INNOPOLIS UNIVERSITY

Machine Learning

Prof. Adil Khan

Objectives

1. A quick recap of last week
2. What is hierarchical clustering? How is it different from k-means? Why is it called hierarchical learning? How does it work?
   - What is Divisive vs. Agglomerative clustering?
   - What is a linkage method? What are its different kinds?
   - What is a Dendrogram? What do we use it for?
3. What is DBSCAN? How is it different from other clustering methods? How does it work?
   - What are core vs. border vs. noisy points?
   - What is “directly density reachable” vs. “density reachable” vs. “density connected” points?

Recap (1)

Unsupervised Learning

- We have only predictors (a.k.a inputs, or features) but no labels

$$\mathbb{D} = \{(x_i) | x_i \in \mathbb{R}^p\}_{i=1}^{m}$$

The **goal of unsupervised learning** is to model the hidden patterns or underlying structure in the given input data in order to learn about the data

- This *underlying structure* is what we usually refer to as *groups* of data
- And these groups are what we refer to as ***Clusters***


SPORTS

WORLD NEWS

Clusters

- Clusters are defined by a center and a spread
- And we assign a given point x^n to a cluster by computing its similarity to the center of the cluster m_k

Recap (2)

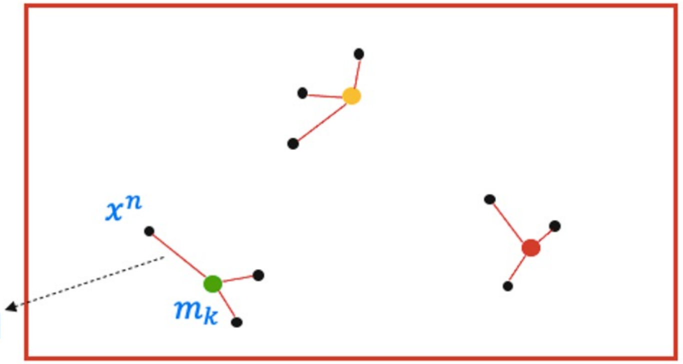
Objective

- Find cluster centers $\{\boldsymbol{m}_k\}_{k=1}^{K}$ and assignments $\{\boldsymbol{r}^{(n)}\}_{n=1}^{N}$ to minimize the sum of squared distances of data points $\{\boldsymbol{x}^{(n)}\}$ to their assigned centers

$\boldsymbol{x}^{(n)} \in \mathbb{R}^D$

$\boldsymbol{m}_k \in \mathbb{R}^D$

$\boldsymbol{r}^{(n)} \in \mathbb{R}^K$



$\boldsymbol{r}^{(n)} = [0, \cdots, 1, \cdots, 0]^T$

$$\min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} J(\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}) = \min_{\{\mathbf{m}_k\},\{\mathbf{r}^{(n)}\}} \sum_{n=1}^{N}\sum_{k=1}^{K} r_k^{(n)} ||\mathbf{m}_k - \mathbf{x}^{(n)}||^2$$

where $r_k^{(n)} = \mathbb{I}[\mathbf{x}^{(n)}$ is assigned to cluster $k]$, i.e., $\mathbf{r}^{(n)} = [0, .., 1, .., 0]^\top$

K-means Clustering

- Initialization: randomly initialize cluster centers
- The algorithm then iteratively alternates between the following two steps
  - Assignment Step: assign each data point to the closest center
  - Refitting Step: move each cluster center to the mean of the data points assigned to it

The objective is solved by alternating minimization

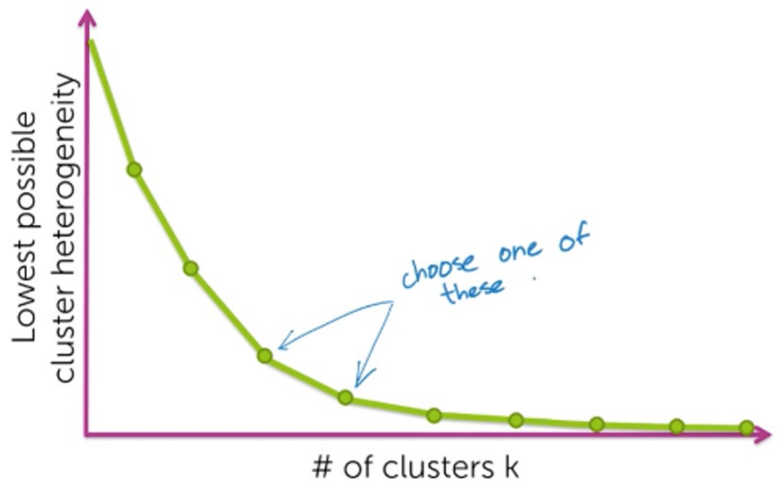
Recap (3)

k-means is sensitive to center initialization and we can get completely different solutions, by converging to a local mode

It can be improved by using a specific way of choosing centers

K-means++

Choosing K – The Elbow Method

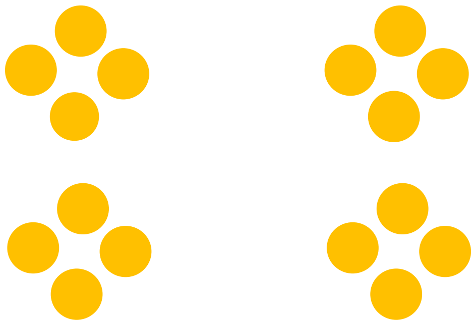


Issues with K-means

1. Not knowing the optimum value of K
2. Also, k-means clustering does not work well when
   - We want to discover clusters of varying sizes, densities and shapes
   - We do not want to include noisy points (outliers) into any clusters

Hierarchical Clustering

How Many Clusters do You See?

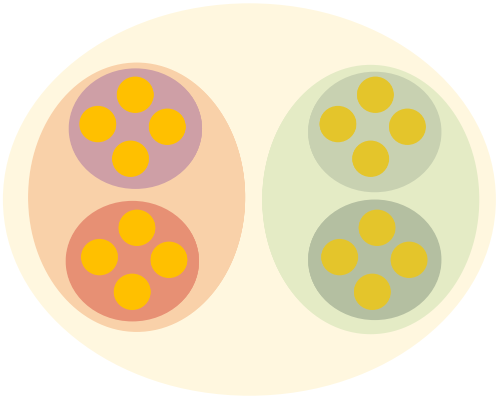


- It all depends on how you are looking at your data, even better – at what granularity do you want to look at your data

- Are you looking for high-level effect or fine-grained details?

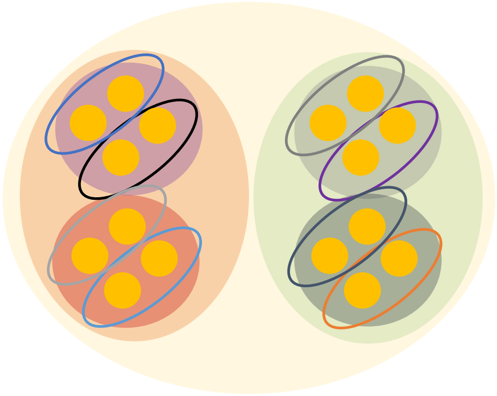
- Thus instead of finding a certain number of clusters, we can change our objective – “find a hierarchy of structure”

Ways of Discovering Hierarchy



- Top-down or Divisive Clustering
  - Start with all items in one cluster,
  - Split recursively

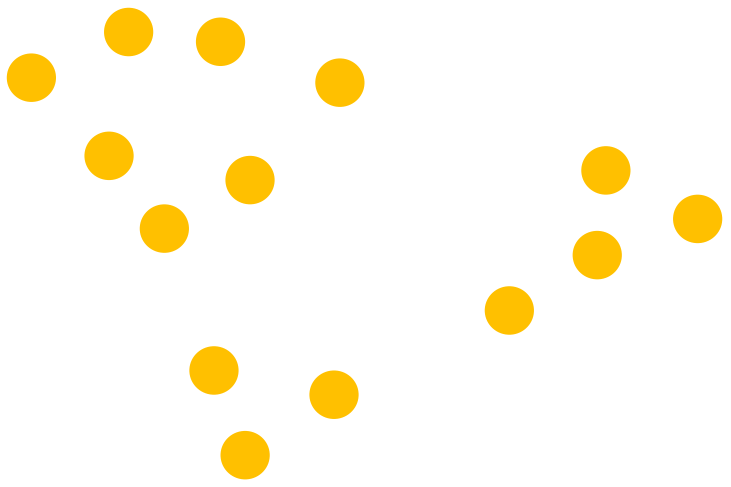
Ways of Discovering Hierarchy (2)



- Bottom-up or Agglomerative Clustering
  - Start with singeltons,
  - Merge clusters using some criteria

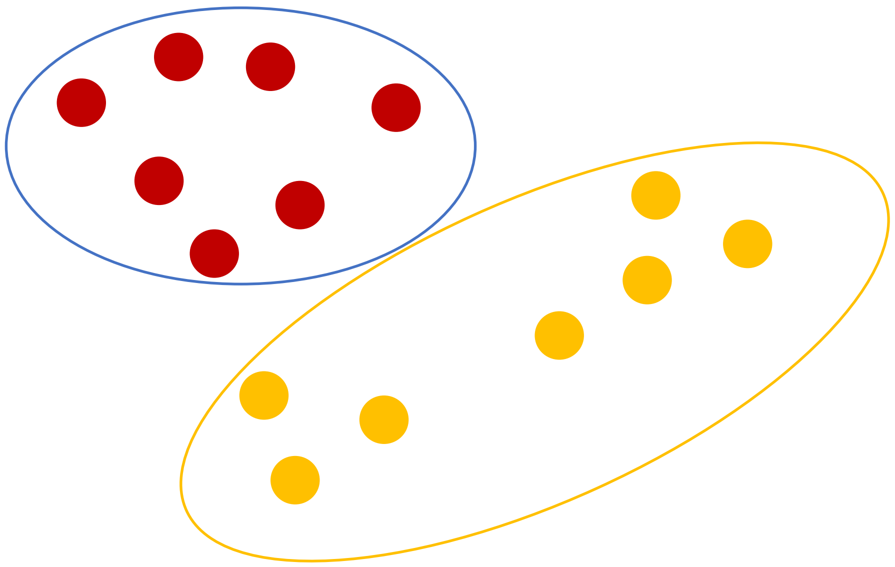
Top-down Hierarchical Clustering (2)

- Simple solution: run k-means recursively



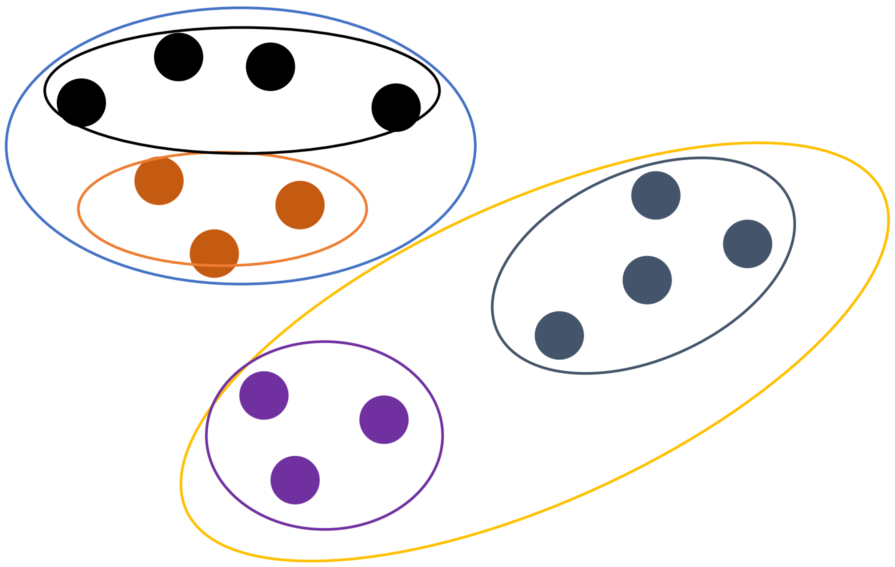
Top-down Hierarchical Clustering (3)

- Simple solution: run k-means recursively



Top-down Hierarchical Clustering (4)

- Simple solution: run k-means recursively



Bottom-up Hierarchical Clustering

- Agglomerative clustering algorithms begin with every observation representing a singleton cluster

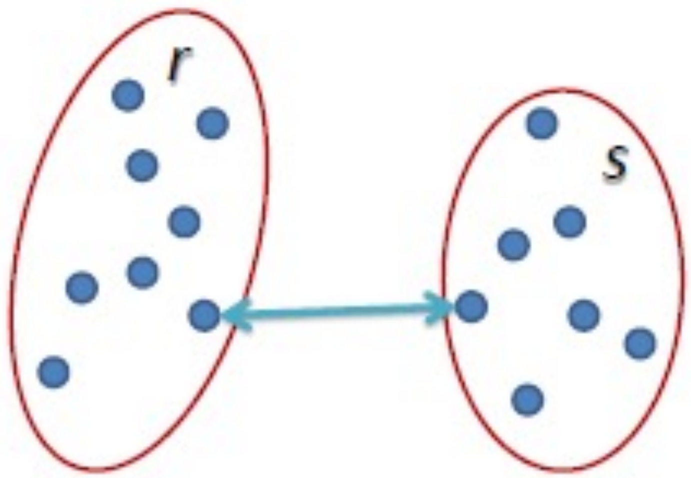
- At each step the closest two (least dissimilar) clusters are merged into a single cluster

- Therefore, a measure of dissimilarity between two clusters (groups of observations) must be defined.

Distance Between Clusters

- Three most common choices are:
  1. Single linkage:

     Distance between the two most similar members of each cluster



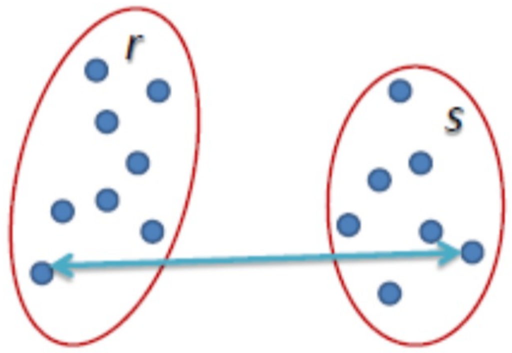
$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$

Distance Between Clusters (2)

- Three most common choices are:
  1. Single linkage:

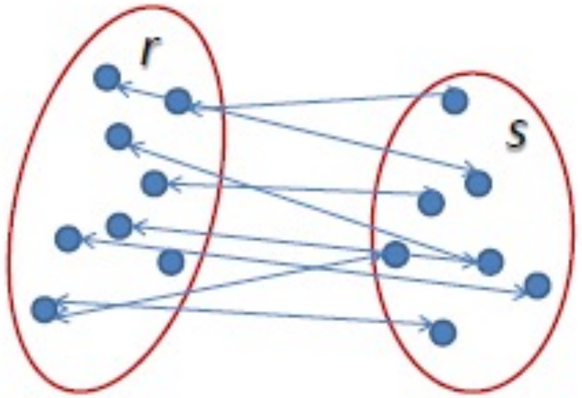
     Distance between the two most similar members of each cluster
  2. Complete linkage:

     Distance between the two most dissimilar members of each cluster



$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

Distance Between Clusters (3)



$$L(r,s)=\frac{1}{n_r n_s}\sum_{i=1}^{n_r}\sum_{j=1}^{n_s}D(x_{ri},x_{sj})$$

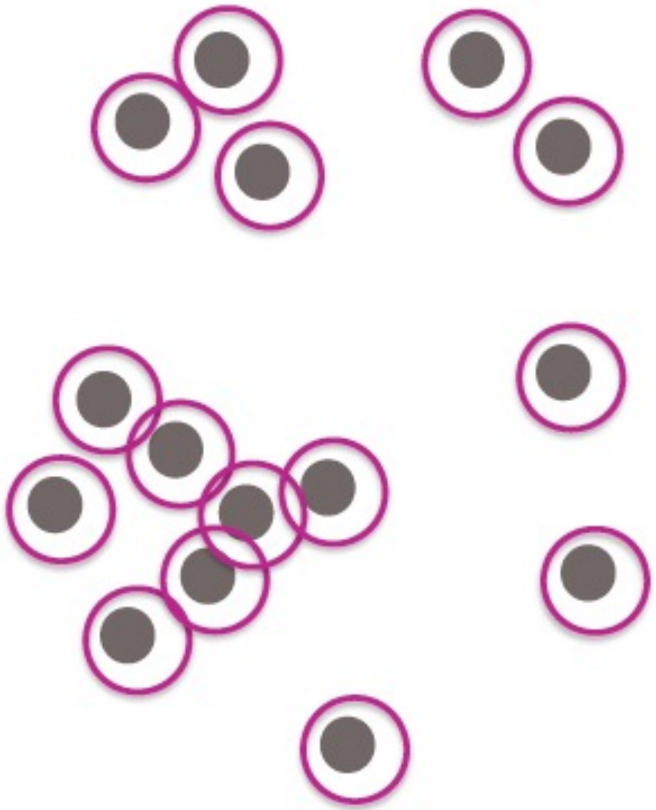
- Three most common choices are:
  1. Single linkage:
     Distance between the two most similar members of
  2. Complete linkage:
     Distance between the two most dissimilar members
  3. Average linkage:
     Average distance between the members of each cluster

Distance Between Clusters (Summary)

- Three most common choices are:
  1. Single linkage:
     Distance between the two most similar members of each cluster
  2. Complete linkage:
     Distance between the two most dissimilar members of each cluster
  3. Average linkage:
     Average distance between the members of each cluster

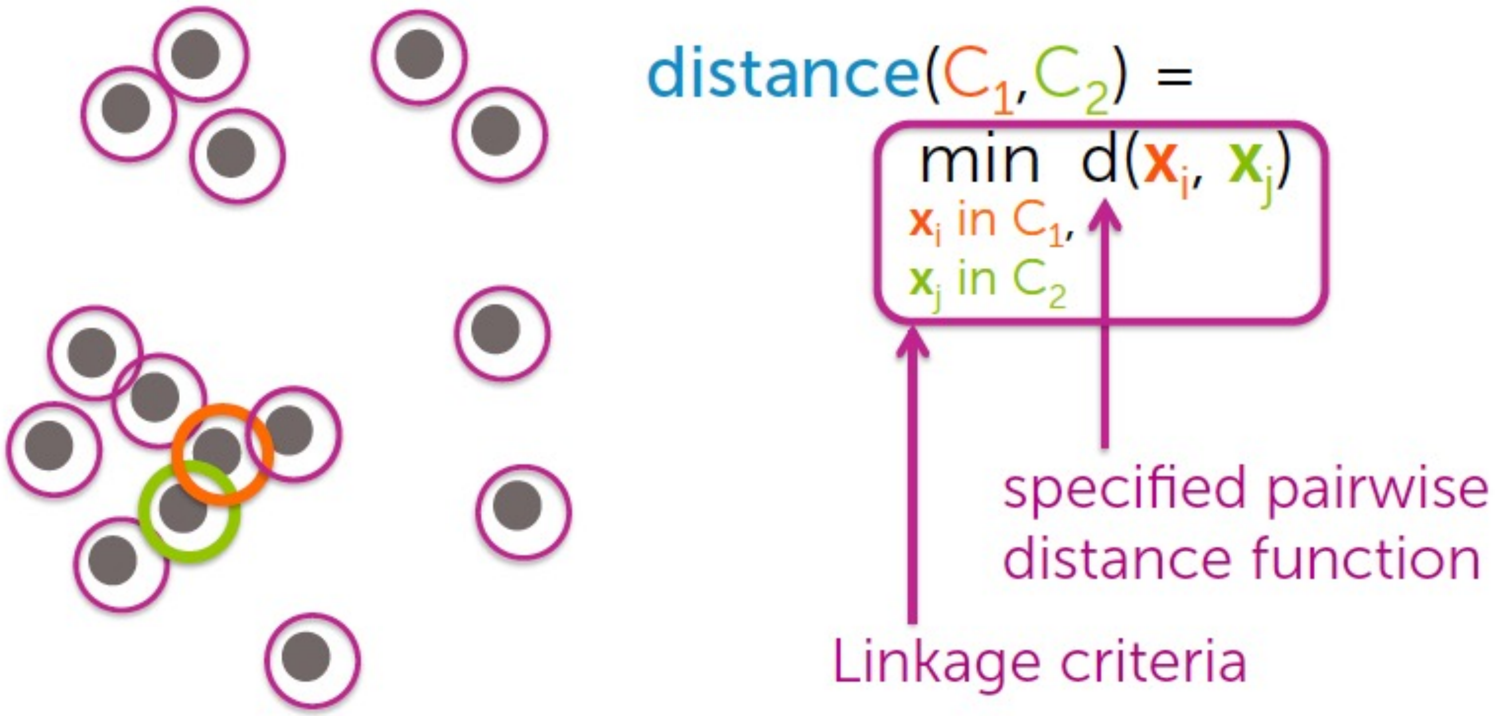
Agglomerative Clustering - Example

1. Make each data point a single cluster



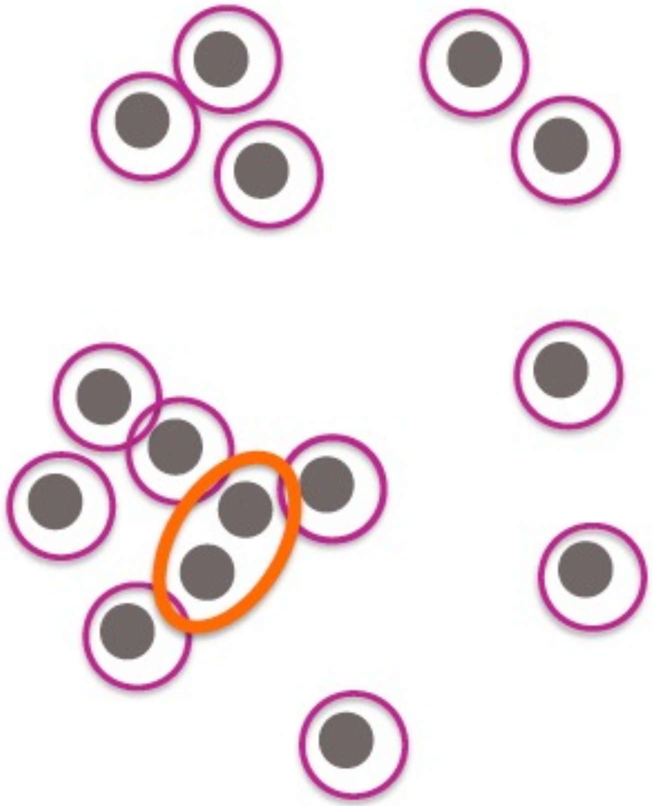
Agglomerative Clustering - Example

2. Compute distance (Single Linkage) between clusters:



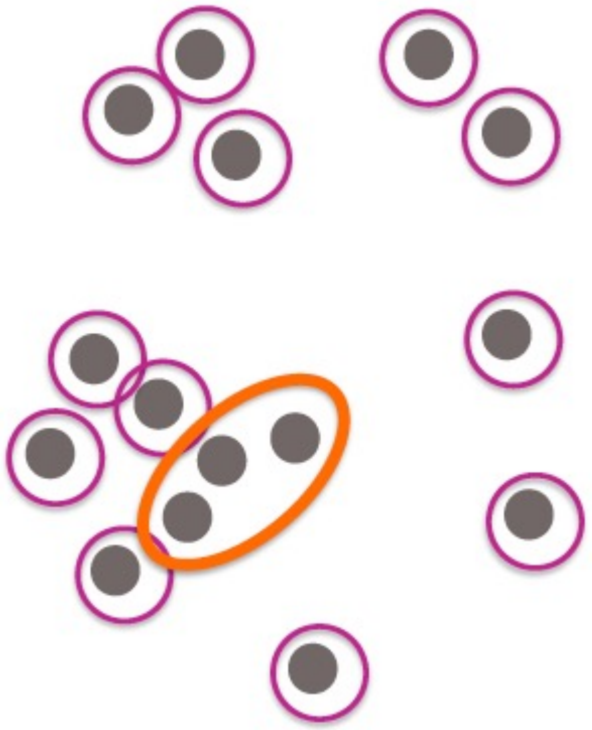
Agglomerative Clustering - Example

3. Merge the two closest clusters



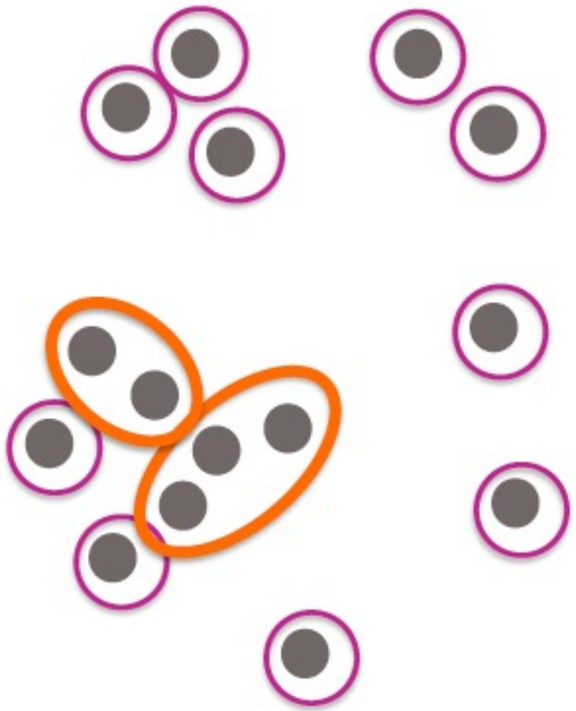
Agglomerative Clustering - Example

4. Repeat step (3) until all points are in one cluster



Agglomerative Clustering - Example

4. Repeat step (3) until all points are in one cluster



Agglomerative Clustering - Example

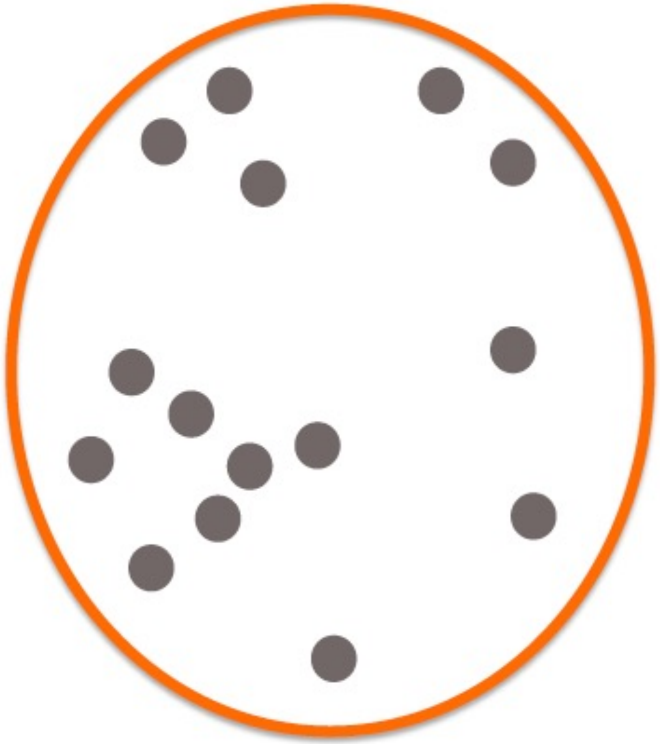
4. Repeat step (3) until all points are in one cluster

Agglomerative Clustering - Example

4. Repeat step (3) until all points are in one cluster

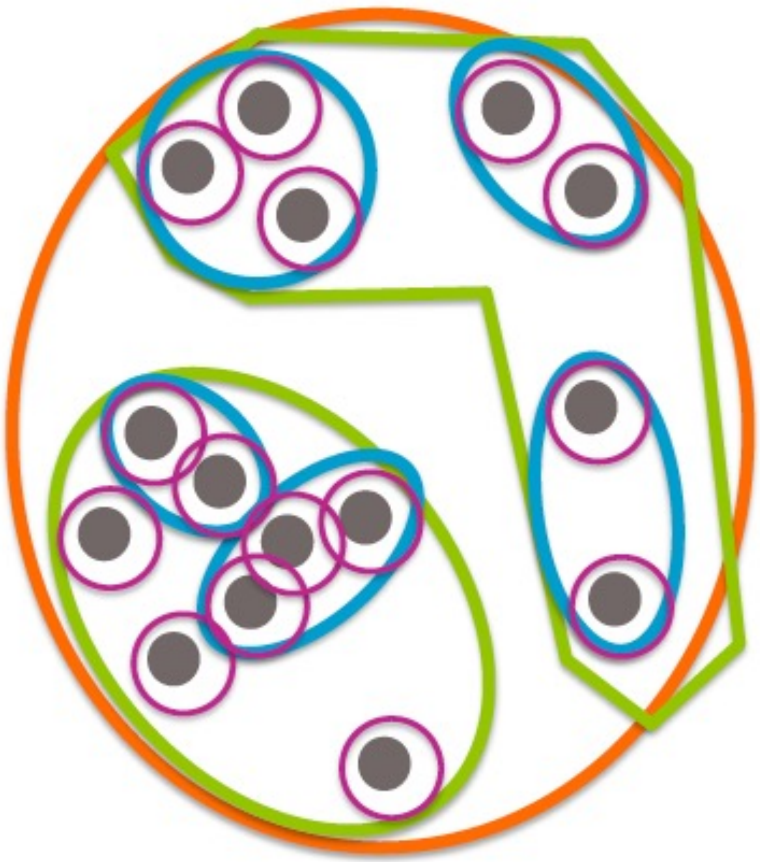
Agglomerative Clustering - Example

4. Repeat step (3) until all points are in one cluster



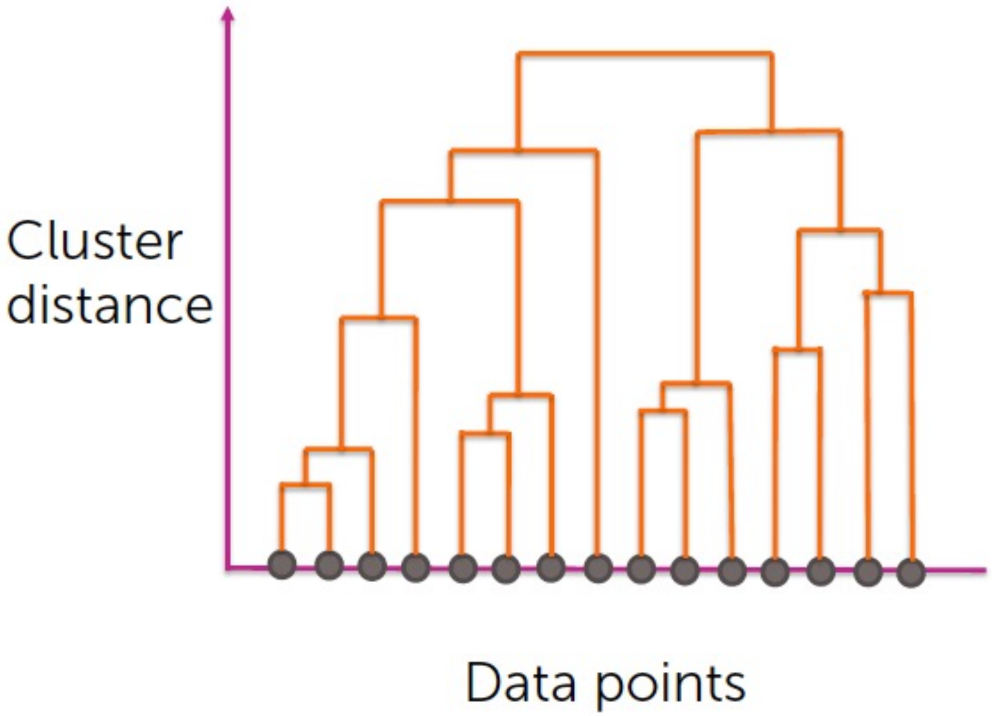
Agglomerative Clustering - Example

- Discovered Hierarchy

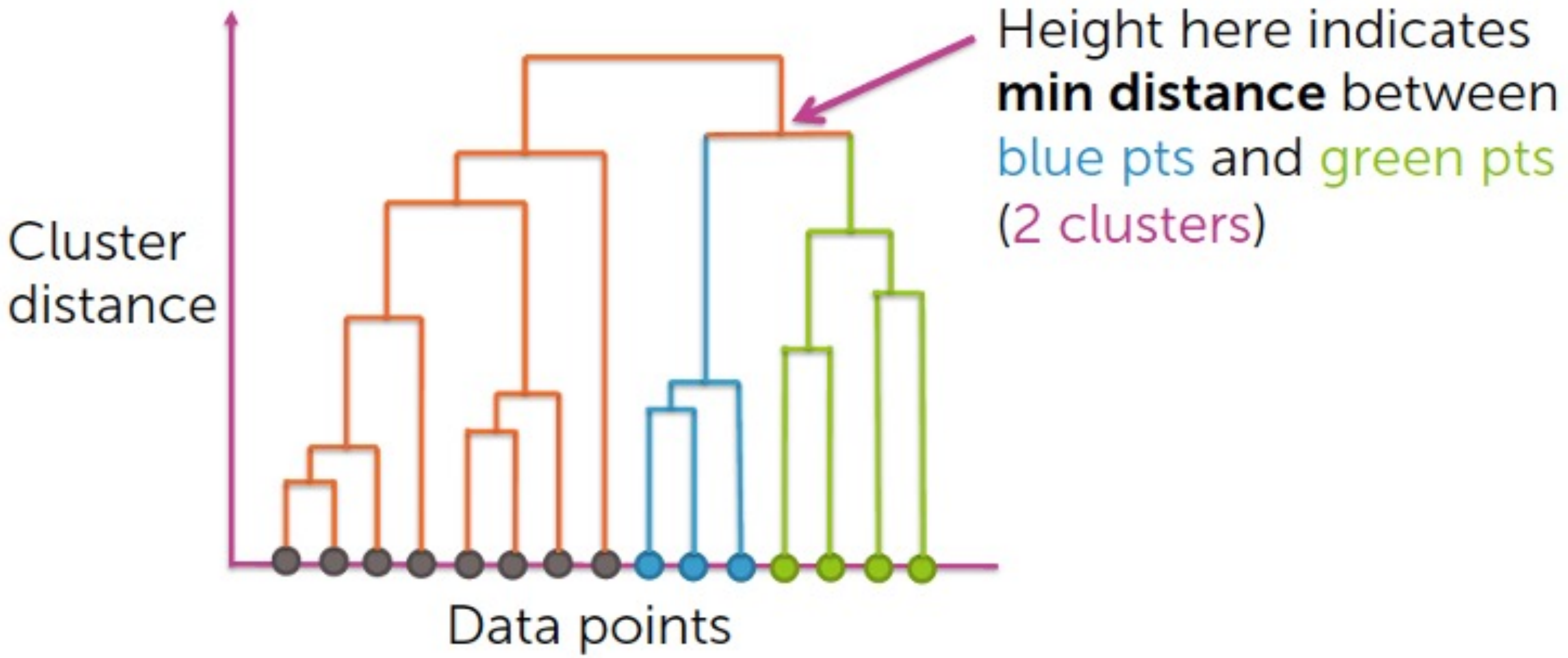


Hierarchy Can Be Viewed as a Tree

- Dendrogram
  - X axis shows data points (ordering is such that it makes the visualization better)
  - Y-axis shows distance between pair of clusters

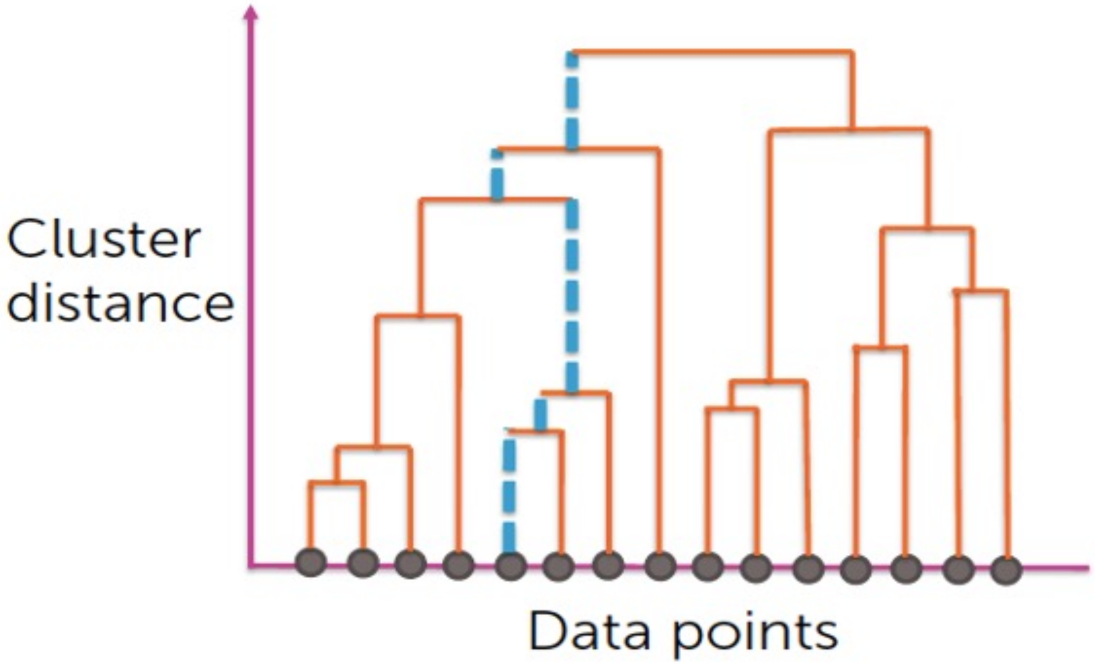


Dendrogram


Height here indicates
min distance between
blue pts and green pts
(2 clusters)
Cluster
distance
Data points

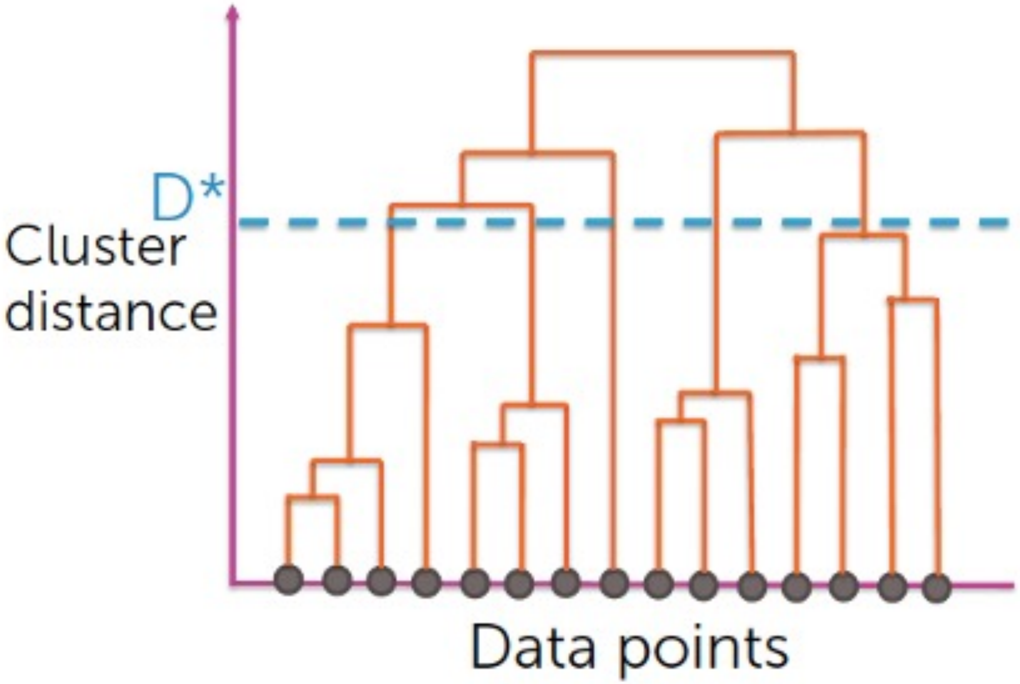
Dendrogram (2)

Path shows all the clusters to which a data point belongs to, and the order in which the clusters merge



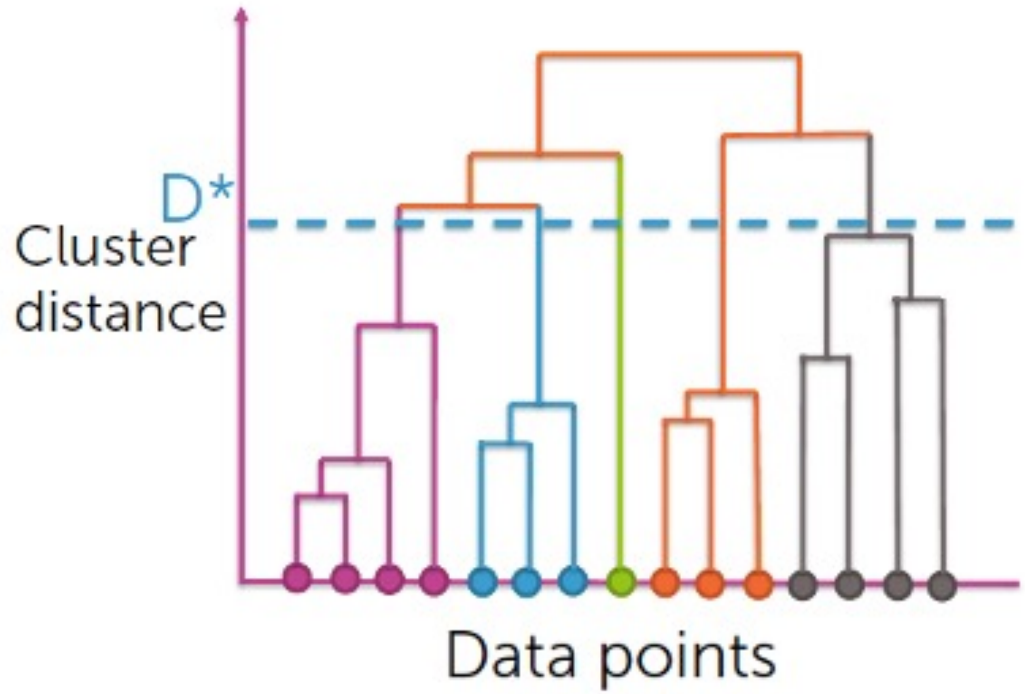
Using Dendrogram

- Choose a distance D^* at which to cut the dendrogram

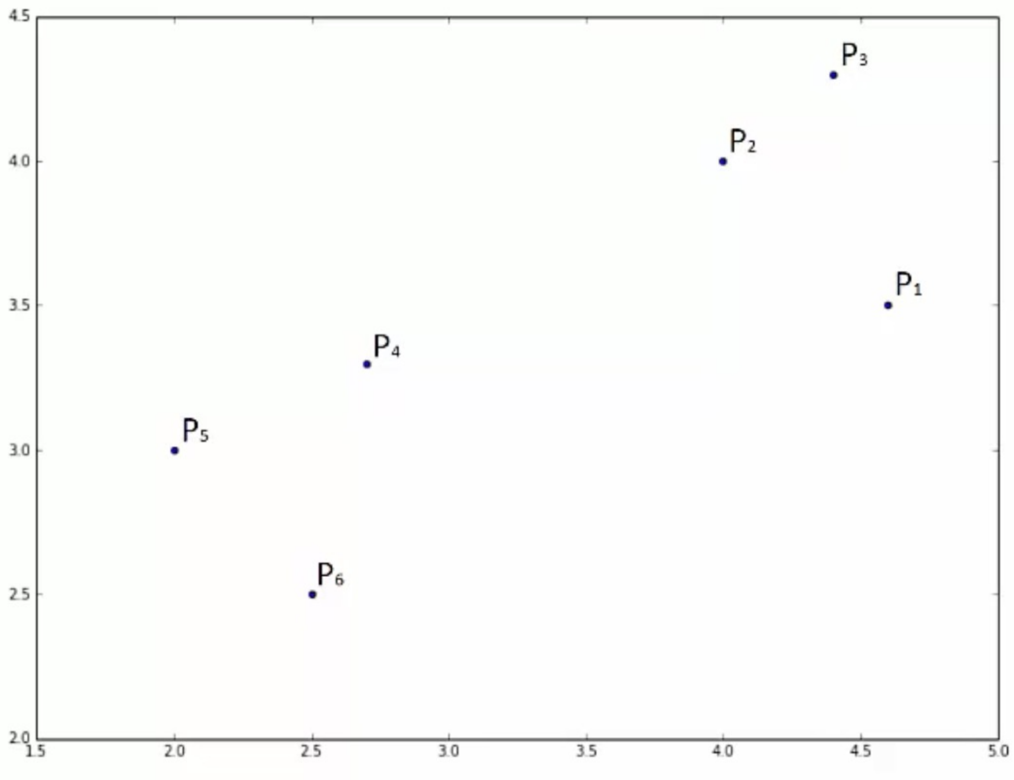


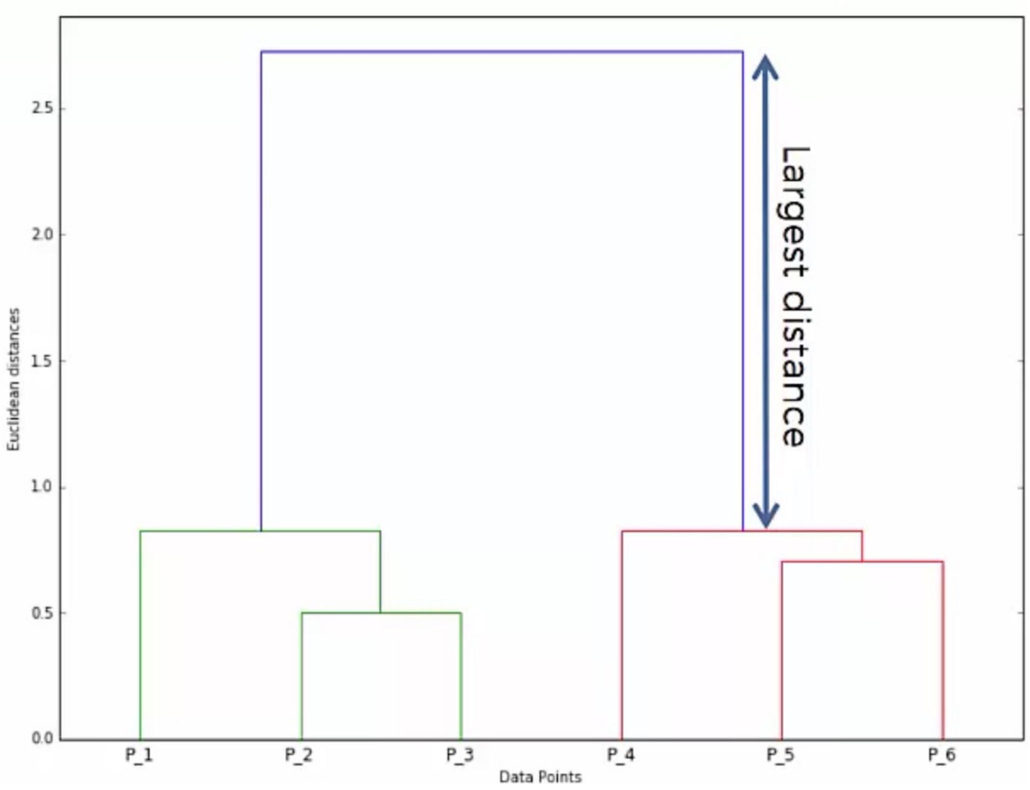
Using Dendrogram (2)

- Every branch that crosses D^* becomes a cluster in the final outcome



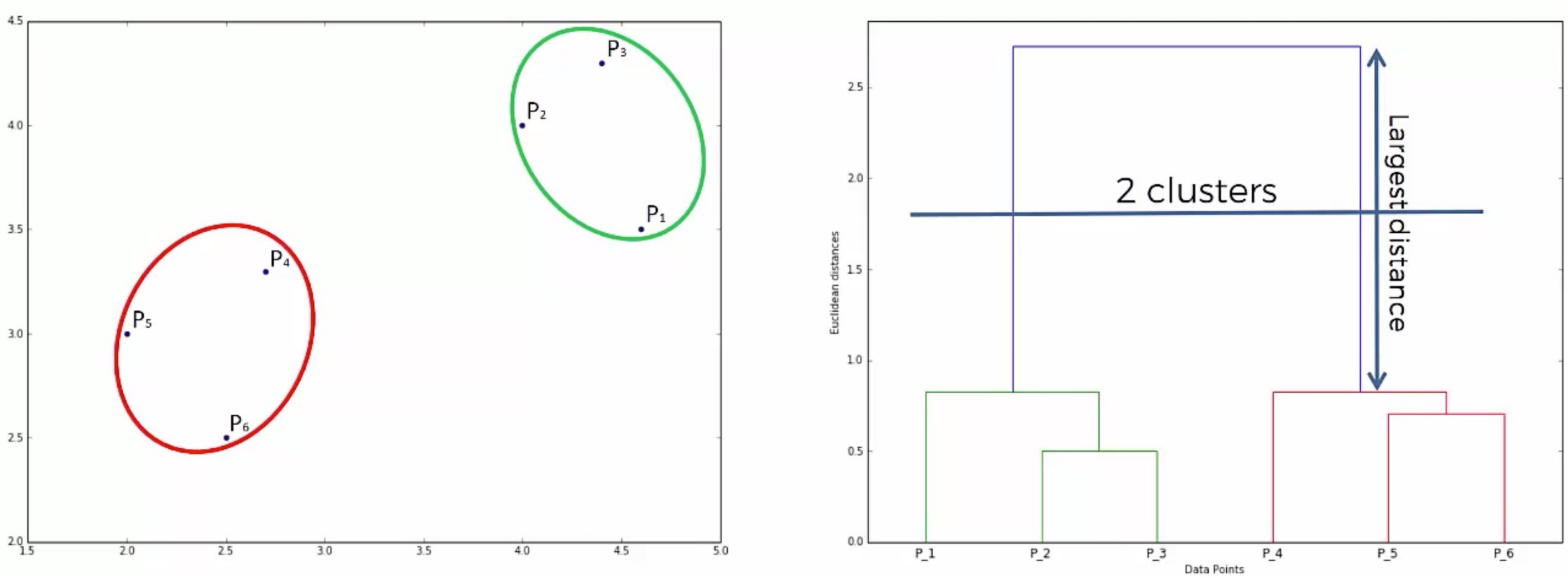
Optimal Cut (Suggestion)



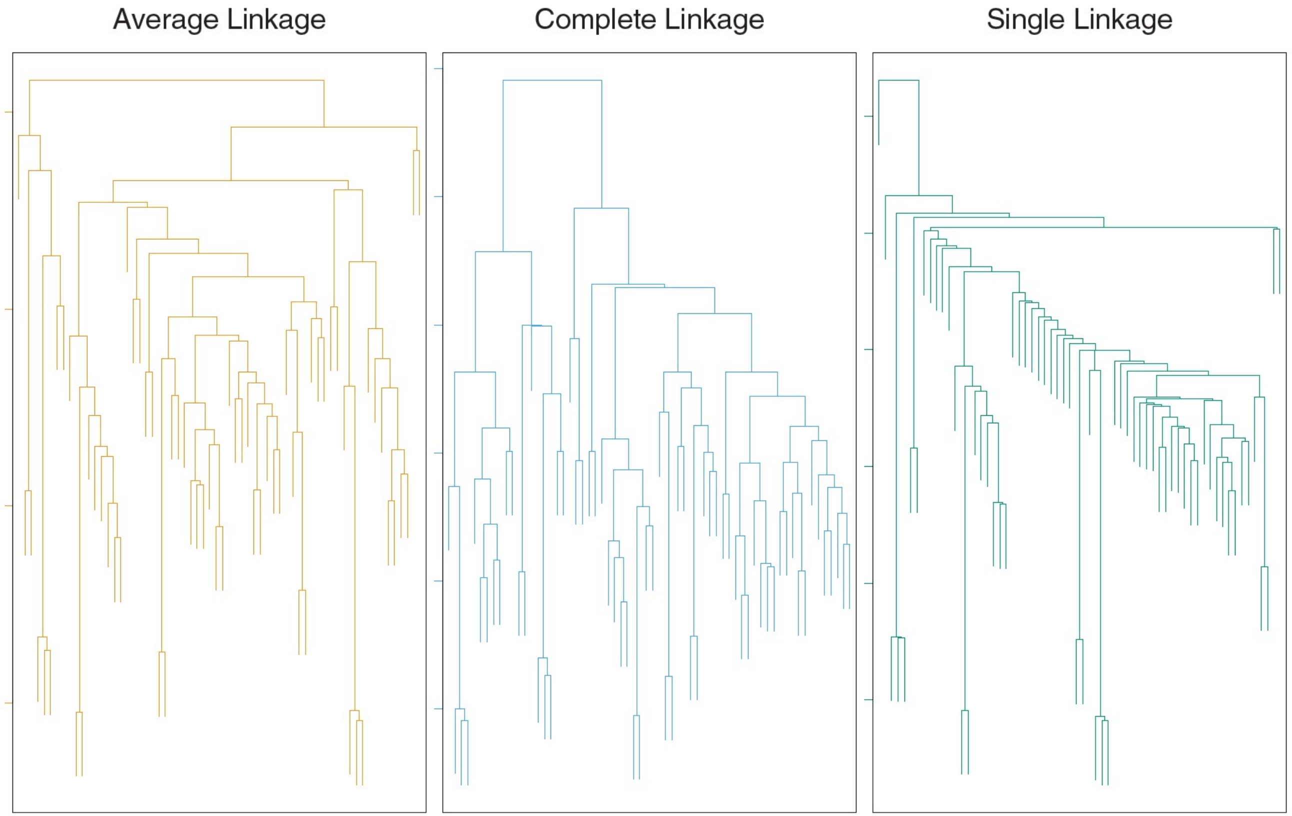


Largest vertical distance uncut by a horizontal line!!

Optimal Cut (Suggestion)



Single vs. Complete vs. Average



Average, complete, and single linkage applied to an example data set.

Average and complete linkage tend to yield more balanced clusters for this example

DBSCAN

Why DBSCAN?

- Same reasons as those of Hierarchical Clustering
- In addition, it helps us identify outliers, and does not let them effect the outcome of clustering

DBSCAN

- Works on the idea of density estimation
- And uses it to perform clustering

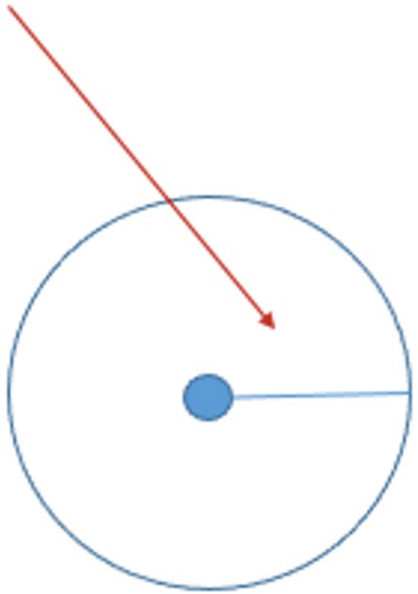
"***locate regions of high density that are separated from one another by regions of low density***"

DBSCAN Terminologies (1)

- Involves the following
  - Neighborhood of point p
  - Number of points inside the neighborhood of p
  - Core points
  - Border points
  - Noise points

DBSCAN Terminologies (2)

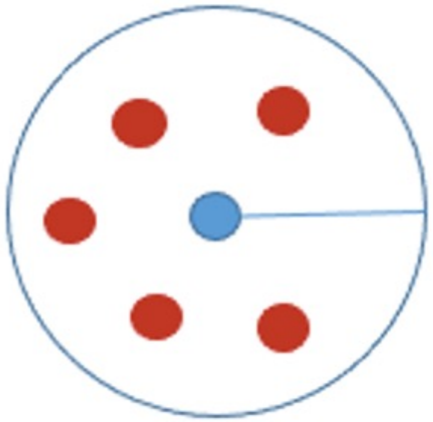
- Neighborhood of p (Eps)
  - Is treated as a *radius* of the circular neighborhood of p



DBSCAN Terminologies (3)

- Number of points inside the neighborhood of p

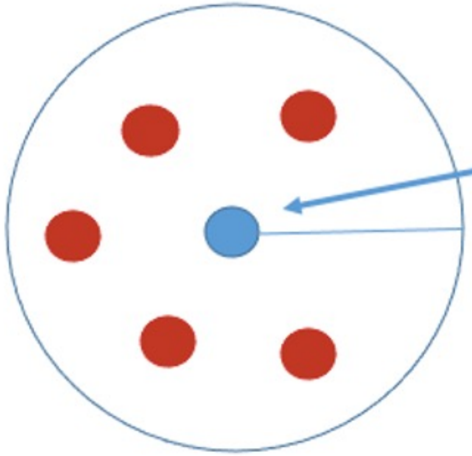
$$N_{Eps}(p) = \{q \in D \mid dist(p, q) \le Eps\ (p)\}$$



DBSCAN Terminologies (4)

- Core point p

$$|N_{Eps}(p)| \geq MinPts$$

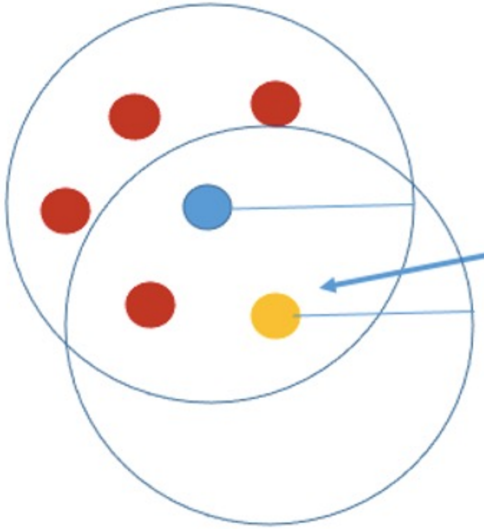


Core point for $MinPts = 4$

DBSCAN Terminologies (5)

- Border point q of a core point p

$$|N_{Eps}(q)| < MinPts \text{ but } q \in Neps(p)$$

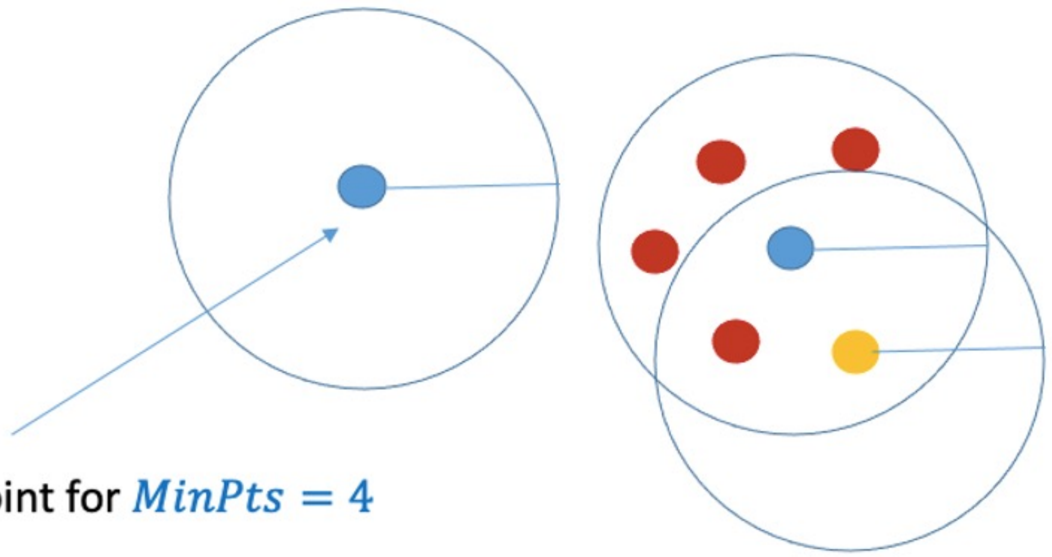


Border point for $MinPts = 4$

DBSCAN Terminologies (6)

- Noise point l

$$|N_{Eps}(l)| < MinPts \ AND \ l \ is \ not \ a \ border \ point$$



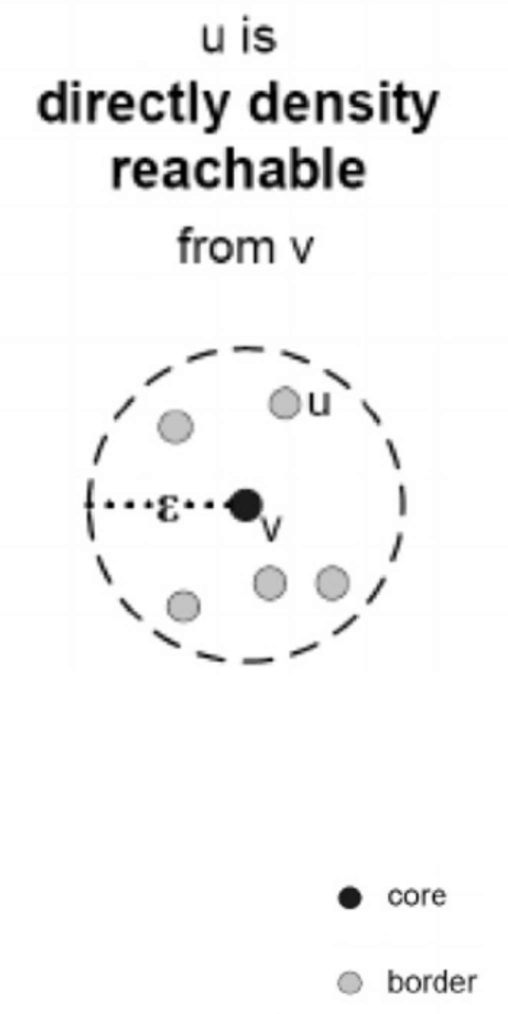
Noise point for $MinPts = 4$

Directly Density Reachable

- Point q is directly density reachable by point p if

  1. $|N_{Eps}(p)| \geq MinPts$ -- That is, p is a core point

  2. $q \in Neps(p)$ – That is, q belongs to the neighborhood of p



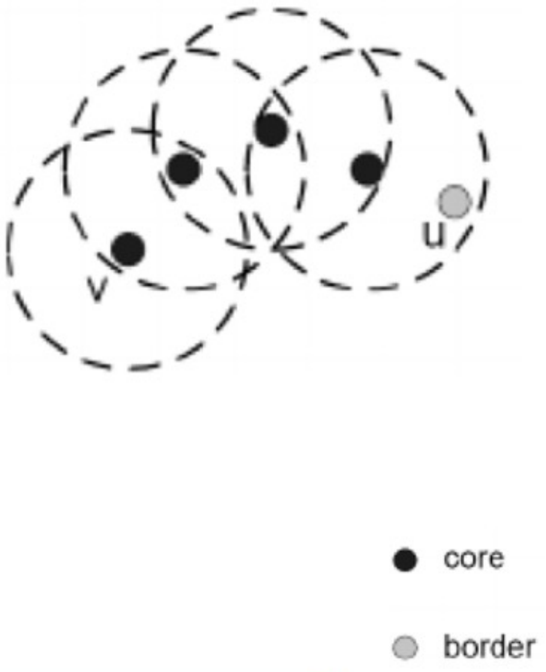
Note that Directly Density Reachable is not symmetric in general!

Density Reachable

- Point q is density reachable by point p if

  1. We can find a chain of points $p_1, p_2, \ldots, p_n$
  2. Such that $p_1 = p$, $p_n = q$
  3. And p_{i+1} is directly density reachable by p_i

u is **density reachable** from v



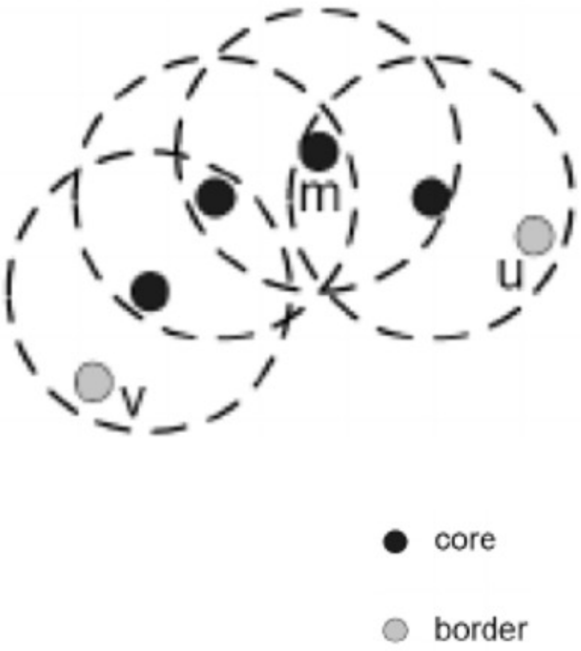
Note that Density Reachable is not symmetric in general, too!

Density Connected

- Point p and q are density connected if

  1. There is a point c
  2. Such that both p and q are density reachable from c

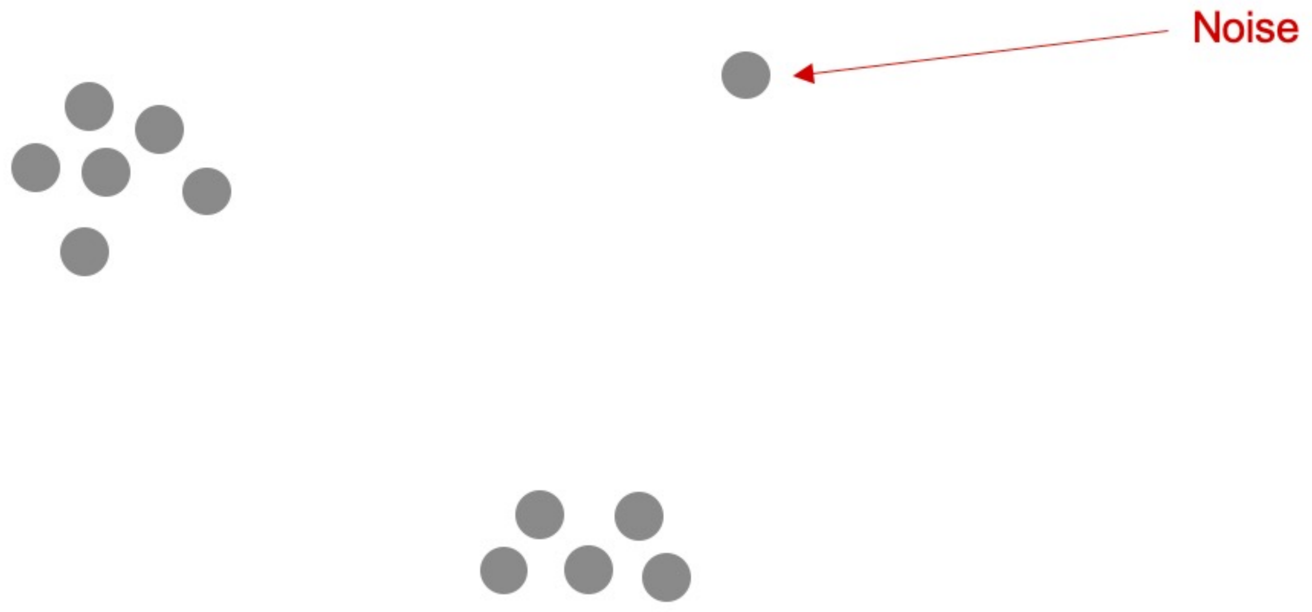
u is **density connected** from v



Note that Density Connected is symmetric!

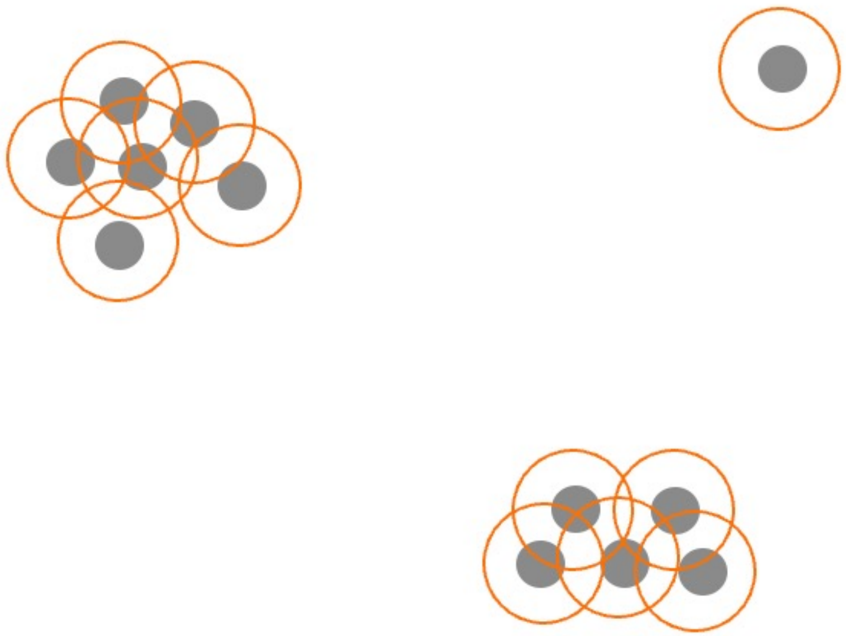
How does DBSCAN work?

We have the following data points



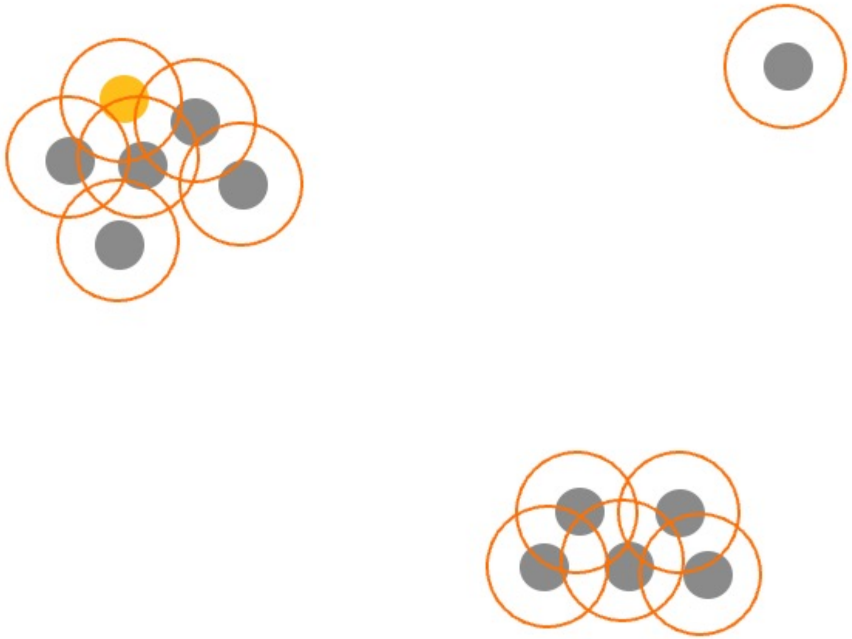
How Does DBSCAN Work?

- DBSCAN starts by identifying the neighboring data points of each data point within some radius (the neighborhood) - $Neps(p)$



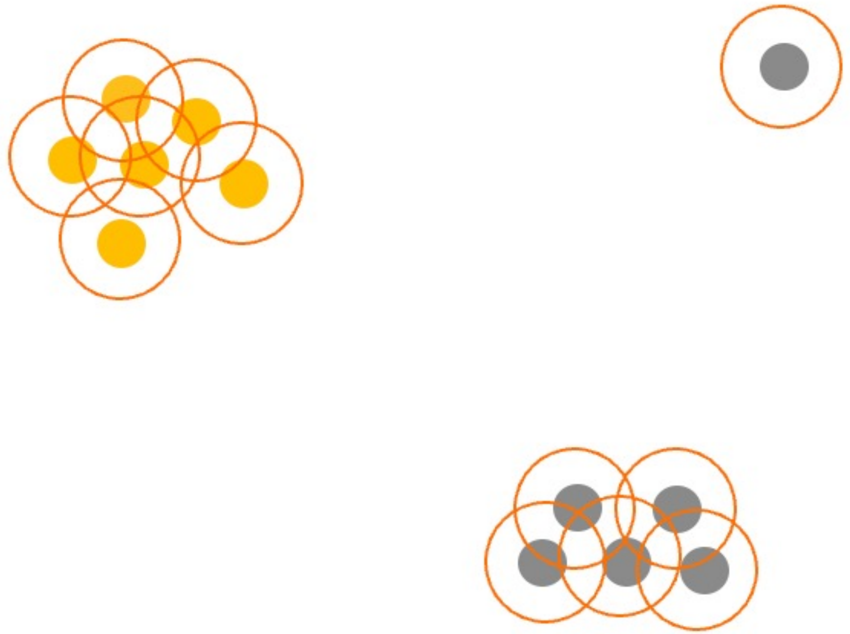
How Does DBSCAN Work?

- Next it starts with an arbitrary point assign it to a cluster.



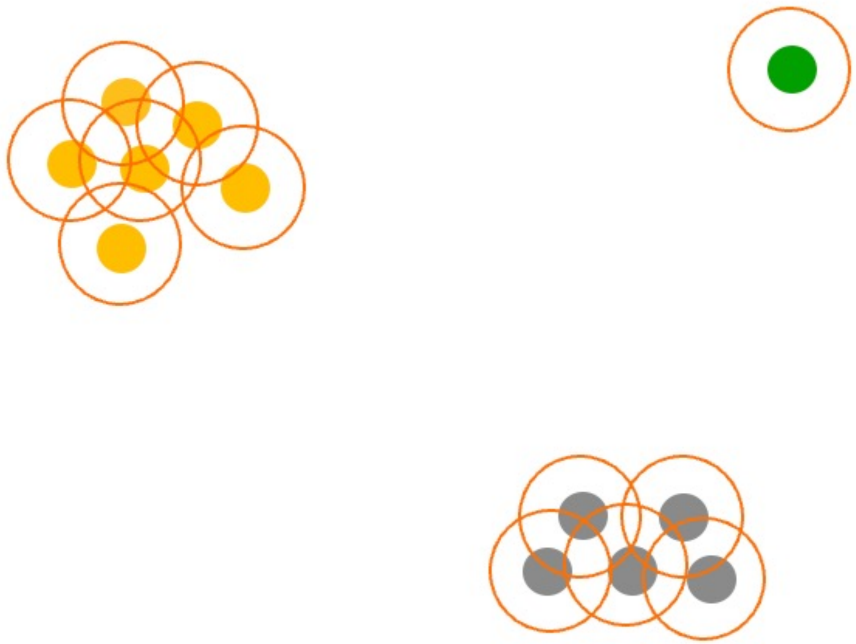
How Does DBSCAN Work?

- Any data point within the radius of another point are in the same cluster



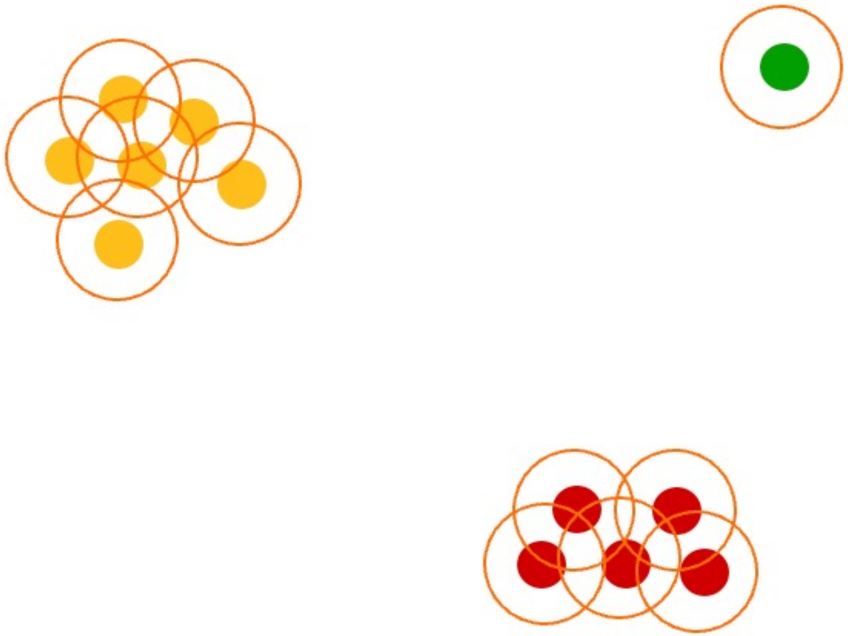
How Does DBSCAN Work?

- Any data point within the radius of another point are in the same cluster



How Does DBSCAN Work?

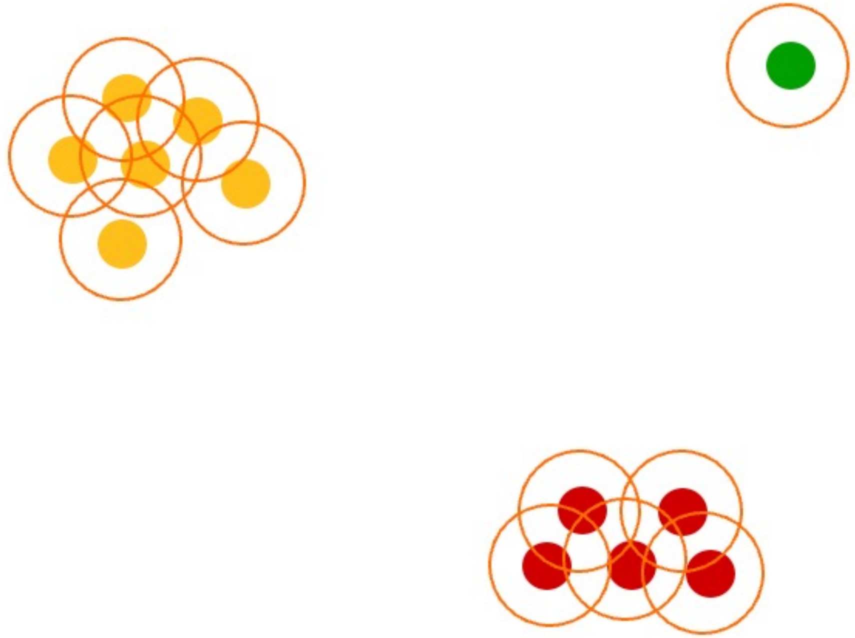
- Once more



Note that the noisy point has been considered as a cluster too!

How Does DBSCAN Work?

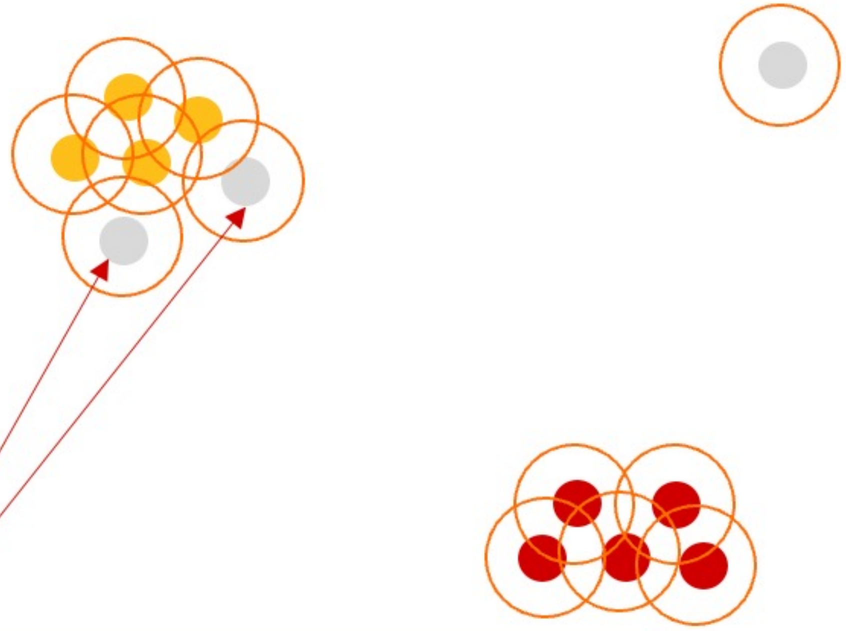
- To avoid Nosie, DBSCAN applies $|N_{Eps}(p)| \geq MinPts$



Let’s set $MinPts$ = 2

How Does DBSCAN Work?

- $|N_{Eps}(p)| \geq MinPts$, where $MinPts = 2$

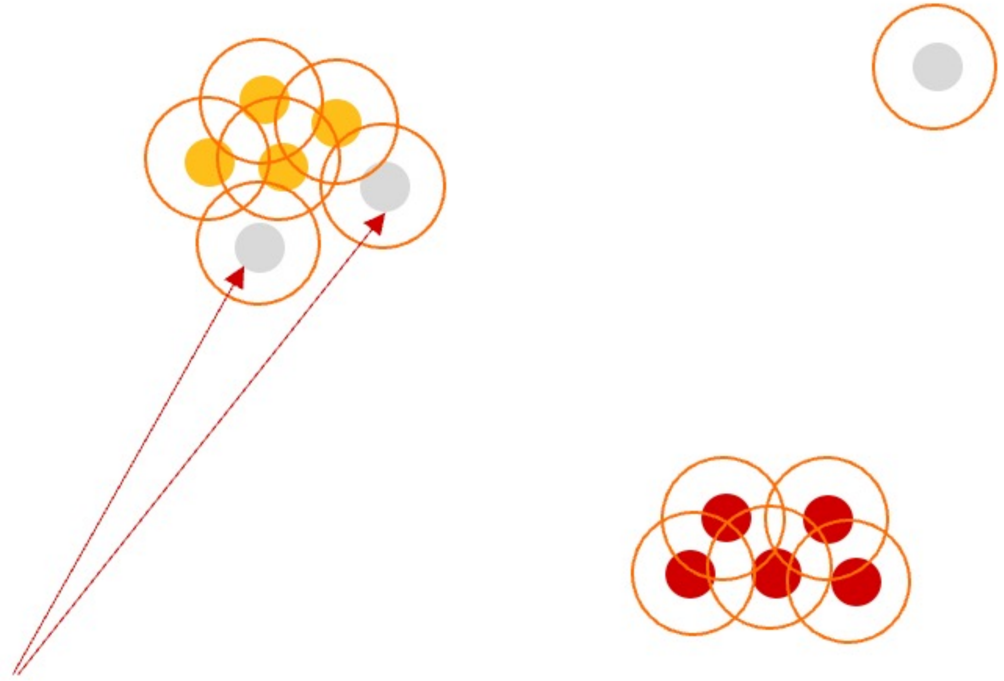


But these were not noise!

How Does DBSCAN Work?

- To fix this, DBSCAN employs the “reachibility”

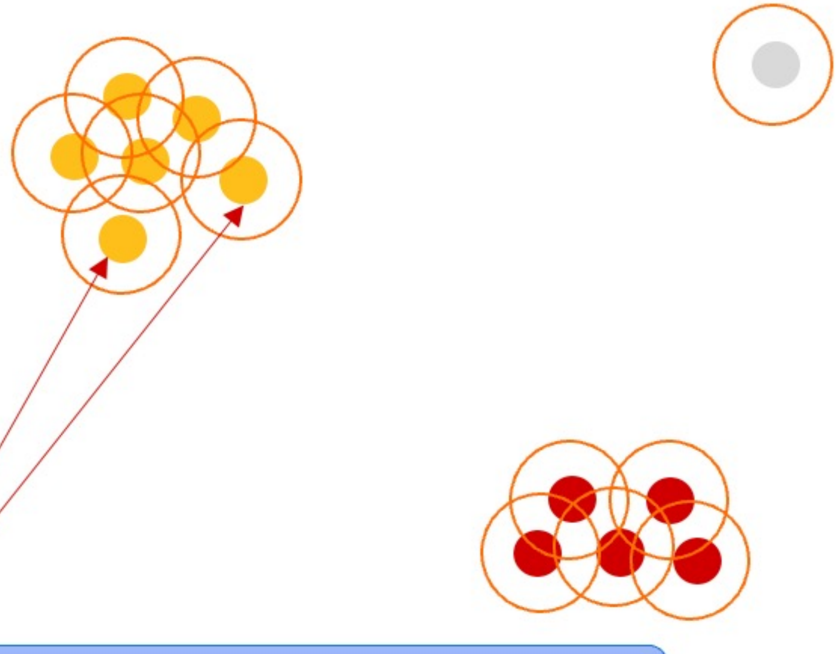
$\forall\, p, q$: if $p \in C$ and q is density reachable from p then $q \in C$



How Does DBSCAN Work?

- To fix this, DBSCAN employs the "reachibility"

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



Problem Fixed!

Thus a Cluster in DBSCAN is:

- Let D be the set of all points.
- Cluster C w.r.t Eps and $MinPts$ is a non-empty subset of D, such that
  1. $\forall\, p, q$: if $p \in C$ and q is density reachable from p then $q \in C$
  2. $\forall\, p, q \in C$, p is density connected to q

How Does DBSCAN Work? (Summary)

- Given Eps and $MinPts$, the DBSCAN discovers clusters in two steps

    1. Choose an arbitrary point from the data that satisfied the core point condition and use it as a seed

    2. Retrieve all points that are density reachable from the seed to create the cluster

Self-exploration

1. Read the original paper on DBSCAN

   https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf

2. Visualize how DBSCAN works at

   https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

Summary (Clustering)

1. Clustering: discovering structures/groups in the data
   - An unsupervised learning problem
2. Principle on which clustering works
3. K-means clustering
   - K-means ++
4. Hierarchical clustering
   - Divisive
   - Agglomerative
5. DBSCAN
   - Density based