대용량 데이터를 처리하기 위한 TFP-tree 기반의 점진적 빈발 패턴 마이닝 기법

이종범*, Minghao Piao*, 신진호**, 류근호*
*충북대학교 데이터베이스/바이오인포메틱스 연구실
**한국전력연구원
*{ jongbumlee, bluemhp, khryu}@dblab.chungbuk.ac.kr
**jinho@kepri.re.kr

TFP-tree based Incremental Frequent Patterns mining Method for Handling Large Data Set

Jong Bum Lee*, Minghao Piao*, Jin-ho Shin**, Keun Ho Ryu*
*Database/Bioinformatics Laboratory, Chungbuk National University
** Korea Electric Power Research Institute, Daejeon, Korea

요 약

이 논문에서는 점진적 마이닝 기법을 사용하여 대용량 전력 사용량 데이터로부터 빈발 패턴들을 찾아내고, 빈발 패턴들을 기반으로 하여 분류 작업을 효과적으로 완성하는데 목적을 두고 있다. 이를 위하여 본 논문에서는 TFP-tree 를 기반으로 하는 점진적 빈발 패턴 마이닝 기법 및 분류 알고리즘에 대해서 설명한다.

1. 서론

현재까지 알려진 대표적인 빈발 패턴 마이닝 알고리즘으로는 Apriori, FP-Growth 그리고 Apriori-TFP 알고리즘[1] 등이 있다.

그 중에서 FP-Growth, Apriori-TFP 은 대표적인 트리 기반의 알고리즘이다. 두 알고리즘은 모두 비점진적 마이닝 기법으로서 실행 속도는 FP-Growth 가 빠른 반면 메모리 사용은 Apriori-TFP 가 더욱 효과적이다. 그리고 주어진 테스트 데이터가 Sparse data 일 경우 Apriori-TFP 가 효과적인 반면 dense data 일 경우는 FP-Growth 가 더 효과적이다 [2]. 이러한 알고리즘의 특성을 기반으로 FP-tree 를 이용한 점진적 마이닝 기법에 대해서는 많은 연구가 진행 되었지만 [3, 4, 5] Apriori-TFP 를 기반으로 하는 점진적 기법에 대해서는 이루어지지 않았다. 특히 전력 사용량 데이터와 같은 대용량 데이터를 다룰 시 메모리 사용이 효과적인 기법이 필요하다.

이를 기반으로 본 논문에서는 대용량 데이터에서 효과적으로 빈발 패턴들을 찾고 이를 이용하여 분류를 할 수 있는 점진적 기법을 제시한다.

2. Incremental TFP

(표 1)은 이 논문에서 사용하는 용어들에 대한 설명이다. *DB* 로부터 기본 빈발 패턴 생성 모델이 될 $T\text{-}tree_{DB}$ 를 생성하고 Pruning 과정을 생략하는 것이 본 알고리즘의 첫 번째 단계이다. 새로 추가된 *db* 의 정보를 $T\text{-}tree_{DB}$ 에 추가하기 위하여 새로운 $T\text{-}tree_{db}$ 를 구축한다.

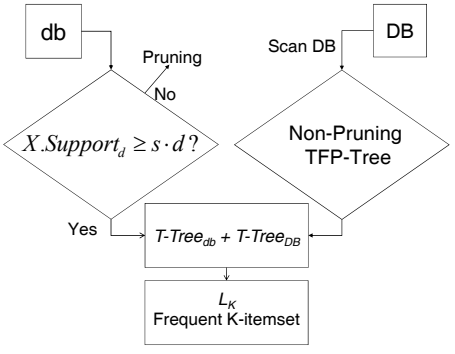
<표 1> 용어들에 대한 설명

Name	Description
DB	*오리지널 데이터 셋*
db	*새로 추가된 데이터 셋*
d	*db 안의 트랜잭션의 수*
s	*db 안의 minimum support*
L_K	*DB∪db 의 Frequent K-itemsets*
$X.Support_d$	*db 의 아이템 X 에 대한 support count*

$T\text{-}tree_{DB}$ 의 크기를 줄이고 효과적인 메모리 사용을 위하여 *db* 로부터 구축된 $T\text{-}tree_{db}$ 에서는 (식 1)를 기반으로 *pre-minimum support* [4]라는 개념을 사용하여 Pruning 작업을 한다.

$$X.Support_d \geq s \cdot d \qquad \text{(식 1)}$$

전체 Incremental TFP 에 대한 설명은 (그림 1)과 같다.

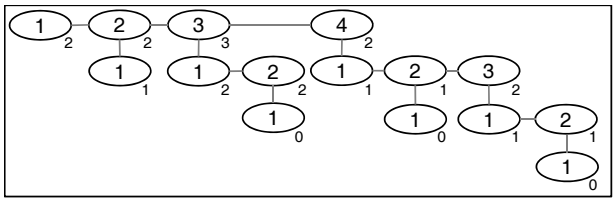


(그림 1) Incremental TFP 알고리즘

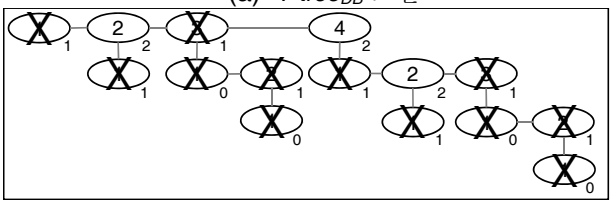
(그림 2)는 (표 2)에 주어진 예제 데이터를 사용하여 전체 알고리즘의 수행 과정을 설명한다.

<표 2> 예제 데이터

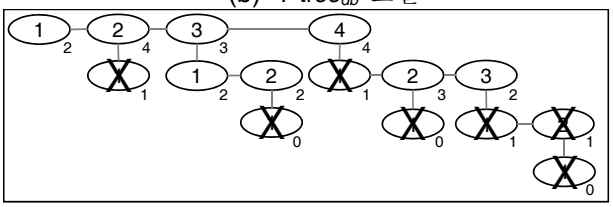
TD	DB	db
1	1, 3, 4	1, 2, 4
2	2, 3, 4	2, 3, 4
3	1, 2, 3	



(a) $T\text{-}tree_{DB}$ 모델



(b) $T\text{-}tree_{db}$ 모델



(c) $T\text{-}tree_{DB} \cup T\text{-}tree_{db}$ 모델 완성

(그림 2) Incremental TFP 구축 예

(그림 2.a)는 *DB* 로부터 Pruning 단계를 거치지 않고 구축된 $T\text{-}tree_{DB}$ 로, *DB* 로부터 구축할 수 있는 최대 트리 이다. 즉, 가능한 많은 빈발패턴을 보존하기 위한 것이다. 새로운 *db* 가 입력되면 $T\text{-}tree_{db}$ 를 구축하며 *pre-minimum support* 를 만족하지 않는 아이템들에 대해서 Pruning 을 한다 (그림 2.b). (그림 2.c)는 기존 $T\text{-}tree_{DB}$ 에 $T\text{-}tree_{db}$ 의 정보들을 추가한 최종 트리로서 주어진 support 에 따라 Pruning 을 하며 조건을 만족하는 빈발 패턴들을 찾는다. 최종 분류를 위해서는 기존의 TFPC 알고리즘 [1] 의 분류 메커니즘을 사용한다.

3. 성능평가

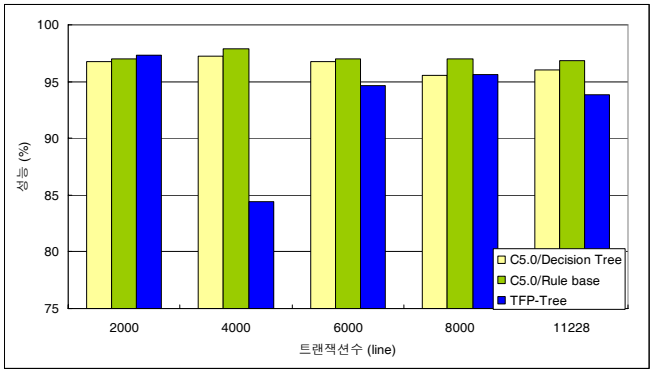
이 실험에서는 한국 전력 연구원으로부터 지원받은 11,228 건의 실제 전기 사용량 데이터를 사용하였다.

<표 3> 크기에 따른 Accuracy

트랜잭션	Number of rules	Accuracy (%)
2000	57	97.3 %
4000	2	84.6 %
6000	3	94.6 %
8000	73	95.6 %
11228	65	93.8 %

데이터 크기가 알고리즘에 미치는 영향을 연구하기 위하여 2000 건의 데이터를 기본단위로 하여 단계별로 데이터 크기를 조절하면서 성능 평가를 하였으며 그 결과는 (표 3)와 같다.

(그림 3)은 이미 상용화되어 현재 가장 많이 사용되고 있는 트리 기반, 및 rule 기반의 C5.0 과의 성능 비교이다.



(그림 3) C5.0 과의 성능비교

실험 결과, 본 논문에서 제시한 알고리즘은 작은 데이터 셋에서뿐만이 아니라 대용량 데이터에서도 비교적 좋은 성능을 나타낸다는 것을 알 수 있다.

4. 결론

본 논문에서는 대용량 데이터를 효과적으로 다루기 위한, TFP-tree 기반의 점진적 분류 알고리즘을 제시하였으며, 실험 결과 데이터 셋의 크기의 제한을 받지 않고 비교적 좋은 성능을 나타냄을 알 수 있다.

사사 표기

이 논문은 2007 년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. R01-2007-000-10926-0)

참고문헌

[1] Frans Coenen. "The LUCS-KDD TFP Association Rule Mining Algorithm" http://www.csc.liv.ac.uk/ ~frans/ KDD/ Software, Department of Computer Science, The University of Liverpool, UK, 2004.

[2] Frans Coenen, Paul Leng, and Shakil Ahmed. "Data Structure for Association Rule Mining : T-Trees and P-Trees" IEEE VOL.16, NO.6, 2004

[3] Muhaimenul Adnan, Reda Alhajj, and Ken Barker, "constructing Complete FP-Tree for Incremental mining of Frequent Patterns in Dynamic Databases" IEA/AIE, pp.363-372, 2006.

[4] Xin Li, Zhi-Hong Deng, and Shiwei Tang. "A Fast Algorithm for Maintenance of Association Rules in Incremental Databases" ADMA, pp56-63, 2006,

[5] William Cheung and Osmar R.Zaiane. "Incremental Mining of Frequent Patterns Without Candidate Gemeration or Support Constraint" ideas, pp.111, 2003