제 26 회 한국정보처리학회 추계학술발표대회 논문집 제 13 권 제 2 호 (2006. 11)

트리 데이터에서 연관규칙 추출을 위한 서브 트리 마이닝

강우준*, 신준**
*그리스도대학교 경영정보학부
e-mail : wjkang@kcu.ac.kr
**성균관대학교 컴퓨터공학과
e-mail : crashjun@ece.skku.ac.kr

Subtree Mining to extract Association rules from Tree Data

Woo Jun Kang*, Jun Shin**
*College of Management and Information Technology, Korea Christian University
**Department of Computer Engineering, Sungkyunkwan University

요 약

XML 트리 데이터들로부터 빈번 서브 트리들을 추출하는 기존 방법들은 복잡하고 다수의 입력데이터 스캐닝을 필요로 할 뿐만 아니라 빈번 서브 트리를 구하기 위해 에지 하나하나의 조인 작업을 필요로 하였다. 이는 결과적으로 많은 수행 시간을 요한다. 본 논문에서는 트리데이터를 레벨 별로 나누고 이를 마치 채로 거르듯이 필터링하여 특정 수치 이상의 출현 횟수를 가지는 노드들만을 남겨 빠르게 빈번한 서브 트리를 찾고, 이를 이용하여 XML 연관규칙들을 생성하는 방법을 제시한다. 제시된 방법을 위해서 PairSet 이라는 새로운 자료구조를 도입하였으며, 이를 이용하는 크로스필터링 알고리즘을 개발하여 제시하였다.

1. 서론

XML 문서는 트리 구조로 이루어져 있으며 이런 구조적 특성으로 인해 XML 마이닝은 주로 대상 문서들의 공통의 서브 트리 패턴들을 발견하는 방법이 가장 많이 연구되고 있는 분야이다. 기존의 RDB 에서의 Apriori 알고리즘을 이용한 빈도 아이템 집합과 후보 빈도 아이템 집합을 구분 지었듯이, XML 마이닝 역시 빈번 서브 트리 집합과 후보 빈번 서브 트리 집합을 근간으로 하여 이루어지는 연구들이 주를 이룬다. 즉 이러한 Apriori 기반의 방식은 1993 년 Rakesh Agrawal[1]에 의해 처음으로 제안된 이후, 많은 연구가 진행되어 현재까지 다양한 알고리즘들이 제안되었다.

XML 문서들로부터 연관규칙을 추출하는 것은 [2]에서 처음 제시되었으며, 이후의 연구들은 대부분 Apriori 의 방식을 따르고 있다. 그러나 Apriori 방식에서의 후보집합 생성방식의 계산 비용은 상당히 크며, 특히 패턴의 수가 많거나 길이가 긴 경우에 더욱 그러하다[3].

연관규칙을 추출하는데 있어서 핵심 포인트는 자주 발생하는 서브 트리를 추출하는 것이다. 지금까지 적지 않은 빈번 서브 트리 생성 알고리즘들이 제안되었으나 대부분의 방식이 단계적인 에지 조인 생성을 이용하므로, 레이블 개수 멱승과 같은 많은 양의 계산 및 공간을 요구한다는 단점이 있다.

본 논문의 목표는 새롭게 고안된 크로스필터링 알고리즘을 이용하여 종래 기술의 방식에 비하여 요구되는 계산량이 감소되고, 수행시간이 짧고 효율적인 방식으로 연관규칙을 추출해내는 것이다. 이를 위하여 본 논문에서 제시하는 XML 연관규칙을 찾기 위한 방법에서는 XML 집합 혹은 트리집합을 페어셋(PairSet)의 형태로 변환하고, 크로스필터링 알고리즘을 이용하여 페어셋(PairSet)에서 빈번하게 발생하는

서브트리 찾고, 크로스필터링이 끝난 후 페어셋(PairSet)의 빈번하게 발생하는 집합(Frequent Set) [F]로부터 연관규칙을 추출하는 프로세스를 수행한다.

이와 같이, 본 논문에서는 가장 복잡하고 많은 시간이 요구되는 에지 조인단계를 생략하기 위해 '페어셋(PairSet)' 이라는 새로운 구조로 이 트리들을 분석하여 저장하는 방식을 사용하였다. 이를 이용하여 최대 빈번 서브 트리들을 추출하여 기존 일부 알고리즘에서 발생했던 빈번 서브 트리를 발견하지 못하거나 중복해서 발견했던 문제점 또한 개선한다.

2. 관련연구

Agrawal 등은 [1]에서 처음으로 연관규칙의 마이닝을 소개하였다. 이후 많은 연구들이 다양한 방법을 통해 진행되어 왔다. [4]는 Apriori 와 AprioriTid 알고리즘을 제시하였다. 이 알고리즘들은 크기가 큰 아이템 집합을 찾기 위하여 여러 번의 루프를 돈다. Apriori 와 비슷한 방식을 사용하는 알고리즘들 역시 너무 많은 반복문을 수행한다. [5]에서는 이를 대체하기 위해 쿼리를 사용하기도 하였다.

몇몇의 연구는 XML 을 이용하여 데이터로부터 추출된 지식들을 표현하는데 있어서 표준모델을 도입하는데 중점을 두었으며, 최근에는 XML 문서들로부터 연관규칙을 추출하기 위한 Tool 들이 제안되기도 하였다[2][5]. 한편, [6]에서는 다양한 XML 데이터로부터 공통의 서브트리를 찾고 이로부터 빈번히 발생하는 트리를 구성하는 알고리즘을 제안하였다.

연관규칙을 찾기 위해서는 우선 빈번히 발생하는 서브트리들을 찾아야 한다. 그런데 이러한 서브트리들을 찾기 위한 후보집합을 찾는데 대한 계산 복잡도가 크기 때문에, 이 분야에서는 어떻게 하면 후보 집합의 크기를 줄일 수 있는가가 큰 이슈이다. [7]에서는 후보집합의 생성 없이 빈번한 패턴을 찾는 방법을 제시하였고 [8]에서는 XML 데이터를 여러 번 스캔하지 않고 연관규칙을 찾는 방법을 제시하였다.

3. 크로스필터링 방법

다음은 본 논문에서 제안하는 크로스필터링 알고리즘을 이용하여 XML 문서들로부터 빈번하게 발생하는 서브트리(Frequent subtree)를 추출하고 이를 바탕으로 연관규칙을 추출하는 방법을 설명한 것이다. 연관규칙이란 $X \Rightarrow Y$ 로 표현되는 규칙으로서, X 이면 Y 이다가 참이 되는 명제를 말한다. 여기서 X 와 Y 는 다음 두 조건을 만족한다.

1) $X \in F, Y \in F$ 2) $(X \not\subset Y) \wedge (Y \not\subset X)$

그림 1 은 크로스필터링 알고리즘을 통해서 연관규칙을 추출하는 과정을 보이기 위한 트리집합의 한 예이다.

Algorithm 1(CF_XAR).
Input : $\mathcal{D}$(A Set of Trees), min_sup.
Output : $\mathcal{FS}$(Set of all frequent HILoP).
(1)$\mathcal{FS} \leftarrow \mathcal{F} \leftarrow \mathcal{C} \leftarrow \phi$
(2) maxDepth ← maxDepth($\mathcal{D}$)
(3) for i ← 0 to maxDepth
(4) Given $\mathcal{D}$, Make the $PairSet[P]$ layer by layer.
(5)$\mathcal{FS}$ ← CrossFilter($[F]$.$[C]$.min_sup)

알고리즘 CF_XAR 은 XML 집합 혹은 트리집합을 페어셋(PairSet)의 형태로 변환한다. 이것은 1) 트리 집합의 모든 트리를 하나씩 DFS 방식으로 방문하면서 PairSet 을 생성하고, 2) 루트 노드에서부터 DFS 방식으로 노드들을 방문하면서, 3) 해당 depth 의 PairSet 에 방문 중인 노드의 key 가 있으면 key 에 대응하는 tid_list 에 현재 방문 중인 트리의 ID 를 추가한다.
그림 1 의 트리집합에서 PairSet 을 만들면 그림 2 와 같이 된다.

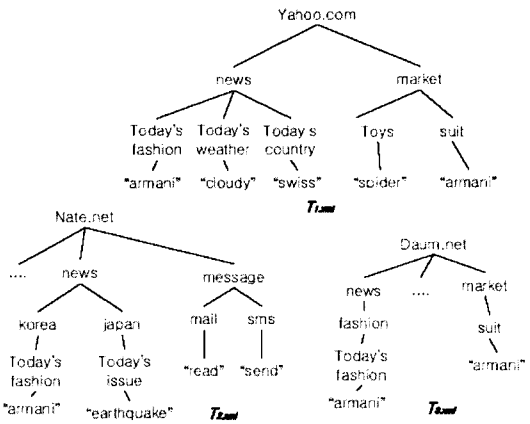

그림 1 XML 문서들의 집합(트리 집합)

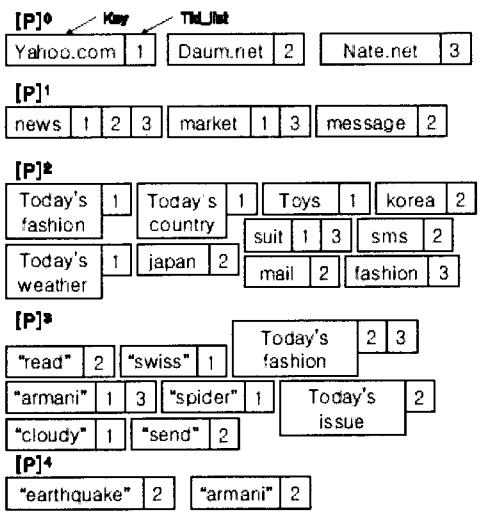

그림 2 PairSet 의 초기값

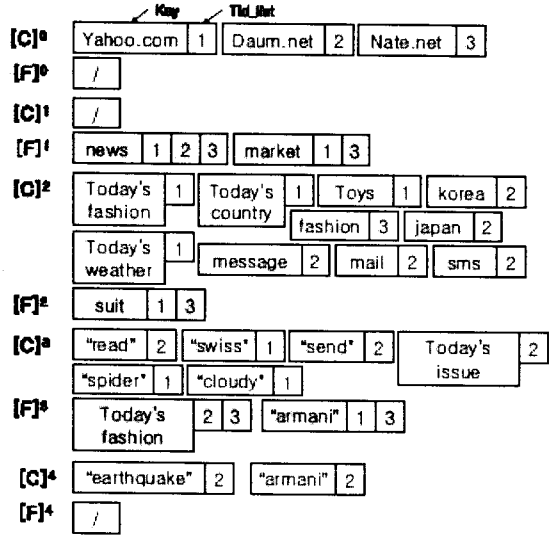

그림 3 Candidate Set 과 Frequent Set

일단 PairSet 을 구성하면 크로스필터링 알고리즘을 이용하여 빈번하게 발생하는 서브트리를 찾는다. 알고리즘 2 는 크로스필터링 알고리즘의 슈도(Pseudo) 코드이다. 크로스필터링 알고리즘은 다음과 같이 세 부분으로 구분할 수 있다. 1) PairSet 을 두 개의 집합 [C]와 [F]로 나눈다. 여기서 [C]는 후보집합(Candidate Set)을 의미하고 [F]는 빈번하게 발생하는 집합(Frequent Set)이다. 이들을 분리하는 기준은 사용자가 정의한 최소지지도 (minsup: minimum support)를 이용한다. 그림 3 에 PairSet 을 후보집합 [C]와 빈번하게 발생하는 집합[F]로 분리한 모습이 도시되어 있다. 2) 인접 레벨의 후보집합[C]에 속하는 어떤 key 가 현 레벨의 빈번하게 발생하는 집합 [F]에 속하면 그 key 에 대응하는 tid_list 를 통합하고 key 를 후보집합[C]에서 삭제한다. 3) 그래도 남아있는 후보집합[C]에 속하는 (key, tid_list)쌍들은 채로 걸러내듯이 다음 레벨로 이동시킨다. 이렇게 해서 빈번하게 발생하는 집합[F]에 남게 되는 (key, tid_list) 쌍은 그림 4 에 나와 있다.

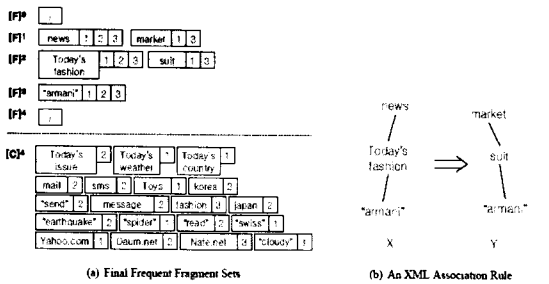

그림 4 크로스필터링의 최종결과(좌)와 만들어진 연관규칙(우)

크로스필터링이 끝난 후 [F]로부터 연관규칙을 추출하면 그림 4 의 오른쪽과 같은 결과를 얻을 수 있다.

Algorithm 2(CrossFilter).
Input : $FrequentPairSet[F]$, $CandidatePairSet[C]$, min_sup.
Output : $\mathcal{FS}$(Set of all frequent HILoP).
(1) for d:= 0 to depth
(2) Separate $PairSet[P]^d$ into $[F]^d$ and $[C]^d$ in accordance with min_sup
(3) $\mathcal{FS} := \mathcal{FS} \cup [F]^d$
(4) for d:=1 to depth
(5) for each key in key | key $\in [C]^{d-1} \cap [F]^d$
(6) $[F]^d.key.tlist := [C]^{d-1}.key.tilst \cup [F]^d.key.tlist$
(7) for l :=d-1 to 0
(8) for each key in key | key $\in [C]^d \cap [F]^l$
(9) $[F]^d.key.tilst := [C]^d.key.tlist \cup [F]^l.key.tlist$
(10) find additional (key,tlist) pairs satisfying min_sup in $([C]^d, [C]^{d-1})$
(11) remove the pairs from both $[C]^d$ and $[C]^{d-1}$
(12) $[C]^d := [C]^{d-1} \cup [C]^d$
(13) return $\mathcal{FS}$

이와 같이, 본 발명에서는 key 와 Tid-list 의 쌍들로 구성된 PairSet 이라는 구조를 도입하고, 크로스 필터링 알고리즘을 개발함으로써, 여러 개의 XML 문서 혹은 트리 데이터들을 마치 채로 걸러내듯이 걸러서 자주 발생하는 서브 트리들만을 추출하고 이로부터 연관규칙을 추출한다.

4. 성능평가

본 논문에서 수행된 모든 실험은 2.4GHz 펜티엄 4, 1GB 메모리의 PC 에서 수행되었고 OS 는 마이크로 소프트의 Windows XP 를 사용하였다. 알고리즘의 구현은 JAVA 를 이용하였다. 그림 5 는 XML 문서의 개수를 100 부터 12000 까지 증가시켰을 때 나타나는 수행시간을 나타내며, 각각의 min-sup 의 값은 0.05, 0.01,0.07 그리고 0.3 이다. 실험에 사용된 데이터들은 [4]에서 제시된 트리 생성기를 이용하였다. 트리 생성을 위해서 사용된 파라미터의 값으로써, 레이블의 수 N=100, 확률 $\rho = 0.2$, 최대 깊이 $d = 3$, 변이(perturbation) $\delta = 25$, 그리고 노드당 fan-out 의 최대 값 $f = 5$ 를 이용하였다. 그림 5,6 은 시뮬레이션의 최종결과를 보여준다. 주어진 트리아이템의 집합 $I = \{I_1, I_2, ..., I_m\}$ 가 존재하고, 주어진 XML 문서의 집합 D 에 대하여 어떤 연관 규칙 $X \Rightarrow Y$ 에 대해 다음과 같이 지지도(support)를 정의한다.

$$\text{support } (X \Rightarrow Y) = \text{freq}(X \cup Y, D) = \frac{|D_{X \cup Y}|}{|D|}$$

$D_{X \cup Y} = \{T_i \mid \forall I_j \in (X \cup Y), I_j \subset T_i$ for some $i \in [1,n], j \in [1,m]\}$, and $D_X = \{T_i \mid \forall I_j \in X,\ I_j \subset T_i$, for some $i \in [1,n], j \in [1,m]\}$.

최소 지지도란 지지도의 최소 임계치를 의미하며, 이 값보다 큰 수만큼 발생하는 서브트리를 추출하는 것이 일반적이다.

5. 결론

본 논문은 비슷한 타입의 다수의 XML 문서로부터 빈번하게 발생하는 패턴을 분석하여 연관규칙을 추출하는데 중점을 두었다. 이를 위해서 PairSet 이라는 새로운 자료구조를 도입하였으며, 이를 기반으로 빈번히 발생하는 항목만을 걸러내는 크로스필터링 알고리즘을 제안하였다. 이와 같이 본 논문에 따른 방법은 기존의 방식에 비해 요구되는 계산량을 감소시키고, 최종적으로 XML 연관규칙을 추출해낸다.

또한, 본 논문에서 제시하는 방법은 이종 환경에서의 XML 문서들을 통합하는 시스템에 요구되는 복잡도와 저장공간 그리고 수행 시간이 적은 효율적인 방식이므로 인티그레이션 시스템에 적용 가능하다.

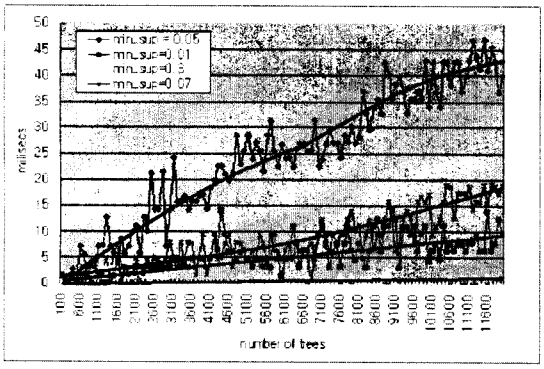

그림 5 트리의 개수와 minsup 의 변화에 따른 수행시간

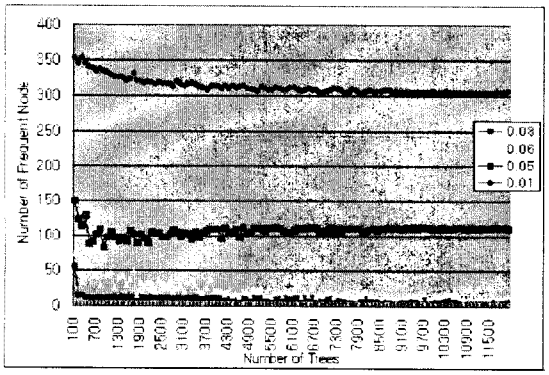

그림 6 트리의 개수에 따른 빈번히 발생하는 노드의 수

참고문헌

[1] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In Proc. of the ACM SIGMOD International Conference on Management of Data, pp.207-216, 1993

[2] D. Braga, A. Campi, S. Ceri, M. Klemettinen, and P. L. Lanzi. Mining association rules from xml data. In Proc. of the 4th International Conference on Data Ware-housing and Knowledge Discovery(DaWaK'02), volume 2454 of LNCS. Springer, pp.21-30,2002.

[3] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W.Chen, J. Naughton, and P. A. Bernstein, editors, 2000 ACM SIGMOD Intl.Conference on Management of Data, pp. 1-12. ACM Press, 05 2000.

[4] R.Agrawal and R. Srikant, Fast Algorithms for Mining Association Rules, In Proc. of the 20th International Conference on Very Large Data Bases, pp.478-499, 1994. [4] T. Imielinski and A. Virmani. MSQL : A query language for database mining. 1999.

[5] R. Meo, G. Psaila, and S. Ceri. A new SQL-like operator for mining association rules. In The VLDB Journal, pp.122-133, 1996.

[6] A. Termier, M.-C. Rousset and M.Sebag. Mining XML data with frequent trees. In DBFusionWorkshop'02, pp.87-96. 2002.

[7] Juryon Paik, Hee Young Yoon, Ungmo Kim. A new method for Mining Association Rules from a collection of XML Documents.In Proc. of ICCSA '05, Ubiquitous Web Systems and Intelligence Workshop (UWSI 2005), volume 3481 LNCS. Springer, pp.936-945, 2005.

[8] R. Meo, G. Psaila, and S. Ceri. An extension to SQL for mining association rules. Data Mining and knowledge Discovery, 2(2), pp.195-224, 1998.