논문 01-26-7A-10 한국통신학회논문지 '01-7 Vol.26 No.7A

공간 데이터 마이닝에서의 질의 처리 최적화 전략

정회원 김 충 석*, 준회원 이 현 창**, 정회원 김 경 창***

Query Optimization Infrastructure in Spatial Data Mining

Choung-Seok Kim* *Regular Member*, Hyun-Chang Lee** *Associate Members*

Kyung-Chang Kim*** *Regular Member*

요 약

최근 각광을 받고 있는 데이터 마이닝 분야에서 데이터 마이닝 툴과 시스템의 등장으로 상호적이고 사용하기 쉬운 GUI 환경의 강력한 데이터 마이닝 질의 언어가 필요하게 되었다. 공간 데이터 마이닝은 공간 데이터에서 유용한 지식을 발견하기 위한 데이터 마이닝의 한 부분이며 공간 데이터는 점, 선, 사각형, 다각형 등으로 이루어져 있다. 공간 데이터 마이닝은 지리정보시스템(GIS)과 더불어 최근에 많은 관심과 연구가 활발히 진행되고 있다. 한편, 공간 데이터 마이닝을 위한 질의 언어와 그 언어에 기반한 공간 데이터 마이닝 질의 처리 및 최적화에 대한 연구가 중요하게 대두되고 있다. 공간 데이터에 대한 마이닝은 일반 관계형 데이터베이스에서의 질의 언어로는 표현이 불가능하다. 본 연구에서는 먼저 공간 데이터 마이닝 질의언어를 정의, 설계하고 질의 언어에 결과 표현 방식과 결과 데이터 집합의 저장을 명시하여 질의 표현의 효율을 높이는 방식을 제시하였다. 또한 공간 데이터 마이닝을 위한 질의 처리 및 최적화 과정을 질의에 기반한 공간 실체화 뷰 의 생성과 유지, 인덱스 활용을 통한 질의 재사용, sampling 마이닝 질의 option 등의 방법론을 이용하여 제시하였다.

ABSTRACT

Data mining is an area that has received a lot of attention recently. The emergence of data mining tools and systems requires a powerful data mining query language that is easy to use and that supports GUI capabilities. Spatial data mining is to discover useful knowledge among spatial data that is comprised of point, line, rectangle, polygon, and other spatial features. Spatial data mining together with Geographic Information System (GIS) are areas dealing with spatial data that has received a lot of attention and where research is active. Data mining of spatial data is not easy to express using the query language provided by existing relational DBMS. Hence, the design of a query language for spatial data mining and query processing and optimization of spatial data mining query based on the query language are important. In this paper, we first define and design a spatial data mining query language that includes presentation of result and specification of result data set storage to enhance the efficiency of queries. In addition, the paper proposes query optimization infrastructure for spatial data mining queries by using optimization strategies like maintenance of materialized view based on spatial query, spatial query reuse through index, and sampling of spatial query.

* 신라대학교 컴퓨터 정보공학부(cskim@silla.ac.kr)
** 경인여자대학 멀티미디어정보전산학부 전임강사(hclee@dove.kyungin-c.ac.kr)
*** 홍익대학교 컴퓨터공학과(kckim@cs.hongik.ac.kr)
논문번호 : 010072-0418, 접수일자 : 2001년 4월 18일
※ 본 연구는 한국과학재단의 특정기초 연구비 지원(과제번호 : 97-0102-04-01-03)을 받았음

Ⅰ. 서 론

최근에 데이터 마이닝과 데이터 웨어하우징 분야의 연구가 빠른 속도로 진행함에 따라 많은 데이터 마이닝 시스템들이 관계형 데이터베이스 및 데이터 웨어하우스에서 지식을 발견하기 위해 개발되었다. 최근 정보기술의 발달로 실생활에서 공간 데이터베이스의 활용과 중요도가 증가하면서 공간 데이터로부터 유용한 지식 또는 정보를 추출하는 공간 데이터 마이닝이 대두되고 있다. 따라서 공간 데이터 마이닝을 위한 질의 언어와 그 언어에 기반한 공간 데이터 마이닝 질의 처리 및 최적화에 대한 연구가 이루어지게 되었다.

공간 데이터 마이닝[17]은 공간 데이터베이스에 저장된 흥미로운 패턴이나 공간 관계 등의 함축적인 지식의 추출을 하기 위한 데이터 마이닝의 한 부분이다[1]. 이러한 공간 데이터 마이닝을 하기 위해서는 공간 데이터에 대한 비교 연산자의 지원, 공간 데이터의 특성을 표현하고 이와 연관된 연산자의 표현 등을 지원할 수 있는 공간 데이터 마이닝 질의 언어가 필요하다. 이러한 질의 언어 설계 시 기존의 SQL을 확장하는 방식을 취하는 이유는 사용자에게 친밀감을 느끼게 하고, 질의 처리에 축적된 기술을 이용할 수 있기 때문이다[2].

공간 데이터 마이닝 질의 언어(SDMQL:Spatial Data Mining Query Language)는 데이터 마이닝에서 필요한 공간 연산자와 공간 관계를 표현하여야 하며 마이닝 규칙들의 집합을 정하고 그들 각각의 규칙들은 마이닝을 위해 다양한 인자들이 필요한데, 이들에 대한 세부적인 사항을 정의해야 한다. 또한 공간 데이터 마이닝의 결과를 사용자에게 보여주는 표현 방식을 질의 언어에 추가하고 사용자의 편이성 및 질의 표현의 형태를 미리 시스템에게 알려줌으로써 불필요한 결과의 계산으로 인한 성능 저하를 방지하고 질의 결과의 효율적인 사용을 할 수 있도록 해야한다. 본 논문에서는 SDMQL의 전체적인 구조를 정의하였고 실제로 마이닝을 위해 적용 가능한 모든 규칙들을 개별적으로 정의하였다.

질의 성능을 향상시키는 효율적인 질의 처리를 위하여 질의 최적화 전략이 반드시 필요하다. 따라서 공간 데이터 마이닝 질의 처리 시에도 질의 최적화 전략이 필요하다. 공간 데이터 마이닝 질의의 특성을 고려한 공간 마이닝 질의 최적화 측면에서는 실체 뷰를 통한 데이터의 재사용성과 공간 데이터 마이닝을 보다 빠르고 신속하게 수행할 수 있는 인덱스 활용을 통한 기반 데이터를 생성하기 위해 노력하여야 한다. 본 논문에서는 실체 뷰와 인덱스를 활용한 공간 데이터 마이닝 질의 최적화 전략을 아울러 제시하였다.

본 논문의 구성은 다음과 같다. 먼저 2장에서는 관련 연구를 언급하고 3장에서는 SDMQL의 설계에 대해서 자세히 설명한다. 4장에서는 공간 데이터 마이닝 질의 처리 및 최적화에 대해서 언급하고 5장에서는 실제 예제를 통한 질의 처리 및 최적화 과정을 보인다. 마지막으로 6장에서는 결론을 맺는다.

Ⅱ. 관련 연구

대표적인 데이터 마이닝 질의언어로는 기존의 SQL에 데이터 마이닝의 개념을 표현할 수 있는 DMQL(Data Mining Query language)[3]이 있으며 공간 연산자, 공간 관계성 표현자, 질의 방식을 나타내는 Spatial SQL[4]은 공간 데이터 마이닝 질의 언어의 기본적인 요소들을 내포한다. 본 논문에서 제시하는 SDMQL은 DMQL과 Spatial SQL을 확장하여 설계하였다.

DMQL에서는 관계형 데이터베이스 상에서 상호작용(interactive)하며 마이닝을 수행할 수 있도록 제공하기 위한 언어로 마이닝에 필요한 데이터 집합을 조회할 수 있는 기능, 지식을 발견하기 위한 마이닝 방법 기술, 데이터 마이닝 처리를 위한 배경 지식(conceptual hierarchy information)의 도입, 각 해당 마이닝 기술마다 필요한 추가적인 옵션 설정 기능을 표현한다. 따라서 SQL-like한 문법을 제공한다고 하지만 위와 같은 많은 기능들을 표현해야 하기 때문에 사용자가 질의를 표현하기도 어렵고 또한 이해하기도 힘든 단점이 있다.

Spatial SQL은 공간 데이터와 비-공간 데이터를 조회할 수 있는 확장된 SQL과 데이터를 그래피컬(Graphical)한 형태로 표현할 수 있는 전용 언어인 GPL(Graphical Presentation Language)로 구성되어진다. 따라서 질의를 수행하기 전에 디스플레이 환경 모드를 GPL 명령어를 통해 수정한 후, 질의 문장을 수행하게 된다. 따라서 데이터를 조회하는 질의 문장 자체는 SQL과 거의 비슷하지만, 데이터를 표현하기 위한 GPL 명령어를 잘 알아야 한다는 단점이 있다.

공간 데이터 마이닝의 최적화를 위해서는 실체화된 뷰을 이용한 데이터 재사용성과 공간, 비-공간

인덱스를 이용한 질의 결과 데이터의 재사용의 방법들이 있다. 본 논문에서는 공간 인덱스[13,14]와 비-공간 인덱스의 사용에 대해서는 기존의 많은 연구가 있었으므로 언급은 하지 않고 현재 거의 미비한 연구 과제인 공간 실체화 뷰에 관련된 관련 연구만을 살펴보도록 하겠다.

데이터 모델별 실체화된 뷰에 대한 관련 연구로는 크게 관계형 뷰와 객체 지향 뷰가 있다. 관계형 뷰 실체화에 대한 관련 연구[10]에서는 뷰 실체화를 위하여 뷰 테이블에 기본 테이블의 레코드 값을 복사하여 유지하는 방법을 제시하였다. 이러한 관계형 뷰 실체화 방법은 뷰 테이블에서 값들을 직접 저장하기 때문에 질의 처리속도가 빠르다. 그러나 객체 식별자 개념을 지원하지 않기 때문에 기본 테이블의 튜플과 뷰 튜플간에 관련성을 유지하기가 어렵다. 따라서 기본 테이블의 튜플이 변경된 경우에 관련된 뷰 튜플들을 변경하기가 매우 어렵고 기본 테이블 튜플의 변경시 실체화된 뷰를 변경하기 위한 알고리즘이 복잡하고 변경 시간이 오래 걸리는 문제점이 있다.

객체 지향 뷰의 실체화 방법은 크게 2 가지가 있다 [12,16]. 첫 번째 방법은 뷰 객체를 실체화시킬 때 실제 데이터를 저장하는 방식이다. 이 방법은 관계형 뷰 실체화 방법과 유사하며, 뷰 객체에 실제 데이터가 저장되어 있으므로 뷰에 대한 질의 수행시 속도가 빠르다. 그러나 뷰 객체와 소스 객체간에 데이터가 중복되므로 중복된 데이터에 대한 일관성 유지 문제가 발생한다. 두 번째 방법은 실체화된 뷰 객체에 데이터가 아닌 소스 객체의 식별자를 저장하는 방식이다. 이 방법은 객체식별자를 이용하여 공간 뷰 객체와 소스 객체간의 데이터를 공유하며, 데이터의 중복이 없으므로 데이터의 일관성 유지가 쉽다. 그러나 뷰 객체를 검색하기 위하여 소스 객체를 추가로 접근해야 하는 오버헤드가 있다.

따라서 본 논문에서는 검색이 주로 발생하고 변경은 자주 일어나지 않는 공간 데이터베이스와 공간 데이터 마이닝의 질의 특성을 고려하여 자주, 반복적으로 사용되는 데이터에 대해서는 실제 데이터 값을 저장하는 실체화 뷰로써 유지하며, 이전에 사용되었던 유사한 질의가 반복적으로 들어오는 경우를 위해서는 인덱스를 이용하여 객체 식별자만을 저장하는 실체화 뷰들을 생성, 관리함으로 2 가지 방법을 모두 적용하도록 하였다.

Ⅲ. SDMQL의 설계

SDMQL에서 사용되는 연산자는 공간 연산자와 공간 관계성 표현자로 구분된다. 공간 연산자는 일정한 값을 반환하는 반면 공간 관계성 표현자는 True 혹은 False를 반환한다. 예를 들면 두 객체 사이의 거리를 구하는 Distance와 Polygon 객체의 면적을 구하는 Area 등 과 같은 연산자는 일정한 값을 반환하므로 공간 연산자이다. 그러나 두 공간 객체가 인접하는지를 나타내는 Adjacency와 중첩 여부를 알려주는 Overlap등은 True 혹은 False를 반환하는 공간 관계 표현자이다.

지금까지는 공간 연산 측면에서 알아보았고 다음으로 데이터 마이닝 측면에서 보면 다음과 같다. 본 연구에서 고려하는 공간 데이터 마이닝 규칙들은 Association rule, Characteristic rule, Discriminant rule, Classification rule 등이다. 이러한 공간 마이닝 규칙들이 필요로 하는 세부적인 사항들이 있는데 Association rule 경우에는 Support와 Confidence를 필요로 하고 Discriminant rule은 비교의 대상을 명시하는 것을 필요로 한다. 그리고 마이닝 규칙들에서 사용되는 일반화의 한계를 표현하는 한계치(threshold) 또한 필요로 한다[3].

마이닝 규칙의 표현의 상세한 내용은 공간 데이터 마이닝 질의 언어의 BNF 문법형식을 나타낸 후 알아보기로 한다. 질의 결과의 표현 방법은 공간 질의나 비-공간 질의 혹은 공간 데이터 마이닝 질의 등의 결과를 테이블 형식으로 볼 것인지 아니면 지도 형식 혹은 그래프 형식으로 나타낼 것인지를 명시하도록 하였으며 이전에 실행했던 질의 결과와의 관계를 고려한 질의 결과 표현 방식 또한 고려하여 명시하도록 하였다. 지금까지의 내용을 바탕으로 설계된 SDMQL의 EBNF 문법을 보면 다음과 같다.

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

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

<constraint>]

SET RESULT DISPLAY
REPORT TYPE *<report formula>*
DISPLAY TYPE *<display type>*

<mining technique>::=ASSOCIATION | CHARACTERISTIC | DISCRIMINANT |CLASSIFICATION | CLUSTERING
<rule spec>::={FOR <class1>WITH<condition1>{VS<class2>WITH<condition2>}}

<attribute-list>::={attribute|<spatial-operation>|",")
<table-list>::=table name{table name|","}
<condition>::=<non-spatial condition> | <spatial condition> { (and | or) <non-spatial condition>|<spatial condition>}
[ON PICK | WITHIN RANGE]
<spatial condition>::= {<spatial-operation> | (and | or) | <spatial-relationship>}
<spatial-operation>::={Direction|Distance|Fusion|Length|Path|Area|Center}
<spatial-relationship>::={Contain|Contain by|Adjacency|Intersect|Connect|Overlap|Equal}
Intersect ::=Intersect(object1, object2)
Connect::= Connect(object1, object2)
Adjacency ::=Adjacency(object1, object2)
Overlap::=Overlap(object1, object2)
Fusion::=Fusion(object1, object2)
Equal::=Equal(object1, object2)
Contain::=Contain(object1, object2)
Contain by::=Contain by(object1, object2)
Direction::=Direction(point, E|W|S|N|NE|NW|SW|SE)
Distance::=Distance(point1, point2|object, object)
Path::=Path(point1, point2|object, object)
Length::=Length(object)
Area::=Area(object)
Center::=Center(object)

<constraint>::=SUPPORT:<numeric-value>,CONFIDENCE:<numeric-value> | THRESHOLD:<numeric-value>
<report formula>::=<report type>{AND <report type>}
<report type>::=TABLE | MAP | GRAPH
<display type>:=NEW | OVERLAY | REMOVE | INTERSECT | HIGHLIGHT
<save results type>::=YES|NO

위에 나타낸 EBNF 문법을 살펴보면 우선 공간 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 계층을 구축 모듈을 통해 정의되고 Knowledge base에 저장된 특정 비-공간 혹은 공간 속성의 개념 계층을 사용한 질의를 가능하게 하고 있다. 하나의 속성에 여러 개념 계층이 존재 할 수도 있기 때문에 개념 계층의 이름과 속성의 이름을 모두 명시하게 하였다.

예를 들면 강수량에 대한 개념 계층이 존재한다고 할 때 건조한 지역에서의 very wet, wet, moderately-wet, moderately-dry등의 개념 계층이 습한 지역의 것과 같을 수 없기 때문이다. ***<rule spec>***은 마이닝 규칙들에서 필요로 하는 추가 구문에 대한 것으로 여기에서는 Discriminant rule의 경우에만 해당이 된다.

SELECT, FROM, WHERE 절은 기존의 SQL 문의 형식을 따르고 있다. **SELECT** 절에서는 선택하고자 하는 속성이나, 속성에 대한 공간 연산이 적용되고, **FROM** 절에는 테이블의 이름이, **WHERE** 절에는 기존 SQL문에서의 조건 값이나 공간 연산이 적용된 조건 등과 사용자와의 GUI를 통해 상호작용 할 수 있도록 질의를 만들기 위한 **ON PICK, WITHIN RANGE**등에 대해 기술한다. ON PICK은 사용자가 특정 지점을 클릭한 경우이고 **WITHIN RANGE**는 사용자가 입력장치로 지정한 범위 안에 있는 것으로 조건을 주는 경우이다.

EXTRACTING RULES WITH ***<constraint>*** 부분에서는 마이닝 규칙들의 세부사항을 나타내는 부분이다. Association rule의 경우 Support, Confidence를 그리고 일반화 과정에서의 한계치를 나타내는 Threshold를 두어 세부사항을 표시하도록 하였다. 마지막으로 살펴 볼 것은 질의 결과를 표시하는 방법에 대한 것으로써 이런 점을 고려함으로써 사용자는 자신이 원하는 질의 결과보고 형식을 지정

하여 볼 수 있고 시스템 측면에서는 사용자가 원하지 않는 불필요한 결과 표시를 하지 않음으로써 성능의 이점을 볼 수 있을 것으로 기대 된다.

REPORT TYPE ***<report formula>***에서 report formula를 지정하고 *TABLE, MAP, GRAPH*의 종류가 있다. 한 개 이상의 report formula를 지정하여 다양한 표현을 가능하게 하였다. 또한 **DISPLAY TYPE** ***<display type>***절에서는 질의 결과를 이전 결과와 관련지어 연산(NEW, OVERLAY, REMOVE, INTERSECT, HIGHLIGHT)을 수행하여 사용자가 다양한 방법으로 이전 질의 결과와 현재의 질의 결과를 사용할 수 있도록 하였다. 그러면 이제까지 설계된 SDMQL를 다양한 데이터 마이닝 예제 질의를 통해 살펴보겠다.

예 1) Association rule의 경우

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

예 1)은 Association rule을 구하는 공간 데이터 마이닝 질의의 예로써 "전공이 컴퓨터공학(CS)인 학생의 GPA와 범위 안의(WITHIN RANGE) 지역과의 연관 규칙을 구하라"라는 질의이다. 마이닝 규칙을 질의 처음에 명시하였고 Association rule의 대상이 되는 조건을 WHERE 절에서 지정을 했으며, Association rule이 필요로 하는 마이닝 세부사항인 support, confidence등을 명시했다.

예 2) Discriminant rule의 경우

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

예 2)는 Discriminant rule을 구하는 공간 데이터 마이닝 질의의 예로써 "1997년 5월의 기후를 산에 인접하는 지역과 바다에 인접한 지역의 기후를 비교하라"라는 질의이다. 마이닝 규칙 명시한 후 개념계층의 사용을 지정하였고 Discriminant rule의 rule spec을 나타내었고 비교 시 고려가 되어야 하는 속성들을 SELECT 절에 일반화의 한계치인 Threshold를 마이닝 규칙의 세부사항을 나타내는 곳에 명시하였다.

예 3) Characteristic rule의 경우

```
Mine rule characteristic as  "B.C temperature &
precipitation"
using hierarchy climate for temperature
using hierarchy climate for precipitation
select temperature, precipitation
from weather_probe, region r1, region r2
where time_period= "summer"  and year=1996
and r1.name= "서울"  and adjacency(r2, r1)
extracting rules with
threshold:0.4
```

예 3)은 Characteristic rule을 구하는 공간 데이터 마이닝 질의의 예로써 "서울과 인접하는 지역들의 1996년 여름의 기후의 특성을 구하시오"라는 질의이다. 위에서의 예와 마찬가지로 마이닝 규칙의 종류를 명시하였고 temperature, precipitation등의 개념 계층을 나타내었고 일반화의 한계치 Threshold를 마이닝 규칙의 세부사항을 나타내는 곳에 명시하여 질의를 완성하였다.

예 4) Classification rule의 경우

```
Mine rule classification as  "region"
select  crimes10000, region_name
from  region r1, region r2
where r1.name= "충청도" and adjacency(r2, r1)
extracting rules with
threshold: 0.3
```

예 4)는 Classification rule을 구하는 공간 데이터 마이닝 질의의 예로써 "충청도에 인접하는 지역들을 인구 10000명당 범죄율로 해당 지역의 Classification rule을 얻어라"라는 질의이다. Classification

rule을 구하는 질의임을 명시하였고 한계치인 Threshold를 주어서 질의를 수행하도록 하였다.

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 WITHIN RANGE
```

예 5)는 Clustering rule을 구하는 공간 데이터 마이닝 질의의 예로써 "범위 안의 지역의 기후를 가지고 Clustering을 수행하라"라는 질의이다. 마이닝 규칙의 종류를 나타낸 후 질의에 사용될 개념 계층의 이름을 표시하였고 Clustering 해야할 속성들을 SELECT 절에 명시하였다.

Ⅳ. 공간 데이터 마이닝 질의 처리 및 최적화

데이터 마이닝은 반복적이고 지속적으로 시행착오를 겪어가며 어떤 적합한 룰을 찾아내는 과정이므로 공간 데이터 마이닝을 위해서는 사용자가 관련 있을 것 같은 공간 데이터를 찾아야 하는데 이런 작업은 실제로는 쉬운 일이 아니며, 따라서 사용자는 여러 번의 시행착오를 겪게 된다. 그러므로 공간 데이터 마이닝 질의는 이전에 사용되었던 질의가 다음 질의에 많은 영향을 줄 수 있게 된다. 따라서 이러한 점을 기본 아이디어로 공간 데이터 마이닝 질의를 처리하도록 한다. [그림 1]은 공간 데이터 마이닝의 효율적인 질의 처리를 위한 전체 구조를 보여주고 있다.

1. 공간 데이터 마이닝 질의 처리 환경

먼저 공간 데이터 마이닝 질의 처리 최적화를 위한 관련 자료 구조에 대해서 상세히 살펴보도록 하자.

- Access Frequency Table list(AFT)
  > 설명 : 이전에 자주 사용되었던 데이터가 실체뷰(MV)로 생성된 후, 이들 테이블을 유지, 관리하기 위한 자료구조로서 다음에 들어올 질의에 사용될 데이터가 전체 또는 일부가 AFT에 존재한다면, 그 만큼의 데이터를 공간 데이터베이스에서 가져오기 위한 I/O가 필요 없이 바로 사용될 수 있어 성능을 향상시킬 수 있다.

  > 자료구조

TBLname	RowSize	AccCnt

- TBLname : 해당 실체화 뷰 table name

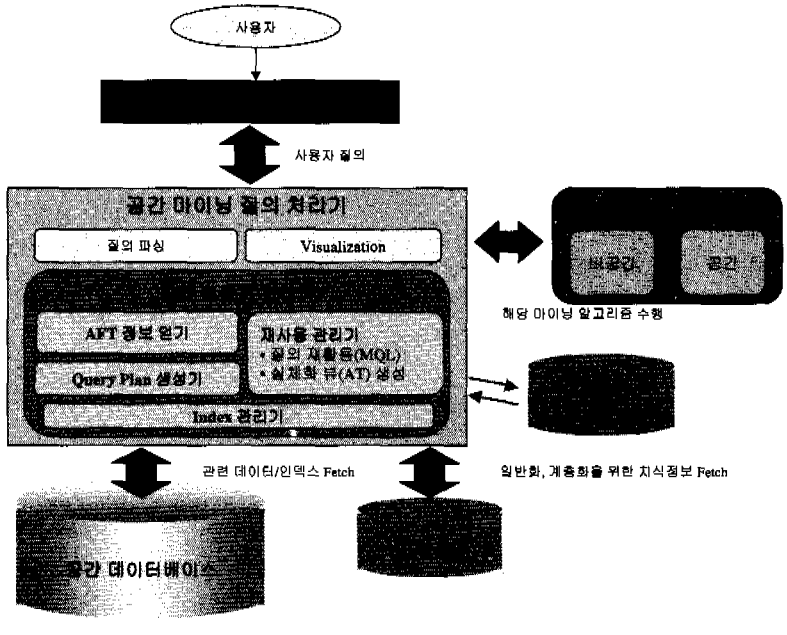

그림 1. 전체 공간 데이터 마이닝 처리 구조

- RowSize : 조건절에 만족한 데이터의 개수
- AccCnt : 실체화 뷰 테이블을 접근한 횟수를 기록하는 필드

● Mining Query list(MQL)

> 설명 : 이전에 사용되었던 질의에 대한 관리를 통해서 데이터의 재 사용성 증대와 자주 재 사용되는 데이터에 대한 실체 뷰의 생성 여부를 판단하기 위한 정보를 관리하는 자료구조이다.

> 자료구조

MQLid	TBLname	조건절	Rel_Size	Sp_size	AccCnt

- MQLid : 들어오는 질의에 대한 Sequence No.로서 최종 결과 테이블 이름으로 사용한다.
- TBLname : 들어온 질의의 From절에 해당하는 table name
- 파싱 조건절 : 들어온 질의의 Where절에 해당하는 조건절
- Rel_Size : 만족하는 릴레이션 레코드 개수
- Sp_size : 만족하는 공간 id 개수
- AccCnt : TBLname을 접근한 횟수를 기록하는 필드로 나중에 해당 자료구조를 관리하는데 사용된다.

● Index list

> 설명 : AFT나 MQL에서 일부 데이터가 공간 마이닝 서버에 없어서 공간 데이터베이스에 접근하여 해당 데이터를 가져오는데, 이때 해당 필드에 대한 인덱스가 없다면 인덱스를 생성하여 공간 마이닝 서버로 가져오고, 인덱스가 존재한다면 바로 공간 마이닝 서버로 가져와서 index list 구조에 저장한다. 이렇게 저장함으로써 다음에 이 데이터가 필요시 해당 인덱스만을 사용하여 질의 플랜을 생성할 수 있어 성능을 향상시킬 수 있다.

> 자료구조

TBLname	Key Field	Index_list_id	Index_Struct

1) Spatial data : R/R*tree 계열
1) Non-spatial data : B tree 계열, Bitmap index

- TBLname : 들어온 질의의 조건 속성 필드가 있는 table name
- Key Field : 들어온 질의의 조건 속성 필드명
- Index_list_id : 질의 수행 후, 색인 결과만을 가져와서 저장해 두기 위한 화일이름이다. 질의수행 결과가 매우 크기 때문에 메인 메모리에 모두 저장할 수 없기에 질의처리기 서버의 로컬 화일/DB로 저장해 두었다가 사용한다.
- Index_Struct : 1) Spatial data : R/R* tree 계열
  2) Non-spatial data : B tree 계열, Bitmap index

사용자가 입력한 공간 마이닝 질의문을 효율적으로 처리하기 위한 모듈로서 크게 두 부분으로 나눌 수 있다. 선행 처리 부분은 입력된 질의문을 파싱하여 AFT를 참고하여 이전에 생성된 실체 뷰와 비교함으로써 이전에 사용했던 데이터를 들어온 질의가 재사용할 수 있는 지를 검사한다. 만약 재사용할 수 없다면 효율적으로 질의를 실행할 플랜[5]을 작성한 후, 공간 데이터베이스에 접근하여 필요한 데이터를 가져오는데, 이때 속성 필드에 인덱스가 없다면 인덱스를 생성하고 인덱스 정보를 인덱스 관리기에서 index list를 통하여 유지, 관리한다.

후행 처리 부분에서는 실체 뷰에서 제공할 수 있는 관련 데이터가 없기 때문에 질의 결과를 저장하고 있는 MQL을 참고하여 이전 질의에서 사용되었던 데이터를 재 사용할 수 있는 지를 검사한다. 만약 들어온 질의가 이전 데이터의 전체 영역에서 처리 가능하다면, 공간 마이닝 질의 서버에서 처리함으로써 공간 데이터베이스를 접근할 필요가 없으며, 이전 데이터의 일부 영역만을 들어온 질의가 사용될 때에는 필요한 데이터 필드만을 공간 데이터베이스에서 가져오며 이때 해당 필드에 인덱스가 없다면, 인덱스를 생성하여 공간 마이닝 서버쪽에서 유지, 관리한다. 이렇게 생성된 인덱스들은 다음에 들어올 질의에 대한 성능 향상을 위해서 사용된다.

공간 데이터 마이닝 질의 처리에 있어서 해당 모듈간 연관 관계와 그들이 사용하는 자료구조에 대한 내용이 [그림 2]에 나타나 있다.

2. 공간 마이닝 질의 최적화 전략

본 연구에서는 공간 마이닝 질의의 최적화 전략에 대해 2 가지 경우로 나누어 처리한다. 첫 번째는 공간 마이닝 질의에 자주 사용되었던 결과 테이블을 실체 뷰로 미리 만들어 AFT에 등록함으로써 빠른 성능을 유도하기 위한 방식이고, 두 번째는 예전

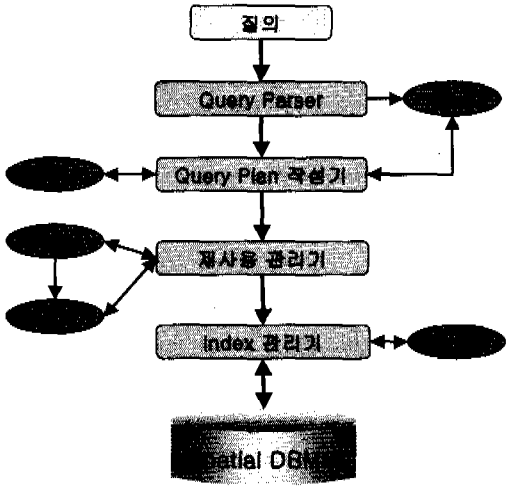

그림 2. 질의 처리 구조

에 생성된 질의 결과 테이블을 다시 사용할 수 있는 공간 마이닝 질의가 들어올 경우에는 그 결과 데이터를 다시 재 사용하기 위해 MQL(Mining Query list)을 사용하는 방식이다.

따라서 AFT와 MQL은 깊은 관련성을 가지고 있으며 [그림 3]은 그러한 관련성을 기반으로 최적화 전략을 추진하는 과정을 도식화하였다.

2.1 AFT 사용 전략

자주 사용되는 실체 뷰를 미리 생성하여 공간 마이닝 질의 처리를 향상시키고자 사용되는 방법으로 AFT에서 실체 뷰들을 관리한다.

Access Frequency Table list(AFT) 관리 프로세스

1) 모든 테이블에 실체 뷰 생성은 거의 불가능함으로 초기에는 타당하고 가능한 실체 뷰를 생성한다. 따라서, 처음에는 계층화(h0<-h1<-h2<-h3)되어 있는 테이블에 대해서 h1을 기준으로 생성하고 계층화가 없는 테이블에 대해서 h0을 기준으로 생성하되 생성할 것인지, 말 것인지를 결정해야 한다. 공간 데이터에 대해서는 공간 데이터를 설명해 주는 테이블을 가장 높은 level(h3)을 기준으로 grouping 한다.
2) MQL에서 빈도수가 높은 테이블에 대해서, 계층도를 참조하여 해당 테이블에 포함되거나 계층의 상위에 있는 테이블들은 MQL에서 삭제되고, 조건식이 다른 경우에는 비-공간 데이터에서는 해당 table을 UNION하고, 공간 데이터에 대해서는 관련 spatial pointer group에 대한 pre-merge를 수행한다.
3) 또한, MQL에서 빈도수가 높은 공간 데이터를 찾아서 spatial pointer group에 대한 각 공간 포인터별 항목에 대한 빈도수를 계산하여 자주 사용하고 있는 spatial pointer들을 찾아서 MV를 생성한다.

2.2 질의 결과 재사용 전략

위의 질의 처리 전략은 사용 데이터가 예전에 자주 사용된 적이 있어 그를 토대로 실체화 뷰를 만들어 질의에 대한 최적화를 하기 위한 전략이다. 하지만, 공간 마이닝의 특성상 공간 마이닝 질의는 실체 뷰가 만들어지기 이전의 유사질의가 다음에 사

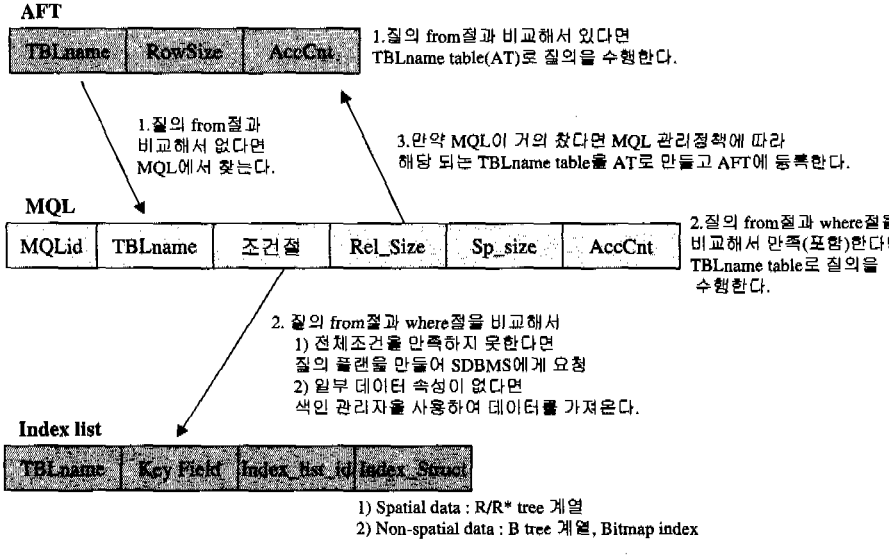

그림 3. AFT와 MQL의 상호 관련성

용될 경우가 많기 때문에 질의 결과 재사용 전략 또한 중요한 최적화 전략이라 할 수 있다.

Mining Query list(MQL) 관리 프로세스

이전에 수행되었던 질의에서 사용했던 데이터의 일부 또는 전부를 포함하는 질의에 대한 처리 속도 향상을 위해서 사용되며, MQL에서 자주 사용되는 테이블들은 실체화 뷰를 생성하여 AFT에 등록한다.

1) 질의에 사용된 테이블들을 MQL에서 찾아서 가장 확률이 높은 것 순으로 조건절을 검사하여 만족하는 테이블을 선택한다. 이것을 간단한 수식으로 표현하면 다음과 같다.

> 만족해야 하는 조건
> = {A(들어온 질의의 조건 필드수) >= Bi(MQL에서의 각 테이블의 필드수) and n(Bi-A) = NULL, 1 <= i <= MQL.Cnt(MQL에 들어온 질의수) }

2) 데이터의 일부가 포함되었을 경우,
- 다른 필드 추가 : 해당 필드만을 인덱스를 사용하여 가져온 후, 나중에 질의 결과와 merge 한다.
- 다른 필드 삭제 : 기존 데이터 필드의 검색 범위가 틀려지므로 새롭게 데이터를 가져와야 하는데, 이때에는 이전에 구성했던 인덱스를 이용하여 새롭게 데이터를 가져온다.

3) MQL 관리 정책

MQL의 버퍼 관리를 위해서 1, 2, 3 순으로 우선순위를 정하여 MQL을 관리하는데, 만약 MQL에 남아 있는 테이블(질의 결과 테이블)들이 모두 같은 순위라면 LRU 정책을 사용하여 유지한다.

MQL의 버퍼 사이즈가 거의 차게 되면, 내보내야 하는 table을 결정은 우선 순위에 따른다. 순위 적용은 아래 도표와 같은 순위로 적용하는데 순위가 낮은 것(1순위)부터 높은 순으로 삭제한다.

이때 버퍼량을 현저하게 줄여 주기 위해 3 순위는 MQL에서 내보내면서 반드시 MV table로 만들어서 AFT에 등록함으로써 동적으로 빠른 응답 시간을 줄 수 있는 실체 뷰를 자동적으로 생성, 유지한다. 2 순위는 일반적으로 MV table을 생성하지 않으나, 버퍼가 차고 3 순위 데이터가 없다면 MV table로 만들 것을 고려한다.

표 1. MQL 관리 정책

순위	AccCnt	Rel_Size	Sp_Size	AT여부
1 순위	↓	↓	↓	×
	↓	↑	↑	×
2 순위	↓	↓	↑	×/○
	↓	↑	↓	×/○
3 순위	↑	↓	↓	○
	↑	↓	↑	○
	↑	↑	↓	○
	↑	↑	↑	○

2.3 AFT와 MQL 활용 예제

아래의 초기 데이터 상에서 다음과 같은 질의(Q)가 들어왔을 때의 AFT와 MQL이 어떻게 반응하는지를 알아보기 위해 다음과 같은 간단한 예제를 준비하였다. 여기에서 계층화를 표현한 방식은 [h <- h1 <- h2 <- h3 ...]의 표현되며 숫자의 의미는 계층화의 높이를 나타내며 숫자가 높을수록 높은 계층임을 나타나며, 질의에 나타난 문자들은 질의가 들어왔을 때에 필요한 테이블의 한 필드라고 생각하자.

데이터 :

전체 데이터 필드 집합(S) = {A, B, C, D, E, F, G, Geo}

공간 데이터 집합(Geo) = {1, 2, 3, 4, 5, 6, 7, 8, 9}

AFT = {{AFT_1, A1B1C1Geo1, 0}, {AFT_2, Geo2Geo1, 0}}

Q1: ABDGeo

=> MQL(MQL_1, ABDGeo, 100, {1, 3}, 1}

=> AFT에서 만족한 데이터를 찾고 못했고 A, B, D가 인덱스가 없기 때문에 각각의 인덱스를 구성한다.

Q2: ABEFGeo

=> MQL(MQL_2, ABEFGeo, 1000, {2, 3, 5}, 1}

=> AFT에서 만족한 데이터를 찾지 못했고 E, F가 인덱스가 없기 때문에 각각의 인덱스를 구성한다.

Q3: ABDFGeo

=> MQL(MQL_3, ABDFGeo, 1000, {2, 3, 7, 9}, 1}

=> Q1에서 사용된 MQL_1을 재사용하고 F는 이전에 만든 인덱스를 사용하여 데이터를 가져온 후 병합한다. 그리고 Q1의 COUNT

는 2가 된다.

Q4: ABFGGeo
=> MQL(MQL_4, ABFGGeo, 150, {2, 3, 7, 8}, 1)
=> 데이터의 재사용이 안되므로 이전에 만든 인덱스를 사용하여 데이터를 가져온 후 병합한다.

AFT 생성을 수행한다고 했을 때, 다음과 같은 단계로 진행된다.

1) 가장 많이 사용된 MQL을 탐색한 후(MQL_1), 관련 질의(MQL_3)와의 병합 후, 제거함으로써 AT을 생성한다.
=> AFT = { {AFT_3, ABDGeo, 0} }

2) 가장 많이 사용된 공간 객체가 어느 것인지를 탐색 후, PRE_MERGE 한다.
=> AFT = { {AFT_4, {2, 3}, 0} }

V. 예 제

5장에서는 2개의 공간 데이터 마이닝 질의에 대한 질의 처리 및 최적화 과정을 예제를 통해서 구체적으로 보이고자 한다.

질의 설명	충청도에 인접하는 지역들을 인구 10000명당 범죄율로 해당 지역의 classification rule을 얻는 질의
사용 질의 (Q1)	Mine rule classification as "region" Select crimes10000, region_name from region r1, region r2 where r1.name = "충청도" and adjacency(r2, r1) extracting rules with threshold: 0.3

[Q1] 만약 AFT가 생성되어 있지 않은 상태에서 처음 들어오는 질의라면

① 질의 파싱
② region table이 예전에 사용되지 않았다면(AFT/MQL 조건 검사) query plan 작성
([5] 논문의 plan 3을 수행한다고 가정)
③ 공간 DBMS에서 질의를 수행하여 결과 데이터(MQL_1)를 마이닝 서버로 보낸 후, 만약 각각 필드에 index가 없다면 생성(local 화일)
Index_list[region, name, index_list_1, btree]
④ local의 결과(MQL_1) 테이블을 이용하여 마이닝 알고리즘을 수행한다.
⑤ rule 검증 후 사용자에게 보여줌
MQL(mql_1, region, 'r1.name = "충청도" ^ r2 adjacency r1', 100, 40, 1)

[Q2] 만약 마이닝 방법이 clustering으로, 조건문에 crimes10000 > 0.5가 추가되어 마이닝 질의가 다시 들어온다면

① 질의 파싱
② region table이 이전에 사용되었고(MQL에서 table 검사), 기존 조건에 변화가 있으므로(MQL의 조건절 검사)
③ 변화된 조건절의 필드에 대해서만 query plan을 작성하여 수행 후, 해당 인덱스를 가져와서 기존 결과 테이블(MQL_1)과 merge해서 재구성한다.
Index_list[region, crimes10000, index_list_2, btree]
④ local의 결과(MQL_2) 테이블을 이용하여 마이닝 알고리즘을 수행한다.
⑤ rule 검증 후 사용자에게 보여줌
MQL(mql_1, region, 'r1.name = "충청도" ^ adjacency(r1,r2)', *100, 40, 2*)
MQL(mql_2, region, 'r1.name = "충청도" ^ adjacency(r1,r2) ^ crimes10000 > 0.5', 80, 35, 1)

*** 나중에 MQL buffer가 차면 AccCnt와 Row_Size을 고려해서 실체 뷰로 만들 테이블을 선정한다.(Sp_Size을 고려하지 않음)
AccCnt : 높음, Rel_Size : 높음 => 3
AccCnt : 높음, Rel_Size : 낮음 => 3
AccCnt : 낮음, Rel_Size : 높음 => 고려(2순위 : 삭제 또는 생성)
AccCnt : 낮음, Rel_Size : 낮음 => 삭제(1순위)

⑥ 사용자에게 보여줌.
위의 두 번째 질의(Q2) 과정의 이해를 돕기 위해 그림으로 도식화하면 [그림 4]와 같다.

VI. 결 론

데이터 마이닝은 반복적이고 지속적으로 시행착오를 겪어가며 어떤 적합한 룰을 찾아내는 과정이므로 조급하게 어떤 목적도 없이 무작정 마이닝을 수행하는 것은 무의미하다. 데이터 마이닝을 하기 위해서는 우선 무엇을 할 것인지을 명확하게 정의한 다음, 타당한 결과를 찾아낼 룰을 여러 마이닝 방법 중에서 찾아야 한다.

본 논문에서는 공간 데이터 마이닝을 위해, 사용자가 쉽게 접근할 수 있는 공간 데이터 마이닝 질의 언어를 설계하였고, 그 언어에 기반한 공간 데이터 마이닝 질의를 효율적으로 처리할 수 있도록 도와주는 공간 마이닝 질의 최적화 방안을 공간 실체화 뷰의 사용성과 이전에 사용된 유사 질의의 재사

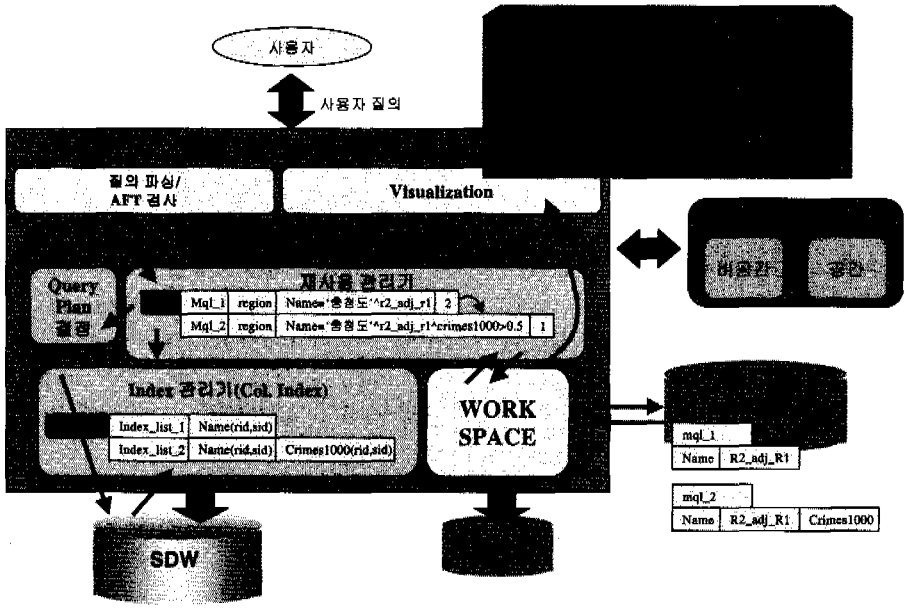

그림 4. 질의 2의 처리 과정

용성과 관련된 인덱스 생성을 통해 제시하였다. 본 논문에서 제시한 질의 최적화 전략을 사용하면 여러 사용자가 사용하는 클라이언트-서버 공간 마이닝 시스템 환경에서 네트워의 부하를 크게 줄일 수 있고, 공간 데이터베이스 서버가 해야 할 일들을 줄여 줌으로써 성능이 향상될 수 있다.

본 논문의 기여는 다음과 같다. 첫 번째는 사용자가 쉽게 공간 데이터 마이닝 질의를 표현할 수 있도록 SDMQL을 정의, 설계하였고 두 번째는 설계된 SDMQL에 기반한 공간 데이터 마이닝 질의의 효율적인 처리를 위한 최적화 방법론을 제시하였다.

향후 연구 과제로 공간 데이터에 대한 변경이 발생할 경우, 어떻게 효율적으로 공간 실체화 뷰에 반영할 것인가에 대한 연구가 필요하다. 또한 공간 데이터 마이닝 질의 최적화를 위한 모듈들을 포함한 프로토타입 시스템을 구현하는 것에 대한 연구가 필요하다.

참 고 문 헌

[1] J. Han, K. Koperski, and N. Stefanovic, "GeoMiner: A System Prototype for Spatial Data Mining", Proc. 1997 ACM-SIGMOD Int'l Conf. on Management of Data (SIGMOD'97), Tucson, Arizona, May, 1997(System prototype demonstration).

[2] Ooi, B. c., "Efficient Query Processing in Geographic Information Systems", LNCS 471, Springer-Verlag, 1990.

[3] J. Han, Y. Fu, K. Koperski, W. Wang, and O. Zaiane, "DMQL: A Data Mining Query Language for Relational Databases", 1996 SIGMOD'96 Workshop. on Research Issues on Data Mining and Knowledge Discovery (DMKD'96), Montreal, Canada, June 1996.

[4] M. Egenhofer "Spatial SQL: A Query and Presentation Language" IEEE [Transactions on Knowledge and Data Engineering 6 (1): 86-95, 1994.

[5] Walid. G. Aref, Hanan Samet,"Optimization Strategies for Spatial Query Processing", VLDB 1991

[6] Frank Olken, "Random sampling from databases", Doctor of Philosophy in CS in the Univ. of California at Berkeley, 1993

[7] R. Ng and J. Han. "Efficient and effective clustering method for Spatial data mining", In proc. int. Conf. VLDB, 1994

[8] K. Koperski and J. Han, "Discovery of spatial association rules in geographic information databases", In proc. 4th Int'l Symp. on Large Spatial Databases, Portland, Maine, Aug. 1995.

[9] M.S. Chen, J. Han, and P.S.Yu. "Data mining: An overview from a database perspective", IEEE Transactions on Knowledge and Data Engineering, 8:886-993, 1996.

[10] Ashish Gupta, Inderpal Singh Mumick, and V. S. Subrahmanian, "Maintaining Views Incrementally", Proc. of ACM-SIGMOD, May, 1993

[11] Yihong Zhao, Prasad M. Deshpande, Jeffrey F. Naughton, Amit Shukla, "Simultaneous Optimization and Evaluation of Multiple Dimensional Queries", Technical Report, Univ of Wisconsin-Madison, CS Department, 1997

[12] Laura M. Haas, Donald Kossmann, Ioana Ursu, "Loading a Cache with Query Result", IBM Technical Report America36, 1999.

[13] N. Beckmann, H. P. Kriegel, R. Schneider, and B.Seeger, "The R* tree: An efficient and robust access method for points and rectangles", In proc. SIGMOD Intl. Conf. on Management of data, 1990

[14] A. Guttman, "R-trees: A dynamic index structure for spatial searching", In proc. ACM-SIGMOD Conf. on Management of Data, 1985

[15] Arthur M. Keller, Julie Basu, "A predicate-based caching schema for client-server database architectures.", In Proc. of the Intl. IEEE Conf. on Parallel and Distributed Information Systems", Austin, TX, USA, September 1994.

[16] 김동현, 문상호, 홍봉희, "실체화된 공간 뷰을 위한 점진적 변경 방법", '96 가을 한국 정보과학회 학술 발표 논문집, Vol 23, 1996.

[17] 오병우, "해시-기반 공간 클러스터링을 사용한 공간 데이터 마이닝", 건국대학교 전자계산학과 박사 학위 논문, 1999.

김 충 석(Choung-Seok Kim) 정회원

현재 : 신라대학교(구.부산여자대학교) 컴퓨터정보공학부 부교수

한국통신학회지 논문지 제 25권 8A호 참조

이 현 창(Hyun-Chang Lee) 준회원

2001년 3월~현재 : 경인여자대학 멀티미디어정보전산학부 교수

2001년 2월 : 홍익대학교 전자계산학과 박사

1996년 2월 : 홍익대학교 전자계산학과 석사

1993년 2월 : 원광대학교 컴퓨터공학과 학사

<주관심 분야> 웹 웨어 하우징, 데이터 웨어 하우스, 데이터 마이닝, 전자 상거래, 멀티미디어

김 경 창(Kyung-Chang Kim) 정회원

현재 : 홍익대학교 컴퓨터공학과 부교수

한국통신학회지 논문지 제 21권 7호 참조