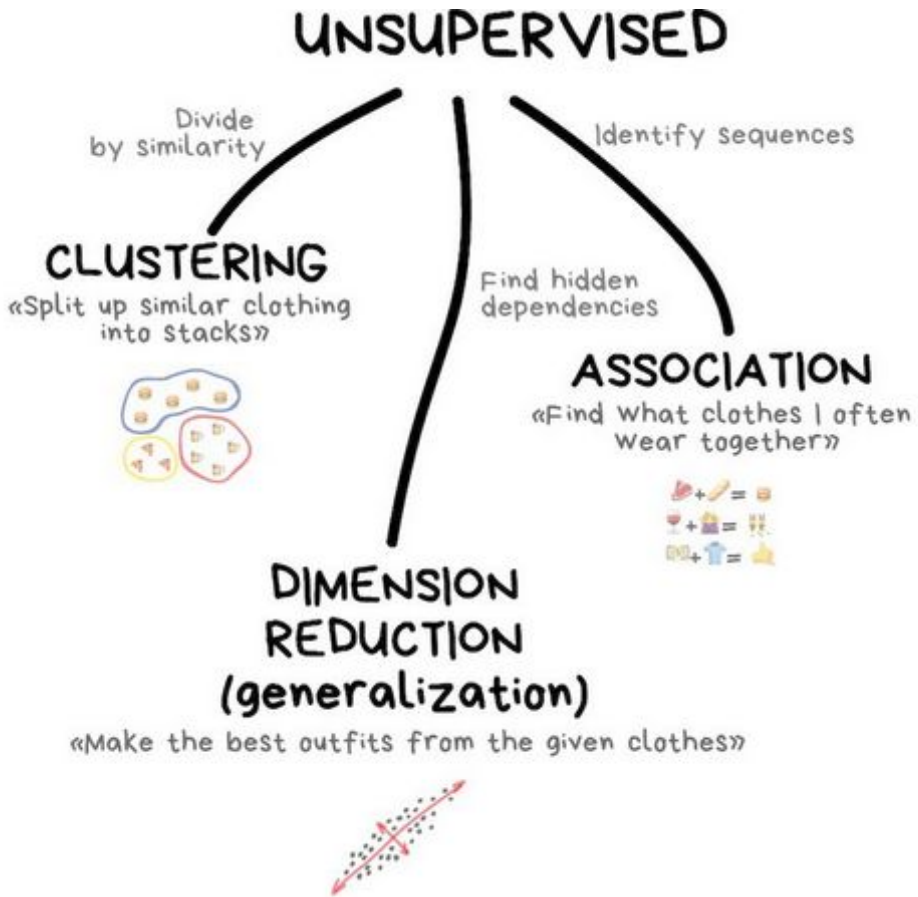
Unsupervised learning

- Labeled data is luxury
- Unlabeled data
- There is no direct measure of success -- no loss function to minimise e.g.
  $L(y, \hat{y}) = (y - \hat{y})^2$
- Part of exploratory analysis by nature



https://vas3k.com/blog/machine_learning/

Dimension reduction

Allows to:

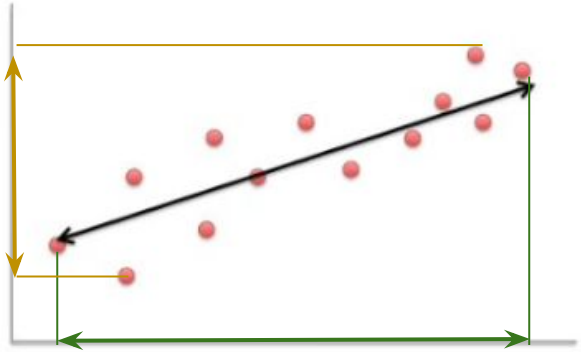
- Transform data, create new variables
- Visualise high dimensional data
- Identify outliers

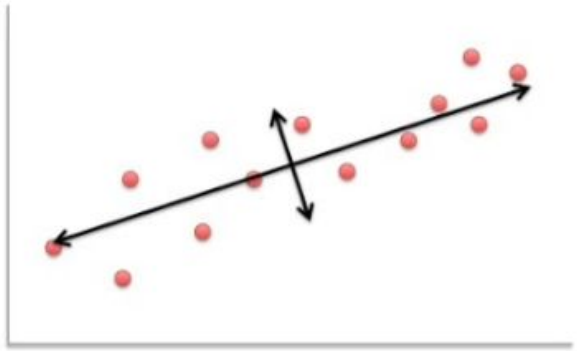
A lot of different techniques....

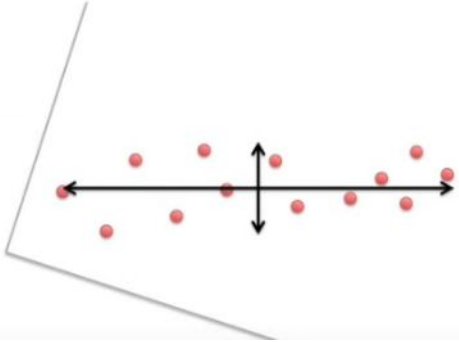
PCA, MDS, nMDS, Factor analysis, ...

PCA

- Data transformation -- creates new dimensions
- Orders new dimensions by amount of variation in it (information/importance)
- Allows for low-dimensional representation of a data
- New dimensions -- linear combination of the original *p* features



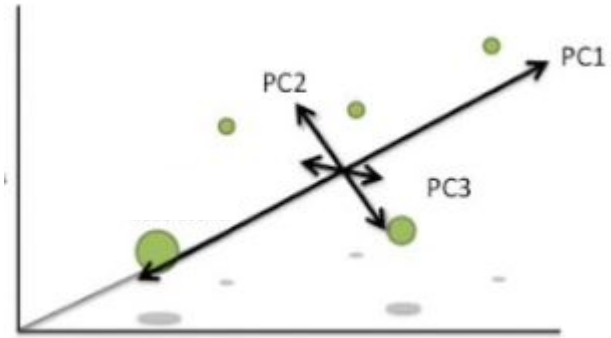


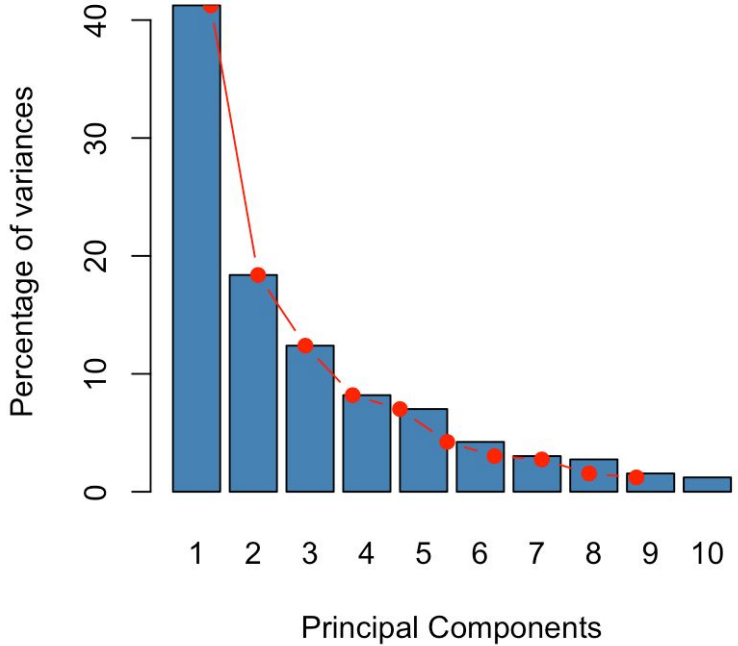


https://www.youtube.com/watch?v=_UVHneBUBW0

PCA transforms data

- We have as many dimensions as we had before
- They are ordered

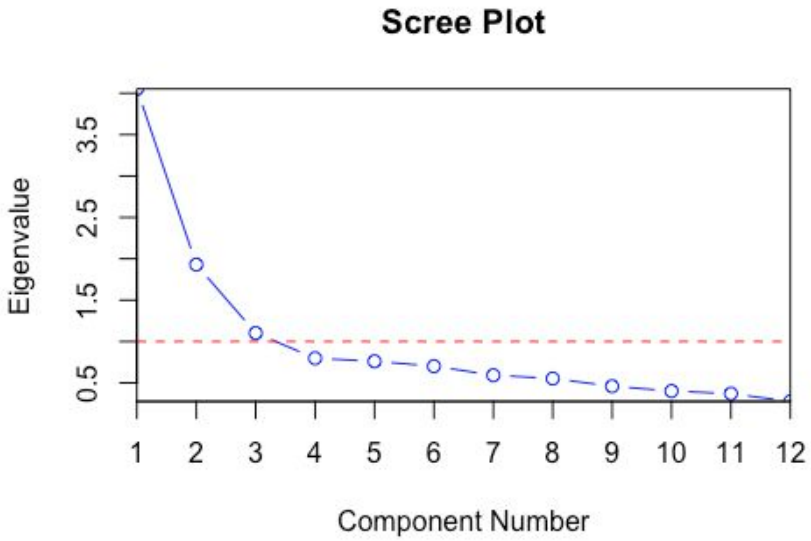


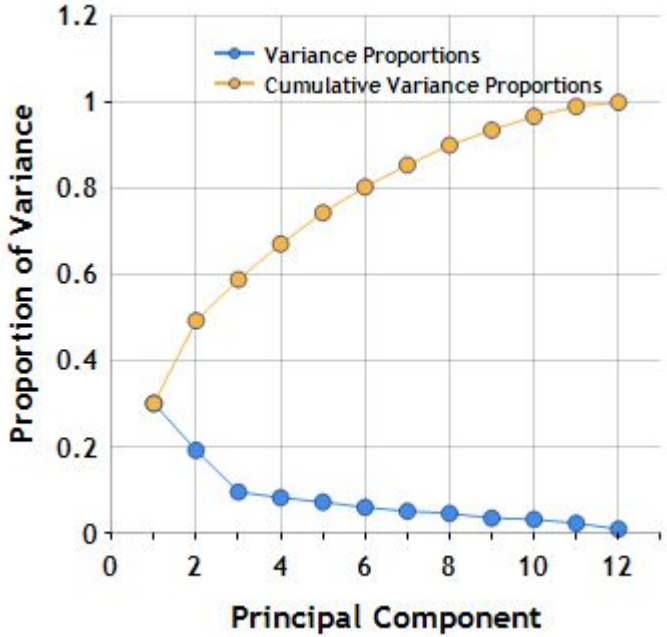


http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/119-pca-in-r-using-ade4-quick-scripts/

PCA transforms data

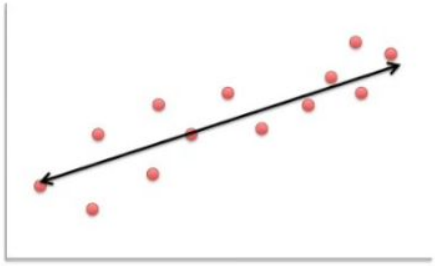
- We have as many dimensions as we had before
- They are ordered





https://machine-learning-course.readthedocs.io/en/latest/content/unsupervised/pca.html

New dimensions in PCA



- Are called principal components (PC)
- Normalized linear combination of the original *p* features

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p$$

$\phi_{11}, \ldots, \phi_{p1}$ -- are called **loadings**

$$\phi_1 = (\phi_{11} \ \phi_{21} \ \ldots \ \phi_{p1})^T$$

Normalized, because

$$\sum_{j=1}^{p} \phi_{j1}^2 = 1$$

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

Different mathematical notations

Ridge

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$$

$$\underset{\beta}{\text{minimize}}\left\{\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2\right\}\quad\text{subject to}\quad\sum_{j=1}^{p}\beta_j^2\le s$$

PCA

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}}\left\{\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2\right\}\ \text{subject to}\ \sum_{j=1}^{p}\phi_{j1}^2=1$$

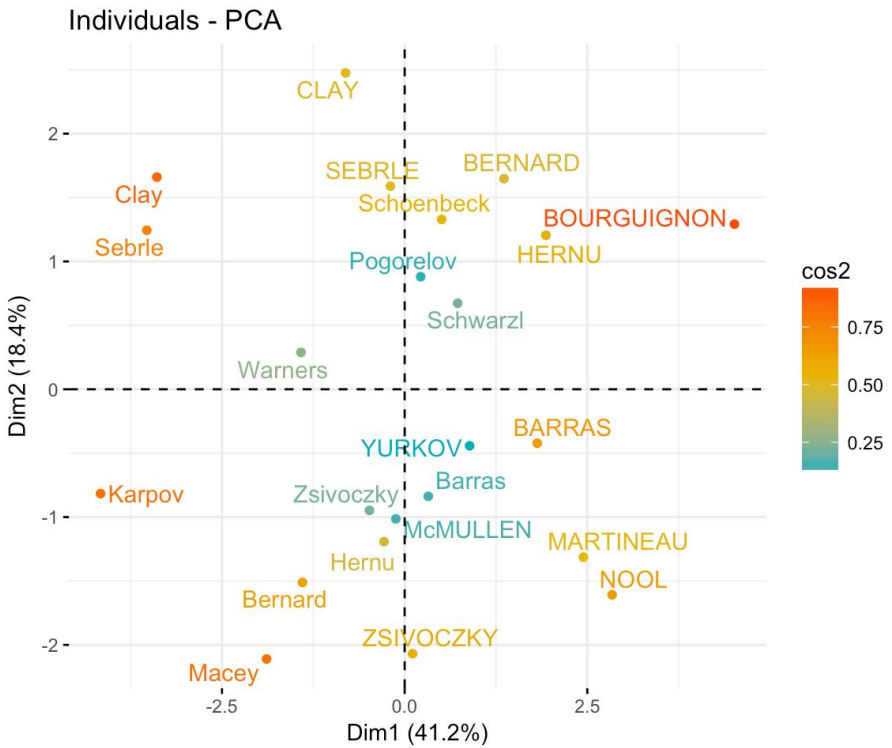
Different mathematical notations

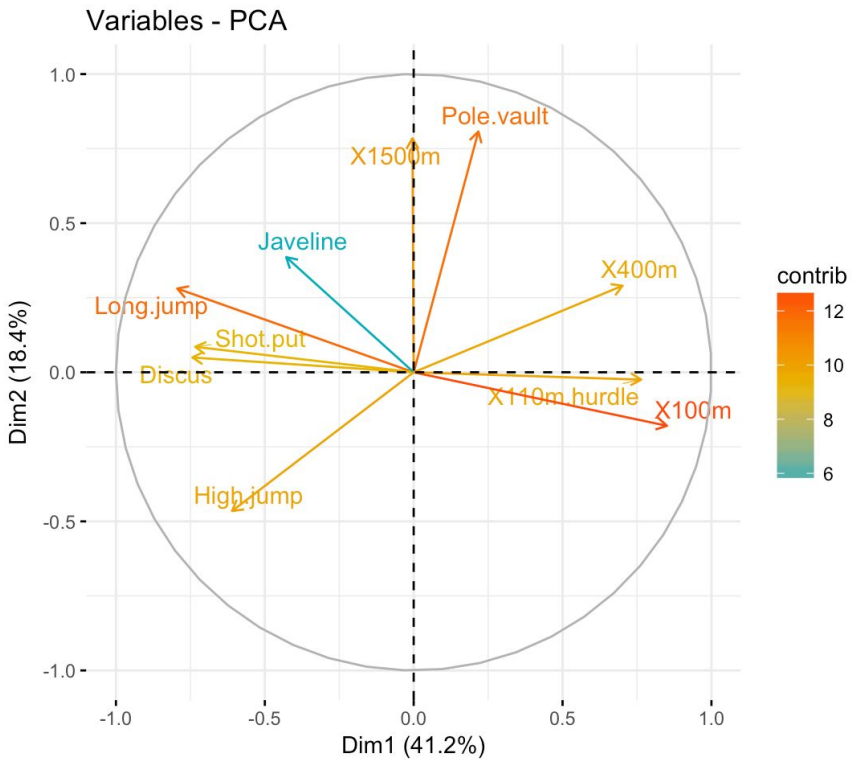
$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

$$\begin{aligned} \underset{\mathbf{c}}{\text{maximize}} \quad & \mathbf{c}^T \mathbf{X}^T \mathbf{X} \mathbf{c} \\ \text{subject to} \quad & \mathbf{c}^T \mathbf{c} = 1 \end{aligned}$$

- eigendecomposition
- computed via the singular value decomposition (SVD)

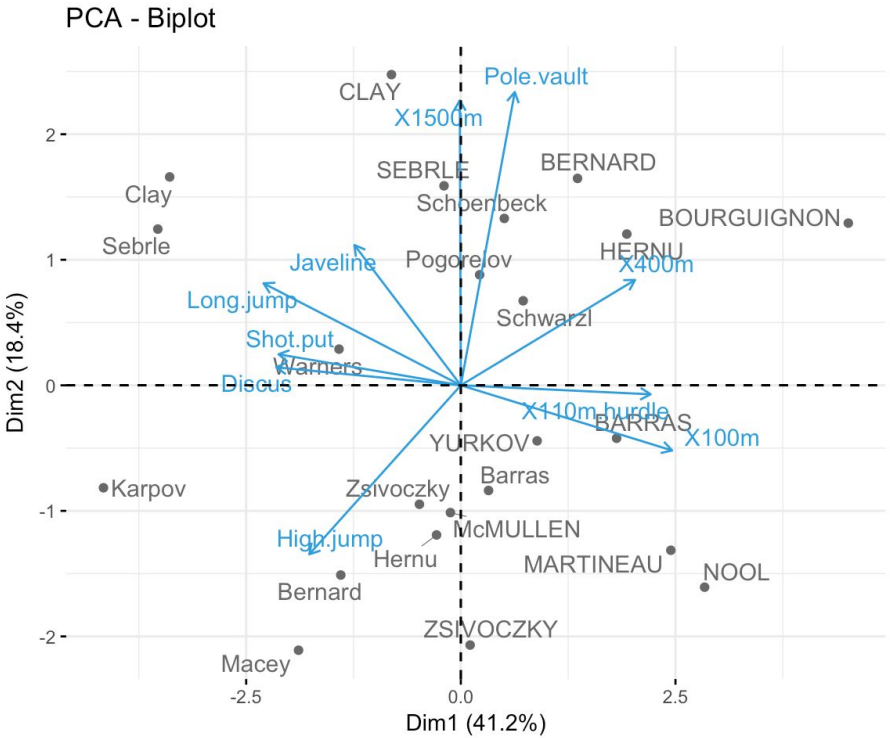
PCA visualization

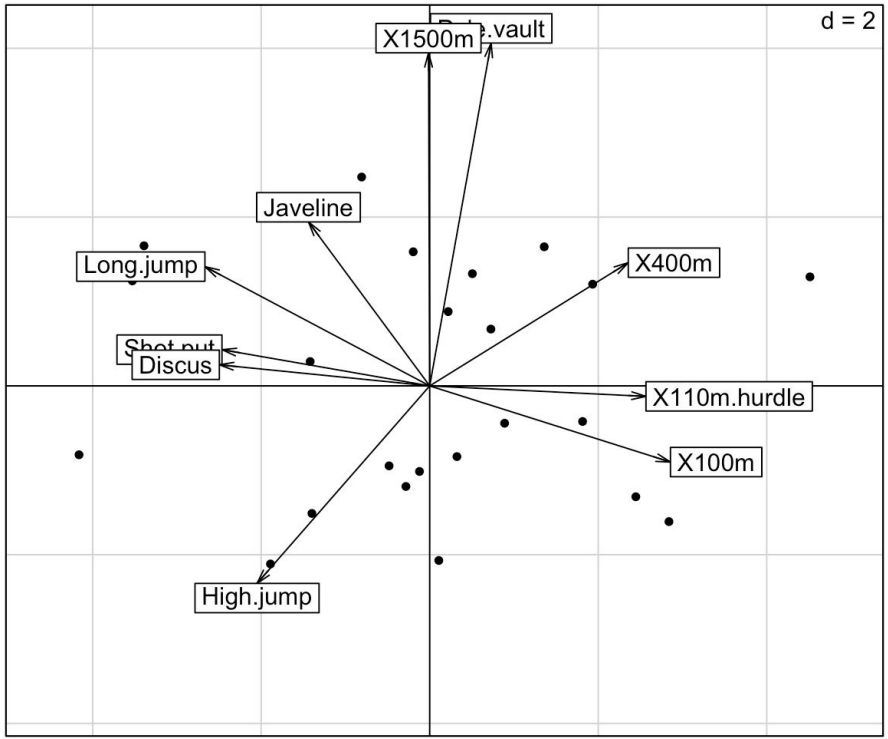




http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/119-pca-in-r-using-ade4-quick-scripts/

PCA visualization





http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/119-pca-in-r-using-ade4-quick-scripts/

Variations

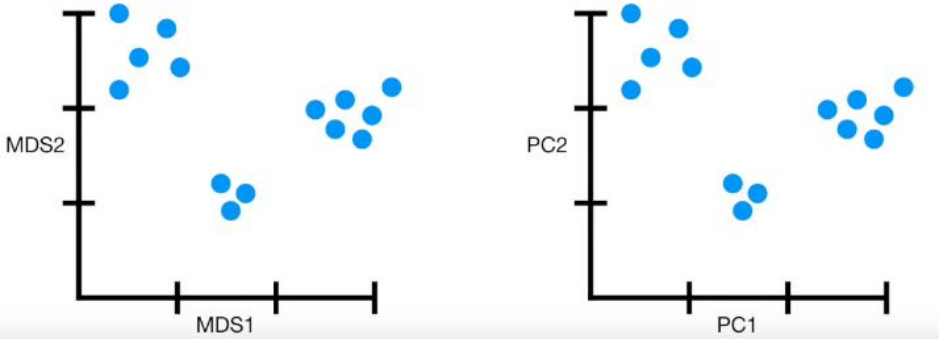
- quadratically regularized PCA
- Sparse PCA
- Nonlinear PCA
- Logistic PCA
- Robust PCA
- Poisson PCA and PCA on ordinal data
- Non-negative matrix factorization
- ...

Multi-Dimensional Scaling (MDS)

- Same as Principal coordinate Analysis
- Very similar to PCA
- Difference: use distance instead of covariance matrix
  (for features, not observations)

Multi-Dimensional Scaling (MDS)

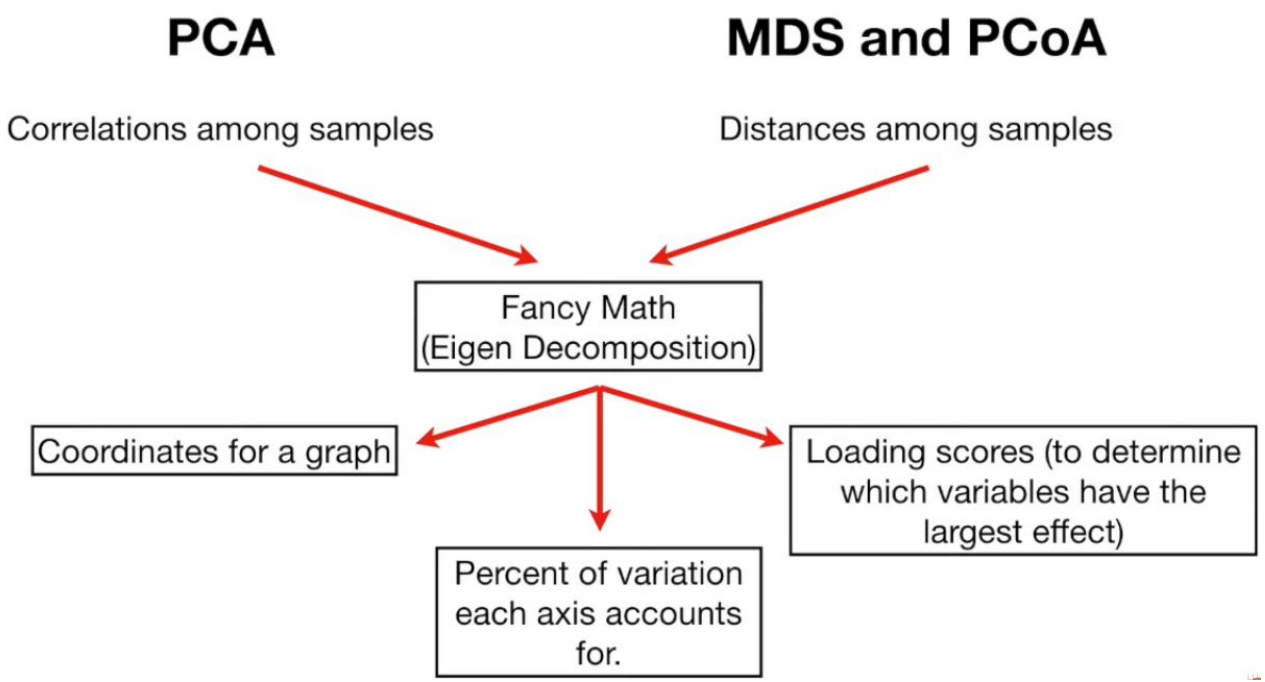
- Same as Principal coordinate Analysis
- Very similar to PCA
- Difference: use distance instead of covariance matrix
  (for features, not observations)
- If we use Euclidean distance: PCA == MDS



Minimizing linear distances is the same as maximizing the linear correlations

Distances

- Euclidean,
- Manhattan,
- Hamming,
- Great distance,
- Log fold change,
- *ect*.



https://www.youtube.com/watch?v=GEn-_dAyYME&t=6s

Non-metric Multi-Dimensional scaling (nMDS)

- Allows for non linear relationship
- Uses iterations to estimate good solution
- Arranges points to maximize rank-order correlation between real world distance and ordination space distance

	Fixed distance	User defined distance
Eigenanalysis	PCA	MDS
Iterative, non-metric		nMDS

nMDS

Arranges points to maximize rank-order correlation between real world distance and ordination space distance

Multivariate data

	Sp A	Sp B	Sp C
Site 1	...	...	...
Site 2	...	...	...
Site 3	...	...	...

User-defined distance matrix (real space)

	Site 1	Site 2	Site 3
Site 1	...		
Site 2	...	...	
Site 3	...	...	...

nMDS

Arranges points to maximize rank-order correlation between real world distance and ordination space distance

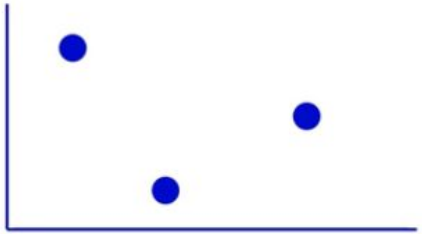
Multivariate data

	Sp A	Sp B	Sp C
Site 1	...	...	...
Site 2	...	...	...
Site 3	...	...	...

User-defined distance matrix (real space)

	Site 1	Site 2	Site 3
Site 1	...		
Site 2	...	...	
Site 3	...	...	...

Ordination space



We define how many dimensions we want beforehand

Euclidean distance matrix (ord. space)

	Site 1	Site 2	Site 3
Site 1	...		
Site 2	...	...	
Site 3	...	...	...

nMDS

Arranges points to maximize rank-order correlation between real world distance and ordination space distance

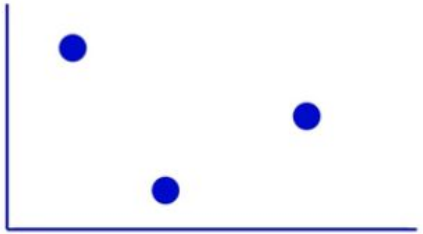
Multivariate data

	Sp A	Sp B	Sp C
Site 1	...	...	...
Site 2	...	...	...
Site 3	...	...	...

User-defined distance matrix (real space)

	Site 1	Site 2	Site 3
Site 1	...		
Site 2	...	...	
Site 3	...	...	...

Ordination space



We define how many dimensions we want beforehand

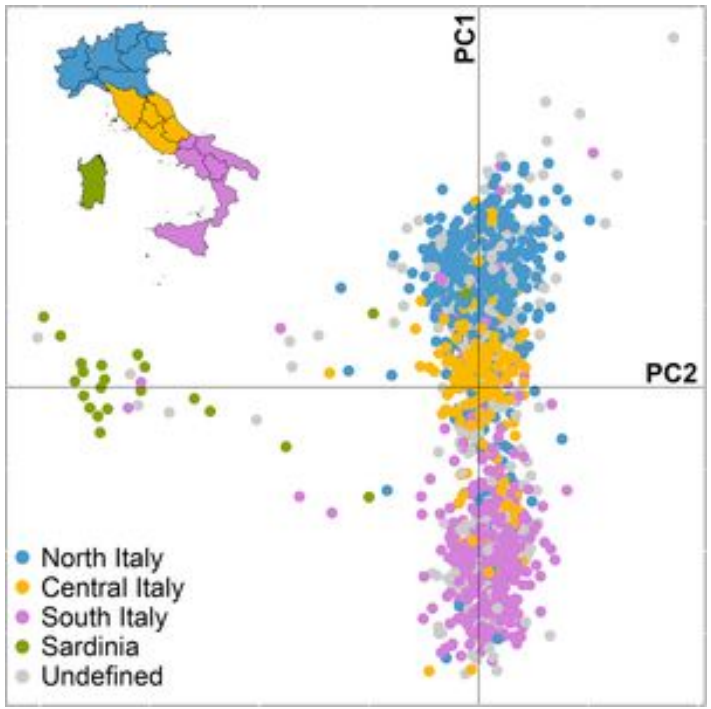
Euclidean distance matrix (ord. space)

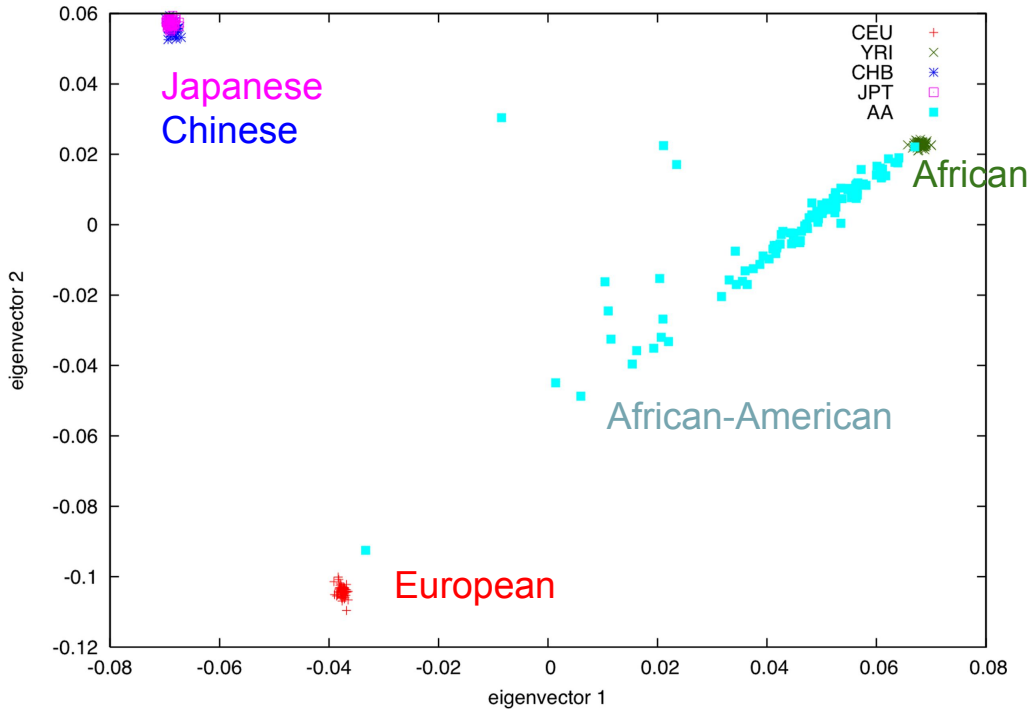
	Site 1	Site 2	Site 3
Site 1	...		
Site 2	...	...	
Site 3	...	...	...

nMDS

- The number of new dimensions has to be specified beforehand
- Finds local, not global solution
- Non linearity, works well with certain types of data e.g. abundance counts
- User specified distance measurements

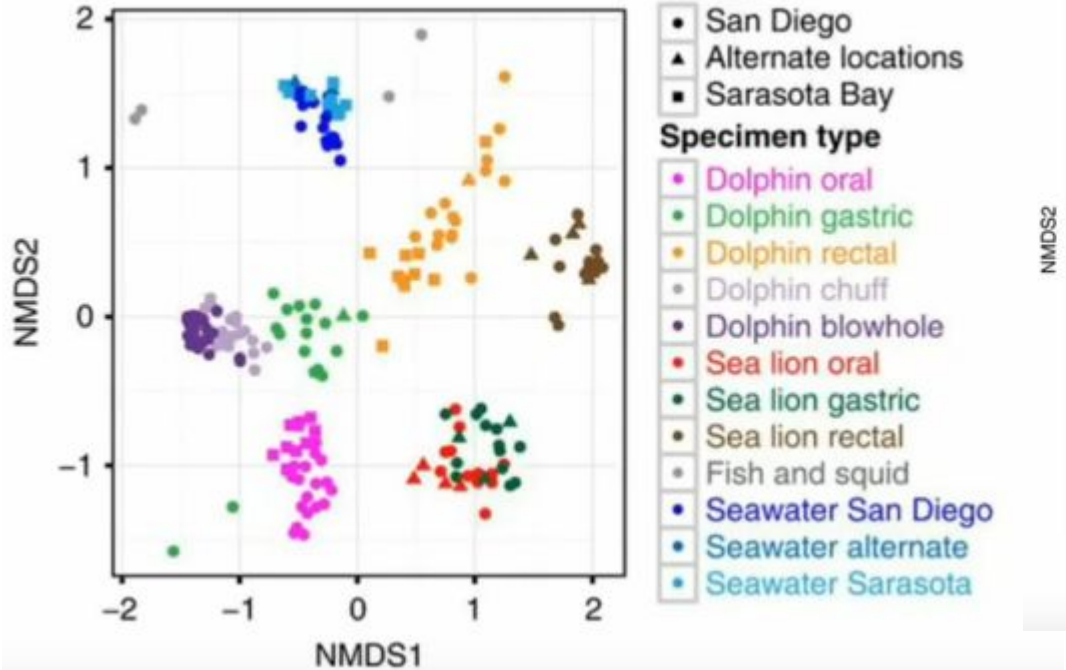
Gradients and structure



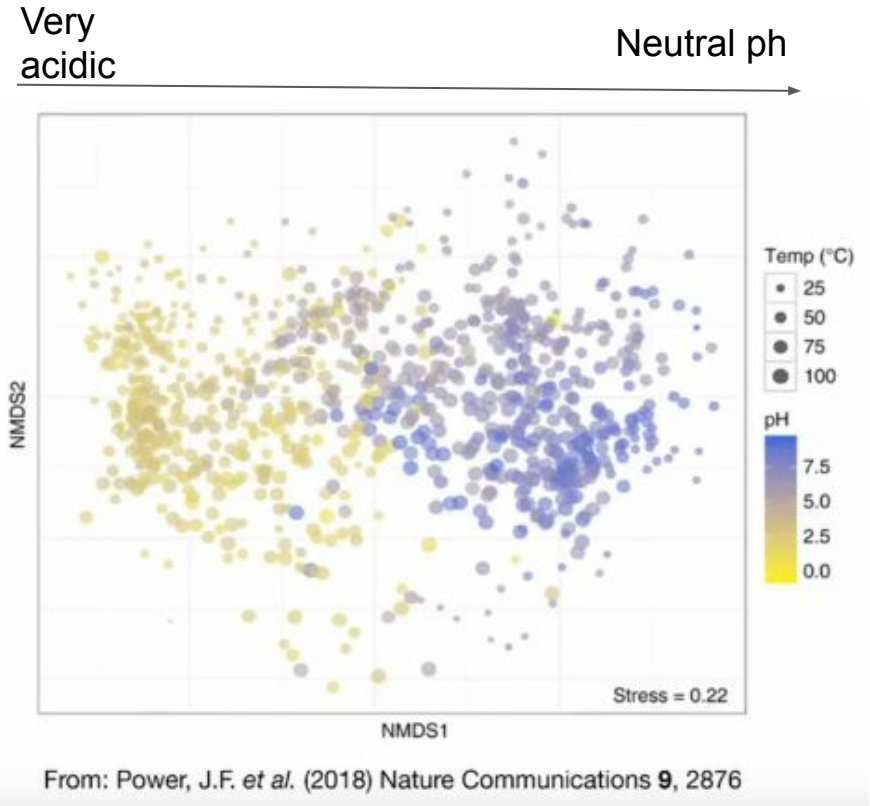


https://bmcgenet.biomedcentral.com/articles/10.1186/s12863-015-0293-x

https://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1000294



https://www.nature.com/articles/ncomms10516

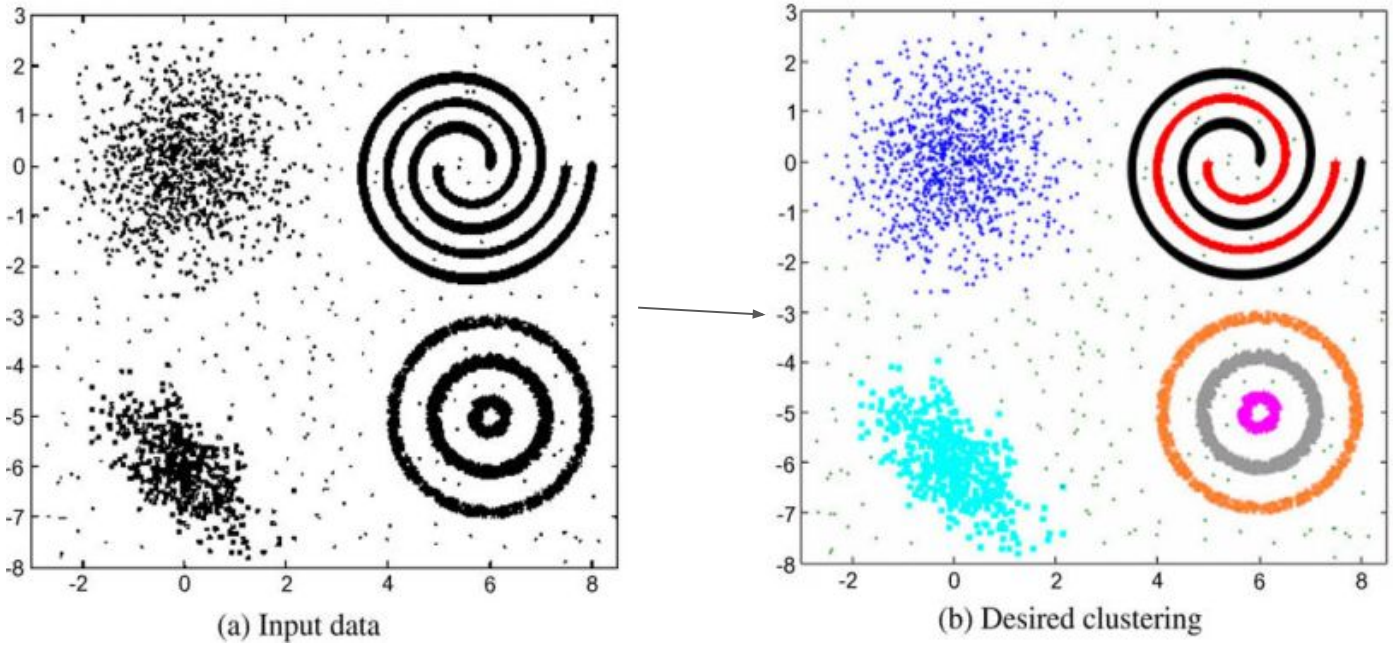


From: Power, J.F. *et al.* (2018) Nature Communications **9**, 2876

https://www.nature.com/articles/s41467-018-05020-y/

Clustering

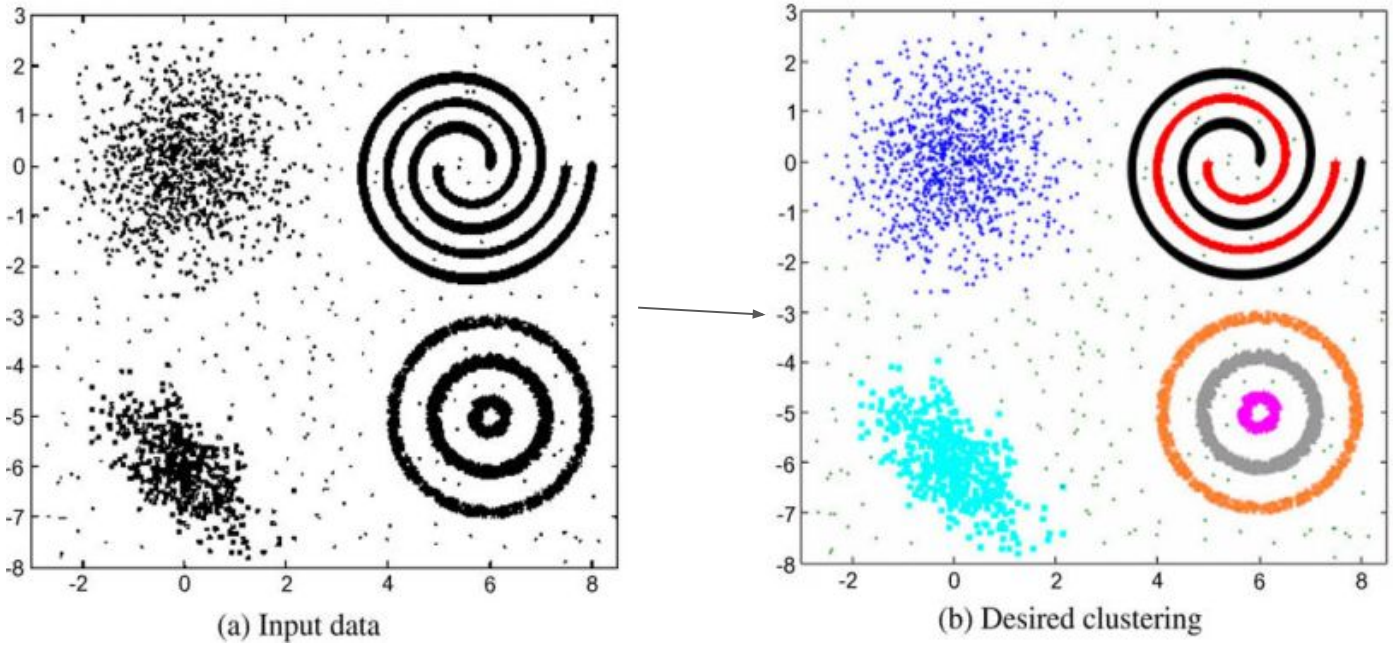
GOAL: group similar stuff together



Jain, A. K. (2010). *Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8), 651–666.* doi:10.1016/j.patrec.2009.09.011

Clustering

GOAL: group similar stuff together → find structure in data



Jain, A. K. (2010). *Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8), 651–666.* doi:10.1016/j.patrec.2009.09.011

Clustering

a task that partitions sample data into clusters, and groups a set of objects in such a way that members in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups.

- Popular in many fields -- Great number of methods

- A lot of application:
  - Market segmentation
  - Detect abnormal behavior
  - image compression
  - label new data
  - ...

Clustering

a task that partitions sample data into clusters, and groups a set of objects in such a way that members in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups.

- Popular in many fields -- Great number of methods

- A lot of application:
  - Market segmentation
  - Detect abnormal behavior
  - image compression
  - label new data
  - ...

Some clustering techniques:

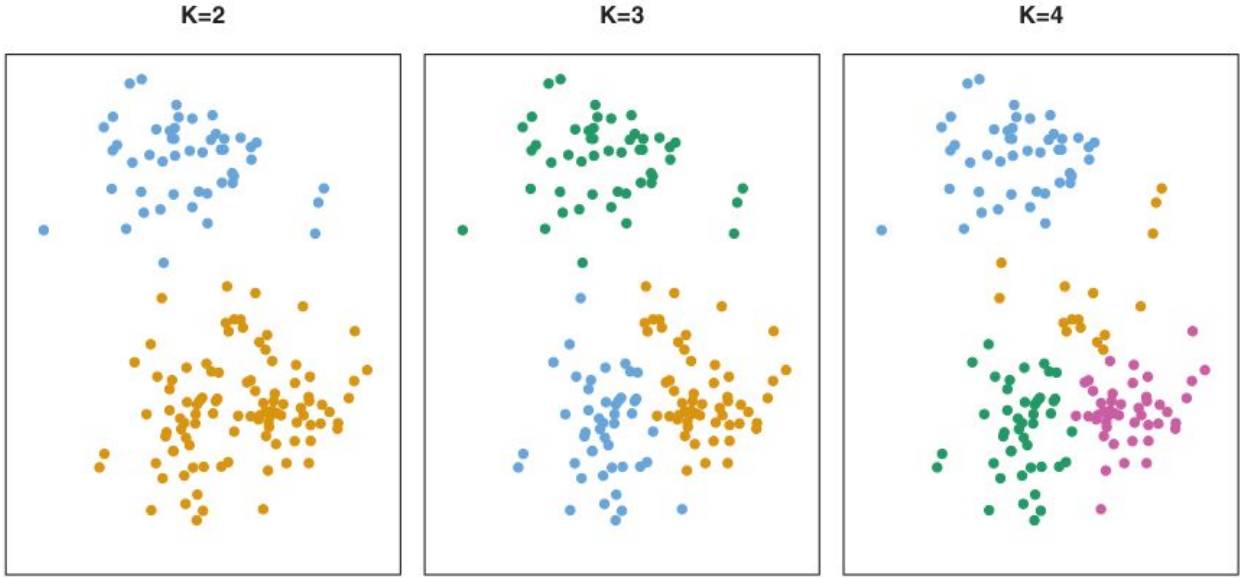
- K-means clustering,
- hierarchical clustering,
- Mean-shift clustering,
- Density-based spatial clustering of applications with noise (DBSCAN),
- Expectation–Maximization (EM) Clustering using Gaussian Mixture Models (GMM)
- ...

K-means clustering

- One of most popular and simple methods
- First published in 1955
- Still widely used, a lot of variations for original algorithm

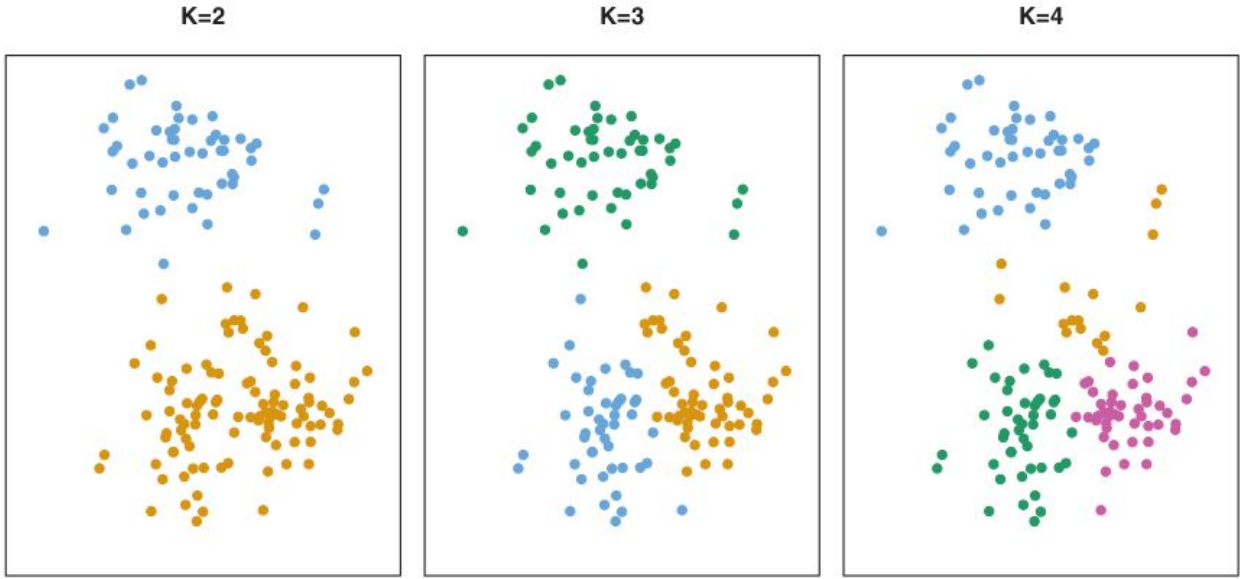
K-means clustering

- Partitions data into K distinct, non-overlapping clusters
- We must first specify the desired number of clusters K



K-means clustering

- Partitions data into K distinct, non-overlapping clusters
- We must first specify the desired number of clusters K



Observations within cluster are more similar to each other than to those in other groups

K-means clustering – defining the problem

Good clusters:

- Within cluster variation is minimal

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

Clusters from 1 to K

Within cluster variation measurement

K-means clustering – defining the problem

Good clusters:

- Within cluster variation is minimal

Define within cluster variation:

- squared Euclidean distance (most common choice)

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

Clusters from 1 to K

Within cluster variation measurement

K-means clustering – defining the problem

Good clusters:

- Within cluster variation is minimal

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

Clusters from 1 to K

Within cluster variation measurement

Define within cluster variation:

- squared Euclidean distance (most common choice)

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

i is an observation

# of observations belonging to cluster *k*

Some predictor *j*

i' is any other observation that is not *i*

K-means clustering – defining the problem

Good clusters:

- Within cluster variation is minimal

Define within cluster variation:

- squared Euclidean distance (most common choice)

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

Clusters from 1 to K

Within cluster variation measurement

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

K-means clustering – defining the problem

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

- Very difficult problem to solve precisely
  (almost K^n ways to partition n observations into K clusters)

K-means clustering – defining the problem

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

- Very difficult problem to solve precisely
  (almost K^n ways to partition n observations into K clusters)

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2,$$

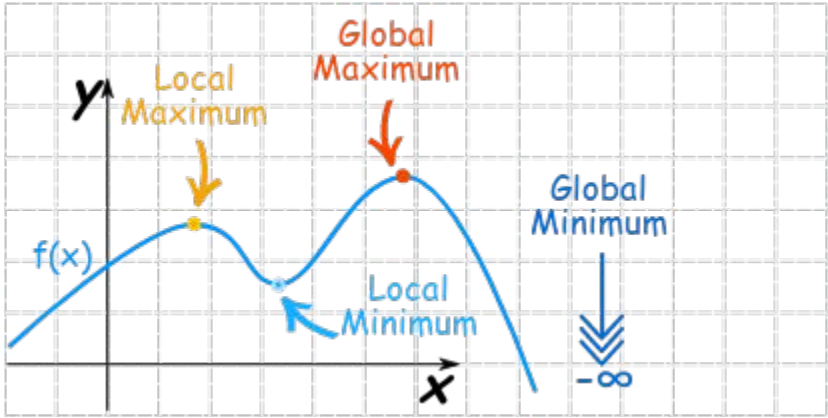
K-means clustering – defining the problem

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

- Very difficult problem to solve precisely
  (almost K^n ways to partition n observations into K clusters)

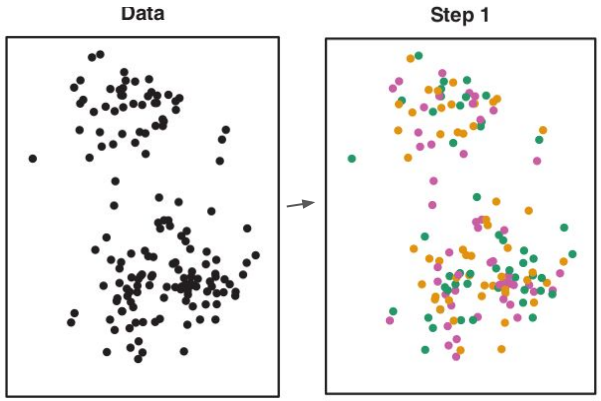
- Find local optimum i.e.
  not THE BEST, but a pretty good solution

$$2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2,$$

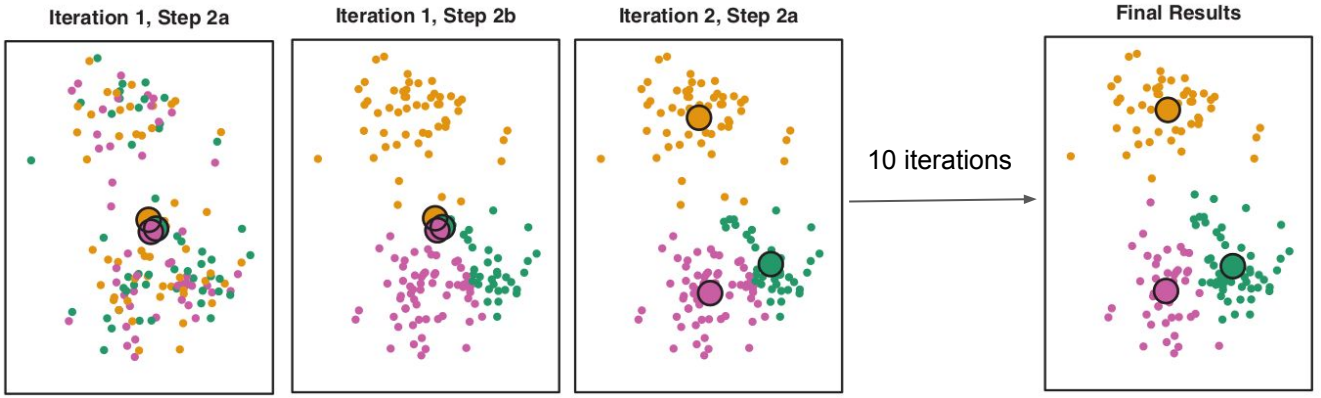


https://www.mathsisfun.com/algebra/functions-maxima-minima.html

K-means clustering – algorithm

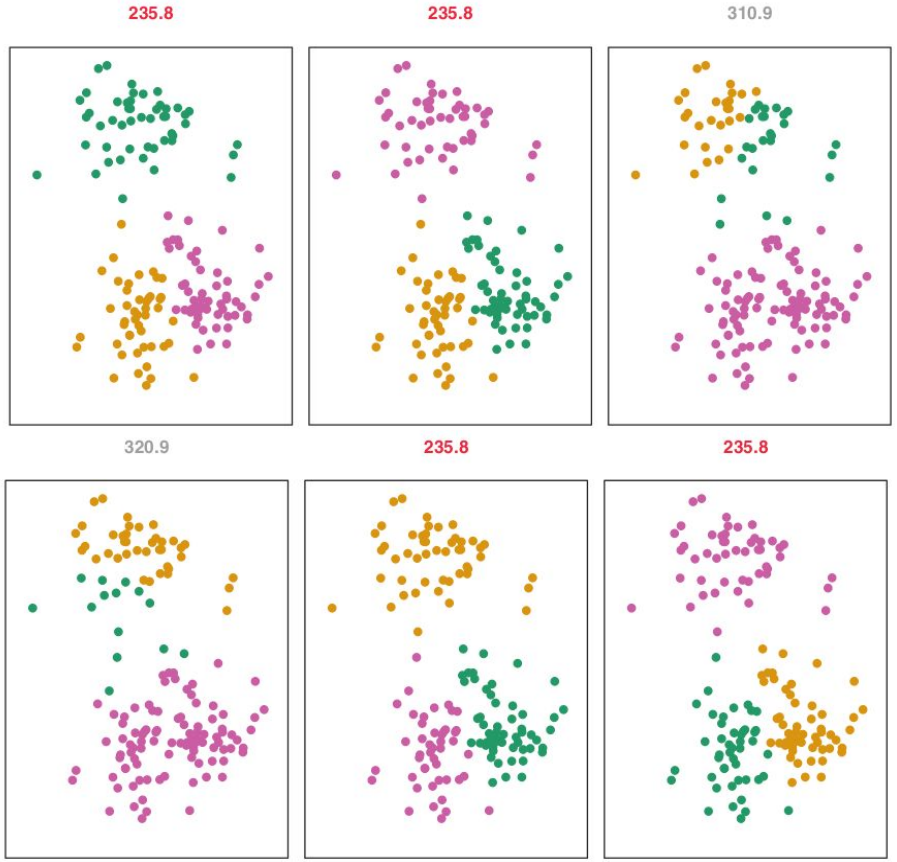


1. Randomly assign a number, from 1 to K, to each of the observations
2. Iterate until the cluster assignments stop changing:
   a. For each of the K clusters, compute the cluster centroid.
   b. Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).



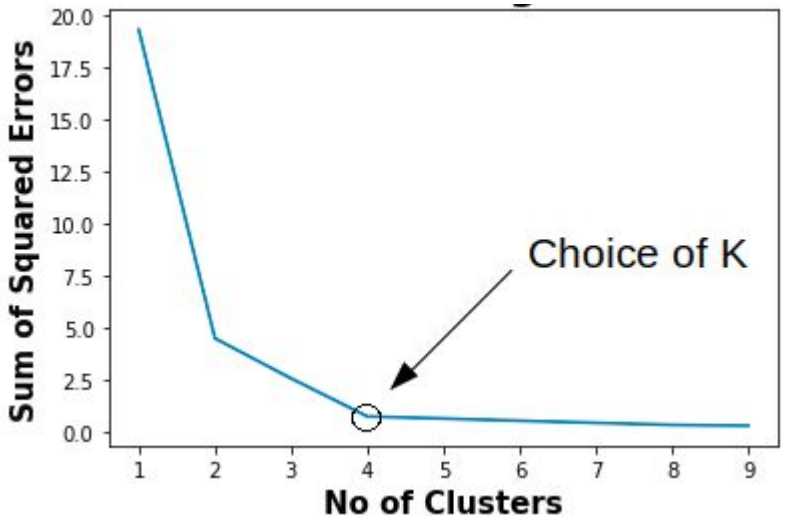
K-means clustering – algorithm

- Finds local minimum, that depends on initial random class assignment
- Need to run clustering algorithm several times with different initial values and then select the best one



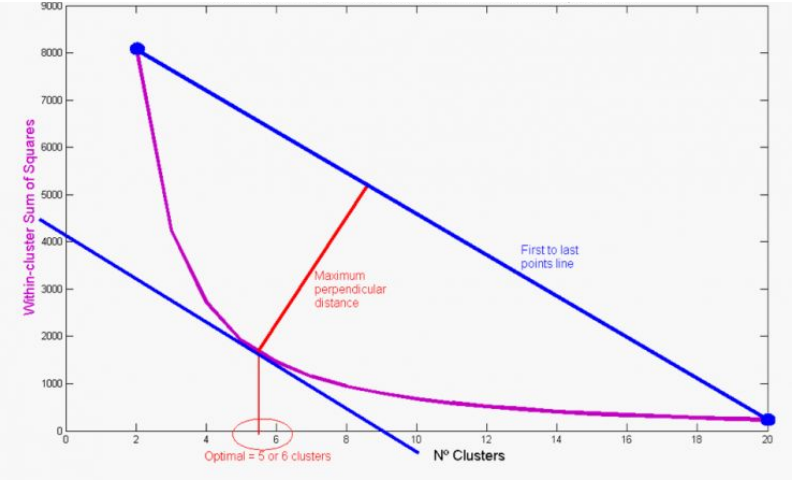
K-means clustering – How to choose K?

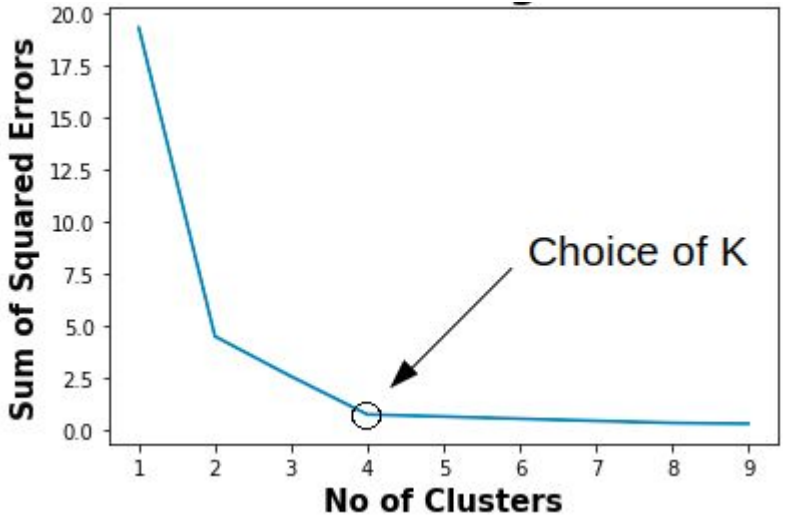
- We need to define number of clusters K beforehand
  - Prior knowledge
  - “Elbow method”



K-means clustering – How to choose K?

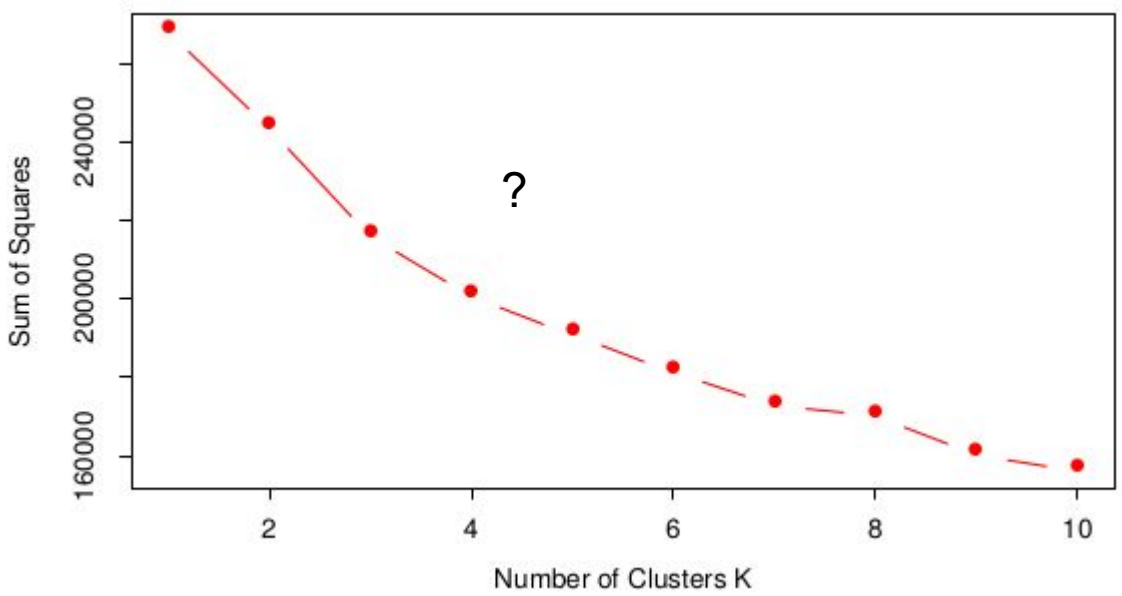
- We need to define number of clusters K beforehand
  - Prior knowledge
  - “Elbow method”





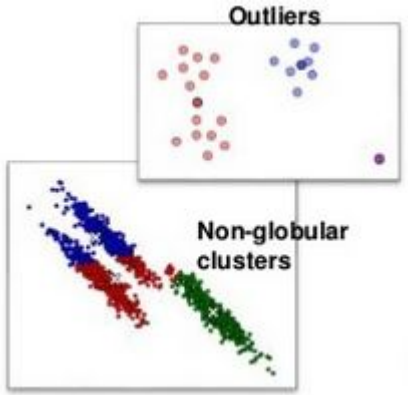
K-means clustering – How to choose K?

- We need to define number of clusters K beforehand
  - Prior knowledge
  - “Elbow method”

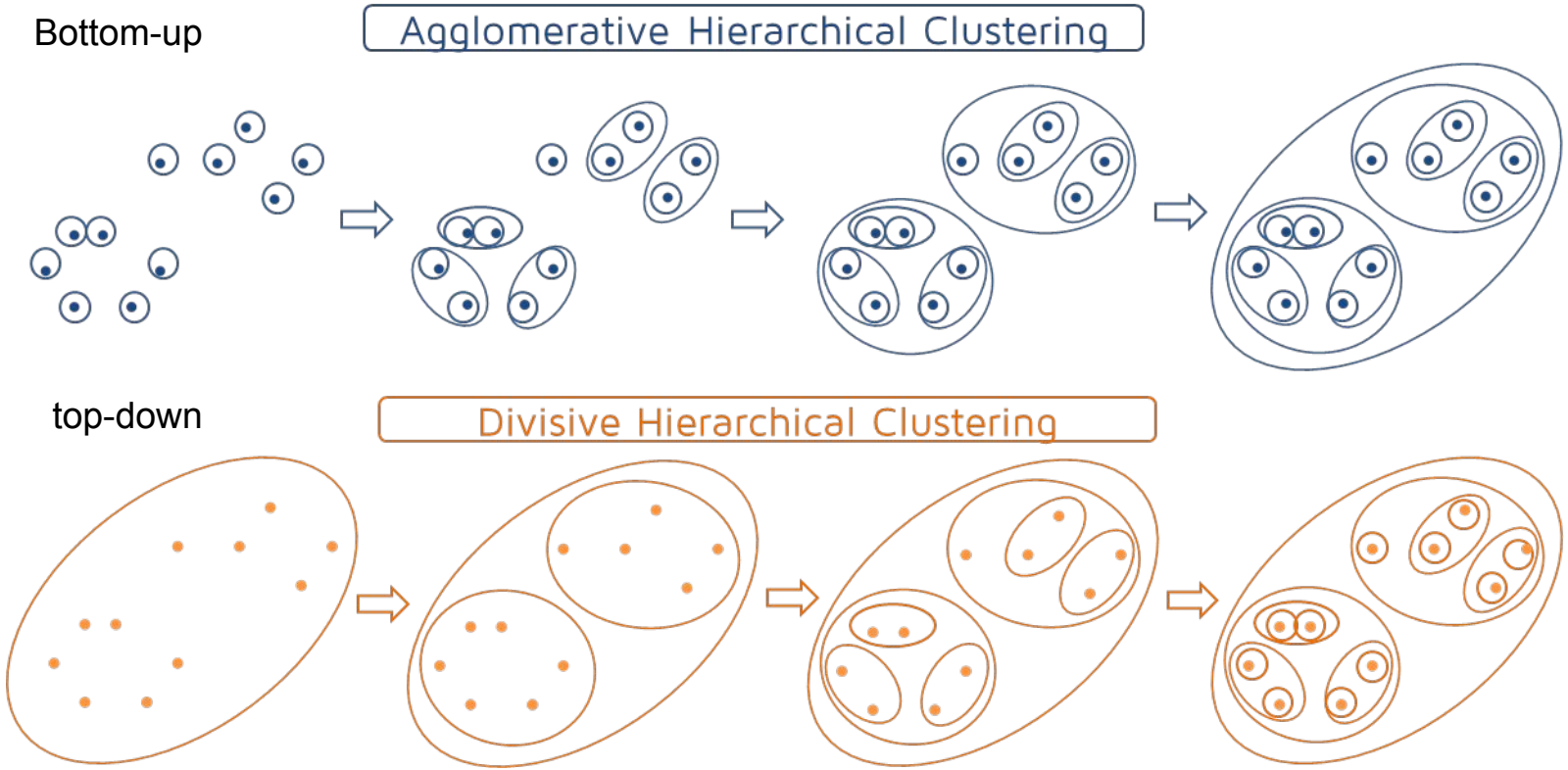


K-means clustering – application issues

- Versatile in usage
- High scalability (performs well when used on large samples)
- Not very robust to perturbations to the data (remove random set of observations n)
- Clusters can be heavily distorted if outliers are present
- Bad with clusters that are not in globular shape
- Not applicable to categorical data
- Curse of dimensionality: inflation of Euclidean distance

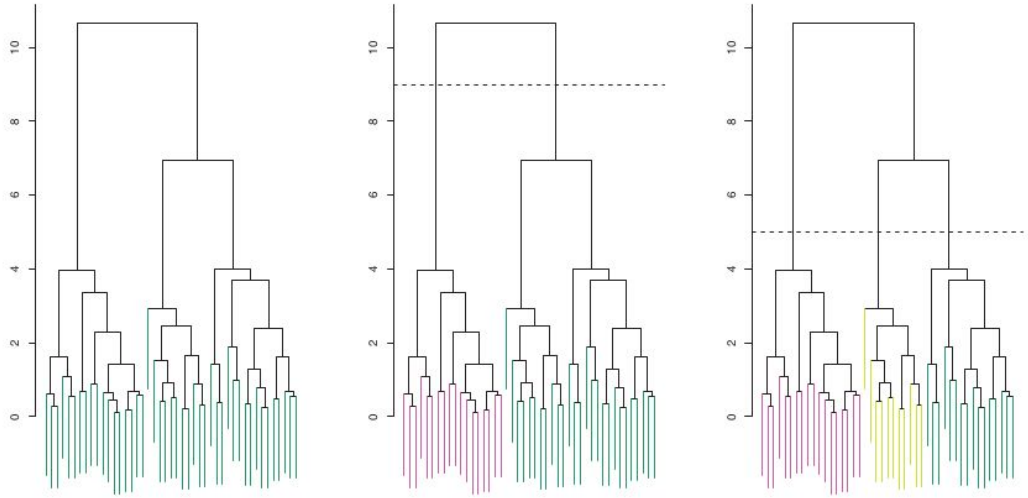


Hierarchical clustering



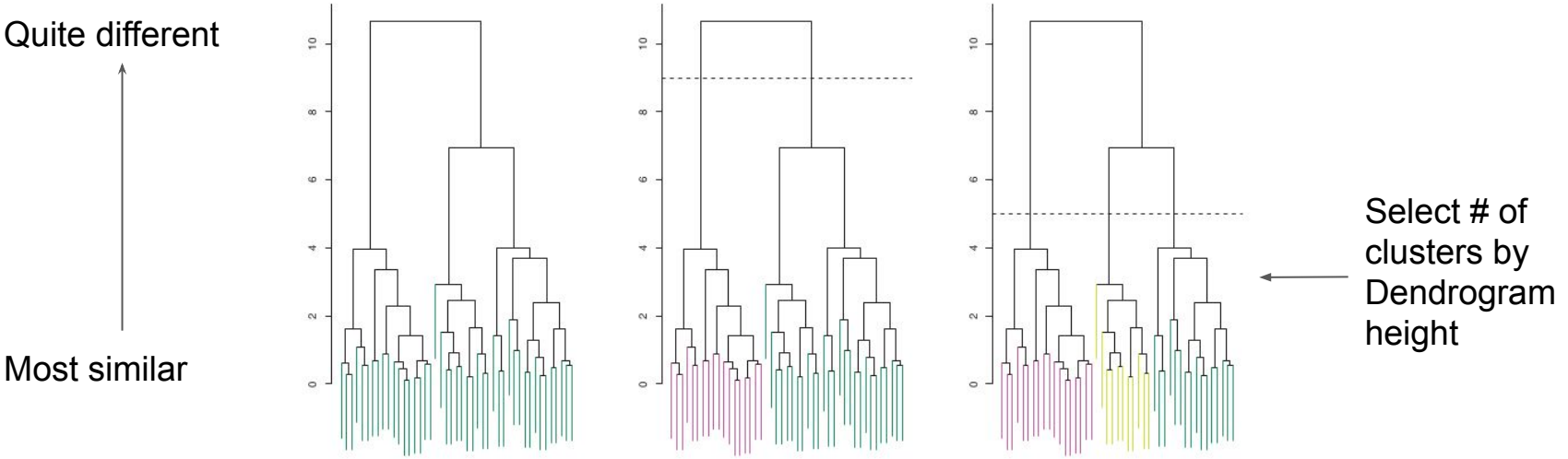
Hierarchical agglomerative clustering (HAC)

- Does not require to specify number of clusters K beforehand
- bottom-up (agglomerate)
- hierarchy of clusters is represented as a tree (dendrogram)

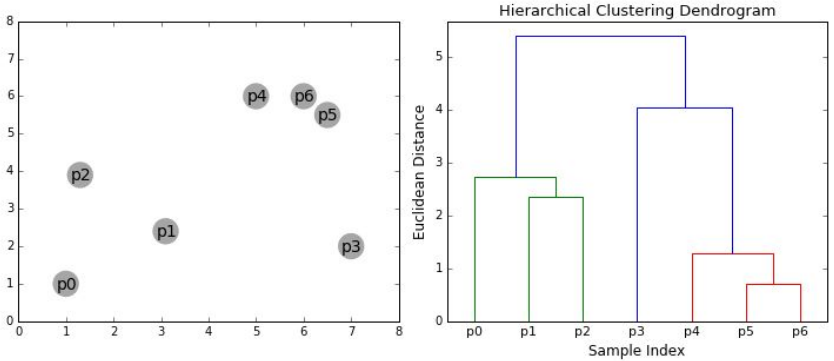


Hierarchical agglomerative clustering (HAC)

- Does not require to specify number of clusters K beforehand
- bottom-up (agglomerate)
- hierarchy of clusters is represented as a tree (dendrogram)

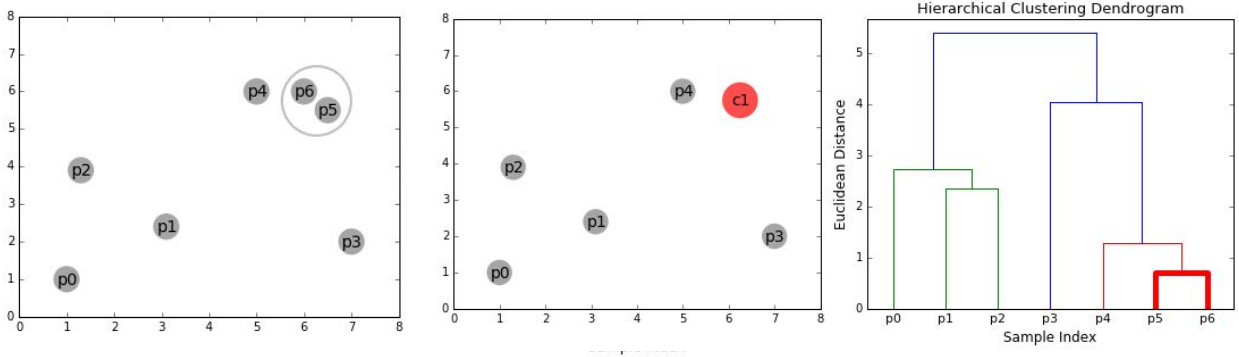


Hierarchical clustering



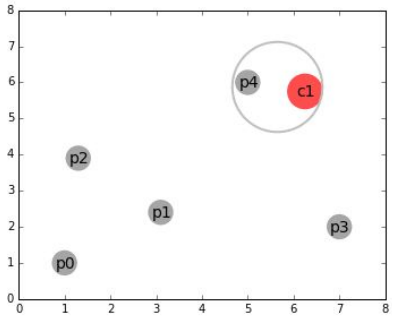
1. Each observation is treated as its own cluster
2. Find pairwise dissimilarity matrix between all clusters (need to define dissimilarity measure)
3. For i = n, n-1, n-2, 2:
   a. Identify the pair of clusters that are least dissimilar (most similar). Fuse these two clusters.
   b. Compute new pairwise inter-cluster dissimilarity matrix among the i − 1 remaining clusters.

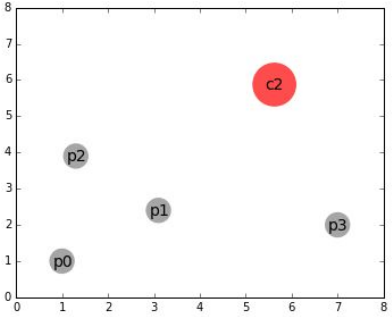
i = n

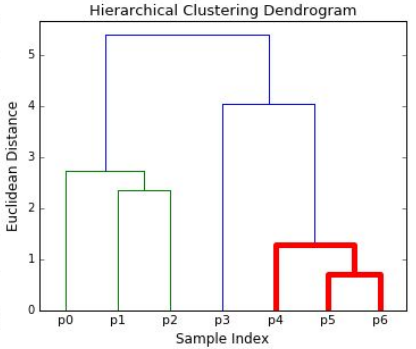


Hierarchical clustering

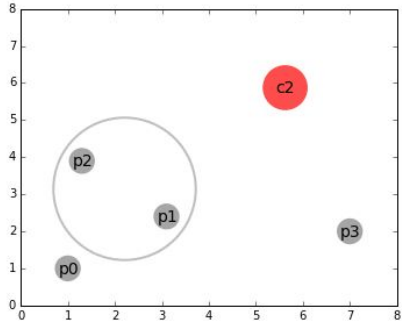
i = n-1

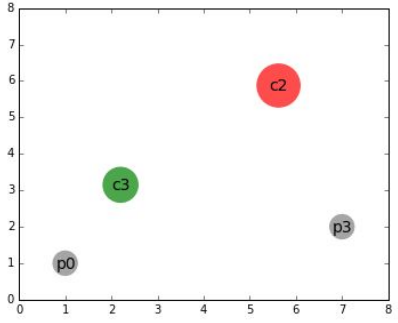


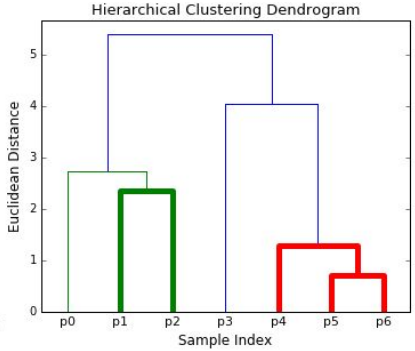




i = n-2

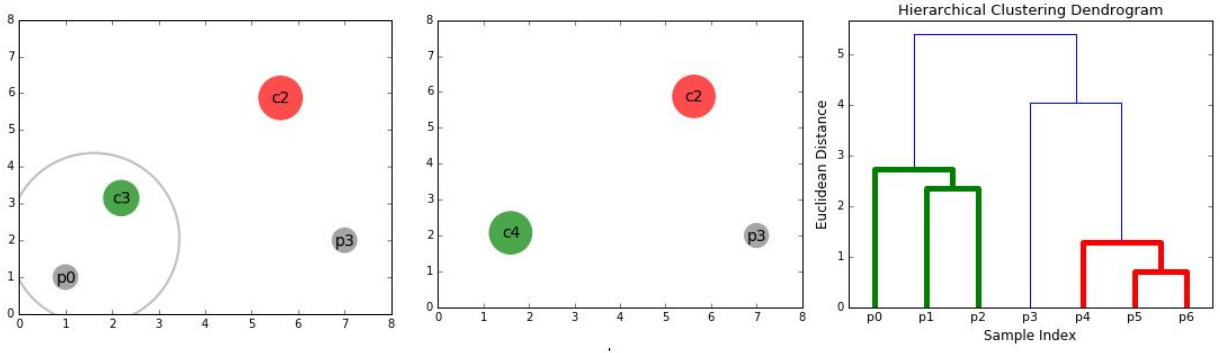




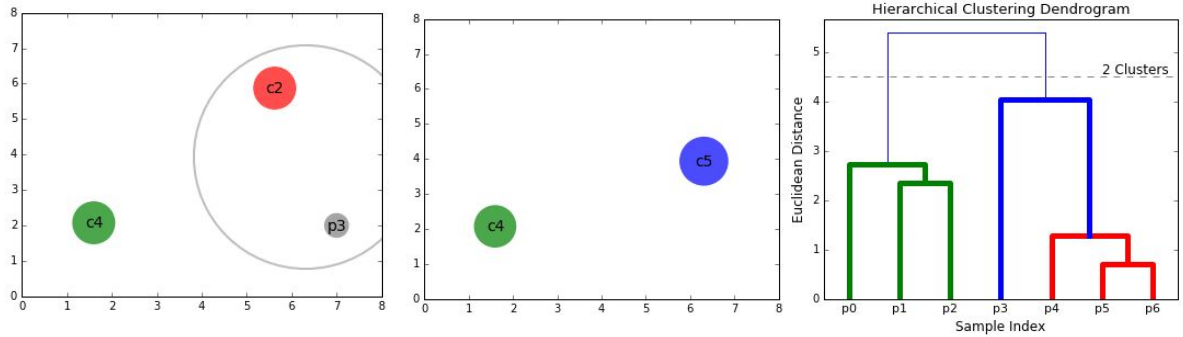


Hierarchical clustering

i = n-3



i = n-4



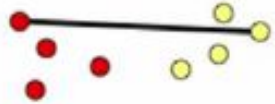
Hierarchical clustering – linkage

- **Single link** *(nearest neighbor)*:
  - Distance between closest elements in clusters
  - Produces long chains

$$D(c_1, c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$$

- **Complete link** *(furthest neighbor)*:
  - Distance between farthest elements in clusters

$$D(c_1, c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1, x_2)$$

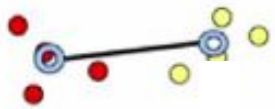


- **Average link**
  (unweighted pair-group average, UPGMA):
  - Average all pairwise distances
  - Less affected by outliers

$$D(c_1, c_2) = \frac{1}{|c_1|}\frac{1}{|c_2|}\sum_{x_1 \in c_1}\sum_{x_2 \in c_2} D(x_1, x_2)$$

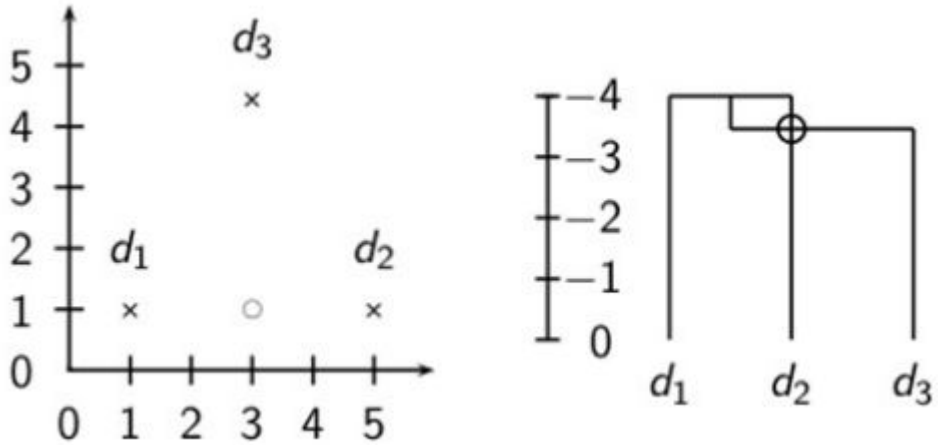
- **Centroid link**
  (Unweighted pair-group centroid, UPGMC):
  - Distance between averages of two clusters
  - Cluster inversion can occur

$$D(c_1, c_2) = D\left(\left(\frac{1}{|c_1|}\sum_{x \in c_1} \vec{x}\right), \left(\frac{1}{|c_2|}\sum_{x \in c_2} \vec{x}\right)\right)$$

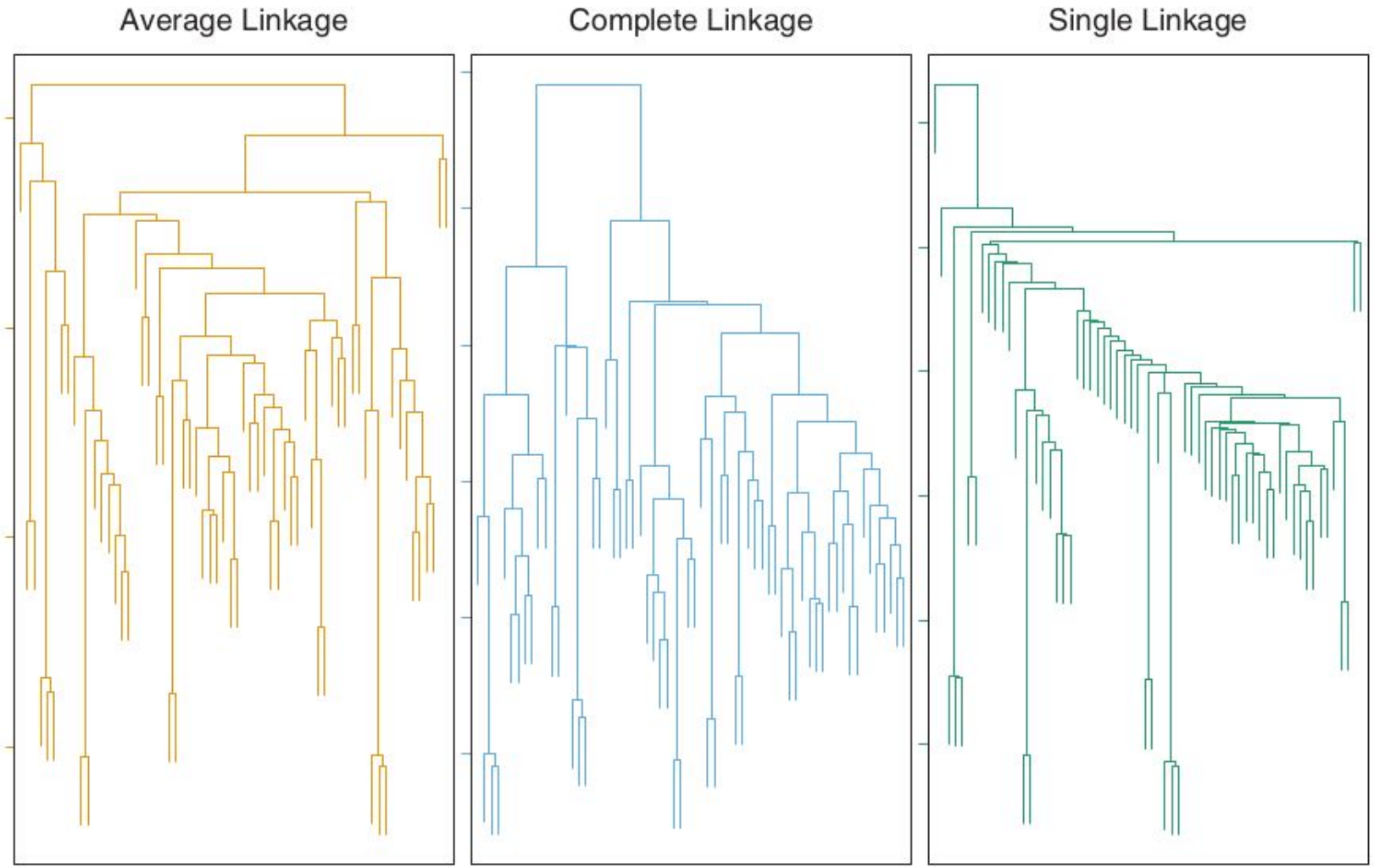


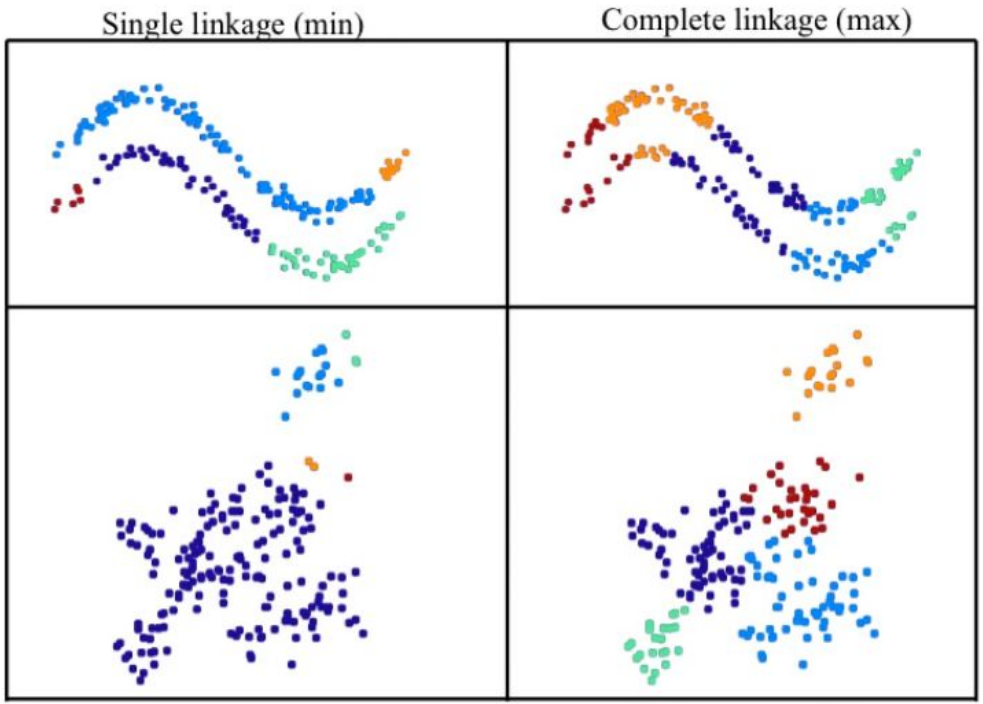
Centroid linkage

- Inversion -- similarity increases during merging process



- First merge similarity measure is -4
- Second merge similarity measure is -3.5

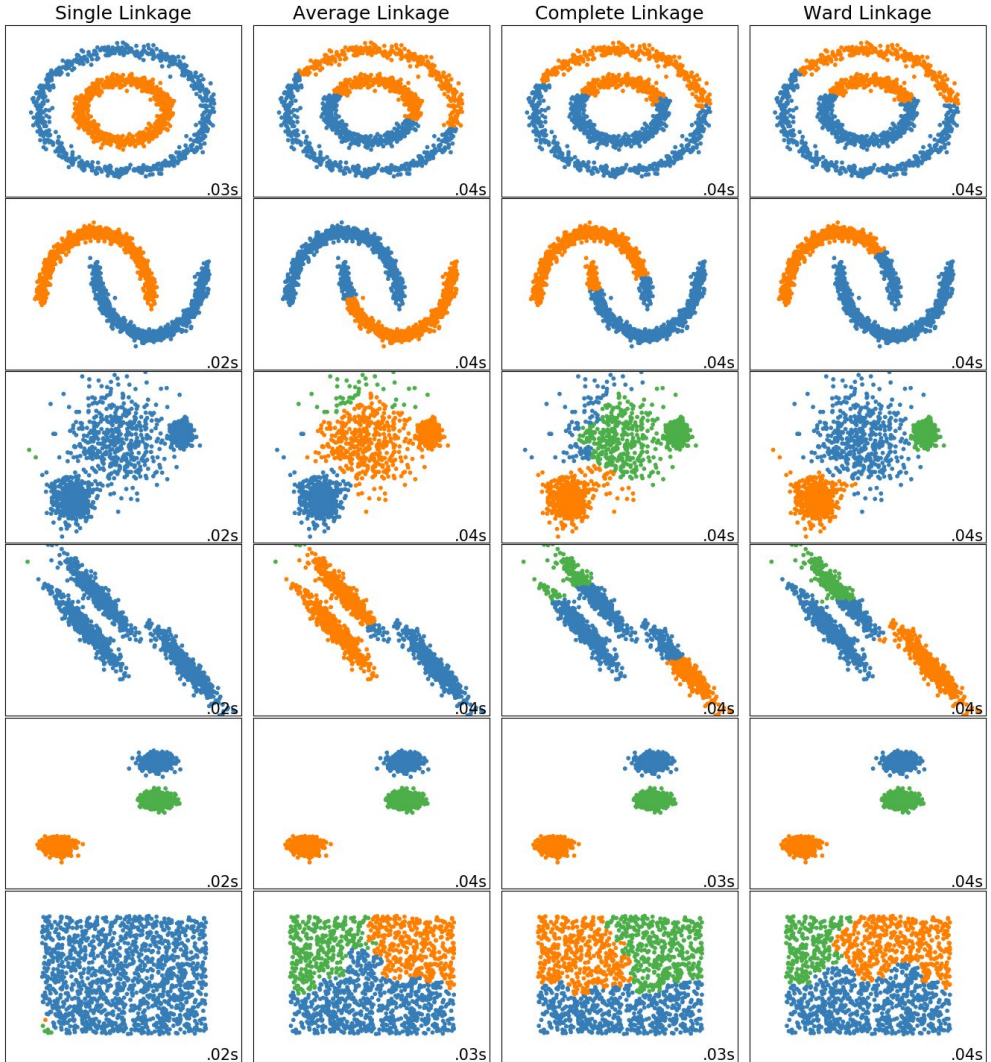

Average Linkage
Complete Linkage
Single Linkage



https://scikit-learn.org/stable/auto_examples/cluster/plot_linkage_comparison.html
https://www.youtube.com/watch?v=OcoE7JlbXvY

Other linkage methods

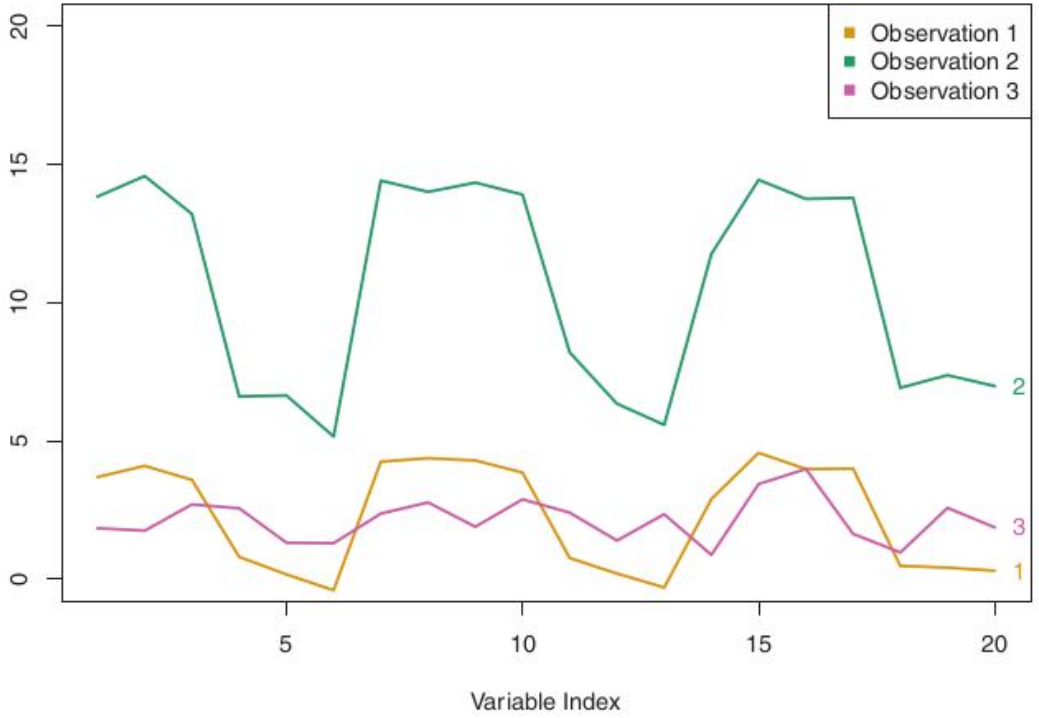
- Ward's method
- Weighted pair-group average (WPGMA)
- Weighted pair-group centroid (median)
- ...



https://scikit-learn.org/stable/auto_examples/cluster/plot_linkage_comparison.html
https://www.youtube.com/watch?v=OcoE7JlbXvY

Dissimilarity (or distance) measurements

- Euclidean,
- Correlation based,
- Manhattan,
- Canberra,
- Binary,
- Minkowski,
- Maximum,
- Chebychev distance,
- Power distance,
- Hellinger distance,
- Mahalanobis distance,
- *ect.*



Small Decisions with Big Consequences

The choice of dissimilarity measurement and linkage type is important

How to find good combinations?

- Try the most common ones (e.g. euclidean distance, average link)
- Identify problems and what causes them, e.g.,

  Have categorical data -> cannot use euclidean distance -> change it

  A lot of features p (high dimension data) -> euclidean distance performs bad

  Clusters possibly have different dispersion -> maybe Ward linkage is better

  Clusters are not globular -> maybe try single linkage
- Field specific practices, methods, solutions…

  Scientific question

  Data

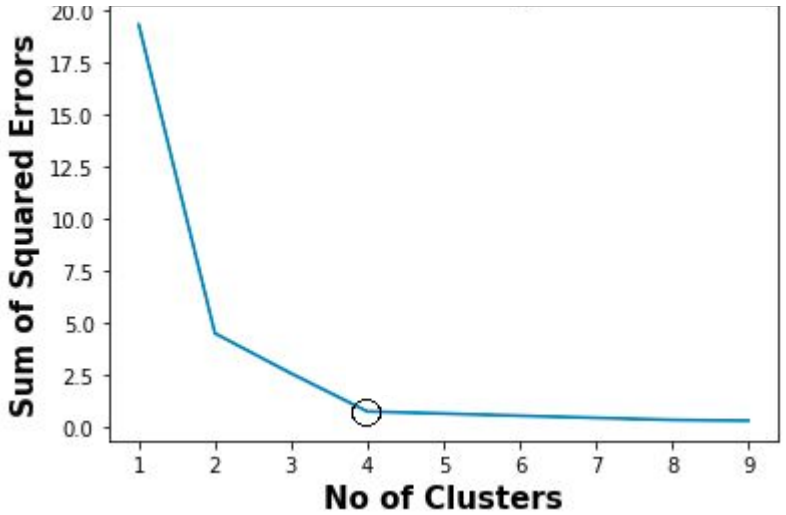
Some more things to consideration

1. Should the variables be scaled to have sd = 1, mean = 0 (or range 0-1)?
   * measured on different scales -- yes
   * different frequency of features e.g. # of socks and computers that one person buys in a year -- likely yes, with sd = 1

If the variables are scaled to have standard deviation one before the inter-observation dissimilarities are computed, then each variable will in effect be given equal importance in the hierarchical clustering performed

Some more things to consideration

2. Where should we cut the dendrogram in order to obtain clusters?
   - Prior knowledge
   - By eye
   - “Elbow method”



Hierarchical clustering

e.g. group of people with a 50-50 split of males and females, evenly split among Americans, Japanese, and French

- Assumption of hierarchical structure might be unrealistic. True clusters might not be nested
- Not very robust to perturbations to the data (remove random set of observations n)
- Clusters can be heavily distorted if outliers are present
- Bad with clusters that are not in globular shape
- Not applicable to categorical data
- Curse of dimensionality: inflation of Euclidean distance
- Do not need to choose K beforehand
- Is more computationally intensive then K-means

K-means clustering – application issues

- A lot of choices -- each has an impact on the results.

  performing clustering with different choices of these parameters, and looking at the full set of results in order to see what patterns consistently emerge

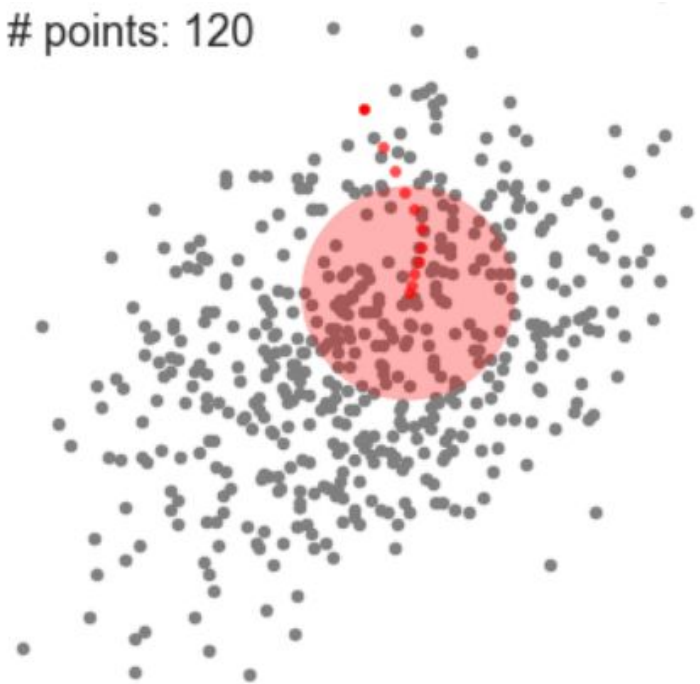
- clustering can be non-robust

  cluster subsets of the data in order to get a sense of the robustness of the clusters obtained

Mean-Shift Clustering

sliding-window-based algorithm that attempts to find dense areas of data points

- Circular sliding window
- Window is centered at a point C (randomly selected)
- Shifting the window iteratively to a higher density region on each step until convergence.



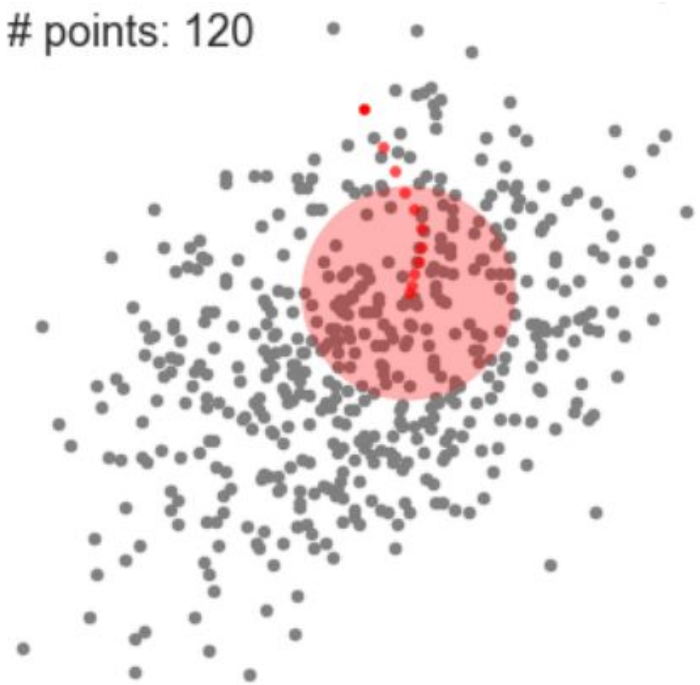
https://www.kdnuggets.com/2018/06/5-clustering-algorithms-data-scientists-need-know.html

Mean-Shift Clustering

sliding-window-based algorithm that attempts to find dense areas of data points

- Circular sliding window
- Window is centered at a point C (randomly selected)
- Shifting the window iteratively to a higher density region on each step until convergence.

HOW to find the direction to shift the window?

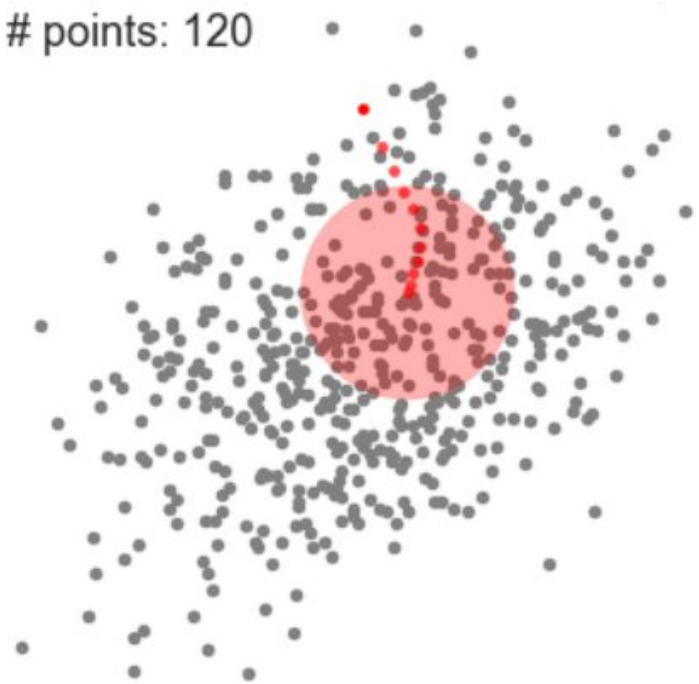


https://www.kdnuggets.com/2018/06/5-clustering-algorithms-data-scientists-need-know.html

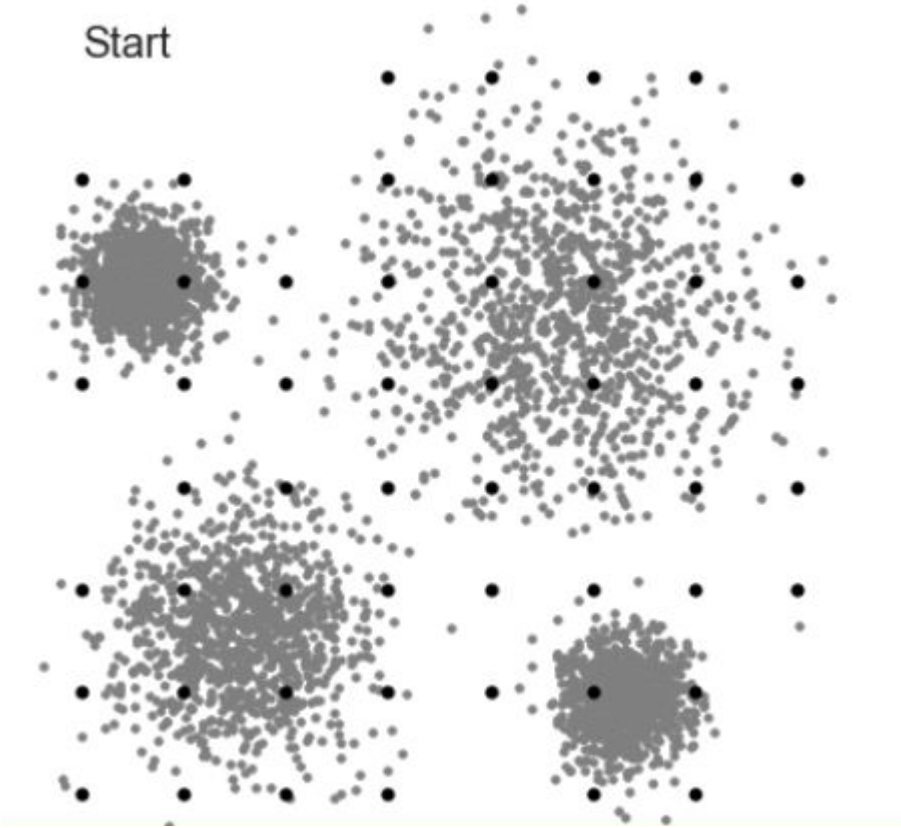
Mean-Shift Clustering

HOW to find the direction to shift the window?

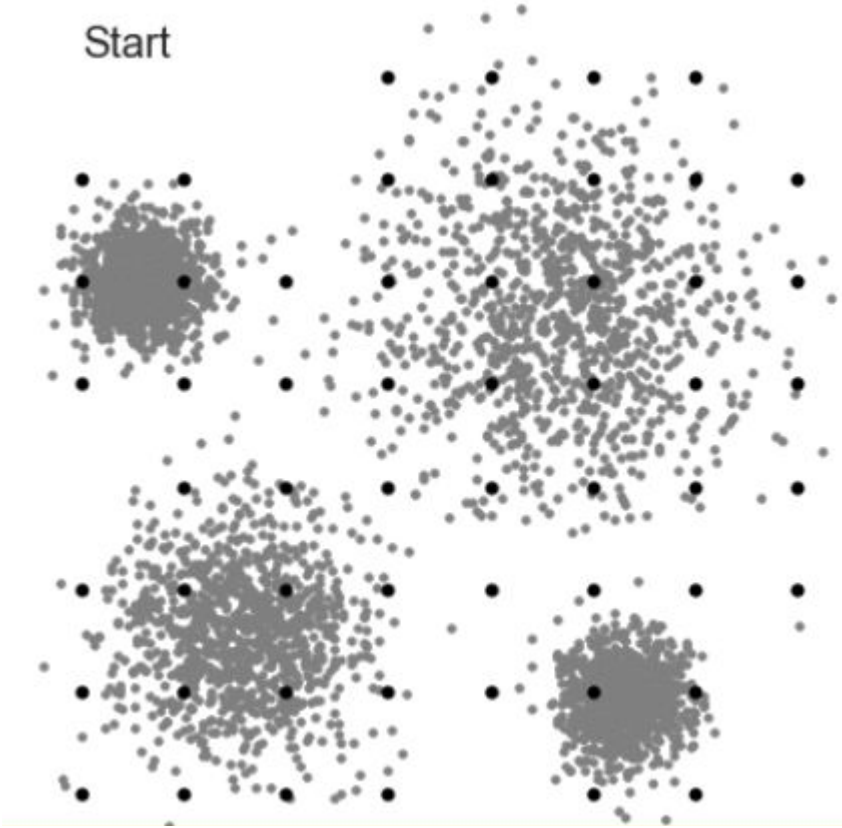
- Density within the sliding window is proportional to the number of points inside it
- If some side of window has more points the mean of all the points in the window will shift to that side

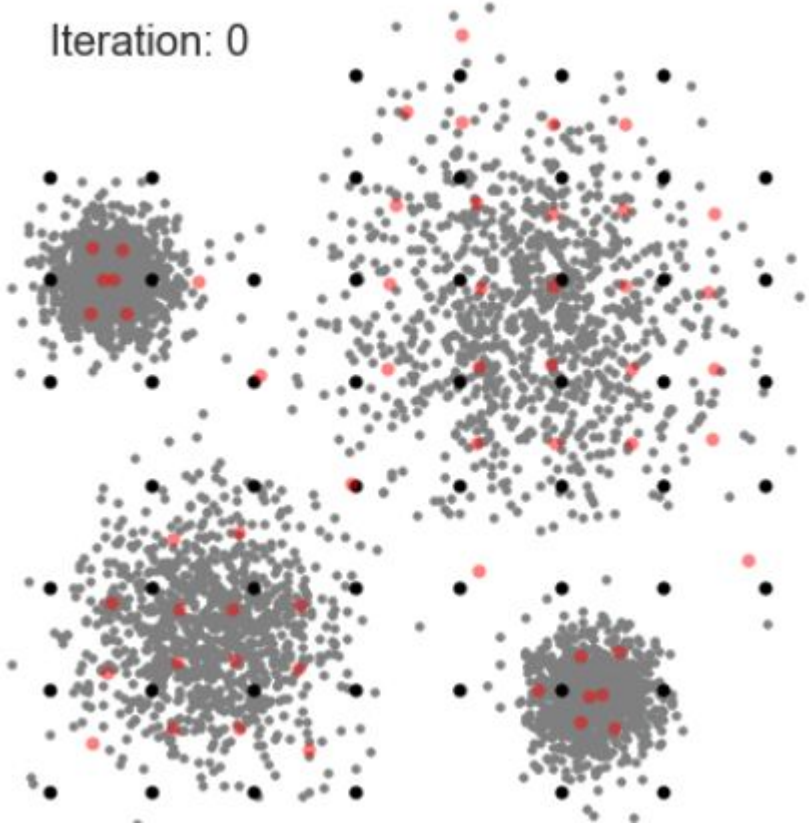


https://www.kdnuggets.com/2018/06/5-clustering-algorithms-data-scientists-need-know.html

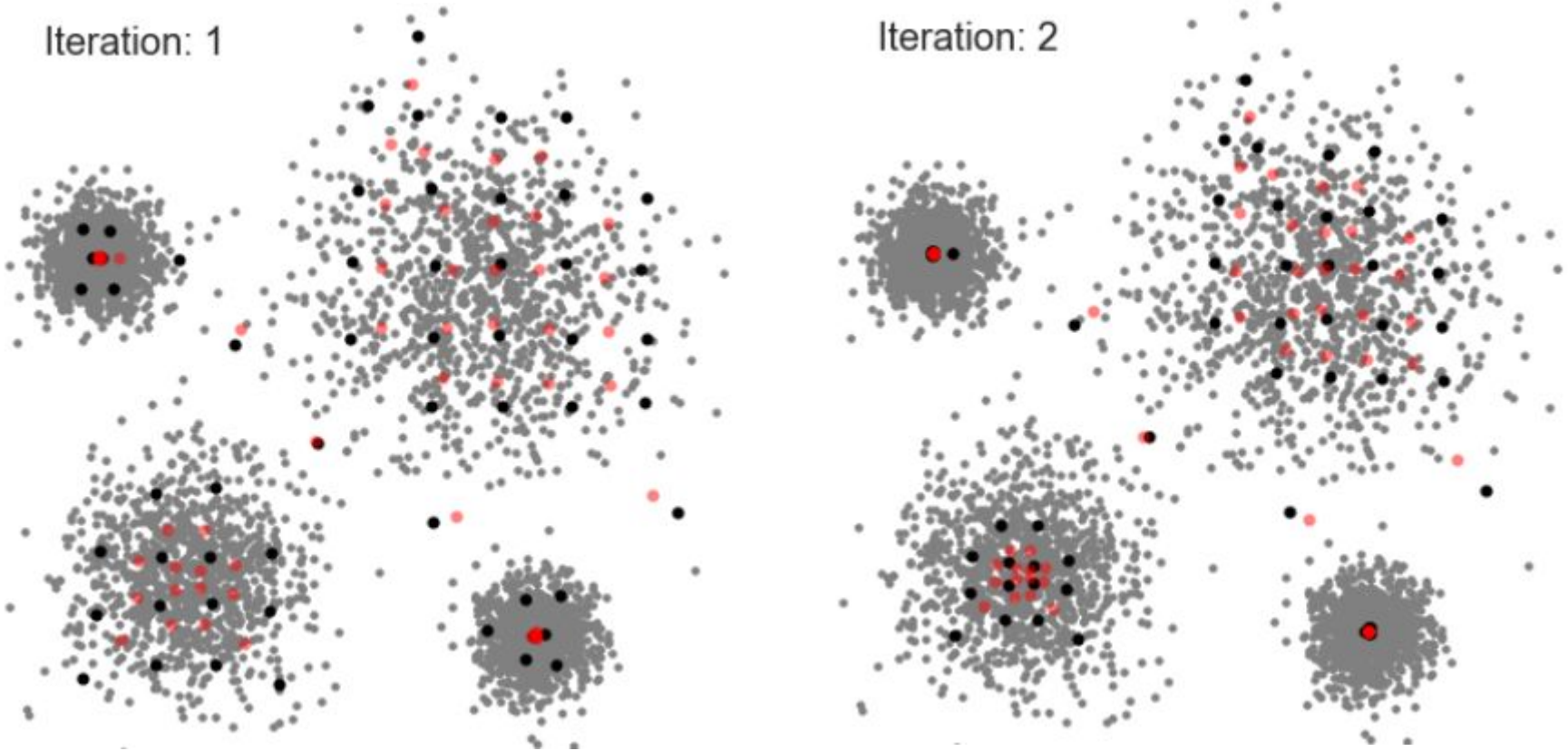


https://www.kdnuggets.com/2018/06/5-clustering-algorithms-data-scientists-need-know.html

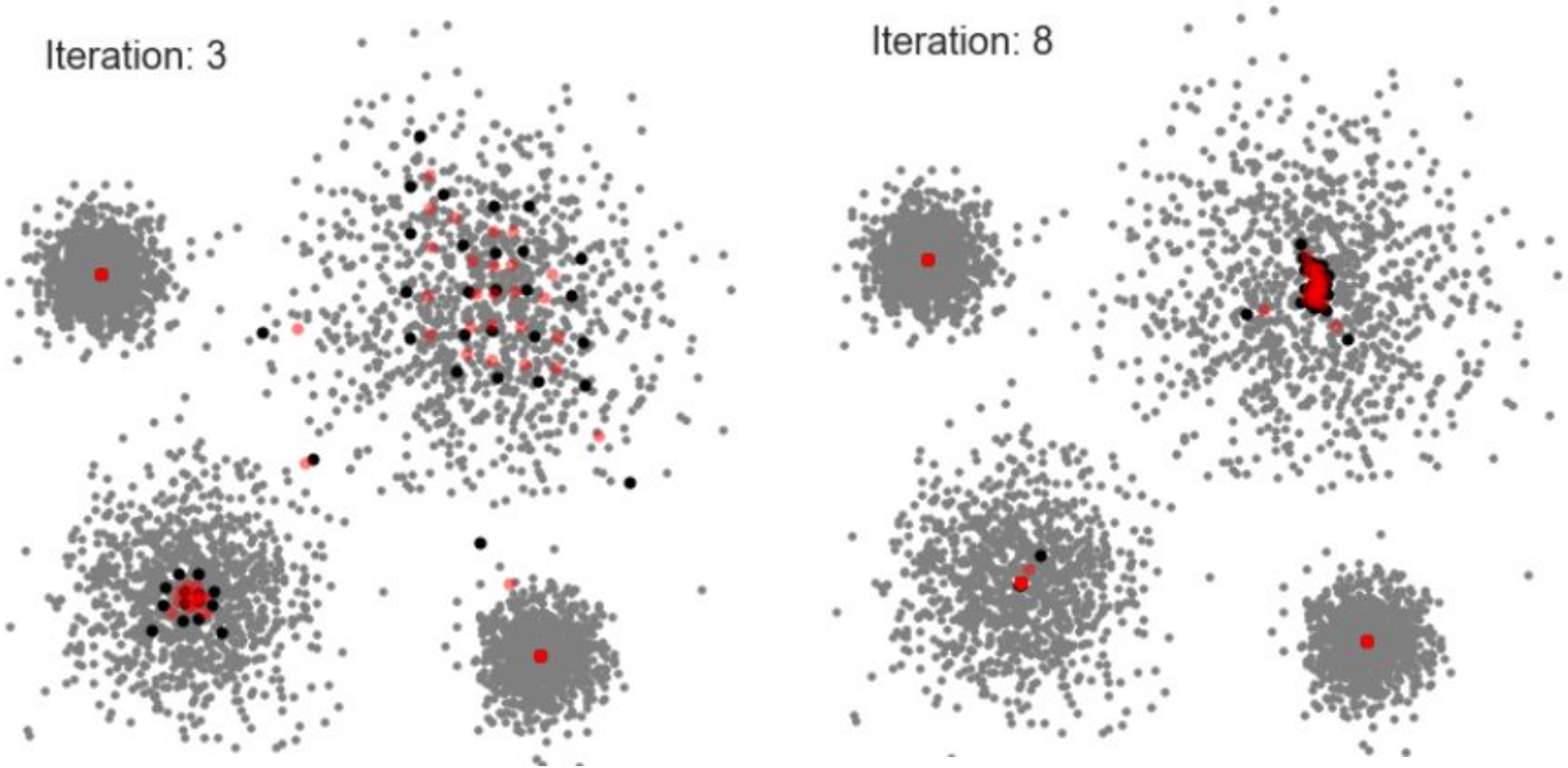




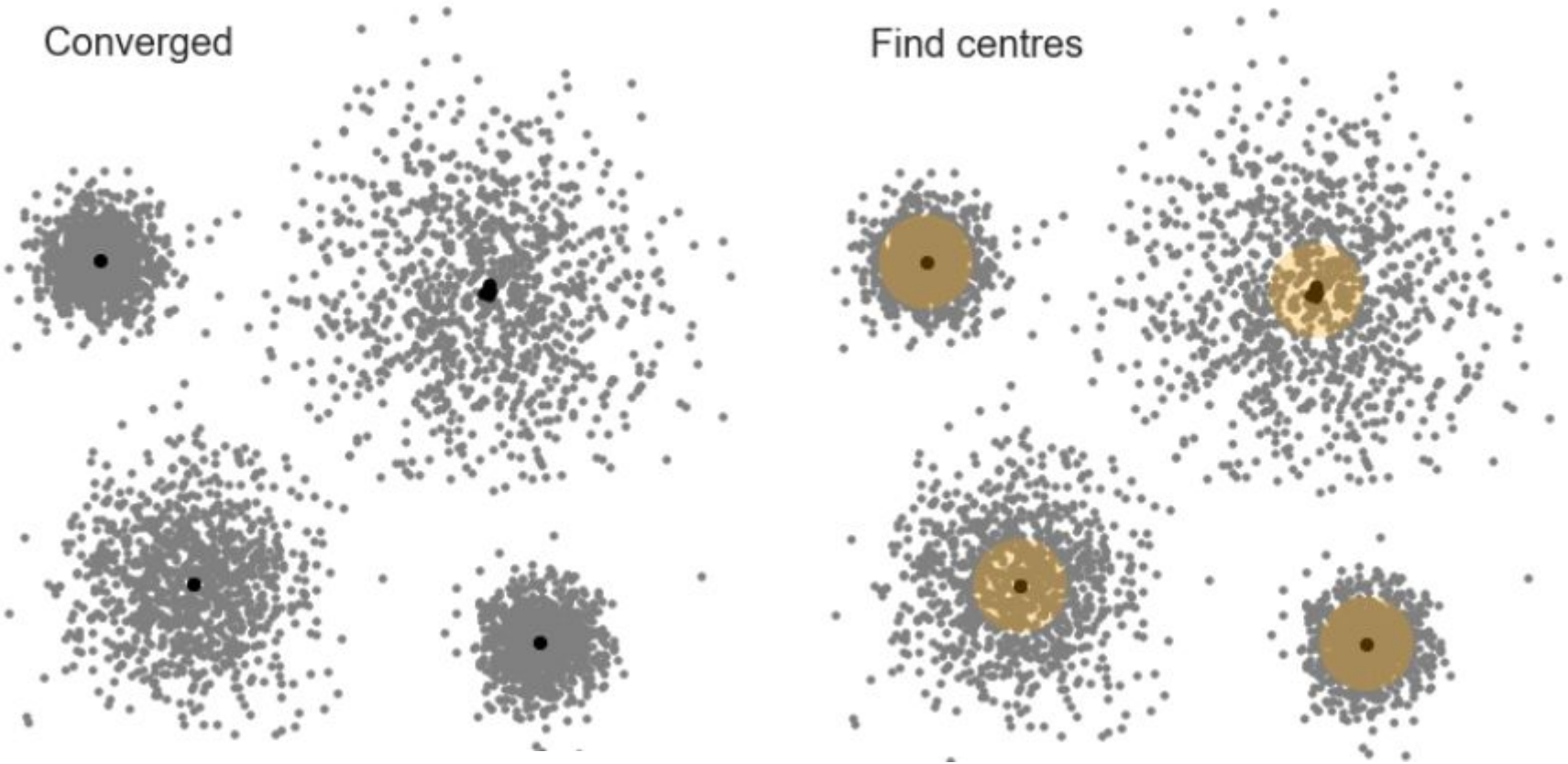
https://www.kdnuggets.com/2018/06/5-clustering-algorithms-data-scientists-need-know.html



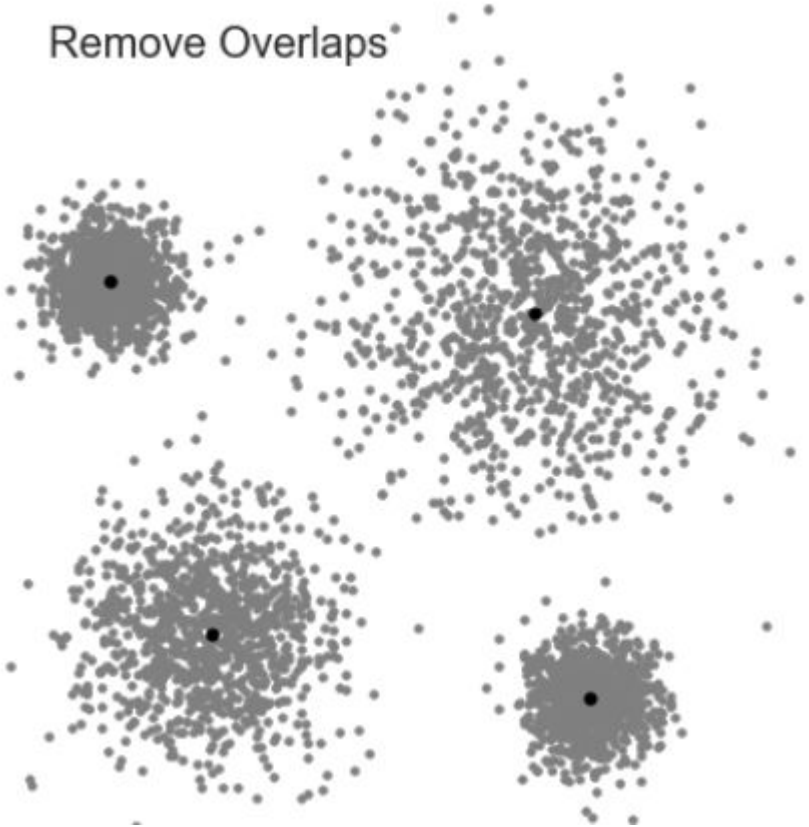
https://www.kdnuggets.com/2018/06/5-clustering-algorithms-data-scientists-need-know.html



https://www.kdnuggets.com/2018/06/5-clustering-algorithms-data-scientists-need-know.html



https://www.kdnuggets.com/2018/06/5-clustering-algorithms-data-scientists-need-know.html



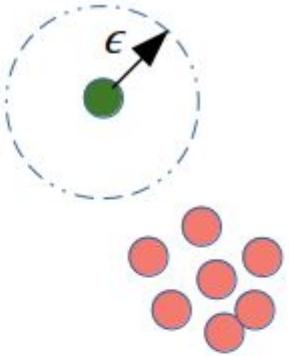
https://www.kdnuggets.com/2018/06/5-clustering-algorithms-data-scientists-need-know.html

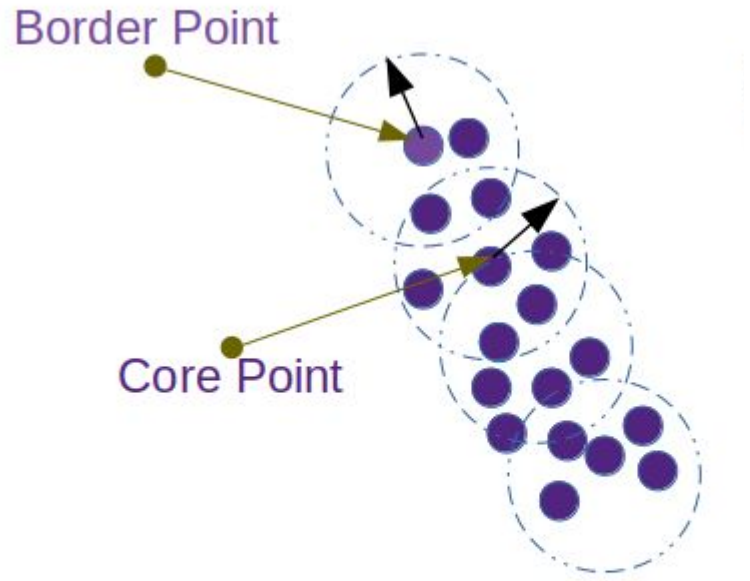
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

density based clustering algorithm similar to mean-shift

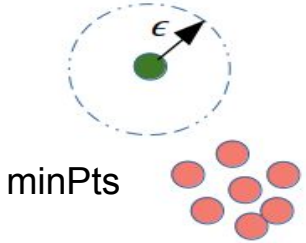
Two parameters:

- ϵ -- Radius of area to search
- minPts -- minPoints

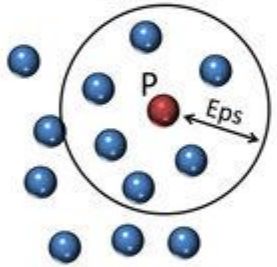




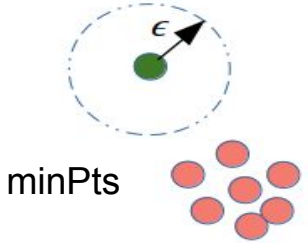
DBSCAN



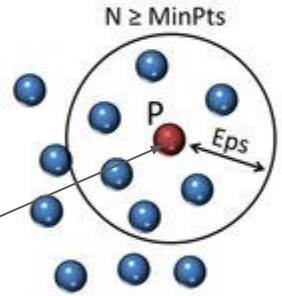
1. Select arbitrary starting data point not visited before. Extract neighborhood of this point using distance epsilon ε (All points which are within the ε distance are neighborhood points)



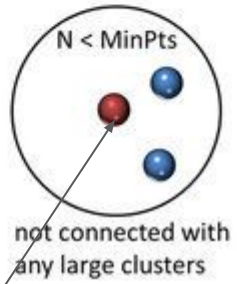
DBSCAN



1. Select arbitrary starting data point not visited before.
   Extract neighborhood of this point using distance epsilon ε
   (All points which are within the ε distance are neighborhood points)
2. If # neighborhood points is sufficient (according to minPoints), point becomes first point in new cluster.
   Otherwise point is labeled as noise
   (might become part of cluster later).
   Point is marked as “visited”

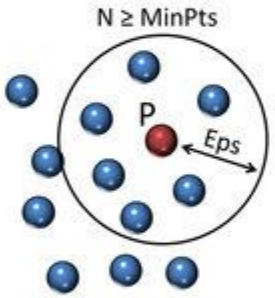


Cluster A point

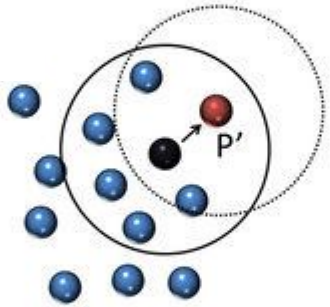


Noise point

DBSCAN

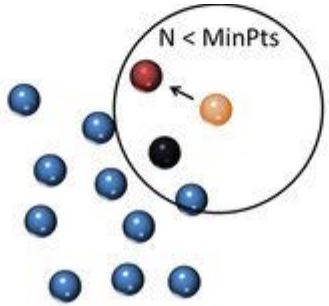


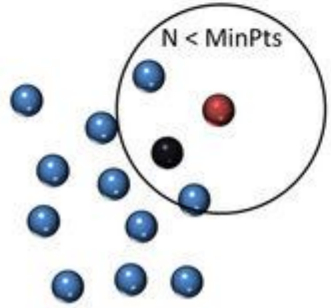
2. If # neighborhood points is sufficient (according to minPoints), point becomes first point in new cluster.
   Otherwise point is labeled as noise
   (might become part of cluster later).
   Point is marked as “visited”



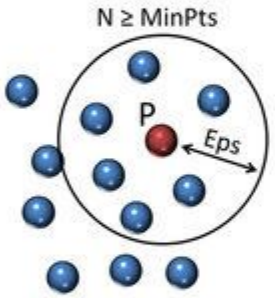
3. All the neighborhood points become part of the same cluster.

4. Select new point from cluster and do step 1-3 until all points in cluster have been visited and labelled.

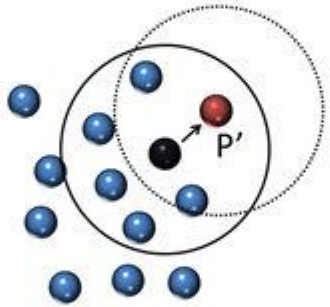




DBSCAN

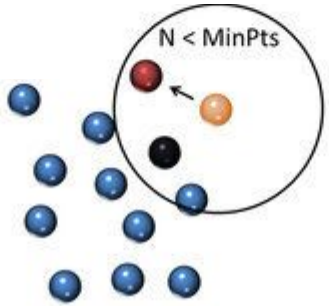


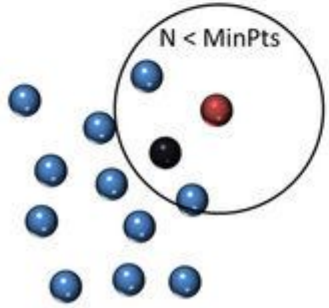
2. If # neighborhood points is sufficient (according to minPoints), point becomes first point in new cluster.
   Otherwise point is labeled as noise
   (might become part of cluster later).
   Point is marked as “visited”



3. All the neighborhood points become part of the same cluster.

4. Select new point from cluster and do step 1-3 until all points in cluster have been visited and labelled.

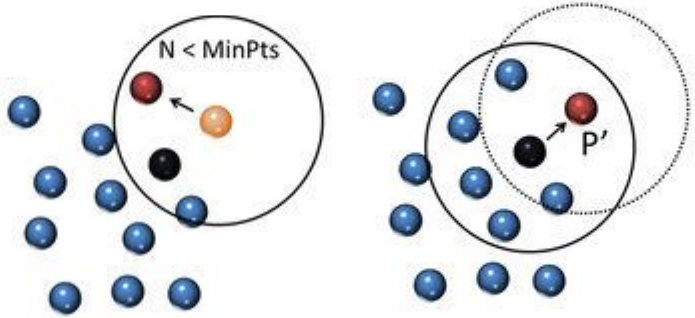


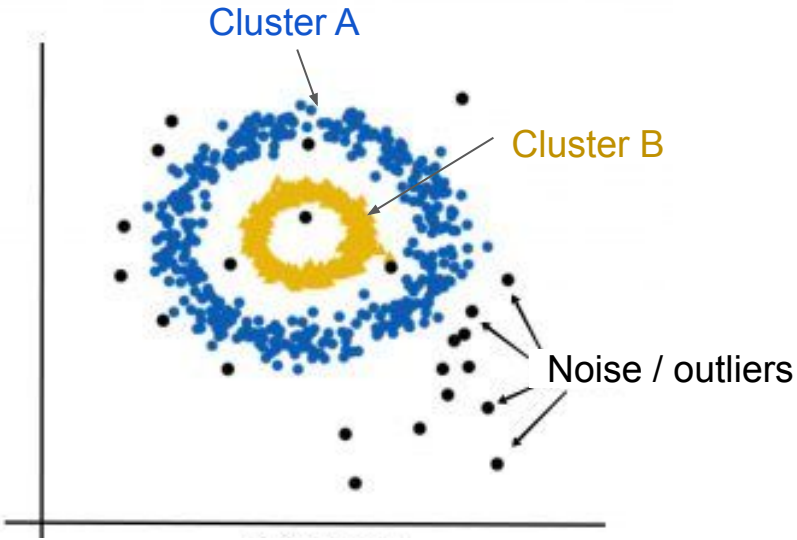


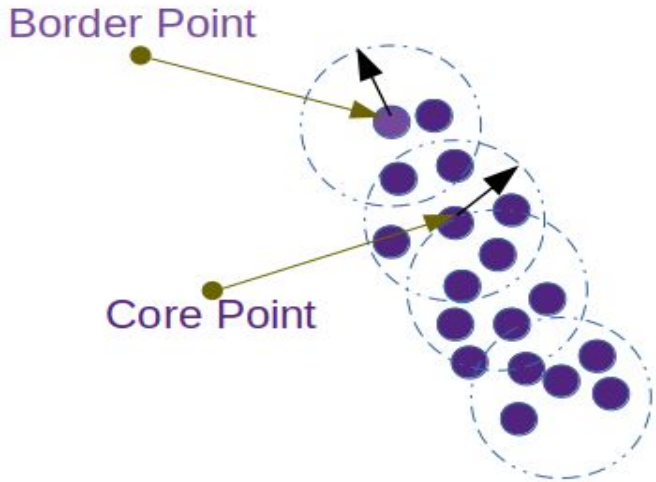
5. Once done with this cluster,
   select new unvisited point and start again
   until all points are visited.

DBSCAN

each point will be marked as “clusterX” or “Noise”

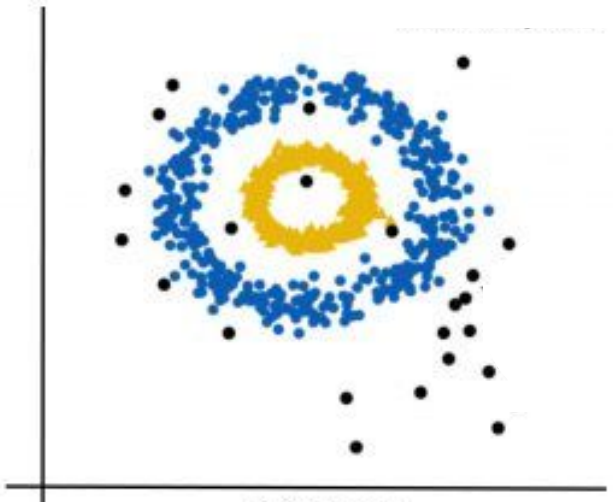


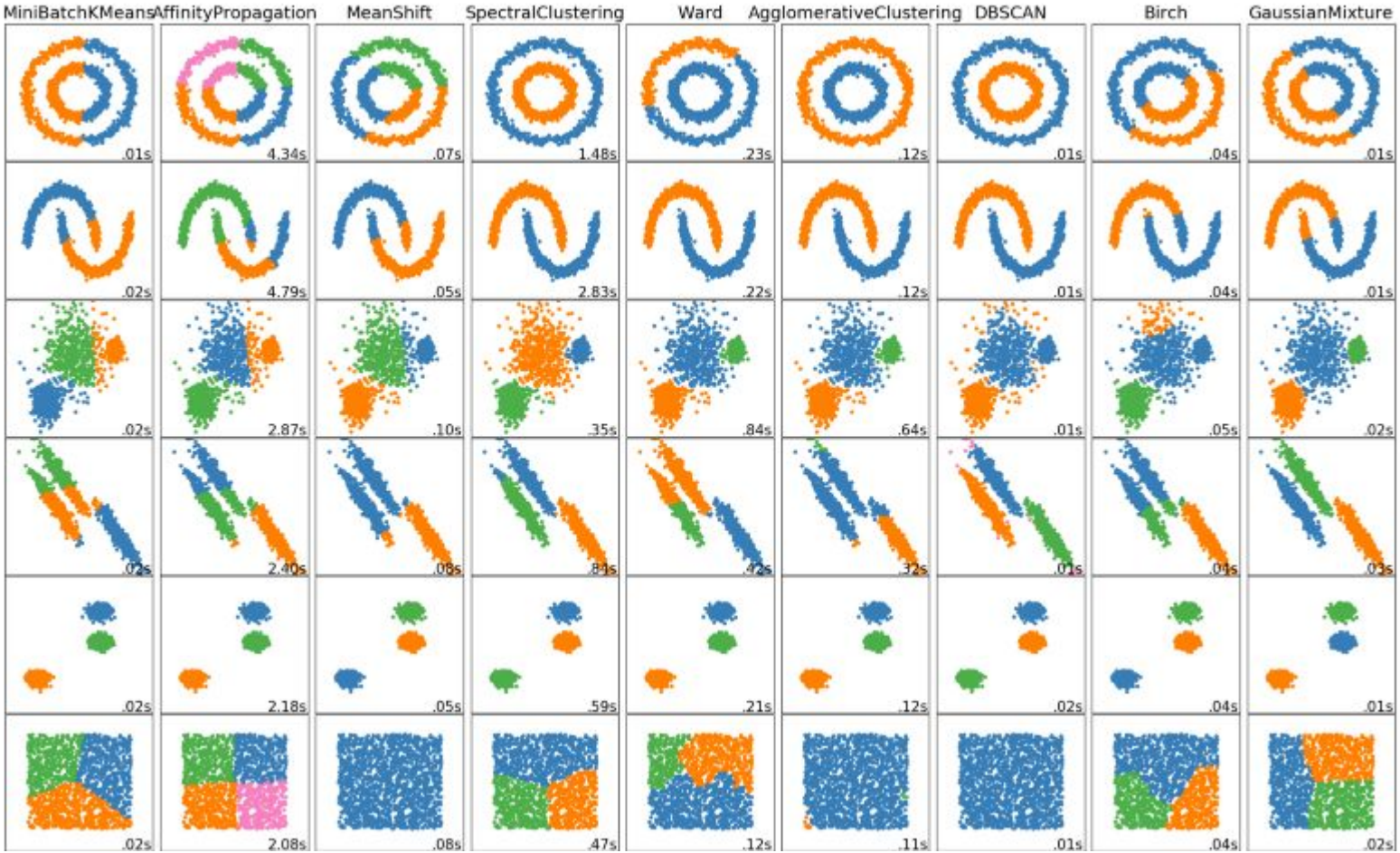




DBSCAN

- Does not require a pe-set number of clusters
- Identifies points as outliers/noise
- Finds arbitrarily sized and shaped clusters quite well
- Does not perform well when the clusters are of varying density
- Does not perform well very high-dimensional





https://www.kdnuggets.com/2018/06/5-clustering-algorithms-data-scientists-need-know.html