Materials: classification and clustering

Noël JAKSE
University Grenoble-Alps, SIMAP Laboratory

Scientific paradigms

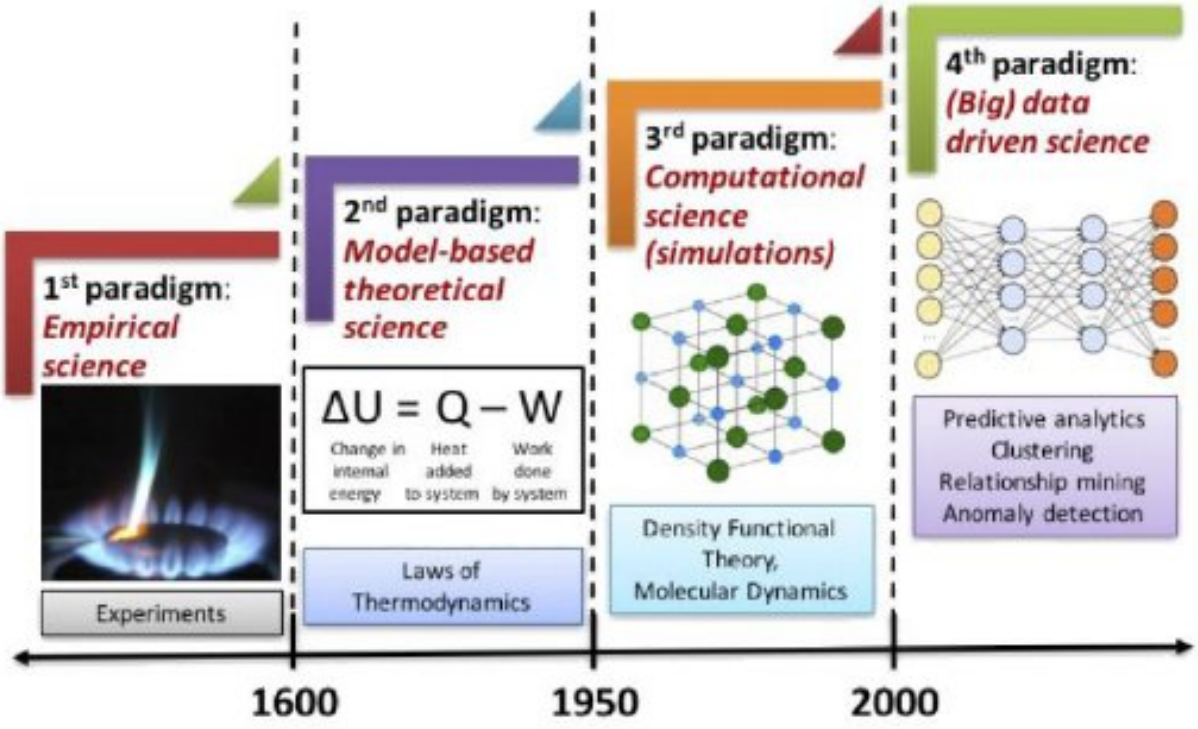
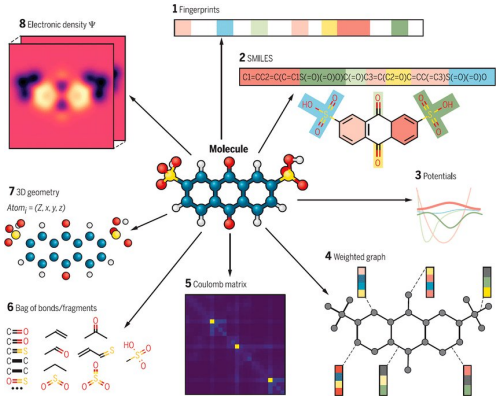


FIG. 1. The four paradigms of science: empirical, theoretical, computational, and data-driven.

A. Agrawal et A.Choudhray, Applied Materials, "Perspective: Material informatics and big data: realization of the 4th paradigm of science in materials science", 4 (2016)

Materials: classification and clustering



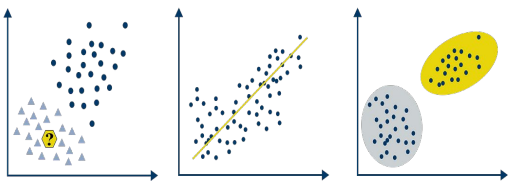
Experimental + Ab initio databases

Datasets :
A resource of materials data

Human brain:
Descriptors :
A representation to describe each material quantitatively

Numerical brain
Intelligent search
(***Machine Learning tools***)

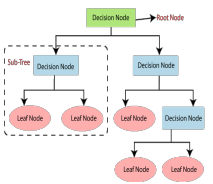
Results (Predictions):
Guide towards new materials
or at least
Define candidates
to be tested in real experiments
Uncover structure/property relationship



Classification: supervised

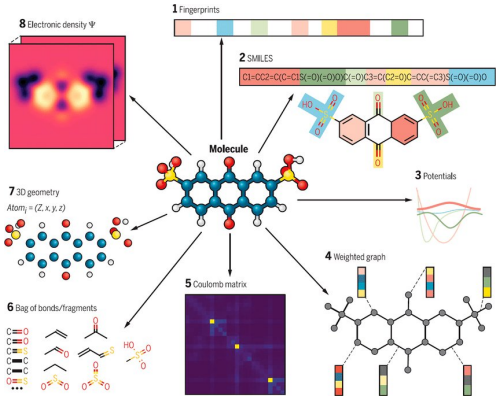
Regression:

Clustering: unsupervised



. . .

Materials: classification and clustering



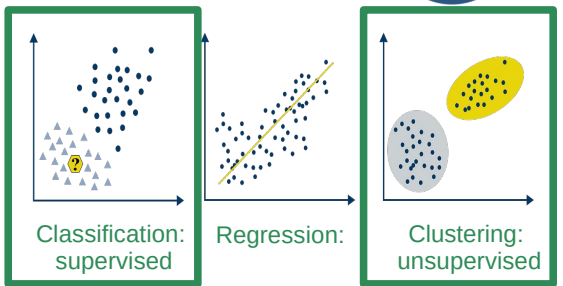
Experimental + Ab initio databases

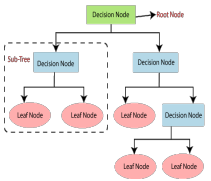
Datasets :
A resource of materials data

Human brain:
Descriptors :
A representation to describe each material quantitatively

Numerical brain
Intelligent search
(***Machine Learning tools***)

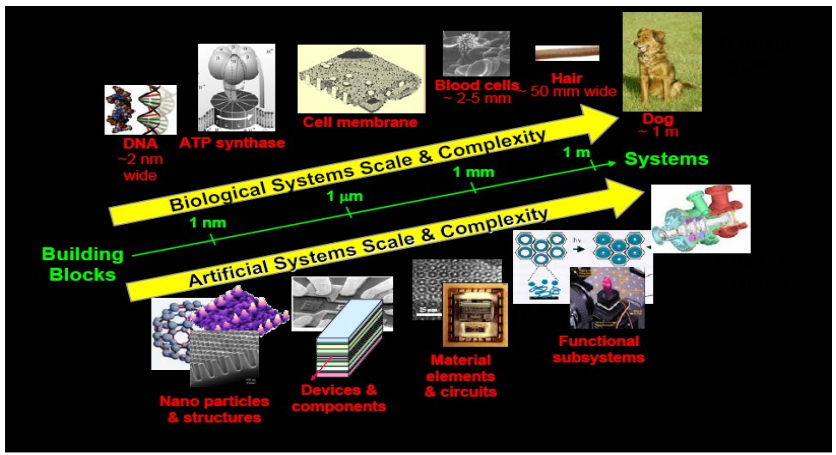
Results (Predictions):
Guide towards new materials
or at least
Define candidates
to be tested in real experiments
Uncover structure/property relationship

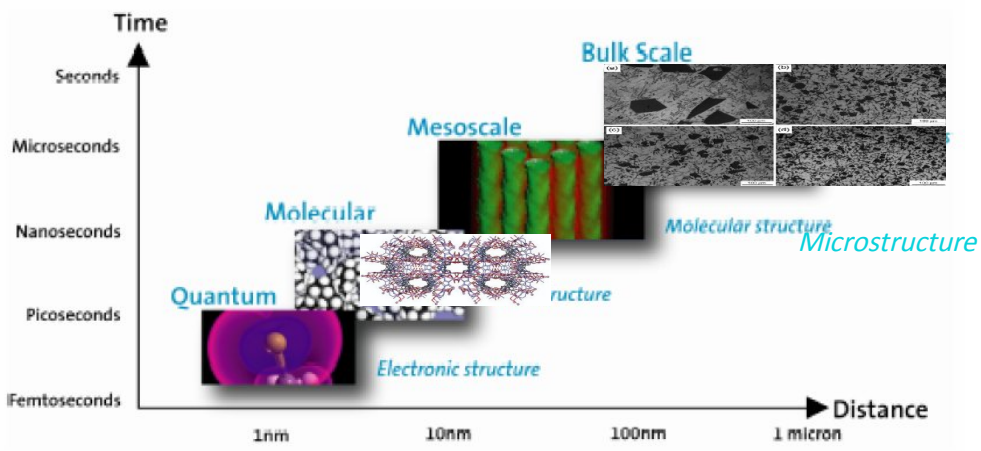




. . .

Atomistics-based materials design





Organic materials

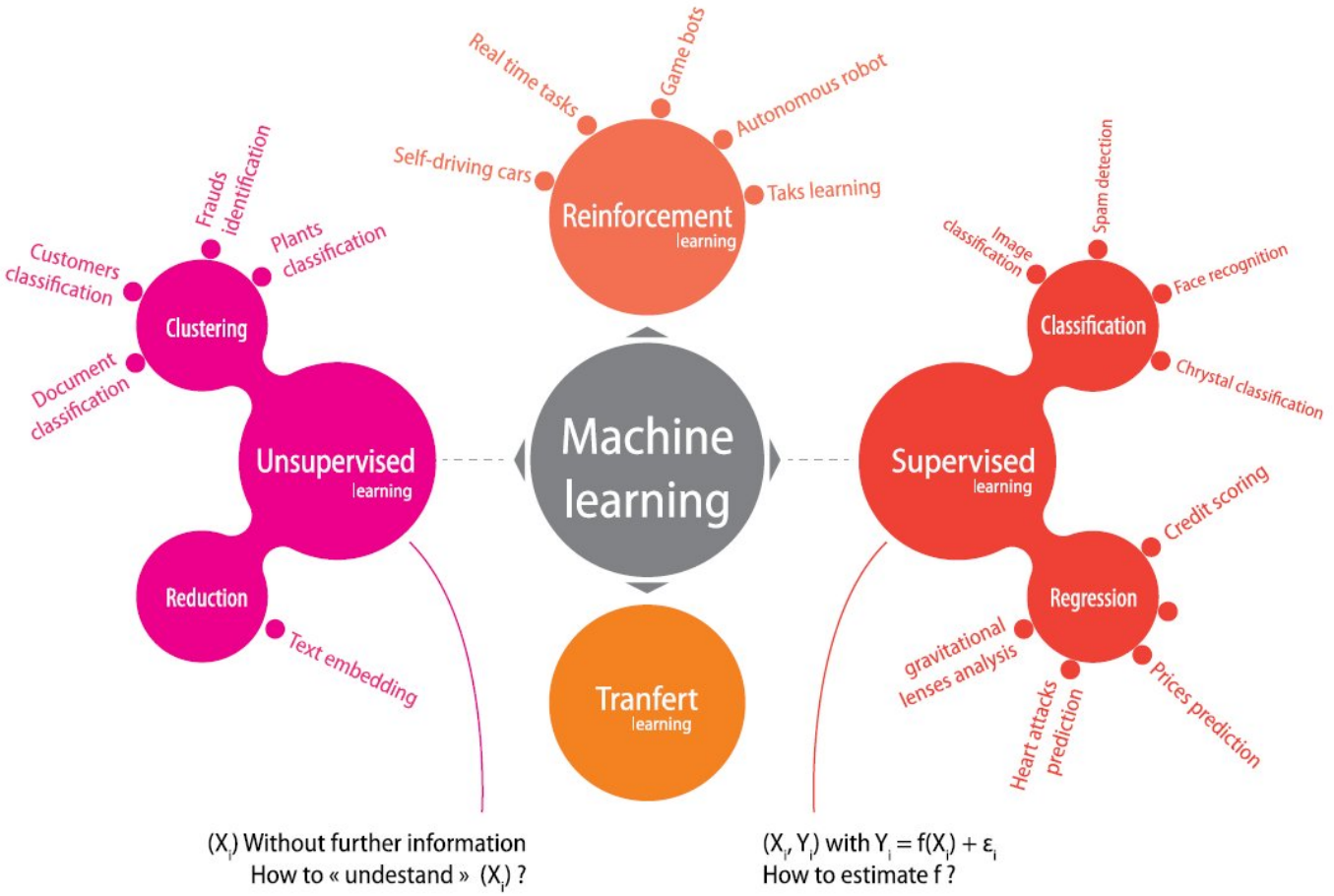
MOF

Metallic materials

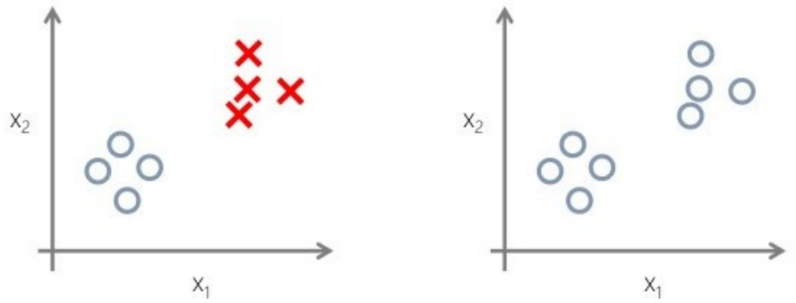
Alloy Nucleation

Alloy Microscrostructure and Properties

Supervised vs unsupervised

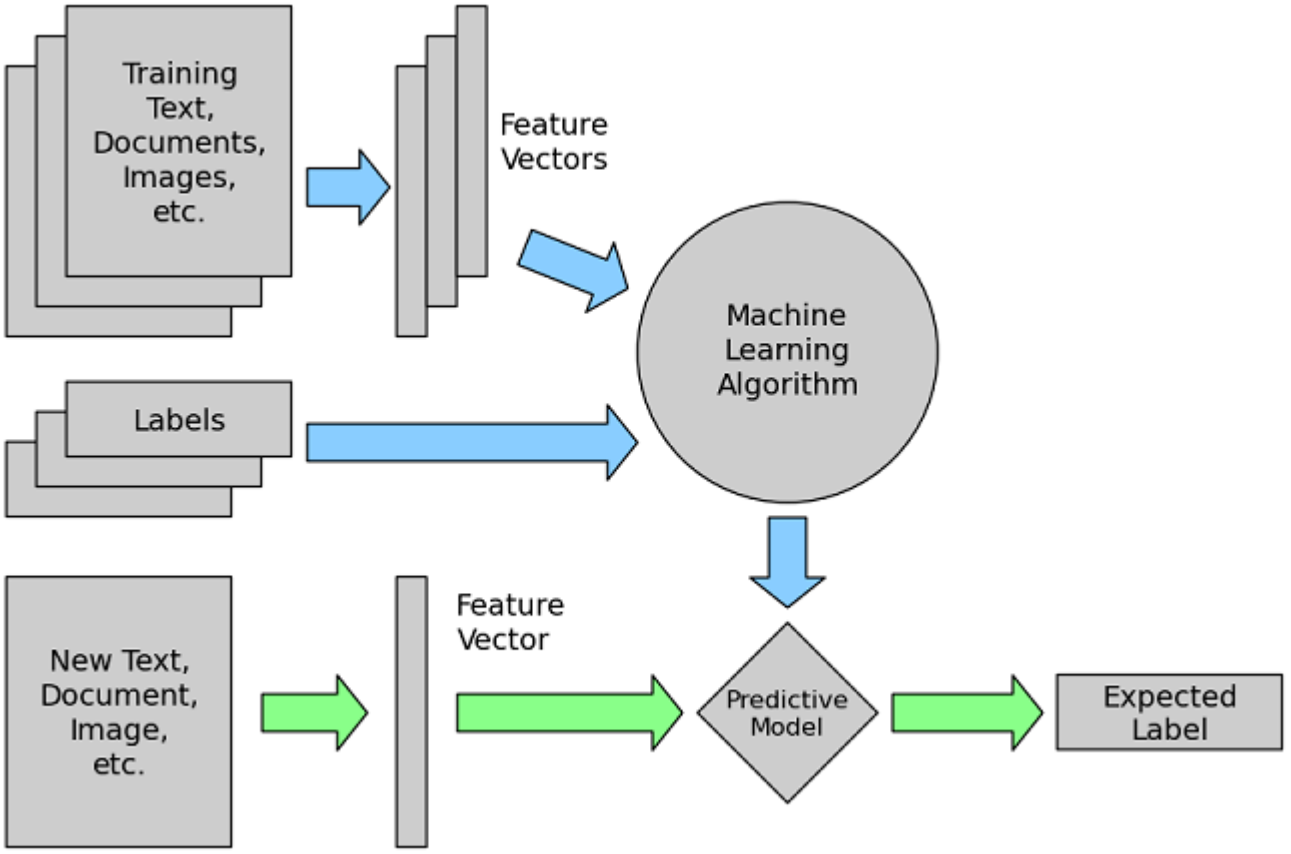


Supervised vs unsupervised

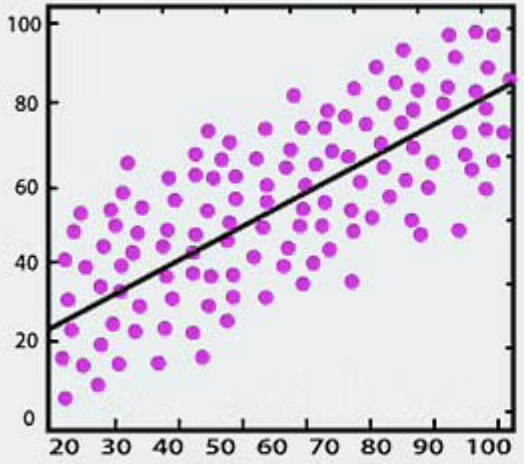


- Supervised learning:
  - **Objective**: learning a function *f* that predicts a variable *y* from a feature ***X***
  - **Dataset**: a training set
- Unsupervised learning:
  - **Objective**: discover a structure, eventually through a model *g,* from an ensemble individuals
  - **Dataset**: a set
- Semi-supervised learning:
  - **Objective**: combine a small amount of labeled data *y* with a large amount of unlabeled data during training. (in between supervised and unsupervised)

Principle of supervised learning

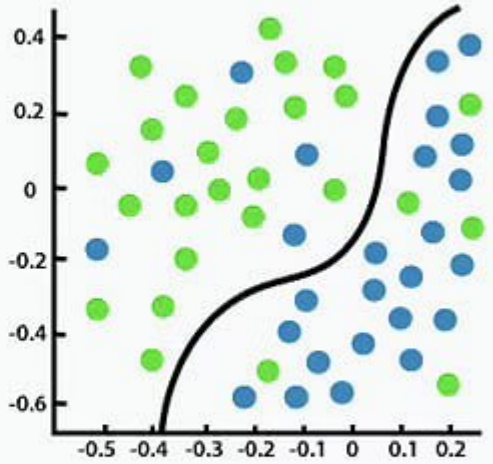


Principle of supervised learning



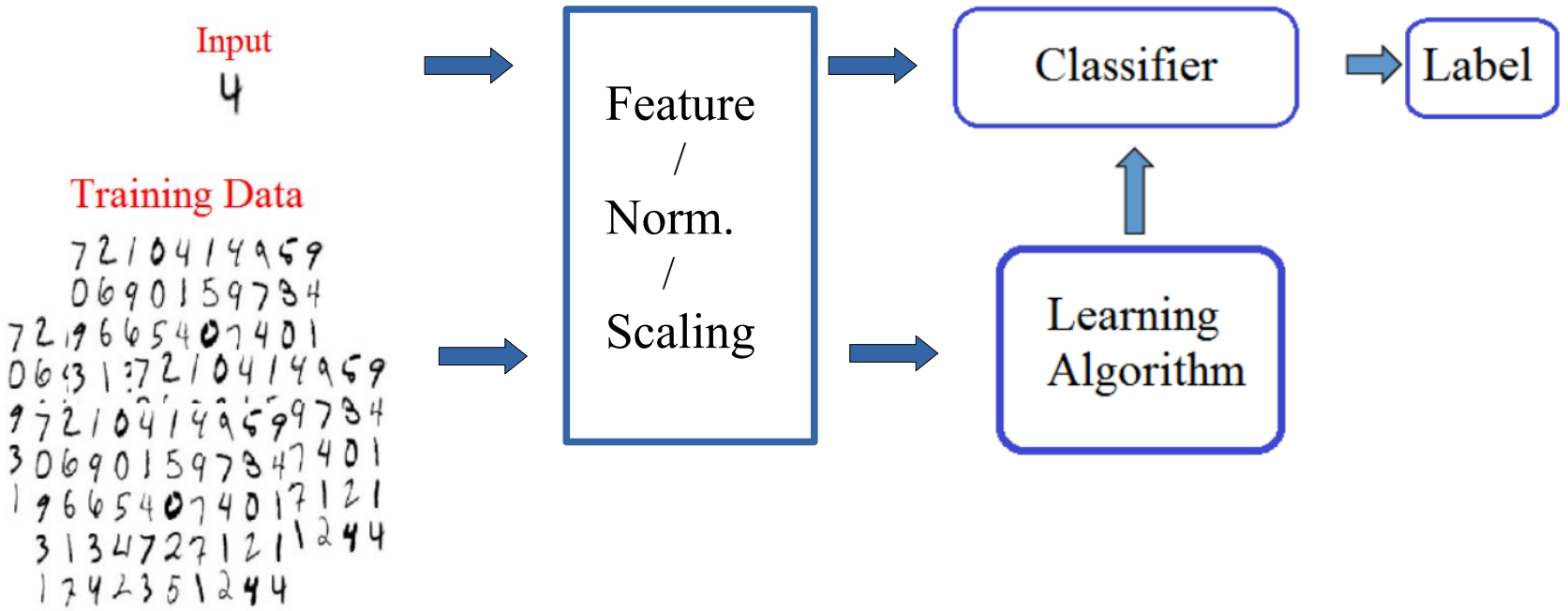
Regression

versus



Classification

Principle of supervised learning



Principle of supervised learning

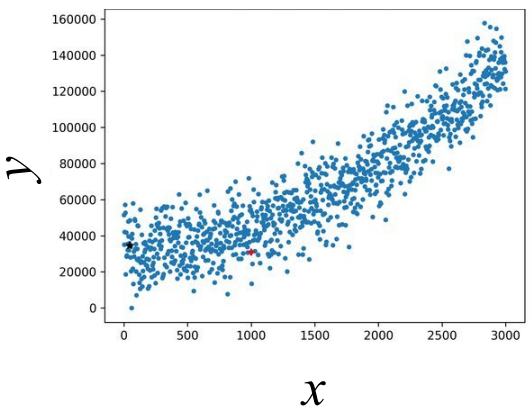
Input

x

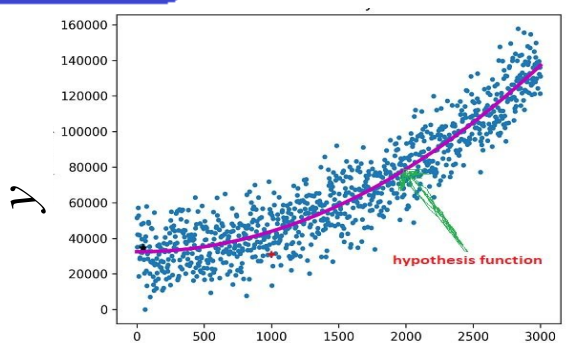
Regressor

y

Training Data



Learning Algorithm



Classification

1. Classification problem:

- Let's consider we have *n* labeled data

$$\{X_i, y_i, i = 1, \cdots n\}$$

- Data in a *d*-dimentional real space $X_i \in \mathbb{R}^d \equiv \mathcal{X}$
- Multi-class classification with *K* classes $y_i \in \mathcal{C} = \{1, \cdots, K\}$
- Specific case of binary classification $y_i \in \mathcal{C} = \{-1, 1\}$

2. Objective:

- For a new feature vector X_+, predict the value of the label $y_+ \in \mathcal{C}$
- For this purpose we use the training data to construct a classifier $\hat{c}$ such that

$$\hat{y}_+ = \hat{c}(X_+)$$

Classification

1. Constant classifiers on a partition:

- Let's consider a partition of $\mathcal{X}$

$$\mathcal{A} = \{A_1, A_2, \cdots, A_M\}$$

- This partition can depend on the data at hand
- The set of constant function $\mathcal{F}_{\mathcal{A}}$ on $\mathcal{A}$
- with the loss defined as: $\ell(y, y') = \mathbf{1}_{yy' \leq 0}$

2. Objective:

- Find a classifier $\hat{c}$ such that

$$\hat{c}_{\mathcal{A}} = \underset{c \in \mathcal{F}_{\mathcal{A}}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, c(X_i)) = \underset{c \in \mathcal{F}_{\mathcal{A}}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{Y_i c(X_i) \leq 0}$$

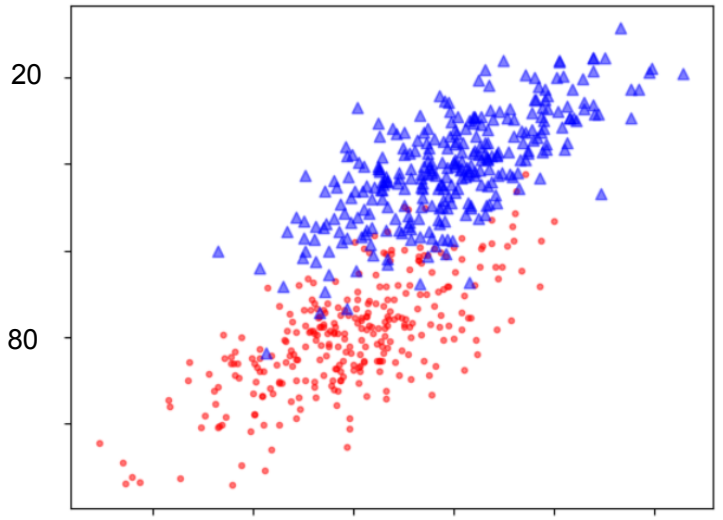
Classification

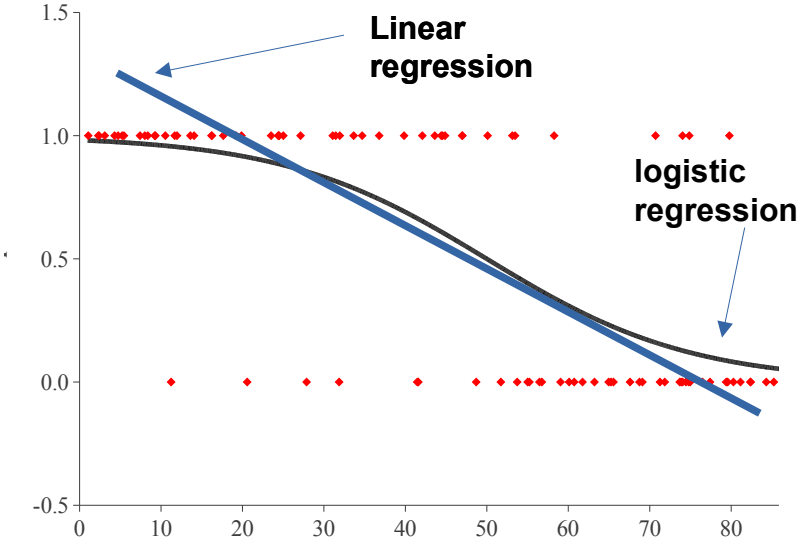
1.Common classifiers (among others):

- Decision Tree – the classifier has dataset attributes classed as nodes or branches in a tree.
- Random Forest – the classifier is a meta-estimator that fits a forest of decision trees and uses averages to improve prediction accuracy.
- K-Nearest Neighbors (KNN) – a simple classification algorithm, where K refers to the square root of the number of training records.
- Linear Discriminant Analysis – estimates the probability of a new set of inputs for every class. (unsupervised but used here on existing classes for an analysis of the dataset)
- Logistic Regression – a model with an input variable (*x*) and an output variable (*y*), which is a discrete value of either 1 (yes) or 0 (no).
- Naive Bayes – a family of classifiers based on a simple Bayesian model that is comparatively fast and accurate. Bayesian theory explores the relationship between probability and possibility.
- Support Vector Machines (SVMs) – a model with associated learning algorithms that analyze data for classification. Also known as Support-Vector Networks.

Classification

1. Logistic regression





Linear regression

$$\pi = \alpha + \beta X$$

logistic regression

$$\log\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta X$$

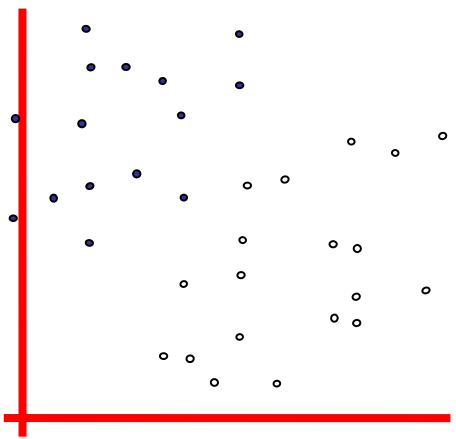
Classification

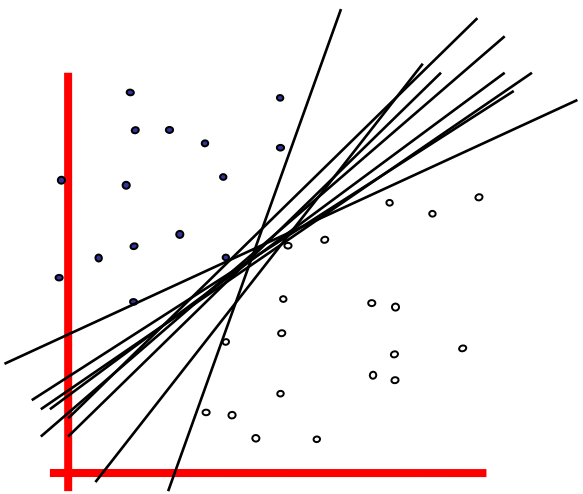
2. Support Vector Machines

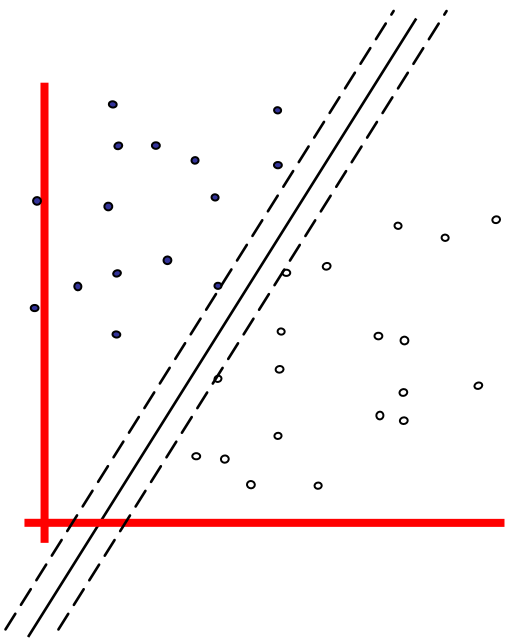
denotes +1

denotes -1

$$f(\boldsymbol{x},\boldsymbol{w},b) = sign(\boldsymbol{w}.\ \boldsymbol{x} - b)$$

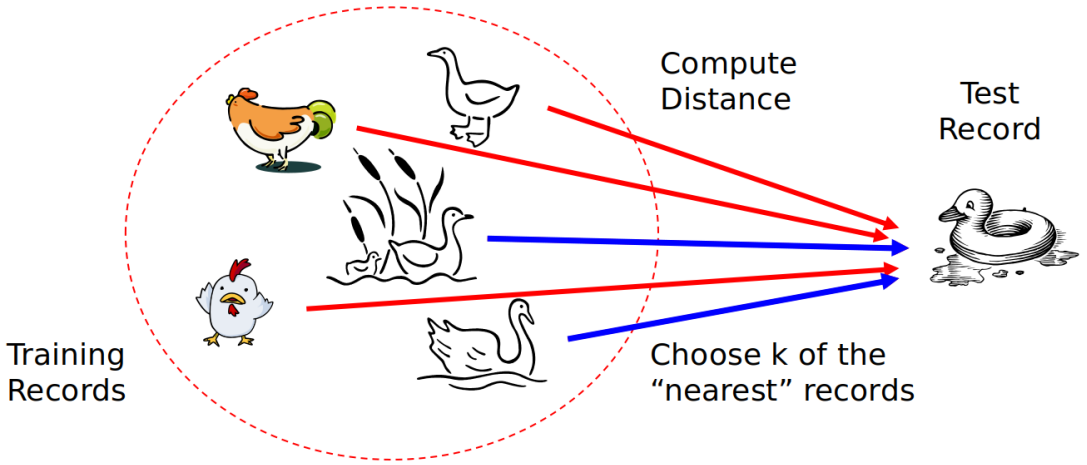


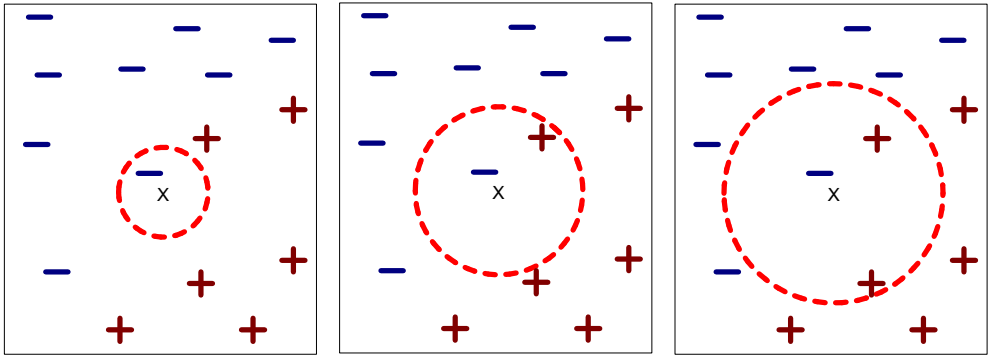




Classification

3. K-nearest-neighbor

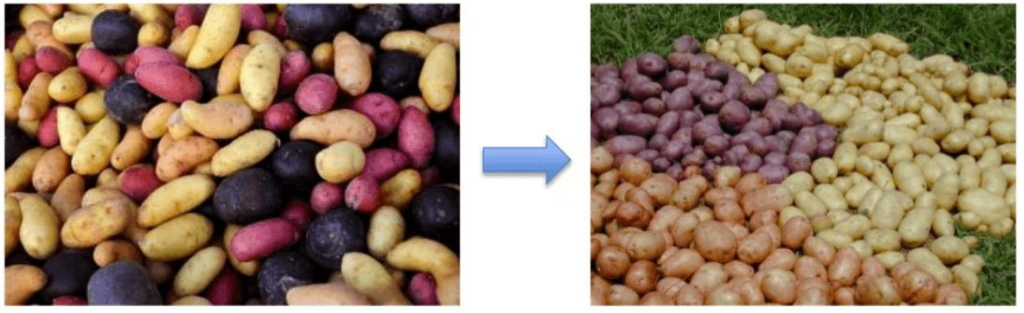




(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

Unsupervised learning

Clustering:

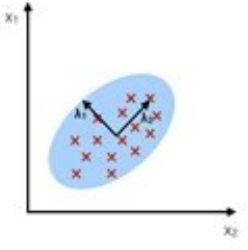


sample

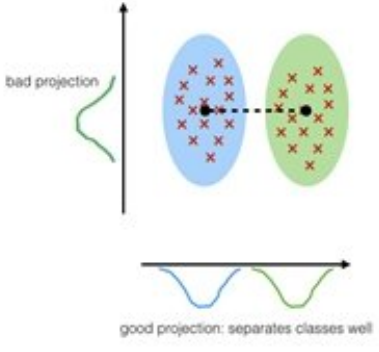
Cluster/group

Dimensionality reduction:

PCA:
component axes that maximize the variance



LDA:
maximizing the component axes for class-separation



Clustering

Different clustering approaches:

- **Exclusive (partitioning)**
  - Data are grouped in such a way that one data can belong to one cluster only.
  - Example: K-means

- **Agglomerative**
  - Every data is a cluster. The iterative unions between the two nearest clusters reduce the number of clusters.
  - Example: Hierarchical clustering

- **Overlapping**
  - Fuzzy sets is used to cluster data. Each point may belong to two or more clusters with separate degrees of membership.
  - Data will be associated with an appropriate membership value.
  - Example: Fuzzy C-Means

- **Probabilistic**
  - Uses probability distribution to create the clusters

Clustering

Clustering can be subjective:

Clustering

Main aspects of clustering:

- **Define your similarity or dissimilarity functions**
  - Most of the case define a distance function between objects

- **The clustering algorithm figures out the grouping of objects**
  - This is done by means of the chosen function
    - Points within a cluster is similar
    - Points across the different clusters are dissimilar or “not so similar”

- **Issues in clustering**
  - How to represent objects ? (vector space, normalisation, scaling …)
  - What is a similarity/dissimilarity function for your spécific data ?
  - What are the algorithm steps of this task ?

Clustering

Properties of similarity functions

- **symmetry**

$$d(x, y) = d(y, x)$$

- **Positive separability**

$$d(x, y) = 0 \text{ if and only if } x = y$$

- **Triangular inequality**

$$d(x, y) \leq d(x, z) + d(z, y)$$

Clustering

Data points: in representation space $\mathbb{R}^d$

$$x = (x_1, x_2, \ldots, x_d)^T$$

$$y = (y_1, y_2, \ldots, y_d)^T$$

- **Minkowski distance:**

$$d(x,y) = \sqrt[p]{\sum_{i=1}^{d}(x_i - y_i)^2}$$

- Euclidean distance : $p = 2 \quad d(x,y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$
- Manhattan distance $p = 1,\ d(x,y) = \sum_{i=1}^{d} |x_i - y_i|$
- “inf”-distance $p = \infty,\ d(x,y) = \max_{i=1}^{d} |x_i - y_i|$

Clustering

Algorithm for k-means

1: initialize *k* clusters randomly : $\{c^1, c^2, \ldots, c^k\}$

2: do

- Cluster assignment: decide the cluster membership of each datapoints x_i by assigning it to the nearsest cluster center

$$\pi(i) = argmin_{j=1,\ldots,k} \left\|x^i - c^j\right\|^2$$

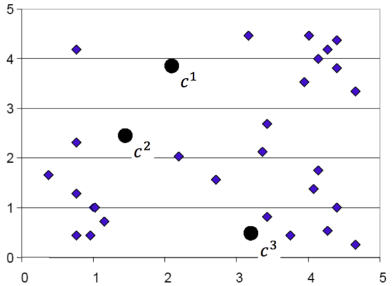
- Center adjustment : adjust the cluster center

$$c^j = \frac{1}{|\{i: \pi(i) = j\}|} \sum_{i:\pi(i)=j} x^i$$

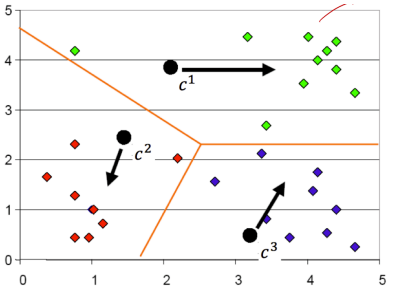
While there is any cluster change

Clustering

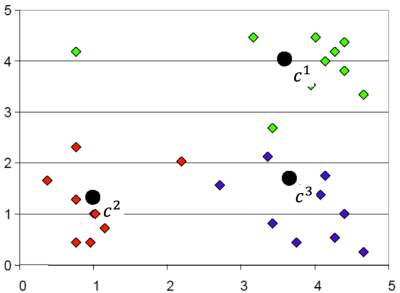
Example step-by-step



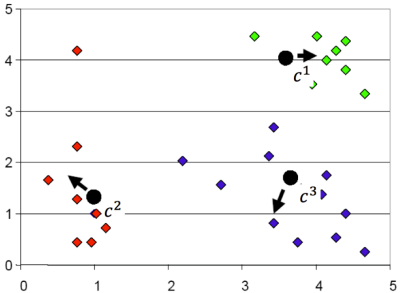
step 1



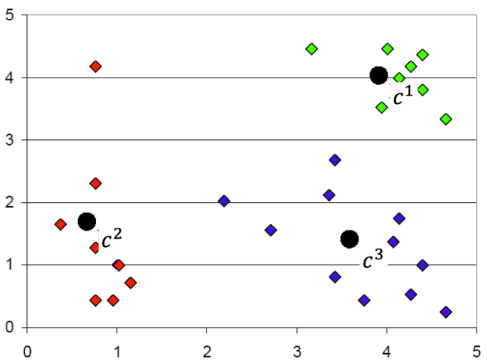
step 2



step 3



step 4



step 5

Principal component analysis (PCA)

Principal component analysis (PCA)

- Procedure which uses the correlations between the variables to identify which combinations of variables capture most information about the dataset
- Geometrically, it identifies the directions in which the cloud of variables is most elongated
- Mathematically, it determines the eigenvectors of the covariance matrix and sorts them in importance according to their corresponding eigenvalues
- PCA is commonly used for dimensionality reduction, i.e. approximating a dataset with a fewer number of variables

Principal component analysis (PCA)

Comprehensive procedure

with a bivariate random Gaussian (,) dataset (notebook):

1. Determine the covariance matrix of (,) :

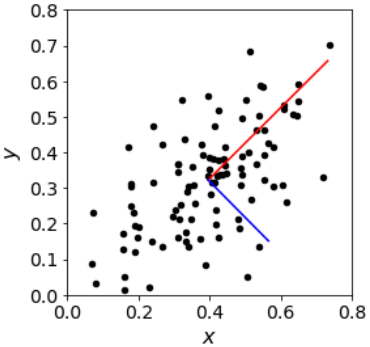
$$C = \begin{pmatrix} Cov(x,x) & Cov(x,y) \\ Cov(y,x) & Cov(y,y) \end{pmatrix}$$

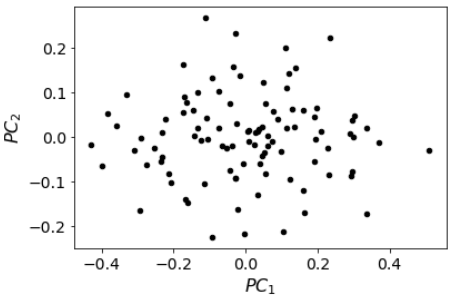
$$\sigma_{xy}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})$$

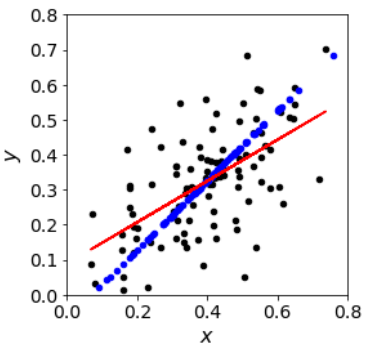
2. Determine the eigenvalues and eigenvectors of

3. Express the data points in the basis of the eigenvectors with new co-ordinates

4. Result along the axis:





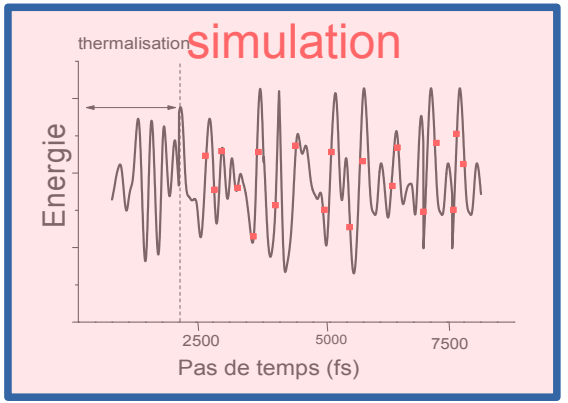


Semi-supervised Learning

- Harnessing the Power of Labeled and Unlabeled Data
- Combining supervised and unsupervised learning approach.
- Addressing the scarcity of labeled data in materials science.
- Reducing the need for expensive and time-consuming experiments.
- Exploiting the vast amount of available unlabeled data.
- Strongly developing field of **Active Learning**

Quantum engineering: An example

Quantum Physics
Electronic
Structure

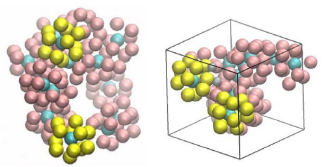


Properties of
Materials
and
Processing

Simplified Hamiltonian
Structure Matching
Force Matching

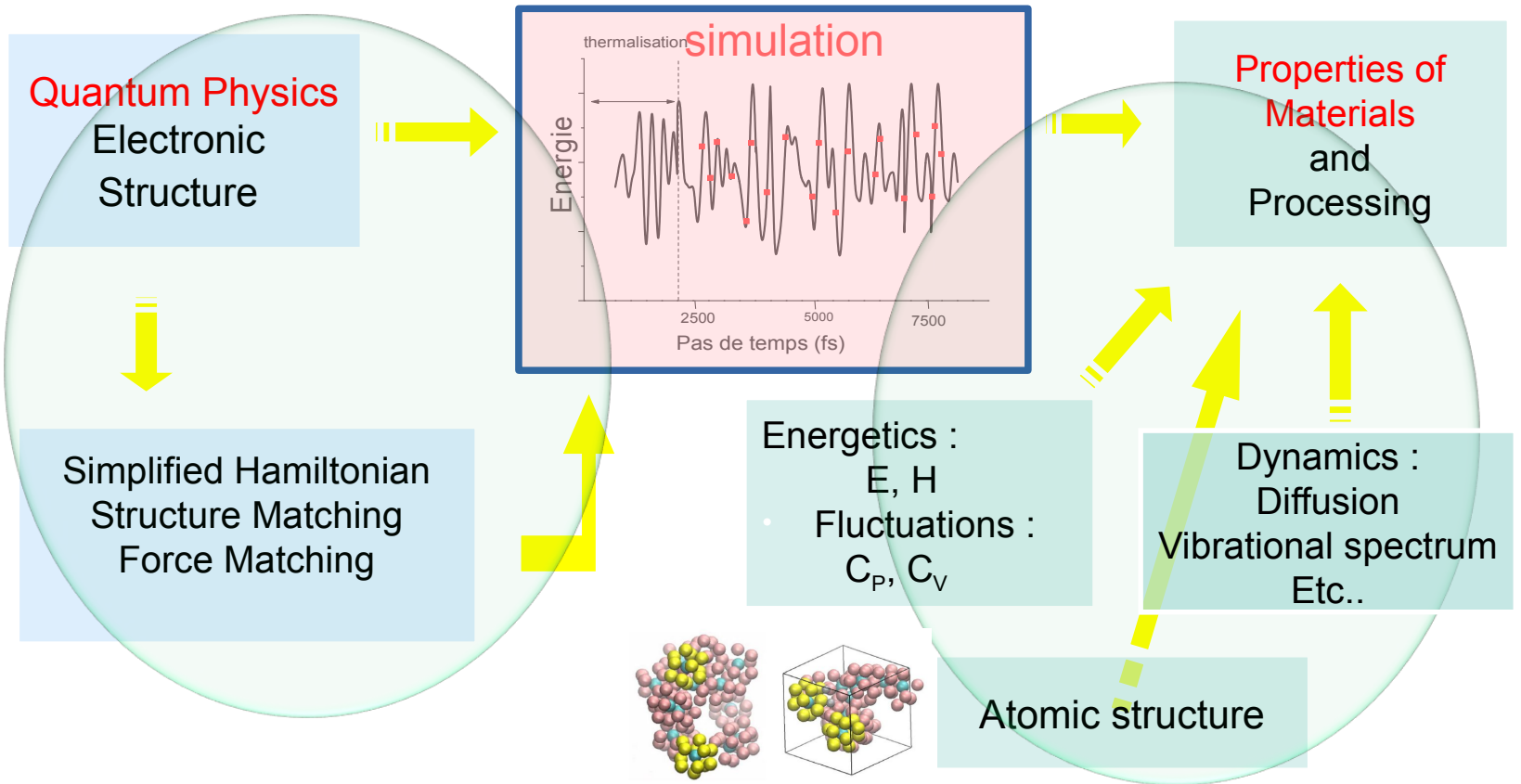
Energetics :
E, H
Fluctuations :
C_P, C_V

Dynamics :
Diffusion
Vibrational spectrum
Etc..



Atomic structure

Quantum engineering: An example

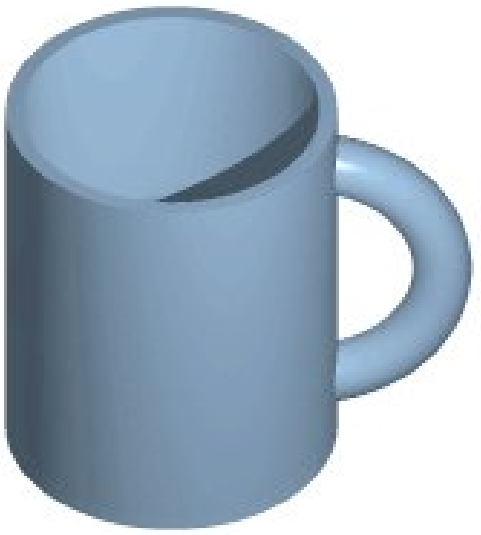


Estimating the forces using ML

Predicting properties using ML

Topological Learning for Structure Identification

- Unsupervised Approach
  - Learn to find interesting structures without prior reference
- Topological information as descriptor
  - Properties which do not change during stretching
  - Tools from Topological Data Analysis to describe topology of local structure



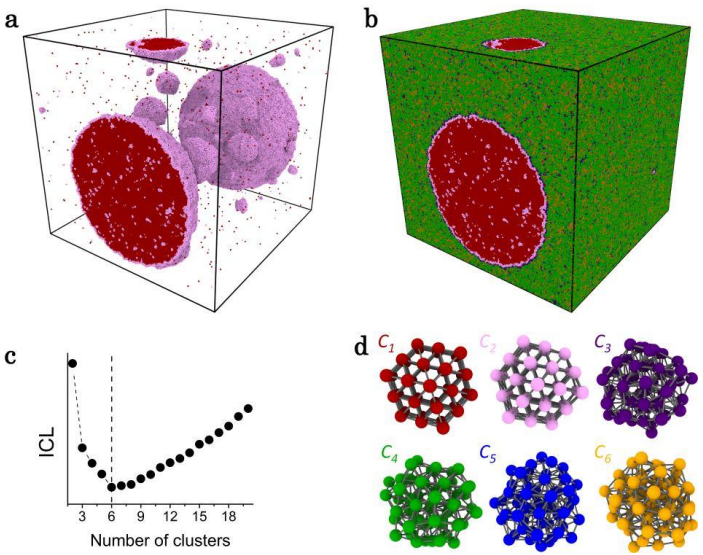
https://commons.wikimedia.org/wiki/File:Mug_and_Torus_morph.gif

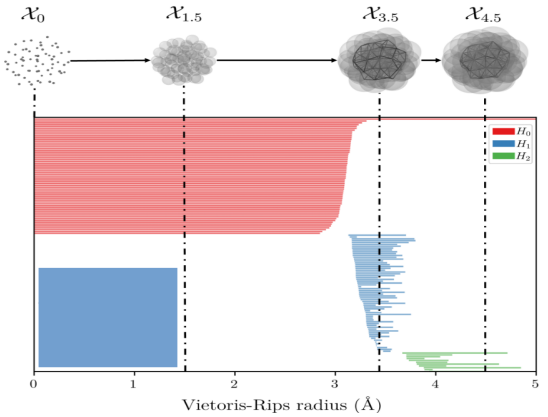
Unsupervised ML : topological learning approach

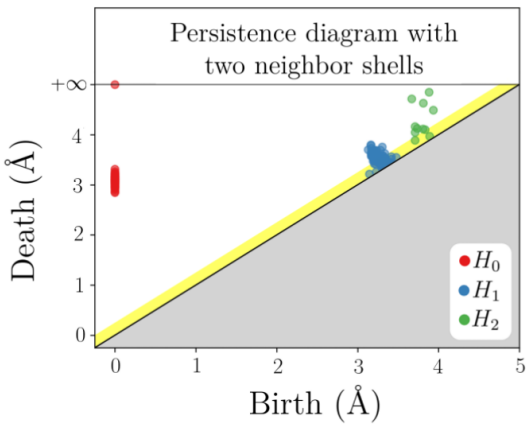
Encode local atomic structures: Persistence Homology from topological data analysis (TDA)

Unsupervised learning approach: Gaussian Mixture Model

Pure Tantalum at 1900 K, 10 million atoms







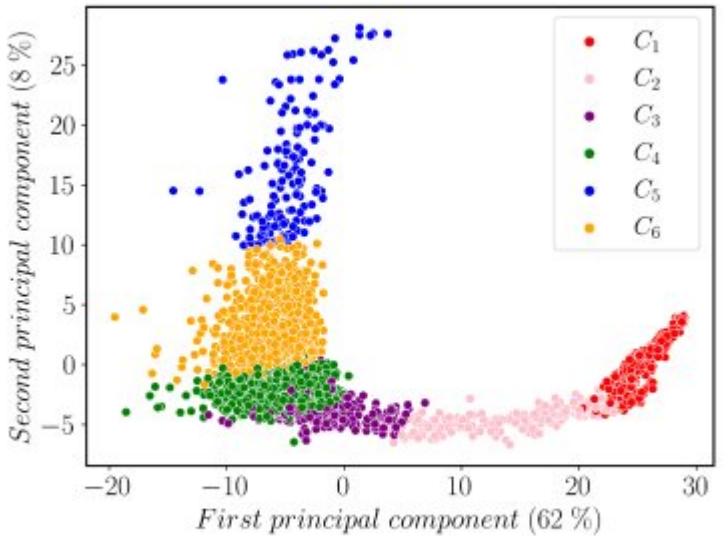
$$m_D(x, y) = \min\{\|x - y\|_\infty, d_\Delta(x), d_\Delta(y)\}$$

Topological vector (H0,H1,H2): descriptor (~ 200 components)

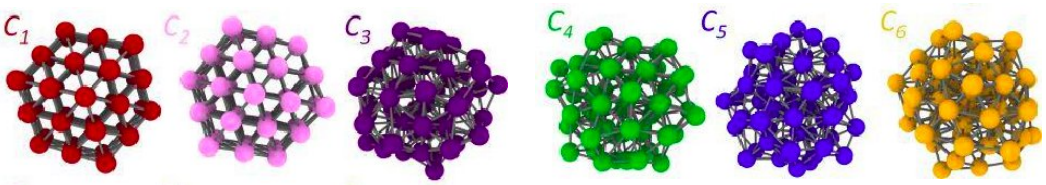
S. Becker et al., Sci Rep. (Nature), **12**, 3195 (2022), PRE **105** 045304 (2022);
S. Becker et al. Neurips (2021)

Unsupervised ML : topological learning approach

Tantalum as a test case: assessment of the clustering and dimensionality reduction



Principal component analysis of the clusters

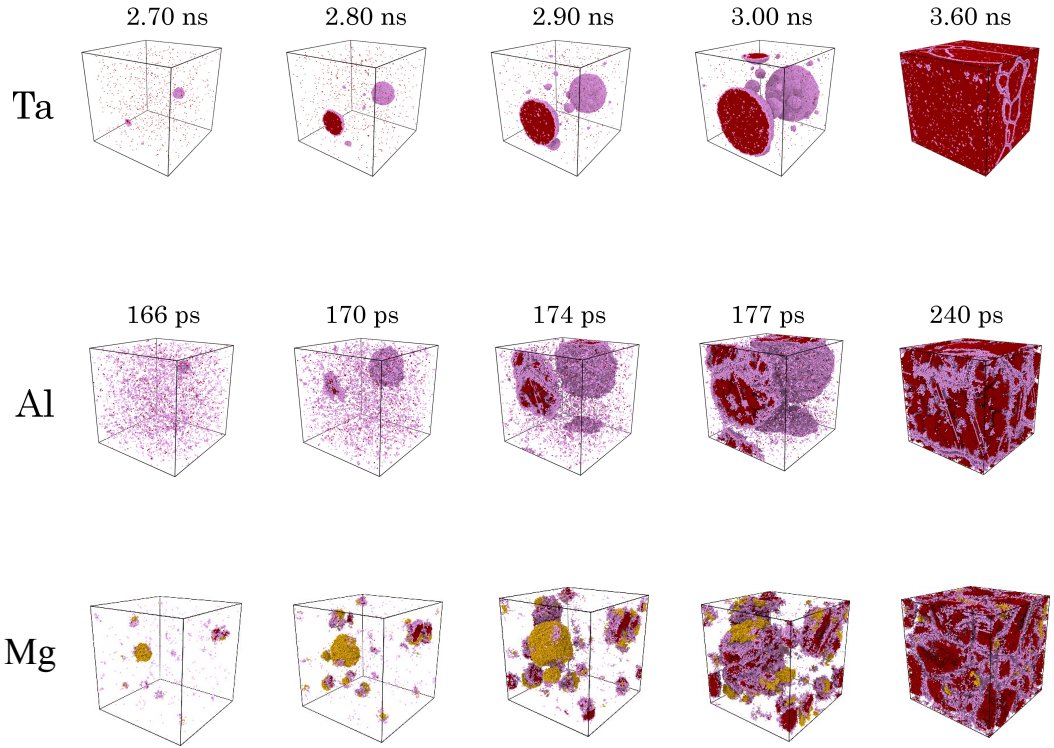


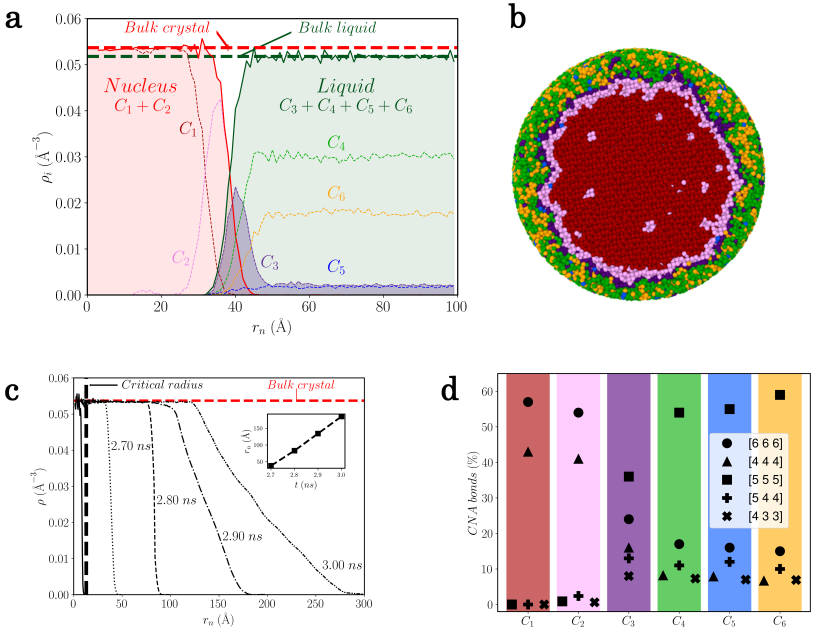
- Clear distinction of most of the clusters
- C_3 plays particular role
- robust for several metals investigated with various underlying crystal structures(Al,Mg,Zr)

S. Becker et al., Sci Rep. (Nature), 12, 3195 (2022), PRE (2022)
S. Becker et al. Neurips (2021)

Unsupervised ML : topological learning approach

Results : morphologies and nucleation pathways





S. Becker et al., Sci Rep. (Nature), **12**, 3195 (2022), PRE **105** 045304 (2022);
S. Becker et al. Neurips (2021)