

빈발 패턴 네트워크에서 연관 규칙 발견을 위한 아이템 클러스터링

오경진^a, 정진국^a, 조근식^b

^a 인하대학교 공과대학 정보공학과 인천광역시 남구 용현동 253, 402-751

Tel: +82-032-875-5863, Fax: +82-032-875-5863, E-mail: okjkillo@eslab.inha.ac.kr, gj4024@eslab.inha.ac.kr

^b 인하대학교 공과대학 컴퓨터정보공학부 인천광역시 남구 용현동 253, 402-751

Tel: +82-032-860-7447, Fax: +82-032-875-5863, E-mail: gsjo@inha.ac.kr

Abstract

데이터마이닝은 대용량의 데이터에 숨겨진 의미있고 유용한 패턴과 상관관계를 추출하여 의사결정에 활용하는 작업이다. 그 중에서도 고객 트랜잭션의 데이터베이스에서 아이템 사이에 존재하는 연관규칙을 찾는 것은 중요한 일이 되었다. *Apriori* 알고리즘 이후 연관규칙을 찾기 위해 대용량 데이터베이스로부터 압축된 의미있는 정보를 저장하기 위한 데이터 구조와 알고리즘들이 제안되어 왔다. 본 논문에서는 정점으로 아이템을 표현하고, 간선으로 두 아이템집합을 표현하는 빈발 패턴 네트워크(FPN)이라 불리는 새 자료 구조를 제안한다. 빈발 패턴 네트워크에서 아이템 사이의 연관 관계를 발견하기 위해 이 구조를 어떻게 효율적으로 사용하느냐에 초점을 두고 있다. 구조의 효율적인 사용을 위하여 한 아이템이 클러스터 내의 아이템과는 유사도가 높고, 다른 클러스터의 아이템과는 유사도가 낮도록 네트워크의 정점을 클러스터링하는 방법을 사용한다. 실험은 신뢰도, 상관관계 그리고 간선 가중치 유사도를 이용하여 네트워크에서 아이템 클러스터링의 정확도를 보여준다. 본 논문의 실험 결과를 통해 신뢰도 유사도가 네트워크의 정점을 클러스터링할 때 클러스터의 정확성에 가장 많은 영향을 미친다는 것을 알 수 있었다.

Keywords:

데이터마이닝; 클러스터링; 연관규칙; 빈발 패턴 네트워크

1. 서론

데이터마이닝(Data Mining)은 대량의 데이터 속에 숨겨진 의미있고 유용한 패턴(Pattern)과 상관관계를 추출하여 의사 결정에 이용하는 작업이다. 정보화 혁명 이후 정보기술의 가속적 발전으로 인해 매일

쏟아져 나오는 데이터의 양은 사람의 힘으로는 도저히 소화할 수 없을 정도로 방대해졌다. 이와 같이 웹의 경우를 보아도 하루에 백오십만 페이지 이상 증가하고 있다. 데이터의 양이 급속도로 증가하고 있어, 그 안에 담긴 정보를 찾아 활용하는 일이 쉽지 않다. 따라서 방대한 양의 축적된 데이터 속에 존재하는 유용한 패턴과 상관관계를 찾아내기 위한 데이터 마이닝 기법의 연구가 지속적으로 이루어지고 있고, 데이터베이스(Databases)에서 존재하는 연관규칙을 찾는 것이 중요한 일이 되었다. 연관규칙은 데이터 안에 존재하는 아이템(Item)간의 종속관계를 찾아내는 작업이며, 마케팅(Marketing)에서는 손님의 장바구니에 들어 있는 아이템간의 관계를 알아 본다는 의미에서 경제 분야에서는 시장바구니분석 (Market Basket Analysis)이라고도 한다. 연관규칙 (Association Rule)은 서비스의 교차판매, 매장진열, 첨부우편, 사기적발 등의 다양한 분야에 활용되고 있다.

전형적인 연관규칙 알고리즘은 두 단계로 작동한다. 첫 번째 단계는 최소지지도(Minimum Support)를 만족하는 아이템 집합인 빈발 항목 집합(Frequent Itemsets)을 찾는 것이다. 두 번째 단계는 최소신뢰도(Minimum Confidence)를 만족하는 모든 빈발 항목 집합으로부터 규칙을 생성하는 것이다. *Apriori* 알고리즘 이후 빈발 항목 집합을 찾기 위한 알고리즘들이 많이 연구되어 왔고, 빈발 항목 집합을 찾기 위해 대량의 데이터베이스로부터 압축된 의미 있는 정보를 저장하기 위한 자료 구조도 제안되어 왔다.

본 논문에서는 빈발 패턴 네트워크(Frequent Pattern Network)라 명명한 새로운 자료 구조를 제안한다. 빈발 패턴 네트워크는 아이템을 표현하기 위한 정점(Vertex)과 두 아이템 집합을 표현하기 위한 간선(Edge)으로 구성되어 있다. 구조의 효율적인 사용을 위해 한 아이템이 클러스터(Cluster) 내의 아이템과는 유사도가 높고, 다른 클러스터의 아이템과는 유사도가 낮도록 네트워크의 정점을

클러스터링(Clustering)하는 방법을 사용한다. 기존의 연관 규칙 알고리즘은 많은 연관 규칙을 생성하는데 반해, 클러스터링 방법은 발견되는 연관 규칙의 수를 줄이는데 도움이 된다.

본 논문의 구성은 다음과 같다. 2장에서는 클러스터링에 대하여 알아보고, 관련된 연구를 분석한다. 3장에서는 본 논문에서 제안한 빈발 패턴 네트워크를 설명한다. 4장에서는 빈발 패턴 네트워크에서의 클러스터링 알고리즘을 설명한다. 5장에서는 구현 및 실험을 통해 빈발 패턴 네트워크의 결과를 분석하고, 마지막으로 6장에서는 결론을 맺는다.

2. 배경 및 관련 연구

2.1 클러스터링

클러스터링은 클러스터 안의 아이템끼리는 높은 유사성을 갖게 하고 다른 클러스터들의 아이템과는 큰 상이성을 갖도록 클러스터로 만들어가는 과정이다. 상이성은 아이템을 표현하는 속성 값에 기초하여 매겨진다. 클러스터링은 데이터 마이닝, 통계학, 생물학, 그리고 기계 학습 분야 등에 많이 이용되고 있다.[1].

클러스터링은 중요한 인간 행위이다. 예를 들어, 어렸을 때 사람은 지속적으로 무의식적인 클러스터링을 시도함으로써 동물과 식물 혹은 사람과 사물을 구분하는 방법을 배운다. 클러스터링은 패턴 인식, 데이터 분석, 이미지 처리, 그리고 시장조사를 포함한 많은 분야에 넓게 사용되고 있고, 최근 데이터 마이닝 연구에서 가장 활발히 활용되고 있다. 클러스터링을 통해 전체적인 데이터의 분포 패턴을 알 수 있고, 각 클러스터에 존재하는 유용한 상관관계를 찾을 수 있다.

클러스터링 기법은 분할 기법(Partitioning Methods), 계층적 기법(Hierarchical Method), 밀도기반 기법(Density-based Method), 격자기반 기법(Grid-based Method), 그리고 모델기반 기법(Model-based Method) 등 5가지로 분류한다[1].

본 논문에서는 탐욕적 방법으로 클러스터링하는 계층적 방법을 사용하고, 상관관계(Correlation), 신뢰도(Confidence), 간선가중치 유사도(Edge Weight Similarity)를 사용하여 클러스터에 아이템을 포함시켰다. 생성한 클러스터를 평가하기 위해 에러기반 기준(Error-based Criterion)을 사용하였다.

2.2 관련 연구

Apriori 알고리즘[7] 이후 빈발 항목 집합을 찾기 위한 알고리즘들이 많이 연구되어 왔고, 빈발 항목 집합을 찾기 위해 대량의 데이터베이스로부터 압축된 의미 있는 정보를 저장하기 위한 자료 구조도

제안되어왔다.

Apriori 알고리즘[7]은 최소지지도를 만족하는 모든 빈발 항목 집합을 생성한다. 하지만 패턴의 길이가 길 경우, 생성된 후보 항목 집합이 최소지지도를 만족하는가에 대한 확인을 위해 계속적인 데이터베이스 접근을 해야 한다. [10]에서는 아이템 제약을 사용하여 후보 항목 집합(Candidate Set)의 생성 시간을 줄이기 위한 3가지 방법을 제안하였지만, 데이터베이스의 접근은 계속 이루어져야 한다.

FP-growth 알고리즘[3]은 간결하고 압축된 정보를 표현하는 빈발 패턴 트리(FP-Tree)를 생성 한다. 빈발 패턴 트리는 후보 항목 집합을 생성할 때 데이터베이스 접근 비용을 효율적으로 줄인다. 하지만 트랜잭션(transaction)에 포함된 아이템이 많으면 트리의 깊이가 깊어지고, 과도한 빈발 패턴 트리의 생성과 소멸이 문제가 된다.

[4]에서 제안한 단편 지지도 맵(Segment Support Map)은 Apriori에 적용되어 많은 수의 아이템 집합을 제거할 수 있다. 그래서 Apriori 알고리즘의 성능을 최적화하는 장점을 가지고 있다. 반면에 이 구조는 아이템을 클러스터링할 수 없어, 그룹별 데이터의 특징을 알 수 없다.

[9]에서는 Frequent Closed Itemset[5]을 기반으로 정보의 손실 없이 적은 수의 규칙을 발견하기 위하여 갈루아 래티스(Galois Lattice)구조를 사용하였다.

[8]에서는 연관 규칙 하이퍼그래프(Association Rule Hypergraph)를 이용하여 아이템과 트랜잭션을 클러스터링 하는 방법을 제안하였다. 연관 규칙을 사용하여 밀접하게 관련된 아이템을 클러스터링하고, 클러스터링된 아이템을 기반으로 트랜잭션을 클러스터링하는 방법을 소개하였다. 이 논문에서는 큰 연관성이 있는 아이템을 클러스터링하기 위해 연관 규칙을 사용하였다. 하지만 연관 규칙을 이용하기 위해 연관 규칙 알고리즘을 수 행해야 한다.

본 논문에서는 아이템의 발생 빈도수(Frequency)를 기반으로 빈발 패턴 네트워크를 형성하여 아이템 클러스터링을 한다. [8]과 같이 빈발 항목 집합을 찾아 그 속에 담겨있는 연관 규칙을 이용하여 네트워크를 형성하지 않고, 네트워크에 존재하는 아이템을 클러스터링하기 때문에 빈발 항목 집합을 찾는 과정에서 발생하는 비용을 줄일 수 있다. [11]에서와 같은 전통적인 접근과는 달리, 클러스터로 탐색 공간을 국한 시킴으로써 효율적인 데이터마이닝 알고리즘을 구성할 수 있다. 클러스터는 기존에 존재하는 알고리즘에 의해 계산된 커다란 연관 규칙의 수를 줄이는데 유용하고, 유사성이 있는 아이템을 클러스터링함으로써 데이터 분포 패턴과 데이터 속성들 사이에 존재하는 흥미 있고 유용한 상관관계를 찾을 수 있다. 이 방법은 데이터베이스의 접근이 한번만 이루어지므로

트랜잭션의 수에 선형적으로 수행되고, 규칙을 발견하기 위해 모든 트랜잭션을 저장할 필요가 없다.

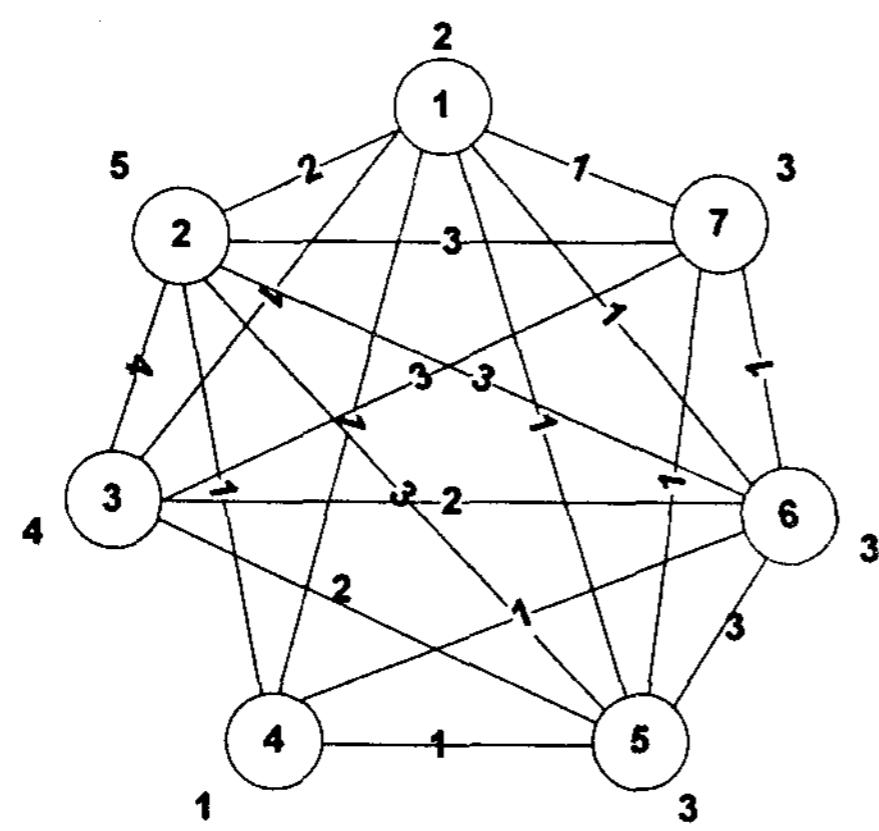
3. 빈발 패턴 네트워크

데이터베이스 안에 모든 아이템은 오름차순으로 존재한다고 가정을 하고, 데이터베이스로부터 모든 트랜잭션을 빈발 패턴 네트워크로 옮긴다. 네트워크는 연관규칙 마이닝 작업을 하는 동안 데이터베이스를 표현하는 간결하고 압축된 자료구조이다. 네트워크는 무방향(Undirected), 자기반복(Self-loop)이 없는 가중치그래프(Weighted Graph)이다.

빈발 패턴 네트워크 $N = (V, E)$ 는 정점들의 집합 V 와 간선들의 집합 E 로 구성되어 있다. 정점 집합 V 는 클러스터되어질 데이터 아이템의 집합을 표현하고, 간선 집합 E 는 V 의 두 정점의 연결을 표현한다. 정점의 가중치는 아이템의 발생 빈도를 나타내고, 간선의 가중치는 두 정점 사이에 공동으로 발생하는 빈도를 나타낸다. 다시 말해서 간선의 가중치는 두 정점(아이템)의 연결 강도를 표현하기 위한 값이다.

[그림 1]은 네트워크의 간단한 예이다. 왼쪽은 데이터베이스에 축적된 트랜잭션을 나타내고, 오른쪽은 트랜잭션을 네트워크에 옮겼을 때의 모습이다.

TID	Item
1	2 3 7
2	1 2 4 5 6
3	1 2 3 7
4	2 3 5 6
5	2 3 5 6 7



[그림 1] 빈발 패턴 네트워크

빈발 패턴 네트워크의 관련 사항을 다음과 같이 정의한다.

정의 1. 빈발 패턴 네트워크

- 네트워크는 정점들의 집합과 간선들의 집합으로 구성된다.
- 네트워크에서 하나의 정점은 하나의 아이템에 해당한다. 정점은 **itemName**, **count**, 그리고 **edges**의 세 가지 속성을 가지고 있다. **itemName**은 아이템의 이름을 가지고, **count**는 아이템의 발생 빈도수를 표현하는 지지도를 저장한다. 그리고 **edges**는 간선들

의 집합이다. 사전적으로 마지막에 위치한 정점은 간선들을 갖지 않는다.

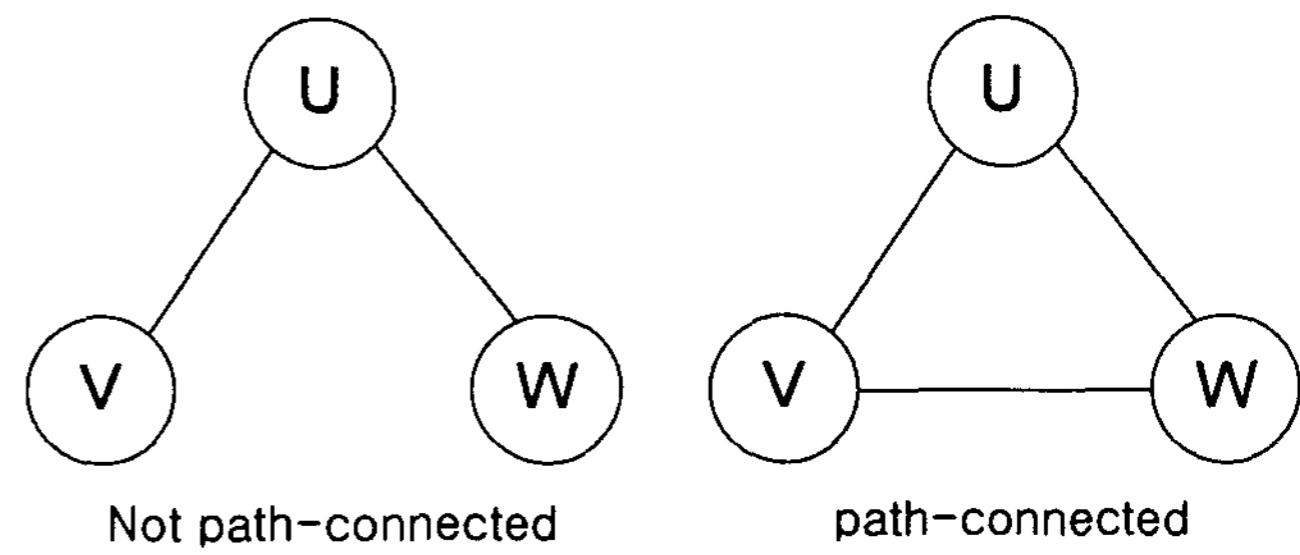
- 네트워크에서 하나의 간선은 2-아이템집합에 해당한다. 간선은 **fromVertex**, **toVertex**, 그리고 **count**의 세 가지 속성을 가진다. **fromVertex**와 **toVertex**는 정점에 연결을 나타내고, **fromVertex**가 사전 편집상 더 빨리 발생한다. **count**는 두 정점(아이템)에 공통으로 발생하는 빈도수를 저장한다.

정점 u 와 v 를 잇는 간선은 $edge(u, v)$ 또는 e_{uv} 로 표기한다. 정점 u 와 v 사이의 정점들과 간선들이 나오는 순차열을 경로(Path)라 하고, $path(u, v)$ 로 표기한다. 정점 u 을 시작 정점(Start Vertex)이라 하고, 정점 v 를 끝 정점(End Vertex)이라 한다. $path(u, v)$ 의 다른 정점들은 내부 정점(Internal Vertex)이라 한다.

정의 2. 연결된 경로(Connected Path)

$path(u, v)$ 위에 있는 정점들의 집합을 X 라 하고, $X = \{x \mid x \in V, x \in (u, v)\}$ 로 표기한다. 여기에서 V 는 빈발 패턴 네트워크에 포함된 정점들의 집합이다. X 안의 어느 두 정점 x 와 y 에 대해 간선 $edge(x, y)$ 가 존재하면 X 를 연결된 경로라 한다.

예를 들어, [그림 2]의 왼쪽 부분은 네트워크가 세 정점 u , v , w 와 두 간선 $edge(u, v)$, $edge(u, w)$ 로 구성되어 있다. 왼쪽 부분은 $edge(v, w)$ 가 존재하지 않기 때문에, $path(u, v)$ 는 연결된 경로가 아니다. 연결된 경로가 되려면 $edge(v, w)$ 를 포함한 그림의 오른쪽 부분과 같이 되어야 한다. 그러므로 연결된 경로는 완전히 연결된 그래프이다.



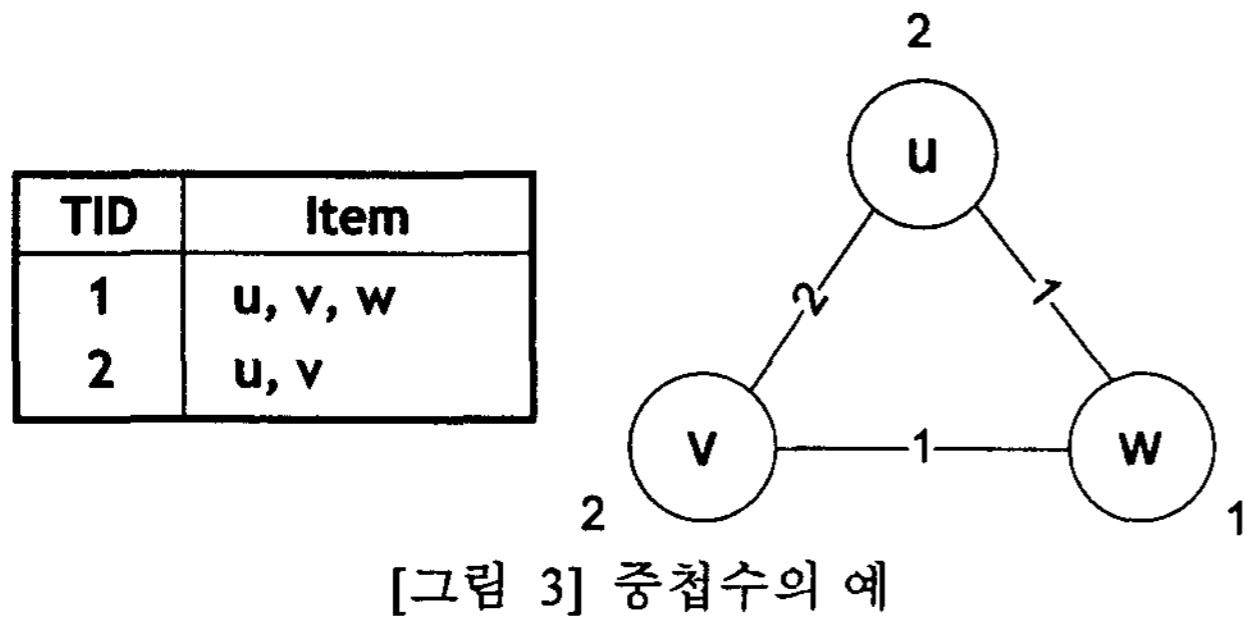
[그림 2] 연결된 경로

정의 3. 중첩수(Countfold)

위의 정의에서 사용된 예를 보자. 네트워크의 경로 안에 세 정점들의 집합을 X 라 하고, x 와 y 가 X 에 포함된다고 하자. $x \neq y$ 이고, e_{xy} 는 x 에서 y 로의 간선이다. 정점 x 의 빈도수는 $count(x)$, 간선 e_{xy} 의 빈도수는 $count(e_{xy})$ 로 표기한다. X 에서 정점 x 의 중첩수를 $countfold(x)$ 라 표시하고 식(1)과 같이 계산한다. 정점 x 에 연결된 모든 간선의 가중치를 더한 다음, 정점 x 의 가중치를 뺀다.

$$\text{countfold}(x) = \sum_{e_{xy} \in E} \text{count}(e_{xy}) - \text{count}(x) \quad (1)$$

[그림 3]은 두 트랜잭션 $\{u, v, w\}$ 와 $\{u, v\}$ 으로 이루어졌다. 두 트랜잭션을 세 정점 u, v, w 와 간선 $\text{edge}(u, v), \text{edge}(u, w), \text{edge}(v, w)$ 로 구성된 네트워크로 옮길 수 있다. 각 정점의 빈도수는 $\text{count}(u) = 2, \text{count}(v) = 2, \text{count}(w) = 1$ 이고, 각 간선의 빈도수는 $\text{count}(e_{uv}) = 2, \text{count}(e_{uw}) = 1, \text{count}(e_{vw}) = 1$ 이다. 정점 v 의 중첩수 $\text{countfold}(x)$ 를 식(1)로 계산하면 1이다.



정의 4. 경로지지도(Path Support)

경로지지도는 경로 전체의 지지도 값(Support Value)이다. 네트워크에서 정점 x 에서 정점 z 까지 경로지지도를 $f(x, z)$ 로 표기하고, 식(2)로 계산한다. $\text{path}(x, z)$ 에서 $x \neq y$ 이고, V 와 E 는 네트워크에서 각각 정점과 간선의 집합이다.

$$f(x, z) = \max\{\min\{\text{count}(e_{xy}) \mid xy \in E\}, \max\{\text{countfold}(x) \mid x \in V\}\} \quad (2)$$

중첩수를 사용하여 $\{u, v, w\}$ 같은 k -아이템집합 ($k > 2$)의 지지도 값을 계산할 수 있다. 하지만 [그림 4]와 같은 트랜잭션이 데이터베이스에 발생하였을 경우, 식(2)를 사용하여 [그림 4]에 있는 네트워크의 3-아이템집합 $\{1, 2, 3\}$ 의 정확한 지지도 값을 계산할 수 없다. 그림에서 데이터베이스는 여섯 개의 트랜잭션을 가지고 있고, 이것을 빈발 패턴 네트워크로 옮길 수 있다. 데이터베이스에 3-아이템집합 $\{1, 2, 3\}$ 을 포함하는 트랜잭션은 존재하지 않는다. 하지만 네트워크에는 3-아이템집합 $\{1, 2, 3\}$ 이 존재하게 되고, 식(2)에 의해 계산된 아이템집합의 경로지지도 값은 2이다. [그림 4]의 경우는 존재하지 않는 3-아이템집합 $\{1, 2, 3\}$ 이 계산된다.

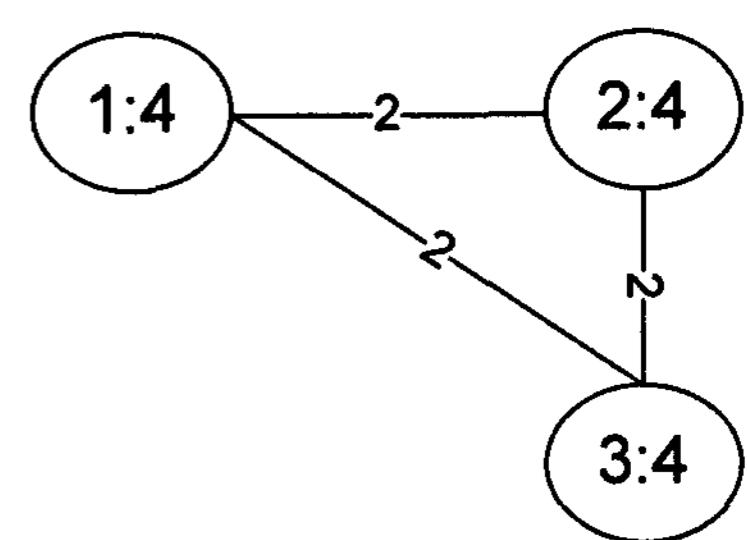
이러한 경우의 문제점을 해결하기 위하여 논리적 패턴을 도입한다.

정의 5. 논리적 패턴(Logical Pattern)

실질적으로 경로가 빈발 항목 집합이 아니고 존재하지 않지만, 네트워크에서 경로가 연결된 경로이며, 경로지지도가 주어진 최소지지도 임계값

θ 와 동일하거나 그보다 큰 경우, 그 경로를 논리적 패턴이라 부른다. 하지만 실질적으로 그 경로는 빈발 항목 집합이 아니다.

TID	Item
1	$1, 2$
2	$1, 2$
3	$2, 3$
4	$2, 3$
5	$1, 3$
6	$1, 3$



[그림 4] 논리적 패턴

4. 연관규칙 발견을 위한 클러스터링

본 장에서는 연관규칙 발견을 위한 클러스터 생성 방법을 설명한다.

빈발 패턴 네트워크 $N = (V, E)$ 은 클러스터 생성 알고리즘의 입력으로 사용된다. 네트워크에서 정점들의 집합은 $V = v_1, v_2, \dots, v_n$, 간선들의 집합은 $E = e_{12}, e_{13}, \dots, e_{1m}$ 으로 표기하고, 간선 $\text{edge}(v_i, v_j)$ 의 가중치는 $w_{ij}(v_i, v_j)$ 로 표기한다. 빈발 패턴 네트워크 클러스터링의 목표는 네트워크의 아이템을 클러스터 안의 아이템끼리는 높은 유사성을 갖게 하고 다른 클러스터들의 아이템과는 큰 상이성을 갖도록 네트워크를 k 개의 부집합(Subsets)으로 그룹화하는 것이다. 생성된 클러스터의 모든 정점은 연결된 경로이고 각각의 클러스터에 포함된 정점의 수는 2와 같거나 크다. 또한 클러스터는 공통 원소를 갖지 않는다.

4.1 알고리즘(Algorithm)

빈발 패턴 네트워크에서 클러스터링 알고리즘을 FPNC(Frequent Pattern Network Clustering)로 부른다. FPNC 알고리즘은 클러스터 안으로 연관 있는 정점을 클러스터링한다. 정점의 집합 V , 최소지지도 θ , 최소신뢰도 ξ 를 입력으로 받아 알고리즘을 수행한 후, 클러스터 C 의 집합을 결과물로 얻는다.

Algorithm FPNC

Input :

- 정점 v_1, v_2, \dots, v_n 의 집합 V
- 최소지지도 θ
- 최소신뢰도 ξ

Output :

- 클러스터 C 의 집합

Method :

1. V 에 속한 초기 정점 v_i 를 선택한다.
2. 인접한 정점으로부터 가장 연관있는 정점 v_j 를 선택한다.
3. 클러스터 C_k 에 정점 v_i 와 v_j 를 병합(Merge)한다.
4. 클러스터 C_k 의 인접 정점의 집합 A 가 정점을 포함하고 있으면 단계 2로 간다.
5. 부집합 $(V - C_k)$ 가 원소를 포함하고 있으면 단계 1로 간다.
6. $v \notin C$ 인 V 의 각각의 정점을 이상점(Outlier)에 할당한다.

[그림 5] Algorithm FPNC

알고리즘의 첫 번째 단계는 가장 높은 차수를 갖는 초기 정점 v_i 를 선택하는 것이다. 무방향 그래프이기 때문에 정점 v_i 의 차수는 연결된 간선의 수이고, $d(v_i)$ 로 표기한다. 가장 높은 차수를 가진 정점을 초기 정점으로 선택하는 이유는 가장 높은 차수를 가진 정점을 포함하는 클러스터가 많은 정점과 연결이 되어 있어 큰 클러스터가 될 확률이 높기 때문이다. 만약 두 정점이 동일하게 가장 높은 차수를 갖게 되는 경우가 발생하면, 정점의 발생 빈도수가 다른 것보다 큰 정점이 초기 정점으로 선택된다. 차수도 같고 정점의 발생 빈도수도 같은 경우에는 사전 순서상 앞에 있는 정점을 선택한다. 이렇게 선택된 정점을 초기 클러스터 C_i 로 할당하고, 그 다음부터 각각의 연관있는 정점은 초기 클러스터 C_i 의 원소로 배치된다.

두 번째 단계는 인접한 정점으로부터 가장 연관있는 정점 v_j 를 선택하는 것이다. 인접 정점 집합(Adjacent Vertices Set) A 의 각각의 정점 v_j 와 초기 클러스터 C_i 의 유사도 $\text{Sim}(C_i, v_j)$ 를 구하여 가장 유사도가 큰 정점을 초기 클러스터 C_i 에 병합한다. 병합에 앞서 인접한 정점 v_j 는 하나의 원소를 가지고 있는 C_j 로 할당한다. 유사도는 상관관계, 신뢰도, 그리고 정규화된 간선 가중치를 사용한다.

클러스터 C_i 와 C_j 사이의 상관관계를 표현하는 상관관계 유사도 corr_{ij} 를 식(3)에 의해 계산된다.

$$\text{corr}_{ij} = \frac{P(C_i \cup C_j)}{P(C_i)P(C_j)} \quad (3)$$

$P(C_i)$ 와 $P(C_j)$ 는 각각 클러스터 C_i 와 C_j 의 확률이고, $P(C_i \cup C_j)$ 는 두 클러스터의 합이다. 식(3)에 의한 결과 값이 1보다 작으면 C_i 와 C_j 가 부정적으로 관련되어 있다는 것을 나타내고, 1보다 크면 긍정적으로 관련되어 있다는 것을 나타낸다. 그리고 결과 값이 1이면 C_i 와 C_j 는 서로 독립적이고 둘 사이에 아무런 상관관계가 없음을 의미한다. 본 논문에서는 상관관계 계산을 위한 과정을 지지도 값을 가지고 활용하므로 식(4)를 가지고 유사도를 계산한다. 식(4)에 사용된 $|T|$ 는 트랜잭션의 수이다.

$$\text{corr}_{ij} = \frac{|T| * \text{support_count}(C_i \cup C_j)}{\text{support_count}(C_i) * \text{support_count}(C_j)} \quad (4)$$

신뢰도 유사도는 식(5)로 계산된다. 클러스터 C_j 는 하나의 원소만을 가지고 있다.

$$\text{confidence}(C_i \Rightarrow C_j) = P(C_j | C_i) = \frac{\text{support_count}(C_i \cup C_j)}{\text{support_count}(C_i)} \quad (5)$$

마지막으로 간선 가중치 유사도는 식(6)에 의해 계산된다.

$$\text{edgeweight}(C_i, C_j) = \frac{\sum w_{ij}}{|E_{ij}|} \quad (6)$$

$|E_{ij}|$ 는 각각의 간선이 C_i 의 정점과 C_j 의 정점을 연결하는 간선 집합 $|E_{ij}|$ 의 카디널리티(Cardinality)이고, w_{ij} 는 $|E_{ij}|$ 에 포함된 간선의 지지도 값이다. 식(6)은 C_i 와 C_j 의 정점을 연결하는 간선들의 평균 가중치 값을 구한다. 세 가지 유사도 방법 중 하나를 선택하여 계산된 값 중에서 인접 정점 집합 A 에서 가장 유사도가 큰 정점을 선택하여 클러스터 C_j 에 정점으로 바꾼다. 여기에서 주목해야 할 점은 가장 유사도가 큰 정점은 클러스터 C_i 의 원소인 각각의 정점 v_i 에 관계를 갖고, E_{ij} 는 클러스터 C_i 와 C_j 사이의 간선들의 집합을 나타내는 점이다.

세 번째 단계는 두 클러스터를 하나의 클러스터 C_k 로 병합한다. 병합을 하기 전에 앞서 클러스터 C_k 에 지지도 값 $\text{support_count}(C_k)$ 를 할당한다. 여기에서 클러스터 C_k 의 지지도 값은 식(2)로 계산된 경로지지도를 의미한다. 그러므로 클러스터 C_k 에 모든 정점 V_k 는 연결된 경로로 되어 있다. 병합된 클러스터는 또한 Frequent Closed Itemset이다[5].

네 번째 단계에서는 반복 조건을 검사해 단계를 반복할지 결정한다. 클러스터 C_k 의 인접 정점 집합 A 가 원소를 가지고 있으면, 두 번째 단계로 돌아가 탐욕적(Greedy) 방법에 의해 A 의 원소 중 가장 유사한 정점을 선택해 알고리즘을 다시 수행한다. A 가 원소를 가지고 있지 않거나, 클러스터 C_k 가 연결된 경로가 아니면, 알고리즘은 클러스터 리스트 C 에 클러스터 C_k 를 첨가한다. 이때 클러스터 C_k 의 카디널리티가 2보다 작으면 클러스터는 없어지고, 없어진 클러스터의 정점은 정점 집합 V 에 남게 된다.

5 번째 단계까지는 클러스터를 발견하는 절차이고, 다섯 번째 단계는 클러스터의 생성을 계속 할 것인가에 대한 결정을 하는 단계이다. 루프(loop)를 지속하기 위해 클러스터의 초기 원소로 사용될 정점이 부집합 $(V - C_k)$ 에 남아 있는지를 확인한다. 초기 정점으로 사용될 정점이 남아 있지 않다면 알고리즘은 6단계로 넘어가고, 남아 있다면 1단계로

돌아가 알고리즘을 계속 수행한다.

마지막 여섯 번째 단계는 클러스터 리스트 C 에 포함된 어떤 클러스터에도 포함되지 않은 정점들에 대한 처리를 하는 단계이다. 하나의 정점을 가진 클러스터는 인정하지 않고 다시 정점 집합 V 로 반환하기 때문에, 클러스터 리스트 C 에 포함된 각각의 클러스터는 카디널리티가 2 이상이다. C 에 포함된 어떤 클러스터에도 속하지 않은 정점을 남겨진 정점들의 집합 R 에 포함시키고 다음과 같이 표기한다.

$$R = \{v \mid v \notin C_j, C_j \in C\} \quad (7)$$

R 에 속한 정점들은 이상점이라 부르고 연관규칙을 생성하는 작업에서 이러한 이상점들은 무시한다. 연관규칙 작업에서 제외되는 이유는 두 가지이다. 첫째, 이러한 이상점들의 대부분은 발생 빈도수가 크지 않아 빈발 항목 집합이 되기 어렵다. 둘째, 이상점들은 다른 정점과의 연관성이 작아 어느 클러스터에도 포함되지 않기 때문에 의미있고 유용한 규칙에 포함되기에 부족한 점이 있다. 이상점을 제외함으로 인해 의미가 적은 연관규칙은 생성되지 않고, 이상점들을 처리하는데 사용되는 시간과 메모리 같은 자원을 소모하지 않을 수 있게 된다.

5. 실험 및 평가

본 논문에서는 빈발 패턴 네트워크에서 아이템을 클러스터링하기 위해 두 가지 인공 데이터 집합(Synthetic Datasets)과 소매 시장 바구니 데이터 집합(Retail Market Basket Datasets)인 실제 데이터를 가지고 실험을 하였다. 인공 데이터집합은 IBM Almaden Research Group에서 만든 생성프로그램을 사용하여 두 가지 인공 데이터 집합을 생성하였다. 첫 번째 데이터는 10,000개의 아이템, 트랜잭션당 평균 10개의 아이템, 빈발 항목 집합당 평균 4개의 아이템, 그리고 100,000개의 트랜잭션을 포함하는 데이터 집합이다(T10.I10.100K with 10K items). 두 번째 데이터는 10,000개의 아이템, 트랜잭션당 평균 40개의 아이템, 빈발 항목 집합당 평균 10개의 아이템, 그리고 100,000개의 트랜잭션을 가지고 있는 데이터 집합이다(T40.I10.100K with 10K items).

실제 데이터는 연관규칙 알고리즘의 효율성을 평가하기 위해 사용되는 소매 시장 바구니 데이터 집합이다. 이 소매 데이터는 5,133명의 고객으로부터 생성된 88,163개의 트랜잭션을 포함하고 있다. 평균적으로 하나의 트랜잭션에 13개의 아이템이 포함되어 있지만, 대부분 고객들은 쇼핑을 할 때 7개에서 11개 사이의 아이템을 구입 한다. 이 실제 데이터는 공개적으로 사용할 수 있고, 이 데이터에 대해 더 세부적인 정 보는 [6]에서 얻을 수 있다.

빈발 패턴 네트워크에서 FPNC 알고리즘으로 생성된 클러스터를 평가하기 위해 비용 함수(Cost Function)를 사용하였다. 클러스터링은 자율학습(Unsupervised Learning)이므로, 클러스터링의 평가 정도는 일반적으로 비용함수에 의해 측정된다. 네트워크에서 생성된 클러스터는 두 정점 사이의 거리를 계산하지 않기 때문에 거리 제곱 합 같은 표준 비용 함수를 사용하지 않는다. FPNC 알고리즘에 의해 생성된 클러스터를 평가하기 위해 정규화 오류율(Normalized Error Rate)을 사용하였고, NER이라 표기한다.

클러스터 C_i 의 오류율(Error Rate, ER)은 식(8)과 같이 계산된다.

$$ER(C_i) = \frac{|E|}{|C_i|} \quad (8)$$

$|E|$ 는 오류 집합(Error Set) E 의 카디널리티이고, 오류집합은 어떤 클러스터에 대하여 내부 유사도(Intra-similarity)보다 상호 유사도(Inter-similarity)가 더 큰 정점들의 집합이다. $|C_i|$ 는 클러스터 내부에 포함된 정점의 수를 표현한다.

정규화 오류율은 식(9)로 계산된다.

$$NER(C) = \frac{\sum ER(C_i)}{|C|} \quad (9)$$

$|C|$ 는 FPNC 알고리즘에 의해 생성된 클러스터의 수이다. 클러스터의 모든 아이템이 실제 영향력 있는 군집의 멤버라는 가정하에 클러스터의 정확성으로써 이 식을 해석할 수 있다. 클러스터에 상호 유사도보다 작은 내부 유사도를 가지는 정점(Missing Item)이 존재하면 NER은 증가하게 된다.

5.1 인공 데이터 집합의 실험 결과

두 인공 데이터는 100,000개의 트랜잭션으로 이루어져 있다. (T10I4D100K)는 총 870개의 아이템을 가지고 있고, (T40I10D100K)는 총 1,647개의 아이템을 가지고 있다. [표 1]은 상관관계 유사도를 사용하여 FPNC 알고리즘을 수행한 후, 정규화 오류율로 클러스터를 평가한 내용이다. 표는 0.1%에서 0.5%까지 각 단계별로 최소지지도 값을 조절하여 실험하였다. $|C|$ 는 생성된 클러스터의 수, $|I_c|$ 는 모든 클러스터에 포함된 아이템의 수, $|E|$ 는 여러 집합 E 에 포함된 아이템의 수, $AVG|E|$ 는 평균 여러 집합의 수이고 NER은 식(9)로 계산된 값을 보여준다. [표 2]는 0.7%의 신뢰도, [표 3]은 정규화 간선 가중치의 유사도를 가지고 실험한 결과이다. 결과로부터 빈발 패턴 네트워크에서 클러스터링은 신뢰도 유사도 방법이 가장 높은 정확성을 보임을 알 수 있다. 상관관계와 간선 가중치는 비슷한 수의

아이템을 포함한 클러스터가 생성되었고 간선 가중치가 더 정확함을 보였다.

[표 1] Error Rate with Correlation Similarity

Data set	T10I4D100K					T40I10D100K				
	Support (%)	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4
C	210	172	128	95	68	38	14	9	5	5
Ic	688	567	412	270	173	94	31	20	13	12
E	75	38	21	10	10	29	5	15	8	7
AVG E	0.36	0.22	0.16	0.11	0.15	0.76	0.36	1.67	1.6	1.4
NER	0.090	0.056	0.046	0.037	0.055	0.261	0.178	0.417	0.4	0.35

[표 2] Error Rate with Confidence Similarity

Data set	T10I4D100K					T40I10D100K				
	Support (%)	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4
C	97	66	40	19	7	8	6	6	4	4
Ic	359	244	146	68	26	21	15	13	9	9
E	11	2	3	0	0	0	0	0	0	0
AVG E	0.11	0.03	0.08	0.0	0.0	0.0	0.0	0.0	0.0	0.0
NER	0.027	0.008	0.019	0.0	0.0	0.0	0.0	0.0	0.0	0.0

[표 3] Error Rate with Edge Weight Similarity

Data set	T10I4D100K					T40I10D100K				
	Support (%)	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4
C	218	184	138	91	72	32	8	7	4	3
Ic	656	550	409	263	174	87	25	18	12	10
E	120	44	32	14	7	1	0	0	0	0
AVG E	0.55	0.24	0.23	0.15	0.10	0.03	0.0	0.0	0.0	0.0
NER	0.057	0.052	0.054	0.044	0.037	0.016	0.0	0.0	0.0	0.0

5.2 소매시장바구니 데이터의 실험 결과

인공 데이터 집합의 실험결과로부터 클러스터 사이의 가장 중요한 유사도는 신뢰도임을 알 수 있다. 실제 시장바구니 데이터 집합에는 신뢰도와 간선 가중치 유사도를 적용하여 실험을 하였다. 데이터 집합은 5,133명의 고객으로부터 생성되었고, 총 942개의 아이템과 88,163개의 트랜잭션으로 구성되어 있다. 트랜잭션은 평균 13개의 아이템을 가지고 있다. [표 4]는 실제 데이터에 관한 결과를 보여주고 있다. 이 결과로부터 실제 데이터에 대한 결과도 인공 데이터에서와 마찬가지로 클러스터링 정확성에 대해 신뢰도 유사도가 강한 영향을 미치는 것을 알 수 있다.

[표 4] Error Rate with Similarity

Data set	Confidence					Edge Weight				
	Support (%)	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4
C	8	6	6	4	4	32	8	7	4	3
Ic	21	15	13	9	9	87	25	18	12	10
E	0	0	0	0	0	1	0	0	0	0
AVG E	0.0	0.0	0.0	0.0	0.0	0.031	0.0	0.0	0.0	0.0
NER	0.0	0.0	0.0	0.0	0.0	0.016	0.0	0.0	0.0	0.0

지지도 값을 변경함으로 인해 생성되는 클러스터의 수를 조절할 수 있다. 더 큰 값의 지지도를 설정하면 더 적은 수의 클러스터가 생성된다. 클러스터링 방법을 적용함으로써 더 응축된 연관 규칙을 가질 수 있다. 더 적은 클러스터를 생성함으로 인해 이상점에 속한 아이템은 더 늘어난다. 하지만 지지도 값을 낮게 적용하거나 트랜잭션이 추가되면, 아이템에 해당하는 정점의 발생 빈도가 주어진 최소지지도를 만족 할 수 있고, 이상점에 속하던 아이템이 클러스터에 속 할 수 있기 때문에 생성된 이 상점집합은 제거하지 않는다.

6. 결론 및 향후 연구

대량의 데이터에 담긴 유용한 정보를 발견하기 위해 최소지지도를 만족하는 빈발 항목 집합을 발견하고, 이러한 빈발 항목 집합으로부터 최소신뢰도를 만족하는 연관 규칙을 생성하는 것은 데이터마이닝 알고리즘에서 중요한 작업이 되었다. 클러스터링은 인간의 중요한 행위 중 하나로써, 전체적인 데이터의 분포 패턴과 데이터 속성들 사이에 존재하는 유용한 상관관계를 찾을 수 있게 해준다.

본 논문에서는 연관규칙 마이닝을 효율적으로 할 수 있는 빈발 패턴 네트워크를 제안하였다. 빈발 패턴 네트워크는 데이터베이스에 존재하는 트랜잭션을 네트워크로 옮김으로써 수많은 데이터베이스 접근을 하지 않아도 될 뿐만 아니라, 빈발 패턴 트리처럼 간결하고 압축된 정보를 표현하는 자료 구조이다. 경로지지도를 통해서 네트워크 상에서 k-아이템집합($k > 2$)의 지지도를 계산할 수 있는 새로운 방법을 소개하였다.

클러스터에 포함되지 않은 생성 빈도수가 낮은 아이템을 연관규칙 마이닝 작업에서 제외함으로써 흥미롭지 않은 규칙의 생성을 미연에 방지한다. 탐색 공간을 FPNC 알고리즘으로 생성된 클러스터로 국한시켜 연관규칙 발견을 위한 작업을 효율적으로 할 수 있게 되었다.

실험의 결과는 서로 연관된 아이템의 클러스터, 클러스터에 포함된 아이템의 수, 여러집합에 포함된 아이템의 수와 FPNC 알고리즘을 통해 생성된

클러스터의 정 확성을 보여주었다.

향후 연구로는 다양한 클러스터링 방법을 사용하여 제안한 빈발 패턴 네트워크에서 가장 효율적인 클러스터링 방법을 찾아내는 비교 연구가 필요하다.

Reference

- [1] Han, J., and Kamber, M. (2005). "Data Mining : Concepts and Techniques," Morgan Kaufmann Publishers
- [2] Agrawal, R., Aggarwal C. C., and Prasad, V.V.V. (2000). "A Tree Projection Algorithm For Generation of Frequent Itemsets," Journal of Parallel and Distributed Computing 61(3), pp. 350-371.
- [3] Han, J., Pei, J., Yin, Y. and Mao, R. (2004). "Mining Frequent Patterns without Candidate Generation : A Frequent-Pattern Tree Approach," Journal of Data Mining and Knowledge Discovery, 8, pp. 53-87.
- [4] Lakshmanan, L. V. S., Leung, C. K.-S. and Ng, R. T. (2000). "Segment Support Map: Scalable Mining of Frequent Itemsets," ACM SIGKDD Explorations Newsletter, 2(2), pp. 21-27.
- [5] Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999) "Discovering Frequent Closed Itemsets for Association Rules," In Proceedings of the 7th International Conference on Database Theory, pp. 398 – 416.
- [6] <http://fimi.cs.helsinki.fi/data/>
- [7] Agrawal R., and Srikant, R. (1994). "Fast Algorithms for Mining Association Rules," Proceedings of the 20th VLDB(Very Large Data Bases) Conference, pp. 580 - 592.
- [8] Han, E-H, Karypis, G., Kumar, V., and Mobasher, B. (1997). "Clustering based on association rule hypergraphs," In Proccedings of SIGMOD'97 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD'97), May.
- [9] Mohammed J. Zaki. (2000). "Generating non-redundant association rules," Conference on Knowledge Discovery in Data, Proceedings of the sixth ACM SIGKDD, pp. 34 – 43.
- [10] Srikant, R, Vu, Q, and Agrawal, R. (1997). "Mining Association Rules with Item Constraints," Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining(KDD), pp. 67-73.
- [11] Lent, B., Swami, A., and Widom, J. (1997). "Clustering Association Rules," In proceeding of the 13th International Conference on Data Engineering, 220-231.