대용량 그래프에서 k-차수 인덱스 테이블을 이용한 RDBMS 기반의 효율적인 최단 경로 탐색 기법

홍지혜, 한용구, 이영구*
경희대학교 컴퓨터공학과
e-mail:{hjhh, ykhan, yklee}@khu.ac.kr

RDBMS based Efficient Method for Shortest Path Searching over Large Graphs using K-degree Index Table

Jihye Hong, Yongkoo Han, Young-Koo Lee*
Dept of Computer Engineering, Kyung Hee University

요 약

최근 소셜 네트워크의 등장과 기술의 발달로 인해 빅 데이터가 등장하였다. 특히, 소셜 네트워크나 웹 데이터 등과 같은 빅 데이터를 이용하는 애플리케이션이 많아지고 있다. 이러한 그래프 데이터는 크기가 매우 방대하여 인-메모리 기법을 통해 연산하기 어렵다. 최근 대용량 그래프 상에서 효율적인 최단 경로 탐색을 위해 부분 최단 경로를 저장하는 인덱스 테이블을 활용한 기법이 제안되었으나, 인덱스 참조율을 고려하지 않아 비효율적이다. 본 논문에서는 인덱스 참조율이 높은 노드의 차수를 이용한 k-차수 인덱스 테이블을 이용한 효율적인 최단 경로 탐색 기법을 제안한다. 실험을 통하여 제안하는 기법이 거리 기반 인덱스를 이용한 기존의 기법에 비해 약 12% 정도 성능이 향상됨을 보였다.

1. 서론

최근 소셜 네트워크의 등장과 센서 기술의 발달 등으로 인해 빅 데이터가 등장하였다. 특히, 소셜 네트워크 데이터나 웹 데이터, 교통 데이터 등 그래프 데이터를 이용하는 애플리케이션이 많아지고 있다. 이와 같은 그래프 데이터는 크기가 매우 빠르게 증가하고 있기 때문에, 데이터의 크기가 매우 방대하여 인-메모리(in-memory) 기법을 통해 연산하기 어렵다. 이에 대용량 그래프 데이터를 디스크 기반(disk-based)으로 처리하는 기법에 대한 관심이 증가하고 있다.

최근 대용량 그래프 상에서 부분 그래프나, 최단 경로 등을 탐색하기 위한 기법[1-3]들이 연구되었다. 특히 FEM 프레임워크[3]는 그래프 탐색을 위한 RDB 기반의 관계형 연산자들을 제안하였다. RDB는 너비 우선 탐색, 도달 가능성 질의 등의 다양한 기능과 안정된 인프라를 제공하여 대용량 그래프를 처리하기에 적합하다. 이 연구는 또한 그래프 내에서 최단 경로를 효율적으로 탐색하기 위해, 미리 계산된 부분 경로를 저장하는 인덱스 테이블을 제안하였다. 인덱스 테이블은 임계 거리 미만의 최단 경로들로만 구축한다.

그러나 이와 같은 인덱스 테이블 구축 방법은 인덱스의 크기가 증가에 따라 인덱스 참조 비율이 낮아지는 문제점이 있다. 최단 경로에 포함될 확률이 높은 경로들로 인덱스 테이블을 구성하면 인덱스 참조 비율이 높아져 대용량 그래프의 최단 경로 탐색 성능을 향상시킬 수 있다.

본 논문에서는 대용량 그래프에서 차수 인덱스 테이블을 이용한 RDB 기반의 효율적인 최단 경로 탐색 기법을 제안한다. 이를 위해 노드의 차수와 최단 경로와의 관계를 분석하고, 차수 기반의 인덱스 테이블로 효율적인 최단 경로를 탐색하는 알고리즘을 제안한다. 실험을 통해 제안하는 차수 기반 인덱스가 기존의 거리 기반 인덱스에 비해 최단 경로 탐색에 효율적임을 보인다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 RDB 기반 그래프 탐색 프레임워크를 소개하고, 3장에서는 차수 인덱스 테이블을 이용한 최단 경로 탐색 기법을 제안한다. 4장에서는 실험을 통해 제안하는 기법을 통해 효율적으로 최단 경로를 탐색할 수 있음을 보이며, 5장에서는 결론을 맺는다.

---

* 교신 저자

이 논문은 2013년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2013R1A2A1A05056375).

2. RDB 기반의 효율적인 그래프 탐색 프레임워크

최단 경로 탐색, 도달 가능성 탐색 등과 같은 대부분의 그래프 탐색 알고리즘들은 질의에 대한 결과를 포함할 가능성이 높은 노드들을 반복적으로 확장하며 결과 값을 찾는 공통된 패턴을 가지고 있다[3]. FEM 프레임워크[3]는 그래프 탐색에 공통적으로 사용되는 연산을 지원하기 위해 3가지 기본적인 연산자를 제공한다.

· F-operator는 현재까지 살펴본 노드들 가운데 다음 순서로 확장할 노드를 선택한다.

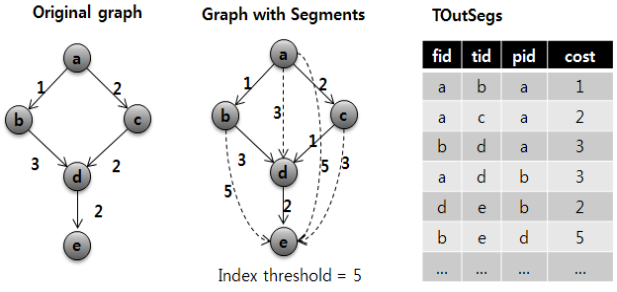
· E-operator는 F-operator를 통해 선택된 노드 집합 F^k에서 도달 가능한 모든 노드들로 확장한 결과 집합 E^k를 반환한다.

· M-operator는 현재까지 방문한 노드들의 집합인 A^k와 확장된 노드 집합 E^k를 기반으로 방문한 노드 집합을 A^{k+1}로 갱신한다.

Breadth First Search(BFS) 방식의 탐색은 한 번의 연산으로 많은 노드를 확장하여 탐색 공간과 시간 비용을 절약할 수 있으나, 대용량 그래프에서 긴 경로를 갖는 최단 거리 탐색이 발생할 경우 알고리즘의 반복 횟수가 성능에 영향을 미친다. FEM 프레임워크는 임계 거리 미만의 경로 세그먼트에 대해 미리 계산된 인덱스 테이블을 구성하여 효율적으로 최단 경로를 탐색하였다. 경로 세그먼트들을 기존의 노드 확장과 동일한 방법으로 확장하여, 불필요한 반복을 줄일 수 있다.

그림 1은 인덱스 테이블의 예시이다. 그림 1(a) 그래프의 모든 노드에서 l_{thd}이하의 거리로 도달할 수 있는 부분 최단 경로인 경로 세그먼트가 TOutSegs에 저장된다. 그림 1(c)와 같이, TOutSegs 테이블은 fid, tid, pid, cost의 4개의 열로 구성된다. fid는 각 경로 세그먼트의 출발 노드, tid는 도착 노드, pid는 tid의 부모 노드이며, cost는 경로 세그먼트의 거리를 나타낸다. 그림 3(b)의 점선은 경로 세그먼트를 나타낸다.

예를 들어, $l_{thd}=5$이므로 $\delta(a,d)=4$인 경로 세그먼트 (a,d) 가 생성되어야 한다. *TOutSegs*의 4번째 레코드는 이 세그먼트를 저장한다.



fid	tid	pid	cost
a	b	a	1
a	c	a	2
b	d	a	3
a	d	b	3
d	e	b	2
b	e	d	5
...	...	...	...

그림 1 인덱스 테이블의 예시

3. k-차수 인덱스 테이블

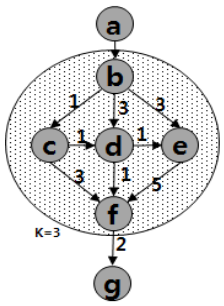
최단 경로 탐색에서 매개 중심성(betweenness centrality)과 차수(degree)가 높은 노드는 탐색의 성능에 중요한 영향을 준다. 매개 중심성은 특정 노드를 거치는 최단 경로의 수를 나타내므로, 차수가 높은 노드가 큰 매개 중심성을 가질 확률이 높다[4,5]. 따라서 차수가 높은 노드가 인덱스 테이블에 포함되면 인덱스 테이블의 참조 비율이 높아서 최단 경로 탐색의 성능을 향상시킬 수 있다.

본 논문에서는 매개 중심성 계산이 모든 쌍의 최단 경로를 찾는 추가적인 연산이 필요한 점을 고려하여, 임계 차수 이상의 노드들 간의 최단 경로를 사전에 계산한 값들로 인덱스 테이블을 구축한다.

정의 1. 지역 최단 경로 부분 그래프 $G=|V,E|$에서 두 노드 $u,v\in V$의 최단 경로이며, $lsp(u,v)=u\rightarrow m_1\rightarrow m_2\rightarrow\ldots\rightarrow v$로 표현한다. 이 때, 지역 최단 경로의 모든 노드들은 V의 원소이다.

정의 2. k-차수 인덱스 테이블 임계 차수가 k일 때, 차수가 k이상인 모든 노드들의 집합 V와 임의의 $u,v\in V$ 사이의 모든 에지 E로 구성된 부분 그래프 $G=|V,E|$내의 모든 지역 최단 경로를 미리 계산하여 저장한 테이블이다.

정의 3. 인덱스 테이블 참조율 최단 경로 탐색 횟수가 n회이고, 찾은 최단 경로에 사전에 계산한 최단 경로가 포함되는 횟수가 m회 일 때, 인덱스 테이블 참조율은 m/n이다. 인덱스 테이블 참조율이 높을수록 인덱스 테이블을 통해 효율적으로 최단 경로를 탐색할 수 있다.



(a) original graph

fid	tid	pid	cost
b	e	d	2
b	d	c	2
b	f	d	3
...	...	...	...

(b) 3-degree index table

그림 2 3-차수 인덱스 테이블의 예시

그림 2는 3-차수 인덱스 테이블의 예시이다. 그림 2(a)의 그래프에서 차수가 3이상인 모든 노드와, 노드들 간의 에지로 부분 그래프 G가 형성되고, G내의 모든 지역 최단 경로를 계산하여 그림 2(b)와 같이 테이블로 구성한다. 거리 기반 인덱스 테이블에서 임계 거리가 2라면, 거리가 3인 경로 세그먼트 (b,f)는 생성되지 않으나, 이 세그먼트는 (a,f), (a,g), (b,g) 등의 경로에 포함되는 매개 중심성이 높은 세그먼트이다. 이처럼 차수 기반의 인덱스 테이블은 매개 중심성이 높은 경로 세그먼트를 생성한다.

알고리즘 1은 차수 인덱스 테이블을 구축하는 알고리즘이다. 인덱스 테이블을 구축하기 위한 임계 차수는 k이다. 일단 모든 노드들의 차수 정보를 계산하고(line 1), 임계 차수 이상의 노드들을 확장할 노드로 선택한다. (line 3) 선택된 노드들을 임계 차수 이상의 노드들과 연결된 에지로 확장하고, (line 4) k-차수 인덱스 테이블에 병합한다. (line 5) 이 과정을 더 이상 확장할 노드가 없을 때 까지

반복하여 k-차수 인덱스 테이블을 구축한다. 반복이 끝나면, k-차수 인덱스 테이블에 추가되지 않은 남은 에지들을 추가한다. (line 7)

알고리즘 1 차수 인덱스 테이블 구축 알고리즘

Algorithm ConstructionDegreeIndex(Threshold k)
• **Input:** degree threshold k • **Output:** k-degree index table
1: Calculate degree of all nodes 2: **REPEAT** 3: Select a set of frontier nodes which have degree value more than k 4: Expand edges using degree threshold k 5: Merge expanded nodes into k-degree index table 6: **UNTIL** There is no frontier node 7: Insert remain edges into degree index table 8. **RETURN** S

최단 경로 탐색은 도착 노드에 도달 가능한 에지가 더 이상 없을 때까지 반복하여 에지를 확장한다. k-차수 인덱스 테이블에 저장된 지역 최단 경로 및 에지를 확장하여, 반복 횟수를 줄일 수 있다. 기존의 거리 기반 인덱스 테이블과 비교하여, 차수 인덱스 테이블에 포함된 에지들의 평균 매개 중심성이 높다. 에지 e의 매개 중심성은 최단 경로에 포함되는 빈도수와 연결되므로, 평균 매개 중심성이 클수록 임의의 최단 경로 탐색에서 인덱스 테이블의 참조율이 높다.

4. 실험

약 17만 개의 에지로 구성된 네트워크 데이터 셋[6]을 상용화 데이터베이스에 저장하여 실험을 진행하였다. 에지 가중치의 범위는 0.03~9.9이다. 제안하는 기법의 우수성을 입증하기위해 기존의 거리 기반 인덱스(distance-based index, DISIDX)와 제안하는 k-차수 인덱스(degree-based index, DEGIDX)를 사용한 최단 경로 탐색 시간을 비교하였다. 거리 기반 인덱스는 임계 거리 3으로 구축하였으며, k-차수 인덱스 테이블은 임계 차수 5로 구축하였다. 인덱스의 크기는 근사하며, 약 30만개의 행으로 구성된다.

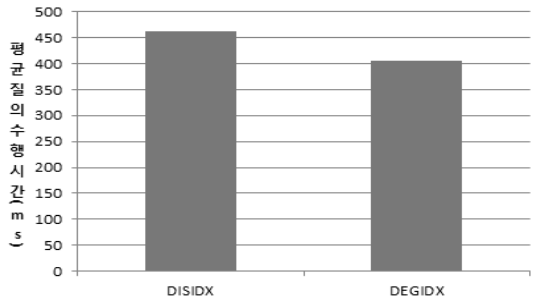


그림 3 최단 경로 탐색 질의 평균 수행 시간

그림 3은 시작 노드와 도착 노드를 임의로 선택하여 최단 경로 탐색 질의를 각각의 인덱스를 이용하여 수행한 결과이다. 총 500회의 최단 경로 탐색 질의를 수행하여 평균 시간을 측정하였다. 평균 질의 수행 시간은 5-차수 인덱스 테이블을 사용하였을 때 약 12% 개선되었다.

5. 결론

본 논문에서는 대용량 그래프에서 k-차수 인덱스 테이블을 이용한 RDB 기반의 최단 경로 탐색 기법을 제안한다. 제안하는 k-차수 인덱스 테이블은 기존의 거리 기반 인덱스 테이블에 비해 인덱스 테이블 참조율이 높아, 효율적인 최단 경로 탐색을 지원한다. 실험을 통하여 제안하는 기법이 거리 기반 인덱스 테이블을 이용한 기법에 비해 약 12% 정도 성능이 향상됨을 보였다.

참고문헌

[1] C. Wang, W. Wang, J. Pei, Y. Zhu, and B. Shi, “Scalable mining of large disk-based graph databases,” In SIGKDD, pages 316-325, 2004.

[2] R. H. Möhring, H. Schilling, B. Sch¨utz, D. Wagner and T. Willhalm, “Partitioning Graphs to Speed Up Dijkstra’s Algorithm,” In: Nikoletseas, S.E. (ed.) WEA 2005. LNCS, vol. 3503, pp. 189 - 02. Springer, Heidelberg.

[3] J.Gao, R.Jin, J.Zhou, J.Yu, X.Jiang and T.Wang, “Relational Approach for Shortest Path Discovery over Large Graphs,” In: PVLDB, 5(4), pages 358-369, 2011.

[4] A. B. David, M. Kamesh, ″A graph-theoretic analysis of the human protein-interaction network using multicore parallel algorithms,″ Proc. 6th Workshop on HiCOMB, 2007.

[5] A. De. Montis, S. Caschili, ″Nuraghes and landscape planning: Coupling viewshed with complex network analysis,″ In: Landscape and Urban Planning, vol. 105, Issue 3, pp. 315-324, 2012.

[6] Condensed matter collaborations 2005 dataset. M. E. J. Newman, ″The structure of scientific collaboration networks,″ Proc. Natl. Acad. Sci. USA 98, 404-409, 2001.