위치 기반 서비스 지원을 위한 연관 클래스 집합 생성 기법

(An Associative Class Set Generation Method for supporting Location-based Services)

김 호 숙 † 용 환 승 ††
(Ho-Sook Kim) (Hwan-Seung Yong)

요 약 최근 이동 컴퓨팅 환경 하에서 위치를 기반으로 하는 다양한 서비스가 점차 증가하고 있다. 본 논문은 이동 컴퓨팅 환경에서 대량의 공간 데이타베이스를 기반으로 하는 위치 기반 서비스를 지원하기 위하여, 요청되는 질의들 사이에 존재하는 의미적으로 연관성이 있는 빈발 항목인 연관 클래스 집합을 제안하고, 이를 효과적으로 찾는 방법에 대해 소개한다. 이때 요청되는 질의들의 시간적 연관 관계, 그리고 이러한 서비스를 제공해 주는 공간 객체들 사이의 거리와 사용자의 접근 특성이 함께 고려된다. 이러한 연구 결과는 이동 환경이 갖는 제약점을 극복하면서 효과적으로 위치 기반 서비스를 지원하는 바탕이 된다. 즉 생성된 연관 클래스 집합은 이동 컴퓨팅 환경에서 지리 정보를 서비스 할 때 관련 자료를 추천하는 시스템에 활용할 수 있고, 지리 정보를 고려한 광고 방송이나 도시 개발 계획 등에 이용할 수 있으며, 이동 사용자를 위한 클라이언트의 캐쉬 정책에 응용될 수 있다.

키워드 : 이동 컴퓨팅, 위치 기반 서비스, 빈발 항목, 연관 규칙, 시-공간 클러스터링

Abstract Recently, various location-based services are becoming very popular in mobile environments. In this paper, we propose a new concept of a frequent item set, called "associative class set", for supporting the location-based service which uses a large quantity of a spatial database in mobile computing environments, and then present a new method for efficiently generating the associative class set. The associative class set is generated with considering the temporal relation of queries, the spatial distance of required objects, and access patterns of users. The result of our research can play a fundamental role in efficiently supporting location-based services and in overcoming the limitation of mobile environments. The associative class set can be applied by a recommendation system of a geographic information system in mobile computing environments, mobile advertisement, city development planning, and client cache police of mobile users.

Key words : mobile computing, location-based services, frequent item set, association rule, spatio-temporal clustering

1. 서 론

무선 네트워크 기술과 이동 정보 기기의 발전은 컴퓨팅 환경에 있어서 이동 컴퓨팅이라는 새로운 패러다임을 낳았다[1]. 특히 PDA나 휴대전화 등의 소형 이동 장비를 이용하여 해당 지역의 지도 정보를 얻거나 웹 사이트를 접속하는 등의 위치 기반 서비스(LBS: Location-based Services)가 빠르게 확대되고 있다[2].

† 비 회 원 : 동의공업대학 컴퓨터정보계열 교수
khosook@ewha.ac.kr
†† 종신회원 : 이화여자대학교 컴퓨터학과 교수
hsyong@ewha.ac.kr
논문접수 : 2003년 9월 18일
심사완료 : 2004년 2월 4일

본 논문은 이동 컴퓨팅 환경에서 대량의 공간 데이타베이스를 이용하는 위치 기반 서비스를 효율적으로 지원하기 위하여 새로운 개념의 빈발 항목인 연관 클래스 집합(associative class set)을 제안하고, 이를 효과적으로 찾는 방법에 대해 소개한다. 빈발 항목(frequent item set)이란 자주 함께 요청되는 항목들의 집합으로 다양한 응용 시스템에서 사용자의 행위 분석을 위한 중요한 기준이 되고 있다. 예를 들면 장바구니 분석(market basket analysis)에서는 효과적인 매장의 전시와 대상 마케팅(target marketing)을 위하여 고객의 구매 성향 분석이 필요하고 이를 위해 구매 트랜잭션의 빈발 항목을 생성한다. 또한 효과적인 웹 서버 관리와 추천 집합

생성을 위해서 웹 접근 로그를 이용하여 웹 페이지의 빈발 항목을 구하기도 한다. 지금까지 이러한 빈발 항목을 찾기 위해 많은 연구[3-6]가 수행되었으나 위치 기반 서비스를 지원하기 위하여 요청 질의의 시간적 특징, 제공되는 객체의 공간상의 특징 및 사용자의 접근 패턴을 동시에 고려한 연구는 수행되지 못하였다.

표 1은 위치 기반 서비스를 지원하는 서버 로그 데이타의 일반적인 형태로 사용자 ID, 사용자의 위치 좌표, 요청되는 서비스를 나타내는 클래스, 결과 객체의 ID, 요청되는 결과 객체의 위치 좌표와 요청 시간 등으로 구성된다. 이를 이용하여 시간-공간적으로 연관성이 있는 빈발 항목인 연관 클래스 집합을 생성하는 단계를 설명해 보자.

표 1 위치 기반 서비스의 접근 로그

Customer ID	User Position	Class	Object ID	Object Position	Access time
A	(323, 334)	Movie center	98	(432,400)	17:03:02
B	(211, 566)	Restaurant	23	(200, 480)	17:04:30
A	(340, 337)	Flower shop	54	(200,189)	17:04:45
A	(355, 339)	Snack bar	74	(450,366)	17:04:50
D	(399, 273)	Hotel	77	(400,250)	20:04:23
D	(402, 250)	Restaurant	63	(410,285)	20:07:40

표 1에는 세 명의 사용자의 접근 기록이 시간 순서대로 적재되어 있다. 사용자 A는 현재 자기 위치와 멀리 떨어져 있는 위치의 특정 영화관을 찾았고 그 영화관과 근접한 스낵 바를 곧 이어 조회했다. 사용자 D는 자기의 현재 위치에서 가장 가까운 호텔을 조회했고 곧 이어 근처의 레스토랑을 조회하였다. 일반적으로 동일한 사용자의 시간적으로 근접한 질의 사이에는 연관성이 존재한다고 가정할 수 있다. 로그 데이타 분석 시 일정 시간을 기준으로 사용자의 의미 있는 접근의 범위를 제한하는 방법은 웹 사용 마이닝(Web usage mining) 등에서 많이 사용된다. [7]에서는 사용자의 세션(session) 구분 시에 30분 간격 내에 발생한 사용자의 클릭 스트림을 한 세션으로 분리하였다. 사용자 별로 30분의 시간 내에 조회된 객체들을 연관성 있다고 가정할 때 표 1의 로그는 {98,54,74}, {23}, {77,63} 세 개의 객체 집합들로 분리될 수 있다. 또한 공간 데이타를 서비스하는 위치 기반 질의에서는 연관성을 갖는 질의의 대상 객체들은 실제 공간상에서도 거리가 가까운 특징을 갖는다고 가정할 수 있다. 예를 들어 한 사용자가 특정 극장을 찾은 후 곧 이어 스낵 바를 찾는다면 이때 사용자가 원하는 스낵 바의 위치는 논리적으로 해당 극장과 가까운 곳으로 제한된다고 볼 수 있다. 즉 이때의 질의는 "X라는 극장과 50미터 이내에 있는 스낵 바의 종류는 무엇인가?"와 같은 형태를 띨 것이다. 표 1의 예에서 사용자 A는 극장을 찾은 후 근접한 시간 내에 꽃집과 스낵 바를 조회하였지만 꽃집은 극장과 스낵바로부터 거리가 매우 멀기 때문에 극장과 스낵 바만이 연관성이 있다고 볼 수 있다. 그 결과 객체 집합 {98, 74}와 {77, 63}은 시간과 공간적 측면에서 동시에 인접한 객체들의 집합으로 볼 수 있다. 우리는 이동 컴퓨팅 환경에서 자주, 함께 조회되는 {극장, 스낵바}, {호텔, 레스토랑}과 같은 시-공간적으로 연관성 있는 클래스 집합을 찾는다.

우리가 아는 한계에서는 이동 환경 하에서 위치 기반 서비스를 지원하기 위하여 사용자의 접근 특성과 요청되는 질의의 시간적 연관성, 그리고 이러한 질의가 제공하는 객체의 공간적 연관성을 함께 고려한 클래스의 빈발 항목을 생성하는 연구는 지금까지 수행되지 못하였다. 우리의 연구의 주된 기여는 다음과 같이 요약될 수 있다.

- 이동 컴퓨팅 환경 하에서 시간과 공간적 제약을 함께 고려하여 위치 기반 서비스를 위한 빈발 항목을 찾는 최초의 시도이며,
- 시간적, 공간적 연관성을 갖는 객체 집합을 찾기 위한 효과적인 세 가지 알고리즘을 제안하였고,
- 논문에서 제안된 연관 클래스 집합은 향후 이동 사용자를 위한 캐쉬 정책이나 이동 광고, 개발 계획 등의 응용에 폭 넓게 적용될 수 있다.

2. 관련연구

트랜잭션 사이에 존재하는 빈발 항목에 대한 연구는 Agrawal 등에 의해 처음 수행 되었다. 대표적인 연관 규칙 알고리즘인 Apriori[3]과 이를 기반으로 한 AprioriAll, AprioriSome[4] 등이 연구되었고, 기존의 알고리즘에 시간 제약조건, 슬라이딩 시간 윈도우, 분류 등을 이용하여 일반화시킨 방법으로 GSP[5]가 제안되었다. 또한 공간 데이타베이스를 이용한 패턴 발견에 대한 많은 연구가 수행되었다. [8]에서는 공간 연관 관계에 대한 개념이 제안되었고, Morimoto는 공간 데이타베이스에 존재하는 객체들 가운데 자주 이웃하는 클래스 집합을 (frequent neighboring class set) 효과적으로 찾는 연구를 수행하였다[6]. 즉 특정 서버 내에 존재하는 객체들을 대상으로 객체가 제공하는 서비스 클래스들이 거리 상으로 가까운 위치에 존재하는 빈도가 일정 수준 이상인 클래스 집합을 찾는 것으로, 공간적으로 이웃하는 빈도가 많은 서비스들 사이에는 높은 연관 관계가 있다는 가정에 바탕을 둔다[6]. 그러나 지리 공간상에 가까이 위치하는 빈도가 높다는 특징만으로 생성되는 빈발 항목은 너무나 일반적인 결과를 나타내므로 실제로 위치 기반 서비스를 효과적으로 지원하기 위한 빈발 항목 생성에는 부족함이 있다. Peng등은 이동 컴퓨팅 환경의 특징인 사용자의 이동성을 고려한 연구를 수행

하였다[9]. 즉 분산 환경에서 사용자의 이동에 대한 로그 파일을 이용하여 사용자의 움직임과 이때 접근되는 객체들의 패턴을 마이닝하고, 그 결과를 각 서버에 대한 데이타 할당에 이용하여 성능을 높이는 결과를 보였다. 그러나 지금까지의 연구들은 이동 환경 하에서 사용자에게 지리 정보를 서비스하는 응용이 갖는 특징인 서비스의 시간적 연관 관계와 서비스되는 객체들의 공간적 특징 동시에 고려하지는 못하였다. 본 논문에서는 효과적으로 위치 기반 서비스를 지원하기 위하여 위의 특징들을 동시에 고려한 새로운 개념의 빈발 항목인 연관 클래스 집합을 제안하고 이를 찾는 알고리즘들을 제안한다.

3. 연관 클래스 집합 생성

본 절에서는 논문에서 사용되는 기본 용어를 정의하고, 연관 클래스 집합을 찾기 위한 알고리즘들을 제안한다. 연관 클래스 집합 생성 단계는 다음과 같다. 먼저 데이타 선정(cleaning)을 통해 질의 로그를 사용자별로 분리하고 분석에 필요한 속성값을 산출한다. 둘째, 시간과 공간적 제약 및 공간 정보를 이용하는 사용자의 접근 패턴의 특징을 고려하여 시-공간 인접 객체(TSNO: Temporal-Spatial Neighbor Object)를 생성한다. 마지막으로 시-공간 인접 객체 집합을 대상으로 연관 규칙을 수행하여 연관 클래스 집합을 찾는다. 이를 그림으로 표현한 것은 그림 1과 같다.

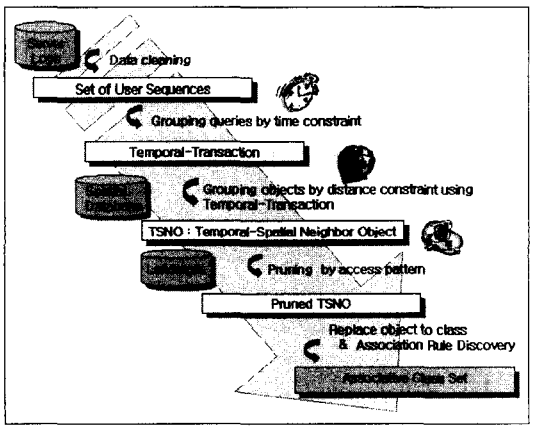

그림 1 연관 클래스 집합 생성 단계도

설명을 위하여 다음의 몇 가지 용어를 정의한다.

- 객체(Object) : 공간 데이타베이스에 존재하는 각각의 데이타로 공간상의 위치를 표현하는 좌표 속성과 객체가 제공하는 서비스의 종류를 나타내는 클래스 속성을 갖는다.
- 클래스(Class) : 객체가 제공하는 서비스 종류를 그룹별로 분류한 것.
- 사용자 시퀀스(User Sequence) : 시간 순서로 정렬된 한 사용자의 질의 리스트.
- 시간 윈도우(Time Window) : 분석가에 의해 연관성을 갖는 시간의 한계로 결정된 시간 간격
- 시간-트랜잭션(Temporal-Transaction) : 사용자 시퀀스의 부분 집합으로 시간 윈도우 내에 요청된 질의 리스트.
- 시간 인접 객체(Temporal Neighbor Objects) : 하나의 시간-트랜잭션 내에 포함된 질의들에 대한 결과 객체 집합.
- 최대 거리(Maximum Distance) : 분석가에 의해 연관성을 갖는 의미 있는 공간의 한계로 결정된 거리
- 시-공간 인접 객체(TSNO : Temporal and Spatial Neighbor Objects) : 시간 인접 객체들 중에서 최대 거리를 기준으로 공간적으로 근접한 하나 이상의 객체들의 부분 집합
- 단일 시-공간 인접 객체(Single TSNO) : 하나의 원소로 이루어진 시-공간 인접 객체
- 연관 클래스 집합 (Associative Class Set) : 시-공간 인접 객체에 속한 오브젝트들의 클래스 속성을 기준으로 연관 규칙을 적용하여 생성된 빈발 항목

3.1 사용자별 시퀀스 생성

연관 클래스 집합을 생성하는 첫 단계는 해당 로그를 사용자 별로 구분하고 시간 순서로 정렬하는 것이다. 분석의 대상이 되는 로그 데이타는 표 1과 같이 여러 사용자로부터 발생되었고 시간 순서로 적재되어 있다. 본 논문에서는 사용자별로 생성된 시-공간 인접 객체의 모든 합집합을 대상으로 연관 클래스 집합을 생성하지만, 동일한 방법으로 개인별로 연관 클래스 집합을 생성할 수 있고, 이렇게 생성된 결과는 개인화 등에 사용될 수 있다.

3.2 시간적 제약 사항을 적용한 트랜잭션 분리

둘째 단계는 사용자별 시퀀스를 이용하여 각각의 질의 발생 시간을 기준으로 시간-트랜잭션 집합을 생성한다. 즉 연관성이 높은 질의는 가까운 시간 간격 내에 질의될 것이라는 가정을 기반으로, 시간 윈도우 내에 발생한 질의들을 묶어서 하나의 시간-트랜잭션으로 구분한다. 본 논문에서는 시간 축을 기준으로 질의들의 시간 간격이 시간 윈도우 값인 δ를 넘지 않는 한계를 유지하면서 한 트랜잭션 내에 속한 질의들의 연관성을 최대로 만들기 위해서 다음과 같은 근사(approximate) 알고리즘을 제안한다. 먼저 한 사용자의 모든 질의들을 이전 질의와의 시간 간격을 기준으로 정렬한다. 전체 질의들의 시작 시간과 마지막 시간 사이의 간격이 δ 이내인 경우는 모

든 질의들을 하나의 시간-트랜잭션으로 볼 수 있지만 시간 간격이 δ 이상인 경우에는 이전 질의와의 시간 간격이 큰 것부터 순서대로 트랜잭션들을 분리하면서 모든 시간-트랜잭션의 시간 간격이 δ 이내에 들어 갈 때까지 반복한다. 이를 정리한 것은 알고리즘 1과 같다.

```
Input : data sequence set of all users (S), time window (TW)
        Si = {t1, t  ··· tn} , Si is one user' s data sequence,  ti is a query in Si
Output : set of temporal-transactions (G)

Generate_Temporal_Transaction ( S, TW )
{
G = { } ; //empty
for i =1 to last user
  {   startTime = timestamp of t1   // t1 is the first query in Si ;
      endTime  = timestamp of tn  // tn is the last query in Si ;
      if (endTime - startTime < TW)
          Gi = Si ; // all queries in Si are a same temporal-transaction
      else
      { Ti = Sorted queries in Si based access time interval(descending order);ⓐ
          // Ti is the list of query ID
        Gi = Partition_transaction (Si, Ti, TW);ⓑ
      } // end else
    G = G + Gi;
  } //end for
} // end Generate_Transaction


Input : subset of a sequence (S), sorted query ID list (T) , Time Window(TW)
Output : set of temporal-transaction (G)

Partition_transaction (S, T, TW);
{
 G = { } ; //empty
 ti = first query ID in T while ti ∈ S ;
    // time interval (ti-1, ti) is the maximum time interval in S
 T = T - ti  ;
 startTime = timestamp of the first query in S ;
 endTime = timestamp of the last query in S ;
 midTime1 = timestamp of ti-1 ;
 midTime2 = timestamp of ti ;

if (midTime1 - startTime < TW)
    { Gi = set temporal-transaction from the first query in S to ti-1;
      G = G + Gi ;
    } // end if
else
    { Si = all queries from first query of S to ti-1;
      G = G + Partition_transaction (Si, T, TW);
    } // end if

if (endTime - midTime2 < TW)
  {  Gi = set temporal transaction from ti to the last query in S;
     G = G + Gi ;
  } // end if
else { S2 = all queries from ti to the last query of S ;
       G = G + Partition_transaction (S2, T, TW);
     } // end else
return G ;
 \// end Partition_transaction
```

<알고리즘 1> 시간 윈도우를 기준으로 시간-트랜잭션을 생성하는 알고리즘

한 사용자 시퀀스에 속한 질의의 전체 수가 n일 때 알고리즘 1의 시간 복잡도는 질의 리스트를 정렬하는 ⓐ 단계에 O(*nlogn*), 트리를 생성하는 ⓑ 단계에서 O(*nlogn*) 만큼의 시간이 소요된다.

3.3 공간적 제약 사항을 적용한 TSNO 식별

실생활에서 이동 사용자는 일정 시간 간격 이내에 서로 다른 목적을 갖고 여러 가지 위치 기반 질의를 할 수 있다. 예를 들면 오전에 병원으로 향하면서 병원과 가까운 약국을 조회한 후에 곧이어 저녁에 있을 약속을 위해 병원과 먼 위치의 레스토랑을 검색하기도 한다. 이러한 경우 병원과 약국은 연관성이 있지만 병원과 레스토랑은 서로 연관성이 없도록 구분되어야 한다.

본 연구에서는 분석가가 정한 연관성을 갖는 공간상의 최대 거리를 기준으로 가까운 객체들끼리 모아서 의미 있는 시-공간 인접 객체들을 생성하는 알고리즘 TIMD와 TICO를 제안한다. 이때 최대 거리의 크기는 응용 시스템의 종류에 따라 결정되어야 한다. 예를 들어 [6]에서는 50m의 거리 내에 존재하는 객체들을 연관성 있는 이웃으로 보았고, 인터넷에서 지리 정보를 검색하는 지도 서비스의 경우에는 관련성 있는 공간 질의 대상 영역이 평균 1.4km * 1.4km의 크기를 갖는 것으로 분석되었다[10].

3.3.1 TIMD(Transaction dIvide algorithm by Maximum Distance)

TIMD는 연관성 있는 모든 객체들이 일정 거리 γ_1내에 존재하도록 연관성 있는 공간의 범위를 제한하는 방법이다. 예를 들어 새로운 꽃집을 개업하기 위해 적당한 위치를 찾는 사람에게는 일정 거리 반경 내에 꽃집과 연관성이 높은 학교나, 아파트 단지, 병원 등이 존재하면서 동시에 꽃집이 없거나 수가 작은 지역이 좋은 후보지가 될 것이다. 이러한 예에서는 공간상의 연관성이 있는 지역이란 모든 객체들이 일정한 거리 이내에 존재하는 것으로 제한될 수 있고, 볼록 다각형의 형태를 (convex polygonal shape) 띨 것이다.

TIMD의 수행 단계는 다음과 같다. 하나의 시간-트랜잭션에 속해있는 객체들에 대하여 먼저 한 객체 a를 선정하고, a와 가장 가까운 객체 b를 찾아서 두 객체 사이의 거리를 측정한다. 만약 a, b 사이의 거리인 dist(a,b)가 γ_1보다 크면 a는 연관성 있는 시-공간 인접 객체를 생성하는 대상에서 제외한다. 만약 dist(a,b)가 γ_1보다 작다면 두 객체 a, b의 중심점 c를 찾고 c에서 가장 가까운 객체 d를 (a, b를 제외한 나머지 중에서) 찾아서 a, b 각각과의 거리가 γ_1보다 작은가를 검사한다. 만약 dist(a,d), dist(b,d)의 크기가 모두 γ_1 보다 작다면 객체 a, b, d는 모두 거리 γ_1의 범위 내에 존재하는 객체들이 된다. 동일한 방법으로 객체들의 중심점과 가장 가까운 객체와 기존의 객체들과의 거리가 γ_1보다 커지게 될 때, 이전 객체들을 하나의 연관 관계가 있는 시-공간 인접 객체로 구분하고, 남아 있는 객체 집합을 대상으로 위의 방법을 계속 수행한다. 이를 정리하면 알고리즘 2와 같다.

TIMD의 시간 복잡도는 선택된 객체들의 중심점에서 가장 가까운 객체를 찾는 ⓐ 단계에 의해 결정된다. 즉 [6]에서와 같이 보로노이 다이아그램을 이용하거나 쿼드트리 등의 공간 인덱스를 사용하는 경우에는 O(*nlogn*)의 시간이 소요되지만 그렇지 않는 경우에는 최대 $O(n^2)$의 시간 복잡도를 갖게 된다. 그러나 본 연구에서는 먼저 각 사용자 별로 시간 윈도우이내에 속하는 질

의의 결과 객체를 선택한 후에 해당하는 시간 인접 객체만을 대상으로 TIMD 알고리즘을 수행한다. 즉 대상이 되는 객체의 수(n)가 작으므로 시간 복잡도가 성능에 큰 영향을 주지 않는다. [10]에 의하면 동일 세션에서 한 사용자가 연속적으로 요청하는 공간 질의 횟수는 평균 17.9 회이며 전체 사용자의 80% 이상이 최대 56 회를 넘지 아니하였다.

```
Input : temporal-transaction (T), maximum distance (MD)
        T = {o1,o2,…on}, oi is an object which is included in a temporal-
transaction Output : set of Temporal and Spatial Neighbor Objects (TSNO)

    Transaction_divide_algorithm_by_maximum_Distance (T, MD )
    {
     TSNO = { } ; // initialize
     while ( T is not empty )
      {  Temp  = { } ; state = true ;
         obj1 = select one object in T ;
         T = T - obj1 ;
         Temp = Temp + obj1;
         while ( state )
             { center = calculate a center position of all objects in Temp;
               obj2 = select the nearest neighbor of the center ; ②
             if  ( all distant from obj2 to each element of Temp < MD ) then
                    {   Temp = Temp + obj2 ;
                        T = T - obj2 ;  } // end if
             else  { TSNO = TSNO + Temp;
                     state = false;  } // end else
         } // end while state
      } // end while T is not empty
    } // end of Transaction_divide_algorithm_by_maximum_Distance ( )
```

<알고리즘 2> TIMD

3.3.2 TICO(Transaction dIvide algorithm by Connected Objects)

TICO는 가까운 거리 내에 위치하는 이웃 객체들의 연결된 집합으로 연관성 있는 공간의 범위를 제한하는 방법이다. 즉 한 객체 O_1을 중심으로 반경 γ_2 범위 이내에 존재하는 모든 객체들을 O_1의 이웃으로 정하고, 다시 각각의 이웃을 중심으로 반지름 γ_2이내에 속하는 모든 객체들을 O_1의 이웃으로 결정하는 방법을 통해 이웃 객체들을 확장하고 이렇게 생성된 모든 연결된 이웃들의 집합을 하나의 시-공간 인접 객체로 결정하는 방법이다. 이 경우 하나의 시-공간 인접 객체에 속한 최초의 객체와 마지막 객체는 매우 먼 거리에 위치할 수 있지만 그 사이를 연결하는 일정 간격 이내의 객체들이 지속적으로 존재하게 된다. 즉 큰 도로를 따라 형성된 상점들의 분포나 호수를 둘러싸고 형성된 레스토랑들의 분포와 같은 예에서는 공간상의 연관성이 있는 지역은 뱀처럼 길게 연결된 모양(snake shape)을 띨 수 있다. 이러한 지역을 서비스하는 위치 기반 서비스에서는 사용자의 이동 경로 상에 있는 관련 객체들이 모두 연관성 있다고 간주되어야 한다. TICO의 생성 방법은 알고리즘 3과 같다.

TICO의 시간 복잡도 역시 공간 인덱스를 사용하는 경우에는 $O(nlogn)$의 시간이 소요되지만 그렇지 않는 경우에는 최대 $O(n^2)$의 시간 복잡도를 갖게 된다. TIMD와 TICO는 수행 시 객체의 선택 순서에 따라 생성되는 시-공간 이웃의 결과가 달라질 수 있는 근사 알고리즘이다. 그러나 일반적인 마이닝 응용에서는 각 이웃 집합의 정확한 지지도보다는 어떤 패턴이 의미 있는 이웃 집합인지를 찾는 것이 더 중요하게 고려되므로[6] 본 연구에서는 근사 알고리즘을 이용하였다.

```
Input : a temporal-transaction (T) , maximum distance (MD))
        T ={o1,o2,…on}, oi is an object which is included in a
temporal-transaction Output : set of Temporal and Spatial Neighbor Objects
(TSNO)

Transaction_divide_algorithm_by_Connected_Objects (T, MD )
{
 TSNO = { } ; // initialize
 while ( T is not empty )
    {  Temp  = { } ; Queue = { } ;
       obj1 = select one object in T ;
       insert obj1 to Queue ;
       T = T - obj1 ;
       Temp = Temp + obj1;
       while (Queue is not empty)
        { obj1 = delete the first object from Queue ;
          S = find all neighbors of obj1 ;
              // obj2 is a neighbor of obj1 when dist(obj1, obj2) < MD and
obj2 ∈ T                Insert S to Queue ;
          T = T - S ;
          Temp = Temp + S;
        }
       TSNO = TSNO + Temp;
    } // end while T is not empty
} // end of Transaction_divide_algorithm_by_Connected_Objects ( )
```

<알고리즘 3> TICO

그림 2는 위의 두 가지 알고리즘을 적용한 예를 보여준다. 그림에서 원 숫자는 시간 -트랜잭션에 의해 조회된 객체들을 접근 시간 순서에 따라 공간상에 표현한 것이다. 이 경우 사용자는 start 위치에서 시작하여 end 위치로 이동하면서 두 지역의 있는 객체의 집합에 대한 질의들을 수행하였다. 이때 일정 시간의 범위 내에 요청된 질의들 중에서 공간적 멀리 떨어져 있는 객체들을 서로 다른 시-공간 인접 객체로 분리되어야 한다. 그림 2의 (c)와 (e)는 두 가지 알고리즘의 적용 결과 객체 5번에 대한 서로 다른 분리 결과를 보여준다.

3.4 표지물 객체의 제거

사용자가 위치 정보를 기반으로 하는 공간 질의를 수행하는 경우에 원하는 목표 객체와 더불어 해당 지역을 대표하는 표지물(landmark)의 기능을 하는 교육 기관 또는 행정 기관에 대한 요청을 함께 수행하는 패턴을 갖는다[10]. 이러한 접근 특징은 지도와 같은 형태로 지역 정보를 표현하는 많은 위치 기반 서비스의 응용에서 발생할 수 있으며 이 경우 표지물에 대한 질의는 연관 클래스 집합 생성을 위한 시-공간 인접 객체에서 제외(prune) 시켜야 한다. 왜냐하면 이때 표지물 정보는 클래스간의 연관 관계가 있어서 요청 된 것이 아니기 때문이다. 한 지역 내에 존재하는 표지물 객체를 결정하는 방법은 지역 전문가가 특정 객체를 선택하여 해당 지역의 표지물로 정할 수도 있고, 로그 분석을 통하여 특정 지역 범위 내에서 관련된 클래스 종류와는 무관하게 여러 질의에서 자주 요청되는 객체를 자동으로 선정할 수

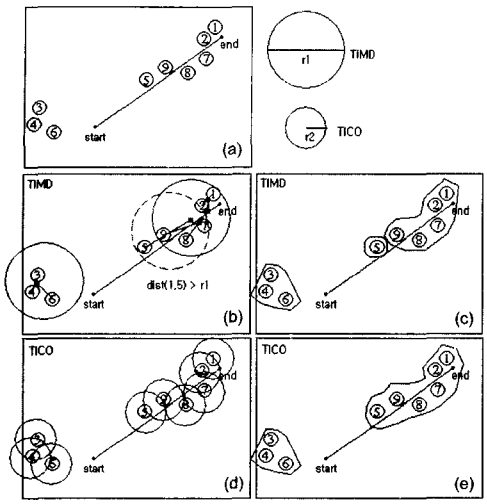

그림 2 객체의 공간상의 거리를 고려한 두 가지 시-공간 인접 객체 분리:

(a) 하나의 시간-트랜잭션에서 질의 된 객체들의 공간상의 분포
Υ_1 : TIMD에 적용되는 최대 거리, Υ_2 : TICO에 적용되는 최대 거리

(b) TIMD 방법 적용 시 객체 1과 객체 5의 거리는 최대 거리인 Υ_1보다 크므로 서로 다른 시-공간 인접 객체로 분리된다.

(c) TIMD 방법의 시-공간 인접 객체 분리 결과 {1,2,7,8,9}, {5}, {3,4,6}

(d) TICO 방법 적용시 1,2,7,8,9,5 번 객체는 모두 최대 거리인 Υ_2 보다 작은 범위 내에 근접한 연결된 이웃들로 하나의 시-공간 인접 객체를 생성한다.

(e) TICO 방법의 시-공간 인접 객체 분리 결과 {1,2,5,7,8,9}, {3,4,6}

도 있다.

3.5 연관 관계 생성

본 절에서는 생성된 시-공간 인접 객체 집합에 연관 규칙을 이용하여 연관 클래스 집합을 생성한다. 이때 실제 객체를 대상으로 연관성을 찾는 것이 아니라 객체들이 서비스하는 클래스들 사이의 연관성을 찾기 위하여 생성한 시-공간 인접 객체에 포함된 객체들을 클래스로 대치한다. 만약 하나의 시-공간 인접 객체에 속한 여러 개의 객체가 동일한 클래스에 속한다면 하나의 클래스로 대치된다. 연관 관계를 생성하기 위한 많은 알고리즘들이 개발되어 있으며 본 논문에서는 Megaputer 사의 PolyAnalyst 4.5[11]를 사용하여 연관 관계를 생성하였다.

4. 수행 평가

4.1 수행 평가 환경의 설정

이동 환경에서 알고리즘의 평가를 유효하게 수행하기 위해서는 데이타와 질의, 사용자의 이동성과 연결의 특징을 비롯한 실험 수행 환경에 대한 설정이 중요한 의미를 갖는다. 본 논문에서는 이동 환경에서의 벤치마크 기준을 설정한 [12]을 기반으로 수행 평가에 필요한 공간 데이타와 질의 테이블을 구성하였다.

실험에서 사용된 공간 데이타의 테이블 구성은 아래와 같다.

표 2 공간 데이타 테이블

Attribute	Type	Example	Note
ObjectID	Number	357	Object, total count = 50,000
ServiceID	Number	30	Class, total count =500
Name	Char(20)	Pusan Fish	Object Name
X	Number	358	X coordinate value (1~1000)
Y	Number	894	Y coordinate value (1~1000)
Attribute	Varchar2(30)	High Price	General attribute
Landmark	Char(1)	N	Y/N
Weight	Number	3	Value of object

수행 평가에서는 인위적으로 생성된 총 50,000개의 객체를 사용하였으며 각각의 객체는 점(point) 형태로 (x, y) 좌표를 이용하여 위치를 표시하는 공간 데이타이다. 대상이 되는 영역은 오른쪽 위와 왼쪽 아래 두개의 점으로 표현되는 사각형의 영역이다. 왼쪽 아래 점의 좌표를 (0,0)으로 하고, 오른쪽 위의 점의 좌표를 (1000,1000)으로 정한다. 클래스의 종류는 500개이며 하나의 클래스에는 50~150개의 객체가 속해 있다. 전 영역에 걸쳐서 200개의 표지물이 분포되어 있으며 중요도(Weight)는 해당하는 객체가 갖는 가치(value)를 수치로 표현한 것이다. 하나의 객체가 갖는 중요도는 응용 프로그램에 따라 다르게 나타난다. [13]에서는 중요도를 동일한 클래스에 속한 다른 객체들의 위치를 기준으로 한 보로노이 영역으로 보았고, 모바일 광고와 같은 응용에서는 광고주가 제공하는 요금에 비례하여 중요도를 결정할 수 있을 것이다. 본 수행평가에서는 각 객체의 중요도를 1부터 최대 20까지 임의로 할당하였다. 이때 표지물의 중요도는 10과 20으로 정하여 다른 객체에 비해 높은 중요도를 갖도록 하였다.

수행 평가에서 사용된 전체 질의 수는 90,735개이며 총 100명의 사용자를 대상으로 각 사용자별로 평균 900여 개의 질의를 수행하였다. 질의 테이블의 구성은 아래와 같다.

각 질의는 현재 사용자의 위치를 나타내는 속성으로 UserX와 UserY 값을 갖는다. Time은 질의 발생 시간으로 0부터 시작하여 증가하는 정수 값으로 정한다. 이는 특정 시간을 기준으로 N 초 이후의 시간을 나타내는

표 3 질의 데이타 테이블

Attribute	Type	Example	note
QueryID	Number	4829	Query ID, total count = 90,735
UserID	Number	29	User ID, total number = 100
PathID	Number	13	Path ID
Point	Number	2	Point ID on each path
UserX	Number	356	X coordinate of user
UserY	Number	432	Y coordinate of User
Time	Number	400	Query time, integer value start from 0
Type	Number	1	Query types
ObjectID	Number	7353	Object ID that is queried
ObjectX	Number	360	X coordinate of Object (0 ~1000)
ObjectY	Number	456	Y coordinate of Object (0 ~ 1000)

표 4 질의 형태 구분

	질의 수	종 류
Type 1	34,099 (38 %)	NLR_Q : 객체의 ID를 이용하여 사용자의 현재 위치로부터 일정 범위 내에 (distance) 위치하는 객체 조회하기
Type 2	8,400 (9 %)	LDQ : 사용자의 위치에서 가장 가까운 (closest to) 특정 서비스클래스에 속하는 객체 조회하기
Type 3	13,500 (15 %)	LAQ : 특정 객체(type2에 속함)로부터 일정 범위(distance) 이내에 있는 특정 속성을 갖는 객체 조회하기
Type 4	5,732 (6 %)	LAQ : 특정 객체(type2에 속함)와 가장 가까운(closest to) 표지물 조회하기
Type 5	29,004 (32 %)	NLR_Q : 객체의 ID를 이용한 질의로 사용자의 현재 위치와 무관한 질의

것으로 간주할 수 있다. 본 수행 평가에서는 한 질의의 결과로 요청되는 객체의 수를 하나로 제한하였고, 이때 객체의 공간상의 위치는 ObjectX와 ObjectY 속성으로 나타낸다.

이동 컴퓨팅 환경에서의 성능 평가 시 사용되는 질의는 크게 3가지로 구분된다[12]. 첫째 Location Aware Query(LAQ)는 오브젝트의 주소나 우편번호 등 위치 관련 속성을 포함하는 질의지만 사용자의 위치와는 관련이 없는 질의이다. 둘째, Location Dependent Query (LDQ)는 질의의 결과가 사용자의 현재 위치와 관련 있는 질의이다. 마지막으로 Non-Location Related Query (NLR_Q)는 상호 등을 이용하여 찾는 질의로 위치와 관련 없는 질의이다. 본 성능 평가에서는 질의의 형태를 5

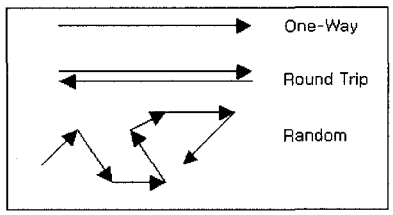

그림 3 움직임의 형태

가지로 구분하였다. 각각의 형태에 해당하는 질의의 수와 전체 질의에 대한 %는 표 4와 같다.

수행 평가에서는 각 사용자가 동일한 속도로 움직인다고 가정하였고, 방향은 8가지 방향(0도부터 45도씩 더해가면서 315도까지 8가지 방향을 설정함)으로 구성되며, 움직임의 형태는 직선형(one-way), 회귀형(Round Trip), 임의형(Random) 3가지 형태로 이루어진다. 각각의 경우를 그림으로 나타내면 그림 3과 같다.

사용자의 움직임을 결정하는 경로는 표 5와 같이 14가지로 이루어지며 각 경로의 형태와 해당 경로의 밀집도는 아래의 표와 같다. 이때 각 경로의 밀집도는 전체 영역의 평균 객체의 밀집 정도를 100으로 할 때 해당 경로를 중심으로 분포한 평균 객체의 밀집 정도를 비교한 것이다.

4.2 수행 평가 결과

4.1절에서 생성한 데이타를 기준으로 시간 윈도우(TW), 최대 거리(MD), 지지도(Support) 등의 입력 파라메터를 변화하면서 네 가지 수행 평가를 실시하였다.

4.2.1 실험 1 : 시간 제약 기준인 시간 윈도우(TW) 증가에 따라 식별되는 시간-트랜잭션 수의 변화

실험 1에서는 질의 시간 순서로 정렬된 사용자별 시퀀스를 대상으로 TW를 20에서 60까지 변화하면서 식별되는 시간-트랜잭션 수의 변화를 살펴보았다. 그 결과 전체적으로 TW가 커질수록 식별되는 시간-트랜잭션의 수는 줄어든다. 이때 TW가 너무 작게 설정되는 경우에는 연관성이 있는 질의들이 서로 다른 시간-트랜잭션으로 분리되는 가능성이 있고, 반대로 TW가 너무 크게 설정 되는 경우에는 연관성이 없는 질의들이 하나의 시간-트랜잭션에 포함되게 되므로 적절한 TW의 설정은 의미 있는 연관 클래스 집합 생성에 중요한 요인이 된다. 그림 4에서 TW가 20에서 30으로 변화하는 구간의 기울기가 다른 구간에 비해 급격하게 변화하는 것을 볼

표 5 경로의 type과 밀집도

PathID	1	2	3	4	5	6	7	8	9	10	11	12	13	14
밀집도	175	127	166	66	77	65	175	166	66	77	131	217	64	67
형태	직선형						회귀형				임의형			

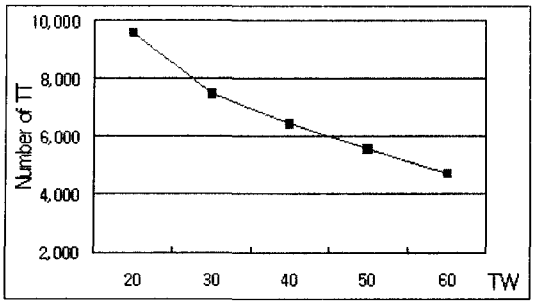

그림 4 TW 변화에 따른 시간-트랜잭션(TT) 수

수 있는데 이는 많은 수의 연속적인 질의가 시간 간격 30 이내에 발생된다는 것을 의미한다. 이러한 경우에 30은 시-공간 인접 객체 생성을 위해 의미 있는 시간 윈도우 값으로 볼 수 있다.

4.2.2 실험 2 : TIMD 알고리즘을 이용한 최대 거리(MD) 증가에 따라 식별되는 시-공간 인접 객체 수의 변화

실험 2에서는 TW를 30으로 하여 생성된 시간-트랜잭션을 대상으로, 객체들 사이의 거리를 기준으로 시-공간 인접 객체를 만들 때, MD을 60부터 100까지 변화하면서 식별되는 시-공간 인접 객체 수의 변화를 살펴보았다. 또한 생성된 시-공간 인접 객체를 대상으로 단일 시-공간 인접 객체를 제거하였고, 객체 수가 두 개 이상인 시-공간 인접 객체에 대해 평균 포함 객체 수를 구하였다. 실험 결과 전체적으로 MD가 커질수록 식별되는 시-공간 인접 객체의 수는 줄어들었고, 평균 포함 객체의 수는 증가하였다. 이때 입력 파라메터인 MD가 너무 작게 설정되는 경우에는 단일 시-공간 인접 객체의 수가 증가하여 많은 수의 시-공간 인접 객체가 제거되는 결과를 볼 수 있다. 결과는 그림 5, 6과 같다.

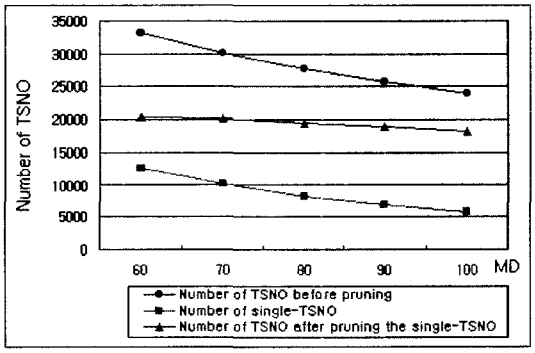

그림 5 TIMD 사용 시 MD증가에 따라 식별되는 시-공간 인접 객체 수

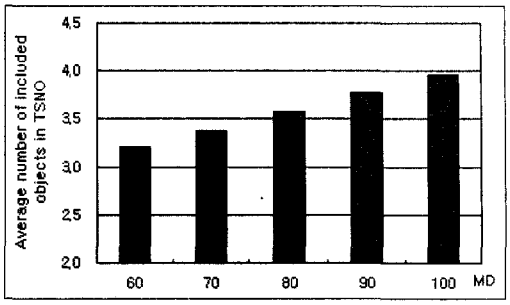

그림 6 TIMD 사용 시 MD 변화에 따른 시-공간 인접 객체들의 평균 포함 객체 수

4.2.3 실험 3 : TICO 알고리즘을 이용한 최대 거리(MD) 증가에 따라 식별되는 시-공간 인접 객체 수의 변화

실험 3에서는 TW 30을 기준으로 생성된 시간-트랜잭션을 대상으로 TICO 알고리즘을 이용하여 시-공간 인접 객체를 식별할 때, MD을 10부터 50까지 변화시키면서 식별되는 시-공간 인접 객체 수의 변화를 살펴보았다. 또한 단일 시-공간 인접 객체를 제거한 후, 시-공간 인접 객체에 대한 평균 포함 객체 수를 구하였다. 실험 결과 전체적으로 MD가 커질수록 최초로 생성되는 시-공간 인접 객체의 수는 줄어들었고, 평균 포함 객체의 수는 증가하였으나 MD가 너무 작은 경우에는 단일 시-공간 인접 객체의 수의 높은 증가로 많은 수의 시-공간 인접 객체가 정제되는 것을 볼 수 있다. 그 결과 최종적으로 생성되는 시-공간 인접 객체의 수는 오히려 MD 값이 40이 될 때까지 증가하는 결과를 볼 수 있다. 또한 그림 6에서와 같이 MD의 크기가 30 이상이 되는 경우에는 최종적으로 생성되는 시-공간 인접 객체의 수의 변화가 거의 없는 결과를 보이는데 이러한 MD 값은 시-공간 인접 객체를 생성하는 의미 있는 최대 거리 값으로 볼 수 있다. 결과를 그림으로 나타낸 것은 그림 7,

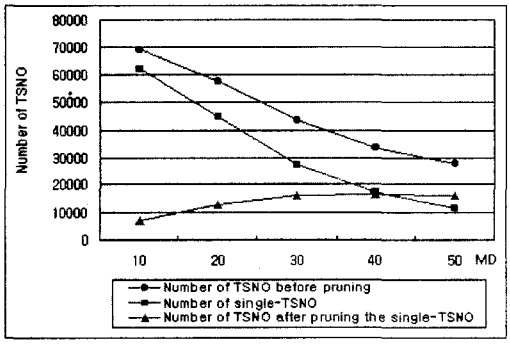

그림 7 TICO 사용 시 MD증가에 따라 식별되는 시-공간 인접 객체 수

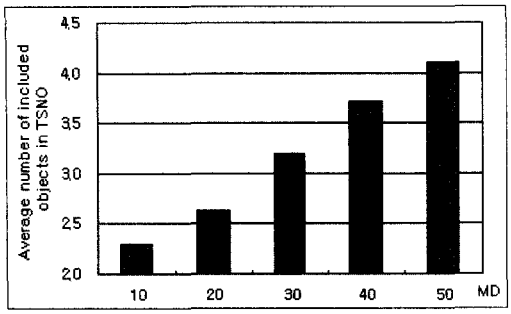

그림 8 TICO 사용 시 MD 변화에 따른 시-공간 인접 객체들의 평균 포함 객체 수

8과 같다.

4.2.4 실험 4 : 지지도 증가에 따라 식별되는 연관 클래스 집합 수의 변화

실험 4에서는 TW = 30, MD = 30일 때 TICO를 이용하여 생성한 시-공간 인접 객체를 이용하여 연관 규칙을 수행하였다. 이때 지지도의 값을 0.01%부터 0.15%까지 변화시키면서 생성되는 빈발 항목 즉 연관 클래스 집합 수의 변화를 살펴보았다. 실험 결과 지지도가 높아짐에 따라 생성되는 연관 클래스 집합의 수가 급격히 줄어드는 것을 볼 수 있었고 결과를 정리한 것은 그림 9와 같다. 이때 생성되는 연관 클래스 집합의 형태는 표 6과 같다. 표 7에서는 생성된 연관 클래스 집합을 이용하여 연관 규칙을 생성한 결과의 일부를 나타낸다. 이때 신뢰도(confidence)를 구하는 식은 아래와 같다.

$$\text{confidence } (A \Rightarrow B) = P(B|A) = \frac{\text{sup\_ count } (A \cup B)}{\text{sup\_ count } (A)}$$

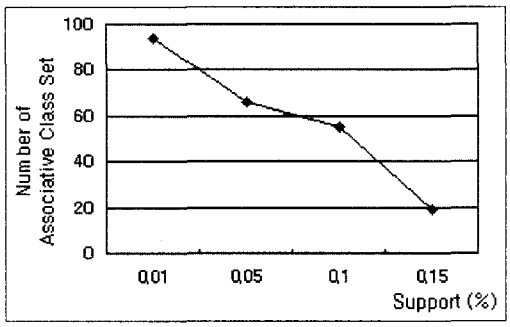

그림 9 지지도 변화에 따라 식별되는 연관 클래스 집합의 수

5. 결 론

본 논문은 이동 컴퓨팅 환경에서 대량의 공간 데이타베이스를 기반으로 하는 위치 기반 서비스 응용 시스템에서 이동 사용자의 질의를 효율적으로 지원하기 위하여 이동 사용자가 요청하는 서비스간의 시간적 연관 관계, 서비스를 제공해 주는 공간 객체들 사이의 거리와 사용자의 질의 패턴을 동시에 고려한 연관 클래스 집합을 제안하였고, 이를 효과적으로 생성하기 위해 시간적 제약사항을 적용한 시간-트랜잭션 생성 알고리즘과 공간적 제약 사항을 반영한 시-공간 인접 객체 생성 알고리즘을 제안하였으며, 실험을 통해 연관 관계를 생성하였다. 생성된 연관 클래스 집합은 이동 컴퓨팅 환경에서 지리 정보를 서비스 할 때 관련 자료를 추천하는 시스템에 활용할 수 있고, 지리 정보를 고려한 광고 방송이나 도시 개발 계획 등에 이용할 수 있으며, 이동 사용자를 위한 클라이언트의 캐쉬 정책에 응용 될 수 있다. 본 논문에서는 연관 클래스 집합을 찾을 때 사용자별로 구하지 않고 서버의 질의 데이타 전체를 대상으로 연관 관계를 생성하였다. 그러나 향후에는 사용자별 history를 기반으로 개인의 특성에 맞는 연관 관계를 생성하여 개인화 등의 응용에 사용할 것이다. 또한 본 논문에서 생성한 연관 클래스 집합을 이용하여 위치 기반 서비스를 수행하는 응용 프로그램을 위한 이동 호스트의 캐쉬와 선반입 정책에 적용하는 방법에 대한 연구를 수행할 것이다.

표 6 연관 클래스 집합의 예

Associative class set # 1	3 classes contain	support count = 25
	class 85	
	class 43	
	class 122	
Associative class set # 2	3 classes contain	support count = 16
	class 452	
	class 367	
	class 347	

표 7 생성된 연관 규칙의 예

	support	confidence
class 43 , class 122 => class 85	0.17%	89.29
class 85 , class 122 => class 43	0.17%	92.59
class 85 , class 43 => class 122	0.17%	89.29
class 367 , class 347 => class 452	0.11%	100
class 452 , class 347 => class 367	0.11%	88
class 452 , class 367 => class 347	0.11%	47

참 고 문 헌

[1] Tomasz Imielinski, Henry F. Korth, "Introduction to Mobile Computing," Mobile Computing, Kluwer Academic Publishers, pages 1-43, 1996.

[2] 류근호, 안윤애, 이준욱, 이용준, "이동 객체 데이타베이스와 위치 기반 정보 검색 서비스의 적용", 한국 정보 과학회 데이타베이스 연구회지, 17권 3호, pp. 57-74, 2001.

[3] Rakesh Agrawal, Tomasz Imielinski and Arun

Sqami, "Mining Association Rules beTWeen Sets of Items in Large Databases," In proceedings of the International Conference on Management of Data (SIGMOD), pp.207-216, 1993.

[4] Rakesh Agrawal and Ramakrishnan Srikant, "Mining Sequential Patterns," In proceedings of International Conference on Data Engineering, pp. 3-14, 1995.

[5] Ramakrishnan Srikant and Rakesh Agrawal, "Mining Sequential Patterns: Generalization and Performance Improvements," Research Report RJ 9994, IBM Almaden Research Center, San Jose, California, 1995.

[6] Yasuhiko Morimoto, "Mining Frequent Neighboring Class Sets in Spatial Databases," In proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2001, pp. 353-358, 2001.

[7] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, Pang-Ning Tan, "Web Usage Mining : Discovery and Applications of Usage Patterns from Web Data," SIGKDD Explorations. ACM SIGKDD, Jan 2000.

[8] K.Koperski, J.Han, "Discovery of spatial association rules in geographic information databases," In Proc. 4th Int'l Symp. on Large Spatial Databases (SSD'95), pp. 47-66, 1995.

[9] Wen-Chih Peng and Ming-Syan Chen, "Mining User Moving Patterns for Personal Data Allocation in a Mobile Computing System," the Proceedings of the 29th International Conference on Parallel Processing (ICPP-2000), August 21-24, 2000.

[11] PolyAnalyst 4.5, http://www.megaputer.com.

[10] 서영덕, 안경환, 홍봉희, "인터넷 GIS의 사용 분석", 한국 정보 과학회 데이타베이스 연구회지, 18권 1호, pp. 41-52, 2002.

[12] Ayse Y. Seydim, Margaret H.Dunham, "A Location Dependent Benchmark with Mobility Behavior," International Database Engineering and Applications Symposium (IDEAS'02), pages 74 -85, 2002.

[13] Baihua Zheng, Jianliang Xu, Dik L.Lee, "Cache Invalidation and Replacement Strategies for Location-Dependent Data in Mobile Environment," IEEE Transactions on Computers, vol. 51, No. 10, pages 1141-1153, October 2002.

김 호 숙
정보과학회논문지 : 데이타베이스
제 31 권 제 2 호 참조

용 환 승
정보과학회논문지 : 데이타베이스
제 31 권 제 2 호 참조