UNSUPERVISED LEARNING

Clustering and k-means

Supervised vs Unsupervised Learning

Supervised learning use **labeled** training set

Classification

Weight	Color	Seeds	Fruit	
150	80	8	0	apple
200	112	6	1	orange
170	120	8	1	orange
210	105	7	1	orange
180	130	9	0	apple

attributes class (target)

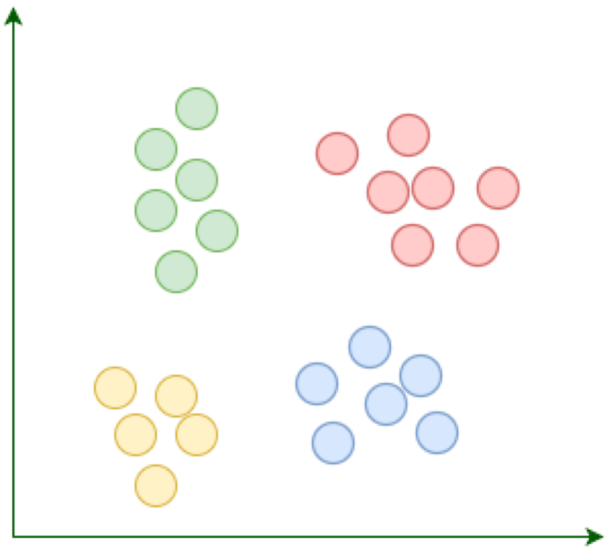
In contrast, **Unsupervised learning** uses only **unlabeled** data: no class nor associated value.

Unsupervised Learning

Two tasks:

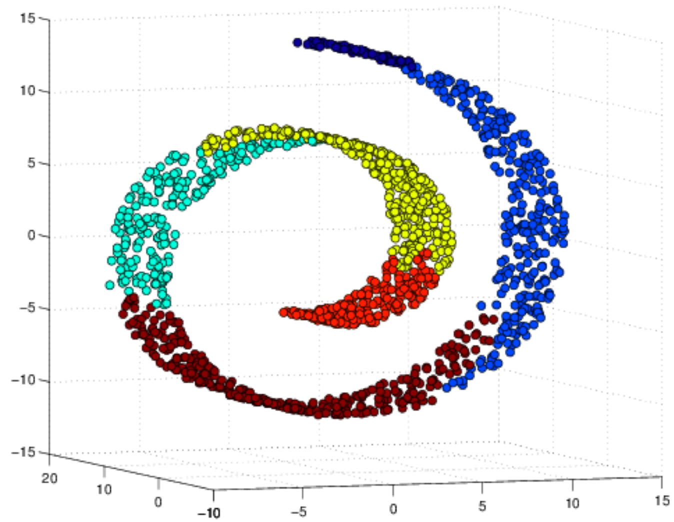
- **Clustering**

find groups in the data



- Representation learning

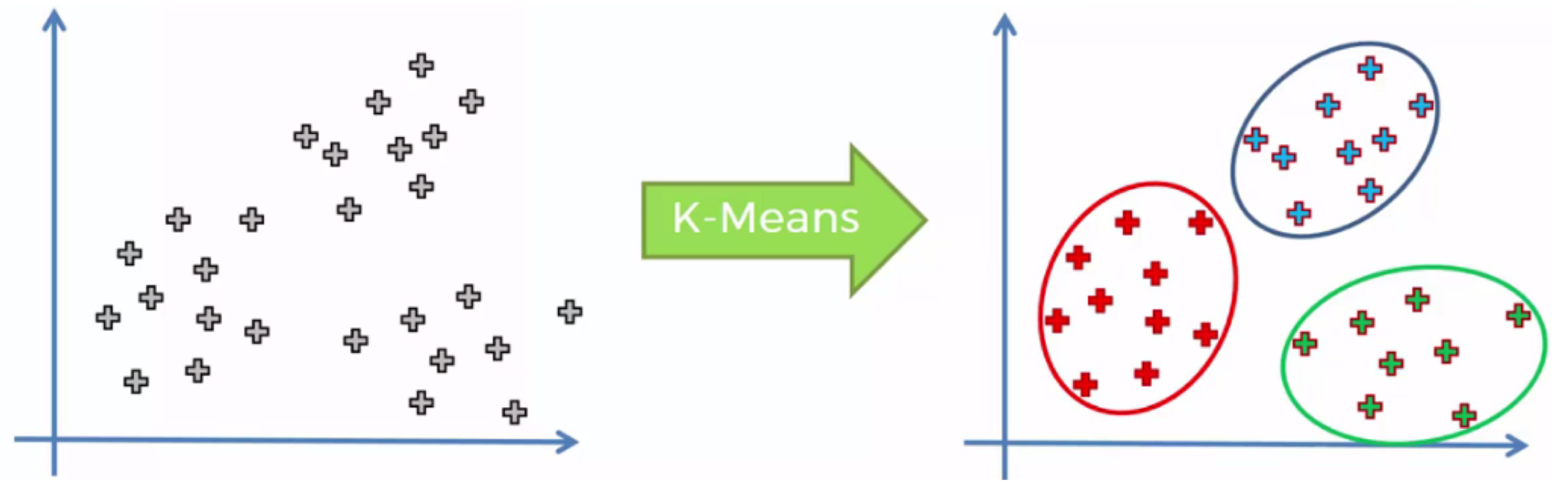
dimension reduction



What is clustering ?

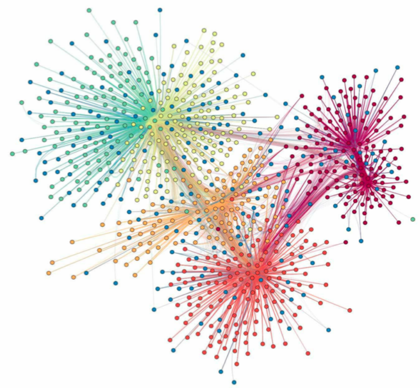
Find groups in the data

- Rely on a **similarity measure** (distance) between data points

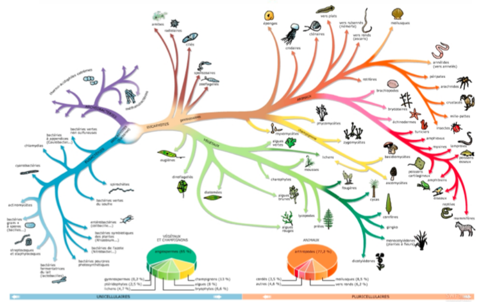


Clustering: applications

- Explore and understand the data
  - Online social networks analysis
  - Epidemiology



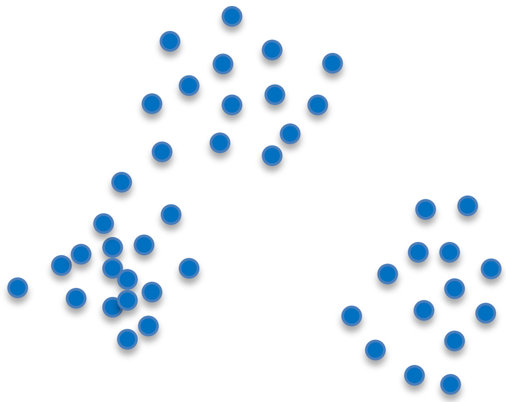
- Summarize data, build taxonomies
  - Information search
  - Biology



- Apply specialized models on each segment
  - Marketing

What is a good clustering ?

It's hard to define precisely what we want



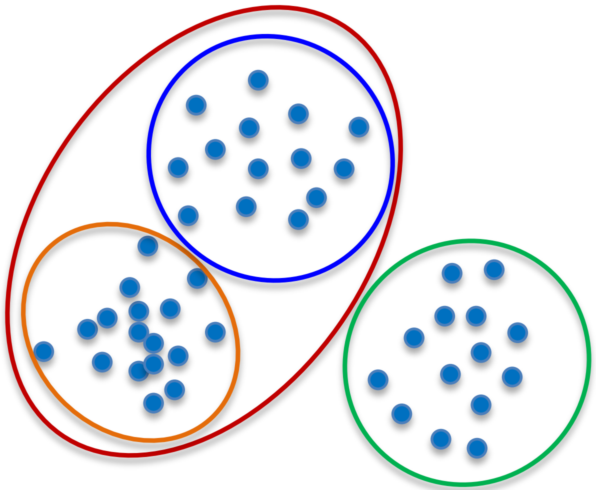
What is a good clustering ?

Two groups or three groups ?

Two examples of *Partitional Clustering*

What is a good clustering ?

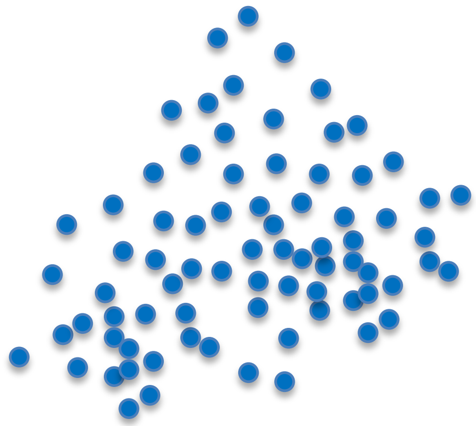
Or maybe some hierarchical structure ?



Hierarchical Clustering

What is a good clustering ?

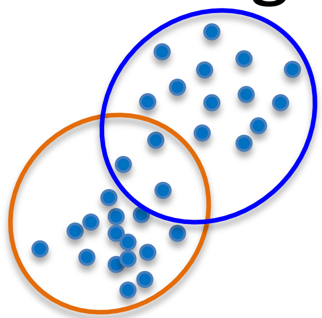
And sometimes, there are no clusters



Other distinctions between clusterings

- **Exclusive** versus non-exclusive

  In non-exclusive (or overlapping) clusterings, points may belong to multiple clusters.

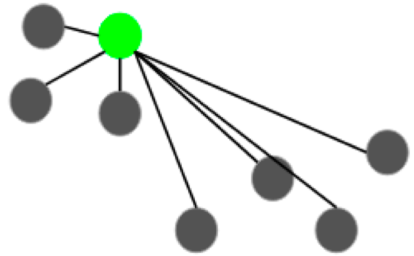


- **Fuzzy** versus non-fuzzy
  - In fuzzy clustering, a point belongs to each cluster with some probability (in [0,1])

k-means basic algorithm

The most popular clustering method

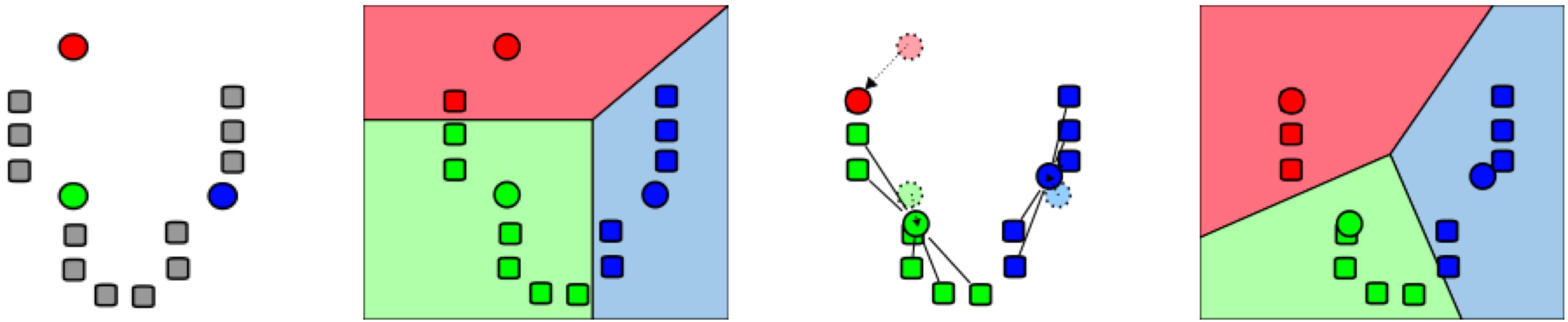
- Each cluster is associated with a **centroid** (center point)
- Each point is assigned to the cluster with the closest centroid
- The number of clusters, K, must be specified



The basic algorithm is very simple:

1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

k-means basic algorithm



1) K initial "means" (here K=3, red, green, blue) are randomly generated within the data

2) K clusters are created by associating every point with the nearest mean. partitions = **Voronoi** diagram

3) The centroid of each of the k clusters becomes the new mean.

4) Steps **2** and **3** are repeated until convergence has been reached.

Source: wikipedia

k-means algorithm properties

- Initial centers are chosen randomly: solution will differ from one run to another.
- Similarity is measured by Euclidean distance, or other measures like cosine or correlation.
- K-means will converge (trust me or read Bottou&Bengio 94)
- Complexity is O(n . K . L . D)

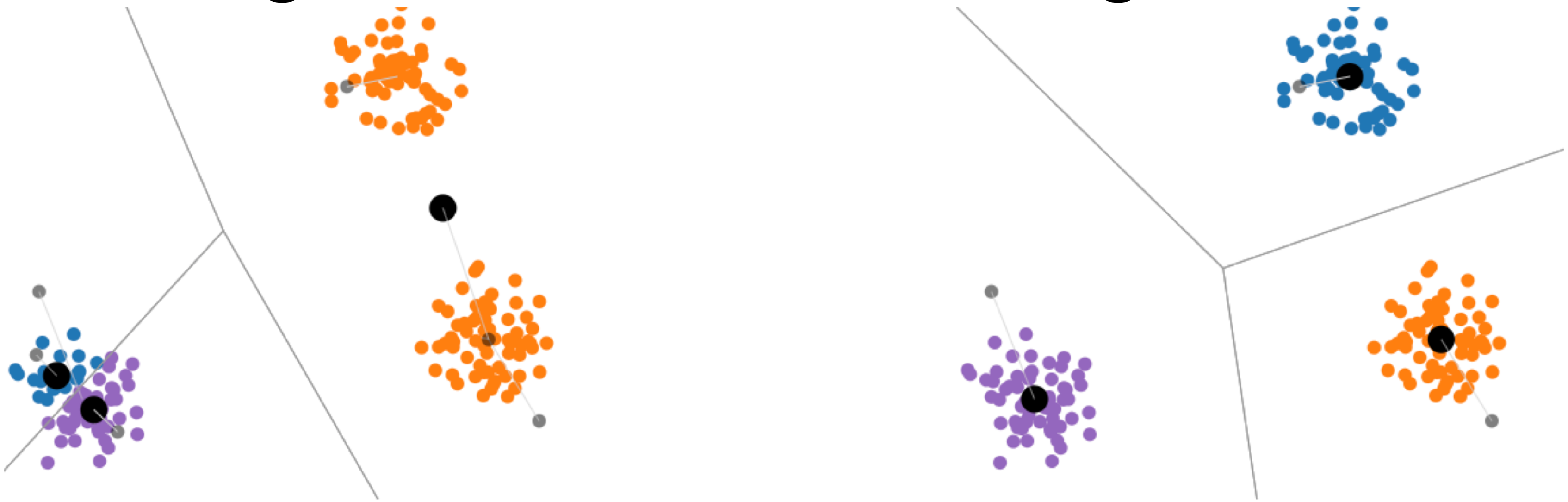
  n = number of points, K = number of clusters

  L = number of iterations, d = number of attributes

Source: wikipedia

k-means : choosing initial centers

The algorithm is sensitive to the initial choice of centers: it can get stuck in a bad configuration

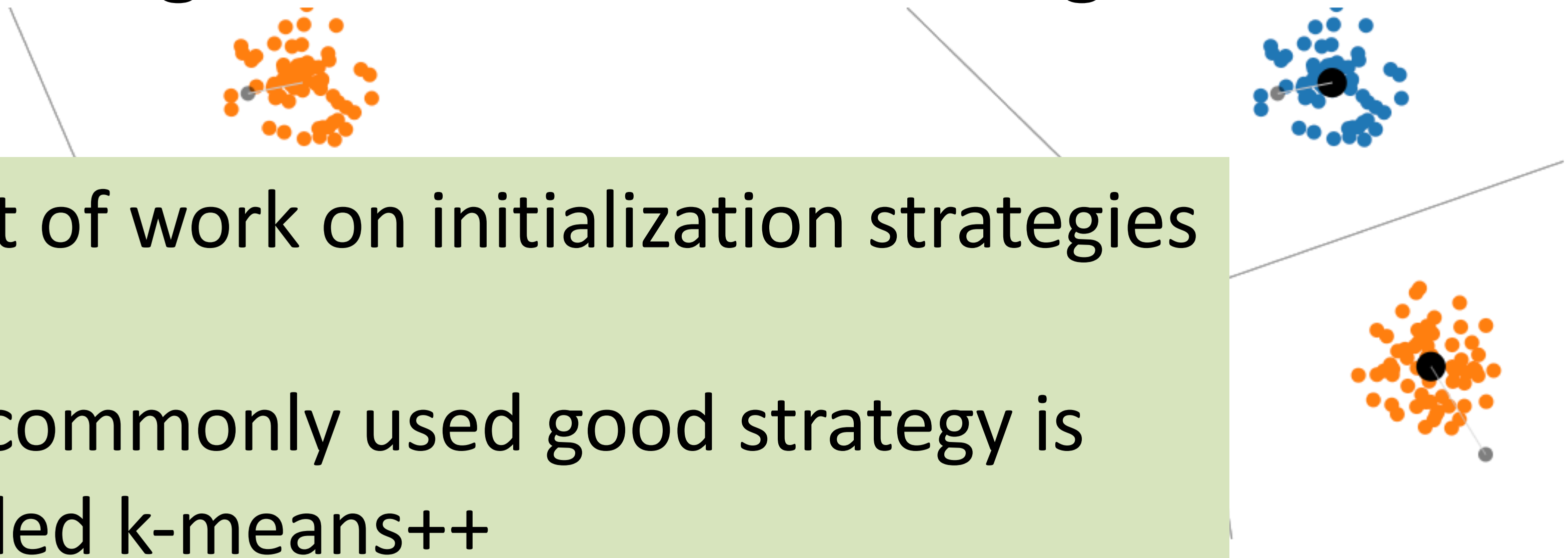


Stuck in local minima   Good solution

Have a look at the interactive demo http://alekseynp.com/viz/k-means.html

k-means : choosing initial centers

The algorithm is sensitive to the initial choice of centers: it can get stuck in a bad configuration



⇒Lot of work on initialization strategies

⇒A commonly used good strategy is called k-means++

k-means : quantifying performance

How to find the best clustering? How to choose K ?

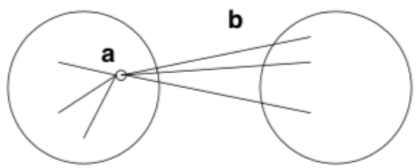
Quantization error (MSE) can be set to zero if K is sufficiently large (but is useful to compare 2 clusterings with the same k)

Silhouette coefficient measures cohesion and separation:

- For an individual point, *i*
  - Calculate ***a*** = average distance of *i* to the points in its cluster
  - Calculate ***b*** = min (average distance of *i* to points in another cluster)
  - The silhouette coefficient for a point is then given by

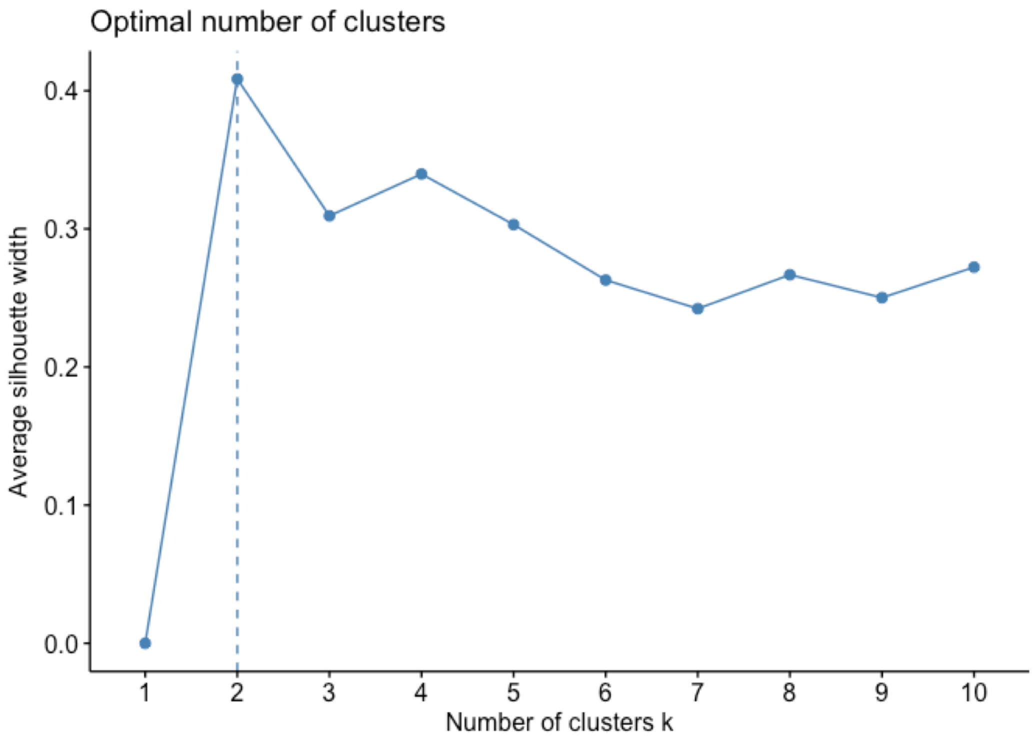
    $s = 1 - a/b$ if $a < b$, (or $s = b/a - 1$ if $a \geq b$, not the usual case)

  - Typically between 0 and 1.
  - The closer to 1 the better.



- **The Average Silhouette Coefficient** of a cluster is the average of the silhouette coefficient of points belonging to the cluster.

k-means : determining the best k

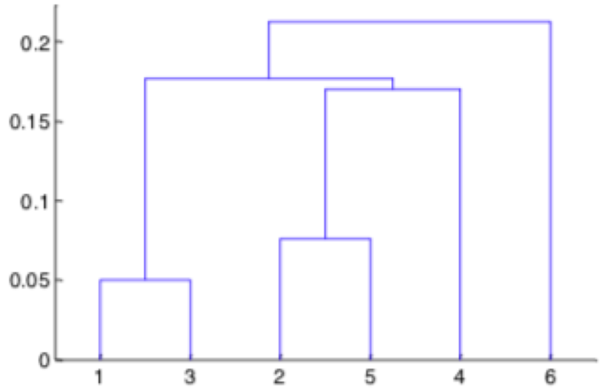


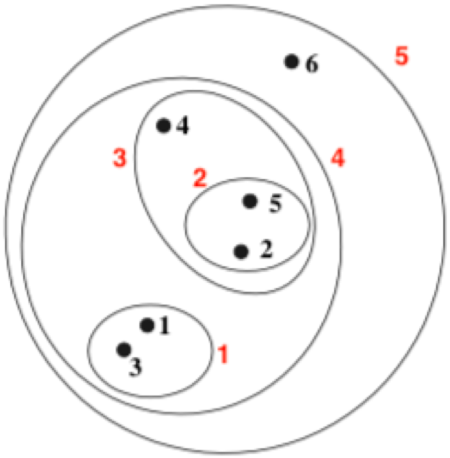
See also Elbow method

Code and examples: https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html
And (in R) https://uc-r.github.io/kmeans_clustering

Hierarchical clustering

Produces a hierarchical tree, can be visualized as a dendrogram (records the sequence of merges)





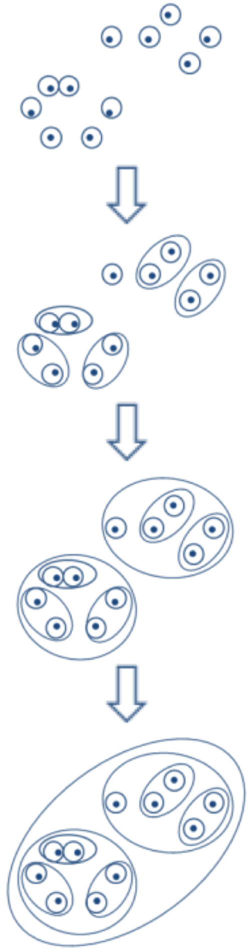
Interpretability, taxonomy

Number of clusters not fixed in advance

Hierarchical clustering

Agglomerative algorithm

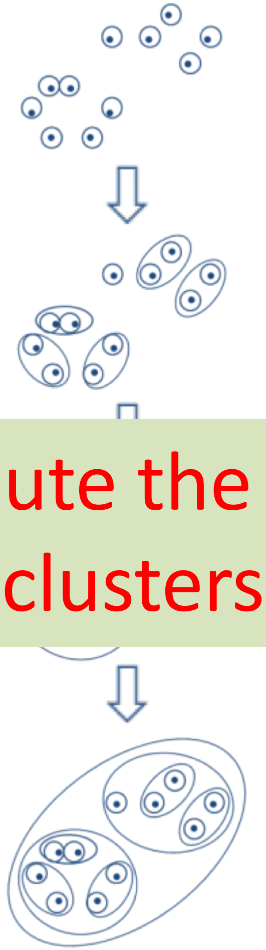
1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two *closest* clusters
5. Update the proximity matrix
6. **Until** only a single cluster remains



Hierarchical clustering

Agglomerative algorithm

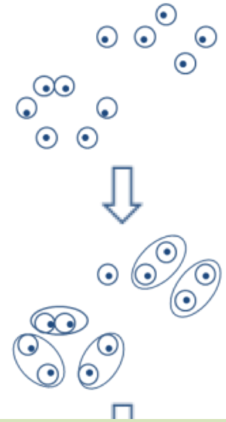
1. Compute the proximity matrix
2. Let each data point be a clu
3. **Repeat**
4. Merge the two *closest* cl
5. Update the proximity matrix
6. **Until** only a single cluster remains



⇒How to compute the distance between two clusters ?

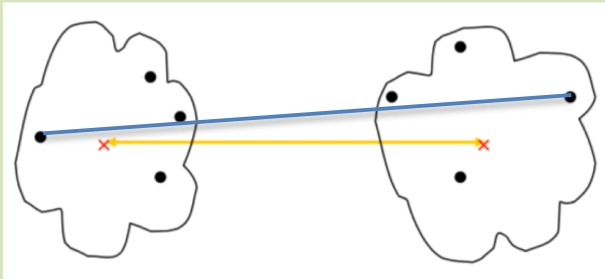
Hierarchical clustering

Agglomerative algorithm



1. Compute the proximity matrix
2. Let each data point be a clu
3. **Repeat**
4. Merge the two *closest* cl
5. Update the proximity ma
6. **Until** only a single cluster re

⇒How to compute the distance between two clusters ?



Validity of a clustering

For supervised classification we have a variety of measures to evaluate how good our model is. For instance:

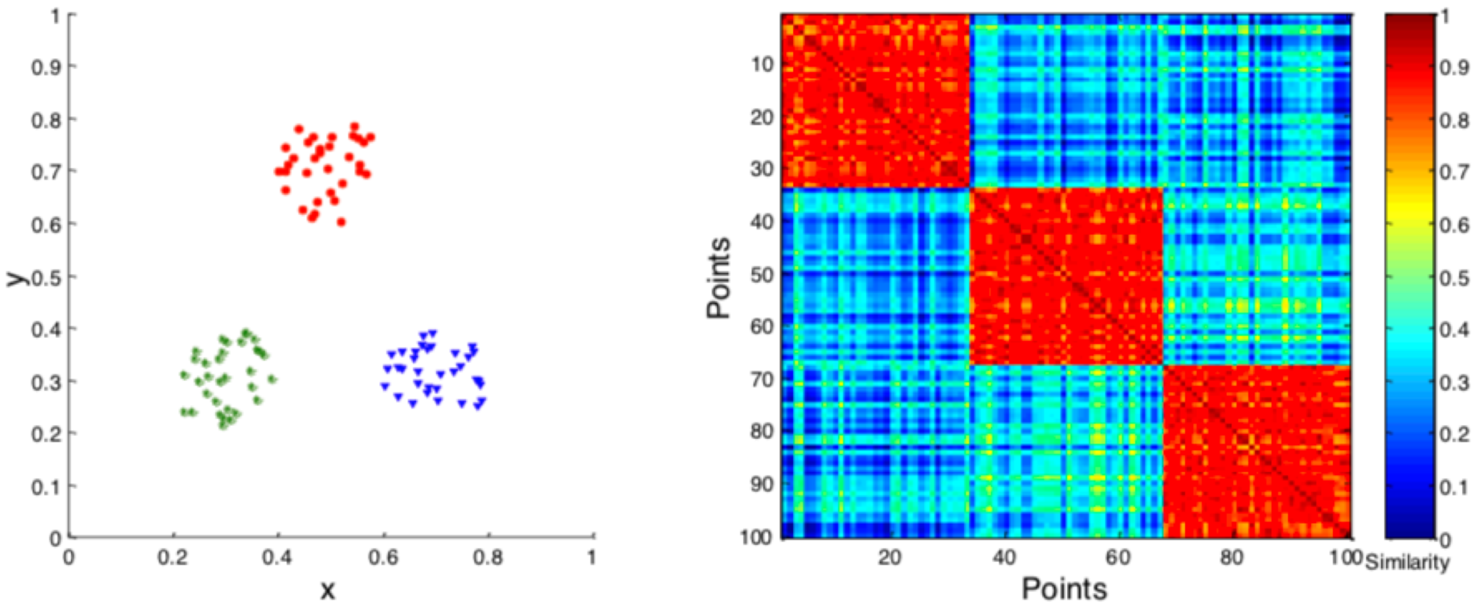
- Accuracy, precision, recall

For cluster analysis, the analogous question is how to evaluate the “goodness” of the resulting clusters?

- But “clusters are in the eye of the beholder”!
- Then why do we want to evaluate them?
  - ➢ To avoid finding patterns in noise
  - ➢ To compare clustering algorithms
  - ➢ To compare two sets of clusters

Similarity matrix

Order the similarity matrix with respect to cluster labels and inspect visually



See https://gmarti.gitlab.io/ml/2017/09/07/how-to-sort-distance-matrix.html

Conclusion

We presented two simple clustering algorithms

- Very useful to understand and summarize the data
- Can also be used to segment the samples and then design local models
- Quality assessment is hard: depend on the application.

Quizz

1. Cite one application of clustering for marketing.
2. Cite one application of clustering in image processing ?
3. What do we need to apply hierarchical clustering to genetic data ?
4. What criteria does k-means algorithm optimize ?
5. Is the result of k-means deterministic ? Why ?
6. What is the best value for k ?
7. Can you give an estimate of agglomerative hierarchical clustering complexity ?

References

Books

- T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning. Springer, 2017 https://web.stanford.edu/~hastie/ElemStatLearn/

Papers

- D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. SIAM, 2007.
- L. Bottou, Y. Bengio. Convergence properties of the K-means algorithms. NIPS’94.
- G. Hamerly. Making k-means even faster. SIMA, 2015.
- C. Elkan. Using the Triangle Inequality to Accelerate –Means. ICML. 2003.

Tutorials

- Introduction for beginners: https://www.surveygizmo.com/resources/blog/regression-analysis/
- Guide to k-means clustering (with code) https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/ or https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a
- Scikit-learn, software tools & tutorials: https://scikit-learn.org/stable/modules/clustering.html

  https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html