A Review of Clustering Techniques with FBPN

Sasirekha .J[#1], Dr. Savithri .V
[#]Research Scholar, Department of computer Science, Mother Teresa Women's University
Kodaikanal, India

***Abstract*— This study predicts the best supervised learning method of clustering techniques in fuzzy back propagation network(FBPN). Image processing algorithms are used to extract the information and patterns derived by process. Classification are done using predictive model of fuzzy technique of back propagation algorithm The values of the features are evaluated by FBPN algorithm.*

Keywords— Clustering, Fuzzy, Back Propagation, Neural Networks

I.INTRODUCTION

Levinski, et al., (2009), describes the approach for correcting the segmentation errors in 3D modeling space, implementation, principles of the proposed 3D modeling space tool and illustrates its application. Paragios, et al., (2003), introduces a knowledge based constraints, able to change the topology, capture local deformations, surface to follow global shape consistency while preserving the ability to capture using implicit function. Suri, et al., (2002), an attempt to explore geometric methods, their implementation and integration of regularizers to improve robustness of independent propagating curves/surfaces. Yuksel, et al., (2006), reveals the 100% classification accuracy of carotid artery Doppler signals using complex-values artificial neural network. Wendelhag, et al., (1991, 1997) results shows variations secondary to subjective parameters when manual measurement methods are employed. A thorough computerized system is necessary to evaluate the pattern recognition using clustering techniques. Our proposed method acts as a tool to predict the same patterns in data effectively and efficiently with less time and less memory allocation.

II.CLUSTERING

Clustering is a technique for partitioning a group of images into meaningful disjoint subgroups. Images that are similar to each other group themselves into a single cluster. All the images in a subgroup are similar to each other. At the same time, the images across the clusters are different. Cluster analysis is different from classification. Clustering is an example of unsupervised learning where there is no idea about the classes or clusters prior to clustering. Some of the important topics to be discussed with respect to clustering algorithms are as follows:

1. Method for finding the similarities and dissimilarities of the images
2. Categorization of clustering algorithms
3. Evaluation of clustering algorithms

Hierarchical clustering

Hierarchical clustering techniques are based on the use of a proximity matrix indicating the similarity between every pair of data points to be clustered. The end result is a tree of clusters representing the nested group of patterns and similarity levels at which groupings change. The resulting clusters are always produced as the internal nodes of the tree, while the root node is reserved for the entire dataset and leaf nodes are for individual data samples. The clustering methods differ in regard to the rules by which two small clusters are merged or a large cluster is split.
Agglomerative methods employ the following procedure:

1. Create a separate cluster for every data instance.
2. Repeat the following steps till a single cluster is obtained:
3. Choose a cluster formed by one of the step 2 results as final, if no more merging is possible.
   (a) Determine the two most similar clusters using similarity measures.

   (b) Merge the two clusters into a single cluster.

Pixel	X	Y
1	4	4
2	8	3
3	7	8
4	12	3

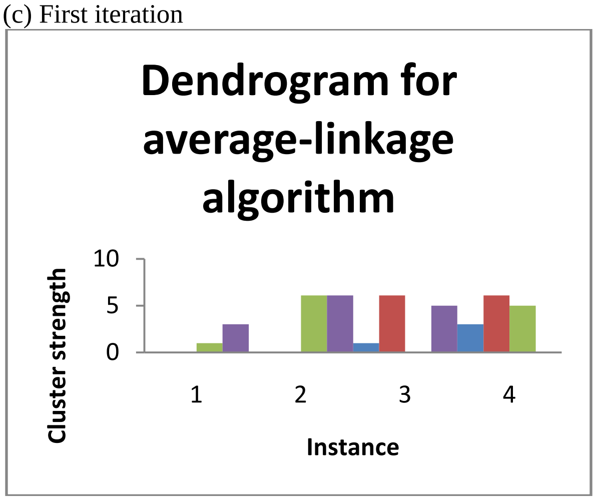
Fig.1(a) Euclid distance

The advantages of hierarchical methods include the fact that there is no need for vector representation for each object. These algorithms are easy to understand and interpret, and are intuitive and simple. Some of the popular algorithms are as follows:

Single-linkage algorithm A single-linkage algorithm is an agglomerative algorithm that takes a single instance and merges it with a cluster with which it is closer. This process is continued till no more merging is possible. Consider the sample image is shown in Fig(a).The Euclidean distance can be calculated between the pixel values for deciding the merging process. The Euclidean distance is shown in Fig(b).The minimum is 4.0, which is the distance between 2 and 4. Therefore, these two instances are merged in the next step as shown in Fig(c).

Distance (P+Q, R)= min(Distance(P, R), Distance(Q, R)). Accordingly, the distance between {1} and {2,4} is the minimum of the distance between {1,2} and {1,4}. Here the minimum is 4.1 Therefore, the resulting cluster is {1, 2, 4} and {3}. There is no point in performing the next iteration as it results in the merging of all the instances. The process therefore ends with two clusters.

(c) First iteration



Complete-linkage algorithm This algorithm is similar to the single-linkage algorithm in most aspects. However, the difference lies in the calculation of the distance between the instance and the cluster.In complete-linkage algorithm, the distance is Distance(P+Q,R)=max {Distance(P,Q), Distance(Q,R)}. The data in Fig(a) is taken as input and the complete linkage algorithm is applied. The initial Euclidean distance is shown in Fig(a)

The minimum is 4.0, which is the distance between 2 and 4.

The distance (P+Q, R) can be calculated as max (Distance (P, R), Distance (Q, R)). Accordingly, the distance between {1} and {2, 4} is the maximum of the distance between {1, 2} and {1, 4}. Here, the maximum is 8.1. Therefore, the resulting cluster is {1, 3} and {2, 4}. There is no point in performing the next iteration as it results in the merging of all the instances. Therefore, the process ends with two clusters. The results of this process can be shown visually as a dendrogram.

Average-linking algorithm The average-linking algorithm is also similar to single-linkage and complete-linkage algorithms. However, the distance between the instance and the cluster is calculated as the average of the individual distances, that is, the distance is

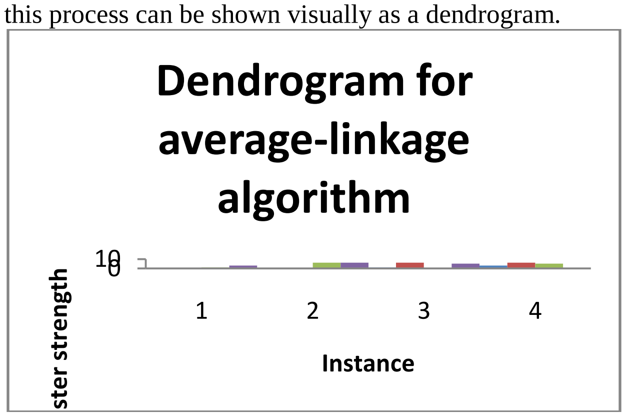
$$D_{average-linkage} = \frac{1}{n_i n_j} \sum_{a \in C_i, b \in C_j} d(a,b)$$

where d(a, b) is the distance between objects a and b ($a \in C_i$, $b \in C_j$), n_i and n_j are the number of objects in the clusters C_i and C_j respectively.

For the original data in Fig(a), the initial Euclid distance is shown in Fig(a).

The minimum is 4.0, which is the distance between 2 and 4. These two instances are merged in the next step as shown in

The distance between {1} and {2, 4} is the average of Distance (P,R) and Distance(Q,R). The results of this process can be shown visually as a dendrogram.



The advantages of the hierarchical algorithms are that they normally yield the correct number of clusters, are helpful in identifying the outliers, and are easy to understand.

Divisive clustering begins with the entire dataset in the same cluster, followed by iterative splitting of the dataset until the single-point clusters are attained on leaf nodes. It follows a reverse clustering strategy against agglomerative clustering. On each node, the divisive algorithm conducts a full search for all possible pairs of clusters for data samples on the node.

Agglomerative hierarchical clustering: This bottom-up strategy starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied. Most hierarchical clustering methods belong to this category. They differ only in their definition of inter cluster similarity.

Divisive hierarchical clustering: This top-down strategy does the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster. It subdivides the cluster into smaller and smaller pieces, until each object forms a cluster on its own or until it satisfies certain termination conditions, such as a desired number of clusters is obtained or the diameter of each cluster is within a certain threshold.

In either agglomerative or divisive hierarchical clustering, the user can specify the desired number of clusters as a termination condition.

A tree structure called a **dendrogram** is commonly used to represent the process of hierarchical clustering. It shows how objects are grouped together step by step. Figure shows a dendrogram for the five objects presented in Figure 7.6, where l =0 shows the five objects as singleton clusters at level 0. At l=1, objects a and b are grouped together to form the first cluster, and they stay together at all subsequent levels. We can also use a vertical axis to show the similarity scale between clusters. For example, when the similarity of two groups of objects, {a, b} and{c, d, e}, is roughly 0.16, they are merged together to form a single cluster.

Four widely used measures for distance between clusters are as follows, where | p-p'| is the distance between two objects or points, p and p'; m_i is the mean for cluster, C_i and n_i is the number of objects in C_i.

Minimum distance: $d_{min}(C_i, C_j) = min_{p \in C_i, p' \in C_j} |p-p'|$

Maximum distance: $d_{max} (C_i, C_j) = max_{p \in C_i, p' \in C_j} |p-p'|$

Mean distance: $d_{mean} (C_i, C_j) = | m_i - m_j |$

Average distance: $d_{avg} (C_i, C_j) = \frac{1}{n_i n_j} \sum_{p \in C_i} \sum_{p' \in C_j} |p-p'|$

When an algorithm uses the minimum distance, $d_{min}(C_i, C_j)$, to measure the distance between clusters, it is sometimes called a nearest-neighbor clustering algorithm. Moreover, if the clustering process is terminated when the distance between nearest clusters exceeds an arbitrary threshold, it is called a single-linkage algorithm. If we view the data points as nodes of a graph, with edges forming a path between the nodes in a cluster, then the merging of two clusters, C_i and C_j, corresponds to adding an edge between the nearest pair of nodes in C_i and C_j. Because edges linking clusters always go between distinct clusters, the resulting graph will generate a tree. Thus, an agglomerative hierarchical clustering algorithm that uses the minimum distance measure is also called a minimal spanning tree algorithm.

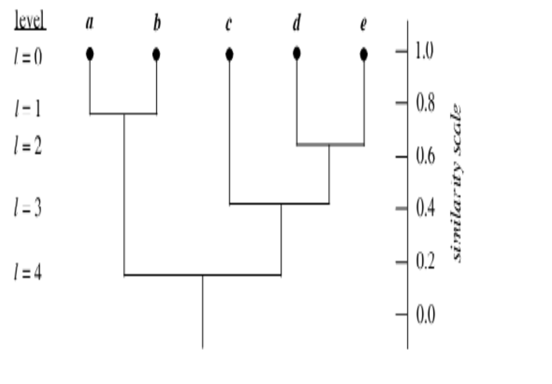


Figure Dendrogram representation for hierarchical clustering of data objects {a, b, c, d, e}.

When an algorithm uses the maximum distance, $d_{max}(C_i, C_j)$, to measure the distance between clusters, it is sometimes called a farthest-neighbor clustering algorithm. If the clustering process is terminated when the maximum distance between nearest clusters exceeds an arbitrary threshold, it is

called a complete-linkage algorithm. By viewing data points as nodes of a graph, with edges linking nodes, we can think of each cluster as a complete subgraph, that is, with edges connecting all of the nodes in the clusters. The distance between two clusters is determined by the most distant nodes in the two clusters. Farthest-neighbour algorithms tend to minimize the increase in diameter of the clusters at each iteration as little as possible. If the true clusters are rather compact and approximately equal in size, the method will produce high-quality clusters. Otherwise, the clusters produced can be meaningless.

The above minimum and maximum measures represent two extremes in measuring the distance between clusters. They tend to be overly sensitive to outliers or noisy data. The use of mean or average distance is a compromise between the minimum and maximum distances and overcomes the outlier sensitivity problem. Whereas the mean distance is the simplest to compute, the average distance is advantageous in that it can handle categoric as well as numeric data. The computation of the mean vector for categoric data can be difficult or impossible to define.

The hierarchical clustering method, though simple, often encounters difficulties regarding the selection of merge or split points. Such a decision is critical because once a group of objects is merged or split, the process at the next step will operate on the newly generated clusters. It will neither undo what was done previously nor perform object swapping between clusters. Thus merge or split decisions, if not well chosen at some step, may lead to low-quality clusters. Moreover, the method does not scale well, because each decision to merge or split requires the examination and evaluation of a good number of objects or clusters.

III. Fuzzy Back Propagation

Fuzzy BP is a hybrid architecture, namely Lee and Lu's Fuzzy BP network (Lee and Lu, 1994). Fuzzy BP is a hybrid architecture which maps fuzzy inputs to crisp outputs. The fuzzy neurons in the model make use of LR-type fuzzy numbers. Besides, triangular type of LR-type fuzzy numbers have been used for simplification of architecture and reduction of computational load. The LR-type fuzzy numbers and their operations required by the fuzzy BP architecture are fist presented. The structure of a fuzzy neuron, the architecture of fuzzy BP, its learning mechanism, and algorithms are elaborated next.

LR-TYPE FUZZY NUMBERS

The LR-type fuzzy numbers are special type of representations for fuzzy numbers, proposed by Dubois and Prade(1979). They introduced functions called L (and R) which map $R^+ \rightarrow [0,1]$ and are decreasing shape functions if

L(0)=1, L(x)<0, $\forall$ x<1,

L(1)=0 or [L(x)>0 $\forall x. and\ L(\infty) = 0$]

Definition

A fuzzy number M̃ is of LR -type if there exist reference functions L (for left), R(for right) and scalars, $\alpha > 0, \beta > 0$ with

$$\mu_{\tilde{M}}(X) = \begin{cases} L\left(\frac{m-x}{\alpha}\right) for x \le m \\ R\left(\frac{x-m}{\beta}\right)\ for\ x \ge m \end{cases}$$

Here, m, called the mean value of M̃, is a real number and $\alpha\ and\ \beta$ are called the left and right spread, respectively.

Here, $\mu_{\tilde{M}}$ is the membership function of zy number M̃. An LR-type fuzzy number M can expressed as $(m,\alpha,\beta)_{LR}$. If $\alpha\ and\ \beta$ both zero the LR-type function indicates a crisp value. For L(z), different functions may be chosen. Dubois and Prade (Dubois and Prade, 1988) mention L(x) = max(0,1-x^p) p>0, L(x) = exp(-x), L(x) = exp(-x^2) to list a few, thus suggestive of a wide scope of L(z). However, the choice of the L and R functions is specific to the problem in hand.

In the case of trapezoidal fuzzy number the LR-type flat fuzzy numbers defined below are made use of

$$\mu_{\tilde{M}}(X) = \begin{cases} L\left(\frac{m_1-x}{\alpha}\right) for x \le m_1, \alpha > 0 \\ R\left(\frac{x-m_2}{\beta}\right)\ for\ x \ge m_2, \beta > 0 \\ 1 \qquad otherwise \end{cases}$$

Briefly, the above equation is represente by the quadruple $(m_1, m_2, \alpha, \beta)_{LR}$. A triangular LR-type fuzzy number can also be represented by the quadruple (m,m,α,β).

FUZZY NEURON

The fuzzy neuron is the basic element of the fuzzy BP model. Given the input vector $\tilde{I} = (\tilde{I}_0, \tilde{I}_1, \ldots, \tilde{I}_l)$ and weight vector $\tilde{w} = (\tilde{w}_0, \tilde{w}_1, \ldots, \tilde{w}_l)$, the fuzzy neuron computes the crisp output O given by

O = f(NET)=f(CE($\sum_{i=0}^{l} \tilde{w}.\tilde{I}$))

where, $\tilde{I}_0$=(1,0,0) is the bias. Here, the fuzzy weighted summation

is first computed and NET=CE(nẽt) is computed next. The function CE is the centroid of the triangular fuzzy number and can be treated as a defuzzification operation which maps fuzzy weighted summation value to a crisp value. Thus, if is the fuzzy weighted summation then the function CE is given by

CE(nẽt) = $net_m + \frac{1}{3}(net_\beta - net_\alpha)$ = NET

The function f is the sigmoidal function which performs nonlinear mapping between the input and output .f is defined as

f(NET) = $\frac{1}{1+\exp(-NET)}$

This is the final computation to obtain the crisp output value O.

In the fuzzy neuron, both input vector Ĩ and weight vector are represented by triangular LR-type fuzzy numbers. Thus, for $\tilde{I} = (\tilde{I}_0, \tilde{I}_1, \ldots, \tilde{I}_l)$ the input component vector $\tilde{I}_i$ is represented by the LR-type fuzzy number$(I_{mi}, I_{\alpha i}, I_{\beta i})$.

To validate the desired output training, testing and validation is done by using confusion matrix network. Confusion matrix is applied for training, validation, testing ROC (Region of characters) using true positive rate vs false positive rate. Accuracy and no number of incorrect responses helps to predict the Fuzzy Back propagation algorithm is best.

IV CONCLUSION

A real time measurement made and classified by fuzzy back propagation network which produces more accurate results than any other method.

It is believed that this will provide a faster solution and effective way for classification of patterns in image processing

Thus concluded that prediction of patterns can be detected effectively and accurately with fuzzy back propagation algorithm and the same can be used as the second observer apart from practitioner's opinion.

References

[1] Cai, W., Chen, S., and Zhang, D., 2007, Fast and robust fuzzy *c*-means clustering algorithms incorporating local information for image segmentation, Pattern Recognition, Vol.40, pp.825–838.

[2] Da-Chuan Cehng, Christian Billich, Shing-Hong Liu, Horst Brunner, Yi-Chen Qiu, Yu-Lin Shen, Hans J Brambs, Arno Schmidt-Trucksass and Uwe HW Schutz, 2011, Automatic detection of the carotid artery boundary on cross-sectional MR image sequences using a circle model guided dynamic-programming, Biomedical engineering Vol.10, pp.1-17.

[3] Levinski, K., Sourin, A., and Zagorodnov, V., 2009, Interactive surface-guided segmentation of brain MRI data**,** Computers in Biology and Medicine, Vol.39, Issue 12, pp.1153–1160.

[4] Markos, G., Tsipouras, Themis, P., Exarchos, Dimitrios, I., Fotiadis, Anna, P., Kotsia, Konstantinos, V., Vakalis, Katerina, K., Naka and Lampros K., Michalis, 2008, Automated Diagnosis of coronary artery disease based on data mining and fuzzy modeling, IEEE transactions on Information technology in biomedicine, Vol.12, Issue 4, pp.447-456.

[5] Paragios, N., 2003, A level set approach for shape-driven segmentation and tracking of left ventricle**,** IEEE Transactions on Medical Imaging, Vol.22, pp.773–776.

[6] Santhiyakumari, N., and Madheswaran, M., 2010, Intelligent medical decision system for identifying ultrasound carotid artery images with vascular disease, International journal of Computer Application, Vol.1, Issue 13, pp.32-39.

[7] Suri, J.S., 2001, Two-dimensional fast magnetic resonance brain segmentation, IEEE Engineering in Medicine and Biology, Vol.20, pp.84-95.

[8] Wendelhag, I., Gustavsson, T., Suurkula, M., Berglund, G., and Wikstrand, J., 1991, Ultrasound measurement of wall thickness in the carotid artery: fundamental principles and description of a computerized analysing system, Clin Physiol., Vol.11, pp.565-577.

[9] Wendelhag, I., Liang, Q., Gustavsson, T., and Wikstrand, J., 1997, A new automated computerized analyzing system simplifies readings and reduces the variability in ultrasound measurement of intima-media thickness, Stroke, Vol.28, pp.2195-2200.

[10] Yuksel Ozbay and Murat Ceylan, 2006, Effects of window types on classification of carotid artery Doppler signals in the early phase of atherosclerosis using complex-valued artificial neural network, Ultrasound in Medicine and Biology, Vol.37, Issue 3, pp.287-295.

[11] Li, B.N., Chui, C.K., Ong, S.H., and Chang, 2009, Integrating FCM and level sets for liver tumor segmentation, Proceedings of the 13th International Conference on Biomedical Engineering, (ICBME 2008), IFMBE Proceedings 23, pp.202–205.

[12] Lei, W.K., Li, B.N., Dong, M.C., and Vai, M.I., 2007, AFC-ECG: an adaptive fuzzy ECG classifier, in: Proceedings of the 11th World Congress on Soft Computing in Industrial Applications (WSC11), Advances in Soft Computing, Vol.39, pp.189–199.