궤적 스트림 데이터로부터 동행 그룹 탐색 기법

강수현, 이기용
숙명여자대학교 컴퓨터과학과
e-mail:{suhyunkang, kiyonglee,}@sookmyung.ac.kr

A Method for Finding Accompany Group from Trajectory Stream Data

Suhyun kang, Ki Yong Lee
Department of Computer Science, Sookmyung Women's University

요 약

객체들의 움직임의 흐름을 나타내는 궤적 데이터에서 함께 움직이는 궤적을 찾아 움직임 패턴을 탐색하는 연구들이 많이 이루어져 왔다. 하지만, 궤적 스트림 데이터에서 궤적의 이동 패턴을 탐색하는 연구는 많이 이루어지지 않았다. 그래서 본 논문은 시간의 흐름에 따라 흘러 들어오는 궤적 스트림 데이터에서 궤적의 이동 패턴을 탐색하여 동행 그룹을 탐색하는 새로운 방법을 제안한다. 기존에도 궤적 스트림 데이터에서 궤적들이 주어졌을 때 궤적들의 이동 패턴을 찾는 연구들은 존재하였으나 발견된 궤적이 언제 생성되었고 언제 소멸되었는지에 대한 정보를 자동으로 출력해 주는 연구는 아직 이루어진 바가 없다. 본 논문에서는 서로 다른 시간에 나타나고 사라지는 모습을 가진 궤적 스트림 데이터에서 동일한 시간에 나타나는 궤적을 찾는 방법을 제안한다. 제안 방법은 객체들의 좌표를 점진적으로 클러스터링을 수행하여 사용자에게 입력받은 지속 시간 이상 클러스터를 유지한 동행 그룹의 객체들을 반환한다. 또한, 기존 연구와 달리 해당 객체들의 지속 시간인 시작과 끝 시간도 자동으로 출력해 준다.

1. 서론

최근 들어 실시간으로 데이터를 수집할 수 있는 위성, 위치 추적과 같은 기기들을 통해 여러 객체들의 움직임을 실시간으로 수집하여 분석하는 일이 많아지고 있다. 이러한 기기들을 통해 수집되는 데이터는 시간과 공간의 정보를 가지고 있으며 계속해서 흘러들어 오는 궤적 스트림 데이터라고 한다.

궤적 스트림 데이터를 사용한 다양한 연구들이 이미 많이 이루어진 바가 있다. 대표적인 예로 여러 객체들의 클러스터링을 통한 움직임의 패턴을 탐색하는 연구들을 많이 볼 수 있다. 예로 들어 궤적 스트림 데이터를 통해 실시간 교통의 혼잡도를 파악하거나 여행 동반자를 찾는 등과 같은 의미 있는 결과를 도출해 낼 수 있다. 하지만 지금까지 궤적 스트림 데이터에서 탐색된 궤적들의 생성과 소멸 시간을 탐색하는 연구는 이루어진 바가 없다.

따라서 본 논문은 이러한 궤적 스트림 데이터로부터 동일한 시간에 나타나는 동행 그룹을 탐색함과 동시에 동행 그룹이 언제 생성되었고 언제 소멸되었는지에 대한 정보를 자동으로 출력해주는 기법을 제안한다. 본 논문의 제안 방법은 움직임을 나타내는 궤적 스트림 데이터가 주어졌을 때, 객체들을 일정 시간 간격으로 클러스터링을 수행한다. 이때 클러스터링은 매 시간 간격 마다 수행하지 않고 최초의 시간 간격에서 수행한 클러스터의 수행 결과를 가지고 다음 시간 간격부터는 점진적으로 클러스터링을 수행해 나간다. 이는 모든 시간 간격마다 전체의 객체들을 클러스터링 하는 데에 생기는 수행 시간을 줄이기 위함이다. 궤적 스트림 데이터는 사람, 동물, 자동차의 움직임으로 생기는 데이터이기 때문에 한순간에 좌표가 급변할 수 없고 점진적으로 좌표가 변할 것이라고 판단된다. 이러한 궤적 스트림 데이터의 특징을 감안하여 본 논문에서는 첫 번째 시간 간격에서 클러스터링을 수행하여 클러스터의 반경이 일정 크기 이하이고 클러스터 내의 객체 수가 일정 수 이상인 동시에 이 클러스터가 일정 시간동안 유지되었다면 해당 클러스터의 생성 시간과 소멸 시간을 함께 자동으로 반환해준다. 제안 방법의 세부 설명은 3장에서 이어진다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구들을 살펴보고, 3장에서는 제안 방법을 자세 설명한 뒤, 4장에서는 제안 방법의 실험 결과를 보이며, 5장에서는 결론을 맺는다.

2. 관련 연구

궤적의 이동 패턴을 찾는 연구는 크게 정적 데이터를 사용한 연구와 스트림 데이터를 사용한 연구로 나눌 수 있다.

정적 데이터에서 유사한 이동 패턴을 가지는 궤적을

찾는 연구는 굉장히 많이 이루어져 왔다. [1]은 특정 시간 내에 두 개의 궤적 사이에 유사한 패턴을 탐색한다. [2][3]은 궤적의 전체를 보고 전체 궤적의 패턴이 유사한 궤적을 클러스터 해준다. 하지만 궤적들은 각 객체 마다 다양한 움직임 패턴을 가지기 때문에 전체 궤적이 유사한 궤적을 찾는 다는 것은 한계가 있다. 그래서 궤적의 전체를 보고 이동 패턴을 탐색하지 않고 부분적으로 유사한 궤적을 탐색하는 연구들도 이루어져 왔다. [4]는 궤적을 선분으로 끊어주고 그 선분들을 통해 유사한 선분을 그룹화 시켜 유사한 부분 궤적을 탐색한다. [5]는 연속적인 시간에서 특정 반경을 가지는 원에 속하는 객체들이 사용자에게 받은 지속 시간동안 무리가 특정 반경의 원 내에서 유지되는지 탐색한다. 하지만 특정 원으로 무리의 지속 여부를 탐색하게 되면 여전히 함께 움직이는 객체가 조금만 원을 벗어나게 되면 이 객체를 놓칠 수 있다는 문제가 있다.

스트림 데이터에서 궤적의 이동 패턴을 찾는 연구는 단순히 유사한 패턴을 이루는 궤적을 찾아 출력해주는 연구와 함께 함께 움직이는 객체들을 출력해주는 두 가지의 출력 형태로 분류할 수 있다.

앞서 설명한 바와 같이 궤적 스트림 데이터에서 유사한 패턴을 이루는 궤적을 출력해 주는 기존 연구들이 있다. [6]는 움직이는 객체들의 궤적 선분을 서로 가까운 선분끼리 소그룹으로 그룹화를 시켜준다. 그리고 새로운 궤적 선분이 들어오면 그 선분을 가장 유사한 소그룹으로 삽입하여 새로운 선분이 추가된 전체 궤적을 반환해 준다.

그리고 궤적 스트림 데이터에서 궤적의 클러스터를 출력해주는 기존 연구들 이외에 같은 움직임 패턴을 보이는 특정 객체들을 출력해 주는 연구들이 있다. [7]은 연속적인 시간 간격 마다 유사한 패턴을 가지는 궤적을 전부 확인하는 것이 아니라 객체들을 버디(buddy)라는 작은 그룹들로 정의하여 이 버디를 매 시간 간격에서 여전히 함께 있는지 확인하고, 일정 시간을 함께 돌아다닌 버디를 반환해준다. [7]은 본 논문과 유사한 연구이지만 이 버디 각각을 매 시간 간격에서 클러스터링 하는 방법 또한 많은 수행시간이 필요할 것으로 판단되고 궤적의 시작과 끝 시간 또한 출력해주지 못한다. 즉, 궤적 스트림 데이터에서 유사한 이동 패턴을 갖는 궤적을 탐색함과 동시에 시작과 끝 시간을 찾는 연구는 아직까지 없는 것으로 파악된다.

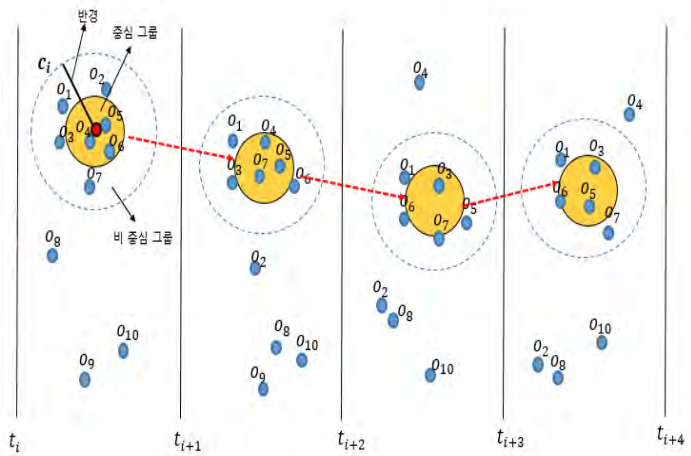
3. 제안 방법

본 장에서는 시간의 흐름에 따라 움직이는 객체들로 구성된 궤적 스트림 데이터를 일정한 시간 간격에서 클러스터링을 수행하여 그 결과 일정 시간 이상 함께 다닌 객체들을 찾는 제안 방법을 상세히 설명한다. 궤적 스트림 데이터는 그림 1과 같이 각 객체의 고유 번호, 시간, 위도, 그리고 경도 좌표를 의미한다. 본 논문에서 제안하는 기법은 (1) 초기 클러스터링 수행, (2) 점진적 클러스터링 수행, (3) 유사한 이동 패턴을 갖는 객체 탐지의 3단계로 구성된다. 아래에서는 3단계를 순서대로 설명한다.

$$
\begin{gathered}
(object_ID_1, time_1, x_1, y_1) \\
(object_ID_2, time_2, x_2, y_2) \\
(object_ID_3, time_3, x_3, y_3) \\
(object_ID_4, time_4, x_4, y_4) \\
(object_ID_5, time_5, x_5, y_5) \\
\vdots \\
(object_ID_n, time_n, x_n, y_n)
\end{gathered}
$$

(그림 1) 궤적 스트림 데이터

3.1 초기 클러스터링 수행



(그림 2) 클러스터 수행 과정

제안 방법은 이동 패턴을 탐색하기 위해 우선 일정 시간 간격마다 같이 움직이는 객체를 확인해야 한다. 우선 그림 2와 같이 사용자에게 시간 간격 파라미터를 입력 받아 궤적 스트림 데이터를 일정 시간 간격으로 분할하여 클러스터링을 수행한다. 본 논문은 이때 모든 간격에서 클러스터링을 수행하는 것이 아니라 그림 2의 가장 첫 번째 시간 간격에서만 클러스터를 수행한다. 클러스터의 경우, 객체의 수가 사용자에게 받은 일정 임계값 T_{size} 이상이여야 하며 클러스터의 반경 또한 일정 임계값 T_{radius} 이하를 만족해야 한다. 본 논문에서는 클러스터링을 수행하기 위해 K-평균 클러스터링(k-means clustering)을 사용하였다. 또한 거리 측정 방식(distance measure)은 유클리디안(euclidean distance)을 사용하였다.

이렇게 첫 번째 간격에서 클러스터링이 수행되고 나면 생성된 클러스터에 궤적 정보를 저장한다. 그림 2의 첫 번째 시간 간격에서 생성된 클러스터의 집합은 C이고, C에 속한 클러스터가 c_i이다. 궤적 정보는 C에 속한 c_i로 설명한다. 궤적 정보는 반경(radius), 중심 그룹(core group), 비 중심 그룹(remainder group), 지속 시간(duration), 시작 시간, 끝 시간으로 이루어져 있다. 반경은 클러스터의 중심점을 기준으로 구한다. 이렇게 반경이 주어지면 그림 2에서 볼 수 있듯이 반경의 일정 비율을 차지하는 노란색 부분 내에 속하는 객체들을 중심 그룹이라고 한다. 그리고 노란색 부분 외에 나머지를 비 중심 그룹이라고 한다. 지

속 시간은 c_i는 t_i, t_{i+1} 시간 간격 사이에 존재하는 클러스터이기 때문에 각각 시작 시간과 끝 시간은 t_i, t_{i+1}으로 저장 된다. 이렇게 저장된 궤적 정보는 다음 단계인 점진적 클러스터링 수행에서 사용된다.

3.2. 점진적 클러스터링 수행

앞서 설명한 바와 같이 초기 클러스터링 결과를 가지고 점진적으로 클러스터링을 수행한다. 점진적인 클러스터링을 수행하는 이유는 매 시간 간격마다 클러스터링을 수행하여 생기는 시간을 줄이기 위함이다. 그래서 본 논문에서는 효율적으로 점진적인 클러스터링을 수행하기 위하여 첫 번째 시간 간격의 클러스터들의 궤적 정보를 이용하여 다음 시간 간격부터는 클러스터를 확인할 때 클러스터 전체를 확인하는 것이 아니라 노란 부분인 클러스터의 중심 그룹만을 확인한다. 이렇게 중심 그룹만을 확인하는 것은 클러스터를 확인하는데 걸리는 수행시간을 좀 더 줄이기 위함이다.

앞서 설명한 것처럼 궤적 정보에 기록된 중심 그룹을 기반으로 다음 시간 간격에서도 여전히 클러스터가 함께 유지되고 있는지 아니면 소멸되었는지 확인한다. 이는 중심 그룹의 객체들이 다음 시간 간격에서도 여전히 존재하는지 확인했을 때 그 비율이 일정 임계값 $T_{maintenance}$ 이상이면 여전히 유지되고 있다고 판단한다. 만약 객체들이 여전히 존재 한다면 그 객체들의 새로운 중심점을 찾고 이 중심점과 가장 멀리 떨어져 있는 객체와의 거리가 사용자에게 받은 거리 $T_{distance}$ 임계값 이하이면 이 객체들이 여전히 유지되고 있을 뿐만 아니라 여전히 가까이 유지되고 있다고 판단한다. 이렇게 클러스터가 함께 유지되고 있다고 판단되면 기존에 저장한 궤적 정보를 업데이트 해준다. 먼저, 중심 그룹은 궤적 정보에 저장된 중심 그룹의 객체와 현재 시간 간격에서 발견된 중심 그룹의 객체들의 새로운 중심점을 기반으로 반경의 일정 비율 내에 속하는 객체들의 교집합을 궤적 정보에 업데이트 해준다. 비 중심 그룹 또한 궤적 정보에 저장된 비 중심 그룹과 새로운 중심점을 기반으로 반경 내에 속하는 객체들의 교집합을 궤적 정보를 업데이트 해준다. 그리고 끝 시간 또한, 클러스터가 함께 유지되었기 때문에 업데이트 해준다.

반면에 새로운 중심점과 가장 멀리 떨어져 있는 중심 객체와의 거리가 사용자에게 받은 거리 임계값 $T_{distance}$이상이면 객체들이 가까이 머물러있지 않고 멀리 퍼져 분리되었다고 판단한다. 분리되었다고 판단되면 중심 그룹에 속한 객체들을 각각 하나의 기준으로 보고 반경 일정 비율 이내에 객체 수를 확인한다. 이 객체 수가 사용자에 받은 일정 임계값 T_{split} 이상이면 중심 그룹이 작은 그룹으로 쪼개어 졌다고 판단한다. 만약 모든 객체의 반경 비율 이내에 객체 수가 일정 임계값 T_{split} 이하이면 중심 그룹이 소멸되었다고 판단한다.

마지막으로 모든 확인이 끝났을 때, 시간 간격에 존재하는 객체들 중에 궤적 정보의 중심 그룹, 비 중심 그룹에 속하지 못한 객체의 수가 일정 임계값 T_{count}이상이면 이 객체들만을 가지고 클러스터링을 수행한다. 이러한 수행을 모든 시간 간격에서 반복한다.

3.3 유사한 이동 패턴을 갖는 객체 탐지

클러스터의 점진적 클러스터링을 통해 지속 여부 판단이 완료되면 마지막으로 함께 이동하는 그룹의 시작 시간과 끝 시간을 출력해 주는 단계이다. 궤적 정보에서 지속 시간이 사용자에게 받은 일정 임계값 $T_{duration}$이상이면 이

```
Input: size threshold, radius threshold, duration threshold,
split threshold, count threshold, maintenance threshold
Output: sets of accompany group, start_time, end_time
T= trajectory information
maintenance objects = the number of existing objects of
object group in next interval
object group = 'objects of core group of trajectory
information
distance = distance between new centroid of core group and
farthest  away object
split count = the number of objects that belong to radius to
the object
count object = the number of objects not in trajectory
  information

//Step 1: initial clustering
initial interval → clustering
   for  c_i in C
     if size of c_i > T_size
       if radius of c_i < T_radius
         T ← T of c_i

// Step 2: incremental clustering
for each interval
  for each T
    if maintenance objects/the number of object group ≥T_maintenance
      if distance  > T_distance
        T′ ← T update
      else        //split
        for each objects of core group
          if radius count > T_split
            T′ ← T update
          else            // remove
      else            //remove
      if count object > T_count
        → Step 1
          T <- T'

//Step 3: accompany group detection
for each T
  duration ≥ T_duration
   output core group, remainder group, start_time, end_time
```

(그림 3) 제안 방법의 의사코드

때, 중심 그룹, 비 중심 그룹, 시작 시간, 끝 시간을 자동으로 반환함으로써 해당 클러스터가 언제부터 언제까지 지속되었는지를 알려준다. 그림 3은 제안 방법의 의사코드이다. Step1은 제안 방법의 첫 번째 단계인 초기 클러스터링, Step2는 점진적 클러스터링, Step3은 유사한 이동 패턴을 갖는 객체 탐지를 나타낸다.

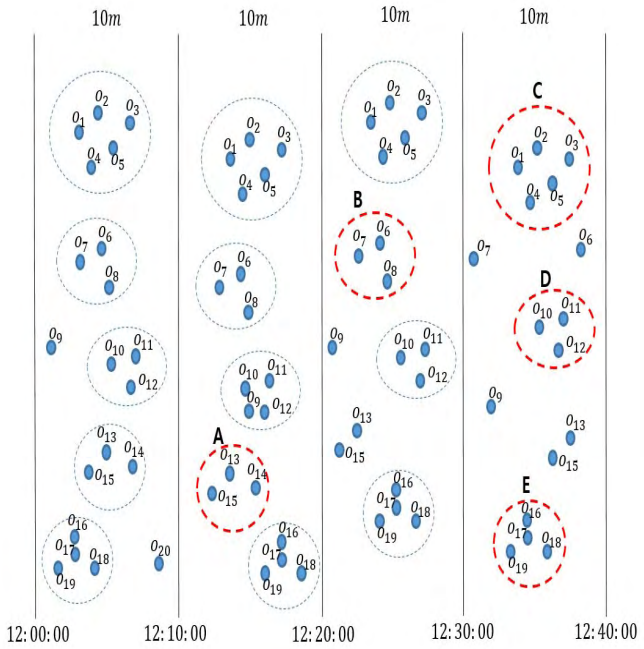
4. 실험 결과

본 장에서는 제안하는 궤적 탐색 방법에 대한 정확성 실험 결과를 보인다. 실험은 가상 궤적 스트림 데이터를 생성하여 사용하였다.

4.1 정확성 실험

실험은 제안 방법이 유사한 이동 패턴을 가지는 궤적들의 시작 시간과 끝 시간을 탐색할 수 있는지를 평가했다. 이 실험을 위해 그림 5와 같은 가상 궤적 스트림 데이터를 사용하였다.

그림 4에서 볼 수 있듯이 가상 데이터는 1~20개의 객체가 시간에 따라 어떻게 움직였는지를 나타내는 궤적 스트림 데이터이다. 각 객체는 생성될 수도 있고 소멸할 수도 있다. 제안 방법은 시간 간격을 10분으로 하여 T_{size} = 3, $T_{duration}$ = 20, T_{radius} = 3일 때, 그림 4에서 볼 수 있듯이 A는 12:00:00~12:20:00, B는 12:00:00~12:30:00, C는 12:00:00~12:40:00, D는 12:00:00~12:40:00, E는 12:00~12:40으로 5개의 궤적 모두 그룹에 속한 객체, 그룹의 시작 시간과 끝 시간을 올바르게 출력함을 확인했다. 즉, 해당 실험을 통해 궤적을 넣었을 경우 동행 그룹과 동행 그룹의 시작, 끝 시간을 100% 올바르게 출력해낼 수 있음을 확인하였다.



(그림 4) 실험 데이터

4. 결론

본 논문에서는 궤적 스트림 데이터에서 객체들의 이동 패턴을 파악하고 동행 그룹을 탐색하는 방법을 제안하였다. 제안 방법은 기존 연구와 달리 특정 동행 그룹의 객체들이 언제 생성되었고 언제 소멸되었는지 까지 자동으로 출력해 준다. 실험 결과 제안 방법은 일정 시간 동안 함께 궤적을 이루는 궤적을 파악했을 뿐만 아니라 시작 시간과 끝 시간을 정확하게 출력할 수 있음을 확인했다.

Acknowledgement

이 논문은 2019년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (No.2018R1D1A1B07045643).

참고문헌

[1]Bakalov, P., Hadjieleftheriou, M., Tsotras, V.J. “Time relaxed spatiotemporal trajectory joins”, Proceedings of the 13th annual ACM international workshop on Geographic information systems, November 04-05, 2005.

[2] Gudmundsson J, van KreveldM, “Computing longest duration flocks in trajectory data”, Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems, November 10-11, 2006.

[3] P. Laube, S. Imfeld, R. Weibel. “Discovering relative motion patterns in groups of moving point objects”, in: International Journal of Geographic Information Science, vol. 19, Taylor & Francis Group, July 2005.

[4] Jae-Gil Lee , Jiawei Han , Kyu-Young Whang, Trajectory clustering: a partition-and-group framework, Proceedings of the 2007 ACM SIGMOD international conference on Management of data, June 11-14, 2007.

[5] Hoyoung Jeung , Man Lung Yiu , Xiaofang Zhou , Christian S. Jensen , Heng Tao Shen, “Discovery of convoys in trajectory databases”, Proceedings of the VLDB Endowment, v.1 n.1, August 2008.

[6] Z. Li, J. Lee, X. Li, J. Han, "Incremental clustering for trajectories", Proc. Database Systems Advanced Applications, pp. 32-46, 2010.

[7] L. Tang et al., "On discovery of traveling companions from streaming trajectories", Proc. ICDE, pp. 186-197, 2012.