IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 2, September 2011
ISSN (Online): 1694-0814
www.IJCSI.org

A Fuzzy Neural Clustering approach for Fingerprint Recognition

Mahendra Kondekar[1], Uday Kulkarni[2] and Santosh Chowhan[3]

**[1] College of Computer Science and
Information Technology, Latur-413512
Maharshtra, India.**

**[2] SGGSIE&T, Nanded,
Maharastra, India.**

**[3] College of Computer Science and
Information Technology, Latur-413512
Maharshtra, India.**

Abstract

Fingerprint recognition is one of the most promising and evergreen biometric recognition technique used presently. In this paper we have proposed Extended Fuzzy Hyperline Segment Clustering Neural Network (EFHLSCNN) with its learning algorithm, which utilizes fuzzy sets as pattern clusters. In this extended version of paper we have used Manhattan distance for calculating distance of hyperline segments. The performance of EFHLSCNN when verified with fingerprint feature vectors, it is found superior than fuzzy hyperline segment clustering neural network (FHLSCNN) proposed by Kulkarni and Sontakke in terms of higher recognition rate and generalization.

Keywords: *Biometrics, Pattern Clustering, Fuzzy Neural Network, FingerCode.*

1. Introduction

Biometrics is the science of identifying individuals by a particular physical characteristic such as voice, eye, fingerprints, height, facial appearance, iris texture, or signature. Fingerprint based personal identification is routinely used in forensic laboratories and identification units around the world [1] and it has been accepted in the courts of law for nearly a century [2, 3]. Fingerprint features are permanent and fingerprints of an individual are unique [15]. Here in this paper we have used one of the most enthusiastic approaches to computer-based pattern recognition i.e. use of fuzzy neural networks for clustering feature patterns. They have been successfully used in many pattern recognition problems [4], [5], [6].

Cluster is a group of patterns having some common properties. Patterns can be grouped into clusters by some predefined criterion. As mentioned by Bezdek [7] the clusters can be formed according to some criterion like distance, angle, curvature, symmetry, connectivity, and intensity. Patterns which are similar are allocated to the same cluster, while the patterns which differ significantly are put in different clusters. Regardless of the clustering method the final result is always a partition of patterns in disconnected or overlapped clusters [10]. The choice of the proper grouping metric is only one aspect of the clustering problem. The fuzzy min-max (FMM) clustering and classification neural network algorithms [11], [12], with their representation of classes as hyperboxes in n-dimensional pattern space and their conceptual simplicity simple but powerful learning process, provided a natural basis for our paper. The derivatives of the original FMM can also be found in [13] and [14]. U. V. Kulkarni and T. R. Sontakke [8] also have proposed FHLSCNN.

In this paper they have used Euclidian distance for calculating distance between hyperline segments. In this extended version of FHLSCNN we have improved the recognition rate of patterns by using Manhattan distance metric. The patterns used for classification and clustering are of Poly U HRF Fingerprint database images of 320*240 sizes at 1200 dpi resolution. The feature extraction process is based on FilterBank based FingerCode feature extraction algorithm. In this algorithm they have used eight different values for with respect to the x-axis θ (0^0, 22.5^0, 45^0, 67.5^0, 90^0, 122.5^0, 135^0, and 157.5^0) . The normalized region of interest in a fingerprint image is convolved with each of these eight filters to produce a set of eight filtered features. These eight directional-sensitive filters capture most of the global ridge directionality information as well as the local ridge characteristics present in a fingerprint. The mean of each sector in each of the eight filtered features defines the components of FingerCode feature vector.

2. Topology of the EFHLSCNN

The EFHLSCNN consists of three layers as shown in Figure 1. The F_R layer accepts an unlabeled input pattern and consists of n processing elements, one for each dimension of the pattern. The F_E layer consists of m hyperline segments that are constructed during training and each node is characterized by the extended hyperline segment membership function as shown in Figure 2. One connection represents one end point for that dimension and the other connection represents another end point of that dimension, for a particular hyperline segment as shown.

The end points of hyperline segments are stored in V and W matrices. Each node of F_C layer represents a cluster and is constructed during training. The transfer function of F_C node performs the union of appropriate hyperline segments. The weights assigned between F_E and F_C layers are stored in the U matrix.

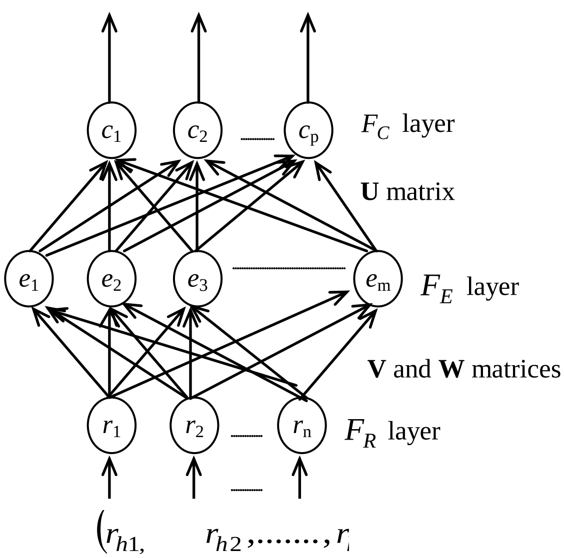

Fig. 1 Extended Fuzzy hyperline segment clustering neural network.

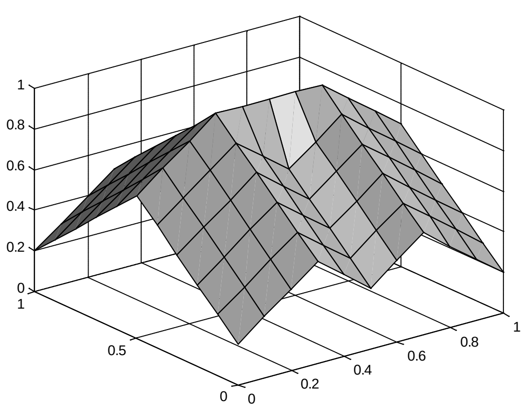

Fig. 2 The plot of extended fuzzy hyperline segment membership function for $\gamma_1 = 1$ with end points w = [0.5 0.3] and v = [0.5 0.7].

3. The EFHLSNN Learning Algorithm

The learning algorithm consists of three steps, creation of hyperline segments, clustering hyperline segments and intersection test. These three steps are described below in detail.

Creation of hyperline segments: The maximum length of hyperline segment is controlled by the parameter, ξ, bounded by $0 \le \xi \le 1$. The value of ξ should be moderately high so that the created hyperline segments will include the patterns, which are close to each other and possibly falling in the same cluster. Assuming $R \in \{R_h \mid h = 1, 2, .., P\}$, where $R_h = (r_{h1}, r_{h2}, ..., r_{hn}) \in I^n$ is the h_{th} pattern belonging to the training set R, the learning process begins by initializing first hyperline segment by the first pattern and then applying unlabeled patterns one by one from the pattern set. The applied pattern is tested for inclusion by calculating the fuzzy membership value with the already created hyperline segments having same end points.

Let $R_h = (r_{h1}, r_{h2}, ..., r_{hn})$ is the h_{th} input pattern, $V_j = (v_{j1}, v_{j2}, ..., v_{jn})$ is one end point of the hyperline segment e_j and $W_j = (w_{j1}, w_{j2}, ..., w_{jn})$ is the other end point of e_j. The fuzzy hyperline segment membership function of j_{th} F_E node is defined as

$$e_j(R_h, V_j, W_j) = 1 - f^3(x, \gamma_1, l), \qquad (1)$$

in which $x = l_1 + l_2$ and the distances l_1, l_2 and l are defined as (2), (3) and (4).

Here in this paper we have used Manhattan distance for computing the values of l_1, l_2 and l as shown in equation 5, 6 and 7 which has given best performance in comparison with Euclidian distance [8] as shown below.

$$l_1 = \max\left(\sum_{i=1}^{n} |w_{ji} - r_{hi}|\right) \qquad (2)$$

$$l_2 = \max\left(\sum_{i=1}^{n} |v_{ji} - r_{hi}|\right) \qquad (3)$$

$$l = \max\left(\sum_{i=1}^{n} |w_{ji} - v_{ji}|\right) \qquad (4)$$

and $f^3(\cdot)$ is a three-parameter ramp threshold function defined as,

$$f^3(x,\gamma_1,l)=0 \qquad \text{if } x=l \text{ otherwise}$$

$$f^3(x,\gamma_1,l)=\begin{cases} x\gamma_1 & \text{if } 0\le x\gamma_1\le 1 \\ 1 & \text{if } x\gamma_1>1. \end{cases}$$

The parameter γ regulates how fast the membership value decreases when the distance between R_h and e_j increases. If the fuzzy membership value calculated is grater than or equal to ξ for any one hyperline segment then the pattern is included by extending that hyperline segment else new hyperline segment is created.

Clustering hyperline segments: The number of clusters constructed depends on the parameters α_2 and β, called as centering and bunching factors, respectively and the values of these parameters are problem dependent. The clustering consists of three steps, I: Determining the centroid, II: Bunching and III: Removal of bunched patterns and hyperline segments. These three steps are described below in detail.

Determining the centroid: To determine the centroid of the cluster, all the patterns are applied to each of the hyperline segments and the patterns that give fuzzy membership larger than α are counted for all hyperline segments. If e_j is the hyperline segment with the maximum count then the centroid is computed as

$$(w_{ji}+v_{ji})/2 \quad \text{for } i=1,2,.........,n. \qquad (5)$$

Bunching: The hyperline segments, which are falling around the centroid and give fuzzy membership value larger than β are bunched in a cluster. Thus the cluster boundaries (i.e. the number of clusters) are governed by the value of bunching factor. As the clusters are being formed the weights in U matrix are also updated as

$$u_{jk}=\begin{cases} 1 & \text{if } e_j \text{ is the hyperline segment} \\ & \text{of the cluster } c_k, \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

for $k=1,2,......,p$ and $j=1,2,......,m$.

Removal of bunched patterns and hyperline segments: The clustered hyperline segments in previous step and the patterns included by these hyperline segments are eliminated. Thus, the next pass uses remaining unclustered hyperline segments and pattern set consisting of remaining patterns for clustering. These three steps are repeated till all the created hyperline segments are clustered.

Let R_p, R_c and R_n represent set of patterns used in the current pass, set of patterns clustered in the current pass and set of patterns that will be used in the next pass, respectively. Then R_n can be described as,

$$R_n=R_p-R_c=\left\{R_n \mid R_n\in R_p \text{ and } R_n\notin R_c\right\} \qquad (7)$$

The R_n calculated in the current pass becomes R_p for the next pass.

Each node of F_C layer represents a cluster. It gives soft decision and the output of *k*th F_C node represents the degree to which the input pattern belongs to the cluster c_k. The transfer function of each F_C node performs the union of appropriate (of same cluster) hyperline segment fuzzy values, which is described as,

$$c_k=\max_{j=1}^{m} e_j\, u_{jk} \qquad \text{for } k=1,2,........,p. \qquad (8)$$

Intersection test: The learning algorithm allows the intersection of hyperline segments from the same cluster and eliminates the intersection between the hyperline segments from separate clusters. If the two hyperline segments from different clusters are intersecting then the intersection is removed by breaking one of the hyperline segment.

4. Simulation Results

The EFHLSCNN is trained with fingerprint feature vector data by setting $\alpha=0.9$, and $\gamma=1$. The value of ξ is set moderately large so that the EFHLSCNN algorithm will create hyperline segments of patterns possibly falling in the same cluster. We have adjusted the value of α close to one so that while computing the centroids, patterns falling around and close to hyperline segment are counted. These values have resulted in the creation of 295 hyperline segments.

The performance of EFHLSCNN algorithm is tested with fingerprint feature vector data. The experiments are carried out with $\alpha=0.9$, $\xi=0.9$, $\gamma=1$ and by varying the bunching factor β. These results are tabulated in the Table 1. The centering and bunching factors are fixed to moderately high value so that EFHLSCNN creates hyperline segments of patterns possibly falling close to each other and belonging to the same cluster and while finding the centroids it will count patterns falling around hyperline segment under consideration. As β increases the number of hyperline segments bunched in cluster decreases, which leads to increase in number of clusters. The performance of EFHLSCNN algorithm is compared with FHLSNN [9], EFHLSNN [14] and FHLSCNN algorithm. The results are depicted in Table 2

The experiments are carried out using fingerprint feature vector and the results obtained are tabulated in Table 2. The timing analysis is also depicted in Table 2.

The FHLSNN algorithm created 200 hyperline segments when trained with the parameters $\theta = 0.14$ and $\gamma = 1$. The results delivered by FHLSNN algorithm are tabulated in the first row of Table 2. This row indicates that the FHLSNN algorithm gives better results with less number of hyperline segments.

Recognition Rate	β	No. of clusters
60.25	0.6	124
68.25	0.65	148
77.75	0.7	174
88.25	0.75	206
92.75	0.8	237
98.00	0.85	263
100	0.9	286

Table 1: The percentage recognition rates obtained from EFHLSCNN algorithm using fingerprint feature vector with number of created clusters

The results obtained from FHLSCNN algorithm for 268 clusters are depicted in the third row of Table 2 with 87.75 recognition rate with 5.133608 seconds recall time. The proposed algorithm gives the recognition rate of 100 percent with 128.9859 and recall time of 4.279710 which is less as compared to FHLSCNN.

Algorithm	Avg.	Training Time	Recall time per pattern
FHLSNN	100	0.1648	0.811566
EFHLSNN	100	0.1631	0.743732
FHLSCNN	87.75	128.9859	5.133608
EFHLSCNN	100	246.4701	4.279710

Table 2: The percentage recognition rates and timing analysis using fingerprint feature vector

The results are depicted in fourth row of Table 2. The experimental result confirms that the EFHLSCNN algorithm generalizes well and yields highest average percentage recognition rates than the FHLSCNN algorithm except increase in training time and less recall time.

These results indicate that EFHLSCNN algorithm gives best average percentage recognition rate as compared to FHLSCNN algorithms using Manhattan distance for calculating distance of hyperline segments. The improvement in generalization performance indicates that clusters are created properly in the pattern space.

Finally, the Table 1 and 2 indicates that the results obtained using fingerprint feature vectors are superior in terms of recognition rate and generalization when we are solving clustering problems.

5. Conclusions

A new extended approach of clustering that utilizes hyperline segments as fuzzy sets that are aggregated into fuzzy set clusters with revision is introduced. The performance of EFHLSCNN algorithm is found superior compared to FHLSCNN algorithm when applied to clustering of fingerprint feature data.

It is observed that the EFHLSCNN algorithm generalizes fit and produces highest average percentage recognition rates as compared with the FHLSCNN algorithm except increase in training time.

6. References

[1] H.C. Lee and R.E. Gaensslen, “Advances in Fingerprint Technology”, second ed., New York: CRC Press, 2001.
[2] Latent Print Examination, http://onin.com/fp/, 2002.
[3] Robert S. Germain, Andrea Califano, and Scott Colville, “Fingerprint Matching Using Transformation Parameter Clustering”, IEEE Computational Science & Engineering, October-December 1997.
[4] Bezdek, J.C. “Computing with uncertainty”, IEEE Communication Mag., pp. 24–36, Sept. 1992
[5] M. H. Hassoun, “Fundamentals of Artificial Neural Networks”, Cambridge, MA: MIT Press, 1995.
[6] R. Krishnapuram, “Generation of membership functions via possibilistic clustering”, in Proc. 1994 IEEE 3rd Int. Fuzzy Systems Conf., vol. 2, June 1994, pp. 902–908.
[7] J. Bezdek, “Pattern recognition with fuzzy objective function algorithms”, New York: Plenum Press, 1981.
[8] Kulkami U. V, Sontakke T. R, Kulkami A. B, “Fuzzy hyperline segment clustering neural network”. IEEE Electronics Letters, 37 (S) , pp 301-303,2001.
[9] U.V. Kulkarni, T.R. Sontakke and G. D. Randale, “Fuzzy Hyperline Segment Neural Network for Rotation Invariant Handwritten Character Recognition”, Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on Neural Network, Washington, DC, USA, 2001, pp. 2918 - 2923 vol.4.
[10] Bogdan Gabrys and Andrzej Bargiela, “General Fuzzy Min-Max Neural Network for Clustering and Classification”, IEEE

TRANSACTIONS ON NEURAL NETWORKS, VOL. 11, NO. 3, MAY 2000

[11] Patrick K. Simpson, "Fuzzy min-max neural networks—Part 1: Classification", IEEE, Trans. Neural Networks, vol. 3, pp. 776–786, Sept. 1992.

[12] Patrick K. Simpson, "Fuzzy min-max neural networks—Part 2: Clustering", IEEE Trans. Fuzzy Syst., vol. 1, pp. 32–45, Feb. 1993.

[13] A. Joshi, N. Ramakrishman, E. N. Houstis, and J. R. Rice, "On neurobiological, neurofuzzy", machine learning, and statistical pattern recognition techniques," IEEE Trans. Neural Networks, vol. 8, Jan. 1997.

[14] M. Meneganti, F. S. Saviello, and R. Tagliaferri, "Fuzzy neural networks for classification and detection of anomalies", IEEE Trans. Neural Networks, vol. 9, Sept. 1998.

[15] M. H. Kondekar, Dr. U. V. Kulkarni & S. S. Chowhan, "Fingerprint Recognition Using Extended Fuzzy Hypersphere Neural Network", JOURNAL OF COMPUTING, VOLUME 3, ISSUE 4, APRIL 2011, ISSN 2151-9617,pp. 101-105.

M. H. Kondekar received the M.C.A. degree from Government Engineering College, Aurangabad in the year 2000. He received the M.Phil. Degree in Computer Science from Y.C.M.O. University, Nashik in the year 2009. He is currently working as Lecturer in the College of Computer Science and Information Technology, Latur, Maharastra, India. His current research interests include various aspects of Neural Networks and Fuzzy Logic, Pattern Recogntion and Biometrics.

Dr.U.V.Kulkarni received the Ph.D. degree in Electronics and Computer Science Engineering from S.R.T.M. University, Nanded in the year 2003. He is currently working as professor in Computer Science and Engineering Department, SGGS Institute of Engineering and Technology, Nanded, Maharastra, India.

S. S. Chowhan received the M.Sc.(CS) degree from Dr. BAM University, Aurangabad in the year 2000. He received the M.Phil. Degree in Computer Science from Y.C.M.O. University, Nashik in the year 2008. He is currently working as lecturer in the College of Computer Science and Information Technology, Latur, Maharastra. His current research interests include various aspects of Neural Networks and Fuzzy Logic, Pattern Recogntion and Biometrics.