The CSI Journal on
Computer Science and Engineering
Vol. 5, No. 2&4 (b), 2007
Pages 33-45
Regular Paper

Learning Cluster Type and Dissimilarity Metric for Each Cluster Using a Set of Possible Cluster Types

Arash Arami[1,2] Babak Nadjar Araabi[1,2] Caro Lucas[1,2] Majid Nili Ahmadabadi[1,2]

[1]Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
[2]School of Cognitive Sciences, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

Abstract

One of the shortcomings of the existing clustering methods is their problems dealing with different shape and size clusters. On the other hand, most of these methods are designed for especial cluster types or have good performance dealing with particular size and shape of clusters. The main problem in this connection is how to define a dissimilarity criterion to make this algorithm capable of clustering general data, which include clusters of different shapes and sizes. Another important objective that must be considered is the computational complexity of any new algorithms. In this paper a new approach to fuzzy clustering is proposed in which a model for each cluster is estimated during learning. Gradually besides, dissimilarity metric for each cluster is defined, updated and used for the next step. In our approach, instead of associating a single cluster type to each cluster, we assume a set of possible cluster types for each cluster with different grades of possibility. Then, a truncation which can be expressed as an attention mechanism focuses on the most probable cluster types for each cluster. This selection step subsides the computational load dramatically while speeds up the clustering. The proposed clustering method which has the capability to deal with partial labeled data is implemented on two families of data, first in presence of partially labeled data, then with fully unlabeled data. Comparing the experimental results of this method with several important existing algorithms, demonstrates the superior performance of proposed method. The merit of this method is its ability to deal with clusters of different shape and size while it computes a fuzzy membership value to different shapes for each cluster.

Keywords: Clustering, Cluster Prototype, Mass Prototype, Linear prototype, Shell Prototype, Fuzzy Membership Function, Attention Control.

1. Introduction

Besides the supervised learning, clustering or unsupervised learning is one of the earliest brain's capabilities, which makes it capable of categorizing patterns of data without supervision. Using this ability, human beings (and lots of animals) can identify the environment around them. Since processing large packages of data considering their vast range seems to be impossible, mankind tends to categorize the objects to assign a finite number of concepts.

More technically, clustering tries to categorize unlabeled data such that the data in each category have the most similarity to each other and dissimilarity to data in other categories. In other words, the goal of clustering is to reveal the organization of pattern into sensible clusters, which will help us to derive useful conclusion about them [1]. The idea of clustering met in many fields, such as life sciences (biology, zoology), medical sciences (psychiatry, pathology), social sciences (sociology, archeology), earth sciences (geography, geology), and engineering [2]. In Several

applications clustering methods are widely used, for instance in machine vision [3], [4] and face recognition [5], images segmentation [6], [7], image compression [8], [9], concept learning [10] and data mining. Generally speaking, clustering can be useful wherever finding rational relativity of data is needed.

Fuzzy clustering which has been originated by fuzzy sets theory regards uncertainty and fuzziness in membership of data points in finding clusters. One of the primary methods for fuzzy clustering was introduced by Dunn [11], which was based on objective functions defined by Euclidean distances. This method was then extended by Bezdak [12] and [13]. Soft clustering algorithms based on L_1 norm were propounded by Jajuga [14]. Furthermore, Yang has done a thorough survey on fuzzy clustering [15]. There also has been adaptive fuzzy clustering method introduced by Gustafson and Kessel [16], in which the clusters' shapes change according to a quadratic distance defined in a fuzzy covariance matrix. A detailed study of this algorithm was presented by Krishnapuram and Kim [17], and an improved approach for estimating the fuzzy covariance matrix in Gustafson-kessel algorithm was proposed by Babuska et al. [18]. Moreover, another clustering algorithm was proposed by Gath and Geva [19], in which the cluster covariance matrix was used in conjunction with an exponential distance, and the clusters were not constrained in volume. Although this algorithm has better performance dealing with different sizes and even shapes of clusters, this algorithm is less robust in the sense that it needs appropriate initialization. Lack of precisely defined initialization usually leads to convergence of algorithm into local optimum points. Recently, a partitional fuzzy clustering method based on adaptive quadratic distance introduced by Carvalho [20]. Also, Bouchachia has enhanced the performance of fuzzy clustering by adding a mechanism of partial supervision [21]. A generalization of fuzzy C-means was proposed in [22] which increased the robustness of fuzzy C-means algorithm. In order to improve the performance dealing with high dimensional data, using a dual-partitioning approach, Tjhi and Chen proposed a new heuristic based co-clustering algorithm [23]. In order to overcome some problems in image segmentation, Cai et al. [24] proposed a fast generalized fuzzy C-means (FGFCM) algorithm by introducing a new localized similarity metric. The proposed similarity metric makes use of local and spatial intensity information. In addition some works have been done in information theoretic clustering in which the information theoretical measures such as mutual information have been used as similarity metric, for instance a robust information clustering algorithm was proposed by Song [25].

Since the goal of clustering is to discover data patterns, finding similarity definition according to different data structures is a crucial and laborious job, and has integral role in success of clustering. This can be more clarified when the data has non-mass type extensions, for example linear type or shell type, in which a wrong similarity/dissimilarity criterion definition leads to an inappropriate clustering. In general cases, there is no prior information about the type of data extension and clusters' model; hence a method which can learn, revises its criterion and uses it in next steps during clustering is essential. In this paper we propose a method to cluster general data based on learning cluster models and similarity/dissimilarity criterion and a simple attention control mechanism for diminishing the computational costs.

In section 2, a brief review of two optimization based fuzzy clustering methods is provided. In section 3, proposed clustering method is introduced and formulated. Also the truncation mechanism and the necessity of this selection mechanism are described in this section. The experimental results and comparison with other methods are presented in section 4. Finally the conclusions are drawn in the last section.

2. Fuzzy Clustering Methods

Since similarity and dissimilarity are themselves fuzzy concepts, fuzzy clustering has been an interesting topic in recent decades. In fuzzy clustering, each datum not only belongs to a cluster, but it belongs also to different clusters with different weights. In order to find the membership values for each data point in each cluster, an optimization process is needed. Therefore, a cost function is needed to be defined. A conventional cost function which widely used for fuzzy clustering is shown in equation 1.

It can be interpreted that, the goal of fuzzy clustering is to minimize the following cost function:

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} U_{ij}^{m} d(x^j, w_i) \tag{1}$$

In which, c is the number of clusters, N is the number of data points, U_{ij} is the membership degree of the *jth* data to the *ith* cluster, m is a fuzzy exponent, ω_i is the *ith* cluster prototype, and $d(x^j, w_i)$ is a dissimilarity criterion between the *jth* datum and *ith* cluster that can be Euclidean distance in its simple form. Fuzzy partitioning condition can be used to prevent achieving redundant results:

$$\sum_{\substack{i=1 \\ \forall j}}^{c} U_{ij} = 1 \tag{2}$$

There are several methods for fuzzy clustering, while in this section fuzzy C-means and Gustafson Kessel are briefly described. Fuzzy C-means clustering is one of the most important methods, in which computing the clusters' prototypes are also involved in the optimization problem. After using the Lagrange multipliers method and by

calculating the derivatives of equation 1 under the constraint (equation 2) with respect to U_{ij} and ω_i these rules for updating are achieved:

$$U_{ij} = \frac{1}{\sum_{L=1}^{c} \left(\frac{d(x^j, \omega_i)}{d(x^j, \omega_L)} \right)^{\frac{1}{m-1}}} \qquad m > 1 \tag{3}$$

$$\omega_i = \left(\frac{\sum_{j=1}^{N} U_{ij}^m x^j}{\sum_{j=1}^{N} U_{ij}^m} \right) \tag{4}$$

In fuzzy C-linear and C-spherical clustering the linear and spherical prototypes are chosen as clusters' prototypes respectively. To clarify the concept of different prototypes, for instance, existence of a linear prototype means that the extension of data of the clusters in some dimensions are significantly more than the other dimensions, and instead of assuming a point in feature space as the center of cluster, a hyper plane can be considered as the model of cluster which will be used as a reference (prototype) to calculated the distance (dissimilarity) of any datum to that cluster. In the similar way, considering a shell prototype for a cluster means that the distance of each datum will be calculated to a shell instead of the fuzzy mean of cluster's data points.

Gustafson-Kessel is another method, in which defining a new distance between data and clusters, detecting the mass and linear clusters have become possible. The distance is formulated below:

$$\begin{aligned} d(x^j, \omega_i) &= d_{A_i}(x^j, \omega_i) = \left\| x^j - \omega_i \right\|_{A_i}^2 \\ &= (x^j - \omega_i)^T A_i (x^j - \omega_i) \end{aligned} \tag{5}$$

In which, *Ai* is a symmetric, real and positive definite matrix. So, the cost function would be changed accordingly:

$$J = \sum_{i=1}^{C} \sum_{j=1}^{N} U_{ij}^m \left\| x^j - w_i \right\|_{A_i}^2 \tag{6}$$

In addition to the partitioning limitation, there exists another limitation here $|A_i| = 1$. This extra limitation makes this method inefficient in dealing clusters with unequal sizes.

The symmetric positive definite matrix *Ai* is defined using Fuzzy scatter matrix, which is formulated below:

$$S_i = \sum_{j=1}^{N} U_{ij}^m (x^j - \omega_i)(x^j - \omega_i)^T \tag{7}$$

$$A_i = \sqrt[n]{|S_i|} \times S_i \tag{8}$$

After calculating the derivatives of equation (6), the updating rules for GK clustering are obtained as:

$$U_{ij} = \frac{1}{\sum_{L=1}^{c} \left(\frac{d_{A_i}(x^j, \omega_i)}{d_{A_i}(x^j, \omega_L)} \right)^{\frac{1}{m-1}}} \qquad m > 1 \tag{9}$$

$$\omega_i = \mu_i = \left(\frac{\sum_{j=1}^{N} U_{ij}^m x_j}{\sum_{j=1}^{N} U_{ij}^m} \right) \tag{10}$$

3. Proposed Algorithm

Clustering different type of clusters (Mass, linear, shell type) needs different kind of dissimilarity functions. When there is no prior knowledge about type and shape of clusters, it is important to use a flexible dissimilarity function between them. Some methods, like GK, have a dissimilarity function with limited flexibility but when the clusters have different sizes, it doesn't work out very well.

In this work an algorithm which is proposed earlier [26] has been modified and become more applicable by reducing its computational burden. The original algorithm which is proposed in our previous work has flexible dissimilarity function. This function is a weighted sum of conventional dissimilarity functions of mass, linear, and shell type fuzzy clustering. During the clustering, the weights of these functions change with respect to the degree that a cluster belongs to each type. Moreover, a soft switching mechanism is applied between each type of dissimilarity function to estimate the proper metric of dissimilarity. In addition, the prototype of each cluster is a combination of different prototypes. The algorithm is applicable to both data including or excluding partially labeled data and is presented in the following, but whenever the labeled data are used, the substitutive step for the latter group is performed.

1) Perform an initial clustering, for example FCM clustering.

2) Consider the U_{ij} s corresponding to labeled data equal to 1 and 0 otherwise. Now find a mass prototype for each cluster according to achieved U_{ij} s.

3) Compute fuzzy scatter matrix using U_{ij} s, and find a linear prototype for each cluster.

4) Using μ_i s and data points, which are computed in first step find a radius about μ_i and find a shell prototype for each cluster accordingly.

5) Form the scatter matrix (SM) for partially labeled data if the ratio of largest eigenvalue to the smallest one, denoted by *Rat,* was smaller than a threshold then consider the initial value of distance weight to mass prototype relatively small.

If there weren't any partially labeled data, then after step 1, form the fuzzy SM (FSM). Perform step 5 for FSM matrix.

6) As mentioned above, the dissimilarity criterion is a weighted sum of distance to different possible prototypes:

$$Dist\ (x^j, \omega_i) = a_{i1} dist_M\ (x^j, \omega_{imass}) + a_{i2} dist_L (x^j, \omega_{ilinear}) + a_{i3} dist_S (x^j, \omega_{ishell}) \quad (11)$$

in which $a_{i1} + a_{i2} + a_{i3} = 1$.

If *Rat* was smaller than a threshold, the cluster would have mass type. If *Rat* was bigger than another threshold, the cluster would have linear type.

7) After computing the final distance of each datum to each cluster, which is the linear combination of its distances to cluster prototypes, U_{ij} s will be computed using equation 12.

$$U_{ij} = \frac{1}{\sum_{L=1}^{c} \left(\frac{Dist\ (x^j, \omega_i)}{Dist\ (x^j, \omega_L)} \right)^{\frac{1}{m-1}}} \quad (12)$$

$m > 1$

8) Then a_1 , a_2 , a_3 , which are coefficients of dissimilarity metric, are updated according to the following formula.

$$a_{i1}(t+1) = a_{i1}(t) + \eta \times a_{i1}(t) U_{ij}(t) \times \left(\frac{MLSd - dist_M\ (x^j, \omega_{imass})}{Sd} \right) \quad (13)$$

in which;

$$MLSd = \frac{\left(dist_L{}^*(x^j, \omega_{ilinear}) + dist_S (x^j, \omega_{ishell}) \right)}{2} \quad (14)$$

$$Sd = dist_M\ (x^j, \omega_{imass}) + dist_L{}^*(x^j, \omega_{ilinear}) + dist_S (x^j, \omega_{ishell}) \quad (15)$$

$$\eta = \frac{1}{N} \quad (16)$$

The formula (13) is the updating rule for a_{i1} weight and the ones for a_{i2} and a_{i3} can be achieved easily substituting the numerator terms. $dist_L^*(x^j, \omega_{ilinear})$ is defined in 2.1.

In order to adjust the sum of a_{ij} for *ith* cluster equal to one according to equation (7) description, below correction is necessary.

$$a_{ij}(t+1) = a_{ij}(t+1) + h_i$$

and

$$h_i = k_i \times \left(\frac{a_{ij}(t+1)}{\sum_{j=1}^{3} a_{ij}(t+1)} \right) \quad or \quad \frac{k_i}{3}$$

$$k_i = 1 - \sum_{i=1}^{C} a_{ij}(t+1)$$

9) Different prototypes are updated then according to achieved U_{ij} s. This updating is similar to FCM method for mass type, FCL method for linear type, and FCS method for shell type. Steps 6 to 9 are performed iteratively to achieve the appropriate solution.

The grade of possibility of each cluster type or in the other hand, degree of membership of each cluster to mass, linear or shell type model is obtained through the following formulas:

$$\begin{aligned} M_{mass}(i) &= a_{i1} / (a_{i1} + a_{i2} + a_{i3}) = a_{i1} \\ M_{linear}(i) &= a_{i2} / (a_{i1} + a_{i2} + a_{i3}) = a_{i2} \\ M_{shell}(i) &= a_{i3} / (a_{i1} + a_{i2} + a_{i3}) = a_{i3} \end{aligned} \quad (17)$$

3.1. Distance Correction Coefficient for Different Prototypes

The distance to a linear prototype, as shown in Figure 1, is less than the distance to a mass prototype even if the cluster has the mass form. This problem can be solved, using a coefficient in distance formula.

Data on the surface, which meets the sphere centre and is perpendicular to linear prototype direction, have equal Euclidean distance from the both mass and linear prototypes; but, other points inside the sphere have less distance to supposed linear prototype. We have:

$$dist_M{}^2 = dist_L{}^2 + r'^2 ; \quad 0 \le r' \le R \quad (18)$$

Where, R is the sphere radius, $dist_M$ is distance from mass prototype, and $dist_L$ is distance from linear prototype.

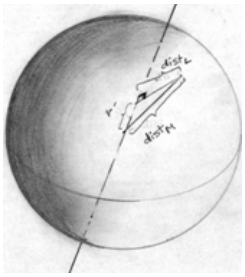


Figure 1. Mass-type cluster model with supposed prototypes

Now, in order to avoid error in specialized contributions to different prototypes, coefficient K corrects the distance in the following manner:

$$dist_L^* = \sqrt{dist_L{}^2 + K} \quad (19)$$

For instance, K for a uniform data distribution in 3D feature space is calculated as below:

$$K = (\frac{1}{\frac{4}{3}\pi R^3})2\int_0^R \pi r^2(R^2 - r^2)dr = R^2/5 \quad (20)$$

More discussion about correction coefficient is presented in Appendix.

It is noticeable to state that R must not be the maximum distance of data to the sphere centre, which affects the robustness of the method; It should define the mass effective radius, for example for a Gaussian distribution, σ can be the effective radius where σ is the variance of data in mass cluster.

3.2. A Truncation-Attention Mechanism

The mentioned algorithm in section 3.1. has significant results which addressed in [26]. The most critical problem in this connection is the higher computational cost of the method.

In order to assess about the computational cost of the proposed algorithm, we have run the algorithm ten times by use of Profiler of MATLAB 2006a (Dual CPU: 2.16, 1GB RAM). The MATLAB profiler analysis shows that about %80 of the computation is done in computing of the different distances.

In order to decrease the computational complexity of the algorithm a truncation or attention mechanism must be designed and applied to improve the mentioned clustering method. With respect to the relatively high computational cost in calculation of distances to different prototypes which is depicted in Table 1, we design a simple attention mechanism which focuses on a subset of possible metrics for each cluster, and enforces the grade of possibility of other metrics to the zero. In other words, this mechanism depletes the grade of possibility of the low possible cluster types.

The focusing mechanism is triggered when one of the grades of possibility of types of clusters becomes so bigger than others and increased in a relatively wide time window. This mechanism aggrandizes the update rate for the most possible cluster type, and decreases the weights of other metrics.

Table 1. Computational cost analysis of the proposed algorithm using Matlab profiler

Operation	Mean Percentage of Computational Cost (based on CPU time)
Computing Distances to the Linear Prototypes	%44
Computing Distances to Shell Prototypes	%19
Computing Distances to Mass Prototypes	%17
Others	%20

If mentioned mechanism is enabled during the whole clustering procedure, it should result in an inappropriate computation of the cluster prototypes which consequently leads to a weak clustering performance. To prevail over this obstacle the attention mechanism must be enabled after a duration in which the clustering algorithm has been working. The amount of this phase of inactivity is directly related to the dataset and the number of clusters and their between distances.

By activating the simple attention mechanism after less than half of clustering epochs and enforce the small aij to zero, in average the 80% of remaining computational costs reduced to less than 55%. So the computational cost of the proposed clustering method reduced to less than 75% of its cost before use of attention mechanism.

4. Experimental Results

The proposed algorithm is implemented for clustering four different datasets. One of the datasets has different clusters of mass, linear, and shell type. Figure 2 shows the normalized data. This dataset has three groups of data including two groups with Gaussian distribution (+ linear-type cluster and Δ mass-type cluster) and a group with uniform distribution (**o** shell-type cluster).

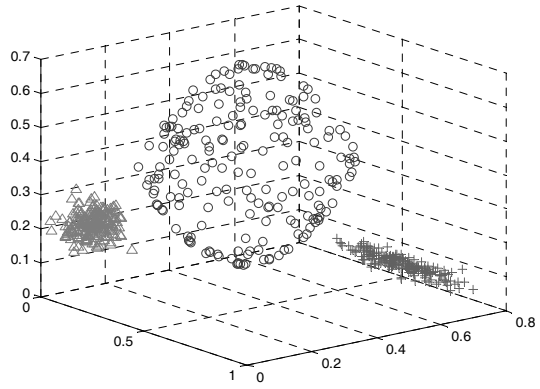


Figure 2. 3D data in the normalized form

The proposed algorithm is applied to this dataset and is compared with several well known clustering algorithms. The results are illustrated in the following figures (Figure 3, Figure 4, and Figure 5). For quantitative comparisons between mentioned methods, some clustering validity indices are utilized. Two of them involve only the U matrix and indicate the degree of crispness of each clustering. These indices are Partition Coefficient [27], and Partition Entropy [28].

Partition Coefficient is described below

$$PC = \frac{\sum_{i=1}^{C}\sum_{j=1}^{N} U_{ij}^{\ 2}}{|X|} \quad (21)$$

$$|X| = N$$

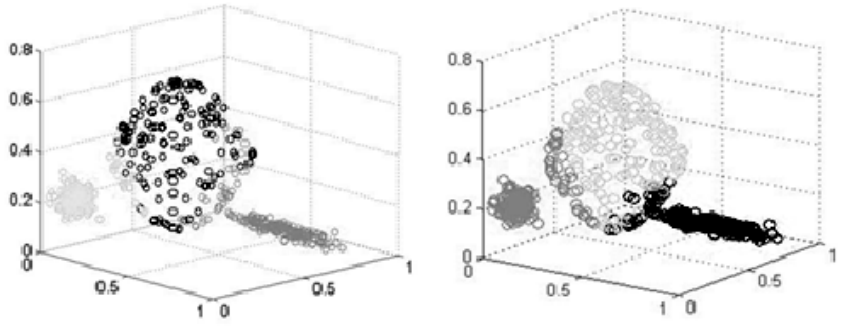


Figure 3. Clustered data, using gustafson-kessel clustering (on the left) and fuzzy c-linear clustering (on the right).

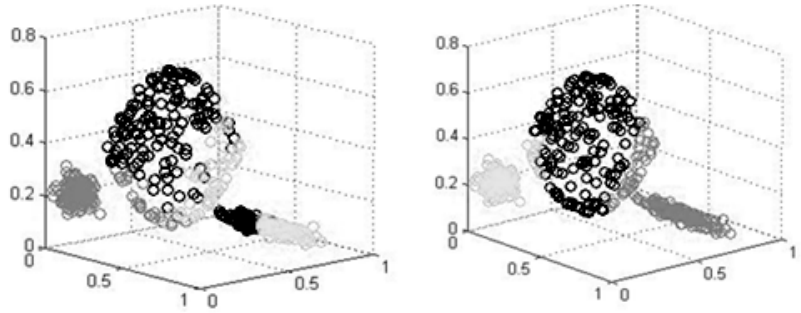


Figure 4. Clustered data, using fuzzy c-spherical clustering (on the left) and fuzzy c-means clustering (on the right).

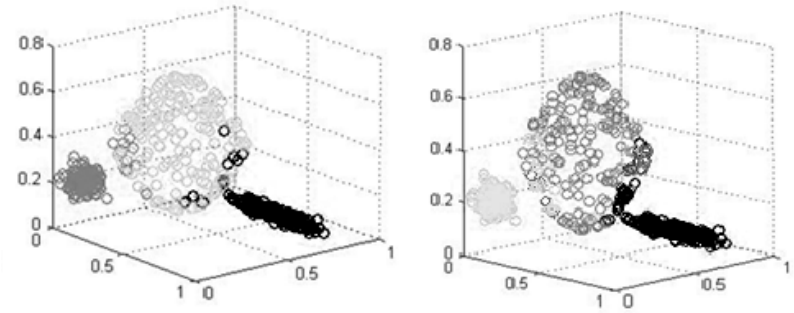


Figure 5. Clustered data, using proposed algorithm with partially labeled observations (on the left) and proposed algorithm with partially labeled observations and attention mechanism (on the right).

Table 2. Quantitative comparisons between different clustering methods

Algorithm / **validity index**	**Cluster validity Index**	Fuzzy C-Linear linear prototype	Fuzzy C-spherical shell prototype	Fuzzy c-means	Gustafson-Kessel	Proposed Algorithm NO P.L.O.	Proposed Algorithm (P.L.O.)	Proposed Algorithm (P.L.O.) and Attention
Expected Value	PC	0.79075	0.8898	0.78106	0.81736	0.6688	0.68419	0.6923
	PE	0.38505	0.2030	0.40154	0.34452	0.35675	0.34349	0.3127
	D	0.385138	0.6106	1.27814	0.0871	0.786775	0.77038	0.6647
	S	906.2569	137.71	89.0652	858.676	67.1291	62.4424	72.8455
Standard Deviation	PC	5.77E-05	0.0003	0.00059	0.00611	0.015455	0.07442	0.08131
	PE	0.000404	0.0021	0.00081	0.012213	0.022689	0.06267	0.09245
	D	0.295625	0.0124	0.01201	0.015224	0.135964	0.18476	0.2017
	S	1687.299	3.9446	1.33235	121.9833	18.77557	29.1823	29.5403

It is obvious that (1/C)<PC<1 and bigger PC is equivalent to crisper clustering.

Partition Entropy is described as:

$$PE = \frac{-\sum_{i=1}^{C}\sum_{j=1}^{N} U_{ij}\ Ln(U_{ij})}{|X|} \qquad (22)$$
$$|X| = N \qquad 0 \le PE \le \ln(c)$$

the smaller is PE, the crisper is the clustering performance.

Two other applied indices also depend on clusters' between distances.

Separation D, which is described below:

$$D = \frac{\min_{i=1}^{C}\ \min_{(i\neq j)\ j=1}^{C} dist_{\min}(\omega_i, \omega_j)}{\max_{l=1}^{C} dist_{\max}(\omega_l, \omega_l)} \qquad (23)$$

the greater is D, the more valid is the clustering.

Separation S, which is described below:

$$S = \frac{\frac{1}{C}\sum_{j=1}^{C}\sum_{i=1}^{C} U_{ij}^{\ 2} dist_{ij}^{\ 2}}{\min_{i=1}^{C}\ \min_{(i\neq j)\ j=1}^{C} dist_{\min}(\omega_i, \omega_j)} \qquad (24)$$

the smaller is S, the more valid is the clustering.

Each of the mentioned algorithms is executed 30 times on the dataset and the results are provided in Table 2.

As can be deduced from Table 2, the proposed method, especially in partially labeled observations case (P.L.O.), and also the FCM method have succeeded more than the others. It is noticeable that the GK powerful method performs weakly due to presence of different cluster sizes. The proposed method also has resulted in satisfying separation S validity, while FCM has led into good separation D validity. Both algorithms do not much differ in PE and PC. The later indices, PE and PC are not individually support the quality of clustering and just describe the crispness of clustering. Since the indices in standard deviation without P.L.Os decreases, it can be deduced that initializing a_1 , a_2 , a_3 should be improved. Moreover, it is noticeable that the utilized simple attention mechanism does not significantly decrease the clustering performance of the proposed method while reduces the CPU run time to less than 80% of its primary form.

Also mentioned clustering methods are applied to the wine dataset. The proposed method can implemented for all 13 dimensions of data, but in this work for subsiding the cost of computations and achieving a better representation the dimension is reduced to three features by use of Local Fisher Discriminant Analysis (LFDA) method. LFDA is proposed by combining the ideas behind Fisher Discriminant Analysis (FDA) and Locality-preserving projection (LPP) methods [29]. This method is capable to be extended to non-linear dimensionally reduction method by use of the kernel trick [29].The dimension reduction result by use of LFDA with *orthonormalized* metric is depicted in Figure 6.

After reducing the number of features to three the proposed clustering algorithm is applied to cluster the resulted data, and is compared with several clustering algorithms. The results are presented in the following figures (Figure 7, Figure 8, Figure 9, and Figure 10).

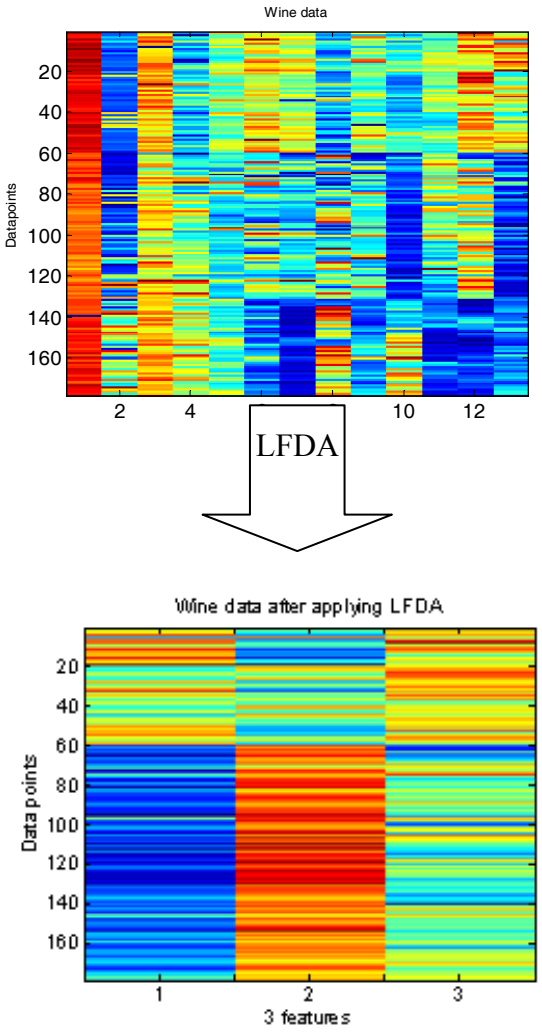


Figure 6. Feature reduction by use of LFDA

Each of the mentioned algorithms is executed 30 times on the dataset and the results are provided in Table 3.

These experimental results show that although Fuzzy C-means leads to rather good performance indices, the proposed algorithm results in better separation D measure. In addition, it is revealed that utilizing attention or truncation mechanism to reduce the computational burden of proposed algorithm does not significantly worsen the performance of the proposed algorithm.

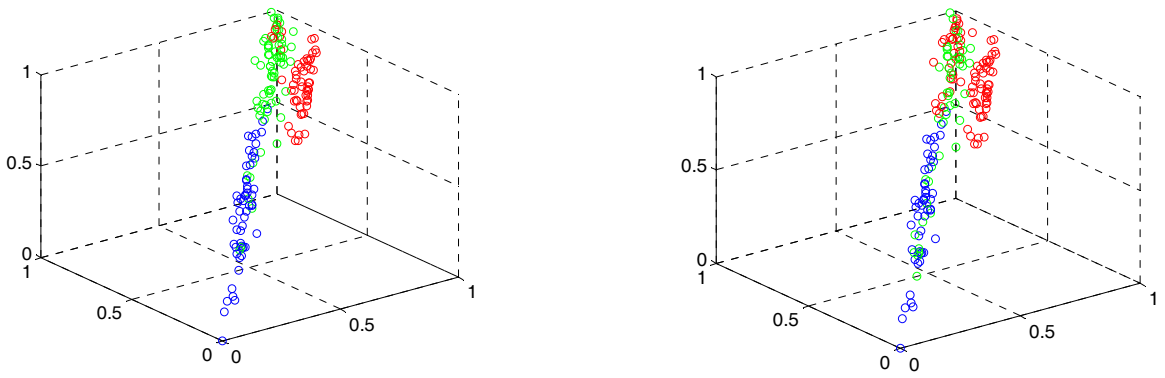


Figure 7. Clustered data, using fuzzy c-means clustering (on the left) and fuzzy mass type clustering (on the right).

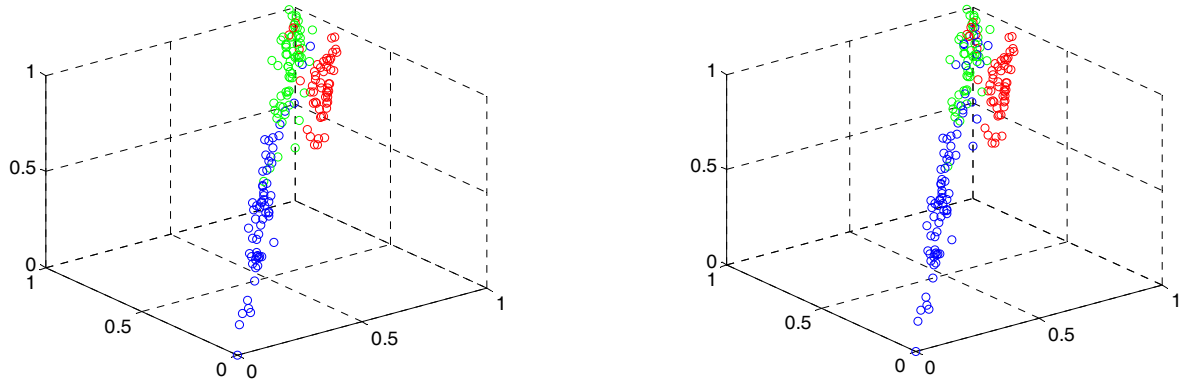


Figure 8. Clustered data, using gustafson-kessel clustering (on the left) and fuzzy c-linear clustering (on the right).

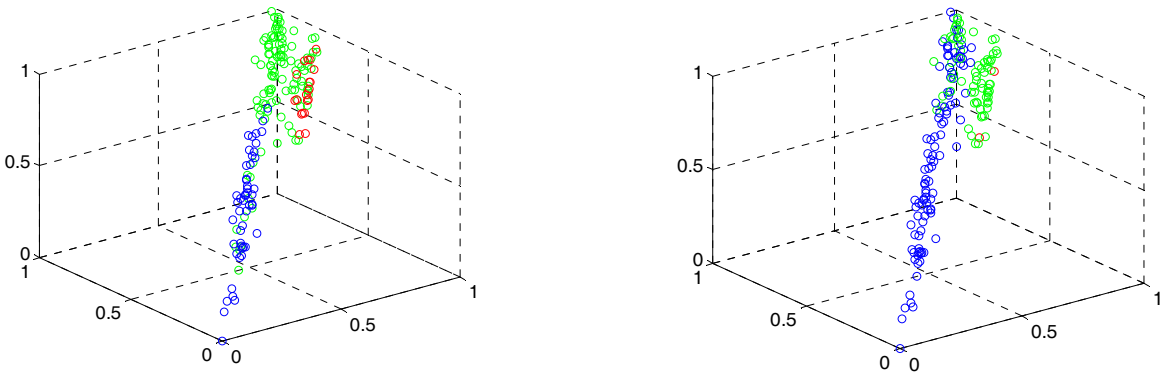


Figure 9. Clustered data, using proposed clustering without partial labeled data (on the left) and proposed clustering with partial labeled data (on the right).

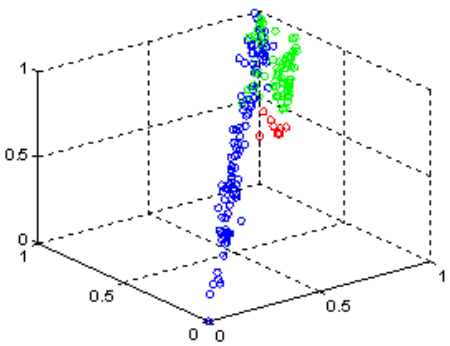


Figure 10. Clustered data, using proposed clustering with partial labeled data and the truncation mechanism.

Table 3. Quantitative comparisons between different clustering methods

Algorithm / validity index	Cluster validity Index	Fuzzy C-Linear linear prototype	Fuzzy c-means	Gustafson-Kessel	Proposed Algorithm NO P.L.O.	Proposed Algorithm (P.L.O.)	Proposed Algorithm (P.L.O.) and Attention
Expected Value	PC	0.76454	0.76476	0.73663	0.360722	0.2997	0.3911
	PE	0.4283	0.4279	0.4963	0.511322	0.665767	0.5326
	D	0.34582	0.50342	0.091	1.254689	0.682633	0.6539
	S	16.7062	8.65314	50.2905	13.74024	10.9249	10.2347
Standard Deviation	PC	8.94E-05	0.000261	5.77E-05	0.135774	0.046701	0.09125
	PE	7.07E-05	0.0003	4.02E-05	0.091954	0.03455	0.08802
	D	0.17646	0.014681	0.00026	0.757172	0.464618	0.70341
	S	13.8880	0.182859	0.17031	15.83105	2.426412	6.9125

Table 4. Quantitative comparisons between different clustering methods

Algorithm / validity index	Cluster validity Index	Fuzzy C-Linear	Fuzzy C-means	Gustafson-Kessel	Fuzzy C-Spherical	Proposed Algorithm (P.L.O.)	Proposed Algorithm (P.L.O.) and Attention
Expected Value	PC	0.67017	0.68876	0.73218	0.80337	0.61716	0.54177
	PE	0.57073	0.56176	0.43127	0.35782	0.44837	0.52168
	D	0.16973	0.49696	0.64677	0.93958	1.26451	0.97676
	S	111.0437	22.72208	58.3828	18.9904	14.04098	16.47594
Standard Deviation	PC	0.04417	0.00734	0.06005	0.00211	0.04192	0.08486
	PE	0.02903	0.01561	0.07178	0.00269	0.04556	0.06810
	D	0.12888	0.03545	0.25783	0.03103	0.38382	0.36169
	S	118.8733	23.76372	62.83825	0.68080	9.96279	7.08683

Table 5. Quantitative comparisons between different clustering methods

Algorithm / validity index	Cluster validity Index	Fuzzy C-Linear	Fuzzy C-means	Gustafson-Kessel	Fuzzy C-Spherical	Proposed Algorithm (P.L.O.)	Proposed Algorithm (P.L.O.) and Attention
Expected Value	PC	0.80873	0.83048	0.812103	0.60810	0.5201	0.6991
	PE	0.37287	0.32386	0.37651	0.64287	0.26257	0.15895
	D	0.04593	0.69642	0.04923	0.17420	0.50337	0.26025
	S	4848.3667	253.63066	495.2205	15550.6333	503.15693	489.23667
Standard Deviation	PC	0.01761	0.0261264	0.04174	0.27069	0.28479	0.00226
	PE	0.03389	0.05291	0.04125	0.44683	0.17594	0.00078
	D	0.06268	0.3531167	0.02977	0.30146	0.49469	0.36607
	S	2154.947	56.02330	42.63201	20743.9965	23.89234	213.03523

The results of implementation of the proposed method with different clustering algorithms on standard Iris data and Abalone data are also compared in Tables 4 and 5 respectively. It must be noticed that each algorithm is executed 30 times on each data set.

By analyzing the results of clustering Iris data after LFDA transformation, depicted in Table 4, it reveals that fuzzy C-Spherical clustering results in the crispest clustering.

However, the proposed clustering algorithm achieved the best separation due to the separation S and D indices. In

addition, it is clear that the attention mechanism does not deteriorate the results of proposed algorithm.

The experimental results of clustering of the LFDA transformed Abalone data -without considering the first feature of the Abalones-, demonstrate that fuzzy C-means algorithm results in better PC index, while the proposed method results in better PE index. In case of separation indices, fuzzy C-means and the proposed algorithm result in better separation D index, however the separation D index of former method is significantly better. Also, it is clear that because of unbalance size (volume) of clusters GK does not result in satisfying values for mentioned indices.

5. Conclusion

This paper presented a new approach to fuzzy clustering, in which during learning, a model for each cluster is estimated. Dissimilarity metric for each cluster is defined, updated and used for the next step. Its strength in dealing with clusters of different type and size is the most important advantage of this method. To ameliorate the computational cost of this algorithm a truncation mechanism which can be expressed as a controller of attention is designed and added to the mentioned clustering algorithm. Proposed clustering method has the capability to deal with partial labeled data as well as fully unlabeled data. This method is implemented on two families of data, first in presence of partially labeled data (10% of data are labeled) and second, with fully unlabeled data. Comparing the experimental results of this method with several important existing algorithms verified its succession both in achieving satisfying values of clustering indices and to estimating each cluster shape. Comparing with different pattern recognition methods which convert the feature space into a space with more dimensions, the proposed method has the capability of computing a fuzzy membership value to different shapes for each cluster in its basic feature space. The simple attention mechanism enforces the fuzzy shape of the clusters to crisp ones while decreases the computational costs. The mentioned capabilities of the proposed clustering algorithm make it useful in shape recognition tasks, or in the problems in which the meaningful quality of features are important, and we want to evade the use of unexpressive combined features.

Acknowledgment

The authors would like to acknowledge anonymous reviewers for their helpful comments. The first author also would like to extend his appreciation to Ahmad Ashoori for useful discussions and comments.

References

[1] S. Theodoridis, and K. Koutroumbas, *Pattern Recognition*, third edition, Elsevier Academic Press, 2006.

[2] M. R. Anderberg, *Cluster Analysis for Applications*, Academic Press, 1973.

[3] H. Frigui, and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Trans. Pattern Analysis and Mach. Intell.*, Vol 21, No. 5, pp. 450-465, 1999.

[4] P. Turaga, A. Veeraraghavana, and R. Chellappa, "Unsupervised View and Rate Invariant Clustering of Video Sequences," *Computer Vision and Image Understanding*, Vol. 113, No. 3, pp. 353-371, 2009.

[5] B. Raytchev, and H. Murase, "Unsupervised Recognition of Multi-View Face Sequences Based on Pairwise Clustering Term with Attraction and Repulsion," *Computer Vision and Image Understanding*, Vol. 91, No. 1-2, pp. 22-52, 2003.

[6] M. R. Rezaee, P. M. J. Van Der Zwet, B. P. F. Lelieveldt, R. J. Van Der Geest, and J. H. C. Reiber, "A Multiresolution Image Segmentation Technique Based on Pyramidal Segmentation and Fuzzy Clustering," *IEEE Trans. Image Processing*, Vol. 9, No. 7, pp. 1238-1248, 2000.

[7] Z. M. Wang, Y. C. Soh, Q. Song, and K. Sim, "Adaptive Spatial Information-Theoretic Clustering Term for Image Segmentation," *Pattern Recognition*, Vol. 42, No. 9, pp. 2029-2044, 2009.

[8] A. Kaarna, P. Zemcik, H. Kalviainen, and J. Parkkinen, "Compression of Multispectral Remote Sensing Images Using Clustering and Spectral Reduction," *IEEE Trans. Geoscience and Remote Sensing*, Vol. 38, No. 2, pp. 1073-1082, 2000.

[9] A. Aiyer, K. (P.) Pyun, Y. Z. Huang, D. B. O'Brien, and R. M. Gray, "Lloyd Clustering of Gauss Mixture Models for Image Compression and Classification," *Signal Processing: Image Communication*, Vol. 20, No. 5, pp. 459-485, 2005.

[10] B. Bhanu, and A. Dong, "Concepts Learning with Fuzzy Clustering and Relevance Feedback," *Engineering Applications of Artificial Intelligence*, Vol. 15, No. 2, pp. 123-138, 2002.

[11] J. C. Dunn, "A Fuzzy Relative to the ISODATA Process and Its Use in Detecting Compact, Well-Separated Clusters," *J. Cybernet*,.Vol 3, pp. 32-57, 1974.

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

[13] R. J. Hathaway, and J. C. Bezdek, "NERF C-Means: Non Euclidean Relational Fuzzy Clustering Algorithms," *Pattern Recognition*, Vol. 27, pp. 429-437, 1994.

[14] P. J. F. Groenen, and K. Jajuga, "Fuzzy Clustering with Squared Minkovsky Distances," *Fuzzy Sets and Systems*, Vol. 120, pp. 227-237, 2001.

[15] M. S. Yang, "A Survey of Fuzzy Clustering," *Math. Computing. Modeling.*, Vol. 18, No. 11, pp. 1-16, 1993.

[16] D. E. Gustafson, and W. Kessel, "Fuzzy Clustering with Fuzzy Covariance Matrix," *Proc, IEEE conf. Decision Contr.*, pp. 761-766, 1979.

[17] R. Krishnapuram, and J. Kim, "A Note on the Gustafson-Kessel and Adaptive Fuzzy Clustering Algorithms," *IEEE Trans. Fuzzy Systems*, Vol. 7, No. 4, pp. 453-461, 1999.

[18] R. Babuska, P. J. van der veen, and U. Kaymak, "Improved Covariance Estimation for Gustafson-Kessel Clustering," *Proc, IEEE Int. Conf. on Fuzzy Systems*, Vol. 2, pp. 1081-1085, 2002.

[19] I. Gath, and A. B. Geva, "Unsupervised Optimal Fuzzy Clustering," *IEEE Trans. Pattern Analysis and Mach. Intell.*, Vol. 7, pp. 773-781, 1989.

[20] F. A. T. Carvalho, C. P. Tenório, and N. L. Cavalcanti Junior, "Partitional Fuzzy Clustering Methods Based on Adaptive Quadratic Distances," *Fuzzy Sets and Systems*, Vol. 157, pp. 2833-2857, 2006.

[21] A. Bouchachia, and W. Pedrycz, "Enhancement of Fuzzy Clustering by Mechanism of Partial Supervision," *Fuzzy Sets and Systems*, Vol. 157, pp. 1733-1759, 2006.

[22] J. Liu, and M. Xu, "Kernelized Fuzzy Attribute C-Means Clustering Algorithm," *Fuzzy Sets and Systems*, Vol. 159, pp. 2428–2445, 2008.

[23] W. C. Tjhi, and L. Chen, "A Heuristic-Based Fuzzy Co-Clustering Algorithm for Categorization of High-Dimensional Data," *Fuzzy Sets and Systems*, Vol. 159, pp. 371–389, 2008.

[24] W. Cai, S. Chen, and D. Zhang, "Fast and Robust Fuzzy C-Means Clustering Algorithms Incorporating Local Information for Image Segmentation," *Pattern Recognition*, Vol. 40, pp. 835–838, 2007.

[25] Q. Song, "A robust information clustering algorithm," *Neural Comput.*, Vol. 17, No. 12, pp. 2672-2698, 2005.

[26] A. Arami, and B. Nadjar Araabi, "A Clustering Method Based on Soft Learning of Model (Prototype) and Dissimilarity Metrics," *Proc, 13th Int. CSI Computer Conf.*, pp. 33-40, 2008.

[27] J. C. Bezdek, "Cluster Validity with Fuzzy Sets," *Journal of Cybernetics*, Vol. 3, No. 3, pp. 58-72, 1974.

[28] J. C. Bezdek, "Mathematical Models for Systematics and Taxonomy," *Proc, 8th Int. Cunj in Numerical Tuxunumy*, pp. 143-166, 1975.

[29] M. Sugiyama, "Dimensionality Reduction of Multi-Modal Labeled Data by Local Fisher Discriminant Analysis," Journal of Machine Learning Research, Vol. 8, pp. 1027-1061, 2007.

Appendix

More Discussion About Distance Correction Coefficient for Linear Type Distance Part

Assuming that each mass of data in n-dimension feature space consist of infinite number of n-1 dimensional surfaces of data. Paying attention to Figure 11, without losing any generality and for ease of description, assume a 3D space and a spherical mass data with maximum distance from its prototype equal to R_{max}. With respect of distribution of data on this mass an effective radius of sphere can be chosen which is denoted by R. By slicing the mass perpendicular to the assumed linear prototype direction, infinite number of surfaces are achieved which drawn by dotted lines.

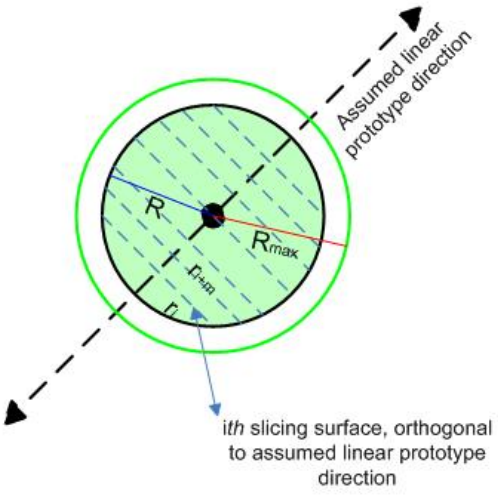


Figure 11. 2D cut of 3D data and infinite slicing surfaces

Using 3-5 and 3-6 for each surface, the correction constant for each datum on each surface easily calculated as below;

$$K_{ij} = R_{ij}^{\ 2} - r_i^{\ 2} \tag{25}$$

r_i is the ith surface radius and $R_{i,j}$ is the jth datum in ith surface distance from mass prototype.

By assuming that the number of data in each surface is a function of data distribution and area of surface, the constant for all data in each surface calculated as below;

$$K_i = \mathrm{F}(\text{distribution - of - data in mass, area of surface } i) \times (R^2 - r_i^2) \quad (26)$$

If there is no information of distribution the effect of this parameter could be neglected or the distribution can be assumed as Gaussian.

And the coefficient for total mass is computed as:

$$K = (1/(\text{mass volume})) \times \sum_{all\ i | r_i < R} K_i \quad (27)$$

For example with assuming the uniform distribution of data in 3D feature space, K can be calculated as fallow;

$$K = (\frac{1}{(4/3)\pi R^3}) 2 \int_0^R \pi r^2 (R^2 - r^2) dr = \frac{R^2}{5} \quad (28)$$

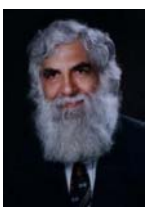
Arash Arami was born in 1983 in Tehran. He received his B.Sc. from University of Tabriz, and his M.Sc. from University of Tehran in 2006, and 2009, respectively, both in Electrical Engineering. He is currently a Ph.D. student of Electrical Engineering in Ecole Polytechnique Fédérale de Lausanne (EPFL) in Switzerland. He is also a doctoral assistant in Laboratory of Movement Analysis and Measurement (LMAM). His research interest includes: Applied Signal Processing, Pattern Recognition, Machine Learning, Intelligent Systems and Intelligent Control.

E-mail: a.arami@ece.ut.ac.ir

Babak N. Araabi received the B.S. degree from Sharif University of Technology, Tehran, Iran, the M.S. degree from University of Tehran, Iran, and the Ph.D. degree from Texas A&M University, TX, USA, in 1992, 1996, and 2001, respectively, all in electrical engineering. In January 2002, he joined the School of Electrical and Computer Engineering, University of Tehran, where he is an associate professor and head of control systems division. He is also a senior researcher at School of Cognitive Sciences, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran.

Dr. Araabi is the author or co-author of more than 50 international journal papers in his research areas, which include machine learning, pattern recognition, neuro-fuzzy modeling, prediction, and system identification.

E-mail: araabi@ut.ac.ir



Caro Lucas received the M.Sc. degree from the University of Tehran, Tehran, Iran, in 1973, and the Ph.D. degree from the University of California, Berkeley, in 1976.

He is a Professor with the Centre of Excellence for Control and Intelligent Processing, Faculty of Electrical and Computer Engineering, University of Tehran, as well as a Researcher with the School of Intelligent Systems (SIS), Institute for Studies in Theoretical Physics and Mathematics, Tehran.

He has served as the Director of SIS (1993–1997), Chairman of the Department of Electrical and Computer Engineering, University of Tehran (1986–1988), Managing Editor of Memories of the Engineering Faculty of the University of Tehran (1979–1991), a Reviewer of Mathematical Reviewers (since 1987), an Associate Editor for the Journal of Intelligent and Fuzzy Systems (1992–1999), and Chairman of the IEEE, Iran Section (1990–1992). He was also a Visiting Associate Professor with the University of Toronto, Toronto, ON, Canada (1989–1990), and the University of California, Berkeley (1988–1989), an Assistant Professor with Garyounis University, Benghazi, Libya (1984–1985), and the University of California, Los Angeles (1975–1976), a Senior Researcher with the International Centre for Theoretical Physics and the International Centre for Genetic Engineering and Biotechnology, both in Trieste, Italy, the Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing, China, and Harbin Institute of Electrical Technology, Harbin, China, a Research Associate with the Manufacturing Research Corporation of Ontario, and a Research Assistant with the Electronic Research Laboratory, University of California, Berkeley. He is the holder of the patent on "Speaker independent Farsi isolated word neurorecognizer." His research interests include biological computing, computational intelligence, uncertain systems, intelligent control, neural networks, multiagent systems, data mining, business intelligence, financial modeling, and knowledge management.

Prof. Lucas has served as the Chairman of several international conferences. He was the founder of the SIS and has assisted in founding several new research organizations and engineering disciplines in Iran. He is the recipient of several research grants from the University of Tehran and SIS.

E-mail: lucas@ut.ac.ir

Majid Nili Ahmadabadi was born in 1967 and received his B.S from Sharif University of Technology of Iran in 1990. He received his M.Sc. and Ph.D. in Information Sciences from the Graduate School of Information Science, Tohoku University, Japan in 1994 and 1997 respectively. In 1997, he joined the Advanced Robotics Laboratory at Tohoku University. Later he moved to the School of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran where he is an associate professor and the head of Robotics and AI Group. He is also a senior researcher at School of Cognitive Sciences, Institute for Research in Fundamental Sciences (IPM), Iran.

In summers 2005 and 2008, Dr. Nili was with Autonomous System Laboratory at EPFL and ETHZ as an invited visiting professor. He is one of Distinguished Lecturers selected by IEEE Robotics and Automation Society for the years 2007 and 2008.

Dr. Nili's main research interest are learning in mulitagent systems, cognitive robotics, biologically inspired learning methods, distributed robotics, object manipulation, and mobile robots.

E-mail: mnili@ut.ac.ir

Paper Handling Data:

Submitted: 28.01.2009
Received in revised form: 19.09.2009
Accepted: 31.10.2009
Corresponding author: Arash Arami,
Control and Intelligent Processing Center of Excellence,
School of Electrical and Computer Engineering,
University of Tehran, Tehran, Iran.