대규모 궤적 데이타를 위한 데이타 마이닝 툴

(A Data Mining Tool for Massive Trajectory Data)

이 재 길 †

(Jae-Gil Lee)

요 약 궤적(trajectory) 데이타는 실세계 어디에서든지 쉽게 찾아볼 수 있다. 최근 들어, 위성, 센서, RFID, 비디오 및 무선 통신 기술의 발전으로 말미암아 이동 객체를 체계적으로 추적하고, 많은 양의 궤적 데이타를 수집할 수 있게 되었다. 이에 따라, 궤적 데이타의 분석에 대한 필요성이 점차 증대되고 있다. 본 논문에서는 **대규모 궤적 데이타를 위한 마이닝 툴**을 개발한다. 본 마이닝 툴에서는 가장 널리 사용되는 마이닝 연산인 **집단화**(*clustering*), **분류**(*classification*), **이상치 발견**(*outlier detection*)을 제공한다. 궤적 집단화는 공통적인 이동 패턴을 발견하며, 궤적 분류는 궤적에 기반하여 이동 객체의 범주를 예측하며, 궤적 이상치 발견은 나머지 궤적들과 크게 다르거나 일관적이지 않은 궤적을 발견한다. 본 마이닝 툴의 가장 큰 장점은 데이타 마이닝 도중에 **부분** 궤적 정보를 활용한다는 점이다. 본 마이닝 툴의 우수성은 다양한 실제 궤적 데이타 셋을 사용하여 입증되었다. 본 논문의 결과로 궤적 데이타 마이닝을 위한 실용적인 소프트웨어를 개발하였고 많은 실제 응용에 적용될 수 있을 것이라 사료된다.

키워드 : 데이타 마이닝, 궤적 데이타, 집단화, 분류, 이상치 발견

Abstract Trajectory data are ubiquitous in the real world. Recent progress on satellite, sensor, RFID, video, and wireless technologies has made it possible to systematically track object movements and collect huge amounts of trajectory data. Accordingly, there is an ever-increasing interest in performing data analysis over trajectory data. In this paper, we develop *a data mining tool for massive trajectory data*. This mining tool supports three operations, *clustering, classification*, and *outlier detection*, which are the most widely used ones. Trajectory clustering discovers common movement patterns, trajectory classification predicts the class labels of moving objects based on their trajectories, and trajectory outlier detection finds trajectories that are grossly different from or inconsistent with the remaining set of trajectories. The primary advantage of the mining tool is to take advantage of the information of *partial* trajectories in the process of data mining. The effectiveness of the mining tool is shown using various real trajectory data sets. We believe that we have provided practical software for trajectory data mining which can be used in many real applications.

Key words : Data Mining, Trajectory Data, Clustering, Classification, Outlier Detection

1. 서 론

최근 들어, 위성, 센서, RFID, 비디오 및 무선 통신 기술의 발전으로 말미암아 이동 객체를 체계적으로 추적하고, 많은 양의 **궤적**(*trajectory*)을 수집할 수 있게 되었다. 궤적 데이타는 이동 객체의 특정 시점에서의 위치를 연속적으로 기록한 데이타이다. 이러한 궤적 데이타의 예로는 차량 이동 데이타, 선박 이동 데이타, 동물 추적 데이타, 태풍 경로 데이타 등 수없이 많다. 이렇듯 많은 양의 궤적 데이타를 분석하고 유용한 지식을 발견할 수 있도록 궤적 데이타를 위한 데이타 마이닝 연구가 활발히 진행되고 있다[1,2].

· 이 논문은 2006년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2006-214-D00129)

† 정 회 원 : 어바나-샴페인 일리노이 주립대학 전산학과 박사후연구원
jaegil@gmail.com

논문접수 : 2008년 11월 6일
심사완료 : 2009년 1월 6일

정보과학회논문지: 컴퓨팅의 실제 및 레터 제15권 제3호(2009.3)

궤적 데이타 마이닝 기법은 크게 3가지로 구분될 수 있다. 첫째, 유사한 궤적들을 같은 그룹으로 묶어주는 **집단화**(*clustering*)이다[3-5]. 응용 예로서는 태풍 경로 데이타에서 많은 태풍들이 공통적으로 움직인 경로를 집단화를 통해 얻어낼 수 있다. 둘째, 다양한 궤적들을 이미 정의된 범주(class) 중의 하나에 할당하는 **분류**(*classification*)이다[6]. 응용 예로서는 레이더를 사용해 얻어낸 선박 이동 데이타에서 각 선박의 종류(예: 어선, 화물선, 유조선)를 분류를 통해 유추해 낼 수 있다. 셋째, 다른 궤적들과 큰 차이를 보이는 궤적을 찾아내는 **이상치 발견**(*outlier detection*)이다[7-9]. 응용 예로서는 주차장의 비디오 감시 시스템을 사용해 얻어낸 사람의 경로 데이타에서 이동 경로가 의심스러운 사람을 이상치 발견을 통해 찾을 수 있다.

궤적 데이타를 위한 집단화, 분류, 이상치 발견과 관련하여 많은 기존 연구가 있지만, 이들 기존 연구의 공통적인 제약 사항은 각각의 **전체** 궤적을 하나의 데이타 객체로 간주한다는 점이다. 일반적으로 궤적 데이타는 여러 위치(점)의 연속으로 이루어져 있기 때문에, 이를 하나의 객체로 간주하면 중요한 정보를 잃어버릴 수 있다. 그림 1과 같이 5개의 궤적이 있다고 가정하자. 이들 5개의 궤적은 서로 다른 방향으로 움직이고 있으나, 표시된 바와 같이 공통적인 **부분** 궤적을 가지고 있다. 기존의 궤적 집단화 알고리즘들은 전체 궤적간의 유사도에 따라 집단화를 수행하기 때문에 공통적인 부분 궤적을 발견할 수 없다.

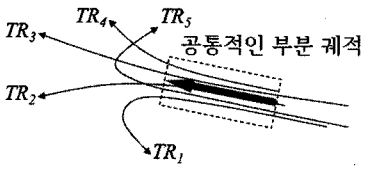


그림 1 기존 궤적 집단화 알고리즘의 한계

Lee 등[3,6,7]은 이러한 문제점을 해결하기 위해 **부분** 궤적 정보를 활용하는 데이타 마이닝 알고리즘들을 제안하였다. 이들 알고리즘들은 기존 알고리즘들에서는 찾아낼 수 없었던 부분 궤적 결과까지도 찾아낼 수 있다는 장점을 가지고 있다. 이들 알고리즘들은 공통적으로 2단계로 구성되어 있다. 첫 번째 단계에서는 궤적을 우선 선분(line segment)으로 분할하고, 두 번째 단계에서는 각각 집단화, 분류, 혹은 이상치 발견 알고리즘을 적용한다. 첫 번째 단계에서 생성된 선분들을 하나의 데이타 객체로 간주하기 때문에 부분 궤적 정보를 활용할 수 있게 된다.

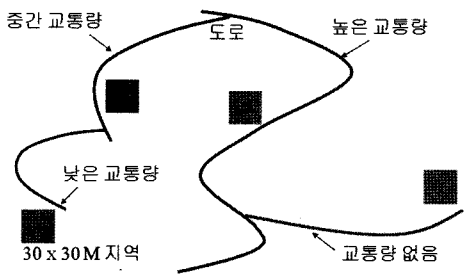


그림 2 동물 이동의 모니터링

부분 궤적 정보를 발견하는 것이 중요함을 **공통적인 부분 궤적**이 필요한 응용의 예를 들어 설명한다. 동물학자들은 차량의 통행량이 동물의 이동, 분포, 서식지에 미치는 영향을 연구하고 있다. 이 연구에서는 교통량이 변화된 도로 근처에서 동물들의 공통적인 행동을 알아내는 것이 중요하다. 그림 2는 실제 동물학 관련 연구 발표에서 발췌되었으며, 사각형으로 표시된 부분이 교통량의 변화에 따라 동물들이 도로 근처를 회피하는 정도를 조사하고 싶은 지역이다. 따라서, 표시된 지역 내에서의 공통적인 부분 궤적을 발견하는 것이 이러한 연구의 목적에 매우 잘 부합된다. 본 데이타 마이닝 툴은 이러한 공통적인 부분 궤적을 찾아내 시각적으로(굵은 선) 표시해준다. 실제 데이타에 대한 실행 결과는 제 3.2절에 있다.

본 논문에서는 Lee 등[3,6,7]이 제안한 궤적 데이타 마이닝 알고리즘을 사용하여 데이타 마이닝 툴을 개발하는 것을 목적으로 한다. 본 데이타 마이닝 툴의 1차적인 구현 동기는 궤적 데이타 분석에 필요한 툴을 얻는 것이고, 2차적인 구현 동기는 실제 마이닝 결과를 바탕으로 기존 마이닝 알고리즘을 개선하고자 하는 것이다. 본 데이타 마이닝 툴은 Windows 플랫폼에서 동작하며 시각적인 결과를 제공하기 때문에, 마이닝 결과를 쉽게 이해할 수 있다. 또한, 동일한 입력 데이타 포맷에 대해 집단화, 분류, 이상치 발견 결과를 동시에 얻을 수 있기 때문에, 다양한 분석이 가능하다는 장점이 있다. 마지막으로, 본 마이닝 결과로 얻어진 통찰력을 바탕으로 결과 품질을 보다 더 개선하기 위해 **시간 부분 궤적 집단**(*temporal sub-trajectory cluster*)이라는 새로운 개념을 제시한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 Lee 등이 제안한 궤적 데이타 마이닝 알고리즘들을 소개한다. 제 3장에서는 본 논문에서 개발한 궤적 데이타 마이닝 툴을 소개한다. 제 4장에서는 시간 부분 궤적 집단을 제안한다. 제 5장에서는 그 밖의 궤적 데이타 마이닝 툴을 소개한다. 마지막으로, 제 6장에서는 결론을 내린다.

2. 궤적 데이타 마이닝 알고리즘

우선 제 2.1절에서는 Lee 등의 알고리즘의 첫 번째 단계에서 수행되는 궤적 분할을 설명한다. 제 2.2절에서는 궤적 집단화 알고리즘[3], 제 2.3절에서는 궤적 분류 알고리즘[6], 제 2.4절에서는 궤적 이상치 발견 알고리즘[7]을 설명한다.

2.1 궤적 분할(trajectory partitioning)

궤적 분할은 특정 궤적을 여러 개의 선분으로 나누는 과정이다. 직관적으로 이동 객체의 이동 방향이 급격히 변한 점에서 분할을 한다. 예를 들어, 그림 3에서 점 p_4와 p_6에서는 반드시 분할해야 한다. 분할점을 선택한 후 선분은 연속적인 분할점을 연결하여 얻어진다.

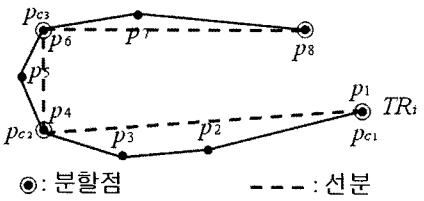


그림 3 궤적 분할의 예시

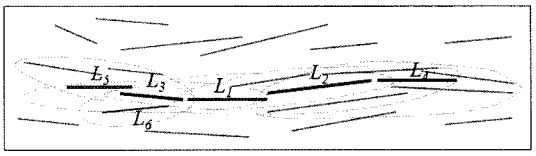
분할점을 선택할 때, **정확성**과 **간결성**의 두 가지 요건을 충족시켜야 한다. 정확성은 선분들의 연속과 원래 궤적과의 차이가 최소화되어야 함을 의미한다. 간결성은 선분들의 개수가 최소화되어야 함을 의미한다. 이 두 가지 요건은 서로 상충하는 요건이다. 극단적으로 모든 점이 분할점이 되면 간결성은 최악이지만, 정확성은 최대가 된다. 반대로 시작점과 끝점만이 분할점이 되면 정확성은 최악이지만, 간결성은 최대가 된다. 이 두 극단 사이의 최적 분할점의 집합을 얻는 것이 궤적 분할의 목적이다.

최적 분할점의 집합은 정보 이론(information theory)의 MDL(minimum description length) 원리에 근거하여 얻어낸다. MDL 비용은 모델 비용과 묘사 비용으로 구성되는데, 이 두 비용의 합이 최소화되는 분할점의 집합을 얻어낸다. 자세한 MDL 비용 수식화는 참고 문헌[3,6,7]에 설명되어 있다.

2.2 궤적 집단화(trajectory clustering)

궤적 집단화 알고리즘은 **밀도 기반** 집단화 알고리즘에 기반한다. 밀도 기반 알고리즘에서 **집단**(*cluster*)은 밀도가 낮은 지역에 의해 분리된 밀도가 높은 지역으로 정의된다. 여기에서 밀도는 특정 반경 내에 있는 데이타 객체의 개수를 의미한다. DBSCAN을 비롯한 밀도 기반 알고리즘은 점 데이타를 위해 개발되었으며, 이들 알고리즘에서는 밀도를 계산할 때 점들 간의 거리만을 고려하였다. 선분을 위한 밀도 기반 알고리즘에서는 선분들 간의 거리뿐만 아니라 선분의 모양까지도 함께 고려해야 한다. 즉, 거리가 가깝고 모양이 비슷한 선분이 있으면 그 선분을 현재의 집단에 계속 포함시켜 나간다. 이 과정을 더 이상 어떤 선분도 현재의 집단에 포함될 수 없을 때까지 반복한다.

그림 4는 밀도 기반 선분 집단화의 예시를 나타낸다. 선분 L_1에서 시작해 선분 L_2와 선분 L_3가 현재 집단에 포함되고, 선분 L_2와 선분 L_3의 이웃에 있는 선분들까지 현재 집단에 추가된다. 유사한 과정을 반복한 결과 선분 L_1~L_6의 타원에 포함되는 선분들이 하나의 집단으로 묶이게 된다. 그림 4에서 볼 수 있듯이 밀도 기반 집단화의 장점은 **임의의 모양**의 집단을 발견할 수 있다는 점이다.



L_6 L_5 ⟷ L_3 ⟷ L_1 ⟷ L_2 ⟷ L_4

그림 4 밀도 기반 선분 집단화의 예시

선분의 집단을 구한 후에, 각 집단 별로 그 집단을 대표하는 이동 형태를 구한다. 이를 **대표 궤적**(*representative trajectory*)라고 부른다. 그림 5에서 분홍색의 선분들이 같은 집단에 속하는 선분들이며, 이 집단을 나타내는 대표 궤적은 굵은 선으로 표시되어 있다. 대표 궤적이 바로 **공통 부분 궤적**을 나타냄을 주목하기 바란다.

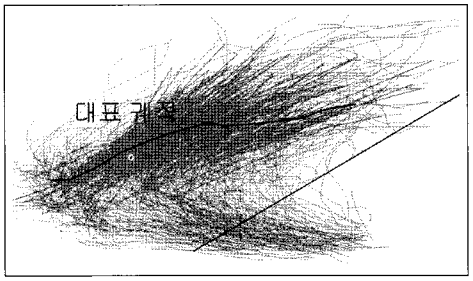


그림 5 대표 궤적의 예시

2.3 궤적 분류(trajectory classification)

궤적 분류에서 가장 핵심적인 연산은 각 범주의 특성을 추출해내는 **특성 추출**(*feature extraction*)이다. 구현된 알고리즘에서는 지역 기반 특성과 궤적 기반 특성의 두 가지 종류의 특성을 추출한다. 첫째, 특정 지역을 한 가지 범주에 속하는 궤적들만이 통과한다면 그 지역은

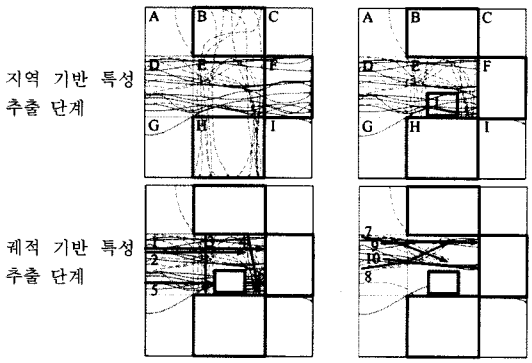


그림 6 지역 기반 특성 추출과 궤적 기반 특성 추출.

해당 범주의 아주 좋은 특성이 될 수 있다. 이를 **지역 기반 특성**이라 부른다. 지역 기반 특성으로 선택되지 않은 선분들은 제 2.2절에서 설명한 집단화 알고리즘에 입력으로 주어진다. 둘째, 같은 범주에 속하는 궤적들이 공통적으로 보이는 이동 패턴이 있다면 그 패턴은 해당 범주의 아주 좋은 특성이 될 수 있다. 이를 궤적 기반 특성이라 부른다. **궤적 기반 특성**은 바로 앞에서 설명한 공통 부분 궤적이다.

그림 6에서 궤적 특성 추출의 두 단계를 보여준다. 두 범주의 궤적들이 각각 실선과 점선으로 표시되어 있다. 윗 줄의 그림에서 굵은 사각형으로 표시된 지역인 **B, F, H, J**는 한 범주의 궤적들만이 통과하고 있으므로 지역 기반 특성으로 추출된다. 아랫 줄의 그림에서 굵은 화살표로 표시된 패턴 **3, 4, 5, 6, 7, 8, 9, 10**은 한 범주의 공통적인 이동 형태를 나타내고 있으므로 궤적 기반 특성으로 추출된다. 패턴 **1, 2**는 다른 범주이지만 서로 유사하기 때문에 궤적 기반 특성으로 추출되지 않고, 대신 보다 작은 단위(fine graunlarity)의 집단을 구성하여 다른 범주 사이에 차이가 생기도록 하였다.

이와 같이 추출된 특성은 SVM(support vector machine)에 공급되어 분류 모델을 구성하는데 사용된다. SVM을 위한 많은 공개 라이브러리가 발표되어 있으며, 본 논문에서는 가장 널리 사용되는 라이브러리인 LIBSVM[1]을 채택한다.

2.4 궤적 이상치 발견(trajectory outlier detection)

궤적 이상치 발견 알고리즘은 **거리 기반** 이상치 발견 알고리즘에 기반한다. 거리 기반 알고리즘에서 **이상치**(*outlier*)는 대다수의 나머지 데이타 객체와 멀리 떨어져 있는 데이타 객체로 정의된다. 다시 말해, 이상치의 근방에는 매우 소수의 데이타 객체만이 존재하게 된다. 구현된 알고리즘에서 궤적 이상치 발견은 두 단계로 진행된다. 우선 궤적에서 추출된 선분에 대해서 **이상 선분**(*outlying partition*)을 가려내게 된다. 이상 선분이란 그 근방에 모양(이동 형태)이 비슷한 선분이 매우 적은 선분을 의미한다. 그리고, 이상 선분을 특정 비율 이상으로 포함하고 있는 궤적이 최종 결과인 이상 궤적으로 결정된다.

그림 7은 이상 궤적을 발견하는 단계를 보여준다. 붉은 색으로 표시된 선분은 주요 이동 형태와 상이하기 때문에 이상 궤적으로 판별된다. 그리고, 이들 이상 선분이 포함되어 있는 해당 궤적은 이상치로 판별된다. 따라서, 전체 궤적이 특이하지 않고 일부분만 특이하더라도 효과적으로 이상 궤적으로 발견될 수 있다.

그림 7 이상 궤적(outlier)의 정의

3. 궤적 데이타 마이닝 툴(tool)

본 논문에서 구현한 데이타 마이닝 툴은 크게 **TRACLUS, TRACLASS, TRAOD**의 세 종류의 패키지로 구성되어 있다. 각각 집단화, 분류, 이상치 발견을 담당한다. 제 3.1절에서는 시스템의 아키텍처를 설명하고, 제 3.2절에서는 실행 화면과 주요 기능을 설명한다.

3.1 시스템 아키텍처

그림 8은 본 논문에서 구현한 데이타 마이닝 툴의 아키텍처를 보여준다. 궤적 입력기는 궤적 데이타베이스에서 궤적 데이타를 읽어서 메모리 상의 데이타 구조에 저장하는 역할을 담당한다. 궤적 분할기는 제 2.1절에서 설명한 알고리즘에 따라 하나의 궤적을 여러 개의 선분으로 분할하는 작업을 담당한다. 이 과정에서 생성된 선분들은 모두 메모리 상에서 유지된다. 집단화 모듈, 분류 모듈, 이상치 발견 모듈은 각각 제 2.2절, 2.3절, 2.4절에서 설명한 연산을 수행하는 작업을 담당한다. 전 과정에서 얻어진 선분들이 입력으로 주어지며, 이들 입력 데이타는 메모리 관리자를 통해 얻어온다. 마지막으로, 결과 출력기는 각 연산의 결과를 시각적으로 화면에 표시하는 작업을 담당한다.

본 마이닝 툴의 개발 기준은 성능을 최대화하며 코드를 간략하게 만드는 것이다. 첫 번째 기준을 충족하기 위해 모든 궤적 데이타 및 선분 데이타를 메모리에 로드한다. 모든 연산은 메모리 상에 로드되어 있는 데이타만을 사용해서 처리된다. 메모리에 모두 로드될 수 없는 경우의 처리는 향후 개발 예정이다. 최근에는 기본 메모리의 크기가 10GB에 이르므로 현재 아키텍처가 큰 문

1) http://www.csie.ntu.edu.tw/~cjlin/libsvm/

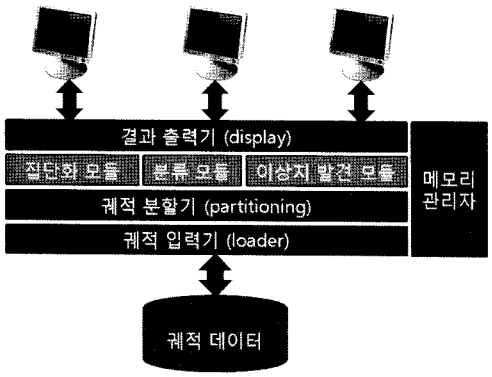


그림 8 궤적 마이닝 툴의 아키텍처

제가 되지 않을 것이라 사료된다. 두 번째 기준을 충족하기 위해 세 가지 연산에서 공통적으로 사용되는 궤적 입력기, 궤적 분할기, 결과 출력기, 메모리 관리자의 코드가 중복되지 않도록 한다.

메모리 관리자는 궤적과 세 가지 모듈의 결과에 해당하는 집단, 특성, 이상치에 대한 C++ 클래스(class)를 정의하고, 이들 클래스의 STL 벡터(vector)를 유지한다. 예를 들어, 궤적에 대한 정보를 유지하기 위해 아래와 같은 클래스의 벡터를 유지한다. 각 궤적 별로 궤적을 구성하는 점과, 그 궤적에서 분할점으로 선택된 점들의 인덱스를 저장한다. 다른 클래스들도 이와 유사한 형태로 정의되어 있다.

```
class trajectory {
private:
    int mTrajectoryId;
    int mNumOfPoints;
    vector<Point> mTrajectoryPoints;
    int mNumOfPartitioningPoints;
    vector<int> mPartitioningPoints;
public:
    // constructor & member functions
};
```

궤적은 여러 개의 선분으로 분할된 후 처리되는데, 이 선분들을 저장하기 위해 아래와 같은 구조체(struct)의 STL 벡터를 유지한다. 이 데이타 구조를 유지함으로써 선분(시작점과 끝점, 즉 16 바이트) 대신에 벡터 내에서의 인덱스(정수, 즉 4 바이트)를 알고리즘 구현에 사용할 수 있다. 이로 인해, 메모리 사용량과 인수 전달에 소요되는 시간을 크게 줄일 수 있다. 반면, 선분의 시작점과 끝점을 알아내기 위해 이 데이타 구조를 참조해야 할 필요가 있는데, 이 데이타 구조는 최신 CPU의 L2 캐시에 충분히 로드될 수 있으므로 참조에 많은 시간이 소요되지 않는다. 경험적으로 볼 때, 분할점은 약 10% 비율로 선정되므로 선분의 개수는 데이타 점의 개수의 약 10%에 불과하다. 펜티엄4의 L2 캐시 크기는 1~2MB이므로, 선분의 개수가 5만~10만일 때에도 L2 캐시에 로드될 수 있다.

```
struct partition {
    Point startPoint;   // starting point
    Point endPoint;     // ending point
};
```

본 마이닝 툴은 약 4만 라인의 C++ 코드로 구성되어 있다. 개발 툴로는 Microsoft Visual Studio 2005를 사용하였으며, 결과 출력을 위해 MFC 클래스를 사용하였다.

3.2 실행 화면

구현된 툴을 운용하기 위해서는 최소 펜티엄3 CPU가 필요하고 펜티엄4 이상의 CPU에서 최적의 성능을 발휘한다. 처리 가능한 데이타 셋의 용량은 장착된 메인 메모리의 크기에 비례한다. 앞서 설명한 데이타 구조에서 각 점을 저장하기 위해 8 바이트가 필요하고, 각 분할점을 저장하기 위해 4 바이트가 필요하고, 각 선분을 저장하기 위해 16 바이트가 필요하다. 데이타 셋의 모든 점이 분할점으로 선택될 경우 분할점 및 선분의 개수는 최대 점의 개수까지 가능하다. 따라서, 최대 약 **(메모리 크기 (바이트) / 28)** 개의 점을 가진 데이타 셋을 처리할 수 있다. 예를 들어, 2GB의 메모리가 사용 가능하다면 최대 약 7천만 개의 점을 가진 데이타 셋을 처리할 수 있다.

본 마이닝 툴의 테스트를 위해 여러 가지 데이타 셋을 사용한다. 집단화, 분류, 이상치 발견에 사용된 데이타 셋의 텍스트 파일 크기는 각각 213KB, 238KB, 235KB 이다. 데이타 셋이 메모리 상에 올려 졌을 때 차지하는 메모리 용량은 점의 개수에 비례한다. 한 점을 나타내는데 X, Y 좌표 각각 4 바이트씩 총 8 바이트가 필요하므로, 데이타가 차지하는 메모리 용량은 대략 **(점의 개수 * 8) 바이트**가 된다. 여기에서 궤적 번호 및 각 궤적 별 점의 개수를 저장하는데 필요한 용량은 그다지 크지 않기 때문에 고려하지 않는다. 집단화, 분류, 이상치 발견에 사용된 데이타가 메모리 상에 올려 졌을 때 차지하는 크기는 앞의 수식에 따라 각각 142KB, 120KB, 161KB 이다.

그림 9는 궤적 집단화 실행 결과 화면이다. 사용된 데이타는 1950년부터 2004년까지 대서양에서 발생한 허리케인의 이동 경로이다. 총 570개의 궤적과 17736개의 점으로 구성되어 있다. 녹색 선은 허리케인의 이동 경로를 보여주며, 붉은 선은 집단화 결과인 공통 부분 궤적들을 보여준다. 일반적으로 대서양에서 발생한 허리케인은 동에서 서로 이동하다가, 상륙하거나 육지에 가까워지면 이동 방향을 남에서 북으로 변경하고, 다시 이동 방향을 서에서 동으로 변경하여 이동하다가 소멸하는

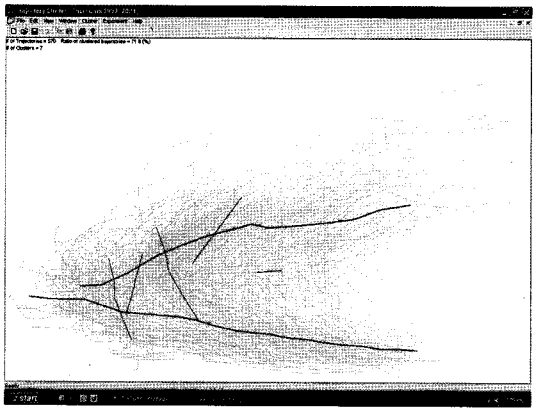


그림 9 궤적 집단화 실행 결과 화면

것으로 알려져 있다. 따라서, 그림 9의 결과는 매우 정확하다고 말할 수 있다.

다음으로 궤적 분류를 위해 사용된 데이타는 1995년 6월에 미국 Oregon 주에서 수집된 동물 이동 경로 데이타이다. 각 범주는 동물의 종을 나타내며, 엘크(elk), 사슴(deer), 소(cattle)의 3개의 범주로 구성되어 있다. 그림 10은 각 종 별로 동물의 이동 경로를 보여주고 있다. 각 종 별로 궤적(점)의 개수는 38(7117), 30(4333), 34(3540)이다.

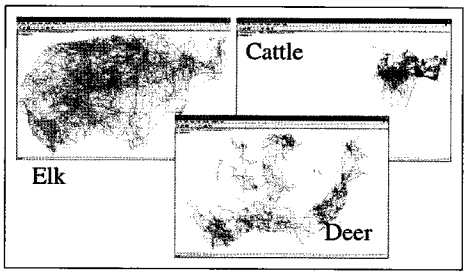


그림 10 궤적 분류에 사용된 데이타

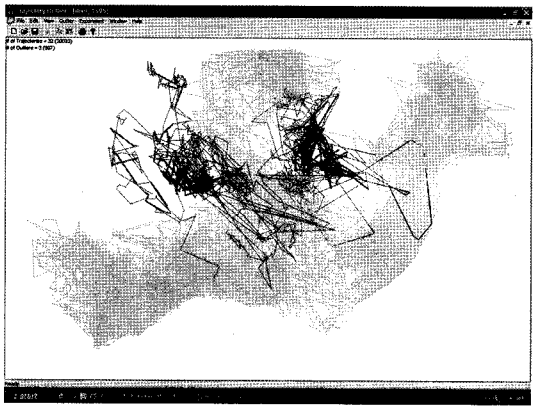


그림 11 궤적 분류 실행 결과 화면

그림 11은 궤적 분류의 실행 결과 화면이다. 회색 선은 동물의 이동 경로를 나타낸다. 우선 사각형은 지역 기반 특성을 나타내고, 굵은 선은 궤적 기반 특성을 나타낸다. 사각형과 굵은 선에서 붉은 색은 엘크, 파란 색은 사슴, 검은 색은 소를 의미한다. 예를 들어, 붉은 색 사각형은 그 지역 안에는 오로지 엘크 밖에 존재하지 않으므로, 어떤 동물의 이동 경로가 그 지역을 지나게 된다면 그 동물은 엘크일 가능성이 높다는 것을 의미한다. 또한, 붉은 색 굵은 선은 엘크들이 공통적으로 지나간 부분 궤적이므로, 어떤 동물의 이동 경로가 이와 겹치게 된다면 그 동물은 엘크일 가능성이 높다는 것을 의미한다. 이와 같이 추출된 두 가지의 특성(feature)들을 사용하여 범주를 알지 못하는 궤적들을 분류해 낼 수 있다. 그림 10의 데이타와 비교해 보면, 그림 11의 특성들은 매우 정확하게 각 범주를 나타낸다고 말할 수 있다. 임의로 20%의 궤적을 골라서 범주를 알지 못한다고 가정하고 분류를 통해 범주를 예측해 보았을 때, 83.3%의 높은 정확도를 보였다.

그림 12는 궤적 이상치 발견 실행 결과 화면이다. 사용된 데이타는 1995년에 미국 Oregon 주에서 수집된 사슴의 이동 경로 데이타이다. 총 32개의 궤적과 20065개의 점으로 구성되어 있다. 녹색 선은 사슴의 이동 경로를 보여주며, 굵은 붉은 선은 이상 선분(outlying partition)들을 보여준다. 이상 선분을 특정 비율 이상 포함하고 있다면 전체 궤적이 가는 붉은 선으로 표시된다. 대부분의 사슴들은 그림 12에서 왼쪽 아래와 오른쪽의 지역에서 움직인 반면, 오직 3마리의 사슴들은 화면의 중앙 지역에서 움직였다. 이 3마리의 사슴의 움직임이 다른 사슴들과 크게 다르게 때문에 이상치로 검출되었다. 따라서, 그림 12의 결과는 매우 정확하다고 말할 수 있다.

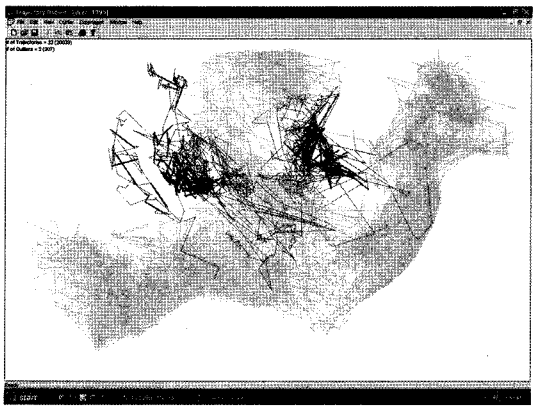


그림 12 궤적 이상치 발견 실행 결과 화면

4. 기존 집단화(clustering) 알고리즘 개선

제 2.2절에서 설명한 부분 궤적 집단은 객체들의 이동 시점을 고려하지 않았기 때문에, 응용에 따라서는 그다지 의미 없는 결과가 산출될 수 있다. 예를 들어, 동물 이동 데이타에서 사슴이 여름에 이동한 경로와 겨울에 이동한 경로가 공통적일 경우, 이러한 정보는 이동 시점의 차이가 크기 때문에 그다지 큰 의미가 없을 수도 있다. 이러한 제약 사항을 없애기 위해 **시간 부분 궤적 집단**(*temporal sub-trajectory cluster*)이라는 개념을 제안한다.

시간 부분 궤적 집단 발견의 핵심은 집단화를 두 단계에 걸쳐 하는 것이다. 첫 번째 단계에서는 제 2.2절과 같이 시간을 전혀 고려하지 않고 위치 정보만을 사용하여 집단화를 수행한다. 여기에서 얻어진 집단 별로 다시 선분 집단화 알고리즘을 적용한다. 이 두 번째 단계에서야 시간을 고려한다. 첫 번째 단계에서는 X-Y 좌표로 구성된 2차원 평면에서 집단화를 수행한 반면, 두 번째 단계에서는 위치-시간 좌표로 구성된 2차원 평면에서 집단화를 수행한다. 주목할 점은 모델의 단순화를 위해 선분의 위치를 오직 1차원 범위(range)로 나타낸다는 점이다. 각 집단 내에서는 선분의 이동 방향(각도)이 거의 동일하므로, 평균 벡터에 선분의 시작점과 끝점을 투영(project)함으로써 거의 오차 없이 선분을 1차원으로 나타낼 수 있다. 그림 13은 이 발견 과정을 요약하여 도시한다.

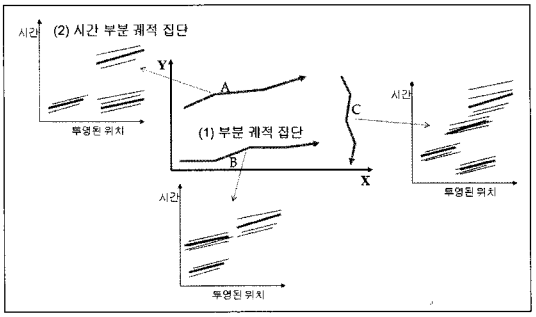


그림 13 시간 부분 궤적 집단 발견의 두 단계

이러한 2단계 집단화의 장점은 간단하면서도 매우 효과적이라는 것이다. X, Y 좌표에 시간까지 고려하게 되면 일반적으로 3차원이 되므로, 집단화 비용과 복잡도가 크게 증가한다. 반면, 제안된 방법에서는 오직 2차원 상에서만 집단화를 수행한다. 두 번째 단계에는 첫 번째 단계에 사용한 선분 집단화 알고리즘과 동일한 알고리즘을 사용할 수도 있고, 다른 특화된 알고리즘을 사용할 수도 있다. 두 번째 단계에서 집단화는 각 집단 별로 수행되기 때문에, 선분의 개수가 적어 매우 빠르게 수행될 수 있다.

그림 14는 시간 부분 궤적 집단 발견의 예시를 보인다. 그림 12의 데이타에 이동 시점을 고려하지 않고 집단화를 수행할 경우 C_1~C_4의 4개의 집단을 얻을 수 있다. C_1에 대해 이동 시점을 고려하여 집단화를 수행할 경우, 그림 14의 왼쪽과 같이 10개의 시간 부분 궤적 집단을 얻을 수 있다. 이들 시간 부분 궤적 집단은 비슷한 경로를 비슷한 시간에 지나간 객체들의 집단임을 쉽게 알 수 있다. 이들 중에서 SP_5와 SP_6을 C_1과 함께 표시하면 그림 14의 오른쪽과 같다.

5. 관련 연구

비교적 많은 궤적 데이타 마이닝 알고리즘들이 발표되어 있는 반면, 실제적으로 데이타 마이닝 툴은 거의

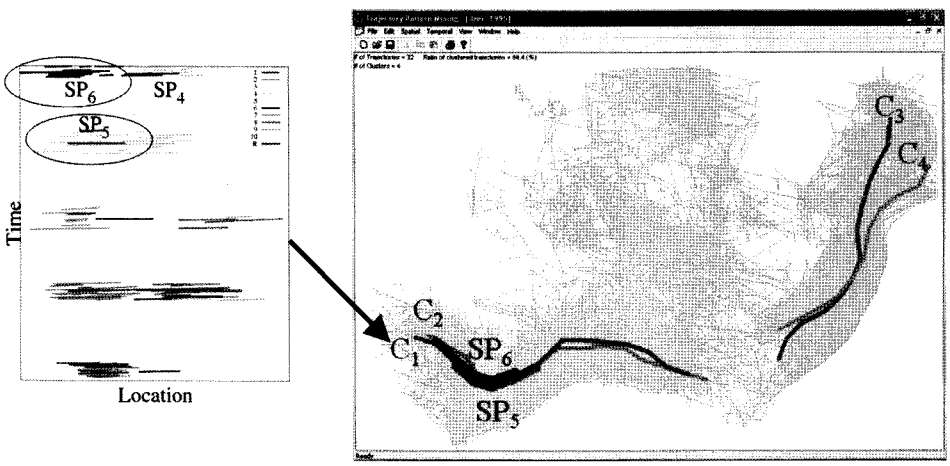


그림 14 시간 부분 궤적 집단의 발견 예

표 1 T-Pattern Miner와 본 마이닝 툴의 비교 요약

	T-Pattern Miner	본 마이닝 툴
응용 분야	방문한 지역들의 패턴을 알아내야하는 응용 (예: 관광지에서 관광객들의 주요 방문 **패턴**, A->B->C)	일반적인 궤적 데이타 분석 응용 (예: 동물 서식지에서 동물의 공통 이동 **경로**)
구현 알고리즘	궤적 패턴 마이닝[10]	집단화[3], 분류[6], 이상치 검출[7]
인터페이스	텍스트 인터페이스 (명령행에서 실행)	그래픽 인터페이스
성능(실행 시간)	1000개의 궤적에 약 5초	집단화의 경우, 약 600개의 궤적에 약 2초
마이닝 정확도	높음	높음
기타 특징	중간 이동 경로와 이동 시점이 고려되지 않음	중간 이동 경로와 이동 시점이 고려됨 (이동 시점 고려는 옵션임)

개발된 바가 없다. 이는 궤적 데이타 마이닝 툴의 개발이 시급하다는 것을 방증한다. 2007년에 Giannotti 등[10]이 개발하여 발표한 공개 소프트웨어인 T-Pattern Miner가 가장 널리 알려져 있으며, 이 소프트웨어는 궤적 데이타에서 **궤적 패턴**(*trajectory pattern*)을 찾아준다. 그림 15에서 궤적 패턴 A-t_1->B'-t_2->B''은 충분히 많은 이동 객체가 A 지역에서 t_1 시간에 B' 지역으로 이동하고 다시 t_2 시간에 B'' 지역으로 이동한다는 사실을 나타낸다.

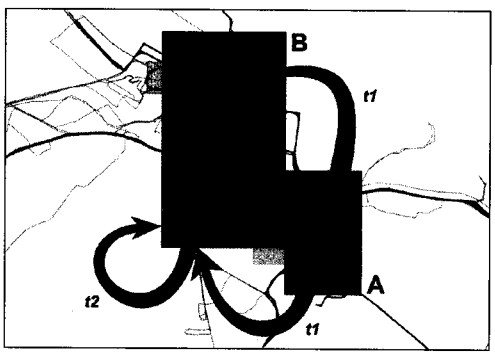


그림 15 궤적 패턴의 예시

좀 더 정형적으로 궤적 패턴은 (S, a)의 쌍으로서 구성된다. S는 k+1개의 위치의 연속으로서 $<(x_0,y_0), (x_1,y_1), ..., (x_k,y_k)>$로 정의된다. a는 연속된 두 위치 사이의 이동 시간의 연속으로서 $<a_1, a_2, ..., a_k>$로 정의된다. 궤적 패턴은 제 2.2절에서 설명한 부분 궤적 집단과 두 가지 큰 차이를 가지고 있다. 첫째, 궤적 패턴에서는 중간 이동 경로가 고려되지 않지만, 부분 궤적 집단에서는 고려된다. 즉, 궤적 패턴에서는 위치 A와 B를 방문했다는 사실이 중요하지, 이 두 위치 사이에 어떻게 이동했는지는 중요하지 않다. 둘째, 궤적 패턴에서는 실제 이동 시간이 고려되지 않지만, 부분 궤적 집단에서는 고려된다. 즉, 궤적 패턴에서는 위치 A에서 B로 이동하는데 t시간이 걸렸다는 사실이 중요하지, 이 두 위치를 언제 이동했는지는 중요하지 않다. 이렇듯 궤적 패턴과 부분 궤적 집단은 서로 다른 정의에서 출발하며, 각자 보다 더 적합한 응용 분야를 가지고 있다.

T-Pattern Miner와 본 마이닝 툴은 서로 다른 응용 분야를 가지고 개발되었기 때문에 직접적인 비교는 어렵지만, 이들의 차이점을 요약하면 표 1과 같다.

6. 결 론

본 논문에서는 대규모 궤적 데이타를 위한 데이타 마이닝 툴을 개발하였다. 집단화, 분류, 이상치 검출의 데이타 마이닝에서 가장 널리 활용되는 세 가지 연산을 제공한다. 본 마이닝 툴의 가장 큰 특징은 Lee 등[3,6,7]이 개발한 알고리즘에 기반하여 부분 궤적 정보를 활용한 마이닝 결과를 제공한다는 점이다. 다양한 실제 궤적 데이타를 사용하여 본 마이닝 툴의 실용성을 검토하였다. 태풍 경로 데이타와 동물 이동 데이타를 사용하여 얻어낸 마이닝 결과가 직관적이고 타당함을 입증하였다. 다양한 대규모의 궤적 데이타가 늘어나고 이러한 데이타에 대한 분석의 중요성이 점차 증대되고 있는 상황에서, 본 궤적 마이닝 툴의 개발은 매우 시의적절하다고 사료된다. 현재 본 마이닝 툴의 집단화 패키지는 미국 NSF의 Move Bank 프로젝트[2]의 일환으로 웹 서비스(web service)를 통해 동물학자 및 일반 사용자들에게 공개될 예정이다[11]. 이러한 실용화 예정 실적은 본 마이닝 툴의 우수성을 직접적으로 입증하는 것이다. 향후 연구로서 궤적 패턴 발견과 같은 보다 다양한 마이닝 연산을 추가 구현함으로써, 본 마이닝 툴의 활용 범위를 한층 넓혀갈 계획이다.

참 고 문 헌

[1] Han, J., Lee, J.-G., Gonzalez, H., and Li, X., "Mining Massive RFID, Trajectory, and Traffic Data Sets," In *Tutorial 14th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*,

2) http://www.movebank.org/

Las Vegas, Nevada, Aug. 2008.
[2] Han, J., Lee, J.-G., and Kamber, M., "An Overview of Clustering Methods in Geographic Data Analysis," In Miller, H. J. and Han, J., eds., *Geographic Data Mining and Knowledge Discovery*, 2nd ed., Chapman and Hall/CRC Press, 2008 (in press).
[3] Lee, J.-G., Han, J., and Whang, K.-Y., "Trajectory Clustering: A Partition-and-Group Framework," In *Proc. 2007 ACM SIGMOD Int'l Conf. on Management of Data*, pp. 593-604, Beijing, China, June 2007.
[4] Li, X., Han, J., Lee, J.-G., and Gonzalez, H., "Traffic Density-based Discovery of Hot Routes in Road Networks," In *Proc. 10th Int'l Symp. on Spatial and Temporal Databases*, pp. 441-459, Boston, Massachusetts, July 2007.
[5] Jeung, H., Yiu, M. L., Zhou, X., Jensen, C. S., and Shen, H. T., "Discovery of Convoys in Trajectory Databases," In *Proc. the VLDB Endowment (PVLDB)*, Vol.1, No.1, pp. 1068-1080, Aug. 2008.
[6] Lee, J.-G., Han, J., Li, X., and Gonzalez, H., "TraClass: Trajectory Classification Using Hierarchical Region-Based and Trajectory-Based Clustering," In *Proc. the VLDB Endowment (PVLDB)*, Vol.1, No.1, pp. 1081-1094, Aug. 2008.
[7] Lee, J., Han, J., and Li, X., "Trajectory Outlier Detection: A Partition-and-Detect Framework," In *Proc. 24th Int'l Conf. on Data Engineering*, pp. 140-149, Cancun, Mexico, Apr. 2008.
[8] Li, X., Li, Z., Han, J., and Lee, J.-G., "Temporal Outlier Detection in Vehicle Traffic Data," In *Proc. 25th Int'l Conf. on Data Engineering*, Shanghai, China, Mar. 2009 (to be published).
[9] Li, X., Han, J., Kim, S., and Gonzalez, H., "ROAM: Rule- and Motif-Based Anomaly Detection in Massive Moving Object Data Sets," In *Proc. 7th SIAM Int'l Conf. on Data Mining*, Minneapolis, Minnesota, Apr. 2007.
[10] Giannotti, F., Nanni, M., Pinelli, F., and Pedreschi, D., "Trajectory Pattern Mining," In *Proc. 13th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pp. 330-339, San Jose, California, Aug. 2007.
[11] Movebank team, Movebank Update #1, http://www.movebank.org/register/Movebank%20Update%20#1.pdf, May 2008.

이 재 길
2006년 7월~현재 어바나-샴페인 일리노이 주립대학 박사후연구원. 2005년 3월~2006년 6월 한국과학기술원 BK21 박사후연구원. 1999년 3월~2005년 2월 한국과학기술원 전자전산학과 전산학전공 박사. 1997년 3월~1999년 2월 한국과학기술원 전산학과 석사. 1993년 3월~1997년 2월 한국과학기술원 전산학과 학사. 관심분야는 시공간 데이타 마이닝, 객체 관계형 데이타베이스 시스템, 정보 검색