**Clustering이란?**

클러스터링은 비슷한 언레이블된 데이터를 그룹화하는 것

분류는 패턴에 따라 사전 정의된 클래스로 인풋데이터를 카테고리화하는 것

Partioning-based clustering? Kmeans, gmm

Density-based clustering? Dbscan, hdbscan

중요도 : 여러 분야에 활용됨

Flight data clustering, detecting anomalies in facility operation, vertiport placement, convective weather modeling

**K-means algorithm**

레이블이 없는 데이터셋을 서로 다른 클러스터로 그룹화하는 비지도 학습

알고리즘

**- K =** 사전 정의된 생성해야하는 클러스터의 수

**- Working mechanism**

단계 **1** 클러스터의 수 **(K)**를 지정합니다**.**

단계 **2. K**개의 인스턴스를 무작위로 선택하여 중심점**(centroid)**을 초기화합니다**.**

단계 **3.** 각 데이터 포인트를 거리 측정법을 사용하여 가장 가까운 중심점에 할당합니다**.**

단계 **4.** 클러스터 내 모든 포인트의 평균을 계산하여 중심점을 재계산합니다**.**

단계 **5.** 알고리즘이 중지 기준을 충족할 때까지 단계 **3**과 **4**를 반복합니다**.**(중지기준은 더 이상 변화가 없을 때, 최대 반복수에 도달했을 때 드등이 가능)

단점 :

K-means 알고리즘은 원을 향해 clutering을 함. 즉 불규칙적인 모양을 지닌 클러스터를 다룰 땐 성능이 나쁨

다른 말로 kmeans알고리즘은 클러스터 모양에 대한 유연성이 부족함

아웃라이어에 로버스트 하지 않음. Noisy 데이터를 다룰 수 없음

Initialization에 민감하미 다른 initialization이 다른 클러스터를 생상할 수 있음

사전에 클러스터의 수를 미리 선정해야함

**클러스터링 평가**

1. Elbow method

데이터셋에서 클러스터의 최적의 수를 결정하기 위해 사용되는 방법 가장 가까운 클러스터 센터와의 거리 제곱합으로 계산

**-** 각 클러스터 내의 데이터 포인트들과 해당

클러스터 중심점 간의 거리 제곱합**(SSE)**를

계산하여 **, SSE**가 급격히 감소하다가

완만해지는 지점을 찾음**.**

**-** 장점 **:** 간단하고 직관적**,** 계산이 빠름

**-** 단점 **:** 주관성**,** 데이터 포인트들이 밀집해

있으면 엘보우가 나타나지 않을 수도 있음

2. Silhouette score

**-** 클러스터링의 품질을 평가하는 지표로**,** 각 데이터 포인트가 자신의 클러스터

내에 잘 맞는지**,** 또는 다른 클러스터와 더 유사한지를 나타냄

**-**실루엣 스코어는 **-1**에서 **1** 사이의 값을 가지며**,** 값이 클수록 클러스터들이 잘 퍼져있으며**,** 클러스터 내의 데이터 포인트들이 유사함을 의미. 낮은 실루엣 스코어는 클러스터가 중첩되거나 안좋음을 의미

**a(i) :** 데이터 포인트 **i**와 같은 클러스터에 속한 데이터

포인트들과의 거리의 평균🡺 동료 들과 얼마나 데이터 포인트가 유사하냐

**b(i) : i**와 다른 클러스터 중 가장 가까운 클러스터까지의 평균

거리🡺 이웃 클러스터와 분리된 저옫

**-** 장점 **:** 클러스터 내부에 데이터 포인트 간의 응집력과 클러스터 간의 분리된

정도를 동시에 고려하므로 클러스터의 품질을 종합적으로 결정 가능**, K**값에 따라

비교하며 최적의 **K**를 찾을 수 있음**.**

**-** 단점 **:** 계산 비용이 비쌈

1. 실루엣 다이아그램

클러스터별로 실루엣 점소ㅜ를 계산하여 시각화한 그래프

**GMM** = Gausian mixture model

Gaussian = 연속확률분포의 유형인 가우시안을 의미

가우시안 분포의 매개변수 평규 mu, 표준 편차 sigma

Mixtuae = 가우시안 분포들의 결합 🡺 파라미터들을 몰라

Model = 몇몇 가우시안 분포들의 결합으로 생성된 데이터라는 가정하에 gmm이 클러스터링 태스크에 사용될 수있음

* Gmm은 알지 못하는 파라미터를 가진 가우시안 분포들의 결함을 사용하는 혼합 모델이고. 데이터들이 이러한 가우시안 분포의 혼합으로부터 형성되었으며, 언라벨된 데이터 포인터의 셋을 클러스터로 그룹핑하는 클러스터링에 사용될 수 있다.

**EM** = Expectation maximaization

언라벨된 데이터 포인트들이 있을 때 해당 데이터들이 두개의 다른 가우시안 분포로부터 나왔다고 가정하자. 이때 우리는 데이터포인트들을 두개의 그룹으로 클러스터링 하고 싶음. 어떻게 할까?

우리가 파라미터들로 가우시안 분포를 추정하고 싶으면, 데이터 분포들이 어떤 가우가우시안분포로 나왔을 것 같은지를 추정해야함. 왜냐면 데이터로 파라미터들을 추정하기 때문에

* Chicken and egg problem : 데이터를 두 분포로 그룹해야하는데 이를 하려면 가우시안 분포를 알아야함. 즉 파라미터들을 알아야하는데 파라미터들은 데이터들로 알 수 있음.
* EM은 반복적인 방법으로 이를 해결함

EM 알고리즘은 GMM의 파라미터를 추정하는 방법임.

EM알고리즘은 gmmdml 파라미터들을 initializing하고 반복적으로 파라미터의 추정값을 향상시킴

Two step iterative algorithm

* 파라미터 initializing과정(ex, randomly place gaussian distribution)이 필요함

Step1 : E-step(expactation), latent variable을 추정함(latent variable은 직접적으로 관찰되거나 측정이 되지 않지만 추정될 수 있는 변수를 의미함)

Step2 : M-step(maximizing), 가장 데이터를 잘 설명하는 모델의 파라미터를 optimizie함.

2-D example

Step1 : initaialization

Covariance matrix의 diagonal element는 변수의 분사를 의미함

Off diagonal element가 0이면 두 변수사이에 상관관계가 없음을 의미

Step2 : Expectation

랜던하게 설정된 파라미터들을 가진 가우시안분포들에 대해 각 데이터마다 어떤 분포에서 나왔는지를 추정함

하드 클러스터링 : 각 데이터포인트는 방드시 하나의 클러스터에만 속함

소프트 클러스터링 : 데이터가 확률을 가지고 클러스터에 속함. 즉 한 데이터포인트가 여러 클러스터에 속할 수 있음

* 이런 확률을 responsibilities(r)을 구함으로 알 수 있음. R은 주어진 데이터에 대한 사후확률임
* Responsibilities에 따라 데이터포인트를 클러스터링 그룹에 할당함

Step3 : Maximization 할당된 데이터 포인트에 따라 gmm 의 파라미터를 다시 추정함

기준을 만족할 때까지 estep과 m-step을 반복함

알고지름 중단기준

Ex) 이전 반복과의 로그-라이클리후드를 비교함 / 최대 반복수를 지정함.

단점. gmm이나 em이 k-means보다 다양한 cluster shape를 다룰 수 있는 것은 맞지만 여전히 partioning based cluster approach임.

* Roughly uniform density에 대해서는 밀도 기반 ㅡ클러스터링이 더 적절할 수있음

노이즈한 데이터는 잘 못다룸

**DBSCAN : Density based spatial clusteringof appliciaiton with noise**

**어느점을 기준으로 반경 x내에 점이 n개 이상 있으면 하나의 군집으로 인식하는 방식이다.**

**그러면 조금 더 구체적인 개념과 용어를 이해해보자**

**먼저 점 p가 있다고 할때, 점 p에서 부터 거리 e (epsilon)내에 점이 m(minPts) 개 있으면 하나의 군집으로 인식한다고 하자. 이 조건 즉 거리 e 내에 점 m개를 가지고 있는 점 p를 core point (중심점) 이라고 한다.**

**DBSCAN 알고리즘을 사용하려면 기준점 부터의 거리 epsilon값과, 이 반경내에 있는 점의 수 minPts를 인자로 전달해야 한**

Step1 : 데이터에서 임의의 점을 고름=> 해당 점은 visited point 가 됨

Step2 : 선택된 데이터로부터 EPS내의 모든 이웃을 추출함. EPS(epsilon)은 이웃으로 여겨질 수 있는 두 포인트의 최소 거리를 말함. 🡺 두 포인트 사이의 거리가 eps보다 작다면 이웃이 됨

Step3 : 추출한 이웃의 수가 minPts이상인지를 확인함

Minpts는 dense하다고 생각될 수 있는 지역에 함깨 있을 수 있는 최소한의 포인트의 수

Stpe4 : 포인트의 유형을 확인함

추출된 이웃 포인트의 수가 minpts 보다 크면, 선택된 점은 core point가 됨.

* Core point는 eps내에 최소한 minpts개의 점을 가져야함

추출된 이웃 포이트의 수가 minpts보다 작으면, 선택된 점은 noise point가 됨

Step5 : 최초의 코어 포인트로부터 클러스터를 만듦

최초의 코어포인트의 이웃에 있는 모든 점(즉 eps내에 있는 점)은 같은 클러스터라 고려함

Step6 : 2-5를 같은 클러스트내에 모든 이웃 포인트들을 찾나앨 때까지 재귀적으로 반복

* eps내에 minpts보다는 적은 점을 갖고 있지만, 코어 포인트의 이웃인 경우에는 border point가 됨
* 코어포인트의 이웃도 아니면 noise point 가 됨

Step7 : 1-6을 데이터 셋의 남은 방문하지 않은 데이터들에 대해 반복함

DBSCAN 알고리즘의 장점은

* K Means와 같이 클러스터의 수를 정하지 않아도 되며,
* 클러스터의 밀도에 따라서 클러스터를 서로 연결하기 때문에 기하학적인 모양을 갖는 군집도 잘 찾을 수 있으며  
  Noise point를 통하여, outlier 검출이 가능하다.

한계점

Roughly uniform density를 갖는 클러스를 찾기 위해 설계됨. 즉, dbscan알고리즘은 다른 것보다 밀도가 높은 클러스터가 있는 ㅇ여러 밀도를 가진 클러스터를 지닌 데이터셋을 다루기 위해 만들어짐

사실 eps와 minpts는 고르기 어려움 우리는 좋은 추측을 위해서는 데이터 분석을 해야함. 그러나 알고리즘의 수행은 파라미터 선택에 민감함. 다양한 밀도를 가진 데이터셋의 경우엔 더 🡺 다양한 밀도를 가진 데이터의 경우에 파라미터의 선택에 매우 민감함

**HDBSCAN : Hierarchicla Density based spatial clustering of application with noise**

Dbscan은 user-defined input parameters로 eps와 minpts가 필요했지만 hdbscan은 minpts만 필요함 🡺 hdbscandl 다양한 밀도의 클러스에 더 flexible한 접근법임

STEP1 : density, sparsity에 따라 공간을 특정한 거리 metirc에 따라 변형함

거리 metric은 mutual reachablity distance dmreach−k(a,b)=max[corek(a),corek(b),d(a,b)]

이를 **mutual reachability**라고 부른다. 두점 a와 b의 거리를 잴때 **{1. a의 이웃과의 거리, 2. b의 이웃과의 거리 3. a와 b자체의 거리}중 max값**

* Noise에 로버스트함

sTep2 : 미니멈 스패닝 트리를 만듦

* 데이터를 weighted graph라 여김. 데이터 포인트를 vertices로, 두 점 간에 edge를 mutual reachability distance로
* Weighted graph가 정의되면, 우리는 그래프의 mst를 만들 수 있음

Step3 : 연결된 컴포넌트의 cluster hierachy tree를 만듦

Mst를 hierachy of connected components로 변환함

* 색은 클러스터내에서 크룹화된 데이터 포인트의 수를 나타냄

Step4 : condense the cluster hierachy tree

* 복잡한 cluster hierachy 를 작은 그룹으로 condensing down함 더 많은 데이터가 각 노드에 존재하도록

선의 너비는 클러스터에 있는 데이터의 수를 나타냄

노들에 데이터 포인트를 붙일 때 기준은?

먼저, hdbscan 알고리즘의 하이퍼 파라미터 minpts가 필요함. 하이퍼 파라미터 값을 가지면, 각 split에서 hierarchy를 걸으며 물어봄

Minpts보다 적은 값을 가지면, 우리는 노드를 부모의 클러스터에 붙임

minpts보다 많은 값을 가지면, 클러스터를 나누고 트리에 남겨둠

Step5 : condensed cluster hierachy treef부터 stable cluster를 추출함

계속해섯 ㅓㄹ아남든 클러스터를 stable cluster라 생각

Coarse region->high distance -> low lambda

Dense region-> low distance -> high lambda

Hdbscan 알고리즘은 one 파라미터 minpts만 결정하면됨. 이 파라미터는 dense하다고 여길 지역에서 클러스터될 포인트의 최소의 숫자를 의미함

그러나, hdbscan 라이브러리를 이용하면 min\_cluster\_size를 결정해야하함

Mincluster\_size는 클러스트라 여길 그룹의 최소 크기를 의미함

타당한 클러스터라고 여겨질 클러스트의 최소 크기를 결정하는 기준을 세우게 함

**Minimum spanning Tree**

Spanning tree란 그래프의 모든 정점을 포함하는 트리를 말한다.

Minimum spannign tree는 최소의 edge weight를 갖는 weighted graph의 spanning ㅅree

Minimum spanning tree problem은 weight의 합을 최소화하면서 그래프의 모든 vertex를 연결하는 트리를 형성하는 edge의 부분칩합을 찾는 것

응용 telephone network design, water supply network design, currency market analysis, airspace network design

**프림 알고리즘**

Step1: Select an arbitary vertexx as the starting vertex and add it to the MST

Step2: Select the edge with smallest cost(ie greecy choice 최적이라 생각하는 것을 결정) from the starting vertex

Step3: Select the edge with smallest cost among the edges connecting the incomplete MST to other vertices

Stpe4: repeat until we have the mst

* Disconnected graph에는 프림알고리즘으로 mst를 찾을 수 없음
* Directed graph에는 프림알고리즘으로 mst를 찾을 수 없음. 모든 노드가 다른 노드에 접근 불가능해서

**데이터 탐색을 위한 비지도 학습 : 차원 축소 소개**

매 비행마다 고차원의 대이처를 처리하는 것은 계산적으로 비쌈, 리얼 타임 분석의 타당에 대한 불확실함을 야기함

LiDAR 센서는 디델인한 point cloud 데이터를 많이 만듬. 차원축소 알고리짐음 중효한 구조적 정보는 보존하면서 point cloud 데이터를 압축시킬 수 있음

카메라에 포작한 시각 정보에서, 차원 축소 알고리즘은 중오한 시각적 특징을 뽑아잴 수 있늠 필수적인 정보는 보유한채로

차원 축소는 데이터셋에서 인풋 변수(feature)의 숫자를 줄이기 위해 사용되는 머신러닝 기법임

(오리지날 데이터의 essence는 보존한채로 고차원 데이터에서 낮은 차원 데이터로 변환하는것)

주된 목표는 필수적인 정보는 유지한채로 데이터 셋을 단순화 시키는 것

필요성

Feature engineering : 차원 축소는 가장 관련있는 피처를 선택하고 인식하는 것을 도와줌, 이건 효과적인 모델 구축에 증요한 일임

Improving Computational efficiency : 고차원 데이터 셋은 종종 비싼 계산 자원을 요구함. 차원 축소는 피처의 수를 줄일 수 있게함. 즉 계산을 더 효율적으로 만듦

Enchacing a way of visualization : 고차원 데이터셋은 시각화하기 어려움. 차원 축소는 고차원 데이터셋을 저차원으로 만드러 시각화하기 쉽게 만듦

Addressing the curse of dimensionality : 차원 축소는 차원의 저주를 다를수 있게함

* 차원의 저주란 고차원의 데이터를 분석할 때 발생하는 다양한 어려움을 말함. Ex) knn 분류 알고리즘은 차원의 저주에 취약함, 차원이 커질수록 점들이 멀어지며 빈공간이 많아짐

유형

* Linear algorithms : 피처들이 linear combinations으로 표현된다고 가정함 ( PCA, LDA)
* Non linear algorithms : kernel pca, t-sne, isometric mapping

차원 축소 d알고리즘은 의도치않게 중요한 정보를 버릴 가능성이 있음 따라서 이 알고리즘을 적용할 땐 주의해야함

**PCA(주성분분석)**

차원 축소 방법중 하나

분산을 최대화하도록 데이터셋을 변화시킴

선형회귀에서는 회귀선과 데이터포인트의 수직 거리를 최소화했지만,

Pca는 데이터 포인트 간의 직교 거리를 최소화함

- 일반적으로 PCA는 다음 단계를 따릅니다.

1단계. D차원 데이터에 대한 데이터 전처리(예: 스케일링) 수행

2단계. 공분산 행렬 계산

3단계. 공분산 행렬의 고유값과 고유벡터 계산

4단계. 고유값을 내림차순으로 정렬하고 해당 고유벡터를 찾습니다.

5단계. 가장 큰 K개 케이스(K≤ D)를 선택하고 해당 케이스를 사용하여 투영 행렬(즉, w)을 만듭니다.

6단계. 투영 행렬을 사용하여 데이터를 저차원 공간으로 변환

장점

차원 축소: 해석하기 어려운 고차원 데이터 세트를 처리할 수 있습니다.

접근성: 쉽게 구현할 수 있습니다.

시각화: 데이터를 저차원 공간으로 변환하여 데이터 시각화에 사용할 수 있어 패턴을 시각화하기가 더 쉽습니다.

모델 성능 향상: 경우에 따라 모델을 피팅하기 전에 PCA를 적용하면 성능이 향상될 수 있습니다.

단점

해석 가능성 손실: 경우에 따라 해석 가능성을 향상시킬 수 있지만 데이터 세트의 차원을 줄이면 일부 정보와 세부 정보(예: 지나친 단순화)가 손실될 수도 있습니다

이상치에 대한 민감도: 데이터 세트의 이상치에 민감합니다. 이상치는 주성분에 영향을 미쳐 잠재적으로 잘못된 결과를 초래할 수 있습니다.

선형 가정: 변수 간의 관계가 선형이라고 가정합니다. 즉, 기본 관계가 비선형이면 제대로 작동하지 않을 수 있습니다.

**데이터 분석을 위한 통계적 접근법 : Sampling**

모집단은 데이터포인트의 전체 세트

샘플은 분석을 위해 선택된 모집단의 부분집합

샘플링은 큰 그룹에서 부분집합은 선택하기위한 통계적 기법

전체 모집단을 조사하는 것은 비용이나, 현실적인 제약 때문에 불가능하니까 실시

Biased sampling : 표본이 모집단을 대표하지 않는 biase가 발생하면 문제

Low response rate :

🡺표본을 더 정확하게 정의할수록 유리한 결과를 생성할 가능성을 높임

**복잡한 분포에서 데이터를 샘플링하는 것의 중요성은 무엇입니까?**

**샘플링이ㅡ 중요성**

• 머신러닝 모델 학습: 머신러닝 기반 모델의 매개변수는 학습 데이터로부터 학습됩니다. 분포에서 직접 샘플링하면 보다 대표적인 훈련 데이터세트를 제공할 수 있습니다.

■ 시뮬레이션 연구: 다양한 엔지니어링 분야에서 시뮬레이션은 기본 분포가 복잡한 시스템을 연구하는 데 사용됩니다. 분포에서 직접 샘플링하면 시뮬레이션 연구를 위한 데이터 세트를 생성할 수 있어 연구자가 시스템 동작을 이해하는 데 도움이 됩니다.

▪ 데이터 탐색: 직접 샘플링을 통해 복잡한 분포의 속성을 탐색할 수 있습니다. 분포를 이해하는 것은 실제 시나리오에서 정보에 근거한 결정을 내리는 데 필수적입니다.

**Acceptance rejection sampling**

간단한 분포에서 샘플을 생상하고 acceptance/rejecton 기준에 따라 샘플을 목표 분포의 샘플로 받아들임

Step1 : simple distribution을 마듦 = proposal distribution

Step2 : proposal distribution이 타겟 distributon을 덮는지 확인

Step3 : 제안 분포에서 샘플을 만듦

Step4 : 0,1사이에 균등 난수를 마듦

Step5: 기준에 따라 샘플을 수용함 accept x0 if u < p(x)/Mq(x)

Step6: 미리 결정된 샘플의 수에 따라 반복

* M은 regection rate에 따라 조정

장점 : 어떤 타겟 분포에도 사용가능, 간단하고 쉬움

단점 proposal distribution이 target에 비해 너무 크면 비효율적 🡺 accpetance rate가 낮으면 비효율적

적절한 proposal distribution을 찾는 것이 어려움

**MCMC 샘플링**

Mcmc 샘플링의 motivation은?

가우시안 분포에서 바로 샘플링은 Box muller transform algorith으로 가능(정규화되어 있고, standard하며, 독립적인 샘플링 가능)

그러나, 복잡한 분포에서 직접 샘플링 하는 것은 complexity나 intractablity 때문에 불가능( 왜냐면 pdf에 대핸 closed form 표현이 없거든)

Regection sampling은 계산적으로 비싸, 특히 high rejection rate를 가질 때(즉, acceptance rate가 낮을 때)

개념

복잡한 확률 분포로부터 직접 샘플을 얻는 방법인 마코브 체인과 몬테 카를로 방법을 결합한 것.

마코브 체인 방법은 시스템의 미래상태는 오직 현재의 상태에만 의존함을 설명하는 수학적 모델.

몬테카를로 (repeated random sampling)을 이용하여 함수의 값을 수리적으로 근사하는 알고리즘

작동 원리

Target distribution p(x)로부터 샘플링을 하고 싶음

Step1 : choose initial value x0 randomly

Step2: Define a proposal distribution

* 메트로폴리스 알고리즘을 활용하려면 proposal distribution은 반드시 대칭적이어야함

Step 3: proposal distribution으로부터 값을 뽑음

Step 4 : implement an acceptance/rejection criterion algorithm

acceptance/rejection을 결정하기 위해 metropolis algorithm이 활용될 수 있음

기준에 충족하지 못했다고 바로 리젝하는 것 대시 re-evaluated 되어야함. P(x1)/p(x0) > 1

Step5 : re-evaluate the initially rejected candidates

. P(x1)/p(x0) > u u = [0,1]사이의 균등분포로부터 랜덤 추출됨

Step6 : 3-5를 최대 반복수 혹은 수렴할 때까지 반복함

장점

Analytical form 없이도 목표 분포를 다룰 수 있음

도입하기에 쉽고 편리함

목표 분포에 대한 가정이나 근사치에 의존하지 않고

단점

느리고 비효율적, 수렴하기까지 많은 샘플과 시간이 필요

수렴을 평가하기가 까다롭고 잘못된 결과를 가져올 수 있음

순차적으로 샘플링하기 때문에 병렬화가 어렵다