분산 네트워크 환경에서 POI추출을 위한 GPS 데이터 분할 방법

오주성*, 허유경*, 박진관*, 백종상**, 정민아*
*목포대학교 컴퓨터공학과
**목포대학교 정보전자공학과
e-mail:ojoos@mokpo.ac.kr

GPS Data Partitioning Method for POI Extraction in Distributed Environment

Joo-Seong Oh*, Yu-Kyung Heo*, Jin-Gwan Park*, Jong-Sang Back**, Min-A Jung*
*Dept of Computer Engineering, Mokpo University
**Dept of Information & Electronic Engineering, Mok-po University

요 약

많은 사람들이 위치 기반 서비스를 사용하면서 위치 기반 서비스에서 사용되는 GPS 데이터는 기하급수적으로 증가하고 있다. 사용자들에게 필요한 정보를 제공하기위해서는 이러한 대량의 GPS 데이터를 처리하여 POI를 추출하고 분석하는 과정이 필요하다. 본 논문에서는 POI를 추출하고 관리·분석하기 위해 MapReduce 환경을 구축하고 DBSCAN 클러스터링 방법을 이용한다. 또한 분산 환경에서 DBSCAN 알고리즘을 수행하기 위해 K-Means를 이용한 데이터 분할 방법을 제안한다.

1. 서론

최근 스마트폰, 태블릿PC 등 모바일 기기들의 사용이 증가함에 따라 목적지까지 가는 길을 검색하거나 카페, 서점 등의 위치를 검색하는 등 위치 기반 서비스(LBS:Location-Based Service)의 사용이 활발해 지고 있다[1,2].

위치 기반 서비스는 사용자가 원하는 위치에 대해 사용자가 필요로 하는 정보를 제공하는 것을 목표로 한다. 이에 따라 사용자들에게 필요한 정보를 제공하기 위해 많은 사람들의 POI(Points Of Interest)를 추출하고 관리·분석하는 것이 필요하다.

본 논문에서는 POI를 추출하기 위한 방법 중 하나인 DBSCAN 클러스터링을 MapReduce에서 사용하기 위해 K-Means 클러스터링을 이용한 데이터를 분할 방법을 제안한다.

2. 관련연구

1) DBSCAN

DBSCAN(Density-Based Spatial Clustering of Applications with Noise)은 많은 어플리케이션에 폭넓게 적용되는 공간 클러스터링 방법이다. DBSCAN은 거리 임계값 ε과 밀도 임계값 MinPoints를 이용하여 군집과 잡음을 분류한다.

임의의 지점 P에서 나머지 지점들과 거리를 계산하여, 거리 임계값(ε)의 범위 안에 있는 지점의 개수가 밀도 임계값(MinPoints)보다 크면 P를 중심점으로 지정한다. P로부터 거리 임계값 내에 있는 지점에서 같은 방법을 실행하여 거리 임계값과 밀도 임계값의 조건을 만족시키면 군집을 형성한다. 이와 같은 방법을 반복하여 군집을 추출하고, 군집에 포함되지 않는 지점은 잡음으로 처리한다[3].

2) MapReduce

MapReduce는 대용량 데이터 셋을 처리하기 위한 프로세스로 Map과 Reduce 두 함수로 이루어졌다.

Map은 사용자가 입력한 데이터 쌍들을 Key/Value 형태의 중간 값으로 출력한다. 각 중간 값은 Key값을 기준으로 그룹화하여 Reduce 단계로 전달된다.

Reduce는 동일한 Key 값을 갖는 데이터 들을 병합하여 Value 셋을 최소화 한다. 일반적으로 각각 Raduce 노드에서 0~1개의 Value 셋이 생성된다[4].

3) POI(Points of Interest) 추출

POI를 찾기 위해 한 사람이 오래 머문 지역을 stay point로 정의 하고 여러 사람들의 stay point를 클러스터링 알고리즘을 통하여 POI로 추출한다. POI를 추출하는 방법으로는 여러 가지가 연구되고 있다.

[5]은 Means shift와 Distance based clustering 방법을 이용하여 GPS 데이터에서 POI를 추출하였다. [6]는 DBSCAN을 이용하여 POI를 찾았다. [7]의 연구는 POI의 추출에 MapReduce와 Canopy 클러스터링을 이용했고 [8]는 Canopy 클러스터링으로 구한 초기 중심값으로 K-Means 클러스터링을 수행하여 POI를 구했다.

4) 데이터 분할 방법

DBSCAN을 분산 환경에서 구현하기 위해서는 데이터를 분할하는 과정이 필요하다. MapReduce는 데이터를 여러 노드로 분산하여 Map과 Reduce작업을 하고 최종적으로 데이터들을 수집하여 결과 값을 출력한다. DBSCAN 수행 시 데이터를 분산하여 클러스터링을 할 경우 클러스터가 형성되어야 될 데이터가 노이즈 데이터가 되거나 최종적으로 데이터를 병합하는 과정에서 데이터가 중복되는 경우가 발생한다. 이러한 이유로 MapReduce를 수행하기 전 데이터를 적절하게 분할하는 과정이 필요하다.

[9]은 데이터를 일정한 크기의 격자형식으로 나눈 뒤 각 격자에 포함되는 데이터의 비용을 계산한다. 동일한 비용의 데이터가 각 노드에서 계산될 수 있도록 데이터를 분산시킨다.

[10]에서는 데이터를 일정한 용량 크기의 블럭으로 분할한 후 각각 블록에 아이디를 부여한다. 최종단계에서 데이터를 병합할 때 아이디를 고려하여 클러스터링을 수행한다.

3. POI 추출을 위한 분산 네트워크 환경

1) K-Means를 이용한 데이터 분할 알고리즘

K-Means는 미리 정의된 클러스터의 개수에 따라 데이터를 분할하고, 각 클러스터의 중심과 데이터간의 평균 유클리디안(Euclidean) 거리를 최소화 시키는 방식으로 동작한다[11].

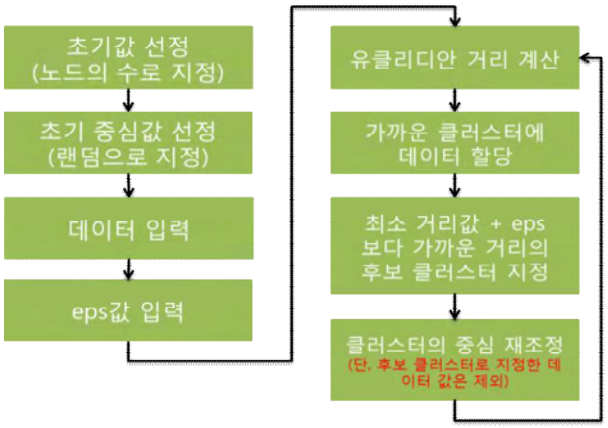
K-Means는 초기 중심값을 선정하고 데이터를 입력한다. 다음으로 클러스터의 중심과 데이터 사이의 유클리디안 거리를 계산하여 가까운 클러스터에 할당하고 데이터의 할당이 끝나면 클러스터의 중심을 재조정한다. 이 과정을 반복하여 수행하고 클러스터에 변화가 없으면 알고리즘을 종료한다.

(그림 1) K-Means로 데이터 분할시 문제점

본 논문에서는 기존의 알고리즘을 그대로 사용할 경우, 그림 2와 같이 DBSCAN 수행 시 군집되어야 할 데이터가 분할되어 로컬 클러스터링 과정에서 노이즈로 처리되는 경우가 발생한다.

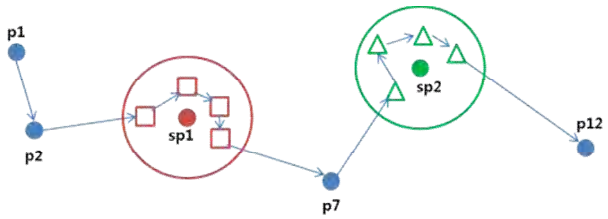
따라서 기존의 K-Means 알고리즘을 수정한다. 기존의 입력 데이터에 추가로 DBSCAN의 파라미터인 eps값을 입력 받는다. 데이터는 초기 중심값들과 유클리디안 거리를 계산하고 거리가 가장 가까운 클러스터에 데이터를 할당한다. 또한 가까운 중심값과 eps값을 더한 범위 내에 다른 클러스터의 중심값이 존재하면 데이터를 그 클러스터의 후보 데이터에 포함시킨다. 이 과정을 수행함으로써 클러스터 외곽에 위치한 포인트들의 군집이 클러스터링 과정에서 분할되어 노이즈 데이터가 되는 현상을 방지할 수 있다.

마지막으로 클러스터의 중심을 재조정하고 위 과정을 반복한다. 단, 클러스터 중심 재조정시 후보 데이터 값들은 계산에 포함 시키지 않는다. 후보 데이터 값이 중심 재조정 과정의 계산에 포함되면 같은 데이터 값이 여러 클러스터에 중복 계산되어 정확한 중심값의 산출이 불가능하다.



(그림 2) K-Means Algorithm

2) sp(Stay Point) 추출 모듈



(그림 3) stay point

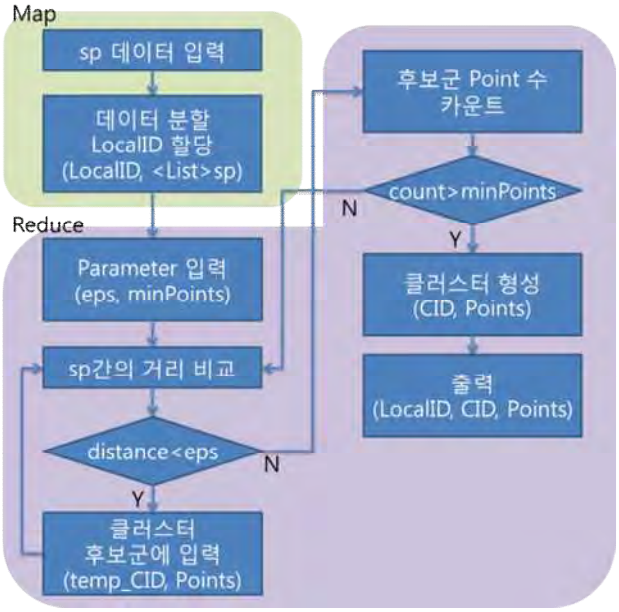
sp추출 모듈은 POI를 추출하기 전의 과정으로 GPS데이터를 sp로 병합하는 작업을 수행하는 프로세스다. sp는 거리 임계값과 시간 임계값을 이용하여 추출한다.

```
    Input : GPSdata(x,y,time), threshold_distance,  threshold_time
    Output : <List>staypoint(x,y)
 1  rawdata <- GPSdata(x,y,time)
 2  start_point <- rawdata
 3  while ( rawdata.hasnext )
 4     distance <- distance + distance(prev_rawdata, rawdata)
 5     time <- time + time(prev_rawdata, rawdata)
 6     if( distance < threshold_distance )
 7        end_point <- rawdata
 8     else
 9        if( time > threshold_time )
10           staypoint(x,y) <- merge(start_point, end_point)
11           start_point <- rawdata(x,y,time)
12           distance <- 0, time <- 0
13
14  emit(staypoint(x,y))
```

(그림 4) stay point 추출 pseudo 코드

GPS points { Pm, Pm+1, ... , Pn }에서 ∀m < i ≤ n 일 때, Pm부터 Pi까지의 거리가 거리 임계값 보다 작고, 이동하는데 걸린 시간이 시간 임계값 보다 클 때 포인트들을 병합하여 sp로 지정한다[12][13]. 본 논문에서는 병합된 포인트들의 영역에서 중심값을 계산하여 sp로 출력한다.

3) DBSCAN을 이용한 POI추출



(그림 5) Local Clustering

본 논문에서 제안한 POI추출 방법에서 DBSCAN 클러스터링은 2단계로 분류된다.

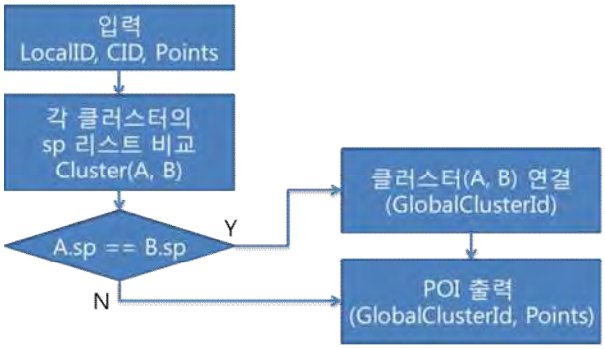
1단계는 로컬 클러스터링 단계이다. 분산 환경에서 DBSCAN 클러스터링을 하기 위해서는 데이터를 분할해야 한다. 이 분할된 구간 내에서 DBSCAN 클러스터링을 수행하는 것을 로컬 클러스터링이라 한다.

로컬 클러스터링 단계에서는 Map과 Reduce단계로 다시 나뉜다. Map 단계에서는 같은 파티션에 소속된 staypoint를 묶어서 LocalID를 부여해 준다.

Reduce단계에서는 Map의 출력값 (LocalID, <List>Point)을 입력 받아 클러스터링을 수행한다. DBSCAN을 수행하기 위해서 파라미터 eps와 minPoints를 입력 받고 sp 데이터들의 거리를 비교한다. 거리가 eps 값보다 작은 값들을 클러스터 후보군에 입력하고 클러스터 후보군 내의 포인트 수를 카운트 한다. 카운트한 값이 minPoints보다 크면 최종적으로 클러스터를 형성하고, 작으면 노이즈로 처리된다.

이 과정을 반복적으로 수행한다. 더 이상 클러스터링 할 포인트가 없을 경우 로컬 클러스터링을 종료하고 각 노드의 클러스터링 결과 값을 수집하여 출력한다.

글로벌 클러스터링 단계는 로컬 클러스터링 단계의 출력값을 입력 받아 각 클러스터의 staypoint 리스트를 비교한다. 동일한 staypoint를 가지고 있는 클러스터는 병합하여 하나의 클러스터로 만들고 같은 GlobalClusterId를 부여해 준다. 이 과정을 더 이상 연결할 클러스터가 없을 때 까지 반복 수행하여 클러스터를 생성하고 생성된 클러스터의 중심값을 POI로 출력한다.



(그림 6) Global Clustering

4. 결론

본 논문에서는 POI를 추출하기위해 MapReduce 환경에서 DBSCAN 클러스터링을 이용하였다. 또한 분산 환경에 적합한 데이터 분할을 위해 K-Means 알고리즘을 수정하여 적용하였다.

기존의 환경에서는 점점 늘어나는 데이터의 양으로 인해 데이터 분석에 많은 시간이 요구되어 사용자에게 시기적절한 정보제공이 힘들어지고 있다. 본 논문에서는 MapReduce를 이용하여 데이터를 분산 처리함으로써 처리속도를 향상시켰고, DBSCAN 클러스터링을 이용하여 사용자의 POI를 추출하는데 정확도를 향상시켰다. 그리고 DBSCAN을 MapReduce에서 수행하기 위한 데이터 분할 방법으로 수정된 K-Means 알고리즘을 사용하여 데이터에 대한 사전정보 없이도 POI를 추출 할 수 있도록 하였다.

ACKNOWLEDGMENT

본 연구는 2014년도 정부 (교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(NRF-2009-0093828)와 미래창조과학부 및 정보통신기술진흥센터의 ICT융합고급인력과정지원사업(IITP-2015-H8601-15-1006)의 연구결과로 수행되었음.

참고문헌

[1] A. Akinori, K. Maruyama, and A Sato et al., “Pedestrian-Movement Prediction Based on Mixed Markov-Chain Model,” Proc. of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2011.

[2] O. Ossama, H. M. Mokhtar, “Similarity Search in Moving Object Trajectories,” Proc. of the 15th

International Conference on Management of Data. Computer Society of India, pp.1-6, 2009.
[3] Hans-Peter Kriegel, Peer Kröger, Jörg Sander and Arthur Zimek1, "Density-based Clustering", WIREs Data Mining and Knowledge Discovery, Volume1, pp.231-240, 2011.3-5
[4] Jeffrey Dean, Sanjay Ghemawat, "MapReduce: simplified Data Processing on Large Clusters", Google, 2004.
[5] A. Kirmse, T. Udeshi, P. Bellver, and J. shuma, "Extracting patterns from location history," Int. Conf. on Advances in Geographic Information Systems, pp. 397-400, 2011.
[6] M. Zignani and S. Gaito, "Extracting human mobility patterns from GPS-based traces," Wireless Days, pp.1-5, 2010.
[7] 정성현, 박영택, "스마트폰 사용자의 관심지점 추출을 위한 MapReduce기반의 Canopy 클러스터링 기법", 한국컴퓨터종합학술대회 논문집, pp.1524-1525, 2013.
[8] 김종환, 이석준, 김인철, "다음 장소 예측을 위한 맵리듀스 기반의 이동 패턴 마이닝 시스템 설계", 정보처리학회논문지, pp321-328, 2014
[9] Yaobin HE, Haoyu TAN, Wuman LUO, Shengzhong FENG, Jianping FAN, “MR-DBSCAN: a scalable MapReduce-based DBSCAN algorithm for heavily skewed data”, Front. Comput. Sci., 2014
[10] Xiufen Fu, Shanshan Hu and Yaguang Wang , “Research of parallel DBSCAN clustering algorithm based on MapReduce”, International Journal of Database Theory and Application, pp.41-48, 2014
[10] 이신원, “K-Means 클러스터링에서 초기 중심 선정 방법 비교”, Journal of Korean Socieity for Internet Information 2012. Dec: 13(6): 1-8
[11] 진홍석, "최소 묘사 길이 원리를 이용한 경로 데이터베이스에서의 집단 발견", KAIST 전산학과 석사학위논문, 2012
[12] Matteo Zignani, Sabrina Gaito, "Extracting Human Mobility Patterns From GPS-based Traces", the 3rd IFIP Wireless Days Conference 2010.