▩ 연구논문

차원간 연관관계와 하이퍼그래프 분할법을 이용한 범주형 속성을 가진 데이터의 클러스터링

- Clustering Data with Categorical Attributes Using Inter-dimensional Association Rules and Hypergraph Partitioning -

이 성 기*
Lee, Sung Gi
윤 덕 균**
Yun, Deok Kyun

Abstract

Clustering in data mining is a discovery process that groups a set of data such that the intracluster similarity is maximized and intercluster similarity is minimized. The discovered clusters from clustering process are used to explain the characteristics of the data distribution. In this paper we propose a new methodology for clustering related transactions with categorical attributes. Our approach starts with transforming general relational databases into a transactional databases. We make use of inter-dimensional association rules for composing hypergraph edges, and a hypergraph partitioning algorithm for clustering the values of attributes. The clusters of the values of attributes are used to find the clusters of transactions. The suggested procedure can enhance the interpretation of resulting clusters with allocated attribute values.

1. 서론

데이터마이닝(data mining)에 있어서의 클러스터링(clustering)은 클러스터 내의 유사성을 최대화하면서 클러스터간의 유사성을 최소화하도록 데이터를 집단화 하는 과정이다. 이렇게 발견된 클러스터들은 데이터 분포의 특성을 설명하는데 사용된다. 예를 들어, 경영적인 측면에 있어서 클러스터링은 서로 다른 고객집단을 특성화하여 목표마케팅(target marketing)을 수행하거나 클러스터의 특징을 바탕으로 고객의 구매행태를 예측하는데 이용된다.

통계학을 바탕으로 한 전통적인 클러스터링 기법들은 데이터 속성(Attribute)들의 차원(dimension)이 높고 많은 양의 데이터를 다루는 경우에는 계산시간이 너무 많이 걸리기 때문에, 대용량 데이터베이스를 대상으로 하는 새로운 클러스터링 방법들이 연구되어져 왔다. 그러나 이러한 클러스터링 방법들은 범주형(categorical)데이터가 아닌 수치(numeric)데이터를 다루기 위해 개발된 방법들이며, 최근에 와서 대용량 데이터베이스의 범주형 데이터 중심의 클러스

* 한양대학교 산업공학과 박사과정
** 한양대학교 산업공학과

터링 방법들이 연구되어지고 있다.

Huang[10]은 수치형 데이터의 클러스터링에 이용되는 k-means방법을 수정하여, 범주형 데이터 각각의 속성값이 서로 일치하면 거리를 0으로 하고 서로 일치하지 않으면 거리를 1으로 하는 k-modes방법을 제안하였다. 이 방법은 범주형 데이터를 수치형 데이터처럼 이용할 수 있는 장점이 있으나, 속성값들 간의 차이를 무조건 0과1로 이분화 함으로써 수치형 데이터에서 발생하는 차이에 비해 범주형 데이터에서 발생하는 차이가 훨씬 더 많은 영향을 미치게 되는 점과 범주형 데이터의 중심을 최빈치(mode)로 설정함으로써 최빈치가 아닌 속성값을 가지는 데이터는 모두 중심에서 멀리 떨어져 있는 것으로 판정되는 단점이 있다. 그래서 범주형 데이터를 수치형처럼 계산하는 방법이 아닌 새로운 방법들이 나타나게 되었다. Gibson et al[5]은 범주형 속성값 들간에 가중치를 할당하고 전파해가는 동적인 시스템을 구성하여 클러스터링을 시도하였으나 이 방법은 수렴이 보장되지 않으며, Zhang et al[12]은 해의 수렴 문제를 해결하였으나 Gibson et al 과 마찬가지로 데이터의 레코드들을 클러스터링하는 것이 아닌 각각의 범주형 속성의 속성값들을 이분화하는 클러스터링을 수행하였다.

Han et al[7]은 연관관계(association rules)에 의해 발생하는 빈발항목집합(frequent itemset)과 이를 바탕으로 한 하이퍼그래프 분할법(hypergraph partitioning)을 이용하여 장바구니 데이터베이스(market basket database)에서 고객들의 트랜잭션(transaction)을 클러스터링하는 방법을 제안하였다. 그 밖에 ROCK[6]과 CACTUS[4]방법이 제안되었으나 ROCK은 샘플링을 통해 클러스터링을 수행함으로써 샘플링에 따라 클러스터가 달라질 수 있고 구성된 클러스터의 특성을 설명할 수가 없으며, CACTUS는 정확성이 떨어지는 단점이 있다.

본 논문은 Han et al[7] 이 제안한 방법을 기반으로 일반적인 여러 개의 범주형 속성들을 가지는 데이터베이스의 레코드(record)들에 대한 클러스터링을 수행하는 알고리듬을 제안하도록 한다. 이 방법은 데이터베이스의 데이터들을 클러스터링 할 수 있도록 알맞은 형태로 변환한 후, 차원간 연관관계를 이용한 하이퍼그래프 분할법을 수행하여 범주형 데이터의 속성값 들을 클러스터링하고 이를 통해 레코드들을 클러스터링한다.

2. 차원간 연관관계(Inter-dimensional Association Rules)

하이퍼그래프 분할법을 이용하여 범주형 속성을 가지는 데이터베이스의 레코드를 클러스터링하기 위해서 먼저 범주형 데이터를 가지는 데이터베이스를 장바구니 데이터베이스 형태인 트랜잭션 데이터베이스로 변환한 후 다차원 연관관계를 찾아낸다.

2.1 범주형 데이터베이스의 변환

범주형 데이터란 수치형이 아닌 데이터를 의미한다. 예를들어, 범주형 속성이란 가격이나 구매 수량과 같은 연속적인 값을 가지는 수치가 아니라, 직업(세일즈맨, 대학생, 연구원, …), 전공(경영학, 산업공학, 체육학, …), 색깔(검정색, 흰색, 노랑색, …)등과 같이 임의의 유한한 속성값의 집합을 가지는 속성이다. 데이터가 A_1, A_2, …, A_m인 m개의 속성을 가지면, 각 속성의 영역(domains)은 $DOM(A_1)$, $DOM(A_2)$, …, $DOM(A_m)$ 으로 나타내며 영역에서 가지는 속성값은 유한하고 순위가 정해지지 않은 값이다. 레코드는 하나의 속성에 대하여 하나의 속성값 만을 가지게 된다. 범주형 데이터베이스는 이러한 여러 개의 범주형 속성들만을 가지는 데이터베이스이다. 범주형 데이터베이스의 레코드들을 클러스터링하기 위하여 먼저 <그림 1>과 같이 원래의 데이터들을 장바구니 데이터 분석 문제와 같은 형태로 변환한다.

<그림 1>은 컴퓨터를 구매한 고객에 관한 정보를 담고 있는 데이터베이스를 간략하게 예로 든 것이다. 실제 데이터베이스는 더 많은 속성을 가지고 있지만 여기서는 직업, 주거지, 구매품 세 가지 속성을 가진다고 가정하고, 각각의 속성영역의 속성값도 실제로는 여러 가지 값을 가

범주형 속성

DB	직업 A_1	주거지 A_2	구매품 A_3
고객1	대학생	서울	노트북
고객2	대학생	경기	PC
고객3	대학생	서울	PC
고객4	대학생	경기	노트북
고객5	회사원	서울	노트북
고객6	대학생	경기	PC
고객7	대학생	서울	노트북
고객8	대학생	경기	PC
고객9	대학생	서울	노트북
고객10	회사원	서울	PC

레코드

변환

속성	직업 A_1		주거지 A_2		구매품 A_3	
속성값	대학생	회사원	서울	경기	노트북	PC
코드	1111	1112	1211	1212	1311	1312
T1	1	0	1	0	1	0
T2	1	0	0	1	0	1
T3	1	0	1	0	0	1
T4	1	0	0	1	1	0
T5	0	1	1	0	1	0
T6	1	0	0	1	0	1
T7	1	0	1	0	1	0
T8	1	0	0	1	0	1
T9	1	0	1	0	1	0
T10	0	1	1	0	0	1

트랜잭션

원래의 데이터베이스 Market Basket 형태의 데이터베이스

<그림 1> 차원간 연관관계 생성을 위한 데이터베이스의 변환

질 수 있지만 각각 두 개의 속성값 만을 가진다고 가정한다. 한 고객을 하나의 트랜잭션으로 볼 수 있으며, 각각의 속성에서 가질 수 있는 속성값이 장바구니 데이터에서의 항목(item)이 된다. 그러므로 전체 항목의 수는 속성영역의 크기를 모두 더한 값이 된다. 변환된 데이터베이스 세 번째 행의 코드는 다차원 Apriori 알고리듬의 실행시 계산의 편의를 위하여 속성값을 변환시킨 코드이다. 일단 총 네 자리수로 구성하며, 코드 1112의 앞의 두 자리 11은 속성(직업)을 나타내고 뒤의 두 자리 12는 하나의 속성값(대학생)을 나타낸다. 코드를 네자리로 구성하면 11~99까지 총 89개의 속성을 표현할 수 있고, 마찬가지로 각 속성에 대하여 89개의 속성값을 표현할 수 있다. 범주형 데이터의 속성들이 가질 수 있는 속성값이 거의 무한히 많은 경우도 있을 수 있지만(날짜, ID, 주민등록번호 등) 클러스터링의 특성을 설명할 수 있도록 의미있는 분석을 하기 위해서는 이러한 속성들은 속성제거(attribute removal)나 속성일반화(attribute generalization)과정을 거쳐 적당한 수의 속성값을 가지도록 한 뒤 분석을 수행하게 되므로 89개 정도면 대부분의 속성을 표현할 수 있을 것이다. 만약, 속성의 수나 속성값이 89개를 넘어설 때에는 코드를 다섯자리나 여섯자리로 확대할 수 있다. 각 셀의 값은 트랜잭션이 항목에 해당하는 속성값을 가지면 1, 그렇지 않은면 0이 된다. 결과적으로 한 속성내의 속성값들 중에서, 속성값을 가지는 하나만 1 이 되고 나머지는 모두 0이 된다.

2.2 차원간 연관관계(inter-dimensional association rules)

연관관계는 트랜잭션에 나타나 있는 항목들간의 관계를 찾아내는 방법이다. 주로 연관관계는 하나의 속성(attribute)에 대한 관계를 찾으려고 한다. 예를 들어, 연관관계 "데스크탑 ⇒ 흑백프린터"를 찾아낼 수 있으며 다음과 같이 표현된다.

구매품(X, "데스크탑") ⇒ 구매품(X, "흑백프린터")

여기서 X는 트랜잭션 데이터베이스에서 항목을 구매하는 구매자이다. 위의 연관관계는 관계내에서 여러번 나타나는 하나의 속성(구매품)만을 포함하고 있으므로 일차원(single-dimensional) 혹은 차원내(intra-dimensional) 연관관계이다.

그러나, <그림 1>에 나타난 데이터베이스는 판매제품 뿐만 아니라 그에 관계되는 정보(직업, 주거지)를 가지는 다차원데이터베이스(multidimensional databases)이다. 데이터베이스의 각

각의 차원(dimension)을 속성(predicate)이라 하며, 여러 개의 속성을 포함하는 연관관계를 찾을 수 있게 된다. 예를 들어, 직업이 대학생이고 주거지가 서울인 사람들이 노트북을 많이 구매한다는 연관관계는 다음과 같이 표현된다.

직업(X, "대학생") ∧ 주거지(X, "서울") ⇒ 구매품(X, "노트북")

위와 같이 두개 혹은 그 이상의 차원(속성)을 가지는 연관관계를 다차원 연관관계(multi-dimensional association rules)라 한다. 또한 여러 개의 속성(직업, 주거지, 구매품)을 포함하고 속성이 연관관계 내에서 단지 한번만 나타나는 연관관계를 차원간 연관관계(inter-dimensional association rules)라고 한다[8].

위와 같은 차원간 연관관계를 찾아내는 과정은 기존의 일차원 연관관계를 찾는 과정에 약간의 제약을 추가한 문제이다. 최근에 연관관계를 찾아내는 효율을 높이기 위한 다양한 알고리듬들이 제시되고 있지만 본 논문에서는 Apriori알고리듬[2]을 이용한다. 이 알고리듬은 후보항목집합(candidate itemsets)을 찾아 최소지지도(minimum support)보다 큰 빈발항목집합(frequent itemsets)을 찾아내고, 이를 반복적으로 수행하는 방법이다. 이 Apriori 알고리듬을 이용하여 빈발속성집합(frequent predicate sets)을 탐색하는 차원간 연관관계 알고리듬을 <그림 2>와 같이 구성하였다.

```
Interdimensional Apriori algorithm : find frequent predicate sets
Input : Database of transactions, min_sup(minimum support threshold)
Output : L, frequent predicate sets in D, confidence, lift(?)

Method :
(1)  L1 = find_frequent_1-predicate_sets(D);
(2)  for(k=2; (Lk-1≠ø)∨(k=m+1); k++) {    // m = number of Attributes
(3)     Ck = apriori_gen(Lk-1, min_sup);
(4)     for each transaction t∈D {    // scan D for counts
(5)        Ct = subset(Ck, t);    // get the subsets of t that are candidates
(6)        for each candidate c∈Ct
(7)           c.count++;
(8)     }
(9)     Lk = {c∈Ck | c.count≥min_sup}
(10) }
(11) return L=Uk Lk;

procedure apriori_gen(Lk-1; frequent_(k-1)-predicate_sets; min_sup)
(1)  for each predicate_set l1∈Lk-1
(2)     for each predicate_set (l2∈Lk-1)∧(l2>l1)
(3)        if (l1[1]=l2[1])∧(l1[2]=l2[2])∧…∧(l1[k-2]=l2[k-2])∧(l1[k-1]<l2[k-1])∧
(4)        (l1[k-1,A]≠l2[k-1,A]) then {    // reduce the number of candidates
(5)           c=l1⋈l2;    // join step generate candidates
(6)           if has_infrequent_subset(c, Lk-1) then
(7)              delete c;    // prune step : remove unfruitful candidate
(8)           else add c to Ck;
(9)        }
(10)       return Ck

procedure has_infrequent_subset(c: candidate k-predicate set;
   Lk-1 frequent (k-1)-predicate_sets);    //use prior knowledge
(1)  for each (k-1)-subset s of c
(2)     if s∉Lk-1 then
(3)        return TRUE;
(4)     return FALSE;
```

<그림 2> Apriori 알고리듬을 이용한 차원간 연관관계 알고리듬

일반적인 범주형 데이터베이스를 변환한 트랜잭션 데이터베이스를 이용하는 차원간 연관관계의 특징은 하나의 속성 내에 있는 속성값은 단지 하나만 1을 가지고 나머지는 모두 0을 가진다는 사실이다. 그러므로 찾아낼 수 있는 최대길이의 연관관계는 데이터베이스가 가지는 속성의 수인 m이다. 이에 관련된 제약이 알고리듬 method의 (2)줄에 최대 m-빈발속성집합까지만 탐색하는 제약이 들어가 있다. 또한 (k-1)-빈발속성집합에서 k-후보속성집합을 결합하는 경우에 같은 속성내의 속성값 들은 하나의 빈발속성집합 내에 함께 들어갈 수 없으므로 결합대상이 되는 두 개의 후보속성집합의 마지막 항목이 같은 속성에 속하는가를 살펴보고, 만약 같은 속성에 속한다면 후보속성집합에서 제외해야 하는 제약이 procedure apriori_gen의 (5)줄에 추가된다. 이 제약은 결합대상이 되는 두 개의 (k-1)-빈발속성집합의 (k-1)번째 항목의 앞의 두자리 코드를 비교하여, 서로 다르면 후보속성집합에 포함시키고 같으면 후보속성집합에서 제외시키는 것이다. 이 제약을 통해 후보속성집합의 생성개수를 상당히 줄일 수 있다. <그림 1>의 트랜잭션 데이터베이스에 대하여 최소지지도를 3으로 하고, Apriori를 이용한 차원간 연관관계 알고리듬을 수행하면 <표 1>과 같은 빈발속성집합이 생성된다. <그림 2>에 나타난 차원간 연관관계 알고리듬을 통하여 구해진 최소지지도를 만족하는 빈발속성집합을 통하여 하이퍼그래프를 구성한다.

<표 1> 차원간 연관관계를 통해 생성된 빈발속성집합

1-빈발속성집합	지지도	2-빈발속성집합	지지도	3-빈발속성집합	지지도
{(1111)}	8	{(1111) (1211)}	4	{(1111) (1211) (1311)}	3
{(1211)}	6	{(1111) (1212)}	4	{(1111) (1212) (1312)}	3
{(1212)}	4	{(1111) (1311)}	4		
{(1311)}	5	{(1111) (1312)}	4		
{(1312)}	5	{(1211) (1311)}	4		
		{(1212) (1312)}	3		

3. 하이퍼그래프 분할법을 이용한 클러스터링

3.1 하이퍼그래프의 구성

하이퍼그래프 H = (V, E)는 점(vertex)들의 집합(V)과 하이퍼에지(hyperedge, E)들의 집합으로 이루어져 있다[3]. 하이퍼그래프는 두 개 이상의 점들을 연결할 수 있다는 관점에서 그래프의 개념을 확장한 것이다. 본 논문에서 하이퍼그래프의 구성은 <표 1>과 같이 차원간 연관관계 알고리듬을 통해 구한 최소지지도를 만족하는 빈발속성집합의 속성값들을 이용한다. 점들의 집합 V는 클러스터링되어야 할 속성값들의 집합이 되고, 각각의 하이퍼에지 e∈E는 각각의 빈발속성집합이다. 그러므로 점들의 총 수는 속성값들의 수이고, 하이퍼에지의 총 수는 빈발속성집합들의 수이다.

속성값들을 하이퍼그래프로 모델링하는데 있어서의 주요 문제점은 빈발속성집합의 속성값들로 하이퍼에지를 구성하고 그에 따른 가중치를 결정하는 것이다. 하이퍼그래프의 가중치의 의미는 이어진 점들이 함께 클러스터를 생성해야 하는 정도를 나타내는 것이므로 가중치가 크면 클수록 이어진 점들이 같은 클러스터에 속할 가능성을 크게 해준다. Han et al[7]은 하이퍼에지의 가중치를 적용하는데 있어 조건부 확률의 개념을 이용한 신뢰도(confidence)의 사용을 제안하였다. 빈발항목집합으로부터 구성할 수 있는 모든 연관관계를 구하고 그 신뢰도들의 평균값을 하이퍼그래프의 가중치로 사용하였다.

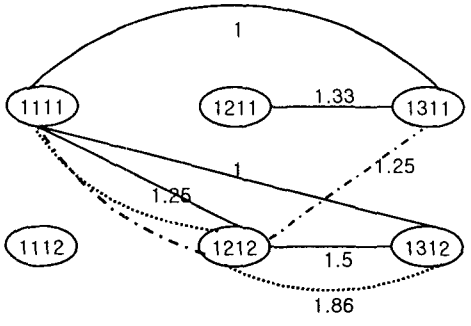
그러나, 신뢰도는 때때로 잘못된 정보를 제공해 주므로 가중치를 사용하는데 적합하지 못하

다. 예를 들어, <그림 1>의 트랜잭션 데이터베이스의 경우, 10개의 트랜잭션 가운데 8명이 대학생이고 서울에 거주하는 고객은 6명이므로 각각 80%와 60%의 지지도를 가진다. 대학생이면서 서울지역에 거주하는 고객은 4명이므로 40%의 지지도를 가진다. 여기서 "직업(X, 대학생) ⇒ 거주지(X, 서울)" 인 연관관계의 신뢰도를 계산하면 P(B|A) = P(A∧B)/P(A) = 40/80 = 50% 가 된다. 이 연관관계는 원래 주거지가 서울인 고객이 이 값보다 큰 60%이므로 잘못된 연관관계를 보여주고 있다. 즉, 사실 대학생이면서 서울에 산다는 것이 의미 있는 연관관계가 되기 위해서는 원래 서울에 사는 사람의 비율이 대학생이면서 서울에 사는 사람의 비율보다 적어야 한다. 그러나 이 경우에는 직업이 대학생이라는 것이 거주지가 서울일 가능성을 감소시키기 때문에 음의 상관관계를 가진다고 볼 수 있다. 음의 상관관계인 두 속성값은 서로 같은 클러스터에 포함되지 않도록 해야 하기 때문에 이러한 음의 상관관계를 가지는 연관관계의 신뢰도를 하이퍼그래프의 가중치로 적용하는 것은 바람직하지 않다.

그러므로 여기서는 신뢰도가 아닌 상관관계의 의미를 가지고 있는 리프트(lift)를 가중치로 사용한다. 리프트는 다음과 같은 식을 사용한다[8].

$$\mathbf{lift}_{A,B} = \frac{P(A \wedge B)}{P(A)P(B)}$$

속성집합 A의 발생이 속성집합 B의 발생과 서로 독립이면 P(A∧B) = P(A)P(B)이며, 그렇지 않으면 속성집합 A와 B는 서로 상관관계가 있다. 여기서 분모는 두 속성집합이 독립일 때 A와 B가 동시에 일어날 확률을 의미하며, 분자는 실제로 두 속성집합이 함께 일어난 확률을 의미한다. 만약 리프트의 값이 1 이면 서로 독립이고, 1 미만이면 음의 상관관계가 있고 1 보다 크면 양의 상관관계가 있게 된다. 양의 상관관계는 하나의 속성집합이 발생하면 그렇지 않은 경우보다 다른 하나의 속성집합이 발생할 가능성이 높아진다는 것을 의미한다. 그러므로, 본 논문에서는 양의 상관관계를 가지는 리프트 값이 1 이상인 연관관계에 대하여만 하이퍼그래프를 구성하도록 한다. <그림 3>은 <표 1>의 빈발속성집합을 하이퍼그래프로 표현한 것이다. 빈발속성집합 {(1111) (1211)}은 리프트 값이 1보다 작기 때문에 하이퍼그래프를 구성하지 않으며, 나머지 빈발속성집합으로부터 6개의 하이퍼에지가 나타난다.



<그림 3> 리프트를 이용한 연관관계의 하이퍼그래프

3.2 속성집합의 클러스터링

빈발속성집합이 트랜잭션의 속성값들의 관계를 보여주지만 이러한 관계는 개별적인 것들이다. 예를 들어, 빈발속성집합 {(1111) (1212)}과 {(1111) (1311)}를 생각해 보자. 이 속성집합은 대학생이고 경기도 거주자인 고객이 많고, 대학생이 노트북을 많이 산다는 것을 나타낸다. 이

것으로부터 우리는 대학생이고 거주지가 경기도인 고객이 노트북을 살 것이라는 것을 짐작할 수 있다. 그러나 이것은 지지도가 낮기 때문에 연관관계를 찾는 알고리듬으로는 찾아낼 수가 없다. 그러므로, 하이퍼그래프를 이용하여 가능한 한 연관이 많이 되어 있는 속성값들로 분할함으로써 클러스터링 하도록 한다. 이것을 위하여 하이퍼그래프를 둘로 분할하는 하이퍼그래프 분할 알고리듬을 이용하며, 분할된 영역으로 인해 잘려지는 하이퍼에지(hyperedge cut)의 가중치를 최소로 하도록 한다. 하이퍼에지 컷을 최소화하는 것은 속성값들을 두 개의 집단으로 쪼개면서 서로 다른 두 개의 집단을 연결하는 관계를 최소화 하기 위한 것이다. 이렇게 나누어진 두 개의 분할 영역을 다시 반복적으로 계속해서 이분화함으로서 클러스터를 만들어 나간다.

하이퍼그래프 분할 알고리듬은 고밀도 집적회로(VSLI)설계에 주로 적용되는 알고리듬이다. 가장 널리 이용되는 방법은 사용자가 지정하는 k개의 분할 영역을 만들때까지 반복적으로 이분화해가는 알고리듬이다. 하이퍼그래프의 최적으로 이분화하는 문제는 NP-hard문제[11]임이 증명되어 있기 때문에 발견적 해법들이 개발되어 왔다. 본 논문에서는 Karypis et al[11]이 제시한 알고리듬과 프로그램을 이용하도록 한다. 이 알고리듬은 HMETIS(http://www.cs.umn.edu/~metis) 프로그램으로 구현되어 있다. HMETIS는 하이퍼에지 컷의 가중치를 최소화함으로써 각각의 분할 영역 안에 있는 점들의 연결강도를 높게 한다.

HMETIS를 이용하여 <그림 3>의 하이퍼그래프를 클러스터링하였다. 클러스터의 수를 3개로 한 후 하이퍼에지 컷을 최소로 하도록 알고리듬을 수행한 결과 하이퍼에지 컷은 2.25이고 3개의 클러스터 {(1111) (1212) (1312)}, {(1211) (1311)}, {(1112)}이 나왔다. 클러스터{(1112)}는 빈발속성집합에 소속된 것이 없으므로 하나의 클러스터로 선택되었으며, 클러스터{(1111) (1212) (1312)}는 이것을 포함하는 빈발속성집합이 {(1111) (1212)}, {(1111) (1312)}, {(1212) (1312)}, {(1111) (1212) (1312)}로 네 개이므로 클러스터가 되었고, 클러스터{(1211) (1311)}도 {(1211) (1311)}, {(1111) (1211) (1311)} 두 개의 빈발속성집합에 포함되므로 하나의 클러스터가 되었다. 이 속성값에 대한 클러스터링 결과는 대체로 같은 속성에 속한 속성값 들이 같은 클러스터내에 속하는 경우가 없으므로 같은 속성내에서는 하나의 속성값만을 가지는 성질에 위배되지 않도록 올바른 결과가 나온 것으로 볼 수 있다.

3.3 트랜잭션의 클러스터링

하이퍼그래프 분할 알고리듬을 통해서 구한 속성값의 클러스터를 이용하여 트랜잭션의 클러스터링을 수행한다. 클러스터 내의 속성값들은 그 클러스터의 특성을 설명해주는 역할을 한다. 트랜잭션의 클러스터링은 클러스터내에 속한 속성값과 트랜잭션내에 속한 속성값을 바탕으로 각각의 클러스터에 속할 점수를 계산하여 트랜잭션이 속할 클러스터를 결정한다. 여기서는 Han et al[7]이 제시한 방법을 그대로 이용한다. 점수는 간단한 비율함수인 $|T \cap Ci|/|Ci|$으로 나타내며, 여기서 T는 트랜잭션이고 Ci는 속성값의 클러스터이다. 트랜잭션은 가장 높은 점수를 가지는 속성값의 클러스터에 포함시키며, 동점의 경우 일치하는 속성값이 더 많은 클러스터에 할당한다.

앞 절에서 나온 클러스터를 가지고 트랜잭션을 클러스터링하면 <표 2>와 같다. 트랜잭션 1, 5, 7, 9는 클러스터 2에 할당되었으며, 트랜잭션 2, 3, 4, 6, 8은 클러스터 3에 트랜잭션 10은 클러스터 1에 할당된다. 클러스터 2에 속한 트랜잭션들은 주거지가 서울지역이면서 노트북을 구입한 고객들이라고 볼 수 있고 클러스터 3에 속한 트랜잭션들은 직업이 대학생이면서 주거지가 경기지역이고 주로 PC를 구매한 고객들로 볼 수 있다. 클러스터 1에 속한 트랜잭션은 회사원인 것을 알 수 있다.

<표 2> 트랜잭션의 클러스터링 결과

$\|T \cap C_i\| / \|C_i\|$ 트랜잭션	클러스터 1 {(1112)}	클러스터 2 {(1211)(1311)}	클러스터 3 {(1111) (1212) (1312)}	소속 클러스터
T1	0	2/2	1/3	2
T2	0	0	3/3	3
T3	0	1/2	2/3	3
T4	0	1/2	2/3	3
T5	1	2/2	0	2
T6	0	0	3/3	3
T7	0	2/2	1/3	2
T8	0	0	3/3	3
T9	0	2/2	1/3	2
T10	1	1/2	1/3	1

3.4 계산의 복잡성(Complexity)

최소지지도를 기준으로 차원간 연관관계를 찾는 알고리듬은 트랜잭션의 수에 대하여 계산량이 선형적으로 증가한다[1]. Apriori알고리듬을 이용한 차원간 연관관계 알고리듬은 지지도가 충분히 크면 대용량 데이터베이스에서도 적당한 시간 내에 연관관계들을 찾아낼 수 있다. 그러나, 지지도가 아주 작은 경우에는 계산시간이 훨씬 더 커지게 되므로 적절한 지지도를 선택하는 것이 필요하다. 최소지지도가 너무 작으면 불필요하게 많은 노이즈(noise)를 포함한 관계들을 찾아내게 되고 지지도가 너무 크면 연관관계가 너무 적어지므로 속성값을 잘 클러스터링할 수 있는 적절한 지지도를 선택해야 한다. Apriori알고리듬의 계산량을 줄이려면 해쉬함수를 이용하거나 다음에 스캐닝(scan)할 트랜잭션의 수를 줄이는 등의 방법을 이용할 수도 있으며, 특히 후보항목집합을 생성하지 않는 FP-tree방법[9]을 이용할 수도 있을 것이다.

하이퍼그래프 분할 알고리듬은 하이퍼에지의 수가 적절한 경우에 좋은 분할 영역을 얻을 수 있으며, HMETIS알고리듬은 몇 분내에 100,000개의 점들을 이분화하여 좋은 분할 영역을 찾을 수 있다. 특히 k 개의 분할영역을 찾는 HMETIS의 계산량은 O((V+E)logk)이다[11]. 점의 수(V)는 전체 속성값의 수와 같으며, 하이퍼에지의 수(E)는 빈발속성집합의 수와 같다.

또한 마지막으로 트랜잭션을 클러스터링하는 부분도 역시 트랜잭션의 수에 대하여 선형적으로 증가하며, 짧은 시간 내에 계산할 수 있다.

4. 실험결과

본 알고리듬의 효율성과 효과를 증명하기 위하여 데이터마이닝 연구분야에서 널리 이용되는 두 개의 실제 데이터를 이용한다. 첫 번째 데이터는 Huang[10] 이 사용한 콩의 질병을 기록한 데이터이며, 두 번째 데이터셋은 ROCK[6] 에 사용된 버섯 데이터이다.

4.1 콩 질병 데이터

콩 데이터는 47개의 개체를 가지며, 35개의 속성으로 표현된다. 35개의 속성 중 하나의 속성값만을 가지는 속성은 제외하고 21개의 속성만을 대상으로 한다. 각각의 개체는 네 가지의 질병 중 한가지의 클래스에 속하며, 각 클래스는 Diaporthe Stem Canker(D), Charcoal Rot(C), Rhizoctonia(R), Phytophthora(P) 이고, 각각 10, 10, 10, 17 개의 개체가 이에 속한다. 각각의 속성들의 속성값들을 코드화 한 후 본 논문이 제시하는 클러스터링 알고리듬을 적용하여, 최소지지도를 20%로 하고 6개 클러스터로 나눈 결과가 <표 3>에 나타나 있으며, 최소지지도를

20%로 하고 8개의 클러스터 나눈 결과가 <표 4>에 나타나 있다.

<표 3> 콩 질병 데이터를 6개로 분할한 클러스터링 결과

	클러스터 1	클러스터 2	클러스터 3	클러스터 4	클러스터 5	클러스터 6
D	0	0	10	0	0	0
C	0	10	0	0	0	0
R	2	0	0	0	3	5
P	15	0	0	1	0	1
할당된 속성값	1212, 1614 1712, 2211 2312, 2413 2511, 2811 3012, 3112	1211, 1311 1811, 2311 2414, 2611 2813, 2912 3011, 3111	1313, 1412 1613, 2112 2314, 2412 2512, 2612 2711, 2911	1512 2011 2012 2212	1111 1612 1912 2013	1411 1511 1812 1911

<표 4> 콩 질병 데이터를 8개로 분할한 클러스터링 결과

	클러스터 1	클러스터 2	클러스터 3	클러스터 4	클러스터 5	클러스터 6	클러스터 7	클러스터 8
D	0	0	8	0	2	0	0	0
C	0	0	0	0	0	10	0	0
R	0	10	0	0	0	0	0	0
P	1	0	0	15	1	0	0	0
할당된 속성값	1512 1612 2011	1411, 1511 2312, 2412 3012, 3111	1313, 1412 1614, 2314 2512, 2612 2811, 2911	1212, 1712 2013, 2112 2211, 2413 2511, 3112	1613 1811 2711	1211, 1311 2311, 2414 2611, 2813 2912, 3011	1111 2012 2212	1812 1911 1912

이 클러스터링 결과는 Huang의 K-mode 알고리듬의 결과[10]와 비교할 때, 평균적으로 좋은 결과를 보여주고 있다. 다만 4개 이상의 클러스터링에 의한 결과보다 그 이상의 클러스터로 나눌 때의 결과가 더 좋은 단점이 있으나 할당된 속성값을 이용하여 클러스터의 특성을 설명해 줄 수 있는 장점이 있다. 이와 같은 장점은 클러스터링을 수행하는 목적이 탐색적이고 클러스터의 특성을 규명하려 한다는 관점에서 매우 중요한 특성이라 할 수 있다.

4.2 버섯 데이터

버섯 데이터는 8124 개의 개체를 가지며, 버섯의 물리적인 특성(색상, 냄새, 크기, 모양)을 설명해주는 22 개의 속성을 가지고 있다. 각각의 개체는 먹을 수 있는 버섯(edible)과 독버섯(poisonous) 두 개의 클래스로 나누어지며, 먹을 수 있는 버섯의 수는 4208 이고 독버섯의 수는 3916 이다. 최소지지도를 20%로 하고 4개의 클러스터로 나눈 결과가 <표 5>에 나타나 있다.

<표 5> 버섯 데이터를 4개로 분할한 클러스터링 결과

	클러스터 1	클러스터 2	클러스터 3	클러스터 4
Edible	640(21%)	3508(86%)	60(94%)	0
Poisonous	2458(79%)	576(14%)	882(6%)	0
할당된 속성값	1412, 1613, 1711 1812, 1913, 2012 2111, 2611, 2713 2812, 2912, 3018 3115	1113, 1114, 1211 1411, 1517, 1811 2214, 2314, 2418 2518, 2916, 3217	1213, 2011, 2213 2313, 2516, 3116 3211	1214, 1311, 1314 1515, 2416, 3011 3014

이 결과는 약 84.3%의 결과를 나타내고 있다. 이는 ROCK[6]의 가장 좋은 결과에는 미치지 못하지만 전통적인 클러스터링 알고리듬의 결과에 비해서는 훨씬 정확한 결과이며, 콩의 경우와 마찬가지로 클러스터에 할당된 속성값들을 바탕으로 클러스터의 특성을 설명해 줄 수 있다.

5. 결론

본 논문에서는 연관관계 생성알고리듬과 하이퍼그래프 분할 알고리듬을 이용하여 일반적인 범주형 데이터베이스에 대한 클러스터링을 수행하는 방법을 제안하였다. 먼저, 관계형 데이터베이스 등과 같은 다차원 데이터베이스를 트랜잭션 데이터베이스로 전환한다. 하이퍼그래프를 구성하기 위해 차원간 연관관계 알고리듬을 이용하여 빈발속성집합을 찾아내고, 리프트 값을 하이퍼에지의 가중치로 하여 하이퍼그래프 분할 알고리듬을 수행한다. 이렇게 구한 속성값들의 클러스터를 이용하여 트랜잭션들의 클러스터를 구한다. 실제 데이터를 이용하여 검증해 본 결과, 이러한 클러스터는 서로 연관이 있는 트랜잭션들을 클러스터링 해주며, 클러스터의 속성값들을 이용하여 클러스터의 특성을 설명해 줄 수 있는 장점이 있다. 보다 좋은 클러스터링 결과를 얻기 위하여 속성값들을 클러스터링 할 때, 중복이 가능하도록 하는 새로운 분할 기법의 개발이 필요로 할 것이다.

참 고 문 헌

[1] Agrawal, R., Mnanila, H., Srikant, R., Toivonen, H., and Verkamo, A.I., "Fast Discovery of Association Rules," Advances in Knowledge Discovery and Data Mining, Santiago, pp. 307-328, 1994.

[2] Agrawal, R., and Srikant R., "Fast Algorithm for Mining Association Rules," Proc. of the 20th Int'l Conference on Very Large Data Bases(VLDB), pp. 487-499, 1994.

[3] Berge, C., Graphs and Hypergraphs, American Elsevier, 1976.

[4] Ganti, V., Gehrke, Y., and Ramakrishnan, R., "CACTUS - Clustering Categorical Data Using Summaries," Proc. of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD), pp. 73-83, 1999.

[5] Gibson, D., Kleinberg, J., and Raghavan, B., "Clustering Categorical Data: An Approach Based on Dynamical Systems," Proc. of the 24th Int'l Conference on Very Large Data Bases(VLDB), pp. 311-322, 1998.

[6] Guha, S., Rastogi, R., and Shim, K., "ROCK: A Robust Clustering Algorithm for Categorical Attributes," Proc. of the 15th International Conference on Data Engineering(ICDE), pp. 512-521, 1999.

[7] Han, E., Karypis, G., Kumar, V., and Mobasher, B., "Clustering Based On Association Rule Hypergraphs," In Proc. of the Workshop on Research Issues on Data Mining and Knowledge Discovery, pp. 9-13, 1997.

[8] Han, J., and Kamber, M., Data Mining: Concepts and Techniques, Academic Press, pp.251-259, 2001.

[9] Han, J., Pei, J., and Yin. Y., "Mining Frequent Patterns Without Candidate Generation," Proc. of the ACM SIGMOD International Conference on Management of Data, pp. 1-12, 2000.

[10] Huang, Z., "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining," In Proc. SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery(DMKD), 1997.

[11] Karypis, G, Aggarwal, R., Kumar, V., and Shekhar, S., "Multilevel Hypergraph Partitioning: Applications in VLSI Design", Proc. ACM/IEEE Design Automation Conference, pp. 526-552, 1997.

[12] Zhang, Y., Fu, A.W., Cai, C.H., and Heng, P.A., "Clustering Categorical Data," Proc.of the 16th International Conference on Data Engineering(ICDE), pp. 305, 2000.