데이터 마이닝의 전처리를 위한 K-means 알고리즘을 이용한 빈발패턴 생성

유희종*, 박지연*
*관동대학교 컴퓨터학과
e-mail:{hjyoo, cypark}@kd.ac.kr

Creation of Frequent Patterns using K-means Algorithm for Data Mining Preprocess

Heui-Jong Yoo*, Chi-Yeon Park*
*Dept of Computer Science, Kwandong University

요 약

우리가 사용하는 데이터베이스 내에는 많은 양의 데이터 들이 들어 있으며, 계속적으로 그 양은 늘어나고 있다. 이러한 데이터들로부터 질의를 통해 얻을 수 있는 기본적이고 단순한 정보들과 달리 고급 정보를 얻게 해주는 방법이 데이터 마이닝이다. 데이터 마이닝의 기법 중에서 본 논문에서는 k-means 알고리즘을 사용하여 트랜잭션을 클러스터링 함으로써 데이터베이스의 트랜잭션 수를 줄여 연관규칙의 대표적인 알고리즘인 Apriori 알고리즘의 단점인 트랜잭션 스캔으로 인한 성능 저하를 개선하고자 한다.

1. 서론

우리가 사용하는 데이터베이스 내에는 많은 양의 데이터 들이 들어 있으며, 사용할수록 그 양은 늘어나고 있다. 이렇게 많은 양의 데이터에서 질의(Query)나 검색(Search)을 통해 얻을 수 있는 정보는 기본적이고 일반적인 정보들이다. 이런 정보들보다 더 나은 고급정보를 얻기 위해 사용하는 기법이 데이터 마이닝(Data Mining)이다. 데이터 마이닝이란 대용량으로 저장되어 있는 데이터로부터 의미 있는 정보를 찾아내는 과정이다.[1]

데이터 마이닝 기술에는 특성화(characterization), 연관규칙(Assocation Rules), 경향분석(trend analysis), 분류화(Classification), 일반화(Generalization), 클러스터링(Clustering) 등 여러 가지 다양한 방법이 있다[2].

연관규칙 기법은 데이터 마이닝 기법 중에서도 가장 많이 연구되고 여러 분야에 적용되는 기법으로, 연관규칙을 생성할 때 사용하는 알고리즘으로는 잘 알려진 Apriori 알고리즘이 있다[3]. Apriori 알고리즘은 후보 항목 생성 시 모든 데이터베이스에서의 데이터 항목에 대한 생성이 아닌, 전 단계의 빈발 항목집합을 대상으로 후보 항목집합을 구성한다. 그러나 이 알고리즘은 데이터베이스의 모든 트랜잭션을 반복적으로 검사하기 때문에 트랜잭션이 많을수록 성능이 저하되는 단점이 있다. 본 논문에서는 K-means 알고리즘을 사용해 트랜잭션을 클러스터링 하여 불필요한 트랜잭션 접근 수를 줄임으로서 Apriori 알고리즘의 최대 단점을 극복하려고 한다.

클러스터링이란 주어진 객체들 중에서 유사한 객체들을 몇 개의 집합으로 그룹화 하여 그 그룹의 특징이나 성격을 파악하는 방법이다[4]. 클러스터링의 종류는 계층적 클러스터링(Hierarchical Clustering), 분할 클러스터링(Partitional Clustering), 최근접 이웃 클러스터링(Nearest Neighbor Clustering) 등이 있다[5].

본 논문의 순서는 다음과 같다. 2장에서는 기존의 클러스터링 방법과 연관규칙에 대하여 간략히 살펴보고, 3장에서는 K-means 알고리즘을 사용하여 클러스터링 하는 방법과 이를 이용하여 연관규칙을 생성하는 방법에 대하여 기술한다. 4장에서는 실험예제를 통해 결과를 확인하고, 마지막으로 5장에서는 결론 및 향후 연구 방향에 대해 기술한다.

2. 관련연구

2.1 클러스터링

클러스터링의 기본 목적은 속성(또는 변수)들의 자연스러운 클러스터(cluster)를 찾아내는 것이다. 자연스러운 클러스터를 찾아내기 위해서는 객체들 간의 연관성(유사도 또는 상이도)을 측정할 수 있도록 양적인 척도 또는 질적인 척도가 필요하며, 분류와는 달리 클러스터의 수나 그 구조에 대해 아무런 가정을 하지 않는다.

클러스터링 기법으로는 클러스터의 초기 중심값 선정과 클러스터간의 유사도를 측정하는 기법 그리고 클러스터를 구축하는 기법에 따라서 계층적 기법과 비계층적 기법이 있다.

계층적 클러스터링은 문서간의 유사도 정보를 토대로 단계적으로 계층적인 클러스터를 형성하는 방법으로 각

문서들을 제각각 하나의 클러스터로 시작하여, 유사도가 높은 두 개의 클러스터를 하나의 클러스터로 만드는 과정을 반복하여 하나의 클러스터가 남을 때까지 반복하는 과정을 갖는다.

비계층적 클러스터링은 임의의 선택된 초기 클러스터로부터 문서를 클러스터에 재배치하는 작업을 반복적으로 수행하여 최종 클러스터를 형성하는 방법으로, 일반적으로 미리 몇 개의 클러스터로 나누어질 것이라고 예상하고 클러스터의 개수 K를 표현해야 한다.

클러스터링 기법 중 K-means 알고리즘은 수치 데이터에 대한 분할적 클러스터링 방법으로 가장 많이 사용되는 기법 중 하나이다. 이 알고리즘은 각 객체를 유사한 특성을 갖는 k개의 클러스터로 분할하는 방법으로, 각 클러스터에 속하는 객체들의 평균을 중심으로 잡고 근접한 거리에 있는 객체들을 묶어서 분할한다.[5].

2.2 연관규칙

연관규칙은 데이터베이스 내의 단위 트랜잭션에서 빈번하게 발생하는 사건의 유형을 발견하는 것이다[3]. 예를 들어, “전체 고객 중에 빵과 버터, 그리고 우유를 구매한 고객이 10% 이상이고, ‘빵과 버터’를 구매한 고객의 50%가 우유도 함께 구매한다.” 이것이 하나의 발견된 사건의 유형, 즉 하나의 규칙이 된다. 여기서 10%는 연관규칙의 지지도(Support)가 되고, 50%는 신뢰도(Confidence)가 된다.[6]

지지도란 생성된 연관규칙이 전체 항목에서 차지하는 비율을 말한다. 즉, 데이터베이스에 속한 전체 트랜잭션 개수 중 그 연관규칙을 지지하는 트랜잭션의 비율을 의미하며, 전체 사건 또는 거래 중에서 어떤 아이템 X와 Y를 포함하는 거래의 정도를 나타낸다. 이것을 식으로 나타내면 다음과 같다.

$$Support = \frac{|X \cap Y|}{N} \text{(N은 전체 트랜잭션의 개수)}$$

신뢰도는 연관규칙의 강도를 의미하며, 전제부를 만족하는 트랜잭션이 결론부까지를 만족하는 비율을 말한다. 즉 어떤 아이템 X를 포함하는 거래 중에서 어떤 아이템 Y가 포함된 거래의 정도를 의미한다. 이것을 식으로 나타내면 다음과 같다.

$$Confidence = \frac{|X \cap Y|}{|X|}$$

지지도를 통해 나온 빈발항목에서 신뢰도를 통해 최종 연관규칙을 얻어내는 것이다

기존에 연구되었던 방법들 중에는 후보 항목집합의 크기를 줄이기 위해 해시기반을 사용한 기법[7]과 주어진 데이터베이스에서 임의의 샘플을 취하여 빈발항목 집합을 탐색하는 방법[8], 후보집합 생성이 없는 빈발항목 집합 마이닝 방법[9] 등이 있다.

대표적인 연관규칙 알고리즘인 Apriori 알고리즘은 주어진 크기의 모든 후보들을 생성하는 후보생성 단계와 생성된 후보들의 지지도를 계산하는 단계로 구성되어 있으며, 각 카운팅 단계에서는 전체 데이터베이스를 스캔한다. 이 과정에서 데이터베이스의 전체 트랜잭션을 검색해야하기 때문에 그만큼 수행속도가 느려지게 된다. 또한, 후보 항목이 많을수록 알고리즘의 성능은 떨어지게 된다.

본 논문에서는 이러한 문제점을 해결하기 위해 데이터베이스에 있는 트랜잭션들을 먼저 K-means 알고리즘으로 클러스터링 하고 클러스터링을 통해 나온 클러스터에서 연관규칙을 찾는다. 이 방법은 후보항목의 개수도 적어지게 되고 전체 트랜잭션보다 클러스터 내에 있는 더 적은 수의 트랜잭션을 검색하게 되기 때문에 Apriori 알고리즘보다 향상 될 것이다.

3. K-means 알고리즘을 이용한 빈발패턴 생성

3.1 K-means 알고리즘

비계층적 군집 방법(Non-Hierarchical Clustering Method)인 K-means 알고리즘은 각 개체를 가장 가까운 중심점에 할당하는 방법이다.

K-means 알고리즘은 객체간의 거리를 유클리디안 거리로 정의한 후 클러스터의 평균을 계산하여 기준함수(criterion function)를 최소화하도록 클러스터를 구성해 나가는 알고리즘이다.

K-means 알고리즘은 (1) k개의 클러스터를 구성하기 위하여 k개의 초기 클러스터 지정, (2) 객체와 클러스터 중심과의 유사도 계산, (3) 해당 객체와 가장 가까운 클러스터에 할당, (4) 새로이 할당된 객체들을 사용하여 해당 클러스터의 중심을 갱신하는 과정을 기준함수 값이 특정 이내로 될 때 까지 (2)~(4) 과정을 반복한다.

기준 함수는 객체들의 오차 제곱의 합으로 식은 다음과 같다.

$$Cost = \sum_{i=1}^{k} \sum_{j=1}^{C_i} |x_{ij} - c_i|^2$$

$$c_i = \frac{1}{C_i} \sum_{j=1}^{C_i} x_{ij}, \; i = 1,2,...,k$$

이 식에서 k는 클러스터 수, C_i는 i번째 클러스터의 객체들의 수, x_{ij}는 i번째 클러스터의 j번째 객체, c_i는 i번째 클러스터이다.

K-means 알고리즘은 거리 행렬을 필요로 하지 않으므로 규모가 큰 데이터베이스에 유리하고, 클러스터내의 중심 계산을 평균으로 하기 때문에 클러스터의 평균이 정의되어 질 수 있는 수치 데이터에만 사용할 수 있다는 제약이 있다.

3.2 클러스터를 이용한 연관규칙

연관규칙 알고리즘의 가장 전형적인 방법인 Apriori 알고리즘은 데이터베이스에서 후보 항목 집합(candidate

itemset)을 구성하고, 구성된 후보 항목집합에서 빈발 항목집합(large itemset)을 탐사하는 과정으로 구성된다. Apriori 알고리즘은 후보 항목 생성 시 모든 데이터베이스에서의 데이터 항목에 대한 생성이 아닌, 전 단계의 빈발 항목집합을 대상으로 후보 항목집합을 구성한다.

빈발 항목집합은 사용자가 정한 최소지지도(minimum support)에 대하여 데이터 항목 집합 X의 지지도 Support(X)와 최소지지도와의 관계가 다음식을 만족해야 하며, 만족할 경우 "빈발하다" 라고 정의한다.

$$\text{Support}(X) \geq \text{minimum Support}(X)$$

빈발 항목은 빈발 항목집합의 원소에 포함되는 항목을 의미한다. k-빈발항목집합은 k개의 항목으로 구성된 빈발 항목집합 L_k로 표현한다.

후보 항목집합은 빈발 항목집합의 원소가 될 가능성이 있는 항목들로 구성된 집합으로 빈발 항목 집합을 탐사하기 위해 사용되는 집합니다. k-후보항목집합은 k개의 항목으로 구성된 후보 항목을 말하며 C_k로 표현한다.

k-항목집합은 항목 집합의 원소가 k개의 항목으로 구성된 집합이다.

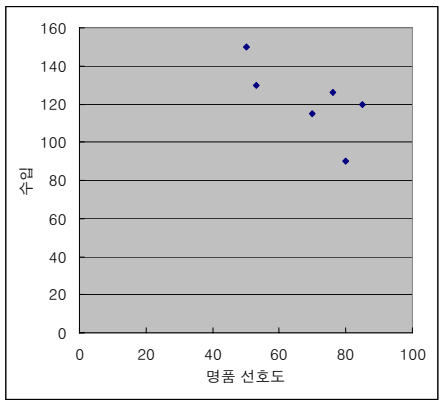
Apriori 알고리즘은 k-빈발항목집합 L_k를 구하기 위해서 (k-1)-빈발항목집합 L_{k-1}로부터 k-후보항목집합 C_k를 구하고 C_k의 지지도를 계산하여 최소 지지도 이상을 만족하는 L_k를 구하는 과정을 반복한다. Apriori 알고리즘의 각 단계의 진행은 데이터 항목의 증가에 따라 반복적으로 진행된다. Apriori 알고리즘은 더 이상의 후보 항목집합을 생성할 수 없을 때까지 반복되어 빈발 항목들을 탐사한다.

4. 실험 및 결과

월수입에 따른 명품 선호도에 대한 데이터가 <표 1>과 같고 데이터의 분포도가 <그림 1>과 같을 때 클러스터의 수 K=2를 발견하기 위해서 K-means 알고리즘을 적용해 보자.

<표 1> 수입과 명품 선호 데이터

ID	수입	명품 선호도
1	150	50
2	130	53
3	90	80
4	120	85
5	115	70
6	126	76



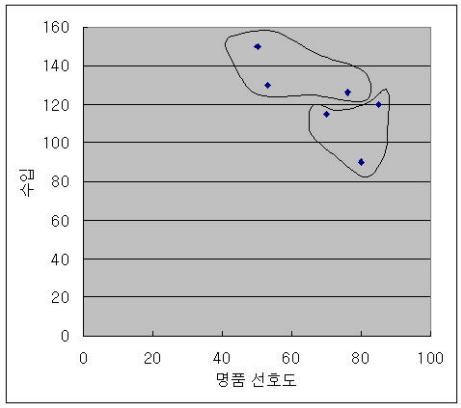
<그림 1> 수입과 명품 선호도 데이터 분포도

먼저 초기 중심점으로 ID1 과 ID3을 선택하여 거리를 구하면 <표 2>와 같다.

<표 2> 초기 중심점과 각 개체 사이의 거리

ID	1	3
1	0.0	67.1
2	20.2	48.3
3	67.1	0.0
4	46.1	30.4
5	40.3	26.9
6	35.4	36.2

선택된 중심점인 ID1과 ID3을 기준으로 가까운 개체를 할당하여 군집을 만들면 <그림 2>와 같이 {1, 2, 6}과 {3, 4, 5}라는 첫 번째 군집이 만들어진다.



<그림 2> 첫 번째 군집 생성

이것은 아직 완전한 군집이라 할 수 없고 생성된 군집을 가지고 <표 3>과 같이 새로운 중심점을 구한 뒤 다시 이 중심점과 각 개체들의 거리를 계산하면 <표 4>와 같다.

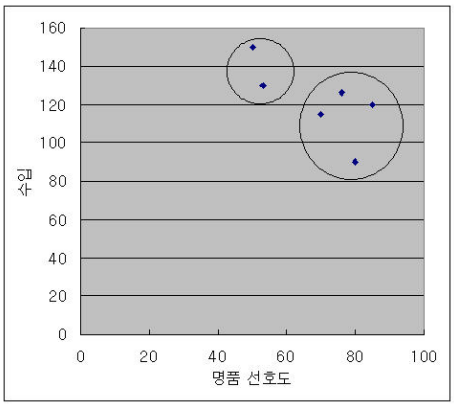
<표 3> 첫 번째 생성된 군집의 중심값

	군집1	군집2
개체	1, 2, 6	3, 4, 5
군집 기준값	(59.6, 135.3)	(78.3, 108.3)

<표 4> 생성된 군집의 중심점과 각 개체 사이의 거리

ID	군집1	군집2
1	17.56	50.40
2	8.46	33.33
3	49.68	18.38
4	29.65	13.48
5	22.81	10.67
6	18.85	17.85

이 때 {1, 2, 6} 군집에 있던 ID6은 두 번째 군집화 작업에서는 <그림 3>과 같이 {1, 2}와 {3, 4, 5, 6}으로 재배치 되어 다른 군집으로 할당된다. 새로운 군집의 중심 값으로 더 이상 개체가 할당되지 않으면 군집 작업은 여기서 종료한다.



<그림 3> 두 번째 군집 생성

<그림 3>의 결과에서 트랜잭션 수가 1/3로 줄어든 것을 알 수 있다. 이것을 Apriori 알고리즘에 적용시키면 빈발항목의 수가 줄어들어 시간의 제약을 받던 Apriori 알고리즘의 더 나은 성능이 기대된다.

실험에서는 클러스터의 밀도나 수입과 명품 선호도의 상관관계 등을 제한하여 실험하였다.

5. 결론

본 논문에서는 데이터 마이닝에서 사용되는 기법 중 연관규칙 알고리즘의 대표적인 알고리즘인 Apriori 알고리즘의 문제점인, 용량이 큰 데이터베이스에서의 트랜잭션 반복 검사로 인한 성능저하를 개선하기 위한 방법으로 Apriori 알고리즘 적용 전 단계인 전처리 단계에서 K-means 알고리즘을 사용하여 트랜잭션들을 클러스터링한 후 클러스터링을 통해 나온 각 클러스터에서 연관규칙을 찾게 되면 전체 트랜잭션보다 클러스터 내에 있는 작은 수의 트랜잭션을 검색하기 때문에 기존의 방법보다 성능은 더 나아지게 될 것이다.

향후 연구 방향으로, k-means 알고리즘의 클러스터링 속도를 개선 할 수 있는 추가적인 알고리즘의 개발이 필요하다. 또한 K-means 알고리즘 적용 시 적절한 k값을 찾는 문제도 아직 남아있다. 향후 k-means 알고리즘의 단점을 극복 할 수 있는 알고리즘을 접목하여 더 나은 결과를 찾아낼 수 있는 연구가 필요하다.

참고문헌

[1] R. Agrawal, T. Imielinski, and A.Swami. "Database Mining : A Performance Perspective.", IEEE Transactions on Knowldege and Data Engineering, 5(6):914-925, Dec., 1993

[2] M.S. Chen, J. Han, and P.S. Yu, "Data Mining: An Overview from a Database Perspective," IEEE Transaction on Knowledge and Data Engineering, Vol. 8, No. 6, pp. 866-883, Dec, 1996.

[3] R. Agrawal and R. Srikant, "Fast Algorithm for Mining Association Rules in Large Databases", Proc. of Int. Conf. on Very Large Databases, pp.487-499, 1994

[4] H. Wang, W. Wang, J. Yang, and P.S. Yu, "Clustering by Pattern Similarity in Large Data Sets," Proceedings of ACM SIGMOD, Wisconsin, pp. 394-405, June, 2002.

[5] A.K. Jain, M.N. Murty, and P.J. Flynn, "Data Clustering: A Review," ACM Computing Surveys, Vol. 31, No. 3, pp. 264-323, 1999.

[6] 김의찬, 황병연, "트랜잭션 클러스터링을 이용한 연관규칙 생성", 한국정보처리학회 춘계학술대회논문집, 제12권 제1호, pp.15-18, 2005.

[7] J.S. Park, M.S. Chen, and P.S. Yu, "An Effective hash-based Algorithm for Mining Association Rules", Proc. of ACM SIGMOD, pp.175-186, May, 1995

[8]H. Toivonen, "Sampling Large Databases for Association Rules", Proc. of Int. Conf. on Very Large Databases, pp. 134-145, Sep., 1996

[9] J. Han, J. Pei and Y. Yin, "Mining Frequent Patterns without candidate generation", Proc. of ACM SIGMOD, pp. 1-12, May, 2000