응용통계연구 제 15권 2호, 2002년, pp. 381-393

K-모드 알고리즘과 ROCK 알고리즘의 개선

김 보 화 [1] 김 규 성 [2]

요 약

K-모드(modes) 알고리즘과 락(ROCK) 알고리즘은 대규모 범주형 자료에 적용 가능한 데이터 군집화 방법이다. 이 논문에서는 두 알고리즘을 고찰하였으며, 두 알고리즘의 단점을 보완한 개선된 데이터 군집화 알고리즘을 제안하였다. 그리고 실제자료에 제안된 방법을 적용한 모의실험을 실시하여 제안된 방법이 데이터 군집화의 성능을 향상시킬 수 있음을 보였다.

주요용어: 계통추출, 데이터 군집화, 범주형 자료, 사후 할당, 연관성 측도.

1. 서론

데이터 군집화(data clustering)는 군집내의 객체(object)들은 비슷한 속성을 갖도록 하고 군집간의 객체들은 서로 상이한 속성을 갖도록 객체들을 나누는 기법이며, 군집을 형성하는 방법에 따라 계층적 군집화(hierarchical clustering) 방법과 분할 군집화(partitioning clustering) 방법으로 구분할 수 있다. 계층적 군집화는 거리가 가까운 객체들을 순차적으로 묶어나가는 병합적(agglomerative)인 방법과 반대로 거리가 먼 객체들을 분리해 가는 분할적(divisive)인 방법으로 나눌 수 있으며, 이 방법에서는 어떤 객체가 하나의 군집에 포함되면 다른 군집으로는 이동하지 않는 성질이 있다. 분할 군집화는 객체들을 몇 개의 군집으로 분할하는 방법이며, 주어진 기준 함수를 최적화하는 군집을 찾는다. 이 방법에서는 군집을 형성하는 과정에서 군집에 객체들의 재 할당이 반복적으로 일어나기 때문에, 초기에 어떤 객체가 부적절하게 군집에 할당된다 하더라도 나중에는 변경될 수 있는 성질이 있다.

전통적인 데이터 군집화 방법들은 양적인 속성을 갖는 자료를 대상으로 개발되어 왔다. 대표적인 분할 군집화 방법의 하나인 K-평균(K-means) 군집화는 객체간의 거리를 유클리디안 거리로 정의한 후 군집의 평균을 계산하여 비용함수를 최소화하도록 군집을 형성해 가는 방법이다(MacQueen, 1967). 이 알고리즘은 간단하여 구현이 쉬울 뿐만 아니라 알고리즘이 수렴하기 때문에 전통적으로 널리 이용되어 왔다(Bezdek, 1980; Selim and Ismail, 1984). K-평균 군집화 방법은 이상치에 민감하게 반응하고 군집들이 대체로 비슷한 크기로 형성되는 경향이 있기 때문에 이에 대한 개선책으로 K-공간 중위수 군집화(k-Spatial medians clustering) 방법(Spath, 1980; Jhun and Jin, 2000)과 이중 K-평균 군집화(허명회,

1) (150-873) 서울시 영등포구 여의도동 16, 한국산업은행
2) (130-743) 서울시 동대문구 전농동 90, 서울시립대학교 컴퓨터·통계학과, 부교수
E-mail : kskim@uos.ac.kr

2000) 방법 등이 제안되었다. 또한 K-평균 군집화 방법은 양적인 속성을 갖는 대용량 데이터에 적용하기 쉽기 때문에 데이터 마이닝 분야에서 군집화 방법으로 효과적으로 활용될 수 있다.

대용량 자료에는 양적인 자료와 범주형 자료가 섞여 있는 경우가 대부분이어서 대용량 자료를 군집화 하기 위해서는 범주형 자료를 대상으로 하는 군집화 방법의 연구가 필수적이다. 범주형 자료를 대상으로 하는 군집분석 방법으로 K-모드(K-modes) 알고리즘(Huang, 1997)과 락 알고리즘(RObust Clustering using linKs, ROCK, Guha 외 2인, 1999) 등이 대표적으로 알려져 있다. K-모드 알고리즘은 분할 군집화 방법의 일종으로, K-평균 알고리즘의 형식을 유지하면서 범주형 자료에 적합하도록 제안된 방법이다. 따라서 K-평균 알고리즘과 마찬가지로 절차가 간단하고 수렴속도가 빠른 반면, 초기 모드 설정에 따라 군집의 결과가 달라질 수 있는 단점이 있다. 락 알고리즘에서는 범주형 자료를 부울형 자료로 변환해서 객체간의 유사성을 정의한 후 데이터의 모든 객체를 동시에 비교하여 유사성이 가장 큰 객체들을 순차적으로 병합해 가는 계층적 군집방법이다. 이 알고리즘에서는 유사성이 가장 큰 객체들을 우선적으로 병합하기 때문에 거대 군집을 형성한 후에도 군집을 형성하지 못하고 남는 객체들이 있을 수 있다.

이 논문에서는 범주형 자료에서 이용 가능한 K-모드 알고리즘과 락 알고리즘의 단점을 보완한 개선된 군집분석 방법을 제안한다. 이 논문은 다음과 같이 구성되어 있다. 제 2장에서는 K-모드 알고리즘과 락 알고리즘을 비교·분석하고, 제 3장에서는 두 알고리즘의 개선된 알고리즘을 제안한다. 제 4장에서는 제안된 알고리즘을 실제 자료에 적용하여 모의실험을 실시하고 결과를 해석하며, 마지막으로 제 5장에서는 간단한 요약과 더불어 향후 연구과제를 제시한다.

2. K-모드 알고리즘과 ROCK 알고리즘

집합 $\mathrm{X} = \{\mathrm{x}_1, ..., \mathrm{x}_n\}$는 개의 객체로 구성되어 있고, 각 객체는 m개의 범주형 변수 값을 갖는다고 하자.

$$\mathrm{x}_i = (x_{i1}, ..., x_{im})', \quad i = 1, ..., n$$

그리고 j번째 범주형 변수 A_j는 l_j개의 수준을 가지며 각 수준을 $c_{j1}, ..., c_{jl_j}$라고 하자.

2.1. K-모드 알고리즘

Huang(1997)이 제안한 K-모드 알고리즘은 K-평균 알고리즘의 기본 구조를 유지하면서 범주형 자료를 처리할 수 있도록 고안된 방법이다. Huang은 범주형 변수의 값들로 구성된 두 객체 x_{i_1}, x_{i_2}의 비유사성을 객체간에 일치하지 않는 변수의 수인 $d(\mathrm{x}_{i_1}, \mathrm{x}_{i_2})$ 로 정의하였다.

$$d(\mathrm{x}_{i_1}, \mathrm{x}_{i_2}) = \sum_{j=1}^{m} \delta(x_{i_1 j}, x_{i_2 j}) \tag{2.1}$$

여기서 $\delta(a,b)$는 두 값이 일치하지 않을 때 1의 값을 갖고, 그렇지 않을 때에는 0의 값을 갖는 지시함수이다.

$$\delta(a,b) = \begin{cases} 0, & \text{만일 } a = b \\ 1, & \text{만일 } a \neq b \end{cases} \tag{2.2}$$

또한 집합 X에 대응되는 모드(mode)인 $\mathrm{q}^* = (q_1^*, .., q_m^*)'$ 는 집합 X내의 객체들과 비유사성이 가장 적은 벡터로 정의하였다. 즉, 모드 q^*는 벡터 $\mathrm{q} = (q_1, .., q_m)'$ 중에서 비유사성의 합 $D = (\mathrm{q}, \mathrm{X})$을 최소로 하는 벡터이다.

$$D(\mathrm{q}, \mathrm{X}) = \sum_{i=1}^{n} d(\mathrm{x}_i, \mathrm{q}) \tag{2.3}$$

집합 X의 모드 q^*는 다음과 같이 찾을 수 있다. 집합 X에서 변수 A_j가 수준 c_{jk}를 갖는 빈도 수를 n_{jk}라고 하면, A_j가 c_{jk}를 가질 상대빈도는

$$f(A_j = c_{jk}|\mathrm{X}) = \frac{n_{jk}}{n}, \ k = 1, ..., l_j$$

가 된다. 그러면 모든 $j(= 1, ..., m)$ 에 대하여

$$f(A_j = q_j^*|\mathrm{X}) \geq f(A_j = c_{jk}|\mathrm{X})$$

을 만족하는 $\mathrm{q}^* = (q_1^*, ..., q_m^*)'$는 비유사성의 합 $D(\mathrm{q}, \mathrm{X})$을 최소로 하므로, 결과적으로 집합 X의 모드가 된다. 즉, 변수별로 빈도가 가장 큰 범주 값들의 조합이 그 집합의 모드가 되는 것이다.

K-모드 알고리즘의 각 단계를 정리하면 아래와 같다.

- 단계 1. K개 군집의 초기 모드 $\{\mathrm{q}_1^{(0)}, ..., \mathrm{q}_K^{(0)}\}$를 선택한다.
- 단계 2. 객체 n개와 초기모드 $\{\mathrm{q}_1^{(0)}, ..., \mathrm{q}_K^{(0)}\}$의 비유사성을 계산하여 비유사성이 가장 적은 군집으로 객체를 할당한 후, K개 군집내의 모드를 갱신하여 갱신된 첫 번째 모드 $\{\mathrm{q}_1^{(1)}, ..., \mathrm{q}_K^{(1)}\}$를 얻는다.
- 단계 3. 모든 객체와 갱신된 모드의 비유사성을 다시 구한 후, 만일 다른 군집의 모드와의 비유사성이 더 적으면 해당 객체를 그 군집으로 다시 할당하고 군집내의 모드를 갱신한다.
- 단계 4. 단계 3을 변화가 없을 때까지 반복 실행한다.

Huang(1997)은 초기 모드 선택 방법으로 두 가지를 제안하였다. 한 가지 방법은 집합 X에서 임의로 서로 다른 K개의 객체를 선택하여 초기 모드로 사용하는 것이다. 다른 방법은 변수별로 범주의 빈도를 구해서 빈도가 가장 큰 범주가 K개의 초기 모드에 골고루 반영되도록 배열하고, 이렇게 정해진 K개의 초기 모드와 가장 유사한 객체를 X집합에서 선택하여 초기 모드로 사용하는 것이다.

2.2. ROCK 알고리즘

Guha외 2인(1999)이 제안한 락 알고리즘은 군집간의 링크값을 이용하여 군집을 순차적으로 병합해 나가는 계층적 군집방법이다. 우선 두 객체 $(\mathrm{x}_i, \mathrm{x}_j)$의 유사성을 다음과 같이 정의한다.

$$\mathrm{sim}(\mathrm{x}_i, \mathrm{x}_j) = \frac{m - \sum_{k=1}^{m} \delta(x_{ik}, x_{jk})}{m + \sum_{k=1}^{m} \delta(x_{ik}, x_{jk})} \tag{2.4}$$

여기서 $\delta(a,b)$는 앞의 식(2.2)에서 정의된 지시함수이다. 유사성 $\mathrm{sim}(\mathrm{x}_i, \mathrm{x}_j)$은 0과 1사이의 값을 가지며, 두 객체가 유사할수록 큰 값을 갖는다. 두 객체간의 유사성이 주어진 기준값(threshold) θ 보다 크면 두 객체를 이웃(neighbor)이라 부르고, 객체 x_i의 이웃군 $G(\mathrm{x}_i)$은 x_i와 이웃인 객체들의 집합으로 정의한다.

$$G(\mathrm{x}_i) = \{\mathrm{x}_i |\ \mathrm{sim}(\mathrm{x}i, \mathrm{x}_j) \geq \theta \ \ (i \neq j)\} \tag{2.5}$$

그리고 두 객체의 링크, $\mathrm{link}(\mathrm{x}_i, \mathrm{x}_j)$,는 두 객체의 이웃군에 속하는 공통 이웃의 개수로 정의하고, 두 군집 C_i와 C_j의 링크는 군집에 속하는 객체들의 링크의 합으로 정의한다.

$$\mathrm{link}(C_i, C_j) = \sum_{\mathrm{x}_p \in C_i, \mathrm{x}_q \in C_j} \mathrm{link}(\mathrm{x}_p, \mathrm{x}_q) \tag{2.6}$$

링크의 값이 클수록 두 객체나 군집은 유사한 것으로 볼 수 있다. 군집의 병합을 위해서 군집간의 링크 값, 군집에 속하는 객체 수 및 유사성의 기준값 θ를 고려한 $g(C_i, C_j)$을 계산한 후, $g(C_i, C_j)$의 값이 가장 큰 두 군집을 병합한다.

$$g(C_i, C_j) = \frac{\mathrm{link}(C_i, C_j)}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}} \tag{2.7}$$

여기서 n_i는 군집 C_i에 속하는 객체의 수이고, $f(\theta) = (1-\theta)/(1+\theta)$이다. 군집의 병합기준 g는 θ의 함수이고 θ값에 따라 병합되는 군집의 수가 달라지기 때문에, 실제 데이터에서는 θ의 값을 자료의 특성에 따라 정해 주어야 한다.

이상의 락 알고리즘의 절차를 정리하면 다음과 같다.

- 단계 1. 주어진 θ에 대하여 객체간의 유사성을 계산하여 객체별로 이웃과 이웃군을 구한다.
- 단계 2. 군집간의 링크값 $\mathrm{link}(C_i, C_j)$을 계산한 후, $g(C_i, C_j)$을 구한다.
- 단계 3. $g(C_i, C_j)$ 값이 가장 큰 두 군집을 병합하고 군집간의 링크 값을 갱신한다. 이때, 병합된 군집과 다른 군집간의 링크 값은 병합되기 전의 링크 값의 합으로 계산한다.
- 단계 4. 군집의 개수가 일정 수에 이를 때까지 단계 2와 단계 3의 과정을 반복 수행한다.

락 알고리즘은 병합적인 방법이므로 최종 군집의 수를 분석자가 임의로 정해줄 수 있다. 그러나 실제 데이터로 락 알고리즘을 구현해보면 미리 설정한 군집의 수에서 병합이 멈추기 어려운 경우가 발생하기도 한다. 예를 들어 최종 군집의 수를 2라고 했을 때, 4절에서 수행한 모의실험 결과를 보면 $\theta = 0.52$ 일 때 거대 군집이 2개 생기고 남은 객체가 12개가 생기는 과정을 거치게 된다. 이 단계에서 병합을 계속하면 남은 객체가 거대 군집에 병합되는 것이 아니라 거대 군집 2개가 서로 병합되어 하나가 되고 남은 객체는 여전히 12개가 되는 상황이 발생한다. 이 과정을 계속하면 거대 군집 하나에 남은 객체 하나로 군집이 형성되는 결과가 초래된다. 따라서 이 예에서는 거대 군집 2개와 남은 객체가 12개일 때 병합을 멈추는 것이 현실적으로 의미 있는 일이다. 형성되는 거대 군집의 수는 자료의 속성과 θ의 값에 따라 달라지며, 군집을 형성하지 못하고 남는 객체의 처리는 락 알고리즘이 사후적으로 보완해야 할 사항이다.

2.3. 두 알고리즘의 비교

범주형 자료에 대한 군집화 방법으로 제안된 K-모드 알고리즘은 군집의 수를 미리 정하고 실행하는 분할 군집화 방법이고, 락 알고리즘은 가장 유사성이 큰 군집들을 단계적으로 병합하는 계층적 군집화 방법이므로 두 알고리즘은 군집을 형성해 가는 과정이 서로 다르다.

양적인 변수와 달리 범주형 자료는 두 객체간의 유사성을 정의해야 하는데, K-모드 알고리즘은 두 객체간의 일치하지 않는 변수의 개수로 두 객체의 비유사성을 정의하는 반면, 락 알고리즘은 전체 범주형 변수 중 일정 비율 이상이 일치하면 서로 이웃이라고 하고, 두 객체간 공통 이웃의 개수를 링크라고 하여 이를 두 군집간의 유사성으로 정의하고 있다. K-모드 알고리즘과 락 알고리즘은 대용량 범주형 자료를 분석의 대상으로 하고 있으므로 알고리즘의 수렴 속도가 중요하다. Murtagh(1992)는 대용량 자료에 대한 군집화에서 분할 군집화 방법이 계층적 군집화 방법 보다 수렴 속도가 빠르다는 것을 증명하였다. 또한 Huang(1997)은 실제 자료에 K-모드 알고리즘과 K-평균 알고리즘을 적용하여 K-모드 알고리즘이 K-평균 알고리즘 보다 수렴 속도가 빠르다는 것을 보였다. 대용량 범주형 자료에 대한 군집화에서 분할 군집화 방법인 K-모드 알고리즘이 계층적 군집화 방법인 락 알고리즘 보다 수렴 속도가 더 빠르다.

다음으로 두 알고리즘의 장·단점을 비교해보자. K-모드 알고리즘은 절차가 간결하고, 범주형 변수로 설명되는 두 객체간 비유사성의 정의가 간단하다. 또한 대용량 자료에서 수렴 속도가 빠르다는 장점이 있다. 반면 군집분석 전에 미리 군집의 수를 정해 주어야 하고, 알고리즘의 첫 번째 단계에서 초기 모드를 어떻게 정하는가에 따라 군집의 결과가 달라질 수 있는 단점이 있다. 락 알고리즘은 두 객체간의 유사성을 정의할 때 링크라는 개념을 이용하여 두 객체 뿐 만 아니라 자료의 모든 객체를 동시 비교한다. 그리고 어떤 군집을 먼저 병합할 것인가를 판단할 때 군집의 크기까지 고려하기 때문에 상대적으로 크기가 작은 군집도 제대로 유지할 수 있다는 장점이 있다. 그러나 범주형 자료를 부울형 자료로 변환해서 사용하기 때문에 과정이 복잡하고, 병합하는 과정에서 거대 군집과 군집을 형성하지 못하고 남아 있는 객체가 있는 과정을 거치며, 병합을 계속 할 경우 남은 객체가 거대 군집에

포함되는 것이 아니라 거대 군집들이 서로 병합되는 일이 생길 수 있다.

3. 개선된 K-모드 알고리즘과 ROCK 알고리즘

3.1. 개선된 K-모드 알고리즘

K-모드 알고리즘의 가장 큰 단점은 초기 모드 값의 설정에 따라 군집의 결과가 달라질 수 있는 점이다. 따라서 데이터의 특성을 고려하여 초기 모드를 설정하면 K-모드 알고리즘의 효율을 높일 여지가 있다. 이 소절에서는 K-모드 알고리즘의 효율을 높일 수 있는 개선 방안을 제안한다.

3.1.1. 초기 모드를 계통 추출

초기 모드를 랜덤하게 선택하는 것보다는 군집형성에 영향을 많이 주는 변수를 중요한 변수로 보고 이 변수의 값이 초기 모드에 골고루 선택되도록 하면 군집화의 효율을 높일 수 있을 것이다. 크기 순으로 정렬된 양적인 자료의 경우 계통추출법이 임의추출법보다 추론의 효율이 높은 점에 착안하여, 범주형 자료의 경우에도 군집을 형성에 영향을 많이 주는 변수를 중심으로 정렬한 후 초기 모드를 계통추출 한다. 계통추출을 하기 위해서는 자료의 정렬이 필수적이므로, 군집을 구분하는 변수와 연관성이 큰 범주형 변수 순으로 다중 정렬을 하거나 혹은 군집 형성에 중요하다고 알려진 변수를 중심으로 정렬을 한다. 군집화를 하려는 대상이 대규모 범주형 자료이므로 자료 정렬에 많은 계산시간이 소요될 수 있다. 이 경우 범주형 변수 전체가 아닌 중요한 일부 변수에 대해서만 정렬을 하면 계산시간이 단축될 것이다.

초기 모드를 계통 추출하는 절차는 다음과 같다.

- 단계 1. 각 범주형 변수에 대하여 연관성 측도(measure of association) 등을 통하여 군집형성에 영향을 많이 주는 변수에 우선 순위를 부여한다.
- 단계 2. 우선 순위를 유지하며 전체 객체를 다중 정렬한다(혹은 일부 중요 변수에 대해서만 정렬).
- 단계 3. 정렬된 자료에서 K개 객체를 계통추출하여 초기 모드로 선택한다.

여기서 단계 3의 계통추출방법은 다음과 같다. 만일 n/K 가 자연수이면 1과 n/K 사이에서 하나의 난수를 발생한다. 이 난수를 a_1이라 하자. 그러면 순차적으로 아래의 순서에 해당하는 객체를 초기 모드로 선정한다.

$$a_i = a_1 + (i-1) \times \frac{n}{K}, \quad i = 2, ..., K$$

만일 n/K 값이 정수가 아니면 순환계통추출을 이용한다. 즉, n개의 객체 중의 하나를 랜덤하게 선정한다. 이를 b_1라고 하자. 그러면 다음에 선택되는 객체는 아래와 같다.

$$b_i = b_1 + [(i-1) \times \frac{n}{K}], \quad i = 2, ..., K$$

여기서 $[c]$는 c를 반올림한 값을 사용한다. 만약 b_i 값이 n보다 큰 경우는$(b_i - n)$ 번째에 해당하는 객체를 선택한다. 이와 같이 초기 모드를 선정하면 군집 형성에 설명력이 높은 변수가 초기 모드 선정에 골고루 반영되는 장점이 있다.

3.1.2. 범주의 개수 고려

자료에서 초기 모드를 임의로 선택할 경우 범주의 수가 적은 변수에서는 각 수준이 골고루 뽑히지 않을 가능성이 크다. 범주의 수가 적은 변수의 수준이 골고루 선택되도록 하기 위해서는 이 변수로 객체들을 정렬한 후 각 범주에서 임의로 객체를 선택하면 된다. 이 방법은 범주의 개수가 적은 변수의 수준이 초기 모드에 골고루 선택되도록 하는 방법이다.

범주의 개수를 고려한 초기 모드 선택방법의 절차는 다음과 같다.

- 단계 1. 범주의 개수가 적은 변수 순으로 변수에 우선 순위를 준다.
- 단계 2. 변수의 우선 순위 순으로 객체를 다중 정렬한다(혹은 일부 중요 변수에 대해서 정렬).
- 단계 3. 정렬된 자료에서 범주의 개수가 적은 변수가 K개의 초기 모드에 골고루 반영되도록 한다.

단계 3에서 K개의 초기 모드를 선택하는 방법은 다음과 같다. 범주의 개수가 적은 변수를 A라 하고 이 변수의 범주의 수를 l이라고 하자. 만일 $K = l$이라면 다중 정렬된 자료에서 변수 A의 범주 수준별로 하나의 객체를 임으로 선택하여 초기 모드로 사용한다. $K > l$인 경우, 변수 A의 각 범주에서 하나의 객체를 임의로 선택하고, 나머지 $(K-l)$개의 많은 객체를 포함하는 범주에서 순차적으로 선택한다. 마지막으로 $K < l$ 인 경우, 많은 객체를 포함하는 범주부터 차례대로 각 범주에서 임의로 하나의 객체를 선택하여 초기 모드로 사용한다.

3.2. 개선된 ROCK 알고리즘

락 알고리즘은 병합과정에서 거대 군집과 군집을 형성하지 못하고 남아 있는 객체가 있는 단계를 거칠 수 있다. 군집을 형성하지 못한 객체들은 다른 객체들에 비해 상대적으로 유사성이 적은 것들이며, 락 알고리즘은 유사성이 큰 객체들을 우선적으로 묶어가기 때문에 알고리즘을 실행하다 보면 유사성이 적은 객체들이 뒤에 남겨지게 된다. 락 알고리즘에서는 일단 군집에 포함된 객체는 갱신을 하지 않기 때문에 유사성이 적은 객체는 그대로 남아 있을 가능성이 크다. 남겨진 객체를 군집에 포함시키기 위해서 병합을 계속하면 남아 있는 객체의 상당수를 군집에 포함시킬 수 있으나, 동시에 이미 형성된 거대 군집들끼리 병합되는 현상이 발생하기도 한다. 따라서 이 단계에서 병합을 계속하여 거대 군집을 서로 병

합하는 것보다는 병합을 멈추고 군집을 형성하지 못한 객체들을 이미 만들어진 군집에 사후 할당하는 방법이 더 효과적일 수 있다.

사후 할당은 군집을 형성하지 못하고 남은 객체들과 락 알고리즘으로 형성된 군집과의 유사성을 구해서 가장 유사한 큰 군집에 남은 객체들을 할당하는 것이다. 남은 객체와 군집의 유사성은 군집의 대표 객체와 남은 객체가 서로 일치하는 범주형 변수의 개수로 정의한다. 이때 군집의 대표 객체는 군집내의 객체를 대상으로 범주형 변수별로 빈도가 가장 큰 값들로 만든다.

사후 할당 과정은 다음과 같다.

- 단계 1. 이미 형성된 각 군집에서 대표 객체를 설정한다.
- 단계 2. 남은 객체와 군집의 대표 객체간의 유사성을 계산한다.
- 단계 3. 유사성이 가장 큰 군집에 남은 객체를 사후 할당한다. 이 때, 만약 유사성이 가장 큰 군집이 2개 이상일 경우, 임의로 하나의 군집을 선택하여 할당한다.

4. 모의실험

모의실험에 이용된 자료는 1984년 미국 의회의 선거 자료로, UCI 기계 학습 자료실(UCI Machine Learning Repository)에서 얻은 것이다. 총 16개의 문항(예를 들면, 범죄 관련 경험 유무, 이민 유무 등)이 있고 각 문항의 응답은 '예' 또는 '아니오'이다. 여기서 무응답은 하나의 범주로 고려하였기 때문에 각 변수는 3개의 범주를 갖는다. 총 435명의 의원이 응답하였고 그 중에서 공화당원은 168명(38.62%)이고 민주당원은 267명(61.38 %)이다.

4.1. K-모드 알고리즘

미국 의회의 선거자료에 군집의 수와 초기 모드 선택 방법을 달리하면서 K-모드 알고리즘을 적용하였다. 개선된 방법에서 연관성 측도를 구하기 위해서 소속정당을 나타내는 변수를 종속변수로 하고 나머지 16개의 변수를 설명변수로 하여 연관성 측도를 구하였다. 두 명목형 변수간의 연관성 측도로 대칭 람다(symmetric lambda)를 사용하였는데, 연관성 측도가 0.5 이상인 것이 6개(37.5%)였고, 연관성이 전혀 없는 변수가 1개(6.25%) 있었다. 군집 형성과 가장 연관성이 큰 변수는 '의사들의 진료비 동결'에 대한 입장을 묻는 변수였다.

자료에서 소속 정당을 나타내는 변수의 범주는 2개이고, 나머지 16개의 변수는 모두 범주의 개수가 3개이다. 범주의 수를 고려한 초기 모드 선택방법에서는 범주의 수가 가장 적은 변수인 소속 정당을 나타내는 변수로 자료를 정렬하고 나머지 16개의 변수는 순서대로 다중 정렬한 후 소속 정당별로 객체를 랜덤하게 선택하여 초기 모드로 사용하였다.

이러한 과정을 100번 반복 실행하여 군집의 오분류률(misclassification rate)을 구한 후, 이에 대한 평균과 표준편차를 구하여 <표 4.1>을 얻었다. 여기서 오분류률은 군집화 결과 형성된 각 군집에서 정당별로 객체의 수를 구하여 적은 정당에 속하는 객체가 오분류되었다고 판단하였다.

<표 4.1> K-모드 알고리즘의 오분류율 (괄호안은 표준편차)

군집의 수	임의로 선택	계통 추출	범주의 수 고려
2	14.13% (3.62%)	13.59% (0.76%)	13.25% (0.67%)
3	12.51% (2.64%)	11.34% (2.48%)	11.32% (2.16%)
4	11.60% (2.19%)	10.76% (2.39%)	10.39% (2.03%)

전체적인 경향을 보면, 군집의 수가 많을수록 오분류률은 작아지는 경향이 있다. 군집의 수가 같으면 초기 모드 선택 방법에 따라 오분류율이 다르게 나타났는데, 대체로 계통 추출한 방법과 범주의 수를 고려한 방법이 임의로 선택하는 방법보다 군집의 오분류률이 작았다.

4.2. ROCK 알고리즘

락 알고리즘에서는 θ값은 두 객체간의 이웃여부를 판단하는 중요한 값이며 θ의 값에 따라 이웃 여부가 달라진다. 미국 의회의 선거자료에서는 16개의 변수가 있으므로, 예를 들어 16개의 변수 중 적어도 11개 이상이 일치할 때 이웃으로 하고 싶으면 $\theta = (16-5)/(16+5) = 0.52$로 하여야 한다. 조건을 더 강화하여 16개의 변수 중 적어도 12개 이상이 일치할 때 서로 이웃이 되게 하려면 $\theta = 0.6$ 으로 하면 된다. 이런 방법으로 여러 개의 θ 값을 구한 후, 미국의회의 선거자료에 락 알고리즘을 적용하여 <표 4.2>를 얻었다.

<표 4.2> 락 알고리즘에 대한 모의실험 결과

θ 값	군집 번호	ROCK 공화당원	ROCK 민주당원	ROCK 오분류률	ROCK 남은 객체	사후할당 공화당원	사후할당 민주당원	사후할당 오분류률
0.52	1	156	42	11.82 %	12	159	43	11.95 %
	2	8	217			9	224	
0.60	1	156	40	11.06 %	19	160	42	11.49 %
	2	6	214			8	225	
0.68	1	149	30	9.30 %	48	159	45	12.41 %
	2	6	202			9	222	
0.77	1	136	16	6.33 %	119	162	50	12.87 %
	2	4	160			6	217	

미국 의회 선거자료에서 락 알고리즘은 주어진 4개의 θ값에서 모두 2개의 거대 군집을 형성하였으며, 동시에 군집에 형성하지 못하는 객체를 생성하였다. θ값이 커질수록 남은 객체를 고려하지 않은 군집의 오분류률은 작아지나 군집을 형성하지 못하고 남은 객체의 수는 증가한다. $\theta = 0.77$일 때 2개의 군집을 형성한 후 군집화의 오분류률은 6.33%로 가장 낮지만, 군집을 형성하지 못한 객체는 435개 중 119개(27.4%)로 가장 많았다. 따라서 오분류율이 낮더라도 전체 자료 중 27%를 무시할 수는 없기 때문에 $\theta = 0.77$ 인 경우의 락 알고리즘은 사후적으로 수정·보완이 필요함을 시사한다.

앞 절에서 설명한대로 군집을 형성하지 못하고 남은 객체들을 이미 형성된 군집에 사후 할당을 하여 군집을 갱신하고 오분류률을 계산하였다. 그 결과, θ값이 커지면서 오분류률이 작아지다가 다시 커지는데 경향이 발견되는데, 그 이유는 값이 커지면 군집을 형성하지 못한 객체의 수가 많아지기 때문에 사후 할당을 하면서 오분류률이 증가하기 때문인 것으로 해석된다. $\theta = 0.6$일 때, 즉 16개의 변수 중 12개 이상의 변수 값이 일치하면 서로 이웃이 되게 할 때 오분류률(11.5%)이 가장 작았다.

락 알고리즘은 $\theta = 0.52, ..., 0.77$ 등 4가지 경우에 모두 거대 군집의 수가 2인 경우로 나타났으므로 K-모드 알고리즘과의 비교는 군집의 수가 2인 경우에 의미가 있다. 오분류률을 비교하면, K-모드 알고리즘은 범주의 수를 고려한 초기 모드 선택 방법의 오분류률이 13.25%로 가장 작았고 락 알고리즘은 인 경우에 사후 할당한 오분류률이 11.49%로 가장 작았다. 즉, 미국 의회 선거 자료에서 군집의 수가 2인 경우 락 알고리즘의 효율이 K-모드 알고리즘의 그것보다 우수하다고 할 수 있다. 그러나 K-모드 알고리즘에서 군집의 수가 2 보다 큰 경우의 오분류률은 오히려 락 알고리즘의 오분류률 보다 작게 나타나 군집의 수에 따라 두 알고리즘의 효율이 달라진다고 할 수 있다.

5. 결론

전통적인 군집화 방법은 주로 연속형 자료를 대상으로 하였다. 그런데 최근의 대용량 자료에는 연속형 자료뿐 아니라 범주형 자료도 다수 포함되므로, 이러한 자료에 군집화 방법을 적용하기 위해서는 전통적인 군집화 방법 외에 범주형 자료를 대상으로 하는 군집화 방법이 필요하다.

대용량 범주형 자료를 대상으로 하는 군집화 방법으로는 K-모드 알고리즘과 락 알고리즘 등이 있다. 이 연구에서는 K-모드 알고리즘과 락 알고리즘을 고찰하였고, 이 두 알고리즘의 단점을 보안하는 개선된 알고리즘을 제안하였다. 또한 실제 자료에 두 알고리즘을 적용하여 군집분석의 결과를 비교하였다.

K-모드 알고리즘은 초기 모드 값에 따라 군집의 결과가 달라질 수 있다는 것이 가장 큰 단점이었는데 자료의 특성을 고려한 초기 모드 선택 방법인 계통추출 방법과 범주의 수를 고려한 방법을 K-모드 알고리즘에 적용한 결과 군집의 효율을 향상시킬 수 있음을 모의실험을 통하여 부분적으로 보였다. 락 알고리즘은 링크라는 개념을 사용하여 군집간 유사성을 정의하고 유사성이 가장 큰 군집들을 단계적으로 병합하는 과정을 반복하는 것으로, 거대 군집이 형성된 후에도 군집을 형성하지 못하고 남는 객체가 있을 수 있다는 단점이 있었다. 이러한 단점은 남은 객체들을 이미 만들어진 군집 중에서 유사성이 가장 큰 군집으로 사후할당 함으로써 극복될 수 있음을 보였다.

이 연구에서는 범주형 변수만으로 이루어진 자료에 대한 군집화 방법을 살펴보았다. 그러나 데이터 마이닝에서 다루는 대용량 자료에는 연속형 변수와 범주형 변수가 함께 있는 경우가 더 일반적이다. 따라서 향후 연속형 변수와 범주형 변수가 함께 있는 경우에 대한 군집화 방법의 연구가 더 필요할 것으로 생각된다.

참고문헌

[1] 허명회 (2000). 이중 K-평균 군집화. <응용통계연구>, 13, 343-352.

[2] Anderberg, M.R. (1973). *Cluster analysis for applications*, Academic Press.

[3] Bezdek, J.C. (1980). A convergence theorem for the fuzzy ISODATA clustering algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(8), 1-8.

[4] Goodman, L. A. and Kruskal, W. (1954). Measures of association for cross classifications. *Journal of the American Statistical Association*, 49, 732-764.

[5] Gordon, A.D. (1999). *Classification*. Chapman and Hall.

[6] Guha, S. Rastogi, R. and Shim, K. (1997). A clustering algorithm for categorical attributes. Technical report, Bell Laboratories, Murray Hill.

[7] Guha, S. Rastogi, R. and Shim, K. (1999). Rock: a robust clustering algorithm for categorical attributes. *Proceedings of the IEEE International Conference on Data Engineering*, Sydney.

[8] Huang, Z. (1997a). Clustering large data sets with mixed numeric and categorical values. *Proceedings of the first pacific-asia conference on KDD*, World Scientific.

[9] Huang, Z. (1997b). A fast clustering algorithm to cluster very large categorical data sets in data mining. *Workshop on research issues on data mining and knowledge discovery.*

[10] Jain, A.K. and Dubes, R.C. (1988). *Algorithms for clustering data.* Prentice Hall, Englewood Cliffs, New Jersey.

[11] Jhun, Myounshic and Jin, Seohoon (2000). On a modified k-spatial medians clustering. *Journal of the Korean Statistical Society*, 29, 247-260.

[12] MacQueen, J.B. (1967). Some methods for classification and analysis of multi -variate observations. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 281-297.

[13] Murtagh, F. (1992). Comments on "Parallel algorithms for hierarchical clustering and cluster validity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(10), 1056-1057.

[14] Selim, S. Z. and Ismail, M. A. (1984). K-means-type algorithms : a generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(1), 81-87.

[15] Spath, H. (1980). *Clustering analysis algorithms for data reduction and classification of*

objects, John Wiley and Sons.

[16] UCI Machine Learning Repository Content Summary, http:/ /www.ics.uci.edu /~mlearn /MLSummary.html.

[2002년 1월 접수, 2002년 7월 채택]

Improvements of K-modes Algorithm and ROCK Algorithm

Bo-Hwa Kim [1] Kyuseong Kim [2]

ABSTRACT

K-modes algorithm and ROCK(RObust Clustering using linKs) algorithm are useful clustering methods for large categorical data. In the paper, we investigate these algorithms and propose improved algorithms of them to correct their weakness. A simulation study shows that the proposed algorithms could increase the performance of data clustering.

Keywords: categorical data, data clustering, measure of association, posterior allocation, systematic selection.

1) Korea Development Bank, 16-3 Youido-dong, Yongdeungpo-ku, Seoul 150-973, Korea
2) Associate Professor, Dept. of Computer Science and Statistics, Univ. of Seoul, Seoul 130-743, Korea
E-mail : kskim@uos.ac.kr