

위치 기반 서비스를 위한 이동 객체의 시간 패턴 탐사 기법

(Temporal Pattern Mining of Moving Objects for Location based Services)

이 준 욱 [†] 백 옥 현 ^{**} 류 근 호 ^{***}
(Jun Wook Lee) (Ok Hyun Paek) (Keun Ho Ryu)

요 약 위치 기반 서비스는 이동중인 사용자에게 위치와 관련된 정보를 제공한다. 최소한의 자원으로 사용자에게 유용한 정보를 개인화하여 제공하는 것은 위치 기반 서비스가 가져야 할 필수적인 기능이다. 이 기능은 데이터 마이닝을 통해 실현될 수 있다. 하지만 기존의 데이터 마이닝 연구는 시간 및 공간 속성을 동시에 고려하고 있지 않다. 따라서 시간에 따라 공간 위치 속성이 변경되는 특성을 갖는 위치 기반 서비스의 대상에는 적절하지 않다. 이 논문에서는 시간 및 공간 속성을 가지는 이동 객체의 위치 데이터로부터 유용한 시간 패턴을 탐사하기 위한 새로운 데이터 마이닝 기법을 제안하였다. 평면 상에서 좌표로 표현되는 이동 객체의 위치 정보를 일반화하기 위하여 contains와 같은 공간 연산을 사용하였다. 또한 이동 패턴 탐사 시 실제 유효한 시퀀스를 만들기 위해 객체의 위치 사이에 시간 제약조건을 적용하였다. 이렇게 생성된 이동 객체 위치의 시퀀스로부터 빈발 이동 시퀀스를 구하여 시간 패턴을 생성하였다. 제안한 기법은 기존과는 다른 시,공간적 접근을 취함으로써 시간과 공간 의미가 중요시되는 위치 기반 서비스에 적합한 새로운 유형의 지식을 제공할 수 있다.

키워드 : 시간 패턴탐사, 이동객체, 위치기반 서비스, 이동 패턴

Abstract LBS(Location Based Services) provide the location-based information to its mobile users. The primary functionality of these services is to provide useful information to its users at a minimum cost of resources. The functionality can be implemented through data mining techniques. However, conventional data mining researches have not been considered spatial and temporal aspects of data simultaneously. Therefore, these techniques are inappropriate to apply on the objects of LBS, which change spatial attributes over time. In this paper, we propose a new data mining technique for identifying the temporal patterns from the series of the locations of moving objects that have both temporal and spatial dimension. We use a spatial operation of contains to generalize the location of moving point and apply time constraints between the locations of a moving object to make a valid moving sequence. Finally, the spatio-temporal technique proposed in this paper is very practical approach in not only providing more useful knowledge to LBS, but also improving the quality of the services.

Key words : Temporal Pattern Mining, Moving Object, Location based service, Moving Pattern

1. 서론

이 연구는 한국과학재단 RRC(정주대정보통신연구센터)의 지원으로 수행되었음.

[†] 비 회 원 : 충북대학교 컴퓨터과학과
junux@dblab.chungbuk.ac.kr

^{**} 비 회 원 : 국방과학연구소 연구원
ohpack@add.re.kr

^{***} 종신회원 : 충북대학교 전기전자및컴퓨터공학부 교수
khryu@dblab.chungbuk.ac.kr

논문접수 : 2002년 1월 29일

심사완료 : 2002년 8월 22일

위치 기반 서비스는 개인의 위치를 파악하여 각종 마케팅이나 고객 서비스등에 있어 개인화 되고 동적인 서비스를 제공한다[1]. 위치 기반 서비스의 대상인 PDA, 이동 전화, 차량, 비행기 등은 자유롭게 이동하면서 시간에 따라 위치가 변경되는 특성을 가진다. 이와 같이 시간의 흐름에 따라 객체가 이동하면서 위치 및 모양이 연속적으로 변경되는 특성을 가지는 데이터를 이동 객체(moving objects)라 한다[2,3]. 이동객체는 이동 점 객체와 이동 영역 객체로 구분되며 연속적으로 위치가 변

화하는 객체는 이동 점 객체로써 모델링된다. 이동 객체의 시간에 따른 변화는 특정한 경향 혹은 패턴을 가진다. 이러한 경향이나 패턴과 같은 지식은 시간 데이터 마이닝 기법을 통해 발견할 수 있다[4,5]. 데이터 마이닝을 통해 발견된 이동 객체의 패턴은 사용자의 이동 경향을 고려한 위치 기반 정보 서비스에 유용하다[4,6].

기존 시간 데이터 마이닝 연구는 이동 객체의 위치 데이터는 거의 고려하고 있지 않다. 예를 들어 이동 객체의 위치 데이터로부터 “고객의 25%는 A→B→C→D 지역 순으로 방문한다”, “20대 여성 고객의 60%는 B→D→E의 이동 패턴을 가진다”, “오후 1시부터 3시 사이에 25%의 고객이 쇼핑 구역에서 식당 구역으로 이동한다”와 같은 유형의 지식을 찾아낼 수 있다. 이와 같이 이동 점 객체에 연관규칙, 순차 패턴, 분류 등 기존의 다양한 데이터 마이닝 기법을 적용함으로써 유용한 지식을 얻을 수 있음을 볼 수 있다[7]. 이러한 지식의 예는 시간 패턴, 구체적으로는 순차 패턴이라는 공통된 지식을 담고 있다. 다시 말하면, 공간 정보가 연속적으로 변하는 이동 객체는 이러한 변화의 패턴을 탐사하는 것이 가장 유용하면서 기본적인 과정이라는 것이다.

이동 객체의 위치가 시간에 따라 변하는 패턴을 탐사하는 문제는 트랜잭션 데이터베이스 분석[8,9]이나 웹상에서의 사용자의 행위 패턴을 분석하기 위한 문제[10, 11,12]와 유사하다. 그러나 이러한 기존의 제시된 기법들은 사용자의 공간적인 속성의 변화에 관하여 고려하지 않고 있어 이러한 접근 방법은 이동 점 객체에 대한 시간 패턴을 탐사하는 문제에는 적합하지 않다. 위치기반 서비스를 위한 이동객체의 시간패턴 탐사와 같이 시간과 공간 속성을 함께 고려하는 문제에 적합한 새로운 접근 방법이 필요하다. 따라서, 이 논문에서는 시간 및 공간 속성을 가진 이동 객체의 시간 패턴을 탐사하기 위한 새로운 마이닝 기법을 제안한다.

이 논문에서는 이동 객체에 대한 정의와 시간에 따른 위치 변화의 경향으로서 이동 패턴(*moving patterns*)을 정의한다. 또한 위치 기반 서비스에 적용 가능한 지식을 추출하는 새로운 마이닝 기법으로 이동 패턴 탐사 기법을 제안한다. 이 기법에서 제안한 아이디어는 다음과 같다.

■ 2차원 평면 상에서 좌표로 표현되는 이동 객체에 대하여 공간 연산을 적용함으로써 위치 정보를 일반화한다. 공간 연산을 사용한 위치 정보의 일반화를 통해 이동 객체의 위치 정보를 의미 있는 지식을 찾을 수 있도록 변환할 수 있다.

■ 이동 패턴 탐사에 있어 불확실한 이동 시퀀스를 유효한 트랜잭션으로 구성하기 위해 이동 객체의 위치 사

이에 시간 제약조건을 적용한다. 이것은 이동 패턴 탐사의 대상이 되는 시퀀스가 명확하게 정의되지 않는다는 것에 기인한다. 시퀀스를 구성하는 지역 간에 최대 시간 제약조건을 두어서 두 위치 사이의 시간이 최대 시간 제약조건 내에 있을 때만 하나의 시퀀스로 생성하도록 한다.

■ 이동 객체 위치에 대한 이동 시퀀스로부터의 의미 있는 패턴을 탐사하는 알고리즘을 제안한다. 이 알고리즘은 Apriori[13]를 이동 패턴 탐사 문제에 적합하게 확장하였다. 시퀀스를 구성하는 요소가 이동 객체의 위치이므로 Apriori에서 사용하는 항목(item) 대신에 위치(location)라는 용어를 사용한다. 또한 Apriori에서 후보 항목을 줄이기 위해 이전 단계의 지식을 사용했던 것과 같이 가지치기 작업을 수행하며, 최소 임계치 이상의 빈발 이동 시퀀스를 생성한다.

이동 객체의 시간 패턴은 시,공간적 접근에 기반함으로써 차량, 항공기, 개인단말기 사용자, 모바일 디바이스 사용자 등에게 공간상의 이동 패턴에서의 위치 의미와 이동 시간에 따른 시간의 의미를 함께 포함할 수 있다. 따라서 제안하는 이동 패턴은 개인화되고 차별화된 위치 기반 정보서비스 및 M-Commerce 등에 매우 실질적으로 적용될 수 있다.

이 논문의 구성은 다음과 같다. 2장에서는 이동 객체 시간 패턴 탐사와 관련하여 기존의 패턴 탐사 연구를 기술하고, 3장에서는 이동 패턴 탐사의 대상인 이동 객체의 표현을 정의한다. 4장에서는 이동 패턴 탐사 문제를 정의하고 5장에서는 문제 정의에 따라 이동 패턴 탐사기법을 제안한다. 6장에서는 제안하는 기법을 구현하고 실험을 통해 그 성능을 평가하여 보인다. 마지막으로 결론에서 향후 연구를 기술한다.

2. 관련연구

위치 기반 서비스는 이동중인 사용자에게 유무선 통신을 이용하여 위치 컨텍스트에 따른 관련된 정보를 제공하는 서비스를 통칭한다. 위치 기반 서비스는 위치 측정 기술, 위치 응용 플랫폼 그리고 응용 서비스 등으로 구성된다[1]. 위치 응용 플랫폼은 위치 기반 서비스를 위한 기반으로써 위치 측정기술에 의해 수집된 사용자 위치정보 관리 및 관제 등을 수행한다. 이러한 플랫폼을 기반으로 하는 위치 기반 응용 서비스는 다양한 서비스 [1]를 가능케 하지만 사용자를 고려하여 사용자의 관심과 위치 그리고 최소한의 자원을 이용한 효과적인 정보 전달이 무엇보다도 중요하다.

위치 기반 응용 서비스를 위해서는 사용자 프로필과

위치정보, 행위정보, 구매정보 등 다양한 정보들이 통합되어 이용되기 때문에 데이터웨어 하우스가 효과적으로 이용될 수 있다. 특히 동적으로 변하는 사용자의 위치정보와 사용자의 관심 및 행위 등을 분석하여 동적인 응용서비스를 가능케 하기 위해서는 데이터 마이닝 기술의 접목이 필요하다. 이러한 데이터마이닝 기술로는 연관규칙, 분류, 클러스터링, 특성화 기법 등을 들 수 있다. 하지만 위치기반 서비스의 대상이 되는 사용자는 시간에 따라 위치 속성이 변화되는 특성을 갖고 있기 때문에 시간속성을 중요하게 고려하는 시간 데이터 마이닝 기법 등의 적용이 요구된다.

기존의 데이터 마이닝 연구 중에서 위치기반 서비스를 위해 유용한 지식 유형으로서 시간패턴을 탐사하기 위한 기법들이 많이 제시되었다. 시간 패턴은 시간 속성을 가진 데이터에서 찾아 낼 수 있는 시간 지식의 유형으로, 이러한 시간 데이터로부터 의미 있는 지식을 탐사하기 위한 여러 연구가 진행되었다. 기존에 연구되었던 시간 패턴은 그 형태에 따라 순차 패턴[2,8,9,10,4,32], 주기 패턴[15,16], 시간 관계[5,15]로 분류될 수 있다.

우선 순차패턴 탐사는 항목집합으로 구성된 트랜잭션들 간에 특정 항목집합이 순차적으로 발생하는 패턴을 탐사하는 기법이다[8]. 순차패턴의 예로, 특정 지역을 방문하는 사용자들은 특정 시간 간격동안 $A \rightarrow B \rightarrow C$ 구역을 순차적으로 방문하는 경향이 있다는 것을 들 수 있다. 순차패턴 탐사는 제시된 기법은 한 트랜잭션 내에서 발생하는 항목들간에 연관성을 탐사하는 연관규칙에 시간 개념을 추가한 것이다[8]. 즉, 연관규칙 탐사는 트랜잭션 내의 연관성을 탐사하는 문제인 반면, 순차패턴은 트랜잭션 상호간 연관성을 탐사하는 문제이다.

순차 패턴의 소개 이후 더 효율적인 기법에 관한 연구들과 다른 시간 관련 빈발 패턴을 마이닝 하기 위한 확장된 연구들이 진행되었다. 이들 중에서 GSP[9]는 시간 제약조건, 시간 윈도우, 항목에 대한 분류를 이용하여 순차패턴에서 찾지 못하는 패턴을 찾고자 하였으며, 또한 트랜잭션 데이터 뿐만 아니라 웹 상에서의 사용자의 접근 패턴을 탐사하기 위해 확장되었다[10,11,12]. [10]에서는 문서나 객체가 하이퍼링크로 연결된 분산된 정보를 제공하는 환경에서의 접근 패턴을 탐사하기 위한 기법을 제공하였다. 사용자의 접근 패턴을 선회 패턴(*traversal pattern*)이라 정의하고 선회 패턴을 탐사하기 위한 알고리즘적인 접근을 시도하였다. [12]에서는 접근 패턴을 효율적으로 탐사하기 위한 자료구조로써 WAP를 제안하였다.

주기 패턴은 시간 상에서 정기적으로 발생하는 현상

으로, 예를 들어 매년 또는 매주, 매일과 같은 시기에 주기적으로 발생하는 패턴을 가리킨다. [16]에서는 주기(cyclic) 연관규칙을 소개하였다. 초점을 맞추고 있는 부분은 전체 시간에서의 연관 규칙이 아니라, 특정한 시간 간격에서만 존재하고 나머지 부분에서는 존재하지 않는 규칙을 탐사하는 것이다. 즉 주기 연관규칙을 탐사함으로써 “매일 7시부터 9시 사이에 커피를 산 고객은 도넛도 산다”와 같은 지식을 얻을 수 있다. 이 연구에서는 주기 연관 규칙을 탐사하기 위해서 기존의 기법을 확장하여 연관 규칙 탐사와 주기를 따로 다루었다. [18]에서는 시간 연관 규칙을 위한 문제를 제시하고 주기적으로 발생하는 사건에 대한 지식을 표현하는 방법을 제안하였다. 시간 패턴을 찾아내기 위해 시스템의 달력(calendar)을 이용하여 시간 속성에 접근하였고 각 시간 단위(시간, 일, 주, 월 및 연)와 마이닝 규칙을 적용할 시간 간격 제약 조건을 기존의 연관규칙에 추가한 형태를 가진다. [15]에서는 시계열 데이터에서 부분적으로 주기적인 순차패턴을 탐사하는 기법을 제안되었다.

마지막으로 순차 패턴과 주기 패턴이 트랜잭션의 발생 시점에 기반을 둔 것과는 달리, 시간 관계는 시간 간격을 가진 여러 사건 간의 시간 관계를 포함한 지식이다[5,15]. 간격을 가지는 사건의 시간 패턴을 탐사하기 위한 기법은 다양한 사건 간의 관계를 구하는 것을 목적으로 하고 있다[17]. 입력은 사건의 시작 시점과 종료 시점을 저장하고 있는 시간 데이터베이스로 가정했으며, 사건 사이의 관계를 나타내기 위해서 Allen의 시간 관계 분류[19]에 기반하였다. 이 기법에서는 두 이벤트가 충분히 가까운 시간 내에 발생하지 않으면 연관성을 찾기가 힘들다고 보고, 특정 시간 윈도우 내에서 발생한 시간 패턴에 대해 관심을 두었다. [5] 역시 Allen의 시간 간격 연산자를 기반으로 한 시간 간격 데이터로부터 시간 관계 규칙을 탐사하기 위한 기법을 제안하였다. [17]과 다른 점은 시간 데이터베이스를 가정하는 것이 아니라 발생 시점을 가진 데이터를 시간 간격 데이터로 변환하여 사용하였다는 것이다.

이러한 시간 패턴 탐사와 관련된 연구들은 시간 데이터를 대상으로 수행되었으나, 지식 탐사의 대상이 되는 데이터는 공통적으로 공간 속성을 가지고 있지 않다. 따라서 시간과 공간 속성을 함께 가지는 이동 객체의 시간 패턴 탐사에 기존의 연구들을 적용한다면 공간 속성 즉 위치와 관계성이 고려된 의미 있는 시간 패턴을 탐사하는데 많은 제약이 따른다는 문제점이 있다. 시간 및 공간 속성을 가지는 데이터에 대한 패턴 탐사를 위한 연구가 일부 수행되었다[20,21]. 그러나 [20]은 공간 데

이타에 대한 지식이 시간에 따라 어떻게 변화하는가를 탐사한다. 즉, 메타규칙(meta-rule)은 규칙 속에서의 또 다른 규칙을 탐사하는 것으로 시간과 공간을 동시에 다루고 있지는 않으며, [21]은 특정 지역 내에서 일어난 환경적인 사건(event)의 패턴을 탐사하는 것으로, 각 지역은 일정한 크기를 가지는 것으로 한정하고 있어 실세계의 공간 영역 상에서의 공간 객체간의 관계 및 의미를 반영하기 어렵다는 문제점이 있다.

이처럼 이동객체의 연속적 위치변화를 효율적으로 패턴화 하여 의미 있는 시간 지식으로 탐사하는 것은 공간과 시간속성을 함께 고려해야 하는 문제점을 갖고 있다. 이러한 패턴 탐사의 문제점을 해결하기 위해 공간 연산의 적용을 통한 공간 영역 일반화 접근법을 적용할 수 있다. 이 논문에서는 이동 객체 위치의 이산적인 모델에 기반하여 이동 객체의 각 위치에 대해서 공간 영역으로 일반화하는 공간 연산을 적용함으로써 실세계의 공간의 의미를 반영하여 이동 객체의 시간 패턴을 탐사할 수 있도록 하였다.

3. 이동 객체의 표현

정적인 객체와 달리 이동 객체의 위치는 시간에 따라 연속적으로 변하는 특성을 갖기 때문에 특정 시점의 정적인 데이터만을 고려하는 기존의 데이터 모델로는 직접 표현할 수 없었다[22,23]. 따라서, 시공간 이동 객체를 표현하기 위한 다양한 데이터 모델이 제시되었다[24, 25,26]. 이동 객체의 위치 변화는 이산적(discrete) 또는 연속적(continuous)으로 발생할 수 있으며, 시점(time points) 또는 시간 구간(time periods)으로 기록될 수 있다. 이러한 이동 객체의 표현상의 방법은 어떤 응용에서 사용되는가에 따라서 다르게 정의될 수 있다[6,27]. 각각의 표현 방법에 따라 장점과 단점이 존재할 수 있으나, 이동 객체의 표현 모델에 대해서 일관성을 가지는 표준이 없고, 이동 객체의 시간 패턴을 탐사하는 데 있어서 고려해야 할 점이 한정되어 있으므로 이동 패턴 탐사 문제를 기술하기에 앞서 이동 객체의 표현을 정의해야 할 필요가 있다.

이동 점 객체의 위치 속성은 (그림 1)과 같이 x, y를 축으로 하는 2차원 공간상에서 연속적으로 변화한다. 각 차원 x, y는 경계를 갖는 유한한 영역으로써 가정한다.

실세계의 이동 객체의 연속적 변화는 컴퓨터 내에서 표현되기 힘들기 때문에 이 논문에서는 (그림 1)에서 연속적으로 이동하는 객체의 이동 위치는 이산적인 시점에서 샘플링 됨을 가정한다. 이 때 시점은 시간 구간의 시작 점과 끝 점을 나타낸다. 이 논문에서 고려하는 이동

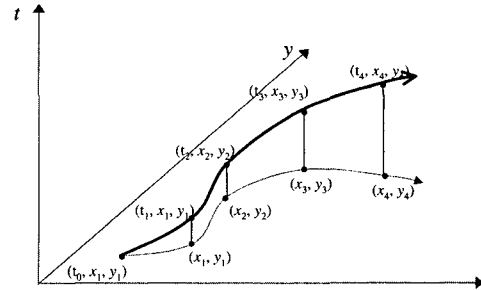


그림 1 이동 객체의 위치 변화

객체의 표현은 일반성을 갖기 위해 기본적인 부분만을 고려한다. 즉, 샘플링된 특정 시점의 위치로 표현하는 것이다. 예를 들어 위치 기반 서비스에서의 공간 속성은 어떤 위치 측정 기술을 사용하느냐에 따라서 x, y 좌표를 가지는 좌표로 표현할 수도 있고, 기지국에 포함되는 영역만으로도 만족하는 경우도 있다. 현재 이동 전화 등을 대상으로 하는 실 세계에서는 기지국이 포함하는 일정 셀 영역만으로 표현하지만, 알고리즘에서 지역만을 대상으로 해서 데이터를 모델링 할 경우, 좌표 값으로 표현되는 이동 점에 대해서는 규칙을 탐사할 수 없으므로 이동 점 객체의 공간 속성은 이동 객체의 평면상의 x, y 좌표로써 표현한다. 이동 점 객체에 대한 추상화된 타입으로써의 Mpoint는 다음과 같이 정의된다.

[정의 3.1]

$$Mpoint = oid, \{(VT_1, L_1), (VT_2, L_2), \dots, (VT_n, L_n)\}$$

단, oid는 유일한 특성을 갖는 객체의 식별자를 가리킨다. VT_i는 유효 시간이고, L_i는 VT_i에 샘플링된 객체의 위치를 나타내며 (x_i, y_i)로 이루어진다.

<표 1>은 이러한 이동 객체를 관계 데이터베이스에 테이블 형태로 표현한 예를 보여준다.

표 1 이동 객체 표현의 예

oid	vt	x	y
100	2001/10/10/13/10	3321000	-233100
100	2001/10/10/13/20	3397000	-463600
100	2001/10/10/13/25	3385000	-523600

4. 이동 패턴 탐사 문제 정의

$L = \{l_1, l_2, \dots, l_m\}$ 은 이동 객체의 공간 위치 속성을 나타내는 좌표의 유한 집합이다. 단, $l_i = (x_i, y_i)$ 이며 각 x_i, y_i 는 평면상에서의 이동 객체의 좌표를 나타낸다.

$A = \{a_1, a_2, \dots, a_n\}$ 은 이동 객체의 공간 위치 속성 값으로 표현되는 영역의 집합이다. 단, 각 $1 \leq j \leq n$ 에 대해 $a_j = (l_1, l_2, \dots, l_k)$ 이고 $l_k = (x_k, y_k)$ 이다. 이것은 하나의 영역을 표현하기 위해 영역이 가지는 대표 좌표 값들을 사용하는 것으로써, 좌표 값으로 표현된 이동 객체의 공간 위치 속성은 공간 연산을 통해 영역으로 변환될 수 있다. 이와 같이, x, y 좌표 값으로 표현된 이동 객체의 공간 정보를 좌표를 포함하는 영역으로 나타내는 것을 위치 일반화라 한다. 예를 들어, (그림 2)와 같은 공간 영역이 주어졌다고 가정하자. $A = \{a_1, a_2, a_3, a_4\}$ 이며, 각 영역의 표현은 $a_1 = ((1,2), (1,4), (4,7), (8,6), (8,3), (5,2))$, $a_2 = ((4,7), (2,10), (6,12), (13,9), (8,6))$, $a_3 = ((8,3), (8,6), (13,7), (15,5), (12,2))$, $a_4 = ((5,1), (5,2), (8,3), (12,2), (12,1))$ 이다. 이 때, 이동 객체 $Mpoint_i$ 의 좌표가 (6,6)이라면, 공간 연산을 통해 a_1 으로 일반화된다. 공간 연산을 이용한 위치 일반화는 5.2절에서 자세히 기술한다.

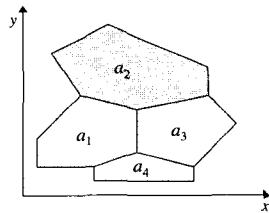


그림 2 공간 영역

하나의 시퀀스 $S = \{s_1, s_2, \dots, s_k\}$ 은 영역의 순서화 된 리스트이다. k 는 시퀀스의 길이를 나타낸다. $s_j = (t_j, a_j)$ 이며, t_j 는 이동 객체가 샘플링 된 특정 시간이고 $a_j \in A$ 이다. 시퀀스를 형성하기 위해서는 시간 제약조건을 적용한다. 시간 제약조건은 시퀀스내에 연속적인 이동으로 영역이 포함되기 위해 인접한 이동이 발생 시간 t_j, t_{j-1} 즉, 트랜잭션 시간간의 시간 간격을 의미한다. 최대 시간 간격으로서의 max_gap 는 다음과 같이 정의한다. $t_j - t_{j-1} \leq max_gap$ 이고, 이 때, $2 \leq j \leq k$

k 개의 길이를 가진 시퀀스, 즉 k 개의 지역으로 이루어진 시퀀스는 k -시퀀스로 표기된다. 하나의 지역은 한 시퀀스에서 여러 번 나타날 수 있다. 한 이동 객체에 대해 시간에 따른 지역 이동의 변화를 순서적으로 나열한 것을 이동 시퀀스(moving sequence)라 부른다.

a_k, b_k 등과 같이 이동 지역을 나타내는 식별자들로 구성된 두 이동 시퀀스에 대하여 시퀀스 $\langle a_1 a_2 \dots a_n \rangle$ 은

만약 $a_1 = b_{i_1}, a_2 = b_{i_2}, \dots, a_n = b_{i_n}$ 인 정수 $i_1 < i_2 < \dots < i_n$ 이 존재한다면 다른 시퀀스 $\langle b_1 b_2 \dots b_m \rangle$ 의 부분 시퀀스이다. 예를 들면 $\langle 2 5 9 8 \rangle$ 은 $\langle 6 2 1 5 9 3 8 \rangle$ 의 부분 시퀀스이다.

이동 시퀀스의 집합 $S = \{s_1, s_2, \dots, s_m\}$ 이라 하자. 각 s_i 는 이동 시퀀스를 나타내며, 이 때 $1 \leq i \leq m$ 이다. 한 시퀀스 s 가 다른 시퀀스 s' 의 부분 시퀀스이면 s' 는 s 를 포함한다고 한다. 시퀀스 s 의 지지도(support)는 s 를 포함하는 전체 이동 시퀀스의 비율 즉, $sup(s) = |\{s_i | s \subseteq s_i\}| / m$ 으로 정의된다. 최소지지도(minimum support threshold)는 시퀀스 s 가 빈발한 것으로 평가되기 위해 만족해야 하는 지지도의 하한 값으로 사용자가 명시하며 이 논문에서는 이 값을 min_sup 라고 명시한다. 따라서 시퀀스 s 는 $sup(s) \geq min_sup$ 이면 빈발 시퀀스이다. 시퀀스에서 하나의 지역이 여러 번 나타날 수 있지만, 하나의 시퀀스 안에서는 최대 한번만 카운트 된다.

[정의 4.1] 이동 객체의 데이터베이스 D , 사용자가 지정한 최소 지지도 min_sup 그리고 사용자가 지정한 지역 간의 시간 제약조건 max_gap 이 주어졌을 때, 이동 패턴을 마이닝 하는 문제는 최소 지지도를 만족하는 모든 빈발 이동 시퀀스를 찾는 것이다. 그러한 각 이동 시퀀스는 이동 패턴이다.

5. 이동 객체의 시간 패턴 탐사

이 절에서는 4장의 정의에 따라 이동 패턴을 탐사하기 위한 기법을 제시한다. 이동 패턴을 탐사하기 위한 알고리즘은 크게 데이터베이스 정렬 단계, 위치 일반화 단계, 이동 시퀀스 추출 단계, 빈발 이동 패턴 마이닝의 네 단계로 이루어진다.

5.1 데이터베이스 정렬 단계

마이닝을 위한 데이터베이스는 객체 식별자를 주 키로, 유효 시간을 보조키로 정렬되어야 한다. 이 과정은 이동 패턴 탐사를 위해 객체 별로 정렬하고 유효 시간에 따라 정렬하는 변환 과정이라 할 수 있다. <표 2>는 이동 객체의 식별자와 유효 시간으로 정렬된 데이터베이스를 보여준다. 각 이동 객체의 유효 시간은 분 단위로 표현한다.

5.2 위치 일반화 단계

이동 객체의 위치 값에 대해 공간 연산을 적용하여 일정한 범위 값을 가지는 영역으로 변환한다. 이 과정에서 객체의 x, y 좌표 점에 대해 특정 영역에 속해있는지 시험하는 공간 연산이 쓰이게 된다. 공간 영역은 다각형(polygon)으로 표현한다. 단, 여기서의 다각형은 가장

표 2 정렬된 데이터베이스

oid	vt	x	y
1	2001/10/30/13/10	15	10
	2001/10/30/13/15	38	15
	2001/10/30/13/25	55	8
	2001/10/30/13/38	65	19
2	2001/11/01/12/30	5	17
	2001/11/01/12/38	7	35
	2001/11/01/12/45	35	16
	2001/11/01/12/56	51	18
3	2001/10/30/14/11	23	15
	2001/10/30/14/17	59	19
	2001/10/30/14/23	77	12
	2001/10/30/14/58	78	35

단순한 형태로서, 구멍이 있거나 여러 개의 다각형으로 구성된 것은 고려하지 않는다. 데이터베이스에 저장되어 있는 모든 영역들에 대해서 x, y 좌표 점을 포함하는지에 대한 테스트가 수행되고, 각각의 영역에 테스트를 수행했을 때, 참 값을 가지는 영역을 반환하게 된다.

연산의 효율성을 위해 각각의 영역에 대해서 공간 인덱싱을 필요로 한다. 즉 영역에 대한 포함 연산의 효율적인 수행을 위해서는 공간객체를 최소경계사각형(MBR)로 표현하여 사각형들의 포함과 겹침관계로 인덱스를 구성하는 R-Tree[28]를 고려할 수 있다. 최적의 성능을 위해서는 겹침의 최소화와 영역의 최소화에 기초한 최적화 인덱스로 R*-Tree[29]를 이용할 수 있다. 이 논문에서 제시하는 이동 패턴 탐사는 2차원 공간과 시간 차원만을 고려한다. 이동객체 인덱싱을 위해 3차원 공간상의 점객체 인덱스를 다룬 STR-Tree[30]나 이동체의 궤적 인덱스에 초점을 둔 TB-Tree[31] 등은 제안하는 기법의 특성에 부합되지 않기 때문에 2차원 영역 인덱싱을 위해 단순히 이 논문에서는 R*-Tree로 가정한다.

특정 영역 내에 특정 좌표 점이 포함되어 있는지 검사하는 메커니즘은 다음과 같다. 우선 임의의 좌표 점 p 에서 시작하는 단 방향 직선 l 을 그린 후, l 과 교차하는 영역 A 의 간선의 개수를 센다. 만약 교차하는 간선의 개수가 짝수이면 점 p 는 A 의 외부에 위치하고, 홀수이면 p 는 A 의 내부에 위치한다. (그림 3)의 예에 이러한 방법을 적용하면 P_1 은 영역 외에 있고, P_3 는 영역 내에 있음을 알 수 있다.

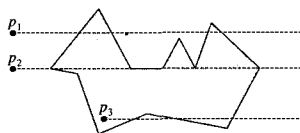


그림 3 영역 내의 점

그러나 P_2 와 같은 경우, 이 방법을 그대로 적용하면 교차하는 간선의 개수가 9개가 되어 영역 내에 있다는 결론이 나오게 된다. 이런 경우를 위해서 l 이 A 의 경계와 동일선상에 있을 경우 계산에 넣지 않으며, 간선 e 의 최소 하나의 끝점이 l 보다 높게 있다면 하나의 교차하는 간선으로 계산한다. 그러면 P_2 를 교차하는 간선의 개수가 6개가 되어 영역 외에 있다는 것을 알 수 있다.

(그림 4)의 알고리즘은 두 좌표 점과, 특정 영역을 입력으로 했을 때 영역이 좌표 점을 포함하고 있는지를 테스트하는 Contains 공간 연산을 보여준다. 좌표 점이 영역 안에 있을 때 참을 반환하고, 그렇지 않을 때는 거짓을 반환한다. 여기서 $Intersect(edge(i),l)$ 함수는 영역의 i 번째 간선이 직선 l 과 교차하는지를 검사하는 함수이며, $Collinear(edge(i),l)$ 함수는 영역의 i 번째 간선이 직선 l 과 수평인지를 검사하는 함수이다.

```

Boolean Contains(p, A)
Begin
  if (p is on one edge of A), return true // p is inside A
  else
    count = 0
    l = horizontal ray containing p
    for (i=1 to n)
      begin
        if (Intersect(edge(i),l) and not Collinear(edge(i),l)) then
          begin
            if (one endpoint of edge(i) is strictly above l) count = count+1
          end
        end for
      if (count is odd)
        return true // p is inside A
      else
        return false // p is not inside A
      endif
    end
  end

```

그림 4 Contains 연산

<표 3>은 <표 2>의 정렬된 데이터베이스의 이동 객체 각각의 공간 속성 값에 Contains 연산을 적용하여 일반화된 지역으로 변환한 예이다.

표 3 Contains 연산 적용 후 위치

oid	vt	Location
1	2001/10/30/13/10	A
	2001/10/30/13/15	B
	2001/10/30/13/25	C
	2001/10/30/13/38	D
2	2001/11/01/12/30	A
	2001/11/01/12/38	E
	2001/11/01/12/45	B
	2001/11/01/12/56	C
3	2001/10/30/14/11	B
	2001/10/30/14/17	C
	2001/10/30/14/23	D
	2001/10/30/14/58	H

5.3 이동 시퀀스 추출 단계

이 단계는 각 이동 점 객체에 대한 이동 시퀀스가 추출되는 단계이다. 즉, 이동 패턴 마이닝에 쓰이는 하나의 트랜잭션이 만들어지는 것이다.

패턴 탐사의 대상이 되는 하나의 시퀀스가 명확하게 정의되는 트랜잭션 데이터베이스에서의 패턴 탐사와 달리, 이동 패턴 탐사의 대상이 되는 시퀀스는 명확하게 정의되지 않는다. 예를 들면, 한 객체 $Mpoint_i$ 이 A, B, C, D 지역을 순서적으로 방문했다고 가정하자. 시간을 고려하지 않을 경우, 이 네 지역은 객체 $Mpoint_i$ 의 이동 경로, 즉 $A \rightarrow B \rightarrow C \rightarrow D$ 가 될 수 있다. 그러나, C 지역에서 D지역을 방문하는 데 걸리는 시간이 한 주라고 가정한다면, 객체 $Mpoint_i$ 의 이동 경로가 여전히 $A \rightarrow B \rightarrow C \rightarrow D$ 라고 보기는 어렵다. 다시 말하면, 객체 $Mpoint_i$ 의 이동 경로는 $A \rightarrow B \rightarrow C$ 와 D, 두 개의 시퀀스로 나뉘어야 한다. 이렇게 의미 있는 이동 시퀀스를 생성하기 위해서 시퀀스를 구성하는 지역 간에 최대 시간 제약조건을 두어서 두 위치 사이의 시간이 최대 시간 제약조건 내에 있을 때만 하나의 시퀀스로 생성하도록 해야 한다.

따라서 이 과정에서는 객체의 공간 속성에 대한 유효 시간을 검사해서 특정 위치에 머무른 기간(duration)이 max_gap 을 초과하면 초과하기 이전의 이동 시퀀스와 이후의 이동 시퀀스로 분리된다.

<표 4>는 <표 3>의 데이터베이스에서 각 객체의 이동 시퀀스를 추출한 것이다. 이 예에서는 max_gap 이 30 분이라고 가정한다. 여기서 유의해야 할 점은 객체 3의 세 번째와 네 번째 위치 사이에 시간 간격이 max_gap 인 30분을 초과하므로 두 개의 이동 시퀀스로 분리된다는 것이다.

표 4 이동 시퀀스

oid	Moving Sequences
1	<A B C D>
2	<A E B C>
3	<B C D>, <H>

5.4 빈발 이동 패턴 마이닝 단계

이 단계는 이동 시퀀스로부터 사용자가 지정한 임계치를 초과하는 빈발한 이동 패턴을 마이닝하기 위한 단계이다. 알고리즘은 효율적으로 후보 집합을 줄일 수 있는 대표적인 연관 규칙 알고리즘인 Apriori[13] 알고리즘을 변형하여 사용한다.

Apriori 알고리즘과 같이 알고리즘의 기본적인 구조

는 데이터에 대해서 여러 번의 스캔 과정을 거친다. 각 패스에서 후보 시퀀스를 생성하고 데이터베이스에서 후보 집합을 포함하는 이동 시퀀스의 개수를 계산한다. 각 패스에서 생성된 빈발 이동 시퀀스는 다음 패스의 후보 시퀀스를 만들기 위해 사용된다. 알고리즘은 패스가 끝난 후 더 이상 빈발 이동 시퀀스가 없거나 생성될 후보 집합이 없으면 끝난다.

F_k 는 빈발 k -시퀀스를 표현하고 C_k 는 후보 k -시퀀스를 표현한다. C_k 는 F_{k-1} 의 셀프 조인, 즉 $F_{k-1} * F_{k-1}$ 을 통해 생성된다. 이 때, F_{k-1} 에 존재하는 각각의 이동 시퀀스 s_1, \dots, s_{k-1} 과 s'_1, \dots, s'_{k-1} 가 존재할 때, s'_1, \dots, s'_{k-1} 가 s_1, \dots, s_{k-2} 을 포함하거나, s_1, \dots, s_{k-1} 가 s'_1, \dots, s'_{k-2} 를 포함할 때 조인이 이루어진다. 그런 다음, C_k 내의 시퀀스에 대하여 F_{k-1} 에 포함되지 않은 부분 시퀀스를 포함하는 시퀀스는 삭제된다. 이 과정은 빈발하지 않은 집합들의 상위 집합들(super sets)은 빈발하지 않다는 관찰에서 수행된다.

후보 집합을 저장하고 이동 시퀀스에 해당되는 후보 집합이 있는지 효율적으로 검색하기 위해 해쉬 트리를 사용한다. 해쉬 트리는 해쉬 테이블과 항목집합을 포함하는 내부노드(internal node)와 잎노드(leaf node)로 구성된다. 내부노드의 해쉬 테이블의 각 버킷은 자식 노드를 가리키는 데, 깊이 d의 내부 노드는 깊이 d+1의 노드를 가리킨다. 후보 항목집합을 해쉬 트리에 추가할 때 루트로부터 잎노드에 이르기까지 깊이를 1씩 증가시키면서 트리에 저장한다. 깊이 d의 내부 노드에서 후보 항목집합의 d 번째 항목을 해쉬 함수에 적용하여 자식 노드를 결정한다.

min_sup 이 2개의 시퀀스라고 가정했을 때, <표 4>의 이동 시퀀스로부터 추출된 빈발 이동 패턴은 {<A>, , <C>, <D>, <A B>, <A C>, <B C>, <C D>, <A B C>}이 되며, (그림 5)는 이 과정을 보여준다.

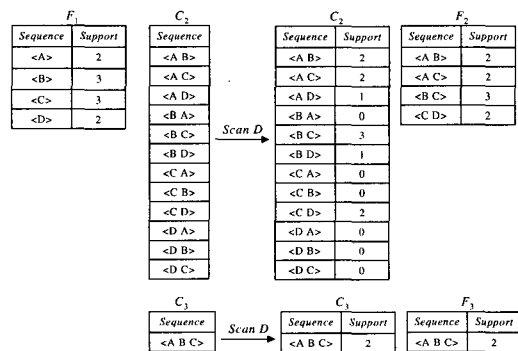


그림 5 후보 시퀀스 생성 예

지금까지 이동 패턴 마이닝을 위한 수행 과정을 네 단계로 분리하여 알아보았으며 (그림 6)은 이러한 각 단계별 알고리즘 MP(Moving Pattern mining)의 전체 구성을 보여준다.

```

MP(database D, max_gap, min_sup, Area A)
begin
  foreach moving objects mpoint in D do
    foreach location x,y of mpoint do
      foreach area a in the A do
        if (contains(mpoint, a))
          set location of mpoint as a
          break;
        for (i=1; end of a mpoint; i++) do
          begin
            if  $v_i - v_{i-1} > max\_gap$ 
              insert with new sequence id
            end
             $F_1 = \{large\ 1\text{-sequence}\};$ 
            for ( $k=2; F_{k-1} \neq \emptyset; k++$ ) do
              begin
                 $C_k = \text{New candidates generated from } L_{k-1}$ 
                foreach moving-sequence c in D do
                  increment the count of all candidates in  $C_k$  that are contained in c
                 $F_k = \text{Candidates in } C_k \text{ with } min\_sup$ 
              end
            Answer = Frequent Sequences in  $U_k F_k$ 
          end
        end
      end
    end
  end

```

그림 6 이동 패턴 마이닝 알고리즘 MP의 전체 구성

6. 실험 및 평가

이 장에서는 이동 패턴 탐사 알고리즘을 구현한 후, 실험 데이터를 통해 제안된 알고리즘의 성능을 분석하였다.

6.1 실험 환경 및 데이터 생성

제안한 알고리즘은 Windows 환경에서 Java 1.3을 이용하여 구현하였으며, 데이터베이스는 Oracle 7.3.4을 사용하였다. 성능 평가를 위해 사용된 시스템은 Pentium PC 200MHz 64Mbyte이며, 데이터베이스 서버로 사용된 시스템은 Ultrasparc 143MHz 64Mbyte이다. 오라클과의 연동을 위해 JDBC 드라이버를 사용하였다.

실험은 [13]에서 사용한 데이터 생성기를 변형하여 이용하여 임의의 데이터를 생성하여 수행하였다. 여기서 변형한 부분은 $|E|$ 로서, 이동 객체의 좌표로 표현된 위치 속성을 좌표가 속해 있는 영역으로 변환 될 때, 검사하는 영역을 구성하는 평균 간선의 개수를 나타낸다. 데이터 생성 파라미터는 <표 5>와 같다.

성능 평가를 위해 $|D|=20000$, $N=2000$ 그리고 $N_s=2000$ 으로 생성 파라미터를 결정하였다. <표 6>은 데이터 생성을 위한 파라미터 설정을 보여준다.

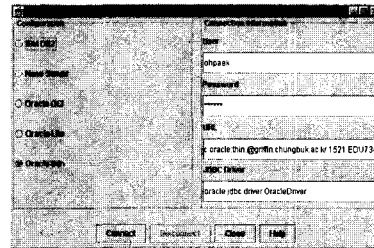
표 5 데이터 생성 파라미터

입력변수	설명
D	데이터베이스에 입력되는 전체 이동 시퀀스의 수
C	이동 시퀀스 당 평균 영역의 개수
E	영역을 구성하는 간선의 평균 개수
S	잠재적 빈발 시퀀스의 평균 길이
N_s	잠재적 빈발 시퀀스의 수
N	영역의 수

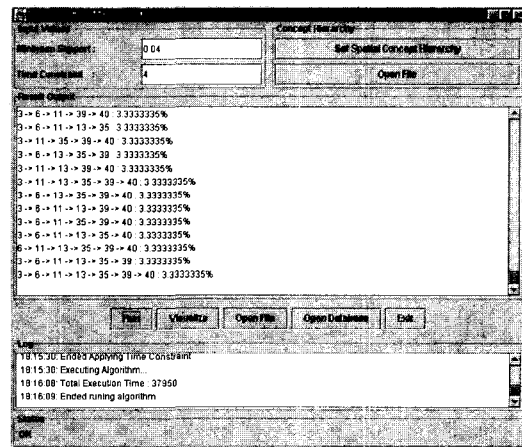
표 6 데이터 생성을 위한 파라미터 값

생성 데이터 이름	C	E	S
C5.E4.S4	5	4	4
C5.E6.S4	5	6	4
C10.E4.S4	10	4	4
C10.E4.S8	10	4	8

실험 데이터의 생성은 시퀀스를 구성하는 지역의 개수, 영역을 구성하는 간선의 개수, 결과 패턴의 평균 길이에 변화를 주어 생성하도록 하였다.



(a) 데이터베이스 연결



(b) 알고리즘 실행 후 결과

그림 7 구현 결과 화면

(그림 7)은 구현된 GUI를 보여주며 자바 스윙 컴퍼넌트를 이용하여 구현하였다. 7(a)는 데이터베이스와의 연결 설정을 위한 창이며, 7(b)는 결과를 출력하기 위한 것으로 다섯 부분으로 구성되어 있다. 사용자의 입력을 받는 부분은 최소 지지도와 공간 속성 간 최대 시간을 입력 받는다. 결과 출력 부분은 알고리즘 수행 결과와 중간 결과들을 화면에 보이도록 하였다. 실행버튼은 알고리즘을 수행하며 대상 데이터를 파일 또는 데이터베이스를 통해 입력받을 수 있도록 하였다. 각 단계의 수행시간과 과정을 로그로 남기고 상태를 표시하기 위한 패널을 두었다.

6.2 성능 평가

성능 평가에서는 생성된 데이터를 이용하여 알고리즘의 최소 지지도의 변경, 전체 이동 시퀀스의 개수, 이동 시퀀스 당 평균 영역의 수를 기준으로 성능을 측정하였다.

(그림 8)은 생성된 각각의 데이터 집합에 최소 지지도를 1%부터 0.35%까지 줄여가면서 수행시간을 측정하고 결과를 보여준다.

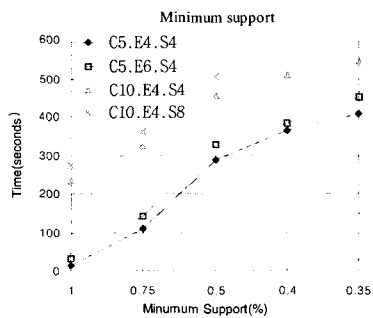


그림 8 최소 지지도의 변경에 따른 수행시간

각각의 데이터 집합은 지지도가 낮아지면 낮아질수록 선형적으로 수행시간이 늘어나는 것을 볼 수 있다. C5.E4.S4와 C5.E6.S4에서 볼 수 있는 것처럼 영역의 다각형을 구성하는 간선의 평균 개수가 커지면 실행 시간도 증가하는 것을 볼 수 있는데, 이는 이동 시퀀스를 구성하는 개개 좌표의 선이 영역의 간선과 교차하는지 검사하는 데 드는 비용이라 할 수 있다. 하나의 시퀀스를 구성하는 영역의 평균 개수가 많을수록, 빈발 이동 패턴을 구성하는 영역의 평균 개수가 많을수록 수행시간이 선형적으로 증가하였다. 이것은 각각의 개수가 늘어날수록 스캔해야 할 데이터베이스가 커지기 때문인 것으로 분석된다. 데이터 셋 C10.E4.S8에 대해서는 지지도가 0.4%가 되는 부분에서는 후보 집합이 해쉬 트리에 저장할 수 없을 만큼 많이 생성되었다.

(그림 9)는 전체 이동 시퀀스의 수, 즉 데이터베이스의 크기를 늘여가며 1%, 1.5%, 2%의 지지도 값에 대해 수행시간을 측정한 결과이다. 이때의 N_s 4000, N 은 2000으로 설정하였으며, $|C|$ 는 5, $|S|$ 는 4로 설정하였다. 데이터베이스의 크기가 늘어날수록 수행 시간은 서서히 증가하였으며, 지지도 1.5%와 2%는 거의 동일한 실행 시간을 보였으나 1%로 떨어지면서 급격히 증가하였다. 이것은 지지도가 낮아지면서 생성되는 후보 집합의 개수가 많아졌기 때문이다.

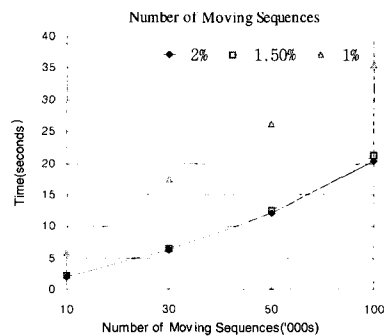


그림 9 전체 이동 시퀀스의 수에 대한 수행시간

(그림 10)은 하나의 이동 시퀀스를 구성하는 영역의 평균 개수를 5부터 20까지 늘여가며 수행시간을 측정한 결과이다. 이때의 $|D|$ 는 20000이며, N_s 는 4000, N 은 2000으로 설정하였다. 한 이동 시퀀스를 구성하는 영역의 개수가 많아질수록 수행시간은 선형적으로 증가함을 볼 수 있다. 특히 지지도 2%와 1.5%의 경우 영역의 개수가 10 이하일 때는 거의 변동이 없었으나 10을 초과하면서 증가의 폭이 커졌다. 이것은 시퀀스의 길이가 길어지면서 생성하는 후보집합의 크기가 커졌기 때문으로 분석된다.

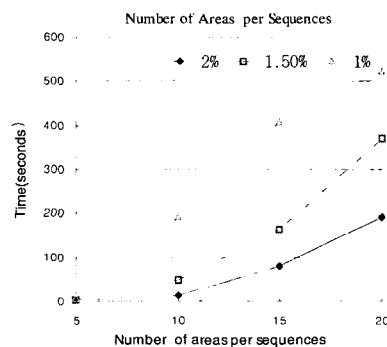


그림 10 이동 시퀀스 당 평균 영역의 개수에 대한 수행 시간

성능 평가를 통해 얻을 수 있는 결과를 정리하면 첫째, 알고리즘의 전체 과정에서 이동 객체의 위치 일반화 단계와 빈발 이동 패턴 마이닝 단계는 서로의 성능에 영향을 미치지 않는다. 둘째, 사용자가 입력한 지지도는 알고리즘의 성능에 큰 영향을 미친다. 이것은 지지도가 낮아질수록 생성하는 후보 집합의 개수가 많아지고 데이터베이스를 스캔해야 하는 비용이 커지기 때문이다. 셋째, 이동 시퀀스의 길이가 길어질수록, 특히 길이가 10을 초과하면 수행시간이 증가함으로써 효율적인 알고리즘의 개발과 저장구조의 개선이 필요함을 알 수 있다. 넷째, 입력 데이터베이스의 크기에 따라 알고리즘의 수행시간은 서서히 증가하였다.

7. 결론 및 향후 연구

현재 위치 기반 서비스에 대한 관심이 증대되고 있으며, 특히 사용자 위치와 정보를 기반으로 하여 개인화되고 위치에 적합한 지식을 사용하여 제시하기 위하여 데이터 마이닝의 사용은 필수적이다. 기존의 데이터 마이닝 기법에서는 동적으로 변하는 이동 객체의 위치 정보에 대한 접근은 이루어지지 않았다.

따라서 이 연구에서는 위치 기반 서비스를 요청하고 받는 개개의 사용자를 이동 객체로 보고, 위치 기반 서비스에 적용하기 위해 시간에 따라 이동하는 사용자의 위치 정보로부터 의미 있는 패턴을 탐사하는 방법을 제안하였다.

이동 패턴을 탐사하기 위한 알고리즘은 네 단계로 구성하였다. 먼저, 데이터베이스를 객체 식별자와 트랜잭션 식별자로 정렬하고, 이동 점 객체의 공간 위치 정보에 대해 공간 연산자를 적용하여 위치 일반화를 수행하였다. 위치 일반화를 통해 이동 객체의 위치 정보를 의미 있는 지식을 찾을 수 있도록 변환하였다. 이동 시퀀스 추출 단계에서는 유효한 이동 시퀀스를 생성하기 위해 시간 제약 조건을 적용하였다. 이렇게 생성된 이동 시퀀스로부터 빈발 시퀀스를 추출함으로써 이동 패턴을 탐사하였다. 실험에서는 제안한 알고리즘을 Java를 사용하여 구현하였고, 성능 평가를 위하여 데이터를 생성하여 실험을 수행하였다. 알고리즘의 성능은 지지도가 낮아질수록, 이동 시퀀스의 길이가 길어질수록, 데이터베이스의 크기가 커질수록 서서히 증가함을 볼 수 있었다. 제안된 기법은 이동 객체의 위치 데이터에 대하여 웹 로그 분석이나 트랜잭션 분석에 적용된 기법과는 다른 시,공간적 접근을 취함으로써 이동 패턴에서의 위치 의미와 이동 시간에 따른 시간의미를 함께 포함할 수 있다. 따라서 제안하는 이동 패턴은 개인화되고 차별화된

위치기반 정보서비스 및 M-Commerce 등에 매우 실질적으로 적용될 수 있다.

향후 이동 객체의 위치 정보 뿐 아니라 속도나 방향 등의 행위정보와 이동 객체가 한 지역에 머무른 시간을 고려하여 패턴을 탐사하기 위한 방법의 연구가 수행되어야 한다. 또한, 다양한 이동 객체 타입을 고려해야 하고, 이동 시퀀스의 길이가 길 때 효율적으로 패턴을 탐사하기 위한 효율적인 기법의 개발이 필요하다.

참고 문헌

- [1] 김옥, 지규인, 이장규, "위치 기반 무선 인터넷 서비스", *Telecommunications Review*, 제10권, 제6호, pp.1260-1269, 2000.
- [2] M. N. Garofalakis, R. Rastogi, and K. Shim, "SPIRIT : Sequential Pattern Mining with Regular Expression Constraints," *Proceedings of the 25th International Conference on Very Large Databases*, 1999.
- [3] O. Wolfson, A. P. Sistla, B. Xu, J. Zhou, and S. Chamberlain, "DOMINO : Databases fOr MovINg Objects tracking," *Proceedings of the ACM-SIGMOD International Conference on Management of Data*, pp.547-549, 1999.
- [4] 류근호, 이준욱, 이용준, "eCRM을 위한 시간 데이터 마이닝 기술", *한국 정보과학회 데이터베이스연구회지*, 제17권, 제1호, 2001.
- [5] 이용준, 서성보, 류근호, 김혜규, "시간간격을 고려한 시간관계 규칙 탐사 기법", *한국정보과학회 논문지*, 제 28권, 제 3호, pp.301-314, 2001.
- [6] 안병익, "LBS기술동향과 전망 - LBS 구조 및 구성", *한국지리정보*, 10월호, pp.52-56, 2001.
- [7] J.S. Song, Y. J. Lee, and K. H. Ryu, "Discovering Temporal Relation Rules from Interval Data," submitted to the *ETRI Journal*, 2001.
- [8] R. Agrawal and R. Srikant, "Mining Sequential Patterns," *Proceedings of the 11th International Conference on Data Engineering*, pp.3-14, 1995.
- [9] R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements," *International Conference on Extending Database Technology*, Springer-Verlag, 1996.
- [10] M.-S. Chen, J. S. Park, and P. S. Yu, "Efficient Data Mining for Path Traversal Patterns," *IEEE Transactions on Knowledge and Data Engineering*, Vol.10, No.2, pp. 209-221, 1998.
- [11] J. Borges, M. Levene, "A Fine Grained Heuristic to Capture Web Navigation Patterns," *SIGKDD Explorations*, Vol.2, No.1, pp.40-50, 2000
- [12] J. Pei, J. Han, B. Mortazavi-Asl and H. Zhu,

- "Mining Access Patterns Efficiently from Web Logs," Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining(PAKDD), 2000.
- [13] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules," Proceedings of the 20th International Conference on Very Large Databases, pp. 487-499, Santiago, Chile, 1994.
- [14] H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovery of Frequent Episodes in Event Sequences," Data Mining and Knowledge Discovery, Vol.1, No.3, pp.259-289, 1997.
- [15] J. Han, G. Dong, and Y. Yin, "Efficient Mining of Partial Periodic Patterns in Time Series Database," Proceedings of the 11th International Conference on Data Engineering, 1999.
- [16] B. Ozden,, S. Ramaswamy, and A. Silberschatz, "Cyclic Association Rules," Proceedings of the 14th International Conference on Data Engineering, 1998.
- [17] P.S. Kam and A. Fu, "Discovering Temporal Patterns for Interval-Based Events," Proceedings of the 2nd International Conference on Data Warehousing and Knowledge Discovery, (DaWak), Springer Verlag, LNCS, London, UK, 4-6 Sept, 2000.
- [18] X. Chen, I. Petrounias, and H. Heathfield, "Discovering Temporal Association Rules in Temporal Databases," Proceedings of the International Workshop on Issues and Applications of Database Technology(IADT'98), pp.312-319, 1998.
- [19] J. F. Allen, "Maintaining Knowledge about Temporal Intervals," Communication of the Association of Computing Machinery, Vol.26, No.11, 1983.
- [20] T. Abraham and J. F. Roddick, "Discovering Meta-rules in Mining Temporal and Spatio-temporal data," Proceedings of the International Database Workshop, Data Mining, Data Warehousing and Client/Server Databases, (IDW'97), pp.30-41, 1997.
- [21] E. Tsoukatos and D. Gunopoulos, "Efficient Mining of SpatioTemporal Patterns," Proceedings of the 7th International Symposium on Spatial and Temporal Databases(SSTD), pp.425-442, 2001.
- [22] Seong Seung Park, Yun Ae Ahn, and Keun Ho Ryu, "Moving Objects Spatiotemporal Reasoning Model for Battlefield Analysis," In Proc. of Military, Government and Aerospace Simulation part of ASTC2001, pp.108-113, Apr., 2001.
- [23] 안윤애, 류근호, "이동 객체의 불확실한 위치 정보 관리", 충북대학교 컴퓨터정보통신 연구, 제 9권, 제 1호, pp.81-91, 2001.
- [24] M. Erwig, R.H. Guting, M. Schneider, and M. Vazirgiannis, "Spatio-Temporal Data Types : An Approach to Modeling and Querying Moving Objects in Databases," GeoInformation, Vol.3, No.3, pp. 269-296, 1999.
- [25] L. Forlizzi, R. H. Guting, E. Nardelli and M. Schneider, "A Data Model and Data Structures for Moving Objects Databases," Proceedings of the ACM-SIGMOD International Conference on Management of Data, pp.319-330, 2000.
- [26] R. H. Guting, M. H. Bohlen, M. Erwig, C. S. Jensen, N. A. Lorentzos, M. Schneider, and M. Vazirgiannis, "A Foundation for Representing and Querying Moving Objects," ACM Transactions on Database Systems, 2000.
- [27] InBae Oh, YoonAe Ahn, EungJae Lee, KeunHo Ryu, HongGi Kim, "Prediction of Uncertain Moving Object Location," In Proc. of Int. Conf. on East-Asian Language Processing and Internet Information Technology 2002 (EALPIIT2002 HANOI), 2002.
- [28] A. Guttman, R-trees: a Dynamic Index Structure for Spatial Searching," In Proceedings of the ACM SIGMOD Conference on the Management of Data, pp.47-57, 1984
- [29] N. Beckmann, H. P. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," ACM SIGMOD Conference, pp.322-331, 1990.
- [30] D. Pfoser, Y. Theodoridis, and C. S. Jensen, "Indexing Trajectories of Moving Point Objects," CHOROCHRONOS Technical Report CH-99-03, October, 1999.
- [31] D. Pfoser, C. S. Jensen, and Y. Theodoridis, "Novel Approaches in Query Processing for Moving Objects," CHOROCHRONOS Technical Report CH-00-3, February, 2000
- [32] R. J. Bayardo Jr., "Efficiently Mining Long Patterns from Databases," Proceedings of the ACM-SIGMOD International Conference on Management of Data, pp. 85-93, 1998.



이준욱

1997년 충북대 컴퓨터과학과 졸업. 1999년 충북대 대학원 전자계산학과(이학석사). 1998년 ~ 1999년 한국전자통신연구원 위촉연구원 근무. 2002년 현재 충북대학교 대학원 컴퓨터과학과 박사과정. 관심 분야는 시간 데이터베이스, 시공간 데이터베이스, CRM, 시간 데이터 마이닝, 시공간 데이터 마이닝



백 옥 현

2000년 충북대학교 정치외교학과(학사).
2002년 충북대 대학원 전자계산학과(이
학석사). 2002년 현재 국방과학연구소 연
구원. 관심분야는 시간데이터베이스, 시
간 데이터 마이닝, LBS



류 근 호

1976년 숭실대 전산과 졸업. 1980년 연
세대학교 산업대학원 전산전공(공학석
사). 1988년 연세대 대학원 전산전공(공
학박사). 1976년 ~ 1986년 육군군수지원
사전산실(ROTC장교), 한국전자통신연구
소(연구원), 한국방송통신대, 전산학과(조
교수) 근무. 1989년 ~ 1991년 Univ. of Arizona 연구원
(TempIS Project). 1986년 ~ 현재 충북대학교 전기전자
및 컴퓨터공학부 교수. 관심분야는 시간 데이터베이스, 시공
간 데이터베이스, Temporal GIS, 객체 및 지식베이스 시스
템, 지식기반 정보검색시스템, 데이터 마이닝, 데이터베이스
보안 및 Bio-Informatics