Module 1 :

Machine Learning Review

Unsupervised Learning Algorithms

Géraldine Schaller-Conti, August 2024

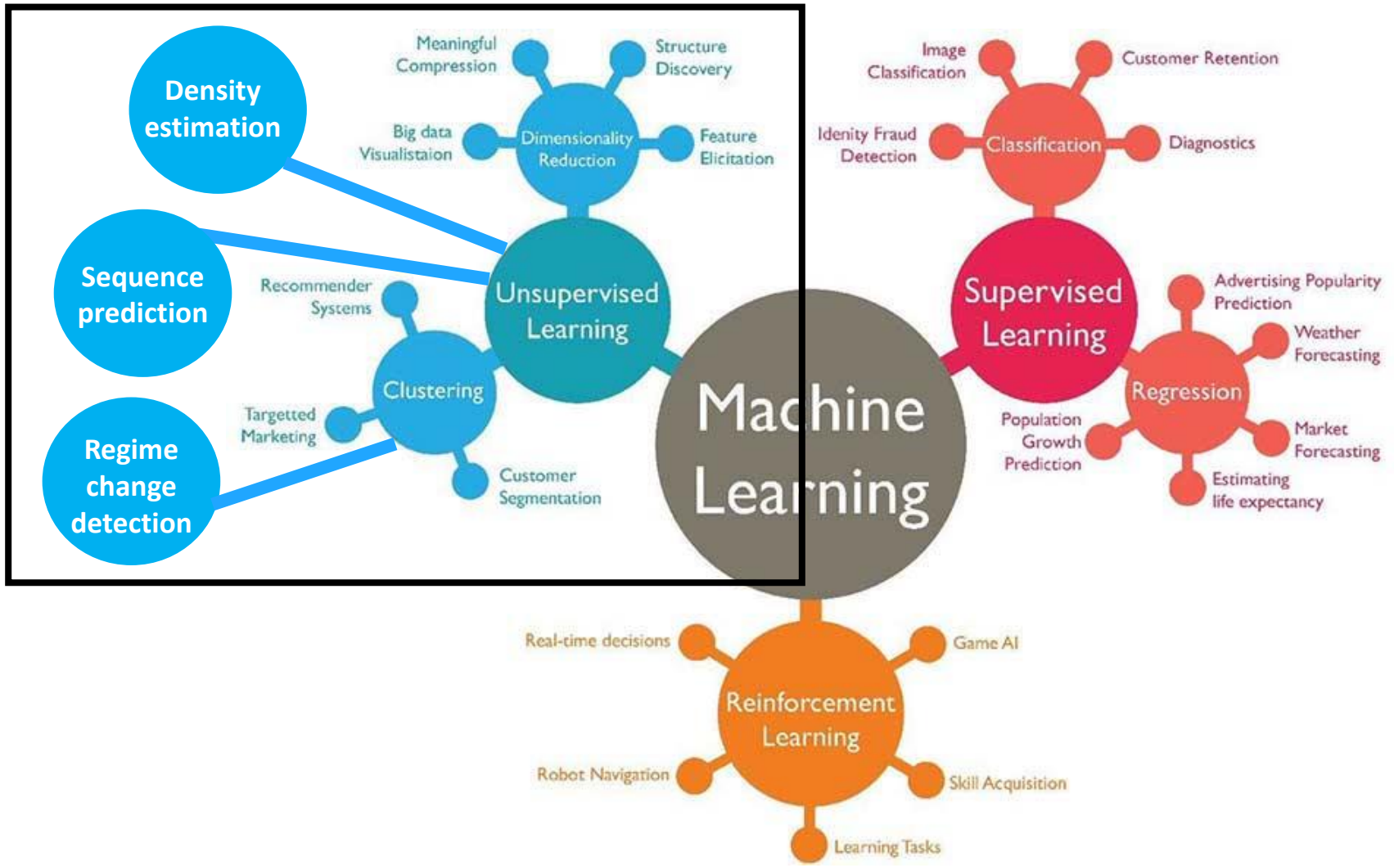
Discussion Session

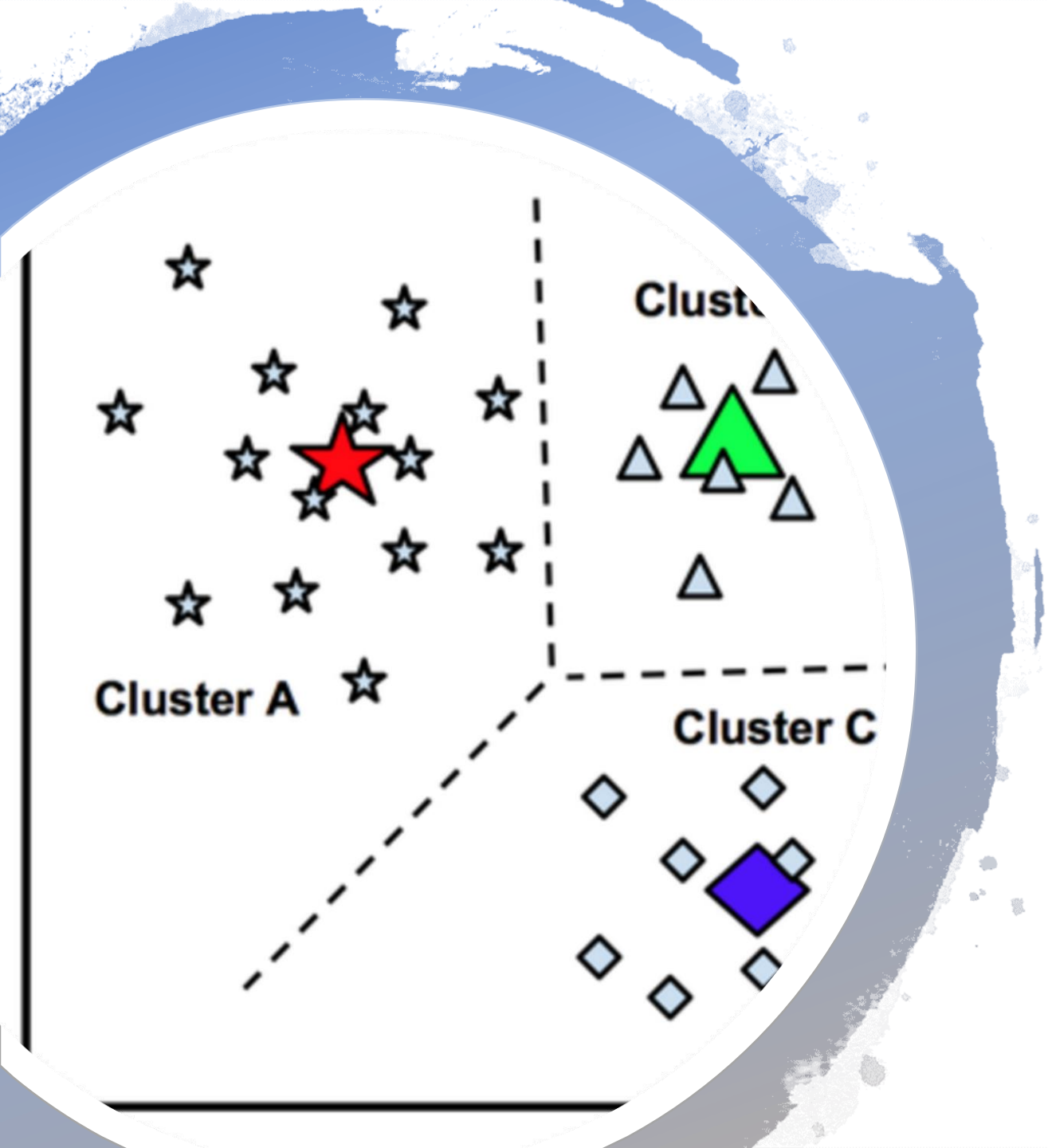
- Review of Notebook 2 (regression)
  - Pipeline for processing numerical attributes
  - Train model
  - linear regression, decision tree, random forest and SVR
  - Compare the models (MSE, MAE, cross-validation)
  - Fine-tune model (grid search, random search)
  - Assess the performance on the test set (RMSE, 95% CL)

Bibliography

- Deep Learning book (Goodfellow, Bengio, Courville)
- Machine Learning @ Stanford (Prof Andrew Ng)
- Hands-On Machine Learning with Scikit-Learn & Tensorflow (Aurélien Géron)

Learning Objectives





Clustering

- K-means

- Hierarchical clustering

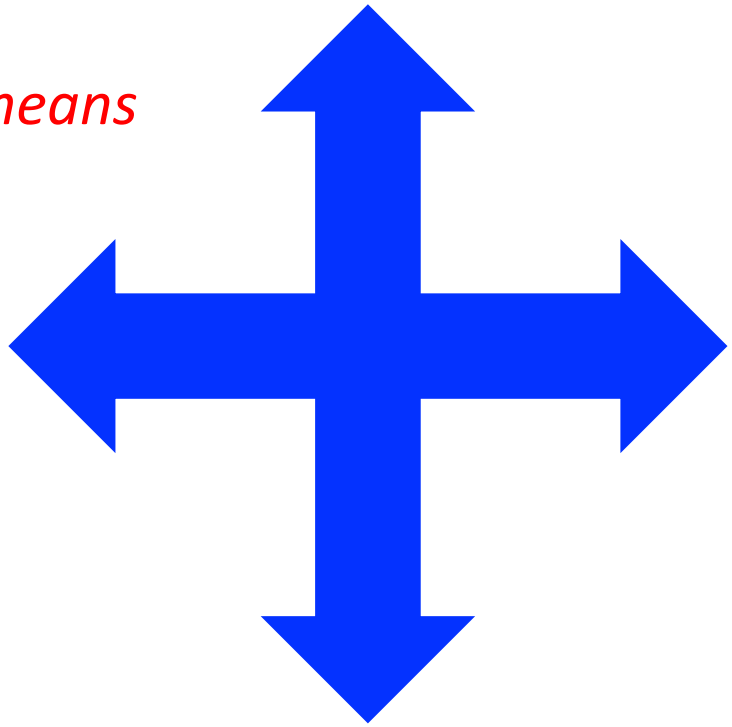
Types of clustering

when efficiency is important

Flat

No explicit structure should relate the clusters to each other

K-means



Hard

Soft

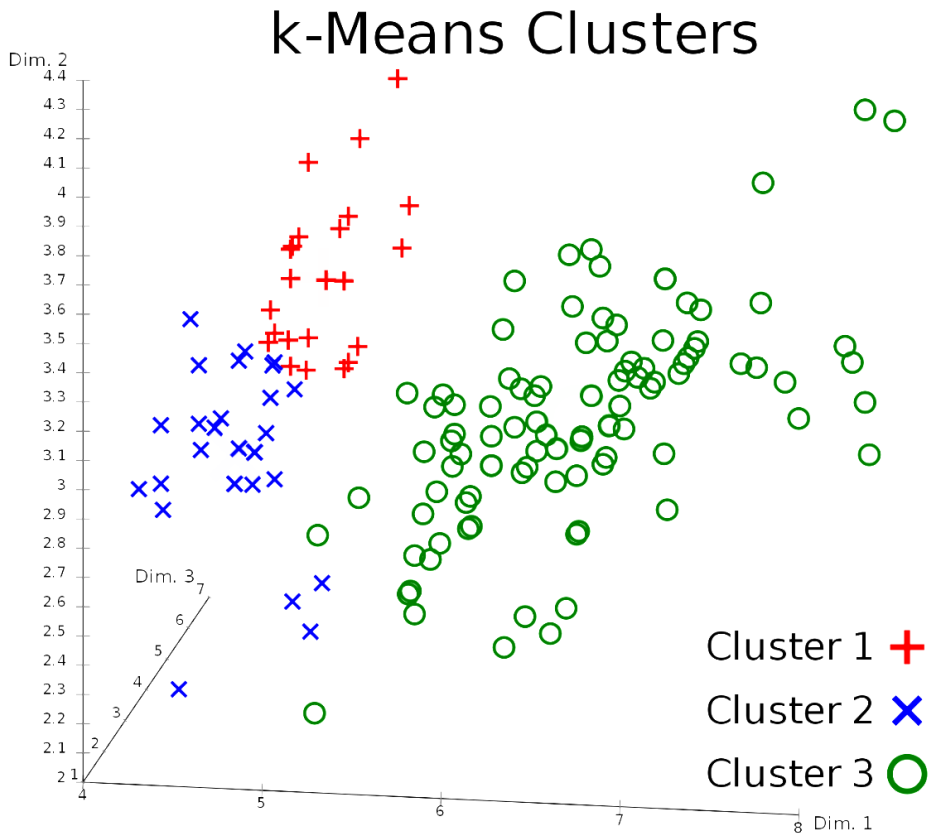
Takes into account uncertainty of the association

Hierarchical

- when potential problem of flat clustering is a concern
- believed to produce better clusters

K-Means : Introduction

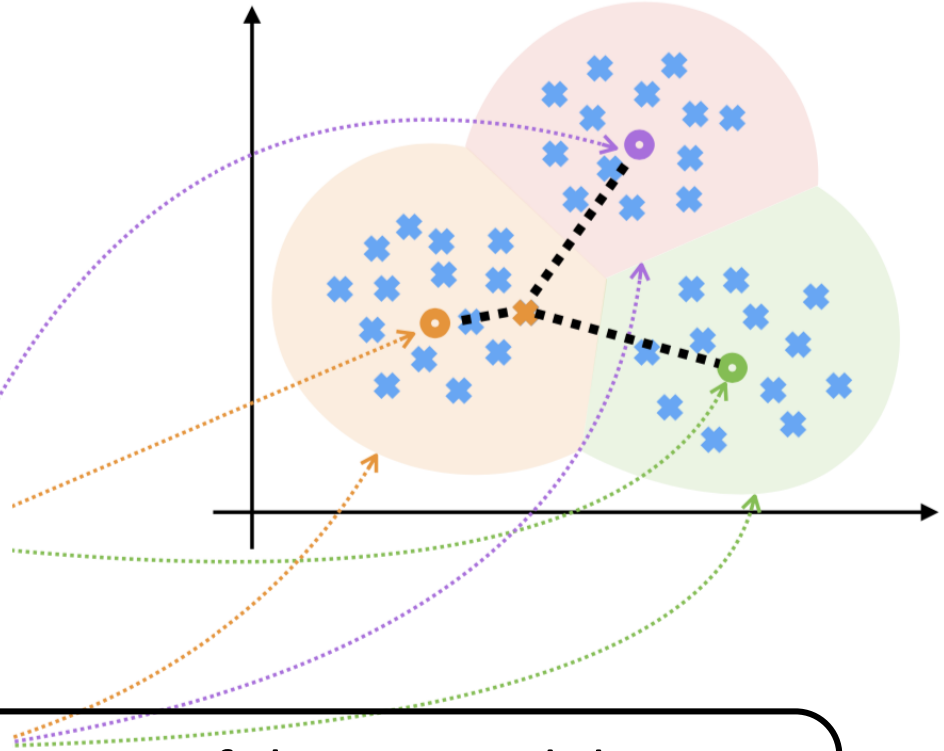
- Partition the dataset into K non-overlapping clusters
  - "K" : *the number of clusters*
  - "means" : data is averaged (ie. finding centroids)
  - Each data point belongs to only one group



Optimization (EM algorithm)

0) Initialize the centroids

2) Maximization step : Compute the centroids for the clusters by taking the average of all the data points belonging to each cluster



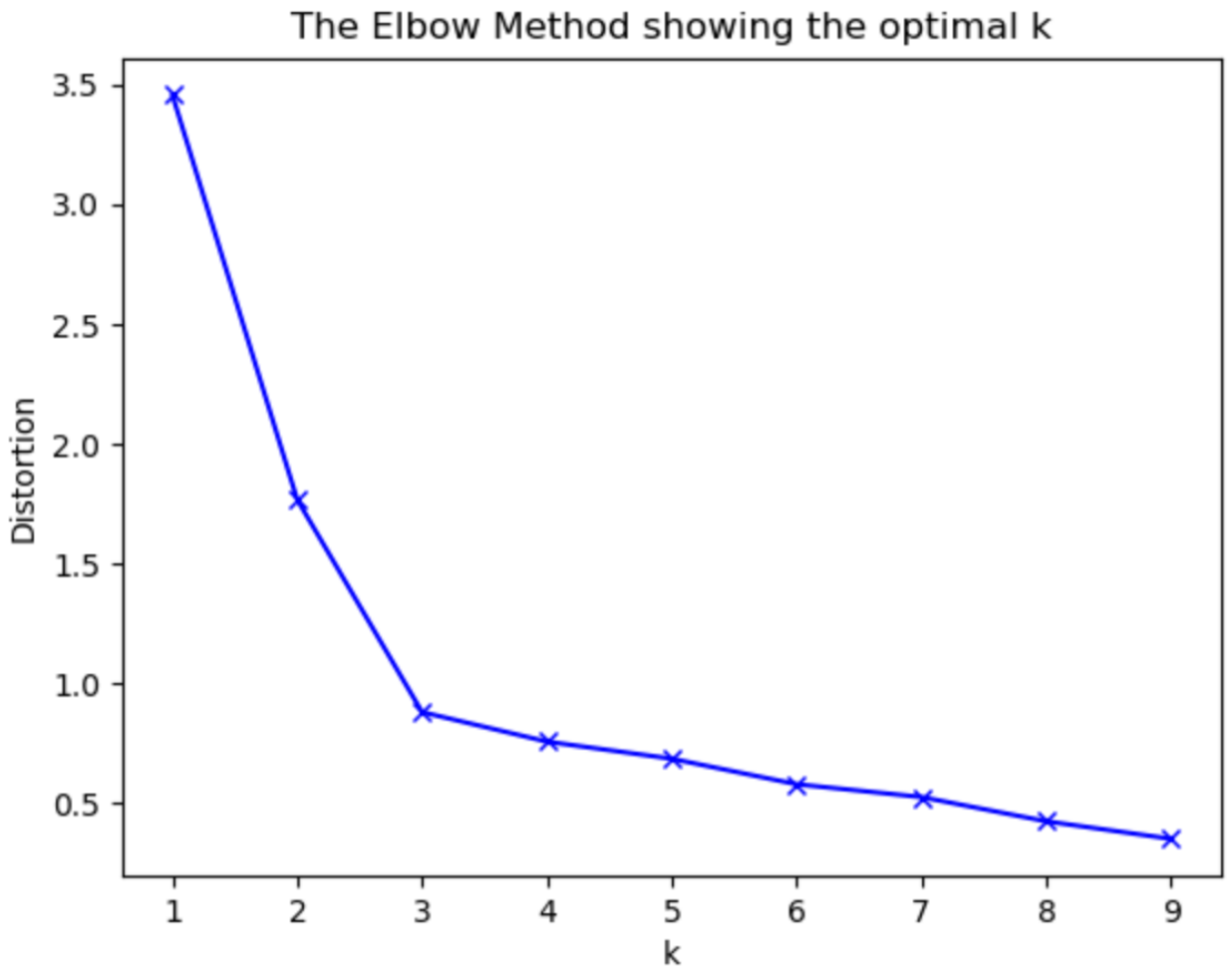
alternate

1) Expectation step : Compute the sum of the squared distance between the data points and centroids and assign each data point to the closest cluster

- Stop criteria :
  - Centroids have stabilized wrt some tolerance
  - Number of iterations has been achieved

Elbow method : based on Sum of Squared Errors between data points and cluster centroids

Number of clusters (K)



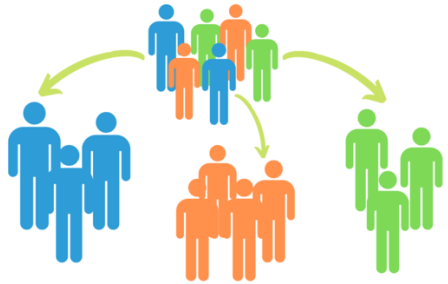
Applications

Customer segmentation

https://towardsdatascience.com/customer-segmentation-using-k-means-clustering-d33964f238c3

Goal : you own a supermarket mall and you want to find who are your target customers

Data : membership card data (ID, age, gender, annual income, spending score)



Document clustering

https://resources.experfy.com/bigdata-cloud/k-means-clustering-in-text-data/

Goal : you want to analyze Facebook/Twitter/Youtube comments of a particular event

Data : social media text data

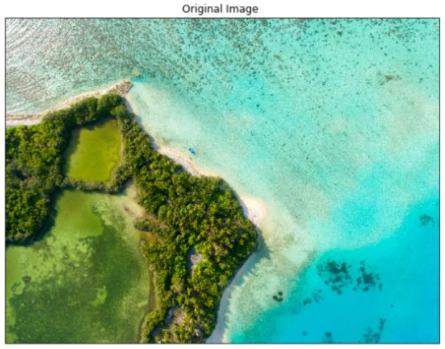
Documents	Online	Festival	Book	Flight	Delhi
D1	1	0	1	0	1
D2	2	1	2	1	1
D3	0	0	1	1	1
D4	1	2	0	2	0
D5	3	1	0	0	0
D6	0	1	1	1	2
D7	2	0	1	2	1
D8	1	1	0	1	0
D9	1	0	2	0	0
D10	0	1	1	1	1

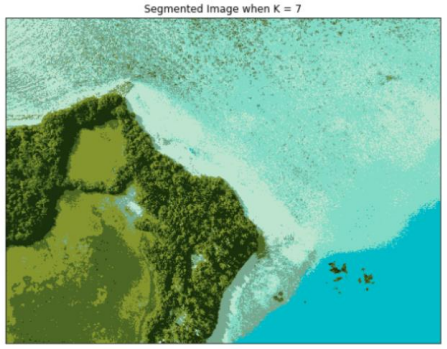
Image compression

https://www.geeksforgeeks.org/image-compression-using-k-means-clustering/

Goal : to compress a digital image to reduce its cost of storage or transmission

Data : colored image (pixels)





In practice

- sklearn.cluster.KMeans

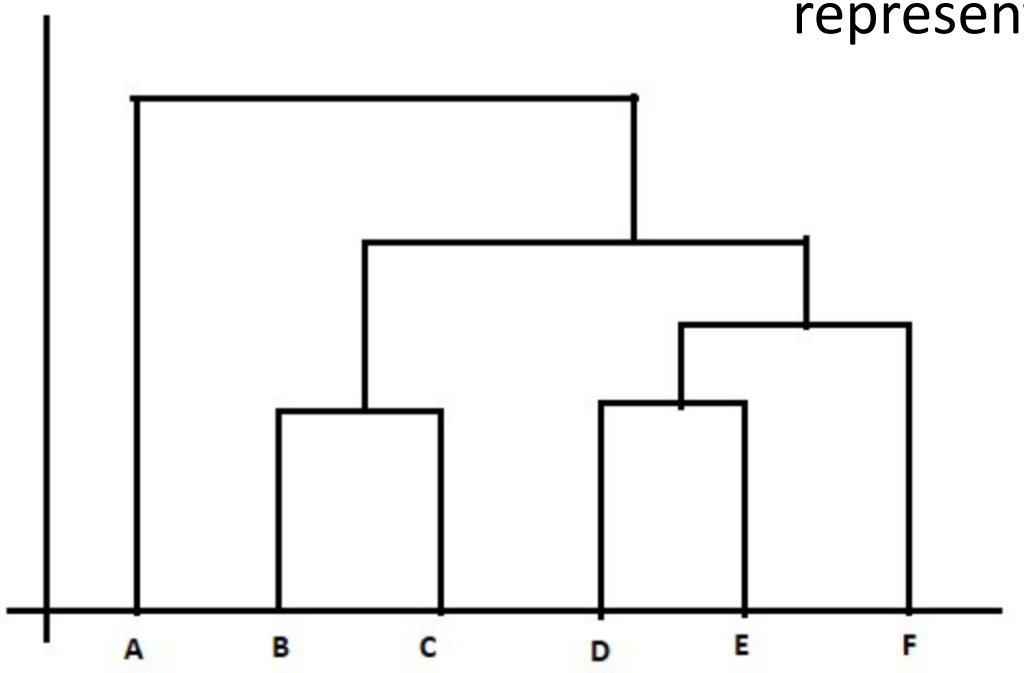
```
>>> from sklearn.cluster import KMeans
>>> import numpy as np
>>> X = np.array([[1, 2], [1, 4], [1, 0],
...               [10, 2], [10, 4], [10, 0]])
>>> kmeans = KMeans(n_clusters=2, random_state=0).fit(X)
>>> kmeans.labels_
array([1, 1, 1, 0, 0, 0], dtype=int32)
>>> kmeans.predict([[0, 0], [12, 3]])
array([1, 0], dtype=int32)
>>> kmeans.cluster_centers_
array([[10.,  2.],
       [ 1.,  2.]])
```

- If large dataset (>10k samples) :
  - sklearn.cluster.MiniBatchKMeans

Hierarchical Clustering : Introduction

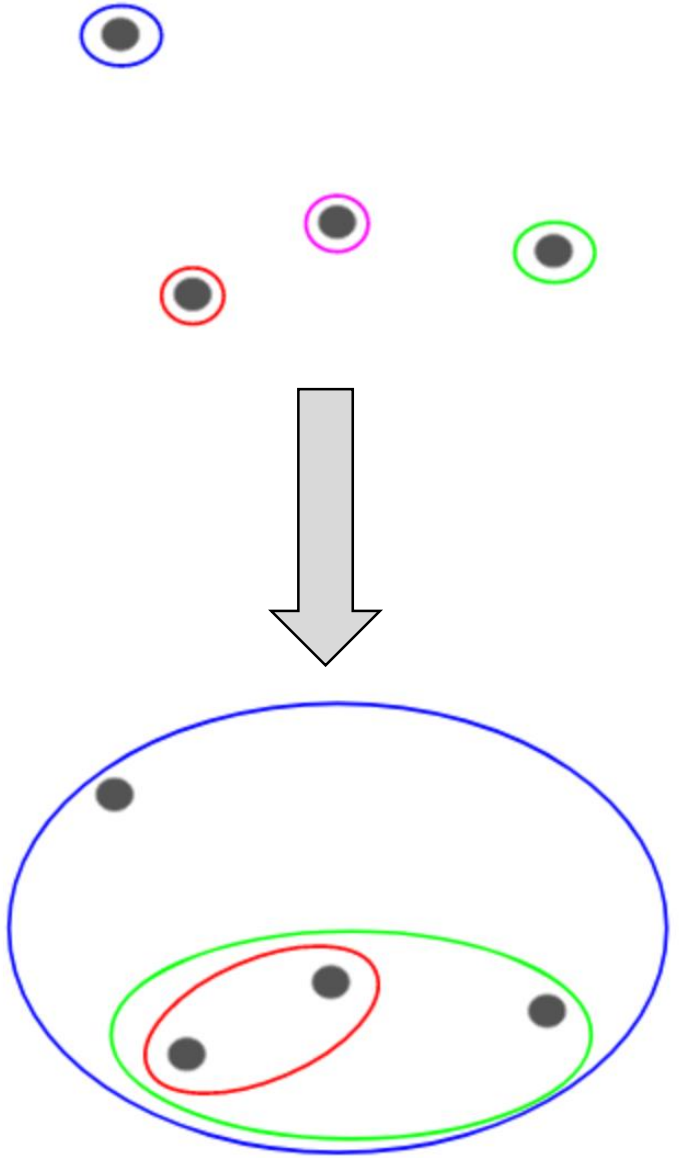
- Build a hierarchy of clusters
- Two types :
  - Agglomerative
  - Divisive (not much used, computationally more expensive and can yield to imbalanced clusters)

Dendogram representation



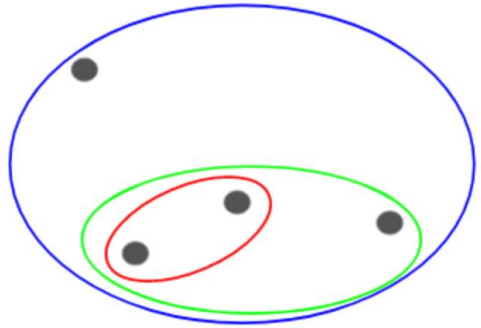
Optimization (Agglomerative)

- Agglomerative algorithm: each data point is considered as an individual cluster
  1) Calculate the proximity of individual points
  2) Iteration :
     - Merge the two closest clusters
     - Update the proximity matrix
- Stop criterium :
  - a single cluster remains

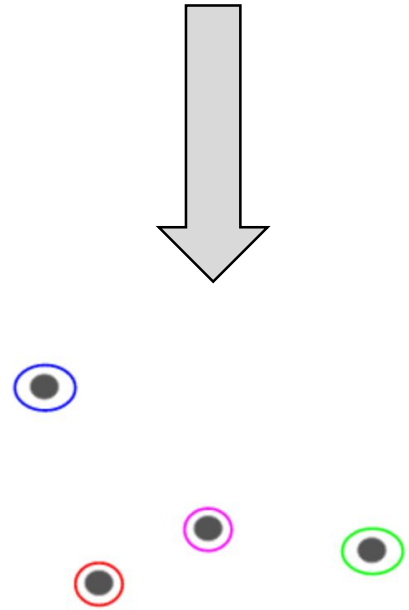


Optimization (Divisive)

- **Initialization**: single cluster (root cluster)
- **Divisive Step**: In each iteration, choose a cluster to divide based on distance choice



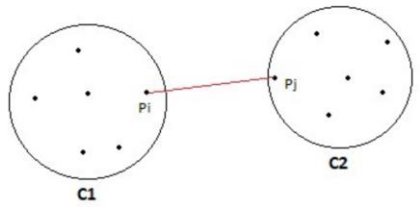
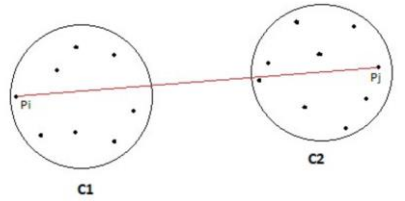
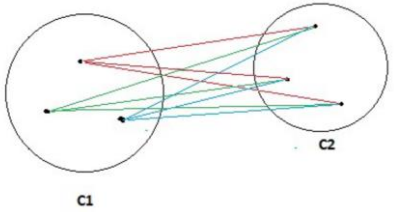
- **Division Process**: The chosen cluster is divided into two or more smaller sub-clusters. This is typically done by finding a cutting point that separates the data points into groups that are more distinct from each other.



- **Stopping Criterion**:
  - predefined number of desired clusters
  - threshold level of similarity
  - maximum number of iterations

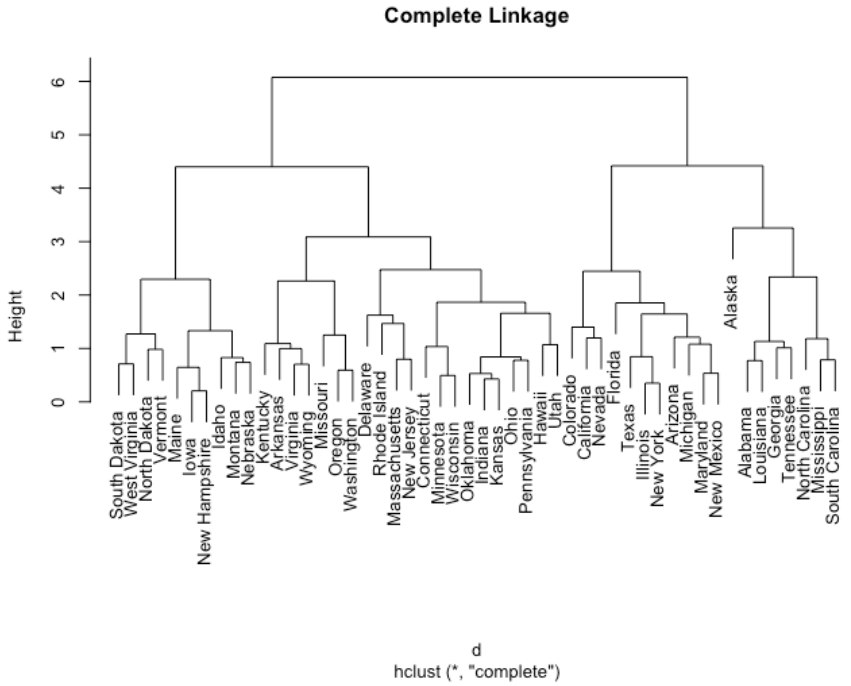
Proximity choice (Linkage)

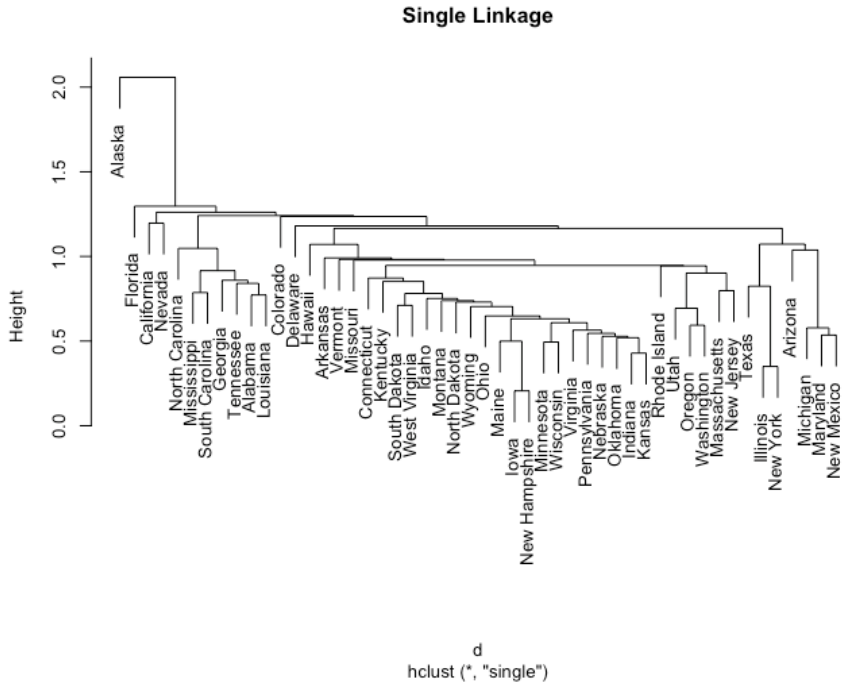
Let's see how to compute proximity…

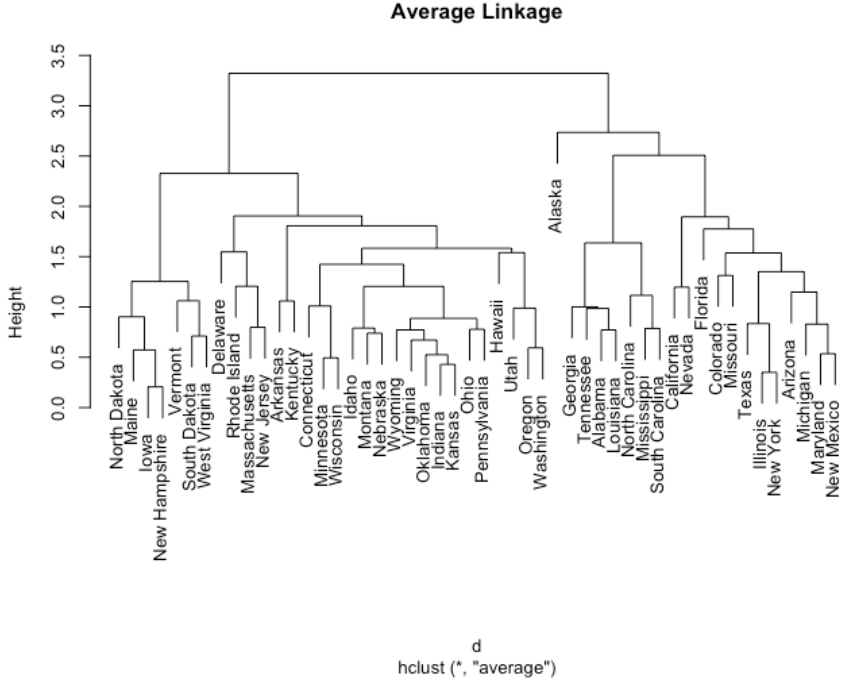
	Single linkage algorithm (MIN)	Complete linkage algorithm (MAX)	Group Average *
Proximity			
Pros	separate non-elliptical shapes if gap between two clusters is not small	Separates well clusters even if there is noise between clusters	Separates well clusters even if there is noise between clusters
Cons	cannot separate clusters properly if there is noise between clusters	Bias towards globular clusters Tends to break large clusters	Bias towards globular clusters

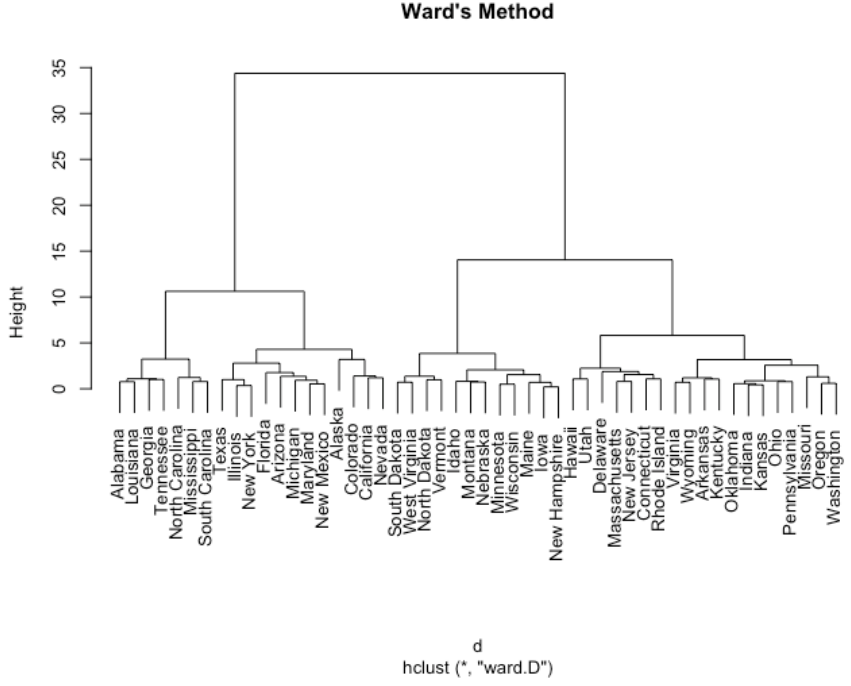
* Ward's method is similar, but it calculates the sum of the squares of the distances

Proximity illustration









Number of clusters K

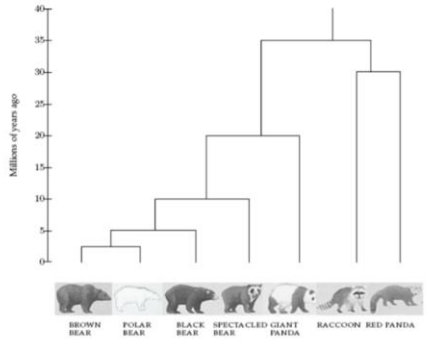
- Cut the dendogram at a certain level to obtain a specific number of clusters
- Dendrogram Visualization: look for a point where the vertical lines are relatively long, indicating a significant merge.
- Elbow Method: Plot the distances (or dissimilarities) at which the merges occur in the dendrogram. Look for an "elbow point" where the distances start to increase rapidly.
- Calinski-Harabasz Index: Also known as the Variance Ratio Criterion, this index measures the ratio of between-cluster variance to within-cluster variance. A higher Calinski-Harabasz index indicates a better clustering solution.
- Davies-Bouldin Index: This index quantifies the average similarity between each cluster and its most similar cluster. A lower Davies-Bouldin index indicates a better clustering solution.

Applications

Animal evolution

Goal : find the phylogenetic tree relating different species together

Data : DNA sequences

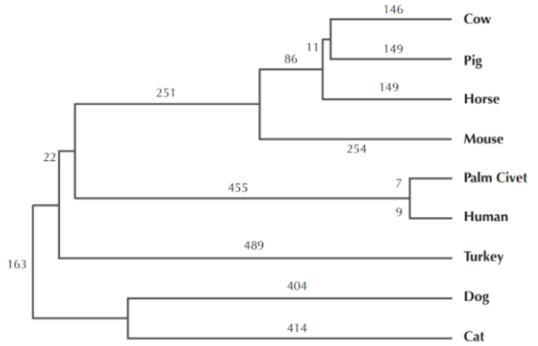


https://towardsdatascience.com/hierarchical-clustering-and-its-applications-41c1ad4441a6

Tracking viruses

Goal : track viral outbreaks and their sources (SARS virus example)

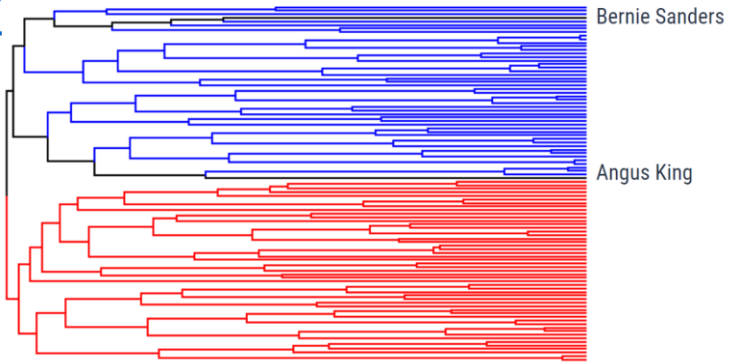
Data : DNA sequences



Find party lines through Twitter

Goal : Cluster US senators into their respective parties

Data : Twitter data



In Practice

- sklearn.cluster.AgglomerativeClustering

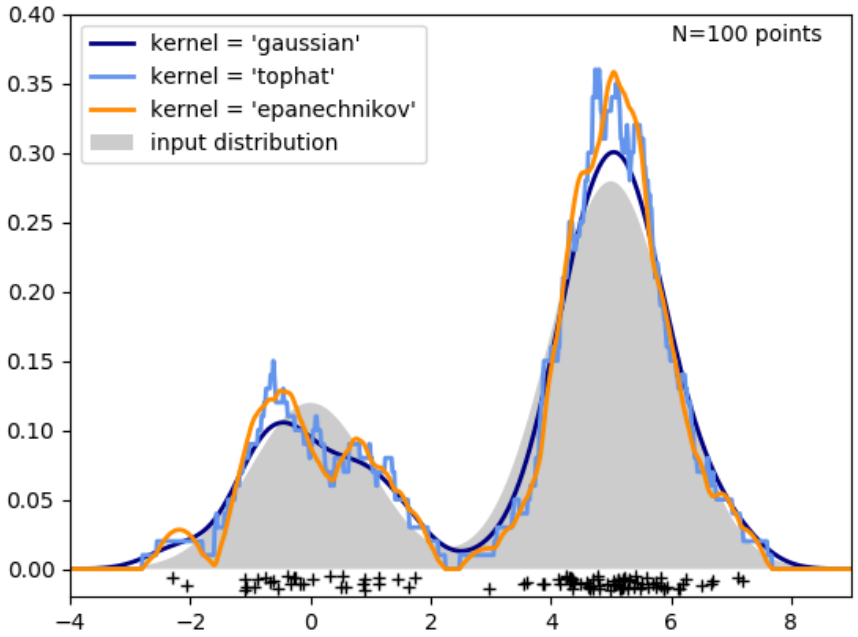
```
#----------------------------------------------------------------------
# 2D embedding of the digits dataset
print("Computing embedding")
X_red = manifold.SpectralEmbedding(n_components=2).fit_transform(X)
print("Done.")

from sklearn.cluster import AgglomerativeClustering

for linkage in ('ward', 'average', 'complete', 'single'):
    clustering = AgglomerativeClustering(linkage=linkage, n_clusters=10)
    t0 = time()
    clustering.fit(X_red)
    print("%s :\t%.2fs" % (linkage, time() - t0))
```

K-means versus hierarchical clustering

	K-means	**Hierarchical clustering**
Advantages	- Faster - Simple	- Deterministic (reproducible)
Disadvantages	- Assumption (spherical, evenly sized clusters) - Choice of K	- Slower - Choice of similarity

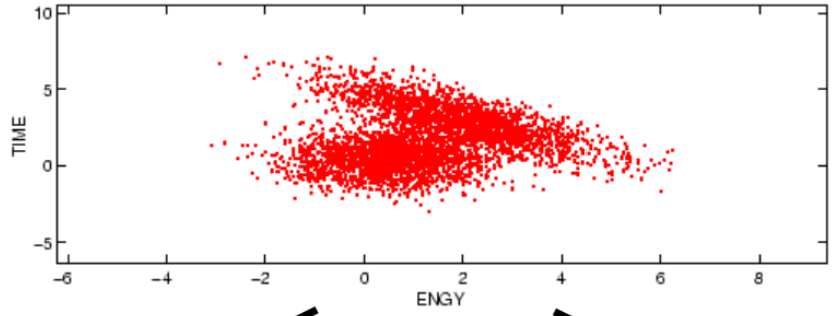


Density estimation

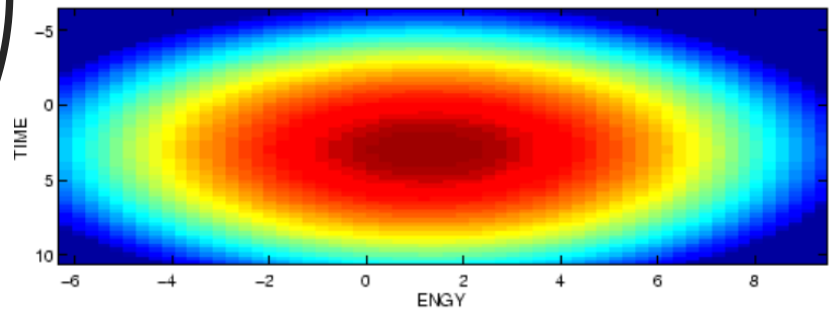
Observe data from a complex but unknown probability distribution

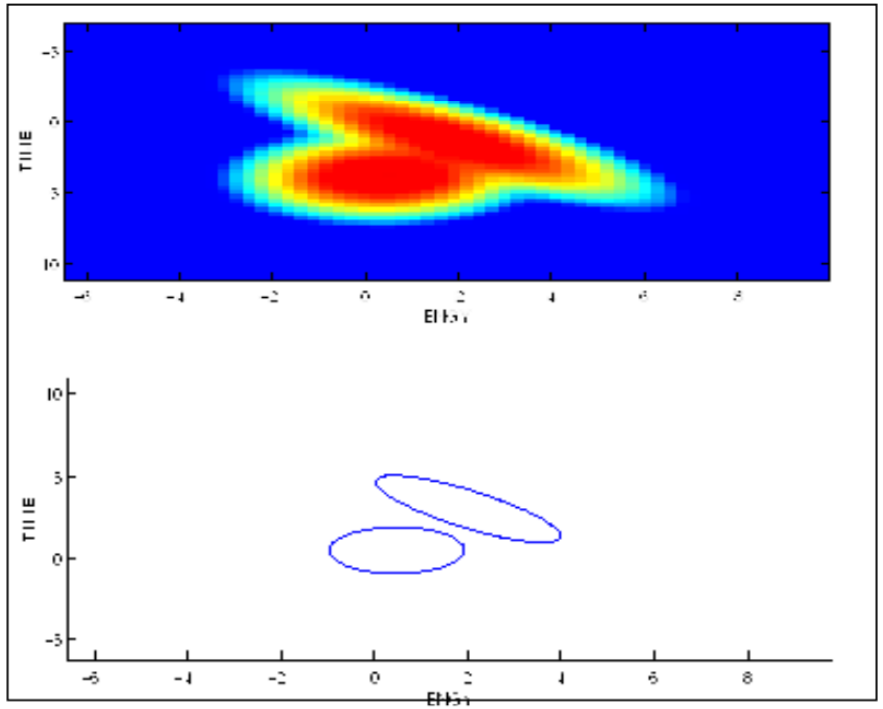
- Gaussian Mixture Model (GMM)

GMM: Introduction 1



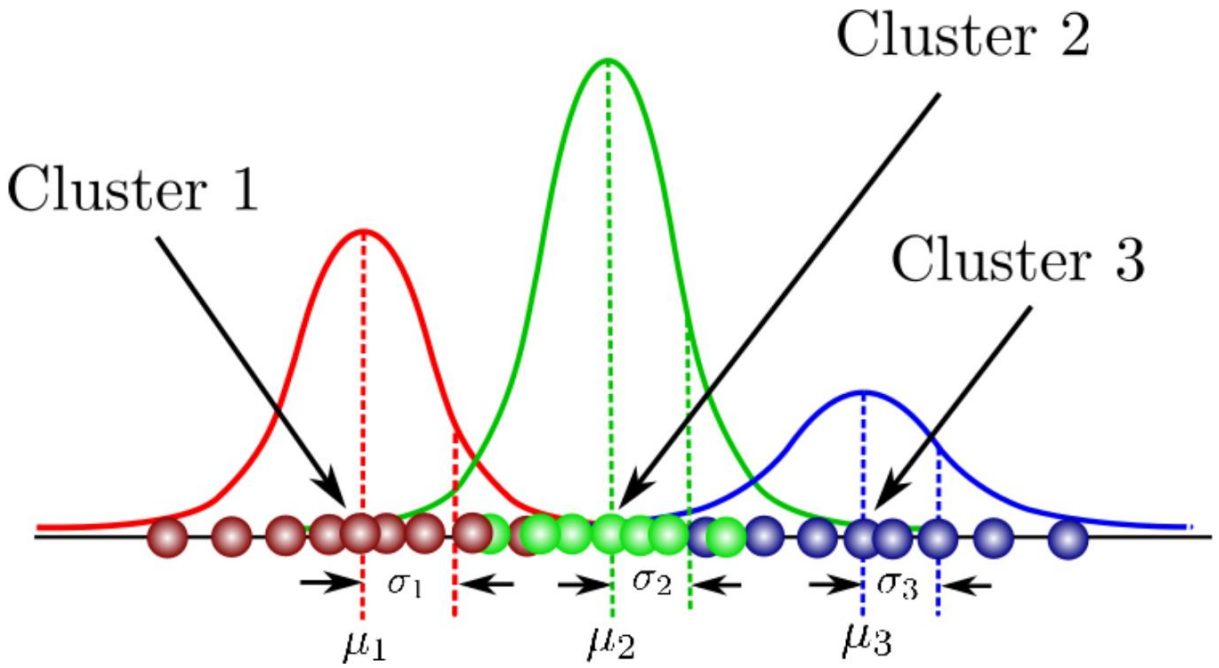
single Gaussian ? ? GMM





GMM: Introduction 2

- Function of several Gaussians $k \in \{1,\ldots, K\}$, where K is the number of clusters of our dataset



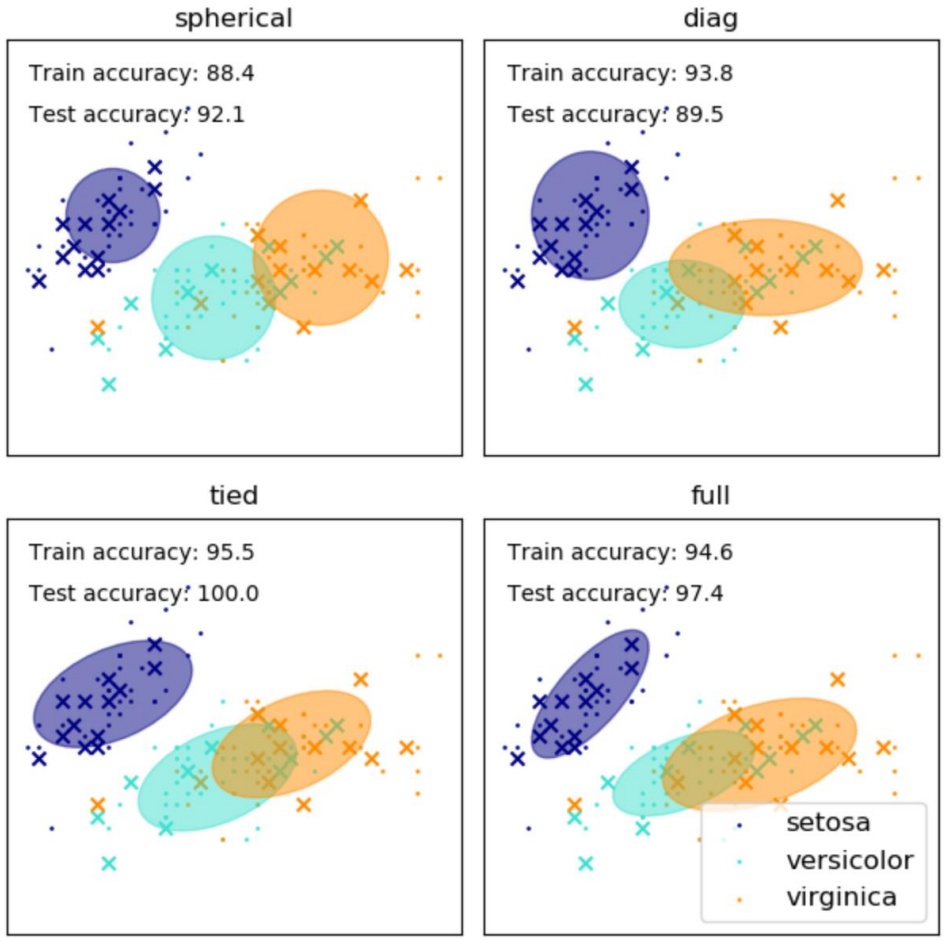
- Parameters :
  - means μ
  - covariances Σ that define the widths
  - mixing probabilities c that define how big the Gaussian will be

Covariances

- sklearn.mixture.GaussianMixture

```
# Try GMMs using different types of covariances.
estimators = {cov_type: GaussianMixture(n_components=n_classes,
              covariance_type=cov_type, max_iter=20, random_state=0)
              for cov_type in ['spherical', 'diag', 'tied', 'full']}
```

"diagonal" situation with circular contours



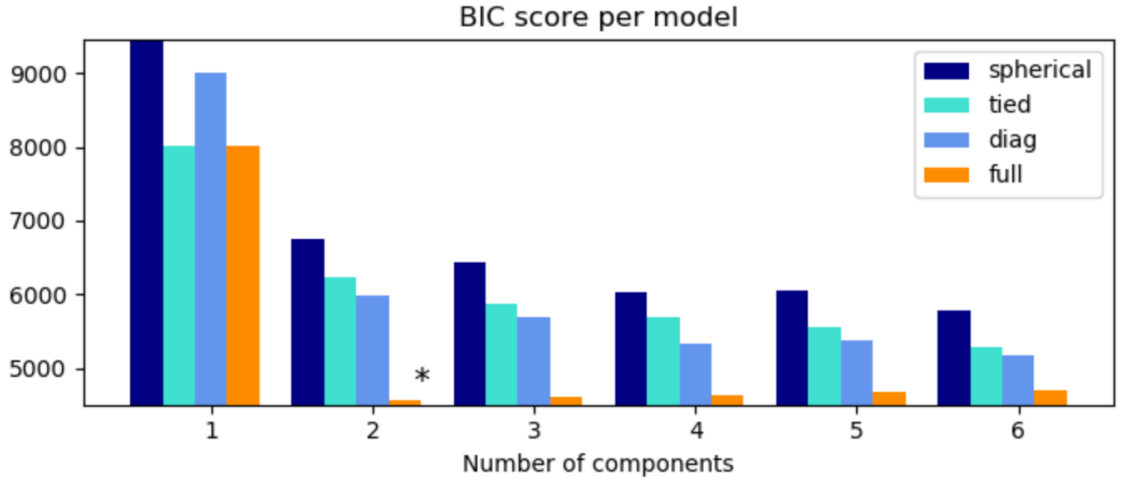
contour axes are oriented along the coordinate axes

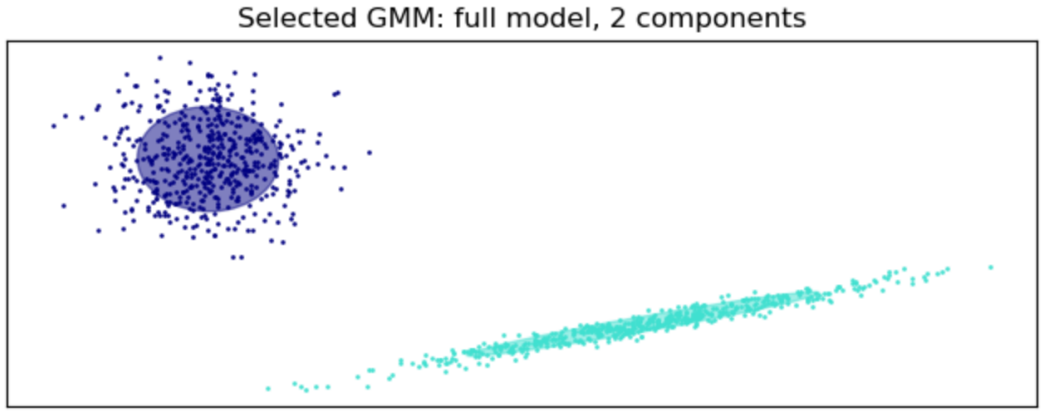
they have the same shape, but the shape may be anything

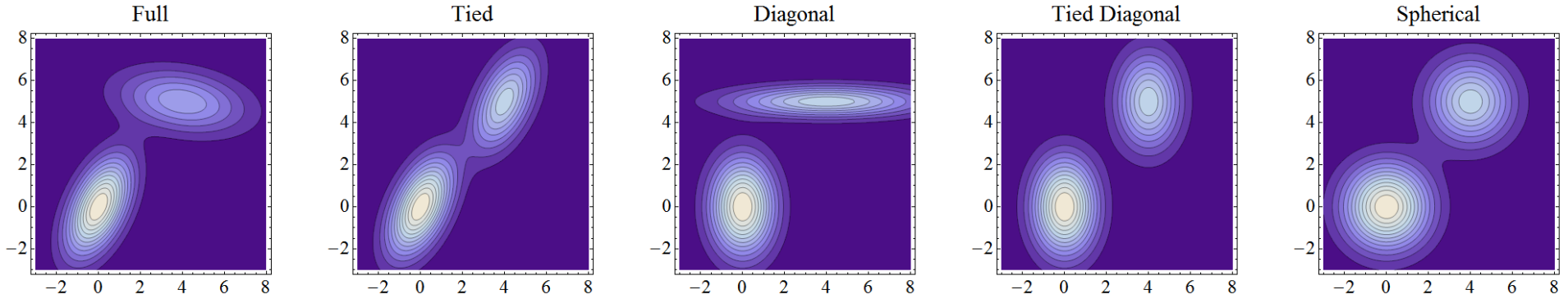
any position and shape

Number of Gaussian (K)

Bayesian Information Criterion (BIC)



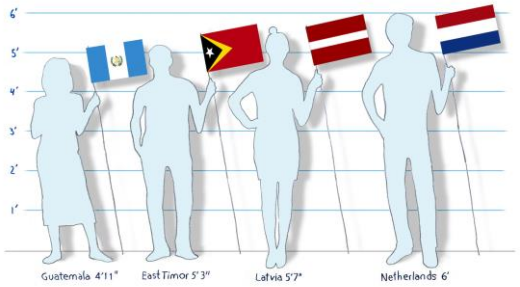




DISCUSSION

Applications

Demographic Statistics

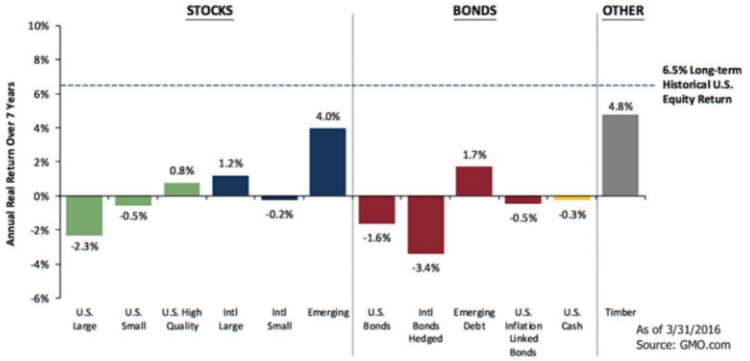


Goal : compute the average height of people of different ethnicities (African-American, Caucasian, Asian and Latino)

https://www.quora.com/What-is-an-example-of-real-world-application-of-Gaussian-Mixture-Models

Data : People's heights

Finance



Goal : forecast asset returns

Data : timeseries of asset prices

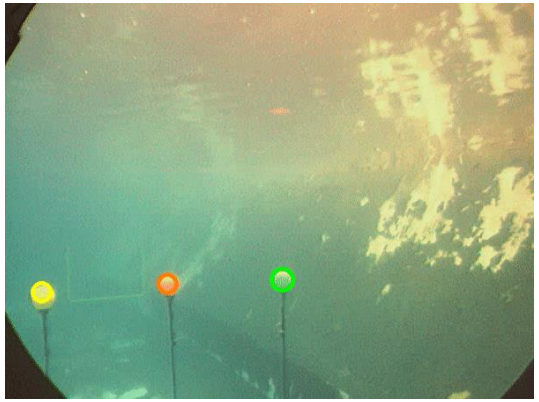
https://blog.quantinsti.com/mixture-models-for-forecasting-asset-returns/

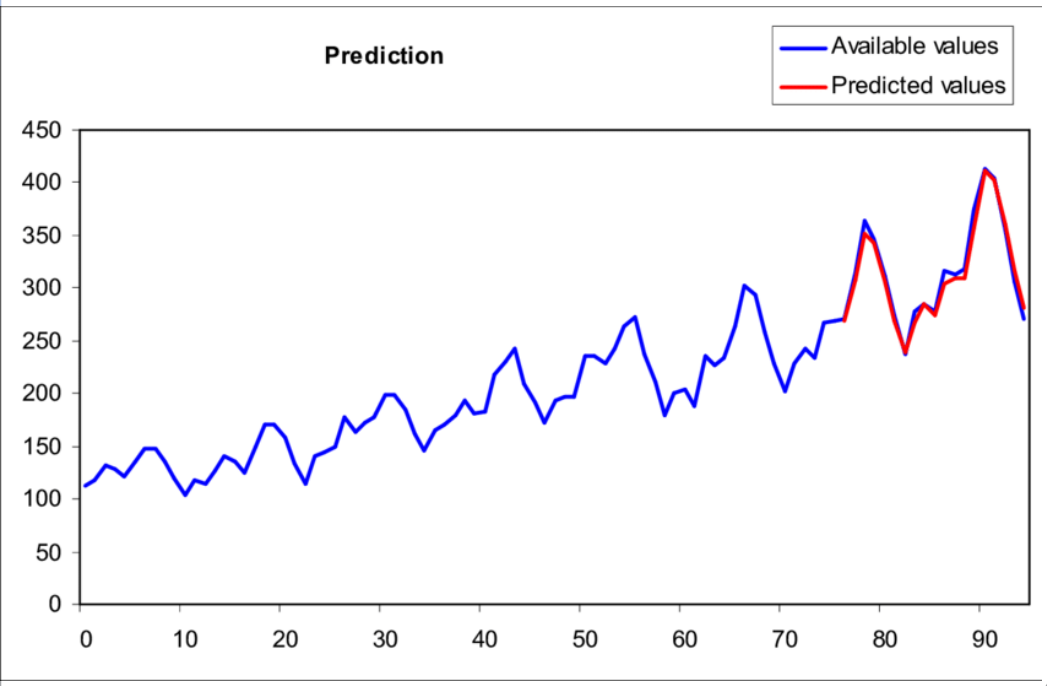
Detect objects in challenging environments

https://medium.com/lifeandtech/color-segmentation-using-gmm-7a8d900e48b

Goal : detect coloured buoys underwater

Data : video sequence





Sequence Prediction

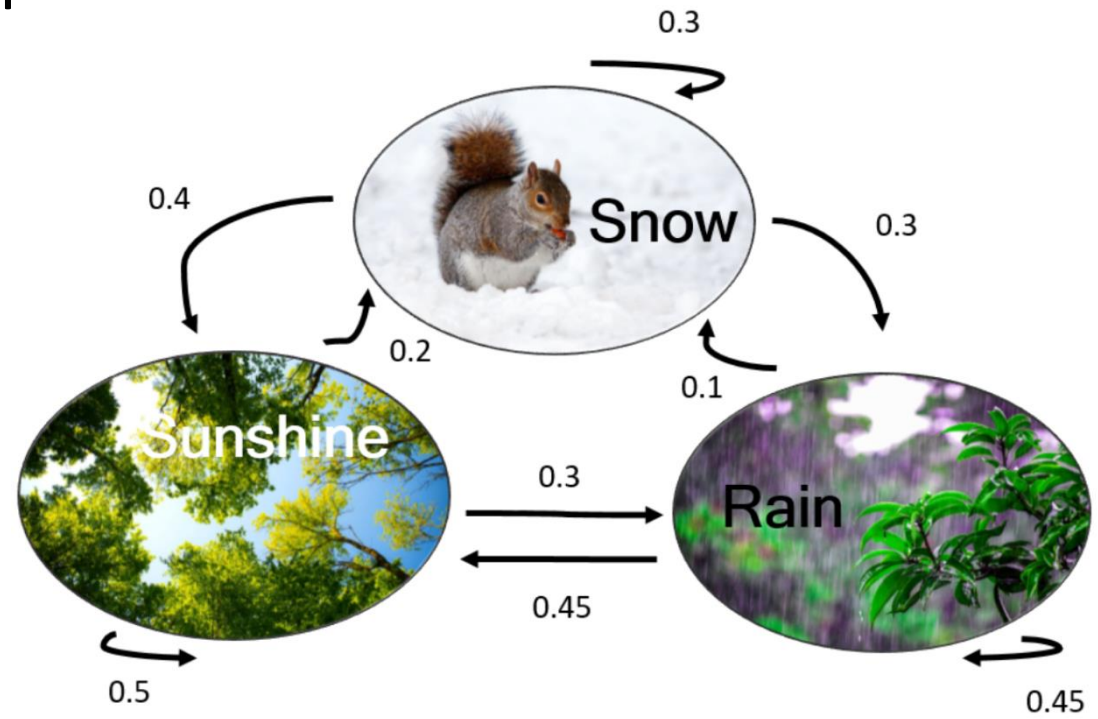
- Hidden Markov Model (HMM)

- Recursive Neural Network (RNN) *(module 2)*

- Long-Short Term Memory (LSTM) *(module 2)*

Hidden Markov Model (HMM)

- Markov Chain: random process, where the probability of being in state j depends *only* on the previous state



$$P = \begin{bmatrix} 0.3 & 0.3 & 0.4 \\ 0.1 & 0.45 & 0.45 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$$

- Invisible Markov chain : we cannot observe it. Each state generates a random one out of k observations

HMM Tasks

- 2 main tasks :

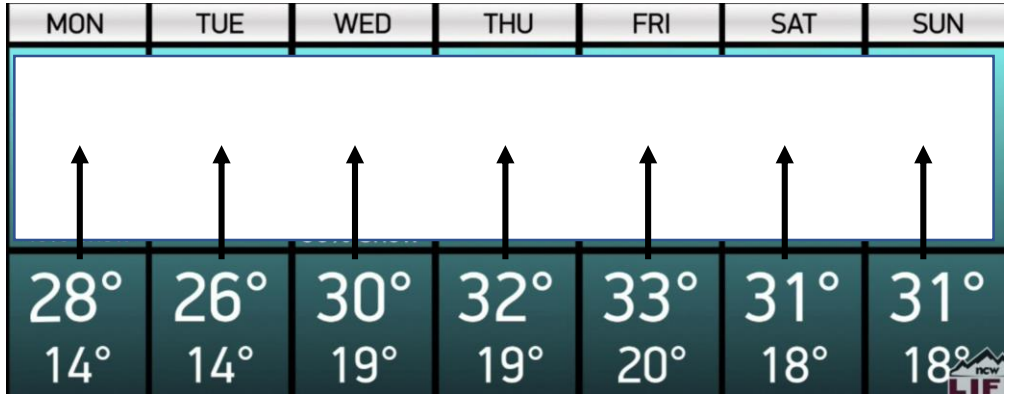
Task1 : Probability of an observation sequence (state estimation)

Calculate the probability that we feel cold for two consecutive days

Task 2 : Most likely explanation (inference/decoding)

- find the most probable hidden states from a series of observations → Viterbi algorithm

In a sequence of 7 days, we know if it was hot or cold. From it, we want to find the most likely weather forecast (snow, rain, sunshine) for this week



Applications

Word recognition (start p.10)

Goal : recognize a word by looking at the characters separately

Data : images of characters

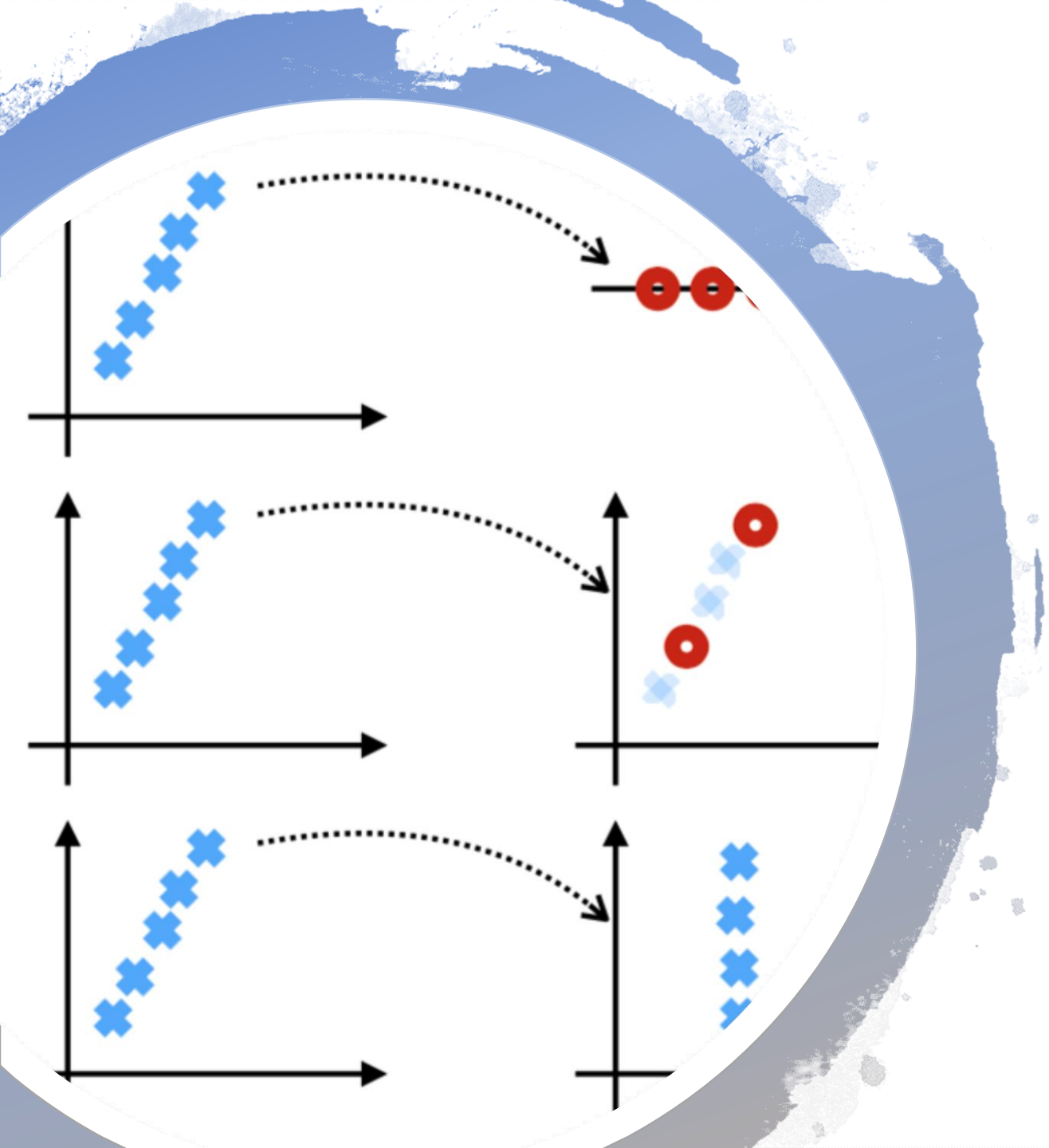
Stock trading

Goal : predict monthly closing prices of the S&P 500 index

Data : monthly prices

In practice

- No implementation anymore in sklearn
- https://pystruct.github.io/
- https://larsmans.github.io/seqlearn/
  - seqlearn.hmm.MultinomialHMM
- https://hmmlearn.readthedocs.io/en/latest/index.html



Feature Extraction

- PCA

- Kernel PCA

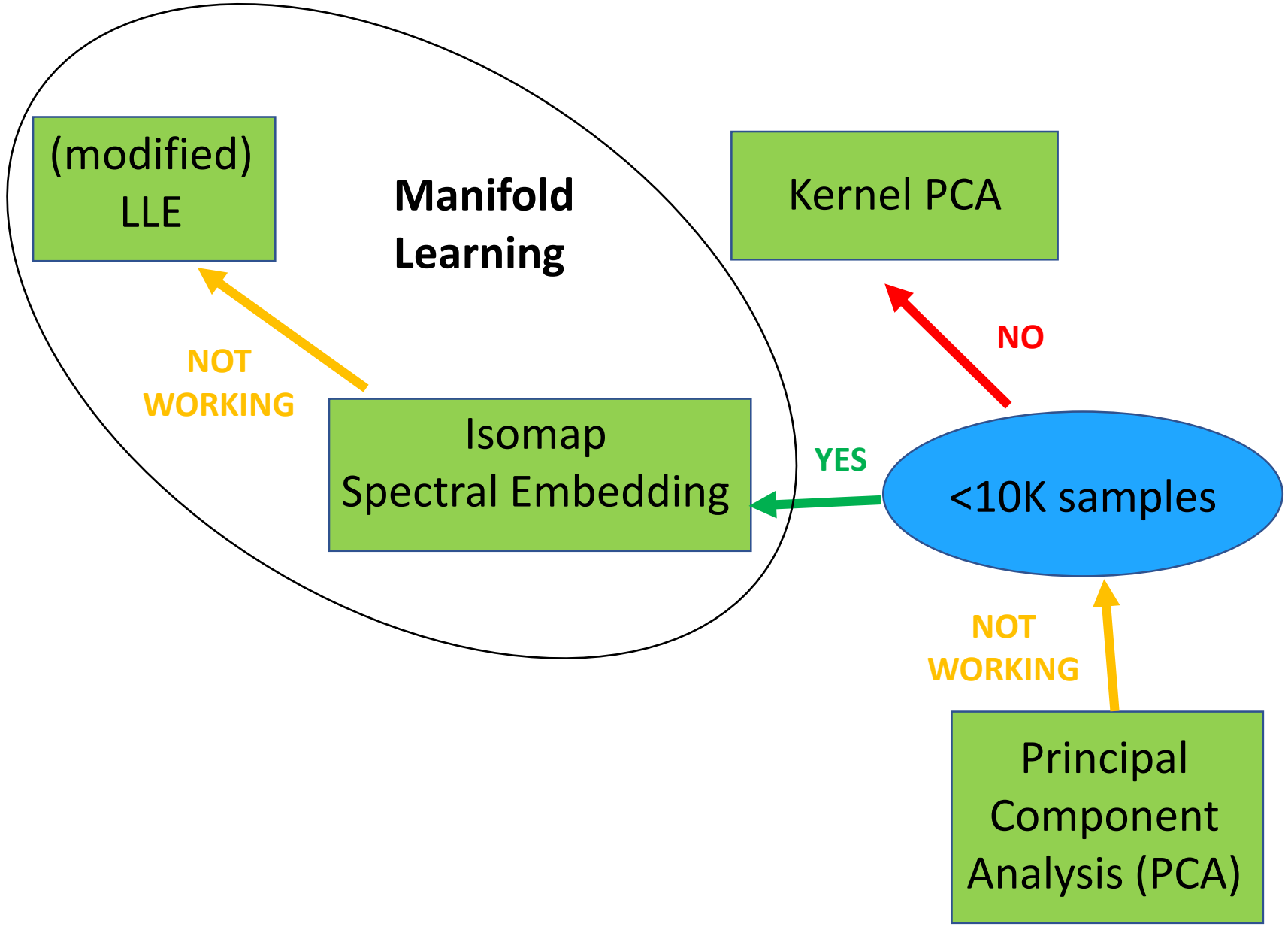
- Manifold Learning

The Curse of Dimensionality

- Many ML problems involve thousands of features for each training instance
  - Extremely slow, much harder to find a good solution

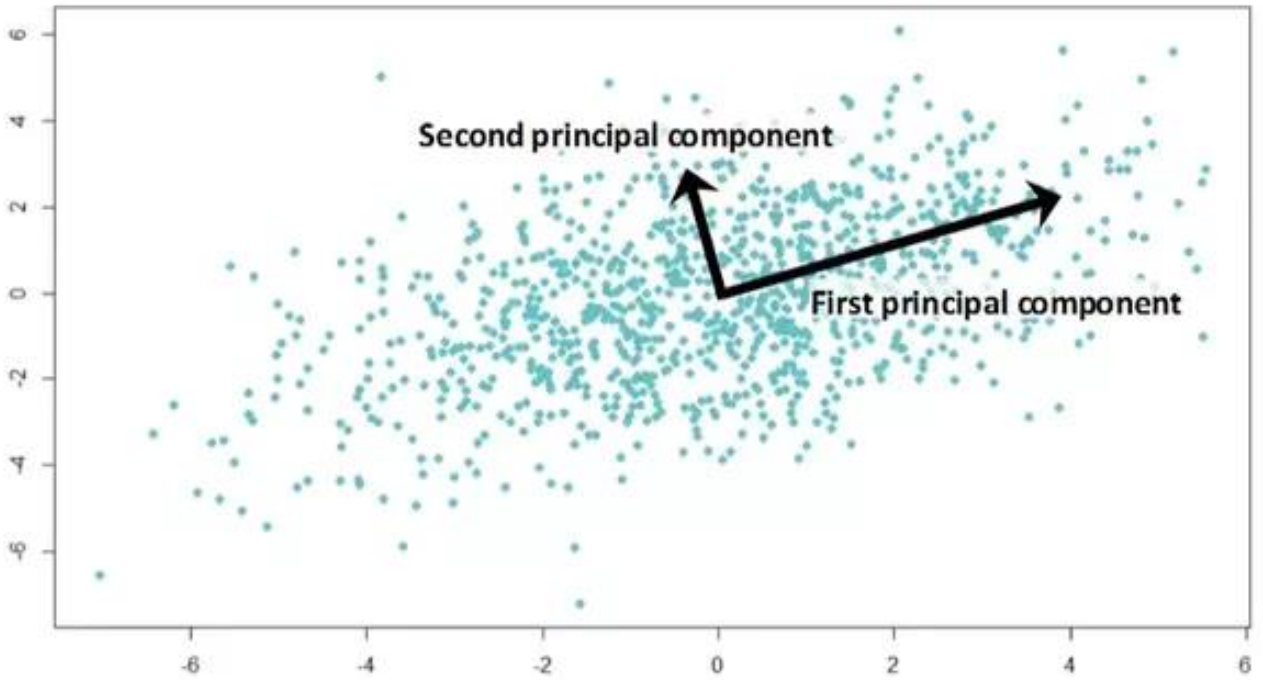
- Solutions :
  - Increase the size of the training set
    - Required size grows exponentially with number of dimensions
  - Reduce the number of features considerably
    - extremely useful for data visualization as well

Feature Extraction Algorithms



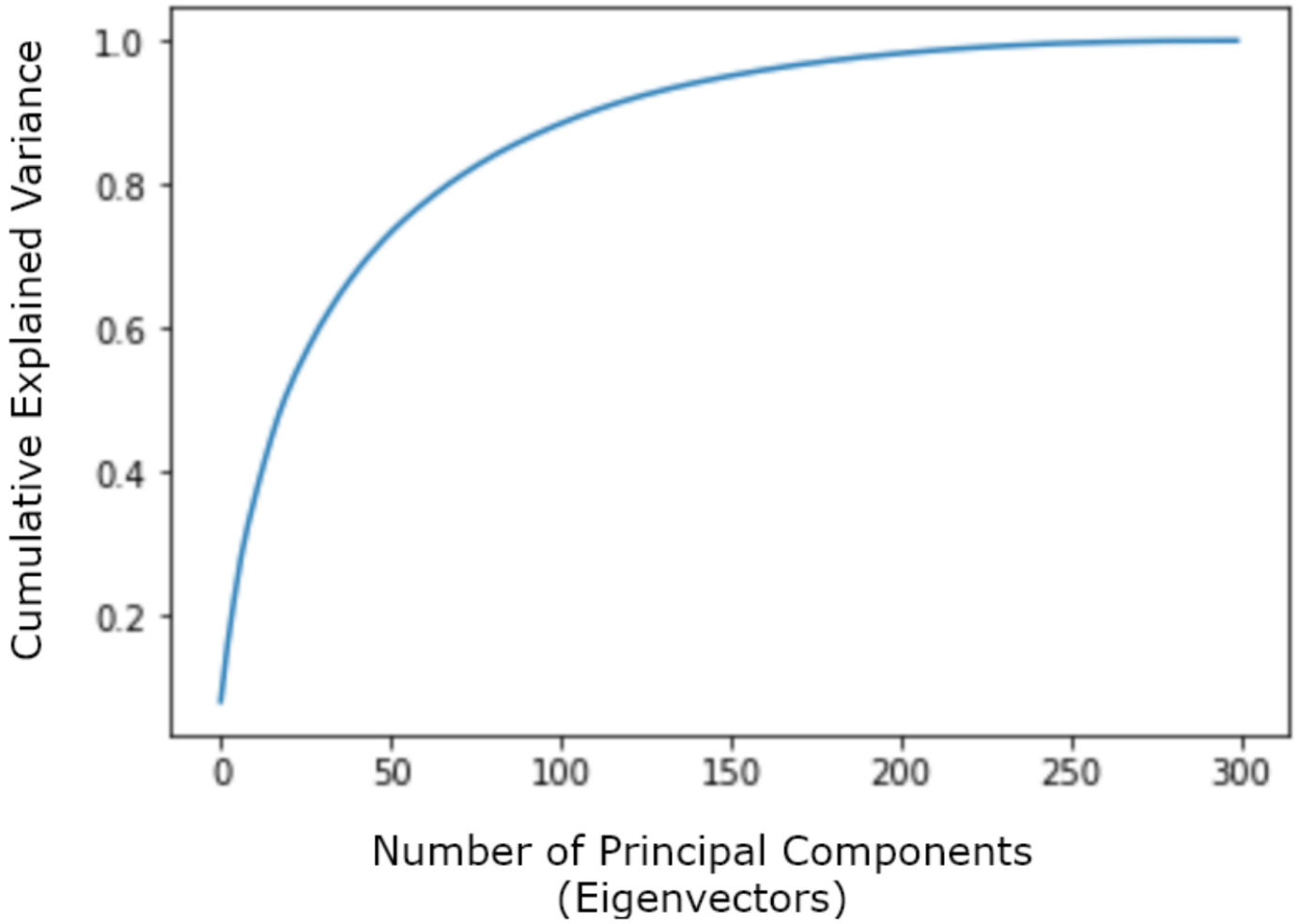
Principal Component Analysis (PCA)

- Find a new set of orthogonal dimensions and ranked according to the variance of data along them.
  - more important principal axis occurs first
  - the new dimensions allow to predict/reconstruct the original dimensions



- Stop criterium :
  - The reconstruction/projection error should be minimized

Number of components



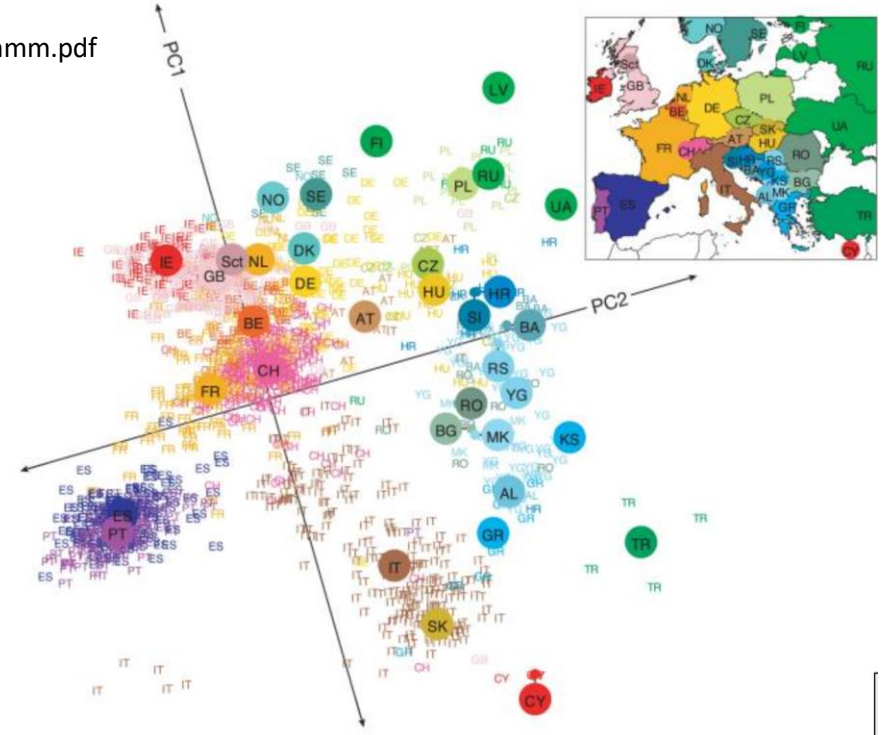
Applications

https://cse.buffalo.edu/~jcorso/t/CSE555/files/lecture_hmm.pdf

Genetic cartography

Goal : characterize genetic variation in European indiviuals

Data : Individual's genotypes



https://hackernoon.com/principal-component-analysis-unsupervised-learning-model-8f18c7683262

Movie recommendation

Goal : build a system that can make personalized movie recommendations to users

Data : historic user-community ratings

	Movie #1 Toy Story	Movie #2 Monsters Inc.	Movie #3 Saw	Movie #4 Ring	Movie #5 Hitch
User #1	4	5	1	NULL	4
User #2	5	NULL	1	1	NULL
User #3	5	4	3	NULL	3
User #4	5	4	1	1	NULL
User #5	5	5	NULL	NULL	3

In practice

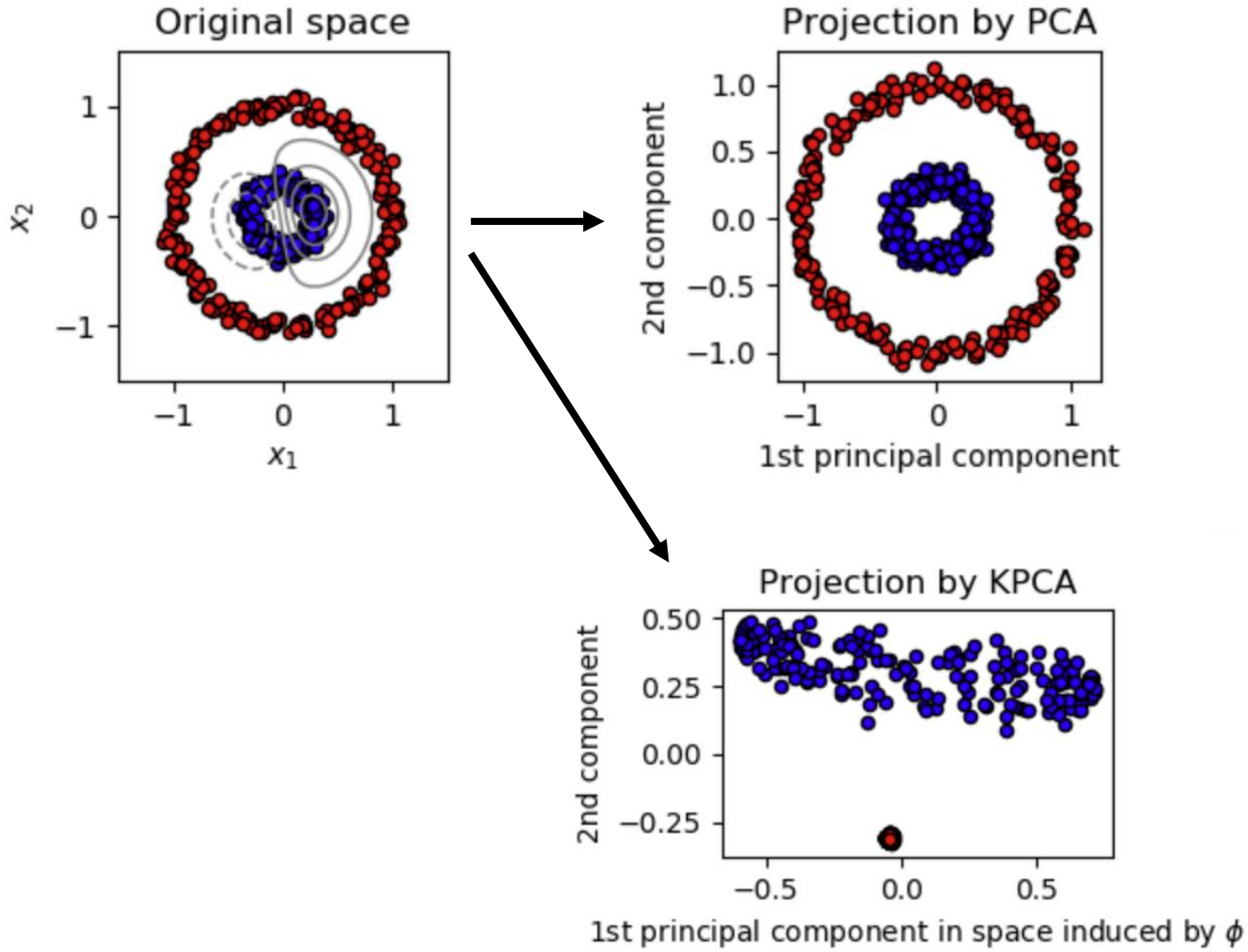
- sklearn.decomposition.PCA

```
>>> import numpy as np
>>> from sklearn.decomposition import PCA
>>> X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
>>> pca = PCA(n_components=2)
>>> pca.fit(X)
PCA(n_components=2)
>>> print(pca.explained_variance_ratio_)
[0.9924... 0.0075...]
>>> print(pca.singular_values_)
[6.30061... 0.54980...]
```

- sklearn.decomposition.FactorAnalysis for mixed data containing categorical data

Kernel PCA

- Non-linear extension of PCA
- Based on kernel (seen for SVC)



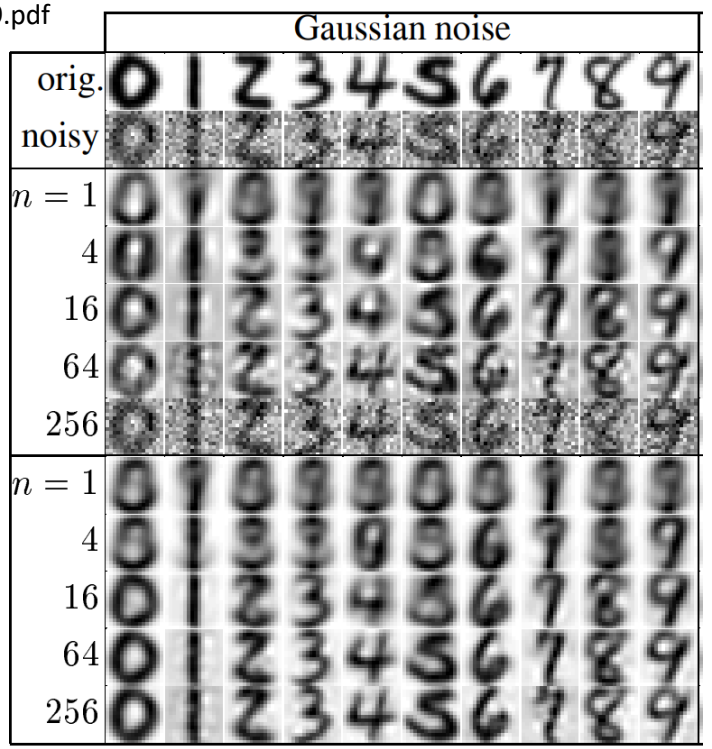
Applications

Image denoising

Goal : denoise images

Data : 256-dim handwritten digits

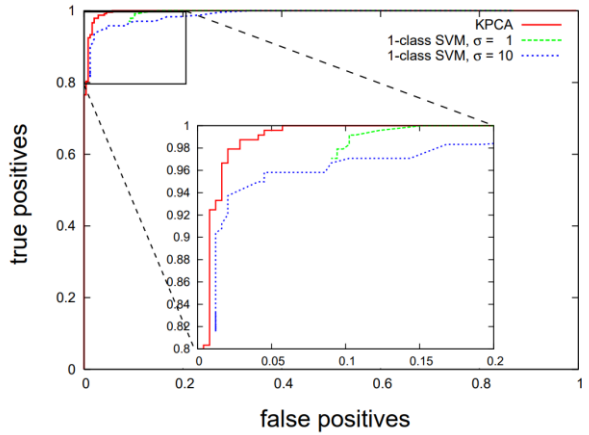
https://alex.smola.org/papers/1999/MikSchSmoMuletal99.pdf



Novelty Detection

Goal : detect novel patterns in new data

Data : breast-cancer data



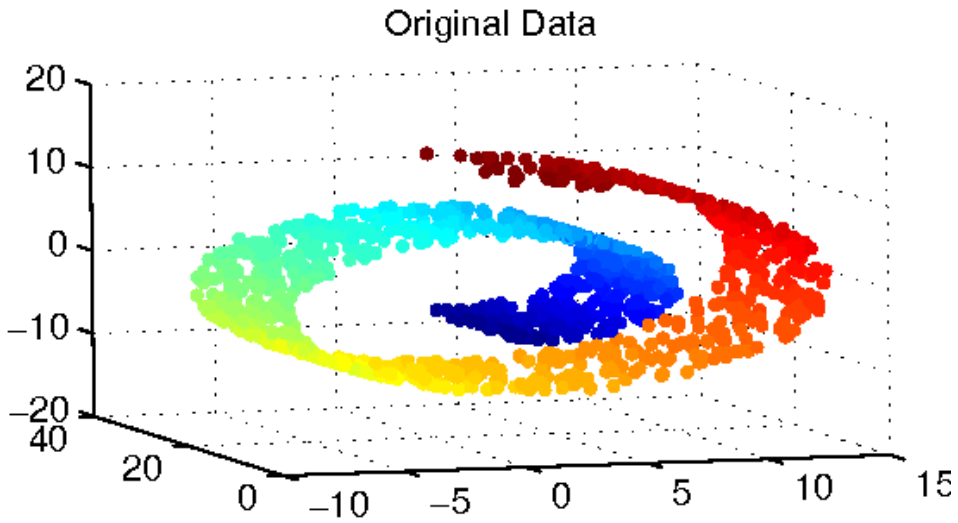
https://www.researchgate.net/publication/222828640_Kernel_PCA_for_novelty_detection

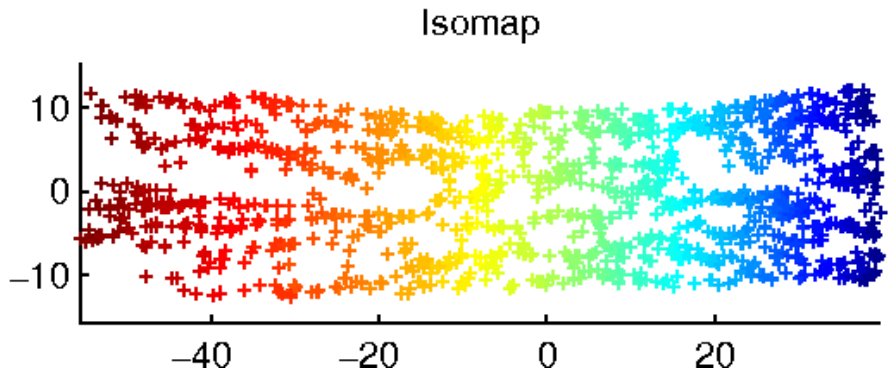
In Practice

```
>>> from sklearn.datasets import load_digits
>>> from sklearn.decomposition import KernelPCA
>>> X, _ = load_digits(return_X_y=True)
>>> transformer = KernelPCA(n_components=7, kernel='linear')
>>> X_transformed = transformer.fit_transform(X)
>>> X_transformed.shape
(1797, 7)
```

Manifold Learning : Introduction

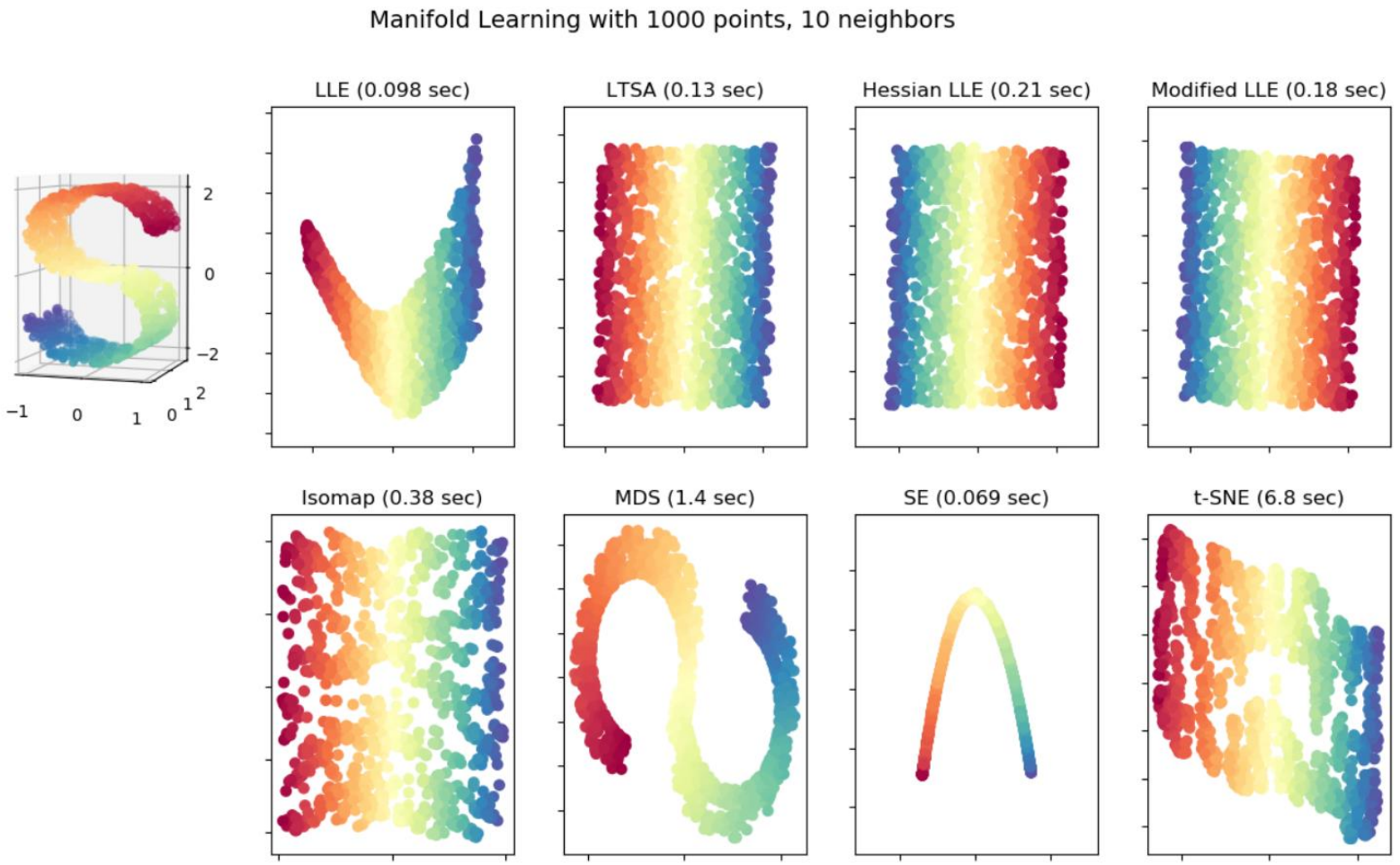
- Non-linear method to visualize the structure of a high-dimensional dataset by reducing its dimension.





- Use cases :
  - Clustering
  - Dimensionality Reduction

Algorithms



- Locally Linear Embedding (LLE), Modified LLE (MLLE), Hessian LLE
- Local Tangent Space Alignment (LTSA)
- Multi-Dimensional Scaling (MDS), Isomap (extension of MDS)
- Spectral Embedding (SE)
- t-distributed Stochastic Neighbor Embedding (t-SNE)

Quiz

https://b.socrative.com/login/student/

Room : CONTI6128