휴리스틱 진화에 기반한 효율적 클러스터링 알고리즘
(An Efficient Clustering Algorithm based on Heuristic Evolution)

류 정 우 † 강 명 구 †† 김 명 원 †††
(Joung Woo Ryu) (Myung Ku Kang) (Myung Won Kim)

요 약 클러스터링이란 한 군집에 포함된 데이타들 간의 유사한 성질을 갖도록 데이타들을 묶는 것으로 패턴인식, 영상처리 등의 공학 분야에 널리 적용되고 있을 뿐 아니라, 최근 많은 관심의 대상이 되고 있는 데이타 마이닝의 주요 기술로서 활발히 응용되고 있다.

클러스터링에 있어서 K-means나 FCM(Fuzzy C-means)와 같은 기존의 알고리즘들은 지역적 최적해에 수렴하는 것과 사전에 클러스터 개수를 미리 결정해야 하는 문제점을 가지고 있다. 본 논문에서는 진화 알고리즘을 사용하여 지역적 최적해에 수렴되는 문제점을 개선하였으며, 클러스터링의 특성을 분산도와 분리도로 정의하였다. 분산도는 임의의 클러스터의 중심으로부터 포함된 데이타들이 어느 정도 흩어져 있는지를 나타내는 척도인 반면, 분리도는 임의의 데이타와 모든 클러스터 중심간의 거리의 비율로서 얻어지는 소속정도를 고려하여 클러스터 중심간의 거리를 나타내는 척도이다. 이 두 척도를 이용하여 자동으로 적절한 클러스터 개수를 결정하게 하였다. 또한 진화알고리즘의 문제점인 탐색공간의 확대에 따른 수행시간의 증가는 휴리스틱 연산을 적용함으로써 크게 개선하였다. 제안한 알고리즘의 성능 및 타당성을 보이기 위해 이차원과 다차원 실험데이타를 사용하여 실험한 결과 제안한 알고리즘의 성능이 우수함을 나타내었다.

키워드 : 클러스터링, 진화알고리즘, 분산도, 분리도, 휴리스틱

Abstract Clustering is a useful technique for grouping data points such that points within a single group/cluster have similar characteristics. Many clustering algorithms have been developed and used in engineering applications including pattern recognition and image processing etc. Recently, it has drawn increasing attention as one of important techniques in data mining. However, clustering algorithms such as K-means and Fuzzy C-means suffer from difficulties. Those are the needs to determine the number of clusters apriori and the clustering results depending on the initial set of clusters which fails to gain desirable results.

In this paper, we propose a new clustering algorithm, which solves the above mentioned problems. In our method we use evolutionary algorithm to solve the local optima problem that clustering converges to an undesirable state starting with an inappropriate set of clusters. We also adopt a new measure that represents how well data are clustered. The measure is determined in terms of both intra-cluster dispersion and inter-cluster separability. Using the measure, in our method the number of clusters is automatically determined as the result of optimization process. And also, we combine heuristic that is problem-specific knowledge with a evolutionary algorithm to speed evolutionary algorithm search.

We have experimented our algorithm with several sets of multi-dimensional data and it has been shown that one algorithm outperforms the existing algorithms.

Key words : Clustering, evolutionary algorithm, intra-cluster dispersion, inter-cluster separability, heuristic

· 본 연구는 과학재단 특정기초연구사업(과제번호:98-0102-01-3) 및 한국과학기술원 뇌신경정보학연구사업(과제번호:M1010700004-1A220000900)의 지원에 의하여 수행되었습니다.

† 학생회원 : 숭실대학교대학원 컴퓨터학과
ryu0914@orgio.net
†† 비 회 원 : (주)씨씨미디어 기획팀 연구원
zycrome@chollian.net
††† 종신회원 : 숭실대학교 컴퓨터학부 교수
mkim@computing.soongsil.ac.kr

논문접수 : 2000년 10월 14일
심사완료 : 2001년 10월 16일

1. 서 론

관찰이나 실험 등을 통해 얻은 데이타들을 분류한다는 것은 과학적 연구의 가장 기본이 되는 목표중의 하나라고 볼 수 있다. 실제 d개의 변수로 구성된 N개의 개체들은 d-차원 공간에 흩어진 N개의 점으로 생각될 수 있으며, 이들이 어떤 의미의 유사성을 가지고 군집(cluster)을 이루고 있는지에 대한 정보는 다변량 자료의 구조를 이해하

는 데 매우 중요한 의미를 가지고 있다.

클러스터링이란 주어진 데이타를 군집화 하는 것으로, 한 군집 내에 있는 데이타들은 유사성(similarity)이 높은 반면 다른 군집에 속하는 데이타들과는 차별성(dissimilarity)이 높도록 데이타를 분류하는 것이다. 클러스터링은 특별한 정보나 배경지식 없이 데이타들 간의 주어진 척도를 이용하여 결과를 이끌어 내므로 비교사 학습(unsupervised learning)에 속하는 패턴 분류 방법으로써 현재 패턴인식, 영상처리 등의 공학분야에 널리 적용되고 있을 뿐 아니라, 최근 많은 관심의 대상이 되고 있는 데이타 마이닝 분야에서 핵심기술로 활발히 연구되고 있다.

클러스터링 알고리즘은 크게 분할적 클러스터링(partitional clustering)과 계층적 클러스터링(hierarchical clustering)으로 나눌 수 있다.

분할적 클러스터링은 임의의 데이타가 단지 하나의 클러스터에 포함되는 단순 클러스터링(hard clustering)과 두 개 이상의 클러스터에 동시에 속하는 것을 허용하는 퍼지 클러스터링(fuzzy clustering)으로 나뉘어 진다. 이와 같은 알고리즘들은 초기값에 따라 지역적 최적해로 수렴될 수 있는 문제점과 사전에 클러스터 개수를 결정해야 하는 문제점, 그리고 잡음에 민감한 문제점을 가지고 있다.

트리 구조로 표현되는 계층적 클러스터링은 초기에 입력 데이타 각각을 하나의 클러스터로 보고 유사성이 가장 큰 데이타부터 묶어 최종적으로 모든 데이타를 하나의 클러스터가 되도록 묶어 올라가는 상향식 트리를 형성하는 통합적 방법(agglomerative method)과 이와 반대 개념인 하향식으로 트리를 형성하는 분리적 방법(divisive method)으로 나뉘어 진다. 이들 방법들은 매 단계 유사한 데이타들만을 고려하므로 분할적 클러스터링처럼 사전에 클러스터 개수를 결정할 필요는 없으나, 어느 단계에서 알고리즘을 멈추게 할 것인지 임계값을 결정해야 하는 문제점을 가지고 있다[1].

본 논문에서는 앞서 설명한 분할적 클러스터링 알고리즘의 문제점을 개선한 알고리즘을 제안한다. 제안한 알고리즘에서는 병렬탐색으로 최적의 해를 찾는 진화알고리즘을 이용하여 전역적 최적해를 찾을 뿐만 아니라 클러스터링의 특성인 '클러스터내의 유사성과 클러스터간의 차별성'을 각각 분산도와 분리도로 나타내고, 입력 데이타들의 분포에 따라 자동으로 적절한 클러스터 개수를 결정한다. 또한 진화알고리즘이 가지고 있는 단점인 탐색공간의 확대에 따른 탐색시간의 증가를 본 논문에서는 휴리스틱 연산(heuristic operation)을 정의하여 개선한다.

본 논문의 구성은 다음과 같다. 먼저 2절에서는 기존의 알고리즘과 문제점들을 살펴보고, 3절에서는 진화알고리즘을 이용한 클러스터링 알고리즘을 제안한다. 4절에서는 제안한 알고리즘을 적용하여 인위적인 다차원 실험데이타에 대한 실험결과를 살펴보고 5절에서 결론을 맺는다.

2. 기존의 클러스터링 알고리즘과 문제점

분할적 클러스터링에 있어서 사전에 클러스터 개수를 결정하는 것은 매우 어려운 문제로서 클러스터링 알고리즘의 응용분야를 축소시키는 중요한 요인이 된다. 일반적으로 데이타의 차원이 3차원 이하일 경우에는 산점도등을 활용한 시각적 관측에 의해 군집관계를 파악하여 개수를 결정할 수 있으나, 차원이 증가하면 이러한 분석을 통해 클러스터의 개수를 결정하는 것이 어려우므로 클러스터 개수를 여러 번 바꾸어 클러스터링을 하여 어느 경우가 입력 데이타의 특성을 잘 표현하고 있는지 조사해야만 한다. 이와 같은 시행착오에 의한 결정은 많은 시간과 노력이 소요될 뿐만 아니라 분석의 정확성도 떨어진다.

특히 방대한 양의 다차원 데이타를 분석하기 위한 정보기술인 데이타 마이닝은 시간과 정확도가 중요한 요인이 된다. 따라서 클러스터링 알고리즘 역시 이러한 요인을 만족하기 위해서는 자동적으로 클러스터링이 이루어질 필요성이 대두되고 있다.

분할적 클러스터링 알고리즘 중에서 가장 보편적으로 많이 사용되는 K-means 알고리즘[2]과 Fuzzy C-Means(FCM)알고리즘[3]은 모든 데이타로부터 각각의 클러스터 중심까지의 거리의 제곱의 합으로 정의되는 목적함수를 최소화하는데 바탕을 둔 알고리즘이다.

이들 알고리즘의 목적함수는 단지 클러스터내의 유사성을 중심과 입력 데이타간의 거리로만 고려하고 있기 때문에 클러스터 개수를 제한하지 않는다면 각각의 데이타가 클러스터의 중심이 되는 경우 항상 최소 값이 된다. 이는 분할적 클러스터링 알고리즘이 사전에 클러스터 개수를 정해주어야 하는 필요성을 보여주고 있다. 또한 이들 알고리즘은 목적함수의 최소 값을 찾는 것이기 때문에 초기 클러스터의 중심 설정에 따라 알고리즘의 성능이 민감하게 좌우된다.

목적함수의 값이 지역적 최적해에 수렴하는 가능성을 줄이기 위해서 Isodata(Iterative Self-Organizing Data Analysis Techniques A) 알고리즘은 분할연산(split operation)과 결합연산(merge operation)을 바탕으로 사

용자와의 대화를 통하여 수행 중에 클러스터 개수가 변하는 것을 허용한다. 그러나 부가적으로 선험적 절차에 있어서 사용되는 많은 매개 변수, 예를 들면, 요구되는 클러스터 중심 개수, 분리 매개변수, 결합 매개변수 등에 의해 알고리즘의 성능이 민감하게 좌우될 뿐 아니라 차원수가 높은 경우 K-means나 FCM에서 사전에 클러스터 개수를 결정하는 것만큼 많은 어려움이 따른다.[2]

최근 이와 같은 지역적 최적해에 빠지는 문제점을 해결하기 위하여 목적함수를 최적화 시키는 방법으로 최적해를 찾아내는 진화알고리즘을 이용한 연구가 이루어지고 있으며[4][5][6], 또한 자동으로 클러스터 개수를 결정해 주기 위해서 통계적 기법이나 진화알고리즘을 적용하는 연구도 진행되고 있다[7][8].

3. 진화알고리즘을 이용한 클러스터링 알고리즘

클러스터링 문제는 입력데이타를 포함하는 입력 공간에서 유사한 데이타들끼리 그룹화 하는 문제로 생각할 수 있다. N개의 데이타를 c개의 클러스터로 그룹화 할 수 있는 경우의 수는 식(1)과 같다.[9]

$$\frac{1}{c!}\sum_{i=0}^{c}\binom{c}{i}(-1)^{c-i}\,i^{N} \qquad (1)$$

이 처럼 클러스터링 문제에 있어서 최적의 클러스터를 찾는 것은 NP-complete문제로 알려져 있으며 또한 어떤 클러스터링이 최적이냐에 대한 수학적 모델이 아직 없다.[10]

본 논문에서 제안한 알고리즘은 진화알고리즘을 적용한 클러스터링 알고리즘이다. 진화알고리즘은 1975년 존 홀랜드(John Holland)에 의해 제안된 전역적 탐색 기법이며, 이 기법은 자연현상의 자연도태와 진화의 메카니즘에 기반을 둔 확률적인 탐색 알고리즘으로서 특히 최적화 문제에 효율적인 알고리즘[11]이다.

따라서 제안한 알고리즘의 흐름은 (그림 1)과 같이 진화알고리즘의 흐름과 비슷하나 교배연산 대신 탐색시간을 줄이기 위해 즉, 진화속도를 향상시키기 위해 휴리스틱 연산을 사용한다는 점에서 차이가 있다.

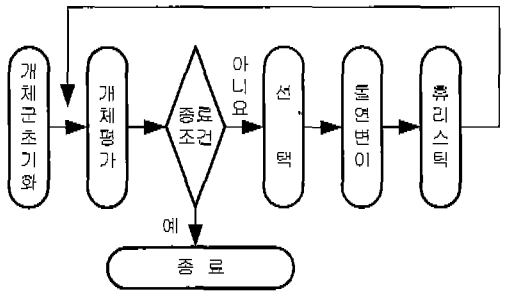


그림 1 진화 과정

3.1 개체군 초기화

진화알고리즘은 문제에 대한 개체(chromosome) 또는 후보해(candidate solution)들의 집단인 개체군(population)을 형성한다. 일반적으로 진화알고리즘에서의 각 개체들은 순서화된 고정길이이며, 한 유전자(gene)를 나타내는 값들의 배열로 표현된다. 제안한 알고리즘에서는 최적의 클러스터 개수와 그에 따른 클러스터 중심위치를 찾기 위해 (그림 2)와 같이 각 클러스터 중심의 집합으로 개체를 표현하고 있다. 따라서 각 개체는 실수로 표현된 가변적 길이를 가지도록 인코딩 한다.

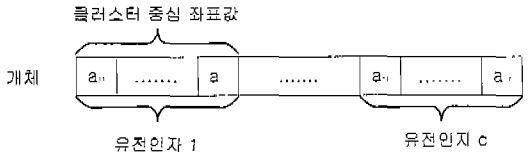


그림 2 개체 표현

(그림 2)는 d차원의 데이타공간에서 c개의 중심점(클러스터)들을 개체로 표현한 것이다.

3.2 개체평가

진화알고리즘에서는 각 개체의 성능을 평가하기 위해서는 적합도를 계산한다. 적합도란 임의의 개체가 문제의 해에 얼마나 적합한 지를 나타내는 척도이다. 이러한 적합도의 관점에서 해가 될 가능성이 있는 것들을 평가하는 환경 역할을 수행하는 것이 적합도 함수이다. 따라서 진화알고리즘의 성능은 적합도 함수에 좌우되므로 문제 해의 특성을 고려한 적절한 적합도 함수를 정의하는 것이 매우 중요하다.

클러스터링의 특성은 클러스터내의 모든 데이타들이 높은 유사성을 가져야 하며, 반면 다른 클러스터에 속하는 데이타들은 높은 차별성을 가져야 한다. 따라서 본 논문에서는 클러스터내의 유사성과 클러스터간의 차별성을 같이 고려한 적합도를 사용한다. 클러스터내의 유사성은 데이타들이 각 차원별로 클러스터 중심으로부터 얼마나 분산되었는가를 표준편차를 사용하여 정의한 분산도로서 나타낸다. 또한 클러스터간의 차별성은 각 클러스터내의 데이타들이 갖는 평균 소속정도의 합으로 정의한 분리도로서 나타낸다. 소속정도란 임의의 데이타가 클러스터에 포함될 가능성을 의미한다.

본 논문에서 d차원을 갖는 N개의 입력데이타 $X=\{x_1, x_2, .., x_N\}$, $x_j=(x_{j1}, x_{j2}, .., x_{jd})$에서 클러스

터링의 특성을 잘 나타내는 중심점 $V=\{v_1, v_2, \ldots, v_c\}$. $v_i=(v_{i1}, v_{i2}, \ldots, v_{id})$을 진화시킴으로써 자동으로 최적의 위치와 개수를 찾아주기 위해 다음과 같은 개체평가가 이루어진다.

모든 입력데이타에 대한 각 클러스터의 소속정도를 구하기 위해 임의의 데이타 x_j에 대해 각 클러스터의 소속정도 u_{ij}를 (식2)와 같이 각 클러스터 중심 v_i과의 거리의 비율로 나타낸다. 여기서 m, c는 각각 클러스터링의 퍼지정도와 클러스터 개수를 의미한다. 퍼지정도 m은 1보다 큰 값을 갖는다. 그 값이 크면 클수록 퍼지한 클러스터링이 생성된다. 즉, 한 데이타가 각 클러스터에 포함될 소속정도가 비슷하게 된다. 일반적으로 m=2이다. 또 $\|\cdot\|$은 내적노름(inner product-induced norm)을 나타낸다.

$$u(x_j, v_i)=u_{ij}=\left(\sum_{k=1}^{c}\left(\frac{\|x_j-v_i\|}{\|x_j-v_k\|}\right)^{\frac{2}{(1-m)}}\right)^{-1} \quad (2)$$

본 논문에서 $S_i \subseteq X$는 (식3)과 같이 S_i는 중심이 v_i인 클러스터에 대한 소속정도가 최대인 데이타(x_j)들의 집합으로 정의한다. 또 클러스터i (S_i)에 포함되어 있는 데이타 개수를 N_i로 표현한다.

$$S_i=\{x_j \mid i=\arg(\max_k[u(x_j, v_k)])\} \quad i=1,2,\ldots,c,\ j=1,2,\ldots,N \quad (3)$$

이와 같이 형성된 각각의 클러스터에 대해서 중심v_i을 평균으로 하고 표준편차 벡터를 계산한다. 이는 클러스터의 특성 중 클러스터내의 유사성에 의하여 클러스터 중심이 될 가능성이 높은 데이타의 주위에는 관련된 많은 데이타들이 분포될 가능성이 높다고 볼 수 있다. 따라서 본 논문에서는 중심 v_i 에 대해서 각 차원별로 표준편차를 계산하여 데이타의 분포를 측정한다. 계산된 모든 클러스터의 표준편차 벡터 요소를 합함으로써 클러스터 중심으로부터 데이타들이 얼마나 분산되어 있는가를 나타내는 분산도(dispersion) $disp(X, V)$을 식(4)와 같이 정의한다. 따라서 분산도가 작을수록 클러스터 중심 근방에 데이타들이 밀집되어 있어 클러스터내의 유사성이 높다는 것을 의미한다.

$$disp(X, V)=\sum_{i=1}^{c}\sum_{k=1}^{d}\left(\sqrt{\frac{1}{N_i}\sum_{x_j \in S_i}(x_{jk}-v_{ik})^2}\right) \quad (4)$$

적합도 함수로서 분산도만을 고려할 경우 최적의 해로는 항상 각각의 입력데이타가 클러스터 중심이 되는 경우이다. 즉 클러스터 개수는 입력데이타 개수가 된다. 이는 기존 분할적 클러스터링 알고리즘의 목적함수를 사용했을 때와 같은 결과를 갖게된다. 따라서 클러스터 개수를 자동으로 결정하기 위해서 클러스터내의 유사성뿐만 아니라, 클러스터간의 차별성을 동시에 고려해야만 한다. [7]에서는 클러스터간의 차별성을 모든 클러스터 중심간의 거리의 합으로 정의하였으나, 본 논문에서는 FCM에서 고려하고 있는 소속정도를 이용한다. 소속정도는 (식2)와 같이 모든 클러스터 중심과의 거리의 비율로 계산되어지기 때문에 가까운 클러스터 사이에 있는 데이타는 낮은 소속정도를 갖게 된다. 따라서 본 논문에서는 이러한 소속정도의 특성을 이용하여 클러스터간의 차별성을 나타낸다. 즉, 각각의 클러스터(S_i)에 대한 평균 소속정도를 합한 것을 분리도(separability) $sep(X, V)$으로 식(5)과 같이 정의한다. 분리도가 크면 클수록 클러스터간의 차별성이 높다는 것을 의미한다.

$$sep(X, V)=\sum_{i=1}^{c}\left(\frac{N_i}{N}\right)^{n}\left\{\frac{1}{N_i}\left(\sum_{x_j \in S_i}u_{ij}^{m}\right)\right\} \quad (\text{단}, 0 \le n \le 1) \quad (5)$$

식(5)에서 변수 n은 클러스터 크기를 제어하는 클러스터 크기 변수(cluster size parameter)라고 할 수 있다. 즉, 변수 n이 0이면 클러스터의 크기에 무관한 반면, n이 1에 가까워지면 클러스터 크기에 따라 가중치를 부여함으로써 클러스터 내에 적당하게 데이타들이 포함될 수 있도록 클러스터링을 하게 된다.

이와 같이 정의된 평가 척도를 사용하여 적합도가 작으면 작을수록 클러스터 특성을 잘 나타내는 개체로 평가되기 위해 식(6)과 같이 적합도 함수 $fit(X, V)$를 정의한다.

$$fit(X, V)=c^{1-l}\times\frac{disp(X, V)}{sep(X, V)} \quad (6)$$

여기서 $0 \le l \le 1$이며, 변수 l은 클러스터 개수를 제어하는 클러스터 개수 변수(number of clusters parameter)로 정의한다. 변수 l이 1에 가까운 값을 가지면 입력데이타에 대해 세밀하게 클러스터링을 하게 되며 반면, 0에 가까우면 대략적으로 클러스터링을 하게 된다.

지금까지 정의된 적합도 함수는 각 개체들의 특성을 평가하여 다음 세대의 개체집단을 선택하기 위한 척도가 된다. 본 논문에서는 다음 세대의 개체집단을 선택하기 위한 방법으로 룰렛휠(roulette wheel)방법과 엘리트 방법(elitist method)을 사용한다. 룰렛휠 방법은 개체의 적합도에 비례하여 개체가 선택될 기회를 주는 확률적 균등 표본 선택 방법으로 항상 가장 우수한 성질을 갖는 개체를 선택한다는 보장이 없다. 따라서 현세대의 개체집단 중에서 가장 우수한 성질을 갖는 개체를 선택하여 다음 세대에 계속 존속시킴으로써 최상의 적합도를 갖는 개체를 지속적으로 개선되게 하는 엘리트 방법을 동시에 사용한다[12].

3.3 진화 연산(Genetic Operation)

기존의 진화알고리즘을 이용한 클러스터링 알고리즘 [4][6][7][13]에서는 진화 연산으로 전통적인 교배연산과 돌연변이 연산을 사용하였다. 그러나 이 두 연산은 어떤 특별한 정보 없이 임의적으로 연산을 수행함으로 진화속도가 저하되는 단점이 있다. 본 논문에서는 이처럼 맹목적인 연산을 하는 교배연산 대신, 부모세대로부터 보다 가능성 있는 자식세대를 생성하기 위한 휴리스틱 연산을 정의하여 진화속도를 향상시킨다. 그러나 휴리스틱 연산은 잘못된 정보를 사용하게 되면 지역적 최적해에 빠지기 쉬운 단점을 가지고 있으므로 이를 보완해 주기 위해 기존의 돌연변이 연산을 그대로 사용한다.

3.3.1 돌연변이 (mutation)

진화알고리즘에서 돌연변이 연산은 한 개체의 유전자에서 임의로 선택된 유전자를 가능한 다른 값으로 바꿈으로써, 개체군의 다양성을 유지하며 탐색의 방향이 지역적 최적해로 향할 경우 여기에서 벗어날 수 있도록 한다.

제안한 알고리즘에서 휴리스틱 연산을 적용하기 때문에 진화된 클러스터 중심의 위치가 최적의 위치일 가능성이 높다. 하지만 보장은 할 수 없다. 따라서 가우시안 분포함수를 이용하여 현재 클러스터 중심으로부터 가까운 곳에서 새로운 중심이 선택될 확률을 높여 개체의 다양성을 유지한다.

돌연변이 연산을 적용하여 새로운 중심을 선택할 때 전체 입력공간을 균일하게 확률을 주어 선택하는 것보다 현재 클러스터 중심으로부터 가까운 지역이 먼 지역보다 선택될 확률을 높이게 하는 것이 진화속도를 향상시킬 수 있다. 하지만 지역적 최적해에 수렴될 가능성도 높아진다. 그러므로 제안한 알고리즘에서는 돌연변이 확률에 의해 선택된 t세대의 중심 $v_i(t)$가 $t+1$세대의 $v_i(t+1)$을 선택하기 위해 $v_i(t)$을 중심으로 하는 가우시안 분포함수를 사용하여 $v_i(t+1)$이 $v_i(t)$ 가까이 에서 선택될 확률이 보다 높도록 하였다. 이때 가우시안 함수의 폭을 결정하는 표준편차 값이 작으면 지역적 최적해에서 벗어날 가능성이 낮아지고, 크면 진화속도가 저하된다. 따라서 본 논문에서는 입력 데이타 범위의 70%로 주어 개체집단이 입력공간을 다양하게 탐색할 수 있도록 한다.

3.3.2 휴리스틱 연산(Heuristic Operation)

교배연산과 돌연변이 연산처럼 확률에 근거하여 모든 개체에 적용하는 맹목적인 연산이 가질 수 있는 진화시간 지연의 문제점을 휴리스틱 연산인 합병연산과 분할연산 그리고 K-means연산을 적용하여 진화속도를 개선한다. 특히 일반적 진화알고리즘에서 재조합 연산으로 알려진 교배연산은 최적의 클러스터 개수를 찾는 제안한 알고리즘에서는 비효율적이다. 왜냐하면 클러스터 개수에 따라 최적의 클러스터 위치가 다르기 때문에 클러스터 개수가 다른 즉, 길이가 다른 개체들을 교배하더라도 더 우수한 개체가 생성될 가능성이 낮기 때문이다.

본 논문에서 제안한 휴리스틱 연산은 가까이 있는 두 클러스터를 합병하고, 큰 클러스터는 두 개의 클러스터로 분할하는 개념에서 비롯된다. 합병 연산을 적용하기 위해서는 분리도를 고려한다. 즉, 클러스터 중심간의 거리가 임계값(θ_M) 보다 작을 경우 두 중심을 합병한다. 또한 분할 연산을 적용하기 위해서는 분산도를 고려한다. 즉, 각 클러스터의 표준편차 백터 요소가 임계값(θ_S)보다 클 경우 두 개의 클러스터로 분할한다. 이 두 연산을 통해 형성된 중심은 대략적인 중심의 위치를 정해줄 뿐 정확한 중심은 아니다. 따라서 K-means 알고리즘을 한 단계만 사용한 K-means연산[5]을 사용하여 위의 두 연산에 의해서 선택된 중심을 동일 세대에 정확한 중심으로 옮겨놓아 진화속도를 향상시킨다.

1) 합병 연산(merge operation)

두 클러스터 중심간의 거리가 임계값 θ_M보다 작으면 합병 연산을 적용하여 (그림 3. 왼쪽)과 같이 두 클러스터의 중심 v_1, v_2을 합하여 한 개의 클러스터로 만든다. 생성된 클러스터 중심은 식(7)과 같이 계산된다.

$$v = \frac{v_1 + v_2}{2} \tag{7}$$

여기서 임계값 θ_M는 평균 중심거리에 상수 α를 곱한 것으로서 식(8)과 같다.

$$\theta_M = \alpha\left[\frac{2}{c(c-1)}\sum_{i=1}^{c}\sum_{j=i+1}^{c} D_E(v_i, v_j)\right],\ 0 < \alpha \le 1 \tag{8}$$

단, $D_E(v_i, v_j) = \sqrt{\sum_{k=1}^{d}(v_{ik} - v_{jk})^2}$

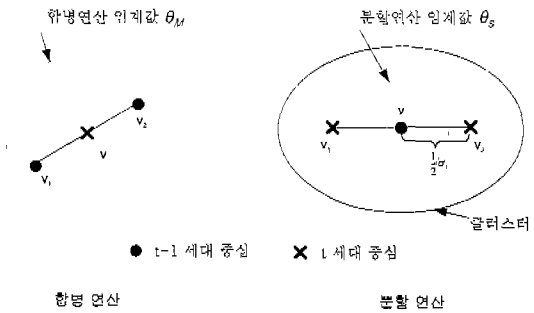


그림 3 합병 / 분할 연산

만약, 한 세대에 두 중심간의 거리가 임계값 보다 작은 것이 2개 이상일 경우에는 그 중 가장 가까운 중심 두 개만을 합병한다.

2) 분할 연산 (split operation)

클러스터의 표준편차 백터 요소들 중 임계값 θ_S보다 크면 분할연산을 적용하여 (그림 3. 오른쪽)과 같이 중심 v를 두 중심 v_1, v_2 로 분할한다. 분할된 중심좌표는 식(9)와 같이 표준편차가 가장 큰 차원만 고려한다.

$$\begin{cases} v_{1k} = v_k + \frac{1}{2}\sigma_k, v_{2k} = v_k - \frac{1}{2}\sigma_k, (\sigma_k = \max_i[\sigma_i]) \\ v_{1i} = v_{2i} = v_i (i \neq k) \end{cases} \quad (9)$$

이때 임계값 θ_S는 모든 클러스터의 표준편차 벡터 요소 합의 평균에 상수 β를 곱한 값으로 식(10)과 같다.

$$\theta_S = \beta \left[\frac{1}{c} \sum_{i=1}^{c} SD(v_i) \right], \quad 1 \leq \beta \quad (10)$$

단, $SD(v_i) = \frac{1}{d} \sum_{k=1}^{d} \left(\sqrt{\frac{1}{N_i} \sum_{x_j \in S_i} (x_{jk} - v_{ik})^2} \right)$

만약, 한 세대에 표준편차 벡터 요소가 임계값 보다 큰 것이 2개 이상일 경우에는 그 중 가장 큰 벡터 요소를 가지는 클러스터 하나만 분할한다.

3) K-means 연산 (K-means operation) [5]

돌연변이 연산, 합병 연산, 분할 연산에 의해 생성된 중심은 클러스터 내에서 정확한 중심이 아닌 대략적인 중심에 위치하고 있어 정확한 중심을 찾기 위해서 더 진화를 시켜야할 필요가 있다. 이러한 문제를 개선하기 위해서 본 논문은 [5]에서 제안한 K-means 알고리즘을 한 단계 적용한 연산을 사용한다. K-means 연산은 앞에서 설명한 연산들을 사용하여 생성된 중심들을 가지고 각 데이타를 가장 가까운 중심으로 재 할당하여 새로운 클러스터 중심을 계산한다. K-means 연산을 통해 매 세대마다 클러스터 중심들을 데이타 분포에 가장 적합한 중심으로 교정함으로써 진화 속도를 개선한다.

위에서 정의된 연산들이 한 세대에서 적용되는 순서는 표 1과 같다.

표 1 연산 수행 순서

```
돌연변이();
if ( 두 중심간의 거리 < θ_M) then
    합병연산(가장 가까운 두 중심);
if ( 클러스터 표준편차 백터 요소 > θ_S ) then
    분할연산(가장 큰 표준편차 벡터요소);
K-means연산();
```

4. 실 험

본 논문에서는 제안한 알고리즘의 타당성을 검증하기 위해 기존의 알고리즘 즉, K-means, FCM, Isodata과 비교 실험하였다. 제안한 알고리즘이 전역적 최적해에 수렴하면서 가장 적합한 클러스터 개수를 자동으로 찾는 것을 시각적으로 확인할 수 있도록 2차원 데이타를 이용하여 기존의 알고리즘과 비교하며 다차원 가우시안 분포 데이타에서는 클러스터링 후의 클러스터 분류율을 기존의 알고리즘과 비교하고, 생성된 클러스터 중심은 가우시안 분포 데이타의 원 중심간의 오차를 계산하여 타당한 지 확인한다. 또한 휴리스틱 연산을 사용하지 않고 교배 연산과 돌연변이 연산을 사용한 알고리즘[12]과 제안한 알고리즘과의 진화속도와 생성된 클러스터 개수 및 적합도를 비교한다.

본 실험에서는 실험 데이타 변수로 표2와 같이 선언하였다.

표 2 실험 데이타 변수

개체 집단크기	30	α	0.5
돌연변이확률	0.2	β	1.5

4.2 2차원 데이타

2차원 데이타, 데이타A[12]와 데이타B 그리고 데이타C를 가지고 제안한 알고리즘과 기존의 알고리즘을 비교하여 그 타당성을 보인다. K-means와 FCM은 초기 중심값에 따라 클러스터링 결과가 다르다. 따라서 각각의 데이타에 대해 임의적으로 초기 중심값을 바꿔가며 제안한 알고리즘은 10회, K-means와 FCM은 각각 20회씩 실험하였다.

데이타A에 대한 실험결과는 (그림 5)과 같다. 그림에서 표현한 중심점은 각 알고리즘을 수행했을 때 평균적으로 발생되는 두 가지 경우의 중심점들을 표현하고 있다. 첫 번째 경우의 중심점((○ : (1.5, 4.0), (7.0,4.0))은 K-means 경우 30% 생성되는 반면 제안한 알고리즘 경우 100% 생성된다. 두 번째 경우의 중심점(◇ : (3.4, 4.0), (7.5, 4.0))은 K-means, FCM의 첫 번째 경우를 제외한 평균 중심점을 나타내고 있다. 특히, FCM은 20회 모두 동일한 위치에 중심점이 생성됨을 알 수 있다. (그림 5)의 오른쪽 그래프는 세대에 따른 적합도 변화를 나타낸 것이다.

데이타B는 본 논문에서 제시한 실험 데이타로서 두 클러스터에 포함된 데이타의 수가 다를 뿐 아니라 타원형으로 형성되어 있는 데이타이다. 타원형으로 이루어진 군집들은 일반적인 거리의 척도인 유클리디언 거리로는 같은 군집으로 형성하기 힘들기 때문에 분산을 고려한 통계적인 거리를 사용한다. (그림 6)는 데이타B에 대한 실험결과로서 (○:(10.08, 12.18), (20.55, 16.98))는 제안

한 알고리즘에 의해 생성된 평균 중심점을 나타낸 반면 K-means는 ((12.96,14.62),(22.74,17.11))과 ((12.79,14.53), (22.60,17.10)) 두 가지 경우의 중심점들을 찾는다. 따라서 ◇은 생성된 평균서 중심점((12.84,14.55), (22.64,17.10)) 의 위치를 의미한다. FCM의 경우 20회 모두 중심점이 ((12.52,14.23),(22.83,17.09))에서 생성되었다.

기존의 알고리즘들은 클러스터링의 특징을 단지 클러스터내의 유사성을 일반적인 거리로만 고려하였으나 제안한 알고리즘에서는 표준편차 벡터를 사용하여 정의하고 동시에 다른 클러스터에 속하는 데이타들의 차별성을 거리의 비율로 고려하여 타원형에서도 기존 알고리즘보다 적합하게 클러스터링이 이루어지는 것을 보여주고 있다.

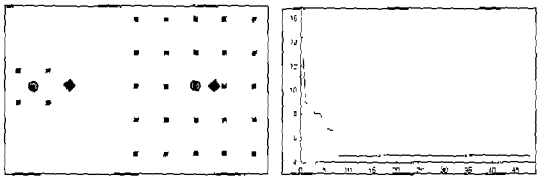


그림 5 데이타A에 대한 실험 (l=0.5, n=0.5)

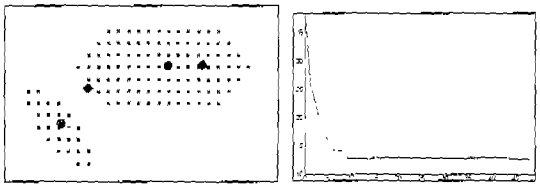
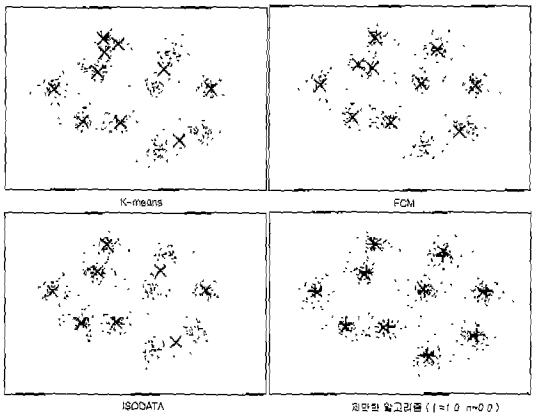


그림 6 데이타B에 대한 실험(l=0.5,n=0.5)



×: 클러스터 중심좌표
+: 가우시안 분포 데이타의 원 중심좌표

그림 7 데이타C에 대한 실험결과

데이타C는 입력공간에 대한 각 차원의 영역이 [0,10] 인 2차원 가우시안 분포 데이타로써 10개의 원 중심으로부터 가우시안 분포를 갖도록 각각 50개씩 데이타를 생성하였다. 따라서 총 500개의 데이타를 가지고 있다. 단, 데이타 생성시 원 중심은 클러스터의 평균 중심은 아니다.

실험결과는 (그림 7)과 같이 K-means, FCM은 각각 25%, 50% 경우만이 제안한 알고리즘 결과와 같았으며 제안한 알고리즘은 100% 똑같은 결과를 보이고 있다. (그림 7)에서 기존의 알고리즘의 결과는 20회 수행중 임의의 한 경우를 보여준 것이다.

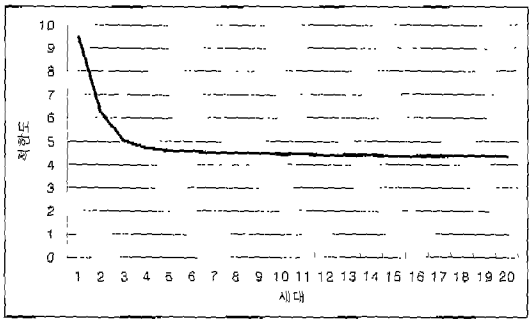


그림 8 세대에 따른 평균 적합도

(그림 8)에서는 세대에 따른 제안한 알고리즘의 평균 적합도 변화를 보여주고 있다. 평균적으로 20세대 내에서 적합도가 수렴되는 것을 알 수 있다. 또한 (표 3)에서는 생성된 클러스터 중심과 가우시안 분포의 원 중심간의 오차를 나타내고 있다. 오차는 데이타의 원 중심과

표 3 중심간의 오차

	가우시안 분포데이타의 원중심	클러스터 중심	오차 (%)
클러스터1	(2.09, 2.99)	(2.13, 2.94)	0.45
클러스터2	(4.46, 2.86)	(4.26, 2.78)	1.52
클러스터3	(6.50, 5.79)	(6.45, 5.84)	0.50
클러스터4	(0.37, 5.77)	(0.40, 5.72)	0.41
클러스터5	(7.49, 8.72)	(7.47, 8.77)	0.38
클러스터6	(3.22, 7.30)	(3.90, 7.36)	1.01
클러스터7	(9.85, 5.82)	(9.86, 5.69)	0.92
클러스터8	(9.40, 2.17)	(9.20, 2.08)	1.55
클러스터9	(6.64, 0.50)	(6.69, 0.37)	0.98
클러스터10	(3.63, 9.76)	(3.72, 9.72)	0.69
평균			0.84

가장 근접한 클러스터 중심간의 거리를 데이타 영역의 대각선 길이(데이타 공간의 최대거리)로 나눈 비율로 나타낸다. 따라서 생성된 클러스터들이 적절하게 데이타의 특성을 표현하고 있는 것을 알 수 있다.

4.2 다차원 데이타

다차원데이타는 데이타C를 생성한 것과 같은 방법으로 생성하였고 단지 차원만 10차원, 20차원으로 확장하여 생성시킨 데이타로서 입력공간에 대한 각 차원의 영역은 각각 [0,1], [0,10]으로 되어 있다. 또한 두 데이타 모두 500개 데이타를 가지고 있다.

지금까지 제안한 알고리즘이 진화알고리즘을 통해 자동적으로 클러스터 개수를 찾을 수 있을 뿐 아니라 최적의 위치까지 찾을 수 있다는 것을 기존 알고리즘과 비교실험을 통해 확인하였다.

본 실험에서는 진화알고리즘이 가지고 있는 단점인 탐색공간의 증가에 따른 진화수렴속도의 증가를 본 논문에서 제안한 휴리스틱 연산을 사용하여 해결하고 있음을 같이 보여주고 있다.

표 4 기존알고리즘과 제안한 알고리즘 비교

	10차원데이타	20차원데이타
	분류율 (%)	분류율 (%)
K-means	80	87
FCM	88	90
제안한 알고리즘	100	100

표 4는 다차원 데이타에서 제안한 알고리즘에 의해 생성된 클러스터가 적합한지 알아보기 위해 클러스터 후의 클러스터 분류율을 기존의 알고리즘과 비교한 결과를 보여주고 있다.

표 5은 10차원데이타에 대해 생성된 클러스터 중심과 가우시안 분포의 원 중심간의 오차를 나타내고 있다. 여기서 오차는 데이타의 원 중심과 가장 근접한 클러스터 중심간의 거리를 데이타 영역의 대각선 길이(데이타 공간의 최대거리)로 나눈 비율로 나타낸다.

20차원데이타에서 가우시안 분포데이타의 원 중심과 생성된 클러스터 중심의 평균 오차 거리는 0.46% 이다.

그림 9, 10은 각각 10차원과 20차원 데이타에 대한 적합도 변화에 따른 클러스터 개수의 변화를 보여주고 있다. 즉 제안한 알고리즘이 최적의 클러스터 개수를 찾으면서 차원에 관계없이 평균적으로 15세대이내에 수렴하는 것을 알 수 있다.

표 5 중심간의 오차

	가우시안 분포데이타의 원중심	클러스터 중심	오차 (%)
클러스터1	(0.25,0.83,0.34,0.50,0.06 0.61,0.97,0.25,0.58,0.66)	(0.25,0.81,0.35,0.50,0.06 0.61,0.95,0.26,0.57,0.66)	1.05
클러스터2	(0.87,0.54,0.36,0.79,0.20 0.18,0.16,0.50,0.03,0.97)	(0.87,0.52,0.35,0.80,0.20 0.16,0.15,0.50,0.02,0.97)	1.10
클러스터3	(0.02,0.76,0.25,0.50,0.56 0.62,0.19,0.30,0.71,0.06)	(0.02,0.76,0.24,0.51,0.55 0.63,0.19,0.31,0.72,0.08)	1.00
클러스터4	(0.73,0.58,0.50,0.53,0.76 0.55,0.08,0.24,0.64,0.06)	(0.72,0.60,0.51,0.52,0.76 0.57,0.08,0.24,0.63,0.07)	1.14
클러스터5	(0.51,0.24,0.04,0.94,0.94 0.79,0.41,0.42,0.12,0.26)	(0.49,0.25,0.02,0.95,0.94 0.78,0.41,0.41,0.12,0.25)	1.14
클러스터6	(0.41,0.32,0.55,0.61,0.80 0.80,0.59,0.76,0.45,0.85)	(0.41,0.31,0.55,0.62,0.79 0.80,0.58,0.77,0.45,0.84)	0.77
클러스터7	(0.49,0.33,0.98,0.66,0.84 0.93,0.41,0.28,0.68,0.99)	(0.47,0.34,0.99,0.65,0.85 0.92,0.42,0.28,0.67,1.00)	1.10
클러스터8	(0.96,0.09,0.46,0.75,0.04 0.09,0.48,0.33,0.58,0.31)	(0.97,0.09,0.47,0.73,0.04 0.08,0.48,0.32,0.60,0.31)	1.10
클러스터9	(0.67,0.12,0.11,0.68,0.20 0.89,0.61,0.76,0.73,0.20)	(0.66,0.12,0.11,0.67,0.20 0.90,0.61,0.75,0.73,0.20)	0.63
클러스터10	(0.38,0.48,0.35,0.95,0.70 0.65,0.17,0.17,0.99,0.97)	(0.38,0.48,0.37,0.95,0.71 0.64,0.17,0.16,0.98,0.97)	0.89
평균			0.99

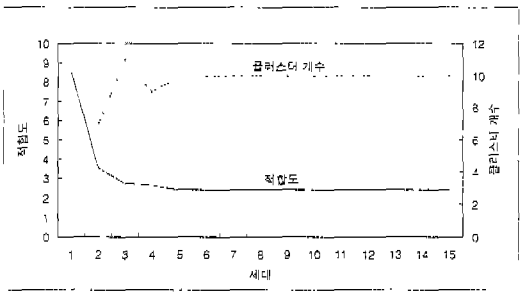


그림 9 적합도와 클러스터 개수(10차원)

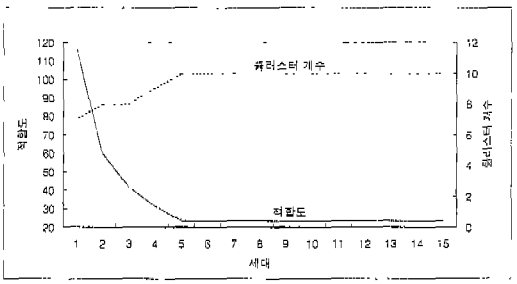


그림 10 적합도와 클러스터 개수(20차원)

또한 그림11, 그림12 는 교배연산과 돌연변이 연산을 적용했을 때와 교배연산 대신 휴리스틱 연산과 돌연변이 연산을 적용했을 때 각각의 성능을 나타내고 있으며 교배연산 대신 휴리스틱 연산을 적용한 경우가 더 빨리 진화하는 것을 알 수 있다.

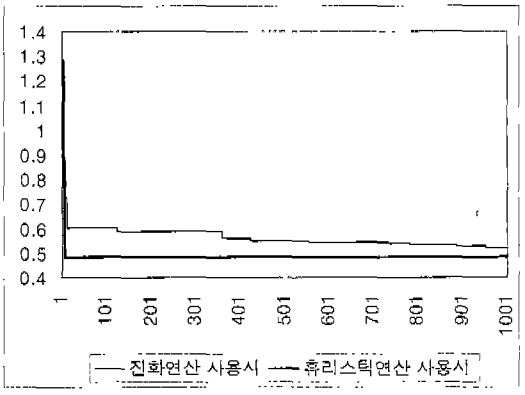


그림 11 1000세대 진화된 적합도값의 변화

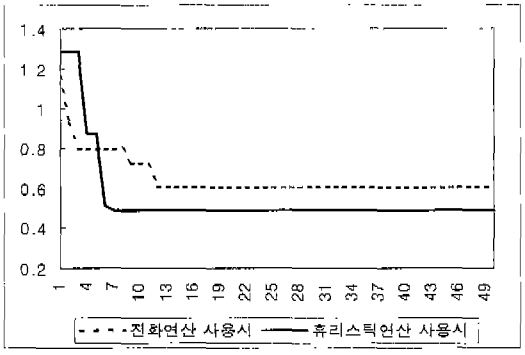


그림 12 20세대 진화된 적합도값의 변화

클러스터링에서는 클러스터간의 상관성이 강하다. 즉, 일부의 클러스터 중심위치 또는 클러스터 개수가 바뀌면 다른 클러스터도 바뀌게 된다. 따라서 부모세대로부터 가능성이 높은 두 개체들을 선택하여 교배연산을 적용하게되면 정보가 교환되어 일부의 클러스터 중심위치 또는 클러스터 개수가 바뀌게 된다. 이는 교배연산을 적용하기 전보다 더 우수한 개체가 생성될 가능성을 낮게 한다.

그러므로 제안한 알고리즘은 선택방법에 의해 선택된 개체들 사이에서 맹목적으로 선택된 두 개체의 정보를 교환하는 교배연산 대신 개체 각각에 대해 클러스터내의 유사성(분산도)과 클러스터간의 차별성(분리도)을 고려하여 휴리스틱 연산을 적용하는 것이 보다 효율적이라고 할 수 있다.

그렇다면 휴리스틱 연산만 적용하여 가장 좋은 결과를 취한 경우와 제안한 알고리즘과의 결과와 어떤 차이가 있는지 알아보기 위해 아래와 같은 실험을 통해 살펴본다.

초기해를 임의로 생성한 크기가 30인 개체집단을 20세대 수행하면 제안한 알고리즘은 최적의 클러스터 개수와 중심의 위치를 찾는다는 것을 앞에서 보았다. 본 실험은 제안한 알고리즘의 수행 시간만큼 휴리스틱 연산만을 이용하여 반복 수행한 다음 결과를 비교한다.

휴리스틱 연산만을 사용하는 경우 적합도 평가 단계와 개체 선택 단계가 제외됨으로 진화알고리즘을 사용하는 경우보다 약 2배가 빠르다는 것을 실험을 통해 알 수 있었다. 그러므로 실험에서는 비교의 공정성을 위하여 크기가 제안한 알고리즘의 2배인 60인 개체집단을 휴리스틱 연산만을 사용하여 20회 반복 수행한다. 이것을 초기 클러스터 중심과 클러스터 개수를 각각 다르게 하여 10회 실시한 결과, 수렴된 평균 적합도는 표6과 같다. 여기서 ( )는 10회 실험 중 적합한 클러스터 개수와 중심을 찾은 횟수를 나타낸다. 따라서 제안한 알고리즘은 10회 모두 옳게 찾았으나 휴리스틱 연산만을 사용한 경우에는 다차원일수록 옳게 찾는 횟수가 적고 제안한 알고리즘과의 적합도 오차도 커지는 것을 알 수 있다. 추가로 크기가 120인 개체집단을 실험한 결과 역시 제안한 알고리즘 보다는 못하지만, 개체집단이 60인 경우보다는 좋은 결과를 보이고 있다.

표 6 휴리스틱 연산만 사용한 경우와 제안한 알고리즘의 성능 비교

방법 / 개체크기 / 데이타	제안한 알고리즘	휴리스틱 연산만 사용한 경우	
	30	60	120
2차원 데이타 (데이타 C)	4.3676 (10)	4.5043 (8)	4.4763 (8)
10차원 데이타	2.4364 (10)	3.1607 (3)	3.0449 (5)
20차원 데이타	23.5227 (10)	33.5304 (3)	30.3321 (5)

따라서, 휴리스틱 연산만을 적용하여 최적의 해를 찾는다는 것을 보장할 수 없으며, 또한 있다 하더라도 제안한 알고리즘 보다 계산시간이 더 오래 걸린다. 더욱이

탐색공간이 커질수록 그 가능성은 더욱 적어진다. 반면, 제안한 알고리즘은 진화 알고리즘을 적용하여 매 세대, 전 세대에서 우수한 성질의 개체를 선택하여 연산을 수행하기 때문에 최적의 해를 찾을 가능성이 보다 높다는 것을 알 수 있다.

5. 결 론

본 논문에서는 진화 알고리즘을 이용하여 자동으로 클러스터 개수를 결정하는 클러스터링 알고리즘을 제안했다. 제안한 알고리즘에서 사용한 적합도 함수는 표준편차 벡터를 계산하여 중심으로부터 포함된 데이타가 얼마나 떨어져 있는지 알 수 있는 분산도(dispersion)와 임의의 데이타와 모든 중심간의 거리의 비율로서 얻어진 소속정도를 고려한 분리도(separation)로 정의하고 있다.

이와 같이 정의된 두 가지의 평가척도를 고려하여 제안한 알고리즘은 가우시안 분포들로부터 인위적으로 생성된 데이타를 사용하여 기존 알고리즘(K-means, FCM, Isodata)에 있어 초기 중심에 따라 지역적 최적해로 수렴될 수 있는 문제를 해결할 뿐 아니라, 클러스터 개수를 자동으로 결정함을 보여주고 있다. 또한 진화알고리즘의 단점인 탐색공간이 커짐에 따라 진화 속도가 기하급수적으로 느려지는 문제점을 휴리스틱 연산인 합병 연산, 분할 연산, K-means연산을 적용하여 개선하고있다.

향후 계획으로는 기존의 분할적 클러스터링 알고리즘에서 다루기 힘들었던 기호적이나 이산적인 값을 포함한 데이타를 적용하는 부분을 연구할 계획이다. 기존의 분할적 클러스터링 방법들은 기호적인 데이타를 수치 데이타로 변환하여 사용하였으며, 그 과정에서 데이타의 성질이 왜곡되어 적절한 클러스터링이 되지 않는 단점을 가지고 있다. 이러한 단점 역시 클러스터링의 응용분야를 축소시키는 요인이 될 수 있다. 예를 들면 데이타 마이닝, 전자상거래에서 생성되는 데이타들은 일반적으로 수치 데이타와 기호적 데이타가 혼합되어 있는 경우가 많다. 이러한 데이타들을 올바르게 분석하기 위해서는 제안한 알고리즘에 기호적 데이타를 분석할 수 있는 척도를 고려하여 보다 일반적인 클러스터링 알고리즘으로 확장되어야 할 것이다.

참 고 문 헌

[1] Brian D Everitt, "Cluster analysis," third edition, John Wiley & Sons, Inc. 1993.

[2] J. T. Tou, R. C. Gonzalez, "Pattern Recognition Principles," Addison-Wesley Publishing Company, Inc. pp. 75-109, 1974.

[3] George J. Klir, Bo Yuan, "Fuzzy Sets and Fuzzy Logic," Prentice-Hall Inc. 1995.

[4] G.Phanendra Babu and M. Narasimha Murty, "Clustering with Evolution Strategies," Pattern Recognition VOL 27, No. 2, pp. 321-329, 1994.

[5] K.Krishna and M. Narasimha Murty, "Genetic K-Means Algorithm," IEEE Trans. Syst., Man. Cybern., VOL. 29, No. 3, pp. 433-439, 1999.

[6] Susu Yao, "Evolutionary Search Based Fuzzy Self-Organising Clustering," Congress on Evolutionary Computation (CEC '99), pp. 185-188. 1999.

[7] 정창호, 임영희, 박주영, 박대희, "진화프로그램을 이용한 퍼지 클러스터링", 정보과학회 논문지(B) 제26권, 제1호, pp. 130-138, 1999.

[8] Hichem Frigui, Raghu Krishnapuram, "A Robust Competitive Clustering Algorithm With Application in Computer Vision," IEEE Tans. on Pattern Analysis And Machine Intelligence, VOL 21, NO 5, 1999.

[9] Knuth, D. The art of computer programming, vol. 1. Fundamental Algorithms of Addison-Wesley Series in Computer Science and Information Processing. Addison-Wesley, Reading, MA, 1973.

[10] YoungJa Park and ManSuk Song, "A Genetic Algorithm for Clustering problems," In Symposium on Genetic Algorithm-98 pp.568-575, July, 1998.

[11] Z. Michalewicz, "Genetic Algorithm + Data Structures=Evolution Programs," Third, Extended Edition, Springer-Verlag, 1995.

[12] Hsiao-Fan Wang, Chen Wang, Guang-Yaw Wu, "Bi-criteria fuzzy c-means analysis," Fuzzy Sets and Systems 64, pp. 311-319, 1994.

[13] 김명원, 류정우, "진화 알고리즘을 이용한 클러스터링 알고리즘", 2000봄 학술발표논문집(B) 제27권 1호 pp. 313-315, 2000.

김 명 원

1972년 서울대학교 응용수학과 졸업. 1981년 University of Massachusetts (Amherst) Computer Science 석사 학위 취득. 1986년 University of Texas (Austin) Computer Science 박사 학위 취득. 1975년 ~ 1978년 한국과학 기술연구소 연구원. 1982년 ~ 1985년 institute for Computing Science & Computer Application (Univ. of Texas) 연구원. 1985년 ~ 1987년 AT&T Bell Labs. (Naperville) Member of Technical Staff. 1987년 ~ 1994년 한국전자통신연구소 책임연구원. 1991년 ~ 1993년 충남대학교 전자계산학과 겸임부교수. 2000년 9월 ~ 2001년 3월 미국 IBM T.J WATSON 연구소 방문과학자. 1994년 ~ 현재 숭실대학교 컴퓨터학부 부교수. 1992년 ~ 1993년 한국신경회로망 연구회 회장. 1992년 ~ 1993년 한국정보과학회 뉴로 컴퓨팅 연구회 회장. 1993년 ~ 1995년 정보과학회 뉴로 컴퓨팅 연구회 위원장. 1998년 ~ 2000년 한국인지학회 부회장. 1997년 ~ 2000년 한국뇌학회 부회장. 1993년 ~ 현재 IEEE Neural Network Council 한국지부장. 2001년 ~ 현재 한국뇌학회 회장. 관심분야는 유연추론, 신경회로망, 퍼지시스템, 진화알고리즘, 패턴인식, 자동추론 기계학습, 데이터마이닝, creativity engineering 등

류 정 우

1998년 2월 숭실대학교 정보과학대학 인공지능학과 졸업(학사). 2000년 2월 숭실대학교대학원 컴퓨터학과 졸업(석사). 2000년 2월 ~ 현재 숭실대학교대학원 컴퓨터학과 박사과정. 관심분야는 신경망, 유전자알고리즘, 퍼지이론, 데이터마이닝, 에이전트

강 명 구

1994년 2월 경희대학교 문리과대학 수학과졸업(학사). 1995년 ~ 1998년 연변과학기술대학 강사. 2001년 2월 숭실대학교대학원 컴퓨터학과 졸업(석사). 2001년 ~ 현재 (주)씨씨미디어 기획팀. 관심분야는 데이터마이닝, 웹마이닝, 웹로그분석, OLAP