Automatic Shape Independent Clustering Inspired by Ant Dynamics

Prithwish Chakraborty [1], Gourab Ghosh Roy[1], Spandan Sinha[1], Soumya Bose[1], Ankur Mondal[2], and Swagatam Das[1]

[1] *Dept. of Electronics and Telecommunication Engg, Jadavpur University, Kolkata, India*
[2] *Dept. of Computer Science, Guru Nanak Institute of Technology*
prithwish1611@gmail.com , myself_gourab@yahoo.co.in, spandan0@gmail.com, 1988.soumya@gmail.com, mondal.ankur@gmail.com, swagatamdas19@yahoo.co.in

Abstract

This article describes a simple heuristic algorithm that can automatically detect any number of well-separated clusters, which may be of any shape, convex and/or non-convex. This is in contrast to most of the existing clustering algorithms that assume a value for the number of clusters and/or a particular cluster structure. The algorithm draws inspiration from the dynamics of ants and iteratively partitions the dataset based on its proximity matrix. A runtime complexity analysis shows that the algorithm runs in quadratic time with respect to the size of the dataset. It can detect outliers from the data and is also able to identify the situation when the data do not have any natural clusters at all. Encouraging results on both real and artificial datasets have been included to show the effectiveness of the proposed technique.

1. Introduction

The objective of clustering is to partition unlabeled data [1] into groups of identical objects. Each group, called a 'cluster', consists of objects that are similar between themselves and dissimilar to objects belonging to other groups. Clustering, or cluster analysis, is prevalent in any discipline that involves analysis of multivariate data. In the past few decades, cluster analysis has played a central role in diverse domains of science and engineering [2-7].

Clustering algorithms can be *hierarchical* or *partitional* [1]. In hierarchical clustering, the output is a tree showing a sequence of clustering with each cluster being a partition of the data set [2]. Partitional clustering algorithms, on the other hand, attempts to decompose the data set directly into a set of disjoint clusters. They try to optimize certain criteria (e.g. a Squared-error function).

The problem of partitional clustering has been approached from the diverse fields of knowledge like statistics (multivariate analysis) [8], graph theory [9], expectation maximization algorithms [10], artificial neural networks [11, 12], evolutionary computing [13, 14], swarm intelligence [15] and so on.

The first task prior to clustering data involves identifying whether or not there is any cluster structure. For example, a data set where all the values are drawn from a uniform distribution or from any uni-modal distribution will not exhibit natural groupings. The task of assessing whether a data set has inherent structure in it is called estimating its clustering tendency. Once it is determined that clustering needs to be performed, identifying a good clustering method becomes a challenge. Most of the clustering algorithms often require the number of clusters to be specified a priori. For example, the K-means [16], fuzzy c-means (FCM) [17] and the single linkage method [1] require the specification of the number of clusters beforehand, which may not be feasible to do in many real-life situations. Several other methods also assume the geometry of the data; typically the clusters are assumed to be convex in

nature. For example, the algorithms based on minimization of squared error criterion (like k-means or FCM or GCUK [18]) assume hyper-spherical clusters of approximately equal sizes.

In this article we propose a very simple heuristic algorithm for automatic detection of the optimal number of natural clusters from a previously unhandled dataset. The algorithm is not biased towards the hyper-spherical shape of clusters and can detect shell-type or solid clusters of any arbitrary shape efficiently.

The proposed algorithm draws inspiration from the foraging behavior of ants, although it does not employ the concepts of the Ant Colony Systems (ACS) [19] or Ant Colony Optimization (ACO) [20] directly. Also as will be evident from what follows, the algorithm has no similarity with other ant-based clustering approaches like [21] and [22].

The rest of the paper is organized in the following way. Section 2 describes the ant-based clustering algorithm in sufficient details. Section 3 analyses the average rum-time complexity of the algorithm. Section 4 presents the simulation results of comparing the algorithm with a few most popular partitional and hierarchical clustering techniques on a variety of artificial datasets. Finally Section 5 concludes the paper with a discussion of possible future research issues.

2. The Ant-inspired clustering algorithm

Clustering techniques may be divided into a no of broad classifications such as (i) sequential, (ii) hierarchical, (iii) cost function optimization, etc. In this paper a sequential technique, modified by drawing inspiration from ant movement, has been used. As ants move along a path they deposit a chemical substance known as pheromone. This pheromone decays over time. Ants are generally attracted by pheromone. Only this aspect of ant dynamics is used in this process. Initially, a pseudo-ant is initialized at a random point of the data set. This pseudo-ant searches for the nearest data point from its present location and memorizes the index of that point. In the next iteration the pseudo-ant deposits *anti-pheromone* on its present location and moves to the next point as memorized from its index. Like pheromone anti-pheromone also decays with time, but unlike pheromone pseudo-ants are repelled from a data point due to presence of anti-pheromone. This may be alternately put forward as pseudo-ants deposit anti-pheromone in its path and those data points are covered by anti-pheromone cloud. This blurs the point from the vision of the ant. Over time, this cloud becomes thinner due to anti-pheromone decay and those data points gradually come into vision of the pseudo-ant. As the ant moves in the afore-mentioned manner it checks whether it has already visited the nearest visible point and terminates indicating a cluster under such a condition. This technique ensures that, without employing any velocity operator or gradient operator the pseudo-ant traverses the data set and in case of closed clusters it moves along the cluster without getting stuck and when finally it has scanned over all the cluster points it comes back to an already visited point over which anti-pheromone cloud has sufficiently thinned so as to make it the nearest visible data point. According to the scheme outlined before the pseudo-ant then terminates it movement indicating a cluster. The algorithm saves this cluster, removes the cluster points from the data set and initializes a new pseudo-ant to search for remaining clusters. This process continues until all the data point has been classified into clusters. Thus the algorithm automatically detects the clusters without requiring any knowledge about no of clusters in the data point. The parameter which ensures proper cluster identification is the pheromone amount deposited on a data point and its rate of decay. These parameters may be estimated from maximum distance between data points and the data set variance, respectively. Also, a data set may have clusters of different densities. This necessitates different amount of anti-pheromone deposit for different clusters which may prove to be quite cumbersome. As such we deposit same amount anti-pheromone on every point, but the pseudo-ant is now modeled to compare the distance form the nearest visible point and compare it with distances between successive points visited by the pseudo-ant, terminating itself when the difference is large.

The algorithm is presented below. Besides every important step the bold number in parenthesis denotes the no of basic operations required to complete the step.

Let; $\bar{x} = \{x_1, x_2, \ldots\ldots x_n\}$ represent data set consisting of n data points where every data point $x_i = \{x_{i,1}, x_{i,2}, \ldots\ldots\ldots x_{i,d}\}, \forall i = 1(1)n$ represents a point in the d dimensional feature space. Also, let us define a 2 -dimensional matrix $dist$ where,

$$dist(p,q) = \sqrt{\sum_{j=1}^{d} (x_{p,d} - x_{q,d})^2}$$

represent the distance of the p-th data point $\vec{x}_p$ from the q-th data point $\vec{x}_q$. Lastly let $cluster$ be a 2 dimensional matrix such that,

$$cluster(i, j) = index\ of\ \ jth\ \ point\ of\ \ ith\ cluster.$$

Then the process may be represented as:

for i-th pseudo-ant

$b = ceil(rand * n_i);$

// where, n_i is the no. of data points left un-clustered to the i-th pseudo-ant.

$cluster(i,1) = b;$ // initializing pseudo-ant.

$act\_dist = dist;$

$flag = 0;$

$temp = 0;$

$t = 0;$

$j = 1;$

$while\ \ j < n_i$

$cluster(i, j) = b;$

$dist(k,b) = dist(k,b) + penalty, k = 1(1)n_i$ **[n_i]**

//anti-pheromone deposit

$b1 = \min(dist(b,k)), k = 1(1)n_i$ **[2*n_i]**

//searching next cluster point

$if\ \ b1 == cluster(i,l), l = 1(1)\ j$

$flag = 1;$ **[j]**

end

//checking for round trip

$temp = temp + act\_dist(b,b1);$

$if\ \ dist(b,k) > jump * temp / j$

$flag = 1;$

end

//checking for jump break

$if\ \ flag == 1$

$remove\ cluster\ \ point\ s\ \ from\ data\ set$

$n_{i+1} = n_i - j;$

$i = i + 1;$

$break$

end

//terminating present cluster

$t = t + factor;$

$dist(k, cluster(i,l)) = dist(k, cluster(i,l)) * (1 - t / n_i);$

$l = 1(1)\ j$

$k = 1(1)n_i$ **[j*n_i]**

//anti-pheromone decay

$b = b1;$

//saving next cluster point

$j = j + 1;$

end

//terminating i-th cluster

$if\ \ n_i < 0$

$end\ \ process$

$else$

$repeat\ \ process\ \ for\ next\ \ pseudo\_ant$

end

Here parameters are $penalty$, $factor\ and\ \ jump$.

One may estimate them as

$E(penalty) = \max(dist)$

$E(jump) = \text{var}(X)$

3. Complexity analysis

In the previous section, no. of basic operations required for those important and potentially time consuming steps have been shown, for a single iteration of a cluster.

If the i-th cluster consists of c_i no of data points and if there are p no of clusters then, the approximate complexity may be given as:

$$\begin{aligned} complexity &\approx \sum_i \sum_j (3 * j + 1) * n_i \\ &= \sum_i (c_i (c_i + 1)/2) + 3) * n_i \\ &\approx \sum_i (c_i{}^2) * n_i / 3 \quad \text{,for large n} \end{aligned}$$

Best case complexity is found when every data point belongs to different clusters or if they belong to the same cluster.

As such an average complexity can be approximated assuming that there are p clusters consisting of c members each.

Then, $n_i = n - c * (i - 1), i = 1(1)p$

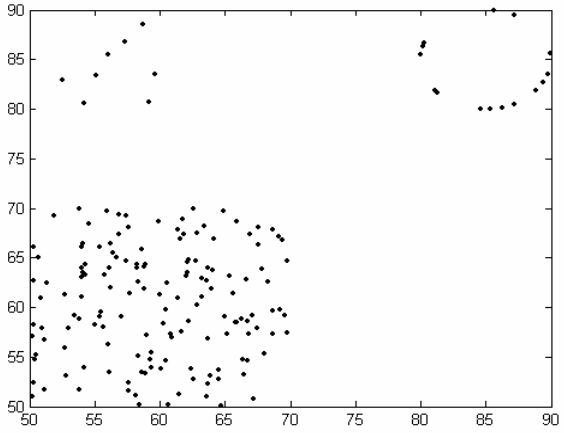
also, $n = c * p$

$$\begin{aligned} \therefore complexity &= c^2 (np - c * (p^2 - p)/2)/2 \\ &\approx c^2 np \\ &= n^2 c \end{aligned}$$

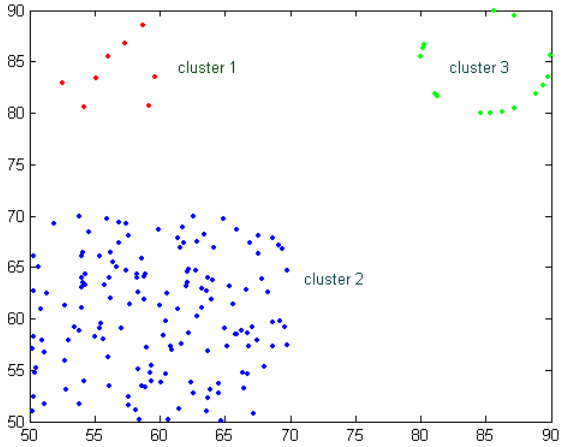
So, $complexity = O(n^2 c)$

4. Experimental results

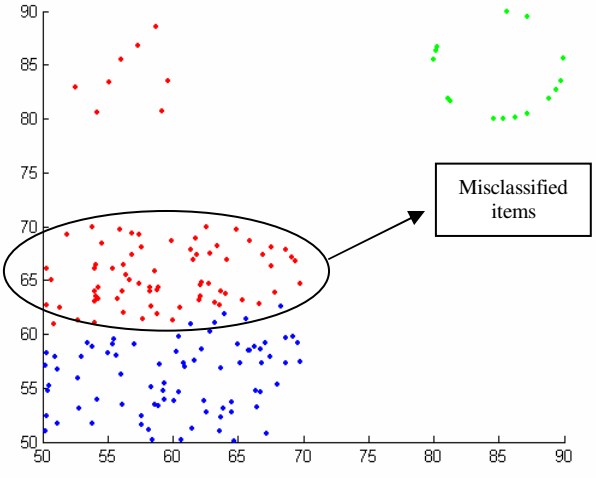
In this section we provide limited results of the simulation experiments that are being undertaken in order to compare the performance of the proposed algorithm with the most widely used partitional and hierarchical clustering algorithms. The algorithms compared include the k-means, FCM, agglomerative single linkage, agglomerative average linkage, and agglomerative complete linkage. Note that except the ant-based method all the other algorithms are non-automatic and for them the correct number of clusters was supplied in each case as an input. Here we show the clustering results over four two-dimensional datasets with clusters of various shape, size and density. Two-dimensional datasets were chosen for the ease of visualizations of the results. More experiments regarding the efficiency of the algorithm on higher-dimensional and real world datasets and comparison in terms of several clustering validity indices will be reported in a future communication. Figures 1 to 5 shows the unlabelled synthetic datasets and the corresponding clustering results with all the algorithms. In each case the misclassified portions of the datasets have been marked.



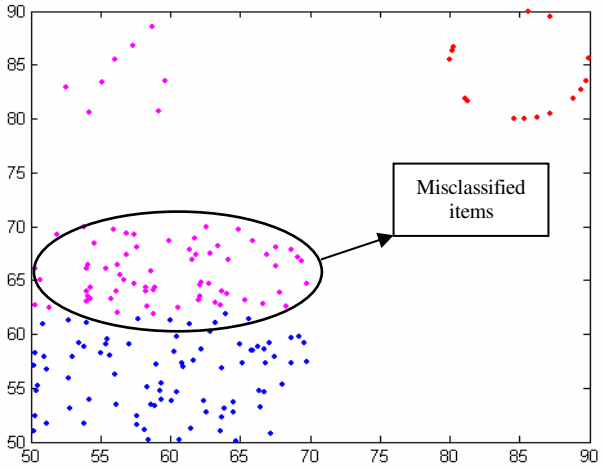
(a) Unlabelled Synthetic_Data_1



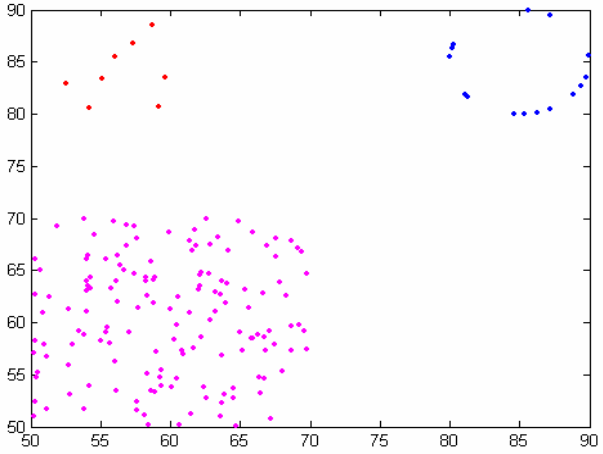
(b) Clustering with the ant-based clustering technique



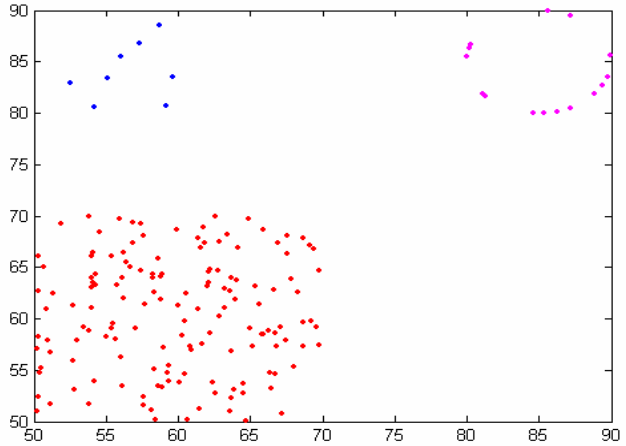
(c) Clustering with the k-means algorithm



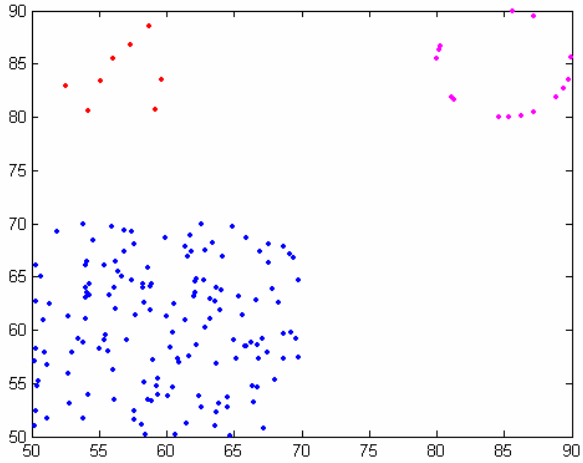
(d) Clustering with the FCM algorithm



(e) Clustering with the complete-link hierarchical agglomerative algorithm.

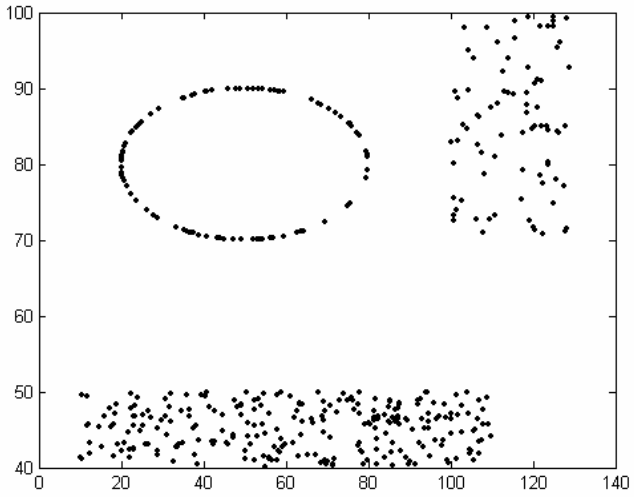


(f) Clustering with the single-link hierarchical agglomerative algorithm.

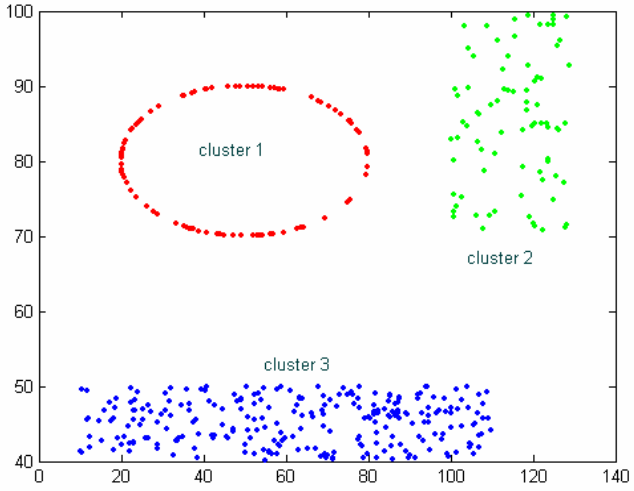


(g) Clustering with the average-link hierarchical agglomerative algorithm.

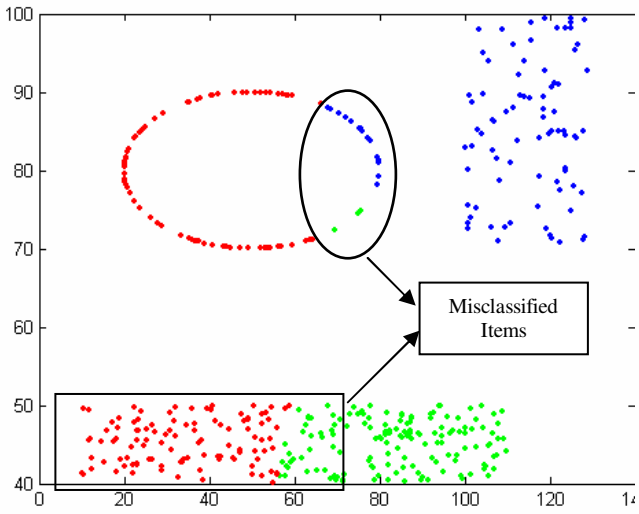
Fig. 1: Clustering results over Synthetic Data_1



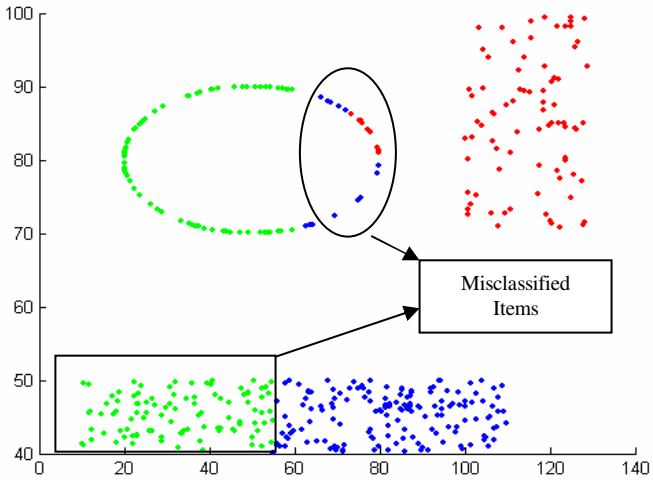
(a) Unlabelled Synthetic_Data_2



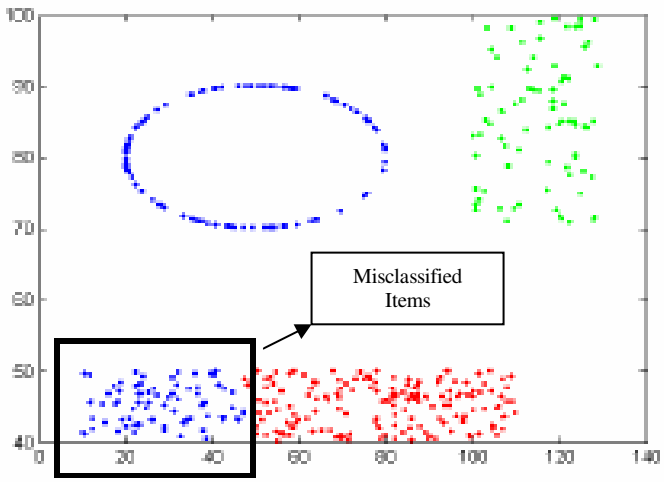
(b) Clustering with the ant-based clustering technique



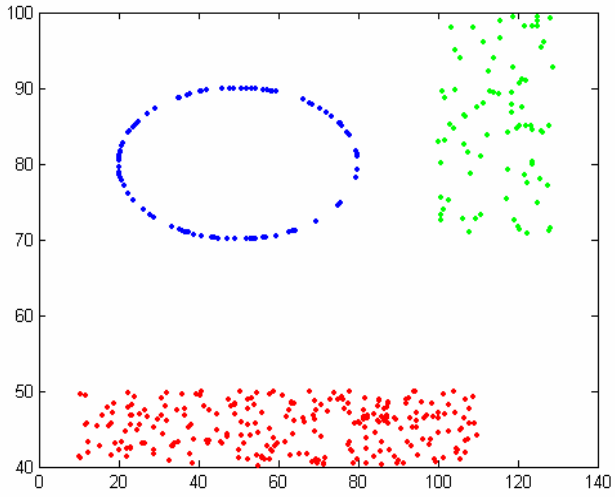
(c) Clustering with the k-means algorithm



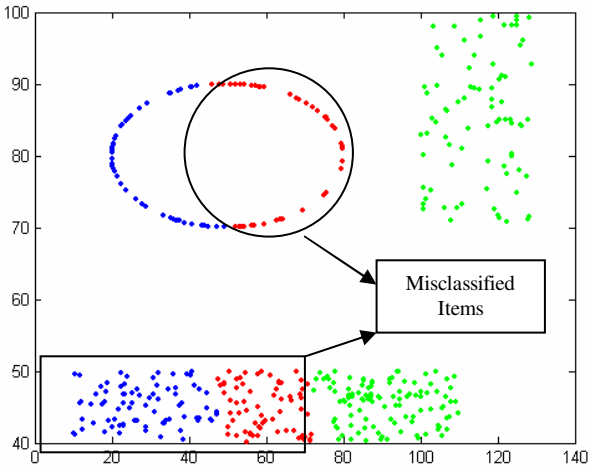
(d) Clustering with the FCM algorithm



(e) Clustering with hierarchical average-link agglomerative algorithm

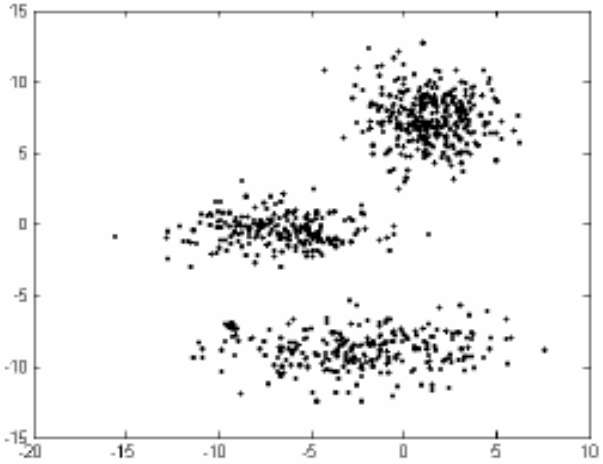


(f) Clustering with hierarchical single-link agglomerative algorithm

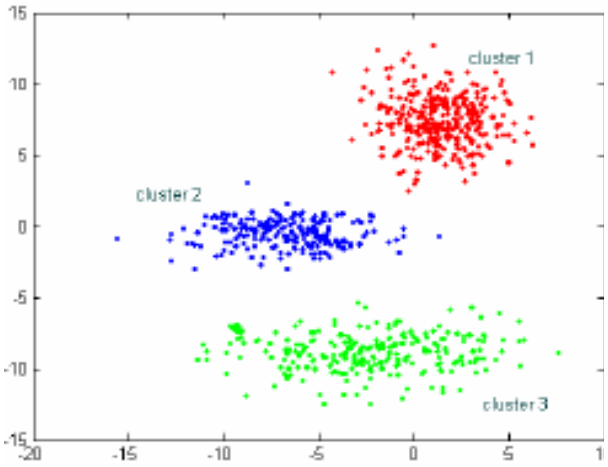


(g) Clustering with hierarchical complete-link agglomerative algorithm

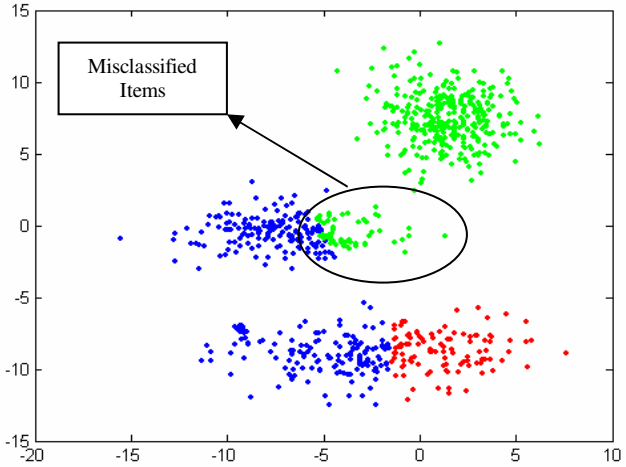
Fig. 2: Clustering results over Synthetic Data_2



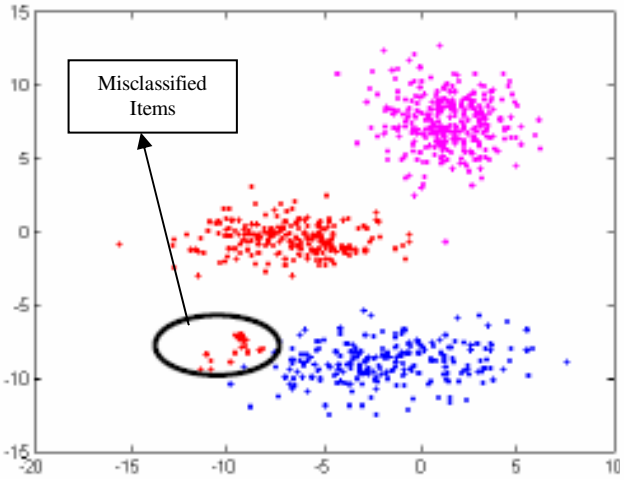
(a) Unlabelled Synthetic_Data_3



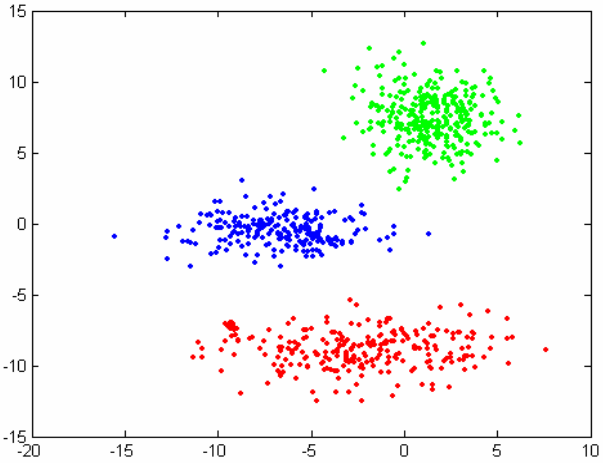
(b) Clustering with the ant-based clustering technique



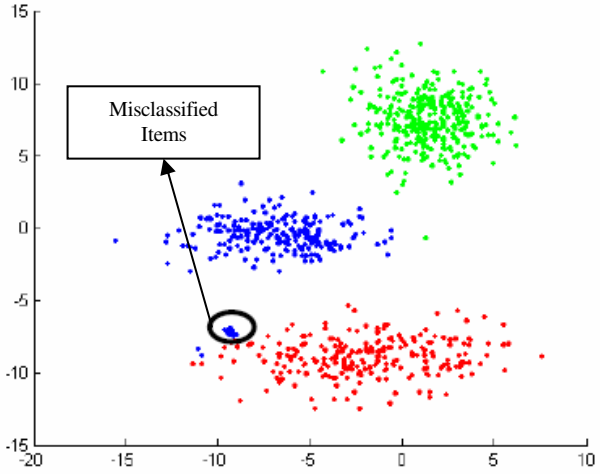
(e)Clustering with hierarchical complete-link agglomerative algorithm



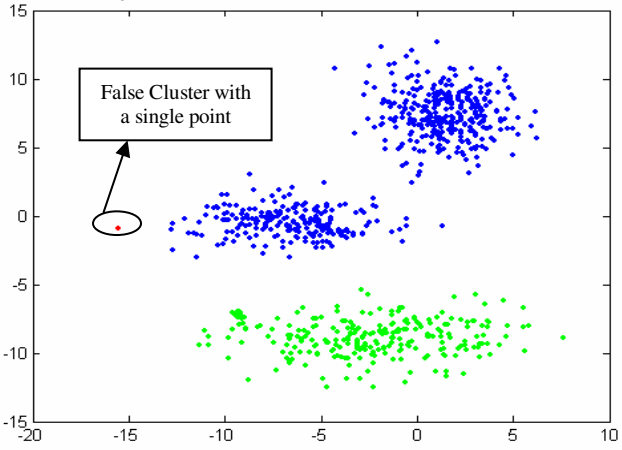
(c) Clustering with the k-means algorithm.



(f) Clustering with hierarchical average-link agglomerative algorithm

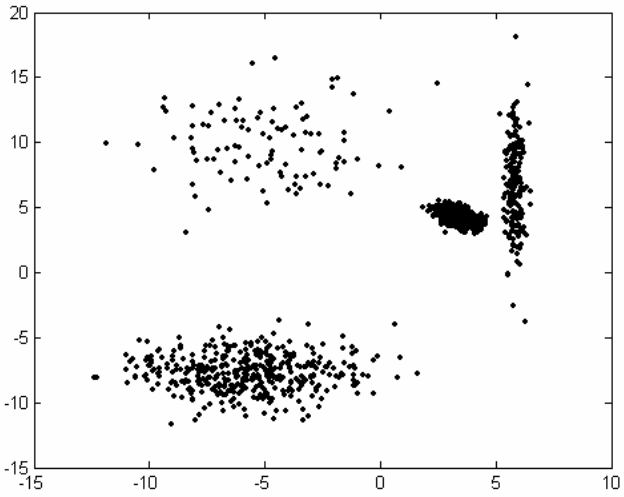


(d) Clustering with the FCM Algorithm

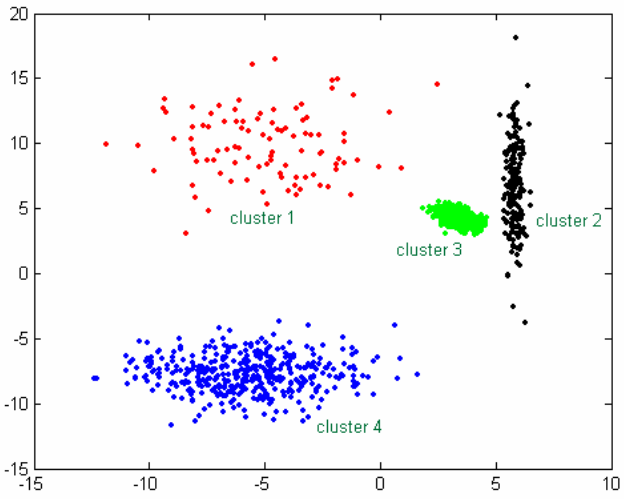


(f) Clustering with hierarchical single-link agglomerative algorithm

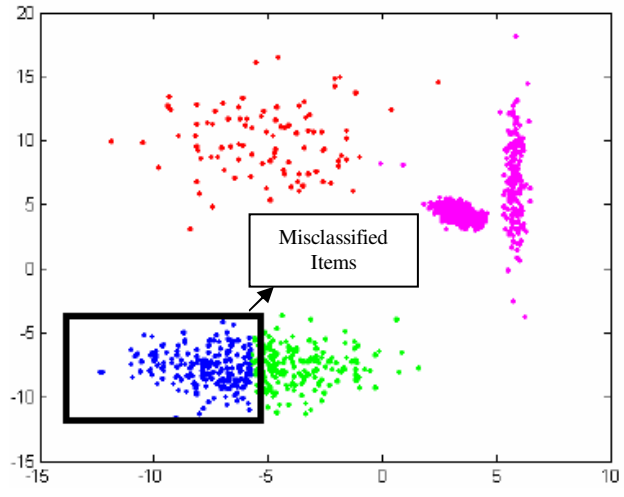
Fig. 3: Clustering results over Synthetic Data_3



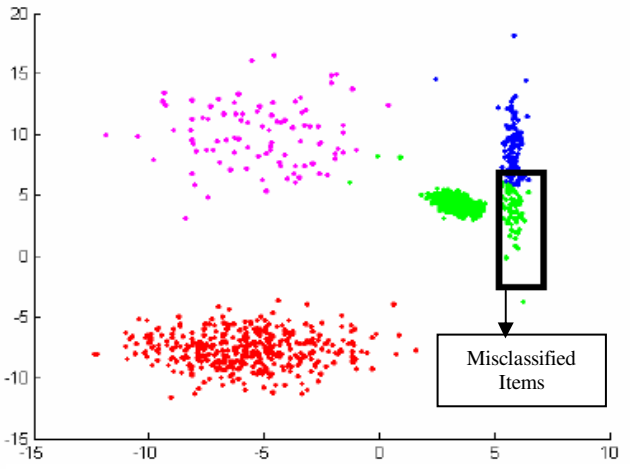
(a) Unlabelled Synthetic_Data_4



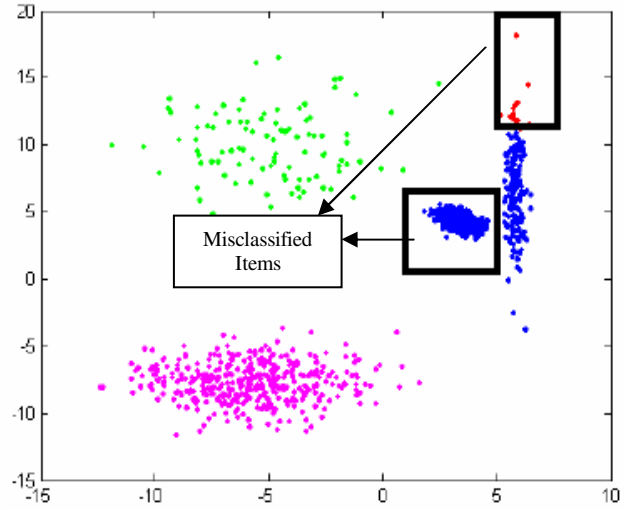
(b) Clustering with the ant-based clustering technique



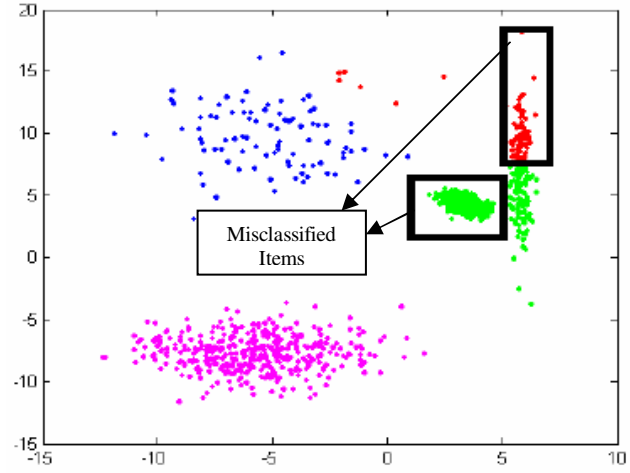
(c) Clustering with the k-means algorithm



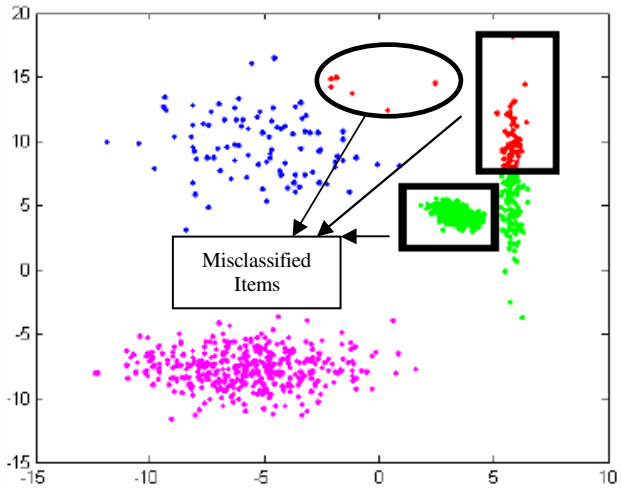
(d) Clustering with the FCM algorithm



(e)Clustering with hierarchical average-link agglomerative algorithm

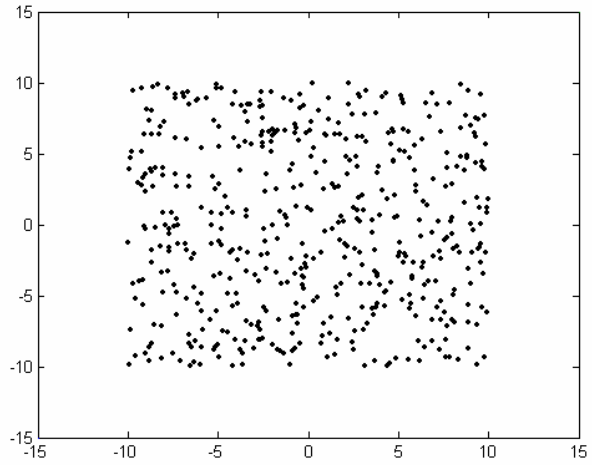


(f) Clustering with hierarchical single-link agglomerative algorithm

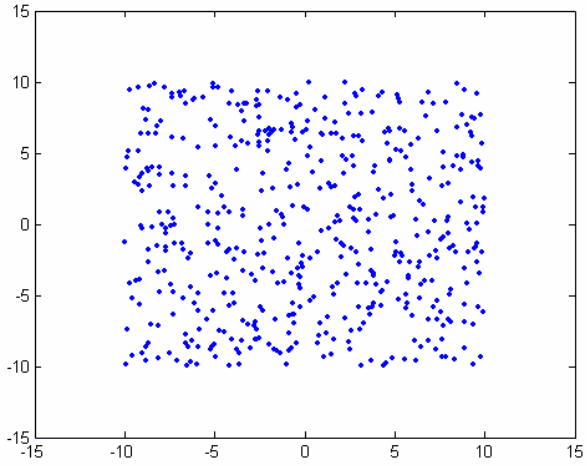


(g) Clustering with hierarchical complete-link agglomerative algorithm

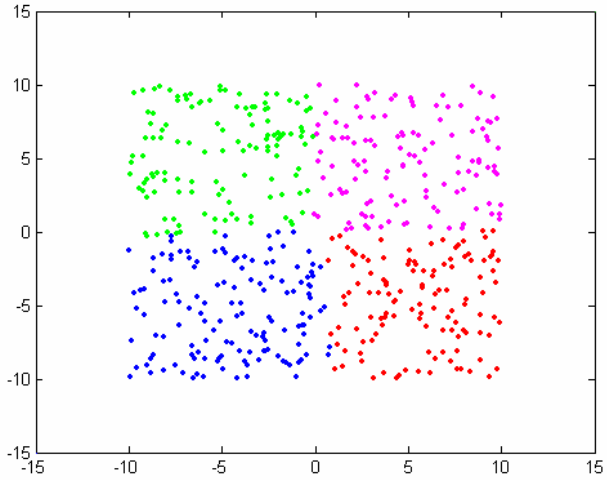
Fig. 4: Clustering results over Synthetic Data_4



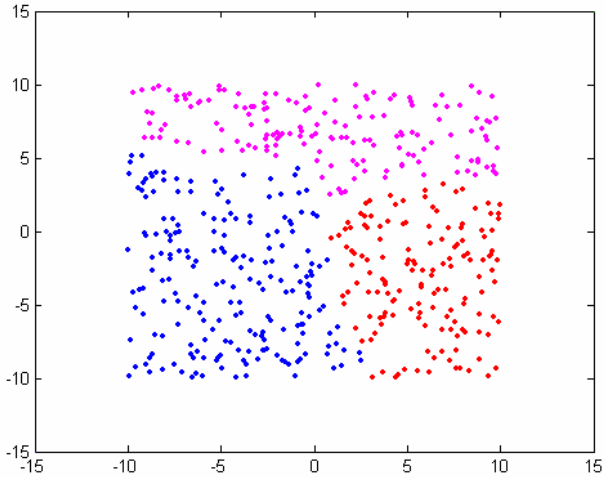
(a) Unlabelled Synthetic_Data_5



(b) Clustering with ant-based method



(c) Clustering with the automatic GCUK [23] algorithm (yielded 4 clusters)



(d) Clustering with the automatic DCPSO [23] algorithm (yielded 3 clusters)

Fig. 4: Clustering results over Synthetic Data_4

Figures 1 to 5 indicate that the performance of the ant-based clustering algorithm remains clearly and consistently superior to the five most popular clustering algorithms. Note that for synthetic data_1, performance of the three hierarchical clustering techniques remain similar to the ant-based algorithm, although the correct number of clusters had to be supplied to them in each case while the proposed method could find the optimal number of clusters easily. Substantial difference of performance occurred between the hierarchical methods and the proposed method for the more complex synthetic datasets 2 to 4. From Figure 3 (f) we see that in case of synthetic data 3, the single linkage hierarchical agglomerative algorithm yielded two big clusters (by merging points

from two actual clusters) and one false cluster containing a single point. In case of synthetic data 4, the clusters are non-spherical, of different sizes and densities and not well-separated. Except the ant-based method, all other clustering algorithms performed poorly on this dataset.

From Figure 5 (a) it is evident that synthetic data 5 is a collection of random dots over a rectangular area. The data was generated using a uni-modal probability density function and does not contain more than one natural clusters. The proposed method could detect only one single cluster in this dataset. However, we see that two recently developed clustering algorithms based on Genetic Algorithms (GA) - CGUK [18] and Particle Swarm Optimization (PSO) - DCPSO [23] could yield respectively 4 and 3 clusters for this data. Hence the two algorithms fail to detect the clustering tendency of the data before partitioning them.

Currently we are experimenting on the performances of the ant based method on higher-dimensional real life datasets and also investigating the effect of initialization and parameter selection on the algorithm. We are also comparing the performance with other metaheuristic clustering algorithms based on GA, PSO, and ACO over the real life machine learning datasets. We intend to publish the detailed results soon in a future communication.

5. Conclusions and Future Scopes

The paper has presented an ingenious approach toward automatic and shape-independent clustering. The main feature of the algorithm is that it is very simple and can handle datasets with complex shaped clusters. The proposed technique can encompass both shell data as well as solid clusters, which is evident from the pictures presented in Section 4. It can also check whether there is any real clustering tendency in the data under test.

However, the parameter estimation needs further research. Estimation using heuristic algorithms may be used, from which empirical formulae can be derived. Also in order to detect intersecting shell clusters a gradient operator may be introduced.

References

1. A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, ACM Computing Surveys, vol. 31, no.3, (1999) 264—323.

2. T Lillesand, R Keifer, Remote Sensing and Image Interpretation, John Wiley & Sons, 1994.

3. H. C Andrews, Introduction to Mathematical Techniques in Pattern Recognition, John Wiley & Sons, New York, 1972.

4. M.R Rao, Cluster Analysis and Mathematical Programming,. Journal of the American Statistical Association, Vol. 22, pp 622-626, 1971.

5. R. O. Duda , P. E. Hart. Pattern Classification and Scene Analysis. John Wiley and Sons, 1973.

6. B.S. Everitt, Cluster Analysis. Halsted Press, third edition (1993).

7. E.W. Forgy, Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of classification, Biometrics, 21, (1965) 768-9.

8. C. T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Transactions on Computers C-20 (1971), 68–86.

9. T. Mitchell, Machine Learning. McGraw-Hill, Inc., New York, NY (1997).

10. J. Mao, A. K. Jain, Artificial neural networks for feature extraction and multivariate data projection. IEEE Trans. Neural Networks.vol. 6, (1995) 296–317.

11. N. R. Pal, J. C. Bezdek, E. C.-K. Tsao, Generalized clustering networks and Kohonen's self-organizing scheme. IEEE Trans. Neural Networks, vol 4, (1993) 549–557.

12. T Kohonen, Self-Organizing Maps, Springer Series in Information Sciences, Vol 30, Springer-Verlag, (1995).

13. E. Falkenauer, Genetic Algorithms and Grouping Problems, John Wiley and Son, Chichester (1998).

14. S. Paterlini, T. Minerva, Evolutionary Approaches for Cluster Analysis. In A. Bonarini, F. Masulli, G. Pasi (eds.) Soft Computing Applications. Springer-Verlag, Berlin. (2003)167-178.

15. A. Abraham, S. Das, and S. Roy, Swarm Intelligence Algorithms for Data Clustering, *Soft Computing for Knowledge Discovery and Data Mining*, O. Maimon and L. Rokach (Eds.), Springer Verlag, Germany, ISBN 978-0-387-69934-9, pp. 279-313, 2007.

16. J. MacQueen, Some methods for classification and analysis of multivariate observations, Proceedings of the Fifth Berkely Symposium on Mathematical Statistics and Probability, 281-297 (1967).

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

18. S. Bandyopadhyay and U. Maulik, Genetic clustering for automatic evolution of clusters and application to image classification, Pattern Recognition, 35, 1197-1208, 2002.

19. M. Dorigo, V. Maniezzo, and A. Colorni, Ant System: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics–Part B, 26 (1): 29–41, 1996.

20. M. Dorigo and L. M. Gambardella, Ant Colony System: A cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation, 1 (1): 53–66, 1997.

21. J. Handl and J. Knowles, and M. Dorigo, Ant-based clustering: a comparative study of its relative performance with respect to k-means, average link and 1D-SOM. Technical Report TR/IRIDIA/2003-24. IRIDIA, Universite Libre de Bruxelles, Belgium, 2003.

22. E. Lumer and B. Faieta, Diversity and Adaptation in Populations of Clustering Ants. In Proceedings Third International Conference on Simulation of Adaptive Behavior: from animals to animate 3, Cambridge, Massachusetts, MIT press, pp. 499-508, (1994).

23. M. Omran, A. Salman, and A. Engelbrecht. Dynamic Clustering using Particle Swarm Optimization with Application in Unsupervised Image Classification. Fifth World Enformatika Conference (ICCI 2005), Prague, Czech Republic, 2005.