15. Density based Clustering
2019년 가을 학기

2019년 08월 19일
https://www.github.com/KU-BIG/KUBIG_2019_Autumn

이 글은 백준걸 교수님의 데이터 마이닝 강의의 강의안을 많이 참고하였다.

이번 강의에서는 거리를 기반으로 두지 않고 밀도를 기반으로 하는 clustering 기법에 대해서 살펴볼 것이다. 왜 distance based clustering 대신 density based clustering을 사용하는지와 대표적인 density based clustering 기법인 DBSCAN에 대해서 알아보는 시간을 가질 것이다.

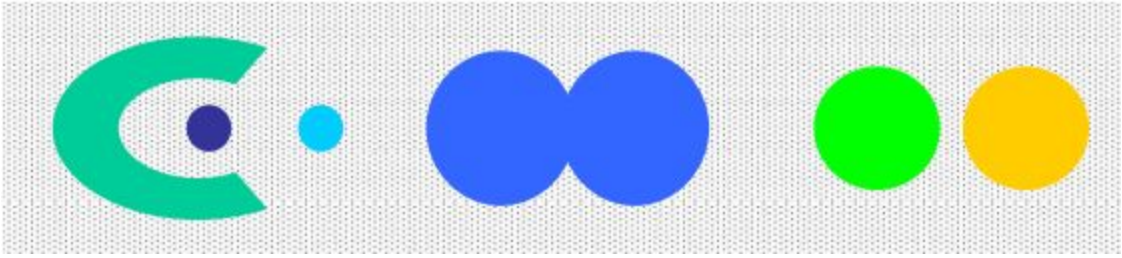
1. Introduction

우리는 앞의 강의에서 거리를 기반으로 둔 K-means clustering과 Hierarchical clustering에 대해서 살펴보았다. K-means clustering은 centroid와 medoid를 중심으로 가장 가까운 cluster에 point를 배정하는 방식이다. Hierarchical clustering은 가까운 point들을, cluster들을 서로 묶어가면서 cluster를 형성하는 방식이다.

이 두 기법은 거리의 정의에 따라서 결과가 다르게 나타난다. 거리를 기반으로 data들을 묶기 때문에 거리에 대한 정의가 달라지면 data를 묶는 기준이 달라지기 때문이다. 즉, distance based clustering에서는 거리에 대한 정의가 무엇보다도 중요하다. 이 뿐만 아니라 거리를 기반으로 했기에 나타나는 단점들은 여러가지가 있다.

K-means는 한 data를 가장 가까운 centroid가 속한 cluster에 배정한다. 이 때문에 cluster는 centroid를 기점으로 구의 형태를 띠게 된다. K-means는 초기 값의 영향을 많이 받는다. 초기값이 outlier로 잡힌다면 그 클러스터는 outlier으로만 구성될 확률이 커서 의미가 없어진다. 또한 data를 모르는 상태에서 cluster 개수를 미리 정해야 한다는 단점도 있다.

Hierarchical clustering의 경우 cluster 개수를 미리 정할 필요가 없다. 많은 경우에 K-means의 초기값 문제를 해결하기 위해서 hierarchical clustering의 결과를 사용하기도 한다. 하지만 hierarchical clustering에도 문제는 존재한다. Cluster간의 distance를 정의하는 방법이 매우 다양하다. 어떤 distance를 사용하느냐에 따라서 cluster의 형태가 정해진다.



위의 그림을 보자. 사람들은 위의 그림에서 쉽게 data들을 clustering할 수 있다. 바탕에 noise가 깔려 있어도 촘촘한 data와 그렇지 않은 부분을 쉽게 구분하여 noise를 무시하기 때문이다. 하지만 distance based clustering의 경우 모든 cluster들을 연결 시킬 것이다. 거리를 기반으로 하기 때문에 무시되어야 할 noise들이 다른 cluster로 이어지는 징검다리가 되기 때문이다.

위와 같은 상황에서는 거리 기반의 군집화 모델을 사용하는 것은 추천하지 않는다. Density based clustering이 있기 때문이다. 밀도를 중심으로 하기 때문에 data들이 빽빽히 모여있는 곳을 cluster로 묶고, 비교적 data가 듬성 듬성하게 있는 곳은 noise로 무시할 수 있다. 마치 noise의 바다 위에 떠있는 cluster 섬들만 잡아낼 수 있다.

2. DBSCAN (Density-based spatial clustering of applications with noise)

DBSCAN은 긴 이름에 모든게 담겨 있다. 밀도를 중심으로 하는(density-based) 군집화 기법(spatial clustering of applications)인데 노이즈(with noise)가 있어도 작동된다. 이 기법은 1996년에, Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu의 논문을 통해 소개되었다. 그리고 지금까지 데이터 분석가들에게 사랑받는 모델로 남아있다.

이 장에서는 DBSCAN에서 어떻게 밀도를 규정하는지에 대해서 살펴볼 것이다. 그 후, DBSCAN의 알고리즘을 간단하게 살펴보겠다. 마지막으로 DBSCAN의 장단점을 다룬 후에 이 길었던 강의의 마침표를 찍을 것이다. 여기까지 따라오느라 감사하다.

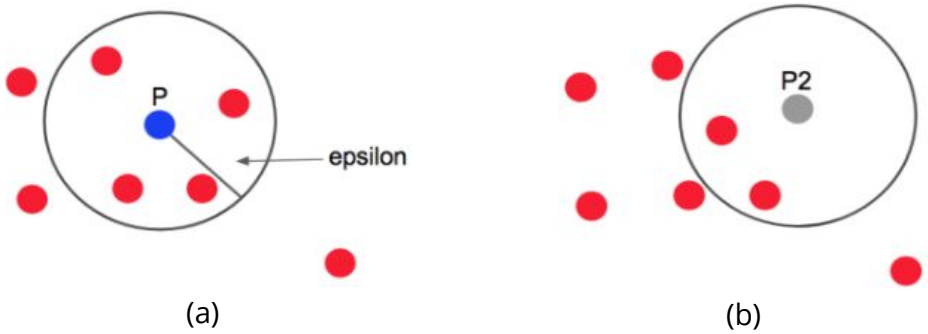
2.1 Density

우리는 distance based clustering을 배우기 앞서 distance를 어떻게 정의할 것인지 다뤘다. Density based clustering도 이와 같이 density를 어떻게 정의할 것인지 다뤄야한다. Density를 모르는 상태로 density 기반의 기법을 사용할 수 없기 때문이다. 앞에서 밀도 기반 군집기법을 설명하면서 DBSCAN은 distance의 굴레에서 벗어난 것처럼 서술했으나, 사실 density도 거리를

기반으로 정의된다. 따라서 distance based clustering과 같이 distance의 정의에 따라 그 결과가 값이 변할 수도 있다는 점은 짚고 넘어간다.

DBSCAN은 밀도 계산하기 위해서 모든 data에 대한 labeling을 실시한다. 모든 data를 대상으로, Core, Border, 그리고 Noise Point를 구별해야지 clustering이 가능해진다. 그리고 이렇게 일일이 labeling을 실시하는 이유는 이를 토대로 밀도를 계산하기 때문이다. 좀 더 자세히 설명하겠다.

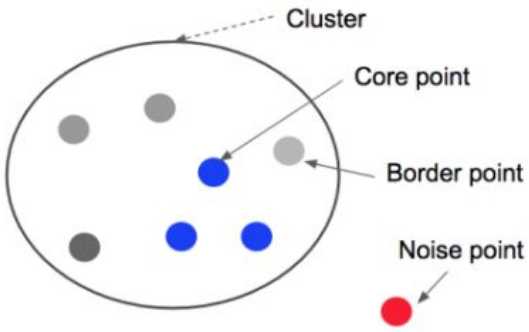
밀도를 규정하기 위해서 필요한 것은 radius(Eps)이다. 한 점을 중심으로 일정한 거리(radius) 이내에 몇 개의 점이 있느냐로 밀도를 계산하기 때문이다. 아래의 그림을 보면서 density를 어떻게 계산하는지 살펴보자. P는 정해진 radius안에 점의 개수가 4개 있기에 4의 밀도를 가진다. 반면에 P2는 radius안에 점이 2개 있기에 2의 밀도를 가진다. Radius안에 점이 많으면 많을수록 빽빽한 밀도를 가지게 되고 noise가 아닐 가능성이 커진다.



https://bcho.tistory.com/1205

이제는 Core, Border, Noise point를 구분하는 방법에 대해서 보자. 정답부터 말하자면 (a)의 P는 core point, (b)의 P2는 border point이다. Noise는 core와 border point가 아닌 point들이다. 밀도를 정의하기 위해서 radius가 필요했다면, core인지 정의하기 위해서는 MinPts가 필요하다. MinPts는 radius안에 최소 몇 개의 점이 있어야 core로 볼 지에 대한 기준이다. 위의 그림을 보면 MinPts가 4로, (a)에서는 이 기준을 만족하였기에 P는 core로 분류되지만, P2는 이 기준을 만족하지 못하기에 core로 분류되지 못한다. 이렇게 분류된 core들은 나중에 cluster를 구성하는 구심점의 역할을 하게 된다.

Core point들을 분류했다면 border point 분류는 더욱 쉽다. Core point의 radius 안에 들어오는 point들이 다 border point들이다. 즉, (a)에서 P를 둘러싼 원 안에 들어온 4개의 point들이 core P에 할당된 border point들이 된다. 이러한 border point들은 cluster의 구심점은 되지 못하지만, cluster에 포함된다. 이렇게 border point까지 다 구분하고 남은 point들은 noise가 되어서 cluster 구성시 제외된다.



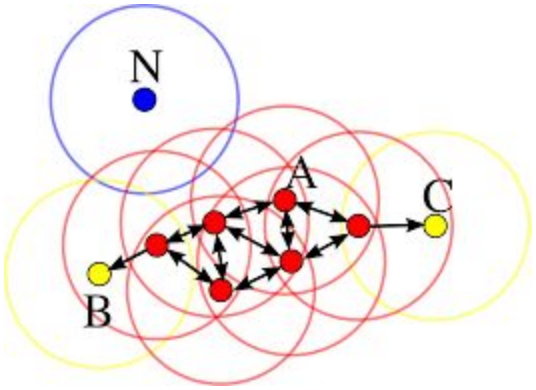
https://bcho.tistory.com/1205

2.2 Clustering

Core point, border point, noise point를 모두 분류한 후의 군집화 과정은 매우 간단하다.

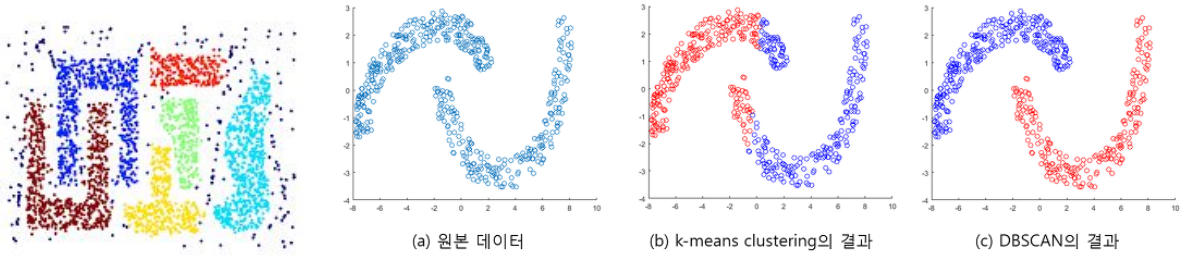
1. 모든 data point들을 Core, border, noise point로 분류한다.
2. 모든 noise point를 제거한다.
3. 임의의 core point에서 시작하여 radius(Eps)안에 들어오는 모든 core point들을 서로 연결하여 하나의 cluster을 형성한다.
4. Radius안에 core point가 더이상 들어오지 않는다면 연결을 그만두고, 연결되지 않은 core point에서 3의 과정을 반복한다.
5. 모든 core point들이 연결이 되어 cluster가 형성되었다면, border point를 속해있는 core point와 연결한다.

아래의 그림을 통해 다시 설명하겠다. 제일 먼저 noise인 N을 제외해야한다. A점에서 시작하여 radius안에 포함되어있는 모든 core point인 빨간점들이 서로 연결한다. 모든 core point들이 서로 연결되어 cluster을 이룬 후 cluster에 border point를 할당한다.



https://en.wikipedia.org/wiki/DBSCAN#cite_note-dbscan-1

DBSCAN으로 clustering을 한다면 다음과 같은 data들도 잘 군집된다.



(a) 원본 데이터 (b) k-means clustering의 결과 (c) DBSCAN의 결과

여러 Clustering을 시각적으로 잘 표현한 글을 아래에 참조하니 참고하기 바란다.

https://michigusa-nlp.tistory.com/27

2.3 DBSCAN 장단점 (영문 위키피디아 참조.[1])

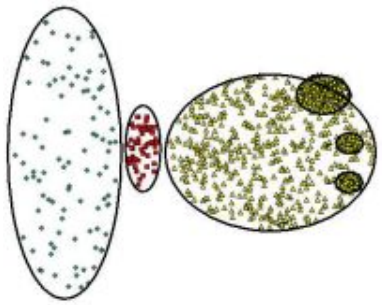
DBSCAN의 장점은 다음과 같다.

1. DBSCAN은 K-means와 다르게 사전에 cluster 개수를 지정할 필요가 없다.
2. DBSCAN은 다양한 형태, 크기의 cluster를 찾아낼 수 있다. 심지어 연결되지는 않았지만, 다른 cluster에 완전히 둘러쌓인 cluster도 찾을 수 있다. MinPts로 인하여, single-link 효과(다른 cluster끼리 약한 결합으로 연결되는 현상)가 경감된다.
3. DBSCAN은 noise를 판별해주며, outlier에 robust하다.
4. DBSCAN은 radius(Eps)와 MinPts만을 모수로 필요로 하며, 데이터베이스에 저장된 데이터 순서에 가장 덜 민감하다.
5. DBSCAN은 region query를 가속화 할 수 있는 데이터베이스와 함께 사용되도록 설계되었다.
6. 모수, radius(Eps)와 MinPts는 데이터가 잘 이해가 된다면, domain expert에 의해 설정 될 수 있다.

[1] https://en.wikipedia.org/wiki/DBSCAN#cite_note-dbscan-1

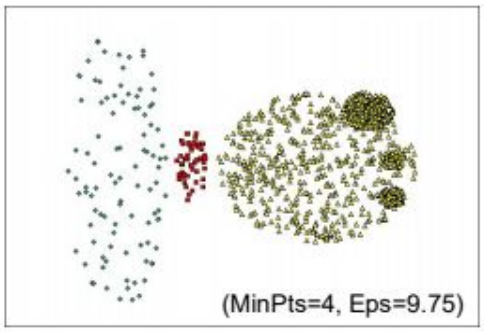
DBSCAN의 단점은 다음과 같다.

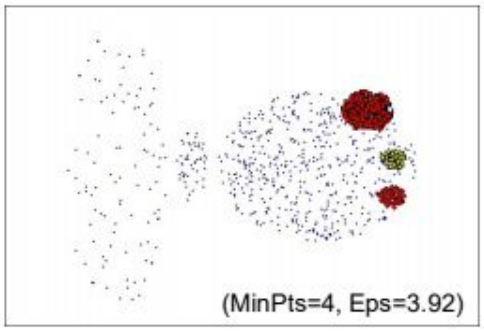
1. DBSCAN은 완전한 deterministic인 것은 아니다. : 하나 이상의 cluster에 속할 수 있는 border point들은 작업 진행시의 data의 순서에 따라서 cluster가 결정된다. 많은 경우에 이러한 현상은 잘 일어나지 않고, clustering 결과에 미치는 영향도 미미하다. Core point와 border point에 있어서 DBSCAN은 deterministic하다. DBSCAN은 border point를 noise로 처리하는 변형으로, 밀도로 연결된 요소에 대한 보다 일관된 통계적 해석뿐만 아니라 완전한 deterministic한 결과를 얻을 수 있다.
2. DBSCAN의 성능은 밀도를 측정하는데 사용되는 distance measure에 의존한다. 가장 널리 사용되는 distance metric은 Euclidean distance이다. 특히 고차원 데이터에서, 이 metric은 차원의 저주(“Curse of dimensionality”)로 인해 거의 쓸모가 없어질 수 있어, 적절한 radius(Eps)를 찾는걸 어렵게 만든다. 하지만 차원의 저주는 Euclidean distance를 쓰는 어떠한 알고리즘에서도 일어난다.
3. DBSCAN은 다양한 밀도를 가진 data set에는 잘 작동하지 못한다. MinPts와 radius (EPS)의 조합이 모든 cluster에 적절하게 선택되지 못하고 일괄적으로 작용하기 때문이다.
4. 만약 data와 단위가 잘 이해되지 않는다면, 의미있는 radius(Eps)를 고르는 것이 어렵다.



Original Points

- Varying densities
- High-dimensional data

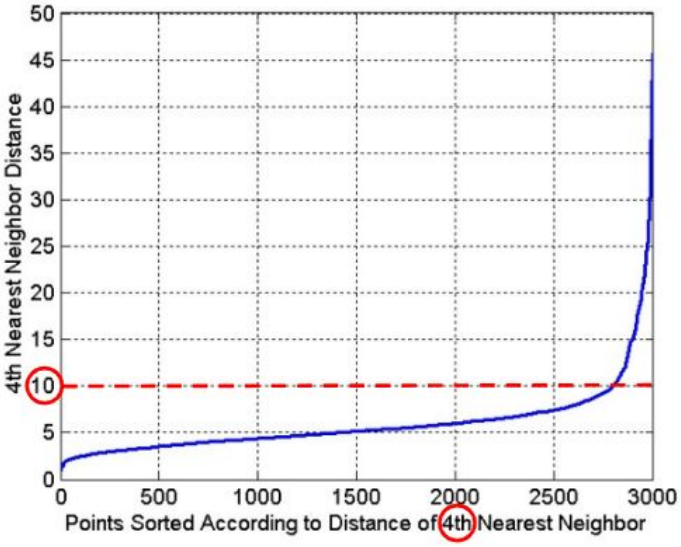




2.4 Parameter 선택

모든 모델이 그렇듯이 DBSCAN도 parameter인 radius(Eps)와 MinPts를 잘 설정하는 것이 중요하다. 따라서 많은 분석가들은 경험을 통해서 나오는 직감, domain 지식을 통해서 parameter를 설정하거나, 시행착오를 통해서 가장 성능이 좋은 모수를 찾아간다. DBSCAN에도 다양한 parameter 설정 방법이 있지만 이 장에서는 k-distance graph 방법만 소개하겠다.

이 radius, MinPts 설정 방법의 아이디어는 k^{th} nearest neighbor를 활용한다. K-nn에서는 가장 가까운 neighbor부터 k번째로 가까운 neighbor의 정보를 활용했다면, 여기에서는 k번째로 가까운 neighbor의 정보만을 이용한다. 정확하게는 모든 point마다 k번째로 가까운 point까지의 거리를 구한다. 그렇게 구해진 거리를 정렬하면 아래의 그림과 같이 된다.



위의 k-distance graph을 보면 4^{th} Nearest Neighbor Distance를 정렬하였다. 여기서 급격하게 distance가 증가하는 elbow point의 y축을 보면 distance가 10인 지점인 것을 확인할 수 있다. Distance가 10보다 큰 point들을 noise로 볼 수 있다. 즉, 이 graph를 통해서 우리는 radius를 10, MinPts를 4로 잡을 수 있다. 대부분의 noise가 아닌 point들은 10의 거리 안에 4개 이상의 point들이 있기 때문이다.

여기까지 KU-BIG 2019 가을학기 Session 강의안이었습니다. 이 강의안이 앞으로의 프로젝트에 많은 도움이 되길 소망합니다. 매 세션마다 참여해주셔서 감사합니다.

- KU-BIG 학술부 일동 -