Homework

Atlas42

March 2022

1 Ý nghĩa tham số radius, min sample trong thuật toán dbscan? Nếu chỉ số lớn, nhỏ ảnh hưởng thế nào tới thuật toán?

Ý nghĩa tham số radius, min sample trong thuật toán dbscan:

Radius mang ý nghĩa là bán kính của đường tròn quanh các điểm dữ liệu để kiểm tra density. Min sample mang ý nghĩa là số điểm tối thiểu trong mỗi đường tròn quay quanh một điểm dữ liệu để điểm đó được coi là một Core point

Sự điều chỉnh của radius và min sample có ảnh hưởng rất lớn đến với kết quả khi áp dụng thuật toán dbscan, radius nhỏ hoặc lớn có thể gây ảnh hưởng tới bán kính của đường tròn, làm cho phạm vi bao phủ nhỏ đi hoặc lớn lên tương đương với việc phân loại các điểm border và noise, min sample lớn hoặc nhỏ ảnh hưởng tới việc phân loại các điểm dữ liệu thành core point.

2 Biến đổi lại và so sánh ba thuật toán: kmean, GMM, dbscan. Khi nào nên sử dụng thuật toán nào? cho ví dụ??

K - means:

Step 1 in an iteration of K-means is to minimize distortion measure Jwrt cluster membership variables r_{nk}

$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\|x_n - \mu_k\right\|^2$

Terms for different data points x_n are independent, for each data point set r_{nk} to minimize: $\sum_{k=1}^{K} r_{nk} \left\|x_n - \mu_k\right\|^2$

Simply set $r_{nk} = 1$ for the cluster center μ_k with smallest distance

Step 2: fix r_{nk}, minimize Jwrt the cluster centers μ_k

$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\|x_n - \mu_k\right\|^2$ switch order of sums

So we can minimize wrt each μ_k separately

Take derivative, set to zero:

$2\sum_{k=1}^{K} r_{nk} \left\|x_n - \mu_k\right\| = 0$

$\iff \mu_k = \frac{\sum_{n=1} r_{nk} x_n}{\sum_{n=1} r_{nk}}$

Rinse and repeat until convergence

GMM:

$$\ell(\theta) = \sum_{n=1}^{N} \log(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \sigma_k))$$

$$\iff \frac{\partial \ell(\theta)}{\partial \mu_k} = \sum_{n=1}^{N} \frac{1}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \sigma_k)} \cdot (\pi_k \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp - \frac{(x_n - \mu_k)^2}{2\sigma^2})'$$

$$= \sum_{n=1}^{N} \frac{1}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \sigma_k)} \cdot \pi_k \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot (\exp - \frac{(x_n - \mu_k)^2}{2\sigma^2})'$$

$$= \sum_{n=1}^{N} \frac{1}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \sigma_k)} \cdot \pi_k \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp - \frac{(x_n - \mu_k)^2}{2\sigma^2} \cdot (-\frac{(x_n - \mu_k)^2}{2\sigma^2})'$$

$$= \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(x_n|\mu_k, \sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \sigma_k)} \cdot \frac{1}{2\sigma^2} \cdot ((x_n - \mu_k)^2)'$$

$$= \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(x_n|\mu_k, \sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \sigma_k)} \cdot \frac{1}{2\sigma^2} \cdot 2(x_n - \mu_k) \cdot (x_n - \mu_k)'$$

$$= \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(x_n|\mu_k, \sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \sigma_k)} \cdot {\sigma^2}_k^{-1}(x_n - \mu_k)$$

$$= \sum_{n=1}^{N} \gamma(z_{nk}) \cdot {\sigma^2}_k^{-1}(x_n - \mu_k)$$

Setting derivative to 0, and multiply by σ_k

$\sum_{n=1}^{N} \gamma(z_{nk})\mu_k = \sum_{n=1}^{N} \gamma(z_{nk})x_n$

$\iff \mu_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})\mu_k x_n$, where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$

DBSCAN:

- Clusters formed are arbitrary in shape and may not have same feature size.

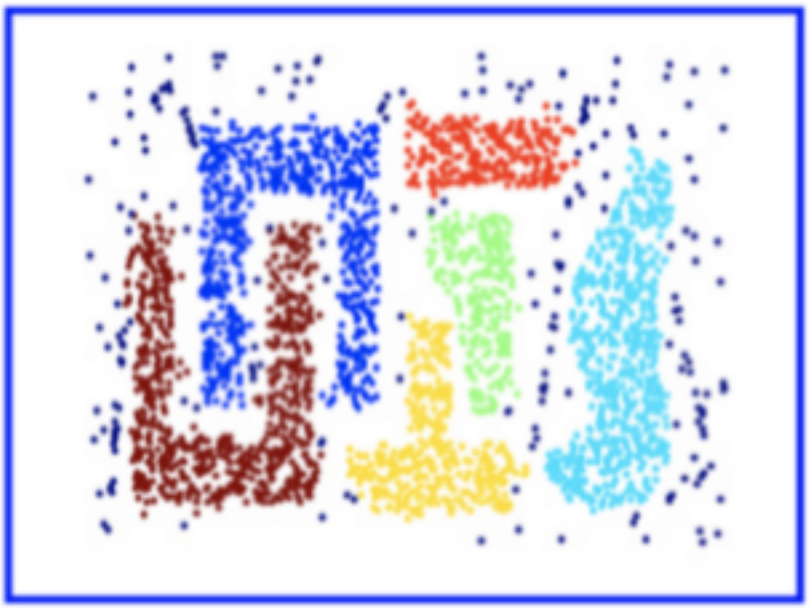
- Number of clusters need not be specified.

- DBSCan Clustering can not efficiently handle high dimensional datasets.

- DBScan clustering efficiently handles outliers and noisy datasets.

- DBScan algorithm locates regions of high density that are separated from one another by regions of low density.

- It requires two parameters : Radius(R) and Minimum Points(M)



The above image depicts a more traditional clustering method that does not account for multi-dimensionality. Whereas the below image shows how DBSCAN can contort the data into different shapes and dimensions in order to find similar clusters.

Kmeans:

- Clusters formed are more or less spherical or convex in shape and must have same feature size.

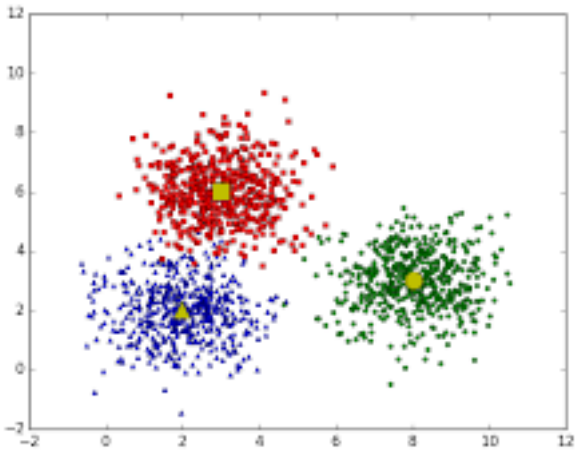
- K-means clustering is sensitive to the number of clusters specified.

- K-means Clustering is more efficient for large datasets.

- K-means Clustering does not work well with outliers and noisy datasets.

- In the domain of anomaly detection, this algorithm causes problems as anomalous points will be assigned to the same cluster as “normal” data points.

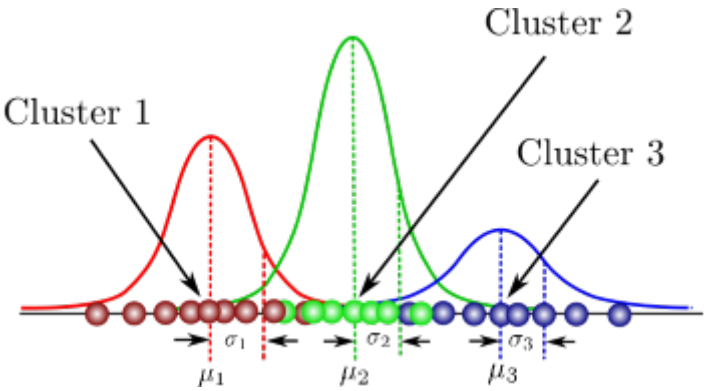
- It requires one parameter : Number of clusters (K)



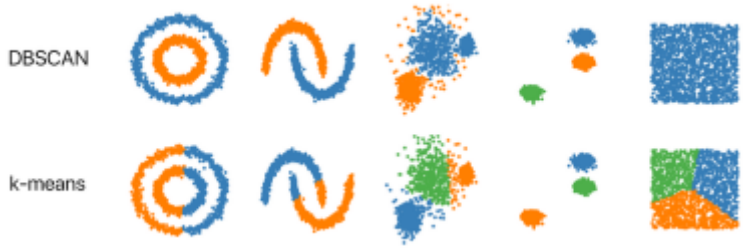
GMM:

- Used for generative unsupervised learning or clustering

- Similar to the k-means algorithm



Kmeans vs DBSCAN



GMM vs Kmeans:

K means only calculates conventional Euclidean distance. i.e K-means calculates distance and GM calculates weights.This means that the k-means algorithm gives you a hard assignment: it either says this is going to be this data point is a part of this class or it's a part of this class. In a lot of cases we just want that hard assignment but in a lot of cases it's better to have a soft assignment. Sometimes we want the maximum probability like: This is going to be 70% likely that it's a part of this class but we also want the probability that it's going to be a part of other classes. It is a list of probability values that it could be a part of multiple distributions, it could be in the middle, it could be 60% likely this class and 40% likely of this class. That's why we incorporate the standard deviation.