UNIVERSITY OF TWENTE.

BIG GEODATA AND MACHINE LEARNING

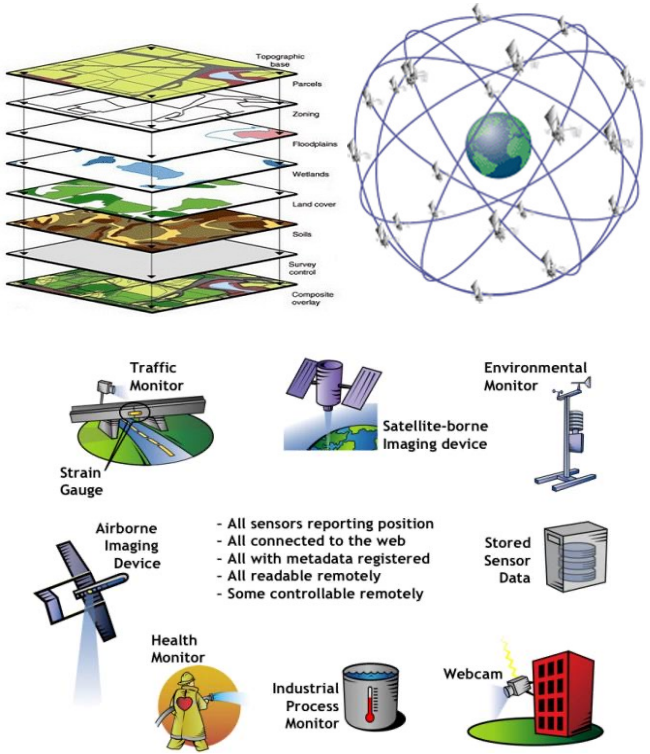
Mahdi KHODADADZADEH

February 2022

ITC FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

GEODATA

- Geospatial data are related to Earth and give geographically referenced information.
- Structured / traditional:
  - Raster: satellite images, model outcomes (e.g. climate simulations), drone images, etc.
  - Vector: trajectory data (e.g. Uber data), geo-located twitter data, etc.
  - Graph: road network data, supply chain network data, etc.
- Unstructured/ novel:
  - UGGC, VGI, etc.



BIG GEODATA

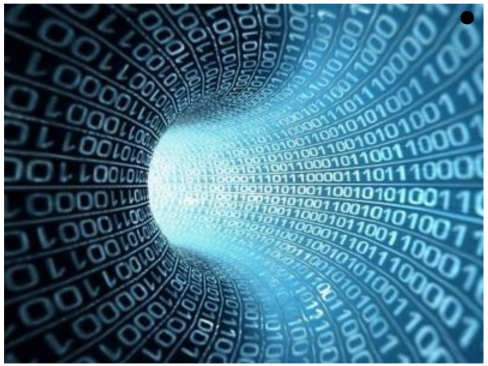
- Large amounts of spatio-temporal data becoming available both to the scientific community and to the general public.

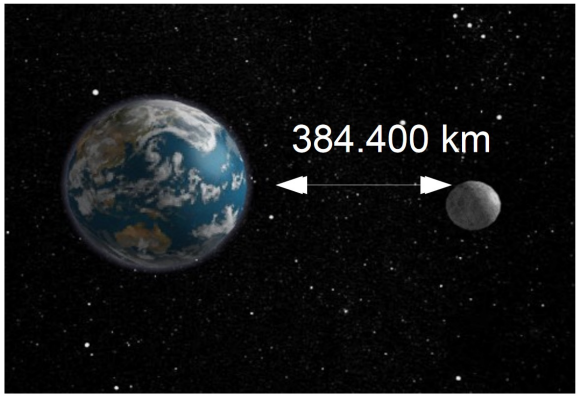
"Data is arguably the most important natural resource of this century… Big data is big news just about everything you go these days. Here in Texas everything is big so we just call it data"

Michael Dell, 2014

BIG GEODATA

- Every day we create 2.5 trillion (10^{18}) bytes of data. 80 % of these are already georeferenced or can be.
- It's a huge dataset, equal to a DVD tower that goes from the Earth to the Moon every day.





M. A. Brovelli,

Politecnico Milano

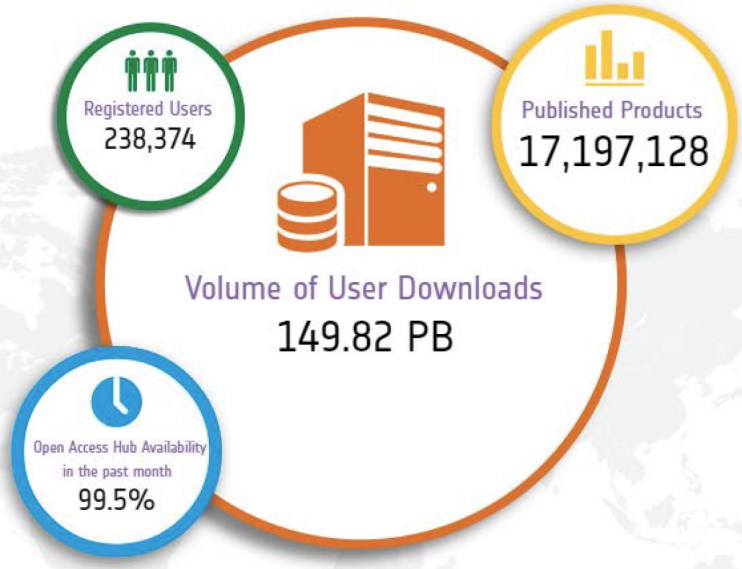
COPERNICUS -- THE SENTINELS

	**Sentinel 1** (A/B/C/D) **SAR Imaging**	All weather, day/night applications, interferometry
	**Sentinel 2** (A/B/C/D) **Multispectral Imaging**	Land applications: urban, forest, agriculture, … Continuity of Landsat, SPOT
	**Sentinel 3** (A/B/C/D) **Ocean & Global Land Monitoring**	Wide-swath ocean colour, vegetation, sea/land surface temperature, altimetry
	**Sentinel 4** (A/B) **Geostationary Atmospheric**	Atmospheric composition monitoring, pollution; instrument on MTG satellites
	**Sentinel 5** (A/B/C) & Precursor **Low-Orbit Atmospheric**	Atmospheric composition monitoring; instrument on MetOp-SG satellites
	**Sentinel 6** **Jason CS (A/B)**	Altimetry reference mission

FREE EO DATA AND INFORMATION SERVICES

Registered Sentinel Users

The real number of users is much higher but unknown due to the free, full & open data policy.



Statistics on
28 Jun 2019

Sentinel Scientific Data Hub

ITC

UNIVERSITY OF TWENTE.

BIG DATA – V IS FOR …

- **Volume:** define Big Data as “Big”
- **Velocity:** data needs to be accessed and processed fast
- **Variety:** structured vs unstructured data sources. One of the biggest challenges of big data
- **Veracity:** trustworthiness of the data
- **Variability:** understanding and interpreting the correct meanings of raw data (which depends on its context).
- **Visualization** data presentation (for decision-making).
- **Value:** Big Data can provide businesses with immense value. Raw data is worthless. Need to integrate and analyze it.

BIG DATA – V IS FOR …

The 42 V's of Big Data and Data Science

Tags: 3Vs of Big Data, Humor

https://www.kdnuggets.com/2017/04/42-vs-big-data-data-science.html

BIG DATA – TOWARDS A DEFINITION

- Big Data refers to **large** and **complex** data sets that are difficult to handle using traditional systems and methods to analyse and extract information.

- Big data usually includes datasets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process data within a tolerable elapsed time
- Big data represents the information assets characterized by such a high volume, velocity and variety to require specific technology and analytical methods for its transformation into value
- Big data is where parallel computing tools are needed to handle data

https://en.wikipedia.org/wiki/Big_data

BIG DATA – ILL-DEFINED TERM

Dataset type	Size range	Fits in RAM?	Fits on local disk?
Small dataset	Less than 2–4 GB	Yes	Yes
Medium dataset	Less than 2 TB	No	Yes
Large dataset	Greater than 2 TB	No	No

Boundaries are fluid, depending on the technology

Jassie C. Daniel. Data Science with Python and Dask. Manning. 2019

BIG GEODATA CHALLENGE

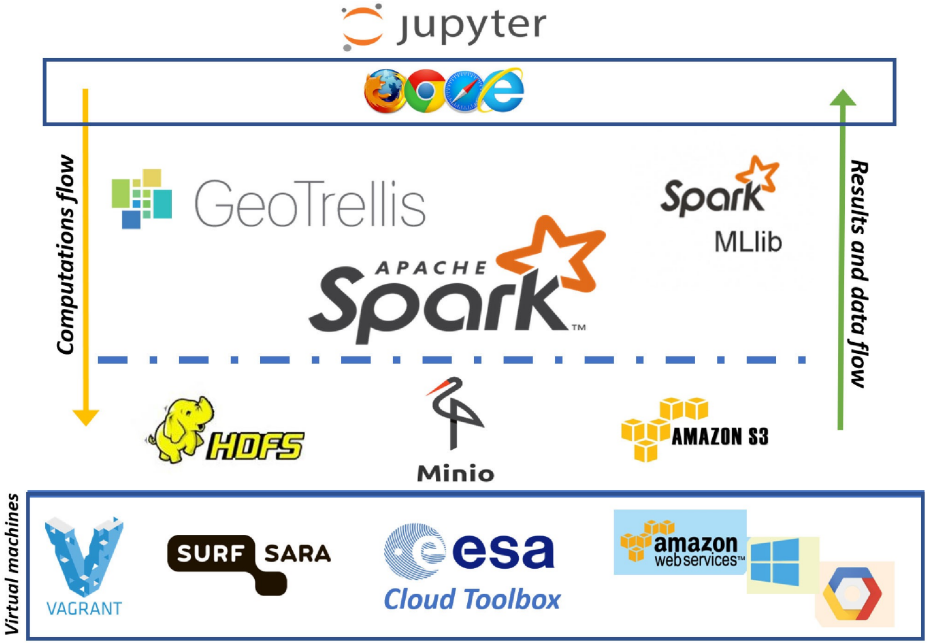
- Data processing and analysis tasks are still mostly performed on **local** workstations and is **time consuming**.
- The real challenge in normal circumstances is to develop **innovative** ways to deal with all the available data and datasets.

The tsunami of data

BIG GEODATA SOLUTIONS

- Numerous spatial computing methods and systems have been developed to tackle the difficulties and **enable** discovery, delivery, analysis, and visualisation of **big geospatial data**.
- Recent developments in both **hardware and software** infrastructure has given big push and new direction to distributed data processing capabilities.
- **Scalable and affordable** big data analysis capabilities are available through:
  - Open-source systems that allow computing clusters on commodity hardware
  - Proprietary cloud-based data storage and computing services

BIG GEODATA SOLUTIONS



From: R. Zurita-Milla, R. Goncalves, E. Izquierdo-Verdiguier and F. O. Ostermann, "Exploring Spring Onset at Continental Scales: Mapping Phenoregions and Correlating Temperature and Satellite-Based Phenometrics," in *IEEE Transactions on Big Data*, vol. 6, no. 3, pp. 583-593, 1 Sept. 2020

GOOGLE EARTH ENGINE

- Google Earth Engine combines a multi-petabyte catalog of satellite imagery and geospatial datasets with planetary-scale analysis capabilities available for free

  https://earthengine.google.com/

WHAT IS THE RIGHT SOLUTION?

- Depending on the nature of spatial big data and the analysis needs
- Not all analysis needs require big data technology!
  - Analyses that can be done **faster by parallel computing** on *a workstation* (e.g. continental scale studies with medium-size data)
  - Analyses **requiring special processing units** (e.g. GPU) due to computational complexity (e.g. machine and deep learning)
  - Analyses **requiring distributed computing** on *a cluster* due to large volume of data or high computational complexity (e.g. global-scale studies with big data, e.g. petabyte scale)

YOU NEED TO KNOW…

- How to use existing methods and tools more efficiently to avoid distributed computing (e.g. data engineering, proper sampling strategy, indexing)
- How to perform parallel computing on a workstation (i.e. parallelism with multi-cores)
- How to perform parallel computing with specialized processing units (e.g., GPUs)
- How to perform out-of-core computing on a computing cluster (e.g., distributed data management, distributed computing)
- How to use large-scale cloud computing infrastructure (i.e. setup, scaling, cost effectiveness)

QUESTIONS?

BREAK?

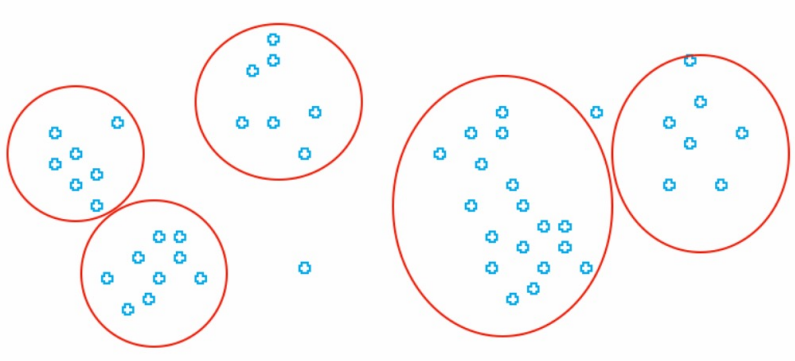
UNSUPERVISED VS SUPERVISED LEARNING

- Unsupervised learning (clustering)
  - Class labels unknown
  - Checking for groups/clusters in the data

- Supervised learning (classification & regression)
  - Supervision: training data
  - Classif./Regres. of unseen/unlabeled data

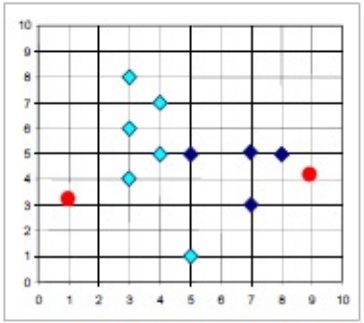
CLUSTERING

- An important task in data mining that aims at identifying groups of elements that are similar among themselves but dissimilar to the elements in other groups.
- It provides a high-level abstraction of the data, which facilitates the extraction of useful information

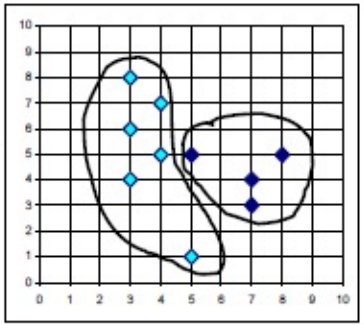


K-MEANS

1. Initialization: arbitrary initialization of the centers

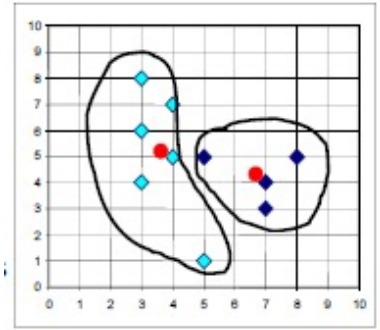


2. Data assignment. Each point is assigned to its closest cluster (center). Ties are broken by randomly assigning the point to one of the clusters. This yields a partitioning of the data.

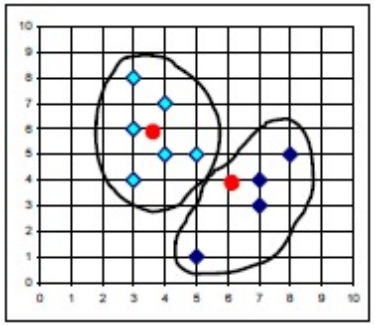


K-MEANS

3. Relocating “centers”. Each cluster representative is moved to the center (arithmetic mean) of the points assigned to it



4. Repeat 2 and 3 until the centers do not longer change

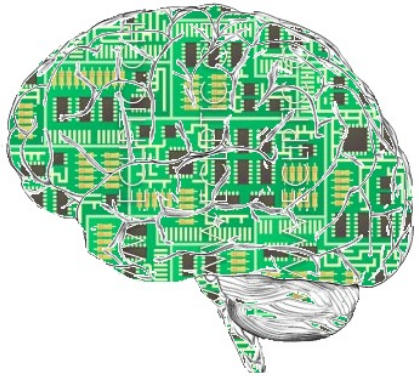


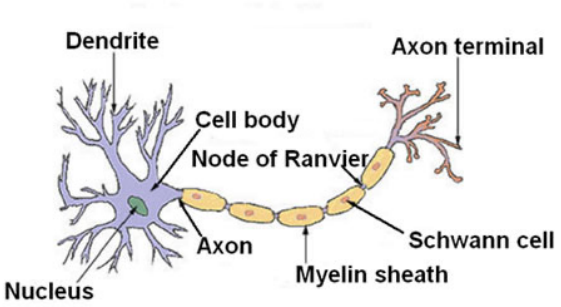
*Each iteration requires N*k comparisons. it takes long time for large datasets*

ARTIFICIAL NEURAL NETWORKS (ANNS)

ANNs are models designed to imitate the human brain through the use of mathematical models.

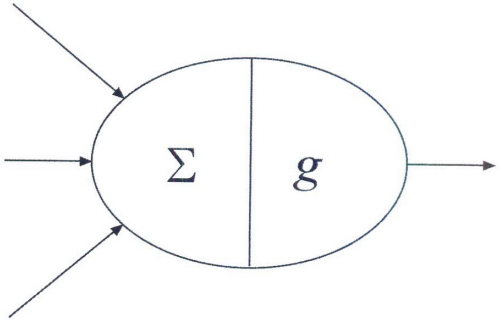
Inspired by the central nervous system and the neurons (and their axons, dendrites and synapses)

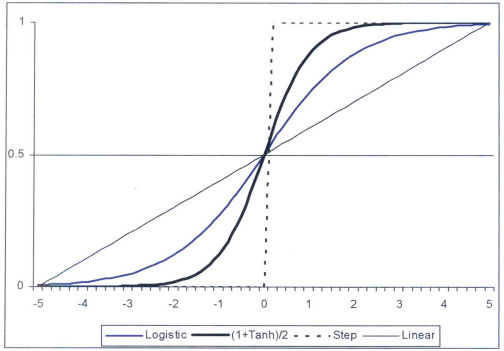




GENERAL NEURON

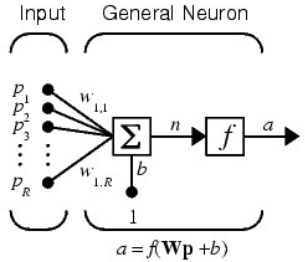
- Input data are weighted, then added up and finally transformed using an activation function
- Several activation functions are possible

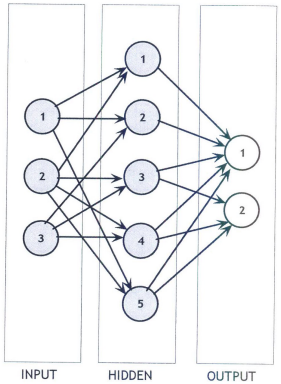


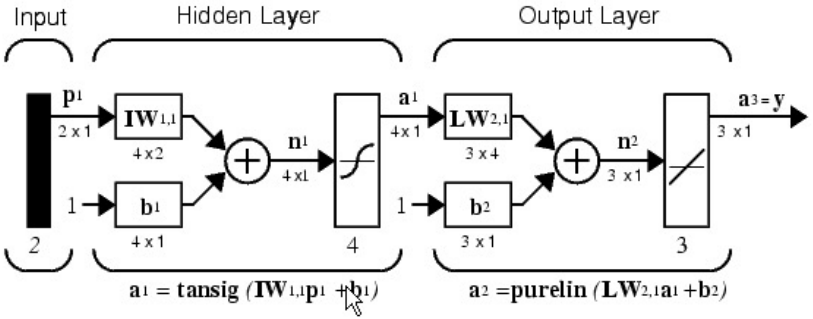


FEEDFORWARD NETWORKS

- Input and output data available
- Change their weights and are able to learn how to solve a certain problem
- Training is done iteratively





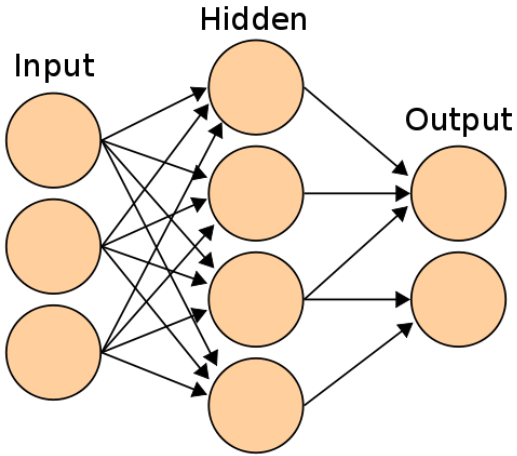


TRAINING / LEARNING

Backpropagation training algorithm:

Forward phase – the inputs are presented to the network

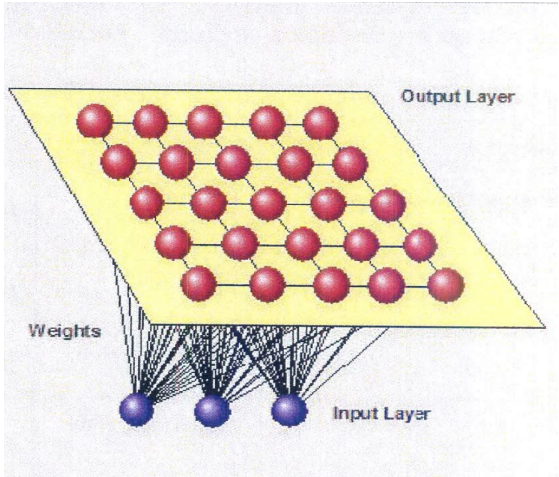
Backpropagation phase – the outputs are compared with the targets and the **weights** are adjusted



SELF-ORGANIZING MAPS

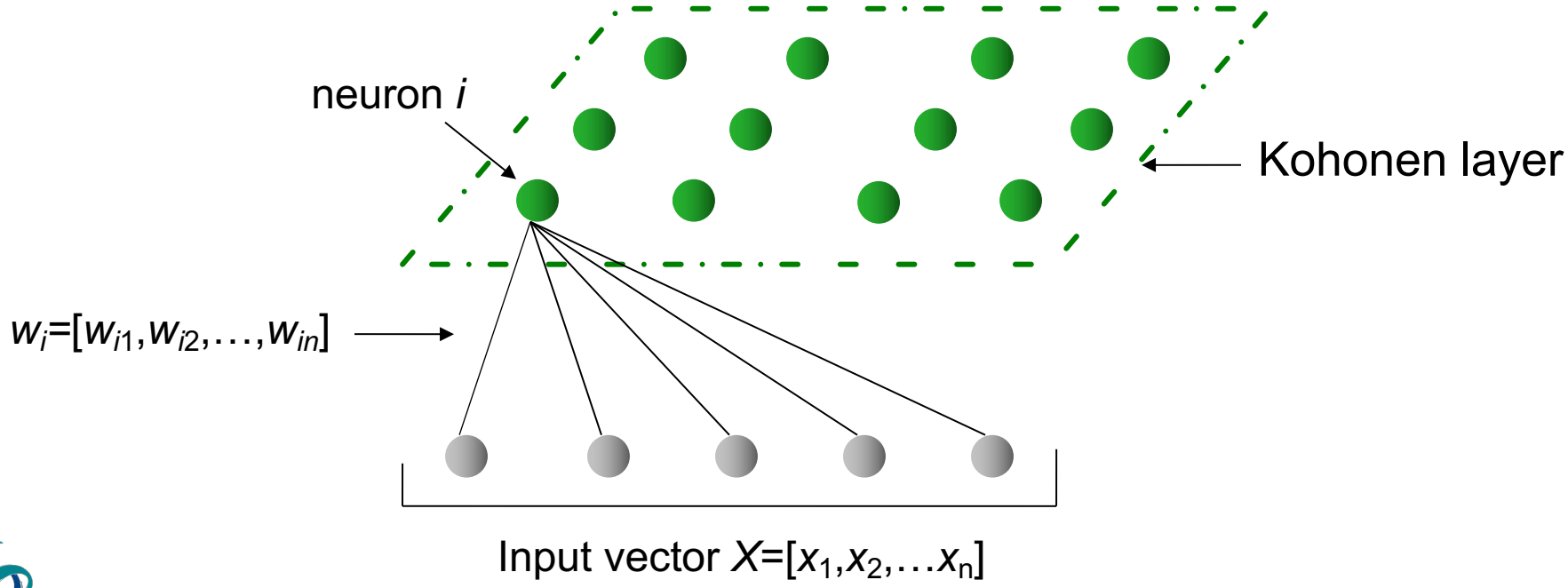
- Developed and formalized in 1992 by Teuvo Kohonen
- Clustering and visualizing high dimensional data
- Detecting patterns in multidimensional data and representing them in much lower dimensional spaces – usually one or two dimensions

- Neural networks try to figure out patterns in the data on their own



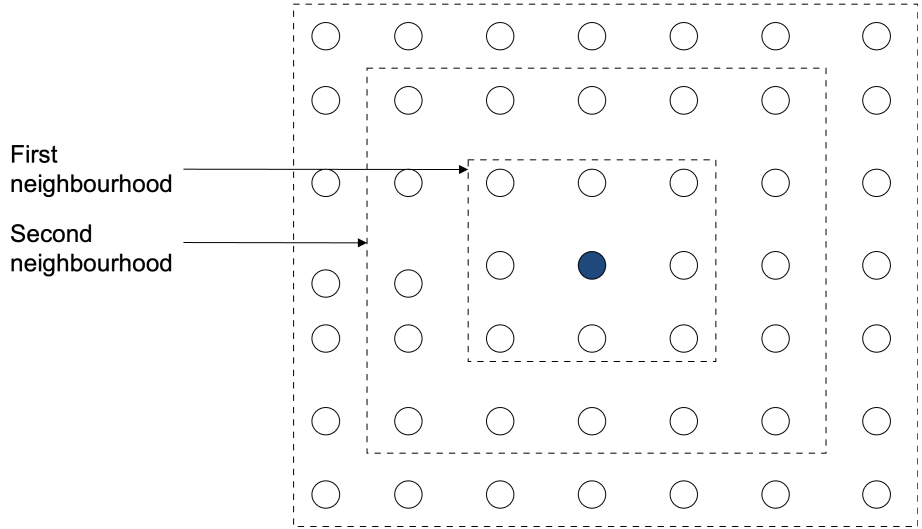
SELF-ORGANIZING MAPS

- The input is connected with each neuron of a lattice (map)



SELF-ORGANIZING MAPS

- Hypothesis: The model self-organizes based on learning rules and interactions
- Goal: Find weight values such that adjacent neurons have similar values
- Neurons maintain proximity relationships as they get updated



SELF-ORGANIZING MAPS: ALGORITHM

- 1) The weights are initialized to random values
- 2) a m-dimensional input vector Xs enters the network;
- 3) The distances di(Wi, Xs) between all the weight vectors on the SOM and Xs are calculated by using (for instance):

$$d_i(W_i, X_s) = \sum_{j=1}^{m} \left(w_j - x_j\right)^2$$

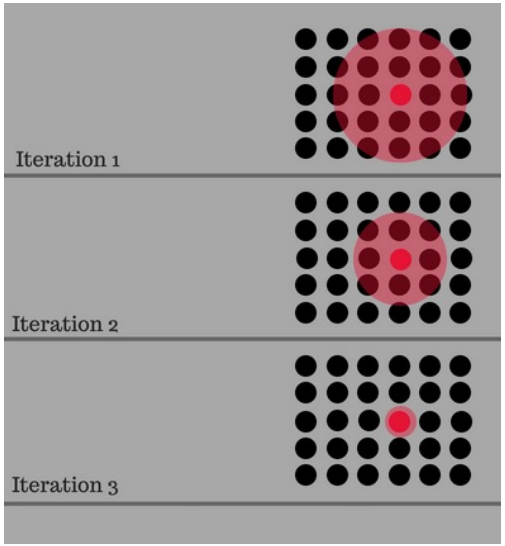
  - Wi denotes the ith weight vector;
  - wj and xj represent the jth elements of Wi and Xi respectively

SELF-ORGANIZING MAPS: ALGORITHM

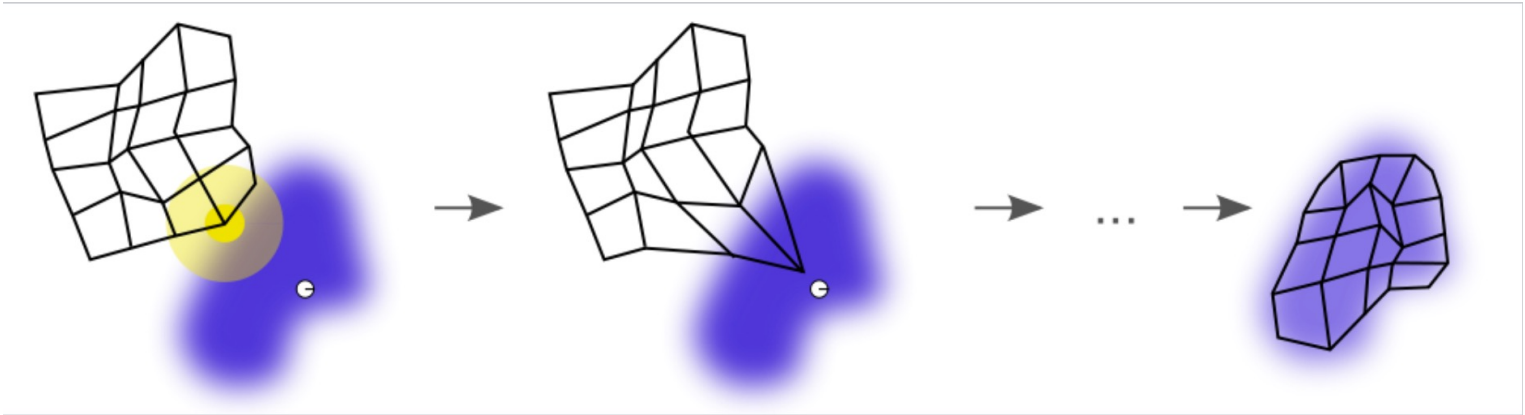
- 4) Find the best matching neuron or “winning” neuron whose weight vector W_k is closest to the current input vector X_i ;
- 5) Modify the weights of the winning neuron and all the neurons in the neighbourhood N_k by applying:
  - $W_{jnew} = W_{jold} + \alpha(X_i - W_{jold})$
- Where α represents the learning rate;
- 6) Next input vector $X_{(i+1)}$, the process is repeated.

→ Input are assigned to neurons that are similar to them

→ Basically, each neuron becomes the center of a cluster



SELF-ORGANIZING MAPS: ALGORITHM



An illustration of the training of a self-organizing map. The blue blob is the distribution of the training data, and the small white disc is the current training datum drawn from that distribution. At first (left) the SOM nodes are arbitrarily positioned in the data space. The node (highlighted in yellow) which is nearest to the training datum is selected. It is moved towards the training datum, as (to a lesser extent) are its neighbors on the grid. After many iterations the grid tends to approximate the data distribution (right).

Wikipedia ☺

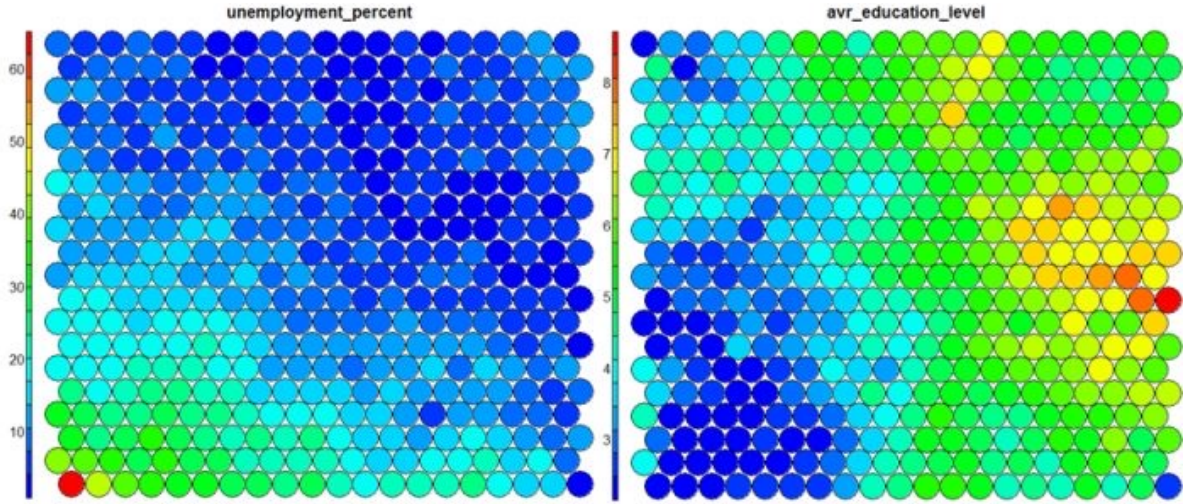
SELF-ORGANIZING MAPS: VISUALIZATION

- SOM visualization are made up of multiple “nodes”. Each node vector has:
  - A fixed position on the SOM grid
  - A weight vector of the same dimension as the input space. (e.g. if your input data represented people, it may have variables “age”, “sex”, “height” and “weight”, each node on the grid will also have values for these variables)
  - Associated samples from the input data. Each sample in the input space is “mapped” or “linked” to a node on the map grid. One node can represent several input samples.
- Similar input samples are placed close together on the SOM grid!

https://www.shanelynn.ie/self-organising-maps-for-customer-segmentation-using-r/

SELF-ORGANIZING MAPS: VISUALIZATION

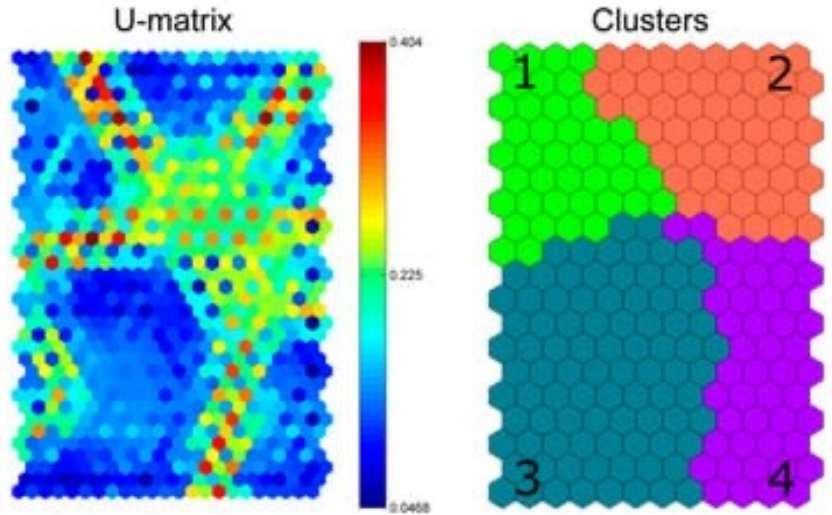
- Typical SOM visualisations are of “heatmaps”.
- Visualisation of different heatmaps allows one to explore the relationship between the input variables.



https://www.shanelynn.ie/self-organising-maps-for-customer-segmentation-using-r/

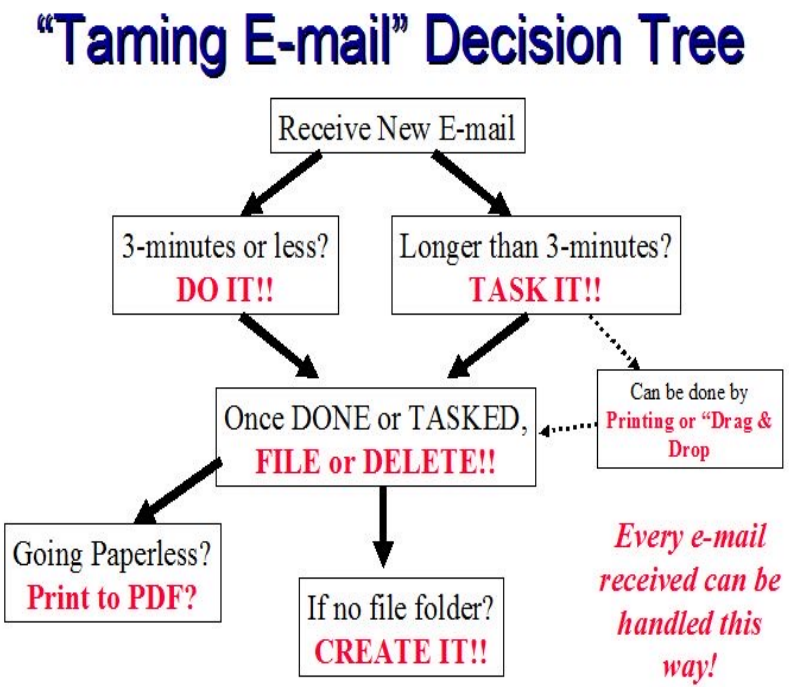
SELF-ORGANIZING MAPS: VISUALIZATION

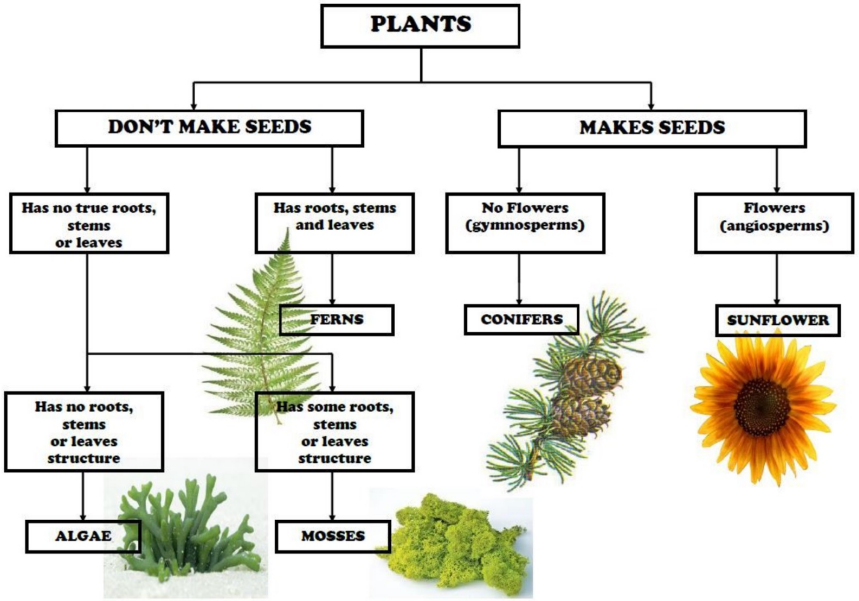
- U-Matrix visualization: the distance between each node and its neighbours.
- Useful visualization to find the “natural number” of clusters in the data without any a priori information.
- High value areas work as cluster separators



https://www.mdpi.com/2071-1050/11/5/1314/htm

WHAT IS A DECISION TREE?





DECISION TREES: TERMINOLOGY

- Root node
- Node
- Terminal node
- Branch
- Split

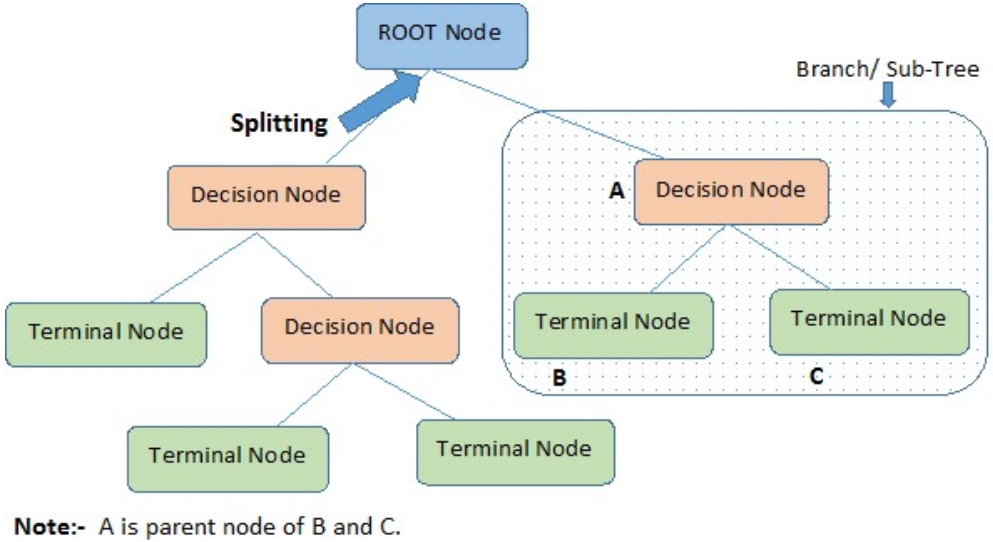
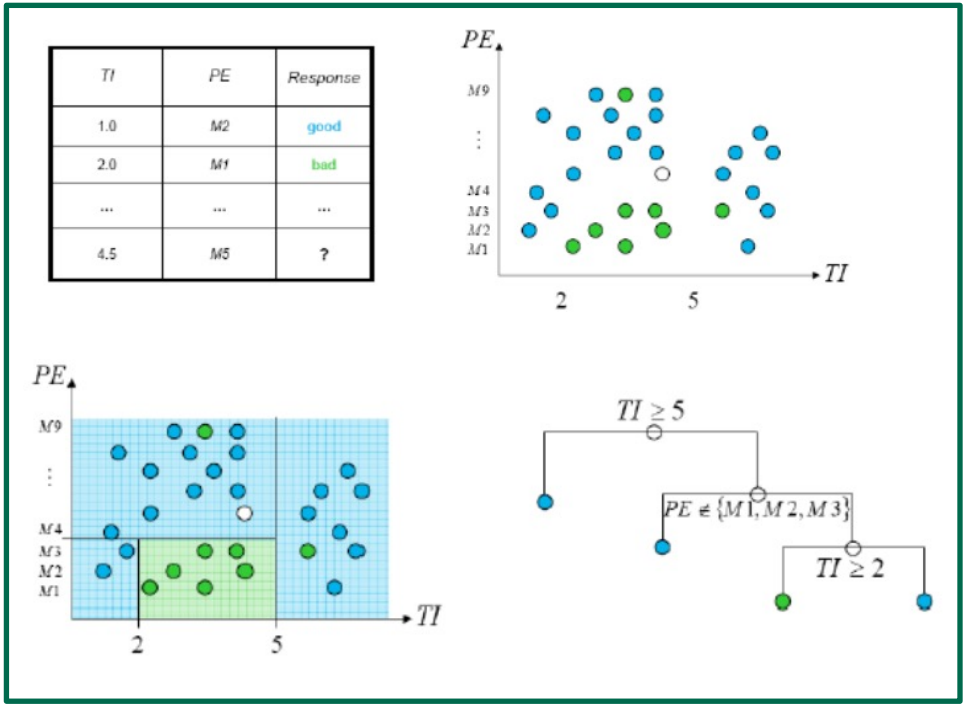
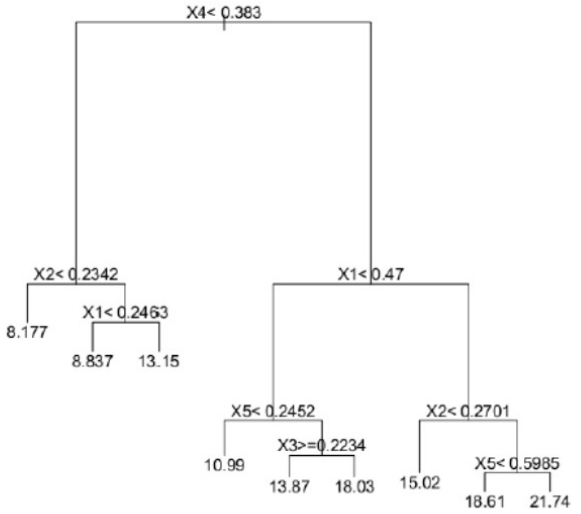


Figure from: https://wiki.pathmind.com/decision-tree

- Attribute (X1, X2, …)
- Response variables (Y)
- DT partition the data so that each unit is as homogeneous as possible wrt the response variable (Y)

DECISION TREES





RANDOM FORESTS

- Leo Breiman continued working on DT and around the year 2000 he found and demonstrated that classification and regression accuracy can be improved by using ensembles of trees (each tree grown in a “random” fashion.

- This work resulted in “random forests”

- Ensemble = a set of elements.
- Ensemble methods are becoming highly popular → computer power

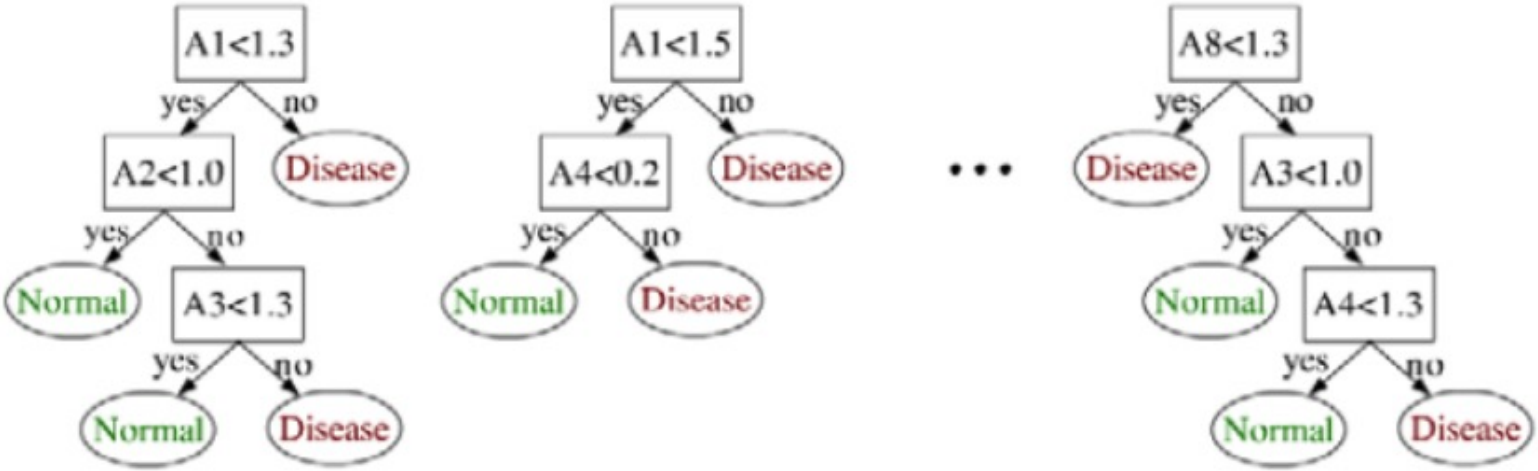
RANDOM FORESTS (IV)

- Input data: N training cases each with M variables

- n out of N samples are chosen with replacement (bagging).
- Rest of the samples to estimate the error of the tree (out of bag)
- m << M variables are used to determine the decision at a node of the tree

- Each tree is fully grown and not pruned

RANDOM FOREST: AN EXAMPLE

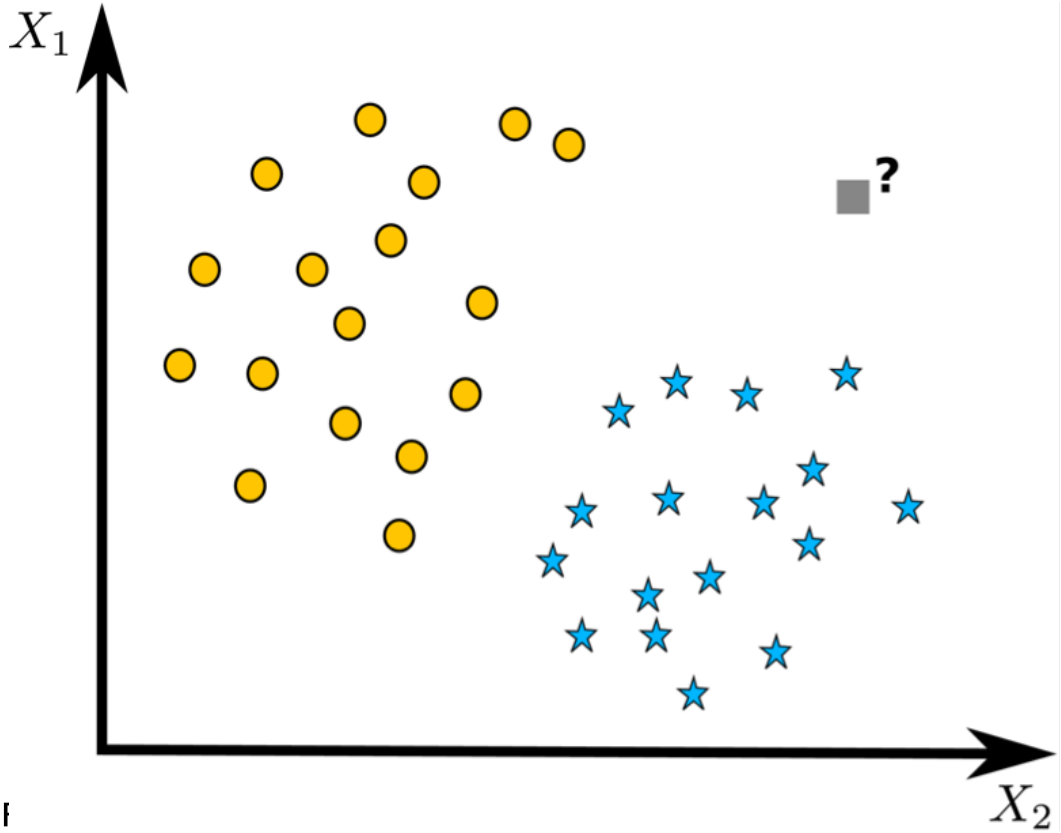


SUPPORT VECTOR MACHINE

- Robust algorithm for classification and regression tasks
- Looks for the plane/hyperplane that maximizes the distance between classes
- It can be seen as a *kernel method (high dimensional data)*

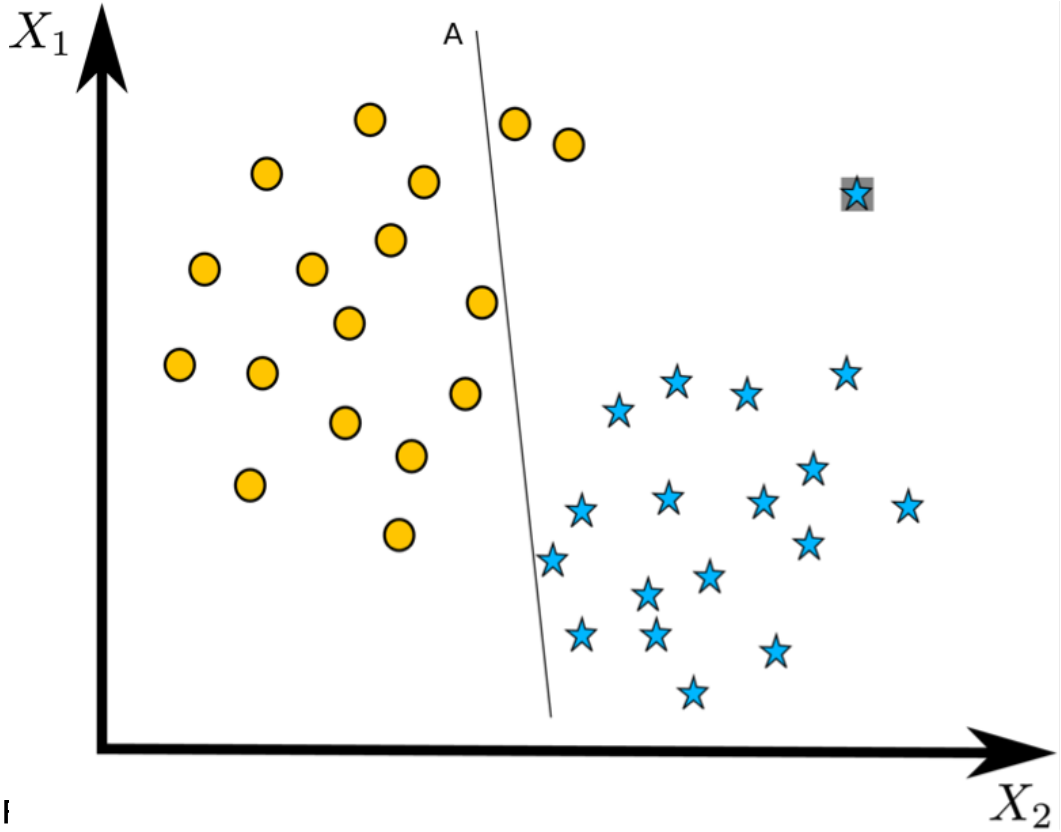
Poll: which class?

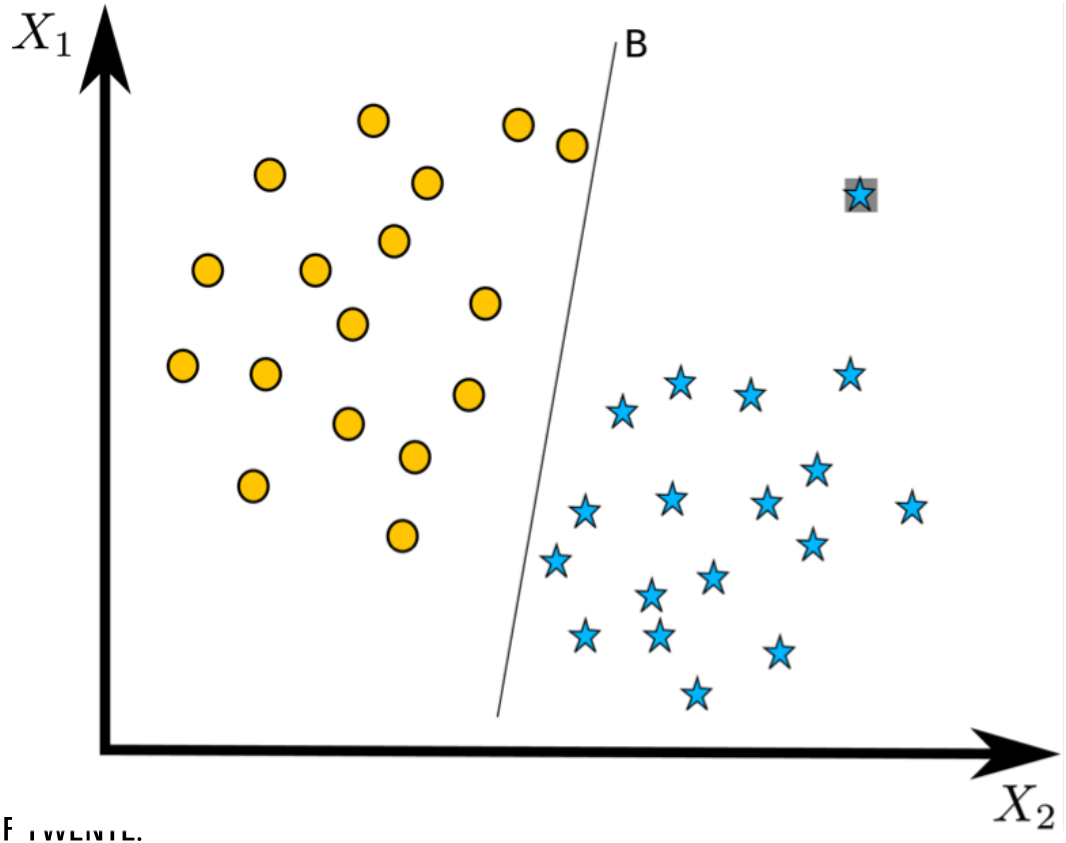
A)
B)

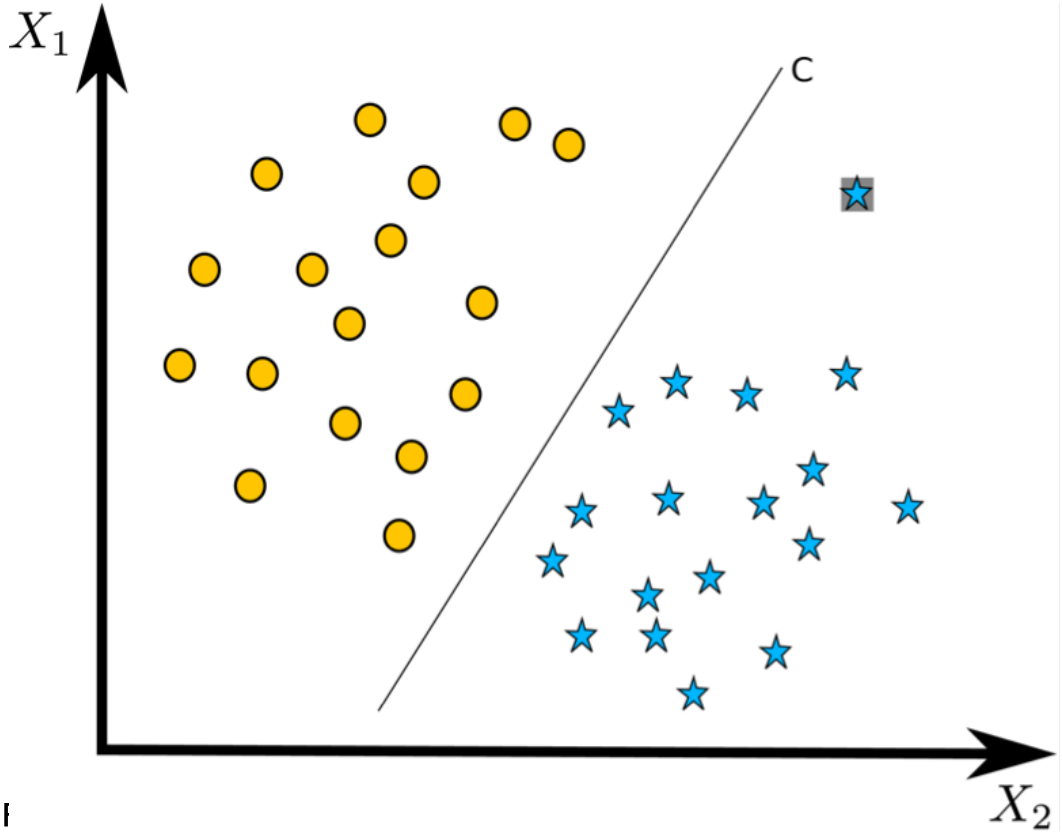


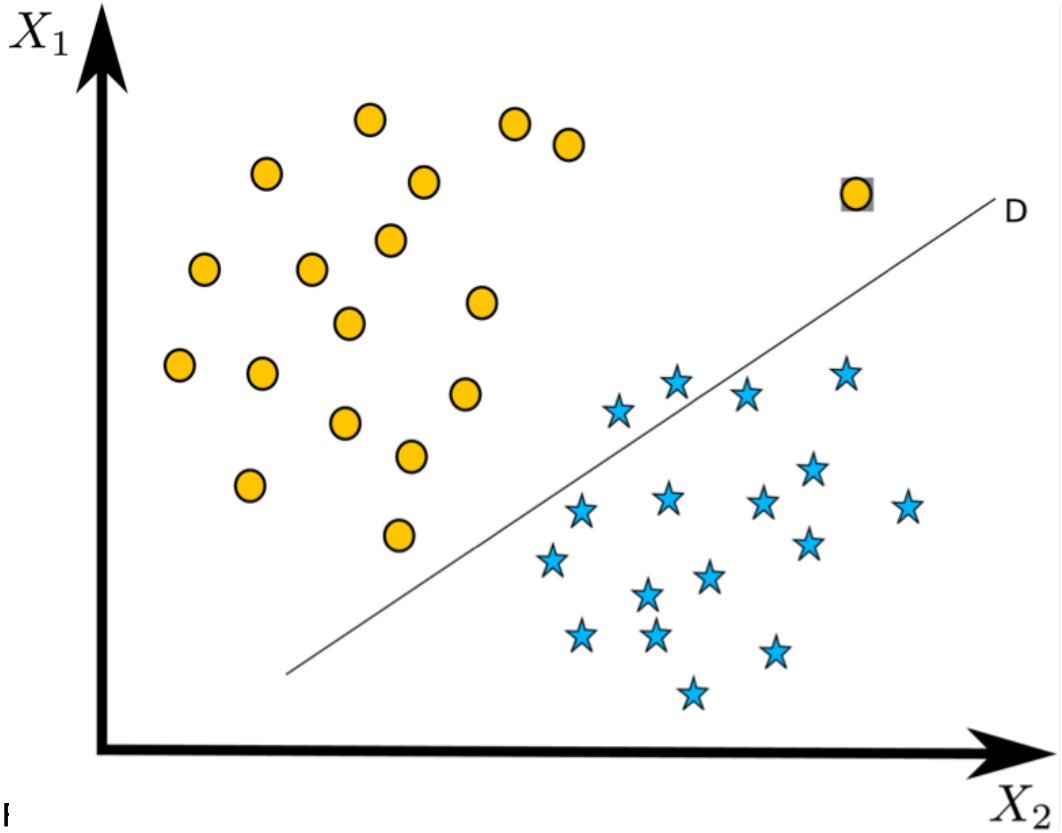
It depends….

on the selected (hyper)plane

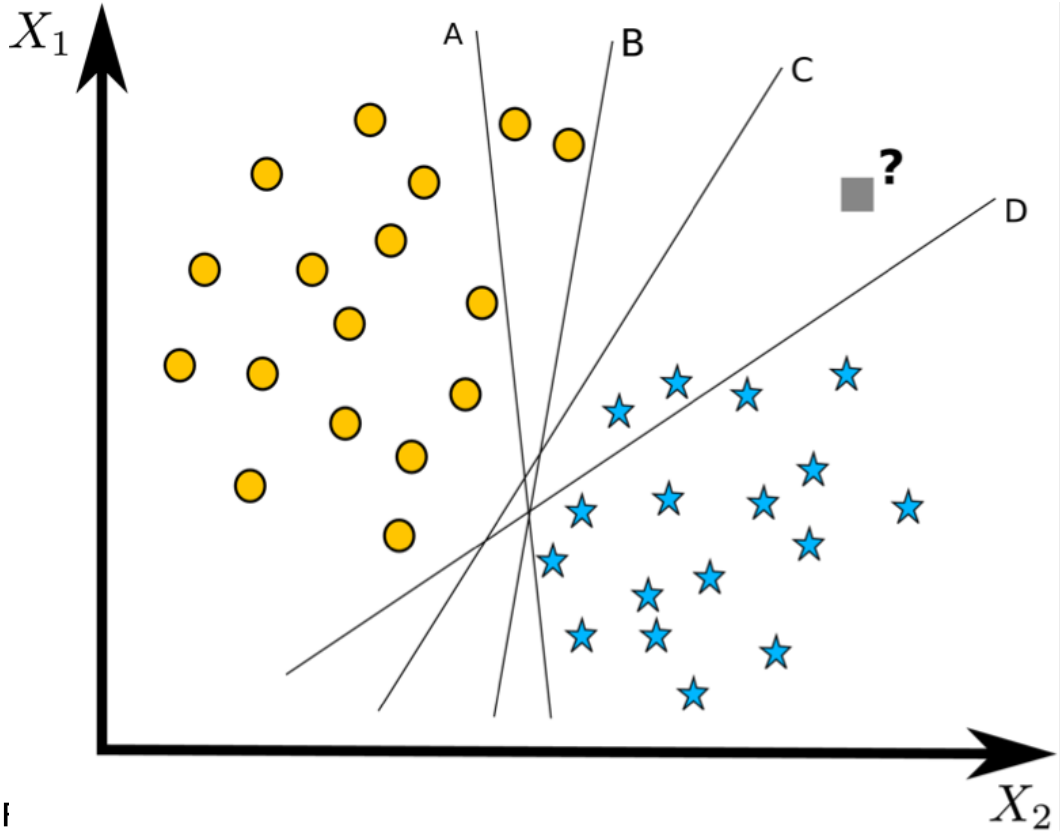

X_1
A
X_2

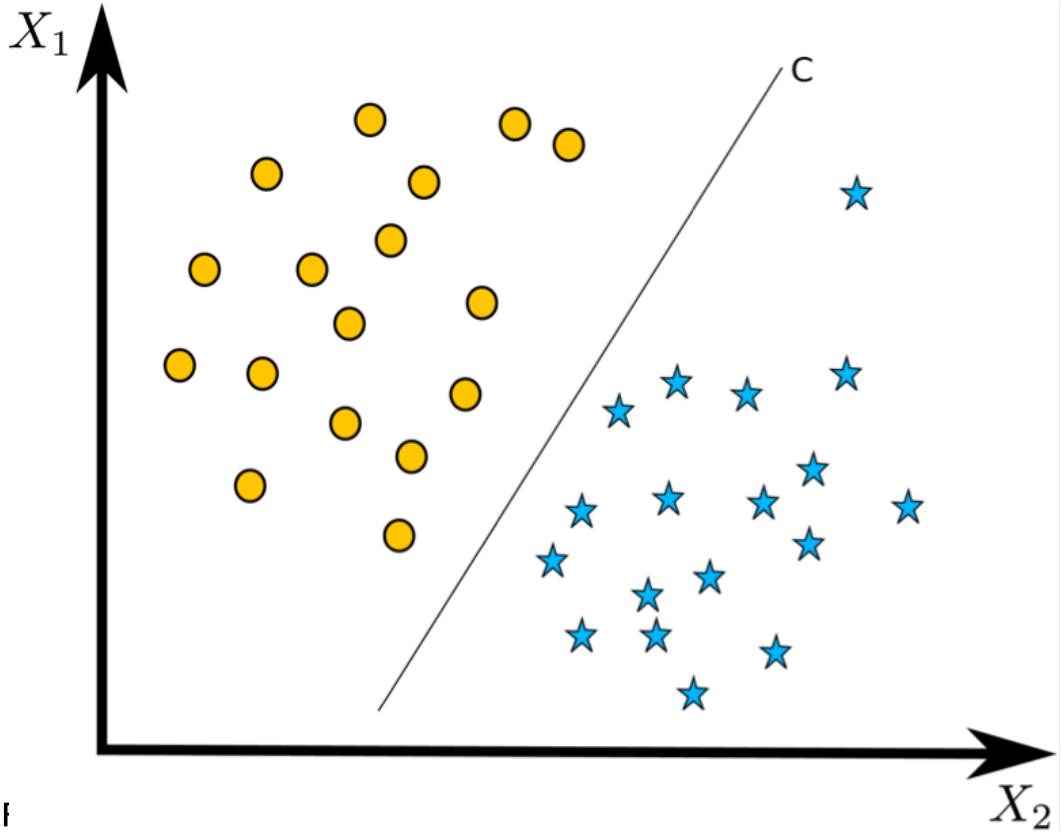

X_1
B
X_2

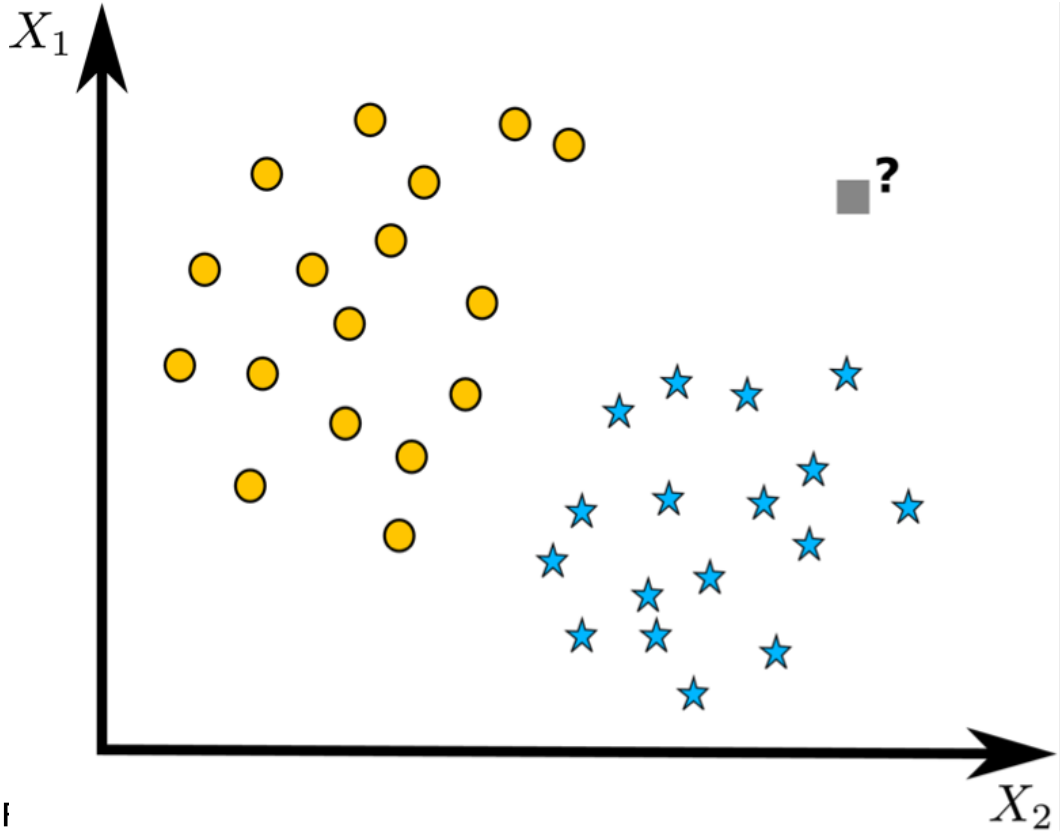

X_1
C
X_2

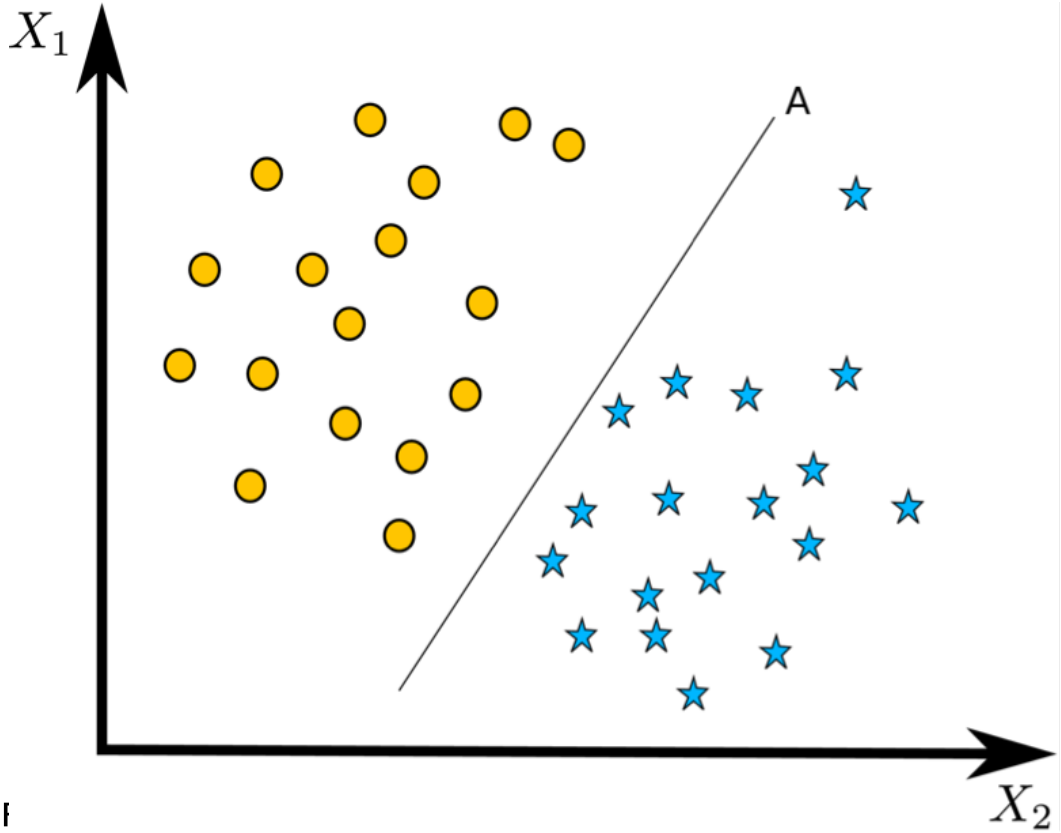

X_1
D
X_2

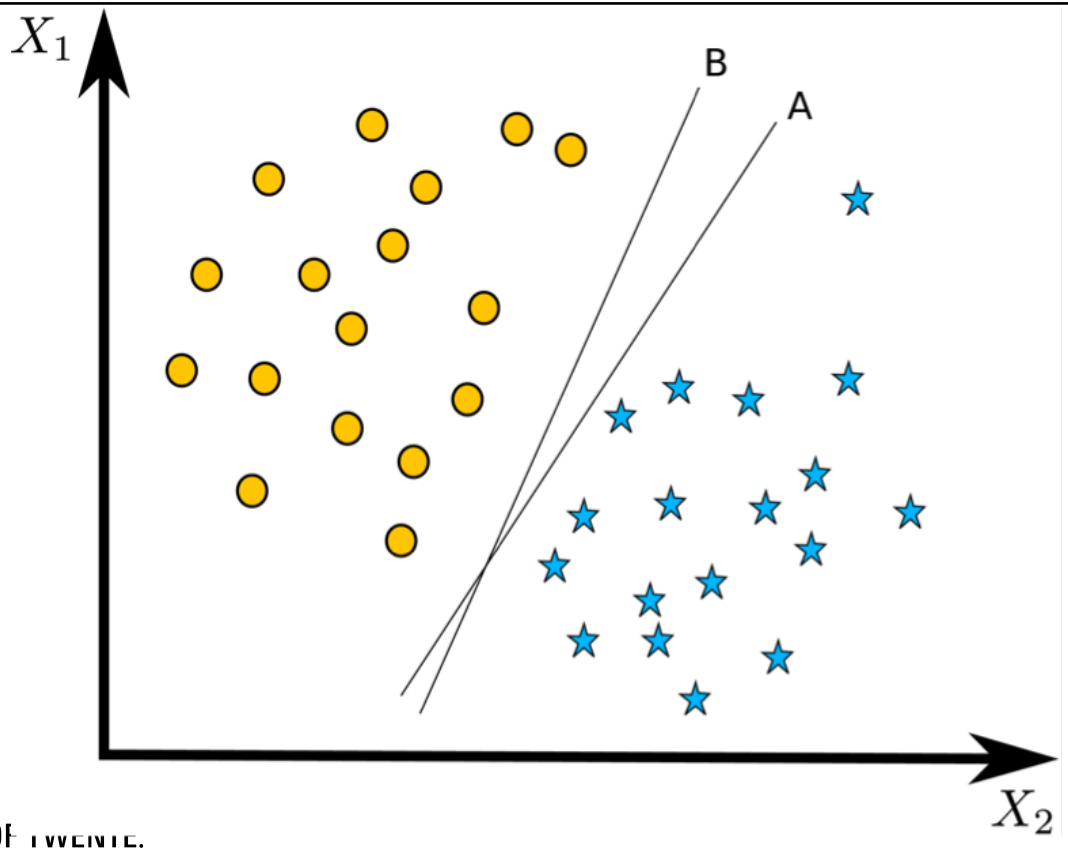
Poll: best Line?



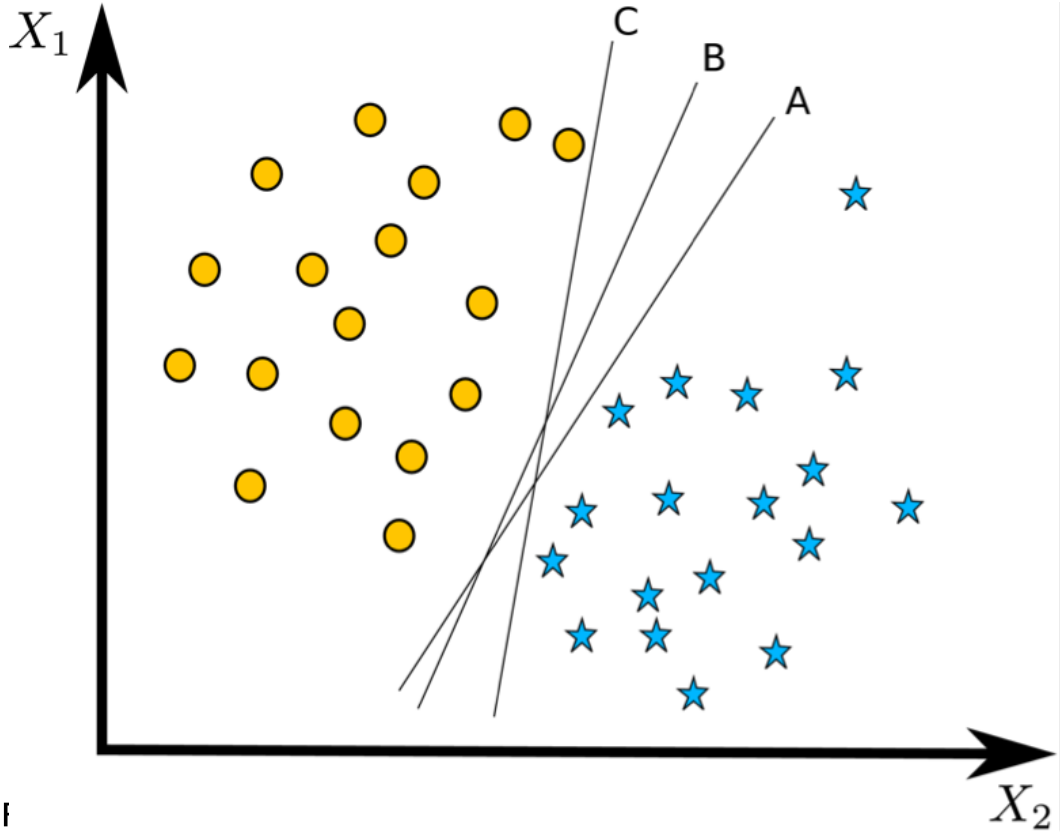

X_1
C
X_2

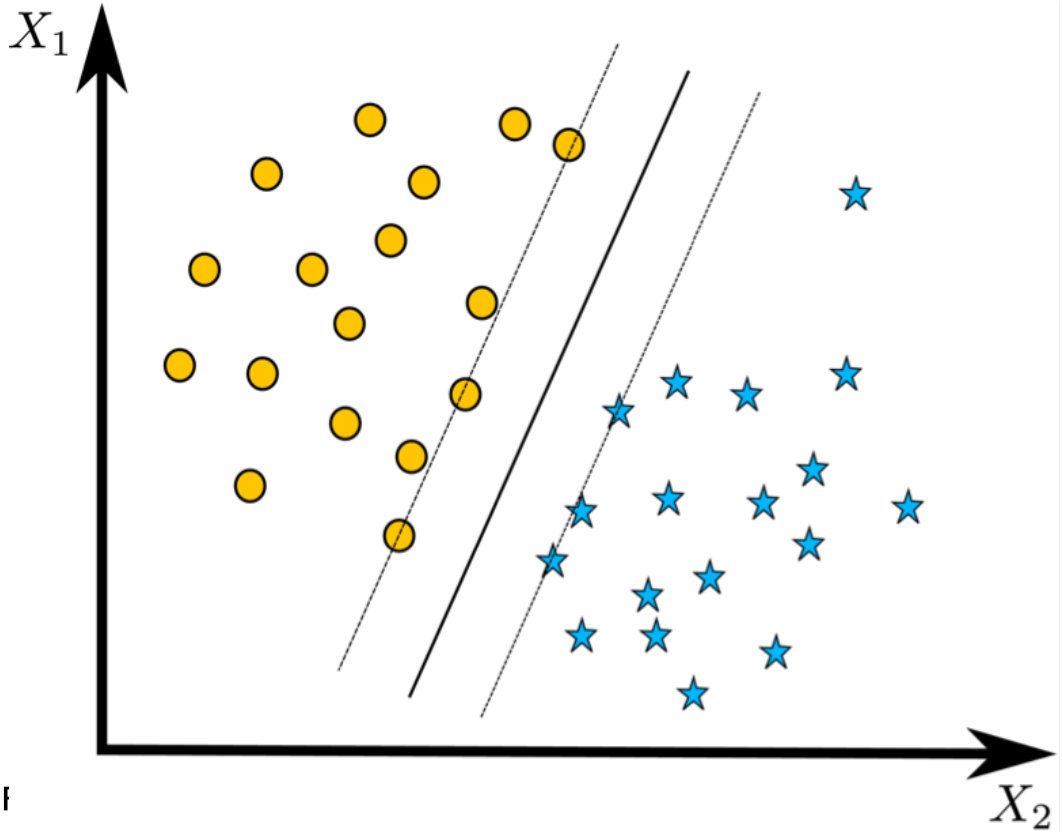

X_1
?
X_2

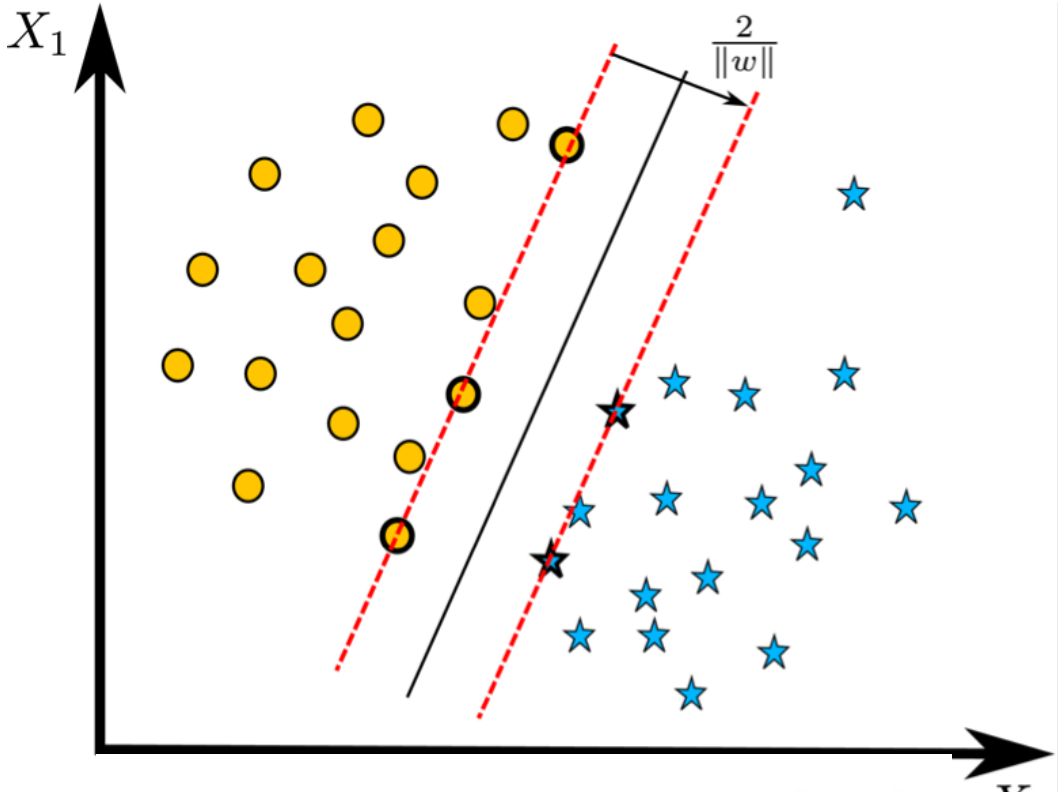

X_1
A
X_2


X_1
B
A
X_2

Poll: best Line?



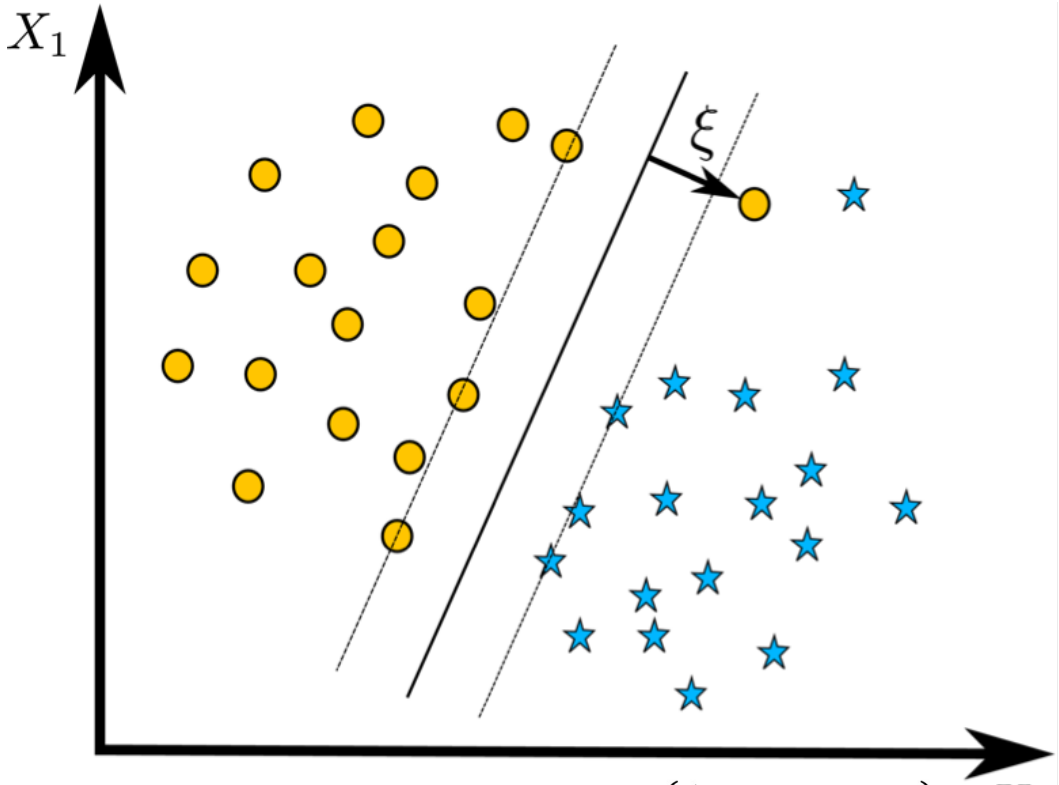

X_1
X_2



UNIVERSITY OF

Maximize margin separation = minimize $\|\mathbf{w}\|$: $\min_{\mathbf{w}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\}$

What happens with outliers?



Errors must be also penalized! $\min_{\mathbf{w}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_i \xi_i \right\}$

Outliers

$$\min_{\mathbf{w}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \right\}$$

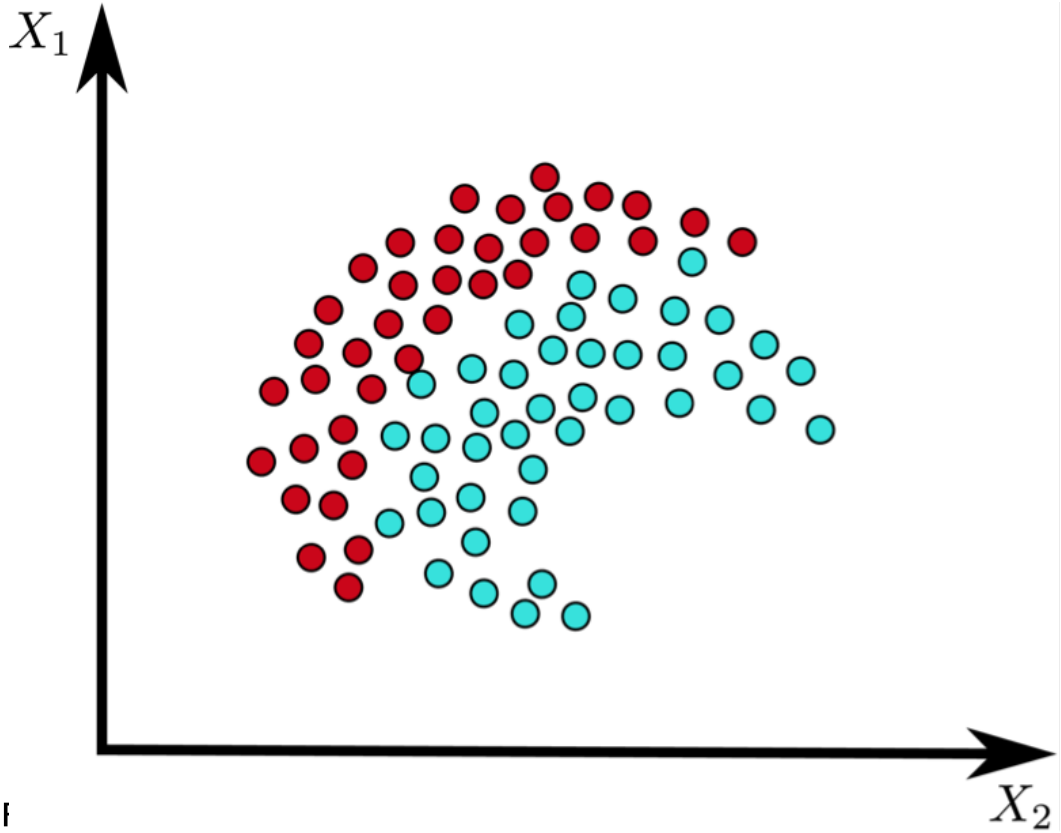
subject to:

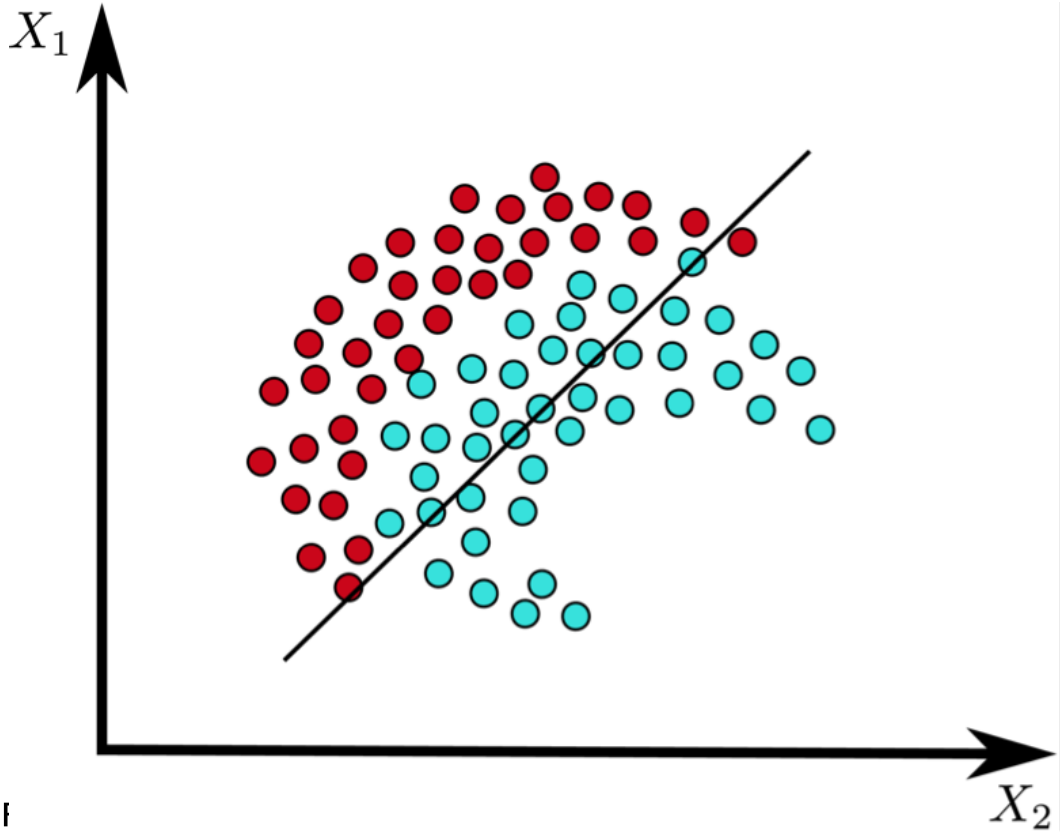
$$\begin{aligned} \mathbf{w}^\top \mathbf{x}_i + b &\geq 1 - \xi_i \quad y_i = +1, \forall i = 1, ..., n \\ \mathbf{w}^\top \mathbf{x}_i + b &\leq 1 - \xi_i \quad y_i = -1, \forall i = 1, ..., n \\ \xi_i &\geq 0 \end{aligned}$$

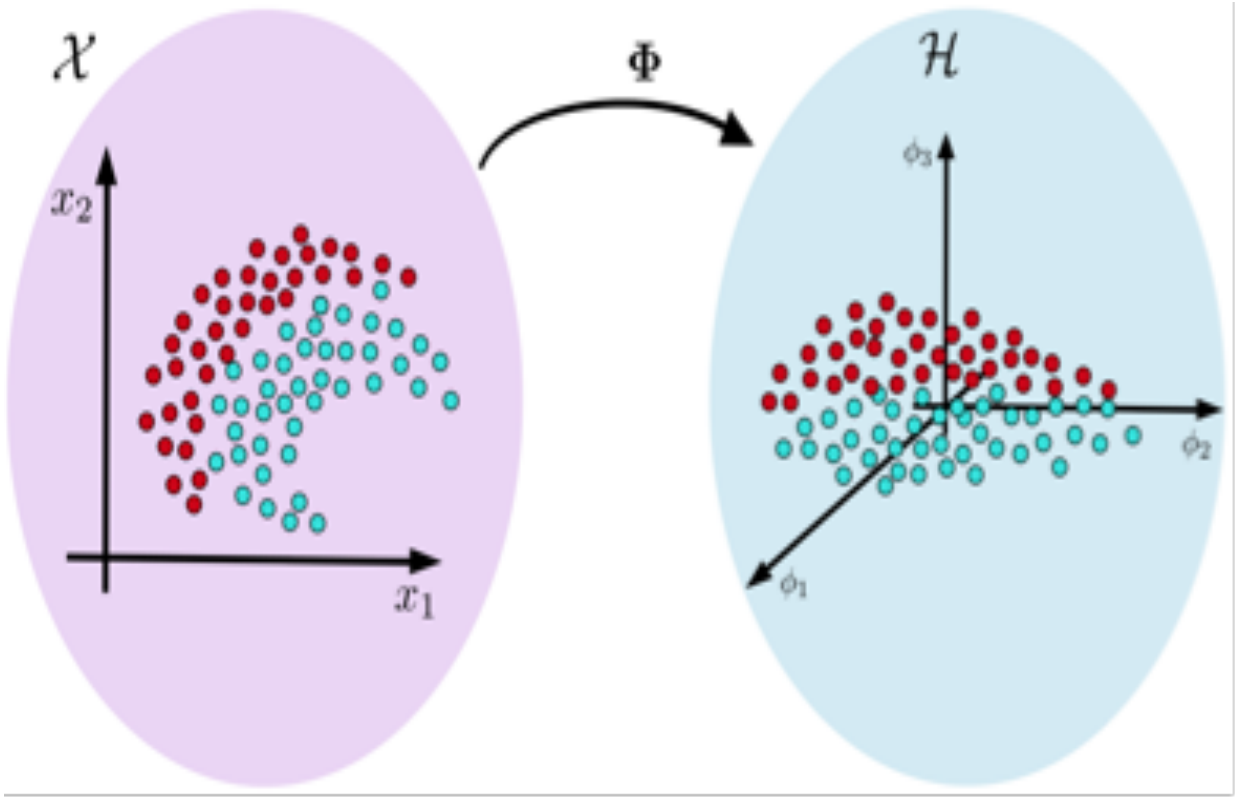
ξ_i is the error associated to misclassify example $\mathbf{x}_i$

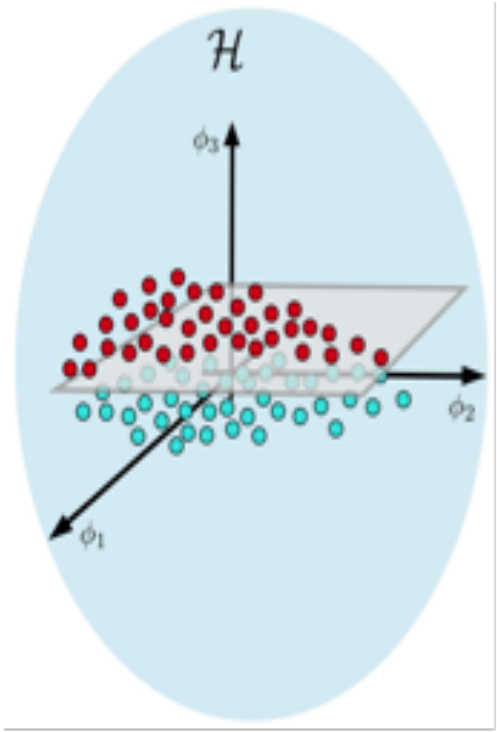
C is a tradeoff parameter controlling the overfitting

Non-linear




X_1
X_2


$\mathcal{X}$
Φ
$\mathcal{H}$
x_2
x_1
ϕ_3
ϕ_2
ϕ_1


$\mathcal{H}$
ϕ_3
ϕ_2
ϕ_1

Φ ?

- Map the data into a space where we have a notion of similarity, namely a dot product space $\mathcal{H}$ (**feature space**), using the **feature mapping**

$$\phi : \mathcal{X} \to \mathcal{H}, \qquad \mathbf{x} \mapsto \phi(\mathbf{x})$$

- The similarity between the elements in $\mathcal{H}$ can now be measured using its associated dot product $\langle \cdot, \cdot \rangle_{\mathcal{H}}$.
- The **kernel function** measures similarity in $\mathcal{H}$:

$$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \qquad (\mathbf{x}, \mathbf{x}') \mapsto K(\mathbf{x}, \mathbf{x}')$$

which we require to satisfy for all $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$

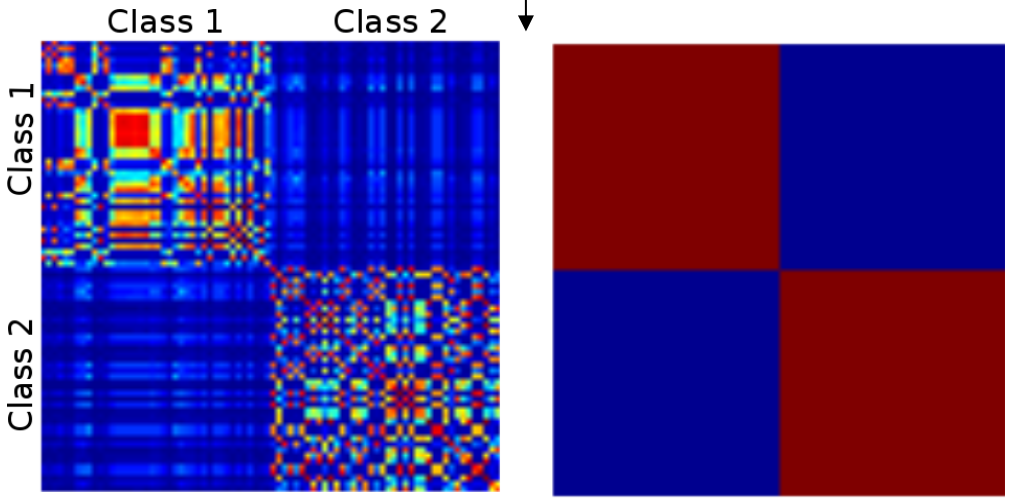
$$K(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle_{\mathcal{H}}$$

Kernel function

Common kernel:

Gaussian Function (RBF): $K(x_i, x_j) = exp(-\|x_i - x_j\|^2 / (2\sigma^2))$

Measures similarity among the samples



SVM: step by step

- **Step 1:** Map examples to a higher dimensional space, $\phi : \mathbb{R}^N \rightarrow \mathcal{H}$

$$\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i), \qquad i = 1, \ldots, n$$

- **Step 2:** Replace dot products $\langle \cdot, \cdot \rangle$ in $\mathcal{H}$ by a kernel function $k(\cdot, \cdot)$

$$\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = K(\mathbf{x}_i, \mathbf{x}_j)$$

- **Step 3:** Solve the same maximum margin problem

$$\max_{\boldsymbol{\alpha}} \left\{ -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \underbrace{\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)}_{=K_{ij}} + \sum_{i=1}^{n} \alpha_i \right\}$$

- **Step 4:** Prediction involves comparing test to train samples with $K(\cdot, \cdot)$:

$$\hat{y}_j = f(\mathbf{x}_j) = sign(\mathbf{w}^\top \phi(\mathbf{x}_j) + b) = sign\left(\sum_{i=1}^{n} \alpha_i y_i \underbrace{\phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)}_{=K_{ij}} + b \right)$$

ITC

RF vs SVM

http://www.researchgate.net/post/Is_random_forest_better_than_support_vector_machines

Sagara Sumathipala

Nagaoka University of Technology

Is random forest better than support vector machines?

I have some problems when comparing machine learning algorithms. Sometimes support vector machines gives better results than random forest. But not always. How do we compare these two algorithms?

TOPICS

Random Forests | Support Vector Machine | Machine Learning | Advanced Machine Learning

Dec 20, 2013

Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?

Manuel Fernández-Delgado MANUEL.FERNANDEZ.DELGADO@USC.ES
Eva Cernadas EVA.CERNADAS@USC.ES
Senén Barro SENEN.BARRO@USC.ES
CITIUS: Centro de Investigación en Tecnoloxías da Información da USC
University of Santiago de Compostela
Campus Vida, 15872, Santiago de Compostela, Spain

Dinani Amorim DINANIAMORIM@GMAIL.COM
Departamento de Tecnologia e Ciências Sociais- DTCS
Universidade do Estado da Bahia
Av. Edgard Chastinet S/N - São Geraldo - Juazeiro-BA, CEP: 48.905-680, Brasil

Editor: Russ Greiner

Abstract

We evaluate **179 classifiers** arising from **17 families** (discriminant analysis, Bayesian, neural networks, support vector machines, decision trees, rule-based classifiers, boosting, bagging, stacking, random forests and other ensembles, generalized linear models, nearest-neighbors, partial least squares and principal component regression, logistic and multinomial regression, multiple adaptive regression splines and other methods), implemented in Weka, R (with and without the caret package), C and Matlab, including all the relevant classifiers available today. We use **121 data sets**, which represent **the whole UCI** data base (excluding the large-scale problems) and other own real problems, in order to achieve significant conclusions about the classifier behavior, not dependent on the data set collection. **The classifiers most likely to be the bests are the random forest** (RF) versions, the best of which (implemented in R and accessed via caret) achieves 94.1% of the maximum accuracy overcoming 90% in the 84.3% of the data sets. However, the difference is not statistically significant with the second best, the SVM with Gaussian kernel implemented in C using LibSVM, which achieves 92.3% of the maximum accuracy. A few models are clearly better than the remaining ones: random forest, SVM with Gaussian and polynomial kernels, extreme learning machine with Gaussian kernel, C5.0 and avNNet (a committee of multi-layer perceptrons implemented in R with the caret package). The random forest is clearly the best family of classifiers (3 out of 5 bests classifiers are RF), followed by SVM (4 classifiers in the top-10), neural networks and boosting ensembles (5 and 3 members in the top-20, respectively).

Keywords: classification, UCI data base, random forest, support vector machine, neural networks, decision trees, ensembles, rule-based classifiers, discriminant analysis, Bayesian classifiers, generalized linear models, partial least squares and principal component regression, multiple adaptive regression splines, nearest-neighbors, logistic and multinomial regression

GARBAGE IN – GARBAJE OUT