International Journal of Electronics Signals and Systems

Volume 1 | Issue 2 Article 6

September 2011

Neural Network Based Fuzzy C-MEANS Clustering Algorithm

suneetha chittineni
Rvr&jcce, chittineni_es@yahoo.com

Raveendra Babu Bhogapathi Dr.
Director-Operations, DELTA Technology & Management Services Pvt. Ltd., Hyderabad, India,
rbhogapathi@yahoo.com

Available online at www.interscience.in

Neural Network Based Fuzzy C-MEANS Clustering Algorithm

Suneetha Chittineni [1], Dr. Raveendra Babu Bhogapathi [2]
[1]Department of Computer Applications, R.V.R. & J.C. College of Engineering, Guntur, India.
[2]Director-Operations, DELTA Technology & Management Services Pvt. Ltd., Hyderabad, India.
E-mail : chittineni_es@yahoo.com, rbhogapathi@yahoo.com

***Abstract* -** In this paper, fuzzy c-means algorithm uses neural network algorithm is presented. In pattern recognition, fuzzy clustering algorithms have demonstrated advantage over crisp clustering algorithms to group the high dimensional data into clusters. The proposed work involves two steps. First, a recently developed and Enhanced Kmeans Fast Leaning Artificial Neural Network (KFLANN) frame work is used to determine cluster centers. Secondly, Fuzzy C-means uses these cluster centers to generate fuzzy membership functions. Enhanced K-means Fast Learning Artificial Neural Network (KFLANN) is an algorithm which produces consistent classification of the vectors in to the same clusters regardless of the data presentation sequence. Experiments are conducted on two artificial data sets Iris and New Thyroid. The result shows that Enhanced KFLANN is faster to generate consistent cluster centers and utilizes these for elicitation of efficient fuzzy memberships.

Keywords : *Pattern Recognition, Fuzzy Clustering, Crisp Clustering, Fast Learning Artificial Neural Network.*

I. INTRODUCTION

Pattern recognition is a field concerned with machine recognition of meaningful regularities in noisy or complex environments. In simpler words, pattern recognition is the search for structures in data. Clustering algorithms can be divided into two classes: crisp and soft (or fuzzy). In crisp clustering, each data object has to be assigned into exactly one cluster. However, the performance of hard clustering becomes worse if some of data clusters are overlapped with each other. Fuzzy clustering approaches can avoid the shortcoming of hard clustering.

The most popular fuzzy clustering technique is fuzzy c-means(FCM).This method was developed by Dunn[1] in 1973 and improved by Bezdek[2] in 1981 and is frequently used in pattern recognition. The FCM method allows each data point belongs to each cluster with a certain degree of membership or belonging in the interval [0, 1] and thus has better ability to describe neutral information or to tolerate disturbance resulting from noisy data. The clustering result of the above clustering methods is usually evaluated by an objective function during the clustering process. The objective function is defined to minimize the total sum of distances between data items and their own centers. Although FCM is an effective algorithm for clustering problems, it sometimes finds worse results because of its random initialization. Hence we are trying to avoid random initialization using neural network algorithm (KFLANN) which produces consistent clusters to improve FCM and also hope that the proposed approach has higher probability to find global optimum. This paper proposes an approach utilizing KFLANN to fuzzy clustering problems.

Many variations of fast learning artificial neural network algorithms have been proposed. A fast learning artificial neural network (FLANN) models was first developed by Tay and Evans [3] to solve a set of problems in the area of pattern classification. FLANN [3] [4] was designed with concepts found in ART but imposed the Winner Take All (WTA) property within the algorithm. Further improvement was done to take in numerical continuous value in FLANN II [4]. The original FLANN II was restricted by its sensitivity to the pattern sequence. The original FLANN II was restricted by its sensitivity to the pattern sequence .This was later overcome by the inclusion of k-means calculations, which served to remove inconsistent cluster formations [6]. The later improvement on KFLANN [7] includes data point reshuffling which resolves the data sequence sensitivity that creates stable clusters. Clusters are said to be stable if the cluster formation is complete after some iterations and the cluster centroids remain consistent.

The remaining of this paper is organized as follows. Section II briefly introduces KFLANN architecture and modified algorithm. Section III describes the fuzzy clustering problems and fuzzy c-means algorithm. Section IV describes the proposed problem of using modified KFLANN to FCM. Section V reports the results of experiments. In the last section, we summarize the proposed approach

II. KFLANN ARCHITECTURE

The KFLANN architecture [6] is shown in Figure 1.KFLANN consists of a single input layer that integrates the source of the patterns. The output layer grows dynamically as new groups are formed during the clustering process. The weight connections between the input node and output node are the direct mapping of each element of input vectors. The dynamic formation of output nodes yield KFLANN is a self-organized clustering ANN model.

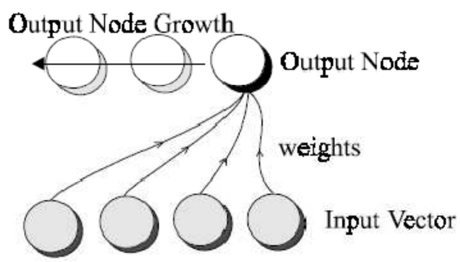

Figure.1. KFLANN Architecture

MODIFIED KFLANN ALGORITHM

The original KFLANN algorithm is described in [6].The segment (Step 4) of the original KFLANN algorithm is modified as follows:

Step 4 Determine the winner from all matched output nodes

Using the following criteria:

If same match is found

Winner =

$$\min \left[\sum_{i=0}^{n} (w_{ij} - x_i)^2 \right] \tag{1}$$

Else

$$\text{Winner} = \max \left(\frac{\sum_{i=0}^{n} \left[\delta_i^2 - (w_{ij} - x_i)^2 \right]}{n} \right) \tag{2}$$

Note: δ_i is the tolerance for i th feature of the input space , W_{ij} used to denote the weight connection of j th output to the i th input, Xi represent the i th feature.

III. FUZZY C-MEANS ALGORITHM

Fuzzy c-means clustering involves two processes: the calculation of cluster centers and the assignment of points to these centers using a form of Euclidian distance. This process is repeated until the cluster centers stabilize. The algorithm is similar to k-means clustering in many ways but incorporates fuzzy set's concepts of partial membership and forms Overlapping clusters to support it. It assigns membership value to the data items for the clusters within a range of 0 to 1.The algorithm needs a fuzzification parameter m in the range [1, n] which determines the degree of fuzziness in the clusters. When m reaches the value of 1 the algorithm works like a crisp partitioning algorithm and for larger values of m the Overlapping of clusters tends to be more. The algorithm calculates the membership value U with the formula,

$$U_{ij} = \frac{1}{\sum_{k=1}^{c} \left[\frac{|x_i - c_j|}{|x_i - c_k|} \right]^{\frac{2}{m-1}}} \tag{3}$$

Where

Uij: is the membership of xi in the jth cluster

m : is the fuzzification parameter

c : is the number of specified clusters

Also the algorithm imposes a restriction which says the sum of memberships of a data point in all the clusters must be equal to one. This constraint is represented by eq. (4).

$$\sum_{j=1}^{c} U_{ij} = 1 \tag{4}$$

The new cluster centers are calculated with the fuzzy membership values using (5).

$$C_j = \frac{\sum_{i=1}^{N} u_{ij}^{\ m} . x_i}{\sum_{i=1}^{N} u_{ij}^{\ m}} \tag{5}$$

This is a special form of weighted average. We modify the degree of fuzziness in xi's current membership and multiply this by xi. The product obtained is divided by the sum of the Fuzzified membership.

The FCM algorithm is given as follows

Step1.initialize c =number of clusters generated from KFLANN

Initialize m= fuzzification parameter

Initialize Cj Cluster centers generated from KFLANN

Step 2. Repeat

For i=1 to N: Update Uij applying (3)

For j=1 to c: Update Ci with (5) with current Uij

Until Cj estimate stabilize

The first loop of the algorithm calculates membership values for the data points in clusters and the second loop recalculates the cluster centers using these membership values. When the cluster center stabilizes (when there is no change) the algorithm ends.

IV. USING ENHANCED KFLANN TO FCM

The generation of fuzzy membership function via KFLANN is a two-step procedure. The first step generates the proper clusters. Then, the fuzzy membership function is generated according to the clusters generated in the first step.

The proposed work is explained as follows.

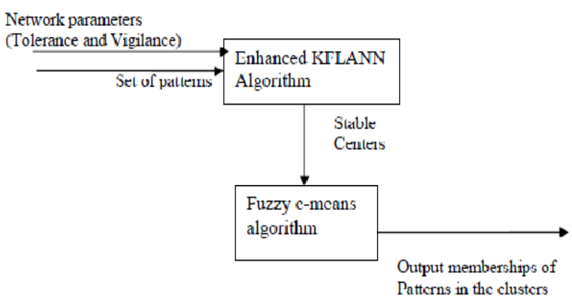

Figure 2. Architecture of proposed algorithm

5. EXPERIMENTS AND RESULTS

To conduct experiments two artificial data sets were used. The descriptions of the data sets are given as follows:

5.1. Iris Dataset

The data set contains 150 random samples of flowers from the Iris species: Setosa, Versicolor, and Virginica. From each species there are 50 observations with 4 attributes each in centimeters. The first class, Setosa, is linearly separable from the other 2, while the latter two are not linearly separable from each other. This means that Versicolor and Virginica intersect with each other and Enhanced K-FLANN is not able to 100% correctly classify the 3 iris species. The following are the 4 attributes of iris data:

- Sepal Length
- Sepal width
- Petal Length
- Petal width

5.2. New Thyroid Data set

The data set contains 215 random samples of thyroid disease. It has total 250 observations.

They fall into 3 classes: 1 . Normal, 2. Hyper , 3. Hypo. Among those 150 samples belong to class 1, 35 samples belong to class 2, 30 samples belong to class 3. It has 5 attributes.

5.3. KFLANN Parameters

It is necessary to initialize the parameters of the KFLANN. Vigilance and Tolerance.

Vigilance (ρ) is calculated by using eq. (6)

$$\rho = \frac{f}{n} \qquad 0.5 < \rho < 1.0 \tag{6}$$

Where 'f' is the Number of features required to be classified in a same clust ' n' is the Total number of features.

Tolerance setting for each feature is calculated by using binary search approach [6] shown by eq. (7).After a complete single epoch, analyses clustering performance. If the results are not up to expected accuracy, tolerance tuning for each feature is performed by using binary search approach [7] and the clustering process is repeated with the tuned tolerance.

Initially $$\delta_i = \frac{\max diff + \min diff}{2} \tag{7}$$

Where maxdiff is the maximum difference of feature value and mindiff is the minimum difference.

The following table shows initialized parameters and the obtained accuracy (%) of Enhanced KFLANN for the two artificial data sets.

5.4. Results of Enhanced KFLANN

Table 1. Network Parameters and Accuracy (%)

Data set	Vigilance (ρ)	Number of features	Tolerance Setting for each Feature	Number of iterations	Number of clusters	Accuracy(%)
Iris	0.75	4	0.39469,0.19396,0.33952,0.6043	100	3	96.66%
New Thyroid	0.8	5	25.375 ,7.8187, 3.1313 ,17.662, 17.881	100	3	92.05%

Table 2. Cluster Centers

Data set	Stable Cluster centers generated from KFLANN		
	Center 1	Center 2	Center 3
Iris	pattern 39	Pattern 70	Pattern 144
New Thyroid	Pattern 7	Pattern 181	Pattern 207

Table 3. Memberships of data patterns for Iris data set and New thyroid data set

IRIS DATA SET. NEW THYROID DATA SET.

S.No.	*Cluster1*	Cluster2	Cluster3	Cluster No.	S.No	Cluster1	Cluster2	Cluster3	Cluster No.
1	*2986.851*	2.746763	1.573316	1	1	139.3971	3.707536	1.382924	1
2	191.6685	2.885931	1.542557	1	2	64.26366	2.110468	1.95844	1
3	3858.889	2.754929	1.570463	1	3	9.422971	1.865716	2.794162	1
4	759.4002	2.8236	1.551529	1	4	51.33209	2.913649	1.569103	1
5	581.6533	2.711706	1.588539	1	5	35.07683	4.617363	1.324648	1
……	…………	….………	…………	…	7	1	0	0	1
38	276.5342	2.883787	1.539367	1	8	56.62557	2.054289	2.017945	1
39	1	0	0	1	……	………	…..……	………	…
……	…………	….………	…………	…	33	61.88661	2.335651	1.799548	1
50	2408.321	2.789512	1.559821	1	34	5.382907	69.32011	1.250311	2
51	1.228892	200.7489	5.5164	2	35	32.01507	2.024563	2.106013	1
52	1.226375	157.0981	5.610944	2	36	35.33623	2.07391	2.04282	1
53	1.119983	4105.509	9.355765	2	…….	…………	…………	………	………...
54	1.075715	97.18593	16.63996	2	149	25.26043	4.951693	1.318459	1
55	1.104747	271.9831	10.97229	2	150	18.29966	6.137635	1.278078	1
…….	………… .	…………	…………	…	151	5.577116	1.90528	3.38022	1
69	1.087675	26.16375	23.5922	2	152	42.71721	3.265431	1.491754	1
70	0	1	0	2	153	29.71523	2.919564	1.603	1
71	1.086382	44.29526	17.56296	2	154	3.269854	10.70055	1.664661	2
……..	…………	…………	…………	…	……. .	…………	…………	………	……
100	1.167795	284.964	7.133888	2	181	0	1	0	2
101	1.125828	9.625437	127.0094	3	182	9.39099	21.73079	1.179945	2
102	1.076179	14.1325	36045.36	3	…….	…………	…………	………	………...
103	1.063898	17.75034	268.5978	3	186	4.710488	1.435074	11.0036	3
104	1.081414	13.70487	431.3972	3	187	2.989129	1.676321	14.5117	3
105	1.08782	12.47977	1665.493	3	……	…………	…………	………	…………
……..	………… ..	…………	…………		206	3.704904	1.452513	24.02364	3
144	1.069566	15.45712	2887.841	3	207	0	0	1	3
145	0	0	1	3	208	2.754269	1.915336	8.70886	3
……..	…………	…………	…………		……	…………	…………	………	…………
149	1.063092	20.44114	95.90442	3	215	12.58883	4.470937	1.434931	1
150	1.068335	19.23415	83.52294	3					

VI. CONCLUSION

The Enhanced KFLANN consistently obtains the same centroids after epoch 3 and also achieves improved accuracy when compared with previous algorithms. The FCM algorithm usually requires randomized initial centers and the result depends on the assumed number of centers. To overcome the initial assumption, the FCM algorithm uses KFLANN output and generates efficient memberships. The KFLANN algorithm which completes the analysis of clustering performance after a single epoch to get the stable clusters as final output. The future work includes applying cluster validation techniques on the output of the both algorithms.

REFERENCES

[1] J. C. Dunn, A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, Journal of Cybernetics 3;1973: 32-57.

[2] Bezdek, J. C., Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, NY, 1981.

[3] L.P.Tay, D.J.Evans, "Fast Learning Artificial Neural Network Using Nearest Neighbors Recall", Neural Parallel and Scientific Computations, 2(1), 1994.

4] D.J.Evans, L.P.Tay, "Fast Learning Artificial Neural Network for continuous Input Applications", Kybernetes.24 (3), 1995.

[5] Kohonen, T., Self Organization and Associative Memory,3rd ed. Berlin: Springer-Verlag, (1989).

[6] Alex T.L.P., Sandeep Prakash, "K-Means Fast Learning Artificial Neural Network, An Alternate Network For Classification", ICANN, 2002.

[7] L. P. Wong, T. L. P. Alex, "Centroid Stability with K-Mean Fast Learning Artificial Neural Networks", International Joint Conference on Neural Network (IJCNN), 2003.