Melbourne Bioinformatics

BIOINFORMATICS + DATA SERVICES + INFRASTRUCTURE, FOR LIFE SCIENCES TODAY

THE UNIVERSITY OF MELBOURNE

COMP90014

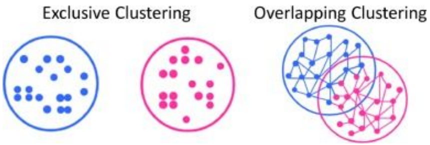
Algorithms for Bioinformatics

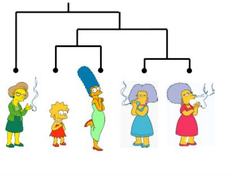
Week 10B: Unsupervised Learning - Clustering II

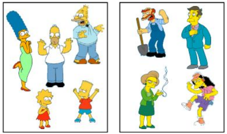
Machine learning: clustering

1. Clustering approaches
2. Density-based clustering
3. Divisive clustering

Clustering approaches







Exclusive

- Data points belong to only one cluster

Overlapping

- Data points may belong to many clusters

Hierarchical

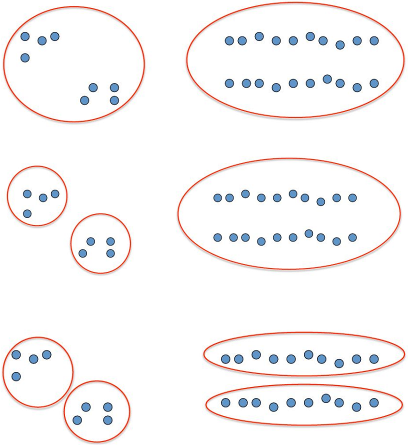
- Assign points to “nested” clusters
- Get all possible clusters for given metric

Partitional

- Split points into “flat” independent clusters
- How many?

Images: Yohoshiva Basaraboyina via medium.com; Eamonn Keogh, Brandeis University

Ingredients for clustering

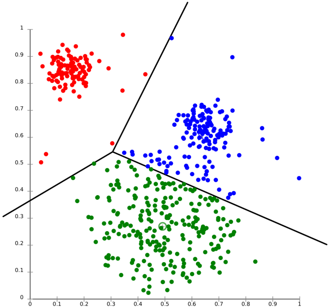


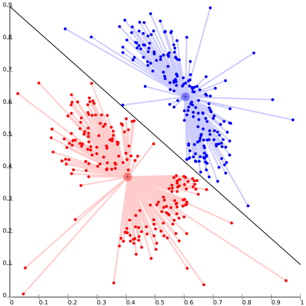
1. similarity metric
   - *e.g.* Euclidean distance
2. a function to evaluate the quality of the clusters
3. clustering algorithm

🤖 clustering is subjective
e.g. how many clusters?
- fixed k clusters
- find the best k to optimize a function

Clustering approaches by cluster definition

1. Centroid-based (*k*-means, *k*-medoids)



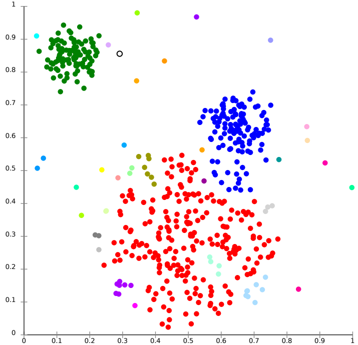


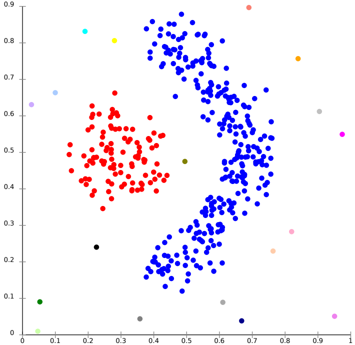
Notion of clusters: Voronoi tessellation/diagram

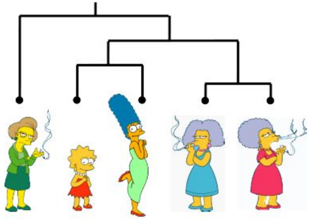
Chire via Wikimedia Commons (1) (2)

Clustering approaches by cluster definition

2. Connectivity-based (hierarchical)





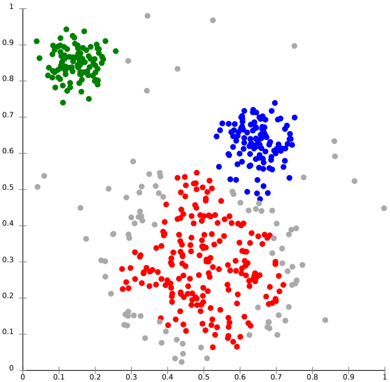


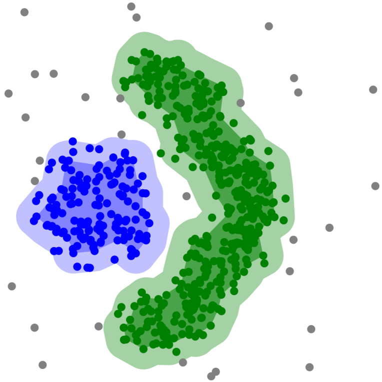
Notion of clusters: cut dendrogram at some depth

Chire via Wikimedia Commons (1) (2)

Clustering approaches by cluster definition

3. Density-based (DBSCAN, OPTICS)





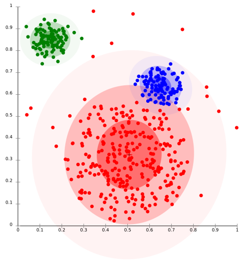
Notion of clusters: connected regions of high density

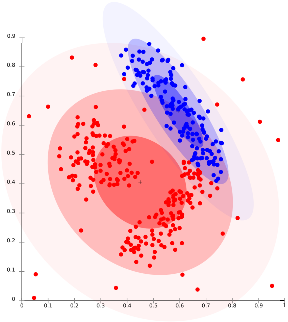
在一定深度上剪切树枝图

Chire via Wikimedia Commons (1) (2)

Clustering approaches by cluster definition

4. Distribution-based (Mixture Models)





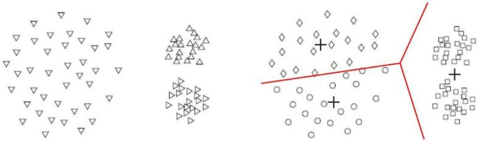
Notion of clusters: distributions on features

5. Network-based

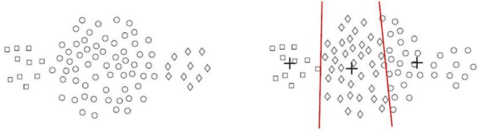
Notion of clusters: graph connectivity

Chire via Wikimedia Commons (1) (2)

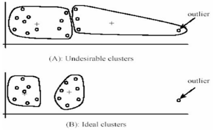
When *k*-means fails

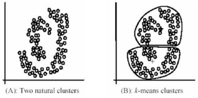


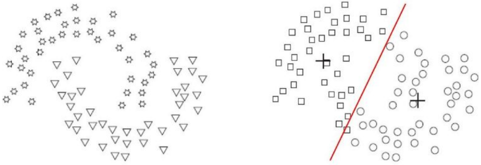
(a) Original points. (b) Three K-means clusters.



(a) Original points. (b) Three K-means clusters.







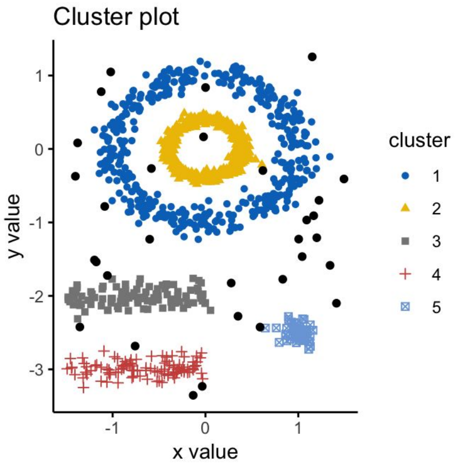
(a) Original points. (b) Two K-means clusters.

Anil Yadav via [slideshare]

Machine learning: clustering

1. Clustering approaches
2. Density-based clustering
3. Divisive clustering

Concepts

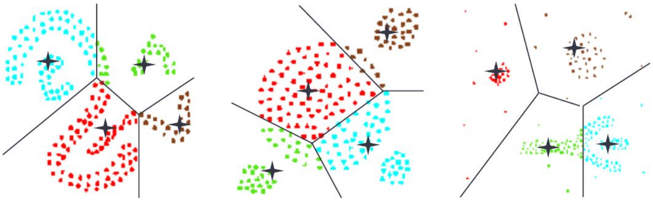


Sagi Shaier via towardsdatascience.com

- clusters are ***dense regions in the data space***, separated by regions of lower object density
- for any point in a cluster, the local point density around that point has to exceed some threshold (ε)
- the set of points from one cluster is spatially connected

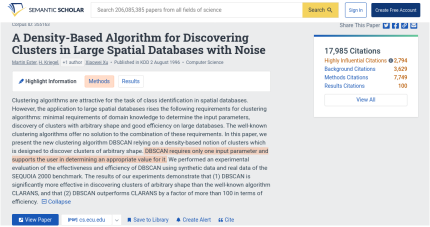
聚类是数据空间中的密集区域，它们被低对象密度的区域所分隔。
对于簇中的任意点，围绕该点的局部点密度必须超过某个阈值（ε）。
一个簇中的点集在空间上是相互连接的。

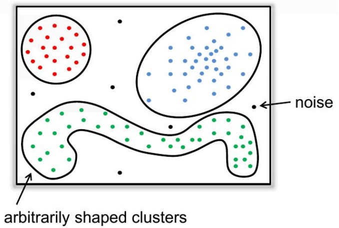
Density-based clustering



- When k-means fails ($K = 4$)
- DBSCAN:
  - discover clusters of arbitrary shape
  - handle noise and outliers

DBSCAN





Ester *et al.*, 1996. KDD-96 Proceedings

- first density-based clustering algorithm
- one the most widely used/cited clustering algorithms

Intuition:

- ***a cluster is a region of high density***
- noise points lie in regions of low density

We need to:

- define neighbourhood of a data point
- define high density

Definitions

对于点p，其ε-邻域包括了所有与p的距离不超过ε的点q。

ε-neighbourhood: objects within a radius ε of an object.

$$N_\varepsilon(p) : \{q \mid d(p,q) \leq \varepsilon\}$$

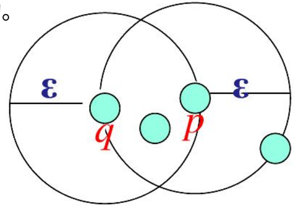
ε: **input parameter.**

High-density: ε-neighbourhood of an object contains at least minpts of objects.

minpts: **input parameter.**

如果一个对象的ε-邻域至少包含了minpts个对象，那么这个邻域被认为是高密度的。

ε-neighbourhood of *p* and *q*: Density of *p* is “high” (minpts $= 4$); Density of *q* is “low” (minpts $= 4$)

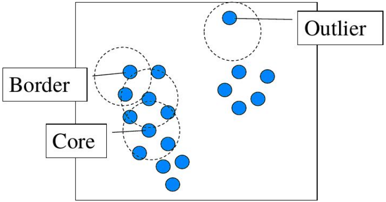


展示了两个点p和q及它们各自的ε-邻域。点p的密度被认为是“高”的，因为它的ε-邻域至少包含了minpts数量（4个）的点。而点q的密度被认为是“低”的，尽管它的ε-邻域和点p是一样的，但是它自己并没有被足够数量的其他点所包围

Shah, 2012 10.1109/NUICONE.2012.6493211

Definitions

Given ε and minpts, categorize the objects into three exclusive categories.



Core point: It has more than minpts objects within ε.

Border point: It has fewer than minpts within ε, but is in the neighbourhood of a core point.

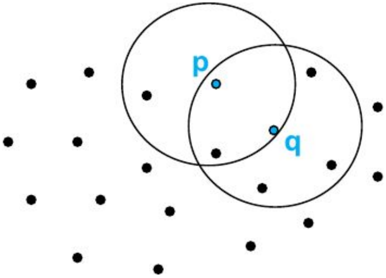
Noise/outlier point: Any point that is not a core point nor a border point.

核心点（Core point）：在一个点的ε邻域内，如果存在超过minpts数量的点，那么这个点就被称为核心点。这意味着核心点周围有足够的邻近点，从而形成一个稠密区域。

边界点（Border point）：如果一个点在核心点的ε邻域内，但其自己的ε邻域内的点数少于minpts，那么它就是一个边界点。边界点位于核心区域的边缘，而不足以自成一个稠密区域。

噪声/离群点（Noise/outlier point）：不属于核心点或边界点的任何点都被认为是噪声或离群点。这些点既没有足够的邻近点构成稠密区域，也不是任何核心区域的一部分。

Direct density reachability



minpts $= 5$

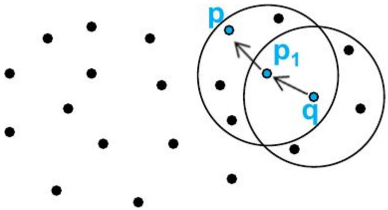
An object p is *directly density-reachable* from object q if:

- q is a core object and
- p is in its ε-neighborhood

- is p directly density-reachable from q?
- is q directly density-reachable from p?
- Density-reachability is an asymmetric relationship

Khan *et al.*, 2018. 10.1109/CEEICT.2018.8628138

Density reachability



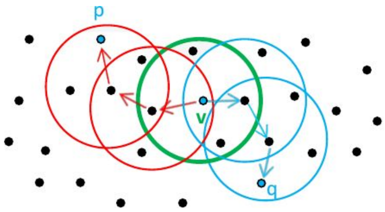
minpts $= 5$

A point p is *density-reachable* from a point q if:

- There is a chain of points $p_1, p_2, \ldots, p_k$, with $p_1 = q$ and $p_k = p$, such that p_{i+1} is directly density-reachable for all $1 < i < k - 1$
- Asymmetric

Khan *et al.*, 2018. 10.1109/CEEICT.2018.8628138

Density connectivity



minpts $= 5$

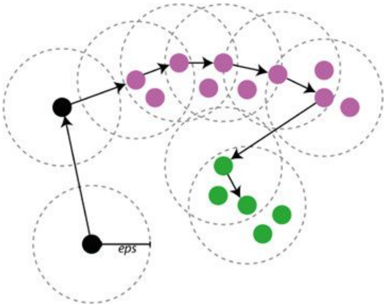
A point p is *density-connected* to a point q if:

- there is a point v, such that both p and q are *density-reachable* from v

- Symmetric

Khan *et al.*, 2018. [10.1109/CEEICT.2018.8628138](https://doi.org/10.1109/CEEICT.2018.8628138)

Cluster definition



Given a data set D of points, parameter ε and minpts:

- a cluster C is a subset of D satisfying two criteria.

Maximality:

- $\forall p, q$ if $p \in C$
  and if q is density-reachable from p,
  then also $q \in C$

Connectivity:

- $\forall p, q \in C$,
  p and q are density-connected

- clusters will contain both core and border points
- noise/outliers:
  - points in D not belonging to any cluster

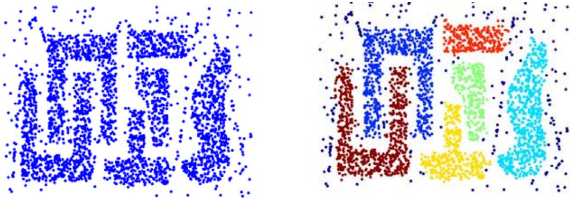
Zielasko *et al.*, 2016. 10.1111/cgf.12881

DBSCAN algorithm

```
Procedure Dbscan(X, ε, minpts):
    foreach unvisited point x ∈ X do
        mark x as visited
        N ← GetNeighbours(x, ε)
        if |N| < minpts then
            mark x as noise
        else
            C ← {x}
            foreach point x′ ∈ N do
                N ← N \ x′
                if x′ is not visited then
                    mark x′ as visited
                    N′ ← GetNeighbours(x′, ε)
                    if |N′| ≥ minpts then
                        N ← N ∪ N′
                if x′ is not yet member of any
                  cluster then
                    C ← C ∪ {x′}
```

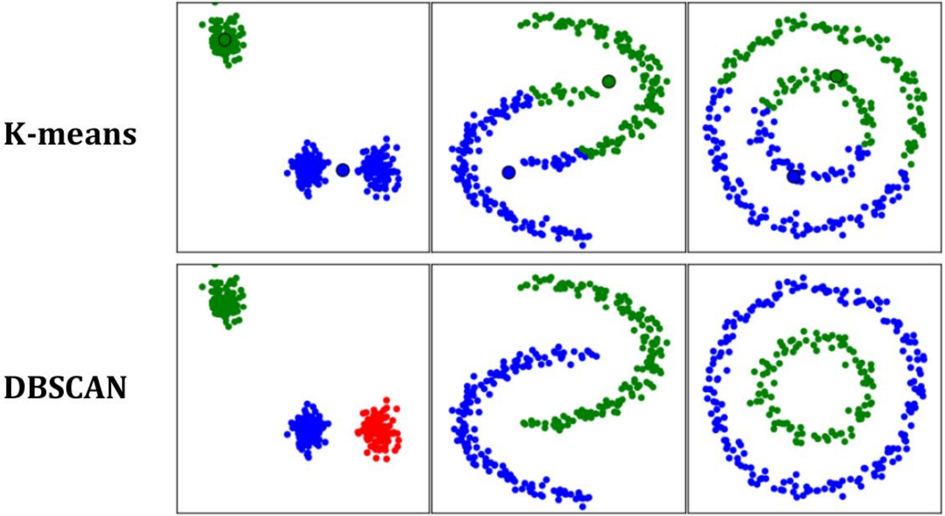
- Input parameters:
  - X points, ε and minpts
- the algorithm proceeds by arbitrarily picking (scanning) up points in the dataset until all points have been visited
- if p is a core point (at least minpts points within a radius of ε) collect all density-reachable points from p and assign to a new cluster
- assign p to noise otherwise
- don't change p's cluster assignment

Complexity and strength of DBSCAN



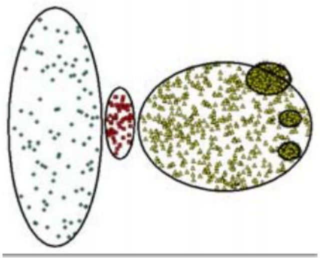
- time complexity:
  - $O(n^2)$ if done naïvely
  - $O(n \times \log n)$ with a spatial index
    - only works in relatively low dimensions
- space complexity: $O(n)$
- can handle arbitrary shapes
- can handle clusters of different sizes
- resistant to noise

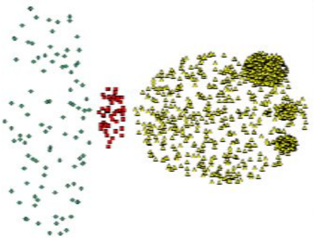
DBSCAN *vs.* *k*-means

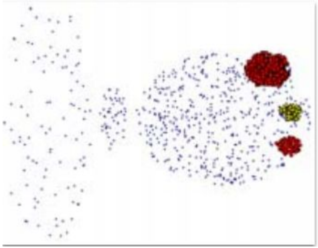


scikit-learn documentation

Weaknesses of DBSCAN







设置ε和minpts可能会很棘手（setting ε and minpts can be tricky）：找到适合所有数据簇的ε值和minpts值可能很难，特别是当簇具有不同密度或大小时

Goal:

- varying densities
- high dimensional data
- overlapping clusters

Different ε configurations:

- setting ε and minpts can be tricky

变化的密度（varying densities）：DBSCAN可能在处理具有不同密度的数据簇时遇到困难。
高维数据（high dimensional data）：在高维空间中，所有点之间的距离都倾向于相似，这会影响DBSCAN算法的性能。
重叠的簇（overlapping clusters）：当不同簇的数据点在空间上重叠时，DBSCAN可能难以正确区分这些簇。

Determining ε

k-距离（k-distance）：

计算每个点的第k近邻的距离（calculate distance of kth nearest neighbor for each point）。
其中k等于minpts减去1（k = minpts - 1）。

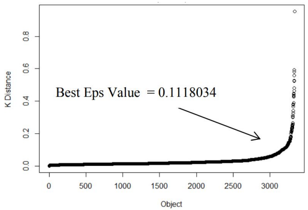
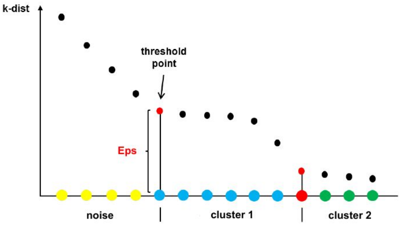


Figure 2 Points sorted by distance to the 3rd nearest neighbor

k-distance: calculate distance of k^{th} nearest neighbor for each point

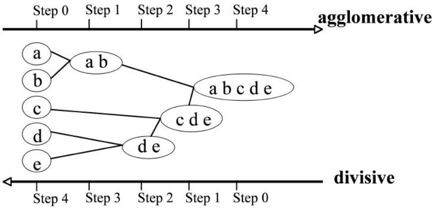
$k = \text{minpts} - 1$



- plot in ascending / descending order
- set ε to the maximum distance before the threshold
- noise points have their k^{th} nearest neighbour at higher distances

Machine learning: clustering

Hierarchical clustering



What are the next clusters to merge?

$$\text{Single linkage:} \quad D_{k,g} = \min(D_{k,i}, D_{k,j})$$
$$\text{Complete linkage:} \quad D_{k,g} = \max(D_{k,i}, D_{k,j})$$
$$\text{Average linkage:} \quad D_{k,g} = \frac{D_{k,i} + D_{k,j}}{2}$$

Agglomerative clustering (bottom-up)

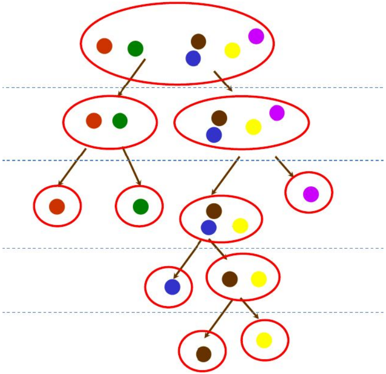
- each data point starts as a single cluster
- join clusters into bigger clusters till we reach one single cluster with all points

Divisive clustering (top-down)

- start with one big cluster
- at each step, split into smaller clusters
- stop at desired number of clusters
- *e.g.* when points are in single clusters

Divisive hierarchical clustering

图片上的文字解释了如何使用任何生成固定数量簇的分区算法来实现分裂式层次聚类。例如，可以使用k-means算法，设置k=2来不断迭代地对簇进行分



- Any partitional algorithm that generates a fixed number of clusters can be used to implement divisive hierarchical clustering
  - *e.g.* k-means, with $k = 2$
  - keep partitioning clusters iteratively

Challenge: use k-means to implement divisive hierarchical clustering on a set of points X.

You can use assume the function `kmeans()` is available (you don't need to implement it yourself).

Hint: start by dividing X into two clusters, then recursively run `kmeans()` on the output until each cluster has only 1 item.

挑战提示是如何使用k-means来实现一个点集X的分裂式层次聚类。可以假设kmeans()函数是可用的，不需要自己实现它。提示是从将X分成两个簇开始，然后递归地对输出进行k-means算法，直到每个簇只有一个元素为止。

Thank you!

Today: Unsupervised Learning - Clustering II

Next time: Supervised Learning