碩士學位論文

기존 $k-mode$ 알고리즘 개선을 통한 범주형 속성 데이터에 대한 효율적인 클러스터링 방법

濟州大學校 大學院

電算統計學科

吳 秀 岷

2006年 12月

碩士學位論文

기존 $k-mode$ 알고리즘 개선을 통한 범주형 속성 데이터에 대한 효율적인 클러스터링 방법

濟州大學校 大學院

電算統計學科

吳 秀 岷

2006年 12月

기존 $k-mode$ 알고리즘 개선을 통한 범주형 속성 데이터에 대한 효율적인 클러스터링 방법

指導敎授 金 鐵 洙

吳 秀 岷

이 論文을 理學 碩士學位 論文으로 提出함

2006年 12月

吳秀岷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

Effective way of clustering on categorical attribution database through improvement of existing k-mode $k-mode$ algorithm

Su-min Oh
(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

그림 차례

표 차례

Effective way of clustering on categorical attribution database through improvement of existing $k-mode$ algorithm

Abstract

In recent years, as importance of categorical database has increased, various researches about categorical data are performed. Especially, k-mode algorithm which is partitional methods for categorical data as an idea of extending k-means algorithm shows excellent capability in time efficiency. However, similarity calculation through simple comparison between random-choice and mode about initial points mode shows low accuracy about result of clustering.

In this thesis, the selecting method of initial points, which is not random-choice about initial points but based on frequency of data attribution has been improved. Also accuracy of clustering results has im proved by giving weights which it follow in frequency.

I. 서 론

미지의 데이터에 대하여 특정한 패턴을 찾고 이를 바탕으로 하는 지식정보의 추출은 매우 중요하다. 그리고 이를 가능하게 해주는 데이터 마이닝 기법은 큰 이슈가 되어왔다. 클러스터링은 다양한 분야의 수많은 데이터들에 대하여 필요로 하는 지식정보를 추출하기 위한 아주 중요한 과정으로 인식되어왔으며, 현제까지 효율적인 클러스터링방법에 대한 연구는 계속되어 왔다.

일반적으로 클러스터링에 대한 연구 중 분할기법을 사용하는 방식들은 일반적으로 거리를 기반으로 하고 있으며, 고전적인 방식의 클러스터링 기법으로서 $k-means$ 알고리즘이 잘 알려져 있다. $k-means$ 알고리즘은 수치형 데이터의 클러스터링에 있어 매우 효율적이라고 알려져 있는 알고리즘이지만, 현실에서는 수치형데이터 뿐만이 아니라 범주형 데이터가 많이 존재하며 중요시 되고 있다. 결국 클러스터링 기법들도 이에 대한 클러스터링을 요구하고 있다. Huang Z (1997)의 연구는 기존의 $k-means$ 알고리즘을 수치형 데이터에서 범주형 데이터로 확장한 $k-mode$ 알고리즘을 제안한다. $k-mode$ 알고리즘은 데이터의 속성간의 비교를 통한 유사도 측정을 기반으로 하며, 수치형데이터에서 범주형데이터로의 확장된 클러스터링 기법을 제공하였다. 하지만 $k-mode$ 알고리즘도 문제점을 가지고 있다.

물론, $k-means$ 알고리즘을 기반으로 수행함으로써 빠른 클러스터링을 하는 장점이 있지만, 범주형데이터의 단순 비교로 인하여 속성의 중요도를 배제하게 된다. 또 초기치 선정에 따른 클러스터링의 정확도 역시 높은 정확도 결과를 보여주지 못하는 문제점을 가지고 있다. 현제 이를 개선한 여러 알고리즘들이 연구되고 있다. 특히, 초기치 선정문제에 있어 이를 데이터의 $subset$을 이용한 개선방식인 $Initial\ point\ refineing\ algorithm$ 은 $k-mode$ 알고리즘의 정확도 향상에 있어 좋은 결과를 보여준다.(Ying S. 2002) 이외에도 클러스터의 밀도중심에 대한 유사도 측정방식의 개선을 기반으로 하는 $k-representative\ algorithm$.(Van H. 2004) 불필요한 범주형속성에 대한 제거를 통해 효율적인 클러스터링을 수행하는 $k-priorities\ \ algorithm$도 기존의 $k-mode$ 알고리즘을 개선한 방법으로서 클러스터링에 효율적이다.(Nam H. 2002)

본 논문에서는 클러스터링의 정확도 및 효율성을 향상시키기 위해서 데이터 속성의 빈도를 기반으로 하는 초기치 선정의 개선된 방식을 제안하며 또한 클러스터링에 영향을 미치는 속성에 대하여 기본적인 가중치를 적용함으로써 클러스터링의 정확도를 보다 향상시킬 수 있는 방법을 제안하였다. 아울러 수치형 속성과 범주형 속성간의 차이를 극복하는 방법을

제안한다.

먼저 II장에서는 관련연구로서 데이터 마이닝에 대한 전반적인 이론으로서 데이터의 종류, 속성 및 여러 가지 방식의 데이터마이닝기법들을 살펴보고, III장에서는 기존 알고리즘에 대한 소개 및 문제점을 분석하였다. 특히 III.2절에서는 이를 개선시킨 알고리즘을 제안하며 그 방식을 자세히 살펴본다. IV장에서는 많은 연구를 통해 잘 알려진 범주형 데이터베이스로 이루어진 UCI Database 중 Soybean Database와 Mushroom Database를 가지고 실험 및 이에 대한 분석을 하였다.

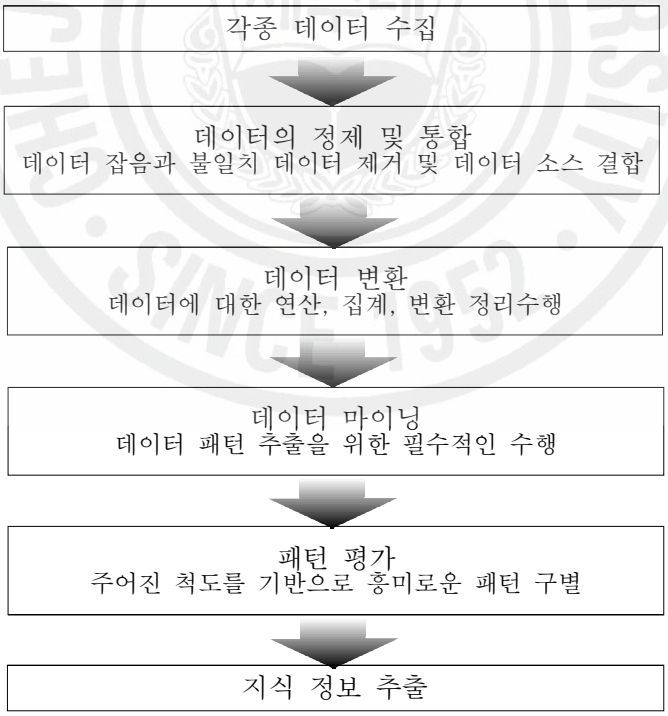
II. 데이터 마이닝

1. 지식발견을 위한 데이터 마이닝

최근 정보 산업 분야에서 데이터 마이닝은 아주 큰 주목을 받고 있다. 정보 산업 분야뿐만 아니라 일상의 주변에서 데이터의 조합으로 이루어진 수많은 정보들을 접하고 있다. 흔히 “정보의 바다”라고 불리는 인터넷을 비롯하여 수많은 분야에서 각종 데이터들을 기반으로 하는 정보들을 접하고 있으며, 이미 인간의 처리능력을 넘어선 한계에 다다랐다. 결국 데이터 마이닝은 대용량의 데이터들로부터 데이터와 사용자 사이에서 보다 효율적으로 지식정보를 추출하는 도구로써 연구가 진행 되고 있다.

데이터 마이닝(Data Mining)은 다양한 종류의 데이터로부터 지식발견(knowledge discovery in database)과정에서 필요한 흥미로운 패턴들을 찾는 중요한 과정이다.

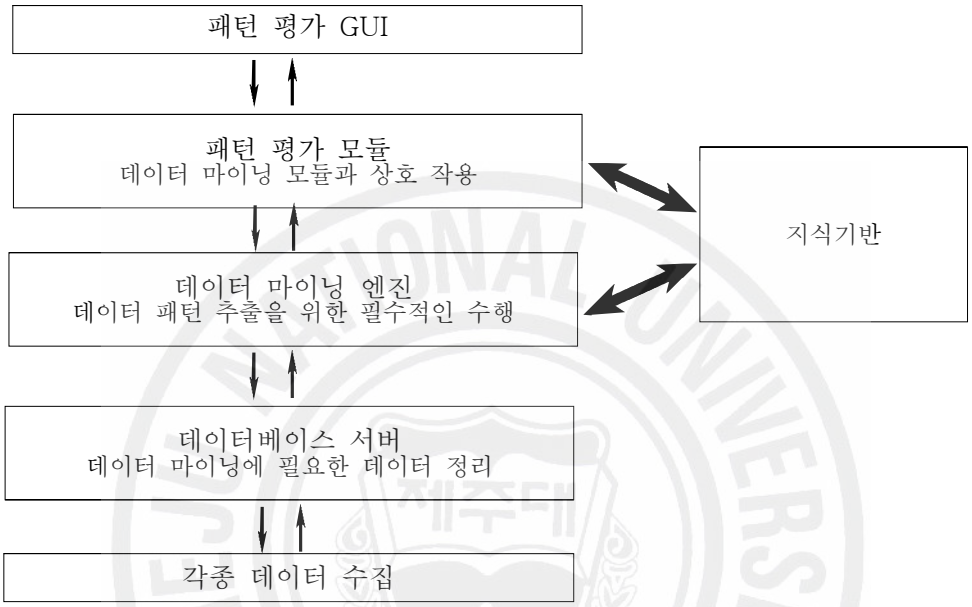
보통 데이터 마이닝을 KDD(Knowledge Discovery in Database:데이터베이스에서의 지식발견)와 동의어로 불리어지고 있다.



그림(1) 지식 정보 추출과정

그림(1)에서 데이터 마이닝은 지식정보 추출을 위한 필수적인 단계로 표현이 되고 있다. 즉 데이터 마이닝은 변환된 데이터를 가지고 일정한 패턴을 찾는 과정을 수행한다. 결국 지식 정보 추출과정의 하나의 중요한 단계로서 표현이 되고 있다.

하지만 정보의 중요성이 더욱 중요시되고 있는 시점에서 데이터 마이닝의 기능 역시 더욱 중요시 되고 있으며, 보다 넓은 시각을 필요로 하고 있다.



그림(2) 개선된 데이터 마이닝 개념

그림(2)에서 데이터 마이닝은 기존의 KDD 보다 진보된 시스템을 보여주고 있다. 데이터 마이닝은 지식정보 및 다른 요소들과 상호 연관을 통해 효율적인 처리를 가능하게 한다. 결국, 데이터 마이닝은 대용량 데이터베이스를 비롯한 다른 여러 가지의 다양한 데이터들로부터 상호작용을 통해 흥미로운 패턴을 찾는 과정으로서 표현된다. 이는 데이터베이스 시스템, 데이터 웨어하우징, 통계, 기계학습, 정보검색 및 신경망, 패턴인식, 공간데이터분석, 영상데이터베이스, 신호처리, 비즈니스, 경제학, 생명정보과학 등의 많은 응용분야에서 활용된다.

또한 데이터 마이닝은 다양한 종류의 데이터만큼이나 다양한 클러스터링 알고리즘 방식들이 존재한다. 클러스터링 방식으로는 분할기법, 계층기법, 밀도기반기법, 격자기반기법, 모델기반 군집화기법, 이상치분석기법등으로 구분한다.

데이터의 종류는 수치속성으로 표현된 수치형데이터와 범주속성들로 이루어진 범주형데이터, 그리고 수치형데이터과 범주형데이터가 혼합되어 존재하는 믹스형데이터등으로 구분한다.

특히 본 논문에서 소개하는 수치형속성 데이터에 대한 효율적인 클러스터링 기법으로는 분할기법을 사용하는 $k-means$ 알고리즘이 있다. $k-means$ 알고리즘은 현재 응용통계분야 분야를 비롯한 많은 분야에서 다양하게 활용되어지고 있다. $k-means$ 알고리즘은 데이터들을 유클리드공간의 점으로 표현하고 데이터간의 유클리드거리를 기반으로 계산하는 방식을 사용한다. 또한 구분되어진 클러스터들 간의 유사도와 상이도를 통해서 클러스터를 구분하게 된다. 이에 대한 것은 III장에서 좀 더 다루도록 하겠다.

2. 데이터속성 유형

클러스터링을 위해서는 먼저 데이터속성의 유형에 대해서 이해해야한다. 일반적으로 데이터의 속성에는 수치형 속성, 범주형 속성 그리고 믹스형 속성으로 구분한다.

1) 수치형 속성 데이터

· 구간척도변수

선형 수치의 연속형 척도로서 수치적으로 측정할 수 있는 속성에 대한 표현을 구간척도변수(*Interval Scaled Variable*)라고 한다. 예를 들어 몸무게, 키, 위도와 경도와 같은 좌표수치들이 대표적인 구간척도변수가 된다. 이러한 속성들에 대한 클러스터링은 결국 측정된 수치의 단위에 따라 군집화에 영향을 받게 되며, 이에 대해서 측정된 수치의 표준화 과정을 수행할 수도 있다. 그리고 데이터들 간의 유사도 측정 및 클러스터의 상이도 측정은 유클리드 거리를 기반으로 한다.

유클리드 거리는 i, j가 p개의 속성을 포함할 때, p개의 데이터 속성에 대한 p차원으로 표현이 가능하게 된다.

즉, 거리 $d(i,j)$는

$$d(i,j)=\sqrt{|x_{i1}-x_{j1}|^2+|x_{i2}-x_{j2}|^2+...+|x_{ip}-x_{jp}|^2}$$

으로 표현이 가능하다.

또한, 구간척도변수에 대한 가중치 적용방식 또한 속성별로 가중치를 적용한다. 가중치(w)는 속성p개에 대해서 다음과 같이 다시 표현이 가능하다.

$$d(i,j)=\sqrt{w_1|x_{i1}-x_{j1}|^2+w_2|x_{i2}-x_{j2}|^2+...+w_p|x_{ip}-x_{jp}|^2}$$

이러한 가중치 적용방식은 범주형데이터 속성의 경우에도 유사하게 적용하게 되며, 본 논문에서 제안하는 가중치 적용방식 역시 위의 개념을 기반으로 한다.

2) 범주형 속성 데이터

· 명목변수

명목변수(*Nominal Variable*)는 둘 이상의 상태를 가지고 있는 이항 변수의 일반형으로서, 색상의 종류(*red*, *blue*, *green*, *pink*, *yellow*, *brown*)와 같이 표현되는 속성이며 명목변수의 속성의 개수(n)는 $1,2,3,...,n$ 으로 이루어진 정수 집합으로 표현한다.

명목변수의 상이도 측정은 변수간의 단순비교를 통하여 계산한다. 그리고 데이터간의 유사도 및 상이도 계산에 필요한 거리(d)는 다음과 같이 표현한다.

$$d(i,j) = \frac{N_n - N_m}{N_n}$$

- i,j : 전체 데이터내의 임의의 객체,
- N_n : 전체 속성의 개수,
- N_m : 전체 속성 중 일치한 속성 개수

즉, 단순 비교를 통하여 데이터간의 거리(d)가 크면 상이도는 증가하며 유사도는 낮게 나타난다.

· 이항변수

0 또는 1로 표현되는 이항변수(*Binary Variable*)는 수치형으로 표현할 수 있다. 하지만 구간척도변수와 같이 유클리드기반의 거리 측정을 통한 유사도를 측정방식을 이용한 유사도 측정은 클러스터링 결과에서 잘못된 오류가 발생한다. 따라서 유사도 측정방식도 이항변수에 적합한 방식을 사용한다.

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

- $q \rightarrow i,j = 1$,
- $r \rightarrow i = 1, j = 0$,
- $s \rightarrow i = 0, j = 1$,
- $t \rightarrow i,j = 0$.

· 서열변수

이산서열변수(*Discrete Ordinal Variable*)는 명목변수와 유사하지만 서열 값들이 의미를 가지고 순서화 되어있다. 예를 들어 스포츠경기의 "금", "은", "동"은 명목변수이긴 하지만 실제 변수가 포함하고 있는 가치는 틀리며, 서열변수 f 에 대하여 $1,\cdots,M_f$로 정의한다. 그리고 서열변수의 상이도 측정은 구간 척도형 변수와 매우 유사하게 계산이 되며, 각 서열에 대한 가중치를 적용함으로서 계산 된다.

i번째 객체의 순위를 r_{if} 라고 할 때, i번째 객체의 f 값을 나타내는 z_{if}는 다음과 같이 표현한다.

$$z_{\mathrm{if}} = \frac{r_{\mathrm{if}} - 1}{M_{\mathrm{if}} - 1}$$

결국 서열변수에서는 각 변수의 범위를 [0.0 , 1.0]으로 표시함으로써 각 변수들에게 일정한 가중치를 적용하여 상이도를 측정한다.

· 비율척도변수

비율척도변수(*Ratio Scaled Variable*)는 박테리아 성장이나 방사능원소의 부패 등이 포함된 지수 범위의 비선형 범위의 양의 값을 갖는 변수이다.

비율척도 변수의 상이도 측정은 구간척도 변수처럼 계산할 경우 클러스터링이 가능은 하지만 데이터의 범위가 왜곡되어 좋은 결과를 기대할 수 없다. 그래서 비율척도변수에 대한 상이도 계산은 로그변환(log*arithmic transformation*)을 통하여 계산한다.

3) 믹스형 속성 데이터

믹스형 속성 데이터는 이미 언급한 1), 2)절의 수치형 속성 데이터와 범주형 속성 데이터가 혼합되어 있는 경우이다. 믹스형 속성 데이터의 클러스터링 방식으로는 일반적으로 유사한 속성변수들끼리 그룹화 하여 그에 맞는 계산을 함으로서 클러스터링을 수행한다.

즉, 유사도 및 상이도 측정에 있어 데이터들을 속성별로 분류하여 계산하고 계산되어진 결과를 다시 합치는 과정을 거쳐 클러스터링을 한다. III장에서 살펴볼 $k-prototype$ 알고리즘이 대표적이다. $k-prototype$ 알고리즘은 수치형 데이터와 범주형 데이터를 구분하여 수치형 데이터는 $k-means$ 알고리즘을 적용하며, 범주형 속성들은 $k-mode$ 알고리즘을 기반으로 계산하게 된다. 결국 이 두 가지 알고리즘을 합쳐서 유사도를 측정하는 방식을 사용한다.

3. 클러스터링 기법

주어진 데이터들에 대해서 사용자가 원하는 최선의 결과를 얻기 위해서 어떠한 클러스터링 기법을 선택하느냐는 매우 중요하다. 여러 가지 기법마다 특징이 있으며 다양한 클러스터링 기법을 활용해보는 것은 지식탐구과정에서도 매우 중요하다.

(1) 분할기법

분할기법(*partitioning methods*)은 n개의 객체에 대하여 k개의 데이터 분할을 만드는 기법이다. 분할기법기반 알고리즘은 클러스터링 되어진 클러스터들이 내부적으로는 강한 유사도를 보이며 클러스터들 간에는 상이도가 크게 된다. 즉, 클러스터들 간의 차이가 크면 잘 된 클러스터링이다.

분할기법의 알고리즘으로는 군집에 있는 객체간의 평균값을 기반으로 하는 $k-means$ 알고리즘 방식이 있으며, $k-means$ 방식은 III.1절에서 자세히 살펴보겠다. 그리고 범주형속성 데이터의 클러스터링 방식으로 본 논문에서 다루게 되는 $k-mode$ 알고리즘이 있으며 역시 III.1절에서 자세히 살펴보겠다.

일반적으로 분할기법은 초기 분할k에 대하여 반복적인 재배정기법(*iterative relocation technique*)을 사용한다. 좋은 결과를 얻기 위해서는 최소한 아래의 두 조건을 만족해야 한다.

(i) 모든 클러스터는 최소한 하나 이상의 객체를 포함한다.

(ii) 모든 객체는 k개의 클러스터로 중복 없이 클러스터링 된다.

(2) 계층기법

계층기법(*hierarchical method*)은 주어진 데이터 객체들을 계층적으로 분할하며, 계층의 분할 형태에 따라 상향식 접근방식과 하향식 접근방식으로 구분할 수 있다. 이러한 방식은 일반적으로 트리형태의 구조를 이루게 된다. 계층기법은 조건이 종료될 때까지 반복하면서 주변의 객체들을 합치는 과정을 수행하며 객체가 하나의 군집으로 되거나 혹은 더 작은 클러스터로 분할될 때 까지 수행된다.

계층기법 클러스터링의 효율을 높이는 방법으로는 트리의 각각의 클러스터링 단계에서 다른 클러스터링 기법을 적용하는 방법이 있다. 트리구조의 계층을 이용한 반복적인 감소를 통한 클러스터링 기법으로 BIRCH(*Balanced Iterative Reducing* and *Clustering Using*

Hierachies) 기법, 중심기반으로하는 구형태의 클러스터나 비슷한 크기의 클러스터들에 대해서 효율적인 CURE(*Clustering Using REpresentatives*)기법, 동적인 모델을 이용한 계층 클러스터링 알고리즘인 Chameleon 기법 등이 있다.

(3) 밀도기반기법

밀도기반기법(*density based method*)은 객체간의 클러스터가 거리를 기반으로 하는 것이 아닌 객체속성의 밀도 값을 기반으로 클러스터링 하게 된다. 이 방식은 객체들의 이상치값을 배재한 형태의 클러스터를 생성하게 된다.

· DBSCAN

DBSCAN(*Density-Based Spatial Clustering of Applications with Noise*) 기법은 데이터들의 밀도를 가지고 클러스터링하는 기법으로서 밀도가 높은 지역을 중심으로 클러스터링하며 이상치 값을 갖는 공간에 대해서도 임의의 형태의 클러스터로 구현이 가능하다.

· OPTICS

OPTICS기법은 DBSCAN기법의 단점을 보완하기 위하여 제안된 기법이다. 밀도를 계산하기 위해 사용되어지는 인자들의 상호작용을 통해 광범위한 인자들로부터 점진적인 클러스터링 순서를 계산하여 클러스터링을 수행한다.

· DENCLUE

DENCLUE(*DENsity-based CLUstEring*) 기법은 밀도분포함수를 기반으로 하는 클러스터링 기법으로서 데이터간의 연관성을 영향력 함수(*influence function*)를 사용하여 모델화한다. 데이터의 밀도는 각각의 데이터의 영향력 함수의 합으로 표현이 되며, 이 함수의 극대값을 클러스터의 군집화에 활용하는 기법이다.

- 안정적인 수학적 기반의 수행 및 고차원데이터에 대한 효과적인 수행
- 분할기법, 계층기법등의 비교적 다양한 기법에의 활용
- 클러스터링 과정에서 발생하는 이상치 값들에 대한 효율적인 클러스터링 효과

(4) 격자기반기법

격자기반기법(*grid based method*)은 객체공간을 격자구조로 이루어진 유한개공간으로 나누고 클러스터링은 이러한 격자 내에서 격자들을 클러스터링하게 된다.

· STING

STING(*STatistical INformation Grid*)는 데이터 공간의 구조를 사각의 격자기반으로 표현하여 클러스터링하는 기법이다. 높은 수준의 셀들은 보다 낮은 수준의 셀로 분할되기 위해 계층 구조내의 질의 응답과정의 반복을 통하여 계산되어지게 되고, 결국 질의에 적합한 셀들이 결과로 구해진다.

이 기법의 장점은 각 셀의 정보가 독립성을 유지할 수 있으며, 병렬처리 및 점진적 수정을 수월하게 함으로서 효율성을 높인다. 하지만 군집의 형태가 수직수평으로만 관계를 함으로써 수행시간에 있어 효율적일 수는 있으나 군집의 정확성은 높지 않다.

· WaveCluster

WaveCluster는 데이터 공간의 다차원 격자 구조를 기반으로 작동하는 밀도기반 알고리즘이다. 클러스터형태가 일반적으로 모자형의 클러스터로 표현이 가능하여 클러스터링 되는 영역과 클러스터링이 되지 않는 영역간의 정보를 보다 효율적으로 표현한다. 결국 클러스터의 형태가 일정하지 않은 데이터에 대해서 효율적으로 작동하며, 기존의 기법들(BIRCH, CLARANS, DBSCAN)보다 우수한 성능을 보여준다.

· CLIQUE

CLIQUE(*CLustering In QUEst*)기법은 밀도기반기법과 격자기반 알고리즘을 혼합시킨 방법으로 고차원데이터에 대해 효율적인 기법이다. 데이터공간을 사각형태의 격자형태로 구분하여 밀도에 따른 사각형의 조밀도를 계산하게 된다. 이 기법은 데이터의 순서에 상관없으며 일반적인 자료분포도 따르지 않는다. 입력된 데이터에 따라 선형적인 확장을 통해 보다 좋은 확장성을 갖는다. 하지만 단순혼합방식으로 인해 군집화 결과의 정확도는 떨어진다.

(5) 모델기반

모델기반(*model Based method*) 기법은 각 군집에 모델을 가정하고 주어진 모델에서 가장 잘 맞는 데이터를 찾는 방식이다. 통계학을 기초하여 군집의 수를 결정하게 되며 이상치 값을 고려한 클러스터링을 하게 된다.

III. $k-mode$ 개선 알고리즘

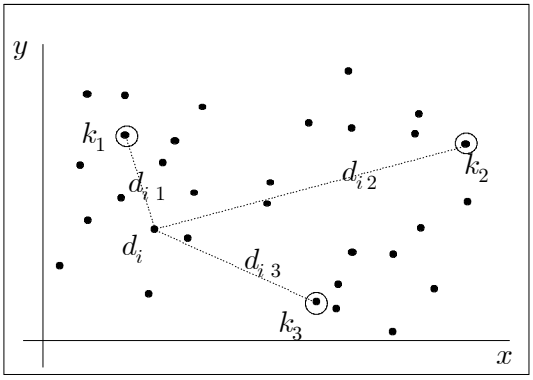
이번 절에서는 분할기법 데이터 마이닝 클러스터링 알고리즘으로 $k-means$, $k-mode$, $k-prototype$ 알고리즘을 소개한다. 각각의 알고리즘은 공통적으로 초기치 k의 선정에 따른 k와 다른 모든 객체들 간의 거리를 측정하여 유사도를 개산하게 되며, 반복수행을 통해서 최적의 클러스터로 분할되는 방식이다. 알고리즘은 수치형데이터, 범주형데이터, 믹스형데이터 들을 클러스터링 하는데 매우 효과적이며 기본적으로 유사한 형태를 취하고 있다. 하지만 각각의 알고리즘들의 거리(d)의 측정방식에서 차이를 두고 있다.

III.2절에서는 $k-mode$ 알고리즘에 대하여 초기치 선정 방식의 개선 및 유사도 측정의 개선을 통한 $k-mode$ 개선 알고리즘을 제안하였으며, 더불어 혼합형 데이터에서 수치형데이터와 범주형 데이터간의 차이를 보완해주는 웨이트(γ)에 대한 제안도 소개한다.

1. 알고리즘 정리

1) $k-means$ $algorithm$

$k-means$ 알고리즘은 데이터속성 중 가장 일반적인 수치형 속성을 갖는 데이터를 위한 클러스터링 알고리즘이다. 일반적으로 응용통계프로그램등에서 기본적으로 사용이 되며 많은 응용이 이루어지고 있는 대표적인 클러스터링 기법이다. $k-means$ 알고리즘은 랜덤하게 선택된 임의의 j개의 초기치 클러스터들을 선정하고 이를 기반으로 클러스터링을 수행한다. 거리 측정은 일반적으로 유클리드 거리를 기반으로하게 된다. 데이터간의 유사도가 높다면 거리는 낮은 수치로 표현이되며, 이는 다시 말해 클러스터가 종료 되었을 때 클러스터간의 상이도는 커진다.



그림(3) 2차원 평면상의 객체와 클러스터간의 거리

그림(3)에서 d_i는 2차원 평면상의 한 점으로 표현된 데이터 객체이다. d_i는 먼저 선정된 초기 클러스터의 중심 k_1, k_2, k_3와 각각의 거리를 계산하고 된다. 그림에서는 k_1과 가장 가깝게 나타나므로 d_i는 k_1의 클러스터로 포함이 된다. 마찬가지 방식으로 반복과정을 통해 모든 데이터를 k_1, k_2, k_3와 거리를 계산해서 포함관계를 찾아낸다. k_1, k_2, k_3들이 포함하는 각각의 데이터들에 대하여 데이터들의 중심을 계산하게 되어 $k_j \rightarrow k'_j$로 클러스터의 중심을 구한다. 다시 거리를 계산하게 되고 β번 반복 후 $k_j^{\beta-1} \rightarrow k_j^{\beta}$ 의 결과가 변화가 없다면 수행을 종료하게 되어 클러스터링을 종료 한다.

· 유사도측정방법

임의의 랜덤선택되어진 k들에 대하여, 나머지 객체들과의 클러스터의 평균($means$)을 기반으로 할당과정을 진행한다.

$$d = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2$$

- p : 주어진 객체의 공간의 점
- m_i : 클러스터의 중심
- C_i : 평균($means$)

일반적으로 거리d는 데이터베이스에서의 모든 객체들의 제곱오차의 합으로 표현한다.

$k-means$ $algorithm$

→ 입력 : n개의 수치형 데이터 베이스, 초기치 선정k의 개수

(1) 입력된 데이터에 대한 초기치 k개의 평균값($means$) 지정

(2) k와 객체간의 유클리드거리기반의 유사도측정

(3) 유사도값에 따른 클러스터 할당

(4) 클러스터의 $means$ 갱신

(5) 클러스터의 변화가 없을 때까지 (2)~(4)단계 반복수행

→ 출력 : 최적 배정 k 클러스터

$k-means$ 알고리즘은 수치형 데이터에 대해 상당히 빠른 속도와 더불어 좋은 결과를 보여준다. 하지만 거리계산방식이 유클리드거리를 기반으로 하기 때문에 범주형 데이터 속성을 같는 데이터에 대해서는 처리할 수가 없다.

$k-means$ 알고리즘의 범주속성데이터에 대한 확장은 $k-mode$ 알고리즘이 잘 알려져 있다.

2) $k-mode$ *algorithm*

$k-mode$ 알고리즘은 $k-means$ 알고리즘을 수치형데이터에서 범주형 데이터로 클러스터링 범위를 확장한 알고리즘이다. 범주속성의 데이터는 단순히 수치적 연산만을 가지고 해결할 수 없다. 그래서 $k-means$ 알고리즘에서 사용되었던 거리개념을 $k-mode$ 알고리즘에 적용하기 위해서 유클리드 거리가 아닌 상이도 측정을 위한 거리개념으로 확장한다.

· 상이도 측정

N개의 레코드들로 이루어진 데이터 베이스에서 임의의 데이터 $N_p,(p \in N)$는 $A_p = \{a_1,...,a_i\}$는 i개의 속성으로 이루어져있다고 하자. 결국 모든 속성들의 집합 A는 $A = \{A_1,...,A_N\}$로 표현된다. 이때 속성들간의 비교를 통해 객체들 간의 상이도 및 유사도 측정에 필요한 거리(d)를 계산한다.

$\{x, y\} \in N, \{x_i, y_i\} \in A_p$에서

$$d(x,y) = \sum_{i=1}^{N} \delta(x_i, y_i), \qquad where \quad \delta(x_i, y_i) = \begin{cases} 0 & (x_i = y_i) \\ 1 & (x_i \neq y_i) \end{cases},$$

로 표현한다.

즉, 비교하는 데이터베이스의 레코드들에 대하여 각각의 범주속성 값이 같을 경우는 0, 같지 않을 경우는 1 로 계산하게 되며, 두 데이터의 거리는 이들에 대한 합으로 계산된다.

결국 계산되어진 거리(d)는 두 객체간의 유사도값이 된다. 이때 거리(d)값이 크면 데이터는 속성이 다른 것이 많다는 것을 의미하므로 상이도가 커지며, 반대의 경우는 거리(d)값이 낮으면 속성이 유사한 것이 많다는 것을 의미하므로 상이도는 낮아지게 된다. 이러한 계산 방식을 통해 객체는 가장 상이도가 낮은 클러스터로 할당이 된다.

	자동차	취미	직업	집	수익
d_1	티코	테니스	학생	아파트	100만원이하
d_2	티코	인라인	학생	단독주택	100만원이하
d_3	벤츠	테니스	의사	단독주택	100만원이상
d_4	소나타	골프	변호사	아파트	100만원이상

그림(4) 범주형 데이터에 대한 속성값 표현

그림(4)에서 초기 선정된 클러스터가 $\{d_1,d_4\}$일 경우, 객체d_1, d_4 은 d_1, d_2, d_3, d_4와 거리를 비교하게 된다. 위에서도 이미 언급했듯이 객체 속성에 대하여 단순히 거리 개념을 사용할수 없다. 이때 데이터의 객체를 비교하게 된다. 속성이 같으면 0, 틀리면 1을 취하며 이에 대한 합이 거리값으로 주어지게 된다. 객체d_1은 $\{d_2,d_3,d_4\}$ 에 대하여{2,5,4}라는 거리값을 갖으며, d_1은 d_2와 클러스터를 이루게 된다. 마찬가지 방법으로 d_4는 d_3와 클러스터를 이룬다.

$k-mode\ algorithm$

→ 입력 : n개의 범주형 데이터베이스, 초기치 k의 개수 지정
(1) 입력된 데이터에 대한 초기치 k들에 대한 $mode$선정
(2) $mode$와 데이터간의 상이도기반의 유사도거리 측정
(3) 유사도 값에 따른 클러스터 할당
(4) 클러스터의 $mode$ 갱신
(5) 클러스터의 변화가 없을 때까지 (2)~(4)단계 반복수행
→ 출력 : 최적 배정 k 클러스터

$k-mode$ 알고리즘은 범주속성의 데이터를 비교함으로써 수치적인 표현이 가능하며 이를 기반으로 데이터들을 클러스터링 할 수 있게 하였다. 또한, $k-means$ 방식과 유사한 계산과정을 수행하므로 빠른 클러스터링 수행시간을 보여준다.

하지만 범주형 속성에 대한 단순 비교만을 하게 되어 비중있는 속성에 대한 처리가 미흡하다는 단점을 가지고 있다. 즉, 좀 더 중요한 속성에는 그 만큼의 가중치를 주는 것이 필

요하다.

이번에는 그림(4)에서 초기 랜덤 선정된 클러스터가 d_1, d_3라고 하자. 객체간의 거리를 구하면 d_3의 경우 d_2, d_4에 대한 유사도 값이 4로 계산된다. 즉, d_3는 d_2, d_4에 대하여 같은 유사도 값을 가지게 됨으로써 유사도값 4에 대하여 같은 클러스터에 포함시키거나 혹은 d_2, d_4 둘 다 클러스터에 포함하지 않아야 한다. 결국 d_3는 잘못된 클러스터링 결과를 발생시킬 수 있다. 결국, 초기치로 선정되는 클러스터는 클러스터링 결과에 영향을 준다.

3) $k-prototype\ \ algorithm$

$k-prototype$ 알고리즘은 수치형속성과 범주형속성이 혼합되있는 데이터에 대해서 클러스터링 할 수 있는 알고리즘이다. $k-prototype$ 알고리즘은 수치형속성과 범주형속성을 구분하여 유사도를 측정하게 되며, 속성간의 웨이트(γ)연산을 추가함으로써 수행된다.

· 거리측정

$N=\{N_1,...,N_n\}$개의 레코드들은 각각 $A_1,...,A_p,A_{p+1},...,A_m$의 속성을 가질 때, p개의 수치형 속성과 $m-p$개의 범주형 속성을 포함한다. 이때 두 속성을 구분하여 연산을 수행하게 되며 거리계산은 다음과 같다.

$$d(x,y)=\sum_{j=1}^{p}(x_j-y_j)^2+\gamma\sum_{j=1}^{m-p}\delta(x_j,y_j)$$

더불어 수치속성데이터와 범주속성데이터간의 차이에 대한 γ 연산을 추가한다. 그리고 γ는 데이터들의 분산을 가지고 정의하게 되며 일반적으로는 경험적으로 주어지게 된다. 하지만 이는 상황에 따라 변하게 되는 불안정한 수치가 된다. 그러므로 항상 적용가능한 방법이 필요하다. 이에 대한 제안은 III.2절에 소개한다.

정리하면,

$$P(W,Q)=\sum_{l=1}^{k}\left(\sum_{i=1}^{n}w_{i,l}\sum_{j=1}^{p}(x_{i,j}-q_{l,j})^2+\gamma\sum_{i=1}^{n}w_{i,l}\sum_{j=p+1}^{m}\delta(x_{i,j},q_{l,j})\right)$$

로 계산한다.

$k-prototype\ algorithm$

→ 입력 : n개의 혼합형 데이터베이스, 초기치 선정k

(1) 입력된 데이터에 대한 초기치 k개를 지정

(2) k와 데이터간의 속성별 정리 및
웨이트 적용기반의 유사도거리 측정

(3) 유사도 값에 따른 클러스터 할당

(4) 클러스터 갱신

(5) 클러스터의 변화가 없을 때까지 (2)~(4)단계 반복수행

→ 출력 : 최적 배정 k 클러스터

$k-prototype$ 알고리즘은 수치형속성과 범주형속성이 혼합된 데이터에 대해서 효과적으로 클러스터링 할 수 있는 장점을 가지고 있다. 하지만 $k-prototype$ 알고리즘은 기존의 $k-means$ 알고리즘과 $k-mode$ 알고리즘에 대하여 단순히 혼합한 방식이므로 기존에 가지고 있는 문제점을 그대로 유지한다. 또한 범주형속성과 수치형속성간의 웨이트(γ)적용 방식에 대한 좀 더 다양한 연구를 필요로 한다.

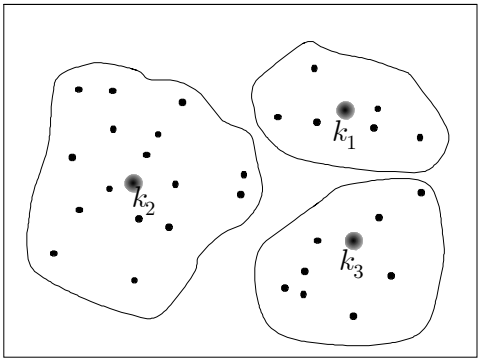
2. $k-mode$ 개선 알고리즘

이미 언급되어진 $k-menas$ 알고리즘과 $k-mode$ 알고리즘은 각각의 데이터속성에 대하여 효과적이다. 하지만 $k-means$ 알고리즘과 $k-mode$ 알고리즘은 몇 가지 문제점들을 가지고 있다. 본 논문에서 최근 많은 연구가 진행되고 있는 범주형데이터 클러스터링을 위한 $k-mode$ 알고리즘을 개선시킨 알고리즘을 제안한다. 먼저 초기치 k의 선정방법의 개선을 통하여 보다 정확한 결과를 구하기 위한 방법을 제안하며, 다음으로 기존의 단순비교를 통한 유사도 계산방식에서 중요한 속성에 대한 가중치 적용 유사도 측정방식을 제안한다. 그리고 혼합형 데이터 속성의 클러스터링을 위한 $k-prototype$ 알고리즘의 웨이트(γ)의 선정 방식을 제안한다.

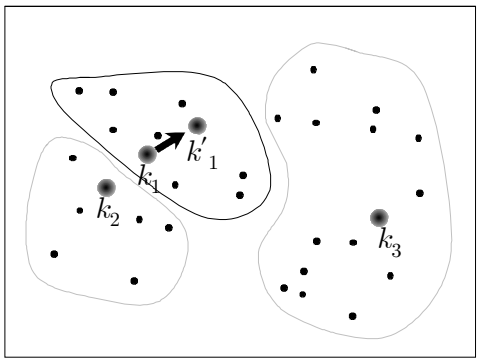
1) 초기치 k 선정 방법 개선

먼저, k_1, k_2, k_3은 $N=\{N_1, \cdots, N_n\}$개의 데이터에 대하여 $A=\{A_1, \cdots, A_i\}$의 속성을 포함하는 데이터에 대하여 선택되어진 초기 클러스터 값이다.

일반적으로 $k-means$ 알고리즘과 $k-mode$ 알고리즘은 초기 선택되어진 클러스터의 중심을 기준으로 클러스터링 되므로 결국 구형태의 클러스터링이 이루어진다. 즉, 유사도 계산에 따른 반경 내에 있는 객체들을 반경의 중심 클러스터에 포함시키는 방식이다. 하지만 $k-mode$ 알고리즘은 범주형 데이터를 기반으로 하므로 구형태의 클러스터링이 되는 것을 표현하지는 않는다. 하지만 개념적인 측면으로 볼 경우, 속성의 수를 차원으로 볼 수 있다. 이는 속성의 수 i에 대하여 i차원에 대한 수치로 변환이 가능하게 되며 알고리즘 수행시 메트릭스 형태로 변환하여 표현이 가능하다.



그림(5) 잘 선택되어진 초기치에 대한 클러스터



그림(6) 반복 클러스터링을 통한 클러스터 중심의 이동

위 그림(5)을 보면 잘 선택되어진 k_1, k_2, k_3에 대해서 클러스터링을 진행할 경우로써, 클러스터링의 결과도 좋은 결과가 나타남을 알 수 있다. 하지만 그림(6)과 같이 k_1과 k_2가 인접하게 랜덤 선택되는 경우가 존재한다. 물론 잘 선택된 k_1, k_2가 좋은 클러스터링을 할 수 있다면 문제가 없다. 하지만, 랜덤하게 선택된 k_1, k_2로부터 항상 좋은 결과를 기대 할 수 없다. 물론, 알고리즘의 반복을 통해 클러스터의 중심이 그림(6)처럼 $k_1 \rightarrow k'_1$ 으로 중심의 위치가 변경이 되어도 마찬가지이다. 결국, 초기치 선정은 클러스터 결과에 많은 영향을 주며, 잘 선택된 초기치는 더 좋은 결과를 기대할 수 있다.

본 논문에서는 $k-mode$ 알고리즘의 개선된 k선정방식을 제안한다. N개의 데이터레코드에 대하여 모든 레코드는 m개의 속성을 포함한다. 이때 임의의 $A_g, (g \in N)$는 m개의 성분들에 대한 집합으로 표현할 수 있으며 A_g는 속성들의 빈도를 표현한 메트릭스로 표현할 수 있다. 그리고 속성 g번째 속성의 범주 빈도의 합은 전체 데이터베이스의 개수가 되며 $N = num[A_{1g}] + num[A_{2g}] + ... + num[A_{ig}], (num[x]: x$의 개수)으로 표현할 수 있다. 이때 i는 각 속성 범주의 개수중 가장 높은 값이 된다. 그리고 속성 m과 i는 초기 데이터 수집 결과에 따라 달라지게 된다. 결국, 메트릭스 A_g는 속성에 대한 빈도로서 표현된다.

$$Database = \begin{Bmatrix} A_{11}A_{12}A_{13}\cdots A_{1m} \\ A_{21}A_{22}A_{23}\cdots A_{2m} \\ A_{31}A_{32}A_{33}\cdots A_{3m} \\ \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \\ A_{N1}A_{N2}A_{N3}\cdots A_{Nm} \end{Bmatrix}, \quad A_g = \begin{Bmatrix} A_{11}A_{12}A_{13}\cdots A_{1m} \\ A_{21}A_{22}A_{23}\cdots A_{2m} \\ A_{31}A_{32}A_{33}\cdots A_{3m} \\ \vdots \quad \vdots \quad \vdots \quad \ddots \quad \vdots \\ A_{i1}A_{i2}A_{i3}\cdots A_{im} \end{Bmatrix}$$

결국, 메트릭스A_g는 m차원에서의 수치적인 표현으로 구현할 수 있다. 이때 임의의 l ,$(l \in m)$번째 속성에 대하여 모든 빈도값이 $A_{l,s}$,$(s \in i)$에서 $num[A_{l,s}] = N$의 경우가 발생한다. 이러한 경우가 발생할 때, 이는 모든 l번째 속성에 대하여 모두 동일한 값을 갖는다는 것을 의미한다. 즉, 클러스터링 수행시 l번째 속성은 차원이 일차원 되어 클러스터링에 아무런 영향을 주지 않는다. 반대로 $A_{ls_1}, A_{ls_2}, A_{ls_3}(s_1, s_2, s_3 \in i)$속성의 성분에 대하여 $num[A_{ls_1}] + num[A_{ls_2}] + num[A_{ls_3}] = N$ 이 되는 경우를 보자. 이는 모든 데이터는 l번째의 성분이 s_1, s_2, s_3의 3가지 성분으로 밀집되어 있음을 뜻한다. 즉, s_1, s_2, s_3성분은 클러스터의 영향을 주는 요인이 된다. 추가적으로 본 눈문의 실험에서는 $\{A_{l1}, ..., A_{li}\}$들의 값에 대하여 전체데이터 N의 갯수의 5%미만의 데이터가 포함되어있는 속성들을 배제하였다. 이러한 방식으로 배제된 속성의 개수를 q라고 하면, 결국 q는 데이터의 속성값이 한 가지 성분이거나 5%미만의 데이터로서 클러스터링에 영향을 주지 않는 것으로 처리한다. 그리고 5%이상의 데이터가 포함된 데이터는 클러스터에 영향을 주는 것들로 판단하고 그 개수를 측정하여 p라고 한다.

속성집합 A는 다음과 같이 표현할 수 있다.

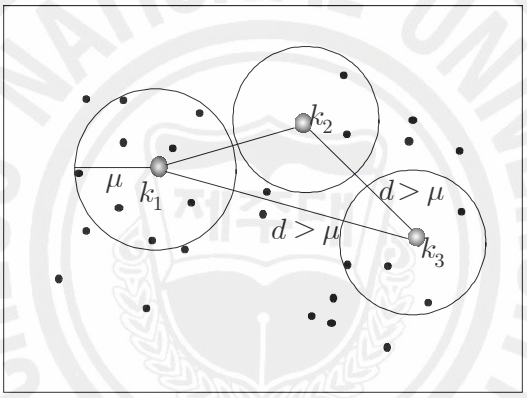
$$\begin{aligned} A &= \{A_1, ..., A_m\} \\ &= \{A_1, ..., A_p, A_{m-p+1}, ..., A_m\} \\ &= \{A_1, ..., A_p, A_{p+1}, ..., A_q\} \end{aligned}$$

이때 초기 클러스터를 결정하는 기준 μ값을 계산한다. 영향을 주는 속성들의 개수를 기반으로 $\mu = p/2 + q$ 로 둔다. 계산되어진 μ는 초기 클러스터 선정에서 클러스터들간의 유사도 측정의 기준이 된다.

즉, 초기치 k개의 선택시 모든 k를 랜덤하게 선택하지 않으며, k_1만을 랜덤하게 선택을 하게 되고 이후의 선태되어질 k들은 k_1과의 거리가 항상 μ이상의 거리에 있는 k들을 선정한다. 또한 $K = 3$일 경우 정해진 k_1, k_2, k_3들은 서로간의 거리가 μ이상이 되도록 하여 k_2, k_3의 관계역시 일정한 거리를 이루도록 한다. μ값에 대한 계산에서 $p/2$는 클러스터링 수행시 데이터가 클러스터링에 영향을 받는 범위를 뜻한다. 만약 $\mu = p + q$로 둘 경우 이는 완전히 동일한 값을 갖는 데이터를 데이터속성을 갖는 객체를 선택하게 되어 의미가 약하게 되며, 만약 $m = q$로 둘 경우 역시 의미가 없어지게 된다. 그래서 $p/2$만큼을 적정선으로 한

다. 만약 임의의 객체에 대한 유사도 값(α)이 ($q< \alpha< p/2+q$)의 범위에 있다고 하면 이는 약한 관계에 있음을 의미한다. 이는 다른 클러스터에 더 큰 영향을 받는다는 것을 의미하며 결국 ($p/2+q< \alpha < m$)의 범위내에 있는 다른 클러스터에 포함이 된다고 본다. 즉, μ계산시 좀더 강한 클러스터링을 요구하기위해서는 $p/2$을 조절하면된다. 아울러 제안하는 방식으로 계산된μ을 기준으로 하는 초기 클러스터는 알고리즘의 반복을 통해 최적의 클러스터로 클러스터링 하게 된다. 추가적으로 사용자는 좀더 강한 유대관계를 같는 클러스터링을 원할 수 있다. 이때는 μ값을 증가하면서 초기치들을 선정할 수 있다.

결국 초기치 클러스터 선정시 객체가 영향을 주는 성분의 개수 중 데이터와 k들과 유사도를 비교함에 있어 영향을 주는 범위를 지정함으로써 공정한 초기치k들을 지정할 수 있다.



그림(7) 개선된 초기치 k선정방식

그림(7)은 초기 랜덤 선택된 k_1을 기준으로 k_1, k_2, k_3들은 모두 기준치(μ)이상을 유지한 k들이 선택되어져 있음을 볼 수 있다. 초기 랜덤 선택되는 k_1이 어떠한 객체가 선택이 되어도 다음 선택되는 k_2, k_3가 일정한 범위이상의 객체를 선택할 수 있으며, 클러스터링 수행시 좀 더 정확한 클러스터링 결과를 기대할 수 있다.

2) 유사도 계산방식 개선

기존 $k-mode$ 알고리즘의 경우 유사도 계산은 δ들의 합으로 정의되어진다. 즉, 유사할수록 δ값이 낮아져서 같은 클러스터로 클러스터링 된다. 하지만 이 방식으로는 아주 긴밀한 자료들에 대한 처리가 부족하다. 즉, 좀 더 중요한 속성에 대하여 가중치(w_s)를 추가연산을 해야한다. 본 논문에서는 δ에 대한 연산방식을 개선하였다.

기존 δ연산에 대하여,

$$\delta(x_i,y_i)=\begin{cases}0 & (x_i=y_i)\\ 1 & (x_i\neq y_i)\end{cases} \rightarrow \begin{cases}1+w_i & (x_i=y_i)\\ 0 & (x_i\neq y_i)\end{cases},(i\in m)\,(x,y\in N)$$

와 같이 개선하였다.

개선된 계산방식으로 구하여진 유사한 클러스터들은 좀 더 큰 값을 가지게 되며, 유클리드거리를 기반으로 하는 거리개념이 아닌 유사도 값이 커지는 개념으로 해석한다. 또한 III.2절에서 제안했던 p,q에 대해서

$$w_i=\begin{cases}1 & (i=p)\\ 0 & (i=q)\end{cases}$$

만큼 추가적인 가중치 연산을 한다. 이는 기본적인 가중치 방법으로서 클러스터링에 영향을 주는 속성의 경우 가중치를 1만큼 추가로 가중하여 유사도를 측정하게 된다. $x_i=y_i$ 인 자료에 대해서, 속성이 영향을 주는 속성(p)일 경우 기본적인 가중치 1만큼을 더 가중하게 되어 결과적으로 유사도는 2가 증가한다. 이는 클러스터링에 영향을 주는 속성에 대해서 가중치를 부여함으로써 보다 정밀한 클러스터링 결과를 계산한다.

● 유사도 계산 예

	A_1	A_2	A_3	A_4	A_5
R_1	8	17	1	44	22
R_2	10	11	46	1	24
R_3	10	9		2	1
R_4	12	10			
R_5	7				
T	1	1	0	0	1

그림(8) 유사도 계산을 위한 속성의 성분 빈도 측정

그림(8)은 $N=\{N_1,...,N_{47}\}$개의 데이터에 대하여 속성 $A=\{A_1,...,A_5\}$이 존재하며, 각각의 속성들은 $\{A_1=5\},\{A_2=4\},\{A_3=2\},\{A_4=3\},\{A_5=3\}$개의 성분으로 구성되있다.

또한 속성에 대한 모든 성분들의 합은 47로써 모든 데이터들을 포함한다. 이때, 47개의 데이터에 대하여 5%는 2(≒2.35)이다. 즉, 데이터들의 속성에 대하여 성분 값이 2보다 작은 클러스터링에 영향을 주지 않는거으로 간주하고 값들을 제외한다. 속성 A_3의 경우, 데이터 성분 값이 2보다 작은 1(R_1)을 제외할 경우 A_3는 한 개의 성분(R_2)만이 남게 된다. 결국 A_3는 클러스터에 영향을 주지 않는 요소(q)로 인식한다. 이러한 방식으로 속성 $\{A_1,A_2,A_5\}$의 경우는 클러스터링에 영향을 주는 속성(p)이 된다. 따라서 μ= (1+ 1+ 1)/2 + 2 = 3.5이며, 유사도 계산시 가중치는 $T=\{1,1,0,0,1\}$에 의해서 적용여부를 결정한다. 따라서, $\{A_1,A_2,A_5\}$일 경우 추가 가중치(w_i)를 적용하며 $\{A_3,A_4\}$일 경우 가중치를 적용하지 않게 된다.

■

3) 속성간 웨이트(γ)적용

추가적으로 본 논문에서 속성간의 웨이트 적용방식을 추가한다. III.1절에서 이미 설명한 $k-prototype$ 알고리즘에 적용되는 웨이트γ값의 추정이다.

일반적으로 $k-prototype$ 알고리즘에서 웨이트 값은 분산값으로 두거나 경험적으로 두게 된다. 하지만 웨이트는 결국 수치형 속성과 범주형 속성의 값에 대한 비교를 의미한다. 수치형 속성의 개수를 C_{num}이라고 하고 범주형 속성의 개수를 C_{cat}이라고 하자. 물론 단순히 웨이트 값을 $\gamma = \frac{C_{num}}{C_{cat}}$로 할 수 있다. 하지만 이는 단순 비교가 되어 보다 신뢰있는 클러스터링 수행에 의미가 약하다.

하지만 수치형 속성중 클러스터링에 영향을 주는 요소를 C_{num}^{p} 라고 하자.또한 범주형 속성중 클러스터링에 영향을 주는 요소를 C_{cat}^{p} 라고 하자.

결국,

$$\gamma = \frac{\frac{C_{num}^{p}}{C_{num}}}{\frac{C_{cat}^{p}}{C_{cat}}}$$

로 정의한다.

이는 웨이트 값을 범주형속성에 대한 수치형속성의 비로서 단순히 비교하는 것이 아닌 실제 클러스터링에 영향을 주는 요소에 대한 비율로 정의함으로서 웨이트(γ) 선정에 대한 경험적이 아닌 보다 신뢰있는 방법을 제안한다.

4) 개선된 $k-mode$ 알고리즘

$k-mode$ 개선 알고리즘

→ 입력 : n개의 범주형 데이터베이스, 초기치 선정 k의 개수

Step 1.

1.1. 입력된 데이터에 대한 μ계산 및 T 설정

1.2. 초기치 k 선정

Step 2.

2.1 전체 객체와 k와의 T에 따른 유사도 계산

2.2 유사한 클러스터에 객체 할당

Step 3.

3.1 클러스터의 변화가 없을 때까지 (Step 2) 반복수행

3.2 변화가 없을 대 정확도 계산 및 출력

→ 출력 : 최적 배정 k 클러스터

IV. 실험결과 및 분석

본 논문에서는 분할 클러스터링분야에서 많은 실험이 이루어지고 있는 UCI Machine Learning의 small Soybean Database 와 Mushroom Database를 대상으로 시뮬레이션 하였다. IV.2절에서는 Soybean Database에 대하여 비교실험을 하였으며, IV.3절에서는 Mushroom Database를 비교실험 하였다. 그리고 IV.3절에서는 현제 많은 연구가 진행중인 다른 연구들과의 결과들에 대해서도 비교하였다.

실험에서는 모든 시뮬레이션을 R-program(ver 2.3.1)를 통해 구현하였다. R은 통계소프트웨어로서 많은 개발이 진행중인 공개소프트웨어이다. R-program은 현제 홈페이지를 통해 프리웨어 버전으로 다운로드(http://www.r-project.org) 할 수 있으며, 활성화된 커뮤니티를 통해 많은 정보가 교류중이다.

1. Database 실험

1) Soybean Database

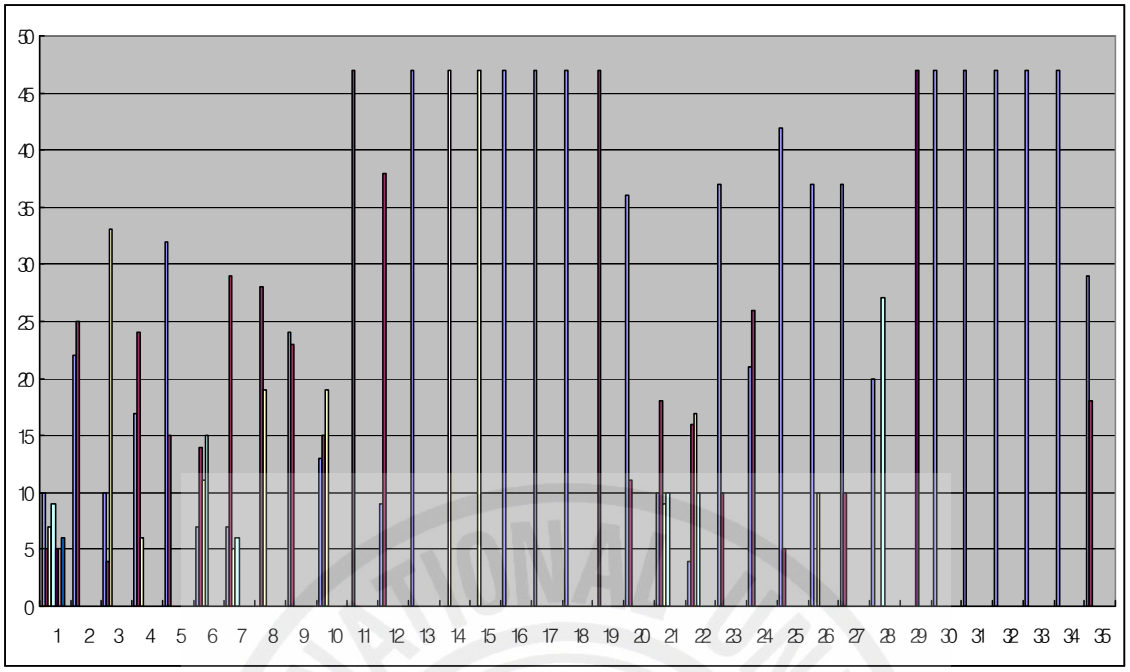
Soybean Database는 $k-mode$ 및 $k-means$ 알고리즘의 테스트 데이터베이스로 매우 유명하다. 데이터베이스는 총 47개의 레코드데이터에 대하여 35개의 속성으로서 이루어져 있으며, 속성들은 범주형 속성인 줄기의 상태, 색깔, 크기 등으로 이루어져 있다.

표(1) Soybean Database의 속성 성분

	속성	속성의 성분	수치형표현
1	date	April, may, June, July, August, september, october, ?.	0,1,2,3,4,5,6,7
2	plant-stand	normal, lt-normal, ?.	0,1,2
3	precip	lt-norm, norm, gt-norm, ?.	0,1,2
4	temp	lt-norm, norm, gt-norm, ?.	0,1,2
5	hail	yes, no, ?.	0,1,2
6	crop-hist	diff-lst-year, same-lst-yr, same-lst-two-yrs, same-lst-sev-yrs, ?.	0,1,2,3,4
7	area-damaged	scattered, low-areas, upper-areas, whole-field, ?.	0,1,2,3,4
8	sevrity	minor, pot-severe, severe, ?.	0,1,2,3
9	seed	none, fungicide, other, ?.	0,1,2,3
10	germination	90-100%, 80-89%, lt-80%, ?.	0,1,2,3
11	plant-growth	norm, abnorm, ?.	0,1,2
12	leaves	norm, abnorm.	0,1

13	leafspots-halo	absent, yellow-halos, no-yellow-halos, ?.	0,1,2
14	leafspots-marg	w-s-marg, no-w-s-marg, dna, ?.	0,1,2,3
15	leafspot-size	lt-1/8, gt-1/8, dna, ?.	0,1,2,3
16	leaf-shread	absent, present, ?.	0,1,2
17	leaf-malf	absent, present, ?.	0,1,2
18	leaf-mild	absent, upper-surf, lower-surf, ?.	0,1,2,3
19	stem	norm, abnorm, ?.	0,1,2
20	lodging	yes, no, ?.	0,1,2
21	stem-cankers	absent, below-soil, above-soil, above-sec-nde, ?.	0,1,2,3,4
22	canker-lesion	dna, brown, dk-brown-blk, tan, ?.	0,1,2,3,4
23	fruiting-bodies	absent, present, ?.	0,1,2
24	external decay	absent, firm-and-dry, watery, ?.	0,1,2,3
25	mycelium	absent, present, ?.	0,1,2
26	int-discolor	none, brown, black, ?.	0,1,2,3
27	sclerotia	absent, present, ?.	0,1,2
28	fruit-pds	norm, diseased, few-present, dna,?.	0,1,2,3
29	fruit spots	absent, colored, brown-w/blk-specks, distort, dna, ?	0,1,2,3,4,5
30	seed	norm, abnorm, ?.	0,1,2
31	mold-groth	absent, present, ?.	0,1,2
32	seed-discolor	absent, present, ?.	0,1,2
33	seed-size	norm, lt-norm, ?.	0,1,2
34	shriveling	absent, present, ?.	0,1,2
35	roots	norm, rotted, galls-cysts, ?.	0,1,2,3

표(1)은 soybean 데이터의 속성들을 보여주고 있다. 총 35개의 속성에 대하여 정리하였으며 속성들의 범주형성분값을 수치형으로 변환하여 실험하였다. 이미 다른 이들의 연구를 통해 soybean 데이터는 $K=4$로 구성되어 있으며, 각각은 $D(Diaporthe\ Stem\ Canker)$, $C(Charcoal\ Rot)$, $R(Rhizoctonia\ Root\ Rot)$, $P(Phytophtoora\ Rot)$로 구분되어 진다. 따라서 본 실험에서도 μ 값에 따른 초기 k의 결정에 대하여 k_1, k_2, k_3, k_4를 지정하여 클러스터링을 수행하였다.



그림(9) Soybean Database 에 대한 속성별 밀집도

그림(9)는 Soybean Database에 대하여 μ값의 결정을 위한 모든 속성의 성분에 대한 빈도를 표현하였다. 그림에서와같이 속성의 빈도가 클러스터링 단계에 영향을 줄 수 있는 속성의 개수는 21(p)개, 속성의 성분이 1개 이하로서 클러스터링에 영향을 주지 않는 개수는 14(q)개이다. 따라서 $\mu = p/2 + q = 24.5$로 계산이 되며, 약 $\mu = 25$로 계산된다. 그리고 이를 기준으로 k값들을 선정한다.

(i)

	k_1	k_2	k_3	k_4
D	2	4	3	1
C	8	2		
R		4	6	
P	12			5

(ii)

	k_1	k_2	k_3	k_4
D		10		
C	1	9		
R			10	
P	8			9

(iii)

	k_1	k_2	k_3	k_4
D		10		
C	1			
R			10	
P				9

그림(10) 기존 $k-mode$ 알고리즘의 최적화 과정

그림(10)은 기존 $k-mode$ 알고리즘의 최적화 진행과정을 보여주고 있다. (i)단계에서 초기 선정된 k들에 대해 유사도 측정에 의한 클러스터결과를 보여주고 있으며, (ii)단계는 (i)단계의 매트릭스의 반복수행을 통해 클러스터링을 진행한 결과이다. 반복수행으로 인해 클러스터들은 재배정되었고 비교적 고르게 클러스터링 되었다. 이때 알고리즘의 반복횟수는 6회로 측정되었다. (iii)단계는 (ii)단계의 결과에 대한 정확도 측정을 위한 매트릭스로서 17개의 데이터가 오분류 되었으며 63.8%의 정확도를 보였다. 물론 이는 랜덤하게 선택되는 k들에 대해서 많은 영향을 받으므로 위 실험을 다시 시행하면 다시 랜덤하게 선택되어진 k에 의해서 더 높은 결과를 보여줄 수도 있으며, 혹은 더 낮은 결과를 보여줄 수도 있다.

(i)

	k_1	k_2	k_3	k_4
D		10		
C			10	
R	9			1
P	4			13

(ii)

	k_1	k_2	k_3	k_4
D		10		
C			10	
R	10			
P				17

(iii)

	k_1	k_2	k_3	k_4
D		10		
C			10	
R	10			
P				17

그림(11) $k-mode$ 개선 알고리즘의 최적화 과정

그림(11)은 개선된 $k-mode$ 알고리즘의 진행과정을 표현하였다. (i)단계에서 이미 잘 선택되어진 k로 인해 초기 클러스터링이 기존 $k-mode$ 알고리즘에 비해 잘 선택되어지고 있음을 볼 수 있다. (ii)단계에서는 반복수행을 통한 클러스터의 이동을 의마하며, 이 때 반복횟수는 3회 반복되었다. (iii)단계는 최적의 매트릭스를 계산한 결과이다. 위 그림에서는 100%의 정확도를 보여주고 있다.

실험에 대한 좀더 정확한 테스트를 위하여 기존 $k-mode$ 알고리즘과 제안하는 $k-mode$ 개선 알고리즘을 비교하였다. 실험은 10회, 100회, 1000회 실험을 반복 진행하였으며, 클러스터링의 정확도 및 알고리즘의 반복 수행횟수를 측정하였다.

%	1	2	3	4	5	6	7	8	9	10
100				1						
98										
95				1			1			
90										
87										
83										
80										
70			1	1	1					
60				4						
50										
40										

그림(12) Soybean Database에 대한 기존 $k-mode$ 알고리즘의 10회 실험

%	1	2	3	4	5	6	7	8	9	10
100		3	4	2	4	1	1	1		
98										
95		2	2	3	1	2		1		1
90		1		1	3	2				
87										
83										
80										
70		2	12	4	2	5			1	1
60		1	5	10	4	4	2		1	1
50		1	4	2	1	1				
40		1								

그림(13) 기존 Soybean Database에 대한 기존 $k-mode$ 알고리즘의 100회 실험

%	1	2	3	4	5	6	7	8	9	10
100		16	37	69	44	22	20	7	5	3
98										
95		11	20	16	19	14	11	2	2	3
90		1	7	8	22	12	4	3	1	
87				3	2		2			
83										1
80										
70		24	86	57	51	21	13	6	2	6
60		10	85	76	58	31	19	7	2	8
50		3	7	18	13	3		1		
40			1	4	1					

그림(14) Soybean Database에 대한 기존 $k-mode$ 알고리즘의 1000회 실험

그림(12), 그림(13), 그림(14)는 기존 $k-mode$ 알고리즘에 대하여 10회, 100회, 1000회의 실험결과를 정확도에 의한 반복수행 횟수에 대하여 정리한 표이다.

표(2) 기존 $k-mode$ 알고리즘의 실험결과표

	10회 실험			100회 실험			1,000회 실험		
	full	*best*	*good*	*full*	*best*	*good*	*full*	*best*	*good*
100	1	1	1	16	16	13	223	223	166
98									
95	2	2	1	12	12	8	98	98	66
90				7	7	5	58	58	38
87							7	7	5
83							1		
80									
70	3			27			266		
60	4			28			296		
50				9			45		
40				1			6		
total	10	3	2	100	35	26	1,000	386	275

표(2)에서 *full*은 모든 데이터에 대한 실험반복횟수이며, *best*의 경우는 잘 클러스터링이 되었다고 할 수 있는 87%이상의 정확도를 보인 경우이다. 그리고 *good*의 경우는 87% 이상의 자료에 대하여 알고리즘의 반복이 5회 이하로서 *best*의 경우에서 비교적 빠른 결과를 보여준 실험의 횟수이다.

다음으로 개선 $k-mode$ 알고리즘의 실험하였다. 마찬가지 방식으로 $k-mode$ 개선 알고리즘에 대하여 10회, 100회, 1000회 반복 수행하였다.

%	1	2	3	4	5	6	7	8	9	10
100		1	2	2						
98										
95			2	1			1			
90										
87										
83										
80										
70				1						
60										
50										
40										

그림(15) Soybean Database에 대한 $k-mode$ 개선 알고리즘의 10회 실험

%	1	2	3	4	5	6	7	8	9	10
100		14	20	6	17					
98										
95		8	14	7	1	1				
90		2		1						
87		1								
83										
80										
70				5	2					
60				1						
50										
40										

그림(16) Soybean Database에 대한 $k-mode$ 개선 알고리즘의 100회 실험

%	1	2	3	4	5	6	7	8	9	10
100		112	297	119	120	5	3	2	1	
98										
95		48	106	85	3	10	1			
90		7	21	5	8					
87		12								
83										
80										
70				15	5					
60				6	3	3	1			1
50				1						
40										

그림(17) Soybean Database $k-mode$ 개선 알고리즘의 1000회 실험

	10회 실험			100회 실험			1,000회 실험		
	full	*best*	*good*	*full*	*best*	*good*	*full*	*best*	*good*
100	5	5	5	57	57	57	659	659	648
98	4	4	3	31	31	30			
95				3	3	3	253	253	242
90				1	1	1	41	41	41
87							12	12	12
83									
80									
70	1						20		
60				7			14		
50				1			1		
40									
total	10	9	8	100	92	91	1,000	965	943

표(3) $k-mode$ 개선 알고리즘의 실험결과표

표(3)를 보면 기본적으로 10회 실험의 경우 87%이상의 정확도는 9회로 나타났으며, 100회 실험의 경우 92회, 1,000회 실험의 경우 965회로 클러스터가 잘 되었다. 특히 알고리즘의 반복이 5회 이하로 나타나는 *good*의 경우는 각각 8회, 91회, 943회로 나타나고 있으며 이를 기존 $k-mode$ 알고리즘과 비교하였다.

표(4) 기존 알고리즘과 개선 알고리즘의 실험결과 비교

	10회 실험		100회 실험		1,000회 실험	
best	기존 $k-mode$	개선 $k-mode$	기존 $k-mode$	개선 $k-mode$	기존 $k-mode$	개선 $k-mode$
>87%	3	9	35	92	386	965
good	기존 $k-mode$	개선 $k-mode$	기존 $k-mode$	개선 $k-mode$	기존 $k-mode$	개선 $k-mode$
>87%	2	8	26	91	275	943

표(4)에서는 Soybean Database 에 대하여 기존 $k-mode$와 $k-mode$ 개선 알고리즘의 실험결과를 정리하였다. 모든 실험에서 기존 $k-mode$ 알고리즘 보다 $k-mode$ 개선 알고리즘이 월등히 우수한 클러스터링 결과를 보여준다.

표(5) 기존 알고리즘과 개선 알고리즘의 평균 반복 횟수

	10회 실험		100회 실험		1,000회 실험	
best	기존 $k-mode$	개선 $k-mode$	기존 $k-mode$	개선 $k-mode$	기존 $k-mode$	개선 $k-mode$
	4.3(3)	3.7(9)	4.4(35)	3.4(92)	4.5(386)	3.4(965)
good	기존 $k-mode$	개선 $k-mode$	기존 $k-mode$	개선 $k-mode$	기존 $k-mode$	개선 $k-mode$
	4(2)	3.3(8)	3.6(26)	3.3(91)	3.9(275)	3.3(943)

표(5)는 알고리즘의 수행횟수를 비교하였다. 모든 실험에 대하여, 평균 반복수행 횟수가 낮음을 알 수 있으며 제안하는 알고리즘이 기존 알고리즘에 보다 클러스터링의 결과의 정확도가 향상되었다. $k-mode$ 개선 알고리즘이 보다 효율적이다.

2) Mushroom Database.

Mushroom Database의 모든 데이터들은 8,124개의 레코드로 구성되어있으며, 각각의 레코드들은 22개의 속성을 표현한다. 각 속성들은 물리적인 색, 모양, 크기, 냄새 등에 대하여 정의되어 있으며, 전체의 데이터 중 식용 가능한 4,208개의 버섯과 3,916개의 독버섯으로 구성($K=2$)되어 있다.

실험에서는 $\mu=13$으로 계산되었으며, 실험은 10회, 100회 반복 수행하였다.

(표 6) Mushroom Database의 속성 성분

	속성	속성의 성분
1	cap-shape	bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
2	cap-surface	fibrous=f, grooves=g, scaly=y, smooth=s
3	cap-color	rown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
4	bruises	bruises=t, no=f
5	odor	almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
6	gill-attachment	attached=a, descending=d, free=f, notched=n
7	gill-spacing	close=c, crowded=w, distant=d
8	gill-size	broad=b, narrow=n
9	gill-color	black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
10	stalk-shape	enlarging=e, tapering=t
11	stalk-root	bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
12	stalk-surface-above-ring	ibrous=f, scaly=y, silky=k, smooth=s
13	stalk-surface-below-ring	ibrous=f, scaly=y, silky=k, smooth=s
14	stalk-color-abobe-ring	brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
15	stalk-color-below-ring	brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
16	veil-type	partial=p, universal=u
17	veil-color	brown=n, orange=o, white=w, yellow=y
18	ring-number	none=n, one=o, two=t
19	ring-type	cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
20	spore-print-color	black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
21	population	abundant=a, clustered=c, numerous=n, scattered=s,several=v,solitary=y
22	habitat	grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

%	1	2	3	4	5	6	7	8	9	10
100										
98										
95										
90										
87					2				1	
83										
80										
70				1		1				
60				1	1		1			
50					1	1				
40										

그림(18) Mushroom Database에 대한 기존 $k-mode$ 알고리즘의 10회 실험

%	1	2	3	4	5	6	7	8	9	10
100										
98										
95										
90										
87				1	1	2	1		4	
83					5			1	3	
80			5					2		
70				6			2			
60				11		20	1	4		1
50					4	3	1		10	
40				2		10				

그림(19) Mushroom Database에 대한 기존 $k-mode$ 알고리즘의 100회 실험

표(7) 기존 $k-mode$ 알고리즘의 실험결과표

	10회 실험			100회 실험		
	full	*best*	*good*	*full*	*best*	*good*
100						
98						
95						
90						
87	3	3	2	9	9	2
83				9		
80				7		
70	2			8		
60	3			37		
50	2			18		
40				12		
total	10	3	2	100	9	2

그림(18), 그림(19)는 Mushroom Database에 대한 10회와 100회 반복 수행을 보여준다. 그리고 실험의 결과에서는 좋은 결과를 보여주고 있지 않음을 알 수 있다. 이에 대한 정리는 표(7)에서 보여준다.

100회 수행 시 *good*(정확도가 87%이상, 반복횟수가 5회 이하)의 경우도 9회로 나타나며, 데이터의 9%만이 좋은 클러스터링을 보여준다.

%	1	2	3	4	5	6	7	8	9	10
100										
98										
95										
90				1						
87				2	1	1				
83				1	1					
80										
70				1						
60					1					
50			1							
40										

그림(20) Mushroom Database에 대한 $k-mode$ 개선 알고리즘의 10회 실험

%	1	2	3	4	5	6	7	8	9	10
100										
98										
95										
90			1			2				
87			2	3	9	7	2			
83				3			2		2	1
80			9	4	1			1		2
70				4	4			2		
60			2	8	1	3	2	1		
50			3	6	5	3	3	1	1	
40										

그림(21) Mushroom Database에 대한 $k-mode$ 개선 알고리즘의 100회 실험

표(8) $k-mode$ 개선 알고리즘의 실험결과표

	10회 실험			100회 실험		
	full	*best*	*good*	*full*	*best*	*good*
100						
98						
95						
90	1	1	1	3	3	1
87	4	4	3	23	23	14
83	2			8		
80				17		
70	1			10		
60	1			17		
50	1			22		
40						
total	10	5	4	100	26	15

그림(20), 그림(21)은 $k-mode$ 개선 알고리즘을 적용한 결과이다. 위 표(8)은 반복 실험별 클러스터링에 대한 전체 데이터와 87%이상으로 잘 클러스터링 되어진 *good*을 구분하였다. 표(7)과 표(8)을 비교하면 10회 100회 반복실험에 대하여 개선 알고리즘이 더 우수한 결과를 보여준다.

표(9) 기존 알고리즘과 개선 알고리즘 평균 반복 횟수

	10회 실험		100회 실험	
best	기존 $k-mode$	개선 $k-mode$	기존 $k-mode$	개선 $k-mode$
	5.6(3)	4.4(5)	6.1(9)	5.1(26)
good	기존 $k-mode$	개선 $k-mode$	기존 $k-mode$	개선 $k-mode$
	5(2)	4(4)	4.5(2)	4.4(15)

표(9)는 기존 알고리즘과 개선 알고리즘의 평균 알고리즘 반복 횟수를 정리한 것이다. 전체 데이터를 대상으로 하는 반복횟수에서는 $k-mode$ 개선 알고리즘이 기존 $k-mode$ 알

고리즘 보다 좋은 결과를 보여주고 있었다. 100회 반복의 *good*의 경우 기존 알고리즘에 비해 개선된 알고리즘이 반복수행수치가 좀 더 높게 나타났으나 잘 된 클러스터의 수는 31개로 기존방식의 9개보다 훨씬 우수한 결과를 보여준다.

2. 연구 비교

최근 기존의 $k-mode$ 알고리즘을 개선한 다양한 알고리즘이 연구되어지고 있다. 이에 대해 본 절에서는 연구되어 발표되어진 실험결과들과 본 논문에서 제안하는 알고리즘들의 클러스터링 결과를 비교하였다.

- *Initial point refineing algorithm* (Ying S. 2002)
- $k-$*representative algorithm* (Van H. 2004)

위의 알고리즘들은 $k-mode$ 방식을 개선한 알고리즘으로 기존의 알고리즘을 잘 개선한 알고리즘이다.

실험에 사용된 데이터는 이미 IV장에서 소개된 Soybean Database 이다.

● *Initial point refineing algorithm* 과의 비교

초기치 선정방법을 개선한 *Initial point refineing algorithm*은 전체 데이터에 대한 *subset*으로부터 클러서트링을 수행하게 된다. *subset*은 *bradly et al*을 기반으로 클러스터링을 수행하게 된다.(bradly P. 1998) 결국 *subset*의 선정에 따라 클러스터링에 영향을 받게 되므로 *subset*의 선정이 매우 중요하다.

Initial point refineing algorithm

step 1 : // sub-sampling
1.0 $CM=0$
  1.1 For $i=1,...,J$
    1.1.1 Let S_i be a small random sub-sample set of Data
    1.1.2 Let SP_i be a randomly seleted K sample from S_i
    1.1.3 $CM_i = Clustering(SP, S_i, K)$
    1.1.4 $CM = CM \cup CM_i$

step 2 : // Refinement
  2.0 $FMS=0$
  2.1 For $i=1,...,J$

2.1.1. Let $FM_i = Clustering(CM_i, CM, K)$

2.1.2. Let $FMS = FMS \cup FM_i$

step 3 : // Selection

3.1. $Let FM = ArgMin_{FM_i}\{Distortion(FM_i, CM)\}$

3.2 Return (FM)

선택되어진 *sudbset*에 대하여 반복수행을 통해 클러스터들을 병합 삭제의 과정을 통해 클러스터는 최적의 결과를 보여준다.

표(10) *Initial point refineing algorithm*과의 클러스터링 결과 비교

정확도	*No refineing algorithm*	*Initial point refineing algorithm*	$k-mode$ 개선 알고리즘
0.98	5	14	10
0.94	2		8
0.89	2		1
0.77	3		
0.70		5	1
0.68	5		
0.66	3	1	

표(10)에서 *Initial point refineing algorithm*은 좋은 결과를 보여주고 있다. 98%의 정확도를 보여주는 것이 14회 이상으로서 기존 $k-mode$ 알고리즘에 비해 *Initial point refineing algorithm*은 좋은 결과를 보여주고 있다. 물론 제안하는 $k-mode$ 알고리즘보다도 우수한 성능을 보여주고 있다. 하지만 전체적으로 보면 *Initial point refineing algorithm*은 클러스터링의 결과가 아주 우수하거나 혹은 상당히 낮은 결과로 측정이 되었다. 비록 제안하는 알고리즘은 98%이상의 정확도에서는 *Initial point refineing algorithm* 보다 낮은 수치를 보였으나 정확도 94%이상 에서는 18회로 제안하는 알고리즘이 우수하게 측정되었다. 또한, 89% 이상의 정확도를 보인 것은 전체 20회 실험중 19회로 *Initial point refineing algorithm* 알고리즘보다 우수한 결과를 보였다.

● $k-representative\ \ algorithm$ 과의 비교

$k-representative\ \ algorithm$ 은 클러스터의 센터를 정하는 방법을 개선한 알고리즘으로서 클러스터링의 센터를 클러스터링 센터의 빈도를 이용하여 계산하는 방법이다. 클러스터링 C는 $Q=q_1,...,q_m$로 정의 되었을 때, $q_j=\{(c_j,f_{c_j})|c_j\in D_j\}$로 표현하게 된다.

$k-representative\ \ algorithm$

1. Initialize a k-partition of D randomly

2. Calulate k-representatives, one for each cluster.

3. For each X_i calculate the dissimilarites
   $d(X_i,Q_l),\quad l=1,...,k$
   Reassign X_i to cluster C_i
   Update both Q_l, $Q_{l'}$

4. Repeat Step 3 until no objet has changed clusters after a full cycle test of the whole data set.

표(11) $k-representative\ algorithm$과의 클러스터링 결과 비교

정확도	$k-representative$ $algorithm$	$k-mode$ 개선 알고리즘
1 ~ 0.978	519($good$)	659
0.977 ~ 0.936	87($good$)	253
0.935 ~ 0.893	80($good$)	41
0.892 ~ 0.851	89	12
0.850 ~ 0.808	86	
0.807 ~ 0.765	94	15
0.764 ~ 0.723	28	5
0.722 ~ 0.680	8	12
0.679 ~ 0.638	7	2
0.638 ~ 0.531	2	1

표(11)에서 $k-representative\ \ algorithm$은 정확도가 0.893 이상인 결과에 대해서 $good$으로 보고있으며 686회의 $good$인 결과를 보여주며 전반적으로 좋은 결과를 보여주고 있음을 알 수 있다. 본 논문에서 제안하는 알고리즘 역시 $good$이상의 결과는 953회로 나타났으며, 0.722~0.680의 범위에서의 정확도만이 낮게 측정되었으며 전반적으로 $k-representative\ \ algorithm$ 보다 우수한 결과를 보여주고 있다.

V. 결론 및 제안

최근 범주형 데이터에 대한 클러스터링 연구는 많은 연구가 진행되고 있으며 특히 분할방식을 사용하는 알고리즘은 범주형 데이터에 좋은 성능을 보여주고 있으며 여러 가지 응용통계 프로그램에도 적용하기 좋은 방식이다. 본 논문에서는 범주형 데이터를 좀 더 정확하고 빠르게 클러스터링 할 수 있도록 $k-mode$ 알고리즘을 개선시킨 $k-mode$ 개선 알고리즘 방식을 제안하였다. 먼저 초기치 k의 선정시 클러스터의 정확도 향상 위하여 클러스터링에 영향을 주는 속성과 그렇지 않은 속성들을 구분하였다. 그리고 그에 따른 μ값의 선정을 통하여 클러스터내의 유사도 증가와 클러스터러들 간의 상이도를 크게 유지하게 함으로써 기존 랜덤하게 선택되는 방식보다 클러스터링 결과의 정확도의 향상을 보였다.

아울러 기존 클러스터링 방식의 유사도 측정방식도 개선하였다. 기존 방식은 단순 비교를 통한 거리기반방식으로 클러스터링에 중요한 요소가 있을 경우 이를 무시하게 된다. 하지만 제안된 방식의 유사도 측정방식은 특정한 속성에 가중치를 줌으로서 k들간의 유사도 계산값들이 커지는 효과를 주게 되어 동일 유사도값 발생의 감소와 더불어 좀 더 정확한 클러스터링을 할 수 있었다. 그리고 가중치 적용시 따로 가중치를 계산하는 것이 아닌 초기 μ값 결정시 클러스터링에 영향을 주는 요소에 대한 가중치적용방식을 사용함으로서 알고리즘의 수행에 영향을 적도록 하였다.

IV장의 실험결과에서도 볼 수 있듯이 제안하는 알고리즘은 기존 $k-mode$ 알고리즘보다 우수한 정확도를 보여주었으며 아울러 Mushroom Database와 같이 레코드가 많은 대용량 데이터에 대해서도 효과적임을 보여주었다.

그리고 실험에서는 추가적으로 알고리즘의 반복수행도 측정하였다. 실험에서 알고리즘의 수행시간을 줄이는 것은 중요한 문제이다. 본 논문에서는 최적의 초기 k값의 선정으로 기존방식 보다 알고리즘의 반복수행 횟수의 감소를 보여줄 것을 예상했지만 예상에 비해 큰 감소를 보여주지는 않았다. 물론 기존방식보다는 적은 반복으로 좋은 결과를 보여주었으며 알고리즘의 정확도 향상에 영향을 더 주었다.

VI. 참고문헌

Anil, 2001, *k−modes Clustering.* − Journal of Classification 18: 35-55 (2001)

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

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

Han·Kamber, *Data Mining Concepts and Techniques.* − Jiawei Han , Micheline Kamber

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

허명희, 2004, *k-평균 군집화와 재현성 평가 및 응용* − 응용통계연구.제17권 1호. 2004년 pp 135-144

이상용, 2002, *대용량 데이터 처리를 위한 하이브리드형 클러스터링 기법* − 두뇌한국21.2002

Michalski, 1983, *Automated construction of classifications: conceptual clustering versus numerical taxonomy* − IEEE trans. pattern Anal. Machine Intell. 5(4), 396-410

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

Ng,R.t. and Han,J., 1994, *Efficient and Effective Clustering Methods for Spatial Data Mining* – Proceedings of the 20th VLDB Conference, Santiago, Chile, pp.144-155

Van H, 2004, *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

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 Instituite of Technology Harbin 150001.P.R.China