Clustering Algorithms

Machine Learning

Prof. Sandra Avila
Institute of Computing (IC/Unicamp)

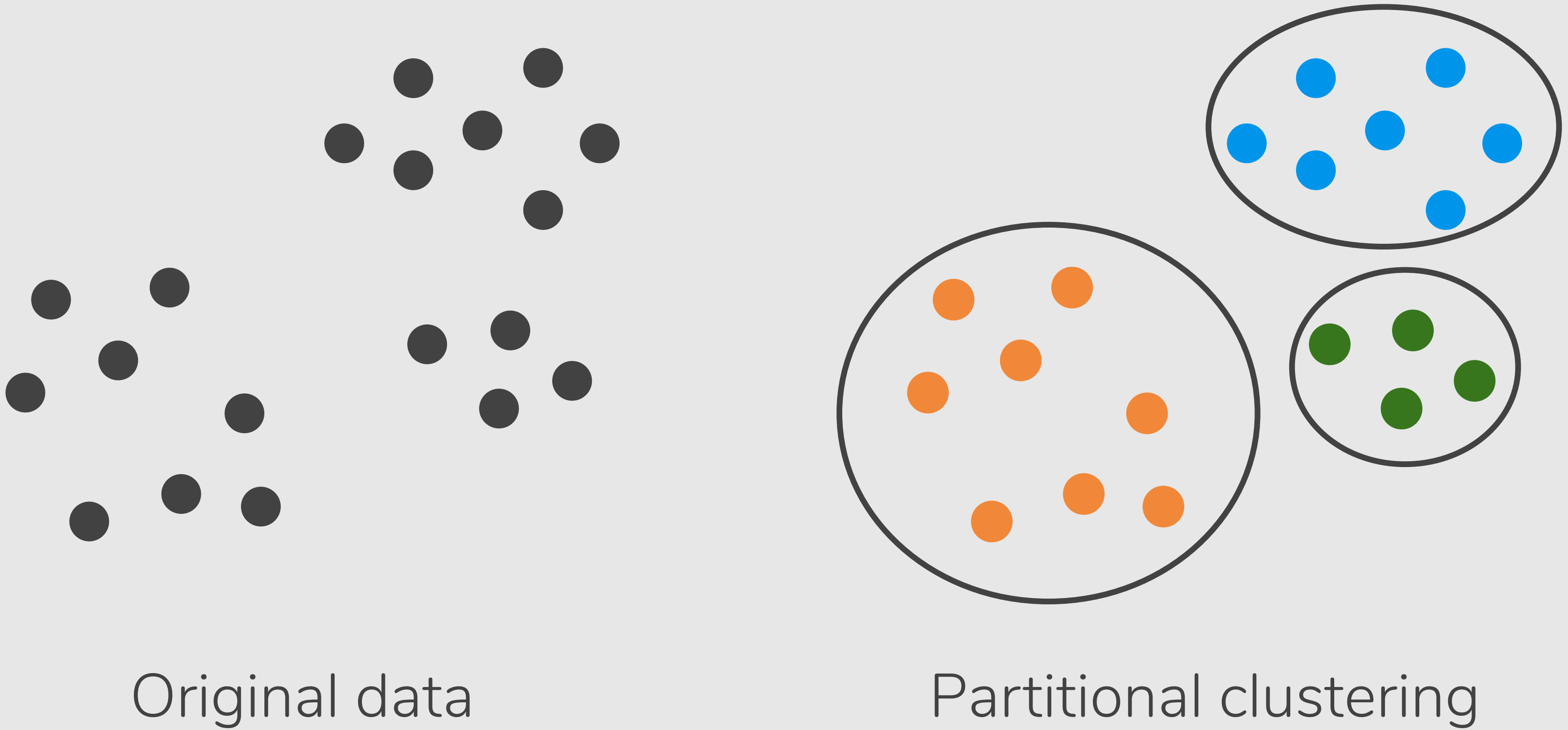
MC886, September 25, 2019

Today's Agenda

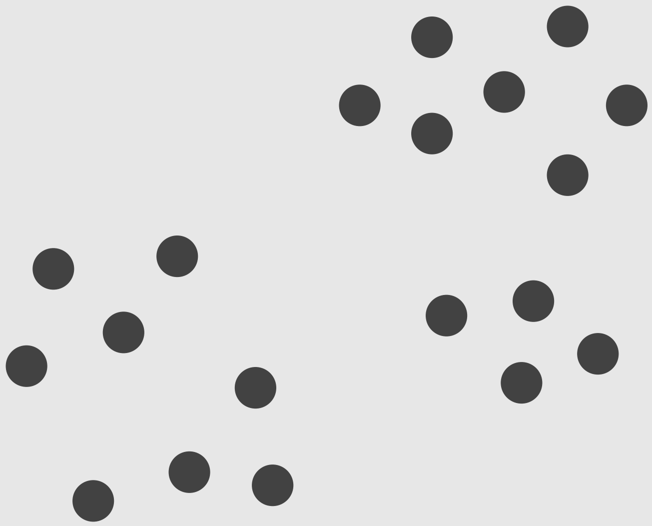
- Hierarchical Clustering
    - DBSCAN Clustering
- Clustering Performance Evaluation

Hierarchical Clustering

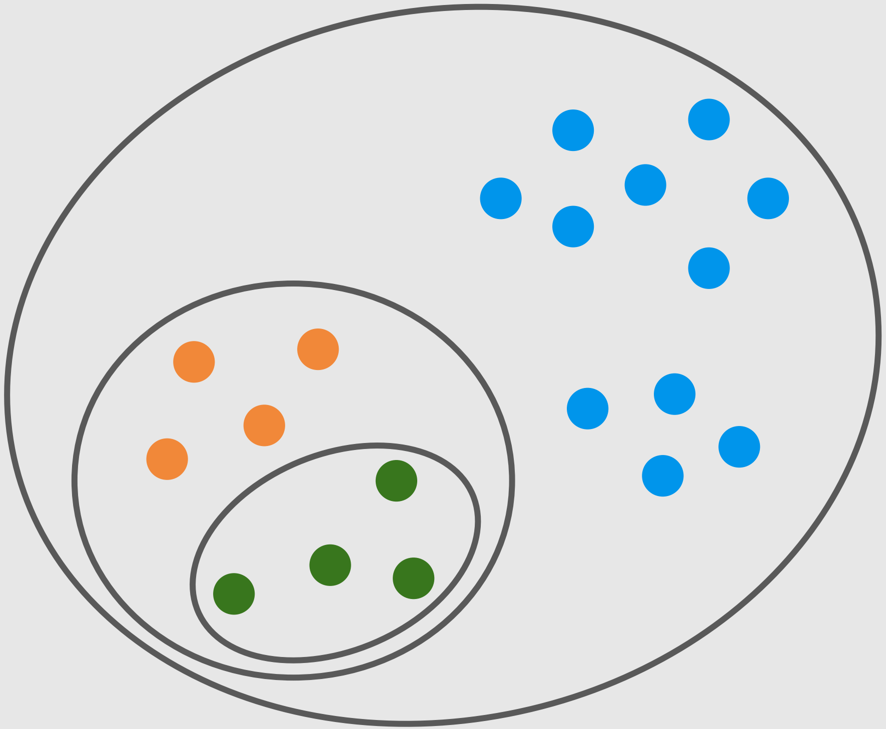
Hierarchical versus Partitional



Hierarchical versus Partitional



Original data



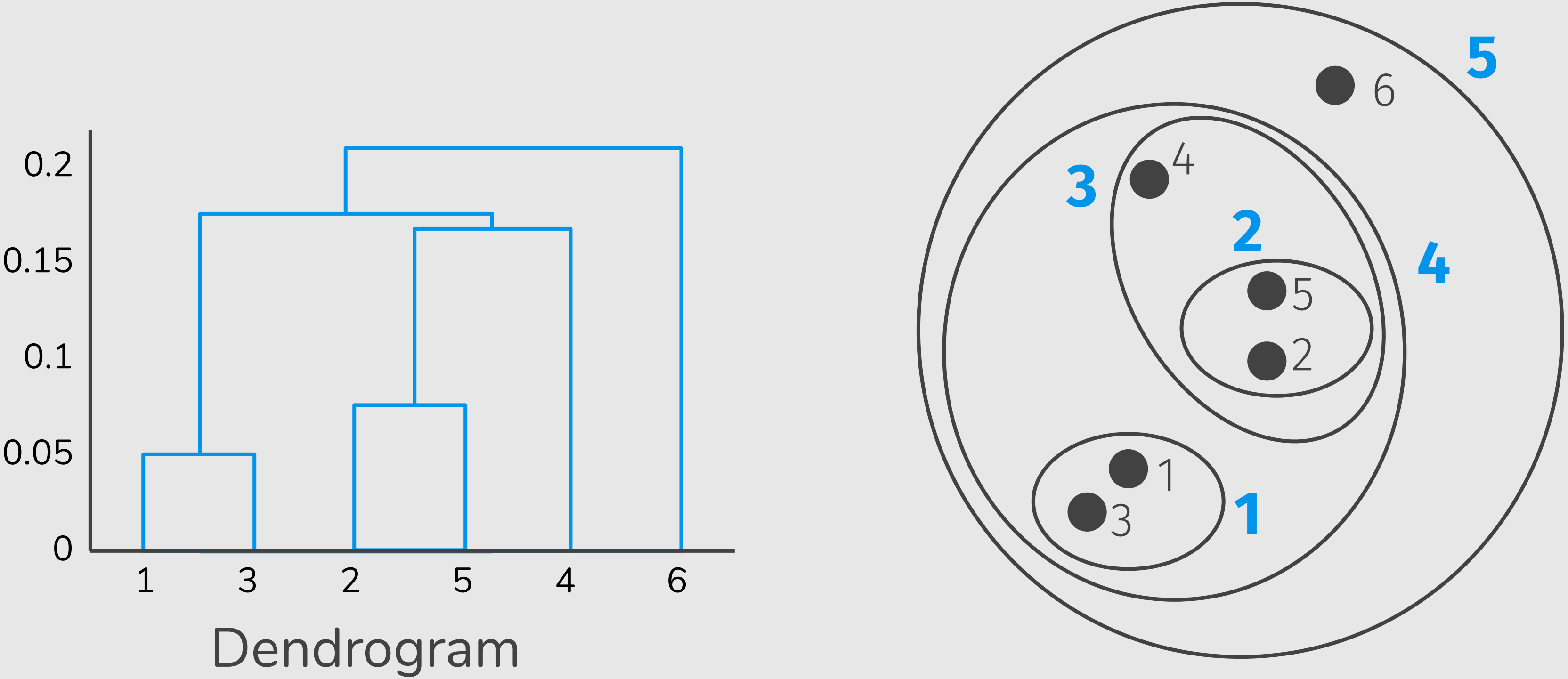
Hierarchical clustering

Hierarchical Clustering

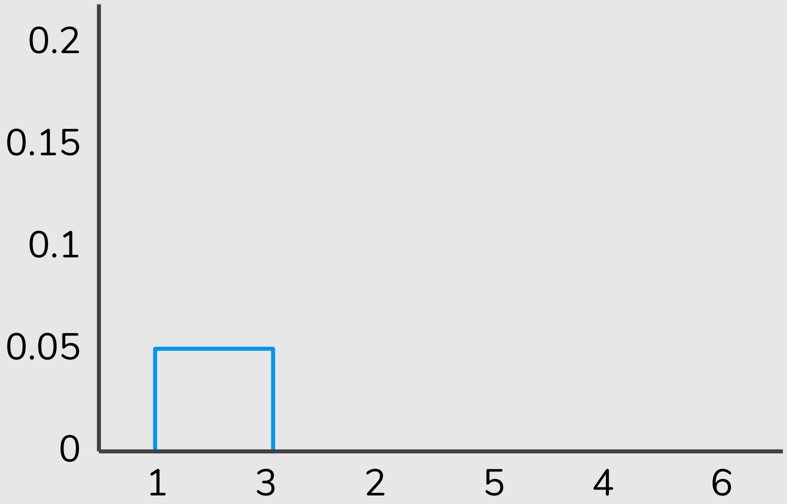
- **Agglomerative** ("bottom up"): each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.

- **Divisive** ("top down"): all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.

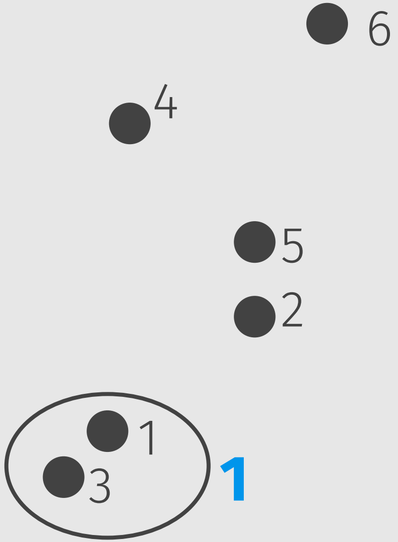
Agglomerative Hierarchical Clustering



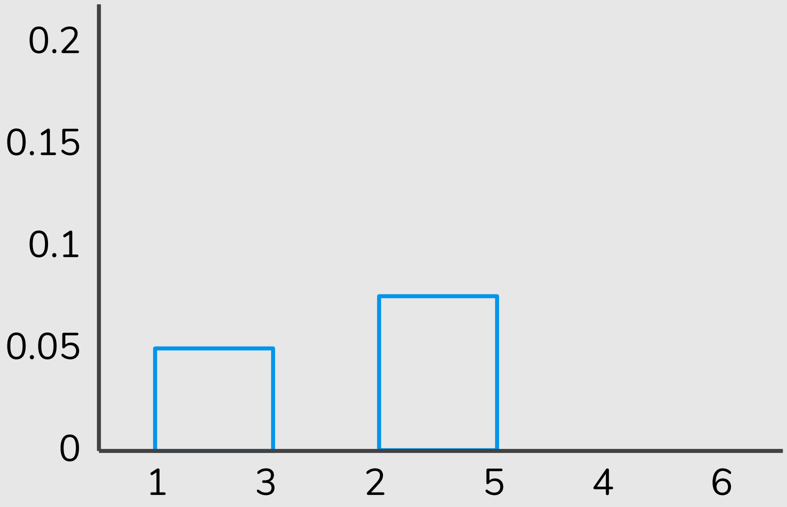
Agglomerative Hierarchical Clustering



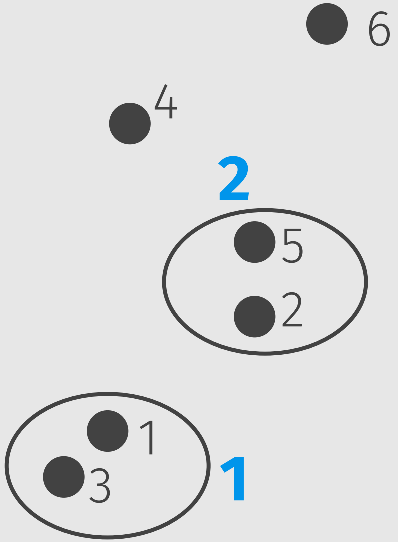
Dendrogram



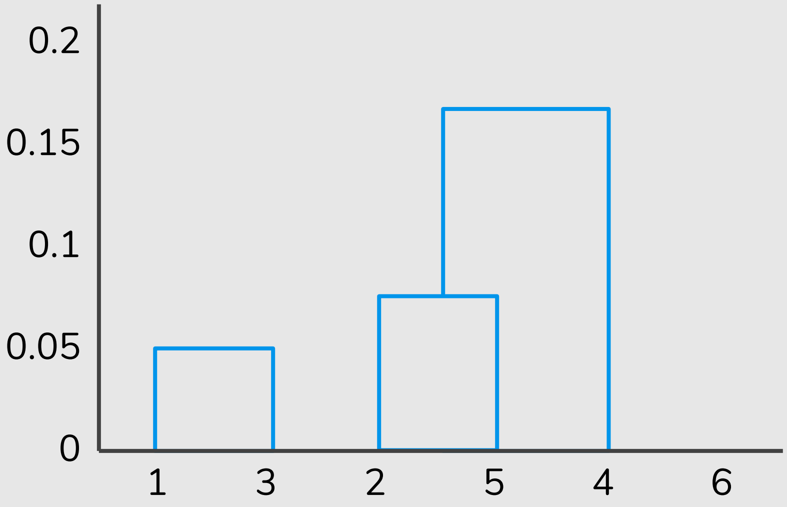
Agglomerative Hierarchical Clustering



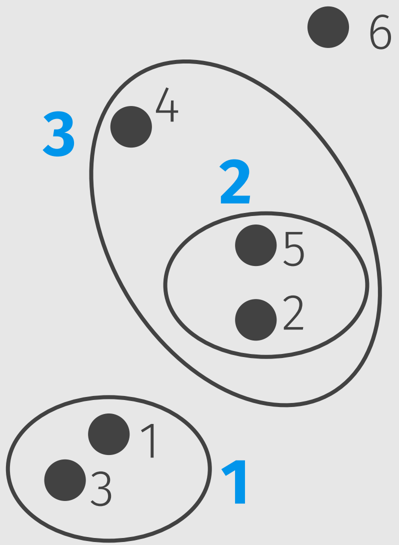
Dendrogram



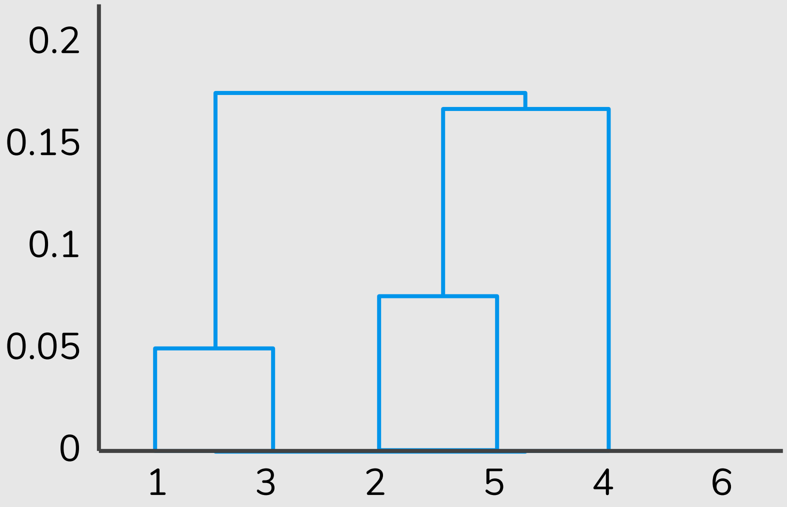
Agglomerative Hierarchical Clustering



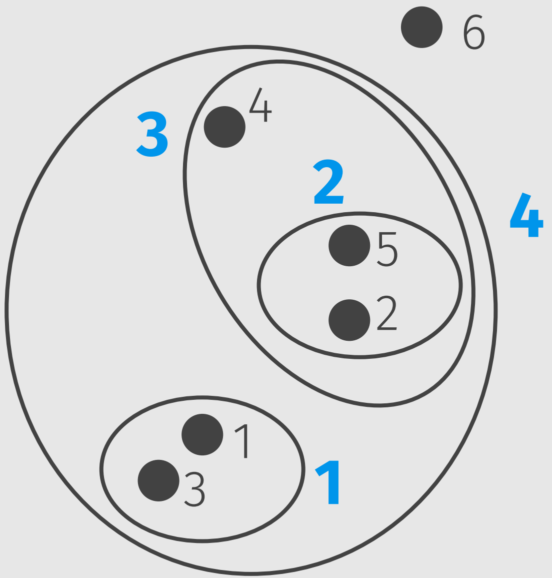
Dendrogram



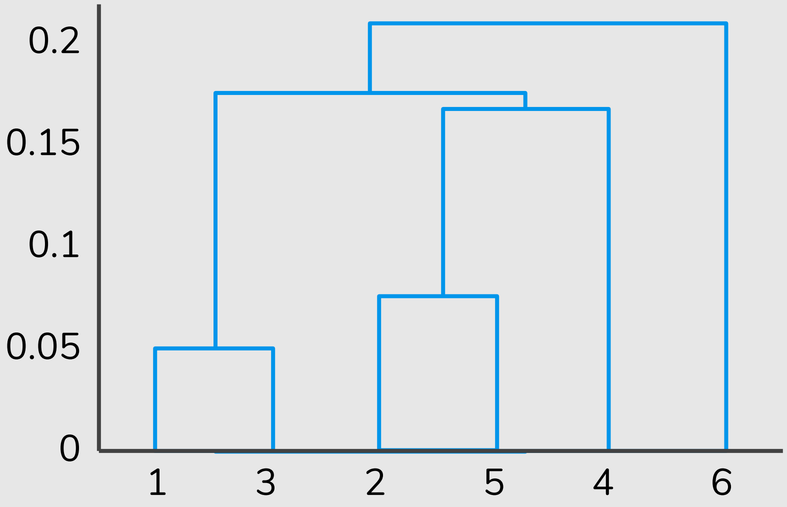
Agglomerative Hierarchical Clustering



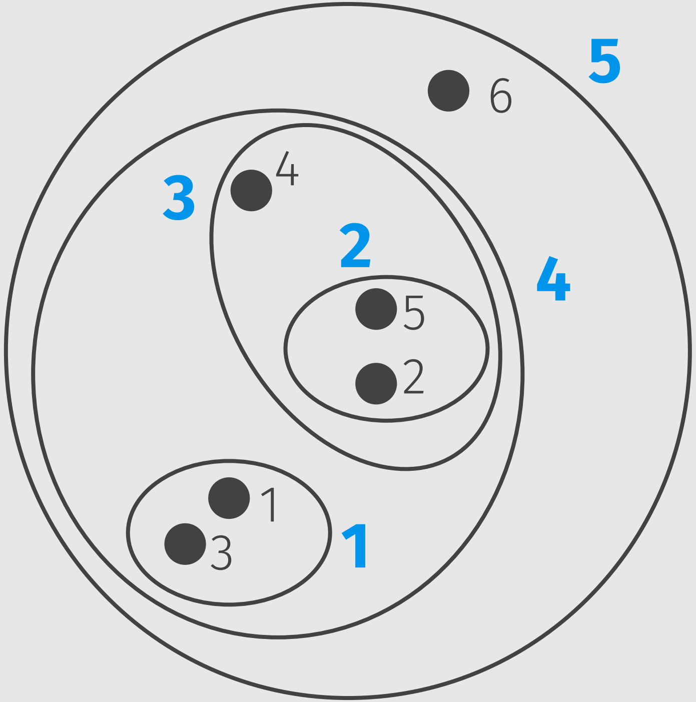
Dendrogram



Agglomerative Hierarchical Clustering



Dendrogram



Agglomerative Hierarchical Clustering

1: compute the **proximity matrix**, if necessary.

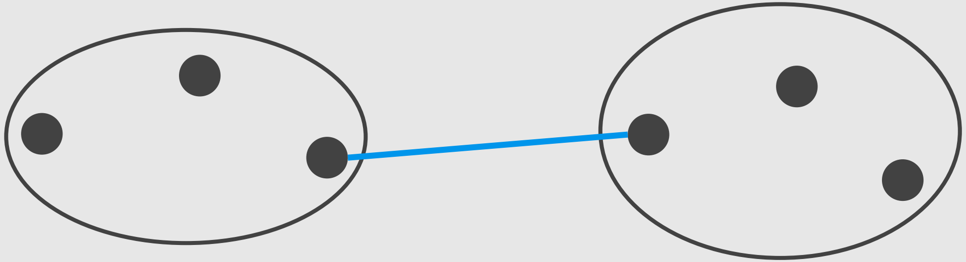
2: **repeat**

3: merge the closest two clusters.

4: update the proximity matrix to reflect the proximity between the new cluster and the original clusters.

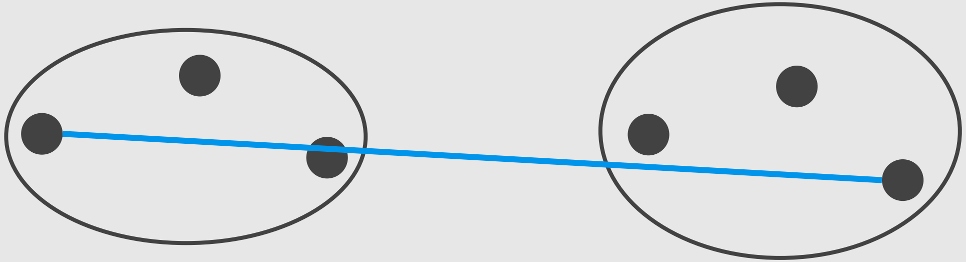
5: **until** only one cluster remains.

Defining Proximity between Clusters



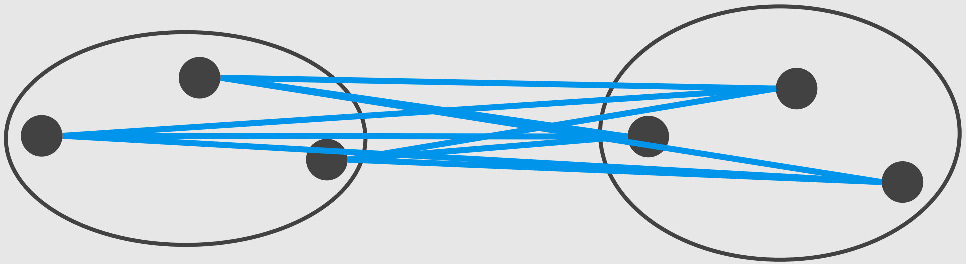
Single link or **MIN**: defines cluster proximity as the **proximity** between the closest two points that are in different clusters.

Defining Proximity between Clusters



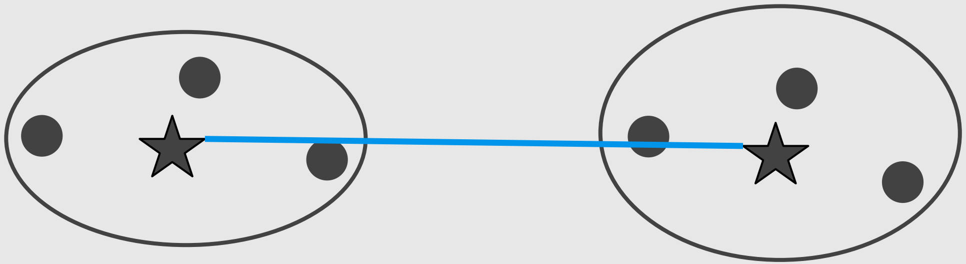
Complete link or **MAX**: takes the proximity between the **farthest** two points in different clusters to be the cluster proximity.

Defining Proximity between Clusters



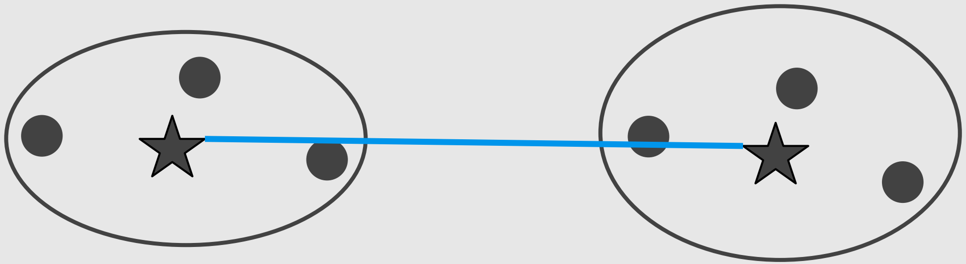
Average: defines cluster proximity to be the **average pairwise** proximities of all pairs of points from different clusters.

Defining Proximity between Clusters



Centroids: the cluster proximity is commonly defined as the proximity between cluster centroids.

Defining Proximity between Clusters



Ward's: measures the proximity between two clusters in terms of the increase in the SSE that results from merging the two cluster.

Agglomerative Hierarchical Clustering

1: compute the **proximity matrix**, if necessary.

2: **repeat**

3: merge the closest two clusters.

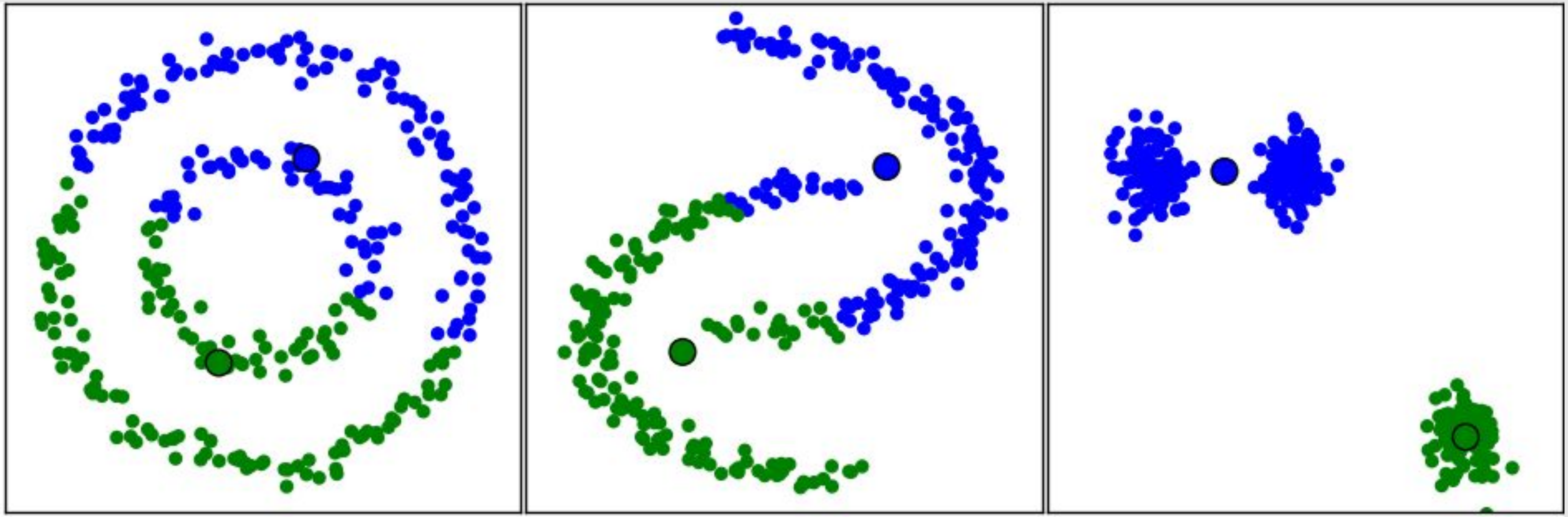
4: update the proximity matrix to reflect the proximity between the new cluster and the original clusters.

5: **until** only one cluster remains.

Today's Agenda

- Hierarchical Clustering
    - **DBSCAN Clustering**
- Clustering Performance Evaluation

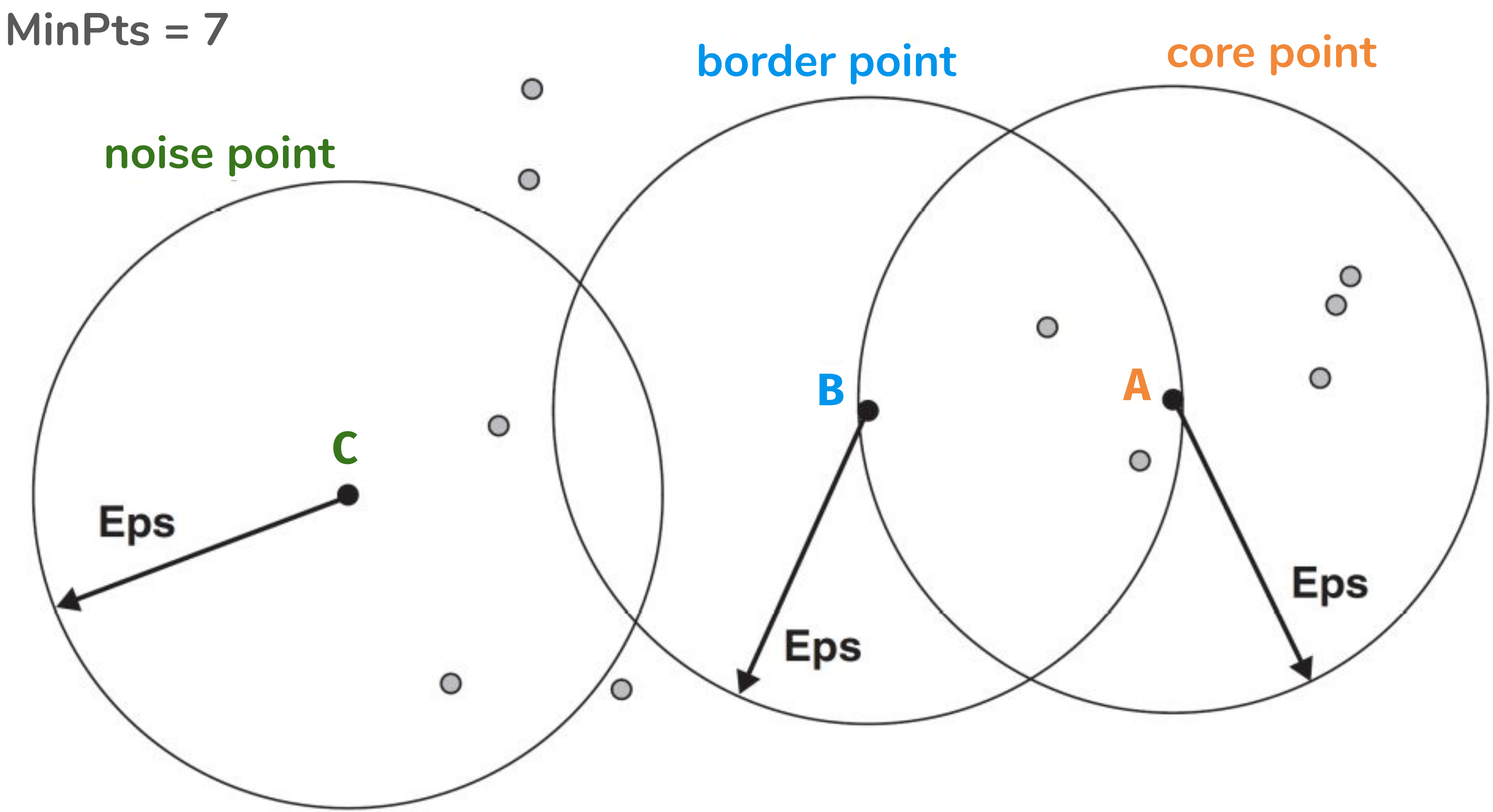
DBSCAN



DBSCAN Clustering

- **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise

- Given a set of points in some space, **it groups together points that are closely packed together** (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions.

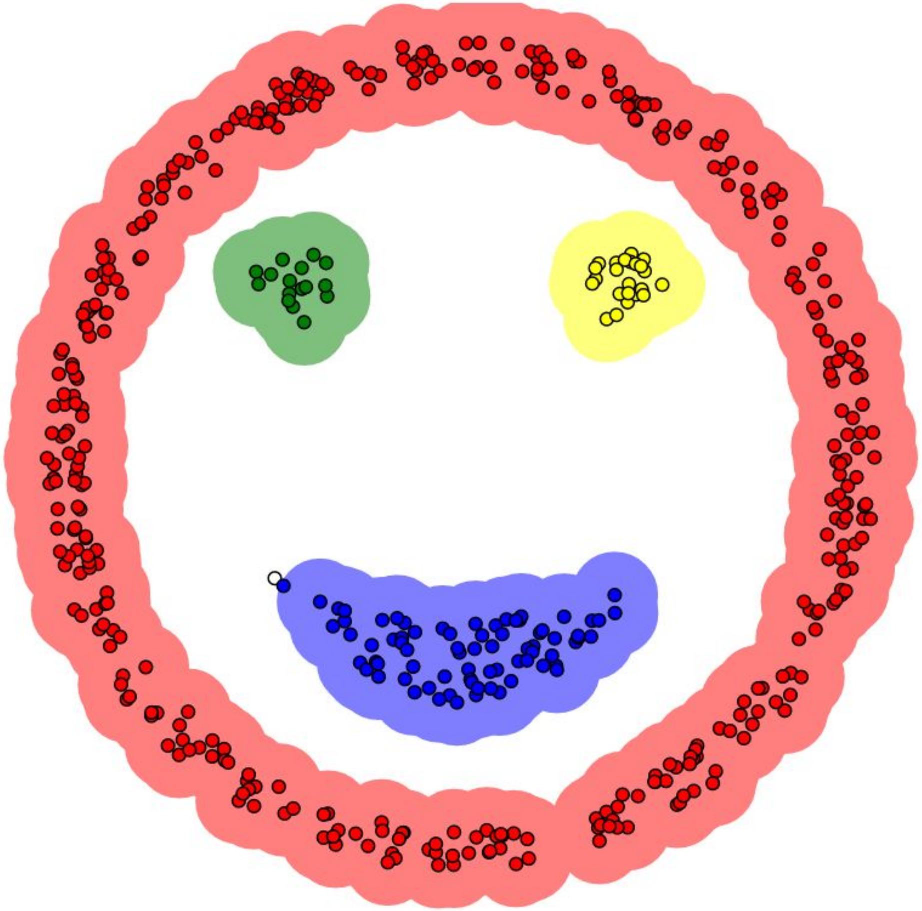

MinPts = 7
border point
core point
noise point
B
A
C
Eps
Eps
Eps

DBSCAN Clustering

- **Core points**: A point is a core point if there are at least *MinPts* within a distance of *Eps*, where *MinPts* and *Eps* are user-specified parameters.

- **Border points**: A border point is not a core point, but falls within the neighborhood of a core point.

- **Noise points**: A noise point is any point that is neither a core point nor a border point.



epsilon = 1.00
minPoints = 4

Restart

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

DBSCAN Algorithm

1. Start with an **arbitrary** point which has not been visited and its neighborhood information is retrieved from the *Eps* parameter.

2. If this point contains *MinPts* within *Eps* neighborhood, cluster formation starts.

   Otherwise the point* is labeled as **noise**.

   * This point can be later found within the *Eps* neighborhood of a different point and, thus can be made a part of the cluster.

DBSCAN Algorithm

3. If a point is found to be a **core** point then the points within the *Eps* neighborhood is also part of the cluster. So all the points found within *Eps* neighborhood are added, along with their own *Eps* neighborhood, if they are also **core** points.

4. The process restarts with a new point which can be a part of a new cluster or labeled as **noise**.

Today's Agenda

- Hierarchical Clustering
  - DBSCAN Clustering
- **Clustering Performance Evaluation**

Clustering Performance Evaluation

Clustering Evaluation

- Evaluating the performance of a clustering algorithm **is not as trivial** as counting the number of errors or the precision and recall of a supervised classification algorithm.

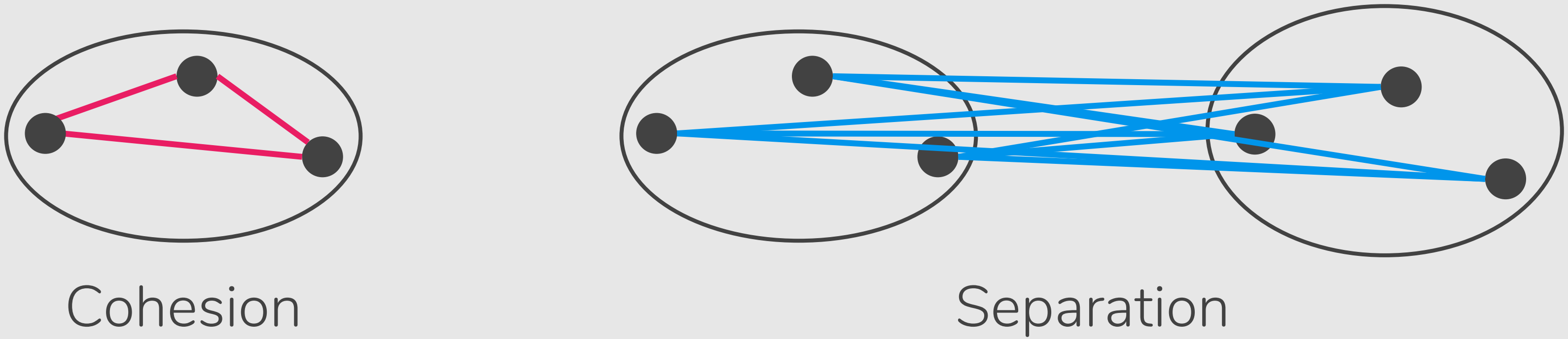
Clustering Evaluation

- Evaluating the performance of a clustering algorithm **is not as trivial** as counting the number of errors or the precision and recall of a supervised classification algorithm.

- Adjusted Rand index
- Mutual Information based scores
- Homogeneity, completeness and V-measure
- **Silhouette Coefficient**

Silhouette Coefficient

- The silhouette value is a measure of how similar a sample is to its own cluster (**cohesion**) compared to other clusters (**separation**).



Silhouette Coefficient

- The silhouette value is a measure of how similar a sample is to its own cluster (**cohesion**) compared to other clusters (**separation**).

- The silhouette ranges from −1 to +1.
    - High value = the clustering configuration is appropriate.
    - Low value = the clustering configuration may have too many or too few clusters.

Silhouette Coefficient

- The Silhouette Coefficient is defined **for each sample** and is composed of two scores:
  - a: The mean distance between a sample and all other points **in the same cluster**.
  - b: The mean distance between a sample and all other points **in the next nearest cluster**.

Silhouette Coefficient

- The Silhouette Coefficient s for **a single sample** is given as:

$$s = \frac{b - a}{max(a,b)}$$

- The score is bounded between -1 for incorrect clustering and +1 for highly dense clustering ($a \ll b$). Scores around zero indicate overlapping clusters.

http://scikit-learn.org/stable/modules/clustering.html#clustering

scikit learn

Home Installation Documentation Examples

Google Custom Search Search ×

Fork me on GitHub

Previous 2.2. Manifold... | Next 2.4. Biclustering | Up 2. Unsupervis...

scikit-learn v0.19.0
Other versions

Please **cite us** if you use the software.

2.3. Clustering

Clustering of unlabeled data can be performed with the module `sklearn.cluster` .

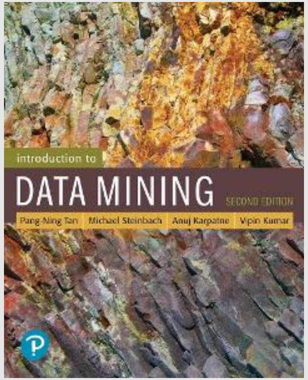
Each clustering algorithm comes in two variants: a class, that implements the `fit` method to learn the clusters on train data, and a function, that, given train data, returns an array of integer labels corresponding to the different clusters. For the class, the labels over the training data can be found in the `labels_` attribute.

Input data

One important thing to note is that the algorithms implemented in this module can take different kinds of matrix as input. All the methods accept standard data matrices of shape `[n_samples, n_features]` . These can be obtained from the classes in the `sklearn.feature_extraction` module. For `AffinityPropagation` , `SpectralClustering` and `DBSCAN` one can also input similarity matrices of shape `[n_samples, n_samples]` . These can be obtained from the functions in the `sklearn.metrics.pairwise` module.

2.3.1. Overview of clustering methods

References



Machine Learning Books

- Pattern Recognition and Machine Learning, Chap. 9 “Mixture Models and EM”
- Pattern Classification, Chap. 10 “Unsupervised Learning and Clustering”
- “Introduction to Data Mining”, https://www-users.cs.umn.edu/~kumar001/dmbook/ch7_clustering.pdf

Machine Learning Courses

- https://www.coursera.org/learn/machine-learning, Week 8