A Genetic Clustering Technique Using a New Line Symmetry Based Distance Measure

Sriparna Saha, *Student Member, IEEE* and Sanghamitra Bandyopadhyay, *Senior Member, IEEE*
Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India-700108
Email: {sriparna_r, sanghami}@isical.ac.in

***Abstract*—In this paper, an evolutionary clustering technique is described that uses a new line symmetry based distance measure. Kd-tree based nearest neighbor search is used to reduce the complexity of finding the closest symmetric point. Adaptive mutation and crossover probabilities are used. The proposed GA with line symmetry distance based (GALSD) clustering technique is able to detect any type of clusters, irrespective of their geometrical shape and overlapping nature, as long as they possess the characteristic of line symmetry. GALSD is compared with existing well-known *K*-means algorithm. Five artificially generated and two real-life data sets are used to demonstrate its superiority.**

***Index Terms*—Unsupervised classification, clustering, symmetry property, line symmetry based distance, Kd-tree, Genetic algorithm**

I. Introduction

Partitioning a set of data points into some nonoverlapping clusters is an important topic in data analysis and pattern classification [1]. It has many applications, such as codebook design, data mining, image segmentation, data compression, etc. Many efficient clustering algorithms [2] [3] have been developed for data sets of different distributions in the past several decades. Most of the existing clustering algorithms adopt the 2-norm distance measure in the clustering process.

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 centroid. The measure of similarity is usually data dependent. One commonly used measure of similarity is the Euclidean distance D between two patterns $\overline{x}$ and $\overline{z}$ defined by $D = \|\overline{x} - \overline{z}\|$. Smaller Euclidean distance means better similarity and vice versa. This measure has been used in the *K*-means clustering algorithm [2]. By this algorithm hyperspherical clusters of almost equal size can be easily identified. This measure fails when clusters tend to develop along principal axes. It may be noted that one of the basic feature of shapes and objects is symmetry. As symmetry is so common in the natural world, it can be assumed that some kind of symmetry exists in the clusters also. Based on this, Su and Chou have proposed a new type of non-metric distance, based on point symmetry. This work is extended in [4] in order to overcome some of the limitations existing in [5]. It has been shown in [6] that the PS distance proposed in [4] has some serious drawbacks. Such as it is unable to detect the proper partitioning where some clusters are symmetrical with respect to some intermediate cluster center. The point symmetry based distance proposed by Su and Chou is only taken into consideration the amount of symmetry of a particular point, the Euclidean distance doesn't have any impact in it. In order to overcome these limitations, a new point symmetry based distance d_{ps} (PS-distance) is developed in [6]. This proposed distance is then used to develop a genetic algorithm based clustering technique, GAPS [6]. From the geometrical symmetry viewpoint, point symmetry and line symmetry are two widely discussed issues. The motivation of our present paper is as follows: to develop a new line symmetry based distance measure, and incorporate it in a genetic clustering scheme that preserves the advantages of the previous GAPS clustering algorithm.

K-means is a widely used clustering algorithm that has also been used in conjunction with the point-symmetry based distance measure in [5]. However *K*-means is known to get stuck at sub-optimal solutions depending on the choice of the initial cluster centers. In order to overcome this limitation, genetic algorithms have been used for solving the underlying optimization problem [7]. Genetic Algorithms (GAs) [8] are randomized search and optimization techniques guided by the principles of evolution and natural genetics, and having a large amount of implicit parallelism. GAs perform search in complex, large and multimodal landscapes, and provide near-optimal solutions for objective or fitness function of an optimization problem. In view of the advantages of the GA-based clustering method over the standard *K*-means [7], the former has been used in this article. In the proposed GA with line symmetry distance based clustering technique (GALSD), the assignment of points to different clusters are done based on the newly proposed line symmetry distance rather than the Euclidean distance. This enables the proposed algorithm to detect both convex and non-convex clusters of any shape and sizes as long as the clusters do have some line symmetry property. A Kd-tree based nearest neighbor search is utilized to reduce the computational complexity of computing the line symmetry distance. Adaptive mutation and crossover operations are used to accelerate the proposed GALSD to converge fast. The effectiveness of the proposed algorithm is demonstrated in identifying line symmetric clusters from five artificial and two real-life datasets. The clustering results are compared with those obtained by the well-known *K*-means algorithm.

II. The Point Symmetry Based Distance

In this section, a new PS distance [6], $d_{ps}(\overline{x}, \overline{c})$, associated with point $\overline{x}$ with respect to a center $\overline{c}$ is described. As shown in [6], $d_{ps}(\overline{x},\overline{c})$ is able to overcome some serious limitations of an earlier PS distance [5]. Let a point be $\overline{x}$. The symmetrical (reflected) point of $\overline{x}$ with respect to a particular centre $\overline{c}$ is $2\times\overline{c}-\overline{x}$. Let us denote this by $\overline{x}^*$. Let $knear$ unique nearest neighbors of $\overline{x}^*$ be at Euclidean distances of d_is, $i=1,2,\ldots knear$. Then

$$\begin{aligned} d_{ps}(\overline{x},\overline{c}) &= d_{sym}(\overline{x},\overline{c})\times d_e(\overline{x},\overline{c}), && (1)\\ &= \frac{\sum_{i=1}^{knear} d_i}{knear}\times d_e(\overline{x},\overline{c}), && (2)\end{aligned}$$

where $d_e(\overline{x},\overline{c})$ is the Euclidean distance between the point $\overline{x}$ and $\overline{c}$. It can be seen from Equation 2 that $knear$ cannot be chosen equal to 1, since if $\overline{x}^*$ exists in the data set then $d_{ps}(\overline{x},\overline{c})=0$ and hence there will be no impact of the Euclidean distance. On the contrary, large values of $knear$ may not be suitable because it may underestimate the amount of symmetry of a point with respect to a particular cluster center. Here $knear$ is chosen equal to 2.

Note that $d_{ps}(\overline{x},\overline{c})$, which is a non-metric, is a way of measuring the amount of symmetry between a point and a cluster center, rather than the distance like any Minkowski distance.

The benefits of using several neighbors instead of just one in Equation 2 are as follows.

1) Here since the average distance between $\overline{x}^*$ and its $knear$ unique nearest neighbors have been taken, this term will never be equal to 0, and the effect of $d_e(\overline{x},\overline{c})$, the Euclidean distance, will always be considered. Note that if only the nearest neighbor of $\overline{x}^*$ is considered and this happens to coincide with $\overline{x}^*$, then this term will be 0, making the distance insensitive to $d_e(\overline{x},\overline{c})$. This in turn would indicate that if a point is marginally more symmetrical to a far off cluster than to a very close one, it would be assigned to the farthest cluster. This often leads to undesirable results as demonstrated in [6].
2) Considering the $knear$ nearest neighbors in the computation of d_{ps} makes the PS-distance more robust and noise resistant. From an intuitive point of view, if this term is less, then the likelihood that $\overline{x}$ is symmetrical with respect to $\overline{c}$ increases. This is not the case when only the first nearest neighbor is considered which could mislead the method in noisy situations.

Note that the complexity of computing $d_{ps}(\overline{x},\overline{c})$ is $O(n)$, where n is the total number of data points. For all the n points and K clusters, the complexity becomes $O(n^2K)$. In order to reduce this, we have used Kd-tree based nearest neighbor search, ANN (Approximate Nearest Neighbor), which is a library written in C++ (obtained from http://www.cs.umd.edu/∼mount/ANN). Here ANN is used to find exact d_is, $i=1$ to $knear$ in Equation 2 efficiently. The Kd-tree structure can be constructed in $O(nlogn)$ time and takes $O(n)$ space [9].

III. The Newly Proposed Line Symmetry Based Distance

Given a particular partitioning, we first find the symmetrical line of each cluster by using the central moment technique [10]. Let the data set be denoted by $X=\{(x_1,y_1),(x_2,y_2),\ldots(x_n,y_n)\}$, then the (p,q)th order moment is defined as

$$m_{pq}=\sum_{\forall(x_i,y_i)\in X} x_i^p y_i^q \qquad (3)$$

By Equation 3, the centroid of the given data set for one cluster is defined as $(\frac{m_{10}}{m_{00}},\frac{m_{01}}{m_{00}})$. The central moment is defined as

$$u_{pq}=\sum_{\forall(x_i,y_i)\in X}(x_i-\overline{x})^p(y_i-\overline{y})^q, \qquad (4)$$

where $\overline{x}=\frac{m_{10}}{m_{00}}$ and $\overline{y}=\frac{m_{01}}{m_{00}}$. According to the calculated centroid and Equation 4, the major axis of each cluster can be determined by the following two items:

1) The major axis of the cluster must pass through the centroid.
2) The angle between the major axis and the x axis is equal to $0.5\times\tan^{-1}(\frac{2\times u_{11}}{u_{20}-u_{02}})$.

Consequently, for one cluster, its major axis is thus expressed by $((\frac{m_{10}}{m_{00}},\frac{m_{01}}{m_{00}}),0.5\times\tan^{-1}(\frac{2\times u_{11}}{u_{20}-u_{02}}))$.

The obtained major axis is treated as the symmetric line of the relevant cluster. This symmetrical line is used to measure the amount of line symmetry of a particular point in that cluster. In order to measure the amount of line symmetry of a point ($\overline{x}$) with respect to a particular line i, $d_{ls}(\overline{x},i)$, the following steps are followed.

1) For a particular data point $\overline{x}$, calculate the projected point $\overline{p}_i$ on the relevant symmetrical line i.
2) Find $d_{ps}(\overline{x},\overline{p}_i)$ by Equation 2. Then the amount of line symmetry of a particular point $\overline{x}$ with respect to that particular line i, will be equal to $d_{ps}(\overline{x},\overline{p}_i)$.

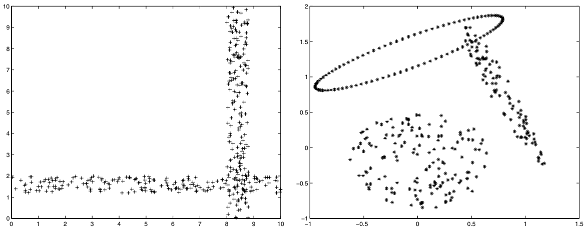
IV. GALSD: The Genetic Clustering Scheme with the Proposed Line Symmetry Based Distance

As mentioned earlier, the GA-based clustering algorithm [7] is used in this article since it is known to provide good clusters when *K* is known. However instead of the Euclidean distance, now the proposed line symmetry distance is used as the distance measure for computing the *clustering_metric* (*M*). The task of the GA is to search for the appropriate cluster centres $z_1,z_2\ldots z_K$ such that M is maximized.

A. String Representation and Population Initialization

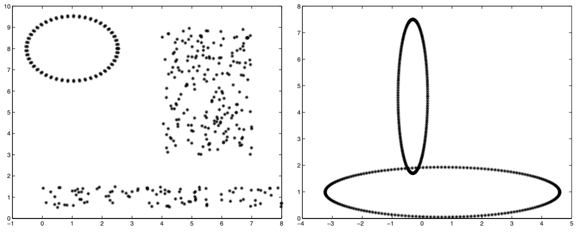
The basic steps of GAPS closely follow those of the conventional GA. Here center based encoding of the chromosome is used. Each string is a sequence of real numbers representing the *K* cluster centers and these are initialized to *K* randomly chosen points from the data set. This process is repeated for each of the $Popsize$ chromosomes in the population, where $Popsize$ is the size of the population.

Thereafter five iterations of the *K*-means algorithm is executed with the set of centers encoded in each chromosome. The resultant centers are used to replace the centers in the corresponding chromosomes. This makes the centers separated initially.



(a) (b)

Fig. 1. (a) *Data1* (b) *Data2*



(a) (b)

Fig. 2. (a) *Data3* (b) *Data4*

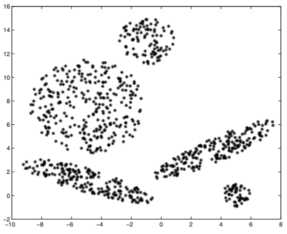


Fig. 3. *Data5*

B. Fitness Computation

In order to compute the fitness of the chromosomes, the following steps are executed.

1) Find the symmetrical line for each cluster. As described in the first paragraph of Section III, for each cluster, we use the moment-based approach to find out the relevant symmetrical line.
2) For each data point $\overline{x}_i,\ i = 1, \ldots N$ where N is the total number of points present in the data set, calculate the projected point $\overline{p}_{ki}$ on the relevant symmetrical line of cluster $C_k,\ k = 1, \ldots K$ where K is the total number of clusters. Then compute $d_{ls}(\overline{x}_i, k) = d_{ps}(\overline{x}_i, \overline{p}_{ki})$ by Equation 2.
3) The point $\overline{x}_i$ is assigned to cluster k iff $d_{ls}(\overline{x}_i, k) \leq d_{ls}(\overline{x}_i, j),\ j = 1, \ldots, K,\ j \neq k$ and $(d_{ls}(\overline{x}_i, k)/d_e(\overline{x}_i, \overline{p}_{ki})) \leq \theta$. For $(d_{ls}(\overline{x}_i, k)/d_e(\overline{x}_i, \overline{p}_{ki})) > \theta$, point $\overline{x}_i$ is assigned to some cluster m iff $d_e(\overline{x}_i, \overline{c}_m) \leq d_e(\overline{x}_i, \overline{c}_j),\ j = 1, 2 \ldots K,\ j \neq m$. In other words, point $\overline{x}_i$ is assigned to that cluster with respect to whose centers its PS-distance is the minimum, provided the total "symmetricity" with respect to it is less than some threshold θ. Otherwise assignment is done based on the minimum Euclidean distance criterion as normally used in [7] or the *K*-means algorithm.

We have provided a rough guideline of the choice of θ, the threshold value on the PS-distance. It is to be noted that if a point is indeed symmetric with respect to some cluster centre then the symmetrical distance computed in the above way will be small, and can be bounded as follows. Let d_{NN}^{max} be the maximum nearest neighbor distance in the data set. That is

$$d_{NN}^{max} = \max_{i=1,\ldots N} d_{NN}(\overline{x}_i), \tag{5}$$

where $d_{NN}(\overline{x}_i)$ is the nearest neighbor distance of $\overline{x}_i$. Assuming that $\overline{x}^*$ lies within the data space, it may be noted that

$$d_1 \leq \frac{d_{NN}^{max}}{2} \quad \text{and} \quad d_2 \leq \frac{3d_{NN}^{max}}{2}, \tag{6}$$

resulted in, $\frac{d_1+d_2}{2} \leq d_{NN}^{max}$. Ideally, a point $\overline{x}$ is exactly symmetrical with respect to some $\overline{c}$ if $d_1 = 0$. However considering the uncertainty of the location of a point as the sphere of radius d_{NN}^{max} around $\overline{x}$, we have kept the threshold θ equals to d_{NN}^{max}. Thus the computation of θ is automatic and does not require user intervention.

After the assignments are done, the cluster centres encoded in the chromosome are replaced by the mean points of the respective clusters. Subsequently for each chromosome *clustering_metric,M*, is calculated as defined below:

```
$M = 0$
For $k = 1$ to $K$ do
  For all data points $\overline{x}_i$, $i = 1$ to $n$ and $\overline{x}_i \in k$th cluster, do
```

$$M = M + d_{ls}(\overline{x}_i, k) \tag{7}$$

Then the fitness function of that chromosome, fit, is defined as the inverse of M, i.e.,

$$fit = \frac{1}{M} \tag{8}$$

This fitness function, fit, will be maximized by using genetic algorithm. (Note that there could be other ways of defining the fitness function).

C. Selection

Roulette wheel selection is used to implement the proportional selection strategy.

D. Crossover

Here, we have used the normal single point crossover [8]. Crossover probability is selected adaptively as in [11]. The expressions for crossover probabilities are computed as follows. Let f_{max} be the maximum fitness value of the current population, $\overline{f}$ be the average fitness value of the population and $f^{'}$ be the larger of the fitness values of the solutions to be crossed. Then the probability of crossover, μ_c, is calculated as:

$\mu_c = k_1 \times \frac{(f_{max} - f^{'})}{(f_{max} - \overline{f})}$, if $f^{'} > \overline{f}$,

$\mu_c = k_3$, if $f^{'} \leq \overline{f}$.

Here, as in [11], the values of k_1 and k_3 are kept equal to 1.0. Note that, when f_{max}=$\overline{f}$, then $f^{'} = f_{max}$ and μ_c will be equal to k_3. The aim behind this adaptation is to achieve a trade-off between exploration and exploitation in a different manner. The value of μ_c is increased when the better of the two chromosomes to be crossed is itself quite poor. In contrast when it is a good solution, μ_c is low so as to reduce the likelihood of disrupting a good solution by crossover.

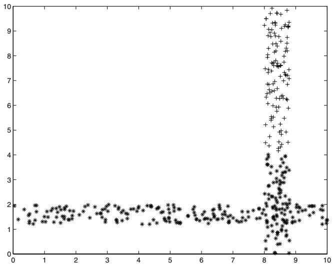


Fig. 4. Clustered *Data1* after application of *K*-means for $K = 2$

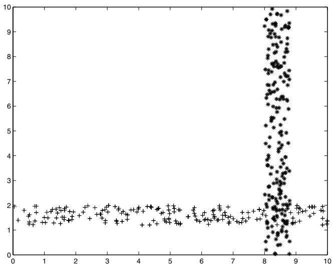


Fig. 5. Clustered *Data1* after application of GALSD-clustering algorithm for $K = 2$

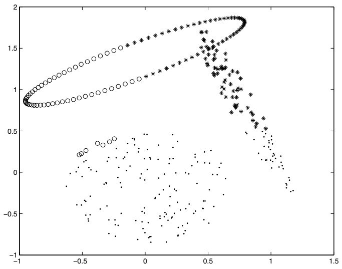


Fig. 6. Clustered *Data2* after application of *K*-means for $K = 3$

E. Mutation

Each chromosome undergoes mutation with a probability μ_m. The mutation probability is also selected adaptively for each chromosome as in [11]. The expression for mutation probability, μ_m, is given below:

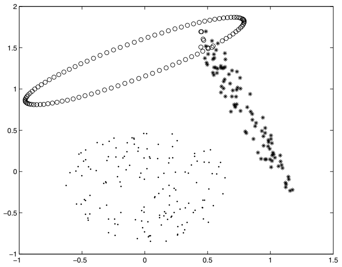


Fig. 7. Clustered *Data2* after application of GALSD-clustering algorithm for $K = 3$

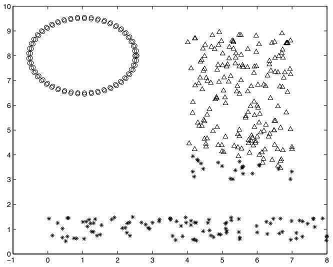


Fig. 8. Clustered *Data3* after application of *K*-means algorithm for $K = 3$

$\mu_m = k_2 \times \frac{(f_{max} - f)}{(f_{max} - \overline{f})}$ if $f > \overline{f}$,

$\mu_m = k_4$ if $f \leq \overline{f}$.

Here, values of k_2 and k_4 are kept equal to 0.5. This adaptive mutation helps GA to come out of local optimum. When GA converges to a local optimum, i.e., when $f_{max} - \overline{f}$ decreases, μ_c and μ_m both will be increased. As a result GA will come out of local optimum. It will also happen for the global optimum and may result in disruption of the near-optimal solutions. As a result GA will never converge to the global optimum. But as μ_c and μ_m will get lower values for high fitness solutions and get higher values for low fitness solutions, while the high fitness solutions aid in the convergence of the GA, the low fitness solutions prevent the GA from getting stuck at a local optimum. The use of elitism will also keep the best solution intact. For a solution with maximum fitness value, μ_c and μ_m are both zero. The best solution in a population is transferred undisrupted into the next generation. Together with the selection mechanism, this may lead to an exponential growth of the solution in the population and may cause premature convergence. To overcome the above stated problem, a default mutation rate (of 0.02) is kept for every solution in the GALSD.

We have used the mutation operation similar to that used in GA based clustering [7]. In GALSD, the processes of fitness computation, selection, crossover, and mutation are executed for a maximum number of generations. The best string seen upto the last generation provides the solution to the clustering problem. Elitism has been implemented at each generation by preserving the best string seen upto that generation in a location outside the population. Thus on termination, this location contains the centers of the final clusters.

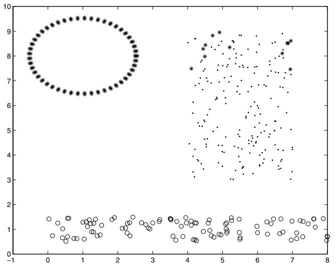


Fig. 9. Clustered *Data3* after application of GALSD-clustering algorithm for $K = 3$

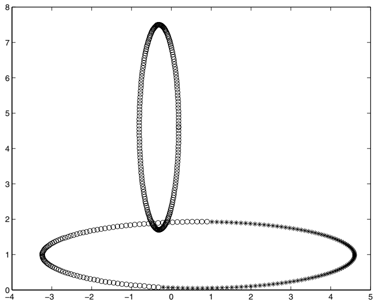


Fig. 10. Clustered *Data4* after application of *K*-means algorithm for $K = 2$

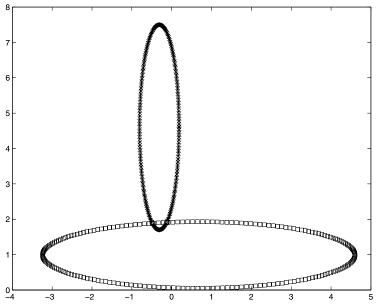
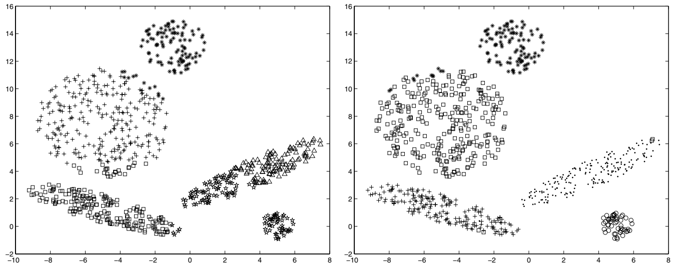


Fig. 11. Clustered *Data4* after application of GALSD-clustering algorithm for $K = 2$



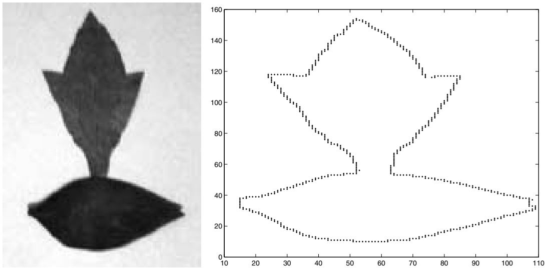
(a) (b)

Fig. 12. Clustered *Data5* for $K = 5$ after application of (a) *K*-means algorithm (b) GALSD-clustering algorithm

V. Implementation Results

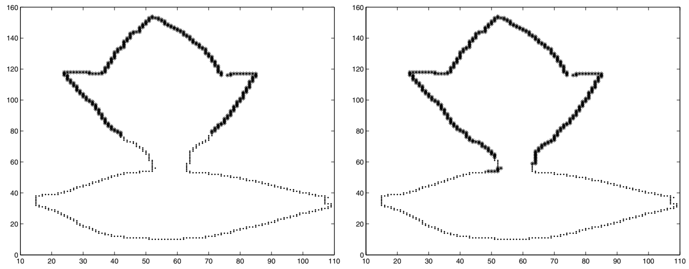
The experimental results comparing the performances of GALSD and *K*-means algorithm are provided for five artificial data sets and two real-life data sets. For the newly developed GALSD-clustering, value of θ is determined from the data set as discussed in Section IV-B. For GALSD, the crossover probability, μ_c, and mutation probability, μ_m, are determined adaptively as described in Section IV-D and Section IV-E, respectively. The population size, P is set equal to 100. The total number of generations is kept equal to 20. Executing it further did not improve the performance.

1) *Data1*: This data set consists of two bands as shown in Figure 1(a), where each band consists of 200 data points. The final clustering results obtained by *K*-means and GALSD are given in Figures 4 and 5, respectively. As expected *K*-means shows poor performance for this data since the clusters are not hyper-spherical in nature. Our proposed GALSD is able to detect the proper partitioning from this data set as the clusters possess the line symmetry property.
2) *Data2*: This data set is a combination of ring-shaped, compact and linear clusters shown in Figure 1(b). The total number of points in it is 350. The final results obtained after application of *K*-means, and GALSD are shown in Figures 6, and 7, respectively, where *K*-means is found to fail in providing the proper clusters. The proposed GALSD is able to detect the proper partitioning.
3) *Data3*: This data set is a combination of a ring-shaped cluster, a rectangular cluster and a linear cluster as shown in Figure 2(a), with total number of points equal to 400. The final results corresponding to *K*-means algorithm, and GALSD are shown in Figures 8 and 9, respectively. As evident from Figure 8, *K*-means fails in correctly detecting the linear cluster; it includes points from the rectangular cluster in the linear cluster. As expected, the ring is properly detected. In the partitioning provided by the GALSD some points of the rectangular cluster are included in the ring because those become more symmetric with respect to the major axis of the ring which passes through the rectangle.
4) *Data4*: This data set contains 400 points distributed on two crossed ellipsoidal shells shown in Figure 2(b). The final results corresponding to *K*-means algorithm, and GALSD are shown in Figures 10 and 11, respectively. As expected *K*-means is not able to detect the proper partitioning but GALSD is able to do so.
5) *Data5*: This data set contains 850 data points distributed on five clusters, as shown in Figure 3(a). The final results corresponding to *K*-means algorithm, and GALSD are shown in Figures 12(a) and 12(b), respectively. *K*-means again fails here in detecting ellipsoidal shaped clusters. But as the clusters present here are line symmetric, the proposed GASLD is able to detect the clusters well.
6) *Two_leaves1*: Most of the natural scenes, such as leaves of plants, have the line symmetry property. Figure 13(a) shows the two real leaves of *Ficus microcapa* and they overlap a little each other. First the sobel edge detector [10] is used to obtain the edge pixels in the input data points which is shown in Figure 13(b). After running the *K*-means algorithm, the obtained



(a) (b)

Fig. 13. (a) *Two_leaves1* data (b) Edge pixels of leaves as input data points



(a) (b)

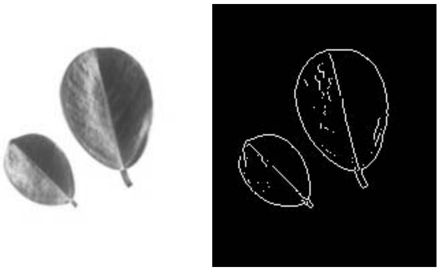
Fig. 14. Clustered *Two_leaves1* data for $K = 2$ after application of (a) *K*-means algorithm (b) proposed GALSD clustering algorithm

partitioning is shown in Figure 14(a). The clustering result obtained after execution of the proposed GALSD algorithm is shown in Figure 13(b). The proposed GALSD demonstrates a satisfactory clustering result.

7) *Two_leaves2*: Figure 15(a) shows the two real leaves of *Fizuslvgi*. The edge map obtained after application of the Sobel edge detector is shown in Figure 15(b). Figures 16(a) and 16(b) show the final clustering result obtained after application of *K*-means and GALSD clustering algorithms, respectively. Both the algorithms perform equally well. *K*-means is able to detect the proper partitioning as the two clusters are completely separated.

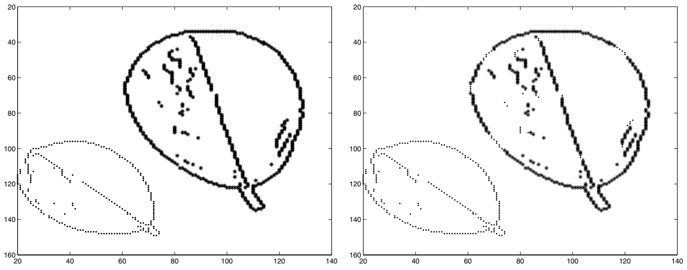
VI. Discussion and Conclusion

In this paper a new line symmetry based distance is proposed which is based on the existing point symmetry based distance. Kd-tree based nearest neighbor search is used to reduce the complexity of symmetry based distance computation. A genetic clustering technique (GALSD) is also proposed here that incorporates the new line symmetry distance while performing cluster assignments of the points and in the fitness computation. The major advantages of GALSD are as follows. In contrast to *K*-means, use of GA enables the algorithm to come out of local optima, making it less sensitive to the choice of the initial cluster centers. The proposed clustering technique can cluster data sets with the property of line symmetry successfully. The effectiveness of the proposed algorithm is demonstrated in detecting clusters having line symmetry property from five artificial and two real-life data sets. Other than the clustering experiments



(a) (b)

Fig. 15. (a) *Two_leaves2* data (b) Edge pixels of leaves as input data points



(a) (b)

Fig. 16. Clustered *Two_leaves2* data for $K = 2$ after application of (a) *K*-means algorithm (b) proposed GALSD clustering algorithm

using leaf example, it is an interesting future research topic to extend the results of this paper to face recognition. Current work is going on to improve the proposed GALSD clustering technique so that it can work perfectly for data sets like *Data3*.

References

[1] B. S. Everitt, S. Landau, and M. Leese, *Cluster Analysis*. London: Arnold, 2001.

[2] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[3] A. K. Jain, M. Murthy, and P. Flynn, "Data Clustering: A Review," *ACM Computing Reviews*, Nov,1999.

[4] C. H. Chou, M. C. Su, and E. Lai, "Symmetry as a new measure for cluster validity," in *2nd WSEAS Int. Conf. on Scientific Computation and Soft Computing*, Crete, Greece, 2002, pp. 209–213.

[5] M.-C. Su and C.-H. Chou, "A modified version of the k-means algorithm with a distance based on cluster symmetry," *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 674–680, 2001.

[6] S. Bandyopadhyay and S. Saha, "GAPS: A clustering method using a new point symmetry based distance measure," *Pattern Recog.*, vol. 40, pp. 3430–3451, 2007.

[7] U. Maulik and S. Bandyopadhyay, "Genetic algorithm based clustering technique," *Pattern Recog.*, vol. 33, pp. 1455–1465, 2000.

[8] J. H. Holland, *Adaptation in Natural and Artificial Systems*. AnnArbor: The University of Michigan Press, 1975.

[9] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, 2nd ed. Springer-Verlag, 2000. [Online]. Available: http://www.cs.uu.nl/geobook/

[10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Massachusetts: Addison-Wesley, 1992.

[11] M. Srinivas and L. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656–667, April, 1994.