그래프 스트림에서 점진적 빈발 패턴 검출

Incremental Frequent Pattern Detection in Graph Streams

정 재 윤, 최 도 진, 복 경 수, 유 재 수+
충북대학교 정보통신공학부

Jaeyun Jeong, Dojin Choi, Kyungsoo Bok, Jaesoo Yoo
School of Information And Communication Engineering, Chungbuk National University

요약

그래프 스트림 데이터에 대한 활용이 증가됨에 따라 빈발 패턴을 검출하는 연구가 활발하게 진행되고 있다. 본 논문에서는 슬라이딩 윈도우 내에 변경된 부분만을 계산하는 점진적인 빈발 패턴 검출 기법을 제안한다. 제안하는 기법은 윈도우에서 변경되는 부분만 계산함으로써 중복된 계산을 감소시킨다. 또한, 간선 관리 테이블을 이용해 관련이 없는 패턴들을 제거함으로써 의미 있는 빈발 패턴만을 검출한다.

I. 서론

그래프 스트림이란 사물간의 관계나 인적 네트워크 등을 그래프 구조로 표현한 데이터가 연속적으로 입력되는 것을 말한다. 최근 이런 그래프 스트림이 IoT 기반 센서 네트워크, 소셜 네트워크 등 다양한 분야에서의 활용이 증가되면서, 그에 따른 다양한 연구가 활발히 진행되고 있다[1].

그래프 스트림에서 빈발 패턴을 찾는 것은 중요한 의미를 갖는다. 빈발 패턴이란 특정 기간 동안 일정 횟수 이상 등장한 패턴들을 말한다. 이 빈발패턴을 이용하면 여러 가지 응용이 가능하다. 예를 들어, 소셜 네트워크에서 사용자간에 빈발하게 교류하는 패턴을 검출하여 커뮤니티를 찾거나, 센서 네트워크에서 빈발 패턴을 이용하여 기계의 고장이나 제품의 결함 등을 찾아낸다[2].

그래프 스트림은 데이터가 끝없이 빠른 속도로 생성되기 때문에 빈발 패턴 검출을 할 때, 빠르고 효율적으로 저장하는 기법과 데이터 생성속도에 상응하는 처리속도가 요구된다. 또한 의미 있는 패턴을 검출하기 위해선 연결성도 고려해야 한다. 만일 연결성을 고려하지 않는다면 서로 관련이 없는 패턴이 발생할 수 있다. 만약 검출된 두 패턴이 연결되어 있지 않다면, 두 패턴은 서로 독립적인 패턴으로 인식 해야만 한다.

[3]에서는 그래프 데이터를 효율적으로 저장하여 빈발 패턴을 찾는 기법을 제안하였다. 하지만, 빈발 패턴을 검출할 때, 중복 연산을 수행함으로써 연산량이 증가한다. 또한, 서로 연결되지 않은 간선들을 같은 패턴으로 인식하는 문제점을 가지고 있다. [4]에서는 이웃하는 간선 정보를 테이블로 관리하여 이웃한 빈발 패턴을 찾는다. 하지만 [3]과 마찬가지로 중복 되는 연산이 수행되는 문제점을 가진다.

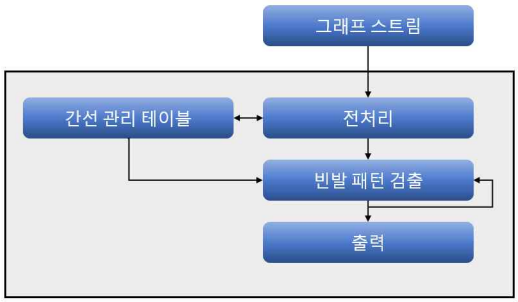
본 논문에서는 그래프 스트림에서 연산량을 감소시키기 위해 점진적 빈발 패턴 검출 기법을 제안한다. 제안하는 기법은 윈도우에서 변경되는 부분만 계산함으로써 중복된 계산을 감소시킨다. 그리고 간선 관리 테이블을 이용하여 연결성을 고려해 관련된 빈발 패턴 검출을 수행한다.

II. 제안하는 빈발 패턴 검출 기법

1. 전체 처리과정

제안하는 기법은 슬라이딩 윈도우 내에 변경된 부분과 기존에 검출한 패턴을 이용하여 계산함으로써 중복연산을 감소시킨다. 또한 서로 관련이 있는 패턴들만 검출하기 위해 연결성을 고려한다.

그림 1은 제안하는 기법의 전체 처리 과정을 보여준다. 전처리 단계는 그래프 스트림이 입력될 때, 그 데이터를 메모리에 효율적으로 저장하기 위한 단계이다. 이때, 간선 관리 테이블을 이용해 간선 정보와 간선 이름을 매핑하여 간선을 간략하게 표현한다. 빈발 패턴 검출 단계에서는 서로 연결된 패턴을 찾기 위해 간선 관리테이블을 이용한다. 한 슬라이딩 윈도우에 대한 빈발 패턴 결과는 출력하고, 다음 윈도우에서 점진적 처리를 하기 위해 결과의 통계 정보는 다시 빈발 패턴 검출 단계로 보낸다.



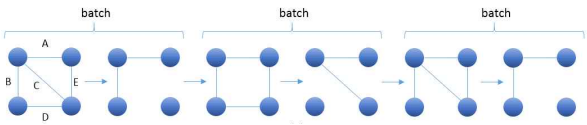
▶▶ 그림 1. 전체 처리과정

+ 교신저자 : yjs@chungbuk.ac.kr
이 성과는 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원(No. 2016R1A2B3007527), 한국과학기술정보연구원의 "초고성능컴퓨팅기반 건강한 고령사회 대응 빅데이터 분석기술개발(K-17-L03-C02-S02)" 사업, 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터 육성 지원사업의 결과로 수행되었음(IITP-2017-2013-0-00881)

2. 전처리 및 간선 관리 테이블

전처리는 그래프 스트림이 입력될 때, 메모리에 저장하는 단계이다. 이 단계에서는 스트림 데이터를 빠르고 효율적으로 저장하기 위해 DSMatrix이라는 2차원 배열구조를 이용한다. 그리고 간선에 대한 정보를 간선의 이름과 매핑하기 위해 간선 관리 테이블을 사용한다.

DSMatrix는 그래프를 메모리에 저장하기 위한 구조이다. 하나의 윈도우 슬라이드는 사용자가 정한 수만큼의 배치로 이루어진다. 그림 2에서는 1개 배치는 2개 그래프, 1개 슬라이딩 윈도우는 3개 배치로 정하고 각 간선 A, B, C, D, E에 대해 발생유무를 1 또는 0으로 표현한다.



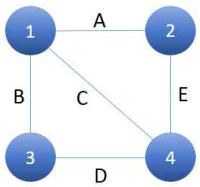
(a) 그래프 스트림 데이터

row	Contents							
A	1	1	;	1	1	;	1	1
B	1	1	;	1	0	;	1	1
C	1	0	;	0	1	;	1	0
D	0	0	;	1	0	;	0	0
E	1	0	;	1	0	;	1	0

(b) DSMatrix

▶▶ 그림 2. DSMatrix 생성 과정

간선 관리 테이블은 간선의 정보와 간선의 이름을 매핑시키고 이웃한 간선인지를 판별하기 위해 사용된다. 그림 3은 간선 관리 테이블을 보여준다. 간선의 연결 정보를 저장할 때 정점은 오름차순으로 정렬한다. 그리고 이웃한 간선인지 판별할 때에는 두 개의 간선에 중복된 정점이 있는지 확인한다. 예를 들어, 그림 3에서 A와 B는 1이라는 정점이 중복되어 있기 때문에, 이웃한 간선이다.



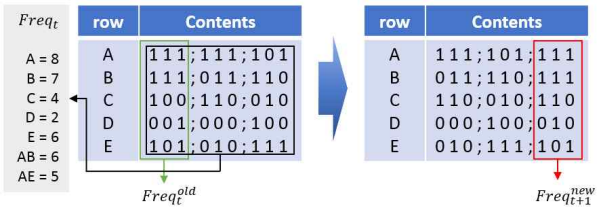
Vertex 1	Vertex 2	Edge Name
1	2	A
1	3	B
1	4	C
2	4	D
3	4	E

▶▶ 그림 3. 간선 관리 테이블

3. 빈발 패턴 검출

이 단계는 하나의 슬라이딩 윈도우 안에서 빈발하는 패턴을 찾는 단계이다. 슬라이딩 윈도우는 변경될 때 겹치는 데이터는 항상 존재한다는 점을 이용하여 중복된 부분의 계산 결과를 활용해 새로운 데이터만 계산하여 점진적으로 빈발 패턴을 검출한다.

기본적인 빈발 패턴 검출은 먼저 단일 간선에 대해 빈발도를 계산한다. 빈발도는 한 윈도우 슬라이드에서 나타난 횟수를 나타낸다. 그림 4는 점진적 빈발 패턴 검출 과정을 보여준다. 그림 4(a)에서는 A=8, B=7, C=4, D=2, E=6 이다. 여기서 임계치가 5라면, C, D를 제외한 A, B, E가 빈발하는 간선이다. 그 다음 빈발하는 간선끼리 AND 연산을 수행하여 다시 빈발도를 계산한다. 즉, AB = 111;011;100 = 6이므로 AB도 빈발하는 패턴이다. 이때, 간선 관리 테이블의 정보를 이용해 B와 E같이 연결되지 않은 간선끼리는 계산하지 않는다. 이 과정을 더 이상 빈발하는 패턴이 나오지 않을 때 까지 반복한다. (a)에서 빈발 패턴은 A, B, E, AB, AE 이다. 그리고 (a)에 $Freq_t$와 같이 단일 간선과 빈발 패턴에 대해서 빈발도를 저장한다. 점진적 빈발 패턴 검출은 앞서 계산된 $Freq_t$를 이용하여 슬라이딩 윈도우의 변경된 부분만을 계산하여 기존 패턴들의 빈발도를 계산한다. 그림 (b)와 같이 새로운 배치가 입력되면 $Freq_t^{old}$와 $Freq_{t+1}^{new}$를 계산한다. 이 두 값은 단일 간선과 검출된 패턴의 빈발도, 즉 A, B, C, D, E, AB, AE의 빈발도를 계산한다. 이렇게 계산한 후 식 (1)을 이용하면 t+1에서 각 단일 간선의 빈발도와 기존 빈발 패턴에 대한 빈발도를 알 수 있다. 그림 (b)에서 C 간선 같이 새롭게 빈발하는 패턴이 등장한 경우엔 기존의 방법 같이 패턴들을 검출한다.



(a) t에서 DSMatrix (b) t+1에서 DSMatrix

▶▶ 그림 4. 점진적 빈발 패턴 검출 과정

$$Freq_{T+1} = (Freq_T + Freq_{T+1}^{new}) - Freq_T^{old} \quad (1)$$

Ⅲ. 결론

본 논문에서는 중복되는 연산량을 줄이기 위해 이전 윈도우에서 계산된 데이터를 활용한 점진적 빈발 패턴 검출 기법을 제안하였다. 그리고 간선 관리 테이블을 통해 이웃하는 간선 정보를 효율적으로 관리하고 서로 연관된 패턴을 찾는데 활용한다. 향후에는 다양한 실험평가를 통해 제안하는 기법의 우수성을 입증할 예정이다.

■ 참 고 문 헌 ■

[1] S. K. Tanbeer, C. K. Leung, J. J. Cameron, "Interactive Mining of Strong Friends from Social Networks and its Applications in E-Commerce," J. Org. Computing and E. Commerce 24, pp.157-173, 2014

[2] C. C. Aggarwal, Y. Li, P. S. Yu, R. Jin, "On Dense Pattern Mining in Graph Streams," PVLDB 3, pp.975-984, 2010

[3] P. Braun, J. J. Cameron, A. Cuzzocrea, F. Jiang, C. K. Leung, "Effectively and Efficiently Mining Frequent Patterns from Dense Graph Streams on Disk," Proc. KES, pp.338-347, 2014

[4] A. Cuzzocrea, Z. Han, F. Jiang, C. K. Leung, H. Zhang, "Edge-based Mining of Frequent Subgraphs from Graph Streams," Proc. KES, pp.573-582, 2015