A Fuzzy Genetic Clustering Technique Using a New Symmetry Based Distance for Automatic Evolution of Clusters

Sriparna Saha and Sanghamitra Bandyopadhyay

Machine Intelligence Unit, Indian Statistical Institute, Kolkata-700108, India.

Email: {sanghami,sriparna_r}@isical.ac.in

Abstract

In this paper a fuzzy point symmetry based genetic clustering technique (Fuzzy-VGAPS) is proposed which can determine the number of clusters present in a data set as well as a good fuzzy partitioning of the data. A new fuzzy cluster validity index, FSym-index, which is based on the newly developed point symmetry based distance is also proposed here. FSym-index provides a measure of goodness of clustering on different fuzzy partitions of a data set. Maximum value of FSym-index corresponds to the proper clustering present in a data set. The flexibility of Fuzzy-VGAPS is utilized in conjunction with the fuzzy FSym-index to determine the number of clusters present in a data set as well as a good fuzzy partition of the data. The results of the fuzzy VGAPS are compared with those obtained by fuzzy version of variable string length genetic clustering technique (Fuzzy-VGA) optimizing XB-index. The effectiveness of the Fuzzy-VGAPS is demonstrated on four artificial data sets and two real-life data sets.

Keyword: Clustering, Cluster Validity Index, Point Symmetry, Kd-tree, Genetic Algorithm, Variable String Length

1. Introduction

Clustering [5] is a fundamental problem in data mining with innumerable applications spanning many fields. 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. One of the basic feature of shapes and objects is symmetry. Su and Chou have proposed a point symmetry (PS) distance based similarity measure [9]. This work is extended in [4] in order to overcome some of the limitations existing in [9]. It has further been shown in [1] that the PS distance proposed in [4] also has some serious drawbacks and a new PS distance (d_{ps}) is defined in [1] in order to remove these drawbacks. For reducing complexity of point symmetry distance computation, Kd-tree based data structure is used.

Fuzzy C-means (FCM) [3] is a widely used clustering technique that uses the principles of fuzzy sets to evolve a partition matrix $U(X)$ while minimizing the measure $\sum_{i=1}^K \sum_{j=1}^n (u_{ij})^m D^2(\bar{c}_i, \bar{x}_j)$ where $D(\bar{c}_i, \bar{x}_j)$ represents the distance from the point $\bar{x}_j$ ($j = 1, \dots, n$) to the center of the i th cluster, $\bar{c}_i$, ($i = 1, \dots, K$), m is the weighting coefficient, and $U(X)$ is the partition matrix of size $K \times n$, which can be represented as $U = [u_{ij}]$, $i = 1, \dots, K$; $j = 1, \dots, n$, where u_{ij} is the membership of pattern x_j to cluster i . However, FCM has three major limitations: it requires the *a priori* specification of the number of clusters (K), it often gets stuck at suboptimal solutions based on the initial configuration (recently, proof of its convergence to a local minima or saddle point of the error function has been provided in [6]) and it can detect only hyper-spherical shaped clusters. In most of the real-life situations the number of clusters in a data set is not known *a priori*. The real challenge in this situation is to be able to automatically evolve a proper value of K as well as providing the appropriate clustering of a data set.

A fuzzy Genetic Algorithm (GA) based clustering technique, Fuzzy-VGA has been proposed in [7], which is able to automatically evolve the appropriate clustering for hyperspherical data sets. However for clusters with other than hyperspherical shapes, this algorithm is likely to fail, as it uses, like the FCM, the Euclidean distances of the points from the respective cluster centroids for computing the fitness value. In order to overcome this limitation, in this article a fuzzy variable string length genetic point symmetry (Fuzzy-VGAPS) based clustering technique is proposed. Here membership values of points to different clusters are computed based on point symmetry based distance rather than Euclidean distance. This enables the proposed algorithm to automatically evolve the appropriate clustering of all types of clusters, both convex and non convex, which have some symmetrical structures. The chromosome encodes the centres of a number of clusters, whose value may vary. A new fuzzy cluster validity index named *FSym-index* is proposed here and thereafter it is utilized for computing the fitness of the chromosomes. The superiority of the pro-

posed genetic clustering technique for evolving the appropriate fuzzy partitioning of a dataset, as compared with the existing Fuzzy-VGA, is demonstrated on four artificial and two real-life data sets having different characteristics.

2. Fuzzy-VGAPS Clustering: Fuzzy Variable String Length Genetic Point Symmetry Based Clustering Technique

In this section, the use of variable string length genetic algorithm using a newly developed point symmetry based distance is proposed for automatically evolving the near-optimal $K \times n$ nondegenerate fuzzy partition matrix U^* . The set $\mathcal{U}$ of all possible nondegenerate partition matrices is represented as $\mathcal{U} = \{U \in R^{K \times n} | \sum_{i=1}^K u_{ij} = 1, j = 1, \dots, n, 0 < \sum_{j=1}^n u_{ij} < n, \text{ and } u_{ij} \in [0, 1]\}$. Here we have considered the best partition to be the one that corresponds to the maximum value of the proposed *FSym*-index which is defined later. Here both the number of clusters as well as the appropriate fuzzy clustering of the data is evolved simultaneously using the search capability of genetic algorithms.

In GAs, the parameters of the search space are encoded in the form of strings (called *chromosomes*). A collection of such strings is called *population*. Initially a random population is created, which represents different points in the search space. An *objective/fitness* function is associated with each string that represents the degree of *goodness* of the solution encoded in the string. Based on the principle of survival of the fittest, a few of the strings are selected and each is assigned a number of copies that go into the mating pool. Biologically inspired operators like *crossover* and *mutation* are applied on these strings to yield a new population. The process of selection, crossover, and mutation continues for a fixed number of generations or till a termination condition is satisfied.

For the purpose of clustering, each chromosome encodes a possible partitioning of the data, the goodness of which is computed as a function of an appropriate cluster validity index. This index must be optimized in order to obtain the best partitions. Since the number of clusters is considered to be variable, the string lengths of different chromosomes in the same population are allowed to vary. As a consequence, the crossover and mutation operators are suitably modified in order to tackle the concept of variable length chromosomes. The technique is described below in detail.

String Representation and Population Initialization: In Fuzzy-VGAPS clustering, the chromosomes are made up of real numbers which represent the coordinates of the centers of the partitions. For each string i in the population ($i = 1, \dots, P$ where P is the size of the population), a random number K_i in the range $[K_{min}, K_{max}]$ is generated. This string is assumed to encode the centres of K_i clusters. For

initializing these centres, K_i points are chosen randomly from the data set. These points are distributed randomly in the chromosome.

Here K_{min} is set equal to 2 and K_{max} is set equal to $\sqrt{n}$ where n is the number of data points present in the data set. 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.

Fitness Computation: This is composed of two steps. Firstly membership values of n points to different clusters are computed by using the newly developed point symmetry based distance d_{ps} . Next, the *FSym*-index is computed and used as a measure of the fitness of the chromosome.

Point Symmetry Based Distance, d_{ps} : The proposed point symmetry based distance $d_{ps}(\bar{x}, \bar{c})$ associated with point $\bar{x}$ with respect to a center $\bar{c}$ is defined as follows: Let a point be $\bar{x}$. The symmetrical (reflected) point of $\bar{x}$ with respect to a particular centre $\bar{c}$ is $(2 \times \bar{c} - \bar{x})$. Let us denote this by $\bar{x}^*$. Let the first and second unique nearest neighbors of $\bar{x}^*$ be at Euclidean distances of d_1 and d_2 respectively. Then

$$d_{ps}(\bar{x}, \bar{c}) = \frac{(d_1 + d_2)}{2} \times d_e(\bar{x}, \bar{c}) \quad (1)$$

where $d_e(\bar{x}, \bar{c})$ is the Euclidean distance between the point $\bar{x}$ and $\bar{c}$.

The computation of point symmetry based distance is highly complex requiring $O(n)$ operations for each point. In order to make the nearest neighbor computation process efficient, we have used the 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>), is used for this purpose. Here ANN is used to find d_1 and d_2 in Equation 1 efficiently. The Kd-tree structure can be constructed in $O(n \log n)$ time and takes $O(n)$ space, while the search complexity is $O(\log n)$.

Computing the Membership Values: For each point $\bar{x}_j, j = 1, 2, \dots, n$, the membership values to K different clusters are calculated in the following way. Find the cluster center nearest to $\bar{x}_j$ in the symmetrical sense. That is, we find the cluster center k that is nearest to the input pattern $\bar{x}_j$ using the minimum-value criterion:

$$k = \text{Argmin}_{i=1, \dots, K} d_{ps}(\bar{x}_j, \bar{c}_i)$$

where the point symmetry based distance $d_{ps}(\bar{x}_j, \bar{c}_i)$ is computed by Equation 1. Here, $\bar{c}_i$ denotes the center of the i th cluster. If the point symmetry distance $d_{ps}(\bar{x}_j, \bar{c}_k)$ is smaller than a pre-specified parameter θ , then we update the membership u_{ij} using the following criterion:

$$u_{ij} = 1, \text{ if } i = k$$

$$u_{ij} = 0, \text{ if } i \neq k.$$

Otherwise, we update the membership u_{ij} by using following rule which corresponds to the normal Fuzzy C-Means [3] algorithm:

$$u_{ij} = \frac{1}{\sum_{i=1}^K \left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{m-1}}}$$

where $m \in [1, \infty)$ is a weighting exponent called the fuzzifier. Here we have chosen $m = 2$. d_{ij} represents the distance from a pattern $\bar{x}_j$ to the cluster center $\bar{c}_i$. The value of θ is kept equal to the maximum nearest neighbor distance among all the points in the data set. It may 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

$$d_{NN}^{max} = \max_{i=1, \dots, N} d_{NN}(\bar{x}_i),$$

where $d_{NN}(\bar{x}_i)$ is the nearest neighbor distance of $\bar{x}_i$. Assuming that reflected point of $\bar{x}$ with respect to the cluster centre $\bar{c}$ lies within the data space, it may be noted that $d_1 \leq \frac{d_{NN}^{max}}{2}$. Ideally, a point $\bar{x}$ is exactly symmetrical with respect to some $\bar{c}$ if $d_1 = 0$. However considering the uncertainty of the location of a point as the sphere of radius d_{NN}^{max} around $\bar{x}$, we have kept the threshold θ equals to d_{NN}^{max} . Thus the computation of θ is automatic and does not require user intervention.

Updating the Centers: The centers encoded in a chromosome are updated using the following equation as in FCM

$$\bar{c}_i = \frac{\sum_{j=1}^n (u_{ij})^m \bar{x}_j}{\sum_{j=1}^n (u_{ij})^m}, \quad 1 \leq i \leq K.$$

Fitness Calculation: The fitness of a chromosome is computed using the $FSym$ -index. Let K cluster centres be denoted by $\bar{c}_i$ where $1 \leq i \leq K$ and $U(X) = [u_{ij}]_{K \times n}$ is a partition matrix for the data. Then $FSym$ -index is defined as follows:

$$FSym(K) = \left(\frac{1}{K} \times \frac{1}{E_K} \times D_K\right), \quad (2)$$

where K is the number of clusters. Here,

$$E_K = \sum_{i=1}^K E_i \quad (3)$$

such that

$$E_i = \sum_{j=1}^n (u_{ij} \times d_{ps}(\bar{x}_j, \bar{c}_i)) \quad (4)$$

and

$$D_K = \max_{i,j=1}^K \|\bar{c}_i - \bar{c}_j\| \quad (5)$$

D_K is the maximum Euclidean distance between two cluster centers among all centers. $d_{ps}(\bar{x}_j, \bar{c}_i)$ is computed by Equation 1.

The objective is to maximize the $FSym$ -index in order to obtain the actual number of clusters and to achieve proper clustering. As formulated in Equation 2, $FSym$ is a composition of three factors, these are $1/K$, $1/E_K$ and D_K . The first factor increases as K increases; as $FSym$ needs to be maximized for optimal clustering, so it will prefer to decrease the value of K . The second factor is the within cluster total symmetrical distance. For clusters which have good symmetrical structure, E_i value is less. This, in turn, indicates that formation of more number of clusters, which are symmetrical in shape, would be encouraged. Finally the third factor, D_K , measuring the maximum separation between a pair of clusters, increases with the value of K . As these three factors are complementary in nature, so they are expected to compete and balance each other critically for determining the proper partitioning. The fitness function for chromosome j is defined as $FSym_j$ i.e., the $FSym$ index computed for the chromosome. The objective of GA is to maximize this fitness function.

Selection: Conventional proportional selection is applied on the population of strings. Here, a string receives a number of copies that is proportional to its fitness in the population. We have used roulette wheel strategy for implementing the proportional selection scheme.

Crossover: For the purpose of crossover, the cluster centers are considered to be indivisible, i.e., the crossover points can only lie in between two cluster centers. The crossover operation, applied stochastically, must ensure that information exchange takes place in such a way that both the offspring encode the centers of at least two clusters. For this, the operator is defined as follows [7]: Let parent chromosomes P_1 and P_2 encode M_1 and M_2 cluster centers respectively. τ_1 , the crossover point in P_1 , is generated as $\tau_1 = \text{rand}() \bmod M_1$. Let τ_2 be the crossover point in P_2 , and it may vary in between $[LB(\tau_2), UB(\tau_2)]$, where $LB(\tau_2)$ and $UB(\tau_2)$ indicate the lower and upper bounds of the range of τ_2 respectively. $LB(\tau_2)$ and $UB(\tau_2)$ are given by

$$LB(\tau_2) = \min[2, \max[0, 2 - (M_1 - \tau_1)]] \text{ and } UB(\tau_2) = M_2 - \max[0, 2 - \tau_1].$$

Therefore τ_2 is given by

$$\tau_2 = LB(\tau_2) + \text{rand}() \bmod (UB(\tau_2) - LB(\tau_2)),$$

if $(UB(\tau_2) \geq LB(\tau_2))$,

$$\tau_2 = 0 \text{ otherwise.}$$

It can be verified by some simple calculations that if the crossover points τ_1 and τ_2 are chosen according to the above rules, then none of the offspring generated would have less than two clusters.

Crossover probability is selected adaptively as in [8]. The expressions for crossover probabilities are computed as follows. Let f_{max} be the maximum fitness value of the current population, $\bar{f}$ be the average fitness value of the popula-

Figure 1. Clustering result on (a) Sym_5.2 using Fuzzy-VGAPS (b) Sym_2.2 using Fuzzy-VGAPS (c) Data_6.2 using Fuzzy-VGAPS (d) Data_4.3 using Fuzzy-VGAPS (e) Sym_5.2 using Fuzzy-VGA (f) Sym_2.2 using Fuzzy-VGA

tion 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:

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

$$\mu_c = k_3, \text{ if } f' \leq \bar{f}.$$

Here, as in [8], the values of k_1 and k_3 are kept equal to 1.0. Note that, when $f_{max} = \bar{f}$, then $f' = f_{max}$ and μ_c will be equal to k_3 . 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.

Mutation: Mutation is applied on each chromosome with probability μ_m . Mutation is of three types. (1) Each valid position (i.e., which is not '#') in a chromosome is mutated with probability μ_m in the following way. A number δ in the range $[0, 1]$ is generated with uniform distribution. If the value at that position is v , then after mutation it becomes $v \times (1 \pm 2\delta)$, if $v \neq 0$, otherwise for $v = 0$ it will be equal to $\pm 2\delta$. The '+' or '-' sign occurs with equal probability. (2) One randomly generated valid position is removed and replaced by '#'. (3) One randomly chosen invalid position is replaced by randomly chosen point from the data set.

Any one of the above mentioned types of mutation is applied randomly on a particular chromosome if it is selected for mutation. The mutation probability is also selected adaptively for each chromosome as in [8]. The expression for mutation probability, μ_m , is given below:

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

$$\mu_m = k_4 \text{ if } f \leq \bar{f}.$$

Here, values of k_2 and k_4 are kept equal to 0.5. This adaptive mutation helps GA to avoid getting stuck at local optimum. When GA converges to a local optimum, i.e., when $f_{max} - \bar{f}$ decreases, μ_c and μ_m both will be increased. As a result GA will come out of local optimum.

Termination: In this paper, we have executed the algorithm for a fixed number of generations. Moreover, the elitist model of GAs has been used, where the best string seen so far is stored in a location within the population. The best string of the last generation provides the solution to the clustering problem.

3 Implementation Results and Comparative Study

The experimental results showing the effectiveness of the Fuzzy-VGAPS algorithm are provided for four artificial data sets viz., (Sym_5.2, Sym_2.2, Data_6.2, Data_4.3) and two real life data sets (*cancer* and *glass* obtained from <http://www.ics.uci.edu/~mllearn/MLRepository.html>). The description of the data sets is given in Table 1. The population size, P , is set equal to 100. Fuzzy-VGAPS is executed for a total of 20 generations. For all the data sets, as is evident from Table 1, Fuzzy-VGAPS is able to find out appropriate number of clusters and the proper partitioning. Figures 1(a), 1(b), 1(c) and 1(d) show the final clus-

tered results obtained after application of Fuzzy-VGAPS on Sym_5_2, Sym_2_2, Data_6_2 and Data_4_3. For *cancer* and *glass* data sets which are of nine-dimensional, it is not possible to show the clustered results visually. For both the real life data sets, Fuzzy-VGAPS is again able to find out the proper number of clusters.

Table 1 also shows the performance of Fuzzy-VGA for all the data sets. As is evident, Fuzzy-VGA is able to detect the proper number of clusters as well as proper clustering for Data_6_2 and Data_4_3 but it fails for Sym_5_2 and Sym_2_2. Figure 1(e) and 1(f) show the clustering result obtained by Fuzzy-VGA on Sym_5_2 and Sym_2_2 respectively. Fuzzy-VGA obtained, incorrectly, 8 and 6 clusters for these two data sets respectively. Clustering results on Data_6_2 and Data_4_3 obtained by Fuzzy-VGA are same as that of Fuzzy-VGAPS and are therefore omitted.

To compare Fuzzy-VGAPS with Fuzzy-VGA clustering algorithm [7] for *cancer* and *glass* data sets, *Minkowski Score* (MS) [2] is calculated after application of both the algorithms. MS is a measure of the quality of a solution given the true clustering. For MS, the optimum score is 0, with “lower scores” being “better”. For *cancer* dataset, MS score is 0.3233 for Fuzzy-VGAPS and 0.37 for Fuzzy-VGA. For *glass* data set, Fuzzy-VGA provided 9 clusters where as Fuzzy-VGAPS is able to detect the proper number of clusters. The MS score is 0.7223 for the latter while it is 1.056 for the former. From the above results it is evident that Fuzzy-VGAPS is not only able to find the proper cluster number, it also provided significantly good clustering (both visually as in Figure 1, and also with respect to the MS scores) in general. Moreover, we have also conducted statistical test ANOVA, and found that the difference in the mean MS values obtained by Fuzzy-VGAPS and Fuzzy-VGA are statistically significant. For *cancer*, mean difference in mean MS obtained by two algorithms over ten runs is $-4.67 \text{ E-}02$ which is statistically significant (significance value is 0.00). While for *glass*, mean difference in mean MS over ten runs obtained by two algorithms is $-3.33 \text{ E-}02$ which is again statistically significant (significance value is 0.00). Due to lack of space results on some more non-convex data sets are excluded from the paper.

4 Conclusion

In this paper a fuzzy clustering technique, Fuzzy-VGAPS, is proposed which finds the membership values of the data points to different clusters based on the point symmetry based distance. It is able to automatically evolve the appropriate clustering of a data set. A new symmetry based fuzzy cluster validity index named *FSym-index* is also proposed in this article. The experimental results on several artificial and real life data sets of varying complexities show that Fuzzy-VGAPS is able to detect proper number of clusters as well as proper clustering from a data set having any

Table 1. Results obtained with the different data sets using Fuzzy-VGAPS and Fuzzy-VGA

Name	No of points	dimension	Actual Cluster No	Obtained Fuzzy VGAPS	Obtained Fuzzy VGA
Sym_5_2	850	2	5	5	8
Sym_2_2	400	2	2	2	6
Data_6_2	300	2	6	6	6
Data_4_3	400	3	4	4	4
<i>cancer</i>	683	9	2	2	2
<i>glass</i>	214	9	6	6	9

type of clusters, irrespective of their geometrical shape and overlapping nature, as long as they possess the characteristic of symmetry. It is well-known that Kd-tree based algorithms can be exponential in the dimensionality of the data. So in future we would like to examine whether the proposed scheme scales up well with dimensionality. The efficiency of Fuzzy-VGAPS on some large data sets needs to be shown also. Finally, comparison with other algorithms, like single linkage algorithm [5], needs to be made. Authors have already started working in this direction.

References

- [1] S. Bandyopadhyay and S. Saha. GAPS: A clustering method using a new point symmetry based distance measure. *Pattern Recog.*, Revised. URL: http://www.isical.ac.in/~sriparna_r/paper.html.
- [2] A. Ben-Hur and I. Guyon. *Detecting Stable Clusters using Principal Component Analysis in Methods in Molecular Biology*. Humana press, 2003.
- [3] J. C. Bezdek. *Fuzzy Mathematics in Pattern Classification*. PhD thesis, 1973.
- [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*, pages 209–213. Crete, Greece, 2002.
- [5] B. S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. London: Arnold, 2001.
- [6] L. Gröll and J. Jäkel. A new convergence proof of fuzzy c-means. *IEEE Transactions Fuzzy Systems*, 13(5):717–720, 2005.
- [7] U. Maulik and S. Bandyopadhyay. Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification. *IEEE Transactions Geoscience and Remote Sensing*, 41(5):1075–1081, 2003.
- [8] M. Srinivas and L. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 24(4):656–667, April, 1994.
- [9] 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*, 23(6):674–680, 2001.