ecoccon

라이프 존 재구성 고안

최창규

목차

01. LIFE ZONE 구성

기존 K-means 알고리즘을 사용해서 나누었던 라이프 존을

다른 알고리즘을 사용하여 구성해보는 것

01. LIFE ZONE 구성

DBSCAN 알고리즘

장점

K-means 알고리즘을 사용하면 상대적으로 밀도가 낮고 퍼져있는 아웃라이어를 배제하지 못해서 발생하는 라이프 존의 의미 감소가 DBSCAN을 활용한 밀도기반 클러스터링으로 판별할 수 있게 되면서 그 의미가 강해졌다.

단점

클러스터마다 다른 반경과 개수를 설정할 수 없어서 모든 클러스터를 동일한 조건으로 구분해야한다.

>> 사람이 2명 이상이라 판단될 때는 k-means알고리즘으로 분류하는게 바람직 할 것으로 생각

01. LIFE ZONE 구성

DBSCAN 알고리즘

직면한 문제점

사람마다 활동반경과 결재횟수가 달라서 모든 사람을 일반화해서 Epsilon, minPts 설정 불가능

해결방안 모색

특정 사람의 epsilon은 결재 시간 별로 정렬해서 결재와 결재 사이의 시간 차이와 결재 장소의 위치 차이를 이용해서 평균 속도를 구한다. 그 속도는 해당 사람의 활동반경을 의미한다고 볼 수 있다.

minPts는 특정 사람의 결재 횟수에 영향을 받는다. 총 결재량, 혹은 일정 기간 동안의 결재량에 비례한 값을 채택하면 된다.

02. 아웃라이어 판단법

- 상대적으로 방문 빈도가 적고 밀도가 낮은 지역
- 결재 시간의 차이에 비해 비정상적인 거리 차가 발생하는 경우

  >> 한 아이디로 두 사람 이상 사용, 온라인 결재, 중복입력…

- 특정 데이터의 정확도가 떨어지는 경우

02. 아웃라이어 판단법

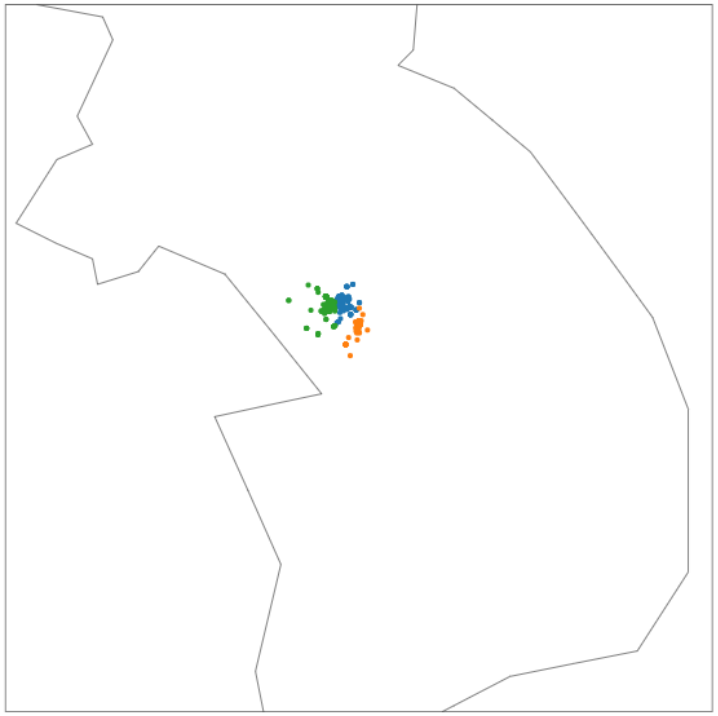
상대적으로 방문 빈도가 적고 밀도가 낮은 지역

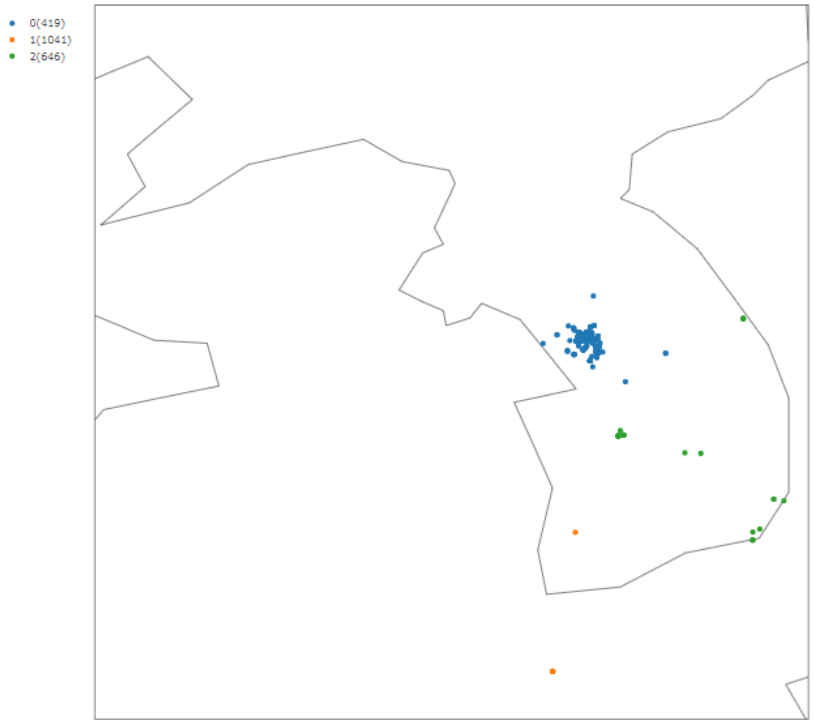
우선적으로는 DBSCAN을 이용해서 걸러지는 아웃라이어 들로 판별하는 방법을 생각했지만 불필요한 클러스터링 과정 으로 인한 최적화 필요

>> Isolation forest 라이브러리 (조금이나마 빨라지겠지만 시간복잡도는 비슷할 것으로 예상)

02. 아웃라이어 판단법

상대적으로 방문 빈도가 적고 밀도가 낮은 지역





DBSCAN을 이용한 아웃라이어 제거 전, 후 라이프 존 결과
>> 아웃라이어 제거의 유무에 따라 확연한 차이가 난다.

02. 아웃라이어 판단법

데이터의 정확도가 떨어지는 경우

Ex) 특정 column의 값이 없거나 주소의 부정확함, 전화번호의 유효성 등

>>이런 값이 많게 되면 라이프 존의 신뢰성이 떨어진다.

DB 문제점

1. ADDR과 SEARCH_ADDR의 일치율이 낮음

   >> ADDR과 SEARCH_ADDR이 도부터 다른 경우도 발견

2. 전화번호와 주소의 정규화 필요

   >>전화번호와 주소의 정규화가 된다면 추가적으로 INVARID한 값을 판별하는 것이 편리해진다.