

論文94-31B-7-22

퍼지 신경망을 이용한 온라인 클러스터링 방법

(A On-Line Pattern Clustering Technique Using Fuzzy Neural Networks)

金載顯*, 徐一弘*

(Jae-Hyun Kim and Il-Hong Suh)

要約

대부분의 클러스터링 알고리즘은 입력 데이터를 클러스터링 하고자 할 때 클러스터의 중심 벡터나 미리 정의해 둔 클러스터 모양을 이용하여 클러스터링을 수행한다. 그러나 실제의 데이터에 대한 아무런 선행 정보가 없을 경우 (클러스터 중심 벡터나 미리 정의된 클러스터 모양에 의해 적절히 표현될 수 없는 경우) 클러스터의 갯수 및 모양에 대한 추정이 불가능하기 때문에, 기존의 방법에 의해서는 입력 데이터를 적절한 클러스터로 소속시키기 어려웠다. 본 논문에서는 이러한 어려움을 극복하기 위하여 클러스터의 표현에 있어서 클러스터의 경계선을 의미하는 여러개의 부클러스터를 도입하였다. 입력 데이터가 각 부클러스터로 얼마나 소속되는지를 표현하기 위하여 퍼지 소속 함수를 사용하였으며, 이러한 입력에 대해 경쟁 학습을 수행하는 신경망을 사용하여 입력 데이터에 가장 가까운부클러스터의 위치를 클러스터의 확장 방향으로 조정하였다. 제안된 방법의 유효성을 보이기 위해 몇가지 수치 예를 제시하였고, 기존의 FMCMC 신경망 알고리즘을 비교 대상으로 하였다.

Abstract

Most of clustering methods usually employ a center or predefined shape of a cluster to assign the input data into the cluster. When there is no information about data set, it is impossible to predict how many clusters are to be or what shape clusters take. (the shape of clusters could not be easily represented by the center or predefined shape of clusters) Therefore, it is difficult to assign input data into a proper cluster using previous methods. In this paper, to overcome such a difficulty, a cluster is to be represented as a collection of several subclusters representing boundary of the cluster. And membership functions are used to represent how much input data belongs to subclusters. Then the position of the nearest subcluster is adaptively corrected for expansion of cluster, which the subcluster belongs to, by use of a competitive learning neural network. To show the validity of the proposed method, a numerical example is illustrated, where FMCMC(Fuzzy Min-Max Clustering) algorithm is compared with the proposed method.

*正會員, 漢陽大學校 電子工學科
(Dept. of Elec. Eng., Hanyang Univ.)

* 본 연구는 한국 학술진흥재단의 신진연구인력

지원에 의한 것임.

接受日字 : 1993年 11月 2日

1. 서론

패턴 분류(pattern classification)는 이미 소속 클래스(class)를 알고 있는 표본 패턴 데이터(sample pattern data)를 이용하여 클래스의 경계선을 형성하는 것이다. 이러한 패턴 분류와는 달리 패턴 클러스터링(pattern clustering)은 소속 표시가 있지 않은 패턴 데이터로 하여금 가장 어울리는 그룹에 속하도록 하는 것을 목적으로 한다. 어떤 입력 데이터가 주어진 경우, 하드 클러스터링(hard clustering)방법에서는 거리가 가장 가까운 클러스터에 소속도 1을 수여하지만, 퍼지 클러스터링 방법에서는 거리에 반비례한 함수(u)로써 소속도를 부여하게 된다. 따라서 퍼지 클러스터링이 입력 패턴 데이터와 각 클러스터들 사이의 관계를 더 세밀하게 표현할 수 있다. 이러한 퍼지 클러스터링을 수행하는 대부분의 방법들은^[2-7] Bezdek의 FCM(fuzzy c-means)이 기초가 되었다.

FCM류의 많은 알고리즘들^[2-7]은 각 패턴 데이터의 모든 클러스터에 대한 소속도의 합이 1이 되도록 하여야 하기 때문에 특정 클러스터의 소속도는 특정 클러스터와 입력값 사이의 거리와 특정 클러스터를 제외한 클러스터들과 입력값 사이의 거리를 고려하여 상대적으로 결정된다. 따라서 클러스터들의 소속도는 서로 영향을 줄 수 밖에 없었다. FCM류 알고리즘들의 문제점은 다음과 같다. 1) 동일 클러스터 내의 같은 거리에 있는 입력 값이라도 입력 값의 위치에 따라 다른 소속도를 나타내게 된다. 2) 노이즈(noise)가 입력되는 경우, 비록 모든 클러스터들로부터 거리는 상당히 떨어져있다 하더라도 노이즈에 대해 $1/n$ (n : 클러스터 갯수)에 가까운 부적당한 소속도를 나타내었다. 이러한 문제점을 해결하고자 제약 조건을 완화하여 각 특정 벡터의 모든 클러스터에 대한 소속도의 합이 0과 1사이라 간주하고 소속도를 구하는 알고리즘이 제안 되었으며, 이에선 Dave의 RFCA(Robust Fuzzy Clustering Aloritm)^[11]와 Krishnapuram, Keller의 PCM(possibilistic c-means)알고리즘^[8] 등이 속하게 된다. 완화된 제약으로부터 구한 소속도는 다른 클러스터로부터 어떠한 영향도 받지 않도록 구성되어 있다. 따라서 각 클러스터로부터 거리가 멀리 떨어진 노이즈에 대해서는 상대적으로 낮은 소속도를 나타내었으며, 클러스터로부터 동일 거리에 있는 입력 패턴에 대해서는 동일한 소속도를 출력할 수 있었다. 주어진 패턴 데이터를 n 개의 클래스에 속하게하는 분류에서는 FCM은 의미가 없지만 여러 클러스터의 패턴 데이터 공유

(sharing)라는 관점에서 보면 FCM은 나름대로의 타당성을 지니게 된다. 즉, 만일 주어진 패턴 데이터가 n 개의 클래스에 속하게 되며, noise 입력이 없다는 선행정보가 있는 경우에 FCM은 의미를 갖게 된다.

FCM이나 PCM에서는 소속되어질 클래스의 갯수를 알고있는 경우에 사용할 수 있으며, 한 개의 클러스터당 한개의 정형(prototype)을 갖는 클러스터링을 수행하였다. 그림 1(a)처럼 clover모양의 데이터 군을 생각해 보자. 그림 1(b)는 1개의 정형을 갖는 클러스터의 소속도의 등고선을 나타내었으며 클러스터에 속하는 입력 벡터 a와 클러스터에 속하지 않는 입력벡터 b가 같은 소속도를 갖는 모순이 발생됨을 알 수 있다. 이는 한 개의 정형을 갖는다는 클러스터링의 결과이다. 따라서 FCM이나 PCM으로 주어진 데이터 군을 정확히 표현하기 위해서는 여러 클러스터를 할당하여 그림 1(c)에서 처럼 구성하여야 한다. 따라서 c-shell clustering과 같이 특정 모양의 클러스터를 정의하고 클러스터링하고자 하는 방법도 도입되었다.

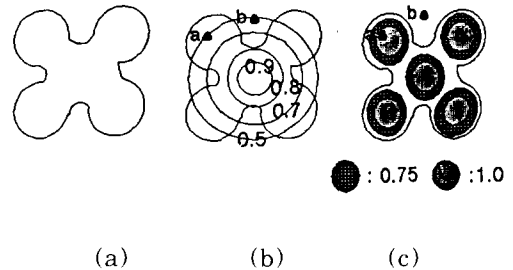


그림 1. membership 함수 분포도 (a) clover 클러스터 (b) 1개 정형 (c) 여러 개 정형

Fig. 1. The distribution map of membership value. (a) clover shapped cluster (b) only one cluster prototype (c) several cluster prototype.

그러나 초기에 필요한 클러스터의 갯수를 추정한다는 것이 매우 어려울 때, FCM, PCM, c-shell clustering과 같은 방법을 적용할 수 없다. 이러한 클러스터링 알고리즘들은 이미 클러스터의 갯수에 대한 선행 정보가 조건이 된다. Fuzzy min-max clustering (FMCC) 신경망 알고리즘^[9]은 이와 같은 어려운 점을 해결하기 위한 알고리즘들 중 하나이다. 즉, 클러스터 갯수에 대한 어떠한 정보도 알 수 없을 때, 필요한 경우 클러스터를 할당함으로써 이를

해결하고자 하였다. 또한 FMCC는 클러스터내의 모든 입력에 대해 소속도를 1로 하여 클러스터내의 모든 특징 벡터를 정형이라고 생각하는 접근 방법이다. FMCC 신경망 알고리즘의 특징은 다음과 같다. 1) 클러스터의 수가 제한되어 있지 않다. 2) H/W로 구현하기 쉽다. 3) On-line 처리가 가능하다. 4) 조정하여야 할 Parameter들의 수가 적다. 그러나 FMCC 신경망 알고리즘에서는 hyperbox라는 클러스터의 모양이 정해져 있고 최대 크기 또한 특정 parameter에 의해 제한되어 있다. hyperbox의 크기를 작게하면 할수록 좀 더 정확한 클러스터의 표현이 가능하다. 그러나 hyperbox의 크기가 작으면 작을수록, 많은 클러스터를 발생시키고, 비록 계층적인 클러스터링 수행중 가장 기본이 되는 클러스터링을 목적으로 한다 하더라도 특징 벡터 공간의 입력 패턴들과 별 차이가 없어지게 되며, 클러스터링의 의미가 상실된다. 따라서 클러스터의 최대 크기에 대한 제한은 가능하다면 없어지는 것이 좋다. 그러나 FMCC에서 클러스터의 최대 크기에 대한 제한이 없어진다면 최악의 경우 모든 클러스터링이 커다란 hyperbox 하나만으로 표현될 수 있다. 즉 hyperbox의 크기가 커지면 커질수록 그림 2에서 처럼 원래 클러스터의 부분이 아닌 영역(risk decision area)을 포함할 가능성이 커진다. (클러스터 밀집도가 떨어진다) FMCC는 주어진 데이터 군을 최대 크기에 대한 제한을 갖는 일정한 모양의 클러스터를 할당함으로써 encoding하고자 하였으나, 주어진 데이터 군에 대한 효율적으로 표현하지 못하였다. 따라서 계층적인 클러스터링을 수행하는 경우라도 클러스터의 모양과 크기에 대한 제약 조건은 없어지는 것이 좀 더 효율적인 접근 방법이 된다. 또한 FMCC 알고리즘은 n차 공간 내에서 모든 n개의 축에 대해 검침을 허용하지 않으므로, 클러스터 core를 제대로 표현하지 못할 수 있다. 이는 클러스터 수축에 의해 이전에 기억하여 둔 data의 상실을 의미한다.

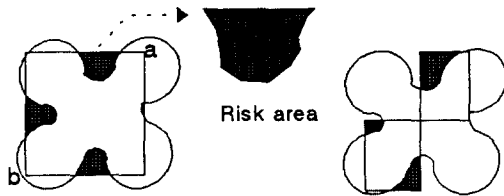


그림 2. hyperbox에 의한 risk decision area
Fig. 2. Risk decision area arising in hyperbox approach.

본 논문에서는 클러스터 자체에 대한 크기나 모양에 대한 제한을 두지않고 동적으로 클러스터를 할당할 때 FMCC에서 발생하였던 risk decision area를 최소화하여 클러스터의 경계선을 학습하는 알고리즘을 제시하고자 한다. 본 알고리즘에서는 부클러스터(sub-cluster)들에 의해 경계선을 나타내고, 경계선 내부의 모든 입력 값에 대해 소속도 1을 가지며 경계선 외부에서는 거리가 멀어짐에 따라 감소하는 비선형 소속도를 갖게 된다. 제시하고자 하는 알고리즘의 특징은 다음과 같다. 1) on-line 클러스터링 특성 2) 클러스터의 동적인(dynamic) 할당 2) 각 클러스터에 독립적인 소속도 할당 3) cluster들의 표현에 있어서 cluster의 경계선을 표현하는 sub-cluster 개념의 NN(neural network) 구현 등이다. 2장에서는 기본적인 클러스터 신경망(basic cluster neural network)과 알고리즘을 설명하였고 3장에서는 이러한 신경망의 확장 network에 대해 설명하였으며 4장에서는 전체 시스템의 성능을 평가할 수 있는 몇가지 예를 들었다. 마지막으로 5장에서 본 논문의 결론 및 향후 연구 과제를 다루었다.

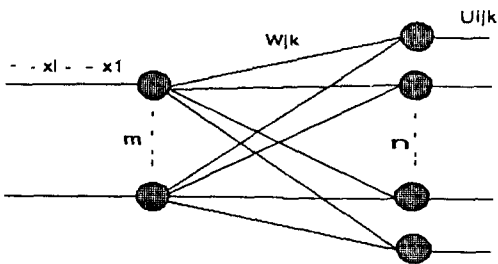
II. (Basic Clustering Neural Network) 알고리즘

입력 벡터들이 구성하는 클러스터의 경계는 명확한 것이 아니고 오히려 애매(fuzzy)하다는 근거에서 BCNN 알고리즘은 클러스터의 표현에 있어서 퍼지 집합 개념을 도입하였다. n차 초월 공간(hyperspace)에서 정의되는 m개의 입력 벡터 $x_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{in})$ 로 이루어진 퍼지 집합(C_j)를 생각할 수 있다. $C_j = \{0.2/x_1, 0.5/x_2, 0.7/x_3, 1.0/x_4, \dots, 0.2/x_m\}$ 클러스터를 퍼지 집합으로 표현하는 경우, 각각의 원소는 해당 클러스터로의 소속도 u_{ij} 를 갖게 된다. 어떤 입력 패턴들에 대해 이러한 퍼지 집합을 구성하기 위해서는 클러스터 모양을 결정하는 클러스터의 핵심(core)를 학습하여야 한다. 클러스터 핵심의 학습은 클러스터 핵심의 경계선을 학습하면 된다. 따라서 클러스터 핵심의 경계선 내부에 존재하는 입력 벡터에 대해서는 소속도 1을 부여하고, 클러스터 핵심의 경계선 외부에 존재하는 입력 벡터에 대해서는 거리(혹은 기준)에 따른 소속도를 부여하여 클러스터를 나타내는 퍼지 집합을 구성한다.

1. 기본적인 클러스터링 신경망(Basic Clustering Neural Network)

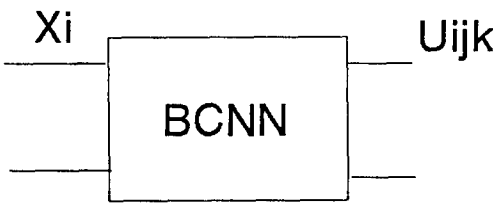
본 클러스터링 알고리즘에서 사용하는 신경망은 경

쟁 학습을 수행하게 된다. 이는 입력 패턴을 표준 패턴과 비교하여 표준 패턴과 더 가까울수록 더 큰 값을 출력하도록 학습시키는 것을 의미한다. 이러한 일반적인 신경망의 구성은 Kohonen의 feature mapping neural network에 기반을 둔 신경망이며, FMMC (fuzzy min-max clustering) 신경망도 이러한 신경망 구성을 이루고 있다. 본 논문에서 사용한 신경망은 이러한 부류의 신경망이며 순차적인 입력에 대해 클러스터링(clustering)을 수행하는 기본적인 신경망의 구조는 그림 3과 같다. 신경망의 구조는 입력을 받아들이는 입력 층과 클러스터의 경계 선을 나타내는 부클러스터 층으로 이루어진다. 이러한 단위 신경망 (Basic Clustering Neural Network)은 한 개의 클러스터를 의미하게 된다.



j : cluster vector k : sub-cluster vector
i : input vector

(a)



(b)

그림 3. 기본적인 클러스터링 신경망(BCNN)
(a) BCNN 구조 (b) 논리적인 표현

Fig. 3. Basic Clustering Neural Network
(a) network configuration of BCNN
(b) logical expression of BCNN.

그림 3에서 패턴 x_i 가 입력되면 모든 클러스터 (cluster)의 부클러스터(sub-cluster)들 중 거리가 가장 가까운 부클러스터와 클러스터가 결정되고 입력

벡터에 대한 부클러스터 벡터의 조정은 클러스터가 확장하는 방향에 대해서만 이루어진다.

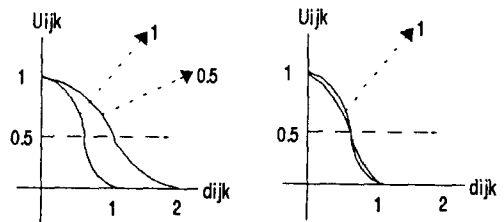
2. membership function

Zymerman과 Zysco가 주장한 membership function model^[10]에 따라 i번째 입력 벡터가 j번째 클러스터의 k번째 부클러스터에 소속되어질 소속도는 식(1)과 같이 정의하였다.

$$u_{ijk} = \frac{1}{1 + (\lambda d_{ijk})^\gamma} \quad (0 < \gamma < \infty, 0 < \lambda < \infty) \quad (1)$$

$$d_{ijk} = \|x_i - w_{jk}\| \quad (2)$$

여기서, λ 는 거리에 대한 스케일 상수이고 γ 는 얼마나 퍼지한 가를 나타내는 상수이다. (그림 4참고) d_{ijk} 는 n차 특징 공간에서 식(2)와 같이 정의되며, 유클리디안 거리를 나타낸다. λ 의 선택에 따라 u_{ijk} 의 값이 0.5가 되는 지점이 결정된다. 즉, d_{ijk} 가 $1/\lambda$ 이 되는 지점에 u_{ijk} 가 0.5의 값을 갖게 된다. 따라서 λ 가 커지면 퍼지함의 정도가 작아지게 되고, λ 가 작아지면 퍼지함의 정도가 커지게 된다. 그러므로 λ 가 크면 클수록 해당 입력 패턴에 대한 부클러스터들의 소속도가 급격하게 감소하게 되고 소속도 0.5를 넘기는 부클러스터의 수가 감소하게 되어 새로운 클러스터의 발생을 유도하게 된다.



(a)

(b)

그림 4. parameter 변화에 따른 소속도 설정

(a) λ 값에 따른 소속도의 변화

(b) γ 값에 따른 소속도의 변화

Fig. 4. Change of membership value due to parameters (a) the change of membership value due to λ (b) the change of membership value due to γ .

γ 값은 1보다 작은 값으로 정해지고 상수 λ 에 대해 독립적으로 u_{ijk} 의 퍼지함 정도를 결정하게 된다. 즉, 어떤 부클러스터의 소속도가 클수록 현재의 입력

과 가까움을 나타낸다.

3. 확장 단위 벡터

어떤 클러스터의 확장은 각 클러스터를 이루는 부클러스터 벡터들 사이의 거리가 멀어짐을 말한다. 예로서 2차원 평면상의 5개의 부클러스터들로 이루어진 클러스터의 확장에 대해 알아보자.

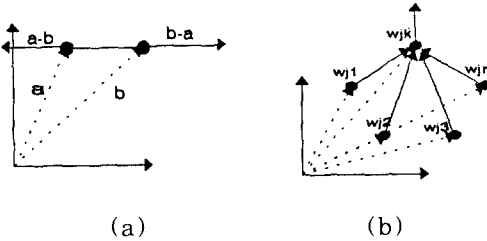


그림 5. 부클러스터 확장 단위 벡터

- (a) 2개의 부클러스터의 확장 벡터
- (b) n개의 부클러스터의 확장 벡터

Fig. 5. The Unit expansion vector of sub-cluster. (a) the expansion vector of two vectors (b) the expansion vector of n vectors.

그림5.(a)에서 부클러스터 a와 b로 이루어진 클러스터가 확장하기 위해서는 클러스터의 각 부클러스터 벡터가 서로 멀어지는 방향으로 진행하여야 한다. 따라서 각 부클러스터 a과 b가 멀어지기 위해서는 a는 a-b의 방향으로, b는 b-a의 방향으로 각각 확장하여야 한다. 이를 일반화하면 그림 5(b)와 식 (3)에서와 같이 구성할 수 있다.

$$a_{jk} = \frac{\sum_{l \neq k} (w_{jk} - w_{jl})}{\left\| \sum_{l \neq k} (w_{jk} - w_{jl}) \right\|} \quad (3)$$

식 (3)에서 a_{jk} 는 k번째 클러스터의 j번째 부클러스터 벡터가 j번째 부클러스터 벡터를 제외한 나머지 부클러스터 벡터로 부터 확장하기 위한 방향을 나타내는 확장 단위 벡터를 표현한 것이다. 따라서 w_k 와 w_k 를 제외한 현재의 모든 부클러스터들(n)사이의 거리의 합 $J_k = \frac{1}{2} \sum_{l \neq k} (w_{jk} - w_{jl})^2$ 을 w_k 에 대해 미분하면 w_k 의 확장 벡터 $\frac{\delta J_k}{\delta w_{jk}} = \sum_{l \neq k} (w_{jk} - w_{jl})$ 를 구할 수 있으며 확장 벡터의 단위 벡터를 구하면 식(3)과 같이 표현될 수 있다. 즉, 특정 부클러스터의 확장 단위 벡터는 부클러스터들의 거리의 합이 멀어지는 방향의 단

위 벡터라 할 수 있다.

4. 확장 조건

확장 조건을 만족할 때 입력 벡터에 가장 가까운 부클러스터(w_k^*)가 입력 벡터에 접근하게 된다. 확장 조건은 4가지 조건 식으로 구성되며, 이러한 조건 식들을 만족하는 경우에 클러스터의 확장을 수행하게 된다.

(조건 1)

$$\left\| \sum_{l \neq k} (x_i - w_{jl}) \right\| > \left\| \sum_{l \neq k} (w_{jk}^* - w_{jl}) \right\| \quad (4)$$

식 (4)는 입력 패턴에 가장 가까운 부클러스터 벡터(w_k^*)와 같은 클러스터 내의 나머지 부클러스터 벡터의 차 벡터 합이 입력 벡터로 부터 나머지 부클러스터 벡터의 차 벡터 합보다 절대값이 작아야 한다는 것을 의미한다.(그림 6 참고) 만일 (조건 1)이 만족되지 못하면 소속도는 1로 처리되고 부클러스터(sub-cluster)의 이동도 이루어지지 않는다. 즉, (조건 1)은 어떠한 입력 벡터가 가장 가까운 클러스터 내부에 존재하는가를 알기 위한 조건이며 만일 입력 벡터가 클러스터 내부에 존재한다면 해당 클러스터에 속하므로 경계선을 나타내는 부클러스터 벡터를 조절할 필요가 없음을 나타낸다.

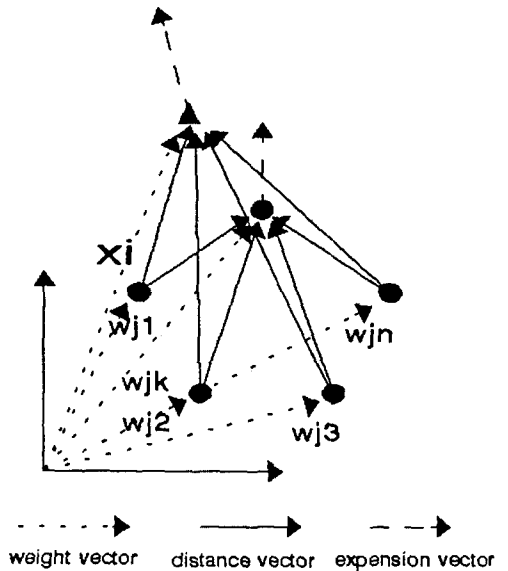


그림 6. (조건 1) 클러스터의 내부와 외부
Fig. 6. The interior and exterior of a cluster.

(조건 2)

$$u_{jk} > \beta \quad (5)$$

(조건 1)를 만족하는 경우, 입력 패턴이 가장 가까운 부클러스터가 속해 있는 클러스터의 외부에 존재한다는 것이며, 외부에 있는 입력 벡터에 대해 가장 가까운 부클러스터가 확장하기 위해서는 식 (5)에서와 같이 입력 벡터에 대해 부클러스터들의 소속도가 δ 보다 커야 한다. 만일 모든 클러스터 내의 모든 부클러스터가 β 보다 작은 소속도를 나타낸다면 새로운 클러스터를 발생시킨다. 이는 새로운 BCNN을 전체 network에 추가함을 의미한다.

(조건 3)

부클러스터(sub-cluster) 조정 방향은 식 (6)를 만족하여야 한다. 즉, 그림 7에서와 같이, i 번째 입력 벡터와 가장 가까운 부클러스터(w_{jk}^*)의 차 벡터를 b_{ijk} 라 할 때, b_{ijk} 와 부클러스터의 확장 단위 벡터(a_{jk}) 사이의 각(θ_{jk})이 w_{jk}^* 를 제외한 n 번째 클러스터 내의 모든 부클러스터에 대한 각(θ_{jk})보다 작아야 한다.

$$|\theta_{jk}| = \left| \cos^{-1} \frac{a_{jk} b_{ijk}}{\|b_{ijk}\|} \right|$$

$$\langle \theta_{jl} | = \left| \cos^{-1} \frac{a_{jl} b_{ljl}}{\|b_{ljl}\|} \right| \quad (6)$$

(for All $l \neq k$)

($b_{ijk} = x_i - w_{jk}$, j -th cluster, k -th sub-cluster, i -th input)

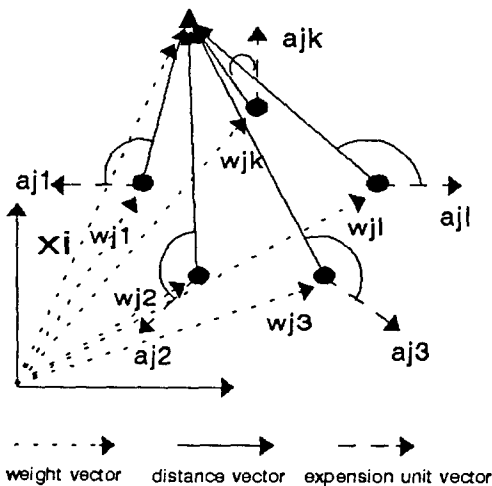


그림 7. 부클러스터의 조정 방향의 제약
Fig. 7. The bounded adaptation direction of sub-clusters.

식(6)으로 인해 부클러스터(sub-cluster)의 진행 방향을 제한함으로써 부클러스터들이 서로 간섭하지 않게 된다. 즉 부클러스터들은 독립적인 클러스터링을 수행하게 된다. 즉, 현재 입력 벡터 방향으로 확장할 적절한 부클러스터가 존재하지 않으므로 입력 벡터 방향으로의 확장을 주도할 새로운 부클러스터 벡터의 삽입이 필요하게 된다. 만일 위의 조건을 만족하지 못하게 되면 입력 벡터 위치에 새로운 부클러스터를 추가한다.

(조건 4)

가중치(weight) 벡터 조종에 있어서 어떤 클러스터의 부클러스터가 다른 클러스터 내부로 확장하는 것(클러스터의 겹침)을 방지하여야 한다. 이를 수행하기 위해 다음의 3단계로 겹침을 확인한다. 먼저, 가장 가까운 가중치 벡터(부클러스터 벡터, w_{jk}^*)를 저장해 두고, 가중치 벡터를 조정한다. 다음으로, 다른 클러스터들 내에서 가중치 벡터(w_{jk}^*)와 가장 가까운 가중치 벡터(w_{im}^{**})와 w_{im}^{**} 가 속한 클러스터(i)를 구한다. 클러스터 i 내에 가중치 벡터(w_{jk}^*)의 포함 여부를 구하기 위해 식(7)을 적용한다. 이 때, 식(7)은 조건 1, 식(4)에서 입력 벡터에 대한 검사를 부클러스터 벡터에 대한 검사로 대체한 것이다.

$$\left\| \sum_{i \neq m} (w_{jk}^* - w_{il}) \right\| > \left\| \sum_{i \neq m} (w_{im}^{**} - w_{il}) \right\| \quad (7)$$

만일 (7)식이 만족 되지못하면 이전에 저장해 두었던 조정 전의 부클러스터 벡터를 회복하고 클러스터의 확장은 이루어지지 않는다.

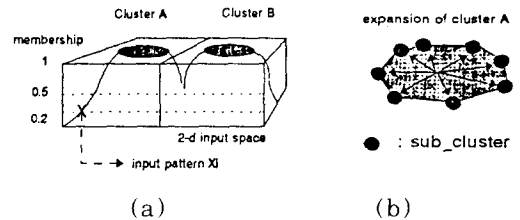


그림 8. 클러스터 내의 소속도의 분포 및 클러스터 확장
Fig. 8. The distribution of membership values in a cluster and cluster's expansion.
(a) distribution of membership values
(b) expansion of a cluster.

그림 8(a)와 같이 입력 패턴(x)이 입력된 경우 x 의 각 클러스터에 대한 소속도가 한계 소속도인 β 보

다 크지 못하므로 A, B 클러스터에 속하지 못하고 새로운 클러스터 C를 만들게 되는 과정을 설명한 것이다. 여기서 β 의 설정은 임의로 설정될 수 있는 변수이다. 그림 8(a)(b)에서 볼 수 있는 것과 같이 A 클러스터 내에 8개의 부클러스터가 있으며 부클러스터 내의 모든 소속도는 1로 설정되어 있다. 또한 각 클러스터를 표현하는 여러 개의 부클러스터들은 입력 공간에 대해 자신의 독자적인 소속도를 갖추고 있으며, 부클러스터 밖의 소속도는 비선형 곡선을 취한다.

5. weight adaptation

만일 위의 조건 (1), (2), (3), (4)가 만족하게 된다면 부클러스터의 재조정은 다음 식(8), (9)과 같이 이루어진다.

$$w^*_{jk} = w^*_{jk} + \Delta w^*_{jk} \tag{8}$$

$$\Delta w^*_{jk} = \mu_{ijk} A(x_i - w^*_{jk}) \tag{9}$$

여기서 A행렬은 symmetric, positive, definite 행렬이며 n차 특징 공간의 각 특징들에 대한 가치를 평가하는 스케일링 요소이다.

이와 같은 부클러스터들의 적응이 이루어지면서 클러스터의 확장을 수행하는 경우, 더 이상 확장할 수 없는 것처럼 보이는 부클러스터가 발생할 수 있다. 즉, 하나의 부클러스터가 주위의 부클러스터들에 의해 둘러싸여 확장 방향을 상실한 것처럼 보이는 경우이다. 그러나 부클러스터의 확장은 어떤 입력 벡터와 이와 가장 가까운 부클러스터의 확장 단위 벡터에 의해 결정되는 것이므로 둘러싸여진 부클러스터는 어느 방향으로든 클러스터의 확장을 위해 조정되어진다.

위에서 설명한 BCNN 알고리즘을 정리하면 다음과 같다.

◆ BCNN 알고리즘

(step 1) 초기 parameter, 부클러스터의 수 결정

(η, λ, β , 초기 부클러스터 수)

(step 2) 패턴 입력

(step 3) 식(1)에 의해 u_{ijk} 를 결정

및 win class j^* 결정

(step 4) 클러스터 확장 조건 검사

```

if ( 조건(1) 만족 ) {
    if ( 조건(2) 만족 ) {
        if ( 조건(3) 만족 ) {
            if ( 조건(4) 만족 ) {
                goto (step 5)
            }
        }
    }
}
    
```

```

else {
    goto (step 2)
}
}
else {
    새로운 부클러스터 생성
    goto (step 2)
}
}
else {
    새로운 클러스터 생성
    goto (step 2)
}
}
else { set  $u_{ijk} = 1$  }
(step 5)
식(8), (9) 적용
부클러스터의 위치 조정
    
```

III. 병렬 BCNN으로의 확장

여러 클러스터를 발생한 경우 그림 9와 같은 병렬 구조를 갖는 BCNN이 이루어진다. 이러한 병렬 구조의 각 BCNN들은 독립적인 클러스터를 의미하고 각 클러스터 내부에는 자신의 부클러스터들을 포함하게 된다. 2장의 (조건3)에 의해 새로운 클러스터가 발생하는 경우 새로운 BCNN이 신경망에 추가된다.

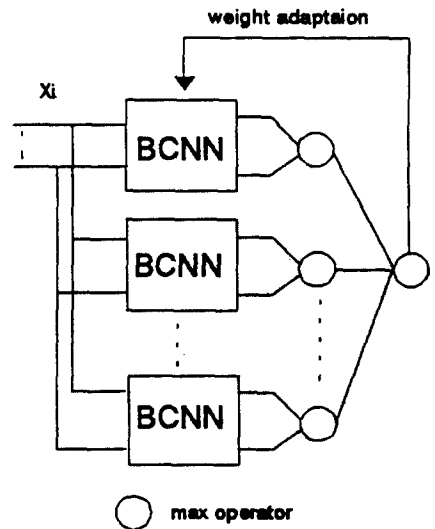


그림 9. parallel BCNN의 구성
Fig. 9. Configuration of parallel BCNN.

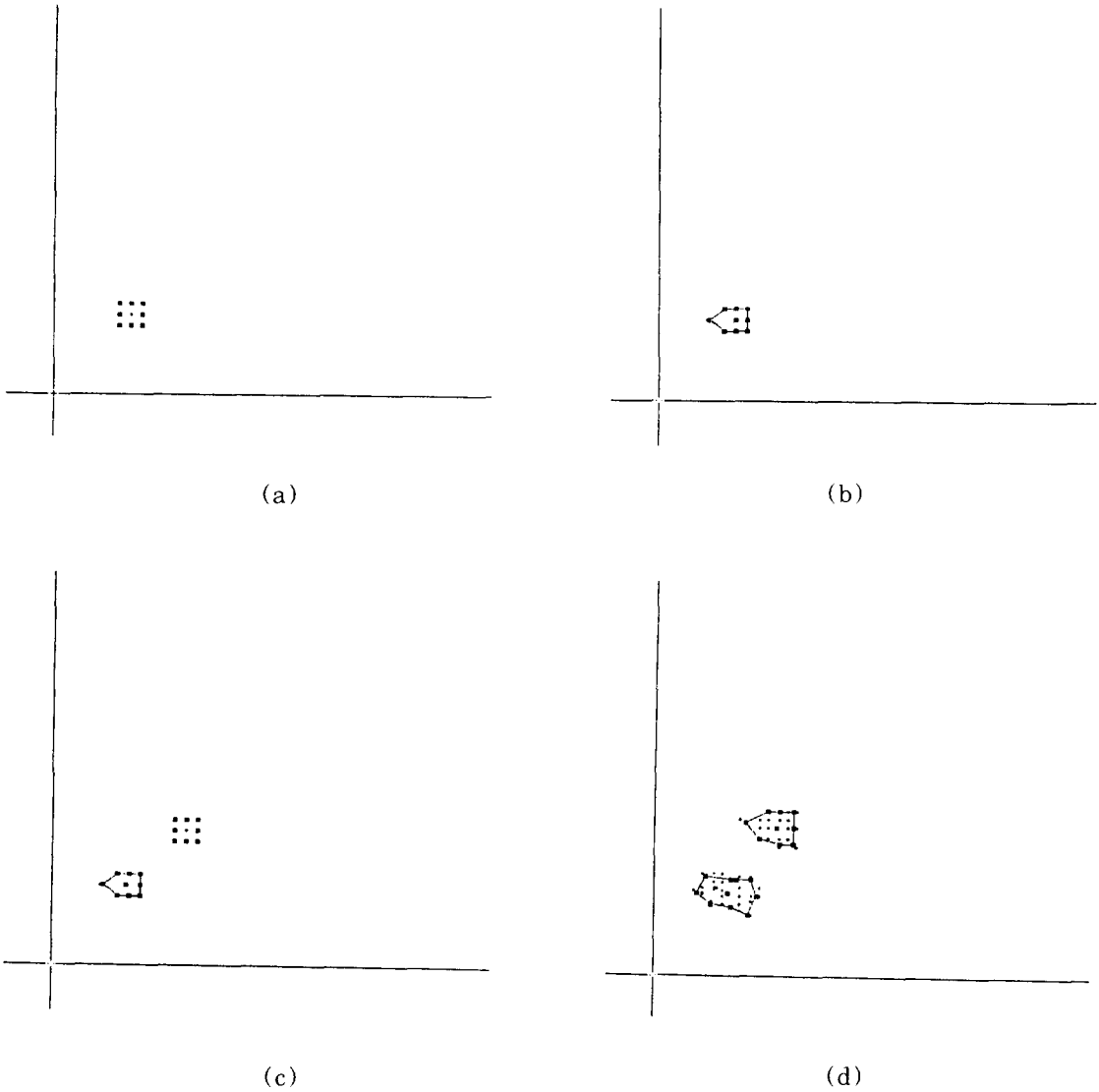


그림 10. 클러스터의 확장 과정

Fig. 10. Procedure of cluster expansion.

이러한 클러스터링 알고리즘으로 수행되는 과정을 그림 10에 보였다. 2차 특징 공간에서 우선 특징 벡터 a 가 입력되고 A 클러스터의 8개 부클러스터들이 8방향에서 단위 거리만큼 떨어진 지점에서 초기화되는 과정을 그림 10(a)에 표현하였다. 그림 10(b)는 특징 벡터 b 가 입력되어 A 클러스터의 입력 벡터에 가장 가까운 왼쪽 중앙의 부클러스터의 소속도가 0.5를 초과하므로 A 클러스터가 확장하는 과정을 나타낸 것이다. 그림 10(c)는 특징 벡터 c 가 입력되어 새로운 클러스터 B가 만들어지는 과정을 나타내었으며,

마지막으로 그림 10(d)는 이러한 과정을 수행하여 만들어진 클러스터의 모습을 나타낸다.

IV. 실험 결과

BCNN 클러스터링 알고리즘이 어떤 data군에 대해 적용될 때, 그 결과에 대한 성능 평가는 분류의 평균 에러율로 정의할 수 있다. 이러한 평균 에러율은 적용한 모든 입력 값 중 잘못 분류된 입력 값의 비율로서 나타낸다.

2-D 및 4-D Gaussian random variable data 군을 분류할 자료로 하여 본 알고리즘과 FMMC와의 성능을 비교 평가하고자 한다. 물론 비교 대상의 선정은 클러스터 갯수를 모르는 경우, on line 클러스터링을 수행하는 알고리즘들 중에서 선정된 것이다. 우선 가우시안 랜덤 변수(Gaussian random variable)을 사용하여 클러스터 a,b,c,d를 만들고 각 클러스터의 confusion matrix를 산정한 것을 그림 11와 표1에 각각 나타내었다. 이러한 confusion matrix의 구성은 우선 각 입력 값에 소속 표시를 붙이지 않은 상태에서 불규칙적인 순서로 입력을 허용한 다음 해당 클래스에 속하는 클러스터들을 분류하고 이러한 클러스터들에 속한 입력 값의 소속을 조사함으로써 이루어졌다. 그림 11에서 수행한 실험의 초기 값은 η (0.33), λ (0.1), β (0.5), 초기 부클러스터 4개로 초기화하였다. 실험에서 알 수 있는 것은 클러스터의 모양에 따라 다르지만 초기 부클러스터의 수는 4개 이상이면 안정된 클러스터링을 수행함을 알 수 있다. 특히, 클러스터의 생성에 관련된 η, λ, β 의 설정이 중요하다. 각 parameter의 결정 순서는 다음과 같다. 1) 먼저 β 값을 결정한다. (보통 0.5가 적당하다) 2) 소속도를 결정하는 η, λ 를 정한다. 를 먼저 결정하고(1.0가 적절하다) η 의 값을 정한다. 이때 λ 의 값은 식(1)의 1/2 거리 지점 설정이라는

개념 하에서 소속도 0.5가 되는 거리를 정하게 된다. 실험에서는 거리 4인 지점을 소속도 0.5가 되도록 한 것이다. 그림 12과 표 2는 FMMC에 의한 클러스터링 결과를 나타낸 것이다. 동일한 입력 값들을 사용하였고 단지 소속도를 산출할 때만 입력 값을 정규화하였다. FMMC에서는 n차 특징 벡터가 입력 값으로 주어진 경우 모든 차수에 대해 hyperbox가 겹쳐짐을 허용하지 않는다. 따라서 수축(contraction)을 수행하게 되며, 이러한 수축 과정을 통해 기존에 학습하였던 데이터를 보존하지 못하게 된다. 이러한 과정은 그림 12에서와 같다. 그러므로 FMMC 알고리즘은 클러스터의 경계선에 대해 잘못 판단할 가능성이 많은 것을 알 수 있다. (표 2) 이때, FMMC의 최대 크기 parameter를 15로 하였다.

그림 13은 BCNN의 η 0.05으로 한 경우의 클러스터링 결과이며, 그림 14는 FMMC의 최대 크기 parameter를 25로 한 경우의 결과이다. 4차원 data set에 대한 실험은 표 3, 4의 confusion matrix로 나타내었다. 2차원 data set에 대한 실험으로 부터 본 알고리즘이 기존의 FMMC 알고리즘보다 클러스터의 형태 및 크기에 있어서 좀 더 효율적임을 알 수 있었으며, 따라서 2차원 및 4차원 data set에 대한 분류화 성능의 개선을 보일 수 있었다.

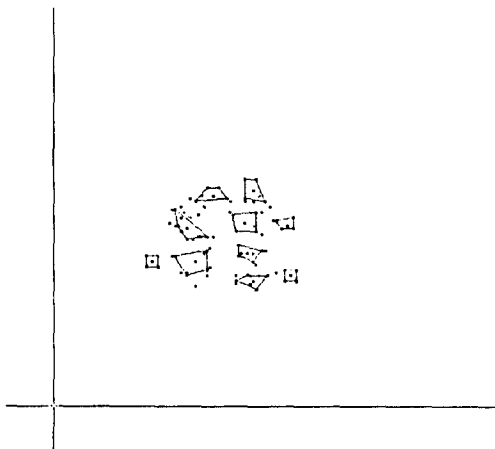


그림 11. 2-D Gaussian random data 군에 대한 BCNN 클러스터링 (η (0.33), λ (0.1), β (0.5), 초기 부클러스터 4개)

Fig. 11. BCNN clustering for 2-D Gaussian random data. (η (0.33), λ (0.1), β (0.5), initial sub_cluster 4)

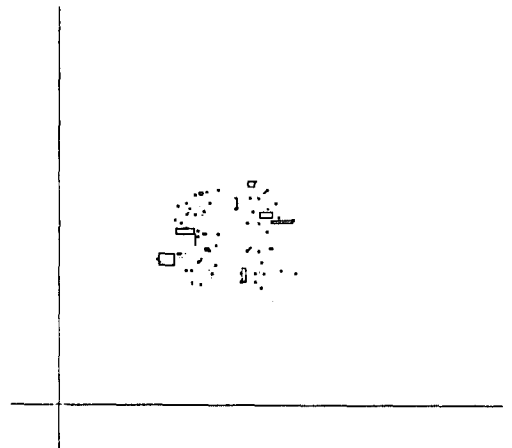


그림 12. 2-D Gaussian random data 군에 대한 FMMC 클러스터링 (클러스터 최대 크기 : 15)

Fig. 12. FMMC clustering for 2-D Gaussian random data. (the maximum cluster volume : 15)

표 1. 2-D Gaussian random data 군에 대한 BCNN confusion matrix, 행 : 기준 class, 열 : 할당 class

Table 1. BCNN confusion matrix for 2-D Gaussian random data. row : reference class, column : assigned class.

	class 1	class 2	class 3	class 4
class 1	92 %	6 %	0 %	2 %
class 2	0 %	100 %	0 %	0 %
class 3	0 %	2 %	96 %	2 %
class 4	4 %	2 %	6 %	88 %

표 2. 2-D Gaussian random data 군에 대한 FMCC confusion matrix, 행 : 기준 class, 열 : 할당 class

Table 2. FMCC confusion matrix for 2-D Gaussian random data. row : reference class, column : assigned class.

	class 1	class 2	class 3	class 4
class 1	72 %	10 %	2 %	20 %
class 2	14 %	80 %	2 %	2 %
class 3	0 %	2 %	90 %	8 %
class 4	4 %	2 %	16 %	78 %

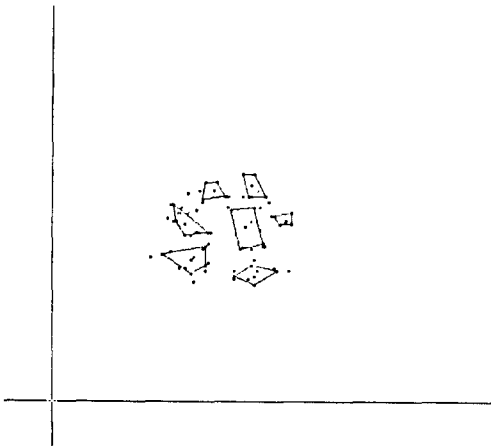


그림 13. 2-D Gaussian random data 군에 대한 BCNN 클러스터링 (η (0.33), λ (0.05), β (0.5), 초기 부클러스터 4개)

Fig. 13. BCNN clustering for 2-D Gaussian random data. (η (0.33), λ (0.05), β (0.5), initial sub_cluster)

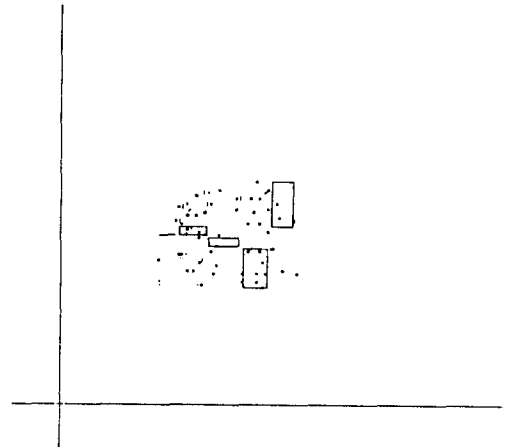


그림 14. 2-D Gaussian random data 군에 대한 FMCC 클러스터링 (클러스터 최대 크기 : 25)

Fig. 14. FMCC clustering for 2-D Gaussian random data. (the maximum cluster volume : 25)

표 3. 4-D Gaussian random data 군에 대한 BCNN confusion matrix, 행 : 기준 class, 열 : 할당 class

Table 3. BCNN confusion matrix for 4-D Gaussian random data. row : reference class, column : assigned class.

	class 1	class 2	class 3	class 4
class 1	90 %	2 %	0 %	8 %
class 2	0 %	94 %	6 %	0 %
class 3	0 %	4 %	96 %	0 %
class 4	0 %	0 %	2 %	98 %

표 4. 4-D Gaussian random data 군에 대한 FMCC confusion matrix, 행 : 기준 class, 열 : 할당 class

Table 4. FMCC confusion matrix for 4-D Gaussian random data. row : reference class, column : assigned class.

	class 1	class 2	class 3	class 4
class 1	82 %	6 %	0 %	12 %
class 2	2 %	88 %	4 %	6 %
class 3	0 %	4 %	90 %	6 %
class 4	2 %	2 %	16 %	80 %

V. 결론 및 고찰

본 논문에서는 입력이 순차적으로 주어질 경우 클러스터 생성과 클러스터의 경계를 나타내는 부클러스터의 확장 및 생성을 통해 클러스터의 모양을 형성하는 클러스터링 알고리즘을 제시하였다. 또한 본 논문을 통해 FMMC의 단점이었던 hyperbox의 형태 및 크기의 제한성을 경계선을 학습하는 부클러스터 확장을 도입함으로써 극복하였고 따라서 risk decision area 및 분류 에러가 기존의 FMMC에서 보다 상당히 줄어들음을 알 수 있었다. 또한 on-line 클러스터링을 수행함으로써 산업계에서 요구하는 data의 실시간 처리에 접근하였다. 그러나 본 논문에서 제시하는 BCNN 알고리즘은 클러스터 갯수를 모르고 순차적인 입력을 분류 대상으로 하는 경우 필수적인 계층적인 클러스터링 중 초기 클러스터링을 목적으로 하였으므로 궁극적인 클러스터링을 수행하기 위해서는 이러한 초기 클러스터링을 통해 생성되어진 클러스터를 입력 값으로 하는 상위의 클러스터링이 필요하다.

參考文獻

- [1] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms," New York : Plenum Press, 1981.
- [2] R. N. Dave and K. Bhaswan, "Adaptive C-shell Clustering and detection of ellipses," IEEE Trans. Neural Networks, vol. 3, no. 5, pp. 643-662, Sept. 1992.
- [3] J. C. Bezdek and R. H. Hathaway, "Numerical convergence and interpretation of the fuzzy c-shells clustering algorithm," IEEE Trans. Neural Networks, vol. 3, pp. 787-793, Sept. 1992.
- [4] R. Krishnapuram, H. Furigui, "The fuzzy shell clustering algorithms for boundary detection and pattern recognition," in Proc. SPIE Conf. Intelligent Robots and Computer Vision X: Algorithms and Techniques (Boston) Nov. 1991, pp. 458-465.
- [5] R. N. Dave, K. Bhaswan, "Fuzzy-shell clustering and applications to circle detection in digital images," Int. J. General Systems, vol. 16, pp. 343-355, 1990
- [6] R. Krishnapuram, O. Nasraoui "The Fuzzy C Spherical Shells Algorithm: A New Approach," IEEE Trans. Neural Networks, vol. 3, no. 5, pp. 787-793, Sept. 1992.
- [6] R. N. Dave, "New measures for evaluating fuzzy partitions induced through C-shells clustering," in Proc. SPIE Conf. Intelligent Robots and Computer Vision X: Algorithms and Techniques (Boston) Nov. 1991, pp. 406-414.
- [7] J. Davis, T. Warms, and W. Winters, "A neural network implementation of the fuzzy c-means clustering algorithm," In Proc. Int. Joint Conf Neural Networks (Seattle), vol. 2, 1991, p. A-953.
- [8] R. Krishnapuram, J. M. Keller, "A Possibilistic Approach to Clustering," IEEE transactions on Fuzzy Systems, vol. 1, NO. 2, May 1993.
- [9] P. Simson, "Fuzzy Min-Max Neural Networks-Part 2: Clustering," IEEE transactions on Fuzzy Systems, vol. 1, NO. 1, May 1993.
- [10] H. J. Zimmerman and P. Zysco, "Quantifying vagueness in decision models," European J. Operational Res., vol. 22, pp. 148-158, 1985.
- [11] R. N. Dave, "Robust fuzzy clustering Algorithm," Sec. IEEE Int. Conf. on fuzzy systems (San Francisco), vol. 1, pp. 1281-1286, March-April 1993.

 著者紹介



金載顯(正會員)

1969年 1月 16日生. 1991년 한양대 전자공학과 졸업 1990년 동 대학원 전자공학과 졸업 (석사). 현재 한양 대학원 전자공학과 박사 과정

徐一弘(正會員)

1955年 4月 16日生. 1977年 서울대 공대 전자공학과 졸업. 1982年 한국 과학기술원 전기 및 전자공학과 졸업(공학). 1982年 대우 중공업 기술 연구소 근무. 1987年 ~ 88年 미국 미시간대 객원 연구원 현재 한양대 공대 전자공학과 교수. 당학회 편집위원. 서울대(科學財團) ERC-ACI研究委員