DOI: 10.3745/KIPSTD.2008.15-D.2.179

이동 객체 데이터베이스에서 빈발 시퀀스 패턴 탐색

Vu, Thi Hong Nhan[†] · 이 범 주[††] · 류 근 호[†††]

요 약

위치 기반 장치의 발전과, GIS 기능의 확장 그리고 위치 정보기술들의 정확성과 가용성이 증가함에 따라서 위치 기반 서비스들의 새로운 영역에 대한 새로운 가능성이 나타나게 되었다. 데이터의 시간과 공간 형태에 따라서 정의되는 Relationship에 기인하여 시공간 데이터 마이닝 영역에서 공간에 대한 지식 검색이 증가할 경우 매우 큰 문제에 직면한다. 이 논문에서는 모바일 환경에서 시공간 패턴 마이닝을 위한 알고리즘들을 제안한다. 이동 패턴들은 All_MOP와 Max_MOP 두 개의 알고리즘을 활용하여 생성된다. 이 알고리즘들은 먼저 모든 빈발 패턴들을 탐사한 후 오직 최대의 빈발 패턴만을 탐사한다. 아울러, 제안한 기법과 기존의 DFS_MINE 기법의 수행 시간 비교를 통하여 제안한 기법이 수행시간에서 다소 우수한 것을 나타낸다. 이러한 제안접근법은 관광 서비스, 교통 서비스 등과 같은 위치 기반 서비스 등에 활용할 수 있다.

키워드 : 시공간데이터마이닝, 이동객체, 빈발패턴

Discovery of Frequent Sequence Pattern in Moving Object Databases

Vu, Thi Hong Nhan[†] · Bum Ju Lee[††] · Keun Ho Ryu[†††]

Abstract

The converge of location-aware devices, GIS functionalities and the increasing accuracy and availability of positioning technologies pave the way to a range of new types of location-based services. The field of spatiotemporal data mining where relationships are defined by spatial and temporal aspect of data is encountering big challenges since the increased search space of knowledge. Therefore, we aim to propose algorithms for mining spatiotemporal patterns in mobile environment in this paper. Moving patterns are generated utilizing two algorithms called All_MOP and Max_MOP. The first one mines all frequent patterns and the other discovers only maximal frequent patterns. Our proposed approach is able to reduce consuming time through comparison with DFS_MINE algorithm. In addition, our approach is applicable to location-based services such as tourist service, traffic service, and so on.

Key Words : Spatiotemporal Data Mining, Moving Object, Frequent Patterns

1. 서 론

최근 데이터베이스로부터의 지식 탐사는 실 세계 응용에서 발생하는 방대한 양의 데이터에 대한 중요한 분석 프로세스로 여겨지고 있다. 따라서 위치기반 서비스, 위치기반 추천서비스, 지능적 물류 관제 서비스, 응급 서비스 같은 시공간 응용분야에서 그 필요성이 증대되고 있다.

시공간 지식 탐사는 시공간 객체들의 변화 이력인 시공간 데이터 집합으로부터 변화에 대한 시간 규칙, 공간 규칙, 시공간 규칙등과 같이 유용한 지식 탐사를 위한 새로운 연구 분야이다. 시공간 데이터 집합에서의 시간 규칙이나 공간 규칙의 탐사는 기존의 시간 데이터 마이닝 기법이나 공간 데이터 마이닝 기법 등을 통하여 탐사될 수 있다. 그러나 시공간 규칙은 객체의 시간 요소와 공간 요소를 함께 고려하여야만 탐사가 가능하다. 시공간 이동 패턴은 이동하는 객체의 위치 패턴으로써 고객의 위치 특성에 따라 개인화되어질 수 있는 시공간 규칙이며, 또한 알맞은 콘텐츠나 서비스 제공을 가능하게 할 수 있는 시공간 규칙이다. 그러나 기존에 제안된 시간 데이터 마이닝이나 공간 데이터 마이닝 기법은 시공간 규칙이나 이동 패턴을 탐사하기가 어렵다. 따라서 동시에 시간과 공간 데이터를 고려하여 시공간 이동 패턴을 탐사할 수 있는 기법이 필요하다. 이 논문에서는 유용한 시공간 규칙을 생성하기 위해 이동 객체 데이터로부터 빈발한 시퀀스 패턴을 탐색하기 위한 기법을 제안한다. 제안한 기법에서는 All_MOP과 Max_MOP의 두 알고리즘을 통해서 이동 객체 데이터로부터 모든 빈발한 패턴과 최대 패턴들을 탐색한다. 이 기법에서는 시간에 따라 순서화된 연관성의 조합을 이용하여 모든 가능한 빈발 패턴을 탐사한 후 탐사된 빈발 패턴 중 최대 빈발 패턴 탐사에 의해 시공

※ 이 논문은 2006학년도 충북대학교 학술연구지원사업의 연구비지원에 의하여 연구되었음.
† 정 회 원 : 한국전자통신연구원, RFID/USN 미들웨어 연구팀
†† 준 회 원 : 충북대학교 전기전자컴퓨터공학부 전자계산학과 박사수료
††† 종신회원 : 충북대학교 전기전자 및 컴퓨터공학부 교수(교신저자)
논문접수 : 2007년 8월 8일, 심사완료 : 2007년 11월 30일

간 규칙을 탐사하게 된다.

이 논문의 구성은 다음과 같다. 2장에서는 시공간 데이터로부터 지식을 탐사하기 위한 기존 연구들을 고찰한 후, 기존 연구의 문제점 및 해결 방안을 기술한다. 3장에서는 시공간 빈발 패턴 탐사를 위해 먼저 이동객체 데이터 및 이동객체의 궤적을 모델링하기 위한 전처리 과정을 기술한다. 4장에서는 처리된 데이터와 궤적에 대하여 시공간 빈발 패턴을 탐사하기 위한 알고리즘을 상세히 기술한다. 그리고 5장에서는 제안한 알고리즘의 구현을 통하여 탐사되는 시공간 패턴에 대한 실험 평가를 수행 후 6장에서 결론을 맺는다.

2. 관련연구

이 장에서는 시공간 마이닝 분야에서 시퀀스 패턴을 탐색하기 위하여 기존에 수행된 연구들에 대해서 기술하고, 현재 이동 시퀀스 패턴 마이닝에 대하여 기술한다.

2.1 시퀀스 패턴

시퀀스 패턴 [1], [7], [8], [12]은 일반적으로 대용량 고객 거래 데이터베이스의 내부 거래 패턴 탐사로 정의된다. 이러한 시퀀스는 시간적으로 순서화된 항목들의 집합이다. 시퀀스 패턴 탐사 기법들의 구분은 공간 검색 탐색 기법과 항목들의 지지도 값 결정에 따라 구분된다[13]. 최근에 이르러, 공간 검색 기반 시퀀스 패턴 탐사는 너비우선 검색이나[14] 깊이 우선 검색 [15],[16]을 이용한다. 따라서 두 가지 시퀀스 패턴 탐사 기법에 대해서 자세히 기술한다.

공간 검색 기반 탐색에 의한 기법: 공간 검색의 탐색은 항목 지지도의 인접 속성에 따른다. 즉, 빈발 항목의 모든 서브집합들은 빈발해야만 한다. 이 속성은 비빈발 항목집합으로부터 빈발한 것을 분류하는 경계에 의존한다. 그러므로 조사할 수 있는 항목집합의 수를 어떻게 제한할 것인가 하는 문제가 발생한다. I={1,…,|I|}는 자연수의 집합에서 항목들과 매핑된다고 가정하자. 여기서 항목들은 자연수들 간의 '<' 관계에 의해 전체적으로 순서화 될 수 있다. XI에 대해서, f:X는 i번째 항목 xi 가 '<' 에 의해 점진적으로 정렬되는 xi=f(xi)가 되었을 때 매핑 된다. 항목집합 X의 n-prefix는 Pre = {xi | xi I X, 1£ i £ n}로 정의된다. 클래스를 C(Pre)로 표기할 경우, Pre I는 C(Pre) ={X I I / |X| = |Pre| + 1와 함께 I이고 Pre는 트리의 노드들로서 X의 prefix 이다. 두 노드들은 에지에 의해 링크되며, 클래스 C의 모든 항목집합들이 부모 클래스 C′의 두 항목집합의 조인에 의해 생성될 수 있다. 항목집합의 하위 인접 속성과 함께 클래스 C의 부모 클래스 C′이 최소 두개의 빈발 항목집합을 포함하지 않는 다면 클래스 C는 임의의 빈발 항목도 포함할 수 없다는 것을 의미한다. 만약 어떤 클래스 C′이 트리 아래의 경로와 만나게 된다면 비빈발항목과 빈발 항목을 구분하는 경계가 만나는 것이다. 이것은 이 경계에서 불필요하기 때문에 공간으로부터 C의 모든 자손들이 제거된다. 따라서, 경계에서 검색을 위해 탐사된 항목들만은 지지도 값들을 결정하는 것이 필요하다.

항목들의 지지도 값 결정에 의한 기법: 이 기법은 두 가지 유형으로 구분될 수 있다. 첫 번째는 데이터베이스에서 발생하는 항목들의 직접적인 카운트를 통해 항목의 지지도를 결정할 수 있는 것이고, 두 번째는 교집합에 의해 항목들의 지지도를 결정하는 것이다. 첫 번째 유형에서의 서브셋 생성과 후보항목 검색은 해쉬 트리나 유사한 데이터 구조를 이용한다. 즉, 각 트랜잭션의 모든 서브셋이 생성되고 서브셋이 후보항목들에 포함되거나 또는 최소 하나의 후보항목들과 함께 공통적으로 나타나게 된다. FP-growth [16]가 대표적인 예이다.

교집합을 이용하여 후보항목들의 지지도를 결정하는 유형은 트랜잭션의 식별자 tid를 둔다. 단일 항목의 tidlist는 이 항목을 포함한 트랜잭션과 그에 대응되는 식별자들의 집합이다. 이러한 것은 tidlist에 모든 항목집합 X가 존재하고 X.tidlist에 의해 표시될 수 있다. 그러므로, 후보항목 Z = X ∈ Y의 tidlist는 Z.tidlist = X.tidlist ∈ Y.tidlist에 의해 얻을 수 있다. 그리고 tidlist는 효율적인 교집합을 허용하기 위한 메모리에서 오름차순 순서로 정렬된다. 마지막으로, 항목집합의 실제 지지도는 |Z.tilist|의 카디널리티가 된다. Eclat[15]가 이 기법의 대표적인 예이다.

2.2 이동 시퀀스 패턴 마이닝

이동 시퀀스 패턴 마이닝은 시간과 공간 차원[5], [10], [11]을 수반하는 객체 위치 정보의 시계열을 다룬다. 시간 및 공간 데이터 마이닝에 대한 이전연구들은 시간과 공간 마이닝[2], [3], [6]에 대부분 초점을 맞추고 있다. 비록 시공간 마이닝에 대한 일부 연구가 수행되어 오고 있지만 이들 연구에서는 주로 이동 객체[9] 인덱싱을 위한 모델과 구조에 초점을 맞추고 있다. 최근 연구[8]에서는 날씨 예측을 위한 시공간 시퀀스 패턴을 탐사하였다. 이 연구에서는 고정된 위치에 대해서 속성들의 시간 변화에 대한 관계를 탐색한다. 그러나 위치 변화와 함께 객체들의 고정된 속성들 간의 관계를 탐색하기 위해 필요한 이동 객체에 대한 패턴을 어떻게 적용할 것인가를 보여주지 못하고 있다.

이 논문에서는 모바일 환경에서 시간과 공간변화에 기반을 둔 시퀀스 패턴을 탐색하는 기법을 제안한다. 아울러 이 논문에서 제안하는 빈발 패턴 탐색 기법과 기존 기법과의 실험 평가를 통하여 제안하는 알고리즘이 시간 효율성 측면에서 약간 우수한 것을 기술한다.

다음 절에서는 제안한 시공간 시퀀스 패턴 탐사 이전에 앞서, 이동 객체 데이터와 이동 객체의 궤적에 대한 모델링 처리를 위한 전 처리 프로세스에 대해서 기술한다.

3. 전 처리 프로세스

이 장에서는 빈발 시퀀스 패턴 탐사를 위한 전처리 과정으로서 이동 객체 데이터의 모델링과 이동 객체 궤적 모델링 및 이동 객체의 위치 데이터를 일반화하는 과정에 대해

서 기술한다.

3.1 데이터 모델링

이동 객체 데이터베이스 D는 위치들의 시간 값들의 합 ($D = \bigcup_{i=1}^{No} D_i$)으로 정의된다. 연속적인 시간 값 Di는 이에 대응하는 객체의 튜플(x, y, t)에 포함된다. 표 1은 이러한 객체들의 튜플을 나타내는 이동 객체 데이터베이스의 예를 나타낸 것이다.

객체들은 명시된 영역(A∈R2, 참조영역)내에서 이동한다. 이동 객체의 위치 정보는 불연속적인 메소드를 사용하여 기술되며, 영역 A의 공간 구조는 분할된 영역의 집합으로 표현된다. 즉, 영역 A는 같은 크기의 유한 집합{$c_1,\cdots,c_n$} ($\cup_{i=1}^{n} c_i = A$, ci ∩ cj=∅, i≠j)으로 분할된다. 또한, 객체들의 이동 경로를 나타내는 궤적들은 해쉬 함수를 이용하여 참조영역 내에서 영역별로 저장된다. 이 논문에서는 각 객체가 최소 시간 인터벌에서 각 영역 내에 위치한다고 가정하고, 각 영역을 위한 최소 하나의 포인트를 제공한다. 그 결과로 궤적은 영역 레이블의 시퀀스로서 정의된다. 그러나 한 셀은 시퀀스에서 몇몇 시간들로 나타나지만, 유용하지 못한 정보를 제공한다. 따라서 우리는 eliminate() 함수를 반복 사용하여 불필요한 정보들을 제거한다. 이러한 함수를 위하여 이동객체의 궤적은 oid.traj = {p_1, $p_2,\cdots,p_n$}로 정의한다. 여기서 p_n은 이동 객체의 포인트를 의미하며, oid.traj의 패턴은 이동 시퀀스로서 moving sequence(oid.traj) = <eliminate(eliminate (label(p_1), label(p_2)),⋯,label(p_n))>로 정의한다.

최소 시간 인터벌과 궤적들의 존속기간 max_span이 주어지면, 이동 시퀀스는 시간적으로 순서화된 영역 레이블의 리스트가 된다. 즉, ms=<c_1, $c_2,\cdots,c_q$>이다. 여기서 c_i는 c_i - c_i-1을 가진 셀 ID이고, 1≤i≤q와 cq-c1≤max_span이 된다. 시퀀스는 K-패턴으로 표기된 K영역으로 구성되어 있다. 이동 시퀀스 S1을 위해서, 영역 c_1이 c_2 이전에 발생한다면, 우리는 영역을 c_1<c_2로 표기한다. 또한 S1 영역이 S2 영역과 일대일로써 순서가 그대로 보존되는 함수 f가 존재한다면, S1은 다른 시퀀스 S2의 서브 시퀀스라고 한다. 즉 1) ci=f(ci), 2) ci < cj 일때 f(ci) < f(cj)이다. 예를 들어, 이동 시퀀스 S1=<c2 c3 c4>가 다른 이동 시퀀스 S2=<c1 c2 c3 c5 c4>의 서브 시퀀스임을 보여준다. 여기에서 이동 시퀀스 <c2 c3 c4>의 모든 영역은 다른 이동 시퀀스 <c1 c2 c3 c5 c4>의 영역과 같이 대응되며, 영역의 순서 역시 보존되어 있음을 의미한다. 즉, 시퀀스 S3 = <c1 c6> 은 다른 이동 시퀀스 S4 = <c1 c2 c3 c5 c4>의 서브 시퀀스가 되지 못한다.

서브시퀀스의 집합을 MS = {$ms_1,\cdots,ms_m$}라고 가정하자. 각 ms_1 은 이동 시퀀스 패턴 1≤ i ≤ m 이다. 시퀀스 ms의 지지도는 ms를 포함하는 모든 시퀀스들의 일부로 정의될 수 있다. 사용자가 명시한 최소 지지도(min_sup)는 각 시퀀스를 만족하는 최하 값이다. 만약 시퀀스 ms가 지지도(ms)≥min_sup를 만족한다면, 이것은 빈발 시퀀스로서 정의된다. 또한 임의의 다른 시퀀스들의 서브 시퀀스가 아니라면 빈발 시퀀스는 최대가 된다. 따라서 이동 객체 데이터베이스에서 모든 빈발한 시공간 시퀀스 패턴을 탐사하기 위해서는 궤적들의 존속기간 (max_span)과 최소 지지도(min_sup)를 만족해야만 한다.

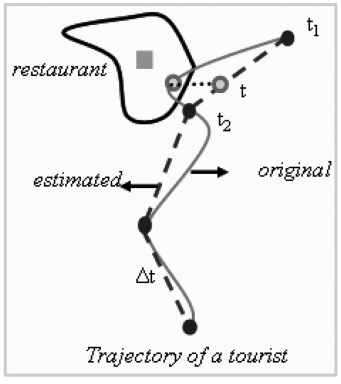
〈표 1〉 이동객체 튜플을 나타내는 데이터베이스의 예

oid	vt	x	y
o1	2006/1/30/ 7:00	10	0
	2006/1/30/ 7:30	20	5
	2006/1/30/ 8:00	25	7
	2006/1/30/ 8:30	35	9
	2006/1/30/ 9:00	33	13
	2006/1/30/ 9:30	28	13
o2	2006/1/30/ 7:00	18	12
	2006/1/30/ 7:30	16	22
	2006/1/30/ 8:00	16	29
	2006/1/30/ 8:30	18	35

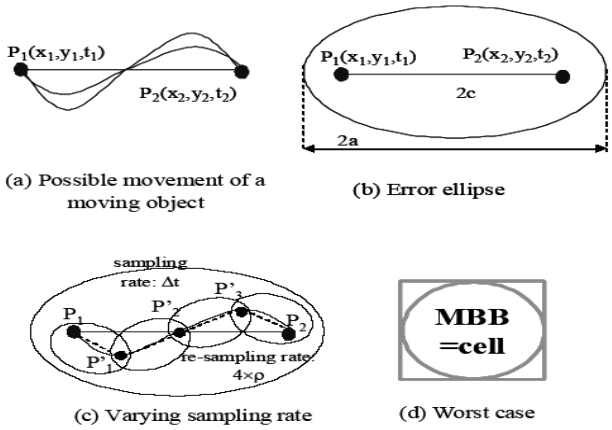
3.2 이동객체의 궤적 모델링

각 이동 차량은 GPS가 부착되어 있어 주기적으로 위치 값이 샘플링 된다고 가정한다. 샘플링 처리는 샘플화 된 데이터의 오류 또는 잘못된 값을 정제하는 과정이다. 그림 1은 관광객의 샘플화 된 포인트들을 보유한 궤적의 예를 묘사한 것이다.

우리는 객체들의 정밀한 표현을 위해서 다음과 같이 두 가지의 가정을 기반으로 샘플화 된 데이터 포인터를 저장한다. 1) 측정 시간에 연결된 에러가 없어야 한다. 2) 하나의 응용 내에서 속도를 고려하였을 때 유사한 궤적을 가지고 있어야 한다. 그러나 실생활에서 이동 객체의 위치는 규칙적인 시간 인터벌에서 GPS 수신자를 이용하여 샘플링 된다. 샘플링 에러는 주어진 객체의 최대 속도 vmax와 시간 t1에서 측정된 P1의 연속적인 두 위치에 의해 타원형의 에러가 제공된다. 이 타원형의 에러는 라인 세그먼트당 샘플링 에러의 크기를 측정하기 위해 사용된다. 최악의 경우는 타원형의 원형을 줄이는 것으로서 그림 2와 같이 고려해야

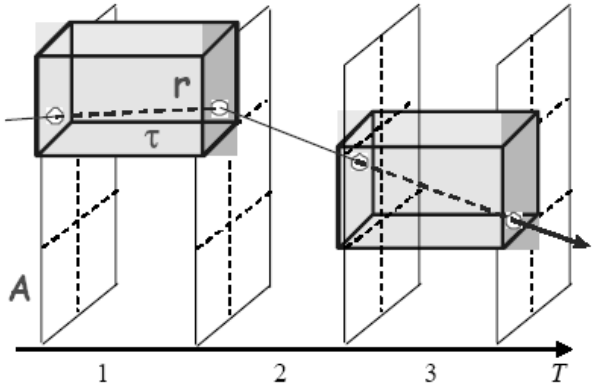


(그림 1) 샘플화 된 포인트들의 궤적 예



(a) Possible movement of a moving object (b) Error ellipse (c) Varying sampling rate (d) Worst case

(그림 2) 다양한 샘플링과 샘플률에 대한 불확실성



(그림 3) 시공간 단위

한다. 좀 더 유용한 연산을 위해서 이 논문에서는 최소 경계 사각형을 사용하여 검색 공간의 셀을 정확히 하였다.

추가적으로, 그리드 임계치 r을 위해 시간적인 확장을 고려하지 않은 참조 영역 A는 같은 크기의 (nx x ny) 셀 배열로 분할된다. 하지만 참조 영역 A가 시간적인 확장을 고려한다면 (그림 3)과 같이 시공간 큐브의 정형화된 시공간 단위로 분할된다.

(그림 3)에서 시공간 단위는 최소 공간과 시간 확장으로 정의된다. 궤적들이 넓게 흩어져 있다면 확실한 시간 인터벌에서 공간적으로 셀을 요구하며, 이때 저장 수단으로는 래스터를 사용한다. 또한 셀 크기 r의 선택은 정확도에 영향을 미치게 된다. 즉, 재 샘플율 ρ와 셀 크기는 객체가 이동하는데 있어서 각 셀에 최소 한번 방문 할 수 있도록 하기 위해서 선택되어야 한다. 여기서 파라미터 r 은 $(v_{max}/\rho) \ll (r/\sqrt{2})$와 같이 선택되어야 한다. 아울러 시간 확장은 미리 결정되어야 하고 애플리케이션에 의존적으로 변경된다. 이것은 $1 \ll \rho\tau$,처럼 선택되고 셀 당 방문 회수를 예상하기 위해 측정된다. 또한 (x0,y0) 좌표를 가진 점들로 구성된 객체 공간은 규칙적인 그리드로서 표현되고 DIR[1:nx,1:ny]배열로 저장된다. 각 엘리먼트 DIR[i,j]는 위치들이 할당된 셀 레이블 cij와 대응된다. [Sx,Sy]가 검색 공간의 2차원 크기면, 각 셀은 [Sx/nx, Sy/ny]의 크기를 가진다. P(a,b) 포인트를 위해서 P는 i= (a-x0)/(Sx/nx) +1에 의해 결정되며, 셀 cij 의 식별자는 j= (b-x0)/(Sy/ny) +1이다. 여기서, 우리는 각 셀이 상위 오른쪽 경계에 대해 오픈되고 하위 왼쪽 경계에 대해서는 폐쇄된 것을 적용한다(그림 4). 모든 궤적은 셀 레이블 cij의 집합으로 변환된다.

3.3 이동객체의 위치정보 일반화

데이터베이스의 한 튜플에 여러 번 액세스가 이루어 질 때, 우리는 액세스 카운트를 한번으로 가정한다. 이것은 eliminate() 함수를 표현하기 전에 궤적들의 위치를 일반화 할 필요가 있기 때문이다. 셀들로 이동 객체들의 위치를 해싱한 후 (그림 5)의 (a) 데이터 집합으로부터 (b)와 같은 결과를 얻을 수 있다.

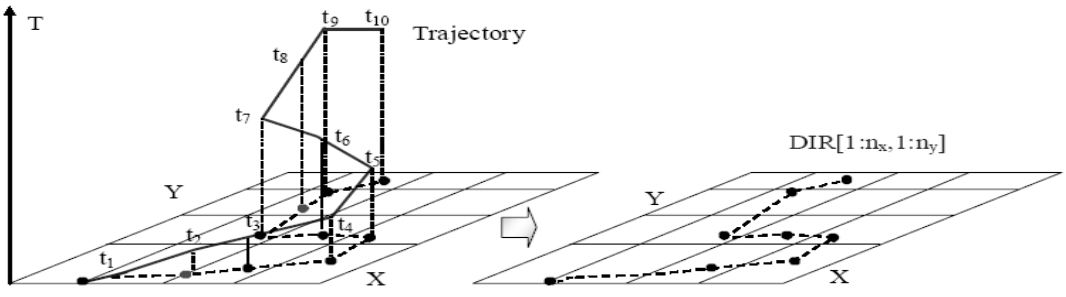
oid	vt	Location
o1	2006/1/30/ 7:00	D10
	2006/1/30/ 7:30	D20
	2006/1/30/ 8:00	D20
	2006/1/30/ 8:30	D30
	2006/1/30/ 9:00	D31
	2006/1/30/ 9:30	D21
o2	2006/1/30/ 7:00	D11
	2006/1/30/ 7:30	D12
	2006/1/30/ 8:00	D12
	2006/1/30/ 8:30	D13

(a) Moving points converted into cell labels

oid	vt	Location
o1	2006/1/30/ 7:00	D10
	2006/1/30/ 7:30	D20
	2006/1/30/ 8:30	D30
	2006/1/30/ 9:00	D31
	2006/1/30/ 9:30	D21
o2	2006/1/30/ 7:00	D11
	2006/1/30/ 7:30	D12
	2006/1/30/ 8:30	D13

(b) Result of generalization

(그림 5) 위치 정보 일반화



(a) All cells the projected trajectory crosses (b) Trajectory generalization

(그림 4) 참조 영역에서의 궤적

[알고리즘 1] 이동 시퀀스 생성 알고리즘

```
Function MovingSequenceExtraction()
    - DB: moving object database
    - A: reference map
    - r: cell size
    - max_span: trajectory's lifespan
Output: set of moving sequences (MS)
Begin
      DIR[1:nx,1:ny]←PartitionReferencePlane(A,r);
      SortedDB ←sort database DB by (oid,t);
      D ←Select from DB trajectories whose lifespan is during the time interval
      max_span[statrt_time, end_time]
      MS ← Ø // initialize the set of moving sequences
      For all moving objects oid in D sorted according to oid and timestamp vt do
            Traj ←DIR[p1.x; p1.y];
            PrevLoc 1; //previous location of moving object
            For (i=2; i<=oid.numberOfPositon() i++)
                  If ( DIR[pi.x; pi.y] ≠ DIR[pi-1.x; pi-1.y] ){
                        Traj ← Traj ∪ DIR[pi.x; pi.y];
                        PrevLoc ← i;
                  }
      MS ← MS ←Traj
     return MS
End
```

4. 시공간 시퀀스 패턴 탐사

이 장에서는 앞 장에서의 수행한 전처리 되어진 데이터베이스로부터 빈발한 이동 시퀀스 패턴들을 탐색한다. 모든 이동 시퀀스 패턴들을 생성하는 단계는 두 단계로 이루어진다. 첫 번째는 모든 빈발한 시퀀스 패턴 탐사와 최대 빈발 시퀀스 패턴 탐사로 나누어진다. 시공간 패턴 탐사를 위해서 데이터베이스가 주키로서 객체 식별자 Oid를 이용하고, 존속기간 max_span내의 궤적의 경로들을 선택할 수 있는 보조키로서 유효 시간 t를 이용한다.

4.1 이동객체의 연속적인 시간 표현

패턴 마이닝의 객체로서 시퀀스가 트랜잭션 데이터베이스에서 명백하게 정의되는 동안 이동 패턴의 객체로서의 시퀀스는 아니다. 따라서 이동 시퀀스들은 빈발 패턴들을 탐사하기 전에 생성되어야한다. 이동 시퀀스는 이동 시퀀스의 정의와 함께 정확히 컴파일 될 때만 유용하다. 즉, 각 시퀀스의 위치들은 시간적으로 순서화 되고, 존속기간은 최대 시간 주기 max_span 내에 존재한다. 표 2는 시간 주기 = 1 과 max_gap = [2006/01/30/ 7:00, 2006/01/30/ 10:00]로 이동 시퀀스의 위치에 대한 시간을 표현하여 별개의 객체 식별자 oid 와 함께 이동 시퀀스의 집합으로 변경된 예를 보여준다. 이동 시퀀스 생성은 MovingSequenceExtraction() 함수를 이용하며, 이 함수를 아래 알고리즘 1에 자세히 기술하였다.

〈표 2〉 이동 시퀀스의 집합 예

oid	Moving sequence
o1	<D10,D20,D30,D31,D21>
o2	<D11,D12,D13>

[알고리즘 2] 모든 빈발 시퀀스 패턴 생성 알고리즘

```
Function All_MOP()

Input:
- MS: set of moving sequences
- min_sup: minimum support
Output: Set of all frequent movement patterns
Begin
MS ← MovingSequenceExtraction(D, A);
F1a set of frequent cells;
For (k=2; Fk-1≠Ø; k++)
Ck generate candidate k-patterns and Prune and update MinInfreqList
For all moving sequences oid in MS do
Increment counter for c∈Ck contained in oid
Fk a set of c∈Ck with c.counter ≥ min_sup;
FMOP set of all frequent patterns in Fk
return FMOP
End
```

4.2 빈발 시퀀스 생성

이 논문에서 빈발 시퀀스는 두 가지 방법으로 생성한다. 첫 번째는 모든 빈발 시퀀스 생성과 최대 빈발 시퀀스 생성이다. 다음 절에서 이 두 가지 방법에 대해서 자세히 기술한다.

4.2.1 모든 빈발 시퀀스 패턴 생성

이동 객체 데이터베이스로부터 모든 빈발 시퀀스를 생성하기 위해 GSP 기반 All_MOP 알고리즘을 제안한다. 하지만 메모리에서 후보 항목들을 제거하기 위해 모든 최소 비 빈발 패턴인 MinInfreqList 리스트를 유지하여야 한다. 초기에 MinInfreqList는 모든 비 빈발 2-패턴들을 포함한다. 새로운 후보 패턴 P가 생성될 때 이 리스트에서 임의의 패턴들의 수퍼 패턴을 체크한다. 이러한 과정을 알고리즘 2에서

[알고리즘 3] 모든 최대 빈발 시퀀스 패턴 탐사

```
Function Max_MOP()
Input:
- MS: set of moving sequences
- min_sup: minimum support
Output: set of frequent maximal patterns
Begin
CandList ←a set of 2-patterns;
MaxFreqList ←a set of 2-spatterns;
While (CandList ≠∅)
Temp generate candidate sequences by joining CandList with itself and Prune and update
MinInfreqList
For all moving sequences oid in MS do
Increment counter for c∈Temp contained in oid
CandList a set of c∈CandList with c.counter ≥ min_sup;
Update MaxFreqList for all c.counter ≥ min_sup;
return MaxFreqList
End
```

자세히 기술하였다.

모든 빈발 패턴의 탐사는 검색 공간이 크면 패턴의 길이가 다소 길어지고 패턴이 변할 수도 있다. 이런 단점을 극복하기 위해 우리는 최대 빈발 시퀀스 패턴을 생성한다.

4.2.2 최대 빈발 시퀀스 패턴 생성

최대 빈발 시퀀스 패턴을 생성하기 위해서 이 데이터 구조는 메모리에서 모든 최대 빈발 패턴 MaxFreqList 리스트에 추가되고 후보 패턴 CandList 리스트에 유지된다. 후보 패턴 생성은 All_MOP에서 기술된 후보 패턴들을 생성하는 방법과 동일하다. MaxFreqList는 모든 빈발 2 패턴과 같이 초기화 된다. Max_MOP에서 생성되는 후보 패턴들은 DFS_MINE [8]와 비교하기 위해 후보의 수를 제한한다. 알고리즘 3은 최대 빈발 시퀀스 패턴을 생성하는 과정을 기술한 것으로써, 입력 조건으로 이동객체 시퀀스 집합과 최소 지지도를 가지고 최대 빈발 패턴 집합을 출력한다.

5. 실험 및 성능평가

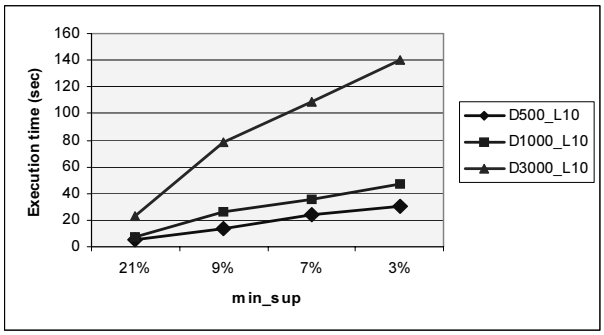
이 장에서는 시공간 패턴 마이닝을 중에서, 이동패턴 생성을 위하여 제안한 알고리즘All_MOP와 기존 알고리즘 DFS_MINE와 비교평가에 대하여 기술한다. 이러한 실험을 통하여 이 논문에서 제안한 알고리즘이 수행시간 측면에서 보다 효율적인 것을 증명한다.

DFS_MINE[8] 알고리즘은 긴 시퀀스 패턴을 매우 빠르게 발견하기 위해 Depth-First-Search-like 접근법을 사용한 알고리즘으로써, 긴 빈발 시공간 시퀀스 발견을 위해 기존의 Breadth-First-Search-like 접근법보다 성능을 향상시킨 알고리즘이다.

이러한 성능 평가를 위하여 All_MOP, Max_MOP 및 DFS_MINE를 C++로 구현하였다. 실험 환경은 Pentium IV, 256MB RAM 그리고 windows XP를 기반으로 하였다. 실험에 사용된 데이터는 이동 객체 데이터 생성기에 의해 생성된 데이터이고, 데이터 생성을 위해 사용된 파라미터는 <표 3>과 같다.

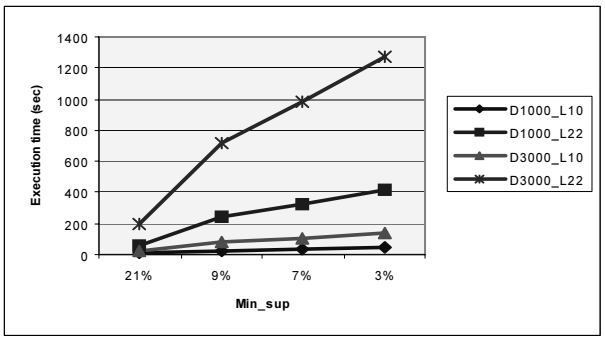
〈표 3〉 데이터 생성을 위한 파라미터

Symbol	Definition	Default values
D	#of distinct objects	
max_span	Life of trajectories	the same weekday
r	Spatial extent	
	Temporal extent	2
L	Average length of object movements	
	re-sampling rate	
R	Outlier percentage	3%
min_sup	Minimum support percentage	9%
v_{max}	maximal velocity	17m/s
grid	interested region	1000 x 1000

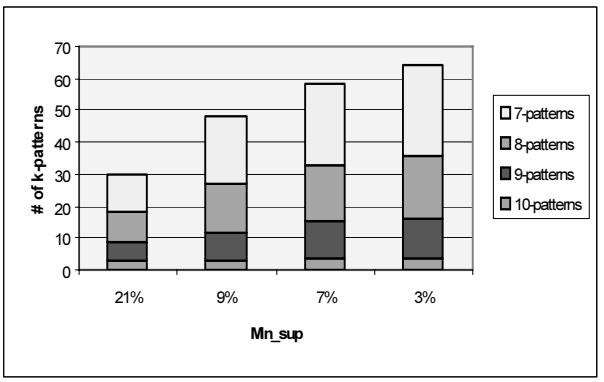


(그림 6) 최대 길이가 같고 궤적수가 다른 데이터들에 대한 All_MOP 알고리즘의 적용 결과

데이터 셋은 다음 원리에 의해 생성하였다. 평균 70%의 객체 궤적들은 평균 30% 궤적들과 같은 경로에서 최대 길이 L을 가진다. 여기서 우리는 [2006/01/01:07:00, 2006/01/01:16:00]와 같은 날 12시간으로 존속 기간 max_span을 고정 하였고, 각 궤적의 시간 포인트들은 L-1 ~ L+10까지 변경하였다. 데이터 셋에서 D500_ L10은 500개의 궤적과 최대 길이 10의 평균 크기를 의미한다.



(그림 7) 각각 1000, 3000개의 궤적수에 대한 최대 길이 변경 (10 및 22) 에 대한 All_MOP 알고리즘 적용 결과



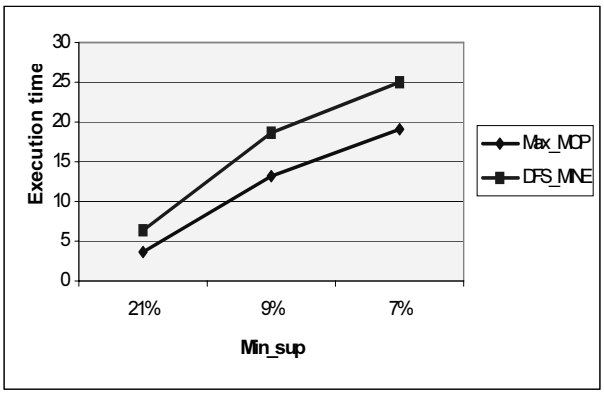
(그림 8) D500_L10의 All_MOP

(그림 6)은 최대 길이를 10으로 고정시키고, 각각의 궤적수 500, 1000, 3000에 대하여 All_MOP 알고리즘을 적용하여 최소지지도 값 (각각 3%, 7%, 9%, 21%) 변화에 따른 실행시간을 나타낸 결과이다. (그림 6)서 보듯이 궤적수에 따라서 실행시간이 증가하고, 최소지지도 값의 증가에 따라 실행시간이 낮아지는 것을 볼 수 있다.

(그림 7)은 궤적수 및 최대 길이 변화에 따른 실행시간을 나타낸 것으로써, All_MOP와 Max_MOP의 실행시간은 궤적 길이의 변화에 따라 서서히 증가하는 것을 나타낸다. (그림 6)과 (그림 7)에서 보듯이 최소지지도의 변경되는 값이 감소함에 따라 빈발 패턴의 수가 증가하는 것을 알수 있다.

(그림 8)은 궤적수 500과 최대길이 10에 대하여 최소지지도 변화에 따른 All_MOP 알고리즘을 적용한 후 각 k개의 빈발패턴 수를 나타낸 것으로써, 패턴의 길이가 패턴의 길이가 짧아질수록 더 많은 패턴들이 생성되는 것을 알 수 있다.

(그림 9)는 이전에 기술한 DFS_MINE 알고리즘과 이 논문에서 제안한 Max_MOP 알고리즘에 대하여 최소지지도 변화에 따른 실행시간 수행 결과를 비교한 것이다. 그림에서 보듯이 이 논문에서 제안한 Max_MOP 알고리즘이 DFS_MINE보다 실행시간이 보다 적게 소비되는 것을 알 수 있다. 최소지지도 21%의 경우 Max_MOP의 경우 약 4 seconds인 반면에 DFS_MINE의 경우 약 6 seconds의 결과를 나타내었다. 이러한 결과는 제안된 All_MOP 알고리즘을 통해서 생성된 후보 패턴들의 수가 DFS_MINE에서 생성된 후보 패턴들의 수보다 훨씬 적기 때문이다.



(그림 9) D500_L10 과 Max_MOP 비교

6. 결 론

최근에 이르러 모바일 환경에서 위치기반 서비스나 개인화 추천 서비스 등이 제공됨에 따라서, 시간과 공간을 모두 고려하여 정보를 얻는 것이 중요시 되고 있다. 이 논문에서는 모바일 환경에서 시간과 공간을 모두 고려한 이동 객체의 시공간 시퀀스 패턴을 탐사하기 위해 All_MOP와 Max_MOP의 두 알고리즘을 제안하였다. 또한 시공간 시퀀스 패턴을 탐사하기 전에 이동객체 데이터와 이동객체 궤적에 대한 모델링과 일반화 과정을 통해 시퀀스 패턴 탐사를 위한 데이터 전처리과정을 수행하였다. 시공간 시퀀스 패턴을 탐사하기 위해 All_MOP알고리즘을 기반으로 전처리 과정이 수행된 데이터로부터 모든 빈발한 패턴들을 탐색하고, Max_MOP 알고리즘을 이용하여 탐사된 빈발 패턴 중 최대 빈발 패턴만을 탐색하였다. 아울러, All_MOP 알고리즘과 DFS_MINE 알고리즘의 비교 실험을 통하여, 빈발 패턴 탐사를 위한 실행 시간에서 All_MOP 알고리즘이 좀 더 효율적인 결과를 나타내었다. 제안된 알고리즘 기법은 관광 서비스, 교통 서비스 등과 같은 위치 기반 서비스 등에서 보다 빠른 빈발 패턴 탐사에 활용될 수 있다.

참 고 문 헌

[1] R.Agrawal & R.Srikant, “Mining Sequential Patterns. Research report,” 1995.

[2] S.Chakrabarti, S.Sarawagi, & B.Dom, “Mining Surprising Patterns using Temporal Description Length,” In Proc. of 24th VLDB, pp.606-617, 1998.

[3] Young Jin Jung and Keun Ho Ryu. Design and Implementation of a SQL based Moving Object Query Process System for Controling Transportation Vehicle,

The KIPS Transactions : Part D, Vol.12-D, No.5, pp.699-708, 2005.
[4] N.Meratnia & R.A.D. By, "Aggregation and Comparison of Trajectories," In proc. of 10th ACM Int'l.Symp. on GIS, pp.49-54, 2002.
[5] H.Yun, D.Ha, B.Hwang, K.H.Ryu, "Mining association rules on significant rare data using relative support," Journal of System and Software, Vol.67, Issue.3, pp. 181-191, 2003.
[6] J.F.Roddick & M.Spiliopoulou, "A Survey of Temporal Knowledge Discovery Paradigms and Methods," IEEE Trans. on Knowledge and Data Engineering, Vol.14, Issue.4, pp.750-767, 2002.
[7] R.Srikant & R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements," in Proc. Int'l Conf. EDBT, 1996.
[8] I.Tsoukatos & D.Gunopulos, "Efficient Mining of Spatiotemporal Patterns," In Poc. On SSTD, LNCS, 2001.
[9] Jeong Seok Park, Young Jin Jung, Moon Sun Shin, and Keun Ho Ryu, A Moving Object Query Process System for Mobile Recommendation Service, The KIPS Transactions : Part D, Vol.14-D, No.7, pp.707-718, 2007.
[10] J.W.Lee, O.H.Paek & K.H Ryu, "Temporal Moving Pattern for Location-Based Service," Journal of system and software, 2004.
[11] J.W. Lee, Y.J.Lee, H.K.Kim, B.H.Hwang & K.H.Ryu, "Discovering Temporal Relation Rules from Interval Data," In Proc. of EurAsia-ICT, LNCS, pp.57-66, 2002.
[12] M.Zaki, "SPADE: An efficient Algorithm for Mining Frequent Sequences," Machine learning Journal, Vol. 42, Issue.1-2, pp.31-60, 2001.
[13] J.Hipp, U.Guntzer, G.Nakhaeizadeh, "Algorithms for association rule mining- A general survey and comparison," ACM SIGMOD, Vol.2, Issue.1, pp.58-64, 2000.
[14] R. Agrawal, T. Imilienski, and A., "Swami, Mining Association Rules between Sets of Items in Large Databases," In Proceeding of ACM SIGMOD, pp.207-216, 1993.
[15] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, "New Algorithms for Fast Discovery of Association Rules," In Proceeding of KDD, pp.283-286, 1997.
[16] J. Han, J. Pei and Y . Yin, "Mining frequent patterns without candidate generation," In Proceeding of ACM SIGMOD, pp.1-12, 2000.

Vu, Thi Hong Nhan

e-mail : nhanvth@dblab.chungbuk.ac.kr
2001년 Vietnam National University, College of Technology(학사)
2004년 충북대학교 전자계산학과(석사)
2007년 충북대학교 전자계산학과(박사)
현 재 한국전자통신연구원, RFID/USN 미들웨어 연구팀 근무
관심분야 : GIS, Moving Object Databases, Stream Database, Data Mining, Sensor network, Ubiquitous&Pervasive computing

이 범 주

e-mail : bjlee@dblab.chungbuk.ac.kr
2000년 서원대학교 전자계산학과 (공학사)
2003년 충북대학교 전자계산학과 (이학석사)
2006년 충북대학교 전자계산학과 (박사수료)
관심분야 : 데이터 마이닝, 시공간데이터베이스, 생물정보학, 엔자임 기능 예측, 스트림데이터마이닝

류 근 호

e-mail : khryu@dblab.chungbuk.ac.kr
1976년 숭실대학교 전산학과(이학사)
1980년 연세대학교 공학대학원 전산전공 (공학석사)
1988년 연세대학교 대학원 전산전공 (공학박사)
1976년~1986년 육군군수 지원사 전산실(ROTC 장교), 한국전자통신연구원(연구원), 한국방송통신대학교 전산학과(조교수)
1989년~1991년 Univ. of Arizona Research Staff (TempIS 연구원, Temporal DB)
1986년~현 재 충북대학교 전기전자 및 컴퓨터공학부 교수
관심분야 : 시간 데이터베이스, 시공간 데이터베이스, Temporal GIS, 지식기반 정보검색 시스템, 유비쿼터스컴퓨팅 및 스트림데이터처리, 데이터마이닝, 데이터베이스 보안, 바이오인포메틱스