진화알고리즘을 이용한 클러스터링 알고리즘

류정우° , 김명원
숭실대학교 컴퓨터학과
ryu0914@channeli.net, mkim@computing.soongsil.ac.kr

A Clustering Algorithm using the Genetic Algorithm

Joungwoo Ryu, Myungwon Kim
School of Computing, Soongsil Univ.

요 약

클러스터링에 있어서 K-means와 FCM(Fuzzy C-means)와 같은 기존의 알고리즘들은 지역적 최소 해에 수렴될 문제와 사전에 클러스터 개수를 결정해야 하는 문제점을 가지고 있다. 본 논문에서는 병렬 탐색을 통해 최적 해를 찾는 진화 알고리즘을 사용하여 지역적 최소 해에 수렴되는 문제점을 개선하였으며, 클러스터의 특성을 표준편차 벡터를 계산하여 중심으로부터 포함된 데이터가 얼마나 분포되어 있는지 알 수 있는 분산도와 임의의 데이터와 모든 중심들 간의 거리의 비율로서 얻어지는 소속성도를 고려하여 클러스터간의 간격을 알 수 있는 분리도를 정의함으로써 자동으로 클러스터 개수를 결정할 수 있게 하였다. 실험데이터와 가우시안 분포에 의해 생성된 다차원 실험데이터를 사용하여 제안한 알고리즘이 이러한 문제점들을 해결하고 있음을 보인다.

1. 서론

클러스터링(clustering)이란 주어진 데이터를 그룹화 하는 것으로, 같은 그룹에 있는 데이터들은 유사성(similarity)이 높은 반면 다른 그룹에 속하는 데이터들과는 비유사성(dissimilarity)이 높도록 상호간의 관계를 정립하는 것이다. 클러스터링은 특별한 정보나 배경지식을 필요로 하지 않고 데이터들 간의 주어진 척도를 이용하여 결과를 이끌어 내므로 비교사 학습(unsupervised learning)에 속하는 패턴 분류 방법으로써 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 지역적 최소해로 수렴될 수 있는 문제점과 사전에 클러스터 개수를 결정해야하는 문제점, 그리고 잡음에 민감한 문제점을 가지고 있다.

트리 구조(dendrogram)로 표현되는 계층적 클러스터링은 입력 데이타의 개수를 하나의 클러스터로 보고 유사성이 가장 큰 데이터부터 묶어 최종적으로 모든 데이터를 하나의 클러스터가 되도록 묶어 올라가는 상향식(bottom-up)으로 트리를 형성하는 집괴적 방법(agglomerative method)과 이와 반대 개념인 하향식(top-down)으로 트리를 형성하는 구분적 방법(divisive method)으로 나뉘어 진다. 이들 방법들은 매 단계 유사한 패턴들만을 고려하므로 분할적 클러스터링처럼 사전에 클러스터 개수를 결정할 필요는 없으나, 어느 단계에서 알고리즘을 멈추게 하는 임계값(threshold)을 정의해야 하는 문제점을 가지고 있다[1].

본 논문에서는 앞서 설명한 분할적 클러스터링 알고리즘의 문제점을 개선한 알고리즘을 제안한다. 이 알고리즘에서는 병렬탐색을 통해 최적의 해를 찾는 진화알고리즘(genetic algorithm)을 이용하여 전역적 최적의 해를 찾을 뿐만 아니라 클러스터가 가져야 할 특성인 같은 클러스터내의 유사성과 클러스터간의 비유사성을 각각 분산도와 분리도로 나타내는 함수를 정의하여 입력데이터들의 분포에 따라 자동으로 클러스터 개수를 결정하도록 하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 기존의 알고리즘과 진화알고리즘을 이용하여 문제점들을 해결하기 위한 최근 연구들을 살펴보고, 3장에서는 진화알고리즘을 이용한 클러스터링 알고리즘을 제안한다. 4장에서는 제안한 알고리즘을 적용하여 실험데이터에 대한 실험결과를 살펴보고 5장에서 결론을 내린다.

2. 관련연구

분할적 클러스터링 알고리즘 중에서 응용분야에 가장 보편적으로 많이 사용되는 K-means알고리즘[2]과 Fuzzy C-Means(FCM)알고리즘[3]은 모든 데이터로부터 각각의 클러스터 중심까지의 거리의 제곱의 합으로 정의되는 목적함수(object function)를 최소화하는데 바탕을 둔 알고리즘이다.

이들 알고리즘의 목적함수는 단지 같은 클러스터내의 유사성을 중심과 입력 데이터간의 거리로만 고려하고 있기 때문에 유사성이 가장 높은 경우는 각각의 데이터가 클러스터의 중심이 되는 경우임으로 목적함수의 값은 클러스터 개수와 입력데이터의 개수가 같은 경우에 항상 최소의 값을 갖는다. 따라서 사전에 클러스터 개수는 결정해야만 한다.

이러한 문제는 입력 공간이 다차원 공간일 때 몇 개의 클러스터로 이루어져 있는지 결정하기 힘들기 때문에 클러스터 개수를 여러 번 바꾸어 어느 것이 주어진 데이터가 갖는 특성을 잘 표현하고 있는지 조사해야만 한다. 또한 이들 알고리즘은 목적함수의 최소값을 찾는 것이기 때문에 초기 설정 값에 따라 알고리즘의 성능이 민감하게 좌우된다.

목적함수의 최소값으로 인한 지역적 최소의 해에 수렴하는 가능성을 줄이기 위해서 Isodata (Iterative Self-Organizing Data Analysis Techniques A) 알고리즘은 분리연산(split operation)과 결합연산(merge operation)같은 부가적인 선험적 절차들을 대화 기법으로 통합하여 수행중에 클러스터 개수가 변하는 것을 허용한 알고리즘이다[4].

그러나 부가적 선험적 절차에 있어서 사용되는 많은 매개 변수, 예를 들면, 요구되는 클러스터 중심 개수, 분리 매개변수, 결합 매개변수 등에 의해 알고리즘의 성능이 민감하게 좌우될 뿐만 아니라 결정하기에도 많은 어려움이 따른다.[2]

최근 이와 같은 반복적 수행을 통한 목적함수의 최적화 방식을 찾아내는 것을 기본 원리로 갖는 알고리즘에 있어서 지역적 최적의 해에 수렴될 수 있는 문제점을 해결하기 위해 진화알고리즘을 이용하는 연구가 이루어지고 있으며[5][6], 또한 자동으로 클러스터 개수를 결정해주기 위해서 통계적 기법이나 진화알고리즘을 적용하는 연구가 진행되고 있다[7].

3. 진화알고리즘을 이용한 클러스터링 알고리즘

클러스터링 문제는 입력공간(feature space)에서 패턴들을 분할하는 문제로 생각할 수 있다. N개의 데이터를 C클러스터로 분할 할 수 있는 경우의 수는 식(1)과 같다.[8]

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

이처럼 클러스터링 문제에 있어서 최적의 클러스터를 찾는 다는 것은 NP-complete문제이므로 본 논문에서 제안한 알고리즘은 1975년 존 홀랜드(John Holland)에 의해 제안된 전역적 탐색 기법으로 자연현상의 자연도태와 진화의 메카니즘(mechanism)에 기반을 둔 확률적인 탐색 알고리즘으로서 특히 최적화 문제에 효율적인 진화알고리즘[9]을 사용하였다.

따라서 제안한 알고리즘의 흐름은 (그림1)과 같이 진화알고리즘의 흐름과 같으며, 각 단계를 어떻게 정의하였는지 살펴보겠다.

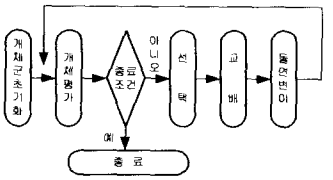

그림 1. 진화과정

(1) 개체군 초기화

진화알고리즘은 문제에 대한 후보해(candidate solution) 또는 개체(chromosome)들의 집단인 개체군(population)을 유지한다. 일반적으로 진화알고리즘에서의 각 후보해들은 순서화된 고정길이이며, 한 유전인자(gene)를 나타내는 값들의 배열로 표현한다.

최적의 클러스터 개수를 찾아주기 위해 제안한 알고리즘에서 개체는 유전인자 값에 해당하는 클러스터 중심좌표의 집합을 나타내며 (Φ_1, Φ_2 ,.....Φ_{pop})로 표현한다. 여기서 Φ_i는 i번째 유전인자가 가지고 있는 중심좌표 (v_{i1}, v_{i2}, ... ,v_{di})를 나타내다. 각각의 개체 길이는 가변적이며 개체 초기화는 입력데이터들과 입력공간에서 균등한 비율로 임의적으로 선택하여 초기 개체군을 구성한다.

(2) 개체평가

각 개체의 성능을 평가하기 위해서는 적합도를 계산한다. 적합도란 임의의 개체가 문제의 해(solution)에 얼마나 적합한지를 나타내는 척도이다. 이러한 적합도의 관점에서 해가 될 가능성이 있는 것들을 평가하는 환경의 역할을 수행하는 것이 적합도 함수(fitness function)이다. 따라서 진화알고리즘의 성능은 적합도 함수에 좌우되므로 문제 해의 특성을 고려한 적합도 함수를 정의해야만 한다.

클러스터의 특성은 클러스터내의 모든 데이터들이 높은 유사성을 가져야 하며, 반면 다른 클러스터에 속하는 데이터들은 높은 비유사성을 가져야한다.

제안한 알고리즘에서는 각각의 클러스터에 대해서 중심을 평균(mean)으로 하고 식(2)에 의하여 클러스터에 포함되는 데이터에 대해서 표준편차 벡터를 계산한다. 모든 클러스터의 표준편차 벡터 요소를 합함으로서 클러스터 중심으로부터 데이터들이 얼마나 떨어져 있는 가를 나타내는 분산도(dispersion) $disp(X,V)$을 식(3)와 같이 정의하였으며, 분산도의 값이 작을수록 중심과 데이터간의 유사성이 높다는 것을 의미한다.

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

$$disp(X,V) = \sum_{i=1}^{c}\sum_{k=1}^{d}\left(\sqrt{\frac{1}{N_i}\sum_{x_j \in S_i}(x_{kj} - v_{ki})^2}\right) \quad (3)$$

여기서 $X=\{x_1, x_2, \ldots, x_n\}$, $x_i=(x_{1j}, x_{2j}, \ldots, x_{dj})$는 입력데이터들의 집합이고 $V=\{v_1, v_2, \ldots, v_c\}$, $v_i=(v_{1i}, v_{2i}, \ldots, v_{di})$는 중심점들의 집합이다. N_i은 클러스터 i에 포함되어 있는 데이터의 개수이며, S_i는 클러스터i의 집합을 의미한다.

각각의 클러스터에 포함된 데이터에 대한 평균 소속정도(membership degree)를 합한 것을 분리도(separation) $sep(X,V)$으로 식(4)과 같이 정의하였으며, 분리도의 값이 크면 클수록 다른 클러스터간의 데이터들의 비유사성이 높다는 것을 의미한다.

$$sep(X,V) = \sum_{i=1}^{c}\left(\frac{N_i}{N}\right)^{n}\frac{1}{N_i}\left(\sum_{x_j \in S_i} u_{ij}^{m}\right) \quad (단, 0 \le n \le 1) \quad (4)$$

식(4)에서 변수 n을 클러스터 크기를 제어하는 클러스터 크기 변수(Cluster Size Parameter)로 정의한다. 즉, (그림2)과 같이 변수 n이 1에 가까우면 클러스터에 포함되는 데이터의 개수에 따라 평균 소속정도에 가중치(weight)를 부여함으로서 어느 정도의 데이터 개수를 가지는 것만을 클러스터로 정의하였다.

이와 같이 정의된 평가 척도를 사용하여 적합도가 작으면 작을수록 클러스터 특성을 잘 나타내는 개체로 평가되기 위해서 식(5)와 같이 적합도 함수 $fit(X,V)$를 정의하였다.

$$fit(X,V) = C^{1-l} \times \frac{disp(X,V)}{sep(X,V)} \quad (5)$$

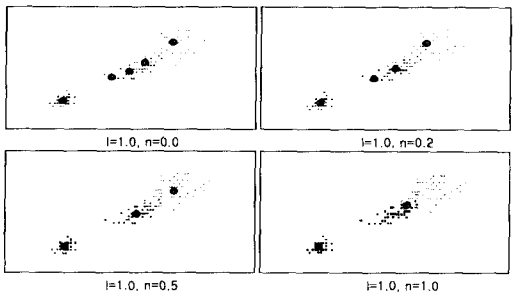

그림 2. n값에 따른 클러스터 크기 변화

여기서 $0 \le l \le 1$이며, 변수 l을 클러스터 개수를 제어하는 클러스터 개수 변수(Number of Cluster Parameter)로 정의하였다. (그림3)와 같이 변수 l이 1에 가까운 값을 가지면 입력패턴에 대해 세부적(detail)으로 클러스터링을 하게 되며 반면, 0에 가까우면 대략적(rough)으로 클러스터링을 하게 된다.

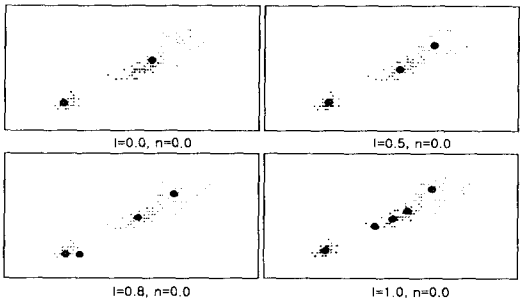

그림 3. l값에 따른 클러스터 개수의 변화

(3) 선택방법

본 논문에서는 다음 세대의 개체집단을 선택하기 위한 방법으로 룰렛휠(roulette wheel)방법과 엘리트 방법(elitist model)을 사용한다. 엘리트 방법은 현재세대의 개체집단 중에서 가장 우수한 성질을 갖는 개체를 선택하여 다음 세대에 계속 존속시킴으로써 최상의 적합도를 갖는 염색체를 지속적으로 개선되게 하는 방법이며, 룰렛휠 방법은 우수한 성질의 개체에 보다 많은 선택의 기회를 주는 방법이다.

(4) 교배연산

본 논문에서는 개체 구조의 특성상 가변길이이며 유전인자들이 위치에 상관없기 때문에 기존의 순서화된 고정길이 개체에서 사용했던 교배연산자를 사용한다는 것은 진화성을 저하하는 요인이 될 수 있다. 따라서 (그림4)와 같은 집합교배 연산자(setcrossover operation)를 정의하였다.

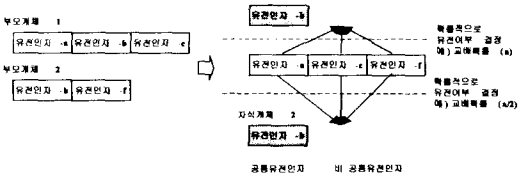

그림 4. 집합교배연산자

집합교배 연산자는 부모개체에서 공통으로 가지고 있는 유전인자는 자식세대에 그대로 전해지고, 서로 다른 유전인자는 생성된 자식개체에 확률적으로 유전여부를 결정함으로서 생성될 수 있는 자식개체 길이(l)의 범위는 식(6)과 같다.

$$c \le l \le (c+n) \quad (6)$$

여기서 c는 공통된 유전인자의 개수이고, n은 비 공통된 유전인자의 개수를 의미한다.

(5) 돌연변이연산

돌연변이 연산은 각 개체에 대하여 임의로 선택된 유전인자를 새로운 성질의 유전인자로 생성시키는 연산자이다.

본 논문에서 정규적 돌연변이 연산자와 가우시안 랜덤 변수(gaussian random variable)를 사용하였다.

- 정규적 돌연변이 연산자

정규적 돌연변이 연산자란 돌연변이 연산을 적용할 때 입력공간상에서 임의로 패턴을 선택하는 경우와 입력패턴들 중에서 임의로 선택하는 경우가 존재하며 그 중 한가지 경우를 임의로 선택하여 새로운 유전인자를 생성시켜 돌연변이 확률에 의해 선택된 유전인자와 바꾸는 연산자이다.

- 가우시안 랜덤 변수

가우시안 랜덤 변수란 가우시안 함수를 사용하여 전혀 다른 유전인자의 성질을 갖도록 하는 정규적 돌연변이 연산과는 다르게 돌연변이 확률에 의해 선택된 특정 유전인자에 비슷한 성질을 갖는 새로운 유전인자가 선택될 확률을 높이는 연산자이다.

4. 실험

제안한 알고리즘의 타당성을 검증하기 위해 두 가지 경우의 실험을 하였다. 첫 번째 실험은 제안한 알고리즘이 전역적 최소해를 찾을 수 있으면서 가장 적합한 클러스터 개수를 자동으로 찾을 수 있는지를 시각적으로 확인할 수 있도록 이차원 실험데이터를 가지고 실험하였으며, 두 번째 실험은 다차원 실험데이터를 사용하였다.

본 실험에서는 (표1)과 같이 매개변수를 사용하여 실험하였다.

표 1. 실험 데이터 변수

진화회수	2,000	교배확률변수	0.5
개체집단 크기	40	돌연변이확률변수	0.2

(1) 이차원데이터에 대한 실험

데이터A[7][10]는 두 클러스터간의 간격이 좁으면서 포함하고 있는 데이터 수가 다르다. 이에 대한 실험 결과는 (그림5)와 같이 K-means와 FCM은 적합한 못한 중심좌표(○)를 찾는 반면, Isodata와 제안한 알고리즘은 올바른 중심좌표(○)를 찾고 있음을 알 수 있다. (그림5)의 오른쪽 그래프는 세대에 따른 적합도 함수의 변화량을 나타낸 것이다.

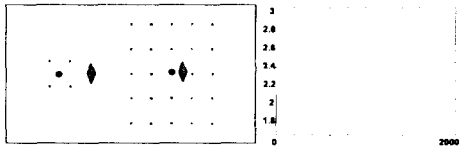

그림 5. 데이터A에 대한 실험결과 (1-0.5, n-0.5)

데이터C는 10개의 가우시안 분포(Gaussian distributions)로부터 500개의 입력데이터를 생성한 것이다.

(그림6)은 데이터C에 대한 실험결과로서 기존의 알고리즘들은 클러스터 특징을 닮지 같은 클러스터에 있는 데이터들의 유사성만을 고려하며, 유사성 또한 거리만으로 고려되는 반면, 제안한 알고리즘에서는 같은 클러스터에 있는 데이터들의 유사성과 다른 클러스터에 속하는 데이터들의 비유사성을 동시에 고려한 결과이다.

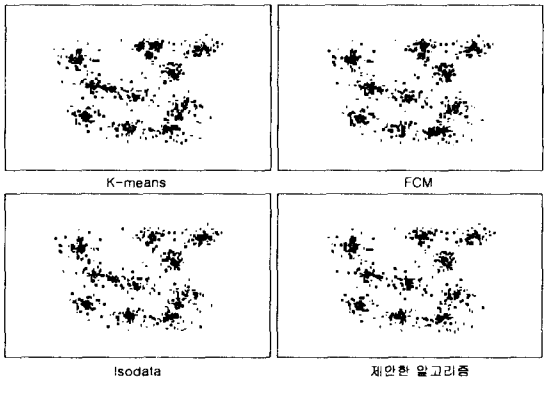

그림 6. 데이터C에 대한 실험결과 (1-0.7, n-0.3)

(2) 다차원데이터에 대한 실험

다차원데이터 역시 데이터C와 같은 방법으로 생성시켰다. (그림7)은 이러한 데이터를 제안한 알고리즘에 적용하였을 때의 실험결과이다. 상단의 중심좌표는 실험데이터를 만들기 위한 가우시안 분포의 평균이고, 하단은 제안한 알고리즘이 자동으로 찾아낸 클러스터 중심좌표이다. 그림에서는 두 개의 좌표가 비슷한 위치에 있다는 것을 확인 할 수 있으며, 또한 클러스터 개수의 변화를 살펴보면, 처음 임의 값으로 시작하여 (그림8.왼쪽)와 같이 9개->7개->11개->10개로 변해 가는 것을 살펴 볼 수 있으며, 그에 따른 적합도의 변화는 (그림8.오른쪽)에서 살펴 볼 수 있다.

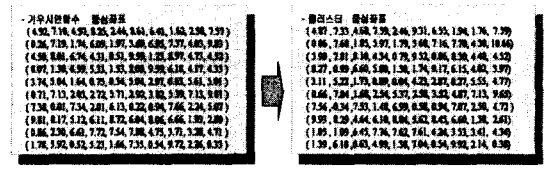

그림 7. 다차원 데이터 실험결과 (1-0.7, n-0.3)

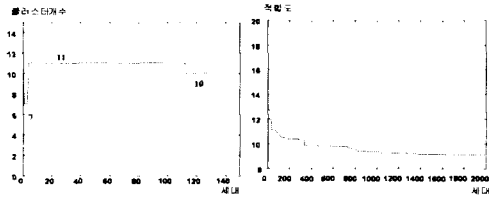

그림 8. 클러스터 개수 / 적합도 변화

5. 결론 및 향후 연구

본 논문에서는 자동으로 클러스터 개수를 결정하는 클러스터링 알고리즘을 제안하기 위하여 클러스터의 특성을 고려한 새로운 목적함수를 정의하여 적합도함수로 사용하였다.

제안한 알고리즘에서 사용한 적합도 함수는 표준편차 벡터를 계산하여 중심으로부터 포함된 데이터가 얼마나 떨어져 있는지 알 수 있는 분산도(dispersion)를 고려함으로써 같은 클러스터내의 유사성을 반영하였으며, 다른 클러스터간에 속해 있는 데이터들의 비유사성은 임의의 패턴과 모든 중심들간의 거리의 비율로서 얻어진 소속정도를 고려한 분리도(seperation)로 정의하였다.

이렇게 정의된 두 가지의 평가척도를 고려하여 자동으로 클러스터 개수를 결정할 수 있게 적합도 함수를 정의하였으며, 또한 가변길이 개체를 사용함으로써 새로운 진화 연산자인 집합교배 연산자(setclrossover operation)를 정의하였다.

제안한 알고리즘에서는 상징적(symbolic)인 데이터를 적용하는 부분은 고려하지 않고 있다.

향후연구계획으로는 제안한 알고리즘을 실제 데이터에 적용하여 보다 일반적인 타당성을 검증하고 지금까지 고려하지 않았던 상징적인 데이터들도 처리될 수 있는 알고리즘으로 확장할 것이다.

6. 참고문헌

[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] Morton Nadler, Eric P. Smith, "Pattern Recognition Engineering", John Wiley & Sons Inc, 1993.

[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, "진화프로그래밍을 이용한 퍼지 클러스터링", 정보과학회논문지(B) 제26권, 제1호, pp. 130-138, 1999

[8] Knuth, D.(1773). The art of computer programming, vol1. Fundamental Algorithms of Addison-Wesley Series in Computer Science and Information Processing. Addison-Wesley, Reading, MA.

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

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