데이터 마이닝 질의 처리를 위한 질의 처리기 설계 및 구현

김 충 석† · 김 경 창††

요 약

데이터 마이닝 시스템은 기본적으로 요약화, 연관화와 분류화 등 다양한 유형의 데이터 마이닝 기능을 포함한다. 이들 기능을 수행하기 위해서 포괄적으로 표현하기 위한 강력한 데이터 마이닝 질의 언어가 요구되며, 사용자에게 보다 친숙한 마이닝 환경을 제공하기 위해서 그래픽 사용자 인터페이스(GUI)를 이용한 데이터 마이닝 질의 언어의 개발이 중요하게 언급된다. 뿐만 아니라 데이터 마이닝 그 자체로서 독립적인 수행이 아니라 수많은 데이터를 포함하며, 의사 결정에 적합한 구조로 설계되어 있는 데이터 웨어하우스와 연관된 데이터 마이닝 질의 처리가 필요하다. 본 논문에서는 먼저 GUI를 통하여 사용자가 쉽게 데이터 마이닝 질의를 수행할 수 있도록 한다. 또한 질의를 처리하기 위한 데이터 마이닝 질의 처리 프레임워크를 제시한다. 데이터 마이닝 질의의 대상은 데이터 웨어하우스에 저장되어있는 데이터이기 때문에 데이터 웨어하우스의 구축이 필요하다. 본 논문에서는 데이터 웨어하우스 구축에 필요한 스키마 생성을 위해서 스키마 생성기를 아울러 개발하여 이용한다. 마지막으로 연관 규칙 발견을 위한 데이터 마이닝 질의를 처리하기 위한 질의 처리기의 구현 내용을 보인다.

Design and Implementation of a Data Mining Query Processor

Chung-Seok Kim† · Kyung-Chang Kim††

ABSTRACT

A data mining system includes various data mining functions such as aggregation, association and classification, among others. To express these data mining functions, a powerful data mining query language is needed. In addition, a graphic user interface (GUI) based on the data mining query language is needed for users. In addition, processing a data mining query targeted for a data warehouse, which is the appropriate data repository for decision making, is needed. In this paper, we first build a GUI to enable users to easily define data mining queries. We then propose a data mining query processing framework that can be used to process a data mining query targeted for a data warehouse. We also implement a schema generator to generate a data warehouse schema that is needed to build a data warehouse. Lastly, we show the implementation details of a query processor that can process queries that discover association rules.

키워드 : 데이터 베이스(Data Base), 데이터 마이닝(Data Mining), 질의처리(Query Processor), 데어터웨어하우스(Data Warehouse)

1. 서 론

최근에 정보화의 발달로 데이터를 생성하고 수집하는 일이 급속하게 증가되고 있다. 수 백만개의 데이터베이스들이 비즈니스 관리와 정부 기관, 자연 과학, 공학 데이터 관리 및 많은 애플리케이션에서 사용되어지고 있다. 이와 같이 데이터와 데이터베이스의 급속한 증가는 데이터를 유용한 정보로 변환할 수 있는 새로운 기술과 도구의 필요를 반영하며, 결과적으로 데이터 웨어하우스와 데이터 마이닝이라는 분야가 중요한 연구 대상으로 자리잡게 되었다[1, 7].

데이터 웨어하우스는 기존의 수많은 데이터베이스에서 사용되는 운용 데이터에 대한 정보의 가치와 역할 인식으로 나타난 용어로서, 전략적인 데이터 사용을 위한 방법이다. 데이터 웨어하우스는 기업 내에서 많은 의사 결정 지원 시스템(decision support system, DSS)과 경영 정보 시스템(executive information system, EIS)의 애플리케이션과 처리를 지원하기 위해서 질적으로 향상되고 통합된 데이터를 갖는 플랫폼 환경이며, 이를 통하여 기존의 운영 데이터를 의사 결정 정보로 제공할 수 있게 된다[2, 3]. 이를 위해서 일반적으로 데이터 웨어하우스는 관계형 OLAP(ROLAP) 서버나 확장 관계형 DBMS시스템을 사용하여 구현 될 수 있다. 이런 서버는 데이터가 관계형 데이타베이스에 저장되었다고 가정한다. 그리고 이런 서버들은 SQL의 확장, 특정한 접근, 효율적으로 다차원 데이터 모델과 연산을 구현하기 위한 구현 방법등을 제공한다.

이와 대조적으로 지식 발견이라고도 하는 데이터 마이

※ 이 연구는 한국과학재단의 특정기초 연구비 지원(과제번호 : 97-0102-04-01-3)을 받았음.
† 종신회원 : 신라대학교 컴퓨터정보공학부 교수
†† 정 회 원 : 홍익대학교 컴퓨터공학과 교수
논문접수 : 2000년 10월 18일, 심사완료 : 2001년 3월 28일

닝은 데이터베이스에 있는 데이터에서 알려지지 않고 잠정적으로 유용한 정보를 추출하는 과정을 의미한다[7]. 데이터 마이닝은 데이터 요약화에서 연관 규칙 마이닝, 데이터 분류화, 특정 패턴을 찾는 등에 이르기까지 다양한 범주의 기능을 수행한다. 이들 기능을 수행하는 데이터 마이닝은 데이터 웨어하우스가 존재하지 않는 기존의 운용 데이터베이스에서도 수행이 가능하다. 그러나 데이터 웨어하우스를 기반으로 수행되는 데이터 마이닝은 이미 데이터 웨어하우스가 정제된 데이터를 내포하기 때문에 일반 운용 데이터베이스를 기반으로 하는 데이터 마이닝 질의 보다 효율적이며 유용한 정보를 얻을 수 있다[8]. 유용한 정보 결과를 습득하기 위해 데이터 마이닝은 질의어, 기본적인 연산과 질의 처리 전략들이 필요하게 되며 데이터 마이닝 기능을 포괄적으로 표현할 수 있는 일반화된 질의 언어와 구문의 설계 방법론 및 질의처리기의 구현이 필요하다[9].

그러므로 본 논문에서는 데이터 마이닝의 수행 결과가 효율적이며, 유용한 정보를 얻을 수 있도록 데이터 웨어하우스를 기반으로 이루어진다. 데이터 웨어하우스 구축을 위한 과정은 본 논문의 범위를 벗어나기 때문에 [11]을 참조한다. 생성된 데이터 웨어하우스를 기반으로 서로 다른 데이터 마이닝 방법들의 내부 메커니즘을 이해하여 마이닝 질의를 수행할 수 있는 일반화된 데이터 마이닝 질의 언어(Data Mining Query Language, DMQL)를 설계하며, 설계된 DMQL에 기반하여 질의를 처리할 데이터 마이닝 질의 처리 프레임워크와 데이터 마이닝 질의 처리기의 시스템 구조를 열거하고 데이터 웨어하우스를 기반으로 연관 규칙 발견 질의를 처리하는 질의 처리기의 설계 및 구현을 제시하고자 한다. 연관 규칙 이외의 다른 데이터 마이닝 질의 처리는 연관 규칙 질의 처리 방법과 유사한 방법으로 수행 가능하므로 본 논문에서는 연관 규칙 질의 처리를 대표로 예를 들었다.

본 논문의 기여는 데이터 마이닝 질의처리 과정을 데이터 웨어하우스 프레임워크에서 접근하였다는 것이다. 이를 위하여 데이터 마이닝 질의 언어를 제시하고 GUI를 통하여 제시된 질의 언어에 기반한 질의를 사용자가 정의할 수 있도록 하였다. 또한 정의한 질의를 처리하기 위한 데이터 마이닝 질의 처리기를 설계 및 구현하였다.

본 논문의 구성은 다음과 같다. 제 2장에서 기존에 이루어진 연구들에 관하여 살펴보며, 제 3장에서 데이터 마이닝 질의의 기반이 되는 데이터 마이닝 질의 언어에 대해서 알아본다. 제 4장에서는 데이터 마이닝 질의 처리의 프레임워크와 질의 처리기의 구성 모듈들에 대해 살펴보며, 제 5장에서는 구현에 관한 내용으로서 데이터 마이닝 질의 처리기에 대한 구현을 다루며, 6장에서 결론으로 끝을 맺는다.

2. 관련 연구

기존의 관계형 시스템에서 그래픽 사용자 인터페이스를 이용해서 데이터베이스에 접근하는 많은 방법들이 존재하였다. 그러나 이들 대부분의 질의 언어는 관계형 질의 언어를 바탕으로 이루어져있다[6]. 이러한 측면에서 관계형 시스템의 성공은 부분적으로나마 관계형 질의 언어의 표준화에 그 영향을 받았다고 할 수 있다. 데이터베이스 시스템에서 SQL-3, OMG 등 최근의 표준화를 위한 활동은 향후 데이터베이스 시스템의 개발에서 표준 데이터베이스 언어의 중요성을 보여주는 단면이다.

이와 같은 관점에서 데이터 마이닝 질의 언어에 관한 연구는 데이터 요약, 연관 규칙, 패턴 탐색 등의 다양한 데이터 마이닝 기능을 포괄적으로 표현할 수 있는 일반화된 질의 언어와 구문의 설계 방법론 및 질의처리기의 구현이 필요하다[9]. 이에 파일 단위로 처리되는 환경에서 개발된 마이닝 질의 언어로서 [10]을 들 수 있다. [4, 6]에서는 관계형 데이터베이스를 기반으로 한 데이터 마이닝 질의 언어가 사용되었다. 데이터 웨어하우스를 기반으로 수행되는 질의 처리는 [12]에서 언급하고 있다. [12]는 데이터 마이닝을 수행하는 질의 처리기가 아니라 데이터 웨어하우스를 기반으로 단순 질의를 처리하는 것이다.

그러므로 이들 마이닝 질의 언어는 파일 단위이거나 관계형 데이터베이스를 기반으로 수행되는 질의 언어이며, 이로 인하여 의사 결정에 적합하게 설계된 데이터 웨어하우스의 일반 사용자들의 사용에 많은 어려움을 안고 있다. 또한 몇 십년간 일반 사용자에게 사용되어 익숙해진 SQL 언어에 유사한 데이터 마이닝 질의 언어의 설계와 설계된 마이닝 질의 언어를 데이터 웨어하우스에 기반한 데이터 마이닝 질의 언어의 수행이 필요하다.

3. 데이터 마이닝 질의 언어

본 논문에서는 데이터 마이닝 질의 언어를 설계하는 것이 목적이 아니다. 본 논문에서 다루는 DMQL은 [6]에서 제안한 DMQL을 기반하여 사용자에게 익숙한 SQL과 유사한 형태의 데이터 마이닝 질의언어이다. 본 장에서는 먼저 DMQL의 구문을 제시하고 유형별 질의 예제를 살펴보고자 한다. DMQL이 포함하는 데이터 마이닝 규칙들은 연관 규칙(Association rule), 특성(Characteristic) 규칙, 차별(Discriminant) 규칙, 분류(Classification) 규칙 등으로서 데이터 웨어하우스를 바탕으로 수행될 데이터 마이닝 질의 유형들이다. 본 논문에서는 데이터 마이닝의 여러 유형들 중에서 데이터 웨어하우스를 기반으로 수행되는 대표적인 데이터 마이닝 규칙인 연관 규칙을 발견하는 데이터 마이닝 질의에 대한 질의 처리가 수행되는 과정을 보이고자 한다.

데이터 마이닝 규칙들은 각각의 유형별로 필요한 추가 세부 항목들이 존재한다. 예를 들면, 연관 규칙의 경우에는 Support와 Confidence를 필요로 하며, 차별 규칙은 비교 대상이 명시 되어야 하는것 등이다. 다음은 본 논문에서 제시한 데이터 마이닝 질의 언어의 EBNF 문법 형식을 나타내었다.

```
<DMQL> :: = USING <DB name>
            [MINE RULE <mining technique> AS<rule-name>]
            {USING HIERARCHY<hierarchy name> for
              <attribute>}
            [<rule spec>]

            SELECT <attribute-list>
             FROM <table-list>
             [WHERE <condition>]
             [EXTRACTING RULES WITH <constraint>]

             SET RESULT DISPLAY
             REPORT TYPE <report formula>

<mining technique> :: = ASSOCIATION | CHARACTERISTIC |
      DISCRIMINANT | CLASSIFICATION | CLUSTERING
<rule spec> :: = {FOR <class1>WITH<condition1>{VS<class2>
      WITH <condition2>}}
<attribute-list> :: = {attribute | ","}
<table-list> :: = table name{table name|","} :
<constraint> :: = SUPPORT : <numeric-value>, CONFIDENCE :
      <numeric-value> | THRESHOLD : <numeric-value>
<report formula> :: = <report type>{AND <report type>}
<report type> :: = TABLE | GRAPH
```

본 논문의 질의 결과 표현 방법은 데이터 마이닝 질의 결과를 테이블 형식과 그래프 형식으로 나타낼 수 있다. 위에 나타낸 EBNF 문법에서 데이터 마이닝 질의를 할 경우 **MINE RULE <mining technique> AS <rule name>**에 사용할 마이닝 규칙을 명시한다. 그리고 개념 계층을 이용하는 부분 **(USING HIERARCHY<hierarchy name> for <attribute>)**에서는 개념 계층 모듈을 통해 정의하며, Knowledge base에 저장된 특정 속성의 개념 계층을 사용한 질의를 가능케 한다. 또한 하나의 속성에 여러 개념 계층이 존재 할 수도 있기 때문에 개념 계층의 이름과 속성의 이름을 모두 명시하게 하였다.

예를 들면 성적에 대한 개념 계층이 존재한다고 할 때 수학에서의 매우 잘함, 잘함, 보통, 못함, 아주 못함 등의 개념 계층이 과학에서의 개념 계층과 같을 수 없기 때문이다. ***<rule spec>***은 마이닝 규칙들에서 필요로 하는 추가 구문에 대한 것으로 여기에서는 차별 규칙의 경우에만 해당이 된다. **SELECT, FROM, WHERE** 절은 표준 SQL 형식을 따르고 있다. **SELECT** 절에서는 선택하고자 하는 속성에 대한 연산이 적용되고, **FROM** 절에는 테이블의 이름이, **WHERE** 절에는 기존 SQL문의 조건 값에 대해 기술한다. **EXTRACTING RULES WITH *<constraint>*** 부분에서는 마이닝 규칙들의 세부사항을 나타내는 부분이다. 연관 규칙의 경우 Support, Confidence를 그리고 일반화 과정에서의 한계치를 나타내는 Threshold를 두어 세부사항을 표시하도록 하였다.

특히, 기존 연구와의 차이로서 본 논문에서 제시하는 질의 형태는 기존의 데이터 마이닝 질의 언어[6] 보다 일반 사용자가 자주 사용하는 형태의 일반 SQL 데이터 마이닝 질의를 제시하고 있다는 사실이다. 한 예로서 질의에서 직접적인 "select from where" 절의 사용은 일반 사용자에게 친근하게 인식될 수 있기 때문이다. 이와 같은 결과는 사용자에게 친밀감을 주게되며, 질의 처리에 기존에 연구가 이루어져왔던 축적된 기술을 활용할 수 있는 장점을 가지기 때문이다.

다음은 DMQL의 다양한 예제 질의를 통해 데이터 마이닝 질의 언어의 사용 예를 살펴본다.

예 1) Association rule의 경우

```
Mine rule association as gpa&birth_place
select gpa, region.name
from student, region
where major = "CS" and birth_place = "서울"
extracting rules with
support : 0.05, confidence : 0.7
```

상기 예제는 연관 규칙을 구하는 데이터 마이닝 질의로써 "전공이 컴퓨터공학(CS)인 학생의 GPA와 서울 지역과의 연관 규칙을 구하라"라는 질의이다. 마이닝 규칙을 질의 처음에 명시하였으며, 연관 규칙의 대상이 되는 조건을 WHERE 절에서 지정하였다. 또한 연관 규칙에서 필요 되어지는 세부 항목인 support, confidence등을 명시하였다.

예 2) Discriminant rule의 경우

```
Mine rule discriminant as "precipitation : Mountain vs. Sea"
using hierarchy climate for precipitation
(for "B.C." with region Mountain
vs "Alberta" with region Sea)
select precipitation, area_name
from weather_probe
where time_period = "May" and year ="1997"
extracting rules with
threshold : 0.4
```

위의 예제는 "1997년 5월 기후에 대해 산간 지역과 바다 지역의 기후를 비교하라"라는 질의이다. 상기 예제에서는 개념 계층의 사용을 지정하였고 Discriminant rule의 rule spec을 나타내었으며, 비교시 고려되어야 하는 속성들을 SELECT 절에 명시하였다. 또한 한계치인 Threshold를 명시하였다.

예 3) Characteristic rule의 경우

```
Mine rule characteristic as "서울 여름 기후의 특성"
using hierarchy climate for temperature
using hierarchy climate for precipitation
```

```
select temperature, precipitation
from weather_probe, region r1
where time_period ="summer" and year =1996
and r1.name ="서울"
extracting rules with
threshold : 0.4
```

상기 예제는 "서울의 1996년 여름 기후의 특성을 구하시오"라는 질의이다. 이전의 예와 마찬가지로 마이닝 규칙의 종류를 명시하였으며, temperature, precipitation등의 개념 계층을 나타내었고 한계치 Threshold를 마이닝 규칙의 세부사항을 나타내는 곳에 명시한 질의를 나타내고 있다.

예 4) Classification rule의 경우

```
Mine rule classification as "region"
select crimes10000, region_name
from region r1
where r1.name="부산"
extracting rules with
threshold : 0.3
```

상기 예제는 Classification rule을 구하는 공간 데이터 마이닝 질의의 예로써 "부산을 인구 10000명당 범죄율로 해당 지역의 Classification rule을 얻어라"라는 질의이다. Classification rule을 구하는 질의임을 명시하였으며, 한계치인 Threshold를 주어서 질의를 수행하는 예제를 보이고 있다.

예 5) Clustering rule의 경우

```
Mine rule clustering as temperature
using hierarchy climate for temperature
select temperature, region.geo
from weather_probe, region
where region.state = "대한민국"
```

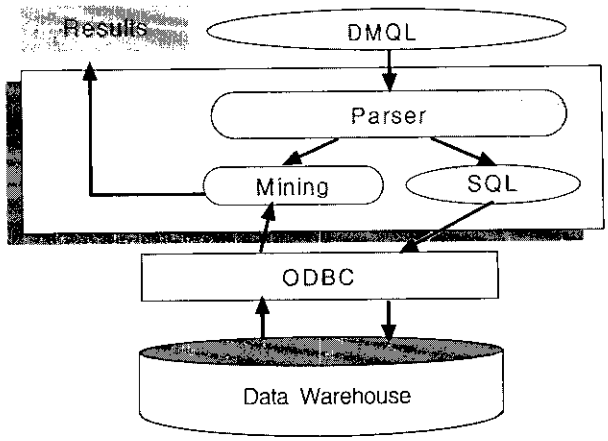
위 예제는 Clustering rule을 구하는 공간 데이터 마이닝 질의의 예로써 "대한민국의 기후를 가지고 Clustering을 수행하라"라는 질의이다. 마이닝 규칙의 종류를 나타낸 후 질의에 사용될 개념 계층의 이름을 표시하였고 Clustering 해야할 속성들을 SELECT 절에 명시하였다.

상기 열거한 여러 마이닝 규칙 예제들은 마이닝 질의를 수행하기 위한 예로서 나열하였으며, 데이터 웨어하우스의 구조와는 별개로 이루어지는 부분이다.

4. 데이터 마이닝 질의 처리기 설계

4.1 데이터 마이닝 질의 처리 프레임워크

본 절에서는 데이터 마이닝 질의 언어 처리기(data mining query processor, DMQP)를 위한 전체적인 프레임워크에 관하여 살펴본다. 먼저, 전반적인 흐름을 그림으로 도시해보면 (그림 1)과 같다.



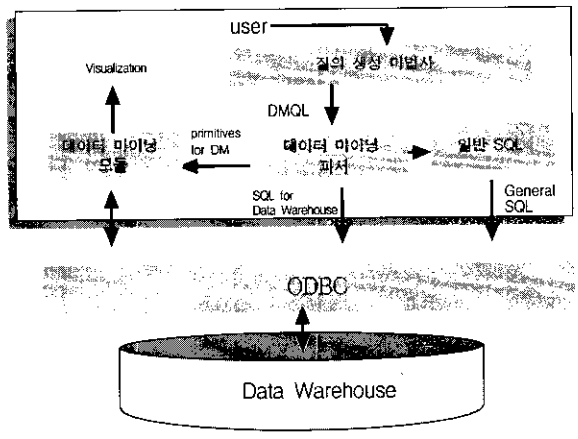
(그림 1) DMQP의 전체 흐름도

(그림 1)에서 데이터 마이닝 질의 언어(data mining query language, DMQL)는 사용자가 그래픽 사용자 인터페이스를 통하여 작성한 마이닝 질의 언어이다. 작성된 질의 언어는 질의 처리기의 파서를 통하여 크게 두 부분으로 구분되어 처리된다. 하나는 "select from where"와 같은 일반 SQL 언어 부분을 추출하여 데이터 웨어하우스에 질의를 수행하는 부분이다. 이 부분에서는 데이터 마이닝을 수행하기 위한 데이터 셋을 찾는 단계이다.

다른 부분은 데이터 웨어하우스로 부터 얻어온 데이터 셋 결과를 대상으로 마이닝을 수행하는 단계이다. 이 부분에서는 마이닝 처리를 위한 변환 단계를 거쳐서 마이닝을 수행하게 된다. 변환 단계는 일반 데이터 값의 의미를 변형시키지 않고 마이닝 처리에 적절하게 변환하는 것이다. 변환된 데이터는 파서를 통하여 임시로 저장된 데이터 마이닝 정보를 이용해서 수행하며, 이로써 사용자가 원하는 데이터 마이닝 결과를 얻을 수 있다.

4.2 데이터 마이닝 질의처리기 구성

상기 데이터 마이닝 질의 언어 처리기의 처리 과정을 세부적으로 살펴보기 위해서 기능별로 질의 처리기의 구조를 구성하여 (그림 2)에 도시하였다.



(그림 2) 데이터 마이닝 질의처리기 구조

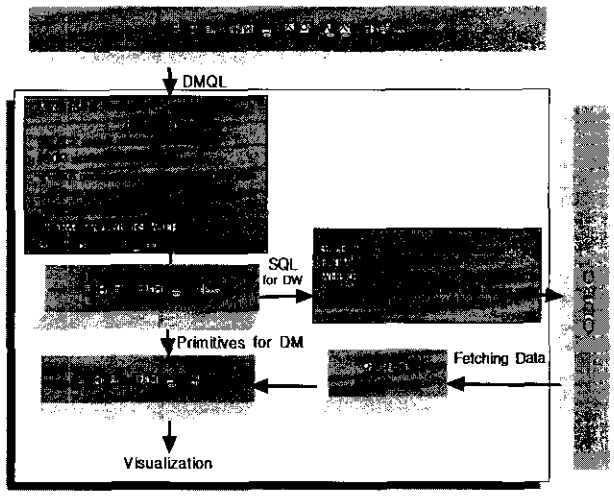
DMQP는 크게 질의 생성 마법사, 데이터 마이닝 파서, 일반 SQL 부분과 데이터 마이닝 모듈들로 구성된 데이터 마이닝 질의 처리 모듈들로 구성되어 있으며, 사용자는 데이터 마이닝을 하기 위한 질의를 DMQP를 통해 수행하고 그 결과를 테이블 형태로 얻을 수 있다. 다음은 각 구성 모듈들의 수행 역할에 관하여 살펴본다. 이해를 돕기 위해서 데이터 마이닝 유형들 중 대표적으로 언급되는 연관 규칙 데이터 마이닝 질의 처리를 예로써 살펴본다.

4.2.1 질의 생성 마법사

이 모듈은 데이터 마이닝 및 데이터 웨어하우스에 대한 질의를 수행할 수 있도록 하는 구문 생성 모듈로써 사용자가 데이터 웨어하우스를 기반으로 한 데이터 마이닝 질의를 수행할 수 있도록 질의를 생성해준다. 실제로 데이터 마이닝 질의를 일반 사용자가 직접 작성하기가 복잡하고 곤란하므로 이를 GUI를 이용하여 쉽게 마이닝 질의를 생성해주는 모듈이다. 또한 사용자가 GUI를 통하여 작성한 일반 데이터 접근 질의는 마이닝 질의 생성 인터페이스를 통하여 동일하게 수행된다. 그러나 질의 생성 마법사에서 데이터 웨어하우스로 전달되는 정보는 일반 SQL 데이터 처리 부분인 "select from where"절 부분을 추출하여 데이터 웨어하우스에 넘겨주어 처리되며, 마이닝과 관련된 "rule, support, confidence"등의 부분은 마이닝 질의 처리기에서 관리가 이루어지게 된다. 실제 동작 내용은 5장에서 살펴본다.

4.2.2 데이터 마이닝 파서

파서 모듈을 세부적으로 살펴보면 (그림 3)과 같다.



(그림 3) 세부적인 마이닝 파싱 모듈

데이터 마이닝 파서는 사용자가 입력한 데이터 마이닝 질의 언어를 파싱 과정을 통하여 데이터 마이닝 모듈에 필요한 마이닝 primitive 들과 학습에 필요한 데이터 셋을 데이터 웨어하우스로부터 얻기 위한 일반 질의 SQL for DW로 분류한다. SQL for DW는 ODBC를 통해 데이터 웨어하우스 서버로 전달되어 관련 데이터 집합을 얻어서 데이터 마이닝 모듈로 전달된다. 이와 같이 웨어하우스로부터 얻어온 데이터 셋은 다양한 형식의 데이터 셋이므로 마이닝 모듈에 적용하기 위해서 인코딩 단계를 거치게 된다. 인코딩 및 디코딩은 데이터 마이닝 모듈에 적용하기 전에 필요하며, 마이닝을 수행한 이후 사용자에게 마이닝 결과를 보여줄 때 필요하므로 인코딩 모듈의 기능을 데이터 마이닝 모듈에 포함하여 처리하였다

4.2.3 데이터 마이닝 모듈

데이터 마이닝 모듈에서 인코딩은 마이닝의 처리효율을 높이기 위해 bit vector로 매핑하는 과정을 적용하였으며, 인코딩된 bit vector 들에 대해 마이닝 적용을 원활하게 하기 위해 다양한 bit-wise 셋 연산자들을 정의하여 마이닝을 수행하였다. 실제로 bit vector들로 마이닝 적용 과정을 다음 절에서 살펴보도록 한다.

4.3 연관 규칙을 발견하는 질의 처리기 설계

다음 <표 1>은 데이터 웨어하우스에서 가져온 소스 데이터 셋 결과라고 가정한다.

<표 1> 소스 데이터 셋

Record_ID	Age	Married	NumCars
100	23	NO	1
200	25	YES	1
300	29	NO	0
400	34	YES	2
500	38	YES	2

이때 트랜잭션은 <100, 23, NO, 1>로서 하나의 레코드로 정의하며, 아이템을 <Age, 20…29>,<Married, No>, <NumCars, 1>과 같은 형태로 모든 아이템 집합들을 정의한다. 이러한 정의에 따라서 <표 1>의 데이터 셋을 인코딩한 결과로 <표 2>와 같은 결과를 얻을 수 있다.

<표 2> 인코딩된 데이터셋

RecID	Age:20…29	Age:30…39	Married:Yes	Married:No	NumCars:0	NumCars:1	NumCars:2
100	1	0	0	1	0	1	0
200	1	0	1	0	0	1	0
300	1	0	0	1	1	0	0
400	0	1	1	0	0	0	1
500	0	1	1	0	0	0	1

이와 같이 데이터 셋을 인코딩한 후 아이템 셋을 구한다. 아이템 셋도 bit vector로 매핑된다. 그 예로 {<age : 20…29>}는 (1000000)으로, {<age : 30…39>, <Married : Yes>}는 (0110000)으로 매핑 될 수 있다. 다음은 아이템 셋을 구하기 위한 알고리즘이다. 이 알고리즘은[10] 가장 큰 아이템 셋을 구하기 위한 알고리즘을 설명한 것이다.

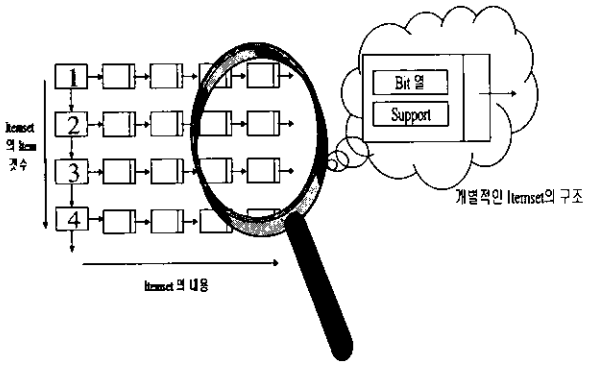
① Itemset의 item의 개수가 1인 것을 구한다.
- 각각이 Minimun support 이상이 되는 것을 선택한다.

② ①에서 선택된 item의 개수가 1개인 itemset을 bit-wise set union 하여 item이 2개인 itemset을 생성한다.
- 각각이 Minimun support 이상이 되는 것을 선택한다.

③ ②에서 선택된 item 이 2개인 것끼리 계산하여 3개의 item을 갖는 itemset을 생성한다.
- 2개의 item을 갖는 itemset을 선택하여 bit-wise set intersection을 계산하여 bit 가 1인 것의 개수가 1인 것을 찾는다.
- 만일 찾았으면,
  그 2개의 itemset을 bit-wise set union 하여 item의 개수가 3개인 itemset을 얻는다.
- 각각이 Minimun support 이상이 되는 것을 선택한다.

④ 같은식으로 k 번째에,
- k개의 item으로 구성된 itemset에서 2개의 itemset을 선택하여 bit-wise set intersection을 계산하여 bit 가 1인것의 개수가 k-1인 것을 찾는다.
- 만일 찾았으면,
  그 2개의 itemset을 bit-wise set union 하여 item의 개수가 k+1개인 itemset을 얻는다.
- 각각이 Minimun support 이상이 되는 것을 선택한다.

위의 알고리즘을 통해 얻은 아이템 셋의 집합은 (그림 4)와 같은 저장구조로 저장된다.



(그림 4) itemset 의 저장 구조

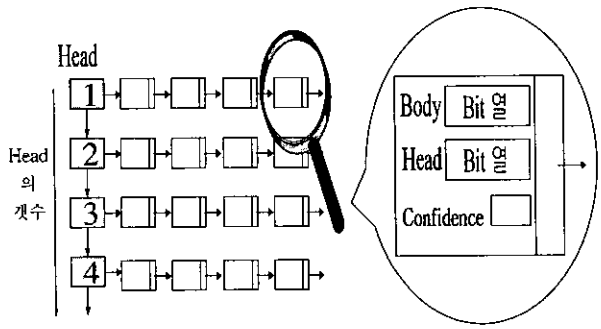
아이템 셋의 저장 구조에 관한 부분은 본 논문의 범위를 벗어나기 때문에 자세한 사항은 생략하였다. (그림 4)의 구조를 이용하여 Minimum confidence 이상이 되는 연관 규칙을 찾는다. 이 때 다음과 같은 방법을 이용한다.

① 위에서 구한 itemset의 집합에서 임의의 2개의 노드를 선택하여 각각 A, B 로 둔다.
② A 와 B를 bit-wise set union 하여 n({A,B})인 것의 support를 구한다.
③ 다음 조건을 만족하는 것을 $A \Rightarrow B$인 연관규칙에 포함한다.

A.IntersectWith(B) = 0 // $A \cap B = \varnothing$

{n({A,B}) / n(A)} * 100 > Min_Confidence

상기 과정을 거쳐서 생성된 규칙은 (그림 5)와 같은 저장 구조로 저장된다. 생성된 최종 연관 규칙들은 디코딩 과정을 거쳐 사용자에게 보여주게 된다.



(그림 5) 연관규칙의 저장 구조

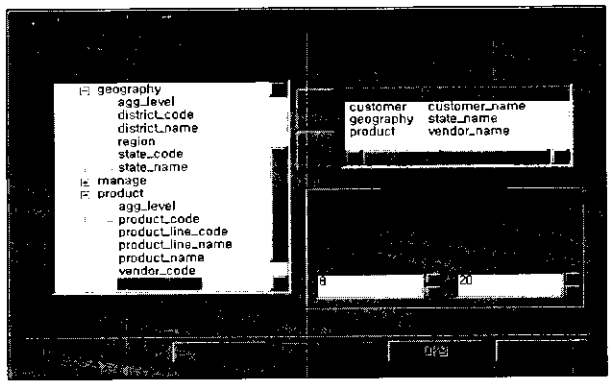
5. 구 현

본 장에서는 이전에 살펴본 데이터 마이닝 질의 처리기의 구축을 보여주기 위하여 연관 규칙을 발견하기 위한 데이터 마이닝 질의 수행에 대해서 살펴본다. 데이터 마이닝 방법중 연관 규칙을 제외한 다른 방법도 유사한 방법으로 구축할 수 있기 때문에 본 논문에서는 생략하였다.

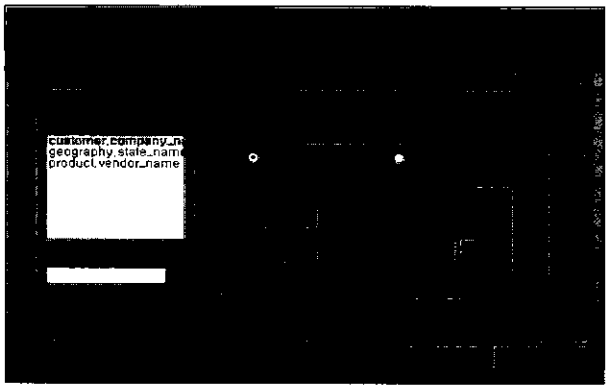
5.1 연관 규칙 질의 처리기의 구현

연관 규칙을 발견하기 위한 질의 처리기는 먼저, 데이터 웨어하우스를 선택 및 접속한 이후 인증 과정을 거치게된다. 인증 과정을 끝마친 다음에 스키마 생성기를 통하여 데이터 웨어하우스를 생성한다[11]. 상기 과정은 질의 처리를 수행하기 위한 데이터 웨어하우스 구축 단계로서 여기에서는 본 과정을 생략한다.

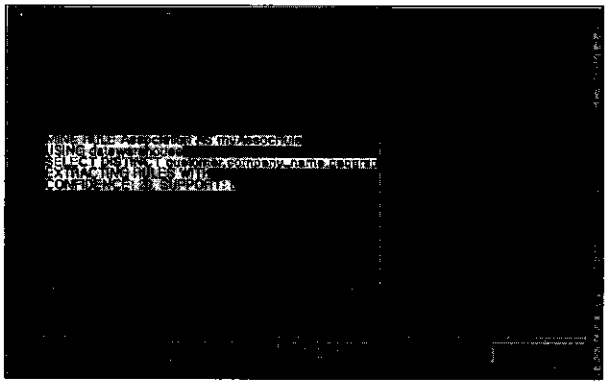
먼저, 데이터 웨어하우스에 대해 마이닝 질의를 수행하기 위해서 사용할 애트리뷰트들을 선택하며, Support와 Confidence를 입력하는 그림을 (그림 6)에 도시하였다. (그림 7)은 마이닝 알고리즘을 수행하기 위해서 선택된 컬럼들에 대해 인코딩 단계를 그림으로 도시한 것이다.



(그림 6) 마이닝 대상 애트리뷰트 선택

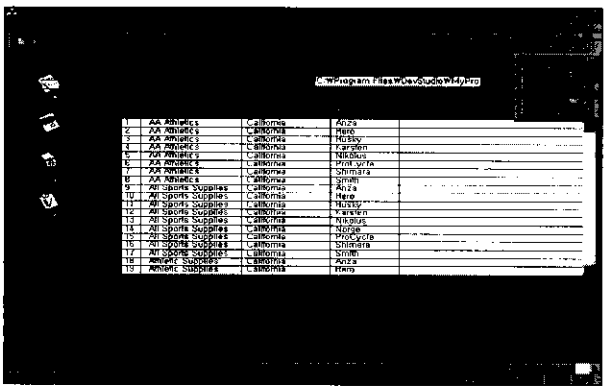


(그림 7) 애트리뷰트의 인코딩 적용



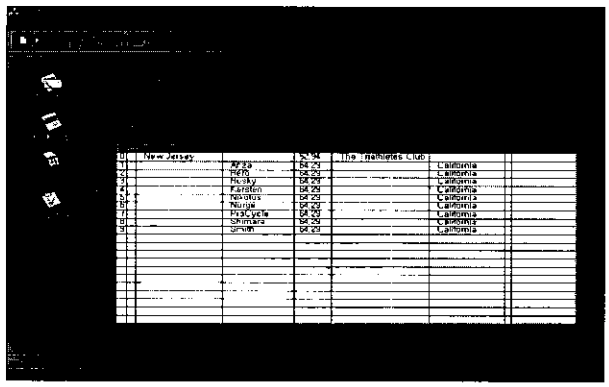
(그림 8) 마이닝 질의 생성 윈도우

상기 과정을 통하여 (그림 8)과 같은 마이닝 질의를 생성할 수 있게 된다. 마이닝 질의 생성 마법사를 이용하여 생성된 데이터 마이닝 질의 언어는 내부적으로 파싱 과정을 통해 마이닝 모듈에 적용하기 위한 마이닝 primitive 들과 질의와 관련되어 데이터 셋을 가져오기 위한 데이터 웨어하우스 SQL로 분류된다. 이때 데이터 웨어하우스 SQL은 ODBC를 통하여 데이터 셋을 가져오게 되며, 데이터 셋은 (그림 9)에서 보이는 List Control을 이용하여 볼 수 있다.



(그림 9) 마이닝에 적용할 데이터셋

마이닝에 적용할 데이터 셋이 정해지면 마이닝 모듈을 적용하기 위한 단계중 "학습" 아이콘을 선택하여 마이닝 모듈을 적용하게 된다. 여기서는 여러 가지의 트랜잭션들 중에서 동시에 발생하는 트랜잭션의 연관 관계를 발견하는 연관 규칙의 수행에 관한 메시지가 나타날 것이다. 또한 마이닝 모듈을 적용할 때 "Learning"이라는 메시지 박스를 띄워 현재 마이닝이 진행되고 있음을 보여준다. 마이닝 작업이 종료되면 "Learning"박스가 사라지면서 마이닝 결과를 볼 수 있게 된다



(그림 10) 연관 규칙을 적용한 마이닝 결과

위와 같은 학습 과정을 통해 최종적으로 다양한 연관 규칙을 얻을 수 있으며, 이렇게 얻은 연관 규칙들에 관한 사항은 (그림 10)과 같은 결과를 얻을 수 있다.

6. 결 론

정보 기술의 발전은 기업들로 하여금 많은 양의 데이터를 기업내부에 축적할 수 있도록 하였다. 이와 같이 축적된 데이터는 데이터베이스 기술과 데이터 웨어하우스라는 기술적 환경에서 활용되며, 대규모의 데이터 집합을 효율적으로 저장하고, 검색하는 기본적인 도구를 제공하였다. 그러나 축적된 데이터로부터 기업의 경쟁력을 강화시킬 수 있는 정보를 획득하기에는 많은 요구사항이 따른다. 이러한 요구사항을 충족시켜주는 새로운 정보 기술의 활용 방법이 데이터 마이닝 기술이다. 즉, 데이터 웨어하우스가 대량의 데이터를 효과적으로 유지 및 관리하기 위해 등장했다면 데이터 마이닝은 그 집적된 데이터의 활용을 높일 수 있는 방법이다.

데이터 마이닝 질의 언어에 관한 연구는 데이터 요약, 연관 규칙, 패턴 탐색 등의 다양한 데이터 마이닝 기능을 포괄적으로 표현할 수 있는 질의 언어와 구문의 설계 방법론 및 질의 처리기의 구현이 요구된다. 이에 데이터 웨어하우스와 데이터 마이닝은 새로운 연구분야를 제시하였으며, 이를 위해 새로운 개념과 방법들을 이용한 질의어, 기본적인 연산과 질의처리 전략들이 필요하게 되었다.

본 논문에서는 기존의 데이터 마이닝 질의 언어를 변형하여 데이터 웨어하우스를 기반으로 수행되는 데이터 마이

닝 질의 언어에 대한 새로운 방향을 제시하고 질의를 수행할 수 있도록 하였다. 이를 위해서 먼저, 데이터 웨어하우스에서 마이닝을 수행하기 위한 데이터 선택 및 질의를 생성할 수 있는 구축 환경을 제시하였다. 또한, 본 논문에서는 여러 데이터 마이닝 유형중에서 연관 규칙을 발견할 수 있는 데이터 마이닝 질의 처리기를 설계 및 구현하였다.

참 고 문 헌

[1] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Advances in knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.

[2] Alex Berson, Stephen J. Smith. Data Warehousing, Data Mining, & OLAP. McGraw-Hill, 1997.

[3] I. Mumick, D. Quass, and B. Mumick. "Maintenance of data cubes and summary tables in a warehouse," In Proceedings of the ACM-SIGMOD Conference, Tucson, Arizona, 1997.

[4] Rosa Meo, Giuseppe Psaila, Stefano Ceri, "A New SQL-like Operator for Mining Association Rules," in Proceedings of the 22nd VLDB Conference, pp.122-133, 1996.

[5] Jiawei Han, Jenny Y. Chiang, Sonny Chee, Jianping Chen, Qing Chen, etc. "DBMner : A System for Data Mining in Relational Database and Data Warehouses," in URL : http : //db.cs.sfu.ca/(for research group)

[6] Jawei Han, Yongjian Fu, Wei Wang, Krzysztof Koperski, Osmar Zaiane, "DMQL : A Data Mining Query Language for Relational Databases," in URL : http : //db.cs.sfu.ca/(for research group)

[7] G. Piatetsky-Shapiro, and W. J. Frawley, Knowledge Discovery in Databases. AAAI/MIT Press 1991.

[8] W. H. Inmon "The Data Warehouse and Data Mining," In Communications of the ACM, November 1996/Vol. 39. No. 11 page 49-57.

[9] Tomasz Imielinski, Heikki Mannila "A Database Perspective on Knowledge Discovery," In Communications of the ACM November, 1996/Vol.39. No.11 page 58-64.

[10] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," Proc. 20th Int'l Conf. Very Large Data Bases, pp.487-500, Sept. 1994.

[11] 정병화, 이현창, 김경창, "데이터 웨어하우스를 위한 스키마 생성기 설계 및 구현", 한국 정보과학회, 제25권 1호, 봄 학술발표논문집, 1998.

[12] A. Gupta, V. Harinarayan, D. Quass, "Aggregate-Query Processing in Data Warehousing Environments," Proc 21st Int'l Conf. Very Large Data Bases, pp.358-369, 1995.

김 충 석

e-mail : cskim@silla.ac.kr

1986년 홍익대학교 전자계산학과 졸업 (이학사)

1988년 홍익대학교 전자계산학과 졸업 (이학석사)

1993년 홍익대학교 전자계산학과 졸업 (이학박사)

1990년~현재 신라대학교(구.부산여자대학교) 컴퓨터 정보공학부 부교수

관심분야 : Distributed Object Processing, Intranet Solution, OOP

김 경 창

e-mail : kckim@cs.hongik.ac.kr

1978년 홍익대학교 전자계산학과(학사)

1980년 한국과학기술원 전산학과(석사)

1990년 University of Texas at Austin 전산학과(박사)

1991년~현재 홍익대학교 정보컴퓨터공학부 부교수

주관심분야 : 객체지향 데이터베이스, 주기억 데이터베이스, OLAP 및 데이터 웨어하우징, Web 데이터베이스