

Support Vector Machines 기반의 클러스터 결합 기법

Support Vector Machine based Cluster Merging

최병인 · 이정훈

Byung-In Choi and Frank Chung-Hoon Rhee

한양대학교 전자전기제어계측공학과

요 약

Convex한 클러스터간의 최적의 거리와 Fuzzy Convex Clustering(FCC) 방법에 의한 효과적인 클러스터 결합 알고리즘을 제시하였다. 또한 두 convex한 클러스터간의 거리 측정 방법의 문제점인 정확성과 수행속도 개선하기 위하여 Support Vector Machines(SVM)을 이용한 빠르고 정확한 거리 측정 방법을 제시하였다. 따라서 데이터의 부적절한 표현 없이 클러스터들의 개수를 크게 더 줄일 수 있었다. 본 논문에서는 제시한 알고리즘의 타당성을 위하여 여러 데이터에 대한 실험 결과를 보여주므로써 제시한 알고리즘을 실제 영상 분할에 적용하여 다른 클러스터링 방법의 결과와 비교분석한다.

Abstract

A cluster merging algorithm that merges convex clusters resulted by the Fuzzy Convex Clustering(FCC) method into non-convex clusters was proposed. This was achieved by proposing a fast and reliable distance measure between two convex clusters using Support Vector Machines(SVM) to improve accuracy and speed over other existing conventional methods. In doing so, it was possible to reduce cluster number without losing its representation of the data. In this paper, results for several data sets are given to show the validity of our distance measure and algorithm.

Key words : Support Vector Machines, SVM, Convex Set, Clustering, Fuzzy Convex Clustering

1. 서 론

기존의 Fuzzy C-means(FCM) 기반의 많은 클러스터링 알고리즘은 클러스터의 표현이 하나의 prototype(점, 선, 타원 등등)으로 고정된다[3]. 그 결과로서, 무정형의 클러스터는 FCM 바탕의 클러스터링 기법들에 대하여 올바르게 표현되지 못하는 경우가 있을 수 있다. 이러한 제한을 극복하기 위해서 convex polytopes를 사용하는 Fuzzy Convex Clustering(FCC) 알고리즘이 제안되었다[2]. 이 FCC 알고리즘은 좀 더 유연한 prototype 인 convex polytopes에 의하여 클러스터를 표현하고 이러한 convex 클러스터들이 입력 패턴에 대하여 적응적으로 확장 가능하고 또한 서로 결합하는 것을 수반한다. 따라서 데이터 집합의 클러스터의 개수에 대한 사전 지식 없이도 임의로 분포되어있는 데이터 집합을 보다 더 적절하게 표현할 수 있다[2]. 반면에, 이러한 장점들에도 불구하고 알고리즘에서 사용되는 클러스터들 사이의 거리 측정 방법 또는 클러스터와 입력 패턴들 사이의 거리 측정방법이 다음과 같은 단점을 가지고 있다.

- 1) 느린 수행속도의 문제점
- 2) 오차의 양

또한 클러스터들의 적절하지 못한 표현을 해결하기 위해

결합할 수 있는 클러스터의 크기를 작게 만들어야 한다. 그 결과로서 많은 바람직하지 못한 클러스터들이 생성될 수 있다. 첫 번째 문제점을 극복하기 위해서 Support Vector Machines(SVM)을 이용하여 최적의 두 convex 클러스터간 거리 측정 방법을 제안하였다[1]. 또한, FCC 알고리즘의 결과로서 생성되는 바람직하지 못한 클러스터들을 효과적으로 결합할 수 있는 클러스터 결합 알고리즘을 제안한다. 제안한 알고리즘에서는 위에서 제안한 거리 측정 방법을 두 클러스터들 간 유사도로 사용하여 minimum spanning tree를 이루고, 그 tree중 적절한 유사도를 한계 값으로 설정하여 이보다 작은 모든 클러스터 쌍들을 결합시켜준다. 따라서 무정형의 클러스터를 표현하는 클러스터들의 개수를 데이터의 적절한 표현을 잃지 않고도 효과적으로 더욱 줄일 수 있었다.

본 논문의 나머지 부분은 다음과 같이 구성되어 있다. 2절과 3절에서는 FCC 알고리즘과 SVM의 기본적인 특성을 간략히 소개한다. 4절에서는 SVM을 바탕으로 한 클러스터들 사이의 거리 측정 방법에 대하여 설명하고, 5절에서는 제안된 거리 측정 방법을 바탕으로 convex 클러스터 결합 알고리즘을 설명한다. 마지막으로, 6절에서는 제안한 방법들의 타당성을 보이기 위해서 몇 개의 데이터 집합들에 대한 결과를 제시하고, 실제적인 영상 분할 예제에 적용하여 FCM과 비교하여 제시한 알고리즘의 우수성을 보인다.

2. Overview of Fuzzy Convex Clustering(FCC) Algorithm

FCC 알고리즘은 기존의 클러스터링 기법들이 고정된 prototype에 의하여 클러스터들을 표현한다는 단점을 극복하

접수일자 : 2004년 4월 28일

완료일자 : 2004년 6월 3일

감사의 글 : 본 연구는 한국과학기술원 영상정보특화연구센터를 통한 국방과학연구소의 연구비 지원으로 수행되었습니다

기 위하여 클러스터들을 convex set에 의한 convex한 도형으로서 표현하였다[2]. 그림 1(b)에서 볼 수 있듯이 클러스터를 표현하기 위하여 convex한 도형을 prototype으로 사용하였다. 그러므로 클러스터의 표현이 기존의 타원, 선, 또는 점이라는 고정된 prototype들 보다 올바른 클러스터의 표현이 가능하게 되었다.

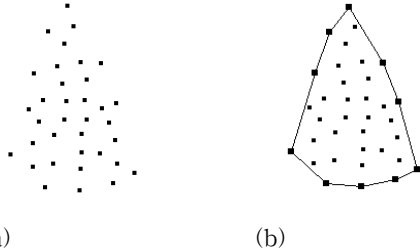


그림 1. FCC 알고리즘에 의한 클러스터 표현: (a) 원 데이터 집합 (b) convex prototype에 의하여 표현된 데이터 집합.
Fig. 1. Representation of a cluster by the FCC algorithm: (a) the original data set (b) the data set represented by a convex prototype.

FCC 알고리즘은 각 입력 패턴에 대하여 단지 convex하다는 제약조건만을 가지고 최대 클러스터 크기(β)에 의하여 입력 패턴과 convex 클러스터를 확장 또는 새로운 클러스터를 생성하고, 클러스터의 개수를 줄이기 위하여 최대 결합 클러스터 크기($\Delta\beta$)에 의하여 결합 가능한 convex 클러스터들을 결합한다. 그러므로 클러스터의 개수에 대한 사전 지식이 필요 없다. 그림 2는 $\Delta\beta$ 의 변화에 따른 FCC 알고리즘의 결과의 예를 나타낸다.

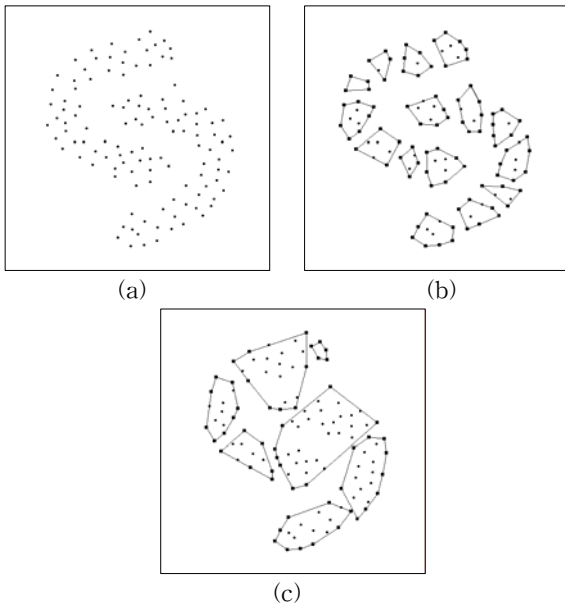


그림 2. FCC 알고리즘의 결과 (a) 원 데이터 집합 ($\Delta\beta = 0.0008$ 일 때) (c) $\Delta\beta = 0.0028$ 일 때.
Fig. 2. Results of FCC algorithm (a) the original data set (b) the result for $\Delta\beta = 0.0008$ (c) $\Delta\beta = 0.0028$.

그림 2(b)에서 볼 수 있듯이 $\Delta\beta$ 를 작게 하면 많은 작은

클러스터들이 생성되고, 클러스터의 개수를 줄이기 위하여 $\Delta\beta$ 를 크게 하면 그림 2(c)와 같이 클러스터의 표현이 부적절하게 될 수 있다. 따라서 본 논문에서는 $\Delta\beta$ 를 작게 하고, SVM을 이용하여 생성된 많은 클러스터들을 적절하게 결합하고자 한다.

3. Support Vector Machines (SVM)

선형적으로 분리 가능한 두 데이터 집합에 대하여, SVM은 다차원 평면과 그 평면에서 가장 가까운 멤버 (support vector) 들 간의 거리 (margin)이 최대이고, 두 데이터 집합을 분할하는 최대 마진 다차원 평면 (maximal margin hyperplane)을 찾을 수 있다[4]-[7]. 이때 최대 마진 다차원 평면은 최적으로 두 데이터 집합을 분할한다. 즉, 두 데이터 집합을 나누는 임의의 다차원평면이 다른 다차원 평면보다 더 큰 최소 마진(minimum margin)을 가진다면, 패턴들이 다차원평면에서 더 멀리 떨어지므로 더욱 신뢰할 수 있게 된다. 따라서 가장 큰 최소 마진을 가지는 최대 마진 다차원 평면은 두 데이터 집합을 나누는 최적의 다차원 평면임을 나타낸다. 주어진 l 개의 패턴 쌍 (x_i, y_i) 들에 대하여, 최대 마진 다차원 평면 $\langle wx \rangle + b$ 는 다음의 quadratic programming problem을 푸는 것에 의해서 얻어질 수 있다.

$$\begin{aligned} & \text{minimize}_{\langle wx \rangle + b} && \langle w \cdot w \rangle, \\ & \text{subject to} && y_i (\langle w \cdot x_i \rangle + b) \geq 1, \\ & && i = 1, \dots, l, \end{aligned} \tag{2.1}$$

여기서 파라미터들인 w , b 와 $y_i = 1$ 는 각각 가중치 벡터(weight vector), 바이어스 (bias), 그리고 클래스 라벨(class label)을 나타낸다.

식 (2.1)의 제한 조건은 다음의 목적 함수 $W(\alpha)$ 를 최대화하는 것으로 단순화할 수 있다.

$$\begin{aligned} & \text{maximize} && W(\alpha) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j \langle x_i \cdot x_j \rangle, \\ & \text{subject to} && \sum_{i=1}^l y_i \alpha_i = 0, \alpha_i \geq 0, \\ & && i = 1, \dots, l, \end{aligned} \tag{2.2}$$

여기서 α 는 Lagrange multiplier를 나타낸다. 위의 2차 최적화 문제를 푸는 것으로 최적의 가중치 벡터 w^* 와 bias b^* 를 다음과 같이 얻을 수 있다.

$$w^* = \sum_{i=1}^l y_i \alpha_i^* x_i, \tag{2.3}$$

$$b^* = - \frac{\max_{y_i=1} (\langle w^* \cdot x_i \rangle) + \min_{y_i=-1} (\langle w^* \cdot x_i \rangle)}{2} \tag{2.4}$$

여기서, w^* 는 마진이 $1/\|w^*\|$ 인 최대 마진 다차원 평면을 형성한다. 선형적으로 분리가 불가능한 데이터 집합에 대해서는 분리 불가능한 정도를 나타내는 변수 (slack variable) ξ_i 를 위의 최적화 문제에 첨가하는 것이 요구되고 다음과 같이 표현될 수 있다.

$$\xi_i = \max(0, \gamma - y_i(\langle w \cdot x_i \rangle + b)) \quad (2.5)$$

식 (2.2)에 ξ_i 를 포함하면 다음과 같은 2차 최적화 문제를 얻을 수 있다.

$$\begin{aligned} & \text{maximize} && W(\alpha) \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j \langle x_i \cdot x_j \rangle, \\ & \text{subject to} && \sum_{i=1}^l y_i \alpha_i = 0, \\ & && C > \alpha_i \geq 0, \quad i=1, \dots, l, \end{aligned} \quad (2.6)$$

여기서 C 는 양의 상수이다.

식(2.6)이 식 (2.2)에서 단지 α 에 상한 값 C 를 첨가한 것과 같으므로, 식 (2.6)의 해답은 식(2.3), (2.4)와 같다.

3. SVM을 이용한 클러스터간 거리 측정

FCC 알고리즘에서는 convex 클러스터들 사이의 거리를 구하기 위하여 클러스터들의 꼭지점들 간 최소 거리와 내부의 단위 셀들 간 최소 거리를 사용하는 두 가지 거리 측정 방법이 제안되었다[2]. 전자의 경우에는 빠른 거리측정이 가능하지만 부정확할 수 있기 때문에 두 클러스터들 사이의 거리를 적절하지 못하게 표현하는 경우가 있다. 후자의 경우는 결과가 매우 정확하지만 이를 위하여 많은 셀들을 사용하여야 하고, 이 셀들이 클러스터의 내부에 존재하는지를 판별해야 하는 과정이 수반되므로 수행 속도가 매우 느릴 수 있고 입력 패턴의 차원이 높아지면 계산량이 기하급수로 커진다. 본 논문에서는 수행 속도와 정확성을 만족시키는 SVM을 이용한 새로운 클러스터 간 거리 측정 방법을 제시한다. FCC 알고리즘의 특성에 따르면, 각각의 클러스터는 반드시 convex 하고 선형적으로 분리 가능하다. 이러한 특성에 의하여 식 (2.2)에 의해 두 클러스터들 사이의 최대 마진 다차원 평면을 구할 수 있다. 따라서 두 convex 클러스터 간 최적의 거리는 최적의 다차원 평면과 support vector사이의 거리인 최소 마진 γ 의 두 배가 된다.

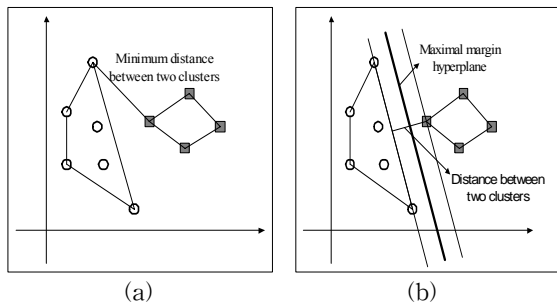


그림 3. (a) Convex 클러스터들의 꼭지점들을 이용한 거리 측정 방법과 (b) SVM을 이용한 거리 측정 방법.

Fig. 3. Distance measure using (a) vertices of convex clusters and (b) SVM.

그림 3(a)는 두 convex 클러스터들의 꼭지점을 사용한 클러스터간 거리를 나타낸다. 그 결과로서, FCC 알고리즘에서 convex 클러스터들이 확장되거나 결합할 때 잘못된 거리를 나타낼 수 있다. 그러나 그림 3(b)는 최적의 클러스터 간 거

리를 구할 수 있다.

SVM에 의한 최대 마진 다차원 평면의 가중치 벡터 w^* 와 바이어스 b^* 를 사용하여, 두 convex 클러스터 A^k 와 A^j 사이의 거리는 다음과 같이 표현된다.

$$d_{svm}(A^k, A^j) = 2 \cdot \frac{\min_{x_i \in A^k \text{ or } A^j} \left(\left\| w^{*T} x_i + b^* \right\| \right)}{\|w\|}. \quad (3.1)$$

Convex 클러스터들이 선형적으로 분리 가능하므로 SVM 학습 알고리즘이 빠르게 수렴할 수 있다. 또한 최대 마진 다차원 평면은 단지 support vector에 의하여 생성되기 때문에 support vector들이 속해있는 꼭지점들의 집합을 사용하여 SVM 학습 알고리즘을 수행할 수 있다. 그러므로 수행속도에 가장 영향을 미치는 데이터의 개수를 크게 줄일 수 있으므로 큰 수행 속도의 이득을 얻을 수 있다. SVM의 특성상 입력 벡터의 차원에 무관하므로 높은 차원의 데이터에 대하여 좋은 수행 결과를 얻을 수 있다.

4. 클러스터 결합 알고리즘

FCC 알고리즘에서 클러스터의 개수를 줄이기 위하여 최대 결합 클러스터 크기, $\Delta\beta$ 를 정함으로써 초기의 convex 클러스터들이 좀 더 큰 클러스터들로 결합하게 된다[1]. 그러나 만약 $\Delta\beta$ 가 너무 크게 정해지면 클러스터들이 반드시 convex 해야 한다는 제한 조건 때문에 클러스터의 부적절한 표현을 야기할 수 있다. 그림 2(b)에서 볼 수 있듯이, 클러스터들이 $\Delta\beta$ 가 증가함에 따라서 바람직하지 못하게 확장된다. 이 문제를 해결하기 위해서 결합 후에 convex 해야 한다는 제한 조건을 완화함에 의해서 convex 클러스터들을 결합하는 알고리즘을 제안한다.

먼저 그림 2(a)처럼 $\Delta\beta$ 를 적당한 크기로 설정한 후 FCC 알고리즘을 수행하여 convex 클러스터들을 구한다. 다음으로 두 클러스터들 사이의 거리를 similarity로 사용하여 클러스터 쌍들을 오름 차 순으로 정렬한 뒤에 각 클러스터들 간의 minimum spanning tree를 생성한다. 그 거리는 다음과 같이 주어진다.

$$\delta_{kj} = d_{svm}(A^k, A^j), \quad (3.2)$$

여기서 d_{svm} 은 SVM을 사용한 클러스터 A^k 와 A^j 간의 거리 측정 방법이다. 다음으로 $\Delta\delta$ 를 클러스터간 거리의 결합 한계 값으로 설정하여 tree의 클러스터 쌍들 중 거리가 $\Delta\delta$ 이하인 결합이 고려되는 클러스터들을 정렬한 순서에 따라서 결합시켜준다. 본 알고리즘은 결합되는 과정에서 알 수 있듯이 클러스터들이 convex 해야 한다는 조건이 필요 없기 때문에 non-convex한 클러스터들의 결합 알고리즘이라고 볼 수 있다. convex한 클러스터들을 설명한 것처럼 그림 4(a)에서는 그림 2의 클러스터들에 대하여 선택된 클러스터 쌍들을 제시한 알고리즘에 따라 묶어주었고, 그 결과 선으로 연결된 convex 클러스터들이 두 개의 non-convex한 클러스터들로 결합되는 것을 보여준다. 그림 4(b)는 convex 클러스터들에 대한 결합되는 클러스터 배열을 나타내는 점진적 계층도(graduated hierarchy)를 보여준다[8].

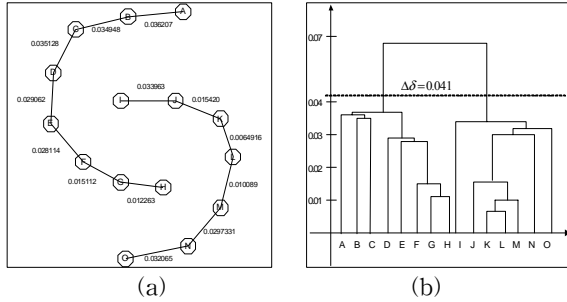


그림 4. 제안한 클러스터 결합 알고리즘의 예(그림 2.(a)) (a) $\Delta\beta$ 가 0.041일 때의 근접도 그래프 (b) (a)의 점진적 계층도.

Fig. 4. An example of the proposed cluster merging algorithm(Fig 2.(a)): (a) proximity graph for $\Delta\delta=0.041$ (b) graduated hierarchy of (a).

제안한 SVM 클러스터 결합 알고리즘은 다음과 같이 요약할 수 있다.

SVM Cluster Merging Algorithm (SCM)

```

Initialize  $\Delta\delta$ ;
FOR all cluster pairs DO
    Calculate weight vector  $w^*$  by SVM for cluster pair  $(A^k, A^j)$ ;
    Set  $\delta_{kj} = d_{svm}(A^k, A^j)$ ;
    Store cluster pairs and  $\delta_{kj}$ ;
END FOR
Sort cluster pairs in increasing order according to  $\delta_{kj}$ ;
Find a minimum spanning tree of cluster pairs;
FOR all cluster pairs of the tree DO
    IF ( $\delta_{kj} < \Delta\delta$ ) THEN
        Merge the cluster pairs into one cluster;
    END IF
END FOR
    
```

제안된 알고리즘에 의하면 적절한 $\Delta\delta$ 의 선택이 중요하다. 그 방법으로는 minimum spanning tree에서 가장 작은 similarity, 곧 가장 큰 클러스터간 거리를 $\Delta\delta$ 의 값으로 선택할 수 있다. 즉, 클러스터간 거리가 가장 크다는 것은 각 클러스터들이 가장 멀리 떨어져 있다는 것을 나타낼 수 있다. 그림 4 에서는 가장 큰 클러스터간 거리가 0.070을 나타내므로 $\Delta\delta$ 는 0.070으로 선택할 수 있다. 그 때의 적절한 클러스터의 개수는 2개가 된다.

5. 실험 결과

제시된 거리 측정 방법과 알고리즘의 타당성을 보이기 위하여 여러 가지 데이터 집합에 대하여 실험을 하였다. 먼저 convex 클러스터들의 꼭지점들 간의 최소 거리와 SVM을 사용한 두 가지 거리 측정 방법을 사용한 제안된 알고리즘의 결과를 비교하였고 2차원 데이터들과 Iris 데이터에 대한 제안된 알고리즘의 결과를 보인다. 다음으로 실제 영상의 Fuzzy C-means(FCM) 알고리즘과 제안된 알고리즘의 영상

분할 결과를 비교한다.

5.1 클러스터간 거리 측정 방법 비교

제안된 알고리즘에서 $\Delta\delta$ 을 0.041로 설정했을 때, 그림 5는 선에 의하여 결합된 클러스터들을 보여준다. 그림 5(a)에서는 몇 개의 클러스터들이 결합되기에 충분히 근접함에도 불구하고 그 클러스터들이 앞서 언급된 것처럼 정확하지 못한 거리의 표현 때문에 결합 되지 않았다. 반면에 그림 5(b)에서는 좀 더 정확한 클러스터들 간의 거리의 표현으로 인하여 모든 convex 클러스터들이 2개의 클러스터들로 결합하였다.

5.2 Several 2-D data set

제안된 알고리즘의 타당성을 보이기 위하여 2 차원 데이터 집합들에 대하여 실험을 하였다. 그림 6은 FCC 알고리즘의 최대 클러스터 크기 $\beta = 0.001$ 로 놓고 FCC 알고리즘을 수행하여 형성된 convex 클러스터들에 대한 제안된 알고리즘의 결과를 보여 준다. 그림 6에서 왼쪽 열은 FCC 알고리즘의 결과를 나타내고, 오른쪽 열은 제안된 알고리즘의 결과를 나타낸다. 그림 6(a)에서 $\Delta\delta$ 는 0.070로 선택되고, 제안된 알고리즘은 FCC 알고리즘에 의하여 생성된 8 개의 convex 클러스터들을 직관적으로 적절한 클러스터의 개수로 예상할 수 있는 두 개의 non-convex 클러스터로 결합된다. 유사하게 그림 6(b)에서는 $\Delta\delta$ 가 0.092로 선택되고, 15개의 클러스터들이 2개로 결합된다. 마지막 예로서, 그림 6(c)는 데이터 집합이 직관적으로 3개의 클러스터들로 나타내어진다는 것을 보인다. $\Delta\delta$ 는 0.068로 설정되고, 14개의 클러스터들이 3개로 결합된다. 세 가지 예에서 볼 수 있듯이, 부적절한 데이터의 표현을 야기하지 않고 다수의 클러스터들이 결합되었다. FCC 알고리즘의 convex라는 제한 조건을 완화하고, 좀 더 적절한 클러스터들 간의 거리 측정 방법을 사용함으로써 제안된 알고리즘은 현저히 클러스터 효율을 개선할 수 있었고 주어진 데이터 집합에 대한 효과적인 표현을 이룰 수 있었다.

5.3 Iris data set

Iris 데이터는 4개의 속성과 3개의 클래스가 있고 각 클래스당 50개의 패턴들로 구성되어있다. 속성 1과 2는 각 클래스의 많은 패턴들의 속성 값들이 중첩되어 있기 때문에 각 클래스의 특성을 잘 나타내지 못하는 반면에 세 번째와 네 번째 속성은 잘 나타내고 있다. 따라서 본 논문에서는 세 번째와 네 번째 속성을 사용하여 제시한 알고리즘을 수행하였다. 그림 7(a)는 FCC알고리즘의 수행 결과를 confusion matrix로 나타낸 것이고 나머지 결과는 $\Delta\delta$ 의 값의 변화에 따른 클러스터링 결과를 나타내었다. $\Delta\delta$ 가 커짐에 따라 클래스 2와 3에 관련된 클러스터들이 묶이게 되는데, 이는 두 클래스가 서로 중첩되어있기 때문이다. $\Delta\delta$ 가 0.091일 경우에는 Iris 데이터의 특성상 클래스 2와 3이 하나의 클러스터로 결합된다.

5.4 Image segmentation

마지막으로 FCM 알고리즘과 제안된 알고리즘의 영상 분할 결과를 비교한다. 그림 9(a)는 제시된 200*200 영상이다. 원 영상에 대한 excess green과 median 값을 속성으로 선택하여 길과 숲과 하늘의 영역에서 90개씩 총 270개의 sample 데이터를 랜덤하게 추출하였다. 이 sample 데이터를 사용하여 FCM 알고리즘과 제시한 알고리즘을 수행하였다. 그리고

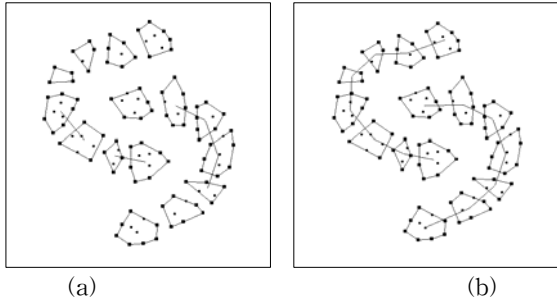


그림 5. 두 거리 측정 방법을 사용한 SCM 알고리즘의 결과: (a) 꼭지점을 이용한 거리 측정 방법 (b) SVM을 이용한 거리측정 방법.
 Fig. 5. Results of SCM for two distance measures : (a) distance measure using vertices (b) distance measure using SVM.

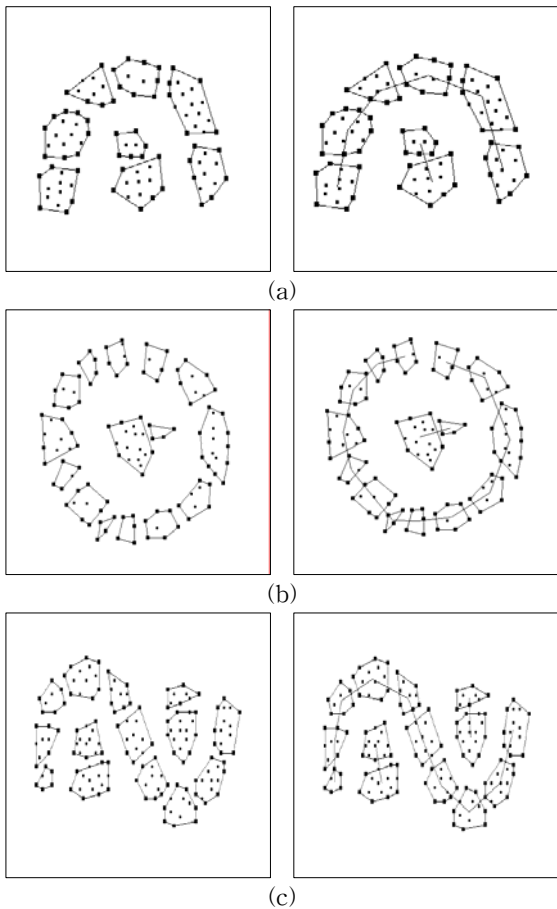


그림 6. 여러 데이터에 대한 SCM 알고리즘의 결과. 왼(오른) 쪽 열은 FCC(SCM) 알고리즘의 결과 (a) $\Delta\delta = 0.070$ 일 때 클러스터 개수(NC)=2 (b) $\Delta\delta = 0.092$ 일 때 NC=2 (c) $\Delta\delta = 0.068$ 이고 NC=3.
 Fig. 6. Results of proposed SCM algorithm for various data sets. Left(right) column are the results of the FCC(SCM) algorithm. (a) $\Delta\delta = 0.070$ and Number of Clusters(NC)=2 (b) $\Delta\delta = 0.092$ and NC=2 (c) $\Delta\delta = 0.068$ and NC=3.

원 영상의 모든 픽셀에 대한 각 클러스터들의 소속 결과를 살펴보았다. 그림 8은 제안된 알고리즘의 결과를 confusion matrix로 표현한 것이다. 그림 9(b)는 클러스터의 개수가 3 일 때의 (NC=3) FCM 알고리즘의 결과를 나타낸다. 그림 9(c)는 FCC 알고리즘에 의하여 생성된 24개의 convex 클러스터들을 제안된 알고리즘에 의하여 3개의 클러스터들로 결합했을 때의 영상분할 결과이다. 그 때의 $\Delta\delta$ 의 값은 0.115 이다. 그림에서 볼 수 있듯이 숲의 noise 부분에서 FCM보다 좋은 결과를 나타내고 있다. 또한 길과 숲의 경계부분에서도 FCM에 비하여 좋은 분할 결과를 나타내고 있다.

$\Delta\delta = 0.000$ 클러스터 개수 = 13 분류 정확도 = 96.7%	$\Delta\delta = 0.034$ 클러스터 개수=9 분류 정확도=94.7%																																						
<table border="1"> <thead> <tr> <th rowspan="2">True Class</th> <th colspan="3">Recognized Class</th> </tr> <tr> <th>1</th> <th>2</th> <th>3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>50</td> <td>0</td> <td>0</td> </tr> <tr> <td>2</td> <td>0</td> <td>47</td> <td>3</td> </tr> <tr> <td>3</td> <td>0</td> <td>2</td> <td>48</td> </tr> </tbody> </table>	True Class	Recognized Class			1	2	3	1	50	0	0	2	0	47	3	3	0	2	48	<table border="1"> <thead> <tr> <th rowspan="2">True Class</th> <th colspan="3">Recognized Class</th> </tr> <tr> <th>1</th> <th>2</th> <th>3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>50</td> <td>0</td> <td>0</td> </tr> <tr> <td>2</td> <td>0</td> <td>50</td> <td>0</td> </tr> <tr> <td>3</td> <td>0</td> <td>8</td> <td>42</td> </tr> </tbody> </table>	True Class	Recognized Class			1	2	3	1	50	0	0	2	0	50	0	3	0	8	42
True Class		Recognized Class																																					
	1	2	3																																				
1	50	0	0																																				
2	0	47	3																																				
3	0	2	48																																				
True Class	Recognized Class																																						
	1	2	3																																				
1	50	0	0																																				
2	0	50	0																																				
3	0	8	42																																				
(a)	(b)																																						
$\Delta\delta = 0.045$ 클러스터 개수=5 분류 정확도=89.3%	$\Delta\delta = 0.091$ 클러스터 개수=2 분류 정확도=66.7%																																						
<table border="1"> <thead> <tr> <th rowspan="2">True Class</th> <th colspan="3">Recognized Class</th> </tr> <tr> <th>1</th> <th>2</th> <th>3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>50</td> <td>0</td> <td>0</td> </tr> <tr> <td>2</td> <td>0</td> <td>50</td> <td>0</td> </tr> <tr> <td>3</td> <td>0</td> <td>16</td> <td>34</td> </tr> </tbody> </table>	True Class	Recognized Class			1	2	3	1	50	0	0	2	0	50	0	3	0	16	34	<table border="1"> <thead> <tr> <th rowspan="2">True Class</th> <th colspan="3">Recognized Class</th> </tr> <tr> <th>1</th> <th>2</th> <th>3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>50</td> <td>0</td> <td>0</td> </tr> <tr> <td>2</td> <td>0</td> <td>50</td> <td>0</td> </tr> <tr> <td>3</td> <td>0</td> <td>50</td> <td>0</td> </tr> </tbody> </table>	True Class	Recognized Class			1	2	3	1	50	0	0	2	0	50	0	3	0	50	0
True Class		Recognized Class																																					
	1	2	3																																				
1	50	0	0																																				
2	0	50	0																																				
3	0	16	34																																				
True Class	Recognized Class																																						
	1	2	3																																				
1	50	0	0																																				
2	0	50	0																																				
3	0	50	0																																				
(c)	(d)																																						

그림 7. 여러 $\Delta\delta$ 에 대한 Iris 데이터의 제안된 알고리즘의 결과.

Fig. 7. Results for various $\Delta\delta$ of the "Iris data" using the proposed algorithm.

클러스터 개수=3 분류 정확도=88.31%	클러스터 개수=3 분류 정확도=85.61%																																						
<table border="1"> <thead> <tr> <th rowspan="2">True Class</th> <th colspan="3">Recognized Class</th> </tr> <tr> <th>1</th> <th>2</th> <th>3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>8820</td> <td>0</td> <td>0</td> </tr> <tr> <td>2</td> <td>3924</td> <td>20777</td> <td>745</td> </tr> <tr> <td>3</td> <td>6</td> <td>0</td> <td>6328</td> </tr> </tbody> </table>	True Class	Recognized Class			1	2	3	1	8820	0	0	2	3924	20777	745	3	6	0	6328	<table border="1"> <thead> <tr> <th rowspan="2">True Class</th> <th colspan="3">Recognized Class</th> </tr> <tr> <th>1</th> <th>2</th> <th>3</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>8219</td> <td>1</td> <td>0</td> </tr> <tr> <td>2</td> <td>4901</td> <td>19701</td> <td>844</td> </tr> <tr> <td>3</td> <td>9</td> <td>0</td> <td>6325</td> </tr> </tbody> </table>	True Class	Recognized Class			1	2	3	1	8219	1	0	2	4901	19701	844	3	9	0	6325
True Class		Recognized Class																																					
	1	2	3																																				
1	8820	0	0																																				
2	3924	20777	745																																				
3	6	0	6328																																				
True Class	Recognized Class																																						
	1	2	3																																				
1	8219	1	0																																				
2	4901	19701	844																																				
3	9	0	6325																																				
(a)	(b)																																						

그림 8. 영상분할 결과 (a) SCM 알고리즘 (b) FCM 알고리즘.

Fig. 8. Image segmentation results for the (a) SCM and (b) FCM algorithms.

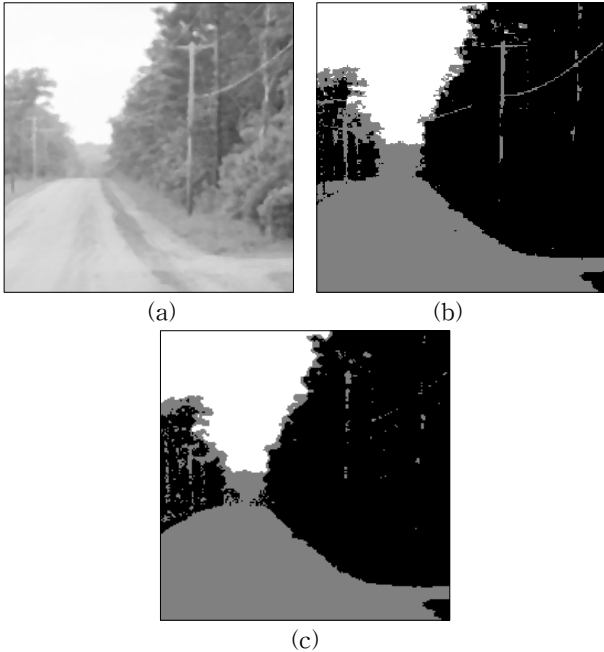


그림 9. (a) 원영상 (b) FCM 알고리즘의 영상분할 결과 (c) SCM 알고리즘의 영상분할 결과.
 Fig. 9. (a) Original image. Segmentation result of the (b) FCM algorithm and (c) SCM algorithm.

5. 결 론

본 논문에서는 제안된 SVM을 사용하여 두 클러스터들 간의 빠르고 최적한 거리 측정 방법과 FCC 알고리즘에 의하여 형성된 convex 클러스터들을 무정형의 클러스터들로 묶어주는 올바른 클러스터 결합 알고리즘에 대한 여러 데이터의 결과와 실제 영상분할에 적용하여 그 결과를 FCM의 결과와 비교하였다. 여러 데이터에 대한 실험 결과에서 볼 수 있듯이 데이터에 대한 적절한 클러스터 표현을 유지하고, 클러스터의 개수를 주목할 만하게 줄일 수 있었다. 실제적인 영상 분할의 결과에서는 기존의 FCM에 의한 결과에 비하여 경계 영역에서의 인식과 noise에 대하여 향상된 결과를 보였다. 이는 각각의 클러스터들이 FCC 알고리즘에 의하여 표현되고 이를 제안된 알고리즘에 의하여 non-convex한 클러스터들로 결합된다. 따라서 기존의 FCM 알고리즘의 point prototype에 비하여 보다 더 올바른 클러스터의 표현에 의하여 나온 결과를 얻을 수 있었다. 제안된 클러스터 간 유사도에 대하여 보다 더 noise에 강한 특성과 효율적인 클러스터의 결합을 위하여 부피, 밀도 등과 같은 여러 클러스터 파라미터들을 결합한 퍼지 소속도 함수가 고려될 수 있을 것이다. 또한 적절한 한계 값의 선택에 대한 문제 또한 연구될 수 있을 것이다.

참 고 문 헌

[1] 최병인, 이정훈, "Support Vector Machines를 이용한 Convex 클러스터 결합 알고리즘," *퍼지 및 지능시스템 학회 추계학술대회 발표논문집*, pp. 267-270, 2002.
 [2] I. Suh, J. Kim, and Frank C.-H. Rhee,

"Convex-set-based fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 3, pp. 271-285, June 1999.
 [3] J. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.
 [4] Y. Freund and R. Schapire, "Large margin classification using the perceptron algorithm," *Machine Learning*, vol. 37, no. 3, pp. 277-296, 1999.
 [5] T. Friess, N. Cristianini, and C. Campbell, "The kernel-adatron: a fast and simple training procedure for support vector machines," in *Machine Learning: Proceedings of the Fifteenth International Conference*, J. Shavlik Ed. Morgan Kaufmann, 1998.
 [6] V. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, 1998.
 [7] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.
 [8] B. Patrice, D. Anaud, and V. Gerard, "Unsupervised fuzzy classification method based on a fuzzy proximity graph and on a graduated hierarchy," *IEEE Int. Conf. Fuzzy Syst.*, vol. 2, pp. 1054-1058, August 1999.

저 자 소 개



최병인(Byung-In Choi)

2001년 : 한양대학교 전자공학과 졸업
 2001년~2003년 : 동 대학원 전자전자제어계측공학과 석사
 현재 : 동 대학원 전자전자제어계측공학과 박사과정

관심분야 : 퍼지 이론, 패턴 인식, Support Vector Machines.

E-mail : bichoi@fuzzy.hanyang.ac.kr



이정훈(Frank Chung-Hoon Rhee)

1981년~1985년 : University of Southern California(USC), Dept. Electrical Eng., BSEE.
 1985년~1987년 : University of Missouri, Dept. of Electrical & Computer Eng., MSEE
 1990년~1993년 : University of Missouri,

Dept. of Electrical & Computer Eng., Ph.D.

1994년~1995년 : 한국전자통신연구원
 1995년~현재 : 한양대학교 전자컴퓨터공학부 부교수
 2002년~현재 : 제어자동화시스템공학회 편집위원
 2003년~현재 : 세계 퍼지 학회(IFSA) 한국 지부 대표

관심분야 : Computer Vision, Pattern Recognition, and all aspects of Computational Intelligence.

E-mail : frhee@fuzzy.hanyang.ac.kr