https://doi.org/10.5392/JKCA.2019.19.01.141

분산 환경에서 경로 질의 기반 서브 그래프 탐색 기법

Subgraph Searching Scheme Based on Path Queries in Distributed Environments

김민영*, 최도진*, 박재열*, 김연동*, 임종태*, 복경수*, 최한석**, 유재수*
충북대학교 정보통신공학과*, 목포대학교 컴퓨터공학과**

Minyoung Kim(cystitis@chungbuk.ac.kr)*, Dojin Choi(mycdj91@chungbuk.ac.kr)*,
Jaeyeol Park(yeols@chungbuk.ac.kr)*, Yeondong Kim(yeon5661@chungbuk.ac.kr)*,
Jongtae Lim(jtlim@chungbuk.ac.kr)*, Kyoungsoo Bok(ksbok@chungbuk.ac.kr)*,
Han Suk Choi(chs@mokpo.ac.kr)**, Jaesoo Yoo(yjs@chungbuk.ac.kr)*

요약

개체 간의 상호 작용을 나타내기 위해 그래프 데이터 형태의 네트워크가 많은 애플리케이션에서 사용되고 있다. 최근에는 빅데이터 기술의 발달로 처리해야할 네트워크의 크기가 점점 커짐에 따라 하나의 서버에서 이를 처리하기 어려워졌기 때문에 분산 처리의 필요성 또한 증가하고 있다. 본 논문에서는 이러한 그래프 데이터가 분산 저장되어있는 환경에서 서브 그래프 탐색을 효율적으로 수행하기 위한 분산 처리 시스템을 제안한다. 불필요한 탐색을 줄이기 위해 데이터의 통계정보를 활용해 확률적인 스코어링을 통해 탐색 순서를 정한다. 그래프 네트워크의 정점과 차수의 관계는 데이터의 종류에 따라 다른 특성을 보일 수 있기 때문에 여러 분포적 특성을 갖는 그래프에 대해 다른 스코어링 방법을 통해 불필요한 탐색을 줄이기 위한 스코어를 계산하여 탐색 순서를 결정한다. 결정된 순서에 따라 그래프가 분산 저장된 서버에서 순차적으로 탐색한다. 성능평가에서는 제안하는 기법의 우수성을 입증하기 위해 기존 기법과의 비교를 수행하였으며, 그 결과 기존 기법보다 탐색 시간이 약 3~10% 향상됨을 보였다.

■ 중심어 : | 그래프데이터 | 그래프 탐색 | 분산 처리 | 서브그래프 매칭 | 빅데이터 |

Abstract

A network of graph data structure is used in many applications to represent interactions between entities. Recently, as the size of the network to be processed due to the development of the big data technology is getting larger, it becomes more difficult to handle it in one server, and thus the necessity of distributed processing is also increasing. In this paper, we propose a distributed processing system for efficiently performing subgraph and stores. To reduce unnecessary searches, we use statistical information of the data to determine the search order through probabilistic scoring. Since the relationship between the vertex and the degree of the graph network may show different characteristics depending on the type of data, the search order is determined by calculating a score to reduce unnecessary search through a different scoring method for a graph having various distribution characteristics. The graph is sequentially searched in the distributed servers according to the determined order. In order to demonstrate the superiority of the proposed method, performance comparison with the existing method was performed. As a result, the search time is improved by about 3 ~ 10% compared with the existing method.

■ keyword : | Graph Data | Graph Search | Distributed Processing | Subgraph Matching | Bigdata |

* 이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-차세대정보 · 컴퓨팅기술개발사업의 지원을 받아 수행되었고(No. NRF-2017M3C4A7069432), 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행되었으며(No. 2016R1A2B3007527), 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 [No.B0101-15-0266, 실시간 대규모 영상 데이터 이해·예측을 위한 고성능 비주얼 디스커버리 플랫폼 개발]

접수일자 : 2018년 12월 20일
수정일자 : 2018년 01월 16일
심사완료일 : 2018년 01월 16일
교신저자 : 유재수, e-mail : yjs@chungbuk.ac.kr

I. 서론

최근 소셜 네트워크, 생물학 네트워크, 단백질-단백질 상호작용 네트워크 등에서 개체와 개체간의 복잡한 관계를 표현하기 위해 그래프 데이터구조가 주목 받고 있다. 그래프는 정점과 간선으로 표현되고 각 정점은 개체, 간선은 개체간의 관계를 나타낸다. 이러한 표현 방식을 바탕으로 데이터 분석 및 데이터 마이닝 등 여러 분야에서 널리 활용되고 있다[1-3]. 또한 SNS, IoT 등에 대한 활용이 증가됨에 따라 대용량의 그래프 데이터가 발생하고 있으며, 적게는 수천만에서 많게는 수십억에 이르는 정점과 간선을 갖는 네트워크가 형성되고 있다. 이로 인해 기존 시스템에서 하나의 서버에 데이터를 저장하여 처리하는 것만으로는 이러한 거대한 데이터를 처리할 수 없어졌으며, 데이터를 분할하여 서로 다른 서버에 저장하는 분산 저장 환경에서 데이터를 처리하는 시스템이 등장하게 되었다[4-6]. 대표적인 디스크 기반 프레임워크인 Apache Hadoop은 Map과 Reduce를 통해 대용량 데이터에 대한 병렬처리를 지원한다[5]. 그러나 데이터가 디스크에 저장되어 있기 때문에 질의 시 디스크에 접근하여 데이터를 읽고 쓰는 비용이 발생한다. 반면 대표적인 메모리기반 플랫폼인 Spark[7]은 질의마다 데이터를 디스크에서 읽지 않고 클러스터 메모리에 적재하여 사용하기 때문에 질의 처리 성능을 높일 수 있다.

그래프 데이터 구조에서 그래프 G는 $G=\{V, E, L\}$로 표현되고 V는 정점, E는 정점과 연결된 간선, L은 정점의 속성이다. 이러한 그래프 구조에서 질의로 주어진 그래프에 대해 일치하는 서브 그래프를 탐색하는 서브 그래프 탐색(Subgraph searching)은 중요한 의미를 갖는다. 소셜 네트워크에서 서브 그래프는 어떤 관계를 중심으로 한 사용자 간의 그룹을 의미할 수 있으며, 단백질-단백질 상호작용 네트워크 에서는 특정 단백질 구조에 대한 분석의 기초가 되기도 한다. 서브 그래프는 질의에 따라 완전 일치 서브 그래프 탐색[8]과 유사 서브 그래프 탐색[9]으로 분류된다. 완전 일치 서브 그래프 탐색은 질의 그래프와 완전하게 일치하는 서브 그래프를 찾는 질의이며, 유사 서브 그래프 탐색은 질의 그래프와 일정 부분 유사도를 갖는 서브 그래프를 찾는 질의이다. 유사 서브 그래프 탐색은 질의의 특성상 질의 그래프와 얼마나 유사한 서브 그래프를 찾을 것인가에 대한 기준을 세워야 하며, 기준에 따라 검색된 결과가 있다고 하더라도 그 그래프가 질의 그래프와 유의미한 관계가 있는지에 대한 검증 작업이 필요하다.

서브 그래프를 찾는 문제는 NP-Complete한 문제이기 때문에 연산 비용을 줄이기 위한 방법으로 filter-and-verification기법[10]이나 휴리스틱 함수에 기반한 경로 탐색 기법[11]등을 이용한 연구들이 제시되었다. filter-and-verification기법은 질의 그래프에 대응하는 필요조건을 검사하여 후보 집합의 수를 줄인다. 이렇게 가지치기된 후보 집합에 대한 검증 단계를 통해 해당 질의 그래프에 대한 결과 집합을 응답하는 기법이다. 휴리스틱 함수에 기반한 경로 탐색 기법은 목표 정점을 찾아가기 위해 최소 비용의 경로를 탐색하여 찾아가는 기법이다. 각 정점에 대한 분기에서 알고리즘은 가장 적은 비용으로 탐색할 수 있는 분기를 선택해 경로를 확장해 나간다. [12]는 이러한 경로 기반 질의 탐색기법을 제안하였다. 질의 그래프를 트리플렛 형태의 세그먼트로 분할하고, 세그먼트 단위로 탐색을 수행한다. 첫 탐색 순서인 헤드 세그먼트에서, 헤드 세그먼트 정점의 레이블과 일치하는 정점을 각 그래프에서 찾아 시작 정점으로 선정한다. 선정한 시작 정점의 이웃 정점의 레이블을 확인하여 세그먼트와 일치하는 트리플렛을 찾아 마스터 역할을 하는 서버로 반환한다. 이어서 분할된 세그먼트에 대한 탐색이 수행되고 마스터는 슬레이브 서버로부터 전송된 결과를 취합한다. 그러나 이 기법은 시작 정점을 선정하는 특별한 기준이 없고, 탐색 비용을 줄일 수 있는 비용모델을 구축하지 않기 때문에 시작 정점이 되는 헤드 세그먼트의 레이블과 일치하는 정점의 개수가 많거나 해당 정점의 차수가 클 경우 탐색비용이 증가한다는 문제점을 가지고 있다.

본 논문에서는 질의 그래프와 동일한 레이블을 갖는 서브 그래프를 찾기 위해 데이터 분산 처리 플랫폼인 Spark에서 경로 질의 기반 그래프 탐색 기법을 제안한다. 그래프의 통계정보를 바탕으로 탐색하지 않아도 되는 정점을 필터링하는 확률 값을 예측하여 불필요한 탐

색을 줄이도록 탐색 순서를 결정한다. 이러한 순서를 바탕으로 질의 그래프를 서브 질의로 분할하고 각 슬레이브 서버에서 탐색을 수행한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문과 관련된 연구들을 기술하고 3장에서는 제안하는 경로 기반 서브 그래프 탐색 기법에 대해 설명한다. 4장에서는 제안하는 기법의 우수성을 확인하기 위해 성능 평가를 보여준다. 마지막 5장에서는 논문의 결과에 대해 기술한다.

II. 관련 연구

filter-and-verification 방법은 서브 그래프 동형성 검사를 수행할 집합의 수를 줄이기 위해 제안되었다. filter-and-verification 방법은 두 가지 단계를 거치는데 단계를 거치기 전에 먼저 그래프의 구조적 특징을 고려한 인덱스를 구축해야 한다. 인덱싱되는 특징으로는 경로, 트리, 사이클 등의 일반적인 특징부터, 해시 테이블 등의 정보를 특징으로 인덱싱 하기도 한다. 첫 번째 단계인 필터링 단계에서 구축된 인덱스를 통해 질의에 대한 결과 집합(응답 집합)이 될 수 없는 그래프의 일부분들을 제외한 후보 집합을 생성한다. 필터링 단계를 거친 후보 집합은 질의 그래프의 응답에 해당하는 결과에 false positive에 해당하는 결과가 포함되어 있기 때문에 후보집합에 대한 검증이 필요하다. 때문에 두 번째 단계인 검증단계에서 기 생성된 후보 집합에 대한 서브 그래프 동형성 검사를 수행하여, 최종적으로 질의 그래프에 대한 결과집합을 생성한다.

filter-and-verification 기법 외에 데이터 그래프의 모든 정점을 탐색하지 않고 질의 그래프와 일치하는 정점만 탐색하여 질의 그래프의 경로를 구축해가는 기법이 제안되었다[13][14]. 그래프는 기본적으로 두 정점과 한 간선으로 이루어진 트리플렛이 모여 구성되어 있으며, [12]는 질의 그래프를 이러한 트리플렛 단위로 분할하여 탐색한다. 마스터의 질의 분할이 끝나면 모든 슬레이브에 분할된 질의 그래프가 브로드캐스트되고, 각 슬레이브들은 전송받은 트리플렛을 탐색한다. 탐색의 시작정점은 분할된 질의 그래프의 한 정점이 선택되면 그 정점과 연결된 간선의 방향을 따라 탐색한다. 한 트리플렛에 대한 탐색이 끝나면 각 슬레이브들은 마스터에게로 탐색 결과를 전송하고 마스터는 이를 기록한 뒤, 다음 트리플렛을 찾으라고 지시한다. 마스터 테이블에는 탐색결과들이 저장되어 탐색된 정점들이 누적되어 기록된다. 분할된 모든 질의에 대한 탐색이 끝나면 마스터의 테이블에 기록된 정점이 누적되어 탐색된 경로가 된다.

서브 그래프 탐색을 위한 여러 기법 중, 본 연구에서는 경로 기반의 탐색 기법을 차용하였다. 경로 기반의 탐색 기법은 데이터 그래프에 대해 여러 가지 특징에 대한 인덱스를 구축할 필요가 없고, 한 정점과 연결된 이웃정점만을 선택해서 탐색하여 데이터 그래프 전체에 대한 서브 그래프 동형성 검사를 수행하지 않아, 전체적인 탐색비용에 대한 이점이 있다. 그러나 [12]는 시작 정점을 선정하는 특별한 기준이 없고, 탐색 비용을 줄일 수 있는 비용모델을 구축하지 않기 때문에 시작 정점이 되는 헤드 세그먼트의 레이블과 일치하는 정점의 개수가 많거나 해당 정점의 차수가 클 경우 탐색비용이 선형적으로 증가한다는 문제점을 가지고 있다. 본 연구에서는 이러한 경로 기반 질의 탐색 방법을 기반으로 탐색 비용을 줄일 수 있는 경로로 탐색하는 방법을 제안한다. 데이터 그래프에 대해 수집된 통계정보를 바탕으로 정점 당 발생할 수 있는 차수의 확률밀도함수를 통해, 해당 정점의 차수가 등장할 확률을 고려하여 탐색하지 않아도 되는 확률을 스코어링한다. 이 값은 질의 그래프와 데이터 그래프의 차수의 차이로 인해 탐색하지 않아도 되는 정점을 필터링할 수 있는 확률이다. 필터링될 확률이 높을수록 적은 비용으로 탐색이 가능하며, 이러한 필터링 확률이 높은 정점 순으로 질의그래프를 분할하여 분할된 서브 질의그래프를 순차적으로 탐색한다.

III. 제안하는 서브 그래프 탐색 기법

1. 특징

전체 시스템 구조는 [그림 1]과 같이 마스터-슬레이브 구조로 동작한다. 마스터는 그래프에 대한 통계정보를 수집하고 질의를 분할한다. 질의는 불필요한 탐색을 줄일 수 있는 값을 계산하여 이 순서로 탐색 순서가 결정되며, 결정된 순서를 바탕으로 트리플렛 형태의 서브 질의로 분할된다. 슬레이브는 결정된 탐색순서대로 분산 저장된 데이터에 대해 탐색을 수행하고, 여러 슬레이브 서버 간 탐색된 결과를 조인하여 마스터서버로 반환한다.

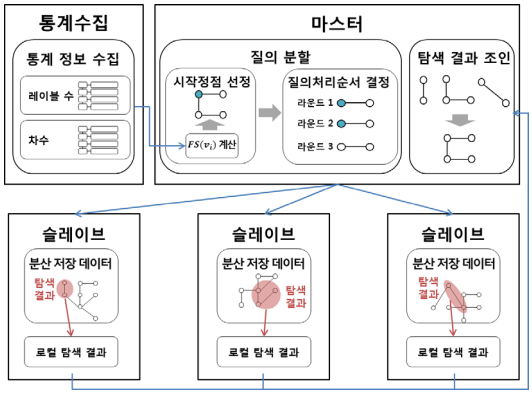


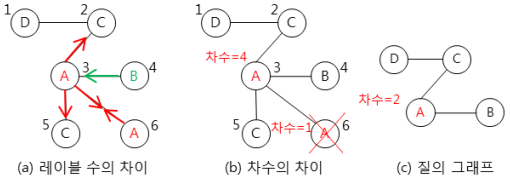
그림 1. 제안하는 기법의 전체 구조

2. 통계 정보 수집

경로를 기반으로 서브 그래프를 탐색하는 경우, 어떤 정점을 언제 탐색하느냐는 탐색에 중요한 영향을 미친다. 시작 정점의 수가 무수히 많은 경우, 무수히 많은 병렬탐색이 이루어지기 때문에 실제로 탐색에 필요한 정점의 수보다 더 많은 불필요한 탐색이 이루어질 수 있다. 또한 적절한 수의 정점을 시작으로 탐색을 이어나간다고 해도 연결된 이웃 정점을 모두 탐색해야 하기 때문에 연결된 간선의 수에 따라 탐색해야할 경우의 수도 무수히 많아진다. 때문에 제안하는 기법에서는 불필요한 탐색을 줄이고 더 적은 탐색으로 질의 그래프와 일치하는 서브 그래프를 탐색하기 위하여 정점의 수와 차수를 고려해 필터링 스코어를 계산한다.

서로 동일한 레이블을 갖는 데이터 그래프와 질의 그래프의 정점들을 $G_L(v_i)$, $Q_L(v_i)$라 하자. 만약, $Q_L(v_i)$과 일치하는 모든 $G_L(v_i)$를 시작으로 탐색한다면 레이블 L의 수가 증가함에 따라 탐색을 시작할 정점의 수도 선형적으로 증가하게 된다. [그림 2](a)와 (b)는 데이터 그래프이며 [그림 2](c)는 질의 그래프이다. [그림 2](a)에서 $G_L(v_i)$의 동일한 레이블 수가 1개인 4번 정점을 시작으로 경로를 탐색할 경우보다, 동일한 레이블 수가 2개인 3, 6번 정점을 시작으로 경로를 탐색하는 것이 더 많은 탐색이 필요하다. 때문에 제안하는 기법에서는 그래프 G의 각 정점들의 레이블 수의 통계를 수집하고, 적은 수의 레이블을 갖는 정점이 시작 정점으로 선정되도록 고려한다.

질의 그래프와 일치하는 정점을 탐색한다고 할 때, 질의 그래프 정점의 차수보다 낮은 차수를 갖는 정점은 탐색할 필요가 없다. [그림 2](b)에서 [그림 2](c)의 A레이블을 갖는 정점을 데이터그래프에서 탐색을 한다면 3, 6번 정점이 탐색될 것이다. 질의 그래프의 A레이블을 갖는 정점의 차수는 2이고 3, 6번 정점의 차수는 각각 4, 1이다. 질의의 차수는 2이기 때문에 차수가 4인 3번 정점은 탐색을 해야 하지만, 차수가 1인 6번 정점은 탐색한다고 해도 질의그래프가 될 가능성이 없다. 따라서 $Q_L(v_i)$보다 낮은 차수를 갖는 $G_L(v_i)$은 필터링할 수 있는 정점이다. 이를 기반으로 본 연구에서는 차수의 차이로 인해 탐색할 정점을 필터링 할 수 있는 스코어를 계산한다.



(a) 레이블 수의 차이 (b) 차수의 차이 (c) 질의 그래프

그림 2. 차수와 레이블의 차이로 인한 필터링

3. 필터링 스코어

현실 세계의 데이터를 그래프 데이터로 모델링하면 어떤 데이터를 사용 하느냐에 따라 정점과 그 간선의 분포가 특정한 경향을 보인다. 단순하게는 그래프의 정점과 그 정점과 연결된 간선이 비교적 고르게 분포된 정규분포의 형태를 보일 것이라 생각할 수 있지만 소셜

네트워크나 단백질-단백질 상호작용 네트워크 등은 소수의 정점에 연견된 간선이 집중되는 경향을 보이며, 이를 power-law형태의 분포라 한다. [15]에 따르면 실제로 많은 그래프 네트워크에서 각 정점과 그 정점의 차수가 power-law형태를 보임을 보였다. 본 논문에서는 이를 기반으로 그래프의 정점과 그 정점과 연결된 간선이 특정한 분포를 보일 경우, 탐색에서 제외될 확률적인 필터링 값을 계산하고 질의 그래프의 어떤 정점을 우선적으로 탐색할 것인지에 대한 탐색 순서를 결정한다. 데이터의 특성에 따라 소셜 네트워크 등에서 나타나는 power-law형 분포를 보일 경우와 여타 다른 그래프에서 데이터의 분포가 대부분 평균에 가까운 정규분포를 보일 경우를 모두 고려하여 두 가지 스코어링 방법을 제안하였다.

탐색 순서는 필터링 값을 통해 결정되며, 필터링 값은 크게 2가지 경우로 나누어 계산한다. 첫 번째 경우는 데이터 그래프의 정점과 그 차수가 비교적 평균치에 가깝게 분포된 정규분포를 따를 경우이며, 두 번째 경우는 어떤 정점에 차수가 집중되는 경향을 보이는 power-law분포를 보일 경우이다. 두 가지 경우에서 해당 분포의 확률밀도함수를 통해 해당 정점의 차수를 예측하고 예측된 값을 통해 확률적인 탐색순서를 결정한다.

(1) 정규분포의 필터링 확률

데이터 그래프의 정점과 그 차수가 고르게 분포된 정규분포를 따른다고 가정하면, G의 k차수를 갖는 정점에 대해 평균 차수가 μ, 표준편차가 σ인 확률밀도함수 $f(k)$는 식(1)과 같다.

$$f(k)=\frac{1}{\sigma\sqrt{2\pi}}exp\left(-\frac{(k-\mu)^2}{2\sigma^2}\right) \quad (1)$$

[그림 3]은 이러한 확률밀도함수를 갖는 데이터 그래프에서 차수를 갖는 정규분포의 확률밀도함수를 나타낸다. 서로 동일한 레이블을 갖는 데이터 그래프와 질의 그래프의 정점은 $G_L(v_i)$, $Q_L(v_i)$이고, 확률밀도함수에서 $G_L(v_i)$, $Q_L(v_i)$의 값은 해당 차수가 발생할 확률이다. $G_L(v_i) \sim Q_L(v_i)$의 면적은 앞서 언급한 차수의 차이에 의해 $Q_L(v_i)$의 차수보다 적은 차수를 갖는 정점 $G_L(v_i)$를 탐색하지 않고 필터링 할 수 있는 확률이다. 따라서 $G_L(v_i)$부터 $Q_L(v_j)$의 차수까지의 면적을 *filtering probability* $FP_{ND}(v_i)$라 하고, 식(2)와 같이 정의한다.

$$FP_{ND}(v_i)=\int_{G_L(v_i)}^{Q_L(v_i)} f(k)dk \quad (2)$$

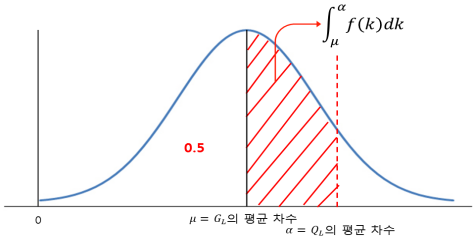


그림 3. 정규분포의 필터링 확률

(2) power-law분포의 필터링 확률

[15]에서는 일반적인 그래프 네트워크에서 정점과 그 차수 사이의 관계는 power-law분포를 보이고, power-law 분포에서 어떤 정점에 대해 그 차수가 발생할 확률이 일반적으로 $P(d) \propto d^{-\alpha}$이며, $\alpha \approx 2$임을 보였다. 이 관계는 [그림 4]에서 확인할 수 있다. 또한 [16]은 power-law분포는 식(3)과 같이 제타분포 또는 파레토 분포라 불리는 확률밀도함수로 표현할 수 있음을 보였다.

$$p(k)=\frac{k^{-\gamma}}{\zeta(\gamma)} \quad (3)$$

power-law형 분포에서 정점과 간선 사이의 관계가 2의 지수승 관계를 갖는다면, 제타함수에서 변수 s를 2로 특정할 수 있다. 그러므로 식(4), 식(5)와 같이 제타함수는 특정한 값으로 수렴하게 된다. 그러므로 power-law 분포의 확률밀수함수는 식(6)과 같이 정의할 수 있다.

$$\zeta(\gamma)=\sum_{n=1}^{\infty}\frac{1}{n^{\gamma}} \quad (4)$$

$$\zeta(2)=1+\frac{1}{2^2}+\frac{1}{3^2}+\cdots=\frac{\pi^2}{6} \quad (5)$$

$$f(x)=\frac{6}{(\pi k)^2} \quad (6)$$

[그림 4]는 이를 바탕으로 전체 그래프 데이터에서 차수 k를 갖는 파레토 분포의 확률밀도함수를 나타낸다. 정규분포와 마찬가지로, $G_L(v_i)$부터 $Q_L(v_j)$의 차수

까지의 면적을 $FP_{PD}(v_i)$라 하고, 식(7)과 같이 정의한다.

$$FP_{PD}(v_i) = \int_{G_L(v_j)}^{Q_L(v_j)} f(k)dk \quad (7)$$

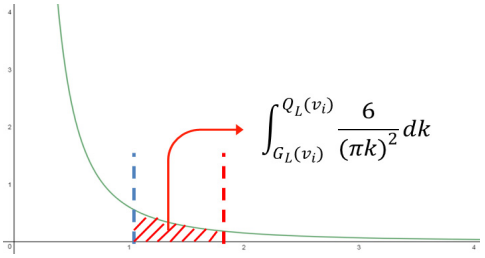


그림 4. power-law 분포의 필터링 확률

(3) 필터링 스코어 계산

질의 그래프가 입력되면 수집한 통계 정보를 바탕으로 모든 Q의 정점에 대해 시나리오 1(그래프의 정점과 차수가 확률적인 분포를 따를 경우)과 시나리오 2(차수가 발생할 확률이 아닌 실제 그래프의 평균차수만으로 계산)에 대해 $filtering\ score\ FS(v_i)$를 각각 식(8), 식(9)와 같이 정의한다.

$$FS(v_i) = \left(\frac{G_L}{G_{all}} \times FP(v_i) \right) \quad (8)$$

$$FS(v_i) = \left(\frac{G_L}{G_{all}} \times avgDegree\ of\ Q_L(v_i) \right) \quad (9)$$

4. 질의그래프 분할

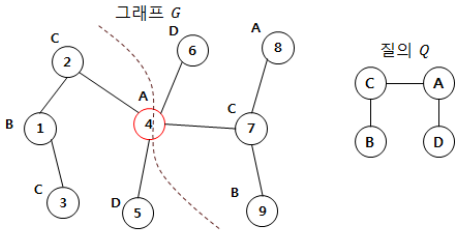
질의 그래프의 정점 중 필터링 스코어가 가장 높은 값을 갖는 정점을 시작정점으로 선정하며, 시작 정점과 연결된 이웃 정점 중 다음으로 높은 값을 갖는 정점으로 이루어진 트리플렛으로 질의 그래프를 분할한다. 이 작업은 시작 정점과 연결된 모든 이웃정점에 대해 수행되며, 이웃정점과의 분할이 끝나면, 시작 정점의 이웃정점과 연결된 정점 중 다음으로 높은 값을 갖는 정점에서 같은 작업이 이루어진다.

시작 정점 및 탐색 순서 결정이 끝나면 탐색을 위한 질의처리 라운드를 등록한다. 라운드는 각 슬레이브에서 병렬로 탐색할 순서이며 이전에 결정한 탐색 순대로 등록된다. 필터링 스코어가 가장 높은 시작 정점은 라운드 0으로 등록된다. 시작 정점과 연결된 이웃 정점 중 다음으로 높은 스코어를 갖는 정점을 다음 라운드로 등록한다. 시작 정점과 연결된 모든 이웃 정점에 대해 라운드 등록이 끝나면 시작 정점의 이웃 정점에 대해 같은 작업을 수행한다. 이로 인해 질의 그래프의 모든 정점이 필터링 스코어가 높은 순으로 분할되어 라운드로 등록된다.

5. 서브 그래프 탐색 수행

필터링 스코어를 통해 질의 그래프가 분할되고 탐색 순서인 질의처리 라운드가 등록되었다. 마스터는 결정한 순서대로 각 슬레이브에게 탐색을 지시한다. 탐색을 지시받은 슬레이브는 자신의 파티션에서 탐색을 수행하고 결과를 결과 테이블에 기록한다. 모든 라운드의 탐색이 끝나면 슬레이브에 결과 테이블에 기록된 정점의 리스트가 최초 질의 그래프의 서브 그래프가 된다.

[그림 5]의 탐색 결과는 이러한 라운드의 수행결과를 보여준다. 질의처리 라운드에 따라 라운드 0이 수행되면 각 슬레이브에서는 시작 정점으로 선정된 레이블을 갖는 모든 정점을 탐색한다. 그 후 라운드 1이 수행되어 그 시작 정점의 레이블을 갖는 정점과 연결된 라운드 1의 레이블을 갖는 이웃 정점을 탐색하여 기록한다. 다음 라운드는 탐색 결과에 기록된 정점의 ID를 통해 탐색된다. 이로 인해 이전 라운드의 수행으로 탐색된 정점의 이웃에 대해서만 탐색이 수행되며, 이전 탐색에서 불필요한 정점에서부터 다시 탐색하는 일이 없기 때문에 탐색 비용을 효율적으로 줄일 수 있다. [그림 5]의 라운드 1의 수행 결과로 3번 정점과 연결된 1번 정점이 탐색되었지만 라운드 2의 수행에서 3-1과 연결된 A레이블을 갖는 정점은 데이터 그래프에 존재하지 않는다. 따라서 라운드 2의 수행 결과에선 (3, 1)이 제외된다. 라운드 3의 수행에서 (3, 1) 이 제거되었기 때문에 1번 정점과 연결된 레이블 D를 갖는 정점은 탐색할 필요가 없다. 이전의 라운드 수행결과에 대해서만 다음 라운드가 수행되는 방식을 통해 결과로 기록되는 테이블의 크기는 점점 줄어들고 불필요한 탐색을 수행하지 않는 이점이 있다.



round 0	C
round 1	C-B
round 2	C-A
round 3	A-D

	탐색 결과
round 0	(2) (3) (7)
round 1	(2,1) (3,1) (7, 9)
round 2	(2,1,4) (3,1) (7, 9, 8) (7, 9, 4)
round 3	(2, 1, 4, 5) (7, 9, 8) (7, 9, 4, 6) (7, 9, 4, 5)

그림 5. 질의 탐색 순서와 그 결과

Ⅳ. 성능 평가

본 연구에서는 제안하는 기법의 우수성을 보이기 위해 기존 기법[12]와의 성능비교와 그래프의 분포에 따른 시나리오에 대한 자체비교를 수행하였다. [12]는 경로 탐색에 대한 비용예측이 없이 분할된 질의 그래프에 대해 탐색을 수행하였다. [표 1]은 성능평가 환경을 보여준다, 성능 평가 환경은 Intel(R) Core(TM) i7-6700 CPU 3.40GHz 프로세서, 30G메모리, 3개의 클러스터 환경으로 구성되었고, Spark의 GraphX로 Scala를 통해 구현하였다. 실험 평가를 위해 실제 데이터 집합과 가상의 데이터 집합을 사용하였다. Standford에서 제공하는 데이터와 Graph Generator 소프트웨어인 GTgraph를 사용하였다. skitter는 인터넷 토폴로지 데이터로, 약 170만개의 정점과 1100만개의 간선으로 이루어져 있다. GTgraph는 skitter데이터와 동일한 정점과 간선의 수를 가지며, 임의로 생성한 데이터이다. 데이터의 분포는 정점과 그 차수가 대체적으로 균일하게 분포되도록 생성하였다. 성능평가로는 그래프의 분포 시나리오에 따른 자체 평가, 질의 그래프의 구조적 차이에 따른 탐색 시간을 비교하였다.

표 1. 성능평가 환경

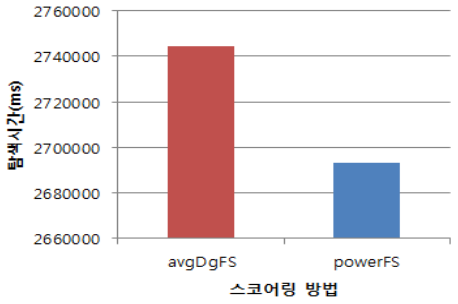
구 분	내 용
프로세서	Intel(R)Core(TM)i7-6700CPU 3.40GHz
메모리	30G
클러스터	3

제안하는 기법은 시나리오에 따라 데이터의 분포를 고려한 스코어링 방법과 평균차수만을 고려한 스코어링 방법으로 나뉜다. 또한 분포를 고려한 스코어링 방법은 데이터의 분포에 따라 다른 확률밀도함수를 사용하기 때문에 각 방법에 대한 자체 성능평가를 수행하였다. 시나리오 1은 분포에 따라 정규분포와 power-law 분포로 나뉘며, 각 분포에서 질의 그래프의 차수가 발생할 확률을 통해 스코어링한다. 정규분포는 normalFS로, power-law분포는 powerFS로 표기하였다. 시나리오 2는 데이터 그래프에서의 평균 차수만을 통해 스코어링하는 방법으로, avgDgFS으로 표기하였다. 실험 데이터 집합에서 정규분포형 데이터는 GTgraph를 사용해 생성한 랜덤 데이터를 사용하였고, power-law분포형 데이터는 실제 데이터 집합을 사용하였다.

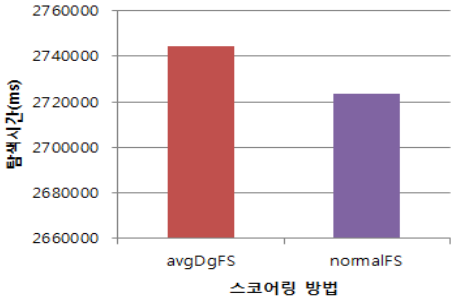
[그림 6]은 시나리오별 탐색 시간을 나타낸 것이다. 데이터 집합에 관계없이 시나리오 1의 확률밀도함수의 필터링 확률과 레이블 수로 스코어를 계산하는 방법이 시나리오 2의 평균차수와 레이블 수로 스코어를 계산하는 방법보다 우세한 성능을 보였다. 이는 그래프의 정점과 차수가 특정한 분포를 따르고 있기 때문으로 보인다. 평균값은 기준이 되는 값만 제시할 뿐, 기준을 벗어난 값에 대해서는 보정을 해주지 않기 때문이다. 반면 시나리오 1의 확률밀도함수를 이용한 계산은 실제 데이터의 통계정보를 이용해 더 적은 탐색으로 질의를 수행할 수 있는 경로를 제시하기 때문에 대부분의 경우에서 우수한 성능을 보였다.

제안하는 기법과 기존 기법과의 비교평가를 위해 질의 그래프의 구조적 유형에 따라 탐색시간의 차이를 평가하였다. 질의 그래프가 연결된 구조적 특성은 탐색 성능에 유의미한 영향을 미친다. [그림 7]의 q1~q4 질의는 구조적으로 큰 차이를 보인다. q1 질의는 한 정점과 다른 정점이 단순하게 연결된 구조이고 q2는 한 정점에

집중된 구조, q3은 두 정점에 집중된 구조이며, 마지막으로 q4는 소수의 정점에 집중된 구조와 단순한 구조가 혼합된 구조의 질의이다.



(a) 실제 데이터



(b) 임의 생성 데이터

그림 6. 스코어링 방법에 따른 탐색 시간

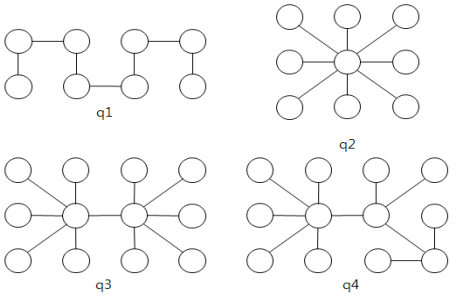


그림 7. 구조에 따른 질의 유형

랜덤 데이터에서 [그림 7]의 질의를 수행시켜 보았다. [그림 8]의 결과를 보면, q1 질의는 성능의 차이가 거의 보이지 않음을 확인할 수 있다. 구조적으로 단순한 질의이기 때문에 데이터가 랜덤으로 생성된 환경에서 성능의 차이가 크지 않음을 알 수 있다. q2~q4질의는 q1 질의에 비해 탐색시간이 긴 경향을 보였다. 한 정점에서 연결된 정점이 많기 때문에 탐색시간이 더 소요된다. 그러나 질의 간 큰 차이를 확인할 수는 없었는데, 데이터의 랜덤성으로 인한 것으로 보인다. 질의 간 탐색 시간의 차이는 크지 않지만 탐색 방법에서는 제안하는 기법의 normalFS와 avgDgFS의 성능이 우수함을 확인할 수 있다.

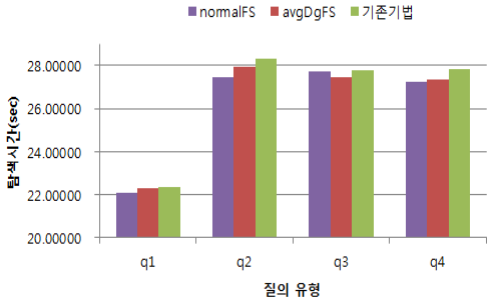


그림 8. 임의 생성 데이터에서 질의에 따른 탐색 시간

[그림 9]는 실제 데이터에서 질의 유형에 따른 탐색 시간을 보여준다. 랜덤 데이터에서는 질의 유형에 따라 큰 성능의 차이는 볼 수 없었던 것처럼, [그림 7]의 q1과 같이 단순한 형태의 질의는 실제 데이터에서의 성능 평가 또한 큰 차이를 보이지 않았다. 한 정점에서 연결된 정점이 하나밖에 없기 때문에 비용 예측에 대한 이점이 거의 없기 때문이다. 그러나 다른 질의의 경우 유의미한 성능 차이를 확인할 수 있었다. q2 의 경우, 한 정점에 극단적으로 집중되는 구조적 특성을 보인다. 이 질의의 경우는 스코어 계산 방법에 의해 집중되는 정점이 시작 정점으로 선정되는 경우가 많았으며, 이로 인해 한 정점에서만 이웃정점에 대해 탐색하게 되었다. q3의 경우 제안하는 기법의 성능이 더 좋은 것으로 나타났지만 q4의 차이에 비하면 약간 못미치는 결과를 보이며, q2와 q4의 중간 정도의 성능을 보여주었다. q4는 기존기법과의 비교에서 가장 눈에 띄는 성능을 보였다. 제안하는 기법에서는 무작위로 탐색하는 것이 아닌, 질의 그래프에 대해 각 정점이 그래프에서 발생할 확률을

예측하여 적은 탐색 횟수로도 질의 그래프를 만들 수 있는 경로의 탐색방법을 예측하기 때문에 q4같은 혼합된 구조의 질의에서 강점을 보였다.

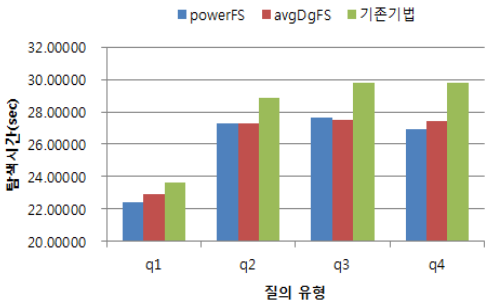


그림 9. 실제 데이터에서 질의에 따른 탐색 시간

V. 결론 및 향후 연구

본 논문에서는 분산 환경에서 서브 그래프 탐색 질의를 처리하기 위한 경로 기반 그래프 탐색 기법을 제안하였다. 데이터 그래프의 정점과 그 정점의 간선 사이의 분포를 통해 질의 그래프 탐색 시 불필요한 탐색을 제거할 수 있는 필터링 값을 계산한다. 계산된 필터링 값을 통해 가장 많은 필터링이 될 수 있는 정점을 탐색의 시작 정점으로 선정하고, 필터링 값에 따라 질의 그래프를 작은 단위의 서브 질의로 분할하여 탐색을 수행한다. 이로 인해 기존 기법보다 빠르게 서브 그래프에 대한 탐색이 가능하다. 제안한 기법은 소셜 네트워크, 단백질-단백질 상호작용 네트워크 또는 여타 정규분포를 따르는 그래프에서 그래프의 분포적 특성에 따라 스코어링 방법을 달리하여 적용하여 여러 데이터 그래프에 적용할 수 있다. 이를 통해 다양한 분포적 특성을 갖는 데이터에 대해 서브 그래프 탐색 질의를 효율적으로 수행할 수 있다. 성능평가 결과, 두 가지 데이터에서 모두 기존 기법보다 탐색 시간이 약 3~10% 향상됨을 확인할 수 있었다. 본 논문에서는 스코어링을 위해 단순히 정점의 수와 정점의 차수를 사용하였지만, 실제로 필터링에 영향을 미칠 수 있는 요소들은 간선의 특성이나 사이클 등이 다양하게 존재한다. 그러므로 향후에는 그래프의 다양한 요소를 고려하여 필터링 성능 향상을 위해 연구할 예정이다.

참 고 문 헌

[1] A. Cuzzocrea, F. Furfaro, G. M. Mazzeo, and D. Saccà, “A grid framework for approximate aggregate query answering on summarized sensor network readings,” Proc. OTM Workshops, pp.144-153, 2004.

[2] A. Fariha, C. F. Ahmed, C. K. Leung, S. M. Abdullah, and L. Cao, “Mining frequent patterns from human interactions in meetings using directed acyclic graphs,” Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, pp.38-49, 2013.

[3] F. Jiang and C. K. Leung, “Mining interesting “following” patterns from social networks,” Proc. International Conference on Data Warehousing and Knowledge Discovery, pp.308-319, 2014.

[4] F. Towards, “Towards a Scalable HDFS Architecture,” Proc. International Conference on Collaboration Technologies and Systems, pp.155-161, 2013.

[5] J. Dörre, S. Apel, and C. Lengauer, “Modeling and optimizing MapReduce programs,” Concurrency and Computation: Practice and Experience, Vol.27, No.7, pp.1734-1766, 2015.

[6] A. Alam and J. Ahmed, “Hadoop Architecture and Its Issues,” Proc. International Conference on Computational Science and Computational Intelligence, pp.288-291, 2014.

[7] X. Liao, Z. Gao, W. Ji, and Y. Wang, “An enforcement of real time scheduling in Spark Streaming,” Proc. International Green and Sustainable Computing Conference, pp.1-6,

2015.

[8] N. Talukder, and M. J. Zaki, “A distributed approach for graph mining in massive networks,” Data Mining and Knowledge Discovery, Vol.30, No.5, pp.1024-1052, 2016.

[9] Y, Tian, R. C. McEachin, C. Santos, D. J. States, and J. M. Patel, “SAGA: a subgraph matching tool for biological graphs,” Bioinformatics, Vol.23, No.2, pp.232-239, 2007.

[10] J. Cheng, Y. Ke, and W. Ng, “Efficient query processing on graph databases,” ACM Transactions on Database Systems, Vol.34, No.1, pp.1-48, 2009.

[11] S. Khuller, B. Raghavachari, and N. E. Young, “Balancing minimum spanning trees and shortest-path trees,” Algorithmica, Vol.14, No.4, pp.305-321, 1995.

[12] J. Balaji and R. Sunderraman, “Distributed Graph Path Queries Using Spark,” Proc. COMPSAC Workshops, pp.326-331, 2016.

[13] X. Zhang and L. Chen, “Distance-aware selective online query processing over large distributed graphs,” Data Science and Engineering, Vol.2, No.1, pp.2-21, 2017.

[14] N. Jing, Y. Huang, and E. A. Rundensteiner, “Hierarchical encoded path views for path query processing: An optimal model and its performance evaluation,” IEEE Transactions on Knowledge and Data Engineering, Vol.10, No.3, pp.409-432, 1998.

[15] M. Faloutsos, P. Faloutsos, and C. Faloutsos, “On power-law relationships of the internet topology,” ACM SIGCOMM 1999 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp.251-262, 1999.

[16] M. L. Goldstein, S. A. Morris, and G. G. Yen, “Problems with fitting to the power-law distribution,” The European Physical Journal B-Condensed Matter and Complex Systems, Vol.41, No.2, pp.255-258, 2004.

저 자 소 개

김 민 영(Minyoung Kim) 준회원

- 2017년 2월 : 충북대학교 정보통신공학과(공학사)
- 2017년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 그래프데이터, 서브그래프 매칭, 분산 컴퓨팅, 빅데이터

최 도 진(Dojin Choi) 정회원

- 2014년 2월 : 한국교통대학교 컴퓨터공학과(공학사)
- 2016년 2월 : 한국교통대학교 컴퓨터공학과(공학석사)
- 2016년 3월 ~ 현재 : 충북대학교 정보통신공학과 박사과정

<관심분야> : 연속 질의 처리, 위치 기반 서비스, 그래프 스트림 처리, 빅데이터 등

박 재 열(Jaeyeol Park) 정회원

- 2014년 2월 : 충북대학교 정보통신공학과(공학사)
- 2016년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2016년 3월 ~ 현재 : 충북대학교 정보통신공학과(박사과정)

<관심분야> : 데이터베이스 시스템, RDF, 실체화 뷰, 빅데이터 등

김 연 동(Yeondong Kim) 준회원

- 2018년 2월 : 충북대학교 정보통신공학과(공학사)
- 2018년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 소셜 사물 인터넷, 데이터베이스 시스템, 빅데이터, 그래프 분석

임 종 태(Jongtae Lim) 정회원

- 2009년 2월 : 충북대학교 정보통신공학과(공학사)
- 2011년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2015년 8월 : 충북대학교 정보통신공학과(공학박사)
- 2015년 9월 ~ 현재 : 충북대학교 정보통신공학과 Postdoc

<관심분야> : 소셜 네트워크, 빅데이터, 텍스트 마이닝

복 경 수(Kyoungsoo Bok) 종신회원

- 2009년 2월 : 충북대학교 수학과(이학사)
- 2000년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2005년 8월 : 충북대학교 정보통신공학과(공학박사)
- 2005년 3월 ~ 2008년 2월 : 한국과학기술원 정보전자연구소 Postdoc
- 2008년 3월 ~ 2011년 2월 : 가인정보기술 연구소 연구원
- 2011년 3월 ~ 현재 : 충북대학교 전자정보대학 정보통신공학부 초빙교수

<관심분야> : 데이터베이스 시스템, 이동 객체 데이터베이스, 이동 P2P 네트워크, 소셜 네트워크 서비스, 빅데이터 등

최 한 석(Han Suk Choi) 종신회원

- 1980년 2월 : 전남대학교 수학교육과(이학사)
- 1986년 8월 : Western Illinois University, Department of Computer Science(이학석사)
- 1997년 2월 : 전북대학교 대학원 컴퓨터과학과(이학박사)
- 1989년 3월 ~ 현재 : 목포대학교 컴퓨터공학과 교수

<관심분야> : 빅데이터 분석, 딥러닝, 지능형 컴퓨터 응용 시스템 등

유 재 수(Jaesoo Yoo) 종신회원

- 1989년 2월 : 전북대학교 컴퓨터공학과(공학사)
- 1991년 2월 : 한국과학기술원 전산학과(공학석사)
- 1995년 2월 : 한국과학기술원 전산학과(공학박사)
- 1995년 2월 ~ 1996년 8월 : 목포대학교 전산통계학과 전임강사
- 1996년 8월 ~ 현재 : 충북대학교 전자정보대학 정교수

<관심분야> : 데이터베이스 시스템, 멀티미디어 데이터베이스, 센서 네트워크, 바이오인포매틱스, 빅데이터 등