

論文97-34C-7-6

Convex polytope를 이용한 퍼지 클러스터링 (Fuzzy Clustering Involving Convex Polytope)

金載顯*, 徐一弘*, 李定勳*

(Jae-Hyun Kim, Il Hong Suh, and Frank Chung-Hoon Rhee)

요 약

대부분의 클러스터링 방법들은 특정 정형을 사용하며 그 결과도 사용되는 정형에 많은 영향을 받는다. 본 논문에서는 임의로 분포되어 있는 데이터를 클러스터의 수 등과 같은 선형적인 지식 없이 적절히 표현하기 위한 컨벡스 형태의 클러스터를 생성하는 새로운 방법을 제안하고자 한다. 여기서 새로운 클러스터가 생성 될 것인지 혹은 클러스터에 새로운 꼭지점을 추가할 것인지 지를 판단하기 위해 클러스터로의 소속도를 나타 내는 비선형 소속함수를 사용하며, 특히 이러한 비선형 함수는 패턴벡터와 가장 가까운 꼭지점과 사이의 거리 및 가장 가까운 꼭지점이 최근 변화된 양을 기준으로 구성 되도록 하여 적응적인 클러스터링이 되도록 하였다. 본 논문에서는 이러한 클러스터링의 장점을 몇 가지 실험적인 예제를 통하여 보여주고자 한다.

Abstract

Prototype based methods are commonly used in cluster analysis and the results may be highly dependent on the prototype used. In this paper, we propose a fuzzy clustering method that involves adaptively expanding convex polytopes. Thus, the dependency on the use of prototypes can be eliminated. The proposed method makes it possible to effectively represent an arbitrarily distributed data set without a priori knowledge of the number of clusters in the data set. Specifically, nonlinear membership functions are utilized to determine whether a new cluster is created or which vertex of the cluster should be expanded. For this, the membership function of a new vertex is assigned according to not only a distance measure between an incoming pattern vector and a current vertex, but also the amount how much the current vertex has been modified. Therefore, cluster expansion can be only allowed for one cluster per incoming pattern. Several experimental results are given to show the validity of our method.

I. 서 론

대부분의 퍼지 클러스터링 방법들은^[1-6] Bezdek에 의해 제안된 퍼지 C-Means(FCM) 알고리즘^[7]을 기반으로 하며 하나의 데이터가 모든 클러스터들로의 소속도의 합은 반드시 1이 되어야 한다는 확률론적 제한하에서 개발 되었다. 소속도들에 대한 이러한 제한은 모든 소속도들이 0인 trivial solution을 피하기 위해 설정되었고 소속도를 probability 혹은 degree of

sharing으로 해석할 수 있는 응용 분야에서는 의미 있는 제한이다^[2]. 그러나 이러한 제한에 의해 생성된 소속도는 상대적인 수치이기 때문에 소속도가 전형성(typicality)을 나타내는 응용 분야에서는 의미가 상실된다^[1,2]. 이러한 문제를 극복하기 위해 Robust 퍼지 클러스터링 알고리즘(RFCA)^[1]과 Possibilistic C-Means(PCM)^[2]이 제안 되었으며, 여기서는 모든 클러스터에 대한 소속도의 합이 1이 될 필요가 없다. 즉, 어떤 클러스터 내의 입력 데이터의 소속도 값이 입력 패턴과 나머지 클러스터들 사이의 거리를 고려함이 없이 결정되어질 수 있다. 그러나 PCM을 비롯한

* 正會員, 漢陽大學校 電子工學科

(Dept. of Elec. Eng., Hanyang Univ.)

接受口字:1996年10月8日, 수정완료일:1997年7月3日

많은 FCM-based 알고리즘들의 고정된 prototype을 사용하는 경우 잘 못 판단하게 될 영역(Risk Decision Region)이 커지기 때문에 임의의 형태를 갖는 클러스터를 효과적으로 표현하기에는 역부족이다^[9].

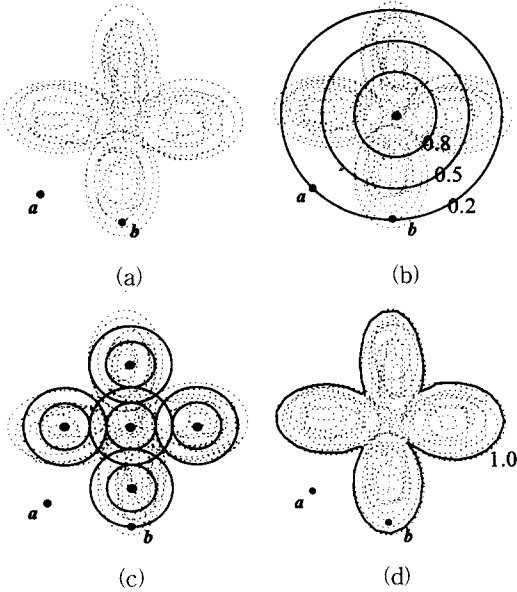


그림 1. 클로버 데이터 클러스터링

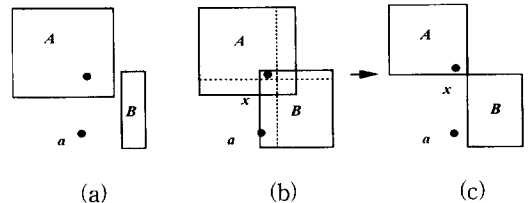
(a) 클로버 모양의 데이터. (b) 한 개의 점 형태의 정형에 의해 표현된 데이터. (c) 다섯 개의 점 형태의 정형에 의해 표현된 데이터. (d) 클로버 모양의 정형에 의해 표현된 데이터.

Fig. 1. Clustering results for clover shaped data set. (a) clover-shaped data set. (b) clover shaped data set represented by only one point-type prototype (c) clover-shaped data set represented by (d) clover-shaped data set represented by a five point-type prototypes

예로서 그림 1(a)와 같은 클로버 모양의 데이터를 고려해 보자. 여기서 동일한 소속도 값을 나타내기 위해서 등고선을 사용하였다. 그림 1(b)로부터 point prototype이 사용된 다면 클러스터 외곽에 위치한 데이터 a 가 클러스터 내부에 존재하는 데이터 b 와 동일한 소속도 값을 갖음을 볼 수 있다. 이러한 불균형은 클러스터가 단일의 prototype을 갖는다 라는 가정에 기인한다. 만일 좀더 나은 클러스터링을 수행하기 위해서는 그림 1(c)와 같이 5개의 point prototype를 갖는 클러스터로 주어진 데이터를 표현하는 것이지만, 필요한 클러스터의 수에 대한 정보를 미리 알 수 없는 상황에서는 사용할 수 없다. 따라서 궁극적으로는 적

절한 개수의 클러스터를 사용하여 그림 1(d)에서 보여지는 것과 같은 클로버 모양의 데이터를 표현할 수 있어야 한다.

은 라인으로 클러스터들을 할당하는 Fuzzy Min-Max Clustering Neural Network(FMCCNN)^[8]과 Fuzzy ART알고리즘^[6]은 필요한 클러스터의 수를 미리 알 수 없는 경우에 사용될 수 있는 클러스터링 방법들이다. 그러나 이러한 경우에도 prototype의 형태가 hyperbox라는 일정한 틀이 있고, 따라서 이러한 몇 개의 hyperbox를 사용하여 주어진 데이터를 효과적으로 표현하지 못할 수 있다. 또한, 이러한 방법들에는 클러스터링 과정에서 풀어야 할 중요한 문제가 있다. 클러스터의 수축 과정이 바로 그것인데, 이 과정은 클러스터를 형성하기 위한 매우 중요한 과정이지만 이전에 매우 잘 클러스터링 된 상태에 나쁜 영향을 끼칠 수 있다. 즉, 수축 과정은 어떤 입력 패턴이 클러스터들로 속하는 정도(소속도)가 동요(unstable, fluctuate)하게 하는 주된 요인이 된다. 그림 2에서 이러한 수축 과정의 단점을 나타내주고 있다.



(a) (b) (c)

그림 2. Contracting process에 의한 소속도의 fluctuating 현상

(a) 초기상태 (b) 클러스터 B의 확장 후 모습 (c) 클러스터 B의 수축 후 모습

Fig. 2. Fluctuating(unstable) membership assignment due to a contraction process: (a) initial state, (b) results after expansion, and (c) results after contraction.

위에서 나열된 문제점들을 해결하기 위해, 특정한 prototype에 얽매지않는 방법을 연구 수행 하였다^[9]. 이러한 기존의 연구에서는 클러스터의 내부와 외부에 대한 구분이 직관적인 공식에 의해 정의되었다. 따라서 정확한 확장을 유도하지 못하였고, 생성된 클러스터들의 해석에 있어서 체계적이지 못하였다. 이를 보완하기 위해 본 연구에서는 convex polytope을 이용한 퍼지 클러스터링 방법을 제안하고자 한다. 본 방법은 데이터가 입력 될 때 마다 가장 가까운 convex 클러스터를 확장 시키는 방법이다. 여기서, 주어진 데

이터는 몇 개의 클러스터들로 표현되고, 각 클러스터는 여러 개의 vertex들을 포함하는 convex polytope에 의해 표현된다.

즉, 입력 패턴이 모든 클러스터의 외곽에 존재할 때 마다 가장 가까운 클러스터의 vertex가 입력 데이터 position으로 확장하게 되고, 여기서 비선형 소속도 함수가 사용되어 새로운 클러스터가 생성될 것인지, 새로운 vertex가 추가될 것인지, 또는 어떤 클러스터의 어떤 vertex가 확장할 것인지를 판단하게 된다. 이러한 소속도 함수는 vertex를 교정할 때마다 입력된 패턴 vector와 현재 vertex사이의 거리 뿐만 아니라, 현재의 vertex가 얼마 만큼 교정되었는지에 따라 결정되어 진다.

본 연구에서 제안하고자 하는 방법은 다음과 같은 장점을 지니게 된다. (1) 매 입력 데이터에 대해 온라인으로 클러스터를 할당 혹은 적응시킨다. (2) 비정형의 convex prototype을 사용함으로써 입력 데이터를 좀 더 효율적으로 표현할 수 있다. (3) 여러 물리적 의미가 있는 파라메타를 적용함으로써 구체적인 클러스터링 환경을 구축할 수 있다.

II. Convex Polytope을 이용한 클러스터 생성

Convex 클러스터 내에서는 1의 소속도를 갖는다. 일련의 입력 데이터에 대해 이러한 convex 클러스터를 성취할 수 있기 위해, 클러스터 내의 가장 가까운 vertex가 입력되는 패턴 벡터에 따라 적절히 조절되어야 한다.

이러한 클러스터의 수정을 위해, 현재의 입력 패턴이 어떤 클러스터 내부에 포함되는지 혹은 클러스터를 수정한 후에도 여전히 convex 클러스터를 유지하는지를 판단하는 것이 중요하다. 만일 입력 데이터가 클러스터의 외부에 있다면 convex한 특성을 해치지않는 한 가장 가까운 클러스터의 vertex가 입력 데이터 position으로 확장하게 된다.

1. 입력 패턴이 Convex Cluster내에 포함되는지를 결정

먼저 어떤 클러스터로부터 입력 패턴을 분리시키는 hyperplane의 존재성은 다음의 Theorem 1에 의해 증명되었다.

Theorem 1: [Eidelheit theorem] set $B \subseteq$

R^n 이 m 개의 vertex를 갖는 convex polytope이라고 가정하고, 만일 어떤 입력 패턴 x 가 set B 의외부에 존재하게 되면, 항상 x 와 B 를 분리하는 separating hyperplane이 존재하게 된다.

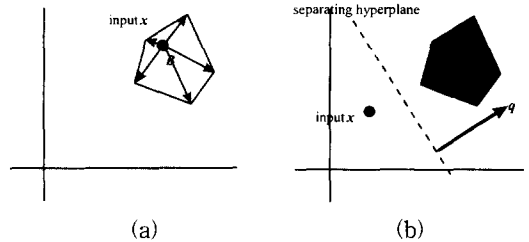


그림 3. Separating hyperplane의 존재성
(a) 입력 x 가 Convex polytope B 내에 포함되는 경우 separating hyperplane은 존재하지 않는다. (b) 입력 x 가 Convex polytope B 내에 포함되지 않는 경우 separating hyperplane이 존재한다.

Fig. 3. Fluctuation due to contraction process.
(a) If an input x is included in a convex polytope B , there is not any hyperplane separating x from B . (b) If an input x is not included in a convex polytope B , a hyperplane separating x from B exists.

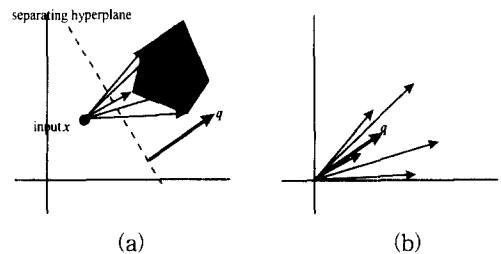


그림 4. 입력벡터와 클러스터B사이의 법선벡터와 Separating hyperplane과의 관계 (a) Separating hyperplane의 convex set B 를 향하는 법선 벡터와 입력 벡터를 원점으로 한 vertex 벡터는 항상 동일한 half space내에 존재한다. (b) 원점으로 벡터 이동 했을 때의 모습

Fig. 4. Relation between an input vector x and a cluster B
(a) An input vector and a normal vector heading to a cluster B from separating hyperplane exist in the same half space (b) normal vectors described based on origin.

Theorem 1로부터 현재의 입력 패턴이 convex set B 에 포함되지 않으면 separating hyperplane이 반드시 존재함을 알 수 있다. 그림 3은 이를 보여준다.

여기서, 이러한 separating hyperplane에 수직이고 convex set B 를 향하는 법선 벡터와 입력 벡터를 원점으로 한 vertex 벡터는 그림 4처럼 항상 동일한 half space에 존재함을 관찰할 수 있다.

또한, 동일한 half space에 존재하는 vector들 중 임의의 vector를 reference vector로 하고 나머지 vector들을 reference vector와 벡터 합을 수행함으로써 구한 모든 벡터들의 내적은 항상 positive라는 성질을 갖고있다. 이러한 성질들을 이용하여 다음과 같이 어떤 separating hyperplane의 존재성을 알 수 있다.

Separating Hyperplane Detecting(SHD) 알고리즘

Given an input vector x ;

Obtain expansion unit vector $e_i^k = \frac{x - a_i^k}{\|x - a_i^k\|}, i = 1, \dots, m.$ (1)

whereis a_i^k denotes the i th vertex vector in the k th cluster;

Among m expansion vectors e_i^k , arbitrarily select an expansion vector, e_i^k , as a reference vector;

FOR all expansion unit vectors for the k -th cluster except for e_i^k **DO**

 Compute f_i^k given as

$$f_i^k = \frac{e_i^k + e_j^k}{\|e_i^k + e_j^k\|}, \text{ for all } i, i \neq j. \quad (2)$$

ENDFOR

Set interior = FALSE;

(The input pattern is initially assumed to lie outside the k th cluster)

REPEAT

IF $f_i^k \cdot f_j^k \leq 0$ **THEN** (3)

 Set interior = TRUE;

 (The input pattern is in the interior side of the k th cluster)

ENDIF

UNTIL (interior = TRUE);

2. 클러스터 수정 후 Convexity 의 유지

이전 절에서는 입력 패턴 x 를 convex set B 로부터 분리 시키는 separating hyperplane의 존재성과 이를 찾는 알고리즘에 대해 논하였다. 이번 절에서는 클러스터의 vertex 수정 (즉, 새로운 vertex의 추가, 혹은 어떤 vertex를 위치 수정)한 후 클러스터의 convexity를 유지하는지에 관해 조사하는 방법을 설명한다. 이 알고리즘은 SHD 알고리즘을 사용함으로써 매우 단순해진다. 클러스터를 확장하기 위한 vertex

수정에는 다음과 같이 두 가지 경우가 있을 수 있다. 하나는 새로운 vertex의 추가이고, 또 다른 하나는 이미 존재하는 vertex의 위치 수정이다. 이는 그림 5에 잘 나타나있고 아래와 같이 정리할 수 있다. 먼저, 모든 클러스터의 외곽에 존재하는 현재입력 패턴 x 의 위치에 가장 가까운 클러스터 및 해당 클러스터의 가장 가까운 vertex를 구한다. 이때 입력 패턴 x 는 해당 클러스터에 새로운 vertex가 추가된다. 다음 현재 입력 패턴 x 의해 구성된 클러스터 내부에 어떤 vertex라도 포함되어 있다면 이러한 vertex를 삭제한다. 삭제되는 경우는 그림 5(a)의 치환의 경우이고 삭제되지 않는 경우는 그림 5(b)에서 와 같이 새로운 vertex가 추가 된다.

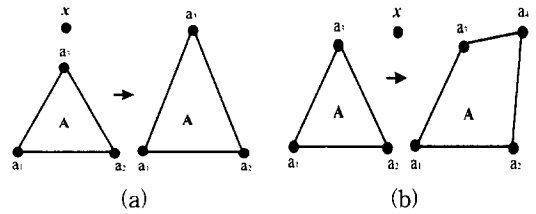


그림 5. Convexity를 유지하면서 클러스터 수정
(a) 기존의 vertex를 새로운 vertex로 치환.
(b) 새로운 vertex의 추가.

Fig. 5. Cluster modification
(a) substitution of an existing vertex with a new one (b) addition of a new vertex.

이 두 가지의 클러스터 수정 경우는 다음 두 알고리즘에서 보여 주고 있다.

Convexity Test (CT) 알고리즘

Set convexity = TRUE; Set vertex number $i=0$;

WHILE (convexity = TRUE **AND** $i \neq$ total number of k -th cluster vertices)

 Increment i ;

 Construct a convex set C_i with all vertices except for i -th vertex;

 Apply the SHD algorithm i -th vertex and vertices of C_i ;

IF interior = TRUE **THEN**

 Set convexity = FALSE;

END IF

END WHILE

Convexity Test After Modification (CTAM) 알고리즘

Construct a new cluster with input pattern x as a new vertex and all the vertices of the

```

k-th cluster;
Apply the CT algorithm;
IF convexity = FALSE THEN
    Replace  $x$  by its nearest neighboring
    vertex;
END IF
    
```

3. Intra-expandability measure of a convex set

클러스터의 확장을 위에서 제시한 것과 같이 수행하게 되면 클러스터는 무한히 커질 수 있다. 이를 보완하기 위해 앞으로 제시될 클러스터링 알고리즘에서는 각 클러스터의 크기(확장된 정도)를 제한하게 된다. 이를 위해 다음과 같이 클러스터의 확장 정도를 정의한다. 먼저, Convex 클러스터의 확장 정도는 intra-expandability measure로 정의되고, intra-expandability measure는 클러스터의 평균 vertex vector와 모든 vertex vector사이의 평균 거리로서 다음 식 (4), (5)와 같이 구할 수 있다.

$$\epsilon_{intra}(A^k) = \frac{1}{m} \sum_{i=1}^m \|a_i^k - c_k\|, \quad (4)$$

여기서 c_k 는

$$c_k = \frac{1}{m} \sum_{i=1}^m a_i^k. \quad (5)$$

이러한 intra-expandability measure는 클러스터의 중심으로부터의 표준 편차로 나타낸 것으로 동일한 표준 편차를 갖는 클러스터들은 모양에 상관없이 동일한 정도로 확장된 것으로 간주된다.

III. Fuzzy Convex Clustering(FCC) 알고리즘

본 연구에서 제시하는 클러스터링 알고리즘에서 퍼지 소속도 함수는 입력 패턴을 어떤 클러스터에 효과적으로 할당하기 위해 사용된다. 즉, 입력 패턴이 어떤 클러스터에 얼마 만큼 속하는 지를 나타내는 방법을 설명한다. 여기서, 소속도 값을 할당하기 위한 2단계 과정을 소개하고자 한다. 먼저 i 번째 vertex가 k 번째 클러스터에 소속될 정도를 다음 식 (6)과 같이 나타낸다.

$$\mu_i^k(k) = \frac{1}{1 + \lambda \|x - a_{i,new}^k\|^r}, \quad (6)$$

where $\lambda = \eta \|a_{i,old}^k - a_{i,new}^k\|$.

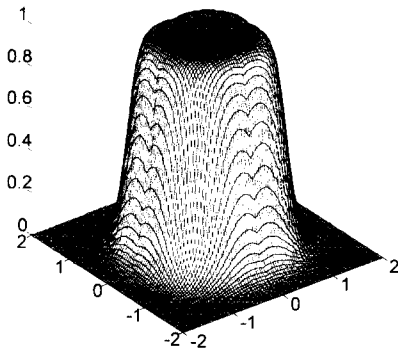
여기서, $a_{i,new}^k$ 는 가장 최근에 변경된 k 번째 클러스터

내의 i 번째 vertex를 나타내며, $a_{i,old}^k$ 는 변경되기 이전의 vertex를 나타낸다. 또한, $a_{i,new}^k$ 는 a_i^k 가 입력 패턴을 포함하기 위해 변경될 때 마다 새롭게 갱신된다. 식(6)에서 λ 는 $\mu_i^k=0.5$ 가 되는 x 와 $a_{i,new}^k$ 의 거리를 정하는 상수이고, η 는 scale 상수이며 η 와 동일한 의미를 갖는다. 그리고 $(0 < \eta < 1)$ 는 fuzziness 혹은 sharp-ness를 나타내는 상수이다. 이러한 소속도 함수는 Zimmermann 등이 소속도 함수의 전형적인 모델로 제안한 것으로부터 착안하였다^[10]. 또한, 식(6)은 소속도 함수가 대상 vertex의 최근에 확장된 정도에 따라 조절된다는 것을 보여준다. 즉, 소속도 함수는 입력 패턴 x 와 $a_{i,new}^k$ 사이의 거리 뿐만 아니라, a_i^k 가 수정된 양에 따라 조절된다. 한편, k 번째 클러스터 내의 입력 패턴 x 의 소속도 값 $m(k)$ 는 다음과 같이 주어진다.

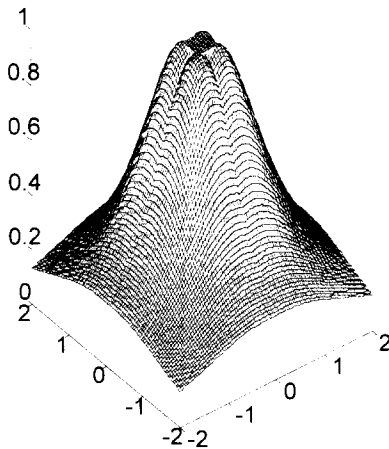
$$m(k) = \begin{cases} 1 & \text{if an input pattern } x \text{ lies inside } k\text{th cluster} \\ \max_i \{\mu_i^k\} & \text{otherwise} \end{cases} \quad (7)$$

또한, 입력 패턴 x 가 어떤 클러스터에 속하는지를 알기 위해서 모든 클러스터들 중 가장 큰 $m(k)$ 를, (i.e., $k = \arg(\max(m(1), \dots, m(N)))$), 여기서, N 은 현재 존재하는 클러스터들의 수를 의미한다.), 구하는 것이 필요하다. 만일 $m(k)$ 가 미리 설정한 특정 threshold 값 이상이면 이를 확장 시키기 위해 CTAM 알고리즘을 적용한다. 그림 6은 2-D 정방형의 클러스터에 대한 소속도 분포를 소속도 값이 0.5가 되는 거리를 나타내는 λ 는 1로 고정하고 퍼지화 상수 η 를 0.2부터 10까지 변화시켜 가면서 함수의 특성을 보인 것이다.

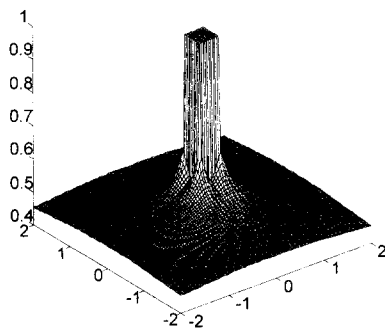
클러스터의 확장에 따라 몇몇 클러스터들의 영역이 겹치는 경우가 발생할 수 있다. 이를 해결하기 위해 SHD 알고리즘을 사용하여 각 클러스터의 vertex들이 다른 클러스터 내에 포함되는지를 조사하게 된다. 만일 하나의 vertex이라도 다른 클러스터에 속해있으면 이는 클러스터간의 겹침이 일어나는 것이므로 클러스터간의 결합을 수행하지 않게 된다. 또한 이러한 두 클러스터간의 겹침치를 조사하는 경우, 클러스터의 중심과 컨벡스 클러스터의 각 vertex를 잇는 선분상의 일정한 간격 위치들이 클러스터의 vertex들로 간주하였다. 마지막으로 아래와 같이 본 알고리즘을 정리하였다.



(a)



(b)



(c)

그림 6. 파라메타 변화에 따른 소속도 함수
(a) $\gamma = 10$, (b) $\gamma = 2$, (c) $\gamma = 0.2$

Fig. 6. Membership function plots for a 2-D square shaped cluster for various fuzzification factors (γ) using a normalized distance (i.e., $\lambda=1$)

(a) $\gamma = 10$, (b) $\gamma = 2$, (c) $\gamma = 0.2$

Fuzzy Convex Clustering(FCC) 알고리즘

Set all necessary initial parameters;

FOR an input pattern x , **DO**

Apply the **SHD** algorithm;

IF x is determined to lie outside all existing clusters **THEN**

Find $\mu_i^*(x)$, for all i and k ;

Find k and $m(k)$ by using (7);

IF $m(k) >$ threshold value θ **THEN**

Apply **CTAM** algorithm to expand k -th cluster;

IF Intra-expandability is larger than a threshold value ? **THEN**

Discard the results of the **CTAM** algorithm;

/* 확장된 클러스터의 크기가 일정 크기 이상인 경우 */

Create a new cluster;

ENDIF

IF Overlapping of a newly expanded cluster by **CTAM** algorithm is detected

THEN

Discard the results of the **CTAM** algorithm.

/* 확장된 클러스터가 다른 클러스터와 겹침이 일어나는 경우 */

Create a new cluster.

ENDIF

ELSE

/* 확장된 클러스터가 가장 가까운 클러스터에 대한 소속도가 일정 이하인 경우 */

Create a new cluster.

ENDIF

ENDIF

ENDFOR

IV. 퍼지 컨벡스 클러스터 결합(Fuzzy Convex Cluster Merging) 알고리즘

FCC 알고리즘을 적용한 후 결과가 너무 많은 클러스터를 발생시킬 수 있다. 이러한 경우를 위해 클러스터 결합 알고리즘을 개발하였다. j 번째 클러스터와 k 번째 클러스터간의 결합은 으로 표기되고 결합된 클러스터는 j 번째 클러스터의 모든 vertex와 k 번째 클러스터간의 모든 vertex들을 포함하게 된다. 이러한 클러스터간의 결합을 수행하기 위해서는 먼저 어떤 클러스터들을 결합할 것인가를 결정하여야 한다. 이를 위해 두 클러스터간의 배척도를 다음 식(8)과 같이 정의하고자 한다. 이러한 클러스터간 배척도의 물리적인 의미는 클러스터 사이의 확장할 수 있는 영역을 뜻하며,

따라서 inter-expandability measure라 명명하고 다음과 같이 정의한다.

$$\epsilon_{inter}(j, k) = \epsilon_{intra}(j \oplus k) - \Delta(j, k)[\epsilon_{intra}(j) + \epsilon_{intra}(k)], \quad (8)$$

여기서, $\Delta(j, k)$ 는 두 클러스터간의 가장 가까운 vertex들의 거리에 반 비례하는 값으로 두 클러스터간의 상관 정도를 나타내는 값이며, 다음식 (9)와 같이 정의된다.

$$\Delta(j, k) = \frac{1}{1 + \min\{\|a_i^k - a_j^j\|\}} \quad (9)$$

따라서 클러스터들의 결합은 두 클러스터 사이의 공간이 적고 두 클러스터 사이의 거리가 가까운 클러스터들을 merge하게 함으로써 수행된다. 물론, 이 경우에도 convex 특성은 계속적으로 만족하도록 한다. 이러한 과정의 종료 조건은 inter-expandability measure가 특정 threshold 값 이내에 있을 때까지 계속된다.

Fuzzy Convex Cluster Merging (FCCM) 알고리즘

Let A^k and A^j be two clusters. And choose an initial value of δ ;

FOR each vertex of A^k **DO**

FOR each vertex of A^j **DO**

Find $\Delta(j, k)$ in (9);

Find $\epsilon_{intra}(j)$ and $\epsilon_{intra}(k)$ in (4);

Find $\epsilon_{inter}(j)$ in (8);

IF $\epsilon_{inter}(j) < \delta$ **THEN**

δ is repeated by $\epsilon_{inter}(j)$;

Perform $j \oplus k$;

FOR each vertex of $j \oplus k$ **DO**

Apply the **SHD** algorithm to eliminate all possible interior vertices of $j \oplus k$;

ENDFOR

ENDIF

ENDFOR

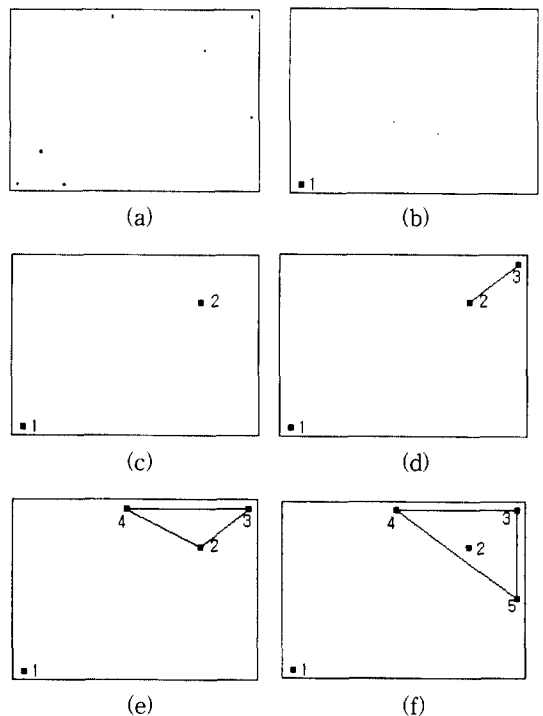
ENDFOR

V. 컴퓨터 시뮬레이션

제안된 방법의 효율성을 입증하기 위해 몇 가지 실험 결과를 제시하고자 한다. 먼저, 그림 7(가)에서와 같이 7개의 입력 vector들이 연속적으로 입력될 때 이를 그림 7(나 - 사)로 FCC 알고리즘을 이용하여

클러스터링하는 과정을 나타내었다. 먼저 그림 7(가)는 7개의 입력 패턴을 나타낸다. 그림 7(나)에서 첫번째 패턴이 입력되고 새로운 클러스터가 생성된다. 다음, 그림 7(다)에서와 같이 2번째 패턴이 입력되고 1번째 클러스터의 외부에 있고 거리가 너무 멀고, 1번째 클러스터로에 대해 낮은 소속도를 나타내는 경우 2번째 클러스터를 생성한다. 그림 7(라)에서 3번째 패턴이 입력되고, 2번째 클러스터로의 높은 소속도를 나타내므로 2번째 컨벡스 클러스터의 새로운 vertex로 등록된다. 그림 7(마)에서 네 번째 패턴이 입력되고, 역시 2번째 클러스터로의 높은 소속도를 나타내므로 2번째 컨벡스 클러스터의 새로운 vertex으로 등록된다. 그림 7(f)에서 5번째 패턴이 2번째 클러스터로의 높은 소속도를 나타내므로 2번째 클러스터의 새로운 vertex으로 등록되며 또한, 2 번째 vertex은 새로운 컨벡스 클러스터의 내부에 존재하므로 삭제된다. 그림 7(바), (사)에서 6,7번째 패턴은 각각 1번째 클러스터에 대해 높은 소속도를 나타내므로 1번째 클러스터의 새로운 vertex으로 등록된다.

다음, 그림 8에서는 FCCM 알고리즘을 수행 과정 및 결과를 나타낸다. 먼저 그림 8(a)는 초기의 FCC 알고리즘을 수행한 후의 모습이며 그림 8(b-d)는 여러 개의 컨벡스 클러스터들을 병합하는 과정을 나타낸다.



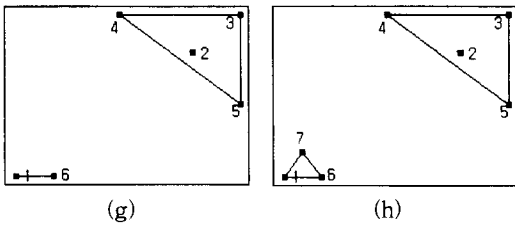


그림 7. FCC 알고리즘의 수행 과정 (b→h)
Fig. 7. A demonstration of the FCC algorithm.

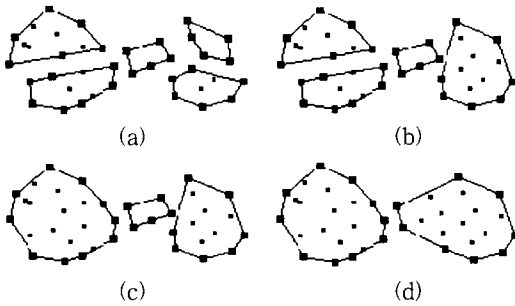


그림 8. FCCM 알고리즘 수행 과정 (a→d)
Fig. 8. A demonstration of the FCCM algorithm.

그림 9에서는 이러한 FCC 알고리즘과 FCCM 알고리즘을 수행하여 Nagy^[11]가 클러스터링하기 힘든 형태로 제안한 데이터 군에 대해 얻은 클러스터링 결과를 보였다. 먼저 그림 9(a)는 클러스터로 가까이 붙은 두개의 데이터 군에 대해 클러스터링을 수행한 결과이다. 그림 9(b)는 두개의 길쭉한 데이터 군의 가까운 거리에 존재할 때의 클러스터링 결과이며, 그림 9(c)는 서로 다른 크기의 두개의 데이터 군이 가까운 거리에 존재할 때의 클러스터링 결과이고, 그림 9(d)는 서로 하나의 데이터 군이 다른 하나의 데이터 군을 싸고 있을 때의 클러스터링 결과이고, 그림 9(e)는 서로 하나의 데이터 밀도를 갖는 2개의 데이터 군에 대한 클러스터링 결과이다.

마지막으로 본 방법과 FMCCNN과의 Clustering 결과를 비교하기 위해 Iris data를 대상으로 2-D 및 4-D에서의 clustering 결과를 소개 하였다. 그림 10의 2-D Iris data의 경우 본 알고리즘으로 실험한 결과 2개의 입력 패턴만이 틀린 클래스로 분류되었고 98개 데이터는 올바른 클래스로 분류되었다. 또한, FMCCNN과의 4-D Iris data에 대한 비교 실험 결과는 다음과 같다. 먼저, 그림 11(a),(b) FMCCNN Clustering 결과는 최대 크기를 0.1로 하였을 때 12개의 분류 오류가 생겼으며, 최대 크기를 이보다 작은 크기

(0.8)로 클러스터링 하였을 때는 좀더 작은 클러스터들이 생기므로, 분류 오류(9개)는 적지만 보다 많은 클러스터들로 이를 표현하게 되었다. 이에 반해 그림 11(c),(d)에서 처럼 FCC & FCCM 알고리즘을 채용한 결과 최대 크기에 해당하는 β 를 0.1로 하였을 때 10개의 분류 오류가 생겼으며, β 를 이보다 작은 크기 (0.8)로 클러스터링 하였을 때는 좀더 작은 클러스터들이 생기므로, 분류 오류(6개)는 적지만 보다 많은 클러스터들로 이를 표현하게 되었다. 또한 위 그림에서 FMCCNN보다는 제안된 알고리즘이 Iris data의 클래스들간의 미세한 경계선 부근을 더 작은 오차를 갖고 클러스터링 하므로 분류 오차를 줄일 수 있었다.

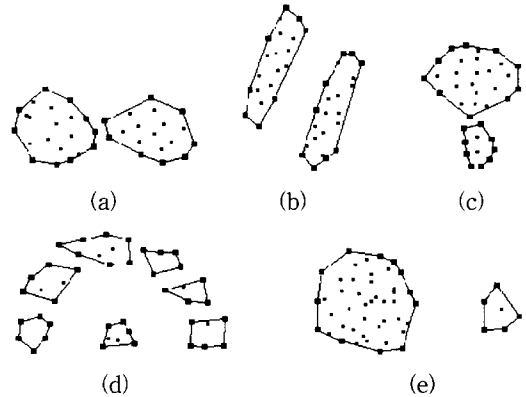


그림 9. Nagy의 데이터에 대해 FCC 알고리즘과 FCCM 알고리즘을 수행하여 얻은 실험 결과
Fig. 9. An experimental results of FCC and FCCM algorithms for Nagy's data set.

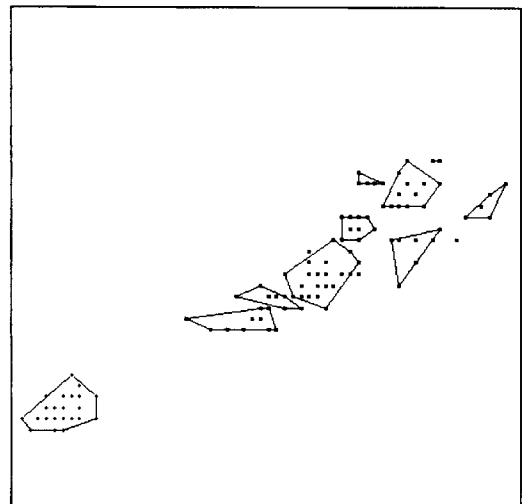


그림 10. IRIS data의 특징 3, 4로 이루어진 data set의 클러스터링 결과
Fig. 10. The 2-level fuzzy clustering results for the iris data using features 3 and 4.

(C : Real Class R: Generated Class)

C \ R	1	2	3
1	50		
2		48	2
3		10	40

(a)

C \ R	1	2	3
1	50		
2		47	3
3		6	44

(b)

C \ R	1	2	3
1	50		
2		46	4
3		6	44

(c)

C \ R	1	2	3
1	50		
2		48	2
3		4	46

(d)

그림 11. 4-D IRIS 데이터의 클러스터링 결과: (a) FMCCNN : (최대크기 : 0.1), (b) FMCCNN (최대크기 : 0.08), (c) FCC & FCCM ($\theta = 0.8, \beta = 0.1, \delta = 0.02$), (d) FCC & FCCM ($\theta = 0.8, \beta = 0.08, \delta = 0.02$)

Fig. 11. Confusion matrices for various maximum allowable cluster sizes of the Iris data using all four features.

(a) FMCCNN : (maximum allowable cluster size : 0.1), (b) FMCCNN (maximum allowable cluster size : 0.08), (c) FCC & FCCM ($\theta = 0.8, \beta = 0.1, \delta = 0.02$), (d) FCC & FCCM ($\theta = 0.8, \beta = 0.08, \delta = 0.02$)

VI. 결론 및 추후 연구 과제

본 논문에서는 convex polytope을 사용하여 새로운 퍼지 클러스터링을 개발 하였다. 여기서 새로운 입력에 대해 어떠한 vertex를 확장 시킬지 혹은 새로운 클러스터를 발생시킬지를 결정하기 위해 비선형 퍼지 소속 함수를 사용하였다. 또한 이렇게 만들어진 여러 개의 클러스터들을 효율적으로 merging하기 위해 intra-클러스터 확장 계수 및 inter-클러스터 확장 계수를 이용한 새로운 merging 알고리즘을 개발하였다. 또한, 시뮬레이션에서 이 두 가지 방법을 온라인으로 결합함으로써 효과적인 클러스터링 결과를 얻을 수 있음을 보였다. 마지막으로 본 논문에서는 제시한 방법의 타당성을 입증하기 위해 컴퓨터 시뮬레이션을 통해 타 방법과의 비교를 수행하였다. 본 방법에서는 여러 가지 파라메타들이 사용되는데 추후에는 이러한 파라메타들을 자동적으로 학습할 수 있는 방법이 요구되는 바 이다.

참고 문헌

- [1] R. N. Dave, Robust Fuzzy Clustering Algorithm, *Proc. 2nd IEEE Int. Conf. Fuzzy Syst.*, March 1993, San Francisco, CA, vol. 1, pp. 1281-1286.
- [2] R. Krishnapuram and J. M. Keller, A possibilistic approach to clustering, *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, May 1993.
- [3] R. Krishnapuram, H. Frigui, and O. Nasraoui, Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation - part I, *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 1, pp. 29-43, Feb. 1995.
- [4] R. Krishnapuram, H. Frigui, and O. Nasraoui, Fuzzy and possibilistic shell clustering algorithms and their application to boundary detection and surface approximation - part II, *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 1, pp. 44-60, Feb. 1995.
- [5] G. Carpenter and S. Grossberg, ART 2: Stable self-organization of pattern recognition codes for analog pattern recognition, *Applied Opt.*, vol. 26, pp. 4919-4930, 1987.
- [6] G. Carpenter, S. Grossberg, and D. Rosen, Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, *Applied Opt.*, vol. 26, pp. 4919-4930, 1987.
- [7] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. NewYork: Plenum, 1981.
- [8] P. Simpson, Fuzzy min-max neural networks-part 2: Clustering, *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 32-45, May 1993.
- [9] 김재현, 서일홍, 퍼지 신경망을 이용한 온라인 클러스터링 방법, *대한전자공학회논문지*, 제31권, B편, 제7호, pp. 199~210, 1994
- [10] H. J. Zimmerman and P. Zysno, Quantifying vagueness in decision models,

European J. Operational Res., vol. 22, pp. 148-158, 1985.

recognition, *Proc. IEEE* 56, pp. 836-882, 1968.

[11] G. Nagy, State of the art in pattern

저 자 소 개

金 載 顯(正會員)

1969년 1월 16일생. 1991년 한양대학교 전자공학과 졸업. 1993년 동 대학원 전자공학과 졸업(석사). 1993년 ~ 현재 동 대학원 전자공학과 박사과정 재학중.

徐 一 弘(正會員)

1977년 서울대 공대 전자공학과 졸업. 1982년 한국 과학 기술원 졸업. 1982년 ~ 1985년 대우중공업 기술연구소 근무. 1987년 ~ 1988년 미국 미시간대 객원 연구원. 현재 한양대 공학대 전자공학과 교수.



李 定 勳(正會員)

1963년 3월 3일생. 1985년 5월 University of Southern California 전자공학사 (BSEE). 1985년 8월 ~ 1987년 5월 University of Missouri 전자공학박사(Ph. D). 1994년 3월 ~ 1995년 8월 한국전자통신연구소 통신시스템 연구단 선임연구원. 1995년 9월 ~ 현재 한양대학교 전자공학과 전임강사. 주관심분야는 컴퓨터비전, 패턴인식, 퍼지시스템 및 신경회로망 모델링.