<DBSCAN Clustering>

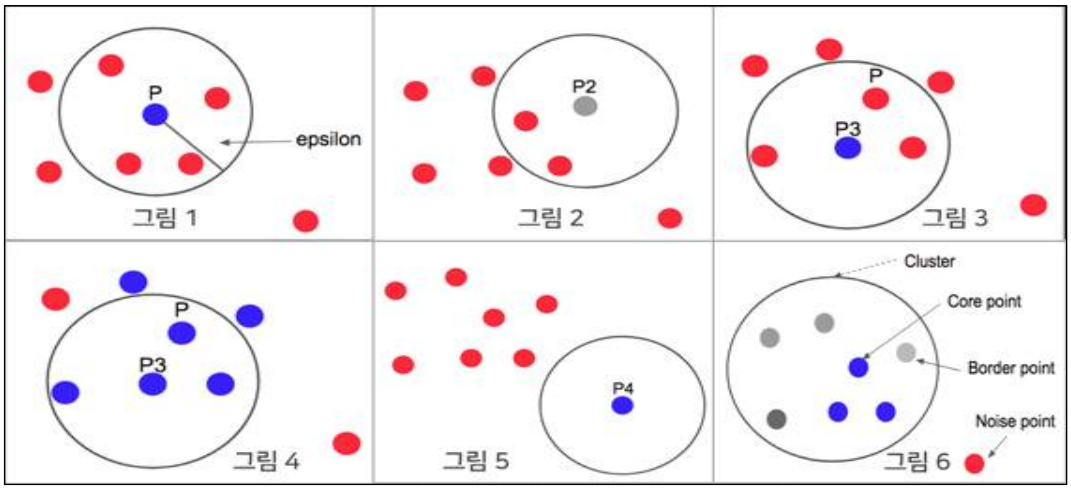
I. 설명

A. Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

B. 어느 점을 기준으로 반경 e내에 점이 n개 이상 있으면, 하나의 군집으로 인식하는 방법
예를 들어, 점 p가 있을 때 점 p에서부터 거리 e(epsilon)내에 점이 m(minPts)개 있으면 하나의 군비으로 인식한다고 가정하면, 거리 e내에 점m개를 가지고 있는 점p를 core point(중심점)이라고 한다.

C. 따라서, DBSCAN 알고리즘을 사용하려면, 기준점 부터의 거리 epsilon값과 이 반경 내에 있는 점의 수 minPts를 인자로 사용해야 한다.

II. 알고리즘



A. (그림1) minPts = 4라고 하면, 파란 점 P를 중심으로 반경 epsilon 내에 점이 4개 이상 있으면 하나의 군집으로 판단할 수 있다.
-> 점이 5개가 있기 때문에 하나의 군집으로 판단되고, P는 core point가 된다.

B. (그림2) 회색 점 P2의 경우, 점 P2를 기반으로 epsilon 반경 내의 점이 3개이기 때문에, minPts = 4에 미치지 못한다.
-> 군집의 중심이 되는 core point는 되지 못하지만, 앞의 점 P를 core point로 하는 군집에 속하기 때문에 이를 border point(경계점)이라고 한다.

C. (그림3) P3는 epsilon 반경 내에 점 4개를 가지고 있기 때문에 core point가 된다.

D. (그림4) 그런데 P3를 중심으로 하는 반경 내에 다른 core point P가 포함이 되어 있는데, 이 경우 core point P와 P3는 연결되어 있다고 하고 하나의 군집으로 묶이게 된다.

E. (그림5) P4는 어떤 점을 중심으로 하더라도 minPts=4를 만족하는 범위에 포함이 되지 않는다. 즉 어느 군집에도 속하지 않는 outlier가 되는데, 이를 noise point라고 한다.

F. (그림6) 정리를 하면,

- **점을 중심으로 epsilon 반경 내에 minPts 이상수의 점이 있으면 그 점을 중심으로 군집이 되고 그 점을 core point라 한다.**
- **core point가 서로 다른 core point의 군집의 일부가 되면 그 군집을 서로 연결되어 있다고 하고, 하나의 군집으로 연결한다.**
- **군집에는 속하지만, 스스로 core point가 안되는 점을 border point라고 하고, 주로 클러스터의 외곽을 이루는 점이 된다.**
- **어느 클러스터에도 속하지 않는 점은 Noise point가 된다.**

III. 장단점

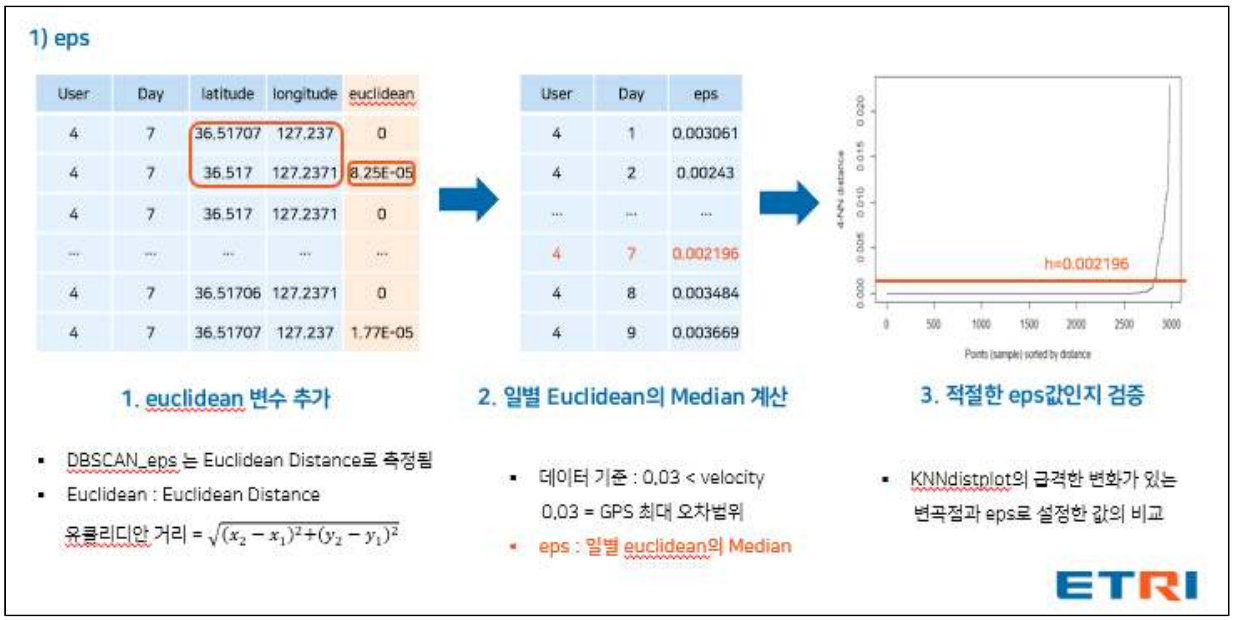
A. 장점

1. **일정 밀도 이상을 가진 데이터를 기준으로 군집을 형성하기 때문에, 노이즈 처리에 강하다.**
2. **k-means와 달리 클러스터의 수를 정하지 않아도 된다.**
3. K-means와 달리 밀도를 가진 코어들이 이어져가는 방식을 취하기 때문에, 기하학적인 모양을 갖는 데이터 분포도 잘 군집화할 수 있다.

B. 단점

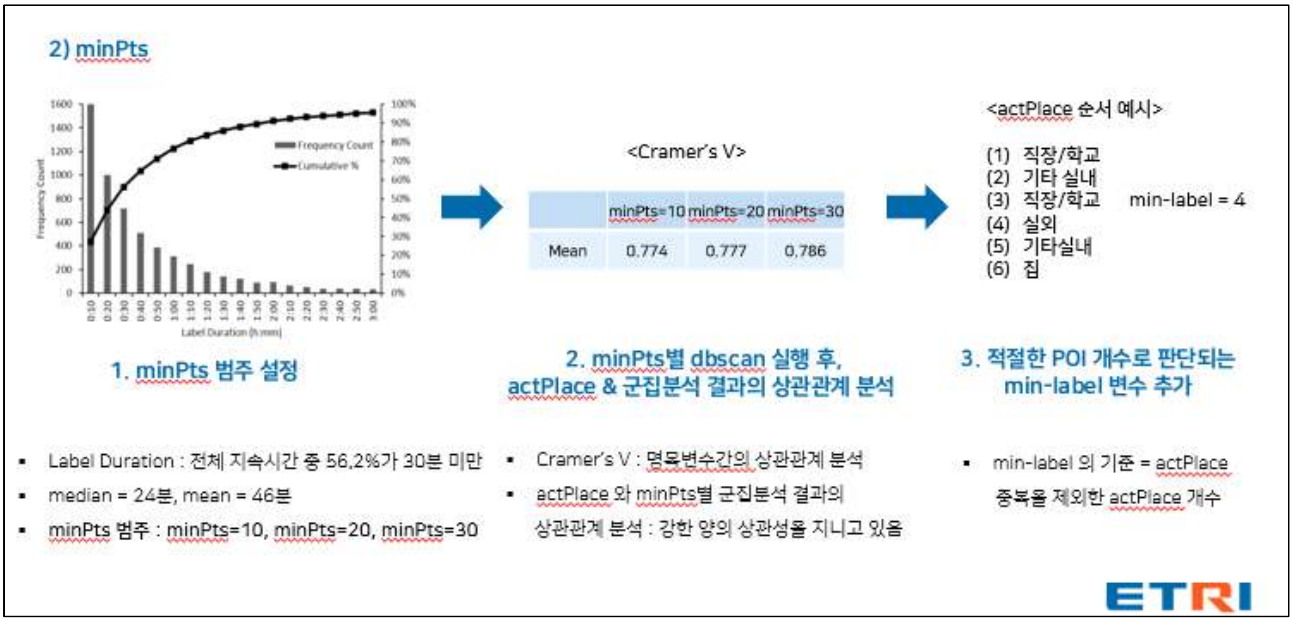
1. 많은 연산을 수행하기에 K-means보다 속도가 느리다.
2. **반지름과 임계치 설정에 많은 영향을 받는다.**
3. **유클리드 제곱거리를 사용**하는 모든 데이터 모델의 공통적인 단점인 'Curse Of dimensionality'가 존재한다. 이는 2차원이나 3차원 등 차원수가 낮은 데이터 세트에는 문제가 되지 않지만, 고차원 데이터 세트로 갈수록 필요한 학습 데이터 양이 급증하는 문제점이며, 이 때문에 많은 연산이 필요해진다.

IV. 이전 프로젝트 활용법



A. 파라미터 선정(eps)

1. gps데이터(위도, 경도)를 활용하여 두 위치 간의 euclidean 거리 계산
2. 일별 euclidean median 계산 (이때, mean 값이 bias가 커서 median 사용)
3. 적절한 eps 값인지 검증(KNNdistplot 활용)



B. 파라미터 선정(minPts)

1. 전체 데이터의 특성을 살펴서 범주 설정 (minPts = 10, 20, 30)
2. minPts별 dbscan 실행 후, 적절한 minPts 설정

<Mean-shift Clustering>

V. 설명

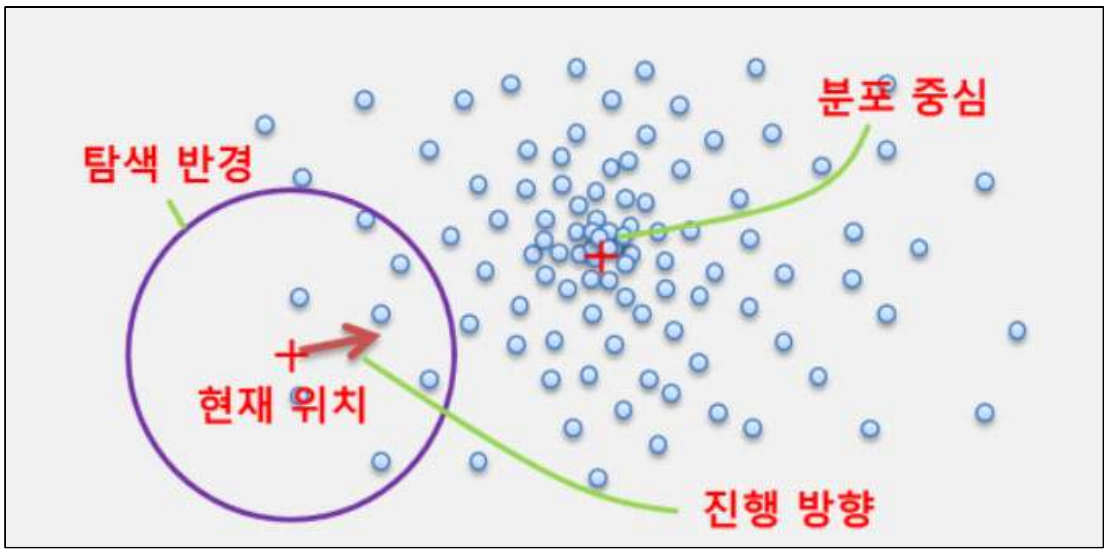
A. 평균 이동 클러스터링

B. 전체 특징 공간을 확률밀도함수로 표현한다. 즉, 어떤 데이터 분포의 무게중심을 찾는 방법으로서, 현재 자신의 주변에서 가장 데이터가 밀집된 방향으로 이동한다.

C. Window(탐색반경)가 data가 밀집한 곳을 찾는다. 이때, window 내의 data sample들의 평균(mean) 위치가 window의 center position으로 업데이트된다.

D. 주로 물체 추적, 실시간 데이터 분석, 영상 분석에 사용된다.

VI. 알고리즘



A. 현재 위치에서 반경 r 이내에 들어오는 데이터들을 구한다.

B. 이들의 무게중심 좌표로 현재 위치를 이동시킨다.

C. 위 과정들을 위치변화가 거의 없을 때까지, 즉 수렴할 때까지 반복한다.

D. 중심점(centroid)의 개수가 군집의 개수(k)가 되는 것이다.

VII. 장단점

A. 장점

1. K-means와 달리 군집의 개수를 미리 설정할 필요가 없다.
2. 물체추적, 영상 세그멘테이션, 데이터 클러스터링 등 다양한 활용방법이 존재한다.

B. 단점

1. window의 반지름 크기 r의 설정에 따라서 클러스터링 결과가 달라질 수 있다.
2. 기본적으로 주변의 지역적인 상황만 보고 진행방향을 결정하기 때문에, Mean-shift를 적용하기 위해서는 먼저 탐색반경(search radius)의 설정이 필요하다. 탐색반경을 너무 크게 잡으면 정확한 무게중심을 찾지 못하게 되며, 반대로 너무 작게 잡으면 local minimum에 빠지기 쉽다.
3. **탐색반경(window)의 크기를 정하는 것이 쉽지 않다.**

<향후 과제로 생각되는 것>

[표 4.3] 사례B의 클러스터의 개수에 따른 클러스터링 결과의 성능평가 수치

클러스터 개수 차이	알고리즘	균질도	분리도	반면영상너비
K = 5	DBSCAN	0.21923	1.13386	0.18621
	K-means	0.14959	0.67736	0.16935
	평균 연결법	0.25403	1.88477	0.20117
K = 10	DBSCAN	0.21708	1.26971	0.17232
	K-means	0.09167	0.60957	0.13010
	평균 연결법	0.18344	1.05797	0.199

- 알고리즘을 추려서, 위와 같은 기준을 설정한 후 모델 검증을 통해 선택하기