무기본형 기초의 퍼지 클러스터링에 대한 빠른 접근

황 철, 이 정 훈
Computational Vision and Fuzzy Systems Laboratory
한양대학교 전자공학과
425-791 경기도 안산시 사1동 1271 번지
{chwang,frhee}@fuzzy.hanyang.ac.kr

Abstract

본 논문에서는 패턴 데이터(pattern data) 의 분할(partitioning)위하여, 계산량의 단축할 수 있는 효과적인 퍼지 클러스터링 알고리즘(fuzzy clustering algorithm)을 제시한다. 본 논문에 제시된 알고리즘은 두 단계로 수행된다. 첫번째 단계는, 개선된 FCM(Fuzzy C-means)방법에 의해 입력 패턴들에 대해, 단지 두 번의 반복 수행과정만을 거쳐, 충분히 많은 개수의 초기 클러스터 중심(center)를 결정한다. 다음 단계에서는, 본 논문에 제시될 클러스터 합치기 알고리즘(cluster merging algorithm)을 통해 각 클러스터의 부피(volume)에 따라 클러스터들을 합치는 과정(merging process)을 하게 된다. 결과적으로, 일정한 제한된 개수의 무정형(amorphous)의 클러스터들로 효과적으로 표현될 수 있다. 본 논문의 마지막에 제시될 실험 결과들은 제시된 방법의 유용성을 보여줄 것이다.

1. 서 론

기존의 패턴데이터를 나누는데 사용되는 클러스터링 방법의 대부분은 각 클러스터의 모양을 한가지의 기본형(prototype)으로 고정시키는 기본형 기초의(prototype-based) 방법이다. 그 결과로, 사용된 클러스터(cluster)의 모양에 따라, 기본형 기초의 방법은 선택된 기본형의 모양에 의존해서, 바람직하지 않은 기본형이 선택될 경우, 하나의 클러스터를 가진 패턴들을 여러 개의 클러스터로 바람직하지 않게 분할(partition)할 수 있다. 그러나, 기본형 기초의 방법은 중심(center)이나 지름(radius)과 같은 기본형의 인자(parameter)에 의존해 기본형의 모양을 결정하는 방법으로, 일반적으로 클러스터링 알고리즘에 적용하기에 용이하다. 결론적으로 클러스터링의 결과를 빠르게 얻을 수 있다. 반면에 무기본형 기초의 클러스터링 방법은 기본형 기초의 방법에 비해 클러스터의 모양을 표현하는데 있어, 보다 더 유연성을 갖는다.[2] 그러나, 클러스터링 결과를 얻기까지의 과정에서 각각의 새로운 입력 패턴과 확장될 수 있는 클러스터의 부피의 계산을 포함하는 분할 의 조건은 매우 많은 계산량을 요구한다. 본 논문에서 두 가지 방법의 장점을 응용해 빠르면서도 효과적인 두 단계로 표현되는 클러스터링 방법을 제시하고자 한다. 첫번째 단계에서는 개선된 FCM에 의해 여러 개의 초기 클러스터 중심의 위치가 얻어진다. 이때 FCM의 개선은 다음과 같이 이루어진다. 새로운 클러스터 중심은 각 클러스터 중심들에 대한 해당 입력 패턴의 멤버쉽 값에 따라 이전 클러스터 중심으로부터 얻어진다. 이때 입력패턴의 순서에 초기 중심의 위치 결정이 민감하지 않게 하기 위해 클러스터의 개수를 충분히 크게 잡아 주는 것이 필요하다. 입력된 모든 패턴에 대해 위와 같은 과정을 수행하고 난 결과 클러스터 중심들은 다음단계에서의 초기 클러스터 중심으로 사용된다. 즉 다시 말해 개선된 FCM에서는 초기 중심의 결정을 위한 단계와 실제 패턴들의 클러스터링을 위한 단계 즉 총 두 번의 반복 연산이 필요하게 된다. 다음 단계로는 첫번째 단계에서 분할된 클러스터들을 각 클러스터들의 부피에 따라 합치는 과정을 수행하는 것이다. 이러한 두 단계를 모두 수행하고 난후의 클러스터의 모양은 무정형의 클러스터 모양을 갖게 된다. 본 논문에서 제시한 클러스터링 방법은 기존 FCM을 다음의 세 가지 방향에서 향상시켰다.

1) 패턴데이터에서 클러스터의 개수에 대한 사전 지식이 필요 없다.
2) 분할을 위한 계산량을 감소시킬 수 있다.
3) 적은 수의 무정형의 클러스터들이 얻어질 수 있다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 먼저, 2절에서는 초기 클러스터들을 결정하는 개선된 FCM에 대해 설명한다. 다음, 3절에서는 2절에서 얻어진 클러스터들의 합치는 방법에 관해 기술한다. 4절에서는, 제시된 방법을 적용한 실험결과를 보여준다. 마지막으로 5절에서는 결론이 주어진다.

2. 본 론

A. 초기 클러스터들을 얻기 위한 개선된 Fuzzy C-means 방법

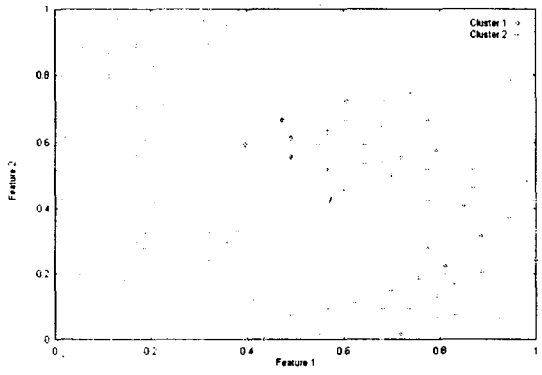
대부분의 FCM기반의 알고리즘은 사용된 기본형에 따라 입력 패턴들이 분할되는 기본형 기반의 방법이다. FCM는 먼저 초기에 C개의 클러스터로 나눌 것이라는 초기 인자를 가지고, c개의 클러스터 중심들을 모든 패턴들과의 퍼지 멤버쉽을 식1에 의해 계산하게 된다.

$$u_{ik} = \left[\sum_{j=1}^{C} \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(m-1)} \right]^{-1} \quad (1)$$

(1)식에서 d_{ik}, d_{jk}는 k번째 패턴과, i번째, j번째의 클러스터 중심들과의 유클리디안 거리(euclidian distance)를 나타낸다. 또, 인자 m은 퍼지화 요소(fuzzification factor)로 사용되었다. 이렇게 각 클러스터 중심들과 모든 입력 패턴들과의 멤버쉽값을 구하고 나면, 기본형의 위치를 아래의 식(2)에 의해 이동시킨다.

$$v_i = \frac{\sum_{k=1}^{N} (u_{ik})^m x_k}{\sum_{k=1}^{N} (u_{ik})^m} \quad (2)$$

(2)의 식에서, N과 x_k는 각각 패턴의 개수와 k번째 패턴임을 나타낸다. 중심을 새로운 위치로 갱신하는 과정은 더 이상 충분히 중심의 위치가 변화하지 않을 때까지 계속 반복한다. 최종 클러스터의 모양은 중심을 기준으로 원의 모양을 그리게 되어 있다. 이때 앞 절에서 설명했던 것처럼, 이런 경우 둥근 원의 형태로 분포되어 있는 패턴들에 대해서는 클러스터링이 용이하지만, 넓게 한쪽 방향으로만 분포되어 있는 클러스터에 대해서는, C의 개수에 따라, 클러스터의 경계에 불필요한 영역을 많이 포함하게 되거나, 또는 하나의 클러스터를 불필요하게 분할하는 경우가 생기게 된다. 아래의 그림 1의 예제는 클러스터의 개수를 2개로 하여 클러스터링을 수행한 결과이다.



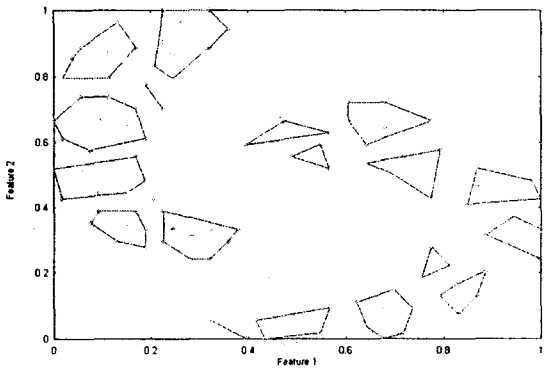
[그림1] 무정형의 분포를 갖는 입력 패턴들에 대한 FCM 클러스터링 결과.

일반적인 FCM은 반복 연산이 끝날 때까지 모든 패턴들에 대해 여러 번 같은 과정을 수행하게 된다. 그러나, 만약 초기의 클러스터 중심의 개수를 충분히 크게 선택한다면, 클러스터 중심의 위치를 결정하기 위해 필요로 되는 반복 연산의 횟수를 줄일 수 있다[4]. 결과적으로, 입력 패턴의 입력 순서에 민감하게 클러스터링 결과가 변화하는 것을 방지하기 위한 연속되는 반복 연산은 불필요하게 된다.

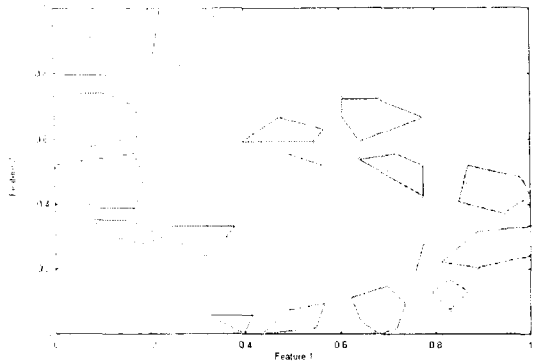
본 논문에서는 입력 패턴들에 대해 단지 두 번의 반복 연산만을 요구하는 개선된 FCM 클러스터링 방법을 제시한다. 두 번의 반복 연산 중 첫번째 연산은 초기 클러스터 중심들의 최종 클러스터 중심으로서의 위치를 결정하기 위한 연산이고, 두 번째 반복 연산은 결정된 최종 클러스터 중심들의 위치를 기준으로 입력 패턴들의 소속 클러스터를 결정하기 위한 단계이다. 클러스터 중심들을 결정하는 과정에서는, 충분히 많은 개수의 클러스터들이 선택이 되고, 모든 클러스터의 중심들은 각 입력 패턴이 입력될 때 마다 입력 패턴과의 멤버쉽 값에 따라 아래 식(3)에 의해 새로운 위치로 갱신된다.

$$v_i^{new} = \frac{v_i^{old} + (u_{ik})^m x_k}{1 + (u_{ik})^m} \quad (3)$$

식 3에서 v_i^{new}와 v_i^{old}는 각각 현재와 이전 중심을 나타낸다. 본 논문에 제시된 방법이 충분히 많은 개수의 클러스터들을 초기에 선택하였을 때, 입력패턴의 순서에 민감하지 않다는 것을 보여주기 위해, 그림 2는 입력패턴의 순서를 각각 다른 순서로 입력하였을 때(C=25)의 결과를 보여주고 있다. 그림2의 클러스터링 결과들과 비교해 그림3에서는 일반적인 FCM방법을 가지고 같은C개수(C=25)를 선택해 클러스터링을 하였을 때의 결과이다. 이를 통해 같은 입력 패턴들에 대해 22번의 반복 연산을 통해 얻어진 그림3의 결과와 의해 2번의 반복 연산만으로 얻어진 그림2의 클러스터링 결과가 매우 비슷하다는 것을 알 수 있다.

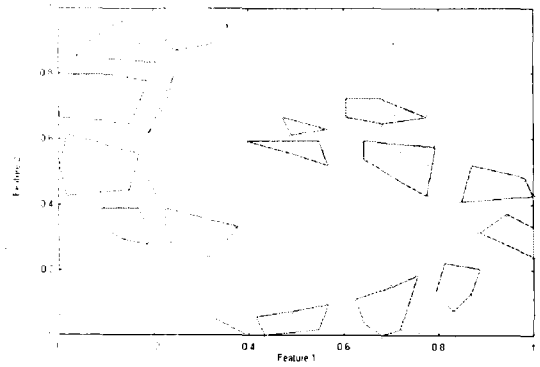


(a)



(b)

[그림 2] C=25 인때 개선된 FCM 방법을 사용하여 입력 패턴 순서에 따른 초기 클러스터링 결과: (a) 원래 패턴 입력 순서, (b) 입력패턴을 무작위의 순서로 입력.



[그림 3] C=25일때 기존의 FCM방법을 사용하여 클러스터링 한 결과.

B. 퍼지 무정형 클러스터링 알고리즘

개선된 FCM 과정을 거쳐 초기 중심들이 결정되고 나면, 여러 개의 클러스터로 분할 되어 있는 클러스터들 간에 합치는 과정을 통해, 클러스터들을 결과적으로 바람직한 모양의 무정형의 몇 개의 클러스터로 확장해 나가는 과정이 필요하게 된다. 그러나 무조건 클러스터를 계속 확장하는 것은 바람직하지 않다. 퍼지 밀집도(fuzzy compactness), 퍼지 부피(fuzzy hyper volume)[5], 클러스터간 확장성(intercluster expandability)[2]등의 클러스터의 확장을 위한 합치는 조건이 필요하게 된다. 여기에서는 하나의 클러스터가 개선되어야 할지 아닐지를 결정하는데 사용되는 내부 클러스터 확장성(intracluster expandability)로서 클러스터의 부피(volume)을 정의 한다. 이러한 부피의 측정하는 방법에 대해 서술 하겠다.

클러스터의 부피를 정확히 측정하는 방법은 클러스터의 각 차원(dimension)을 위해 합리적인 작은 축 간격 l를 선택하고, 내부 단위면적의 개수를 세어 실제와 측정 부피의 오차를 최소화하는 것에 의해 구현될 수 있다[2]. 그러나 이러한 방법은, 낮은 차원의 문제에서는 만족스러운 연산을 수행할 수 있지만, 높은 차원의 문제에서는 많은 계산량을 요구하게 될 수 있다. 따라서, 클러스터의 부피를 측정하는데 있어서, 클러스터안의 모든 패턴벡터와 그 평균 c_j사이의 유클리디안 거리의 평균이 사용된다. 이러한 경우에, R^n차원의 클러스터 A^j에 대한 내부클러스터 확장성 $\varepsilon_{intra}(A^j)$의 측정은

$$\varepsilon_{intra}(A^j) = \frac{1}{n_j}\sum_{i=1}^{n_j}\left\|x_i^j - c_j\right\|^2 \qquad (4)$$

$$\text{where } c_j = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i^j.$$

클러스터를 확장하는 과정을 수행하기 위해 위에서 언급된 내부클러스터 확장성을 사용한다. 합치는 과정은 다음과 같은 순서로 진행된다. 먼저, 합치기 위해 선택된 두개의 클러스터의 부피를 식4에 의해 측정한다. 다음으로, 두개의 클러스터를 합쳤을 때의 클러스터 내부의 부피를 식5에 주어진 클러스터간 확장성에 의해 측정한다.

$$\varepsilon_{inter}(A^i, A^j) = \varepsilon_{intra}(A^i \oplus A^j) - \Delta(A^i, A^j)\cdot [\varepsilon_{intra}(A^i) + \varepsilon_{intra}(A^j)] \qquad (5)$$

식5에서⊕는 클러스터들을 합치는 연산자이고, $\Delta(A^i,A^j)$는 식 6에 주어진 멤버쉽 함수이다.

$$\Delta(A^i, A^j) = \frac{1}{1+\left\|c_i - c_j\right\|} \qquad (6)$$

결국 ⊕는 식 5에 의해 합쳐진 클러스터들의 모든 패턴들을 포함하는 최소부피의 클러스터를 만들어 낸다. 식 5를 사용하면, 클러스터의 합치기 과정은 결과 클러스터는 다른 클러스터와 겹치지않도록 합쳐진 클러스터의 크기가 타당하도록 허락되는 조건에서 클러스터간의 확장성의 연산을 제한해야 한다. 타당한 클러스터의 크기를 갖도록 하기위한 조건은 아래의 식7과 같은 조건으로 주어진다.

$$\varepsilon_{inter}(A^i, A^j) < \min\{\varepsilon_{intra}(A^i), \varepsilon_{intra}(A^j)\}\alpha \qquad (7)$$

위의 식7에서 $\alpha(0<\alpha<1)$ 은 클러스터의 확장을 제어하는 크기조정 요소(scaling factor)이다. 클러스터들을 합치기 위한 조건은 $\alpha\to1(\alpha\to0)$ 일때 더 유연해진다(엄정해진다). 식7과 같은 조건을 만족하는 클러스터쌍이 여러 개 존재할 수 있다. 이러한 경우는, 조건을 만족하는 클러스터쌍 중 가장 작은 클러스터간의 확장성값을 갖는 클러스터쌍을 합치도록 한다. 또한 식4와 식5를 통해 연산하려면 합쳐진 클러스터내의 모든 패턴들의 평균을 계산하는 것으로, 새로운 중심을 필요로 하게 된다. 이러한 과정은

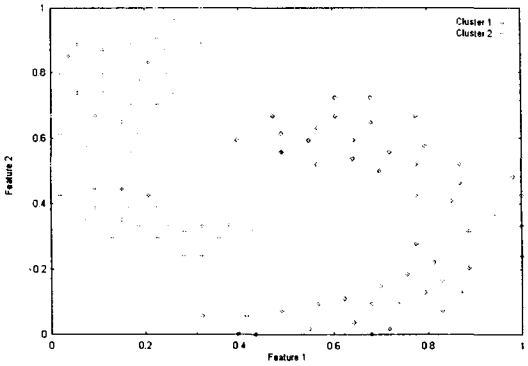
식7의 조건을 만족하는 클러스터쌍이 더 이상 없을 때 까지 계속 수행한다. 개선된 퍼지 클러스터링 알고리즘을 아래와 같이 정리하였다.

Fuzzy Nonprototype-based Clustering Algorithm

Choose C initial cluster centers;
Set an initial value for $m(m>1)$;
FOR all input patterns **DO**
  Obtain the membership value of each cluster center using (1)
  Update all cluster centers with input pattern by (2)
ENDFOR
REPEAT
  Set initial value for α;
  Set δ=LARGE;
  Set merge = FALSE;
  FOR all pairs of clusters (A^i, A^j) **DO**
    Find $\varepsilon_{intra}(A^i \oplus A^j)$, $\varepsilon_{intra}(A^i)$, and $\varepsilon_{intra}(A^j)$ using (4)
    Find $\Delta(A^i, A^j)$ and $\varepsilon_{inter}(A^i, A^j)$ using (6) and (5);
    IF $\varepsilon_{inter}(A^i, A^j) < \min\{\varepsilon_{intra}(A^i), \varepsilon_{intra}(A^j)\}\ \alpha$ **AND** $\varepsilon_{inter}(A^i, A^j) < \delta$ **AND** $A^i \oplus A^j$ dose not overlap with other clusters **THEN**
      Replace δ by $\varepsilon_{inter}(A^i, A^j)$;
      Set $i^*=i$ and $j^*=j$;
      Set merge = TRUE;
    ENDIF
  ENDFOR
  IF merge = TRUE **THEN**
    Perform $A^{i^*} \oplus A^{j^*}$;
    Set index $\{i^*, j^*\}$ to i and resort remaining cluster Indexes in increasing order;
  ENDIF
UNTIL (merge = FALSE)

3. 실험 결과

입력 패턴들을 클러스터링 하기 위해 본 논문에서 제시한 클러스터의 합치기 방법의 효용성을 나타내는 결과를 제시한다. 초기에 필요로 되는 클러스터링 요소들은 다음과 같이 주어졌다. i) 퍼지화 요소 m은 2, ii) 초기 클러스터의 개수 C는 25, iii) 합치기 요소 α 는 0.5로 선택했다. 그림 4는 그림 2에 보여지는 초기 클러스터들을 사용해 클러스터간의 합치기를 수행한 최종 결과를 보여주고 있다. 그림 2의 각 결과는 입력 패턴의 순서에 따라 초기 클러스터들의 모양에는 조금씩 차이가 있었지만, 클러스터 합치기 과정후의 최종 결과는 그림 4와 같이 같은 결과를 보였다. 즉 최종적으로는 입력패턴의 입력 순서에 영향을 거의 받고 있지 않음을 알 수 있다. 또한 그림 1에서 보여준 기존의 FCM과 비교해 볼 때, 좀더 합리적인 클러스터링 결과를 보여주고 있으며 또한 클러스터의 모양에 있어서는 무정형을 나타내고 있다. 이 결과는 본 논문에 제시한 방법이 무정형으로 분포된 패턴들에 대해서 효과적임을 나타낸다.



[그림 4] 그림 2에서의 서로 다른 순서로 입력된 패턴들에 대한 클러스터링 결과.

4. 결 론

본 논문에서 패턴들을 분할하는데 소요되는 계산량을 감소시킬 수 있는 효과적인 퍼지 클러스터링 알고리즘을 제시하였다. 그 방법은 두 단계로 나뉘어 있으며 다음과 같다. 첫번째 단계에서는, 입력 패턴들에 대해 단지 두 번의 반복 연산만을 요구하는 개선된 FCM에 의해 수많은 초기 클러스터들을 결정하는 것이다. 다음 단계에서는 클러스터들은 클러스터 합치기 알고리즘에 의해 각 클러스터의 볼륨을 따라 클러스터들간의 합치기 과정을 통해, 몇 개의 클러스터로 확장해 나아가는 과정이다. 결과적으로, 제한된 개수의 무정형의 클러스터들이 얻어진다. 실험결과는 이러한 제시된 방법이 무정형으로 분포된 패턴들에 대해 효과적이었음을 보여 주었다.

참고 문헌

[1] R. Krishnapuram and J. Keller, "A possibilitic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol.1, no.2, pp. 98-110, May 1993

[2] I. H. Suh, J. H. Kim, and F. C. H. Rhee, "Convex-set-based fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 7, no.3, pp. 271-285, Jun. 1999

[3] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981

[4] L. Tsug and S. Yang, "Genetic algorithms for clustering, feature selection, and classification," in *Proc. 1997 Int. Conf. Neural Networks*, Houston, TX, vol.III, pp. 1612-1616, June 9-12, 1997.

[5] B. Partice, D. Arnaud, and V. Gerard, "Unsupervised fuzzy classification method based on a fuzzy proximity graph and on a graduate hierarchy," in *Proc. 1999 Int. Conf. Fuzzy Systems*, Seoul, Korea, vol.II, pp. 1054-1058, Aug. 22-25, 1999.