碩士學位論文

초기 모드 결정 방식을 개선한 K-mode 알고리즘

濟州大學校 大學院

電算統計學科

梁 舜 喆

2006年 12月

초기 모드 결정 방식을 개선한 K-mode 알고리즘

指導敎授 金 鐵 洙

梁 舜 喆

이 論文을 理學 碩士學位 論文으로 提出함

2006年 12月

梁舜喆 의 理學 碩士學位 論文을 認准함

審査委員長 _____

委 員 _____

委 員 _____

濟州大學校 大學院

2006年 12月

K-mode Algorithm Improving Initial Mode Decision Methodologies

Soon-Cheol Yang

(Supervised by professor Chul Soo Kim)

A thesis submitted in partial fulfillment of the requirement for the degree of Master of Science

Department of Computer Science and Statistics

Graduate School

Cheju National University

December 2006

목 차

List of Figures

List of Tables

Abstract

Data mining is the process of uncovering previously unknown patterns and relationships in large databases using sophisticated statistical analysis and modeling techniques such as classification, association rule mining, clustering, etc.. Spacially, clustering is an important data mining problem. Clustering, in data mining, is useful for discovering groups and identifying interesting distributions in underlying data. The *k*-means algorithm is well known for its efficiency in clustering large data sets. However, working only on numeric values prohibits it from being used to cluster real world data containing categorical values. Huang presented an algorithm, called K-mode algorithm, to extend the K-means paradigm to categorical domains(1997). K-mode algorithm suffers from initial staring conditions effect (initial mode, the number of initial mode).

This paper improved the problem of K-mode algorithm using Max-Min method that is a kind of methods to decide initial values in K-means algorithm. We introduce new similarity measures to deal with categorical data sets using means of cluster. Tested with the Mushroom data sets and Small Soybean data sets the proposed algorithm has shown a good performance for the two aspects (accuracy, run time).

Ⅰ. 서론

최근 컴퓨터의 하드웨어의 발전과 더불어 인터넷이 빠른 속도로 발전하면서 데이터의 양적 팽창이 이루어졌고 이에 따라 데이터를 유용한 정보와 지식으로 바꿔야 하는 시대에 도달하였다. 이러한 시대의 흐름에 따라 수많은 데이터에서 정보와 지식을 추출하는 데이터 마이닝(data mining) 분야가 각광을 받고 있다.

데이터 마이닝은 대용량 데이터에서 의미 있는 규칙이나 패턴을 찾기 위한 데이터 탐색 및 분석과정이라고 할 수 있고, 데이터 마이닝은 분류(classification), 예측(prediction), 유사 그룹 및 연관규칙(association rule), 군집화(clustering), 그리고 의사결정나무(decision tree) 등을 비롯한 다양한 기법들이 사용될 수 있으며, 마케팅을 비롯한 제품 개발, 프로세스 개선, 의사 결정 시스템 등에 폭넓게 활용된다.

데이터 마이닝에서 사용되는 기법 중 하나인 군집화(clustering)는 주어진 관찰치 중에서 유사한 것들을 몇몇의 집단으로 그룹화하여, 각 집단의 성격을 파악함으로써 데이터 전체의 구조에 대한 이해를 돕고자 하는 방법이다. 대용량 데이터에서 개개의 관찰치를 요약하는 것보다는 전체를 유사한 관찰치들의 군집으로 구분하여 복잡한 전체보다는 그것을 잘 대표하는 군집들을 관찰함으로써 전체 데이터에 대한 의미 있는 정보를 얻어낼 수 있다. 데이터 마이닝 기법 중 분류(classification)기법은 미리 정의된 집합으로 데이터를 구분하는 것에 반해, 군집화는 유사도에 기반한 데이터를 구분하는 기법으로 데이터가 갖는 속성값에 따라 각 그룹의 범위나 성격이 다르게 정의될 수 있다.

군집화 방법은 크게 계층적 군집화(hierarchical clustering)방법과 비계층적 군집화(non-hierarchical clustering)방법으로 구분할 수 있다. 계층적 군집화 방법은 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. 비계층적 군집화 방법은 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.

기존의 데이터 군집화 방법들은 수치형 데이터(numerical data)를 대상으로 개발되어 왔다. 대표적인 비계층적 군집화 방법의 하나인 K-means 군집화 방법은 객체간의 거리를 유클리디안 거리(Euclidean distance)로 정의한 후 군집의 평균을 계산하여 비용함수를 최소화하도록 군집을 형성해 나가는 방법이다(MacQueen, 1967). 이 방법은 간단하고 구현이 쉬울 뿐만 아니라 알고리즘이 수렴하기 때문에 전통적으로 널리 이용되어 왔다. 그러나 K-means 방법은 군집의 평균이 정의되어있을 때에만 적용할 수 있다. 즉, 수치형 데이터일 경우에만 적용할 수가 있다. 대용량 데이터에서는 수치형 데이터와 범주형 데이터(categorical data)가 섞여 있는 경우가 대부분이기 때문에 대용량 데이터를 군집화하기 위해서는 범주형 데이터를 대상으로 하는 군집화 방법의 연구가 필수적이다.

범주형 데이터를 대상으로 하는 대표적인 군집화 방법은 K-mode 방법(Huang, 1997)과 ROCK(RObust Clsutering using linKs. ROCK, Guha 외 2인, 1999) 방법이 있다. K-mode 방법은 K-means 방법의 형식을 유지하면서 범주형 데이터에 적합하도록 제안된 방법이고 ROCK 방법은 객체간의 유사도를 정의한 후 데이터의 모든 객체를 동시에 비교하여 유사도가 가장 큰 객체들을 순차적으로 병합해 가는 계층적 방법이다.

이 논문에서는 특히 K-mode 방법에 대해서 자세히 살펴보고 개선된 K-mode 방법을 제안한다.

이 논문은 다음과 같이 구성되어 있다. 제 2장에서는 데이터 마이닝의 개념과 기법에 대해서 살펴보고, 제 3장에서는 기존에 제안된 군집화 알고리즘에 대해서 살펴보고, 제 4장에서는 초기 모드 결정 방식을 개선한 K-mode 방법을 제안한다. 제 5장에서는 제안된 방법을 실제 데이터에 적용하여 실험을 실시하고 결과를 해석하며 마지막으로 제 6장에서는 결론과 더불어 향후 연구 과제를 제시한다.

Ⅱ. 데이터 마이닝

1. 데이터 마이닝의 개념

데이터 마이닝이란 간단히 말하면 대량의 데이터로부터 지식을 추출하는(또는 캐내는) 것을 말한다. 즉, 대용량 데이터에서 의미 있는 규칙이나 패턴을 찾기 위한 데이터 탐색 및 분석과정이라 할 수 있다. 또한 많은 사람들이 '데이터베이스에서의 지식 발견'(Knowledge Discovery in Database : KDD)과 동의어로 취급하고 있다. 어떤 사람은 데이터 마이닝을 단순히 KDD의 필수적 절차쯤으로 여긴다. 그림 1은 지식발견 절차를 나타내고 있으며, KDD는 다음과 같은 단계들의 반복적 연속으로 이루어져 있다.

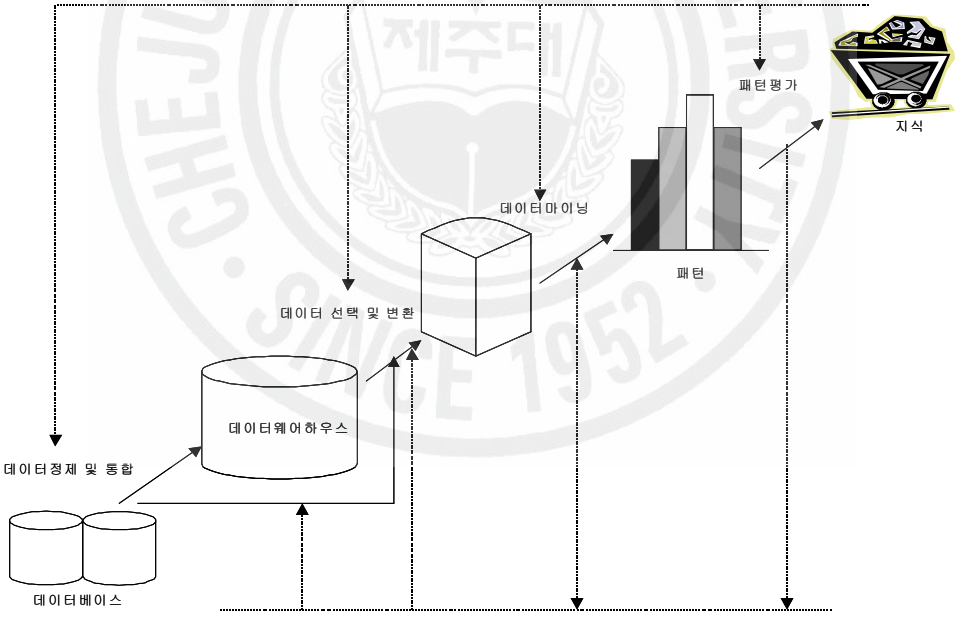

Figure 1. Data mining concept of the KDD

1. 데이터 정제 : 잡음과 불일치 데이터 제거
2. 데이터 통합 : 다수의 데이터 소스들의 결합
3. 데이터 선택 : 분석 작업과 관련된 데이터들이 데이터베이스로부터 검색

4. 데이터 변환 : 요약이나 집계 등과 같은 연산을 수행함으로써, 마이닝을 위해 적합한 형태로 데이터를 변환하거나 합병한다.
5. 데이터 마이닝 : 데이터 패턴을 추출하기 위하여 지능적 방법들이 적용되는 필수적 과정
6. 패턴 평가 : 몇 가지 흥미 척도들을 기초로, 지식을 나타내는 진짜 흥미로운 패턴들을 구별한다.
7. 지식 표현 : 사용자에게 채굴된 지식을 보여주기 위하여 시각화와 지식 표현 기법들이 사용된다.

데이터 마이닝 단계에서는 사용자나 지식 베이스와 상호작용이 가능하다. 흥미로운 패턴들이 사용자에게 제시된 후, 새로운 지식으로서 지식 베이스에 저장될 수 있다. 이러한 견해에 따르면, 데이터 마이닝은 평가를 위해 숨겨진 패턴을 찾아내는 필수적인 단계이면서도 전체 과정에서 보면 하나의 단계에 불과하다. 그러나 데이터 마이닝 기법이 발전하면 새로운 지식창출이 용이해지므로 연구가 필요하다고 할 수 있다.

2. 데이터 마이닝 기법

1) 연관성 분석(Association Analysis)

연관성 분석(Association Analysis)은 주어진 데이터의 집합에서 함께 빈번하게 발생하는 속성값(attribute-value) 조건들을 나타내는 연관 규칙(association rule)들을 발견하는 것이다. 계속해서 많은 데이터들이 수집되고 저장되어 오는 동안 산업분야에서는 데이터베이스 내의 연관 규칙을 발견하는 것이 유용하다는 것을 알게 되었다. 대규모 비즈니스 트랜잭션 데이터들 사이에서 흥미 있는 연관관계의 발견은 카탈로그 디자인, 교차 마케팅, 손실 원인 분석 등의 비즈니스 의

사결정 프로세스에 많은 도움이 된다.

연관성 분석의 가장 전형적인 활용 중에 하나는 장바구니 분석(market basket analysis)이다. 이 프로세스는 고객들의 장바구니에서 서로 다른 품목들 사이의 연관관계를 발견함으로써 고객의 구매습관을 분석한다. 이러한 연관규칙은 고객들이 빈번하게 함께 구매한 품목들에 대한 직관을 갖게 함으로써 소매상들이 마케팅 전략을 세우는데 많은 도움을 준다.

연관 규칙은 $X \Rightarrow Y$ 형식을 갖는다. 다시 말하면, $A_i(for \quad i \in \{1,\cdots,m\})$ 와 $B_j(for \ j \in \{1,\cdots,n\})$ 는 속성값이 쌍일 때, " $A_1 \wedge \cdots \wedge A_m \rightarrow B_1 \wedge \quad \cdots \wedge B_n$ " 형식을 갖는다. 연관 규칙 $X \Rightarrow Y$ 의 의미는 "X 에 있는 조건들을 만족시키는 데이터베이스 투플은 Y 에 있는 조건들도 만족시킬 가능성이 있다."로 해석한다.

2) 의사결정나무분석(Decision Tree Analysis)

의사결정나무(decision tree)는 의사결정규칙(decision rule)을 그림 2와 같은 나무구조로 도표화하여 분류(classification)와 예측(prediction)을 수행하는 분석방법이다. 이 방법은 분류 또는 예측의 과정이 나무구조에 의한 추론규칙(induction rule)에 의해서 표현되기 때문에, 신경망이나 판별분석, 회귀분석 등에 비해서 연구자가 그 과정을 쉽게 이해하고 설명할 수 있다는 장점을 가지고 있다.

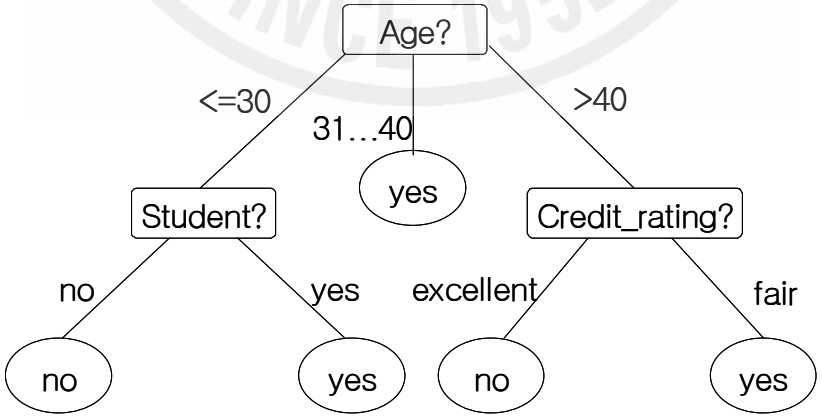

Figure 2. example of decision tree

의사결정나무분석을 위해서 CHAID(Kass. 1980), CART(Breiman et al. 1984),

C4.5(Quinlan, 1993)와 같은 다양한 알고리즘이 제안되어 있으며, 최근에는 이들의 장점을 결합하여 보다 개선된 알고리즘들이 제안되고 상용화되고 있다.

일반적으로 의사결정나무분석은 다음과 같은 단계를 거치게 된다.

1. 의사결정나무의 형성 : 분석의 목적과 자료구주에 따라서 적절한 분리기준(split criterion)과 정지규칙(stopping rule)을 지정하여 의사결정나무를 만든다.

2. 가지치기 : 분류오류(classification error)를 크게 할 위험(risk)이 높거나 부적절한 추론규칙(induction rule)을 가지고 있는 가지(branch)를 제거한다.

3. 타당성평가 : 이익도표(gains chart)나 위험도표(risk chart) 또는 검증용 자료(test data)에 의한 교차타당성(cross validation) 등을 이용하여 의사결정나무를 평가한다.

4. 해석 및 예측 : 의사결정나무를 해석하고 예측모형을 설정한다.

3) 신경망(Neural Networks)

신경망 또는 인공신경망(artificial neural networks)에 관한 연구는 뇌 신경생리학으로부터 영감을 얻어 시작되었다. 자료분석 분야에서 신경망은 복잡한 구조를 가진 자료에서의 예측(prediction) 문제를 해결하기 위해서 사용되는 유연한 비선형모형(nonlinear models)의 하나로 분류될 수 있다. 그러나 신경생리학과의 유사성 때문에 일반적으로 다른 통계적 예측모형에 비해 보다 흥미롭게 받아들여지고 있다. 신경망은 은닉마디(hidden units)라고 불리는 독특한 구성요소에 의해서 일반적인 통계모형과 구별되어진다. 은닉마디는 인간의 신경세포를 모형화한 것으로써, 각 은닉마디는 입력변수들의 결합을 수신하여 목표변수에 전달한다. 이때 결합에 사용되는 계수들을 연결강도(synaptic weights)라고 부르며, 활성함수는 입력값을 변화하고 이를 입력으로 사용하는 다른 마디로 출력하게 된다.

신경망은 많은 입력변수를 가지고 있으며 입력변수들과 목표변수 간의 관계가 복잡한 비선형 형태를 가질 때 유용하다. 그러나 목적함수를 최적화하는 계수값을 찾는 것은 매우 어려운 작업이고 은닉층과 은닉마디의 개수를 정하기가 쉽지 않으며, 은닉층과 은닉마디가 많으면 많을수록 신경망은 더욱 복잡해지는 단점이 있다.

신경망에는 여러 가지 다양한 모형이 있으나, 그 중에서도 자료분석을 위해 가장 널리 사용되는 모형은 MLP(multi-layer perceptron) 신경망이다. MLP는 입력층(input layer), 은닉마디로 구성된 은닉층(hidden layer), 그리고 출력층(output layer)으로 구성된 전방향(feed-forward) 신경망이다. 그림 3은 입력층, 은닉층, 그리고 출력층으로 이루어진 MLP 신경망의 구조이다.

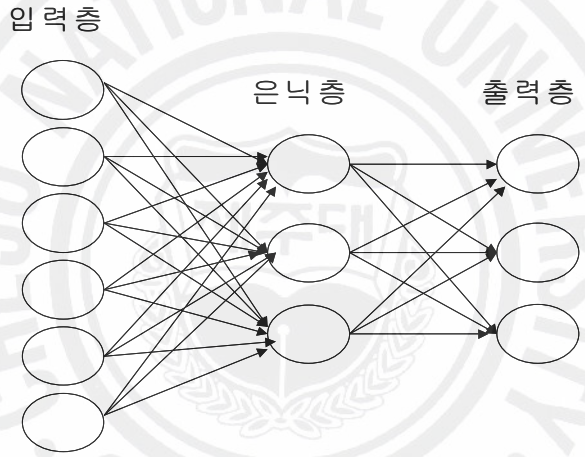

Figure 3. MLP Structure

Ⅲ. 관련 알고리즘 연구

1. K-means 알고리즘

Macqueen(1967)이 제안한 K-means 알고리즘은 수치형 데이터를 군집화하는 비계층적 방법이다.

K-means 알고리즘의 첫 단계에서는 데이터에서 k 개의 관측값을 랜덤하게 선택하여 초기값으로 결정하여 초기 군집의 중심을 이루게 한다. 다음 단계에서는 초기값으로 설정되지 않은 관측값들을 가장 가까운 초기값이 중심이 군집에 배정하여 초기 군집을 형성한다. 형성된 군집의 중심을 다시 계산하여 관측값들과의 거리를 계산한 다음, 가장 가까운 중심의 군집으로 관측값을 이동하여 새로운 중심을 구하고, 군집의 중심의 이동이 없을 때까지 이 과정을 반복한다.

알고리즘은 다음에 요약되어 있다.

1. 주어진 자료에서 랜덤하게 k 개의 초기값 $(s_1, s_2, \cdots, s_k)$ 을 선택한다.

2. 각 관측값 (x_i) 에 대해 초기값 (s_1) 까지의 거리를 계산한다.

$$d_j = \|x_i - s_j\|,\ j = 1, \cdots, k,\ i = 1, \cdots, n$$

3. 각 x_i 를 단계 2에서 계산된 k 개의 d_j 중 가장 작은 d_j 를 주는 초기값 쪽의 군집으로 할당한다.

4. 군집의 중심을 다시 계산한다.

5. 군집의 중심이 변화가 없을 때까지 단계 3, 4를 반복한다.

Figure 4. K-means algorithm

2. K-means 알고리즘에서 초기값 결정 방법

1) KA(Kaufman Approach) 방법

KA방법은 Kaufman과 Rousseeuw(1990)가 제안한 방법으로서 Macqueen이 제안한 방법에 의해서 랜덤하게 초기값을 선택했을 때 발생할 수 있는 문제점을 해결하고 가능하면 형성될 군집의 내부에서 초기 군집의 중심이 선택되도록 초기값을 단계적으로 설정해 나간다.

이 방법의 첫 단계에서는 주어진 자료의 가장 중앙에 위치한 관측값을 첫 번째 초기값으로 선택하고 첫 번째 초기값을 제외한 나머지 모든 관측값에 대해서 초기값과 관측값과의 거리를 계산한다. 또 관측값과 자신의 주변에 있는 관측값과의 거리를 고려하도록 기준을 정해 선택된 초기값과 일정한 거리에서 떨어져 있으면서, 주변에 관측값들이 모여 있는 영역에서 다음 초기값을 설정하게 한다. 이 과정은 k 개의 초기값이 단계적으로 모두 선택될 때까지 반복하게 된다.

다음은 KA방법에 대한 요약이다.

1. 주어진 데이터에서 가장 중앙에 위치한 값을 첫 번째 초기값 s_1 으로 선택한다.

2. 나머지 모든 자료 $x_i (x_i \neq s_1,\ i \neq j),\ j = 1, \cdots, n$ 에 대하여,

   a. 초기값으로 선택되지 않은 관측값 $x_i (x_j \neq s_1,\ i \neq j), j = 1, \cdots, n$ 에 대하여 $C_{ji} = max(D_j - d_{ji},\ 0)$ 을 계산한다.

   여기서 $d_{ji} = \| x_i - x_j \|$, $D_j = min(d_{sj})$ (x_s 는 선택된 초기값)

   b. x_i 에 대해서 $\sum_j C_{ji}$ 를 계산한다.

3. $\sum_j C_{ji}$ 를 최대화하는 x_i 를 두 번째 초기값으로 선택한다.

4. k 개의 초기값이 모두 선택될 때까지 단계 2, 3을 반복한다.

Figure 5. KA process

2) Max-Min 방법

Bae(2005)년에 제안한 방법으로 Macqueen 방법의 초기값을 랜덤하게 선택하였을 때 생기는 문제점을 해결하기 위하여 제안된 KA방법이 정교하지만 자료가 많아짐에 따라 초기값 설정에 따른 시간이 많이 걸리는 단점을 보완하기 위한 방법으로 단계적으로 초기값을 선택하되 선택된 초기값들이 다음 초기값을 정하는 데 정보를 줄 수 있도록 하고, KA방법처럼 많은 계산을 하지 않도록 고안했다.

Max-Min방법에서는 우선 자료에서 랜덤하게 하나의 관측값을 선택하여 첫 번째 초기값으로 선택하고, 첫 번째 초기값에서 나머지 관측값과의 거리를 구하여 그 거리를 최대로 하는 관측값을 두 번째 초기값으로 선택한다. 즉, 초기값을 선택함에 있어 초기값들이 한 곳에 모이는 현상을 방지하기 위해 처음 두 초기값은 머리 있는 것을 택하게 한다.

다음 단계의 초기값을 구하기 위해서 초기값에 선택되지 않은 나머지 관측값들에 대하여 첫 번째 초기값과의 거리와 두 번째 초기값과의 거리를 구하면 각 관측값에 대해 두 종류의 계산된 거리가 얻어진다. 이 두 거리 중, 최소값을 선택하여 각 관측값에 대해 두 초기값과의 거리로 대응하게 한다. 각 관측값에 대응되어 있는 관측값과 두 초기값과의 거리를 비교하여 이 값을 최대로 하는 관측값을 구하여 세 번째 초기값으로 선택함으로써 초기값들이 적절하게 떨어져 선택되도록 하였다.

이 다음 단계의 초기값을 선택하기 위해서는 이전 단계까지 초기값으로 선택되지 않은 관측값에 대하여 위의 과정을 반복적으로 시행하여 k 개의 초기값이 모두 선택될 때까지 계속 실시한다.

Max-Min 방법에 대한 알고리즘은 다음에 설명되어 있다.

1. 자료에서 랜덤하게 하나의 관측값을 선택하여 첫 번째 초기값 s_1 을 결정한다.

2. 나머지 관측값, $x_i(x_i \neq s_1)$, $i=1,\cdots,n$ 에 대하여 첫 번째 초기값 (s_1)과의 거리를 최대로 하는 관측값을 두 번째 초기값 s_2 로 선택한다.

3. 다음 단계의 초기값 s_m , $m=3,\cdots,k$ 를 구하기 위해서 이전 단계에서 구해진 초기값을 추가하면서 다음 단계를 반복하여 k 개의 초기값들을 선택한다.

나머지 관측값 $x_i(x_i \neq s_l,\ l=1,\cdots,m-1),\ i=1,\cdots,n$ 에 대하여

a. 이미 구해진 초기값들과 x_j 와의 거리를 각각 구하여 이들의 최소값 sd_i 을 계산하여 대응시킨다.

$x_i <- sd_i = min\{\|x_i - s_1\|, \cdots, \|x_i - s_{m-1}\|\},\ i=1,\cdots,n\ (x_i \neq s_l,\ l=1,m-1)$

b. 각 관측값 x_i 에 대응하는 sd_i 를 비교하여 이들의 값을 최대로 하는 관측값을 다음 초기값으로 선택한다.

$$s_m = x_p \leftarrow max_{1 \leq i \leq n}(sd_i) = sd_p$$

4. k 개의 초기값이 모두 선택될 때까지 단계 3을 반복한다.

Figure 6. Max-Min process

3. ROCK 알고리즘

Guha 외 2인(1999)이 제안한 ROCK 알고리즘은 군집간의 링크(links)값을 이용하여 군집을 순차적으로 병합해 나가는 계층적 군집방법이다.

우선 두 객체 x_i, x_j의 유사도(similarity measure)을 다음과 같이 정의한다.

집합 $X = x_1, \cdots, x_n$는 n 개의 객체로 구성되어 있고, 각 객체는 m 개의 범주형 변수 값을 갖는다고 하자.

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

그리고 j 번째 범주형 변수 A_j 는 l_j 개의 수준을 가지며 각 수준을 $c_{j1}, \cdots, c_{jl_j}$ 라고 하자.

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

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

$$\delta(a, b) = \begin{cases} 0, & if \quad a = b \\ 1, & if \quad a \neq b \end{cases} \tag{2}$$

유사도 $sim(\mathrm{x}_i, \mathrm{x}_j)$은 0과 1사이의 값을 가지며, 두 객체가 유사할수록 큰 값을 갖는다. 두 객체간의 유사도가 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.

$$G(\mathrm{x}_i) = \{ \mathrm{x}_i \mid sim\ (\mathrm{x}_i, \mathrm{x}_j) \geq \theta \quad (\ i \neq j\ ) \} \tag{3}$$

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

$$link(C_i, C_j) = \sum_{\mathrm{x}_p \in C_i, \mathrm{x}_q \in C_j} link(\mathrm{x}_p, \mathrm{x}_q) \tag{4}$$

즉, 링크의 값이 클수록 두 객체나 군집은 유사한 것으로 볼 수 있다. 군집의 병합을 위해서 군집간의 링크 값, 군집에 속하는 객체 수 및 유사도의 기준값

θ 를 고려한 $g(C_i, C_j)$ 을 계산한 후, $g(C_i, C_j)$ 의 값이 가장 큰 두 군집을 병합한다.

$$g(C_i, C_j) = \frac{link(C_i, C_j)}{(n_i + n_j)^{1+2f(\theta)} - n_i^{1+2f(\theta)} - n_j^{1+2f(\theta)}} \quad (5)$$

여기서, n_i 는 군집 C_i 에 속하는 객체의 수이고, $f(\theta) = (1-\theta)/(1+\theta)$ 이다. 군집의 병합기준 g 는 θ 의 함수이고, θ 값에 따라 병합되는 군집의 수가 달라지기 때문에, 실제 데이터에서는 θ 의 값을 자료의 특성에 따라 정해 주어야 한다.

ROCK 알고리즘의 단계는 다음과 같다.

1. 주어진 θ 에 대하여 객체간의 유사도를 계산하여 객체별로 이웃과 이웃군을 구한다.

2. 군집간의 링크값 $link(C_i, C_j)$ 을 계산한 후, $g(C_i, C_j)$ 을 구한다.

3. $g(C_i, C_j)$ 값이 가장 큰 두 군집을 병합하고 군집간의 링크 값을 갱신한다. 이 때, 병합된 군집과 다른 군집간의 링크 값은 병합되기 전의 링크 값의 합으로 계산한다.

4. 군집의 개수가 일정 수에 이를 때까지 2와 3의 과정을 반복 수행한다.

Figure 7. ROCK algorithm

ROCK 알고리즘은 링크라는 개념을 이용하여 두 객체 뿐 만 아니라 자료의 모든 객체를 동시 비교하여 유사한가를 판단한다. 그리고 어떤 군집을 먼저 병합할 것인가를 판단할 때 군집의 크기까지 고려하기 때문에 상대적으로 크기가 작은 군집도 제대로 유지할 수 있다는 장점이 있다. 그러나 병합하는 과정에서 거대 군집과 군집을 형성하지 못하고 남아 있는 객체가 존재 할 수 있으며, 병합을

계속 할 경우 남은 객체가 거대 군집에 포함되는 것이 아니라 거대 군집들이 서로 병합되는 일이 생길 수 있고 기준값(θ)에 따라 군집의 수와 크기가 달라지기 때문에 최적의 기준값을 정해야 하나 현재로서는 경험적으로 정하는 방법이외에는 존재하지 않는다.

4. K-mode 알고리즘

Huang(1997)이 제안한 K-mode 알고리즘은 K-means 알고리즘을 범주형 속성 영역으로 확장한 알고리즘이다. 알고리즘의 기본 전개 단계는 K-means 알고리즘과 동일하며 다음의 요소들을 범주형 객체의 처리에 맞게 변형하였다.

ROCK 알고리즘과 마찬가지로, 집합 $\mathrm{X}=\mathrm{x}_1,\cdots,\mathrm{x}_n$는 n 개의 객체로 구성되어 있고, 각 객체는 m 개의 범주형 변수 값을 갖는다고 하자.

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

Huang은 범주형 변수의 값들로 구성된 두 객체 $\mathrm{x}_{i_1},\mathrm{x}_{i_2}$의 비유사도(dissimilarity measure)를 객체간에 일치하지 않는 변수의 수인 로 정의하였다.

$$d(\mathrm{x}_{i_1},\mathrm{x}_{i_2})=\sum_{j=1}^{m}\delta(x_{i_1 j},x_{i_2 j}) \tag{6}$$

여기서 $\delta(a,b)$ 는 앞서 정의한 식(2)과 동일하다.

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

$$D(\mathrm{q},\mathrm{X})=\sum_{i=1}^{n}d(\mathrm{x}_i,\mathrm{q}) \tag{7}$$

집합 X 의 모드 q^{*} 는 다음과 같이 찾을 수 있다. 집합 X 에서 변수 A_j 가 수준 c_{jk} 를 갖는 빈도수를 n_{jk} 라고 하면, A_j 가 c_{jk} 를 가질 상대 빈도는

$$f(A_j = c_{jk} \mid \mathrm{X}) \geq \frac{n_{jk}}{n},\ k = 1, \cdots, l_j \tag{8}$$

가 된다. 그러면 모든 $j(=1,\cdots,m)$ 에 대하여

$$f(A_j = c_{jk} \mid \mathrm{X}) \geq f(A_j = c_{jk} \mid \mathrm{X}) \tag{9}$$

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

K-mode 알고리즘의 단계는 다음과 같다.

1. K 개의 군집의 초기 모드 $\{\mathrm{q}_1^0,\cdots,\mathrm{q}_\mathrm{K}^0\}$ 를 선택한다.

2. 객체 n 개와 초기모드 $\{\mathrm{q}_1^0,\cdots,\mathrm{q}_\mathrm{K}^0\}$ 의 비유사도를 계산하여 비유사도가 가장 적은 군집으로 객체를 할당한 후, K 개 군집내의 모드를 갱신하여 갱신된 첫 번째 모드 $\{\mathrm{q}_1^1,\cdots,\mathrm{q}_\mathrm{K}^1\}$ 를 얻는다.

3. 모든 객체와 갱신된 모드의 비유사도를 다시 구한 후, 만일 다른 군집의 모드와의 비유사도가 더 적으면 해당 객체를 그 군집으로 다시 할당하고 군집내의 모드를 갱신한다.

4. 단계 3을 변화가 없을 때까지 반복 실행한다.

Figure 8. K-mode algorithm

앞에서 언급한 것처럼 K-mode 알고리즘은 K-means 알고리즘을 확장한 것이라 할 수 있고, 이러한 확장으로 K-mode 알고리즘은 범주형 데이터를 변환 작업 없이 바로 적용할 수 있다. 이렇게 함으로써 대용량의 범주형 데이터를 효율

적으로 클러스터링 할 수 있다. 또한 결과에 대한 해석이 바로 이루어 질 수 있다는 장점을 갖고 있다. 또한 계층적 방법보다 수행 속도 역시 빠르다는 장점을 갖고 있다.

Ⅳ. 제안 알고리즘

K-mode 알고리즘의 단점은 분석하기 전에 미리 군집의 수를 정해 주어야 하고 알고리즘의 첫 번째 단계에서 초기 모드를 어떻게 정하는가에 따라 군집의 결과가 달라질 수 있다는 점이다. 따라서 데이터의 특성을 고려하여 초기 모드를 설정하면 K-mode 알고리즘의 효율을 높일 수 있다. 이 논문에서는 제 3장에서 언급한 Max-Min방법을 범주형 데이터 군집화 방법인 K-mode 알고리즘에 사용할 수 있도록 변형하였고, 수치형 데이터에서의 거리의 개념인 유사도를 이용하여 초기 모드를 선택함으로써 K-mode 알고리즘의 효율을 높일 수 있는 방안을 제시한다.

1. 유사도

Max-Min 방법을 K-mode 알고리즘에 적용하기 위해서는 수치형 데이터의 거리개념인 유사도를 효율적으로 정의하여야 한다. 여기서 정의할 유사도란 앞서 언급한 ROCK 알고리즘의 유사도와 비슷하나 변수들의 속성의 수준을 가중치로 더하여 좀더 세밀하게 정의하였다.

두 객체 $(\mathrm{x}_i, \mathrm{x}_j)$의 유사도를 다음과 같이 정의한다.

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

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

그리고 j 번째 범주형 변수 A_j 는 l_j 개의 수준을 가지며 각 수준을 $c_{j1},\cdots,c_{jl_j}$ 라고 하자.

$$d(\mathrm{x}_{i_1},\mathrm{x}_{i_2}) = \frac{\sum_{j=1}^{m}\delta(x_{i_1 j},x_{i_2 j})}{\sum_{j=1}^{m}\delta(x_{i_1 j},x_{i_2 j})+w} \tag{10}$$

여기서, $\delta(a,b)$ 는 식(2) 과는 반대로 두 값이 일치하지 않을 때 0의 값을 갖고, 그렇지 않으면, 1의 값을 갖는 지시함수이다.

$$\delta(a,b) = \begin{cases} 1, & if \quad a=b \\ 0, & if \quad a \neq b \end{cases} \tag{11}$$

$$w = 2 \times \sum_{j=1}^{m} \frac{1}{|l_j|} \qquad subject \quad to \quad : \quad \delta = 0 \tag{12}$$

가중치(w)는 $\delta=0$ 일 때, 즉, 동일한 변수에 대한 각 객체의 값이 일치하지 않을 때 그 변수의 (1/수준수)들의 합이 된다. 즉, 2개의 수준을 갖는 변수보다 3개의 수준을 갖는 변수가 서로 다를 가능성이 높기 때문에 이를 유사도에 포함한 것이다.

기존 K-mode 알고리즘에서는 수준수가 다른 변수를 수준수가 동일한 변수로 취급하여 상이도를 계산하였기 때문에 이 부분을 개선하였다. 이 유사도는 0과 1 사이의 값을 가지며 두 객체가 서로 유사하면 유사할수록 1에 가까워진다.

Table 1. Categorical object X, Y

변수 / 객체	직업	지역	취미	결혼유무
X	공무원	서울	독서	유
Y	교사	서울	운동	유

표1 은 객체 X,Y 를 나타낸다. X,Y 는 4개의 변수로 구성되어 있고, 직업이라는 변수는 {공무원, 교사, 학생} 3개의 수준으로 되어 있고, 지역은 {서울, 경기, 부산, 제주} 4개의 수준, 취미는 {독서, 운동, 음악감상} 3개의 수준, 결혼유무

는 {유, 무} 두개의 수준으로 구성되어 있다. X,Y 의 유사도를 계산해보면 다음과 같다.

$$d(\mathrm{X},\mathrm{Y}) = \frac{2}{2+2\times\left(\frac{1}{3}+\frac{1}{3}\right)} = 0.6$$

ROCK 알고리즘에서 사용하는 유사도 식(1)을 이용하여 X,Y 의 유사도를 계산해 보면 $sim(\mathrm{X},\mathrm{Y}) = 0.333$ 이 된다. 따라서 수준수를 가중치로 이용하여 유사도를 계산하는 것이 좀 더 세밀하다는 것을 알 수 있다.
이 유사도는 초기 모드를 구할 때 사용할 뿐만 아니라 군집에 할당하기 위하여 모드(또는 초기 모드)와 비교할 때도 사용된다.

2. 유사도를 이용한 초기 모드 결정

K-means 알고리즘에서 Max-Min방법은 첫 번째 초기값 (s_1) 은 랜덤하게 하나의 관측값을 선택하고 두 번째 초기값 (s_2) 은 첫 번째 초기값과의 거리를 최대로 하는 관측값으로 선택한다. 이 방법을 그대로 범주형 데이터에 적용하려면 두 번째 초기 모드를 찾기 위해서 첫 번째 초기 모드와 모든 데이터간 유사도를 계산해야 한다. 이렇게 되면 대용량 데이터일 경우 알고리즘 수행시 많은 시간이 소요된다. 알고리즘 수행시간을 줄이기 위해서 데이터의 10%를 랜덤하게 추출하여 사용하고 첫 번째 초기 모드를 랜덤하게 선택하는 것이 아니라 추후에 군집에 데이터를 할당하기 위하여 초기 모드들과 데이터간 유사도를 비교하는 것을 감안하여 유사도의 분산을 가장 크게 만드는 객체를 첫 번째 초기 모드 (q_1^0) 로 선택하고 첫 번째 초기 모드와 유사도가 가장 낮은 객체를 두 번째 모드 (q_2^0) 로 선택한다.

그림 9는 제 5장에서 실험데이터로 사용된 mushroom 데이터에서 랜덤하게

1000개를 추출한 후 1번 군집과 2번 군집으로 나누었다. 즉, 1번부터 542번까지는 1번 군집, 543번부터 1000번까지는 2번 군집으로 구성되었고, 모든 데이터에 대해서 유사도를 구한 후 유사도의 분산이 가장 큰 849번 객체와 다른 모든 데이터의 유사도를 산점도로 출력한 것이다. 그림 10은 유사도의 분산이 가장 작은 374번 객체와 다른 모든 데이터의 유사도를 이용하여 산점도를 출력하였다.

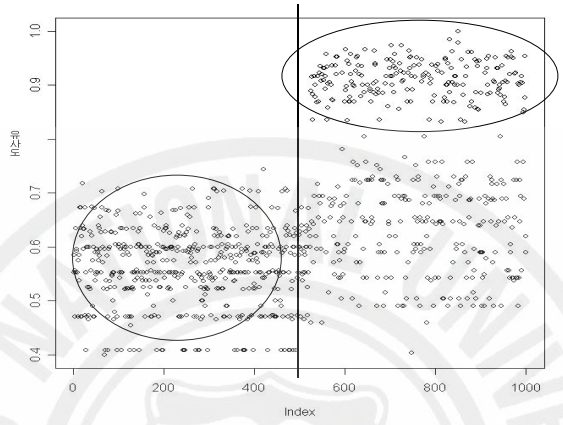

Figure 9. Scatter plot of the biggest object in similarity variance

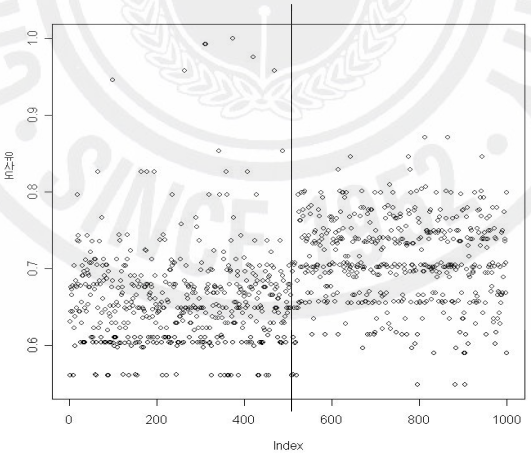

Figure 10. Scatter plot of the smallest object in similarity variance

그림 9과 그림 10을 비교해보면 분산이 높은 그림 9에서는 유사도 계산에서 비교 대상인 849번 객체를 중심으로 동일한 군집의 객체끼리는 높은 유사도를 보이고 다른 군집과의 유사도는 비교적 낮은 편이다. 또한 군집의 경계선이 확연

하게 드러남을 알 수 있다. 반대로, 그림 10은 유사도 계산의 비교 대상인 374번과 대부분의 객체가 비슷한 유사도를 보이고 있다. 물론 군집의 경계인 543번 객체를 전후로 유사도가 약간의 차이를 보이고 있으나 추후에 군집을 할당할 경우에 오분류할 가능성이 높기 때문에 초기모드로 선택할 필요가 없다.

K 가 3이상일 경우 Max-Min 방법을 수정하여 초기 모드를 선택한다. Max-Min방법에서는 앞서 언급했듯이 세 번째 이상의 초기값을 선택하기 위해 처음 두 초기값과 초기값에 선택되지 않은 나머지 관측값들에 대해서 거리를 구하고이 두 거리 중, 최소값을 각 관측값에 대응시킨다. 대응되어 있는 초기값과 관측값과의 거리를 최대화하는 관측값을 구하여 초기값으로 정한다. 이렇게 계산하는 이유는 선택되어질 세 번째 이상의 초기값 $(s_3,\cdots,m : m=3,\cdots,k)$ 들이 앞서 선택한 초기값들과 적절하게 떨어져서 선택되도록 하기 위함이다.

그러나 이러한 방법은 유사도를 사용하는 범주형 데이터에서는 적절하지 못하다. 유사도는 동일한 군집에서는 높게 나타나는 것은 틀림없으나 다른 군집과의 유사도도 높게 나타나는 경우가 있다. 즉, 대부분 변수들간의 속성이 다른 군집과 동일하나 특정 변수 몇 개에서 다른 속성을 가질 경우 다른 군집과도 전반적으로 높은 유사도를 보이게 된다.

Table 2. Process of initial mode decision in Small Soybean data(1)

Index	1	2	3	4	5	6	7	8	9	10
q_1^0	0.703	0.671	0.607	0.742	0.639	0.696	0.703	0.607	0.656	0.688
q_2^0	0.705	0.716	0.72	0.725	0.685	0.716	0.709	0.716	0.77	0.736
최소값	0.703	0.671	0.607	0.725	0.639	0.696	0.703	0.607	0.656	0.688
q_3^0	0.897	0.91	0.832	1	0.88	0.931	0.941	0.85	0.891	0.937
최소값	0.703	0.671	0.607	0.725	0.639	0.696	0.703	0.607	0.656	0.688

위의 표 2는 5장 실험데이터로 사용된 Small Soybean 데이터의 유사도 일부

분을 Max-Min 방법을 사용하여 세 번째 모드 (q_3^0) 를 선택하기 위해 만든 것 중 1번 객체부터 10번 객체까지 q_1^0 , q_2^0 와의 유사도를 나타낸 것이다. 유사도의 분산이 가장 큰 11번 객체가 q_1^0 로 선택되었고, 11번 객체와 유사도가 가장 작은 21번 객체가 q_2^0 로 선택되었다. 또한 실제로 Max-Min방법을 이용하면 4번째 객체가 q_3^0 로 선택되었다. q_4^0 를 선택하기 위해서 Max-Min 방법을 이용하면 1번 객체 또는 7번 객체가 선택되어야 한다. 그러나 Soybean 데이터에서는 1번부터 10번까지는 동일한 군집(1번 군집)이다. 동일한 군집에서 초기 모드가 두개 이상 선택되는 것은 알고리즘 수행시 효율이 떨어질 수밖에 없다.

동일한 군집에서는 객체간 유사도가 가장 높게 나타난다는 점에 착안하여 다음과 같이 초기 모드를 선택한다.

초기 모드 $q_m^0,\ m=3,\cdots,k$ 를 구하기 위해서 이미 구해진 초기 모드를 추가 하면서 초기모드를 제외한 나머지 객체 $x_i(x_i \neq q_l^0, l=1,\cdots,m-1)$에 대해서 이미 구해진 초기모드들과 x_i와의 유사도를 각각 구하고 이 유사도의 최대값을 구한 후 x_i 와 대응시킨다.

$$x_i \leftarrow D_i = \max\left\{ d(x_i, q_1^0),\ d(x_i, q_2^0), \cdots, d(x_i,\ q_{m-1}^0)\right\},\ i=1,\cdots,n \qquad (13)$$

$$here,\ x_i(x_i \neq q_l^0,\ l=1,\cdots,m-1)$$

각각의 객체 x_i에 대해서 얻어진 D_i 를 비교하여 이들의 값을 최소로 하는 객체를 다음 초기값으로 선택한다.

$$q_m^0 = x_m \leftarrow min_{1 \leq j \leq n}(D_j) = D_q \qquad (14)$$

위와 같이 초기 모드와 나머지 객체간의 유사도 중 최대값을 구하는 이유는 앞서 언급한 대로 동일한 군집에서는 객체간 유사도가 가장 높게 나타나기 때문이다. 예를 들어, 1번 군집에서 초기 모드가 설정되면 1번 군집의 객체들과 초기

모드간의 유사도는 매우 높게 나타나고 나머지 객체들과의 유사도는 비교적 낮게 나타게 되고 만약, 높게 나타나더라도 최대값 중 가장 낮은 유사도를 선택하게 되므로 다음번의 초기 모드 설정에서 제외가 된다.

표 3은 표 2와 동일하나 Soybean 데이터의 1번부터 5번 객체, 42번부터 46번 객체까지 표현한 것이다. Max-Min 방법을 사용하면 q_3^0 은 4번 객체가 되고, q_4^0 는 1번 객체나 7번 객체가 선택되어 동일한 군집 내에서 또 다른 초기 모드가 선택된다. 그러나 본 논문에서 제안한 방법에 따르면 초기 모드들과 나머지 객체간의 최대 유사도 중 가장 작은 값을 가지는 5번 객체가 선택된다. 또한, q_4^0 는 46번 객체가 선택된다. 그림 11은 제안한 알고리즘을 단계별로 요약한 것이다.

Table 3. Process of initial mode decision in Small Soybean data(2)

Index	1	2	3	4	5	42	43	44	45	46
q_1^0	0.703	0.671	0.607	0.742	0.639	0.639	0.623	0.607	0.607	0.633
q_2^0	0.705	0.716	0.72	0.725	0.685	0.728	0.77	0.782	0.736	0.708
최대값	0.705	0.716	0.72	0.742	0.685	0.728	0.77	0.782	0.736	0.708
q_3^0	0.928	0.871	0.91	0.88	1	0.74	0.727	0.696	0.747	0.664
최대값	0.928	0.871	0.91	0.88	1	0.74	0.77	0.782	0.747	0.708

1. 대용량 데이터일 경우 데이터의 10%를 표본으로 추출하여 객체간의 유사도를 구한다.
   a. 소용량 데이터일 경우 모든 객체간 유사도를 구한다.

2. 유사도의 분산이 가장 큰 객체를 첫 번째 초기모드로 결정하고 q_1^0 과 유사도가 가장 낮은 객체를 두 번째 초기모드 q_2^0 로 결정한다.

3. 초기 모드 $\mathrm{q}_m^0,\ m=3,\cdots,k$ 를 구하기 위해서 이미 구해진 초기모드를 추가 하면서 초기모드를 제외한 나머지 객체 $\mathrm{x}_i(\mathrm{x}_i \neq \mathrm{q}_l^0,\ l=1,\cdots,m-1)$ 에 대해서 다음을 계산한다.

$$\mathrm{x}_i \leftarrow D_i = \max\left\{ d(\mathrm{x}_i, \mathrm{q}_1^0),\ d(\mathrm{x}_i, \mathrm{q}_2^0),\cdots,\ d(\mathrm{x}_i,\ \mathrm{q}_{m-1}^0)\right\},\ i=1,\cdots,n$$

$$here,\ \mathrm{x}_i(\mathrm{x}_i \neq \mathrm{q}_l^0,\ l=1,\cdots,m-1)$$

4 각각의 객체 x_i 에 대해서 얻어진 D_i 를 비교하여 이들의 값을 최소로 하는 객체를 다음 초기값으로 선택한다.

$$\mathrm{q}_\mathrm{m}^0 = \mathrm{x}_\mathrm{q} \leftarrow min_{1 \leq j \leq n}(D_j) = D_q \quad .$$

5. 각각의 객체들별로 모드와 유사도를 계산하여 가장 유사한 군집에 객체를 할당한다.

6. 모든 객체들에 대해서 군집으로 할당이 끝나면 모드를 갱신한다.

7. 단계 5를 변화가 없을 때까지 반복 실행한다.

Figure 11. The proposed algorithm

Ⅴ. 실험 결과 및 분석

1. 실험 환경

실험은 Microsoft WindowsXP Professional, PentiumⅣ, 3.2GHz, 512MB RAM을 기반으로 하여 공개 소프트웨어 통계 패키지인 R-project를 사용하여 알고리즘을 구현하였다.

2. 실험 데이터

본 논문에서는 군집화 알고리즘의 실험에 빈번하게 사용되는 UCI Machine Learning Repository의 Small Soybean 데이터와 Mushroom 데이터를 사용하였다.

Small Soybean 데이터는 47개의 객체를 가지고 있으며 각각의 객체는 35개의 범주형 속성으로 이루어져 있다. 각각의 속성들은 월(date), 잎 모양, 줄기 상태, 크기 등으로 구성되어 있다.

표 4는 Small Soybean 데이터의 속성의 성분의 일부분을 보여주고 있다. 이 데이터를 실험에 좀 더 적합하게 하기 위하여 속성의 성분을 수치화하였다. 예를 들면, date의 속성을 1부터 6까지 숫자로 표현하였다. 또한 결측치는 0으로 표시하여 실험에 포함하였다. 이전 연구결과에 의하여 Small Soybean 데이터는 4개의 군집(D, C, R, P)으로 구성되어 있음이 밝혀졌고, 군집을 좀더 쉽게 설명하기 위하여 1~4번을 군집의 번호를 부여하였다. 1번 군집은 10개, 2번 10개, 3번 10개, 4번 17개의 데이터를 가지고 있다.

그리고, 제안한 알고리즘이 대용량 데이터에 적합한지를 검정하기 위하여 Mushroom 데이터를 사용하였다. Mushroom 데이터는 총 8124개이고, 그중 4208

개는 식용 버섯을 나타내고, 3916개는 독성 버섯이다. 각각의 데이터들은 버섯의 물리적 특성인 색, 크기, 냄새, 모양 등으로 22개의 속성을 가지고 있다. 또한 각각의 속성은 모두 범주형 데이터로 구성되어 있다. 예를 들어 크기 속성의 성분값은 narrow, broad 로 구성되어 있다. Mushroom 데이터도 Small Soybean 데이터와 마찬가지로 속성의 성분을 수치화 하였다.

Table 4. Part of attributes in Small Soybean data

	속성	속성의 성분
1	date	April, may, June, July, August, september, october, ?.
2	plant-stand	normal, lt-normal, ?.
3	precip	lt-norm, norm, gt-norm, ?.
4	temp	lt-norm, norm, gt-norm, ?.
5	hail	yes, no, ?.
6	crop-hist	diff-lst-year, same-lst-yr, same-lst-two-yrs, same-lst-sev-yrs, ?.
7	area-damaged	scattered, low-areas, upper-areas, whole-field, ?.
8	sevrity	minor, pot-severe, severe, ?.
9	seed-슷	none, fungicide, other, ?.
10	germination	90-100%, 80-89%, lt-80%, ?.
11	plant-growth	norm, abnorm, ?.
12	leaves	norm, abnorm.
13	leafspots-halo	absent, yellow-halos, no-yellow-halos, ?.

3. 실험 결과

제안한 알고리즘의 성능을 평가하기 위하여 기존의 K-mode 알고리즘과 정밀도와 수행시간을 비교하였다. 제안한 알고리즘은 소용량 데이터일 경우 모든 객체에 대해서 유사도를 구하고, 대용량 데이터일 경우 데이터의 10%를 표본으로 추출한 후 모든 객체에 대해서 유사도를 구하기 때문에 기존 K-mode 알고리즘에서 랜덤하게 초기 모드를 선택하는 방법보다는 수행속도가 느릴 수밖에 없다. 그러나 데이터들의 특성을 파악하여 초기 모드를 결정했기 때문에 모드와 객체간의 유사도 비교 후 군집할당, 모드의 갱신단계에서의 수행시간을 비교하였다.

1) Mushroom 데이터

Mushroom 데이터에서 1000개를 랜덤하게 추출하여 실험 데이터를 만들었다. 1000개 중 521개는 식용 버섯이고, 나머지 479개는 독성 버섯이다. 이 데이터에서 제안한 알고리즘대로 데이터의 10%(100개)를 표본을 다시 추출하여 실험하였다.

Table 5. Accuracy of two algorithms in Mushroom data

정밀도	90%초과	90~80%	80~70%	70~60%	60%미만
K-mode	0회	25회	11회	33회	31회
제안	3회	28회	27회	31회	11회

표 5는 기존 K-mode 알고리즘과 제안한 알고리즘을 Mushroom 데이터 1000개를 이용하여 실험한 결과다. 기존 K-mode 알고리즘은 80%이상의 정밀도를 보인 횟수가 25회로 비교적 높게 나타났지만 70%미만의 정밀도를 보인 횟수도 64회 나타났다. 제안한 알고리즘은 80%이상의 정밀도를 나타낸 횟수가 31회이고, 또한 정밀도가 90%가 넘는 경우도 3회가 나타났다. 그리고 70%이상의 정밀

도를 보인 횟수도 58회로 높게 나타났다. 따라서 기존 K-mode 알고리즘보다 제안 알고리즘이 좀더 높은 정밀도를 보였다.

Table 6. Runtime of two algorithms in Mushroom data

	평균수행시간	최장수행시간	최단수행시간
K-mode	284.1983초	735.69초	132.11초
제안	254.52초	616.22초	142.5초

표 6은 Mushroom 데이터에 대한 기존 K-mode 알고리즘과 제안 알고리즘간의 수행시간을 비교한 것이다. 앞서 언급했듯이 제안 알고리즘은 표본을 추출한 후 표본의 모든 데이터의 유사도를 계산하기 때문에 초기 모드를 결정하는 부분은 수행시간에 포함하지 않았다. 기존 K-mode 알고리즘은 평균 284.1983초 수행하였다. 최장수행시간은 735.69초이고, 최단수행시간은 132.11초로 나타났다. 제안 알고리즘에서는 평균 254.52초 수행하였고, 가장 오래 수행한 시간은 616.22초이고, 가장 빨리 수행한 시간은 142.5초로 나타났다. 결과로 봤을 때 제안한 알고리즘이 수행시간 편차도 작고, 평균수행시간도 30초정도 빨리 수행하여 속도 측면에서 기존 K-mode알고리즘 보다 우수하다고 할 수 있다.

2) Small Soybean 데이터

Small Soybean 데이터는 총 객체수가 47개로 소용량 데이터이므로 표본을 추출하지 않고 모든 객체간의 유사도를 계산하여 초기 모드를 선택하였다.

표 7은 Small Soybean 데이터에 대한 제안 알고리즘 수행결과로 정밀도 100%로 나타났다. Small Soybean 데이터는 객체 수가 적기 때문에 모든 객체에 대한 유사도를 계산할 수 있기 때문에 초기 모드를 한번에 선택할 수 있고 또한 제안 알고리즘을 이용하여 각 군집당 1개씩 초기 모드가 결정되었다. 초기 모드는 11번, 21번, 5번, 46번이 선택되었고 Small Soybean 데이터는 1번부터 10번까지 1번 군집, 11번부터 20번까지 2번 군집, 21번부터 30번까지 3번 군집 마직막

으로 31번부터 46번까지 4번 군집으로 이루어져 있다. 따라서 제안 알고리즘은 초기 모드를 효율적으로 선택했다고 할 수 있다.

Table 7. Clustering result by proposed algorithm in Soybean data

	q_1	q_2	q_3	q_4
1		10		
2			10	
3	10			
4				17

Table 8. Runtime of two algorithms in Small Soybean data

	평균수행시간	최장수행시간	최단수행시간
K-mode	7.7376초	8.6초	4.97초
제안	7.55초		

표 8은 Small Soybean K-mode 알고리즘과 제안 알고리즘의 수행시간을 비교한 것이다. Small Soybean 데이터를 K-mode 알고리즘을 이용하여 100회 반복하였다. 수행시간은 제안 알고리즘 7.55초, K-mode 알고리즘은 평균 7.7376초로 제안알고리즘이 약간 빠르다고 할 수 있으나 K-mode 알고리즘에서 최단수행시간이 4.97초로 나타났지만, 이때 수행 결과는 1개의 객체를 오분류 하였다. 이것을 제외하고는 모두 7.5초 이상 수행시간이 소요됐다.

Ⅵ. 결론 및 연구 과제

범주형 데이터 군집화 방법은 최근 컴퓨터의 발전으로 인해 그 동안 축적된 정보를 지식으로 전환하기 위하여 활발히 연구가 진행되는 분야이다. 대표적인 범주형 데이터 군집화 방법은 K-mode 알고리즘과 ROCK알고리즘이라 할 수 있는 데 K-mode 알고리즘은 사용자가 이해하기 쉽고 간편하면서 속도 빠르기 때문에 널리 이용되는 방법이다. 그러나 K-mode 알고리즘은 초기 모드를 랜덤하게 선택하기 때문에 효율성이 떨어진다. 본 논문은 이 단점을 해결하기 위하여 K-means 알고리즘의 초기값 결정 방법인 Max-Min 방법을 범주형 데이터에 사용할 수 있도록 수정하였다. 초기 모드를 결정하기 위하여 대용량 데이터일 경우 표본을 추출하여 유사도를 계산한 후 유사도의 분산이 가장 큰 객체를 첫 번째 초기 모드로 설정하고 이 초기 모드와 유사도가 가장 낮은 객체를 두 번째 초기 모드로 결정한 다음 앞서 결정된 초기 모드들과 객체간의 최대 유사도를 구한 후 이 값을 최소로 하는 객체를 찾아서 다음번 초기 모드로 선택하도록 하였다. 5장에서도 알 수 있듯이 기존 K-mode 알고리즘보다 제안 알고리즘이 정밀도와 군집할당과 모드 갱신의 속도면에서 더 우수함을 알 수 있었다.

본 논문에서 제안한 알고리즘은 비계층적 군집화 방법이기 때문에 데이터의 변환없이 그대로 현실 세계에 적용할 수 있고, 기존의 K-mode 알고리즘보다 정확한 군집화 결과를 나타내준다.

그러나 초기 모드를 결정하기 위해서 표본을 추출하여 모든 객체간의 유사도를 계산해야 한다. 데이터가 커지면 커질수록 많은 객체간의 유사도를 계산해야 하기 때문에 속도면에서 성능이 떨어질 수 있다. 또한 데이터의 대상이 범주형 데이터이기 때문에 수치형과 믹스(mix)되어 있을 경우 군집화를 수행할 수 없다.

따라서 믹스형 데이터일 경우 군집화할 수 있는 방법과 초기 모드 선택시 속도 향상 문제에 대해서 연구해볼 필요가 있다.

Ⅶ. 참고 문헌

Anil., 2001, $K-modes$s Clustering. — Journal of Classification 18: 35-55

Bae W., 2005 , A Study on K-Means Clustering - The Korean Communications in Statistics Vol. 12 No. 2, 2005 pp. 497-508

Bradley P., 1998, Refining initialization of clustering *algorithms* — In: Ahsl, A.(Ed.) ,Proc.4th Internet. Conf. on Knowledge Discovery and Data Mining. AAAI Press, New York.

Bradley P., 1998, Refining initial points for $k-means$ clustering. — In: Proc, 15th Internat.Conf.on Machine Learning .Morgan Kaufmann, Los Altos,CA

Guha, S. Ratogi, R. and Shim, K., 1997, A clustering algorithm for categorical attributes.-Technical Report, Bell Laboratories, Murray Hill

Guha, S. Ratogi, R. and Shim, K., 1999, ROCK : A robust clustering algorithm for categorical attributes. -Proceedings of the IEEE International Conference on Data Engineering, Sydney.

Huang Z, 1997a, Clustering large data sets with mixed numeric and categorical values. — Proceedings fo the First Pacific Asia Knowledge Discovery and Data Mining Conference, Singapore: World Scientific .pp 21-34

Huang Z., 1997b, A fast clustering *algorithm* to cluster very large categorical data sets in data mining. — Proceedings of the SIGMOD Workshop on Research Issues on Data Minig and Knowledge Discovery, Dept. of Computer Science, The University of British Columbia, Canada, pp.1-8

Huang Z, 1997 An alternative extension of the *k−means algorithm* for clustering categorical data. — Int. J. Appl. Math. Comput. Sci., 2004, Vol.14,No.2,241-247

Huang Z, 1998, Extensions to the *k−means algorithm* for clustering large data sets with categorical values. — Data minig Knowledge, Vol2, No.2, pp. 283 -304

허명희, 2004, k-평균 군집화와 재현성 평가 및 응용-한국 통계학회 응용통계연구.제17권 1호. pp 135-144

김보화, 김규성, 2002, K-모드 알고리즘과 ROCK 알고리즘의 개선 -한국 통계학회 응용통계연구, 제15권 2호 pp 381-393

Nam H ,2002 , k-priorities : An Efficient Clustering *algorithm* for Categorical Data Sets. — KIST 석사학위논문

Ying S. ,2002, An iterative initial-points refinement *algorithm* for categorical data clustering. — Pattern Rocognition Letter 23 (2002) 875-884

Zengyou H., Clustering Mixed Numeric and Categorical Data. - Departmen of Computer Science and Engineering. Harbin Institute of Technology Harbin 150001.P. R. China