한국지능정보시스템학회논문지 제14권 제1호 2008년 3월(pp.1~17)

빈발 패턴 네트워크에서 아이템 클러스터링을 통한 연관규칙 발견

오경진
인하대학교 컴퓨터정보공학과
(okjkillo@eslab.inha.ac.kr)

정진국
인하대학교 컴퓨터정보공학과
(gj4024@eslab.inha.ac.kr)

하인애
인하대학교 컴퓨터정보공학과
(inay@eslab.inha.ac.kr)

조근식
인하대학교 컴퓨터정보공학과
(gsjo@eslab.inha.ac.kr)

데이터 마이닝은 대용량의 데이터에 숨겨진 의미있고 유용한 패턴과 상관관계를 추출하여 의사결정에 활용하는 작업이다. 그 중에서도 고객 트랜잭션의 데이터베이스에서 아이템(item) 사이에 존재하는 연관규칙을 찾는 것은 중요한 일이 되었다. Apriori 알고리즘 이후 연관규칙을 찾기 위해 대용량의 데이터베이스로부터 압축된 의미있는 정보를 저장하기 위한 데이터 구조와 알고리즘들이 많이 제안되어 왔다. 연관규칙을 발견하기 위한 기존의 연구들은 모든 규칙을 찾아내지만, 사람이 분석하기에 너무 많은 규칙이 생성되기 때문에 규칙을 분석하기 위한 일 또한 많은 과정을 거쳐야 한다.

본 논문에서는 빈발 패턴 네트워크(Frequent Pattern Network)라 부르는 자료 구조를 제안하고 이를 활용하였다. 네트워크는 정점과 간선으로 구성되며 정점은 아이템을 표현하고, 간선은 두 아이템 집합을 표현한다. 아이템의 빈도수를 이용하여 빈발 패턴 네트워크를 구성하고, 아이템 사이의 유사도를 측정한다. 그리고 클러스터 내의 아이템과는 유사도가 높고, 다른 클러스터의 아이템과는 유사도가 낮도록 클러스터를 생성한다. 클러스터를 이용해 연관규칙을 생성하고 실험을 통해 Apriori와 FP Growth 알고리즘과의 성능을 비교를 하였다. 그 결과 빈발 패턴 네트워크에서 신뢰도 유사도를 이용하는 것이 클러스터의 정확성을 높여줌을 볼 수 있었다. 그리고 전통적인 방법과 비교를 통해 빈발 패턴 네트워크를 이용하는 것이 최소지지도에 유연성을 가짐을 알 수 있었다.

논문접수일 : 2007년 02월 게재확정일 : 2007년 10월 교신저자 : 하인애

1. 서론

데이터 마이닝(Data Mining)은 대량의 데이터 속에 숨겨진 의미있고 유용한 패턴(Pattern)과 상관관계를 추출하여 의사 결정에 이용하는 작업이다(Han and Kamber, 2005). 정보화 혁명 이후 정보기술의 가속적 발전으로 인해 매일 쏟아져 나오는 데이터의 양은 사람의 힘으로는 도저히 소화할 수 없을 정도로 방대해졌다(Tan et. al, 2006). 데이터의 양이 급속도로 증가하는데 반해 필요한 정보를 접하기 어려워 정보의 빈곤현상을 겪고 있어서 데이터에 담긴 정보를 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(Agrawal et. al, 1993). 연관규칙(Association Rule)은 데이터 안에

존재하는 아이템(Item) 간의 상관관계를 찾아내는 작업이며, 마케팅(Marketing)에서는 손님의 장바구니에 들어있는 아이템 간의 관계를 알아본다는 의미에서 시장바구니분석(Market Basket Analysis)이라고도 한다(Michael and Gorden, 1997). 연관규칙은 서비스의 교차판매, 매장진열, 첨부우편, 사기적발 등의 다양한 분야에 활용되고 있다.

전형적인 연관규칙 알고리즘은 두 단계로 이뤄진다. 첫 번째 단계는 최소지지도(Minimum Support)를 만족하는 빈발 항목집합(Frequent Itemsets)을 찾는 것이다. 그리고 두 번째 단계에서는 모든 빈발 항목집합으로부터 최소신뢰도를 만족하는 규칙을 생성한다. Apriori 알고리즘(Agrawal and Srikant, 1994) 이후 빈발 항목집합을 찾기 위한 알고리즘들이 많이 연구되어 왔고, 대량의 데이터베이스로부터 압축된 의미있는 정보를 저장하기 위한 자료구조도 제안되어왔다(Han et. al, 2004). (Agrawal and Srikant, 1994)의 방법들은 데이터에 존재하는 모든 빈발 항목집합을 찾아낸다. 그리고 이를 바탕으로 모든 연관규칙을 효율적으로 발견할 수 있다. 하지만 최소지지도를 만족하는 모든 규칙을 발견하기 때문에, 규칙을 분석하기 위해서 추가적인 많은 과정을 거쳐야 한다. 이런 많은 규칙 때문에 사람이 분석하는 것은 불가능하다고 볼 수 있다. 그리고 연관규칙 마이닝을 하는 전체 과정에서 빈발 항목집합을 찾는 과정에 대부분의 비용을 차지한다(Liu et. al, 1999).

본 논문에서는 빈발 패턴 네트워크(Frequent Pattern Network)라 부르는 네트워크 자료 구조를 제안하고 이를 활용하여 연관규칙을 발견한다. 빈발 패턴 네트워크는 아이템의 빈도수를 이용하여 형성하는데, 아이템을 표현하기 위한 정점(Vertex)과 두 아이템 집합을 표현하기 위한 간선(Edge)으로 구성되어 있다. 빈발 패턴 네트워크에서 연관규칙 발견을 위해 클러스터링을 이용한다. 클러스터(Cluster) 내의 아이템과의 유사도는 높고, 다른 클러스터에 존재하는 아이템과는 유사도가 낮도록 클러스터를 형성한다. 클러스터링(Clustering) 방법은 생성되는 규칙의 수를 줄이고, 데이터 집합 전체의 분포와 특징을 쉽게 알 수 있게 해준다.

빈발 패턴 네트워크를 이용한 연관규칙 생성은 최소지지도에 따른 네트워크의 재구성이 필요 없기 때문에 빈발 항목집합을 찾아내는데 드는 시간을 줄일 수 있다.

본 논문의 구성은 다음과 같다. 제 2장에서는 연관규칙 마이닝에 대하여 알아보고, 관련된 연구를 분석한다. 제 3장에서는 본 논문에서 제안한 빈발 패턴 네트워크를 설명한다. 그리고 제 4장에서는 빈발 패턴 네트워크에서의 클러스터링과 연관규칙 생성을 설명한다. 제 5장에서는 구현 및 실험을 통해 빈발 패턴 네트워크에서 생성된 클러스터를 평가하고, 이를 바탕으로 연관규칙을 생성하여 기존의 방법들과 결과를 비교한다. 마지막으로, 제 6장에서는 실험을 통해 얻은 내용으로 결론을 맺는다.

2. 연관규칙 마이닝

연관규칙은 한 항목의 그룹과 다른 항목의 그룹 사이에 존재하는 연관성을 규칙의 형태로 표현한 것이다(Pasquier et. al, 1999). 연관규칙 마이닝은 사용자에 의해 적절하게 입력된 지지도(Support)와 신뢰도(Confidence)라는 척도를 이용하여 데이터 상호간의 연관성을 파악할 수 있다. 신뢰도는 아이템 집합 A를 포함하고 있는 트랜잭션(Transaction) 중 아이템 집합 B 역시 포함하고 있는 트랜잭션의 비율을 의미하며, 지지도는 모든 트랜잭션에 대해 아이템 집합 A, B를 둘 다 포함하고 있는 트랜잭션의 비율을 나타낸다. 연관규칙은 일반적으로 다음

과 같이 정의되어 진다. $I=\{i_1, i_2, \ldots\ldots, i_m\}$를 아이템들의 집합이라 하고, 각 트랜잭션은 I에 속하는 아이템들의 부분 집합으로 구성되어진다. 이때 연관규칙은 아래와 같은 형태로 기술되어질 수 있다.

$$A \rightarrow B(A \subset I,\ B \subset I,\ \text{and}\ A \cap B = \varnothing)$$

A → B가 내포하고 있는 의미는 트랜잭션에 A 항목 집합이 존재하면 B 항목 집합도 트랜잭션 내에 동반하여 나타나는 경향이 있음을 뜻한다. 이와 같이 연관규칙을 찾기 위해 빈발 항목집합을 찾기 위한 알고리즘들이 많이 연구되어 왔고, 대량의 데이터베이스로부터 압축된 의미있는 정보를 저장하기 위한 자료 구조도 제안되어 왔다. 몇 가지 알고리즘을 살펴보면, 먼저 Apriori 알고리즘은 최소지지도를 만족하는 모든 빈발 항목집합을 생성한다. 하지만 패턴의 길이가 길 경우, 생성된 후보 항목집합(Candidate Set)이 최소지지도를 만족하는가에 대한 확인을 위해 계속적인 데이터베이스 접근을 해야 한다. (Srikant et. al, 1997)에서는 아이템 제약을 사용하여 후보 항목집합의 생성 시간을 줄이기 위한 방법을 제안하였지만, 데이터베이스의 접근은 계속 이루어져야 한다.

그리고 FP Growth 알고리즘(Han et. al, 2004)은 간결하고 압축된 정보를 표현하는 빈발 패턴 트리(FP Tree)를 생성한다. 빈발 패턴 트리는 후보 항목집합을 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.

단편 지지도 맵(Segment Support Map)(Lakshmanan et. al, 2000)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.

갈루아 래티스(Galois Lattice) 구조(Zaki, 2000)는 Frequent Closed Itemset(Pasquier et. al, 1999)을 기반으로 정보의 손실 없이 적은 수의 규칙을 발견하기 위해 사용되었다.

위의 방법들은 최소지지도를 만족하는 모든 빈발 항목집합을 찾아내는 과정에 많은 비용을 소비해야하는 문제점이 있다. 본 논문에서는 이러한 문제를 해결하기 위해 빈발 항목집합을 생성하지 않고 네트워크에 존재하는 아이템을 클러스터링하여 연관규칙을 생성하고자 한다.

클러스터링은 클러스터 안의 아이템끼리는 높은 유사성을 갖게 하고 다른 클러스터들의 아이템과는 큰 상이성을 갖도록 클러스터를 만들어가는 과정이다. 클러스터링은 데이터 마이닝, 통계학, 생물학, 그리고 기계 학습 분야 등에 많이 이용되고 있다(Han and Kamber, 2005).

(Han et. al, 1997)에서는 연관규칙 하이퍼그래프(Association Rule Hypergraph)를 이용하여 아이템을 클러스터링하고, 아이템 클러스터를 이용하여 트랜잭션을 클러스터링한다. 이 방법은 연관규칙의 조건부와 결론부의 아이템들로 하이퍼그래프를 구축하여 클러스터링 과정을 진행한다. 아이템을 클러스터링하기 위해 연관규칙을 사용하였기 때문에 빈발 항목집합을 발견하는 비용을 소비해야 한다.

본 논문에서는 아이템의 발생 빈도수(Frequency)를 기반으로 빈발 패턴 네트워크를 형성하여 아이템 클러스터링을 한다. 클러스터로 탐색 공간을 국한시킴으로써 효율적인 데이터 마이닝 알고리즘을 구성할 수 있다. 클러스터를 이용하여 생성되는 연관규칙의 수를 줄일 수 있고, 유사성이 있는 아이

템을 클러스터링함으로써 데이터 분포 패턴과 데이터 속성들 사이에 존재하는 흥미 있고 유용한 상관관계를 찾을 수 있다. 또한 최소지지도 값의 변화에 따라 빈발 패턴 네트워크를 다시 구축할 필요가 없기 때문에, Apriori나 FP Growth 알고리즘과 달리 최소지지도에 유연함을 가지게 됨을 실험을 통해 보여줄 것이다.

3. 빈발 패턴 네트워크

3.1 빈발 패턴 네트워크의 구성

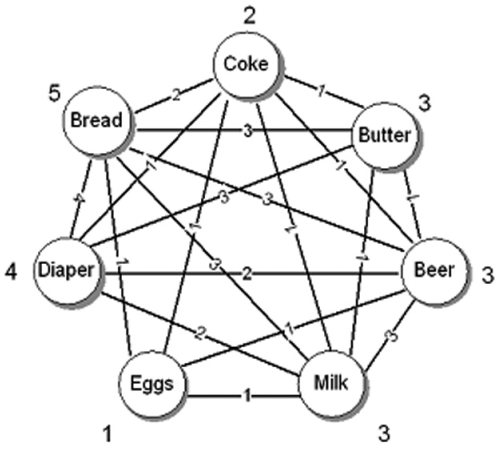
데이터베이스 안의 모든 아이템은 오름차순으로 존재한다고 가정을 하고, 데이터베이스로부터 모든 트랜잭션을 빈발 패턴 네트워크로 옮긴다. 네트워크는 연관규칙 마이닝(Mining) 작업을 하는 동안 데이터베이스를 표현하는 간결하고 압축된 자료 구조이다. 이 네트워크는 무방향(Undirected)이고, 자기반복(Self-loop)이 없는 가중치 그래프(Weighted Graph)이다.

빈발 패턴 네트워크는 정점들의 집합과 간선들의 집합으로 구성된다. 하나의 정점은 하나의 아이템에 해당하며, 각 정점은 아이템 식별자, 정점 빈도수, 그리고 간선 집합의 세 가지 속성을 가지고 있다. 아이템 식별자는 아이템의 이름을 가지고, 정점 빈도수는 아이템의 발생 빈도수를 표현하는 지지도를 저장한다. 그리고 간선 집합은 정점에 연결된 간선들의 집합이다. 사전적으로 마지막에 위치한 정점은 간선들을 갖지 않는다. 하나의 간선은 두 정점을 연결하고, 양단의 두 정점(시작 정점과 끝 정점)과 간선 빈도수의 세 가지 속성을 가진다. 시작 정점과 끝 정점은 정점에 연결을 나타내고, 시작 노드가 사전 편집상 더 빨리 발생한다. 간선 빈도수는 두 정점(아이템)에 공통으로 발생하는 빈도수를 저장하고, 두 정점의 연결 강도를 표현한다. 정점 u와 v를 잇는 간선은 edge(u, v) 또는 e_{uv}로 표기한다.

정점 u와 v 사이의 정점들과 간선들이 나오는 순차열을 경로(Path)라 하고, path(u, v)로 표기한다. 정점 u를 시작 정점(Start Vertex)이라 하고, 정

TID	Item
1	Bread Diaper Butter
2	Coke Bread Eggs Milk Beer
3	Coke Bread Diaper Butter
4	Bread Diaper Milk Beer
5	Bread Diaper Milk Beer Butter

(a)



(b)

[그림 1] 빈발 패턴 네트워크

점 v를 끝 정점(End Vertex)이라 한다. path(u, v)의 다른 정점들은 내부 정점(Internal Vertex)이라 한다. 본 논문에서 경로는 클러스터에 포함된 모든 아이템의 순차열을 뜻한다. 따라서 4개의 아이템을 포함하는 클러스터의 경로에 포함되는 아이템은 시작 정점, 끝 정점 그리고 내부 정점 2개로 이루어진다. 예를 들어, 클러스터가 Bread, Diaper, Beer, Milk로 구성될 때 path(Bread, Diaper)는 시작 정점을 Bread, 끝 정점을 Diaper, 내부 정점을 Beer, Milk를 가지는 경로가 된다. 경로에 대한 자세한 설명은 3.2와 3.4에서 살펴보도록 하겠다.

[그림 1]은 빈발 패턴 네트워크의 간단한 예이다. (a)는 데이터베이스에 축적된 트랜잭션을 나타내고, (b)는 트랜잭션을 빈발 패턴 네트워크에 옮겼을 때의 모습이다. 아이템은 원으로 표현하고, 두 아이템을 잇는 선은 간선으로 표현한다. 원 내부의 문자는 아이템의 이름을 나타내고, 아이템 위의 숫자는 트랜잭션에 나타나는 아이템의 빈도수를 나타낸다. Bread 위의 숫자 5는 트랜잭션에 Bread라는 이름을 가진 아이템이 다섯 번 존재함을 의미한다. Bread와 Diaper를 잇는 간선 위의 숫자 4는 트랜잭션에서 Bread와 Diaper가 동시에 존재하는 경우가 네 번임을 의미한다.

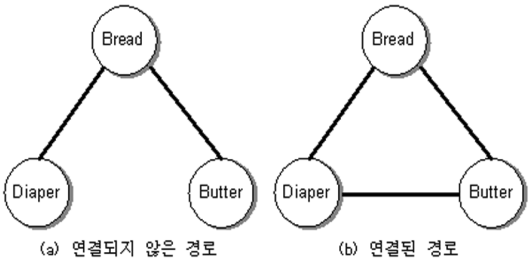
3.2 연결된 경로(Connected Path)

path(u, v)에 있는 정점들의 집합을 X라 할 때, X는 다음과 같이 표현한다.

$$\{X = x | x \in V,\ x \in path(u,\ v)\}$$

여기에서 V는 빈발 패턴 네트워크에 포함된 정점들의 집합이다. X안에서 임의의 두 정점 x와 y에 대해 간선 edge(x, y)가 존재하면 X를 연결된 경로라 한다.

예를 들어, [그림 2]의 (a)는 네트워크가 세 정점 Bread, Diaper, Butter와 두 간선 edge(Bread, Diaper), edge(Bread, Butter)로 구성되어 있다. (a)는 edge(Diaper, Butter)가 존재하지 않기 때문에, path(Bread, Diaper)는 연결된 경로가 아니다. 연결된 경로가 되려면 edge(Diaper, Butter)를 포함한 (b)와 같이 되어야 한다. 그러므로 연결된 경로가 되려면 그래프는 완전히 연결되어야 하고, 이렇게 연결된 경로는 하나의 패턴이 될 수 있다.



[그림 2] 연결된 경로

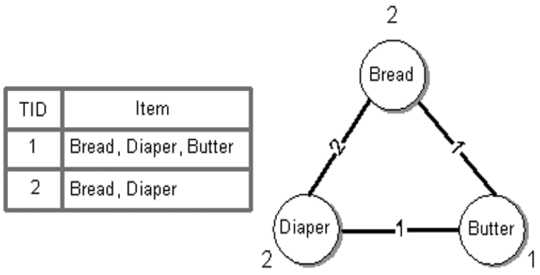
3.3 중첩수(Countfold)

중첩수는 경로 지지도를 계산하기 위한 방법으로, 아이템이 얼마나 빈번하게 발생하는지, 다른 아이템들과 얼마나 연관성이 있는지를 알아보기 위한 것이다.

예를 들어서, 네트워크의 경로 안에 세 정점들의 집합을 X라 하고 x와 y가 X에 포함된다고 할 때, x≠y이고 e_{xy}는 x에서 y로의 간선이다. 정점 x의 빈도수는 count(x), 간선 e_{xy}의 빈도수는 count(e_{xy})로 표기한다. X에서 정점 x의 중첩수를 countfold(x)라 표시하고 식 (1)을 이용하여 계산한다.

$$countfold(x) = \left[\sum_{e_{xy} \in E} count(e_{xy})\right] - count(x) \quad (1)$$

정점 x에 연결된 모든 간선의 가중치를 더한 다음, 정점 x의 가중치를 빼면 중첩수를 구할 수 있다.



[그림 3] 중첩수의 예

[그림 3]은 두 트랜잭션 {Bread, Diaper, Butter}와 {Bread, Diaper}로 이루어졌다. 두 트랜잭션을 세 정점 Bread, Diaper, Butter와 간선 edge(Bread, Diaper), edge(Bread, Butter), edge(Diaper, Butter)로 구성된 네트워크로 옮길 수 있다. 각 정점의 빈도수는 count(Bread) = 2, count(Diaper) = 2, count (Butter) = 1이고, 각 간선의 빈도수는 count(eBreadDiaper) = 2, count(eBreadButter) = 1, count(eDiaperButter) = 1이다. 정점 Bread의 중첩수 countfold(Bread)를 위의 식으로 계산하면 1이다.

3.4 경로지지도(Path Support)

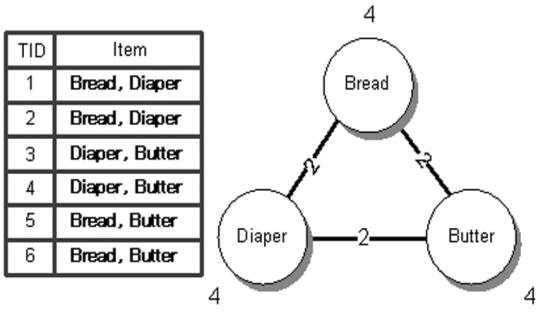
경로지지도는 경로 전체의 지지도 값(Support Value)이다. 빈발 패턴 네트워크에서 클러스터링은 Apriori 알고리즘에서와 같이 모든 빈발 항목집합을 생성하지 않고, 최소지지도를 만족하고 연관성이 높은 아이템을 클러스터로 병합하는 과정을 거친다. 이러한 차이 때문에 k-아이템집합의 지지도를 계산하는 방법이 달라진다. [그림 3]에서와 같이 3-아이템집합 {Bread, Diaper, Butter}가 빈발 항목집합의 의미를 만족하는 경로가 되기 위해서는 edge(Bread, Diaper), edge(Diaper, Butter), edge(Bread, Butter)가 모두 최소지지도를 만족해야한다. 또한 클러스터에 새로운 아이템이 포함될 경우에도 k-아이템집합의 경우 ${}_kC_2$의 간선의 가중치가 최소지지도를 만족해야하고 이를 계산해야 한다. 따라서 클러스터의 모든 아이템을 표현하는 경로의 지지도를 계산하여 최소지지도를 만족하면 클러스터에 포함된 모든 아이템은 최소지지도를 만족하는 빈발 항목집합이 된다.

빈발 패턴 네트워크에서 정점 x에서 정점 z까지 경로지지도를 f(x, z)로 표기하고, 식 (2)로 계산한다. path(x, z)에서 임의의 두 정점 a와 b는 a≠b이고, V와 E는 빈발 패턴 네트워크에서 각각 정점과 간선의 집합이다.

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

중첩수를 사용하여 [그림 3]의 {Bread, Diaper, Butter}같은 k-아이템집합 (k>2)의 지지도 값을 계산할 수 있다. 경로지지도는 중첩수와 간선 가중치 중 큰 값을 가지게 되는데, [그림 3]의 모든 아이템의 중첩수는 1이고, 간선의 가중치중 가장 작은 값은 1이다. 따라서 [그림 3]의 경로 지지도는 1이 된다.

하지만 [그림 4]에서와 같은 트랜잭션이 데이터베이스에 발생하였을 경우, 식 (2)를 사용하여 네



[그림 4] 논리적 패턴

트워크의 3-아이템집합 {Bread, Diaper, Butter}의 정확한 지지도 값을 계산할 수 없다.

[그림 4]에서 데이터베이스는 여섯 개의 트랜잭션을 가지고 있고, 이것을 빈발 패턴 네트워크로 옮길 수 있다. 데이터베이스에 3-아이템집합 {Bread, Diaper, Butter}을 포함하는 트랜잭션은 존재하지 않는다. 하지만 네트워크에는 3-아이템집합 {Bread, Diaper, Butter}이 존재하게 되고, 경로지지도 계산식에 의해 계산된 아이템집합의 경로지지도 값은 2이다. 따라서 [그림 4]의 경우는 존재하지 않는 3-아이템집합 {Bread, Diaper, Butter}이 계산된다. 이러한 경우 실질적으로 경로가 빈발 항목집합이 아니고, 존재하지 않지만 네트워크에서 경로가 연결된 경로이며, 경로지지도가 주어진 최소지지도 임계값 Θ와 동일하거나 그보다 크게 된다. 이러한 경로를 논리적 패턴이라 부른다.

4. 클러스터링과 연관규칙 발견

본 장에서는 빈발 패턴 네트워크에서 아이템 클러스터링을 통한 연관규칙 발견을 설명한다.

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

4.1 클러스터링 알고리즘

빈발 패턴 네트워크에서 클러스터링 알고리즘을 FPNC(Frequent Pattern Network Clustering)로 부른다. FPNC 알고리즘은 네트워크에 존재하는 연관있는 정점을 클러스터에 추가한다. 클러스터는 주어진 최소지지도를 만족하는 아이템의 집합으로 이루어진다. [그림 5]와 같이 정점의 집합 V, 간선의 집합 E, 최소지지도 Θ, 최소신뢰도 Ic를 입력으로 받아 알고리즘을 수행한 후, 클러스터 C의 집합을 결과물로 얻는다. 알고리즘은 여섯 단계로 이루어져 있고, 클러스터링을 하기 위해 모든 정점을 처리한 후 종료된다.

알고리즘의 첫 번째 단계는 가장 높은 차수를 갖는 초기 정점 v_i를 선택하는 것이다. 무방향 그

Algorithm FPNC

Input :
- 정점 v_1, v_2, …… , v_n의 집합 V
- 간선 e_{12}, e_{13}, …… , e_{m-1m}의 집합 E
- 최소지지도 Θ
- 최소신뢰도 Ic

Output :
- 클러스터 C의 집합

Method :
1. V에 속한 초기 정점 v_i를 선택한다.
2. 인접한 정점으로부터 가장 연관있는 정점 v_j를 선택한다.
3. 클러스터 C_k에 정점 v_i와 v_j를 병합(Merge)한다.
4. 클러스터 C_k의 인접 정점의 집합 A가 정점을 포함하고 있으면 단계 2로 간다.
5. 부집합 (V - C_k)가 원소를 포함하고 있으면 단계 1로 간다.
6. v∉C인 V의 각각의 정점을 이상점(Outlier)에 할당한다.

[그림 5] 알고리즘 FPNC

래프이기 때문에 정점 v_i의 차수는 연결된 간선의 수다. 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 같을 경우에는 사전 순서상 앞에 있는 정점을 선택한다. 이렇게 선택된 정점을 초기 클러스터 C_i로 할당하고, 그 다음부터 각각의 연관있는 정점은 초기 클러스터 C_i의 원소로 배치된다.

두 번째 단계는 인접한 정점으로부터 가장 연관있는 정점 v_j를 선택한다. 인접 정점 집합(Adjacent Vertices Set) A의 각각의 정점 v_j와 초기 클러스터 C_i의 유사도 Sim(C_i, v_j)를 구하여 가장 유사도가 큰 정점을 초기 클러스터 C_i에 병합한다. 병합에 앞서 인접한 정점 v_j는 하나의 원소를 가지고 있는 C_j로 할당한다. 유사도는 상관관계(correlation), 신뢰도(confidence), 정규화된 간선 가중치(edge-weight)를 사용한다. 상관관계는 연관규칙의 개선도(lift)와 같은 개념이고, 신뢰도는 연관규칙의 신뢰도를 사용한 것이다. 간선기반의 유사도는 연결된 모든 간선의 가중치의 합으로 계산된다(Jay et. al, 1997).

클러스터 C_i와 C_j사이의 상관관계를 표현하는 상관관계 유사도 $corr_{ij}$를 식 (3)으로 계산한다.

$$corr_{ij} = \frac{P(C_i \cup C_j)}{P(C_i)P(C_j)} \tag{3}$$

P(C_i)와 P(C_j)는 각각 클러스터 C_i와 C_j의 확률이고, P($C_i \cup C_j$)는 두 클러스터의 확률이다. 식 (3)에 의한 결과 값이 1보다 작으면 C_i와 C_j가 부정적으로 관련되어 있다는 것을 나타내고, 1보다 크면 긍정적으로 관련되어 있다는 것을 나타낸다. 그리고 결과 값이 1이면 C_i와 C_j는 서로 독립적이고 둘 사이에 아무런 상관관계가 없음을 의미한다. $P(C) = \frac{support_count(C)}{|T|}$이므로 식 (3)은 식 (4)로 계산된다. 유사도 계산에 사용된 |T|는 트랜잭션의 수이고, support_count(C)는 클러스터 C에 포함된 아이템의 빈도수를 의미한다.

$$corr_{ij} = \frac{|T| * support\_count(C_i \cup C_j)}{support\_count(C_i) * support\_count(C_j)} \tag{4}$$

신뢰도 유사도는 식 (5)에 의해 계산된다. 클러스터 C_j는 하나의 원소만을 가지고 있다.

$$confidence(C_i \Rightarrow C_j) = P(C_i|C_j) = \frac{support\_count(C_i \cup C_j)}{support\_count(C_i)} \tag{5}$$

마지막으로 간선 가중치 유사도는 식 (6)에 의해 계산된다.

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

$|E_{ij}|$는 각각의 간선이 C_i의 정점과 C_j의 정점을 연결하는 간선 집합 $|E_{ij}|$의 카디널리티(Cardinality)이고, w_{ij}는 $|E_{ij}|$에 포함된 간선의 지지도 값이다. 간선가중치 계산식은 C_i와 C_j의 정점을 연결하는 간선들의 평균 가중치 값을 구한다. 세 가지 유사도 방법 중 하나를 선택하여 계산된 값 중에서 인접 정점 집합 A에서 가장 유사도가 큰 정점을 선택하여 클러스터 C_j에 정점으로 바꾼다. 여기에서 주목해야 할 점은 가장 유사도가 큰 정점은 클러

스터 C_i의 원소인 각각의 정점 vi에 관계를 갖고, E_{ij}는 클러스터 C_i와 C_j 사이의 간선들의 집합을 나타내는 점이다.

세 번째 단계는 두 클러스터를 하나의 클러스터 C_k로 병합한다. 병합을 하기에 앞서 클러스터 C_k에 지지도 값 support_count(C_k)를 할당한다. 여기에서 클러스터 C_k의 지지도 값은 경로지지도 계산식으로 계산된 값을 의미한다. 그러므로 클러스터 C_k에 모든 정점 V_k는 연결된 경로로 되어 있다. 병합된 클러스터는 Frequent Closed Itemset이다 (Pasquier et. al, 1999).

네 번째 단계에서는 반복 조건을 검사해 단계 2를 반복할지 결정한다. 클러스터 C_k의 인접 정점 집합 A가 원소를 가지고 있으면, 두 번째 단계로 돌아가 탐욕적(Greedy) 방법에 의해 A의 원소 중 가장 유사한 정점을 선택해 알고리즘을 다시 수행한다. A가 원소를 가지고 있지 않거나, 클러스터 C_k가 연결된 경로가 아니면, 알고리즘은 클러스터 리스트 C에 클러스터 C_k를 첨가한다. 이때 클러스터 C_k의 카디널리티(cardinality)가 2보다 작으면 클러스터는 없어지고, 없어진 클러스터의 정점은 정점 집합 V에 남게 된다.

4단계까지는 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 여섯 번째 단계로 넘어가고, 남아 있다면 첫 번째 단계로 돌아가 알고리즘을 계속 수행한다.

마지막 여섯 번째 단계는 클러스터 리스트 C에 포함된 어떤 클러스터에도 포함되지 않은 정점을 처리하는 단계이다. 하나의 정점을 가진 클러스터는 인정하지 않고 다시 정점 집합 V로 반환하기 때문에, 클러스터 리스트 C에 포함된 각각의 클러스터는 카디널리티가 2 이상이다. C에 포함된 어떤 클러스터에도 속하지 않은 정점을 남겨진 정점들의 집합 R에 포함시키고 다음과 같이 표기한다.

$$R = \{v | v \notin C_j,\ C_j \in C\}$$

R에 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 항목집합이 되기 어렵다. 둘째, 이상점들은 다른 정점과의 상이성이 커서 어느 클러스터에도 포함되지 않기 때문에 의미있고 유용한 규칙에 포함되기에는 부족한 점이 있다. 이상점을 제외함으로 인해 의미가 적은 연관규칙은 생성되지 않고, 이상점들을 처리하는데 사용되는 시간과 메모리 같은 자원을 소모하지 않을 수 있게 된다. 클러스터에 포함될 수 있는 아이템이지만 알고리즘에 의해 이상점에 속하는 아이템들에 대한 오류율은 실험을 통해 알아본다.

4.2 연관규칙의 발견

연관규칙은 생성된 클러스터를 이용해 발견한다. 빈발 항목집합을 생성하지 않고 클러스터의 아이템을 이용하기 때문에 k-아이템 집합에 대한 규칙의 생성을 단계적으로 진행해야한다. 예를 들어, 임의의 클러스터 C_i가 4개의 아이템으로 구성되어 있다면 2-아이템 집합부터 4-아이템 집합에 해당하는 연관규칙을 생성한다. <표 1>은 4개의 아이템으로 구성된 클러스터를 이용하여 생성되는 아이템 집합을 표현한 것이다. 클러스터에 포함되는 아이템은 최소지지도를 만족하기 때문에 각각의 아이템 집합은 빈발 항목집합에 해당되므로 k-아

이템집합에 해당하는 항목들에 대하여 신뢰도를 이용하여 연관규칙을 생성하면 된다. 빈발 패턴 네트워크에서 정점이 각 아이템의 빈도수를 표현하고, 간선이 두 아이템의 빈도수를 표현하고, 경로가 클러스터의 모든 아이템에 대한 빈도수를 표현하기 때문에 신뢰도를 간단하게 구할 수 있다.

<표 1> 클러스터의 아이템 집합

클러스터 아이템	ABCD
2-아이템집합	AB, AC, AD, BC, BD, CD
3-아이템집합	ABC, ABD, ACD, BCD
4-아이템집합	ABCD

이러한 방법은 단계적인 연관규칙 생성과정을 요구하지만 빈발 항목집합을 생성하는 것보다 적은 비용이 든다. 이것은 수행속도 비교 실험에서 확인할 수 있다.

5. 실험 및 결과

5.1 실험 환경 및 데이터 집합

본 논문의 실험은 Microsoft Visual Studio 환경에서 C++프로그래밍 언어를 이용하여 이루어졌으며, 빈발 패턴 네트워크에서 아이템을 클러스터링하기 위해 두 가지 인공 데이터 집합(Synthetic Datasets)과 소매 시장바구니 데이터 집합(Retail Market Basket Datasets)인 실제 데이터를 가지고 실험을 하였다. 인공 데이터 집합은 IBM Almaden Research Group에서 만든 생성프로그램을 사용하여 두 가지 인공 데이터 집합을 생성하였다. 첫 번째 인공 데이터는 10,000개의 아이템, 트랜잭션당 평균 10개의 아이템, 빈발 항목집합 당 평균 4개의 아이템, 그리고 100,000개의 트랜잭션을 포함하는 데이터 집합이다(T10.I4.100K with 10K items). 두 번째 인공 데이터는 10,000개의 아이템, 트랜잭션당 평균 40개의 아이템, 빈발 항목집합 당 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 시장바구니(Market basket) 데이터 집합이다. 이 소매 데이터는 5,133명의 고객으로부터 생성된 88,162개의 트랜잭션을 포함하고 있다. 평균적으로 하나의 트랜잭션에 13개의 아이템이 포함되어 있지만, 대부분 고객들은 쇼핑을 할 때 7개에서 11개 사이의 아이템을 구입한다. 이 실제 데이터는 빈발 항목집합 마이닝 데이터집합(FIMI) 저장소에서 공개하고 있는 데이터이다. 데이터에 대한 세부적인 정보는 FIMI[1]에서 얻을 수 있다.

5.2 실험 평가 방법

실험은 두 가지로 구성된다. 첫 번째는 FPNC 알고리즘을 이용하여 빈발 패턴 네트워크에 존재하는 아이템 사이의 유사도를 측정하여 클러스터를 생성한다. 두 번째는 생성된 클러스터의 아이템을 이용하여 연관규칙을 생성하고, Apriori와 FP Growth 알고리즘의 수행 속도와 생성된 규칙의 수를 비교한다.

빈발 패턴 네트워크에서 FPNC 알고리즘으로 생성된 클러스터의 정확성을 평가하기 위해 비용함수(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 합과 같은 표준 비용함수를 사용하지 않고, 오

1) http://fimi.cs.helsinki.fi/data.

류기반의 비용함수를 사용한다. FPNC 알고리즘에 의해 생성된 클러스터를 평가하기 위해 정규화 오류율(Normalized Error Rate)을 사용하였고, NER이라 표기한다(Davy et. al, 2006).

클러스터 C_i의 오류율(Error Rate, ER)은 다음과 같이 계산된다.

$$ER(C_i) = \frac{|E|}{|C_i|} \quad (7)$$

|E|는 오류 집합(Error Set) E의 카디널리티이고, 오류 집합은 어떤 클러스터에 대하여 내부유사도 (Intra-similarity)보다 상호유사도(Inter-similarity)가 더 큰 정점들의 집합이다.

$|C_i|$는 클러스터 내부에 포함된 정점의 수를 표현한다. 다른 클러스터에 포함될 경우 더 큰 유사도를 가질 수 있음에도 불구하고 탐욕적 클러스터링 방법을 사용하기 때문에 낮은 유사도를 가지고 클러스터에 포함된 아이템이 오류집합에 포함된다.

정규화 오류율은 식 (8)에 의해 계산된다.

$$NER(C) = \frac{\sum ER(C_i)}{|C|} \quad (8)$$

|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.

FPNC를 통해 생성된 클러스터로부터 단계적인 연관규칙을 생성한다. 전통적인 연관규칙 탐사 알고리즘인 Apriori와 FP-Tree 알고리즘을 통해 생성된 규칙의 수와 수행속도를 비교한다. Apriori 알고리즘은 빈발 항목집합 생성에서 연관규칙 생성까지의 시간을 비교 대상으로 하고, FP-Growth 알고리즘은 FP-Tree 생성에서 연관규칙 생성까지의 시간을 비교 대상으로 한다. 빈발 패턴 네트워크는 아이템 클러스터링부터 클러스터로부터 연관규칙 생성까지의 시간을 비교 대상으로 한다.

5.3 실험 결과 및 평가

5.3.1 클러스터

두 인공 데이터는 100,000개의 트랜잭션으로 이루어져 있다. (T10I4D100K)는 총 870개의 아이템을 가지고 있고, (T40I10D100K)는 총 942개의 아이템을 가지고 있다. <표 2>는 상관관계 유사도를 사용하여 FPNC 알고리즘을 수행한 후, 정규화 오류율로 클러스터를 평가한 내용이다. 표는 0.1%에서 0.5%까지 각 단계별로 최소지지도 값을 조절하

<표 2> 상관관계 유사도에 따른 오류율

Data set		T10I4D100K					T40I10D100K			
Support (%)	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
|C|	210	172	128	95	68	165	173	165	157	155
|Ic|	688	567	412	270	173	865	840	787	727	680
|E|	75	38	21	10	10	26	39	42	54	71
AVG.|E|	0.36	0.22	0.16	0.11	0.15	1.58	1.68	0.8	0.63	0.77
NER	0.090	0.056	0.046	0.037	0.055	0.212	0.239	0.128	0.130	0.157

여 실험하였다. |C|는 생성된 클러스터의 수, $|L_c|$는 모든 클러스터에 포함된 아이템의 수, |E|는 에러 집합 E에 포함된 아이템의 수, AVG. |E|는 평균 에러 집합의 수이고 NER은 정규화 오류율로 계산된 값을 보여준다. <표 3>은 0.7%의 신뢰도를, <표 4>는 간선 가중치의 유사도를 가지고 실험한 결과이다. 이 실험의 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. 인공 데이터 집합의 실험결과로부터 신뢰도 유사도가 클러스터의 정확성에 많은 영향을 주는 것을 알 수 있다. 하지만 클러스터에 포함되는 아이템의 수가 적다. 따라서 연관규칙 발견에는 신뢰도 유사도보다는 정규화 오류율이 조금 높지만 더 많은 클러스터를 포함하고 있는 간선 가중치 유사도를 이용해 생성된 클러스터를 사용한다.

<표 3> 신뢰도 유사도에 따른 오류율

Data set	T10I4D100K					T40I10D100K				
Support(%)	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
|C|	97	66	40	19	7	72	72	62	44	33
|Ic|	359	244	146	68	26	544	455	373	281	205
|E|	11	2	3	0	0	10	10	8	6	2
AVG.|E|	0.11	0.03	0.08	0.0	0.0	0.14	0.14	0.13	0.17	0.06
NER	0.027	0.008	0.019	0.0	0.0	0.036	0.022	0.013	0.012	0.005

<표 4> 간선 가중치에 따른 오류율

Data set	T10I4D100K					T40I10D100K				
Support(%)	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
|C|	218	184	138	91	72	155	159	163	157	145
|Ic|	656	550	409	263	174	829	785	747	706	649
|E|	120	44	32	14	7	47	53	79	105	147
AVG.|E|	0.55	0.24	0.23	0.15	0.10	3.06	2.03	1.83	1.30	1.01
NER	0.057	0.052	0.054	0.044	0.037	0.092	0.094	0.124	0.089	0.084

<표 5> 각 유사도에 따른 오류율

Data set	Confidence					Edge Weight				
Support(%)	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
|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

상관관계 유사도는 많은 아이템을 클러스터에 포함하고 있지만 정규화 오류율이 높아 실제 시장 바구니 데이터 집합에는 신뢰도와 간선 가중치 유사도 두 가지를 적용하여 실험을 하였다. 데이터 집합은 5,133명의 고객으로부터 생성되었고, 총 16470개의 아이템과 88,162개의 트랜잭션으로 구성되어 있다. 트랜잭션은 평균 13개의 아이템을 가지고 있다.

지지도 값을 변경함으로 인해 생성되는 클러스터의 수를 조절할 수 있다. 더 큰 값의 지지도를 설정하면 더 적은 수의 클러스터가 생성 된다. 클러스터링 방법을 적용함으로써 더 응축된 연관규칙을 가질 수 있지만 더 적은 클러스터를 생성함으로 인해 이상점에 속한 아이템은 더 늘어난다. 하지만 지지도 값을 낮게 적용하거나 트랜잭션이 추가되면, 아이템에 해당하는 정점의 발생 빈도가 주어진 최소지지도를 만족할 수 있고, 이상점에 속하던 아이템이 클러스터에 속할 수 있기 때문에 생성된 이상점 집합은 제거하지 않는다.

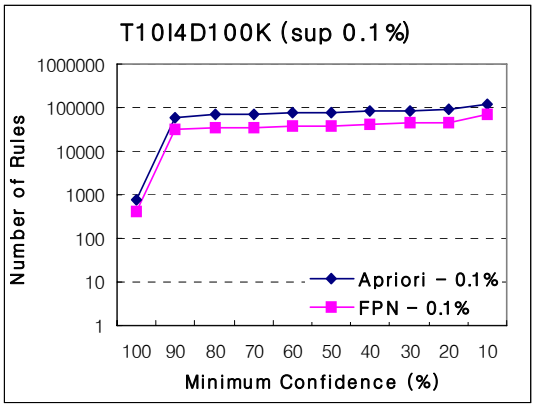
5.3.2 연관 규칙

FPNC 알고리즘으로 생성한 클러스터로부터 결론부에 아이템이 하나인 연관규칙을 생성하였고, 이를 Apriori, FP-Growth 알고리즘을 통해 생성된 아이템의 수와 비교하였다. [그림 6]과 [그림 7]은 생성된 연관규칙의 수를 나타내며 클러스터를 이용한 연관규칙 생성이 더 적은 수의 연관규칙을 생성함을 알 수 있다.

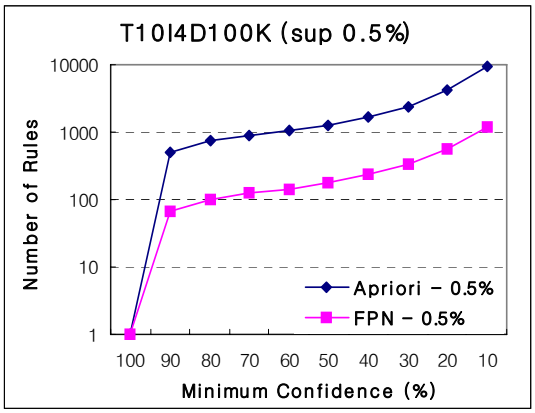
Apriori 알고리즘과 FP-Growth 알고리즘은 생성되는 연관규칙의 수가 같기 때문에 그림에서는 Apriori 알고리즘과 FPN의 수만을 표시하였다.

[그림 8]과 [그림 9]는 각 알고리즘의 연관규칙 마이닝 수행 시간을 비교한 것이다.

실험은 최소신뢰도 70%에서 최소지지도의 변화에 따른 알고리즘 수행 속도 비교이다.



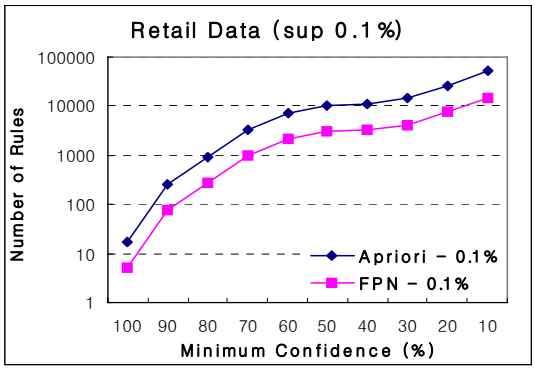
(a)



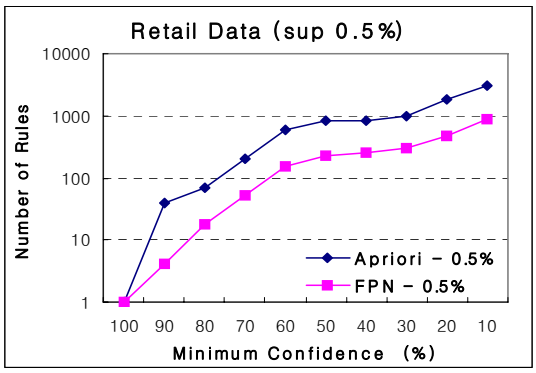
(b)

[그림 6] 연관규칙의 수 : T10I4D100K

Apriori 알고리즘은 최소지지도가 변하면 최소지지도에 맞는 후보 항목집합과 빈발 항목집합을 생성한다. 즉 최소지지도가 0.1%일 경우와 0.2%일 경우의 빈발 항목집합이 다르다는 것이다. 따라서 빈발 항목집합 발견에 드는 비용이 지속적으로 들어간다. FP-Growth 알고리즘은 후보 항목집합을 생성하지 않지만 최소지지도에 따라 FP-Tree가 새롭게 생성되어야 한다. Apriori와 마찬가지로 최소지지도가 0.1%일 경우와 0.2%일 경우의 FP-Tree는 다르다.

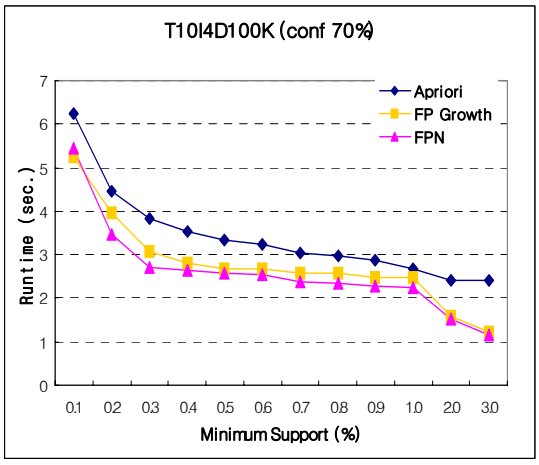


(a)



(b)

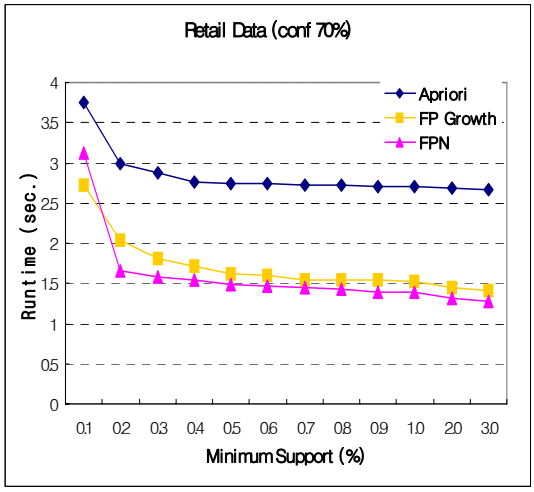
[그림 7] 연관규칙의 수 : Retail Data



[그림 8] 세 알고리즘의 수행속도 비교(T10I4D100K)

빈발 패턴 네트워크를 이용한 연관규칙의 발견은 네트워크 구축 시간으로 인해 초기 연관규칙의 생성 속도가 느리다. 하지만 Aprioir 알고리즘보다 빠른 것을 볼 수 있고, 최소지지도 값이 변해도 클러스터와 연관규칙 생성만 수행하기 때문에 빈발 항목집합 생성에 드는 비용을 줄일 수 있다. 그 결과로 [그림 8]과 [그림 9]와 같이 전체 연관규칙 마이닝 수행 시간이 감소된 것을 볼 수 있다.

실험의 결과로부터 빈발 패턴 네트워크 구축에 드는 비용이 들어가지만 빈발 항목집합을 생성하지 않고 연관규칙을 생성하기 때문에 전체 연관규칙 마이닝 수행속도가 향상된 것을 볼 수 있다. 그리고 이후의 최소지지도 변화에는 유연함을 보임을 파악할 수 있다.



[그림 9] 세 알고리즘의 수행속도 비교(Retail)

6. 결론 및 향후 연구

대량의 데이터에 담긴 유용한 정보를 발견하기 위해 최소지지도를 만족하는 빈발 항목집합을 발견하고, 이러한 빈발 항목집합으로부터 최소신뢰

도를 만족하는 연관규칙을 생성하는 것은 데이터 마이닝 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 발견을 효율적으로 할 수 있는 빈발 패턴 네트워크를 제안하였다. 빈발 패턴 네트워크는 데이터베이스에 존재하는 트랜잭션을 한 번의 데이터베이스 접근으로 네트워크에 옮김으로써 반복적인 데이터베이스 접근 비용을 줄일 수 있는 구조이다. 또한 FP Tree처럼 간결하고 압축된 정보를 표현한다. 그리고 경로지지도를 통해 k-아이템집합(k>2)의 지지도를 계산하는 방법을 소개하였고, 네트워크를 이용하여 연관있는 아이템을 클러스터링한 후, 클러스터로부터 생성된 연관규칙을 생성하였다.

생성 빈도수가 낮고 연관성이 적은 아이템을 연관규칙 마이닝 작업에서 제외시킴으로써 흥미롭지 않은 규칙의 생성을 미연에 방지하였다. FPNC 알고리즘으로 생성된 클러스터는 연관규칙 마이닝을 위한 탐색 공간을 국한시켜 연관규칙 발견시간을 단축시킨다.

FPNC 알고리즘을 통해 클러스터 집합과 포함된 아이템 그리고 에러집합에 포함된 아이템의 수를 알아보고 빈발 패턴 네트워크에서 생성되는 클러스터의 정확성을 비교하였다. 또한 클러스터로부터 생성된 연관규칙의 수와 수행속도를 비교하였다.

실험의 결과로부터 빈발 패턴 네트워크에서는 신뢰도 유사도 방법을 이용하여 아이템을 클러스터링하는 것이 높은 정확성을 가지는 것을 알 수 있었다. 그리고 간선 가중치 유사도는 신뢰도보다 높은 오류율을 가지지만, 더 많은 아이템을 클러스터에 포함시켜 연관규칙 마이닝에 더 적합함을 알게 되었다.

또한 실험의 결과를 통해 빈발 패턴 네트워크 구조를 통한 연관규칙 발견이 빈발 항목집합을 생성하지 않기 때문에 최소지지도에 유연성을 가짐을 알 수 있었다.

향후 연구로는 클러스터링 알고리즘을 개선시켜 좀 더 많은 아이템이 클러스터에 포함될 수 있도록 하는 연구가 필요하고, (Liu et. al, 1999)에서와 같이 드문 아이템(rare item)이 포함되어 있는 데이터 집합에 대한 연관규칙 발견이 수행될 경우에 빈발 패턴 네트워크를 이용해 보고자 한다.

참고문헌

[1] Agrawal, R. C. and C. Aggarwal, V.V.V., “A Tree Projection Algorithm For Generation of Frequent Itemsets”, Journal of Parallel and Distributed Computing, Vol.61, No.3 (2000), 350~371.

[2] Agrawal, R., and R. Srikant, “Fast Algorithms for Mining Association Rules”, In Proceedings of the 20th VLDB(Very Large Data Bases) Conference, (1994), 580~592.

[3] Agrawal, R., T. Imilienski, and A. Swami, “Mining association rules between sets of items in large datasets”, In Proceedings of the 1993 ACM SIGMOD International Conference, (1993), 207~216.

[4] Christian, B. and K. Rudolf, “Induction of Association Rules : Apriori Implementation”, In 15th Conference on Computational Statistics (Compstat 2002, Berlin, Germany) Physica Verlag, Heidelberg, Germany, 2002.

[5] Han, E. H., G. Karypis, and V. Kumar, “Scalable parallel data mining for association rules”, In Proceedings of 1997 ACM SIGMOD International Conference on

Management of Data, (1997), 337~352.

[6] Han, E-H., G. Karypis, V. Kumar, and B. Mobasher, "Clustering Based On Association Rule Hypergraphs", In Proceedings of SIGMOD '97 Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD '97), 1997.

[7] Han, J. and M. Kamber, "Data Mining : Concepts and Techniques", Morgan Kaufmann Publishers, 2005.

[8] Han, J., J. Pei, Y. Yin, and R. Mao, "Mining Frequent Patterns without Candidate Generation : A Frequent-Pattern Tree Approach", Journal of Data Mining and Knowledge Discovery, Vol.8(2004), 53~87.

[9] Lakshmanan, L. V. S., C. K-S. Leung, and R. T. Ng, "Segment Support Map : Scalable Mining of Frequent Itemsets", ACM SIGKDD Explorations Newsletter, Vol.2, No.2(2000), 21~27.

[10] Lent, B., A. Swami, and J. Widom, "Clustering Association Rules", In Proceedings of the 13th International Conference on Data Engineering, (1997), 220~231.

[11] Liu, B., W. Hsu, and Y. Ma, "Mining Association Rules with Multiple Minimum Supports", In ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-99), (1999), 125~134.

[12] Liu, B., W. Hsu, and Y. Ma, "Pruning and summarizing the discovered associations", In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999.

[13] Mannila, H., H. Toivonen, and I. Verkamo, "Efficient algorithms for discovering association rules", International Conference Knowledge Discovery and Data Mining (KDD), (1994), 181~192.

[14] Michael, J. A., B. and L. Gorden, "Data Mining Techniques : For Marketing, Sales, and Customer Support", John Wiley and Sons, 1997.

[15] Ng, R. and J. Han, "Efficient and effective clustering method for spatial data mining", In Proceedings of the 20th VLDB Conference, (1994), 144~155.

[16] Park, J. S., M. S. Chen, and P. S. Yu, "An effective hash based algorithm for mining association rules", In Proceedings of the ACM SIGMOD International Conference on Management of Data, (1995), 175~186.

[17] Pasquier, N., Y. Bastide, R. Taouil, and L. Lakhal, "Discovering Frequent Closed Itemsets for Association Rules", In Proceedings of the 7th International Conference on Database Theory, (1999), 398~416.

[18] Srikant, R., Q. Vu, and R. Agrawal, "Mining Association Rules with Item Constraints", In Proceedings 3rd International Conference Knowledge Discovery and Data Mining (KDD), (1997), 67~73.

[19] Tan, P-N., M. Steinbach, and V. Kumar, "Introduction to Data Mining", Addison-Wesley, 2006.

[20] Zaki, M. J., "Generating Non-Redundant Association Rules", Conference on Knowledge Discovery in Data, Proceedings of the sixth ACM SIGKDD, (2000), 34~43.

[21] Jay, J., Jiang, David, and W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy", In Proceedings of International Conference on Research on Computational Linguistics, (1997), 19~33.

[22] Davy, J, Tom, B., Koen, V., and Geert, W., "Evaluating the performance of cost-based discretization versus entropy-and error-based discretization", Computers and Operations Research, Vol.33, No.11(2006), 3107~3123.

Abstract

Discovering Association Rules using Item Clustering on Frequent Pattern Network

Kyeong-Jin Oh* · Jin-Guk, Jung* · Inay Ha* · Geun-Sik Jo*

Data mining is defined as the process of discovering meaningful and useful pattern in large volumes of data. In particular, finding associations rules between items in a database of customer transactions has become an important thing. Some data structures and algorithms had been proposed for storing meaningful information compressed from an original database to find frequent itemsets since Apriori algorithm. Though existing method find all association rules, we must have a lot of process to analyze association rules because there are too many rules. In this paper, we propose a new data structure, called a Frequent Pattern Network (FPN), which represents items as vertices and 2-itemsets as edges of the network. In order to utilize FPN, We constitute FPN using item's frequency. And then we use a clustering method to group the vertices on the network into clusters so that the intracluster similarity is maximized and the intercluster similarity is minimized. We generate association rules based on clusters.

Our experiments showed accuracy of clustering items on the network using confidence, correlation and edge weight similarity methods. And We generated association rules using clusters and compare traditional and our method. From the results, the confidence similarity had a strong influence than others on the frequent pattern network. And FPN had a flexibility to minimum support value.

Key Words : Frequent Pattern Network, Cluster, Association Rules

* College of Computer Software and Media Technology, Sangmyumg University