대용량업무자료에서 련관규칙을 발굴하기 위한 한가지 병렬처리알고리듬

윤룡한, 김진성

위대한 수령 **김일성**동지께서는 다음과 같이 교시하시였다.

《새로운 과학분야를 개척하며 최신과학기술의 성과를 인민경제에 널리 받아들이기 위한 연구사업을 전망성있게 하여야 합니다.》(《**김일성**전집》 제72권 292페지)

련관규칙발굴은 빈발항목모임탐색으로 진행되며 주어진 업무자료기지에서 빈발항목모임을 발굴하기 위한 대부분의 방법들은 Apriori방법에 의거하고있다.

선행연구[1]에서는 Apriori알고리듬을 리용하여 자료를 분할처리하는 알고리듬을 제기하였으나 Apriori알고리듬이 업무자료기지를 여러번 반복순환하는것으로 하여 업무개수가 많은 자료기지에 적용할 때 발굴시간이 긴 결함이 있다.

선행연구[2]에서는 구간사귐법을 리용한 빈발항목모임발굴방법을 제안하였고 선행연구[4]에서는 개선된 Apriori알고리듬을 제안하였으나 이 알고리듬들도 업무자료기지의 업무개수가 큰 경우에는 적용할수 없는 결함이 있다.

론문에서는 Apriori를 리용하는 선행연구[1]의 방법과 순환을 리용하는 선행연구[3]의 병렬처리알고리듬보다 속도가 빠른 위치모임을 리용한 병렬처리알고리듬을 제기하고 선행방법과의 성능비교를 진행하였다.

정의 1[1] $I:=\{i_1, i_2, \cdots, i_m\}$ 을 항목들의 모임, $D:=\{t_1, t_2, \cdots, t_n\}$ $(t_i \subseteq I)$ 을 업무자료기지라고 하면 $\text{suppcount}_T(M):=|\{k \mid 1\le k\le n,\ M\subseteq t_k\}|$ 을 항목모임 $M\subseteq I$ 의 지지수, $\text{supp}_T(M):=\text{suppcount}_T(M)/n$ 을 항목모임 $M\subseteq I$ 의 지지도라고 부른다.

정의 2[4] 항목모임 $M\subseteq I$ 의 지지수가 주어진 지지수보다 크면 항목모임 M 을 빈발항목모임이라고 부른다.

항목의 개수가 k 인 빈발항목모임전부의 모임을 F_k 로 표시한다.

정의 3 항목모임 $X\subseteq I$ 에 대하여 X 를 포함하는 업무번호들의 모임 $P_X:=\{j \mid X\subseteq t_j,\ 1\le j\le n\}$ 을 X 의 위치모임이라고 부른다.

위치모임을 리용한 빈발항목모임발굴알고리듬에 대하여 보자.

① 업무자료기지 D, 최소지지수 min_sup 를 입력한다.

② 자료기지를 순환하면서 1－빈발항목모임들과 그것의 위치모임을 찾는다.

③ $(k-1)$－빈발항목모임족 F_{k-1} 로부터 k－빈발항목모임족 F_k 를 다음과 같이 얻는다.

F_{k-1} 에 포함되는 두 $(k-1)$－빈발항목모임들의 가능한 모든 쌍 A, B 에 대하여 합모임 $C=A\cup B$ 의 농도가 k 이면 그것이 빈발항목모임인가를 다음과 같이 판정한다.

A 와 B 의 위치모임 P_A, P_B 를 $P_A=\{j_1, j_2, \cdots, j_{p_1}\}$, $P_B=\{l_1, l_2, \cdots, l_{p_2}\}$ 로 표시하자.

일반성을 잃지 않고 $p_1<p_2$ 라고 가정하자.

$\text{flag}(j_1),\ \text{flag}(j_2),\cdots,\ \text{flag}(j_{p_1})$ 을 1로, $\{1,\ 2,\ \cdots,\ n\}\setminus\{j_1,\ j_2,\ \cdots,\ j_{p_1}\}$ 의 원소들을 번호로 가지는 flag 는 0으로 설정한다.

모임 $P=\{l_j \mid \text{flag}(l_j)=1,\ 1\le j\le p_2\}$ 의 원소수가 min_sup 이상이면 C 를 F_k 에 추가한다.

④ F_{k-1} 로부터 F_k 를 구성하는 과정을 $F_k=\varnothing$ 일 때까지 반복한다.

⑤ $F=\bigcup_{j=1}^{k-1}F_j$ 를 출력한다.

정리 1 위치모임을 리용한 빈발항목모임발굴알고리듬은 빈발항목모임들을 모두 발굴한다.

증명 알고리듬의 방법으로 발굴한 k −빈발항목모임족을 F'_k 라고 하자.

$k=1$ 인 경우 알고리듬의 걸음 ②로부터 $F_1=F'_1$ 이다.

$k=p-1$ $(p\ge 2,\ k\in\mathbf{N})$ 인 경우 $F_{p-1}=F'_{p-1}$ 이라고 하자.

$\forall A\in F_p$ 에 대하여 A 의 부분모임들가운데서 원소의 개수가 $p-1$ 인 부분모임족을 $S(A)$ 라고 하면 $S(A)\subseteq F_{p-1}=F'_{p-1}$ 이다.

임의의 두 모임 $P,\ Q\in S(A)$ 에 대하여 $|P\cap Q|=p-2$ 이면 $|P\cup Q|=p$ 즉 $P\cup Q=A$ 이다. 임의의 빈발항목모임의 부분모임도 역시 빈발항목모임이므로 $S(A)$ 의 구성으로부터 $|P\cap Q|=p-2$ 인 $P,\ Q\in S(A)$ 가 존재한다.

$|P\cap Q|=p-2$ 이므로 $P,\ Q$ 에 알고리듬의 걸음 ③을 적용하여 얻어진 모임은 $p-$ 빈발항목모임이다. 그러므로 $A\in F'_p$ 즉 $F_p\subset F'_p$ 이다.

또한 $F'_p\subset F_p$ 이므로 $F_p=F'_p$ 이다.

따라서 알고리듬은 빈발항목모임들을 모두 발굴한다.(증명끝)

정리 2 위치모임을 리용한 빈발항목모임발굴알고리듬은 Apriori알고리듬보다 실행시간을 단축한다.

다음으로 위치모임을 리용한 병렬처리알고리듬에 대하여 보자.

몇가지 기호들을 약속하자.

D 를 업무자료기지, Gmin_sup 를 대역최소지지수, SCL 을 지지수목록(매 항목들의 지지수를 보관한다.), NB 를 최소지지수보다 작은 항목들의 모임족, N 을 자료분할알고리듬에 의하여 나뉘여진 부분자료모임들의 개수, FIL_k 를 원소의 개수가 k 인 빈발항목모임이라고 하자.

국부빈발항목모임발굴알고리듬은 다음과 같다.

① 업무자료기지 D, 최소지지수 min_sup 를 입력한다.

② 지지수목록 SCL 과 부의 경계 NB 를 0으로 초기화한다.

③ 자료분할알고리듬을 리용하여 업무자료기지 D 를 분할한다.

④ 위치모임을 리용한 빈발항목모임발굴알고리듬을 리용하여 부분자료기지 D_i $(1\le i\le N)$ 에서 FIL_i $(1\le i\le N)$ 들을 발굴한다.

빈발항목모임을 발굴할 때 비빈발로 되는 모임들을 NB 에 추가하며 매 항목모임들의 지지수를 SCL 에 보관한다.

대역빈발항목모임발굴알고리듬(병합알고리듬)은 다음과 같다.

① FIL을 FIL_1로 설정한다.

② $FIL_2, \cdots, FIL_N$에 포함되는 모든 항목모임들에 대하여 그 모임이 FIL에 존재하면 그 모임의 지지수를 더해주고 존재하지 않으면 그 모임을 FIL에 추가한다.

③ FIL에 있는 모든 항목모임 Item_i들에 대하여 Item_i의 지지수 $SC(\mathrm{Item}_i)$가 대역최소수 $G\min_\sup$보다 작은 경우 Item_i가 NB에 있으면 $SC(\mathrm{Item}_i)$에 NB에서 Item_i의 지지수 $SC(\mathrm{Item}_i).NB$를 더한 다음 $SC(\mathrm{Item}_i) < G\min_\sup$이면 FIL에서 Item_i를 삭제한다.

④ 대역빈발항목모임족 FIL을 출력한다.

임의로 생성한 0과 1로 이루어진 5 000×18, 7 000×22, 10 000×20형행렬을 업무자료기지로 놓고 선행연구[1, 3]의 방법과 론문의 방법들을 적용하여 빈발항목모임의 생성시간을 비교한 결과는 표와 같다.

표. 빈발항목모임생성시간의 비교결과

업무개수/개	방법[1]	방법[3]	론문의 방법
5 000	4 200ms	3 950ms	3 930ms
7 000	7 420ms	6 950ms	5 640ms
10 000	9 560ms	8 780ms	8 120ms

표에서 보는바와 같이 론문에서 제안한 방법이 업무의 개수가 작은 자료기지에 대하여서는 선행연구들의 방법들과 실행시간이 비슷하지만 업무개수가 커지면 속도가 상대적으로 더 빠르다.

참 고 문 헌

[1] N. Y. Suryavanshi et al.; Int. J. Comput. Appl., 112, 4, 37, 2015.

[2] K. Spandana et al.; Int. J. Comput. Appl., 155, 10, 22, 2016.

[3] A. Parveen et al.; Int. J. Comput. Appl., 146, 2, 16, 2016.

[4] M. G. Ingle et al.; Int. J. Comput. Appl., 112, 4, 37, 2015.

주체107(2018)년 12월 5일 원고접수

A New Parallel Algorithm for Mining Association Rules in Large Transaction Database

Yun Ryong Han, *Kim Jin Song*

We propose a new parallel algorithm for mining frequent itemset using position set and evaluate its effectiveness. The experimental results indicate that the proposed algorithm is faster than Apriori algorithm and the algorithm that is using transposition.

Key words: frequent itemset, position set, association rule mining