neural networks to cluster data. The detected cluster may be a set of clusters of different geometrical structures. Neurons compete with each other based on the point symmetry distance instead of the Euclidean distance. The symmetry-based competitive learning (SBCL) algorithm works well for cases where data sets do not contain crossed geometrical structures or they do not overlap too much. In this paper, we propose a symmetry-based version of the K-means (SBKM) algorithm. The SBKM algorithm is more effective than the SBCL algorithm. The SBKM algorithm is presented as follows:

Step 1: Initialization. We randomly choose K data points from the data set to initialize K cluster centroids, $\underline{c}_1$, $\underline{c}_2 \ldots, \underline{c}_K$.

Step 2: Coarse-Tuning. Now, use the ordinary K-means algorithm with the Euclidean distance to update the K cluster centroids. After the K cluster centroids converge or some kind of terminating criteria is satisfied, we then proceed to the fine-tuning procedure.

Step 3: Fine-Tuning. For pattern $\underline{x}$, find the cluster centroid nearest it in the symmetrical sense. That is, we find the cluster centroid k^* which is nearest to the input pattern $\underline{x}$ using the minimum-value criterion:

$$k^* = Arg \min_{k=1,\ldots,K} d_s(\underline{x}, \underline{c}_k), \tag{3}$$

where the point symmetry distance $d_s(\underline{x}, \underline{c}_k)$ is computed by (2). If the point symmetry distance $d_s(\underline{x}, \underline{c}_{k*})$ is smaller than a prespecified parameter θ, then assign the data point $\underline{x}$ to the $k^*\mathrm{th}$ cluster. Otherwise, the data point is assigned to the cluster centroid k^* using the following criterion:

$$k^* = Arg \min_{k=1,\ldots,K} d(\underline{x}, \underline{c}_k), \tag{4}$$

where $d(\underline{x}, \underline{c}_k)$ is the Euclidean distance between the input pattern and the cluster centroid $\underline{c}_k$.

Step 4: Updating. Compute the new centroids of the K clusters. The updating rule is given below:

$$\underline{c}_k(t+1) = \frac{1}{N_k} \sum_{i \in S_k(t)} \underline{x}_i, \tag{5}$$

where $S_k(t)$ is the set whose elements are the patterns assigned to the $k\mathrm{th}$ cluster at time t and N_k is the number of elements in S_k.

Step 5: Continuation. If no patterns change categories or the number of iterations has reached a prespecified maximum number, then stop. Otherwise, go to Step 3.

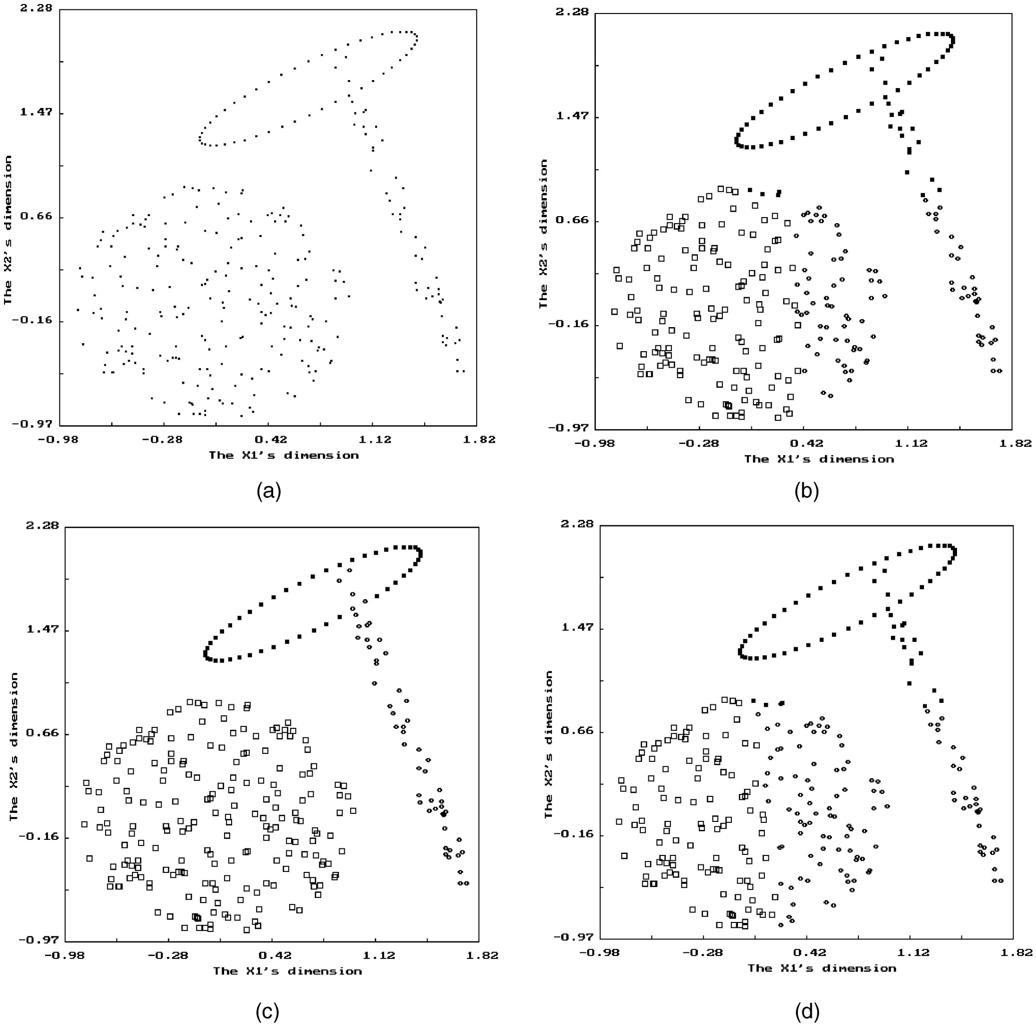


Fig. 5. (a) The data set contains a combination of ring-shaped, compact, and linear clusters. (b) The clustering result achieved by the K-means algorithm with the Euclidean distance. (c) The final clustering result achieved by the SBKM algorithm. (d) The clustering result achieved by the SBCL algorithm.

4 Experimental Results

We used four examples to compare the SBKM algorithm and the SBCL algorithm. In addition, we use one example to show how to use the point symmetry distance in face detections The parameter θ is chosen for 0.18 and this was kept the same irrespective of the data sets used. Since the K-means algorithm is very sensitive to the choice of cluster centers, the best results in ten trials for each data set are reported in this paper. However, we want to emphasize that the majority of the ten trials have the same clustering results.

Example 1. We generated a mixture of spherical and ellipsoidal clusters, as shown in Fig. 2a. There is no clear border between the clusters. The total number of data points is 577. According to the SBKM algorithm, we first clustered the data sets using the ordinary K-means algorithm with the Euclidean distance. Fig. 2b shows the clustering result. We notice that there are several misclassified data points. We then used the point symmetry distance as the dissimilarity measure and entered the fine-tuning procedure. The clustering result is given in Fig. 2c. Obviously, the clustering performance was greatly improved. Fig. 2d shows the clustering result of the SBCL algorithm. The two clustering results shown in Fig. 2c and Fig. 2d are identical.

Example 2. This data set contains 300 data points distributed on two crossed ellipsoidal shells, as shown in Fig. 3a. We use this example to illustrate that the proposed algorithm incorporated with point symmetry distance can also be applied to detect ring-shaped clusters even if they are crossed. The detection of ring-shaped clusters from a digital image is important in industrial applications. Fig. 3b shows the clustering result achieved by the ordinary K-means algorithm with the Euclidean distance. Fig. 3c illustrates the final result achieved by the SBKM algorithm. Fig. 3d shows the clustering result of the SBCL algorithm. We find that the SBCL algorithm cannot work well for this case.

Example 3. The data set consists of two crossed lines. On each line, 200 data points are distributed, as shown in Fig. 4a. The clustering result achieved by the ordinary K-means algorithm with the Euclidean distance is shown in Fig. 4b. The final clustering result of the SBKM algorithm is given in Fig. 4c. This example shows that the proposed algorithm works well for clusters with linear structures even if they are crossed. Fig. 4d shows the clustering result of the SBCL algorithm. We notice that the SBCL algorithm cannot work well for this example.

Example 4. This data set is a combination of ring-shaped, compact, and linear clusters, as shown in Fig. 5a. The ring-shaped cluster and the linear cluster are crossed to each other. The total number of data points is 300. Most clustering algorithms based on objective function minimization fail to detect this kind of

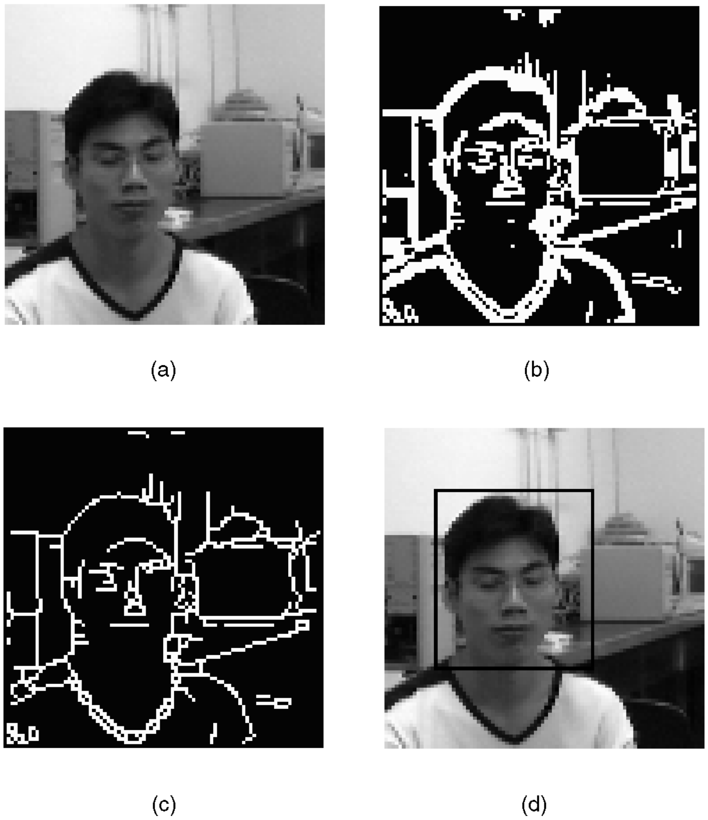


Fig. 6. (a) The original image. (b) The result after applying Sobel operator. (c) The thinned image. (d) The detected human face.

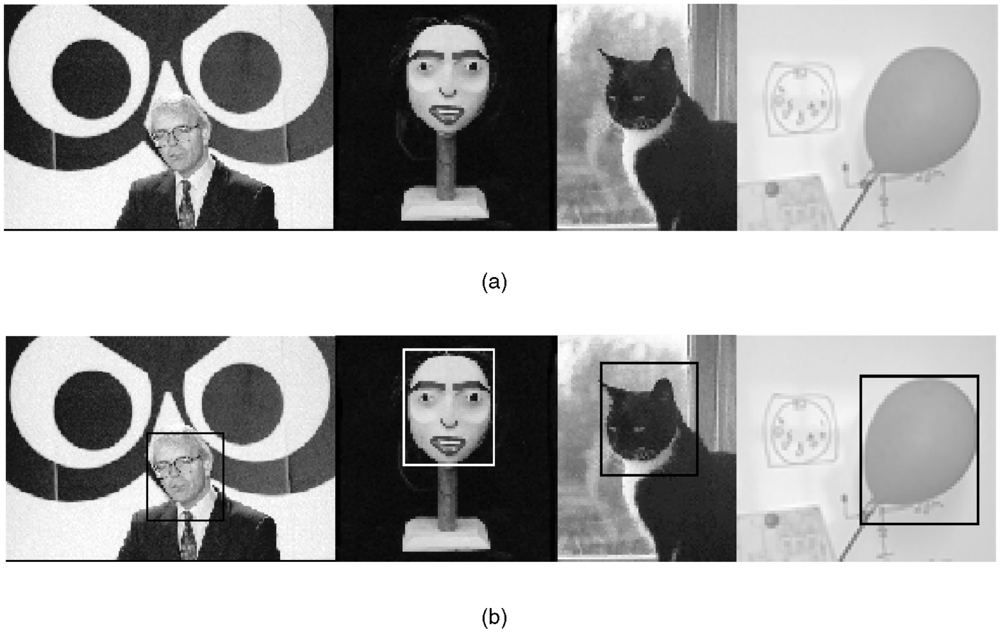


Fig. 7. Face detection: (a) The four original images and (b) the detected faces.

data sets because their performance depends on the dissimilarity measures used to generate a partition of the data sets. As we discussed in Section 2, those popular metrics can only characterize clusters which are compact. The clustering result achieved by the ordinary K-means algorithm with the Euclidean distance is shown in Fig. 5b. The final clustering result of the SBKM algorithm is illustrated in Fig. 5c. This example shows that the proposed algorithm works well for a set of clusters of different geometrical structures. Fig. 5d shows the clustering result of the SBCL algorithm. Obviously, the SBCL algorithm can't get a good experimental result in this example.

Example 5. In this example, we show how to apply the point symmetry distance in face detection. Detecting a face in a complex background is a difficult problem. In fact, it is not a trivial task since the sizes and orientations of the objects of interest may vary a lot. We assume the human face can be approximated by an ellipsoidal-shape. First, we use the Sobel operator to find the edge map of the original image and use a thinning algorithm [24] to thin the edge image. The idea of proposed approach is based on the symmetrical feature of the human face. We assume the object pixels in an edge image to be the data points and measure the degree of symmetry of these object pixels within a image region by using the point symmetry distance. Then, the image region with the largest degree of symmetry contains a face candidate. Using the proposed approach, the face region can be located and clipped out of the image. Fig. 6a gives an example of the original image, which included a human face in a complex background. Fig. 6b shows the edge image by using the Sobel operator. The thinned image is given in Fig. 6c. Fig. 6d illustrates that the proposed algorithm can be used to detect a human face in a complex background. We also used the same algorithm to test the image shown in Fig. 7a. For these four images, the detected faces are shown in Fig. 7b.

5 Conclusions

Although the SBCL algorithm and the SBKM algorithm both use the point symmetry distance as the dissimilarity measure, the SBKM algorithm outperformed the SBCL algorithm in many cases, e.g., in Examples 2, 3, and 4. The proposed SBKM algorithm can be used to group a given data set into a set of clusters of different geometrical structures. The price paid for the flexibility in detecting clusters is an increase of computational complexity. Besides, we can also apply the point symmetry distance to detect human faces. The experimental results are encouraging.

Acknowledgments

This work was partly supported by the National Science Council, Taiwan, R.O.C., under the grant NSC 88-2213-E-032-014.

References

[1] A.K. Jain and R.C. Dubes, *Algorithms for Clustering.* Englewood Cliffs, N.J.: Prentice Hall, 1988.

[2] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

[3] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum, 1981.

[4] J. Hartigan, *Clustering Algorithms.* New York: Wiley, 1975.

[5] J. Tou and R. Gonzalez, *Pattern Recognition Principles.* Reading, Mass.: Addison-Wesley, 1974.

[6] E. Ruspini, “A New Approach to Clustering,” *Information Control,* vol. 15, no. 1, pp. 22-32, July 1969.

[7] G.H. Ball and D.I. Hall, “Some Fundamental Concepts and Synthesis Procedures for Pattern Recognition Preprocessors,” *Proc. Int'l Conf. Microwaves, Circuit Theory, and Information Theory,* pp. 281-297, Sept. 1964.

[8] T. Kohonen, “The “Neural” Phonetic Typewriter,” *IEEE Computer,* vol. 27, no. 3, pp. 11-12, Mar. 1988.

[9] T. Kohonen, *Self-Organization and Associative Memory,* Third ed. New York, Berlin: Springer-Verlag, 1989.

[10] J. Mao and A.K. Jain, “A Self-Organizing Network for Hyperellipsoidal Clustering,” *IEEE Trans. Neural Networks,* vol. 7, pp. 16-29, Jan. 1996.

[11] G.A. Carpenter and S. Grossberg, “A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine,” *Computer Vision, Graphics, and Image Processing,* vol. 37, pp. 54-115, 1987.

[12] G.A. Carpenter and S. Grossberg, “ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns,” *Application Optics,* vol. 26, no. 23, pp. 4919-4930, Dec. 1987.

[13] C.T. Zahn, “Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters,” *IEEE Trans. Computer,* vol. 20, pp. 68-86, Jan. 1971.

[14] D. Blostein and N. Ahuja, “Shape From Texture: Integrating Texture-Element Extraction and Surface Estimation,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 12, pp. 1233-1251, Dec. 1989.

[15] M.C. Su, N. DeClaris, and T.K. Liu, “Application of Neural Networks in Cluster Analysis,” *Proc. IEEE Int'l Conf. System, Man, and Cybernetics,* pp. 1-6, 1997.

[16] M.C. Su and H.T. Chang, “Self-Organizing Neural Networks for Data Projection,” *Proc. Fifth Int'l Computer Science Conf.,* pp. 206-215, Dec. 1999.

[17] F. Attneave, “Symmetry Information and Memory for Pattern,” *Am. J. Psychology,* vol. 68, pp. 209-222, 1995.

[18] W. Miller, *Symmetry Groups and Their Applications.* London: Academic Press, 1972.

[19] H. Weyl, *Symmetry.* Princeton, NJ.: Princeton Univ. Press, 1952.

[20] D. Reisfeld, H. Wolfsow, and Y. Yeshurun, “Context-Free Attentional Operators: the Generalized Symmetry Transform,” *Int'l J. Computer Vision,* vol. 14, pp. 119-130, 1995.

[21] H. Zabrodsky, S. Peleg, and D. Avnir, “Symmetry as a Continuous Feature,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 12, pp. 1154-1166, Dec. 1995.

[22] K. Kanatani, “Comments on “Symmetry as a Continuous Feature”,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 3, pp. 246-247, Mar. 1997.

[23] M.C. Su and C.H. Chou, “A Competitive Learning Algorithm Using Symmetry,” *IEICE Trans. Fundamentals of Electronics, Communications, and Computer Sciences,* vol. E82-A, no. 4, pp. 680-687, Apr. 1999.

[24] R.C. Gonzalez and R.E. Woods, *Digital Image Processing.* Addison-Wesley, 1989.