김일성종합대학학보
(자연과학)
주체103(2014)년 제60권 제7호

JOURNAL OF **KIM IL SUNG** UNIVERSITY
(NATURAL SCIENCE)
Vol. 60 No. 7 JUCHE103(2014).

페지화된 FP-Growth알고리듬을 리용한 련관규칙추출방법에 대한 연구

리효성, 안영진

현재 트랜잭션내부련관규칙추출에서 가장 많이 리용되고있는 알고리듬은 빈발패턴나무의 증식을 리용한 FP-Growth알고리듬이다.[2]

FP-Growth알고리듬은 나무구조를 리용한 련관규칙추출방법만을 제안한 알고리듬으로서 그 실현을 위한 구체적인 방도들은 정의되여있지 않다.

대용량자료기지에서 FP-Growth알고리듬을 리용하여 련관규칙추출을 진행하는데서 나서는 문제는 빈발패턴나무를 구성하고 그것을 통하여 빈발항목모임을 탐색하는 과정에 기억기자원의 제한으로부터 성능이 크게 떨어지는것이다.

그러므로 론문에서는 빈발패턴나무의 구성과 관리를 효과적으로 진행할수 있는 페지화방법을 제기하고 그에 따라 FP-Growth알고리듬을 개선하였다.

1. 빈발패턴나무의 페지화방법

1) 빈발패턴나무페지화의 구조

페지화된 FP-나무는 물리적으로 페지, 마디, 슬로트들로 구성된다.

페지는 FP-나무구성화일의 기억기입출력처리단위이며 개념적으로는 FP-나무의 마디들을 포함하고있다. FP-나무의 뿌리마디로부터 잎마디에 포함되는 항목들의 렬은 트랜잭션자료기지의 매개의 레코드들을 반영하고있으며 탐색과정에 자식마디들을 지적하기 위하여 슬로트라는 개념을 도입한다. FP-나무의 마디들은 슬로트를 리용하여 자식마디와의 련결을 진행한다. 슬로트는 자식마디의 위치 즉 페지번호와 마디번호를 포함하고있는 요소이다.[1] 매개의 마디들은 트랜잭션자료기지의 항목수와 같은 개수의 슬로트들로 구성되며 슬로트들은 자식마디의 위치와 함께 항목의 지지도를 포함한다.

2) 페지화된 FP-나무의 정의

원천자료기지 D에 대하여 다음의 구조를 가진 나무를 페지화된 FP-나무라고 부르고 다음과 같이 표시한다.

$$\begin{aligned}\mathrm{PFPT} = \{&P_i,\ N_j,\ S_{jl},\ B_k \mid 0 < i \le \mathrm{file\_size},\ 1 < k \le \mathrm{node\_count},\\ &0 < l \le \mathrm{node\_size},\ N_j \subset \mathrm{Pfloor}(j\ /\ \mathrm{page\_size},\ S_{jl} \subset N_j,\\ &B_k = [S_{jl}\text{로 부터 } N_k\text{까지의 } \text{룽} \mid l = \exists a,\ 0 \le a \le \mathrm{node\_size}],\\ &S_{jl} = [\mathrm{item},\ \mathrm{supportitem}]\}\end{aligned}$$

여기서 P_i는 페지, file_size는 FP-나무구성화일의 크기 즉 페지총수, N_j는 마디(일반나무에서의 마디와 같은 의미를 가진다.), page_size는 페지의 크기 즉 한 페지가 포함하는 마디수,

node_count는 FP-나무구성화일의 마디총수 즉 $\text{node\_count} = \text{file\_size} \times \text{page\_size}$, S_{jl} 은 마디에 포함되여 실제로 항목을 표현하는 슬로트, node_size는 마디의 크기 즉 마디에 포함되는 슬로트의 개수, B_k 는 슬로트에 련결되는 자식마디를 결정하는 릉, item은 원천자료기지의 항목을 표현하는 요소, supportitem은 item의 지지도 즉 원천자료기지에서의 출현회수이다.

3) 페지화된 FP-Growth알고리듬

페지화된 FP-Growth알고리듬의 기본내용은 빈발패턴나무구조를 화일에 페지단위로 보관하고 페지캐쉬화방법을 도입하여 나무의 증식과 참조를 페지단위로 진행함으로써 화일입출력회수를 훨씬 줄이는것이다.

련관규칙을 발굴하는 과정에 수행되는 빈발패턴나무참조회수는 원천자료기지의 크기에 따라 매우 방대해지므로 화일입출력회수를 줄이는것으로 대용량자료기지에서의 련관규칙추출속도를 개선하는것이 이 알고리듬의 목적이다.

FP-Growth알고리듬이 FP－나무의 증식과정과 구축된 FP-나무를 리용하여 빈발항목모임을 추출하는 과정을 걸치기때문에 이 2가지 알고리듬들에 페지화방법을 적용하여야 한다.[2]

페지화방법을 적용한 FP－나무의 증식알고리듬과 빈발항목모임추출알고리듬은 다음과 같다.

알고리듬 1(페지화된 FP－나무증식알고리듬)

① PFPT를 생성하고 초기화한다.

② 원천자료기지를 초기조사하고 빈발1－항목들의 모임과 그 지지도를 구하여 지지도크기순서로 정렬한다.

③ 원천자료기지를 자료레코드단위로 다시 조사하면서 다음의 단계(④－⑩)들을 수행한다.

④ 읽어낸 원천자료레코드의 항목들을 빈발1－항목들의 지지도순서로 PFPT에 추가한다.(⑤－⑩)

⑤ 뿌리마디로부터 시작하여 지지도순서로 정렬된 빈발1－항목들의 정보를 포함하는 슬로트들을 탐색하면서 차례로 나타나는 슬로트의 지지도값을 1만큼 증가시킨다.

⑥ 조건을 만족시키는 슬로트가 나타나지 않으면 정렬된 원천자료레코드의 항목들을 해당한 항목부터 시작하여 PFPT에 추가하는 동작을 진행한다.

⑦ 완충기에 마디를 추가할수 있는 페지(여유용량을 가진 페지)가 존재하는가를 검사하고 없으면 PFPT화일로부터 완충기에로 읽어들인다.

⑧ PFPT화일에도 찾으려는 페지가 없다면 완충기에 새로 페지를 창조한다.

⑨ 만약 완충기에 빈자리가 없다면 가장 많이 리용된 페지순서로 일정한 개수만큼 선택하여 PFPT화일의 해당한 위치에 쓰기하고 페지들을 소거한 다음 단계 ⑨, ⑩을 다시 반복한다.

⑩ 조건에 맞는 페지가 선택되면 새로운 마디를 창조하고 마디의 해당한 위치에 새 항목을 추가하고 지지도를 1로 한다.

이 알고리듬의 입력자료는 련관규칙을 추출하기 위한 원천자료기지이다.

알고리듬 2(페지화된 FP-Growth알고리듬)

① 알고리듬 1의 단계 ②에서 얻은 정렬된 빈발1－항목모임목록을 지지도의 거꿀순서로 순환하면서 단계 ②－⑤를 수행한다.

② PFPT에서 빈발1－항목모임에 포함되는 모든 마디들을 순환하면서 단계 ③, ④를 수행한다.

③ 해당한 마디의 모든 조상마디들을 탐색하면서 탐색방문회수가 최소지지도보다 큰 마디의 항목들을 지지도와 함께 목록에 추가한다.

④ 목록에 추가된 항목모임들의 모든 부분모임들을 구하여 같은 부분모임들끼리 지지도를 합하여 빈발항목모임에 추가한다.

⑤ 빈발항목모임의 모든 부분모임을 지지도와 함께 빈발항목색인구조에 추가한다.

⑥ 발견된 빈발항목모임들의 모든 부분모임들을 조합하여 최소확신도를 만족시키는 모든 련관규칙들을 구하여 그것의 지지도, 확신도와 함께 련관규칙목록에 추가한다.

맺 는 말

페지캐쉬화방법에서 화일접근회수를 감소시키는 원리는 다음과 같다.

우선 페지단위로 캐쉬에 보관한 페지는 항목모임의 지지도를 계산하기 위한 연산에서 재리용될 확률이 높다.

또한 리용된 페지의 린접페지들은 계산과정에 리용될 확률이 높으므로 접근요청이 들어온 페지와 린접페지들을 한번의 화일읽기로 캐쉬에 적재하고 연산을 진행하도록 하여 화일입출력회수를 요청회수의 10%이상으로 감소시킬수 있다.

페지화를 리용한 개선된 FP-Growth알고리듬에서는 원래의 FP-Growth알고리듬에 대용량자료기지에서 련관규칙추출속도를 향상시키기 위한 페지화방법을 도입하고 페지화를 리용하기 위한 처리과정을 추가하여 성능을 높이였다.

참 고 문 헌

[1] J. WeiHan et al.; Data Mining Concepts and Techniques, Morgan Kaufmann, 23～35, 2006.

[2] J. G. Carbonell et al.; Kowledge Discovery in Database: PKDD' 2006, Springer, 100～127, 2006.

주체103(2014)년 3월 5일 원고접수

Method of Mining Association Rules using Paged FP-Growth Algorithm

Ri Hyo Song, An Yong Jin

The most popular method for mining association rules is the FP-Growth algorithm.

In this algorithm, the concept of FP-Tree for indexing itemsets of rows is defined and used for extracting frequent itemsets by reducing the times of calculating supports. But this algorithm is insufficient for extracting frequent itemsets on large scale of database, as its memory management method for FP-Tree is not suitable. To solve this problem, we suggested the paging method for improving the performance of FP-Growth and upgraded its memory management method by paged FP-Growth algorithm.

Key words: association rule, FP-Growth