International Journal of Database Theory and Application
Vol. 2, No. 2, June 2009

Process of Extracting Uncover Patterns from Data: A Review

Julian Dermoudy[1], Byeong-Ho Kang[2], Debnath Bhattacharyya[3*], Seung-Hwan Jeon[4], and Alisherov Farkhod A.[5]

[1,2]School of Computing and Information Systems, University of Tasmania, Australia
[3]Computer Science and Engineering Department, Heritage Institute of Technology Kolkata-700107, India.
[2]Hannam University, Daejeon-306791, Korea
{Dermoudy,bhkang}@utas.edu.au[1, 2], debnathb@gmail.com[3], jeonseung66@hotmail.com[4], sntdvl@yahoo.com[5]*

*(*correspondent author)*

Abstract

Extracting required patterns from huge amount of mixed data is an area of interest to the researchers. Various promising and already established algorithms are currently using in the name of Data Clustering. Clustering is used for partitioning the data into number of data sets or group. In this paper, we review four popular clustering algorithms from data mining perspective.

Keywords: *Clustering, Data mining, genetic algorithm, fuzzy.*

1. Introduction

Fast retrieval of the relevant information from the databases has been a significant issue. Different techniques have been developed for this purpose; one of them is Data Clustering. Data clustering is a method in which we make cluster of objects that are somehow similar in characteristics. The criterion for checking the similarity is implementation dependent.

There is an increasing interest in the use of clustering methods in pattern recognition, image processing and information retrieval, clustering is also used in other areas like, bio-medical engineering, cancer research, bio-informatics, GIS and marketing. Clustering actually, known as unsupervised classification in the area of numerical analysis, vector quantization, and learning by observation. The field of spatial analysis of point patterns is also related to cluster analysis. The importance and interdisciplinary nature of clustering is evident through its vast literature.

Clustering algorithms are used extensively not only to organize and categorize data, but are also useful for data compression and model construction. By finding similarities in data, one can represent similar data with fewer symbols for example. Also if we can find groups of data, we can build a model of the problem based on those groupings.

Another reason for clustering is to discover relevance knowledge in data. Francisco Azuaje et al. [1] implemented a Case Based Reasoning (CBR) system based on a Growing Cell Structure (GCS) model. Data can be stored in a knowledge base that is indexed or categorized by cases; this is what is called a Case Base. Each group of cases is assigned to a certain category. Using a Growing Cell Structure (GCS) data can be added or removed based on the learning scheme used. Later when a query is presented to the model, the system retrieves the most relevant cases from the case base depending on how close those cases are to the query.

M.N.M Sap and Ehsan Mohebi, explained self organizing map could be used as a tool in exploratory phase of data mining and pattern recognition [2]. In that paper, authors proposed a technique using rough set theory to handle cluster uncertainty.

There are many clustering methods available, and each of them may give a different grouping of a dataset. The choice of a particular method will depend on the type of output desired, the known performance of method with particular types of data, the hardware and software facilities available and the size of the dataset.

In general, we are considering the cluster, class or group analysis here in this paper. In cluster analysis [3], the terms cluster, group, and class have been used in an essentially intuitive manner without a uniform definition. Everitt suggested [4] that if using a term such as cluster produces an answer of value to the investigators, then it is all that is required.

Generally, the common sense of a cluster will combine various plausible criteria and require [3], for example, all objects in a cluster to

a. share the same or closely related properties;
b. show small mutual distances or dissimilarities;
c. have "contacts" or "relations" with at least one other object in the group; or
d. be clearly distinguishable from the complement, i.e., the rest of the objects in the data set.

Carmichael et al. also suggested [5] that the set contain clusters of points if the distribution of the points meets the following conditions:

a. There are continuous and relative densely populated regions of the space.
b. These are surrounded by continuous and relatively empty regions of the space.

But, in reality, the above two conditions or rules violet frequently.

Data clustering (or just clustering), also called cluster analysis, segmentation analysis, taxonomy analysis, or unsupervised classification, is a method of creating groups of objects, or clusters, in such a way that objects in one cluster are very similar and objects in different clusters are quite distinct. Data clustering is often confused with classification, in which objects are assigned to predefined classes. In data clustering, the classes are also to be defined [3]. To elaborate the concept a little bit, we consider the following example:

A gene expression data set can be represented by a real-valued expression matrix.

$$D = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix},$$

where n is the number of genes, d is the number of experimental conditions or samples, and xij is the measured expression level of gene i in sample j. Since the original gene expression matrix contains noise, missing values, and systematic variations, preprocessing is normally required before cluster analysis can be performed. We are considering lossless grayscale images to avoid noise to some extent, however, the noise removal is a separate problem altogether.

Gene expression data can be clustered in two ways. One way is to group genes with similar expression patterns, i.e., clustering the rows of the expression matrix D. Another way is to group different samples on the basis of corresponding expression profiles that is, clustering the columns of the expression matrix D.

Image segmentation is the decomposition of a graylevel or color image into homogeneous tiles [6]. In image segmentation, cluster analysis is used to detect borders of objects in an image.

Clustering constitutes an essential component of so-called data mining, a process of exploring and analyzing large amounts of data in order to discover useful information [7]. Clustering is also a fundamental problem in the literature of pattern recognition. Figure 1 gives a schematic list of various data-mining tasks and indicates the role of clustering in data mining.

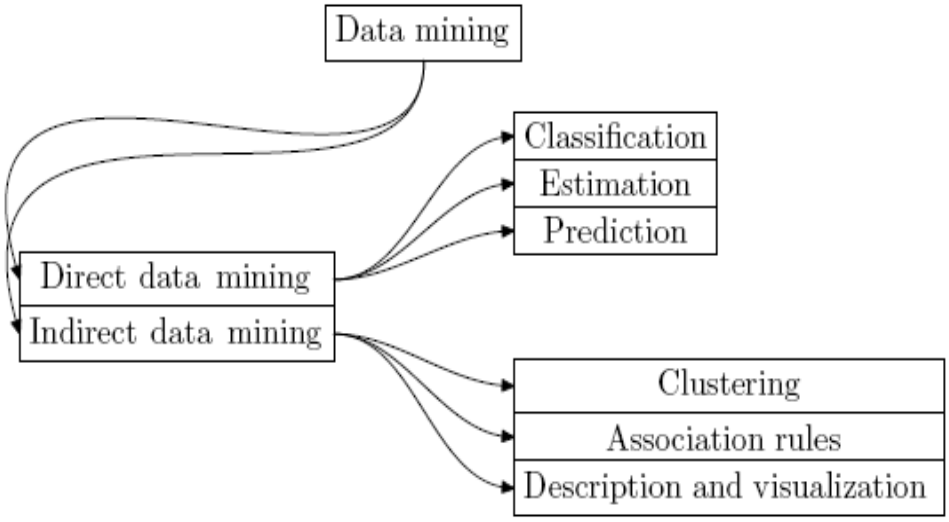

Figure 1. Data-mining tasks

2. Previous works

Roughly speaking, by data clustering, we mean that for a given set of data points and a similarity measure, we regroup the data such that data points in the same group are similar and data points in different groups are dissimilar. Obviously, this type of problem is encountered in many applications, such as text mining, gene expressions, customer segmentations, and image processing, to name just a few.

Fashing and Tomasi suggested [8] that he mean shift algorithm is a very general iterative procedure to the extent that some well-known clustering algorithms are its special cases. The mean shift algorithm is not only an intuitive and basic procedure but also a deterministic process.

It is more efficient than gradient descent or ascent methods in terms of adapting to the right step size.

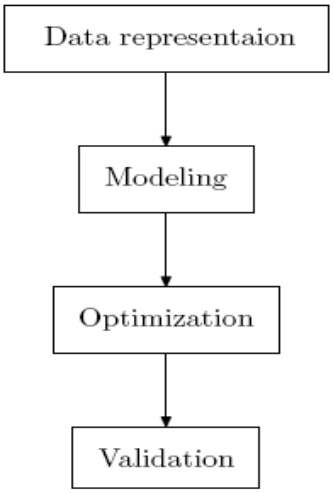

Figure 2. Process of data clustering.

There are also some factors that make the mean shift algorithm not popular. For example, the computational cost of an iteration of the mean shift algorithm is O(n2), where n is the number of data points in the data set. The mean shift algorithm is also not suitable for high-dimensional data sets and large data sets.

As a fundamental pattern recognition problem, a well-designed clustering algorithm usually involves the following four design phases: data representation, modeling, optimization, and validation [9]. Figure 2 shows the schematic diagram also. The data representation phase predetermines what kind of cluster structures can be discovered in the data. On the basis of data representation, the modeling phase defines the notion of clusters and the criteria that separate desired group structures from unfavorable ones. The goal of clustering is to assign data points with similar properties to the same groups and dissimilar data points to different groups. In our experiment, we are searching for the similar properties of a particular class, if not similar then discard.

An investigation has carried [10] out to formulate some theoretical results regarding the behavior of a genetic-algorithm-based pattern classification methodology, for an infinitely large number of training data points n, in an N-dimensional space RN. It has proved that for $n \rightarrow \infty$, and for a sufficiently large number of iterations, the performance of this classifier (when hyperplanes are considered to generate class boundaries) approaches that of the Bayes classifier, which is the optimal classifier when the class distributions and the a priori probabilities are known. It has shown that the optimum number of hyperplanes generated by the proposed classifier is equal to that required to model the Bayes decision boundary when there exists only one partition of the feature space that provides the Bayes error probability. Extensive experimental results on overlapping data sets following triangular and normal distributions with both linear and non-linear class boundaries are provided that conform to these claims. The claims also hold good when circular surfaces are considered as constituting elements/segments of boundaries. It is also shown experimentally that the variation of recognition score with a priori class probability for both the classifiers is similar.

A two-stage clustering algorithm has been proposed [11], which employs variable string length genetic scheme and a multiobjective genetic clustering algorithm. It is based on the novel concept of points having significant membership to multiple classes. An iterated version of the well-known Fuzzy C-Means is also utilized for clustering.

A new genetic search strategy involving chromosome differentiation into two classes and a restricted form of crossover operation is defined [12]. Its application to multi- dimensional pattern recognition problems is studied. Superiority of the classifier is established for four sets of different artificial and real life data.

The concept of chromosome differentiation, commonly witnessed in nature as male and female sexes, is incorporated in genetic algorithms with variable length strings for designing a nonparametric classification methodology. Its significance in partitioning different landcover regions from satellite images, having complex/overlapping class boundaries, is demonstrated. The classifier is able to evolve automatically the appropriate number of hyperplanes efficiently for modeling any kind of class boundaries optimally. Merits of the system over the related ones are established through the use of several quantitative measures [13].

Automatic identification of genes has been an actively researched area of Bioinformatics. S. Bandyopadhyay, U. Maulik and D. Roy in 2008, described [14] many of the current gene finding methods used in the computational intelligence techniques. In this article, a detailed survey on the existing classical and computational intelligence based methods for gene identification has carried out. This includes a brief description of the classical and computational intelligence methods before discussing their applications to gene finding. In addition a long list of available gene finders is compiled. For convenience of the readers, the list is enhanced by mentioning their corresponding web sites and commenting on the general approach adopted.

3. Clustering techniques

We review four of the most popular widely used clustering techniques in this section. These are:

a. K-means Clustering Algorithm,
b. Fuzzy C-means Clustering Algorithm,
c. Clustering Using Genetic Algorithms (fixed k), and
d. Variable String Length GAs (VGA) for Automatic Clustering.

3.1. K-means Clustering Algorithm

Problem: Cluster a set X of n points in d dimensional space into c clusters.

X: set of n points

$X = \{ x_1, x_2, \ldots, x_n\}$

$x_i = [x_{i1}, x_{i2}, \ldots, x_{id}]$ for $i=1,2,\ldots,n$.

V: set of c cluster centers

$V = \{v_1, v_2, \ldots, v_c\}$

$v_j = [v_{j1}, v_{j2}, \ldots, v_{jd}]$ for $j=1,2,\ldots,c$.

Initialization:

Select randomly chosen c points from the data as the initial centers.

$v_j = x_k$ where $k= rand([1\ldots n])$ for $j=1,2,\ldots,c$.

Clustering:

Assign $x_k \rightarrow$ Cluster j such that x_k is closest to v_j.

$d(x_k, v_j) = min\ d(x_k, v_j)$ for $j=1,2,\ldots,c$.

$d(x_k, v_j) = ||x_k - v_j||$

Output :

n_i = number of points in cluster $i=1,2,\ldots,c$.

x_j^i = the *jth* point in cluster *i, i=1,2,…,c,* and *j=1, 2,…, n_i.*

Update:

Compute new cluster center as the centroid of the points assigned to the cluster

$v_i^* = (1/n_i)$ S x_j^i for $i=1, 2, \ldots, c$

Loop or Terminate:

Terminate if there is no/negligible change in centroids

If $v_i^* = v_i$, for $i=1, 2, \ldots, c$ → terminate.

Otherwise go to Clustering step.

Minimizes *J*, the mean squared error:

J = S S $d^2(x_k^j, v_j)$, $j=1,2,\ldots c$ and $k=1, 2, \ldots, n_j$.

Analysis of algorithm with pictorial support:
Initialize Centers, partition the data into *c=3* clusters, shown in Figure 3.
Assign points, shown in Figure 4.
Update Centers, shown in Figure 5.
Reassign Points, shown in Figure 6.
Update Centers, shown in Figure 7.
Reassign Points and Update Centers, shown in Figure 8.
Final Clustering after a few iterations, shown in Figure 9.

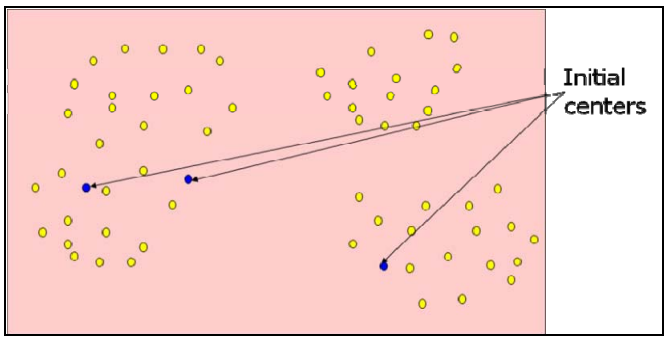

Figure 3. Data Points With 3 Initial Centers.

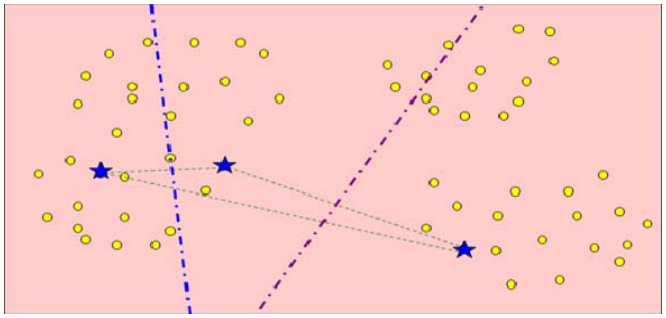

Figure 4. Assign Points.

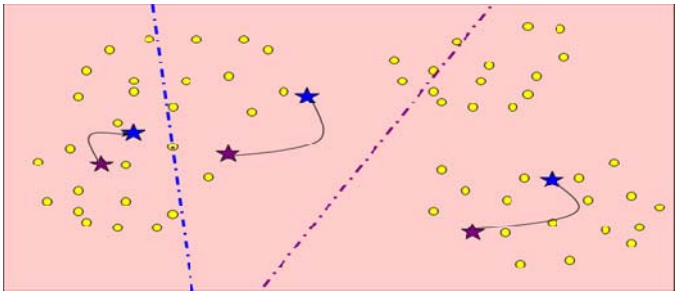

Figure 5. Update Centers.

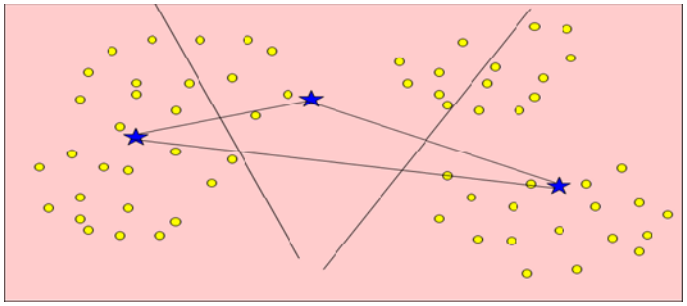

Figure 6. Reassign Points.

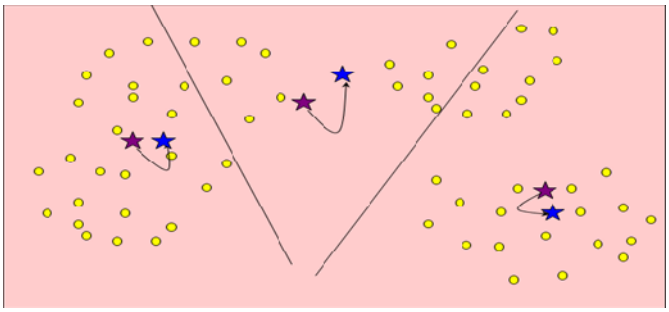

Figure 7. Update Centers.

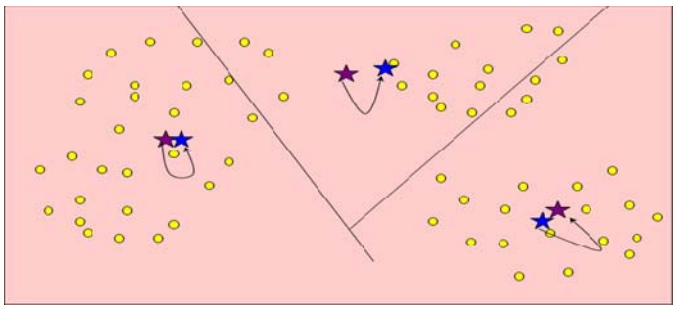

Figure 8. Reassign Points And Update Centers.

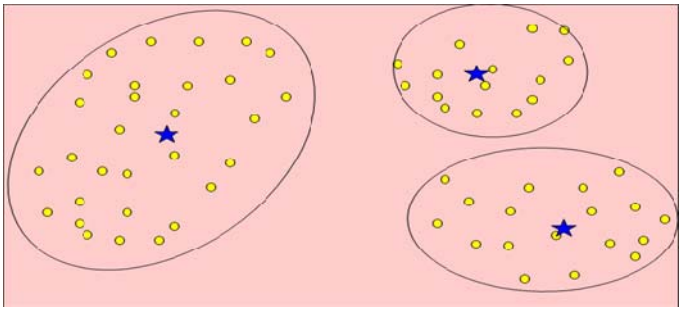

Figure 9. Final Clustering.

Main Output:

A c-partition of X, which is a (c x n) matrix U.

Common Additional Output:

V: set of c cluster centers

$V = \{v_1, v_2, \ldots, v_c\}$

Illustration shown in Figure 10a-c, with following set: n=188, d=2, c=4

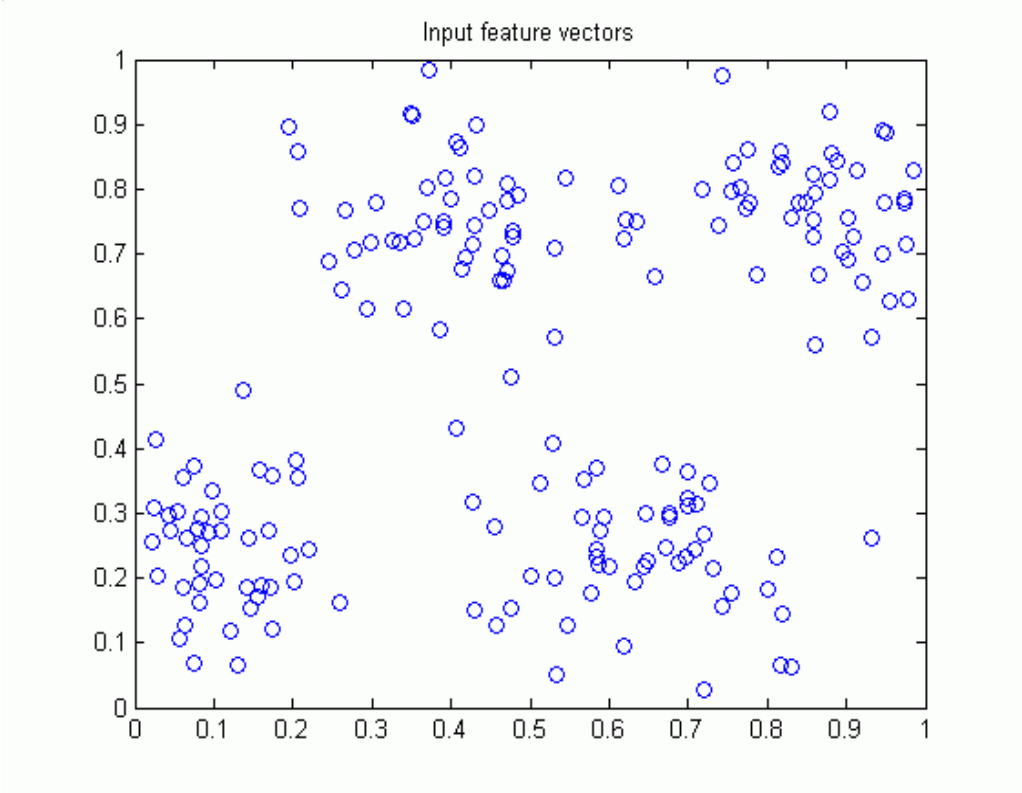

Figure 10a.

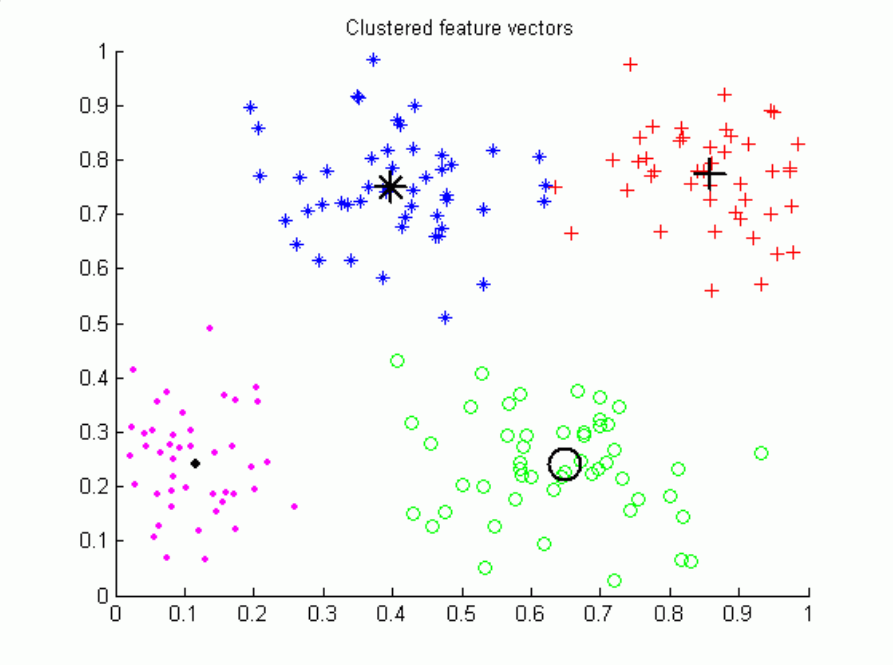

Figure 10b.

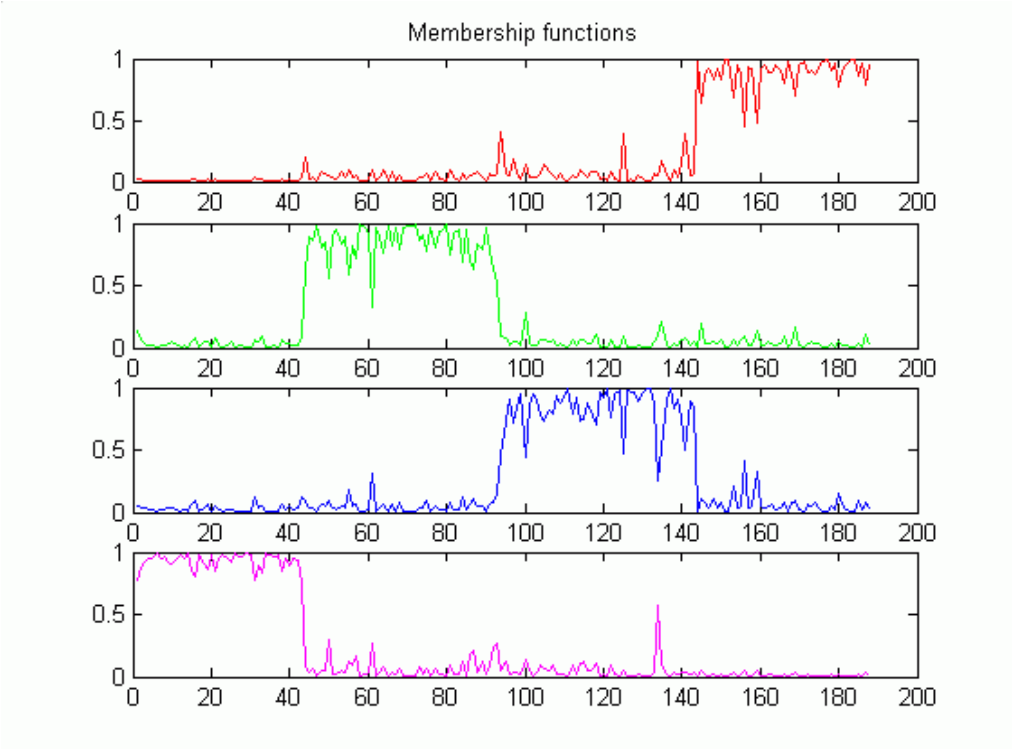

Figure 10c. Rows Of U (Membership Functions).

Objective Function:

Objective function J_m to be minimized in FCM,

$$\min_{(u, v)} \{ J_m (U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}{}^{m} D_{ik}{}^{2} \}$$

m -->Degree of fuzzification or fuzzy coefficient, m>=1

Constraint, $\sum_{i=1}^{c} u_{ik} = 1, \forall k$

Distance, $D_{ik}{}^{2} = ||X_k - V_i||^2_A$

A norm: $||x||_A = \sqrt{(x, x)}_A = \sqrt{x^T A x}$

Minimizing J_m (as this is our minimization problem): Zeroing the gradient of J_m with respect to V .

$$u_{ik} = [\sum_{j=1}^{c} (D_{ij} / D_{jk})^{2/(m-1)}]^{-1}, \forall i,k \ U$$

Zeroing the gradient of J_m with respect to U:

$$V_i = [\sum_{k=1}^{n} (u_{ik}^{m}) (x_k) / \sum_{k=1}^{n} (u_{ik}^{m})] , \forall i,k \ U$$

3.2. Fuzzy C-Means Clustering Algorithm

Number of clusters $1 < c < n$

Terminating Criterion

Maximum number of iterations (generally 100)

Minimal change in cluster centers

Weighting exponent (Fuzziness degree)

m=1: crisp

m=2: Typical

Guess initial c cluster centers

Algorithm:

Alternating Optimization (AO)

$i=0$

REPEAT

$i=i+1$

Update U_t

Update V_t

UNTIL TERMINATION
Output (U_t, V_t)

Limitation 1: Gets stuck at local optima depending on the choice of the initial cluster centers

- Application of better optimization methods like
- GAs, SA, PSO, DE

Limitation 2: Requires K to be specified

- Iterative Algorithms
- Variable String Length GAs, SA, PSO, DE …
- Computing cluster validity indices

3.3. Fuzzy C-Means Clustering Algorithm

Representation:

Cluster centers encoded in the chromosomes
For a d-dimensional space
length of chromosome = $d * K$

$\{ \underbrace{(v_{11}, v_{12}, \ldots, v_{1d})}_{\text{Center 1}} \underbrace{(v_{21}, v_{22}, \ldots, v_{2d})}_{\text{Center 2}} \ldots \underbrace{(v_{c1}, v_{c2}, \ldots, v_{cd})}_{\text{Center } c} \}$

Example 1:

Let $d=2$ and $K=3$, i.e., two-dimensional space, number of clusters = 3
Chromosome ➔ {(51.6 72.3) (18.3 15.7) (29.1 32.2)}
represents 3 cluster centers
(51.6, 72.3), (18.3, 15.7) and (29.1, 32.2).

Fitness computation:

This consists of three phases.

Phase 1: Cluster assignment
each point is assigned to the nearest cluster center.
All ties are resolved arbitrarily.

Phase 2: The cluster centers encoded in the chromosome are replaced by the mean points
of the respective clusters.
$v_i^* = (1/n_i)$ S x_j^i for $i=1, 2, \ldots, c$
v_i replaced by v_i^* in the chromosome

Phase 3: fitness = 1/(clustering metric J)
Compute J = S S $d^2(x_k^j, v_j)$, $j=1,2,\ldots c$ and $k=1, 2, \ldots, n_j$.
Maximization of fitness leads to minimization of J

Genetic Operations:

Crossover:

Single point crossover with a fixed crossover probability.

- For chromosomes of length c
  - a random integer p is generated in the range $[1, c]$

- portions of the chromosomes lying to the right of *p* are exchanged to produce two offspring.

Mutation:
Since we are considering floating point representation, we use the following mutation. A number *d* in the range [0, 1] is generated with uniform distribution. If the value at a gene position is v, after mutation it becomes

$v=v \pm 2 * d * v$, *if* $v \neq 0$,
$v=v \pm 2 * d$ *if* $v=0$.

Termination:

- GA clustering is run for a fixed number of generations
- Elitism incorporated
- best string (one with the lowest *J*) is taken as the solution of the clustering problem.

Result :
(c=5, n=250, d=2, iter=100, Pop = 20, $Prob_{crossover}$ *=0.8,* $Prob_{mutation}$ *=0.01)*, Figure 11a-b.

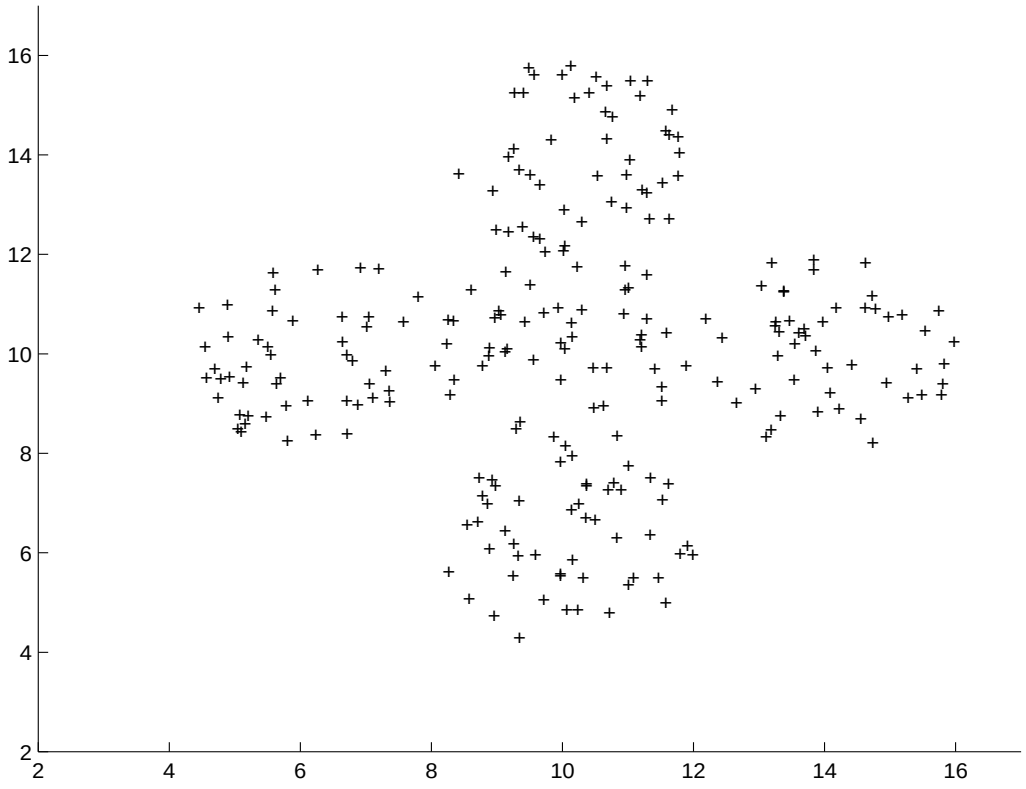

Figure 11a.

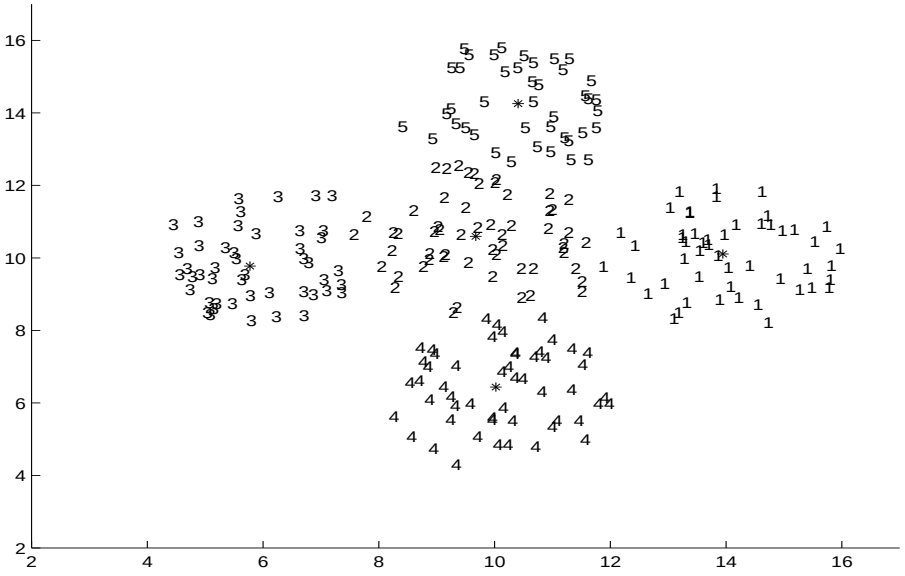

Figure11b.

Clustering for unknown *k*:

Limitation 2 of K-means

- Knowledge about the number of clusters K
- In most 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 the ***number of clusters*** and provide the ***appropriate clustering***.

3.4. Variable String Length GAs (VGA) for Automatic Clustering Algorithm

Each chromosome encodes different number of clusters [15],

- Chromosome $i \rightarrow c_i$ clusters

Representation:

- For a *d*-dimensional space, chromosome length = $d * c_i$
- $c_i = rand()$ mod $K^* + 2$
  - K* is a soft estimation of the upper bound of the number of cluster and rand() return an integer.

Fitness computation:

- **Phase 1**: assign each point to its nearest cluster center.
  - All ties are resolved arbitrarily.
- **Phase 2**: Compute the new centroids, and replace them in the chromosome
- **Phase 3:** Use cluster validity index as the fitness criterion
  - The K-means metric *J* cannot be used if the number of clusters is variable
    - Will be minimum (=0) when each point constitutes a cluster

Validity indices:

Davies-Bouldin (DB) index

DB-index is the ratio of the within cluster scatter to the between cluster separation.

Within cluster scatter is defined as

$S_i = (1/n_i)$ S $||x_k^i - v_i||$

x_k^i → *kth* point in *ith* cluster

Between cluster separation is defined as

$d_{ij} = ||v_i - v_j||$

Compute $R_i = \max \{(S_i + S_j)/d_{ij}\}$ over all i,j, $i \neq j$

DB index is then defined as

$DB = (1/K)$ S R_i, for $i = 1$ to K.

Objective is to minimize *DB* index for proper clustering.

Crossover:

- Probabilistic operation
- Cluster centers are considered to be indivisible
  - i.e., the crossover can only lie in between two clusters centers.
- Parent chromosomes P_1 and P_2 have c_1 and c_2 cluster centers respectively
- C1, crossover point in P1
  - $C_1 = rand()$ mod c_1
- C2, crossover point in P2
  - LB <= C2 <= UB
  - LB(C2) = min[2, max[0,2-(c_1-C1)]
  - UB(C2) = [C2 – max[0,2-C1]]
- Therefore, C2 is given by C2 = LB(C2) + rand() mod (UB(C2) – LB(C2))

Mutation:

- probabilistic in nature
- Generate d in the range [0,1] with uniform distribution
- If the value at a gene position is v, after mutation it becomes
  - $(1 \pm 2d) * v$, when $v \neq 0$
  - $\pm 2d$, when $v = 0$.
  - '+' or '-' sign occurs with equal probability.

Fitness computed by fuzzy cluster validity index:

- Xie-Beni Index
  - $XB = (J_m)/(n * d^2_{min})$
  - J_m → FCM measure
  - d_{min} → minimum intercluster distance
  - n → number of points
  - Minimum value of XB corresponds to proper clustering

4. Result

Clustering Gene Expression Microarray Data:

A microarray (gene expression) data set is usually arranged in a ***G X C*** matrix ***M* of *G* genes** and ***C* conditions**, where each row represents a gene and each column represents an experimental condition (a time point).

Each cell M_{ij} of the matrix represents the expression level of the ***i* th** gene at the ***j* th** condition.

Table 1. Example of Gene Expression Data at Different Time Points.

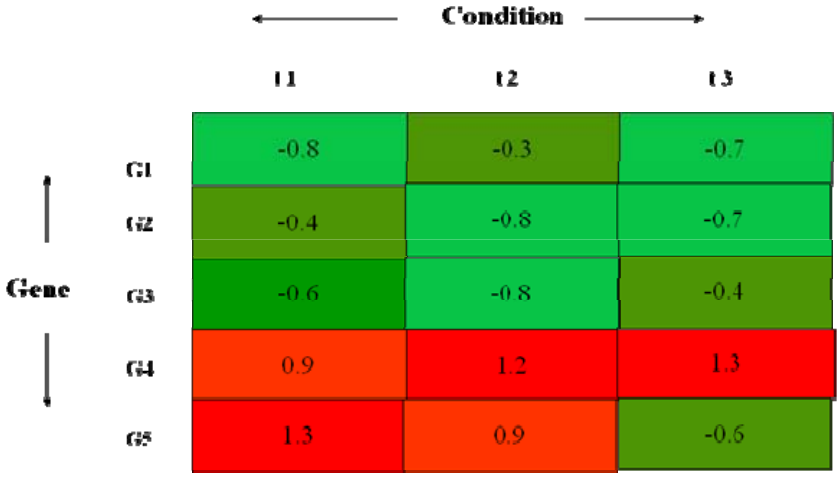

Table 2.

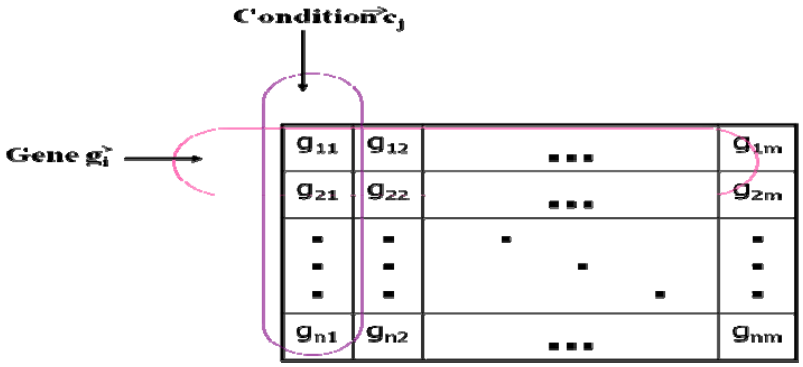

Table 3.

Data Sets	No. of genes	No. of time points	No. of clusters
Yeast Sporulation	6118	7	7
Human Fibroblasts Serum	517	13	10

The Sporulation data is filtered producing 474 prominently expressed genes are found. Both the data set is normalized so that each row has mean 0 and variance 1.

Table 4. Performance Comparison *Silhouette index (S)*

Algorithm	Data set	
	Sporulation	**Serum**
FCM	0.5879	0.3304
Average Linkage	0.5007	0.2977
Single objective GA minimizing XB index	0.5837	0.3532

Silhouette index (S) = (b-a)/max(a,b)
a and b: average distance of a point from the other points of the cluster to which it is assigned and the minimum of the average distances of the point from the points of the other clusters.

5. Conclusion

We have compared k-Means with Fuzzy C-Means, but, k-Means is not considered in our Performance Comparison table, because k-Means is not giving proper expected by stuck into local optima.

In the next phase, we will compare Multiobjective problems with the given result here in this paper. We actually, interested on Multiobjective problems, as cancer cells can not be identified by any single feature. And we will cluster or extract the cancer cells by using Multiobjective Clustering technique.

Acknowledgement

This work was supported by the Security Engineering Research Center, granted by the Korea Ministry of Knowledge Economy. And this work has successfully completed by the active support of Prof. Tai-hoon Kim, Hannam University, Republic of Korea.

References

[1] Azuaje, F., Dubitzky, W., Black, N., and Adamson, K., "Discovering Relevance Knowledge in Data: A Growing Cell Structures Approach," IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 30, No. 3, June 2000, pp. 448.
[2] M.N.M Sap and Ehsan Mohebi, "Hybrid Self Organizing Map for Overlapping Clusters", IJSIP, SERSC, Korea, Vol. 1, No. 1, December 2008, pp. 11-20.
[3] Guojun Gan, Chaoqun Ma and Jianhong Wo, "Data Clustering: Theory, Algorithms and Applications", ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia, ASA, Alexandria, VA, 2007.
[4] B. Everitt, "Cluster analysis", 3rd edition. New York, Toronto: Halsted Press, 1993.
[5] J. Carmichael, J. A. George and R. Julius, "Finding natural clusters", Systematic Zoology, Vol. 17, Issue 2, 1968, pp. 144–150.
[6] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, Issue 5, 2002, pp. 603– 619.
[7] M. Berry and G. Linoff, "Mastering Data Mining", John Wiley and Sons, New York, 2000.
[8] M. Fashing and C. Tomasi, "Mean shift is a bound optimization", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, Issue 3, 2005, pp. 471–474.

[9] J. Buhmann, "Data clustering and learning", The Handbook of Brain Theory and Neural Networks, Cambridge, MA: The MIT Press, 2003, pp. 308-312.
[10] S. Bandyopadhyay, C.A. Murthy and Sankar K. Pal, "Theoretical performance of genetic pattern classifier", Journal of the Franklin Institute, Vol. 336, 1999, pp. 387-422.
[11] S. Bandyopadhyay, A. Mukhopadhyay and U. Maulik, "An Improved Algorithm for Clustering Gene Expression Data", Bioionformatics, Oxford University Press, vol. 23, no. 21, 2007, pp. 2859-2865.
[12] S. Bandyopadhyay and S. K. Pal, "Pattern Classification with Genetic Algorithms: Incorporation of Chromosome Differentiation", Pattern Recognition Letters, Vol. 18, 1997, pp. 119-131.
[13] S. Bandyopadhyay and S. K. Pal, "Pixel Classification Using Variable String Genetic Algorithms with Chromosome Differentiation", IEEE Transactions on Geoscience and Remote Sensing, vol. 39, no. 2, 2001, pp. 303-308.
[14] S. Bandyopadhyay, U. Maulik and D. Roy, "Gene Identification: Classical and Computational Intelligence Approaches", IEEE Transactions on Systems, Man and Cybernetics, Part C, vol. 38, no. 1, 2008, pp. 55-68.
[15] Bandyopadhyay, S. Maulik, U., "Non-parametric Genetic Clustering: Comparison of Validity Indices", IEEE Transactions on Systems, Man and Cybernetics, Part-C, vol. 31, no. 1, 2001, pp. 120-125.