ADAPTIVE FUZZY SYSTEM FOR 3-D VISION

FINAL REPORT
ON
NASA/JSC CONTRACT
# NAG 9-509

CONTRACT PERIOD: 4-01-91 TO 3-31-92

SUBMITTED BY

Dr. Sunanda Mitra
Principal Investigator and
Associate Professor,
Department of Electrical Engineering.
Texas Tech University
Lubbock, Texas 79409-3102

To
Dr. Robert Lea
Fuzzy Logic Technical Coordinator,
Information System Directorate,
NASA- Johnson Space Center PT4
Houston, Texas 77058

April 15,1992

PROJECT SUMMARY

A novel adaptive fuzzy system using the concept of the Adaptive Resonance Theory (ART) type neural network architecture and incorporating fuzzy c-means (FCM) system equations for reclassification of cluster centers has been developed.

The Adaptive Fuzzy Leader Clustering (AFLC) architecture is a hybrid neural-fuzzy system which learns on-line in a stable and efficient manner. The system uses a control structure similar to that found in the Adaptive Resonance Theory (ART-1) network to identify the cluster centers initially. The initial classification of an input takes place in a two stage process; a simple competitive stage and a distance metric comparison stage. The cluster prototypes are then incrementally updated by relocating the centroid positions from Fuzzy c - Means (FCM) system equations for the centroids and the membership values. The operational characteristics of AFLC and the critical parameters involved in its operation are discussed. The performance of the AFLC algorithm is presented through application of the algorithm to the Anderson Iris data, and laser-luminescent fingerprint image data. The AFLC algorithm successfully classifies features extracted from real data, discrete or continuous, indicating the potential strength of this new clustering algorithm in analyzing complex data sets.

This hybrid neuro-fuzzy AFLC algorithm will ehnance analysis of a number of difficult recognition and control problems involved with Tethered Satellite Systems and on-orbit space shuttle attitude controller.

I. INTRODUCTION

Cluster analysis has been a significant research area in pattern recognition for a number of years[1]-[4]. Since clustering techniques are applied to the unsupervised classification of pattern features, a neural network of the Adaptive Resonance Theory (ART) type[5],[6] appears to be an appropriate candidate for implementation of clustering algorithms[7]-[10]. Clustering algorithms generally operate by optimizing some measures of similarity. Classical, or crisp, clustering algorithms such as ISODATA[11] partition the data such that each sample is assigned to one and only one cluster. Often with data analysis it is desirable to allow membership of a data sample in more than one class, and also to have a degree of belief that the sample belongs to each class. The application of fuzzy set theory[12] to classical clustering algorithms has resulted in a number of algorithms[13]-[16] with improved performance since unequivocal membership assignment is avoided. However, estimating the optimum number of clusters in any real data set still remains a difficult problem[17].

It is anticipated, however, that a valid fuzzy cluster measure implemented in an unsupervised neural network architecture could provide solutions to various real data clustering problems. The present work describes an unsupervised neural network architecture[18],[19] developed from the concept of ART-1[5] while including a relocation of the cluster centers from FCM system equations for the centroid and the membership values[2]. Our AFLC system differs from other fuzzy ART-type clustering algorithms [20],[21] incorporating fuzzy min-max learning rules. The AFLC presents a new approach to unsupervised clustering, and has been shown to correctly classify a number of data sets including the Iris data. This fuzzy modification of an ART-1 type neural network, i.e. the AFLC system, allows classification of discrete or analog patterns without a priori knowledge of the number of clusters in a data set. The optimal number of clusters in many real data sets is, however, still dependent on the validity of the cluster measure, crisp or fuzzy, employed for a particular data set.

II. ADAPTIVE FUZZY LEADER CLUSTERING SYSTEM AND ALGORITHM

A. AFLC System and Algorithm Overview

AFLC is a hybrid neural-fuzzy system which can be used to learn cluster structure embedded in complex data sets, in a self-organizing, stable manner. This system has been adapted from the concepts of ART-1 structure which is limited to binary input vectors[5]. Pattern classification in ART-1 is achieved by assigning a prototype vector to each cluster that is incrementally updated[10].

Let $X_j = \{ X_{j1}, X_{j2}, \dots X_{jp} \}$ be the j th input vector for $1 \leq j \leq N$ where N is the total number of samples in the data set and p is the dimension of the input vectors. The initialization and updating procedures in ART-1 involve similarity measures between the bottom-up weights (b_{ki} where k = 1,2,...,p) and the input vector (X_j), and a verification of X_j belonging to the i th cluster by matching of the top-down weights (t_{ik}) with X_j. For continuous-valued features, the above procedure is changed as in ART-2[6]. However if the ART-type networks are not made to represent biological networks, then a greater flexibility is allowed to the choice of similarity metric. A choice of Euclidean metric is made in developing the AFLC system while keeping a simple control structure adapted from ART-1.

Figure 1

Figures 1(a) and 1(b) represent the AFLC system and operation for initialization and comparison of cluster prototypes from input feature vectors, which may be discrete or analog. The updating procedure in the AFLC system involves relocation of the cluster prototypes by incremental updating of the centroids v_i, (the cluster prototypes), from FCM system equations[2] for v_i and μ_{ij} as given below :

$$V_i = \frac{1}{\sum_{j=1}^{N_i} (\mu_{ij})^m} \sum_{j=1}^{N_i} (\mu_{ij})^m X_j \; ; \qquad 1 \leq i \leq C \tag{1}$$

$$\mu_{ij} = \frac{\left(\frac{1}{\|X_j - v_i\|^2}\right)^{1/(m-1)}}{\sum_{k=1}^{C}\left(\frac{1}{\|X_j - v_k\|^2}\right)^{1/(m-1)}}; \quad 1 \le i \le C; \; 1 \le j \le N \tag{2}$$

where N_i is the number of samples in cluster i and C is the number of clusters. The v_i's and μ_{ij}'s are recomputed over the entire data sample N.

As described here, AFLC is primarily used as a classifier of feature vectors employing an on-line learning scheme. Figure 1(a) shows a p-dimensional discrete or analog-valued input feature vector, X to the AFLC system. The system is made up of the comparison layer, the recognition layer, and the surrounding control logic. The AFLC algorithm initially starts with the number of clusters (C) set to zero. The system is initialized with the input of the first feature vector X. Similar to leader clustering, this first input is said to be the prototype for the first cluster. The normalized input feature vector is then applied to the bottom-up weights in a simple competitive learning scheme, or dot product. The node that receives the largest input activation Y is chosen as the prototype vector as is done in the original ART-1.

$$Y_i = \max\{\sum_{k=1}^{p} X_{jk} b_{ki}\}; \qquad 1 \le j \le N \tag{3}$$

Therefore the recognition layer serves to initially classify an input. This first stage classification activates the prototype or top-down expectation (t_{ik}) for a cluster, which is forwarded to the comparison layer. The comparison layer serves both as a fan-out site for the inputs, and the location of the comparison between the top-down expectation and the input. The control logic with an input enable command allows the comparison layer to accept a new input as long as a comparison operation is not currently being processed. The control logic with compare imperative command disables the acceptance of new input and initiates comparison between the cluster prototype of Y_i i.e., the centroid v_i and the current input vector, using equation (4). The reset signal is activated when a mismatch of the first and second input vectors occurs according to the criterion of a distance ratio threshold as expressed by equation (4)

$$R = \frac{\sqrt{d^2(X_j, v_i)}}{\frac{1}{N_i}\sum_{k=1}^{N_i}\sqrt{d^2(X_k, v_i)}} < \tau \qquad (4)$$

where : k = 1....N_i the number of samples in class i and $\sqrt{d^2(X_j, v_i)}$ is the Euclidean distance as indicated in equation(5).

$$d^2(x_j - v_i) = \|x_j - v_i\|^2 \qquad (5)$$

If the ratio R is less than a user-specified threshold τ, then the input is found to belong to the cluster originally activated by the simple competition. The choice of the value of τ is critical and is found by a number of initial runs. Preliminary runs with τ varying over a range of values yield a good estimate of the possible number of clusters in unlabeled data sets.

When an input is classified as belonging to an existing cluster, it is necessary to update the expectation (prototype) and the bottom-up weights associated with that cluster. First, the degree of membership of X to the winning cluster is calculated. This degree of membership, μ, gives an indication, based on the current state of the system, of how heavily X should be weighted in the recalculation of the class expectation. The cluster prototype is then recalculated as a weighted average of all the elements within the cluster. The update rules are as follows: the membership value μ_{ij} of the current input sample X_j in the winning class i, is calculated using equation (2), and then the new cluster centroid for cluster i is generated using equation (1). As with the FCM, m is a parameter which defines the fuzziness of the results and is normally set to be between 1.5 and 30. For the following applications, m was experimentally set to 2.

The AFLC algorithm can be summarized by the following steps :

1. Start with no cluster prototypes, C = 0.
2. Let X_j be the next input vector.

3. Find the first stage winner Y_i, as the cluster prototype with the maximum dot-product.

4. If Y_i does not satisfy the distance ratio criterion, then create a new cluster and make its prototype vector be equal to X_j. Output the index of the new cluster.

5. Otherwise, update the winner cluster prototype Y_i by calculating the new centroid and membership values using equations (1) and (2). Output the index of Y_i. Go to Step 2.

A flow chart of the algorithm is shown in Figure 2.

Figure 2

III. OPERATIONAL CHARACTERISTICS OF AFLC

A. Match-based Learning and the Search

In match-based learning, a new input is learned only after being classified as belonging to a particular class. This process ensures stable and consistent learning of new inputs by updating parameters only for the winning cluster and only after classification has occurred. This differs from error-based learning schemes, such as backpropagation of error, where new inputs are effectively averaged with old learning resulting in forgetting and possibly oscillatory weight changes. In [5] match-based learning is referred to as resonance, hence the name Adaptive Resonance Theory.

Because of its ART-like control structure, AFLC is capable of implementing a parallel search when the distance ratio does not satisfy the thresholding criterion. The search is arbitrated by appropriate control logic surrounding the comparison and recognition layers of Figure 1. This type of search is necessary due to the incompleteness of the classification at the first stage. For illustration, consider the two vectors (1,1) and (5,5). Both possess the same unit vector. Since the competition in the bottom-up direction consists of measuring how well the normalized input matches the weight vector for each class i, these inputs would both excite the same activation pattern in the recognition layer. In operation, the comparison layer serves to test the hypothesis returned by the competition performed at the recognition layer. If the hypothesis is disconfirmed by the comparison layer, i.e. $R > \tau$, then the search phase continues until the correct cluster is found or another cluster is created. Normalization of the input vectors (features) is done only in the recognition layer for finding the winning node. This normalization is essential to avoid large values of the dot products of the input features and the bottom-up weights and also to avoid initial misclassification arising due to large variations in magnitudes of the cluster prototypes. The search process, however, renormalizes only the centroid and not the input vectors again.

B. Determining the Number of Output Classes

AFLC utilizes a dynamic, self-organizing structure to learn the characteristics of the input data. As a result, it is not necessary to know the number of clusters a priori; new clusters are added to the system as needed. This characteristic is necessary for autonomous behavior in practical situations in which nonlinearities and nonstationarity are found.

Clusters are formed and trained, on-line, according to the search and learning algorithms. Several factors affect the number, size, shape, and location of the clusters formed in the feature space. Although it is not necessary to know the number of clusters which actually exist in the data, the number of clusters formed will depend upon the value of τ. A low threshold value will result in the formation of more clusters because it will be more difficult for an input to meet the classification criteria. A high value of τ will result in fewer, less dense clusters. For data structures having overlapping clusters, the choice of τ is critical for correct classification whereas for nonoverlapping cluster data, the sensitivity of τ is not a significant issue. In the latter case the value of τ may vary over a certain range, yet yielding correct classification. Therefore the sensitivity of τ is highly dependent on specific data structure as shown in Figure 1(c). The relationship between τ and the optimal number of clusters in a data set is currently being studied.

C. Dynamic Cluster Sizing

As described earlier, τ is compared to a ratio of vector norms. The average distance parameter for a cluster is recalculated after the addition of a new input to that cluster; therefore, this ratio (R) represents a dynamic description of the cluster. If the inputs are dense around the cluster prototype, then the size of the cluster will decrease, resulting in a more stringent condition for membership of future inputs to that class. If the inputs are widely grouped around the cluster prototype, then this will result in less stringent conditions for membership. Therefore, the AFLC clusters have a self-scaling factor which tends to keep dense clusters dense while allowing loose clusters to exist.

D. The Fuzzy Learning Rule

In general, the AFLC architecture allows learning of even rare events. Use of the fuzzy learning rule in the form of equations (1) and (2), maintains this characteristic. In weighted rapid learning[5], the learning time is much shorter than the entire processing time and the adaptive weights are allowed to reach equilibrium on each presentation of an input, but the amount of change in the prototype is a function of the input and its fuzzy membership value (μ_{ij}). Noisy features which would normally degrade the validity of the class prototype are assigned low weights to reduce the undesired affect. In the presence of class outliers, assigning low memberships to the outliers lead to correct classification. Normalization of membership is not involved in this process. However, a new cluster of outliers only can be formed during the search process[22]. Development of such outlier/noise cluster in AFLC is currently under progress.

Weighted rapid learning also tends to reinforce the decision to append a new cluster. This is due to the fact that, by definition, the first input to be assigned to a node serves as that node's first prototype, therefore, that sample has a membership value of one. Future inputs are then weighted by how well they match the prototype. Although the prototype does change over time, as described in the algorithm, each sample retains its weight which tends to limit moves away from the current prototype. Thus the clusters possess a type of inertia which tends to stabilize the system by making it more difficult for a cluster to radically change its prototype in the feature space.

Finally, the fuzzy learning rule is stable in the sense that the adaptive weights represent a normalized version of the cluster centroid, or prototype. As such, these weights are bounded on [0,1] and are guaranteed not to approach infinity.

E. AFLC as a General Architecture

As with most other clustering algorithms, the size and shape of the resultant clusters depends on the metric used. The use of any metric will tend to influence the data toward a solution which meets the criteria for that metric and not necessarily to the best solution for the data. This statement implies that some metrics are better for some problems than are others. The use of a Euclidean metric is convenient, but displays the immediate problem that it is best suited to simple circular cluster shapes. The use of the Mahalanobis distance accounts for some variations in cluster shape, but its non-linearity serves to place constraints on the stability of its results. Also, as with other metrics, the Euclidean and Mahalanobis distance metrics lose meaning in an anisotropic space.

IV. TESTS AND RESULTS: FEATURE VECTOR CLASSIFICATION

A. Clustering of the Anderson Iris Data

The Anderson Iris data set[23], consists of 150 4-dimensional feature vectors. Each pattern corresponds to characteristics of one flower from one of the species of Iris. Three varieties of Iris are represented by 50 of the feature vectors. This data set is popular in the literature and gives results by which AFLC can be compared to similar algorithms.

We had 52 runs of the AFLC algorithm for the Iris data for 13 different values of τ, with 4 runs for each τ. Figure 1(c) shows the τ-C graph. With Euclidean distance ratio and τ ranging between 4.5 and 5.5, the sample data was classified into 3 clusters with only 7 misclassifications. The misclassified samples actually belonged to Iris versicolor, cluster #2, and were misclassified as Iris virginica, cluster

#1. From Figure 1(c) it can be observed that the optimal number of clusters can be determined from the τ -C graph as the value of C that has $\frac{dC}{d\tau} = 0;\ for\ \ C \neq 1$, for the maximum possible range of τ.

Figure 3, shows the input Iris data clusters using only three features for each sample data point. Figure 4a shows the computed centroids of the three clusters based on all four features. The intercluster Euclidean distances are found to be 1.75 (d_{12}), 4.93 (d_{23}), and 3.29 (d_{13}). d_{ij} is the intercluster distance between clusters i & j. The comparatively smaller intercluster distance between clusters 1 and 2 indicates the proximity of these clusters. Figure 4b shows a confusion matrix that summarizes the classification results.

Figure 3

Figure 4

B. Classification of Noisy Laser-luminescent Fingerprint Image Data

Fingerprint matching poses a challenging clustering problem. Recent developments in automated fingerprint identification systems employ primitive and computationally intensive matching techniques such as counting ridges between minutae of the fingerprints[24]. Although the technique of laser luminescent image acquisition of latent fingerprint provide often identifiable images[25], these images suffer from amplified noise, poor contrast and nonuniform intensity. Conventional enhancement techniques such as adaptive binarization and wedge filtering provide enhancement at the expense of significant loss of information necessary for matching. Recent work[26] presents a novel three stage matching algorithm for fingerprint enhancement and matching. Figure 5b shows the enhanced image of 5a subsequent to selective Fourier spectral enhancement and bandpass filtering. We used the AFLC algorithm to cluster three different classes of fingerprint images using seven invariant moment features[26],[27] computed from images that are enhanced[26]. A total of 24 data samples are used, each sample being a 7-dimensional moment feature vector. These moment invariants are a set of nonlinear functions which are invariant to translation, scale, & rotation. The three higher order moment features are given less weights thus reducing the affect of noise and leading to proper classification. The τ-C graph for the fingerprint data in Figure 1(c) shows a range of τ from 3.0 to 4.5 for which proper classification resulted. The fingerprint data has also been correctly classified by a k-nearest neighbor clustering using only four moment features[26]. Euclidean distances of these clusters indicate that the clusters are well separated which is consistent with the comparatively larger range of τ found for proper classification. Figures 5a and 5b represent one fingerprint class before and after enhancement. Figure 6a shows the computed centroids of three fingerprint clusters. Figure 6b shows a confusion matrix that indicates correct classification results.

Figure 5, Figure 6

V. CONCLUSION

It is possible to apply many of the concepts of AFLC operation to other control structures. Other approaches to Fuzzy ART are being explored[20],[21] that could also be used as the control structure for a fuzzy learning rule. Choices also exist in the selection of class prototypes. With some modification, any of these techniques can be incorporated into a single AFLC system or a hierarchical group of systems. The characteristics of that system will depend upon the choices made.

While AFLC does not solve all the problems associated with unsupervised learning, it does possess a number of desirable characteristics. The AFLC architecture learns and adapts on-line, such that it is not necessary to have a priori knowledge of all data samples or even of the number of clusters present in the data. However the choice of τ is critical and requires some a priori knowledge of the compactness and separation of clusters in the data structure. Learning is match-based ensuring stable, consistent learning of new inputs. The output is a crisp classification and a degree of confidence for that classification. Operation is also very fast, and can be made faster through parallel implementation. A recent work[28] shows a different approach to neural-fuzzy clustering by integrating Fuzzy C - means model with Kohonen neural networks. A comparative study of these recently developed neural-fuzzy clustering algorithms is needed. Future work will involved further modification of the AFLC system and algorithm for analyzing simulation data of the TSS system[29] and for automated attitude controller design of on-orbit shuttle[30].

VI. ACKNOWLEDGMENT

This work was partially supported by a grant from NASA-JSC under contract #NAG9-509, a grant from Northrop Corporation under contract #HHH7800809K, and a Texas Instruments Graduate Student Fellowship to Scott Newton through Texas Tech University. The authors gratefully acknowledge many excellent and thoughtful suggestions from J. Bezdek for improving this paper.

VII. REFERENCES

[1] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, New York , 1973.

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

[3] A. K. Jain and R. C. Dubes, Algorithms for Clustering Data, Englewood Cliffs, NJ, Prentice-Hall, 1988.

[4] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press Inc., San Diego, 1990.

[5] G. A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-organizing Neural Pattern Recognition Machine," Computer Vision, Graphics & Image Processing, 37, pp. 54-115, 1987.

[6] G.A. Carpenter and S. Grossberg, "ART2: Self organization of stable category recognition codes for analog patterns" Applied Optics 2 pp. 4919-4938, 1987

[7] R. Lippmann, "An Introduction to Computing with Neural Networks," IEEE ASSP Magazine, No 4, pp. 4-21, 1987.

[8] L. I. Burke, "Clustering Characterization of Adaptive Resonance," Neural Networks, Vol. 4, pp 485-491, 1991.

[9] Y.H. Pao, Adaptive Pattern Recognition and Neural Networks, Addison-Wesley Publishing Co., 1989.

[10] B. Moore, "ART 1 and Pattern Clustering," Proc. of the 1988 Connectionist Models Summer School, Eds. D. Touretzky, G. Hinton, and T. Sejnowski, Morgan Kaufman Publishers, San Mateo, CA, 1988.

[11] G. H. Ball and D. J. Hall, "ISODATA, an Iterative Method of Multivariate Data Analysis and Pattern Classification," Proceedings of the IEEE International Communications Conference, Philadelphia, PA, June 1966.

[12] L. A. Zadeh, "Fuzzy Sets," Information and Control, 8, pp. 338-353, 1965.

[13] J. C. Bezdek, "A Physical Interpretation of Fuzzy ISODATA," IEEE Transactions on Systems, Man and Cybernetics, vol. 6, pp. 387-389, 1976.

[14] J. C. Bezdek, "A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-2, no. 1, pp. 1-8, 1980.

[15] E. E. Gustafson and W. C. Kessel, "Fuzzy Clustering with a Fuzzy Covariance Matrix," Proceedings of the IEEE CDC, San Diego, CA, pp. 761-766, 1979.

[16] I. Gath and A. B. Geva, "Unsupervised Optimal Fuzzy Clustering," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pp.773-781, 1989.

[17] X. L. Xie and G. Beni, "A Validity Measure for Fuzzy Clustering," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-13, No. 2, p. 841-847, 1991.

[18] S. C. Newton and S. Mitra, "Self-organizing Leader Clustering In A Neural Network Using A Fuzzy Learning Rule," SPIE Proc. on Adaptive Signal Processing, Vol. 1565, July 1991.

[19] S. C. Newton and S. Mitra, "Applications of Neural Networks in Fuzzy Pattern Recognition," presented at the Research Conference on Neural Networks For Vision and Image Processing, Boston University, May 10-12, 1991.

[20] P. Simpson, "Fuzzy Adaptive Resonance Theory," Proceedings of the Neuroengineering Workshop, H. Szu and M. Sayeh, eds. September, 1990.

[21] G. Carpenter, S. Grossberg, and D. Rosen, "Fuzzy ART: An Adaptive Resonance Algorithm for Rapid, Stable Classification of Analog Patterns," Proceedings of the 1991 International Joint Conference on Neural Networks," July, 1991.

[22] R.N. Dave, "Characterization and detection of noise in clustering" Pattern Recognition Letters 12, No. 11, November, 1991

[23] E. Anderson, "The Irises of the Gaspe Peninsula," Bull. Amer. Iris Soc., Vol. 59, pp. 2-5, 1935.

[24] "Computer Graphics in the Detective Business", IEEE Computer Graphics and Applications, Vol. 5 pp. 14-17, April 1985.

[25] C.R. Menzel, "Latent Fingerprint Development with Lasers", ASTM Standardization News, Vol. 13 pp. 34-37, March 1985.

[26] E. Kaymaz, "Fingerprint Analysis and Matching". M.S.E.E. Thesis, Texas Tech University, December 1991.

[27] M.K. Hu, "Visual Pattern Recognition by Moment Invariants" IRE Trans. Information Theory, Vol. IT-8, pp. 179-187, 1962

[28] J.C. Bezdek, E.C.K. Tao, and N.R. Pal, "Fuzzy Kohonen Clustering Networks", Proc. of IEEE Int. Conference on Fuzzy Systems, pp. 1035-1043, San Diego, March 8-12, 1992.

[29] R. N. Lea, J. Villarreal, Y,. Jani and C. Copeland, "Fuzzy Logic Based Tether Control". Proc. of NAFIPS-'91, p. 398-402, May 14-17, 1991.

[30] R. N. Lea, J. Hoblit and Y. Jani, "Performance Comparison of a Fuzzy Logic Based Attitude Controller with the Shuttle On-orbit Digital Auto-pilot", Proc. of NAFIPS-'91, p. 291-295, May 14-17, 1991.

FIGURE CAPTIONS

Figure 1. Operation characteristics of AFLC Architecture . 1(a) shows the initial stage of identifying a cluster prototype, 1(b) shows the comparison stage using the criterion of Euclidian distance ratio $R > \tau$ to reject new data samples to the cluster prototype. The reset control implies the deactivation of the original prototype and activation of a new cluster prototype and 1(c) shows the τ - c graph for choosing τ for unlabelled datasets.

Figure 2. Flow-chart of the AFLC Algorithm

Figure 3. Iris Data Represented by Three-Dimensional Features

Figure 4a. Computed Centroids of Three Iris Clusters Based on All Four Feature Vectors

Figure 4b. Iris Cluster Classification Results shown as a confusion matrix

Figure 5a. A Noisy Laser-luminescent Fingerprint Image

Figure 5b. The Enhanced Image of 5a. by Selective Fourier Spectral Filtering

Figure 6a. Computed Centroids of Three Fingerprint Clusters in Seven-Dimensional Vector Space

Figure 6b. Fingerprint Data Classification Results

Adaptive Fuzzy Leader Clustering

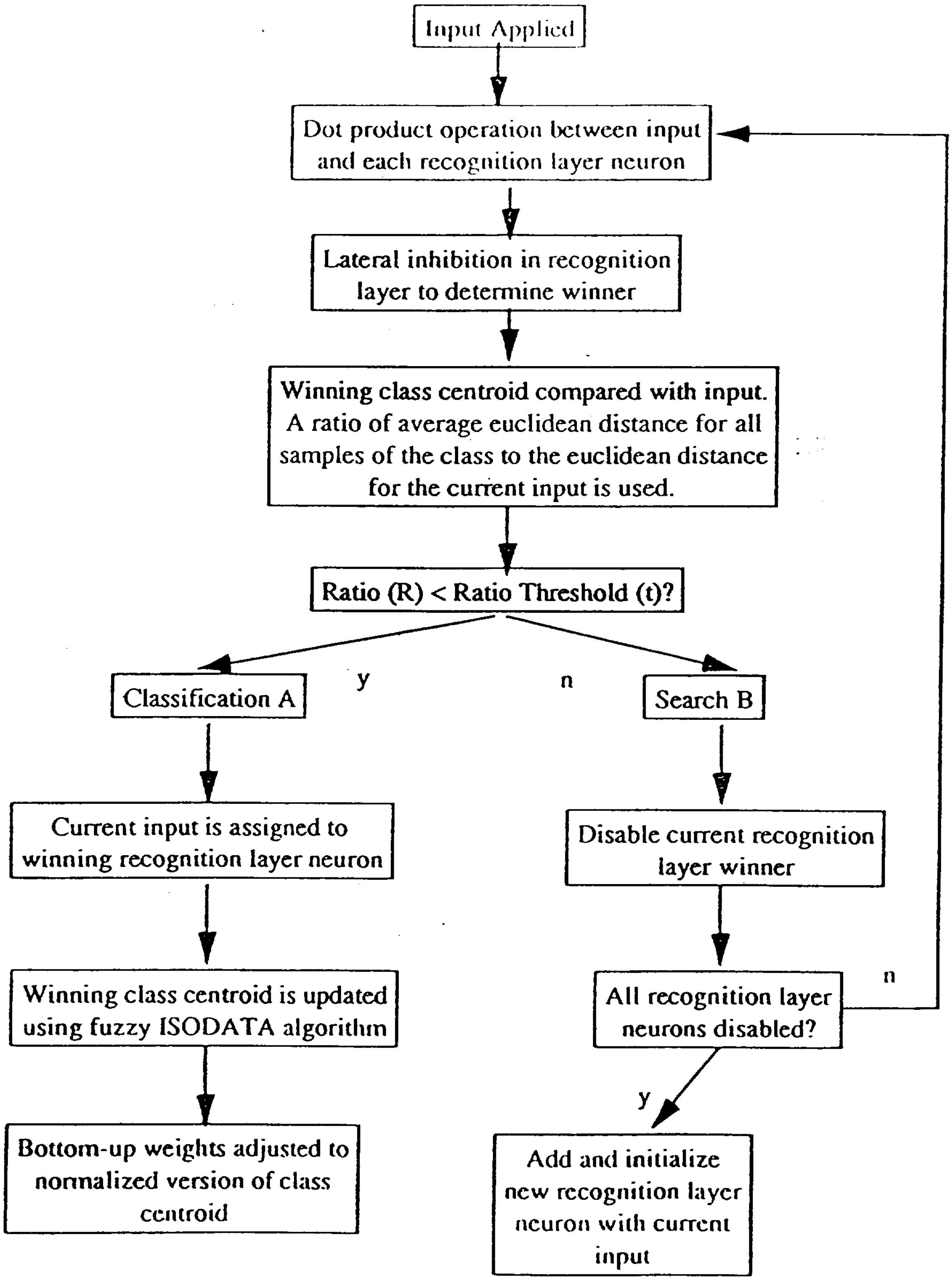
Input Applied
Dot product operation between input and each recognition layer neuron
Lateral inhibition in recognition layer to determine winner
Winning class centroid compared with input. A ratio of average euclidean distance for all samples of the class to the euclidean distance for the current input is used.
Ratio (R) < Ratio Threshold (t)?
y
n
Classification A
Search B
Current input is assigned to winning recognition layer neuron
Disable current recognition layer winner
Winning class centroid is updated using fuzzy ISODATA algorithm
All recognition layer neurons disabled?
n
y
Bottom-up weights adjusted to normalized version of class centroid
Add and initialize new recognition layer neuron with current input

Figure 2.

ANDERSON IRIS DATA

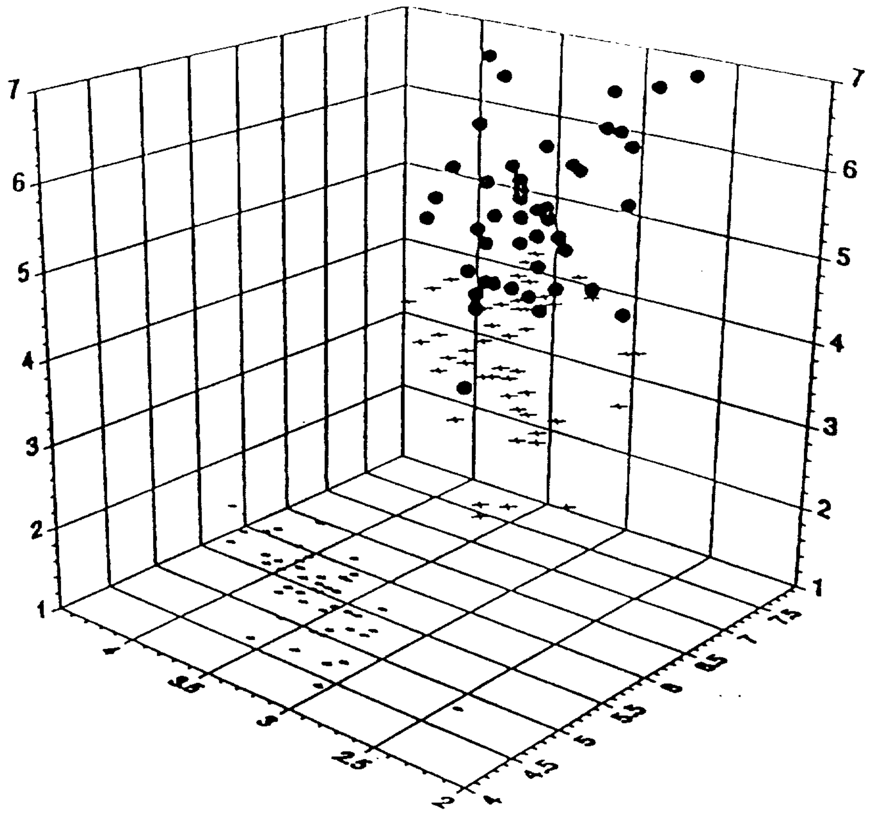

Figure 3.

CLUSTER No.	CLUSTER CENTROID VECTOR xi, i=1,2,3,4			
1	5.95	2.76	4.33	1.34
2	6.72	3.05	5.66	2.14
3	5.00	3.42	1.46	0.24

Figure 4a.

		ACTUAL		
		1	2	3
	1	50	7	
OUTPUT	2		43	
	3			50

Figure 4b.

Fingerprint Data

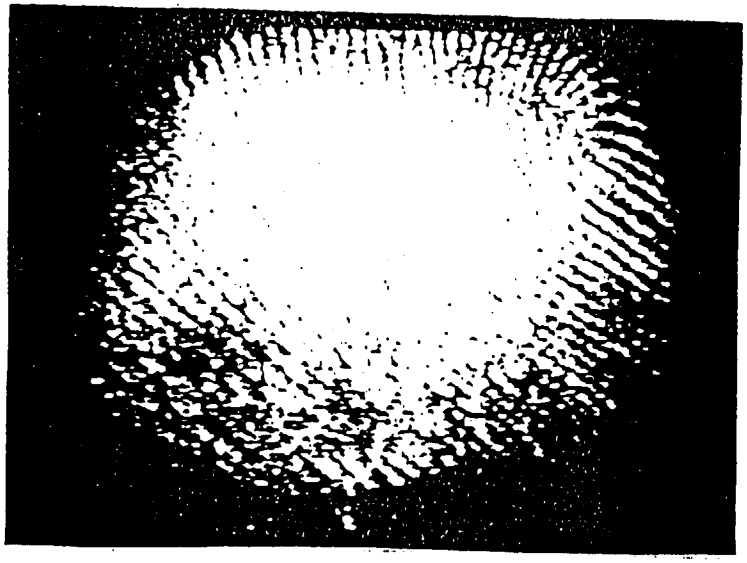

Figure 5a.

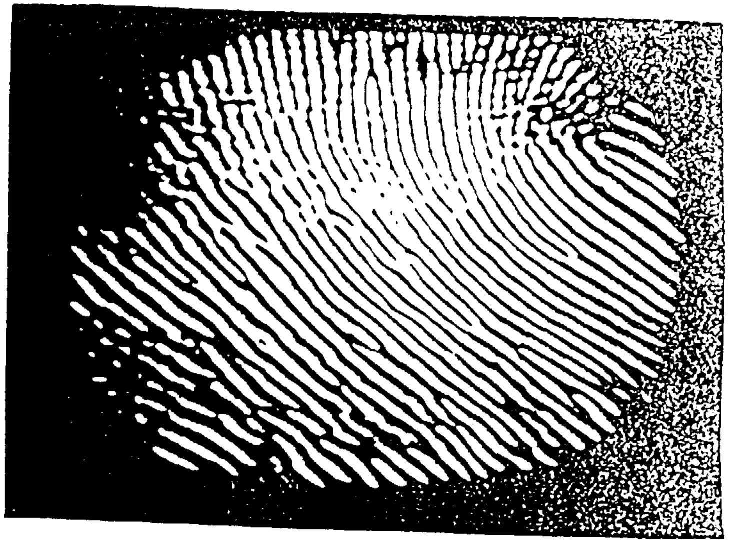

Figure 5b.

CLUSTER No.	CLUSTER CENTROID VECTOR X_j, i = 1,2 ... 7						
	1	2	3	4	5	6	7
1	199.48	15254.10	75.25	14.99	488.95	-822.35	278.50
2	262.52	15956.27	22750.70	4500.20	23888973.00	530694.06	-11096280.00
3	538.03	803.625	57.02	34.38	427.83	112.10	881.09

Figure 6a.

		ACTUAL	
OUTPUT	1	2	3
1	8		
2		8	
3			8

Figure 6b.