반구조적 데이터의 효율적인 최소경계 스키마 추출 기법

박경현[U] *김록원 양은주 최은선 류근호
충북대학교 데이터베이스 연구실
*한국전자통신연구원 전자상거래 연구부
{khpark, ejyang, eschoi, khryu}@dblab.chungbuk.ac.kr
*rwkim@etri.re.kr

An Efficient Technique for Extracting Lower Bound Schema from Semistructured Data

Kyoung Hyun Park[U] Rock Won Kim Eun Joo Yang Eun Sun Choi Keun Ho Ryu
Dept. of Computer Science, Chungbuk National University
*ETRI

요 약

반구조적 데이터는 기존의 스키마와는 달리 고정된 스키마가 없고 주어진 데이터 인스턴스에 대해 하나 이상의 스키마가 존재한다. 따라서 여러 개의 스키마 추출이 가능한데 그중 가장 정확한 스키마를 추출해야 하는 문제(Schema Extraction)가 발생한다. 이러한 문제를 해결하기 위해 지금까지 여러 가지 스키마 추출 기법들이 제안되었는데 대표적인 것으로 데이터가이드(DataGuide)를 이용하여 최대경계 스키마를 추출하는 방법과 데이터로그(DataLog)를 이용하여 최소경계 스키마를 추출하는 방법이 있다.

이 논문에서는 기존의 데이터로그를 이용하는 방법보다 향상된 최소경계 스키마 추출 기법을 제안하고 이전의 스키마 추출 기법들과 비교함으로써 알고리즘의 성능을 살펴본다

1. 서 론

반구조적 데이터(semistructured data)는 자기 서술적(self-describing)이고 고정된 스키마가 존재하지 않는다(schemaless)는 점에서 기존의 데이터와 구분된다[Abite99]. 반구조적 데이터의 이러한 특징으로 인해 반구조적 데이터를 효율적으로 저장하고 질의를 최적화하며 사용자에게 편의성을 제공해 주기 위해서 반구조적 데이터의 스키마 추출이 필수적으로 요구된다.

일반적으로 반구조적 데이터로부터 스키마를 추출할 때 주어진 데이터 인스턴스내의 각각의 객체에 대하여 그 객체에서 나가거나 들어오는 간선들과 그 객체들과의 관계에 따라서 스키마를 분류하기 때문에 분류하는 규칙에 따라 여러 개의 스키마를 추출할 수 있게 된다. 이것은 최악의 경우 반구조적 데이터내에 존재하는 객체 수만큼의 스키마가 생성될 수도 있음을 나타낸다.

따라서 추출 가능한 여러 개의 스키마중 가장 정확한 스키마를 추출해야 하는 문제가 발생하게 되는데 이러한 문제를 해결하기 위해서 지금까지 반구조적 데이터의 스키마 추출을 위한 여러 가지 방법들과 이론들이 제안되었다[Gold97, Bune96, Calv98, Nest98].

그중 대표적인 스키마 추출 기법으로 데이터가이드와 데이터로그를 이용하는 방법이 있다.

데이터가이드는 Lore 프로젝트[Mchu97]에서 소개된 스키마 추출 기법으로 주어진 반구조적 데이터에 대한 모든 엘리먼트 경로를 포함하는 최대 경계 스키마를 생성해 낸다[Gold97]. 이에 반해 [Nest98]은 데이터로그에 기반하여 엘리먼트들의 공통된 경로만을 포함하는 최소경계 스키마(lower bound schema) 추출 기법을 제안하고 있다.

특히, 최소경계 스키마를 생성하는 [Nest98]에서는 데이터로그를 이용한 최대 고정점(fixpoint) 방법을 이용하기 때문에 모든 데이터에 존재하는 객체들이 최대 고정점에 도달할 때까지 반복해서 접근해야 한다. 이는 스키마 추출시에 많은 비용을 요구하며 이에 따라 스키마 갱신의 부담을 증가시킨다.

이 논문에서는 이러한 데이터로그의 문제점으로 나타나는 최대 고정점 방법을 개선할 수 있는 효과적인 알고리즘으로 시뮬레이션을 이용한 최소경계 스키마 추출 기법을 제안하고 기존의 스키마 추출 기법들과의 비교를 통해서 시뮬레이션을 이용한 최소 경계 스키마 추출 기법의 성능을 평가해 본다.

2. 관련연구

반구조적 데이터는 일반적으로 레이블과 방향성이 있는 그래프(labeled directed graph) 모델[Papa95]로 표현되고 기존의 스키마 추출 기법들도 이러한 그래프 모델에 기반을 두고 있다.

이러한 데이터 모델을 바탕으로 한 반구조적 데이터의 스키마 추출은 사용자의 편의성, 저장의 효율성, 질의의 최적화등을 위한 필요성으로 인해 많은 연구가 이루어져 왔다.

데이터가이드가 그 중 하나로 Lore 프로젝트에서 소개한 반구조적 데이터 모델인 OEM(Object Exchange Model)를 바탕으로 최대 경계 스키마를 추출하고 있다.

데이터로그는 최대 고정점을 이용하여 최소 경계 스키마를 추출할 수가 있다. 그러나 단순히 최대 고정점을 이용하여 타입을 추출하게 되면 많은 수의 타입이 생성되고 경우에 따라서 실제 데이터와 비슷한 양만큼의 타입이 생성되는 경우가 발생하기 때문에 클러스터링을 이용하여 타입의 수를 줄이고 있다[Nest98].

데이터가이드는 최대경계 스키마를 생성하기 위해 주어진 데이터 그래프의 모든 노드와 간선을 한 번만 검색하므로 O(mn)의 시간이 소요되는[Gold97] 반면 데이터로그는 주어진 데이터 그래프의 노드를 방문할때 모든 객체들이 분류된 초기 타입 릴레이션으로부터 그 릴레이션이 고정점에 도달할 때까지 반복하여 검사하기 때문에 $O(n^2)$이라는 시간이 걸리므로 데이터로그에 비해 시스템에 많은 부담을 주게 된다.

3. 최소경계 스키마 추출

3.1 스키마 추출 모델

이 논문에서는 반구조적 데이터의 스키마 추출 기법에 대한 입력 데이터로 XML 문서를 사용한다. 그러나 XML 문서는 기본적으로 구조적 문서를 정의하는 모델로부터 시작되었기 때문에 XML 데이터 모델과 반구조적 데이터 모델사이에는 약간의 차이가 존재하게 된다.

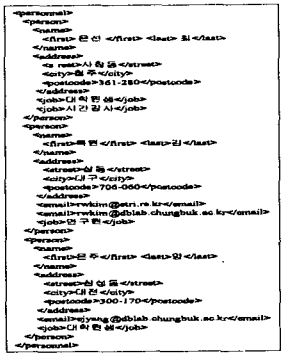

그림 1. XML 문서

따라서 XML 문서의 스키마를 추출하는데 반구조적 데이터의 스키마 추출 기법을 적용하기 위해서는 XML 데이터 모델을 레이블과 방향성이 있는 그래프 모델로 변경해 주어야 하는 작업이 필요하게 된다.

그림 1과 그림 2은 XML 문서와 이에 대응되는 반구조적 데이터의 데이터 모델을 보여주고 있다. XML 데이터 모델은 레이블이 노드상에 표현되고 반구조적 데이터에서는 간선상에 표현된다. 따라서 XML 데이터 모델을 반구조적 데이터 모델로 변형하기 위해서는 단지 XML 데이터의 노드에 있는 레이블을 노드로 들어오는 간선상에 표현해 주면 된다.

따라서 위의 방법을 통해 XML 문서로부터 생성된 반구조적 데이터 모델을 대상으로 하여 기존의 스키마 추출 기법과 이 논문에서 제안하는 시뮬레이션을 이용한 스키마 추출 기법을 적용하여 XML 문서로부터 최소 경계 스키마를 추출할 수 있다.

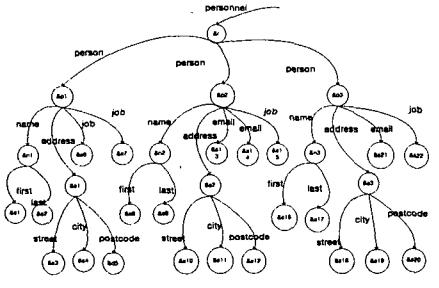

그림 2. 반구조적 데이터 모델

3.2 시뮬레이션

정점의 집합을 V라 하고, E⊆ V^2을 만족하는 간선의 집합을 E, 그리고 정점 v를 레이블 ⟨v⟩로 매핑하는 함수를 ⟨· ⟩ : V→ A이라 할 때 레이블이 있는 그래프(labeled graph)는 G=(V, E, A, ⟨· ⟩)로 나타낼 수 있다. 이때 정점 v를 계승하는 정점(successor)들을 post(v)={u|(v,u)∈E}라 하면 정점들의 집합상에서 이진 릴레이션(binary relation) ≤ ⊆ V^2인 이진 릴레이션에 대해 u≤ v은 다음의 두 조건을 만족할 때 정점 v는 정점 u를 시뮬레이트(simulate)한다고 한다: (1) ⟨u⟩ = ⟨v⟩ , (2) u'∈post(u)인 모든 정점에 대해 u'≤ v'이고 v'∈post(v)인 정점 v'가 존재한다.

이런 시뮬레이션을 이루기 위한 조건은 아래 그림 3에서 잘 보여진다.

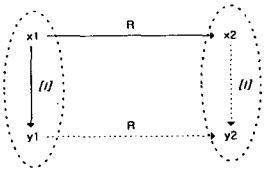

그림 3. 시뮬레이션 다이어그램

그림 3에서 직선으로 이루어진 간선들이 존재할 때 점선들로 이루어진 간선들이 존재하고 이에 대응하는 정점 y2가 존재하면 이진 릴레이션 R은 시뮬레이션이 된다. 다시 말해서, 간선 x2[l]y2는 x1[l]y1을 시뮬레이트한다.

이러한 시뮬레이션을 반구조적 데이터의 스키마를 추출하는데 적용하기 위해서는 기존의 정의에 약간의 조건이 더 필요하다. 두 그래프를 반구조적 데이터의 인스턴스를 나타내는 데이터 그래프와 이에 대한 스키마 그래프라고 하면 우선, 데이터 그래프의 간선 x1[l]y1이 간선 x2[l']y2에 시뮬레이트되어야 한다. 이때 l'는 레이블 l에 대응되는 레이블이나 와일드카드(_)를 나타낸다. 둘째로, 시뮬레이션 관계의 두 그래프는 루트가 존재해야 한다. 즉 데이터 그래프와 스키마 그래프의 루트 r과 r'에 대해서 rRr'가 존재해야 한다. 마지막으로, xRy에서 y가 원자 타입의 노드이고 스트링이나 정수형 같은 타입의 값을 가지면 x도 반드시 원자 타입의 노드이고 같은 타입을 값을 가져야 한다.

이러한 시뮬레이션의 개념은 데이터 그래프와 스키마 그래프 사이에 대한 관계의 유효성을 검사하는 데 이용할 수 있다. 하지만 스키마 그래프가 생성되기 이전에 주어진 데이터 그래프에 대한 스키마 그래프를 생성하기 위해서는 주어진 데이터 그래프의 어떤 노드가 같은 그래프내의 다른 노드와 시뮬레이션 되는지를 판단하여 스키마 추출에 이용한다.

3.3 시뮬레이션을 이용한 스키마 추출

시뮬레이션은 그래프들간의 일치성을 검사하는데 사용된다. 시뮬레이션의 이러한 성질은 주어진 데이터 그래프 G에 대해 타입 정보 추출을 가능하게 한다. 즉, 하나의 그래프를 대상으로 그래프가 내포하고 있는 타입 정보를 시뮬레이션을 이용하여 추출할 수가 있다.

시뮬레이션을 이용하여 스키마를 추출하기 위해서는 먼저 몇 가지 정의가 필요하다. 주어진 그래프 G의 임의의 정점 v에 대해 시뮬레이션 관계에 있는 정점들의 집합을 sim(v)라고 정의한다. 즉, sim(v)는 v가 가지고 있는 출력 간선을 포함하는 정점들을 의미한다. 주어진 임의의 정점 v에 대해서 부모 정점들과 자식 정점들을 post(v)={u|(v,u)∈E}와 pre(v)={u|(u,v)∈E}로 각각 정의할 수 있는데 여기서 E는 그래프에 속하는 간선들의 전체 집합을 의미한다.

```
foreach (node in graph G) {
  labels = getNodeLabels (v);
  foreach (v' in graph G) {
    labels ' = getNodeLabels (v');
    if (labels ⊆labels ')
      sim (v).add(v ');
  }
  remove(v) = pre(v) - pre( sim (v));
}
previsim (v) = V;
while (vertex v such that remove(v)   ≠ ∅) {
    {assert for v, remove(v) = pre(  previsim (v)) - pre( sim (v));}
    foreach (node v in graph G) {
      u = pre(v);
      foreach (p in remove(u)) {
        foreach (w in remove(v)) {
          if (w ∈sim (p)) {
            sim (p) = sim (p) - w;
            foreach (w'' in pre(w)) {
              if (post(w '') ∩ sim (p) = ∅)
                remove(p) = remove(p)   ∪ w";
            } // foreach
          } // if
        } // foreach
      } // foreach
    } // foreach
    previsim (v) = sim (v);
} // while
```

그림 4. 시뮬레이션 알고리즘

데이터로그를 이용하여 스키마를 추출할 경우에는 스키마를 얻기 위한 순환조건으로 객체의 내항 프리디킷의 확장에 대해서 만족하는지를 검사한다. 마찬가지로 시뮬레이션을 이용하는 경우에도 이러한 순환조건이 수행되어져야 하기 때문에 임의의 정점 v에 대해서 remove(v)라는 함수를 정의해야 한다. 이것은 어떤 정점 v에 대해서 v의 pre(v)에 속하지 않는 객체들의 집합을 의미한다. 만약 어떤 정점 v에 대해서 초기에 sim(v)의 집합을 얻었을때 이 sim(v)에 속하는 모든 객체들이 간선을 통한 객체들간의 관계가 고려되지 않은 상태가 된다. 그러므로 어떤 정점 v가 주어지면 그 정점 v의 pre(v)의 원소 u에 대한 sim(u)를 구하고 sim(u)에서 remove(v)에 해당하는 객체들을 제거함으로써 간선을 통한 객체들 상호간의 관계를 고려할 수 있게 된다.

그림 4은 이와 같은 이론을 기반으로 스키마를 추출하는 알고리즘을 보여준다. 그림 4에 나타나는 알고리즘은 두 단계의 과정을 통해서 주어진 데이터 그래프 G에 대한 스키마를 추출해 낸다. 이러한 시뮬레이션 알고리즘을 통해서 얻은 최소 경계 스키마는 그림 5에서 보여주고 있다.

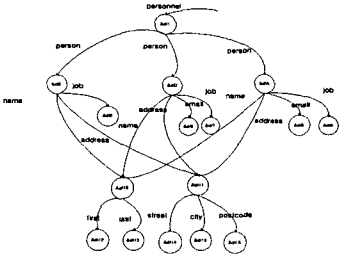

그림 5. 시뮬레이션을 이용한 최소 경계 스키마 그래프

4. 데이터로그와 시뮬레이션

데이터 로그와 시뮬레이션은 모두 최소 경계 스키마를 생성한다. 스키마 그래프 S의 Root노드를 어떤 데이터 객체 X라고 가정하면 (Root, X)는 R에 있다. 시뮬레이션 조건은 Root로부터 나오는 각각의 간선이 X로부터 나오는 동일한 간선을 가진다는 것이다. 여기서는 2개의 Root의 간선이 있기 때문에 이 조건은 아래와 같다.

$$Root(X) \Rightarrow (\exists Y,\ Z(link(X,person,Y) \wedge Person(Y) \wedge link(X,company,Z) \wedge Company(Z)))$$

이것은 only-if조건인데 아래의 최대 고정점 의미를 가지는 데이터로그 규칙으로 기술할 수 있다.

```
Root(X)      :-link(X,person,Y), Person(Y),
               link(X,company,Z), Company(Z)
Person(X)    :- link(X,name,N), string(N),
               link(X,works-for,N), Company(Y)
Company(X)   :- link(X,name,N), string(N),
               link(X,manager,Y), Person(Y)
```

이러한 데이터로그 프로그램과 시뮬레이션 사이의 관계는 매우 밀접하다. 즉, 데이터로그 프로그램을 위한 모델은 시뮬레이션과 일치하며 그 역도 성립한다. 더 일반적으로 데이터로그 타입화 규칙은 최소 경계 스키마와 일치한다.

결론적으로, 시뮬레이션 알고리즘의 초기화는 데이터로그 규칙으로부터 만들어지는 타입 릴레이션과 같다. 또한 스키마 추출의 결과를 가지고 스키마 그래프를 생성하기 위해서는 각각의 sim집합에 대응하는 규칙들이 존재해야 된다. 그러므로 그래프 시뮬레이션 알고리즘은 데이터로그의 최대 고정점을 대신하여 시간비용을 줄이는 역할을 하며 스키마 추출을 위한 초기화 방법과 스키마 추출 결과에 대한 스키마 그래프 생성은 데이터로그 방법과 동일하다.

5. 실험 및 성능분석

알고리즘의 성능분석을 위해 실험에 사용한 데이터는 XML 데이터로 트리로 표현되는 데이터와 그래프로 표현되는 데이터를 사용하였고 128M의 펜티엄II 400Mz상에서 실험이 이루어졌으며 모든 알고리즘은 JAVA를 이용하여 구현되었다.

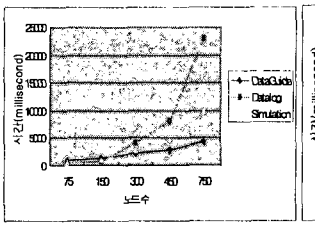

그림 6. 트리 데이터에 대한 스키마 추출 시간

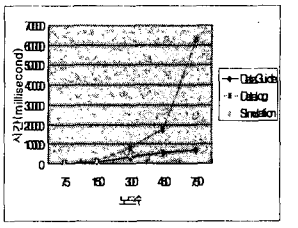

그림 7. 그래프 데이터에 대한 스키마 추출 시간

시뮬레이션을 이용한 알고리즘은 정점 n과 간선 m에 대해 O(mn)의 시간을 보장하는 것이 특징이다. 물론 데이터로그와 마찬가지로 초기화 단계에서 $O(n^2)$의 시간이 소요된다. 그러나 remove(v)를 이용하여 sim(v)의 원소들을 제거해가기 때문에 $O(n^2)$의 시간이 소요되는 데이터로그보다 성능이 우수하다. 그림 6과 그림 7은 각각의 알고리즘에 대한 성능평가를 보여주고 있다.

특히 입력 데이터가 그래프 형태일 경우에는 객체들간의 링크 속성으로 인해 최고 고정점에 도달하기까지의 반복횟수가 증가하기 때문에 시뮬레이션이 데이터로그보다 훨씬 좋은 성능을 보이고 있다.

6. 결론 및 향후연구

이 논문에서는 반구조적 데이터의 최소경계 스키마를 추출하는 기존의 방법보다 효율적인 방법인 시뮬레이션을 이용한 최소경계 스키마 추출 기법을 제안하고 기존의 스키마 추출기법들과 비교함으로써 제안된 알고리즘의 성능을 평가하였다.

제안된 시뮬레이션을 이용한 방법은 최소경계 스키마를 생성하며 이것은 문서를 데이터베이스에 저장할 때 논리적인 저장 구조를 최적화하는데 아주 유용하며 사용자에게 데이터베이스에 저장되어있는 문서 구조를 보여주는데 이용될 수 있다. 향후 연구로서는 최소경계 스키마와 최대경계 스키마를 이용하여 반구조적 데이터나 XML 문서를 효율적으로 저장하는 방법에 대한 연구가 필요하다.

7. 참고 문헌

[Abite99] S. Abiteboul, P. Bunneman, D. Suciu. Data on the Web: From Relations to Semistructured Data and XML. Morgan Kaufmann 1999

[Bune96] P. Buneman, S. Davidson, G. Hillebrand, and D. Suciu. A query language and optimization techniques for unstructured data. In SIGMOD, pages 505-516, Montreal, 1996.

[Calv98] D. Calvanese, G. Giacomo, and M. Lenzerini. What can Knowledge representation do for semi-structured data? In Proc. Of the 15th National Conf. On Artificial Intelligence(AAAI-98)

[Gold97] R. Goldman, J. Widom. DataGuide : Enabling Query Formulation and Optimization In Semistructured Databases. In Proc. of the 23rd VLDB Conference Athens, Greece, 1997

[Mchu97] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J Widom. Lore: A Database Management System for Semistructured Data. SIGMOD Record, 26(3), September, 1997.

[Nest98] S. Nestorov, S. Abiteboul, R. Motwani: Extracting Schema from Semistructured Data. In SIGMOD, pages 295-306, 1998