http://dx.doi.org/10.5392/JKCA.2015.15.12.024

RDF 질의 처리 성능 향상을 위한 실체 뷰 선택 기법

Materialized View Selection Scheme for enhancing RDF Query Performance

박재열*, 윤상원*, 최기태*, 임종태*, 이병엽**, 신재룡***, 복경수*, 유재수*
충북대학교 정보통신공학부*, 배재대학교 전자상거래학과**, 광주보건대학교 보건행정과***

Jaeyeol Park(yeols@chungbuk.ac.kr)*, Sangwon Yoon(ysone88@chungbuk.ac.kr)*,
Kitae Choi(choikitae@chungbuk.ac.kr)*, Jongtae Lim(jtlim@chungbuk.ac.kr)*,
Byoungyup Lee(bylee@pcu.ac.kr)**, Jaeryong Shin(sjr@ghc.ac.kr)***,
Kyoungsoo Bok(ksbok@chungbuk.ac.kr)*, Jaesoo Yoo(yjs@chungbuk.ac.kr)*

요약

시맨틱 웹의 발전과 함께 RDF 데이터에 대한 사용이 증가되고 있다. RDF 데이터는 트리플로 구성되어 있으며 질의 처리 시 높은 조인 비용이 요구된다. 실체 뷰는 질의 처리 비용을 감소시키는 기법으로 알려져 있다. 실체 뷰는 질의 처리의 결과 또는 중간 결과를 저장 공간 내부에 물리적으로 저장하여 질의 처리 시 전체 데이터베이스의 접근이 아닌 실체 뷰의 접근으로 질의를 처리한다. 본 논문에서는 이를 해결하기 위해 의사 결정 트리를 사용하여 실체 뷰를 선택한다. 제안하는 기법은 의사 결정 트리를 통해 질의 처리 시간뿐만 아니라 실체 뷰의 크기 및 유지비용을 고려한다. 성능평가를 통해 제안하는 기법이 기존 기법에 비해 제한된 저장 공간에서의 실체 뷰는 증가하였고 동일 개수의 실체 뷰의 유지비용은 감소함을 보인다.

■ 중심어 : | RDF | 의사 결정 트리 | 실체 뷰 | 단축 경로 |

Abstract

With the development of the semantic web, a large amount of data being produced nowadays is in RDF format. RDF is represented by a triple. An RDF database consisting of triples requires the high cost of join query processing. Materialized view is known as a scheme to reduce the query processing cost by accessing materialized views without accessing the database. It is physically stored the results or the intermediate results of the query processing in a storage area. In this paper, we propose a materialized view selection scheme by using decision tree to solve such a problem. The decision tree considers the size and maintenance costs of the materialized view as well as the profit of query response times. It is shown through performance evaluation that the proposed scheme increases the number of materialized views in the limited storage space and decreases the update rates of the materialized views.

■ keyword : | RDF | Decision Tree | Materialized View | Shortcut |

* 이 논문은 미래창조과학부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업(IITP-2015-H8501-15-1013), 교육부와 한국연구재단의 지역혁신인력양성사업(No.2013H1B8A2032298), 2013년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2013R1A2A2A01015710), 2014년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No.2014R1A1A2055778).

접수일자 : 2015년 06월 25일 심사완료일 : 2015년 07월 24일
수정일자 : 2015년 07월 24일 교신저자 : 유재수, e-mail : yjs@chungbuk.ac.kr

I. 서 론

90년대 초 WWW(World Wide Web)의 제안으로 다양한 웹 서비스들이 개발되고 있다. 일반적인 웹은 HTML를 기반으로 하이퍼링크를 통해 다양한 문서들을 연결하고 태그를 이용하여 다양한 시각적인 표현 방식으로 제공한다. 그러나 일반적인 웹은 컴퓨터가 문서의 내용을 자동적으로 이해하고 처리할 수 없다는 문제점이 있다. 다양한 웹 문서가 인터넷을 통해 생성됨에 따라 컴퓨터가 웹 문서 내용을 자동적으로 인식하고 이해할 수 있는 시맨틱 웹이 제안되었다[1][2].

시맨틱 웹상의 자원을 표현하기 위해 W3C에서는 RDF를 제안하였다. RDF는 웹 자원의 정보를 표현하기 위한 표준 언어로 웹 자원에 대한 메타데이터를 표현한다. RDF는 주어(subject), 술어(predicate), 목적어(object)로 구성된 트리플 구조로 표현된다. RDF 트리플은 주어와 목적어를 노드(node)로 표현하고 술어(Predicate)를 간선(edge)으로 연결하여 그래프로 표현할 수 있다. RDF를 통해 웹 자원간의 의미적 연관성을 표현함으로써 보다 지능적인 정보 검색은 물론 자동화된 다양한 웹 서비스를 제공할 수 있게 되었다[1][2].

일반적인 RDF 데이터베이스는 주어, 술어, 목적어로 구성된 단일 테이블로 저장된다[3-6]. 이 RDF 데이터베이스는 메타데이터의 속성을 정의함으로써 풍부한 데이터 표현을 할 수 있지만 질의 처리 시 많은 조인 비용이 소모된다[4][5][7][8]. 조인 연산 중 가장 큰 비용을 차지하는 부분은 셀프-조인 연산(self-join)이다. RDF 트리플 형식의 데이터가 증가하여 테이블의 크기가 수없이 커진다면 셀프-조인 시 많은 비용이 소요된다. 최근 RDF 데이터에 대한 조인 비용 감소를 위한 많은 연구들이 진행되었다. RDF 질의 처리 비용을 감소시키기 위해 트리플 데이터를 분산하여 저장하는 방법[9], 트리플 패턴에 대해 색인 구조[3][10] 등에 대한 연구들이 진행되었다. 이러한 기법들은 질의 처리 시 트리플 데이터의 효율적인 접근하기 위한 기법으로 동일 질의 또는 유사 질의 처리에도 반복적인 질의를 수행해야 한다.

최근 기존 질의 처리 결과 및 중간 결과를 저장하고 이를 질의 처리에 활용하기 위한 실체 뷰 관리 기법에 대한 연구들이 진행되고 있다[4][5]. 실체 뷰는 오라클 데이터베이스에서 처음 도입된 것으로 질의의 결과를 저장하는 데이터베이스 객체이다. 이러한 실체 뷰는 테이블의 행이나 열의 집합 또는 조인(join), 집계 처리(aggregte operation) 결과일 수도 있다. 일반 뷰는 단지 질의 정보만이 저장되어 관리되며 사용 시 질의를 다시 처리하여 사용자에게 제공한다. 실체 뷰는 질의 처리의 결과를 저장 공간 내부에 물리적으로 저장된다. 질의 처리에 사용 시 전체 RDF 데이터베이스의 접근이 아닌 실체 뷰의 접근으로 질의 처리를 할 수 있기에 일반 뷰보다 빠르다. 하지만 실체 뷰는 하나 또는 그 이상의 테이블의 복사본이라고 볼 수 있기 때문에 데이터 저장소의 공간도 실제적으로 많이 차지하게 된다. 그렇기 때문에 빈번히 사용 되는 특정 질의나, 질의 처리 비용이 고가인 질의 등의 결과를 실체 뷰로 저장해야 한다. [4]에서는 후보 실체 뷰를 선택 시 질의 빈도 및 후보 실체 뷰의 이익을 고려하여 모델링하였다. 실제 환경에서 한정되어 있는 저장 공간 안에 후보 실체 뷰를 실체화 시키는 것을 목적으로 하고 있기에 [5]에서는 질의 빈도 및 단축 경로의 이익뿐만이 아니라 단축 경로의 갱신 비용과 크기도 같이 고려하여 모델링 하였다. [5]는 [4]의 모델에 후보 실체 뷰의 크기 및 갱신율을 단순 뺄셈으로 확장했기 때문에 후보 실체 뷰 선택 시 역전 상황이 발생한다. 본 논문에서는 역전 상황을 방지하며 효율적인 후보 실체 뷰의 선택을 위하여 의사 결정 트리를 사용한다. 의사 결정 트리는 후보 실체 뷰의 이득, 실체 뷰 크기, 그리고 갱신율을 고려하여 구축한다. 구축된 의사 결정 트리는 후보 실체 뷰를 3가지 그룹으로 구분되는 결과를 만들어 낸다. 3가지 그룹으로 질의 처리 시 효율적인 후보 실체 뷰를 선택하여 저장 및 관리를 한다.

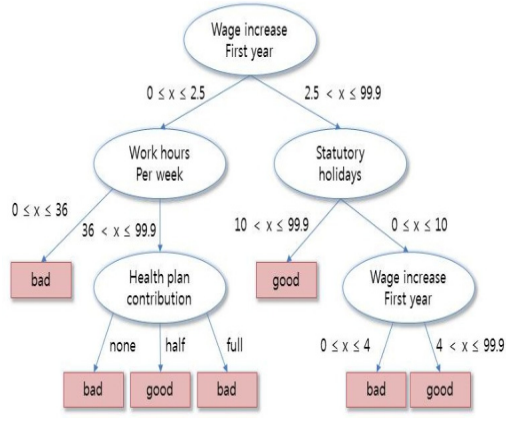
본 논문의 구성은 다음과 같다. II장에서는 관련 연구로써 의사 결정 트리와 기존 후보 실체 뷰 선택 기법을 설명한다. III장에서는 제안하는 의사 결정 트리를 이용한 RDF 기반의 후보 실체 뷰 선택 기법을 기술한다. IV장에서는 기존 기법과 제안하는 기법의 성능 평가를 통하여 제안하는 기법의 우수성을 나타낸 뒤, V장에서는 본 논문의 결론과 향후 연구 방향을 제시한다.

II. 관련 연구

1. 의사 결정 트리

의사 결정 트리는 데이터 마이닝에서 일반적으로 사용되는 방법론으로, 몇몇 입력 변수를 바탕으로 목표 변수의 값을 예측하는 모델을 생성하는 것을 목표로 한다. 의사 결정 트리는 그래프의 일종으로 단말 노드를 제외한 중간 노드(속성)는 분할 기준이 되는 속성을 의미하며 단말 노드는 상위 노드들에 의해 분류된 결과 클래스들의 집합을 의미한다. 상위 노드와 하위 노드를 잇는 경로는 상위 노드의 분할 기준이 된다. 데이터 집합이 주어졌을 때 의사 결정트리의 중간 노드의 경로에 따라 부분 집합으로 분할되며 최종적으로 단말 노드에서는 유사한 결과 클래스 값을 가지는 부분 집합으로 구성된다[11][12].

[그림 1]의 의사 결정 트리는 검사대상의 현재 건강 상태를 5가지 중간 노드로 분류하는 트리이다. 각각의 노드들은 조건들을 가지고 있다. 조건에 따라 다른 하위 노드로 구분 되어 최종 값(good, bad)으로 분류됨을 볼 수 있다.

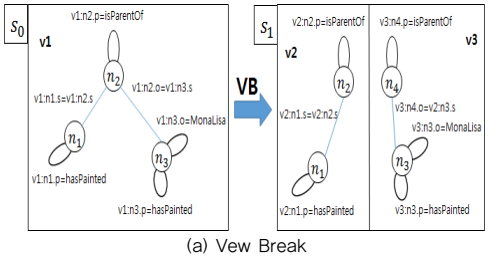


Wage increase First year - 입사 첫해의 임금 상승률
Work hours Per week - 주당 근무 시간
Statutory holidays - 법정 공휴일 수
Health plan contribution - 건강보험

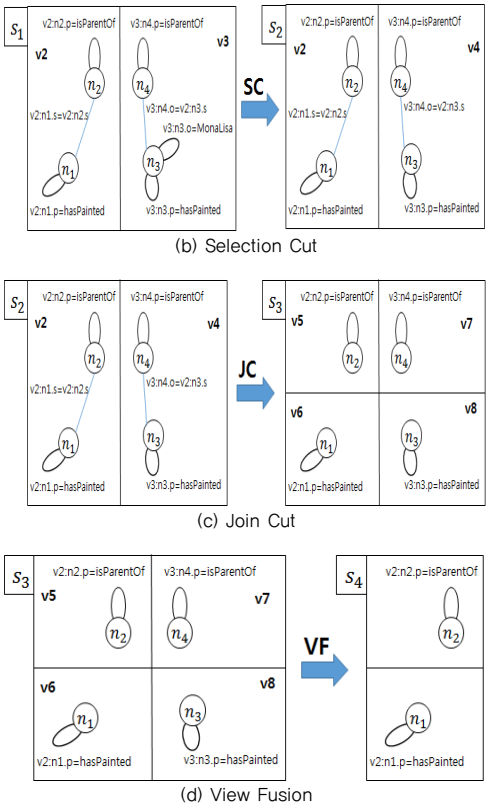
그림 1. 현재 건강 상태 분류 의사 결정 트리

2. 기존 후보 실체 뷰 선택 기법

RDF 트리플 데이터베이스는 단일 테이블로 구성되어 있어 질의 처리 시 조인 비용이 크다. 실체 뷰는 질의 처리 시 조인 비용 및 효율을 높여주는 방법으로 알려져 있다. 하지만 실체 뷰는 제한된 저장 공간의 일부분을 차지하기 때문에 효율이 높은 실체 뷰가 선택되어 구축되어야 한다. 실체 뷰의 문제를 해결하기 위해 [13]에서는 질의 결과(실체 뷰)를 [그림 2]와 같이 4단계의 변환 과정을 거쳐 압축하여 관리한다. [그림 2]는 질의 q(X,Z):-t(X, hasPainted, Z), t(X, isParentOf, Y), t(Y, hasPainted, MonaLisa)에 해당하는 실체 뷰의 4단계의 변환 과정을 보여준다. 첫 번째 변환 View Break는 노드와 노드 사이의 노드를 분할하는 단계이다. 두 번째 변환은 Selection Cut으로서 노드의 인스턴스 값을 삭제하는 단계이다. 세 번째 변환 Join Cut 각 뷰 안의 노드들 간의 조인관계를 잘라준다. 마지막으로 View Fusion은 위의 세 단계와 달리 전체의 뷰의 개수를 줄여주는 단계로써 서로 같은 뷰들을 융합해준다. 각각의 변환 단계들은 그에 맞게 질의를 재작성하여 관리된다. 4개의 변환 단계들은 휴리스틱 알고리즘을 통하여 모든 가능한 경우의 수를 계산하게 되며 그중 가장 효율이 좋은 단계를 선택하여 실체화 한다. 실체화 뷰는 초기의 상태보다 뷰의 개수가 증가하게 되지만 크기가 줄어 더 많은 질의 결과를 실체화 할 수 있으며 대용량의 단일 데이터베이스의 접근하여 질의 처리하는 비용보다 작은 여러 테이블의 조인 비용들이 훨씬 저렴하다는 관점에서 적용된 기법이다.



(a) Vew Break



(b) Selection Cut

(c) Join Cut

(d) View Fusion

그림 2. 실체 뷰 4단계 변환 과정

[13]에서는 이미 구축되어진 실체 뷰 또는 구축 되어진 실체 뷰의 크기를 감소시켜 더 많은 실체 뷰의 구축을 할 수 있도록 기법을 제안하였다. 본 논문에서는 실체 뷰의 크기의 감소 목적이 아닌 [4][5]와 같이 효율이 좋은 실체 뷰의 선택을 목표로 한다.

[4][5]에서는 조인 비용을 줄이기 위하여 단축경로(shortcut) 선택 기법을 제안하였다. 본 논문에서는 단축 경로를 후보 실체 뷰로 정의한다. 후보 실체 뷰는 단일 테이블로 이루어진 데이터베이스에서 효율적인 질의 처리를 위하여 질의의 결과 또는 질의의 중간 결과를 실체 뷰로 만들어 사용하는 기법이다. 저장 공간의 제한이 있기 때문에 모든 후보 실체 뷰를 구축할 수가 없다. [4]에서는 질의의 빈도와 질의 처리에 있어 후보 실체 뷰의 이익을 고려하는 이득 모델을 사용하여 후보 실체 뷰 선택의 방법을 제시하였다. 이득 모델을 통하여 모든 후보 실체 뷰의 이득 값을 계산한다. 저장 공간의 크기에 따라 이득이 가장 높은 후보 실체 뷰 부터 실체화하여 관리한다. [4]에서는 후보 실체 뷰의 크기 및 유지비용을 고려하지 않는다. 실제 환경의 저장 공간은 한정되어 있기에 [5]에서는 [4]의 이득 모델에서 후보 실체 뷰의 크기와 유지비용을 고려하는 확장된 이득 모델을 제안하였다.

[5]의 문제점은 역전 상황의 발생이다. 예를 들어 후보 실체 뷰 a는 여러 질의 처리에 사용가능 하지만 크기 및 유지비용의 크다. 후보 실체 뷰 b는 하나의 질의 처리에 사용 가능하며 크기 및 유지비용이 작다. 이때 저장 공간은 후보 실체 뷰 a 또는 후보 실체 뷰 b 하나만을 저장할 수 있다. 실질적으로 후보 실체 뷰 a 하나만 구축하는 것이 좋지만 후보 실체 뷰의 크기와 유지비용이 크기 때문에 후보 실체 뷰 b가 선택될 가능성이 생기게 된다.

표 1. 관련 연구들의 비교

관련 연구	특성과 문제점
View selection [13]	· 4가지의 변환 과정을 거쳐 실체 뷰의 크기를 줄여 관리 · 질의 빈도 및 실체 뷰 사용 횟수 고려하지 않음
Shortcuts [4]	· 실체 뷰 생성 방법 제안 · 실체 뷰의 크기 및 유지비용 고려하지 않음
Extension Shortcut [5]	· [4]에서 고려하지 않은 실체 뷰의 크기 및 유지비용을 고려 · 실체 뷰 선택 시 역전 상황 발생

III. 제안하는 실체 뷰 선택 기법

1. 제안하는 전체 처리 과정

본 논문에서는 기존 연구의 역전 상황을 방지하며 효율적인 후보 실체 뷰를 선택하기 위하여 의사결정 트리 사용하여 후보 실체 뷰 선택 기법을 제안한다. 실제 환경에서는 저장 공간이 한정 되며 데이터의 갱신이 나타난다. 이를 고려하기 위하여 후보 실체 뷰의 이익뿐만 아니라 후보 실체 뷰의 크기와 갱신율을 고려하는 의사결정 트리를 구축한다.

[그림 2]는 본 논문의 제안하는 기법의 전체 처리 과정을 나타낸다. 먼저 구축 되어있는 RDF 트리플 그래

프(RDF 트리플 데이터베이스)로부터 후보 실체 뷰 집합을 생성한다. 생성된 후보 실체 뷰 집합의 정보를 2개의 테이블로 관리하며 실체 뷰 선택과 후보 실체 뷰 이익 계산 및 의사 결정 나무의 재구축에 사용된다. 각각의 후보 실체 뷰는 의사 결정 트리의 3가지 속성(BenefitF, Size, Update)을 거쳐 3가지 그룹으로 분류된다. 3가지 그룹 중 첫 번째 그룹은 가장 효율이 좋은 그룹으로써 우선적으로 선택되어 실체화가 된다. 두 번째 그룹은 첫 번째 그룹이 구축된 후 저장 공간의 여유 공간이 있을시 실체화되며, 세 번째 그룹은 효율이 떨어지는 그룹으로써 실체화에 있어서 배제된다.

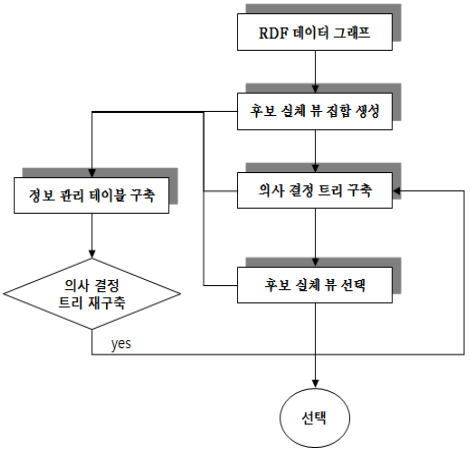


그림 2. 제안하는 실체 뷰 처리 과정

2. 후보 실체 뷰

[그림 3]과 같이 RDF 트리플은 주어와 목적어를 술어로 연결한 그래프로 노드와 노드 사이에 간선이 있는 방향성 그래프이다. 노드는 트리플 중 주어와 목적어의 값들의 집합이고 간선은 술어의 값들의 집합이다. RDF 트리플 그래프는 후보 실체 뷰 선택의 기반 데이터 구조로 한다. 후보 실체 뷰 또한 RDF 그래프로 나타낸다. 본 논문에서는 RDF 그래프 G = {N, E}로 표현하며, N은 노드의 집합이고 E는 간선의 집합을 나타낸다.

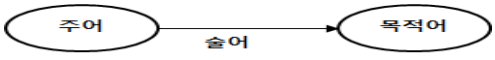


그림 3. 트리플 그래프

후보 실체 뷰들은 RDF 그래프에서 각각의 특정 부분 그래프의 가상 경로를 의미한다. 어떠한 부분 그래프를 실체화하여 관리할 때 최상의 이익을 도출할지 알수 없다. 그렇기에 후보 실체 뷰는 RDF 그래프에서 길이(서로 다른 간선의 수)가 2 이상의 가능한 모든 부분 그래프의 가상 경로로 생성된다. 이렇게 생성되어진 후보 실체 뷰를 $subv$로 표현하고 후보 실체 뷰의 집합 SUBV = $\{subv_1, subv_2 \ldots, subv_n\}$ 이다. 생성된 후보 실체 뷰의 집합은 본 장의 4절에서 의사 결정 트리를 이용하여 질의 처리 시 효율이 높은 후보 실체 뷰를 선택하여 실체화한다.

본 논문에서 그래프와 후보 실체 뷰의 예로서 [그림 4]을 사용하고 있다. 그래프는 각 학교의 학생 정보를 표현하는 트리플로서 심플하게 구성되어 있다. 후보 실체 뷰는 실제로 존재하는 것이 아니라 가상으로서 존재하고 있다. 대학교, 이름, 학번이 필요한 질의 처리가 있다면 총 3번의 자기 조인이 필요 하다. 이 질의의 결과 또는 부분 결과를 $subv_1$ 또는 $subv_3$, $subv_5$로 관리하고 있다면 자기 조인의 횟수를 감소할 수 있게 된다. 이처럼 질의 비용을 절감시키는 $subv_1$ 또는 $subv_3$, $subv_5$를 후보 실체 뷰이다.

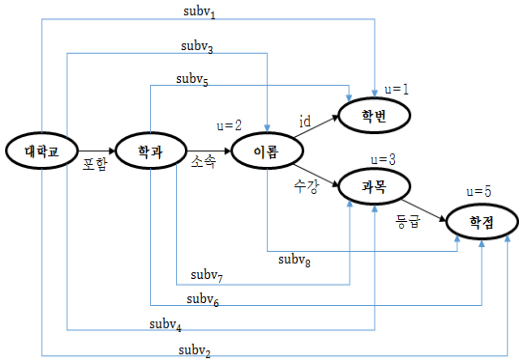


그림 4. 대학생 정보 관리 RDF 데이터베이스 그래프

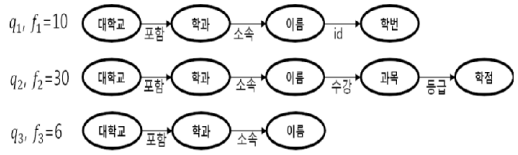


그림 5. 3가지 질의 예제

질의는 질의의 집합 Q = {$q_1, q_2, \bullet\bullet\bullet, q_k$}으로 이루고 있다. 각각의 질의 q_k는 특정 경로에서의 결과 처리를 뜻하며, 각각의 빈도수 f_k를 가지고 있다. 질의 예로서 [그림 5]에 묘사된 q_1, q_2, q_3를 사용한다.

3. 정보 관리 테이블

정보 관리 테이블은 뒤에 나올 의사 결정 트리 구축과 이익 함수 및 의사 결정 트리의 속성에 사용되는 파라미터 값을 관리한다. 정보 관리 테이블을 관리함으로써 후보 실체 뷰의 재구축 시 빠른 선택을 할 수 있다. 또한 질의 처리 시 질의를 처리할 수 있는 실체 뷰가 메모리의 존재 여부를 알 수 있어 질의 처리의 효율을 높일 수 있다.

[표 2]는 질의에 관련된 정보를 관리하는 테이블이다. subv set은 질의 처리 시 사용 가능한 후보 실체 뷰의 집합을 나타내고, frequency는 질의의 빈도수, bestV는 subv set 중 해당 질의 처리에 가장 큰 이익을 가지는 후보 실체 뷰를 나타낸다. 예를 들어 질의 q_1을 처리 시 메모리에 실체 뷰 $subv_1$가 존재 한다면 선택하여 처리한다. 존재하지 않는다면 8개의 후보 실체 뷰의 접근이 아닌 실체 뷰 $subv_3$, $subv_5$ 2개 접근하여 선택하게 된다.

표 2. 질의 정보 관리 테이블

Query	subv set	frequency	bestV
q_1	$subv_1$, $subv_3$, $subv_5$	10	$subv_1$
q_2	$subv_2$, $subv_3$, $subv_4$, $subv_6$, $subv_7$, $subv_8$	30	$subv_2$
q_3	$subv_3$	6	$subv_3$

[표 3]은 의사 결정 트리를 통하여 후보 실체 뷰 선택 시 의사 결정 트리의 속성에 해당하는 정보를 관리한다. related query는 후보 실체 뷰를 사용하여 질의 처리가 가능한 질의 집합을 나타내며, 이득 함수에 사용된다. size는 해당 후보 실체 뷰의 크기(트리플의 수)를 나타내며, Update 횟수는 과거부터 현재까지의 후보 실체 뷰에 속하는 노드들의 갱신 횟수의 총 합을 나타낸다. 한 노드의 갱신 횟수는 [그림 4]에서 노드 위의 u로써 나타낸다. 예를 들어 후보 실체 뷰 $subv_1$의 갱신 횟수는 노드 이름의 갱신 횟수 2와 노드 학번의 갱신 횟수 1을 합한 3이다. Level은 실체 뷰가 관리되어지는 곳을 나타낸다.

표 3. 후보 실체 뷰 정보 관리 테이블

SUBV	related query	size	Update 횟수	Level
$subv_1$	q_1	10,000	3	disk
$subv_2$	q_2	200,000	10	virtual
$subv_3$	q_1, q_2, q_3	8,000	2	memory
...	...	...	...	...
$subv_8$	q_2	200,000	8	disk

의사 결정 트리를 통하여 후보 실체 뷰 선택 시 [표 3]의 related query의 정보와 [표 2]의 frequency의 정보를 사용하여 이득 값을 구한다. [표 3]의 size와 Update 횟수는 각각 의사 결정 트리 속성 Size, Update에 매칭되어 사용된다.

4. 이득 함수

이득 함수는 후보 실체 뷰 선택 시 가장 큰 영향을 미치는 질의 처리 비용의 감소에 따른 이득 값을 구한다. 이득 값을 구하는데 있어서 가장 중요한 것은 질의 처리 비용의 절감과 질의 빈도이다. 질의 처리 절감 비용과 질의 빈도로 구해진 이득 값은 의사 결정 나무 3가지 속성 중에 가장 높은 정보 이득을 가지며 최상의 노드에 위치한다. 이는 후보 실체 뷰 선택 문제에 있어 후보 실체 뷰의 크기 및 갱실율 보다 중요하다는 것을 의미한다.

기존 논문 [4]에서는 후보 실체 뷰의 사용 없이 질의 처리 비용에 후보 실체 뷰를 사용한 질의 처리 비용을 뺀 값에 빈도를 곱하여 구하였다. 예를 들어, 질의 q_1을 처리하는 후보 실체 뷰 $subv_1$의 이득 값을 구한다면, f_1 ×{비용(대학교→학과→이름→학번) − 비용(대학교 $\overrightarrow{subv_1}$ 학번)}으로 계산이 된다. 즉, 총 후보 실체 뷰 $subv_1$의 이득 값은 아래와 같이 계산 한다.

$$BenefitF(subv_i) = \sum_{q_k \in RQ_i} f_k * cost(q_k) - cost(q_k, subv_i) \quad (1)$$

cost(q_k)는 후보 실체 뷰 없이 질의 q_k를 처리하는 비

용이고, cost(q_k, $subv_i$)는 후보 실체 뷰 $subv_i$를 사용하여 질의 q_k를 처리하는 비용이다. f_k는 질의 q_k에 해당하는 질의 빈도이다. RQ_i는 후보 실체 뷰 $subv_i$를 사용하여 질의 처리를 할 수 있는 질의의 집합이다.

후보 실체 뷰를 선택 하여 구축하는데 있어서 대부분 저장 공간의 크기가 제한되는 것이 일반적이다. 제한된 공간에서 선택된 후보 실체 뷰의 수가 많을수록 총 이득이 늘어 날것이다. 또한, 동일한 공간에 후보 실체 뷰의 개수가 같을 때 갱신 비용에 따라 이득의 최대화가 달라질 것이다. [5]에서는 이 둘을 고려하여 식(1)에 후보 실체 뷰의 크기와 갱신 비용을 고려하였다. 우리는 크기와 갱신율을 의사 결정 트리로 고려하고 있다.

[4][5]에서는 식(1)을 기반으로 이익을 구하고 있다. 식(1)은 다음과 같을 때 정확성이 떨어진다. 질의 q_i의 질의 처리 시간이 10, 빈도 f_i는 5이고 질의 q_j의 처리 시간이 5, 빈도 f_j는 5이다($\{q_i, q_j\} \in Q$, $i \neq j$). 이때, 질의 q_i, q_j를 후보 실체 뷰 $subv_i$, $subv_j$를 사용하여 처리하였을 때 각각 9와 4의 비용이 든다. 후보 실체 뷰 $subv_i$, $subv_j$의 이익은 5의 값으로 동일하게 구해진다. 실제로는 10에서 9보다 5에서 4로 처리 비용의 감소된 것이 더 큰 이득이다.

본 논문에서는 정확성을 더욱 높이기 위하여 이득률로 구한다. 즉, 아래와 같이 계산 될 수 있다.

$$BenefitF(subv_i) = \sum_{q_k \in RQ_i} f_k * \frac{cost(q_k) - cost(q_k, subv_i)}{cost(q_k)} \quad (2)$$

위의 예를 식(2)에 대입하여 이익 값을 구한다면 후보 실체 뷰 $subv_i$의 이익은 0.5이고 후보 실체 뷰 $subv_j$의 이익은 1이 구해진다.

5. 실체 뷰 생성

본 논문에서는 [그림 6]에서 묘사된 의사 결정 트리를 사용하여 후보 실체 뷰 선택을 한다. 의사 결정 트리는 데이터 마이닝에서 일반적으로 사용되는 방법론으로, 몇몇 입력 변수를 바탕으로 목표 변수의 값을 예측하는 모델을 생성하는 것을 목표로 한다[11][12]. 의사 결정 트리는 하나의 나무 구조를 이루고 있으며 노드(속성)라 불리는 구성요소들로 이루어져 있다. 각 노드는 루트 노드에서부터 단말 노드까지 높은 정보 분류 이득을 가진 순서대로 구축된다. 실체화할 후보 실체 뷰 선택을 위하여 3가지 속성 BenefitF, Size, Update를 의사 결정 트리에서 고려한다. BenefitF은 이득 값으로 본 장의 4절에서 설명하였다. Size는 후보 실체 뷰의 크기로서 후보 실체 뷰를 실체화 하였을 때의 총 트리플의 수이다. Size는 [표 3]의 size의 정보를 사용한다. Update는 각 후보 실체 뷰의 갱신율이다. 갱신율은 [표 3]의 Update 횟수와 총 질의 횟수를 사용하여구한다. 단, RDF 데이터베이스 그래프의 노드의 변화율은 전체 질의 처리 횟수 보다 작아야한다. 3개의 속성 각각의 조건 a, b, c는 전체 후보 실체 뷰의 평균으로 하였다.

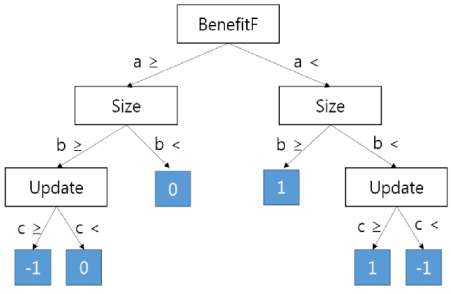


그림 6. 의사결정 트리

[그림 6]의 의사 결정 트리는 3개의 속성을 고려하여 후보 실체 뷰를 최종적으로 {1, 0, -1} 3개의 그룹으로 분류하여 후보 실체 뷰 선택을 한다. 그룹 1에 해당되는 후보 실체 뷰는 이익 값으로 순위를 매겨 저장 공간에 실체 뷰로 우선시되어 저장이 된다. 그룹 -1에 해당되는 후보 실체 뷰는 그룹 1에 해당되는 후보 실체 뷰가 저장 공간에 저장된 후 저장 공간이 남았다면 이익 순위에 따라 저장되어 진다. 그룹 0은 효율이 떨어지는 후보 실체 뷰의 집합으로 후보 실체 뷰의 가상 경로만 저장된다.

IV. 성능평가

본 절에서는 앞에서 제안된 이득 함수와 크기 및 갱신율을 포함한 의사 결정 트리를 이용한 후보 실체 뷰

선택 기법 성능에 대한 성능 평가를 수행한다. 성능 평가는 본 논문에서 제안하는 기법과 [4][5] 기법를 저장 크기에 따른 실체 뷰 개수와 동일한 실체 뷰의 개수일 때의 갱신율 및 갱신 비용을 비교 평가한다. 실험 환경은 Intel Core i5-3570 CPU 3.4GHz, 8 GB RAM, Windows7 환경을 사용하였다.

실험의 데이터 집합은 1,200,000개의 RDF 트리플과 단축 경로의 총 수는 13개를 사용하여 제안하는 기법과 기존 기법 2가지를 구축 하였다. 각 노드의 크기는 총 합이 1,200,000개가 될 수 있는 범위 안에서 랜덤하게 구성 되었다. 각각의 노드는 0.3이하의 갱신율과 갱신율의 총 합이 1이 되도록 구성하였으며, 질의는 임의의 8가지의 타입을 사용하였고, 질의 빈도수는 각 질의 총 합이 200이 되는 범위 안에서 임의 값을 생성하였다.

실체 뷰의 저장 공간의 크기에 따른 실체 뷰의 개수를 기존 기법 2개와 비교하였다. 동일한 저장 공간에서 실체 뷰의 구성 개수가 많다면 임의 질의에 이용 가능한 실체 뷰의 개수가 많을 것이며, 그에 따른 평균 질의 처리 비용이 감소할 것이다.

본 논문에서 제안하는 방법은 DTB(Decision Tree benefit)으로 나타낸다. NB(Normal Benefit)은 [4]에서 제안한 방법이다. 후보 실체 뷰 선택 시 후보 실체 뷰의 크기 및 갱신율을 고려하지 않고 질의의 빈도만을 교려한 방법이다. EB(Extension Benefit)은 [5]에서 제안한 방법이다. 후보 실체 뷰 선택 시 NB에서처럼 질의 빈도를 고려하되 후보 실체 뷰의 크기 및 갱신율을 고려하였다. NB와 EB는 본 논문에서 제안하는 방법과 같이 후보 실체 뷰의 성능을 비교하는 기준이 된다.

[그림 7]은 3개의 후보 실체 뷰 구성 방법을 일정 저장 공간 안에서 실험 결과를 보여준다. x축은 본 논문에서 제안하는 의사 결정 트리를 이용하여 나온 결과인 그룹에 해당하며, y축은 저장 공간 안에 구축된 실체 뷰의 개수이다. 성능 결과는 의사 결정 트리의 결과 그룹 중 그룹 1(이 성능 결과는 group 1로 표시)에 해당하는 실체 뷰들의 크기의 합과 그룹 1 및 그룹 -1(이 성능 결과는 group -1로 표시)에 해당하는 실체 뷰들의 크기의 합의 해당하는 DTB와 NB, EB의 실체 뷰의 개수를 비교 하였다. NB는 실체 뷰의 크기를 고려하지 않기 때문에 실체 뷰의 개수가 가장 적은 것을 볼수 있으며, 실체 뷰의 크기 및 갱신 비용을 고려한 EB 보다 DTB의 실체 뷰의 개수가 많이 구성 된 것을 볼 수 있다. DTB는 NB 보다 평균적으로 55%, EB 보다 27% 많은 실체 뷰들을 구성한다.

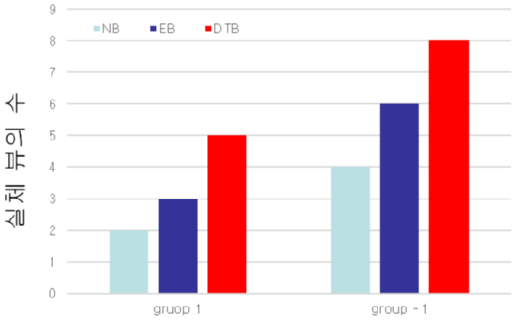


그림 7. 일정 공간에서의 실체 뷰 개수

[그림 8]은 3가지의 실체 뷰 구축 방법을 사용하여 실체 뷰의 개수가 동일할 때 갱신율의 비교를 나타낸다. x축은 본 논문에서 제안하는 의사 결정 트리를 이용하여 나온 결과인 그룹에 해당하며, y축은 실체 뷰들의 갱신율의 합을 나타낸다. [그림 7]에서 나타나듯이 [그림 8]의 group 1은 5개의 실체 뷰, group -1은 8개의 실체 뷰를 사용하여 DTB와 NB, EB의 갱신율을 비교하였다. group 1일 때 DTB는 NB 보다 33%, EB 보다 30% 낮은 갱신율을 보이며, group -1일 때 DTB는 NB 보다 16%, EB 보다 13% 낮은 갱신율을 가진다.

[그림 9]는 동일한 개수의 실체 뷰를 구축한 3가지 기법의 유지비용을 보여준다. x축은 group 1, group -1의 각각의 실체 뷰의 개수 5개, 8개를 나타내며, y축은 각각의 group에 해당하는 유지비용의 크기를 나타낸다. group 1의 실체화 뷰 개수를 구성할 때 DTB는 NB 보다 79%, EB 보다 71% 낮은 유지비용을 보이며, group -1의 실체화 뷰 개수를 구성할 때 DTB는 NB 보다 55%, EB 보다 30% 낮은 유지비용을 가진다.

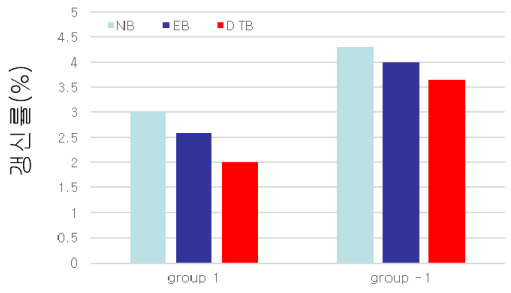


그림 8. 실체 뷰 수에 따른 갱신율

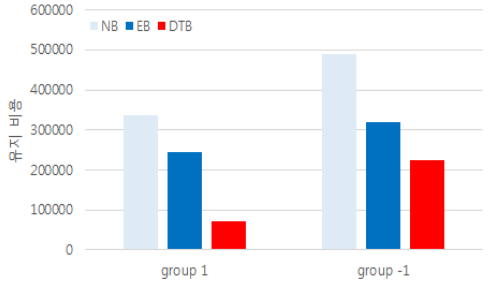


그림 9. 실체 뷰의 동일 개수에 따른 유지비용

V. 결론

본 논문에서는 의사 결정 트리를 이용하여 효율적인 후보 실체 뷰 선택 기법을 제안하였다. 제안하는 기법은 후보 실체 뷰의 이득, 실체화 크기, 그리고 갱신율을 고려하여 의사 결정 트리를 구축한다. 의사 결정 트리는 이익 값을 먼저 고려하기 때문에 기존 논문의 문제점인 역전 상황을 해결하였다. 의사 결정 트리를 통하여 선택된 후보 실체 뷰는 3가지 그룹으로 분류되어 관리 된다. 3가지 그룹으로 분류되어 선택되어진 후보 실체 뷰는 제한된 공간 안에 다양한 실체 뷰를 관리하여 보다 다양한 질의 유형의 처리 효율을 높일 수 있다. 실험을 통하여 제안하는 기법은 기존 기법에 비해 일정 공간에서의 실체 뷰 개수가 27%~55% 많이 구축되는 것을 확인할 수 있다. 동일한 개수의 실체 뷰가 존재 시 갱신율은 제안하는 기법이 기존 기법에 비해 13%~33%, 유지비용은 31%~79% 감소하는 것을 확인하였다. 향후 연구로는 본 논문에서 제안하는 기법의 우수성을 입증하기 위해 질의의 유형과 데이터의 종류를 다양화 및 의사 결정 트리 학습 알고리즘을 통한 의사 결정 트리와의 성능 평가를 실시할 예정이다. 또한 실체 뷰를 메모리와 디스크 두 단계로 구성하여 관리 방법과 교환 기법을 연구하여 실질적인 대용량 RDF 데이터 관리 시스템에 적용할 예정이다.

참 고 문 헌

[1] S. Decker, S. Melnik, F. van Harmelen, D. Fensel, M. Klein, J. Broekstra, M. Erdmann, and I. Horrocks, “The Semantic Web: The Roles of XML and RDF,” IEEE Internet Computing, Vol.4, No.5, pp.63-73, 2000.

[2] D. Abadi, A. Marcus, S. Madden, and K. Hollenbach. “Scalable semantic web data management using vertical partitioning,” Proceedings of international conference on Very large data bases, pp.411-422, 2007.

[3] T. Neumann and W. Gerhard, “RDF-3X: a RISC-style Engine for RDF,” Proceedings of the VLDB Endowment, Vol.1, No.1, pp.647-659, 2008.

[4] V. Dritsou, P. Constantopoulos, A. Deligiannakis, and Y. Kotidis, “Optimizing query shortcuts in RDF databases,” Proceedings of Extended Semantic Web Conference on The Semantic Web: Research and Applications, pp.77-92, 2011.

[5] 강승석, 신준호. “트리플 데이터베이스 단축 경로 이득 함수와 구성 인자 실험 분석,” 한국전자거래학회지 제19권, 제1호, pp.131-143, 2014.

[6] 복범, 이병욱, “관계형 데이터베이스 기반의 RDF 데이터 저장구조 개선에 관한 연구,” 한국인터넷정보학회 춘계학술발표대회, pp.149-150, 2013.

[7] D. J. Abadi, A. Marcus, S. R. Madden, and K. Hollenbach, “SW-Store : a vertically partitioned DBMS for Semantic Web data management,”

The VLDB Journal, Vol.18, No.2, pp.385-406, 2001.

[8] P. Constantopoulos, V. Dritsou, and E. Foustoucos, “Developing query patterns,” Proceedings of European conference on Research and advanced technology for digital libraries, pp.119-124, 2009.

[9] 김천중, 김기연, 윤종현, 임종태, 복경수, 유재수, “대규모 RDF 데이터의 분산 저장을 위한 동적 분할 기법,” 한국정보과학회논문지, 제41권, 제12호, pp.1126-1135, 2014.

[10] 김기연, 윤종현, 김천중, 임종태, 복경수, 유재수, “대규모 RDF 데이터의 특성을 고려한 효율적인 색인 기법,” 한국콘텐츠학회논문지, 제15권, 제1호, pp.9-23, 2015.

[11] 장윤경, 유병섭, 어상훈, 김경배, 배해영 “데이터 웨어하우스에서 의사결정 트리를 이용한 실체화 뷰 선택 기법,” 한국정보처리학회 춘계학술대회, pp.63-66, 2006.

[12] 이병엽, 박용훈, 유재수 “의사결정트리를 통한 자동차산업의 구매패턴 분류,” 한국콘텐츠학회논문지, 제15권, 제1호, pp.9-23, 2015.

[13] F. Goasdoué, K. Karanasos, J. Leblay, and I. Manolescu, “View selection in semantic web databases,” Proceedings of the VLDB Endowment, Vol.5, No.2, pp.97-108, 2011.

저 자 소 개

박 재 열(Jaeyeol Park) 준회원

- 2014년 2월 : 충북대학교 정보통신공학과(공학사)
- 2014년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

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

윤 상 원(Sangwon Yoon) 준회원

- 2014년 2월 : 충북대학교 전자공학과(공학사)
- 2014년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 데이터베이스 시스템, RDF, Provenance Index, 빅데이터 등

최 기 태(Jieun Han) 준회원

- 2014년 2월 : 충북대학교 정보통신공학과(공학사)
- 2014년 3월 ~ 현재 : 충북대학교 정보통신공학과 석사과정

<관심분야> : 데이터베이스 시스템, 분산 컴퓨팅, 부하분산 처리, 빅데이터 등

임 종 태(Jongtae Lim) 정회원

- 2009년 2월 : 충북대학교 정보통신공학과(공학사)
- 2011년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2011년 3월 ~ 현재 : 충북대학교 정보통신공학과 박사과정

<관심분야> : 데이터베이스 시스템, 시공간 데이터베이스, 위치기반 서비스, 모바일 P2P 네트워크, 빅데이터 등

이 병 엽(Byoungyup Lee) 종신회원

- 1991년 2월 : 한국과학기술원 전산학과(공학사)
- 1993년 2월 : 한국과학기술원 전산학과(공학석사)
- 1997년 2월 : 한국과학기술원 경영정보공학(공학박사)
- 1993년 1월 ~ 2003년 2월 : 대우정보시스템 차장
- 2003년 3월 ~ 현재 : 배재대학교 전자상거래학과 교수

<관심분야> : XML, 지능정보시스템, 데이터베이스시스템, 전자상거래학

신 재 룡(Jaeryong Shin) 정회원

- 1996년 2월 : 충북대학교 정보통신공학과(공학사)
- 1998년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2002년 8월 : 충북대학교 정보통신공학과(공학박사)
- 2003년 3월 ~ 현재 : 광주보건대학 보건행정과 교수

<관심분야> : 실시간데이터베이스, 내용기반검색 등

복 경 수(Kyungsoo Bok) 종신회원

- 1998년 2월 : 충북대학교 수학과(이학사)
- 2000년 2월 : 충북대학교 정보통신공학과(공학석사)
- 2005년 2월 : 충북대학교 정보통신공학과(공학박사)
- 2005년 3월 ~ 2008년 2월 : 한국과학기술원 전산학과 Postdoc
- 2008년 3월 ~ 2011년 2월 : ㈜가인정보기술 연구소
- 2011년 3월 ~ 현재 : 충북대학교 정보통신공학과 초빙부교수

<관심분야> : 데이터베이스 시스템, 위치기반서비스, 모바일 P2P 네트워크, 소셜 네트워크 서비스, 빅데이터 등

유 재 수(Jaesoo Yoo) 종신회원

- 1989년 2월 : 전북대학교 컴퓨터공학과(공학사)
- 1991년 2월 : KAIST 전산학과(공학석사)
- 1995년 2월 : KAIST 전산학과(공학박사)
- 1995년 3월 ~ 1996년 8월 : 목포대학교 전산통계학과(전임강사)
- 1996년 8월 ~ 현재 : 충북대학교 정보통신공학부 및 컴퓨터정보통신연구소 교수
- 2009년 3월 ~ 2010년 2월 : 캘리포니아주립대학교 방문교수

<관심분야> : 데이터베이스 시스템, 빅데이터, 센서네트워크 및 RFID, 소셜 네트워크 서비스, 분산 객체컴퓨팅, 바이오인포매틱스 등