Unsupervised Learning

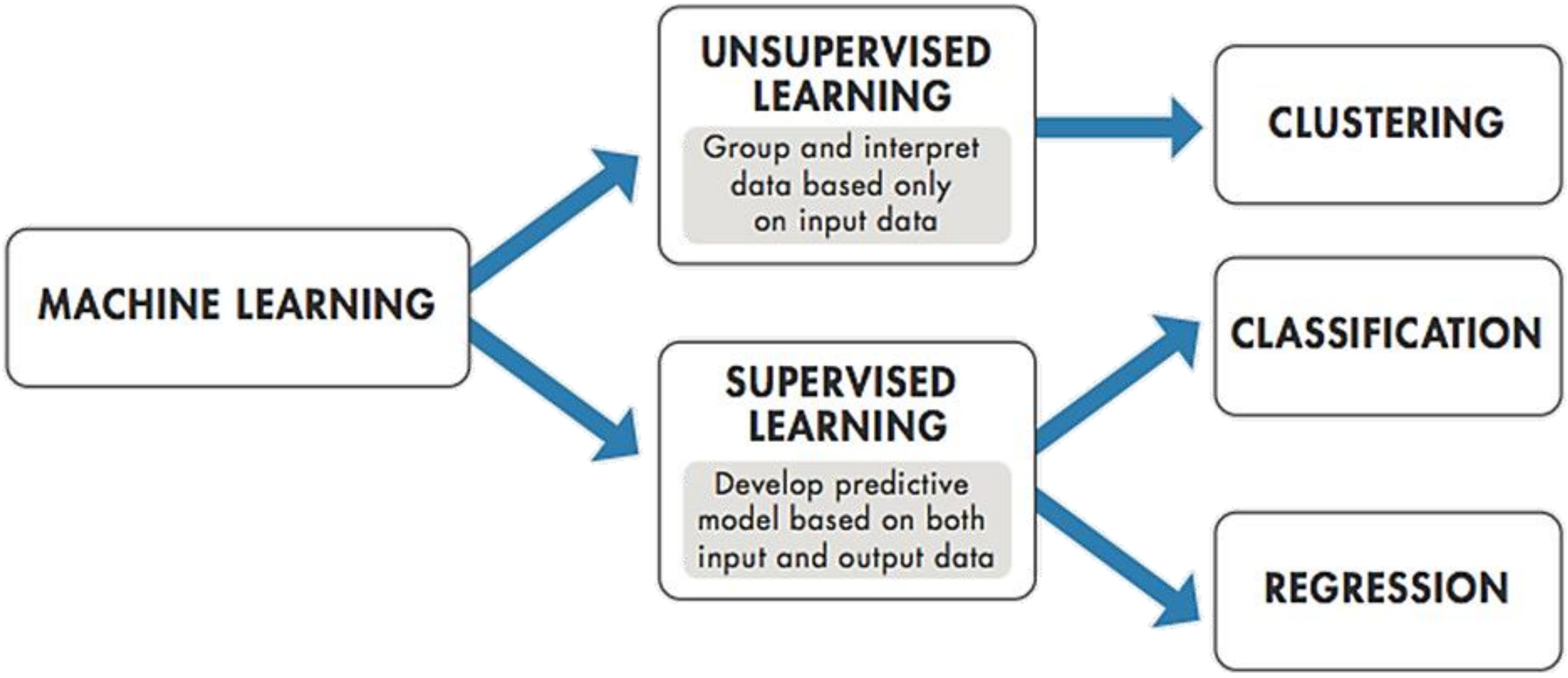
KASHTANOVA VICTORIYA

INRIA, EPIONE

UNIVERSITÉ CÔTE D'AZUR

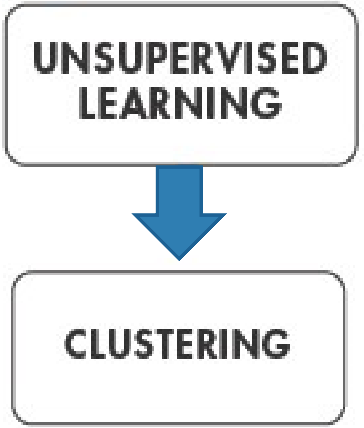
Inria INVENTEURS DU MONDE NUMÉRIQUE

Machine learning methods

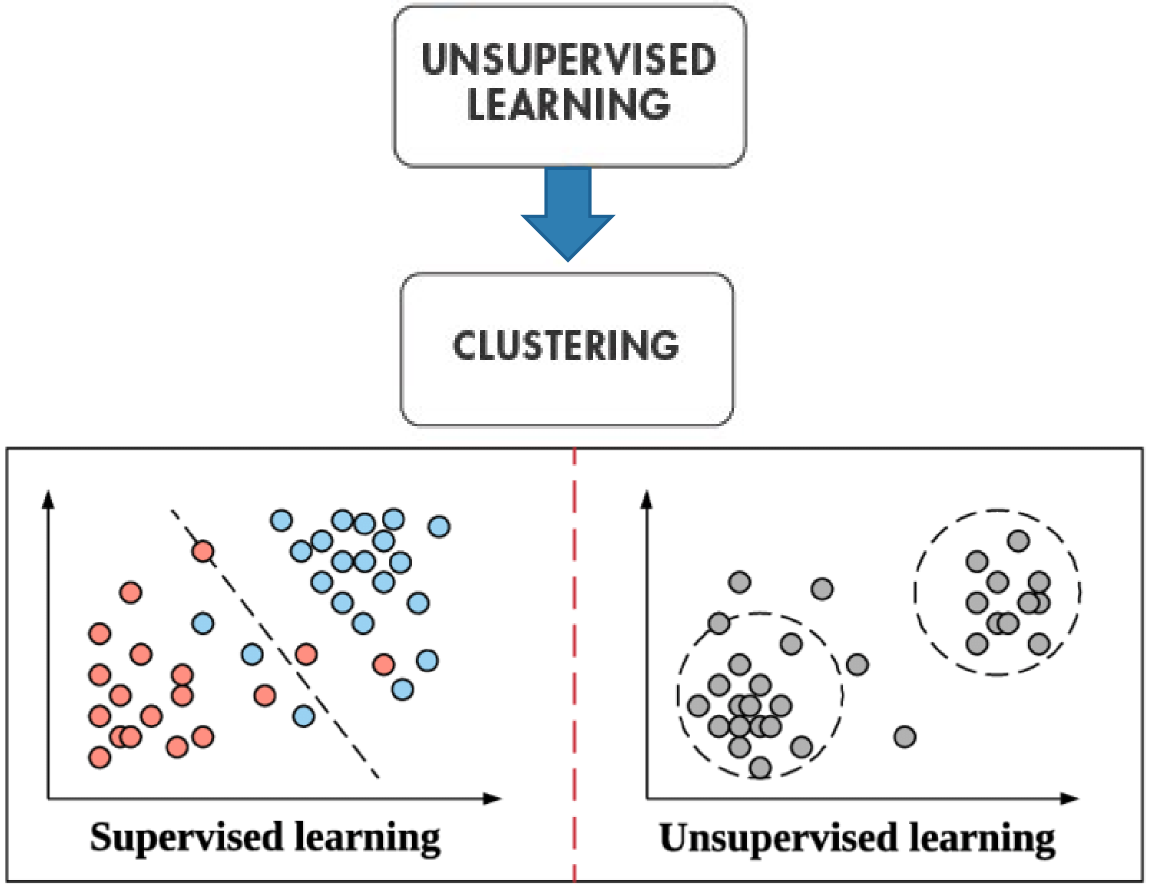


Unsupervised methods

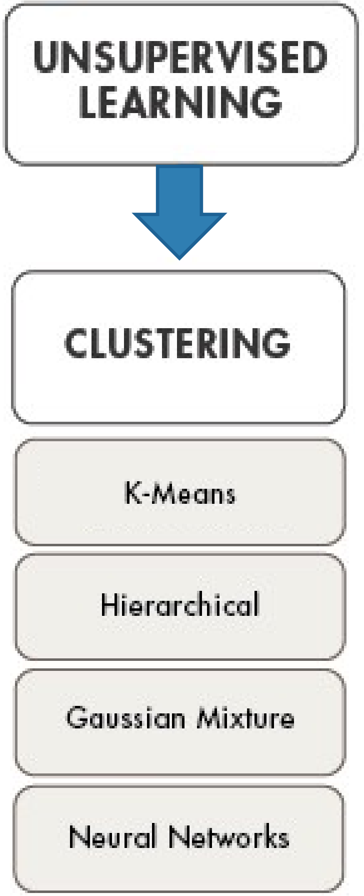
Unsupervised methods



Unsupervised methods



Unsupervised methods

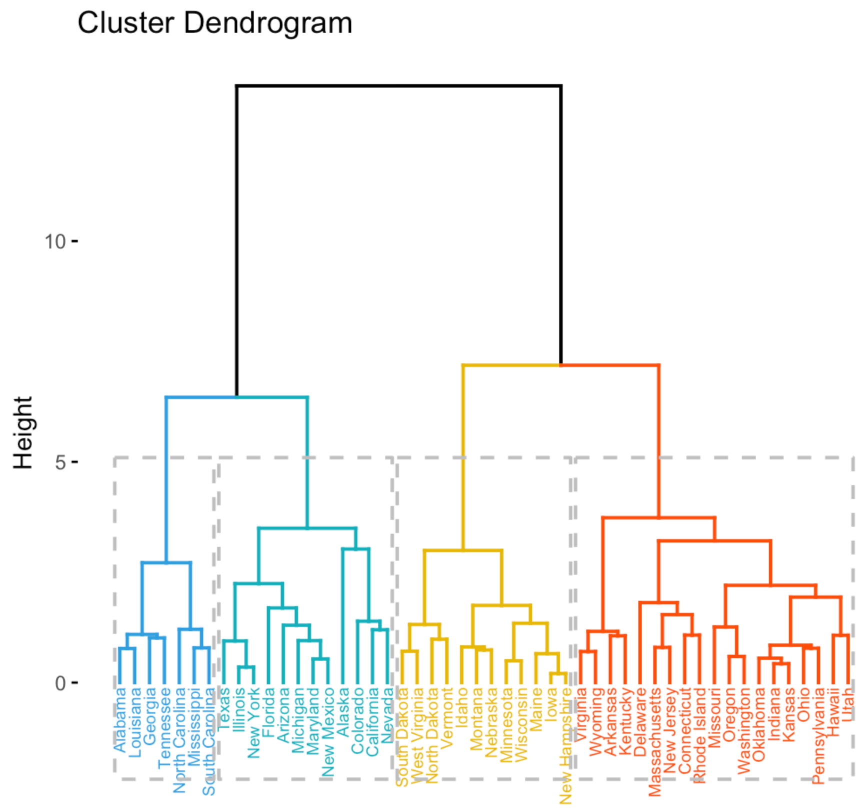


Hierarchical (Agglomerative) clustering

1. Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item.
2. Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one fewer cluster.
3. Continue the process until all items are clustered into a single cluster of size N.

<u>**Linkage Metrics :**</u>

- **Single-linkage** (by closest item)
- **Complete-linkage** (all items closer then in another cluster)
- **Average-linkage** (by closest average)



Hierarchical clustering : Example

	BOS	**NY**	**CHI**	**DEN**	**SF**	**SEA**
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

Hierarchical clustering : Example

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

{Boston} {New-York} {Chicago} {Denver} {San Francisco} {Seattle}

Hierarchical clustering : Example

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

{Boston} {New-York} {Chicago} {Denver} {San Francisco} {Seattle}

{Boston, New-York} {Chicago} {Denver} {San Francisco} {Seattle}

Hierarchical clustering : Example

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

{Boston} {New-York} {Chicago} {Denver} {San Francisco} {Seattle}

{Boston, New-York} {Chicago} {Denver} {San Francisco} {Seattle}

{Boston, New-York, Chicago} {Denver} {San Francisco} {Seattle}

Hierarchical clustering : Example

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

{Boston} {New-York} {Chicago} {Denver} {San Francisco} {Seattle}

{Boston, New-York} {Chicago} {Denver} {San Francisco} {Seattle}

{Boston, New-York, Chicago} {Denver} {San Francisco} {Seattle}

{Boston, New-York, Chicago} {Denver} {San Francisco, Seattle}

Hierarchical clustering : Example

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

{Boston, New-York, Chicago} {Denver} {San Francisco, Seattle}

{Boston, New-York, Chicago, Denver} {San Francisco, Seattle} **OR** {Boston, New-York, Chicago} {Denver, San Francisco, Seattle}

(**Single linkage**) (**Complete linkage**)

Hierarchical clustering : Example

	BOS	NY	CHI	DEN	SF	SEA
BOS	0	206	963	1949	3095	2979
NY		0	802	1771	2934	2815
CHI			0	966	2142	2013
DEN				0	1235	1307
SF					0	808
SEA						0

{Boston, New-York, Chicago} {Denver} {San Francisco, Seattle}

{Boston, New-York, Chicago, Denver} {San Francisco, Seattle} **OR** {Boston, New-York, Chicago} {Denver, San Francisco, Seattle}

(**Single linkage**) (**Complete linkage**)

{Boston, New-York, Chicago, Denver,San Francisco, Seattle}

Hierarchical clustering : Advantages & Drawbacks

- **Deterministic**
- Flexible with respect to linkage criteria
- Can select number of clusters using dendogram

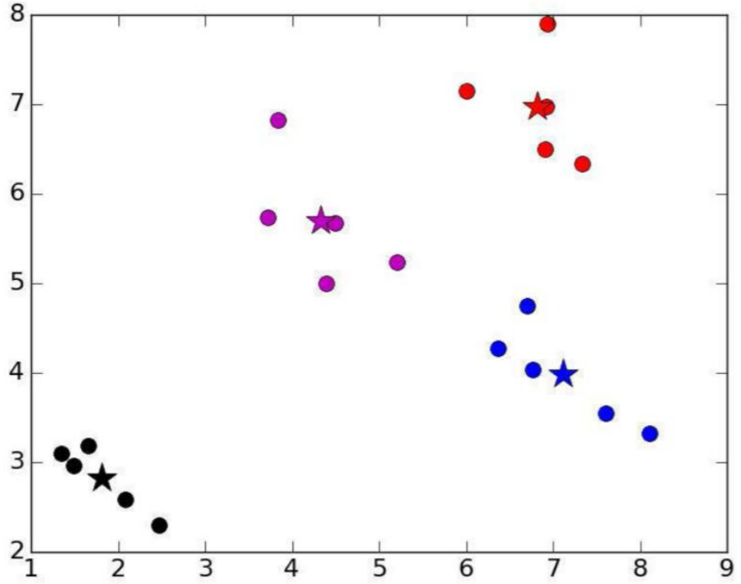
- **Slow** (Naïve algorithm has n^3 complexity)

K-means method

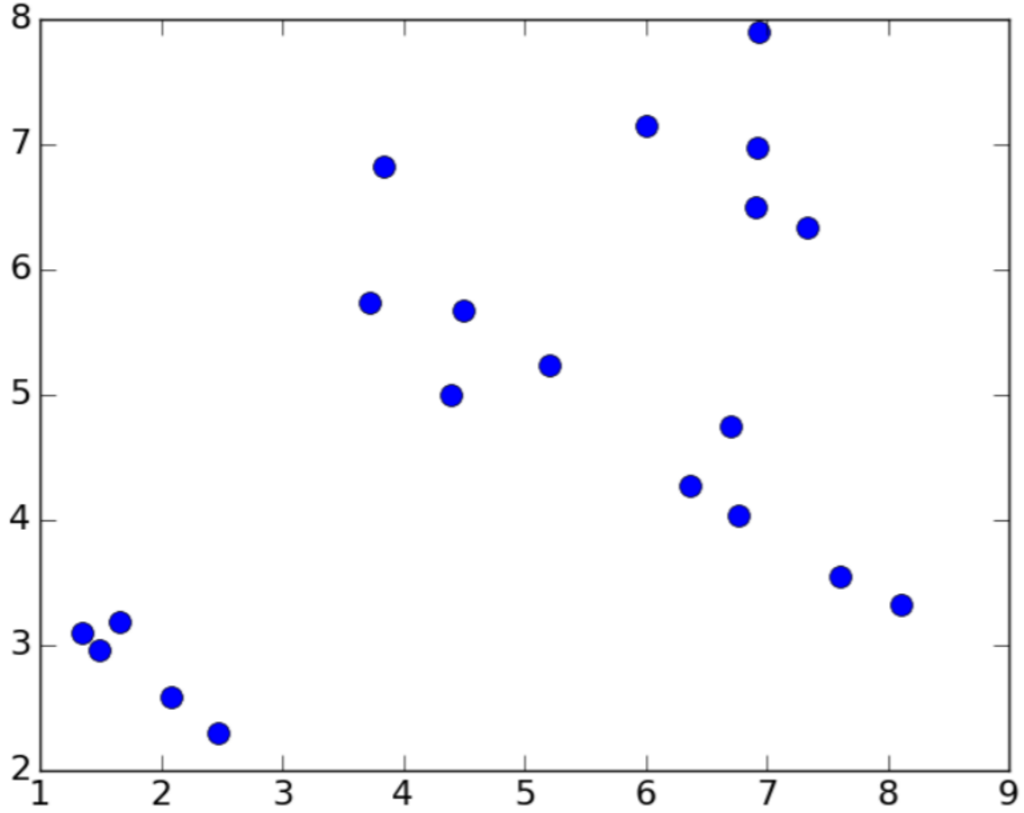
K-means algorithm is the most popular and yet simplest of all the clustering algorithms.

<u>Algorithm :</u>

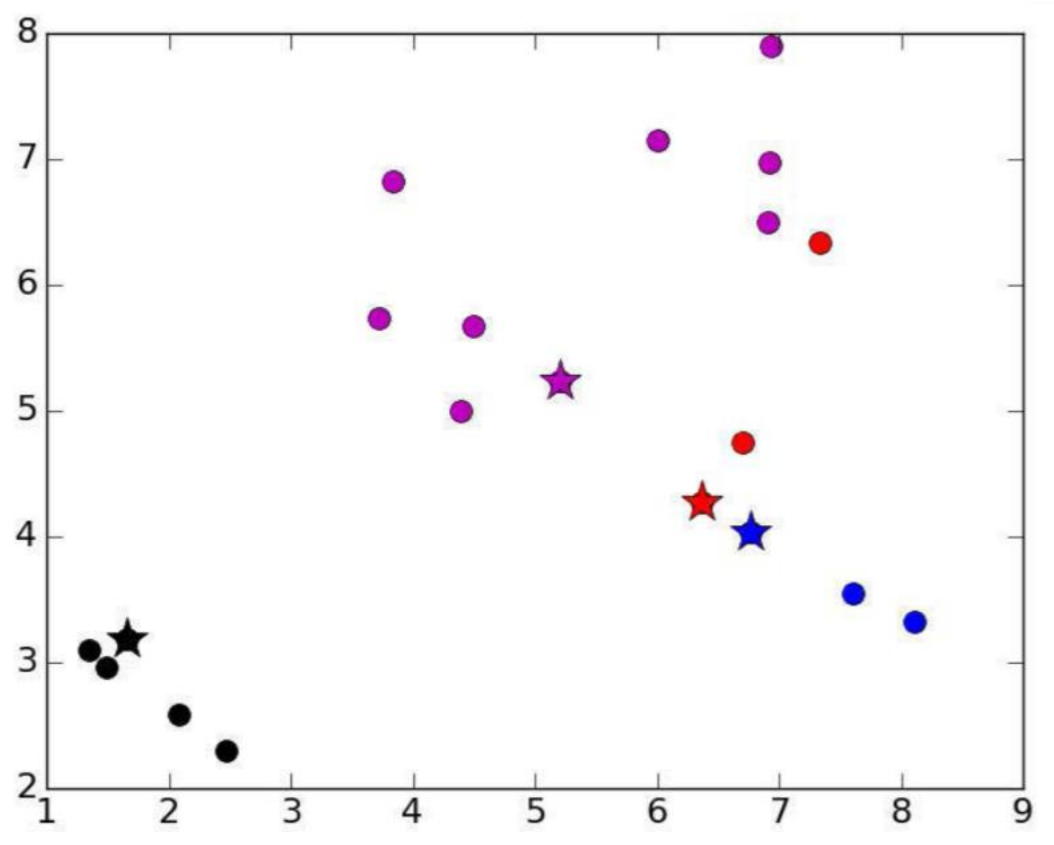
1. Select the number of clusters k that you think is the optimal number.
2. Initialize k points as "centroids" randomly within the space of our data.
3. Attribute each observation to its closest centroid.
4. Update the centroids to the center of all the attributed set of observations.
5. Repeat steps 3 and 4 a fixed number of times or until all of the centroids are stable (i.e. no longer change in step 4).



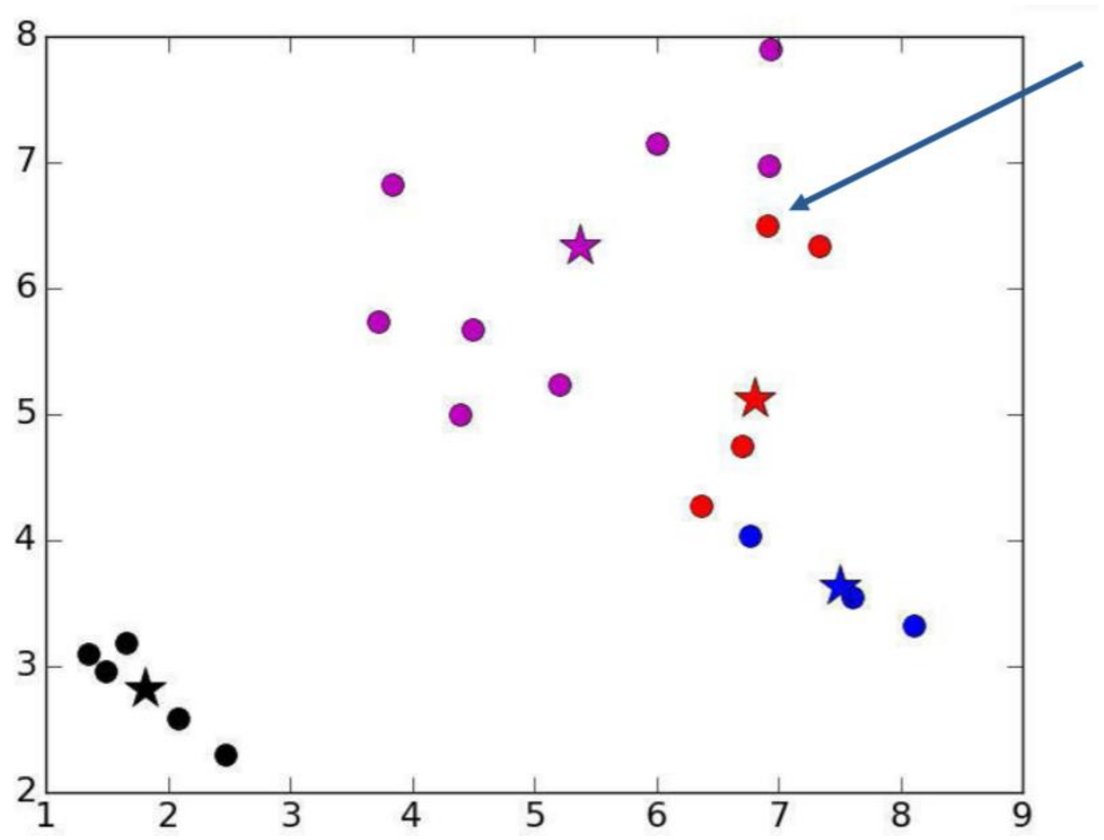
K-means method : Example



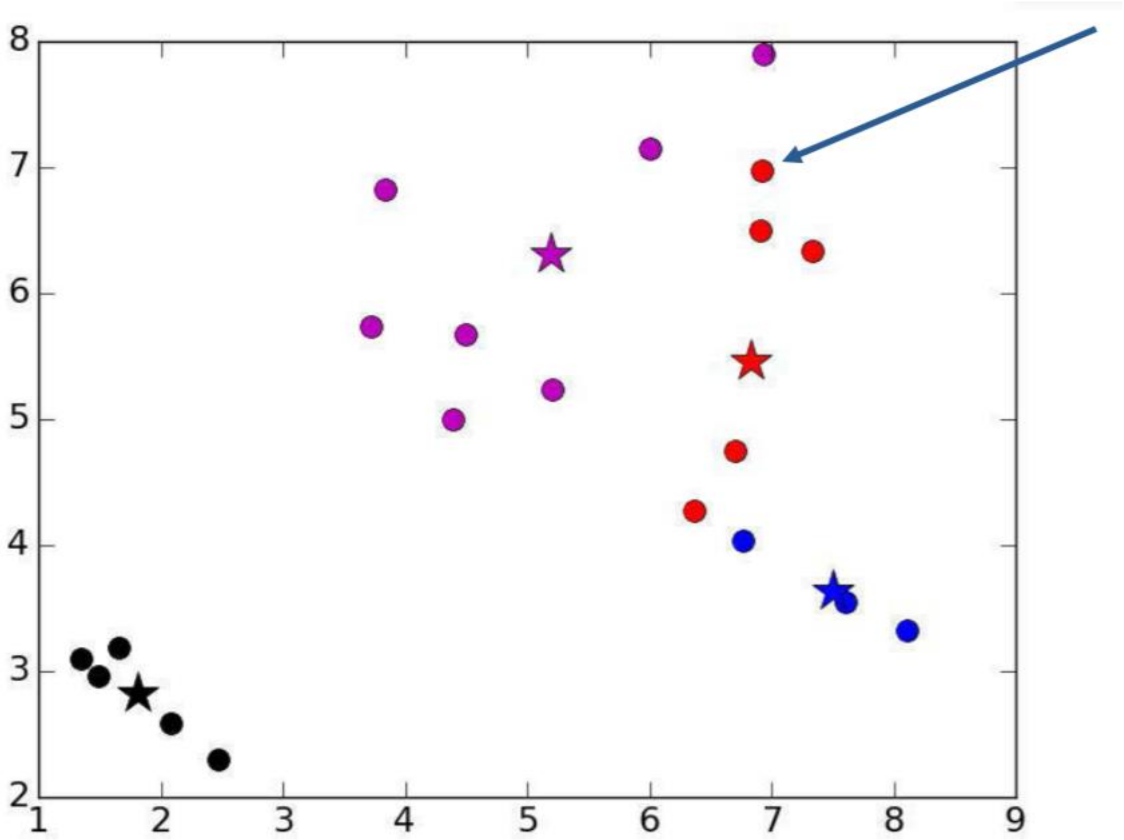
K-means method : Initial centroids



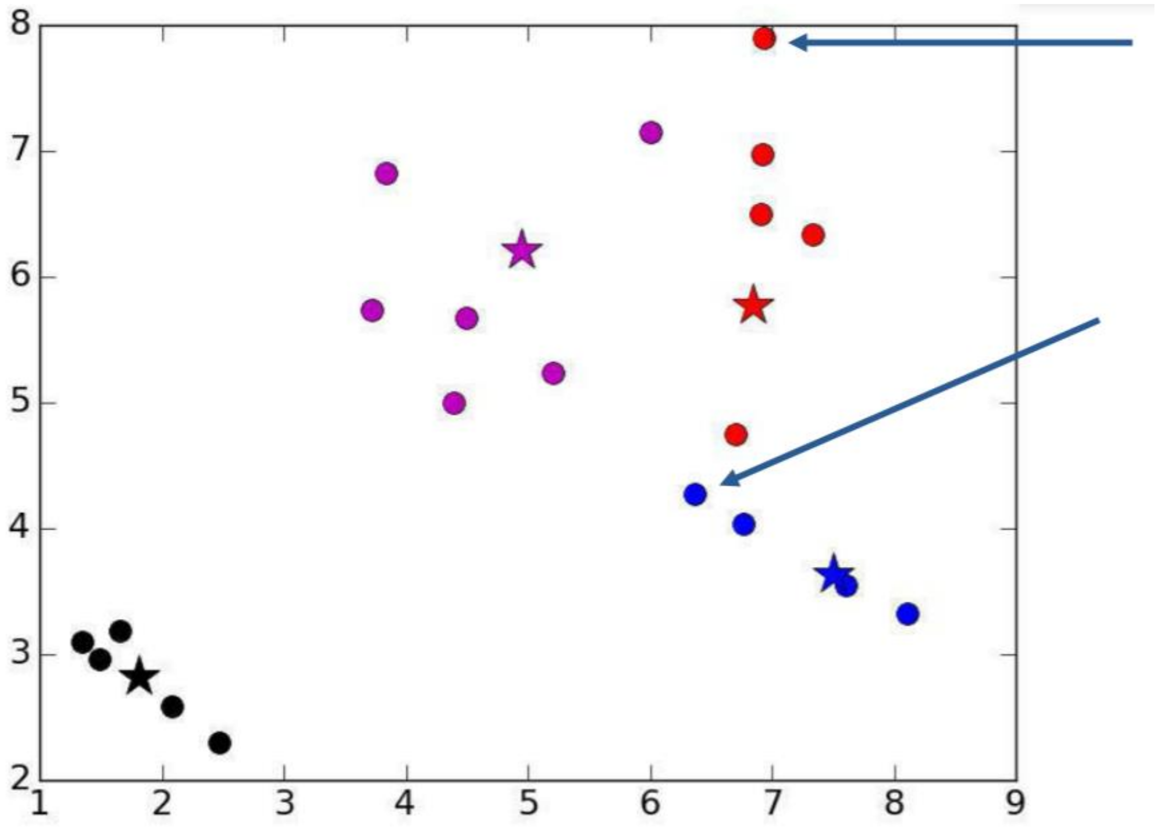
K-means method : 1st iteration



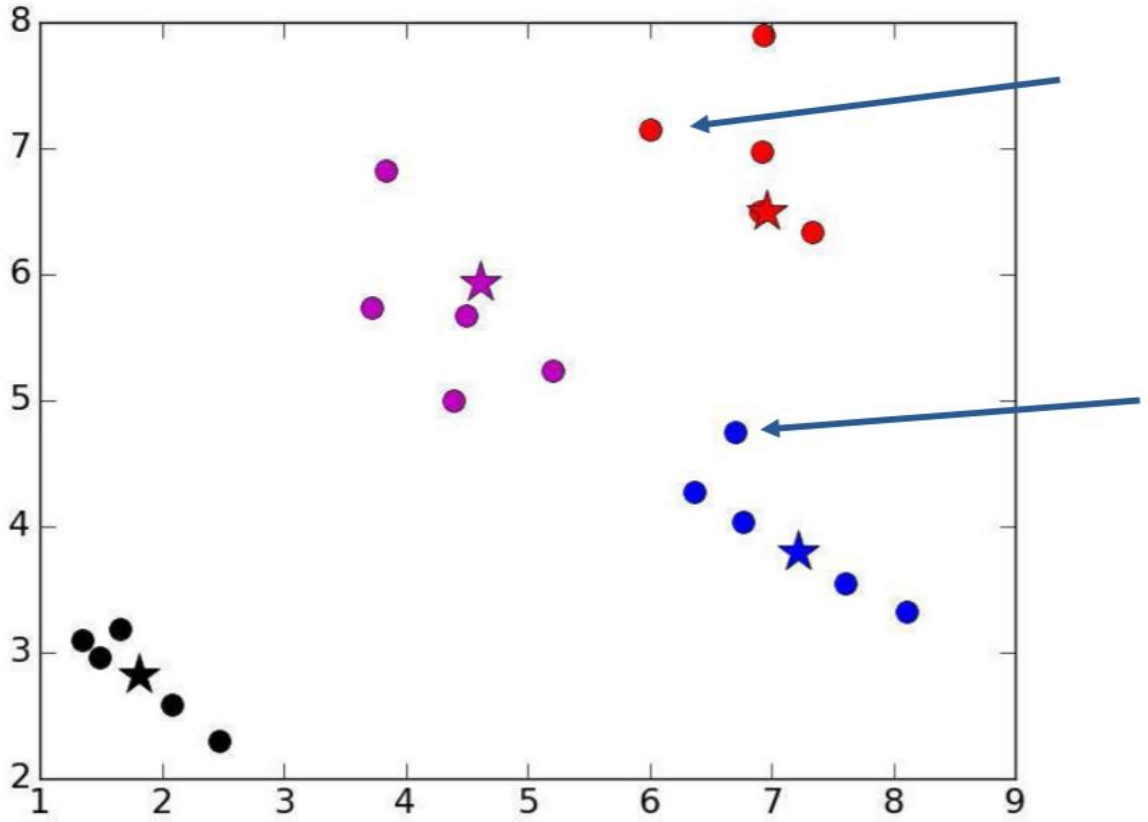
K-means method : 2nd iteration



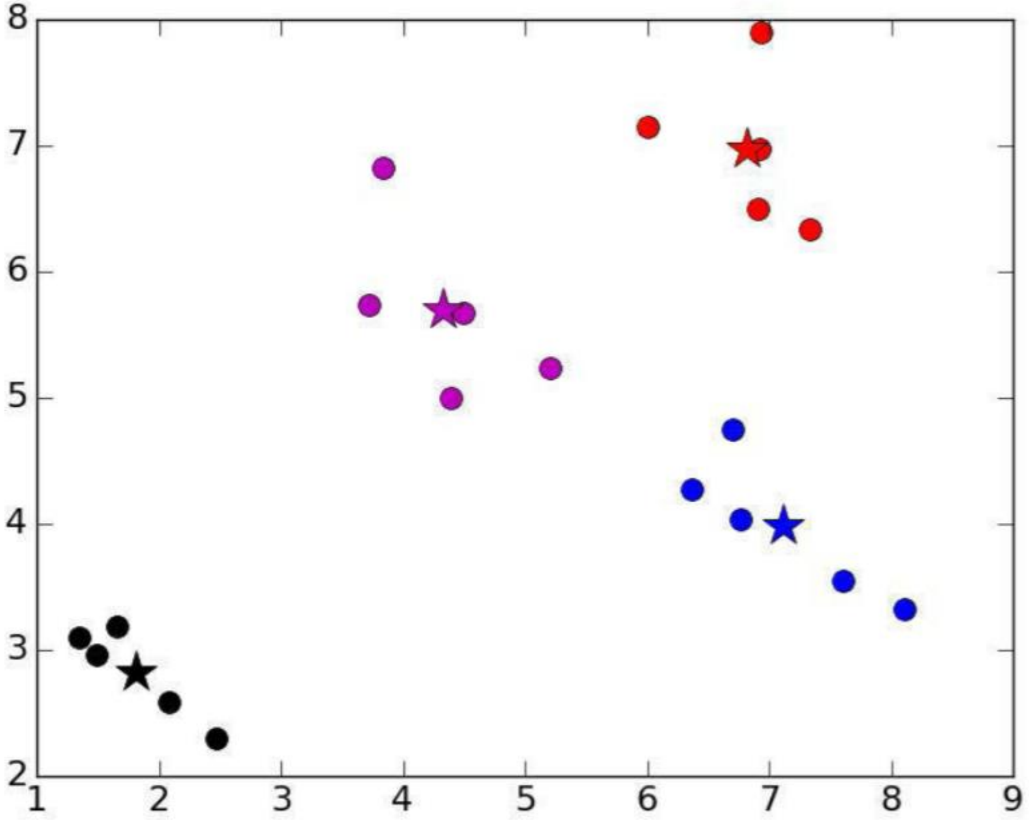
K-means method : 3d iteration



K-means method : 4th iteration



K-means method : 5th iteration

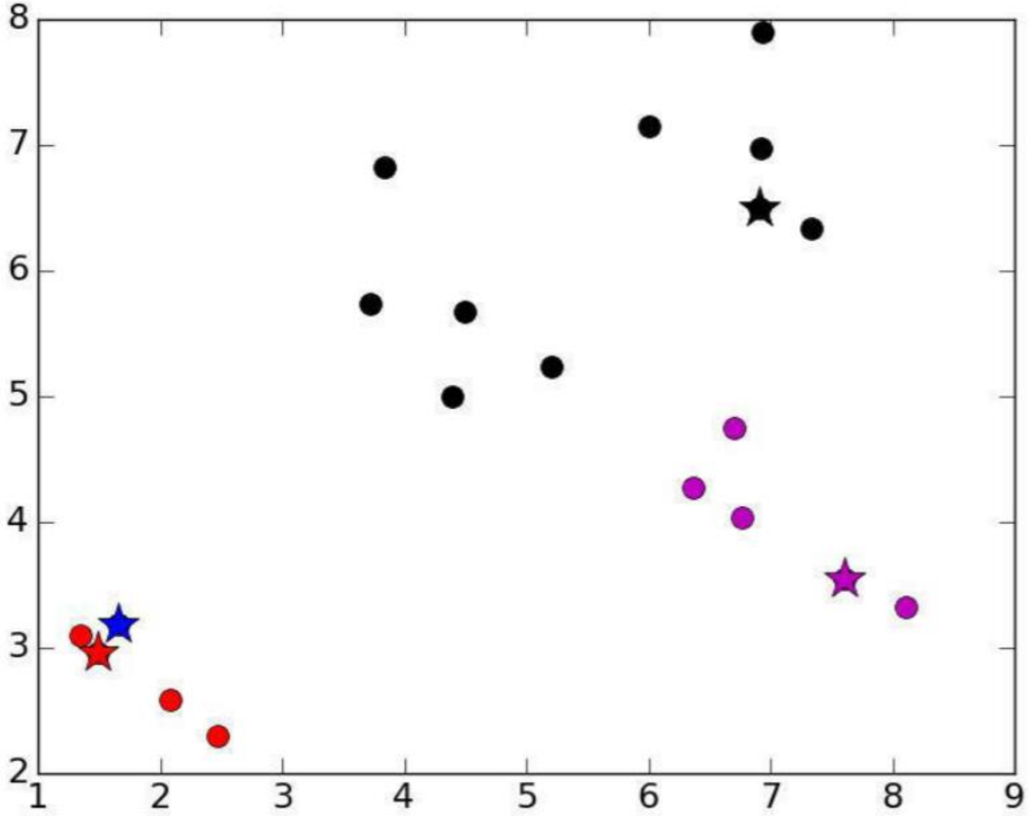


K-means method : Advantages & Drawbacks

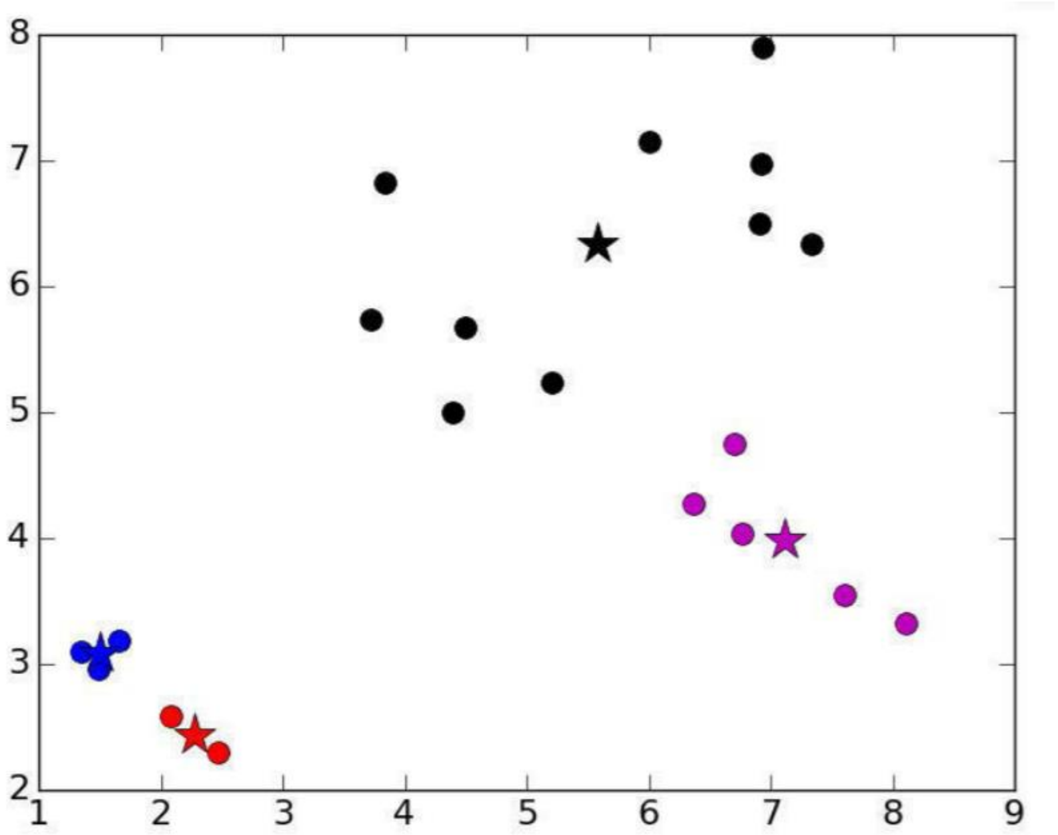
- **Simple**
- Fast

- Gets only local optimas
- Result can depend upon initial centroids
- Choosing the “wrong” k can lead to strange results

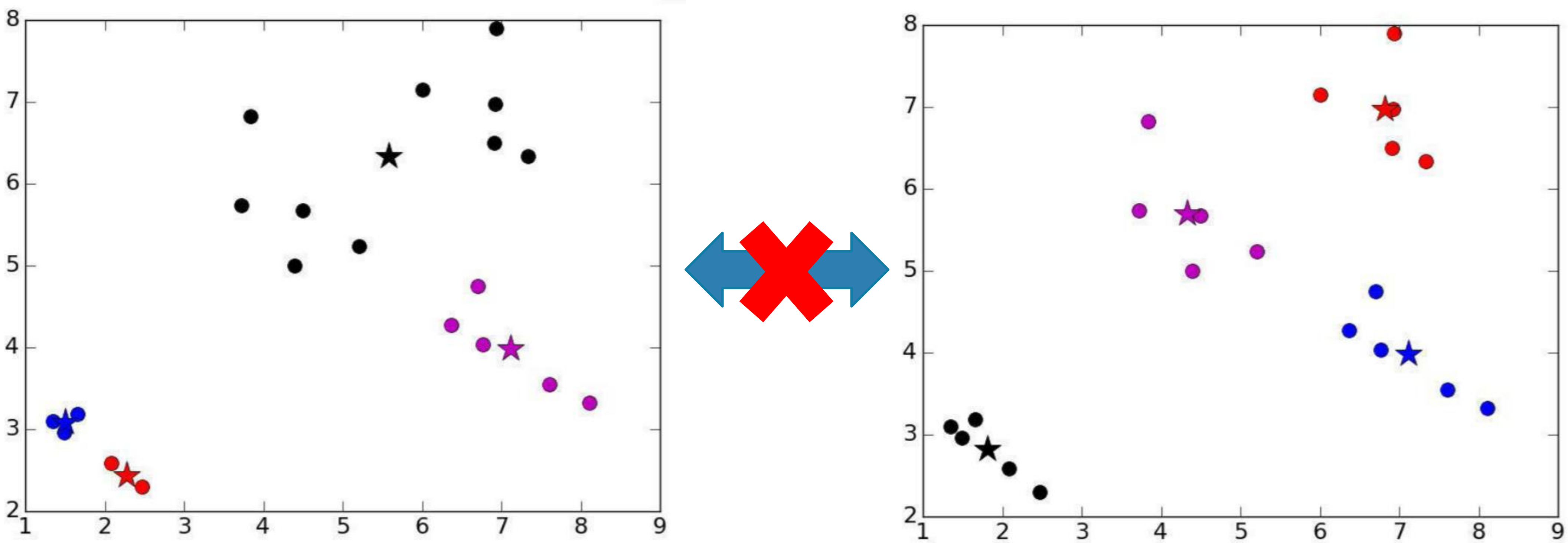
K-means method : Bad initialisation



K-means method : Bad initialisation



K-means method : Bad initialisation

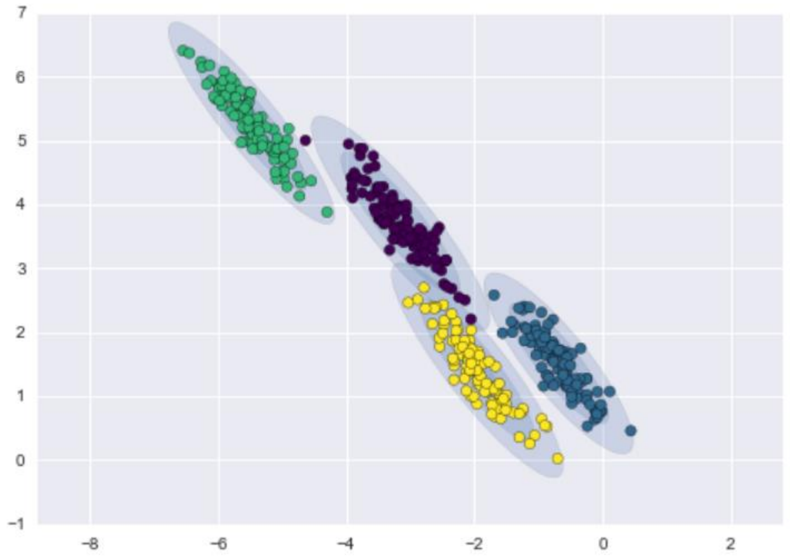


Gaussian mixture method

- The model of a mixture of Gaussian distributions assumes that **our data is a mixture of multidimensional Gaussian distributions** with certain parameters.

- **Uses an expectation–maximization approach** which qualitatively does the following:
    1. Choose starting guesses for the location and shape
    2. Repeat until converged:
        a) ***E-step*** : for each point, find weights encoding the probability of membership in each cluster
        b) ***M-step*** : for each cluster, update its location, normalization, and shape based on *all* data points, making use of the weights

- The result of this is that each cluster is associated not with a hard-edged sphere, but with a smooth Gaussian model.

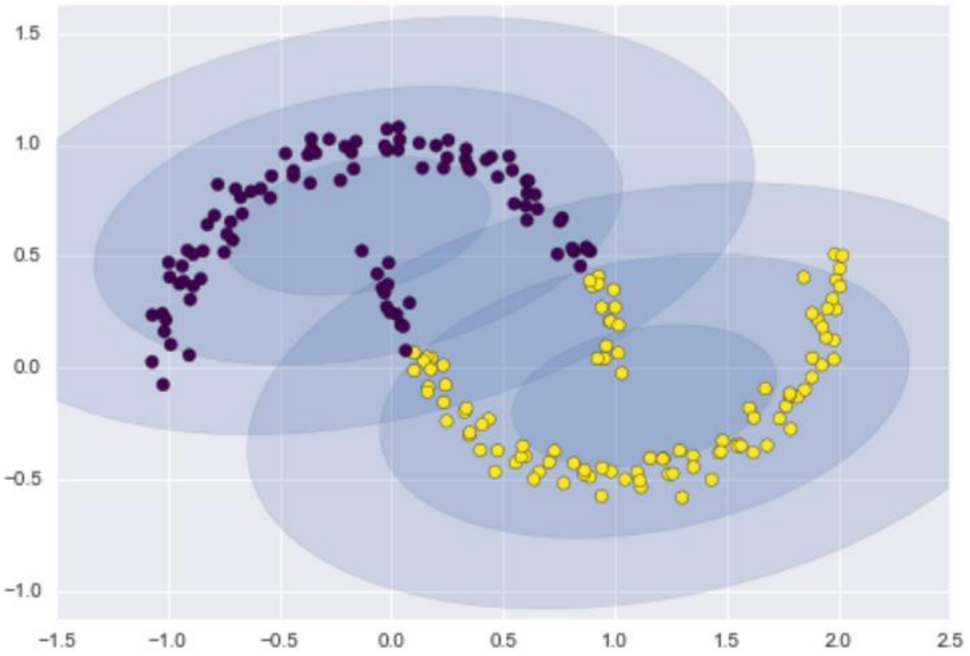


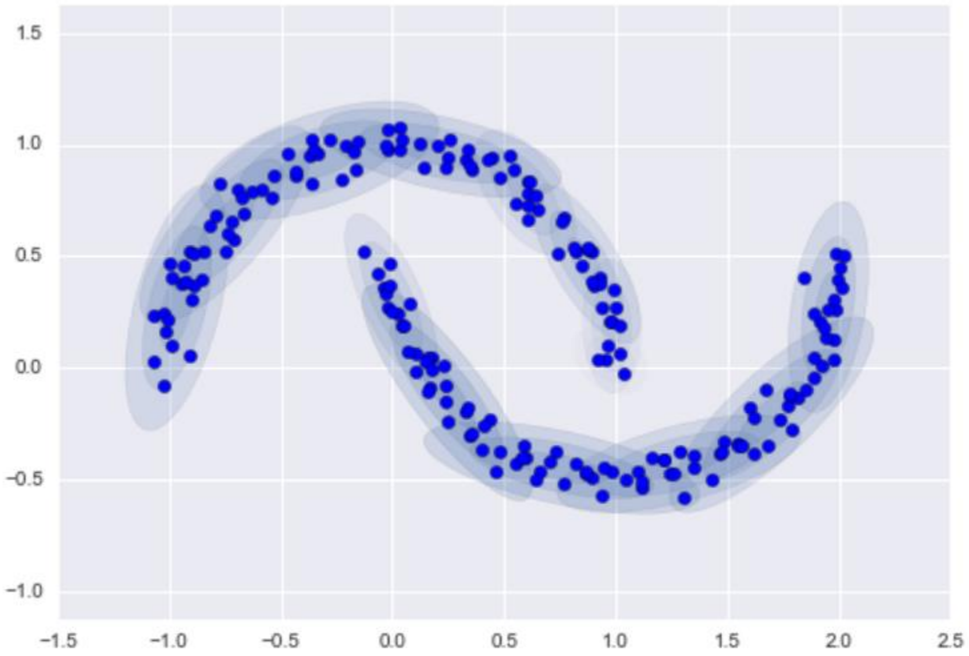
Gaussian mixture method: Advantages & Drawbacks

- **General case of K-means**
- Flexible in terms of cluster covariance
- Can work successfully with mixed distributions

- Gets only local optimas
- Not very successful for data with complex geometry

Gaussian mixture method : Examples





Model evaluation methods

Accuracy metrics

Internal :
(not imply the knowledge about the true labels of the objects)

- Silhouette

External :
(true labels of objects are known)

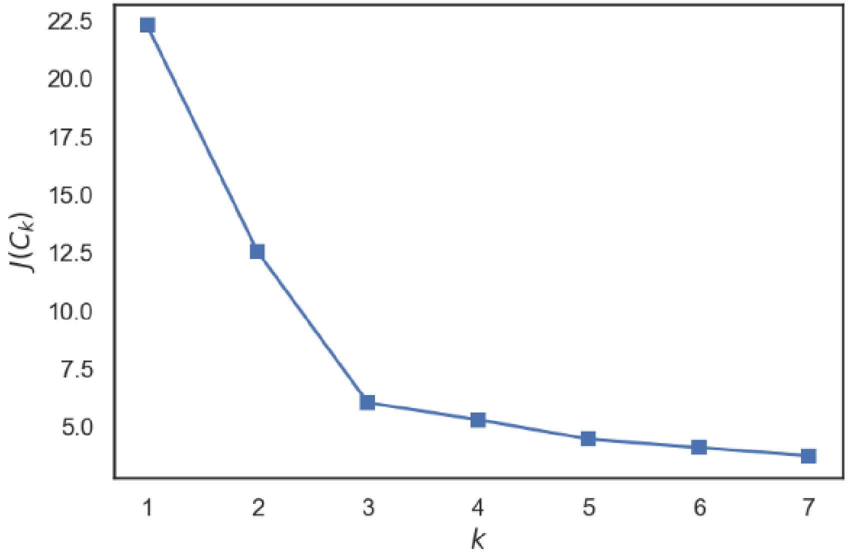
- Adjusted Rand Index (ARI)
- Adjusted Mutual Information (AMI)
- Homogeneity
- Completeness
- V-measure

Accuracy dependence on hyper-parameters curve

<u>For example :</u>

For K-means method, we can use like a internal metric the sum of squared distances between the observations and their centroids :

$$J(C) = \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \mu_k|| \to \min_C,$$



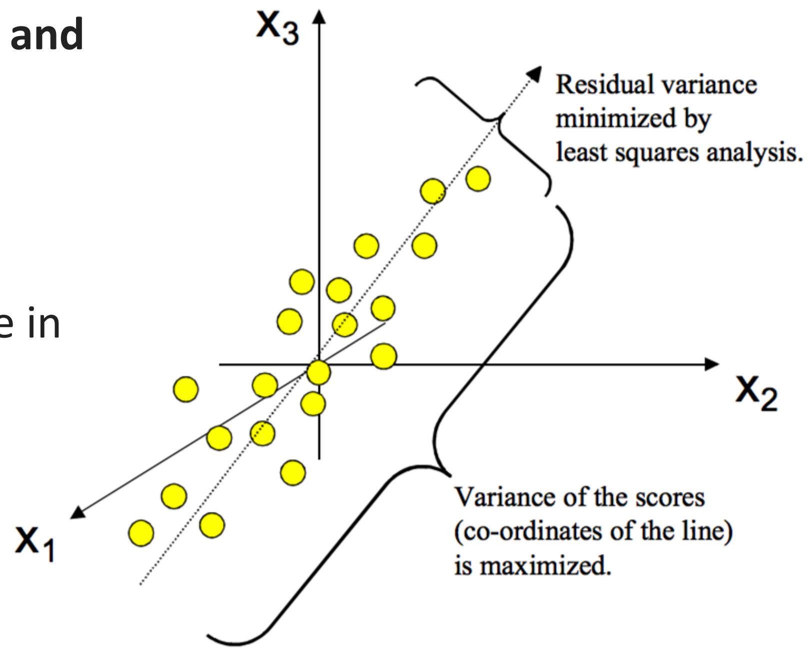
Dimensionality reduction methods

Principal Component Analysis (PCA)

- PCA is a very flexible tool and allows analysis of datasets that may contain, for example, multicollinearity, missing values, categorical data, and imprecise measurements.

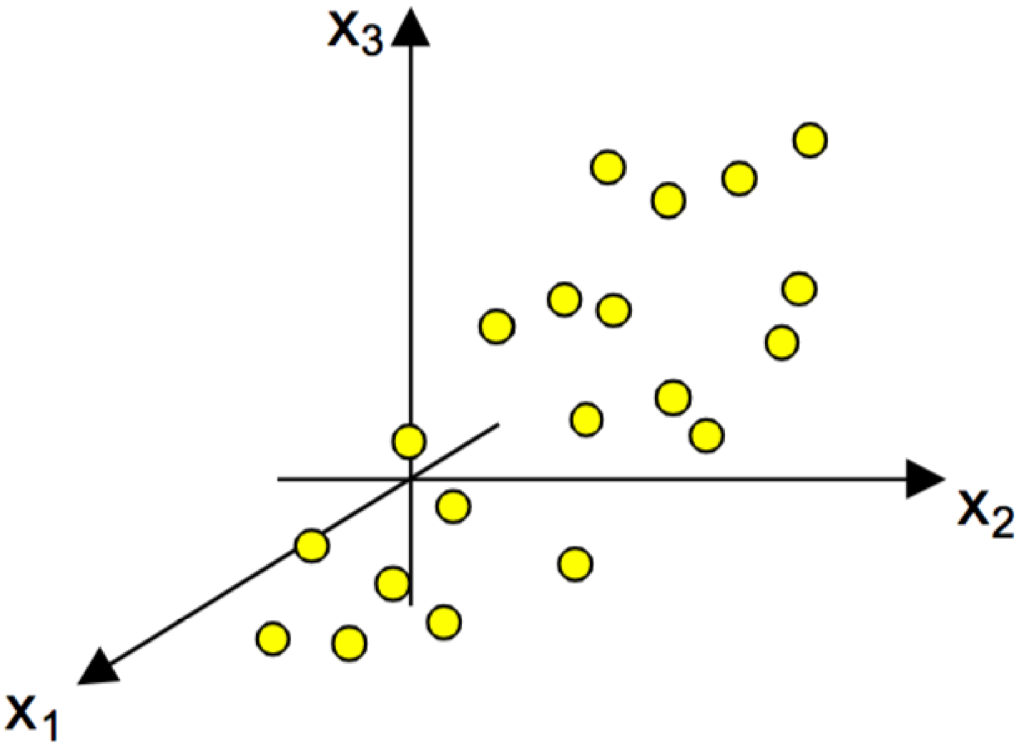
- **The goal is to extract the important information from the data and to express this information as a set of summary indices called principal components.**

- Statistically, PCA finds lines, planes and hyper-planes in the K-dimensional space that approximate the data as well as possible in the least squares sense. A line or plane that is the least squares approximation of a set of data points makes the variance of the coordinates on the line or plane as large as possible.



PCA : geometrical approach

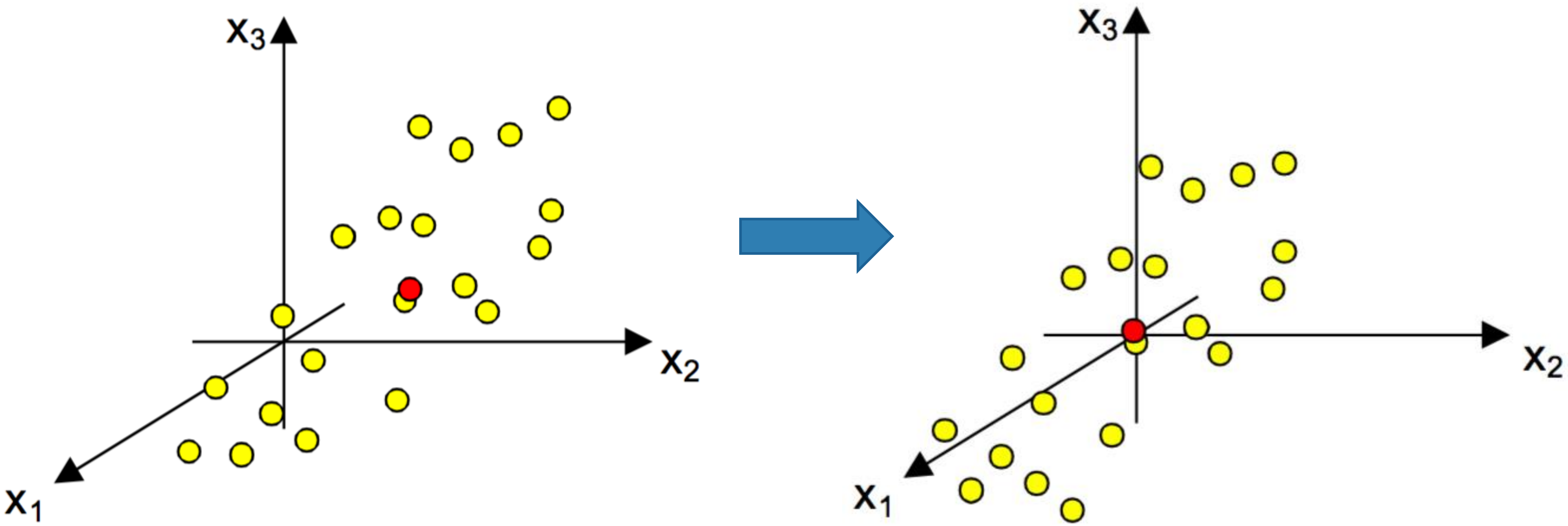
Data points in 3D



PCA : geometrical approach

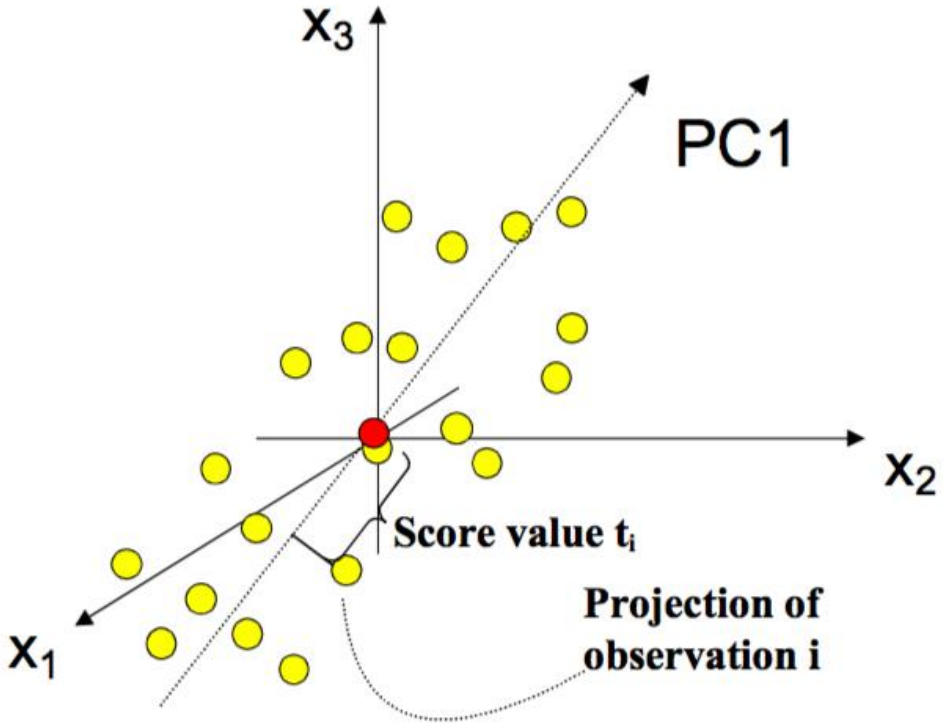
Mean centering :

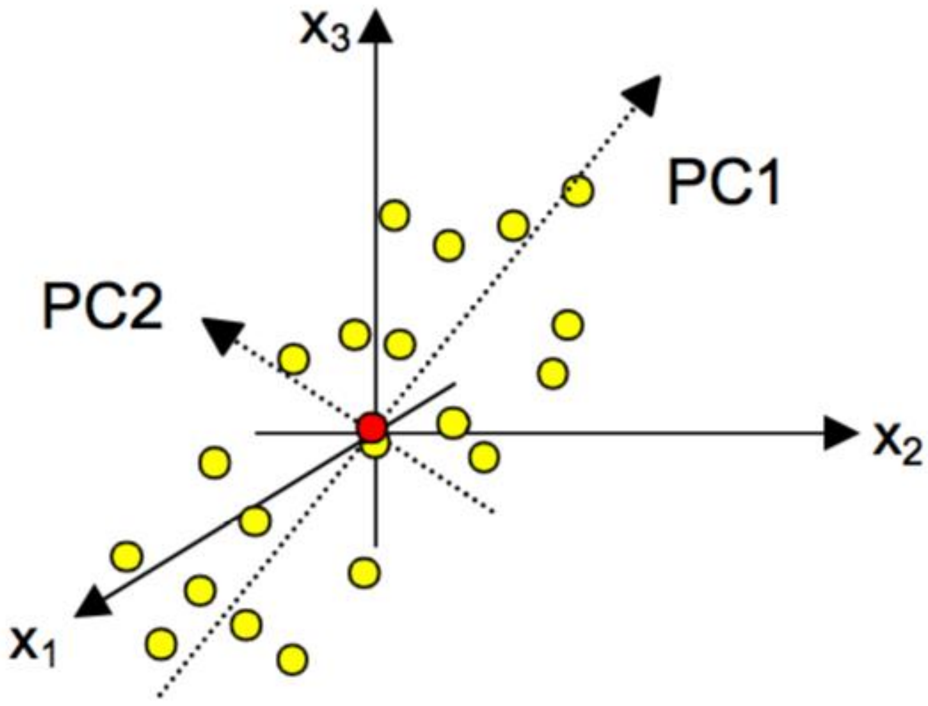
1. Calulate mean
2. Subtract of the averages (mean in the center of coordinates)



PCA : geometrical approach

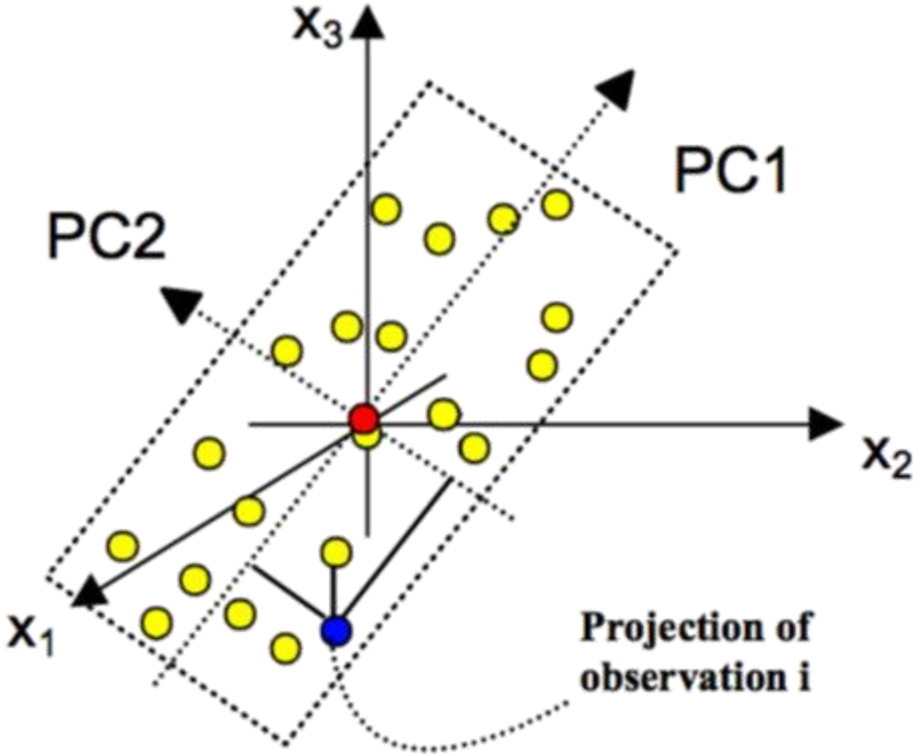
Principal components calculation

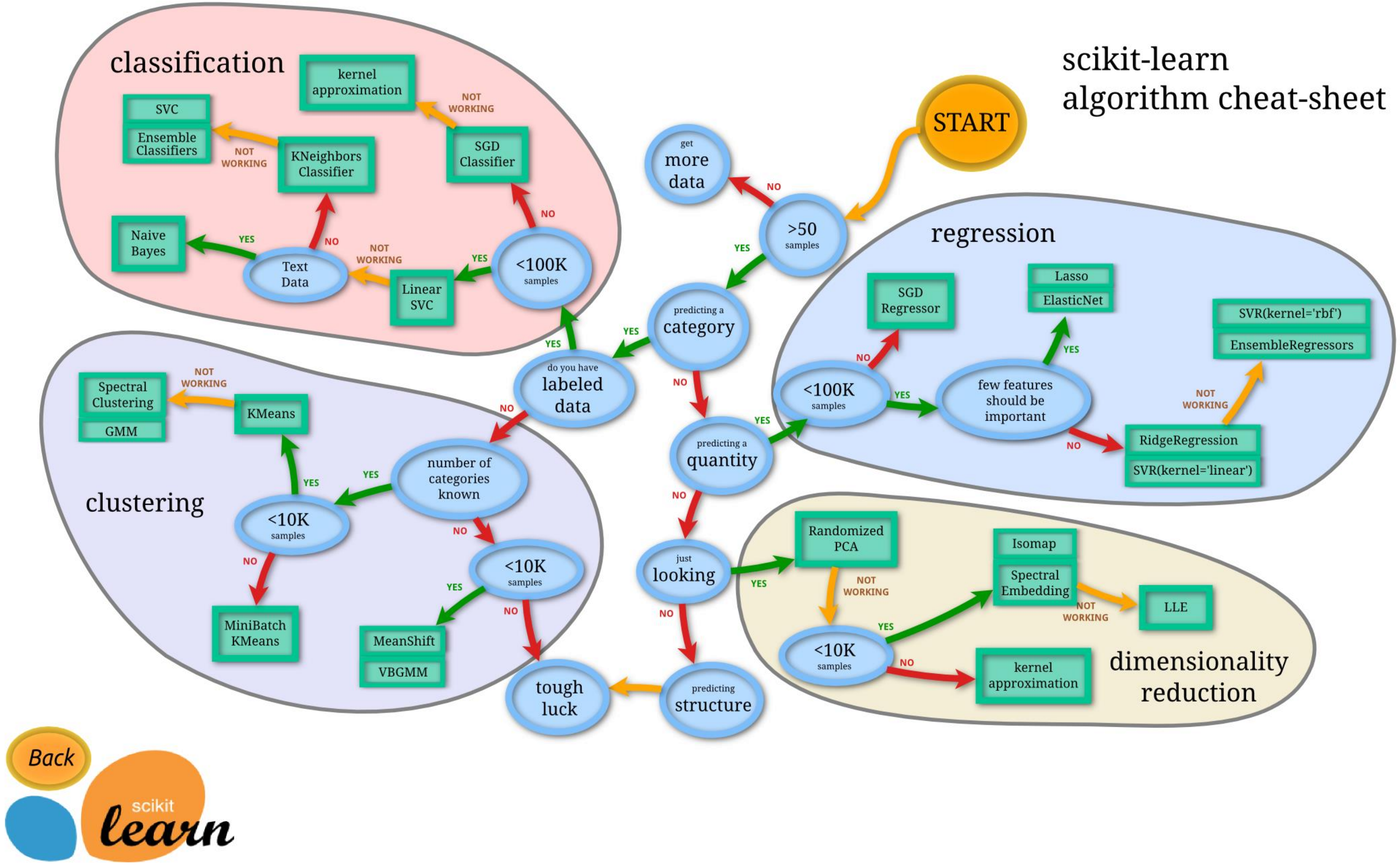




PCA : geometrical approach

Result : Two principal components define a model plane




scikit-learn
algorithm cheat-sheet
START
classification
regression
clustering
dimensionality
reduction
>50
samples
get
more
data
predicting a
category
do you have
labeled
data
predicting a
quantity
just
looking
predicting
structure
tough
luck
<100K
samples
SGD
Classifier
kernel
approximation
Linear
SVC
Text
Data
Naive
Bayes
KNeighbors
Classifier
SVC
Ensemble
Classifiers
number of
categories
known
<10K
samples
KMeans
Spectral
Clustering
GMM
MiniBatch
KMeans
MeanShift
VBGMM
SGD
Regressor
few features
should be
important
Lasso
ElasticNet
RidgeRegression
SVR(kernel='linear')
SVR(kernel='rbf')
EnsembleRegressors
Randomized
PCA
Isomap
Spectral
Embedding
LLE
kernel
approximation
YES
NO
NOT
WORKING
Back
scikit
learn

Sources

- MIT course “Introduction to Computational Thinking and Data Science” (Prof. Eric Grimson, Prof. John Guttag)
- Open Machine Learning Course (by Yury Kashnitsky, mlcourse.ai)
- YouTube lections “Algorithms and Concepts” (by CodeEmporium)