2003년도 대한전기학회 하계학술대회 논문집 2003.7.21-23

변형된 Mountain 방법을 이용한 G-K 클러스터링 성능 개선

김승석, 전병석, 김주식, 유정웅, *이진국
충북대학교 전기전자공학부, *충주대학교 전기공학과

Improving the G-K Clustering Performance using the Modified Mountain Method

Sung-Suk Kim, Byeong-Seok Jeon, Joo-Sik Kim, Jeong Woong Ryu, Chin-Gook Lhee

Abstract - G-K 클러스터링이 가지는 우수한 클러스터 분류 성능에도 불구하고 데이터의 편중 및 분포 밀도에 의하여 클러스터링의 결과과 만족스럽지 못하는 경우가 발생한다. 제안된 방법에서는, G-K 클러스터링에서 데이터의 분포 및 밀도 등과 같은 다양한 조건에 대한 문제를 동시에 고려함으로써 클러스터링 결과를 개선한다. G-K 클러스터링에서 일부 파라미터의 수동적 파라미터 결정 방법을 Mountain 방법을 이용하여 능동적인 알고리즘으로 대치하여 클러스터 최적화 과정을 더욱 용이하게 한다. 이러한 클러스터링의 장점은 뉴로-퍼지 모델의 규칙 감소와 성능개선으로 나타나며 이를 시뮬레이션을 통하여 보이고자 한다.

1. 서 론

최근 뉴로-퍼지 시스템의 발달은 다양한 분야에 효과적으로 사용되고 있다[1][2]. 학습을 통한 뉴로-퍼지 시스템은 빠른 학습 능력과 우수한 성능을 가지지만 초기 파라미터에 대하여 많은 영향을 받는다[3]. 또한 뉴로-퍼지 시스템의 구조동정 및 파라미터 동정에서의 효과적인 구성 또한 시스템에 영향을 준다. 시스템의 전제부 결정에서 클러스터링 방법은 일반적인 그리드 분할 방법이 가지는 지수함수 형태로 증가하는 규칙 문제에 적응적으로 사용할 수 있다[2][3]. 일반적인 클러스터링 방법으로 Fuzzy C-Mean (FCM)등이 이용되고 있으나 유클리디언 노름을 사용, 데이터의 편중이나 밀도에 대하여 좋은 성능을 나타내지 못하였다. 또한 Gaussian Mixture Model을 통한 클러스터링 방법은 Expectation-Maximization 알고리즘을 통하여 일반적인 데이터의 분포에 대하여 좋은 결과를 볼 수 있지만 데이터의 편중이나 밀도에 대한 문제에서는 좋지 않은 클러스터 파라미터를 추정할 수 있는 문제를 가지고 있었다[4-6].

제안된 방법에서는 Gustafson-Kessel (G-K) 알고리즘을 이용하여 이러한 문제점을 해결하고자 하였으며 G-K 알고리즘이 가지는 문제점을 변형된 Mountain 방법을 이용하여 해결하였다[7][9]. Mahalanobis 노름을 기반으로 하는 G-K 알고리즘과 각 데이터 분포의 합으로 표현되는 변형된 Mountain 방법을 이용하여 데이터의 분포 및 밀도를 동시에 고려하는 능동적인 클러스터링을 실시하였다[7-9]. 이를 시뮬레이션 결과를 통하여 제안된 방법의 유용성을 보이고자 한다.

2. G-K 클러스터링 성능개선

2.1 G-K 클러스터링

G-K 클러스터링은 데이터 집합에서 다른 기하학적인 형태의 클러스터를 검출하기 위하여 유클리디언 노름이 아닌 Adaptive distance 노름을 이용한 FCM 알고리즘의 확장된 형태이다[7]. 이를 식으로 표현하면, 각 각의 클러스터는 다음과 같은 내적 노름을 가지는 행렬 A_i를 가진다.

$$D^2_{ikA_i}=(x_k-v_i)^T A_i(x_k-v_i) \tag{1}$$

여기서 행렬 A_i는 최적화 변수로 이용되며, 데이터의 위상학적 구조에서 각 클러스터에 대하여 적응적(Adaptive)이다. 즉 $A=(A_1,A_2,\ldots,A_c)$로 표현할 수 있다. 또한 G-K 알고리즘의 목적함수는 다음과 같이 정의한다.

$$J(X;U,V,A)=\sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m D^2_{ikA_i} \tag{2}$$

여기서 $U\in M_{fc}$, $V\in R^{n\times c}$, $m>1$이고, 이는 다음과 같이 해를 구할 수 있다.

$$(U,V,A)=\arg\min_{M_{fc}\times R^{n\times c}\times PD^n} J(X;U,V,A) \tag{3}$$

여기서 PD^n은 $n\times n$의 양(positive)으로 정의된 행렬이다. 고정된 A에 대하여 다음 식으로 적용할 수 있다.

$$\mu_{ik}=\frac{1}{\sum_{j=1}^{c}(D_{ikA}/D_{jkA})^{2/(m-1)}},\quad 1\le i\le c,\ 1\le k\le N \tag{4}$$

$$v_i=\frac{\sum_{k=1}^{N}(\mu_{ik})^m x_k}{\sum_{k=1}^{N}(\mu_{ik})^m},\quad 1\le i\le c \tag{5}$$

여기서, 목적함수는 A_i에 대하여 직접적으로 최적화 할 수 없다. 가능한 해를 구하기 위해서, A_i는 몇가기 방법을 이용하여 제한을 두는데, 제안된 방법에서는 행렬에 대하여 적용하였다.

$$\|A_i\|=\rho_i,\ \rho_i>0,\ \forall i \tag{6}$$

라그랑지 곱에 의한 최적화를 실시하면 A_i는 다음과 같이 얻을 수 있다.

$$A_i=[\rho_i \det(F_i)]^{-1/n}F_i^{-1} \tag{7}$$

여기서 F_i는 i번째 클러스터의 퍼지 공분산 행렬로서 다음과 같다.

$$F_i=\frac{\sum_{k=1}^{N}(\mu_{ik})^m(x_k-v_i)(x_k-v_i)^T}{\sum_{k=1}^{N}(\mu_{ik})^m} \tag{8}$$

식(7) 와 식(8)을 식(6)에 대입하면 x_k와 v_i에서의 일반화된 Mahalanobis 거리 노름을 얻을 수 있다. 여기서 공분산은 U에서 소속 정도에 의하여 가중처리된다.

G-K 알고리즘
초기화 단계:
주어진 데이터 집합 X에 대하여, 클러스터의 수 $1<c<N$를 초기화하고 가중 퍼지 멱지수 $m>1$, 종결조건 $\varepsilon>0$을 설정한다.

단계 1: 중심을 계산한다.

$$v_i^{(l)}=\frac{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m x_k}{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m},\quad 1\le c\le N$$

단계 2: 클러스터의 공분산행렬을 구한다.

$$F_i=\frac{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m (x_k-v_i^{(l)})(x_k-v_i^{(l)})^T}{\sum_{k=1}^{N}(\mu_{ik}^{(l-1)})^m}$$

단계 3: 노름을 계산한다.

$$D_{ikA_i}^2=(x_k-v_i^{(l)})^T[\rho_i \det(F_i)^{1/n}F_i^{-1}](x_k-v_i^{(l)})$$

단계 4: 분할 행렬을 갱신한다.

$$\mu_{ik}=\frac{1}{\sum_{j=1}^{c}(D_{ikA}/D_{jkA})^{2/(m-1)}}$$

단계 5: 종결조건을 검사한다.
$\|U^{(l)}-U(l-1)\|<\varepsilon$이면 알고리즘을 종료하고, 이외의 경우 단계 1로 돌아간다.

2.2 변형된 Mountain 클러스터링

Mountain 클러스터링은 Yager와 Filev에 의하여 제안되었으며 밀도 비율을 기반으로 클러스터 중심에 가깝도록 평가하는 간단하면서도 효과적인 방법이다. 이방법은 FCM처럼 초기 클러스터 중심을 획득하는데 사용할 수 있다. 또한 사람이 작업하는 것과 같이 데이터 집합의 시각적인 클러스터형성을 기반으로 하고 있다.
일반적인 mountain 함수는 다음과 같이 표현할 수 있다.

$$m(v)=\sum_{i=1}^{N}\exp\left(-\frac{\|x_i-v_j\|^2}{2\sigma^2}\right) \tag{9}$$

여기서 σ는 사전에 정해진 상수이다.
이 경우, 중심과의 거리에 따라 mountain 함수는 크게 달라지게 된다. 즉, 데이터 밀도 비율에 의하여 mountain 함수의 크기는 영향을 받고 상수 σ에 의하여 유연성이 결정이 된다.
제안된 방법에서는 각각의 데이터간의 거리를 모두 측정하고, 각 데이터에서 가지는 mountain 함수의 크기를 모두 더하여 각데이터의 밀도정보로 이용하였다.

$$m(x_j)=\sum_{i=1}^{N}\exp\left(-\frac{\|x_i-x_j\|^2}{2\sigma^2}\right) \tag{10}$$

각 함수는 모든 데이터와의 계산된 값의 합을 가지고 있으며, 이는 데이터 분포와 밀도에 대한 정보를 가지고 있다.

3. 개선된 G-K 클러스터링

G-K 알고리즘은 Mahalanobis 노름을 이용하므로써 다양한 형태의 클러스터를 효과적으로 추정한다. 하지만 상수 ρ을 항상 1로 고정함으로써 각 클러스터는 동일한 체적(volume)을 가지려는 성질을 가진다. 이 경우, 편중되거나 밀도가 서로 다른 데이터 집합을 경우 클러스터의 추정 결과가 원하는 결과와 달라지는 문제점을 가지고 있다. 즉 데이터 집합의 형태에 적응적으로 클러스터를 추정하기 위해서는 체적을 의미하는 ρ를 능동적으로 추정하여야 한다. 변형된 Mountain 방법에 의한 함수의 값은 각 클러스터에 대하여 체적을 간접적으로 나타내므로 이를 정규화 과정을 통하여 ρ에 적용한다.

$$m(v_i)=\max(m(x_k)\mid_{x_k\in c_i})-\min(m(x_j)\mid_{x_j\in c_i}) \tag{11}$$

$$\rho_i=\frac{m(v_i)}{\sum_{j=1}^{c}m(v_j)} \tag{12}$$

각각의 정규화된 $m(v_i)$는 ρ_i로 적용되어 G-K 알고리즘으로 들어간다.
이를 그림으로 보면 다음과 같다.

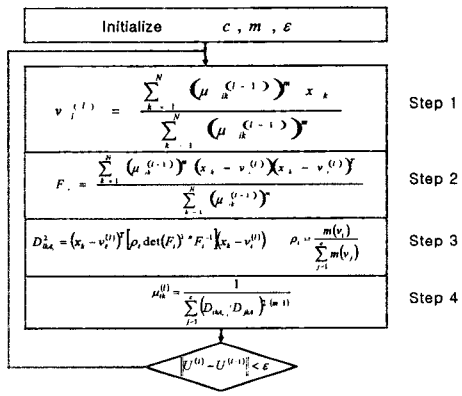

그림 1. 제안된 알고리즘

FCM은 데이터의 패턴이 잘 분리되어 있거나 같은 크기나 형태를 가질 때 좋은 성능을 보이는 반면 그렇지 않은 데이터의 집합에 대하여는 데이터와는 다른 클러스터 정보를 추정하였으나 G-K 클러스터링은 이러한 문제점을 어느정도 해결할 수 있다.

4. 시뮬레이션 및 결과

실험에 사용된 데이터는 편중된 분포와 밀도를 가지도록 임의로 750개의 데이터 집합을 생성하였다. 그림 2에서 볼 수 있드시 각 클러스터의 크기와 밀도는 크게 차이가 난다.
먼저 변형된 Mountain 방법을 이용하여 각 클러스터의 Mountain 함수의 크기를 결정한다. 이 경우 각 데이터 집합의 두 개의 차원을 가지므로 각각의 거리는 유클리디언 노름을 이용하여 계산하였다. 이렇게 계산된 함수의 크기는 각 데이터에 대한 각각의 거리를 모두 합산하여 저장한다. 이 함수의 크기는 G-K 클러스터링 알고리즘이 각회 수행될 때마다 추정되는 ρ_i의 계산에 이용된다.
먼저 그림 2은 전체 데이터 분포를 나타내었으며 그림 3은 FCM 알고리즘을 이용하여 클러스터를 추정하였을

경우 추정된 클러스터와 중심을 나타내었다. 또한 그림 4은 일반적인 G-K 알고리즘을 이용하였을 경우 추정결과를 나타내었다.

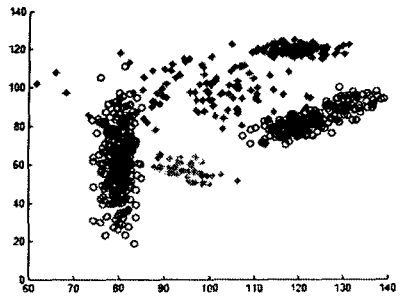

그림 2. 클러스터 분포

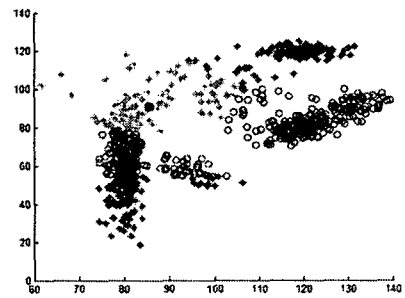

그림 3. FCM 알고리즘에 의한 클러스터 추정

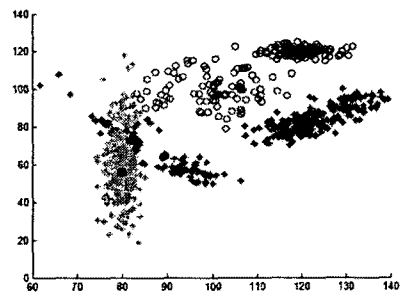

그림 4. G-K 알고리즘에 의한 클러스터 추정

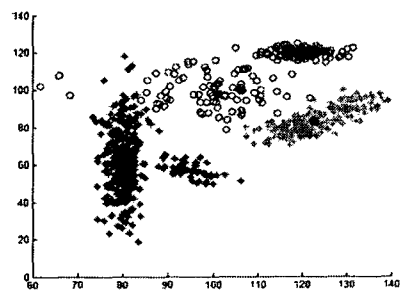

그림 5. 제안된 방법에 의한 클러스터 추정

그림 5에서 볼 수 있듯이 제안된 방법에 의한 클러스터링의 결과가 이전의 연구 결과보다 우수한 것을 알 수 있다. 이를 표로 표현하면 다음과 같다.

표 1. 성능 평가 : 오차

	FCM	G-K	제안된 방법	비고
1번째	42	36	19	
2번째	153	17	4	
3번째	0	0	0	
4번째	0	0	0	
5번째	39	50	0	
종합	234	103	23	

5. 결 론

클러스터링은 대규모 데이터를 효과적으로 표현하거나 관리할 수 있다는 장점으로 활발해 연구되고 있는 분야이다. 이러한 클러스터링 알고리즘에서, 클러스터의 파라미터들이 얼마나 더 정확하게 클러스터를 표현하는 것이 중요한 관심이다. 제안된 방법에 의한 클러스터 추정은 FCM 알고리즘이 가지는 유클리디언 노름에 의한 추정 방법에 의한 문제점을 해결하였으며, 일반적인 G-K 클러스터링의 동일체적을 유지하려는 문제점을 변형된 Mountain 방법을 통하여 해결하였다. 추후 연구과제로는 생성된 클러스터 파라미터를 적절하게 표현할 수 있는 퍼지 소속함수 연구등을 통하여 뉴로-퍼지 시스템의 성능을 개선하는 것이다.

[참 고 문 헌]

[1] J. S. R. Jang, C. T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing : A Computational Approach to Learning and Machine Intelligence, Prentice Hall, 1997.

[2] Simon Haykin, Neural Network : A Comprehansive Foundation, Prentice Hall, 1999.

[3] J. S. R. Jang, "ANFIS : Adaptive-Networks based Fuzzy Inference system", IEEE Trans. on System, Man, and Cybern, Vol. 23, No. 3, pp. 665-685, 1993

[4] Timothy J. Ross, Fuzzy Logic with Engineering Application, McGraw-Hill, 1995.

[5] Todd. K. Moon, "The Expectation-Maximization Algorithm", IEEE Signal Processing Magazine, 1996.

[6] Guorong Xuan, Wei Zhang, Peiqi Chai, "EM algorithm of Gaussian Mixture Model and Hidden Markov Model", Image Processing, Proceedings, International Conference on, Vol. 1, pp. 145-148, 2001.

[7] Robert Babuska, Fuzzy Modeling for Control, Kluwer Academic, 1998.

[8] Uzay Kaymak, Magne Setnes, "Fuzzy Clustering With Volume Prototypes and Adaptive Clustering Merging", IEEE Trans on Fuzzy Systems, Vol. 10, No. 6, pp. 705-712, 2002.

[9] Raghu Krishnapuram, Jongwoo Kim, "A Note on the Gusftafson-Kessel and Adaptive Fuzzy Clustering Algorithms", IEEE Trans on Fuzzy Systems, Vol. 7, No. 4, 1999.