

Algorithms for Unsupervised Learning

Dimensionality Reduction and
Density Estimation

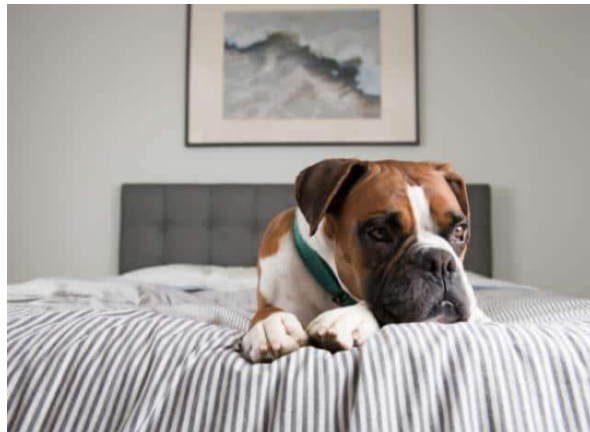
Algorithms for Unsupervised Learning

- Manifold Learning (Dim. Red. & ID est.):
 - PCA
 - K-PCA
 - ISOMAP
 - t-SNE
 - Autoencoders
- Density Estimation:
 - Histograms
 - Kernel Density Estimation
 - k-Nearest Neighbor
 - Generative Adversarial NN
- Clustering:
 - k-means/c-means, kernel k-means, spectral clustering...
 - Hierarchical clustering
 - Density Based clustering
 - Self Organizing Maps

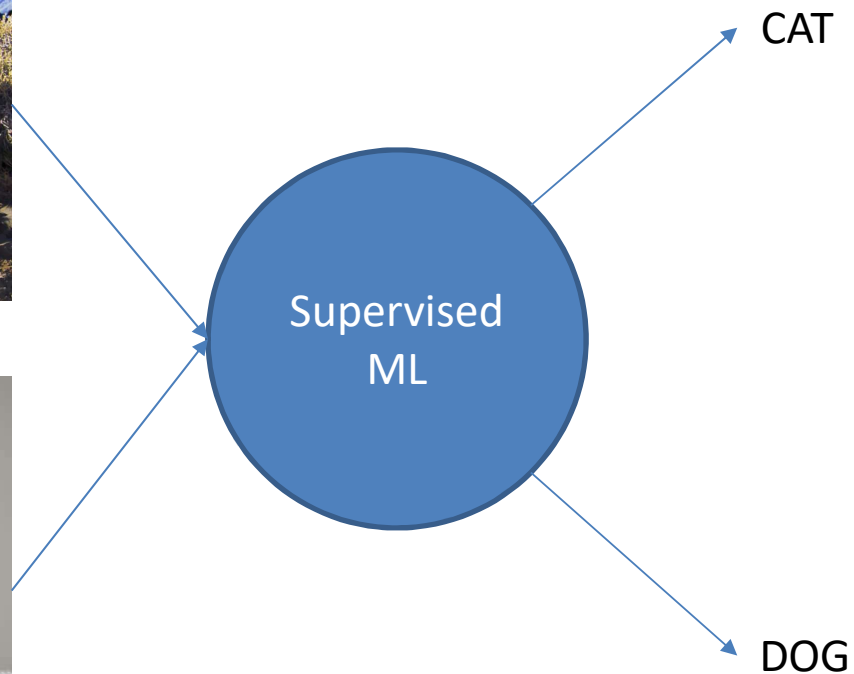
Dimensionality reduction and Intrinsic Dimension estimation.

MANIFOLD LEARNING METHODS

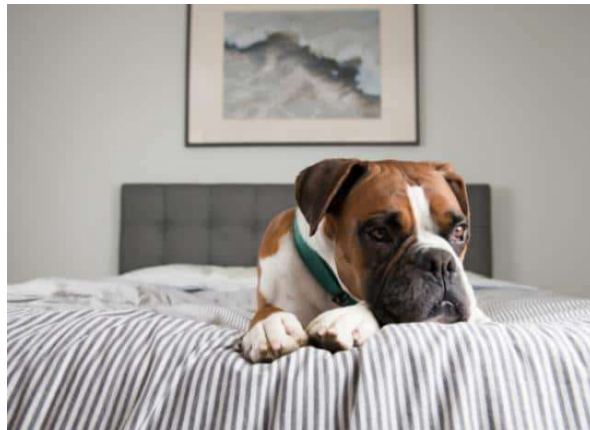
A view from supervised learning



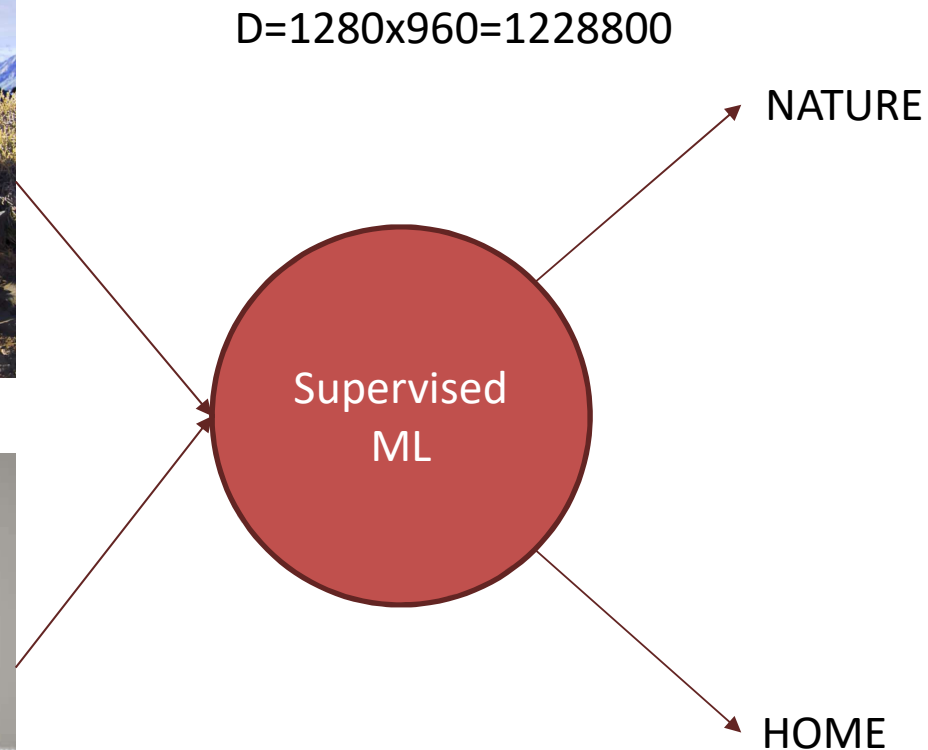
$D=1280 \times 960 = 1228800$



A view from supervised learning



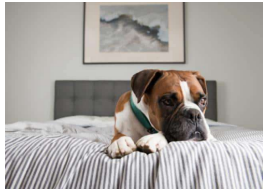
$D=1280 \times 960 = 1228800$



A view from supervised learning

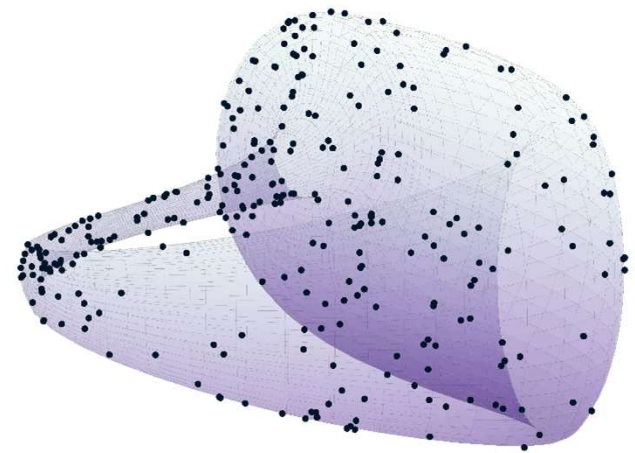
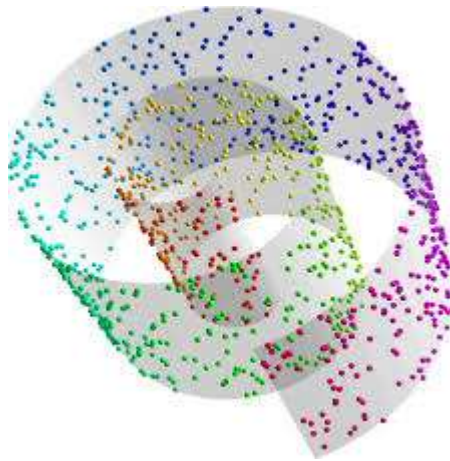
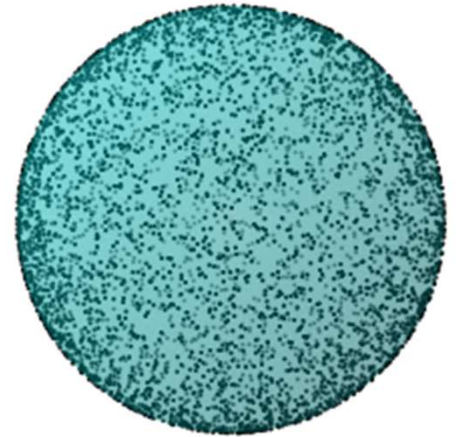
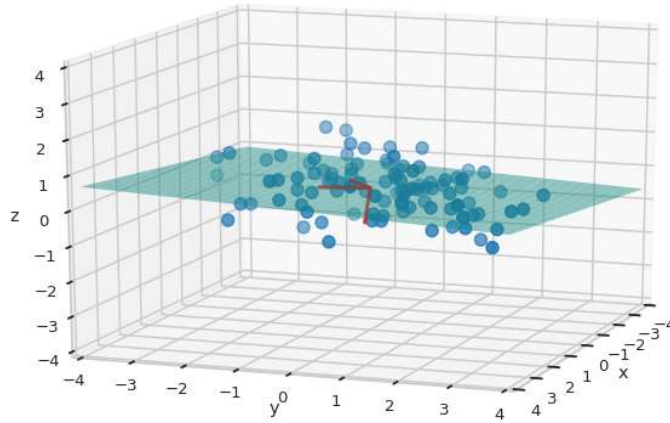
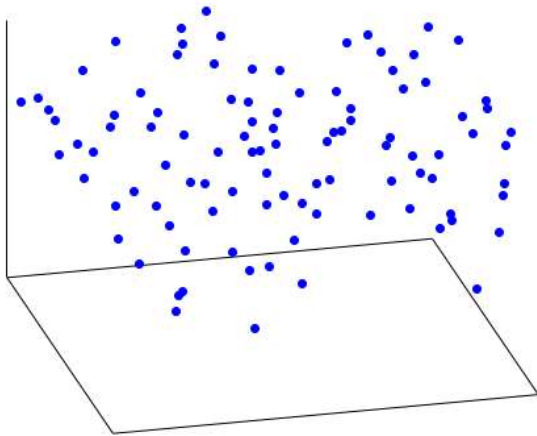


$$D=1280 \times 960 = 1228800$$

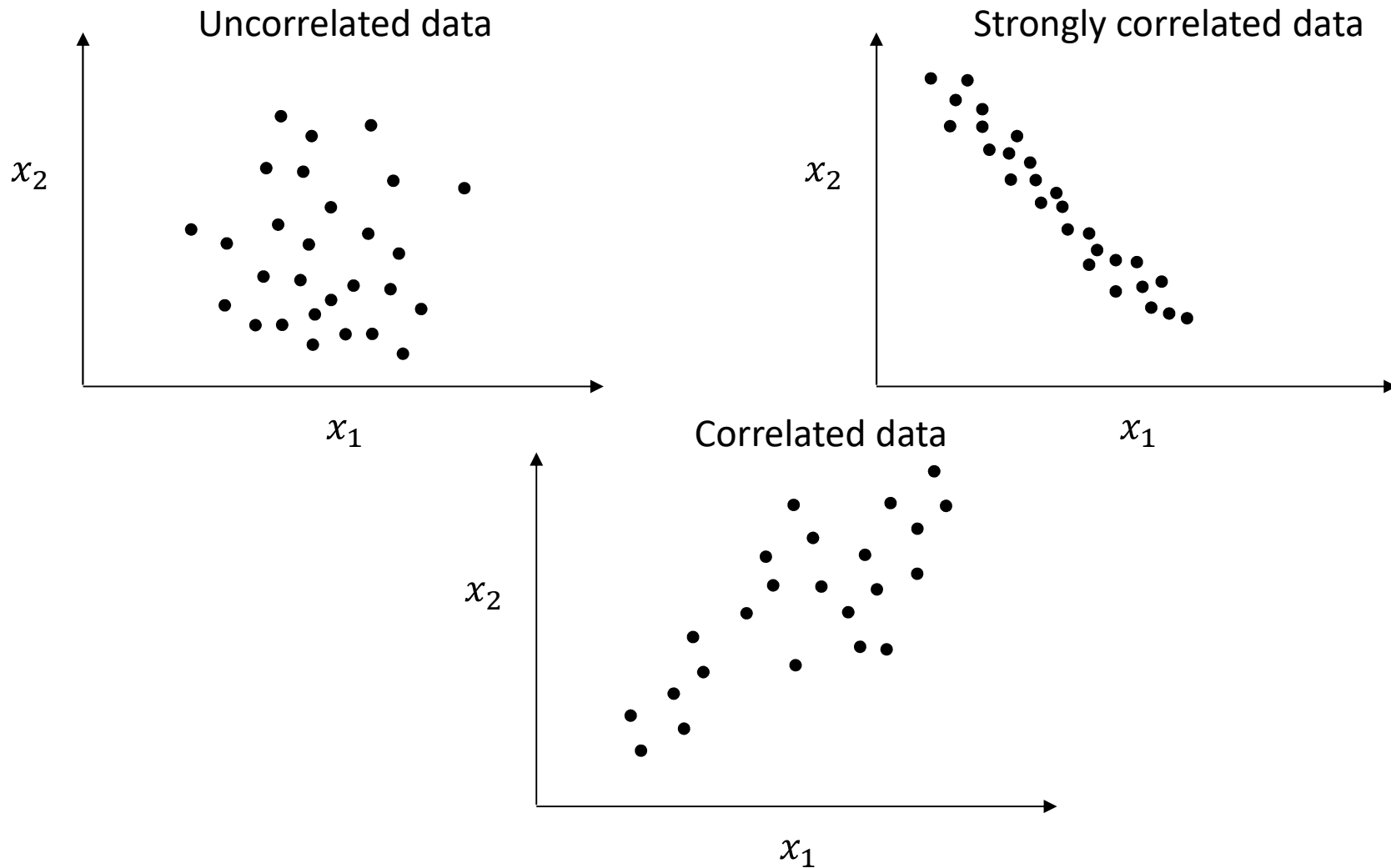


The data lay in a manifold whose dimension is the total number of INDEPENDENT classification tasks that can be performed.

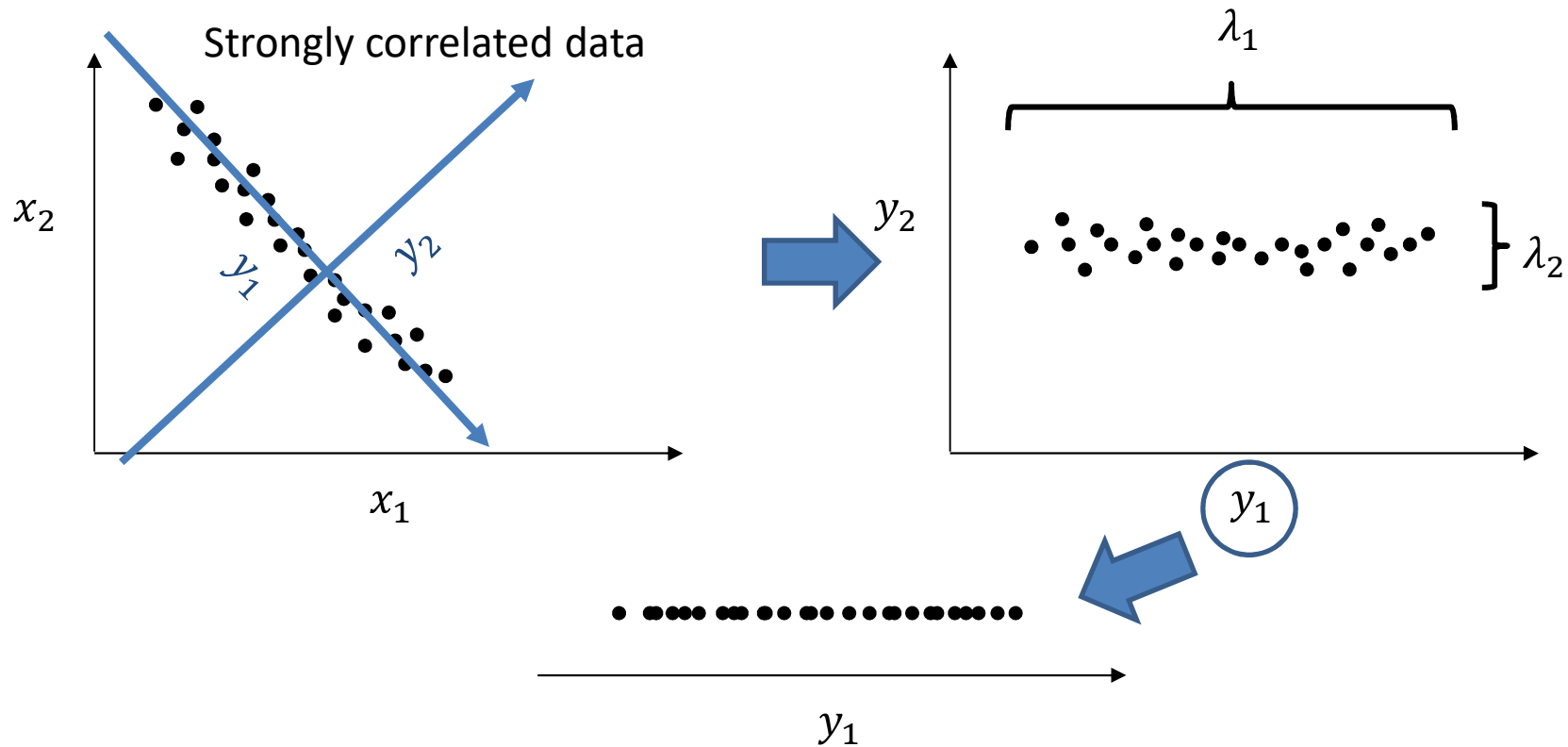
MANIFOLDS



STRONG CORRELATIONS REDUCE DIMENSIONALITY



What is the task that we want to perform?



NOTE: We just need a rotation of the space that reduces (cancels) covariances

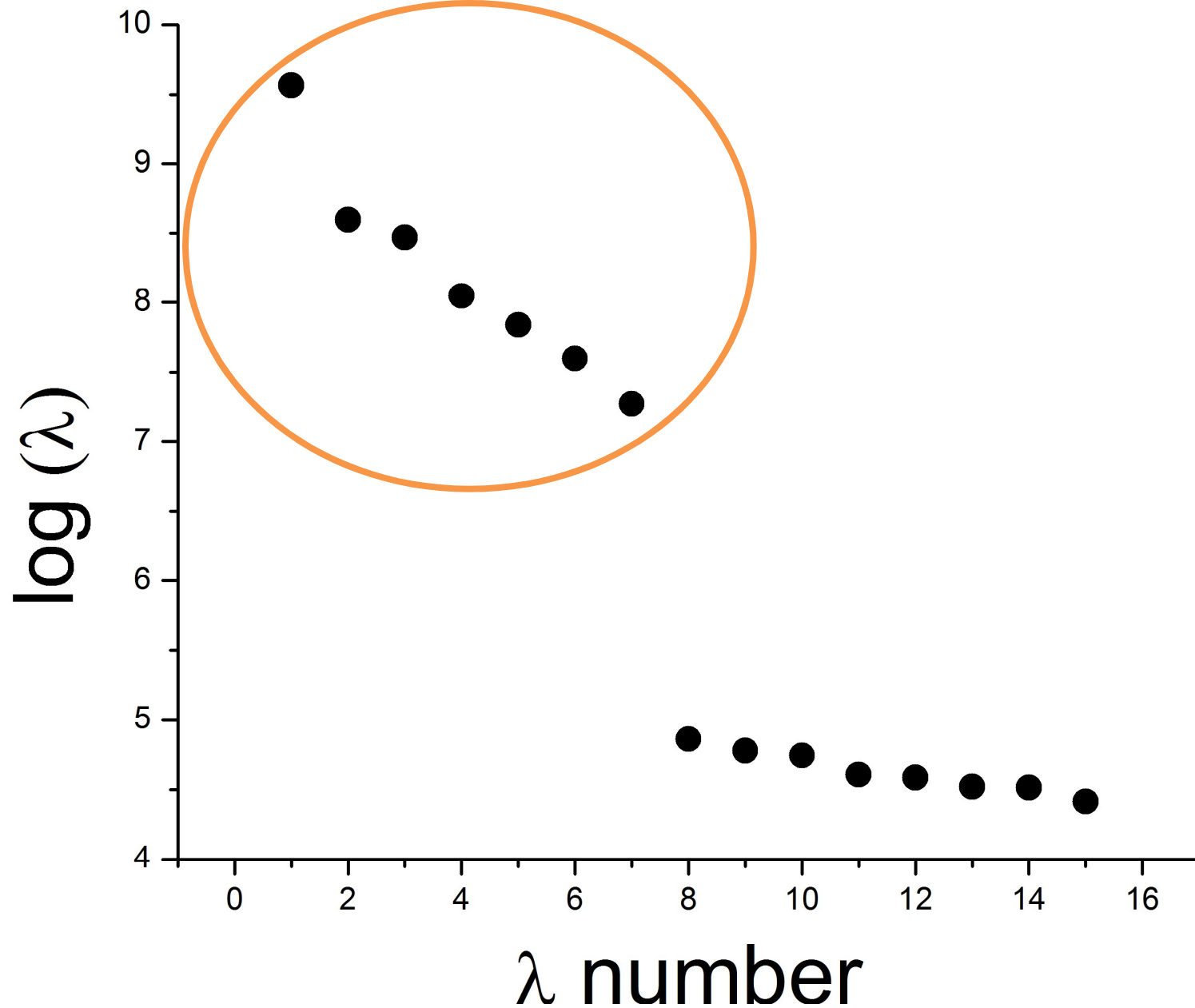
DIAGONALIZATION OF THE COVARIANCE MATRIX

PCA. How to...

Given n observations $\{x_1, x_2, \dots, x_n\}$ of m -dimensional column vectors

1. Compute the mean vector $\mu = \frac{1}{n} \sum_{i=1}^n x_i$
2. Compute the covariance matrix by MLE $\mathbb{C} = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T$
3. Compute the eigenvalue/eigenvector pairs (λ_i, u_i) of \mathbb{C} with $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_m$ (Diagonalize/SVD) and sort.
4. Choose the number of dimensions (d) in which project the data
5. Project by $y_i^{(j)} = x_i^T u_j$

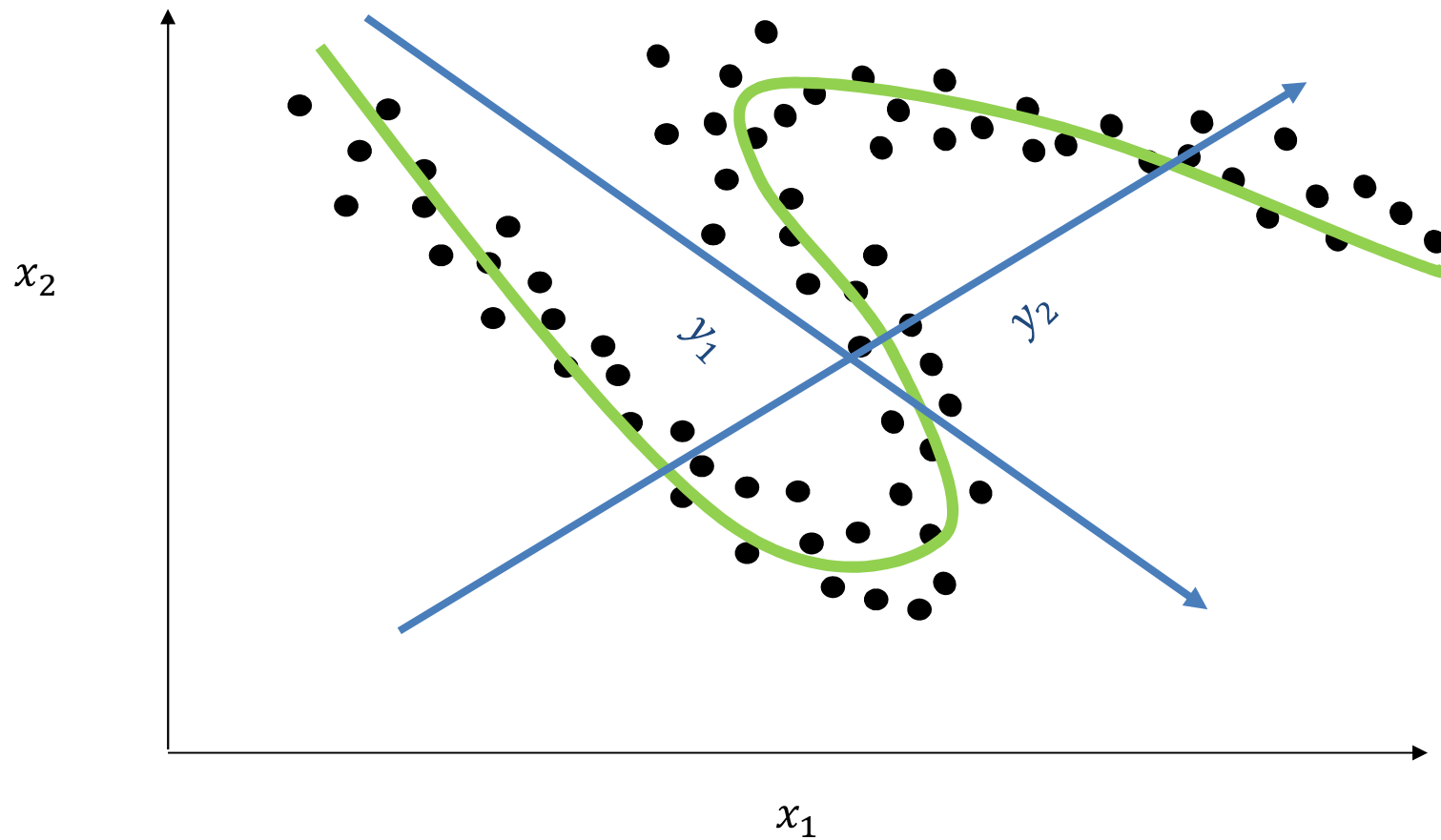
Choosing the number of PC's



PCA problems

- Poorly suited for non-linear transformations

Addressing the non-linear problem

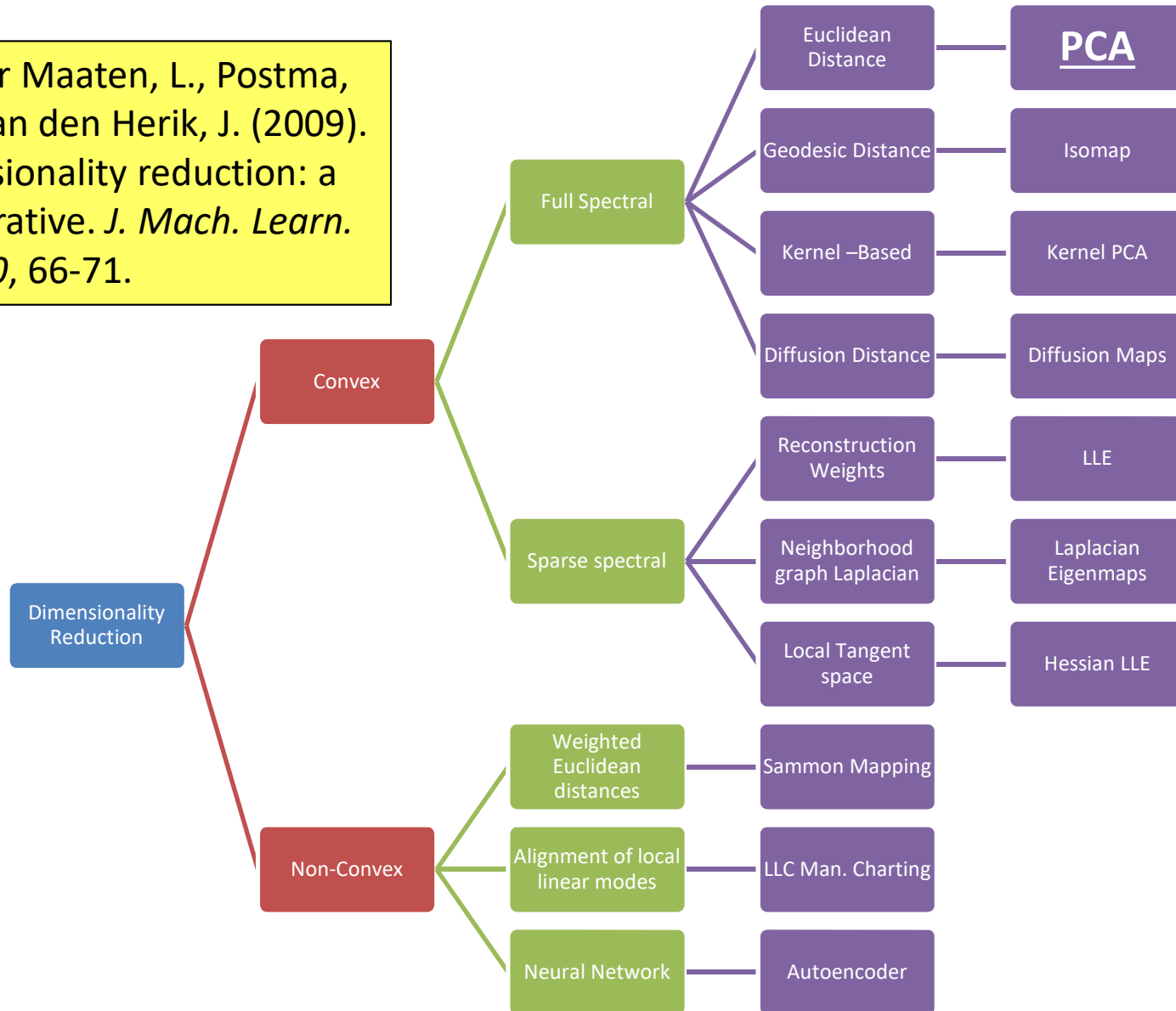


PCA problems

- Poorly suited for non-linear transformations
- Not able of capturing invariances
- By using the covariance along the whole dataset, is poor suited for problems not well described by this parameter.
- Not scale invariant
- Focused on large pairwise distances

Beyond PCA

Van Der Maaten, L., Postma, E., & Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J. Mach. Learn. Res.*, 10, 66-71.



Focusing on preserving the distances: Multidimensional Scaling

Δ^{ij} Distance in the ambient space

δ^{ij} Distance in the projected space

Stress: Measure of the discrepancy between both distances.
We try to minimize it.

Focusing on preserving the distances: Multidimensional Scaling

$$S = \sum_{i,j} \left((\Delta^{ij})^2 - (\delta^{ij})^2 \right)^2$$

Classical MDS: Formally equivalent to PCA when using Euclidean Distance.
Analytical minimization.

$$S = \frac{\sum_{i,j} \left((\Delta^{ij}) - (\delta^{ij}) \right)^2}{\sum_{i,j} \delta^{ij}}$$

Metric MDS: Preserves the relative importance of the distances.
Numerical minimization.

$$S = \frac{\sum_{i,j} \left(g(\Delta^{ij}) - f(\delta^{ij}) \right)^2}{\sum_{i,j} f(\delta^{ij})}$$

Non-metric MDS: You choose what to preserve. Numerical minimization.

Classical MDS

1.- Obtain the Gram matrix

$$\hat{G}^{ij} = -\frac{1}{2} \left(\Delta^{ij} - \frac{1}{N^2} \sum_{i,j} \Delta^{ij} - \frac{1}{N} \left(\sum_i \Delta^{ij} + \sum_j \Delta^{ij} \right) \right)$$

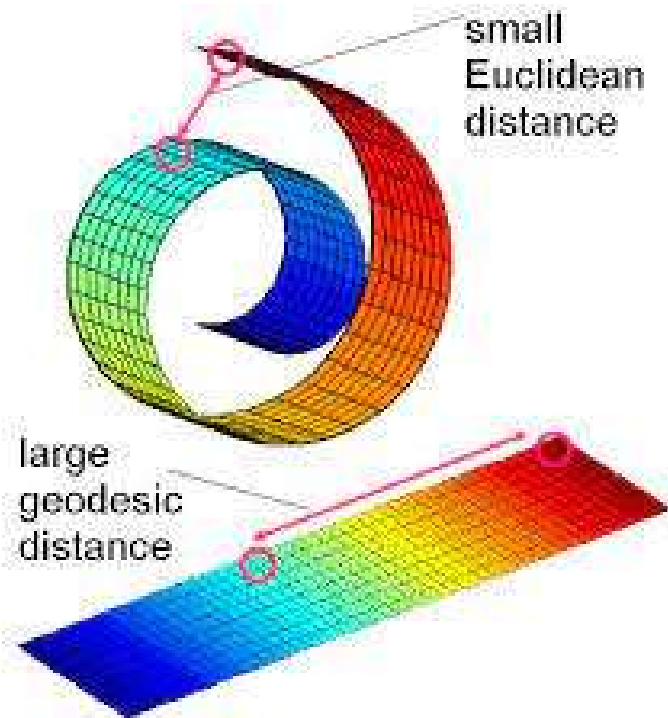
This is called double centering.

2.- Diagonalize it

3.- From the spectrum, decide d

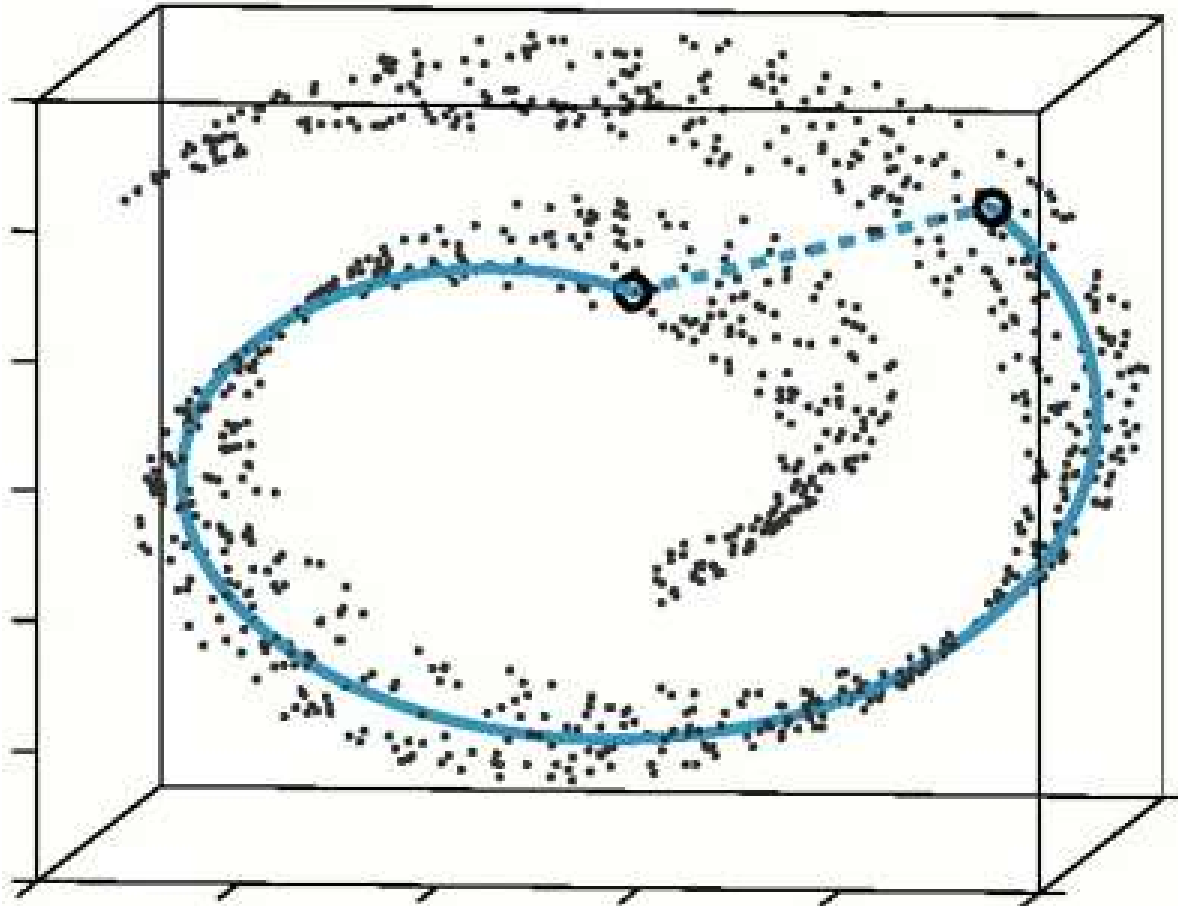
Note that this procedure does not requires Δ^{ij} being computed as the Euclidean distance!!!

ISOMAP



Uses classical MDS setting Δ^{ij} equal to the geodesic distance instead of the Euclidean distance.

Geodesic distance



Geodesic distance

- Determine the neighbors.
 - All points in a fixed radius.
 - K nearest neighbors
- Construct a neighborhood graph.
 - Each point is connected to the other if it is a K nearest neighbor.
 - Edge Length equals the Euclidean distance
- Compute the shortest paths between two nodes

Geodesic distance: Naïve algorithm

1. Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes.
2. Set the initial node as current. Mark all other nodes unvisited. Create a set of all the unvisited nodes called the unvisited set.
3. For the current node, consider all of its unvisited neighbors and calculate their tentative distances. Compare the newly calculated tentative distance to the current assigned value and assign the smaller one. For example, if the current node A is marked with a distance of 6, and the edge connecting it with a neighbor B has length 2, then the distance to B (through A) will be $6 + 2 = 8$. If B was previously marked with a distance greater than 8 then change it to 8. Otherwise, keep the current value.
4. When we are done considering all of the neighbors of the current node, mark the current node as visited and remove it from the unvisited set. A visited node will never be checked again.
5. If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the unvisited set is infinity (when planning a complete traversal; occurs when there is no connection between the initial node and remaining unvisited nodes), then stop. The algorithm has finished.
6. Otherwise, select the unvisited node that is marked with the smallest tentative distance, set it as the new "current node", and go back to step 3.

Kernel PCA

- Apply a kernel function (non-linear transformation) to the distances in such a way that we only weight those that we are interested in.
- For instance: You are only interested in distances within a certain radius σ . You can

use: $\widetilde{\Delta}_{ij} = 1 - e^{-\frac{\Delta_{ij}}{\sigma}}$

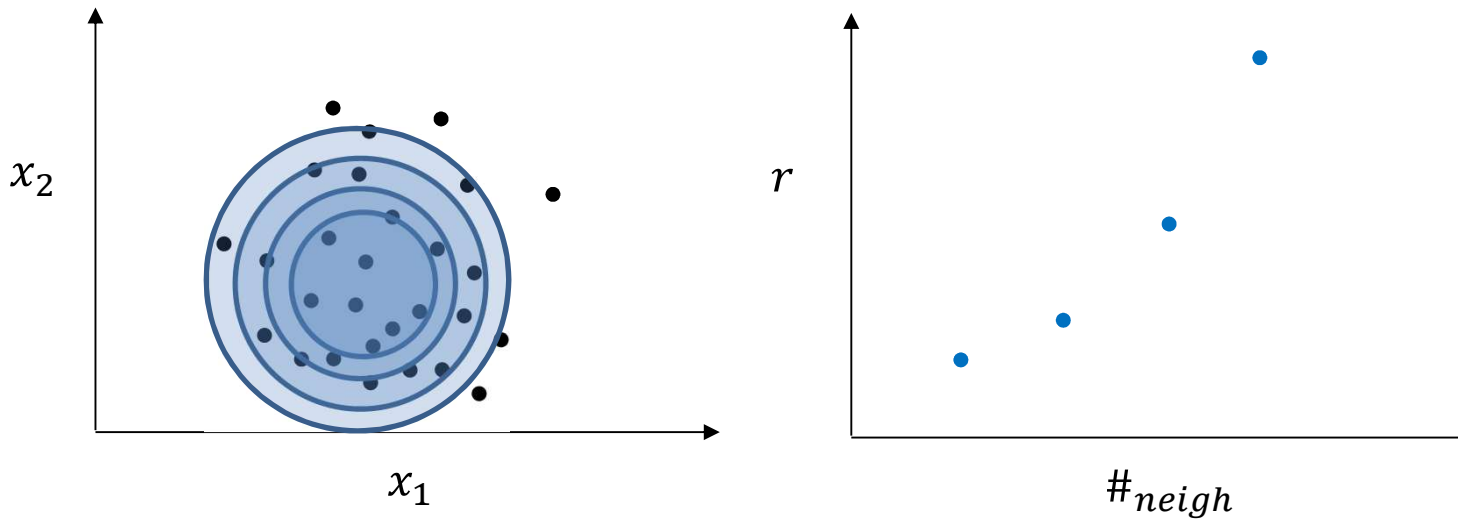
t-SNE or UMAP

- These methods aim to reproduce the probabilities of neighborhood between data points.
- Optimize a loss function that measures the overlap between these probabilities in the ambient and projection spaces.
- Generally excellent performance for visualization, take care when using them for pre-processing.
- Random initialization, not always reproducible results.

Directly estimation of the Intrinsic dimension

Minimum number of parameters required to describe the data while minimizing the information loss.

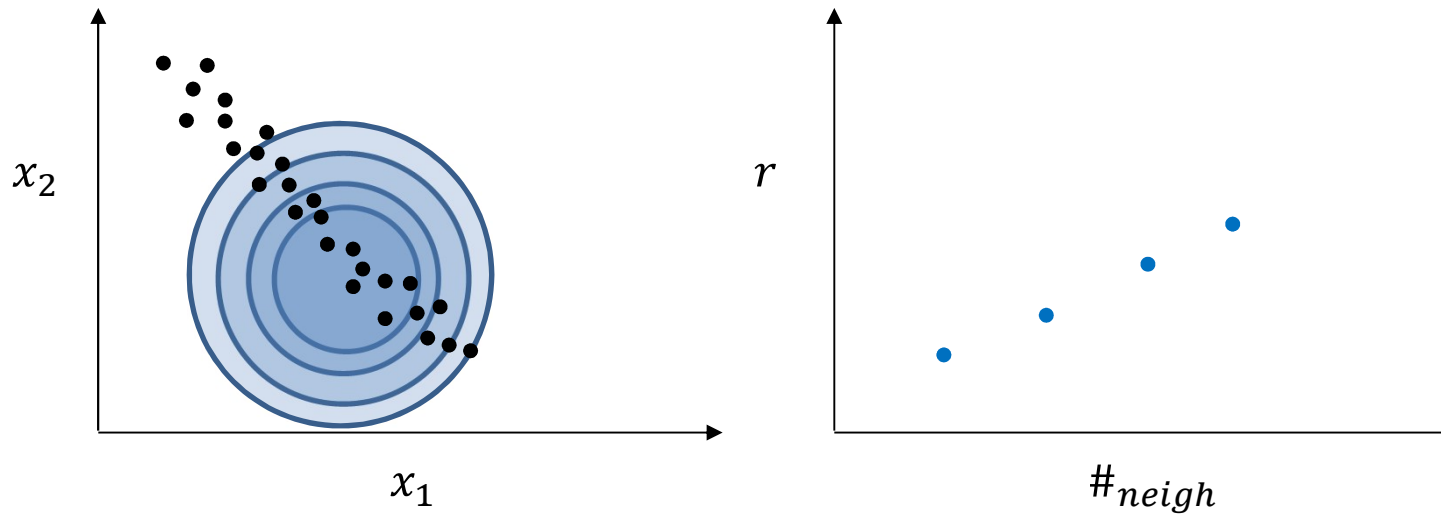
Classical approach $\#_{neigh} = \rho r^d$



Directly estimation of the Intrinsic dimension

Minimum number of parameters required to describe the data while minimizing the information loss.

Classical approach $\#_{neigh} = \rho r^d$



Directly estimation of the Intrinsic dimension

In practice:

$$\#_{neigh} = \rho r^d \longrightarrow \log(\#_{neigh}) = \log \rho + d \log r \quad \text{Linear fit}$$

Only true at constant density:
Modern methods try to
disentangle both measures

TWO-NN

$$\mu_i = \frac{r_{i,2}}{r_{i,1}};$$

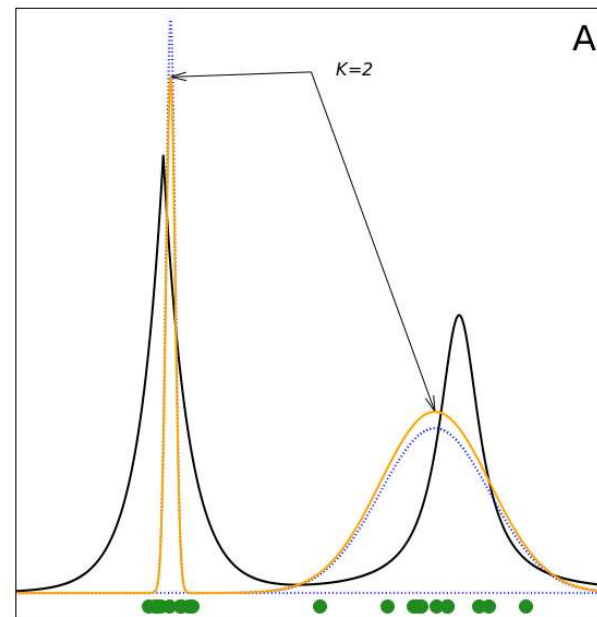
$$P(\mu|\rho) = P(\mu) = \frac{d}{\mu^{1+d}}$$

DENSITY ESTIMATION

Gaussian mixture models

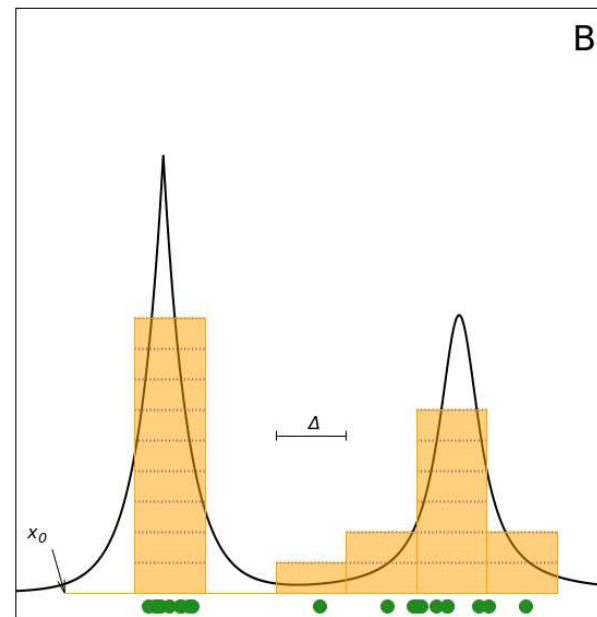
- Parametric approach
(We enforce a functional form to $p(x)$)

- $p(x) = \sum_{i=1}^K w_i e^{-\left(\frac{x-\mu_i}{s_i}\right)^2}$
- The only parameter is K



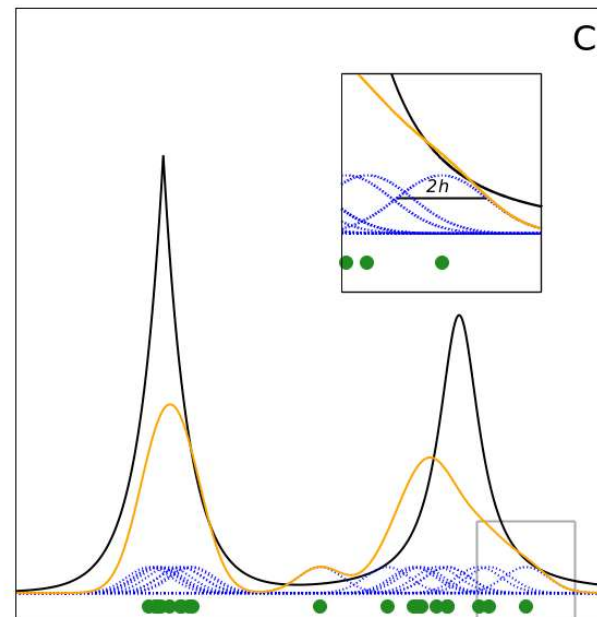
Histograms

- Non-parametric approach
(We don't enforce a functional form to $p(x)$)
- Count the number of points within a binwidth Δ
- Two parameters: Δ & x_0
- Poor performance at $d > 2-3$



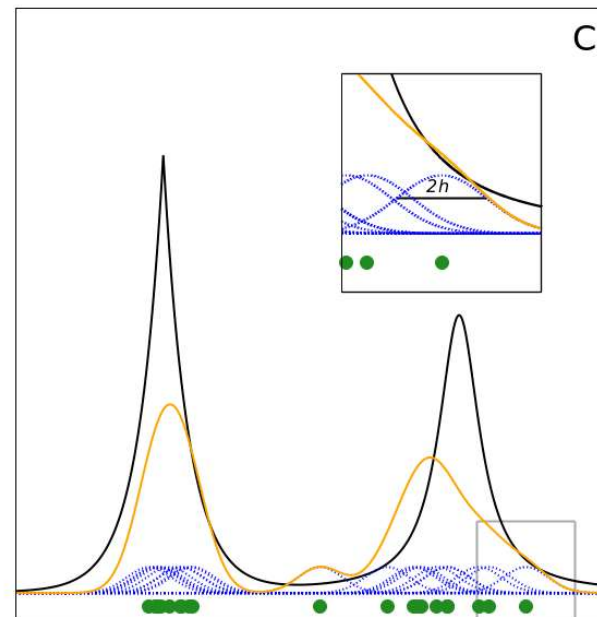
Kernel density estimation

- Non-parametric approach
(We don't enforce a functional form to $p(x)$)
- $$p(x) = \sum_i \frac{1}{h\sqrt{2\pi}} e^{-\left(\frac{x-x_i}{h}\right)^2}$$
- Puts a gaussian in the top of each point, then sum.
- Single parameter: h



K-Nearest neighbor density estimation

- Non-parametric approach (We don't enforce a functional form to $p(x)$)
- $p(x) = \frac{k}{r_k^d}$
- The density at a point is the inverse of k times the volume occupied by its k nearest neighbors.
- Single parameter: k



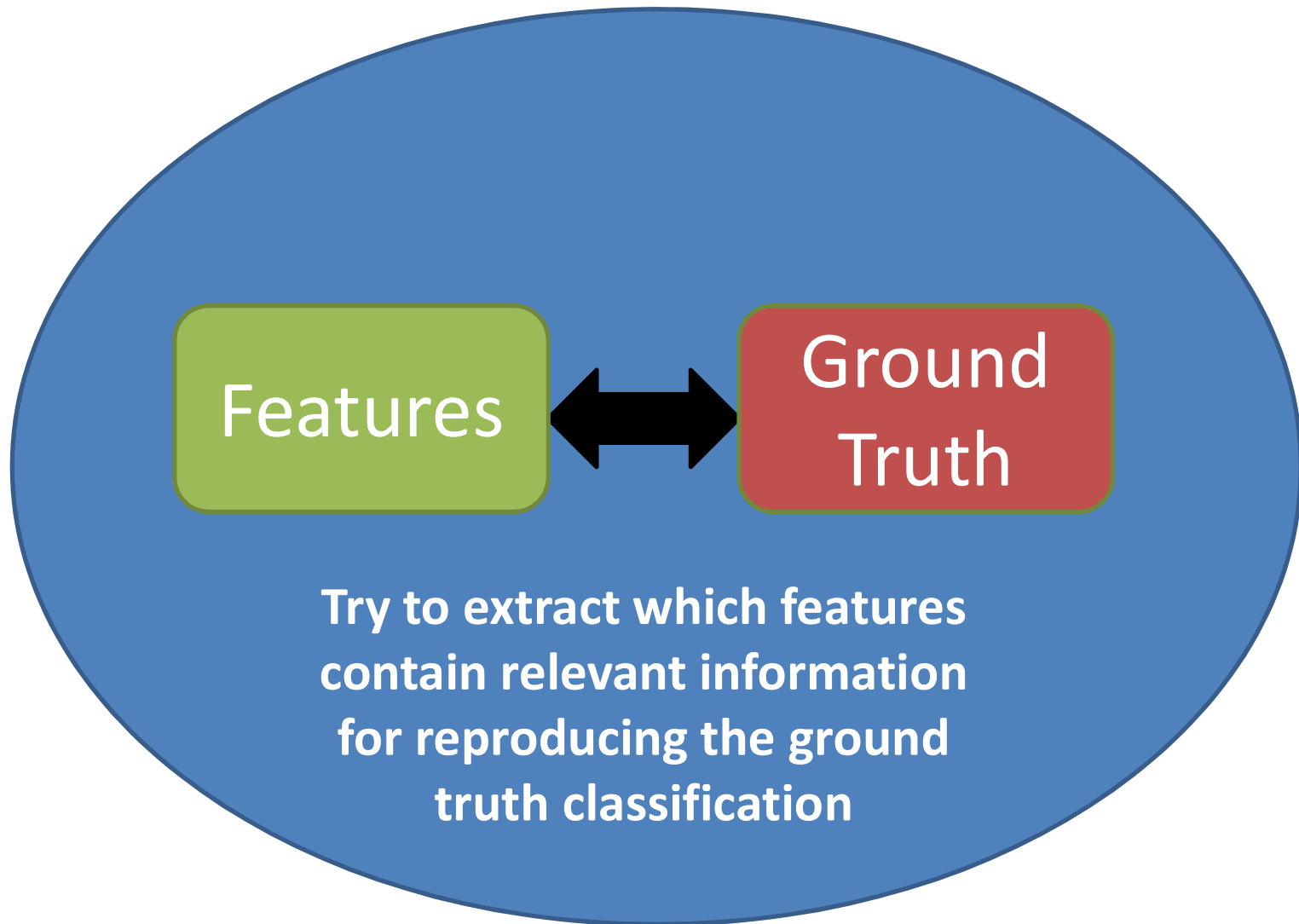
Feature Selection

Master in High Performance
Computing

Feature Selection & Dimensionality reduction

- Highly related with the problem that we want to solve
- It may need feedback from the whole process
- Feature selection usually depends on labeled data while dimensionality reduction does not.
- Feature selection can be based on expertise...

Feature selection



My stone collection (3)



Metálico	1.-Talco	6.- Ortosia	Azufre	Magnetita	Escamosa	Blanco	Negro	Naranja
Vítreo	2.- Yeso	7.- Cuarzo	Aragonito	Galena	Concoidea	Gris	Amarillo	Transparente
Graso	3.- Calcita	8.- Topacio	Rejalgar	Cinabrio	Lisa Yeso Especular	Marrón	Granate	Rojo
Adamantino	4.- Fluorita	9.- Corindón	Azurita	Mercurio	Fibrosa	Dorado	Verde (opaco)	Verde (trasp.)
Anacarado	5.- Apatito	10.- Diamante	Malaquita	Oro	Rosetas	Morado	Azul	Combinado

- Weight
- Light wavelength
- Shape
- Volume
- Rugosity
- ...

Of course, if you are an expert, you already know which are the relevant features... but, I'm quite dummy...

Techniques for Feature Selection (1)

Variable Ranking

- Which variables explain better the labels?
- Quantify it and sort by the score

Var ID	Score
A	0.9
B	0.1
C	0.2
D	0.7
E	0.03
F	0.85



Var ID	Score
A	0.9
F	0.85
D	0.7
C	0.2
B	0.1
E	0.03

Techniques for Feature Selection (1)

Variable Ranking

- Which variables explain better the labels?
- Quantify it and sort by the score
 - Linear correlation coefficient (R).

Techniques for Feature Selection (1)

Linear Correlation coefficient

$$R(i) = \frac{\sum_{k=1}^m (x_{k,i} - \bar{x}_i)(y_k - \bar{y})}{\sqrt{\sum_{k=1}^m (x_{k,i} - \bar{x}_i)^2 \sum_{k=1}^m (y_k - \bar{y})^2}}$$

- For continuous variables and outputs.
- Goodness of linear fit
- Easy to extend to linear fit of functions of variables (i.e. take log of x).

Techniques for Feature Selection (1)

Variable Ranking

- Which variables explain better the labels?
- Quantify it and sort by the score
 - Linear correlation coefficient (R).
 - Single variable classifier. Jaccard index, F-score, etc.

Techniques for Feature Selection (1)

Single variable classifier

Confusion matrix

R ↓ C→	Forest	Indust.	Urban	Water	Total
Forest	68	7	3	0	78
Indust.	12	112	15	10	149
Urban	3	9	89	0	101
Water	0	2	5	56	63
Total	83	130	112	66	391

- Based in the correspondence between the ground truth classification and the ones that comes from the single variable
- In some cases, requires labeling (assign the variable to a class).
- The confusion matrix can summarize some of them
- We will go in deeper detail when talking about external validation

Techniques for Feature Selection (1)

Variable Ranking

- Which variables explain better the labels?
- Quantify it and sort by the score
 - Linear correlation coefficient (R).
 - Single variable classifier. Jaccard index, F-score, etc.
 - Mutual information between variable and the target.

Techniques for Feature Selection (1)

Information Theoretic Ranking

$$I(i) = \int \int_{x_i, y} p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)p(y)}$$

- A kind of single variable classifier.
- In non-continuous variables, the integrals become sums
- Extensible to continuous variables by non parametric density estimation
- Using a Gaussian distribution for estimating the density will lead to a similar criteria to the correlation coefficient.
- Is a formalization of the intuition that the higher the joint distribution, the higher the mutual information, i.e. the higher should it be in the rank.

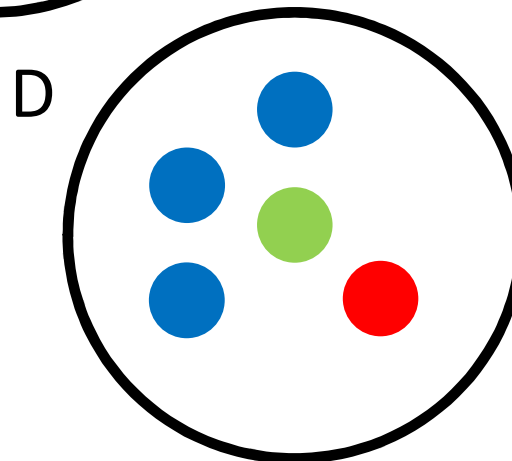
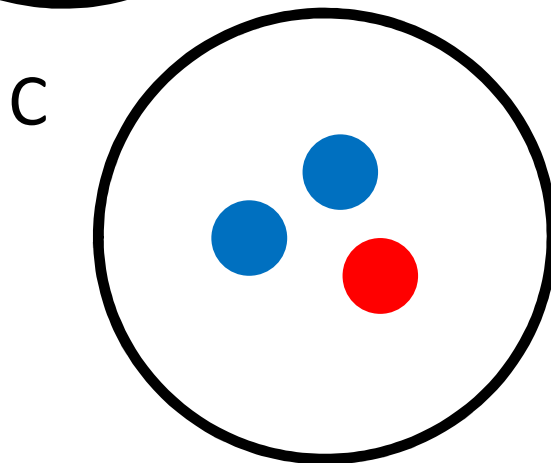
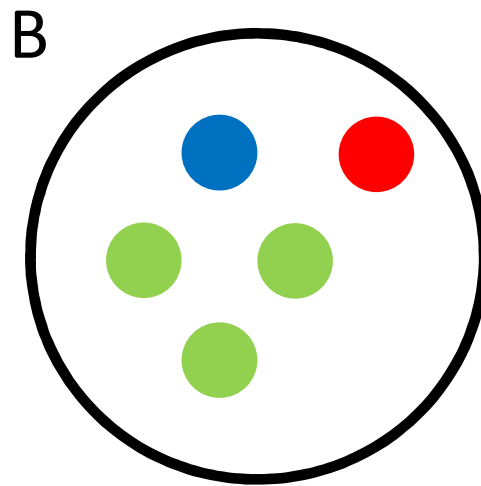
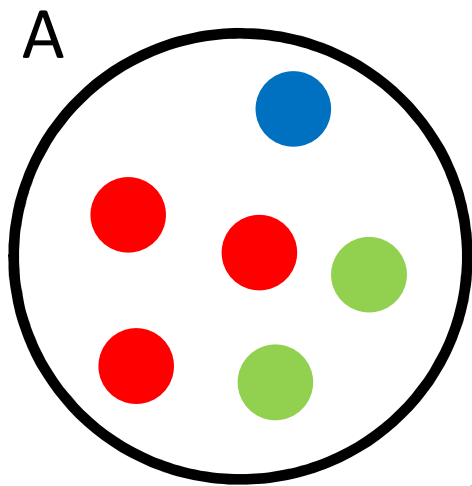
Techniques for Feature Selection (1)

Information Theoretic Ranking

- The probabilities, in a discrete case, are estimated from frequency counts.
- Imagine a three class problem (red, green, blue) with a discrete variable that can take 4 values (A,B,C,D).
 - $P(y)$ are 3 frequency counts.
 - $P(x)$ are 4 frequency counts.
 - $P(x,y)$ are 12 frequency counts.

Techniques for Feature Selection (1)

Information Theoretic Ranking



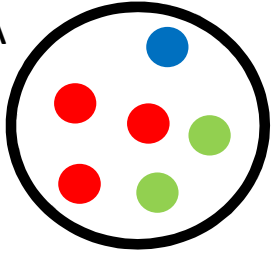
$$p(A, red) = \frac{3}{19}$$

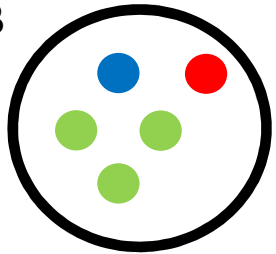
$$p(A) = \frac{6}{19}$$

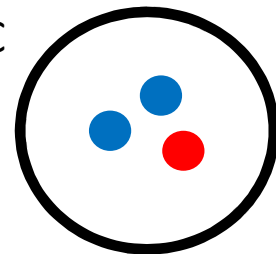
$$p(red) = \frac{6}{19}$$

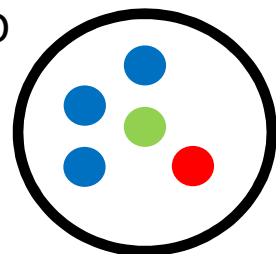
Techniques for Feature Selection (1)

Information Theoretic Ranking

A 

B 

C 

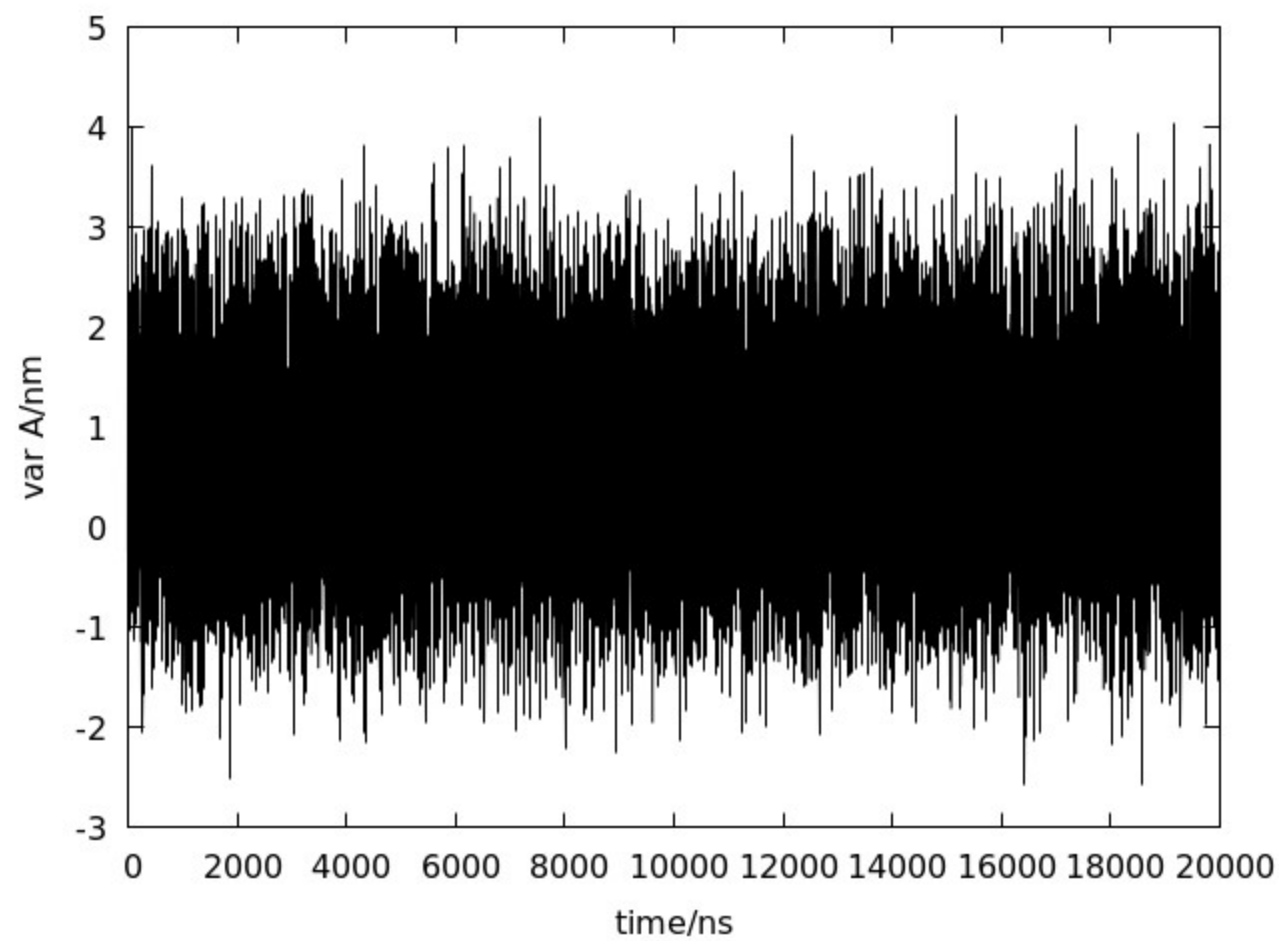
D 

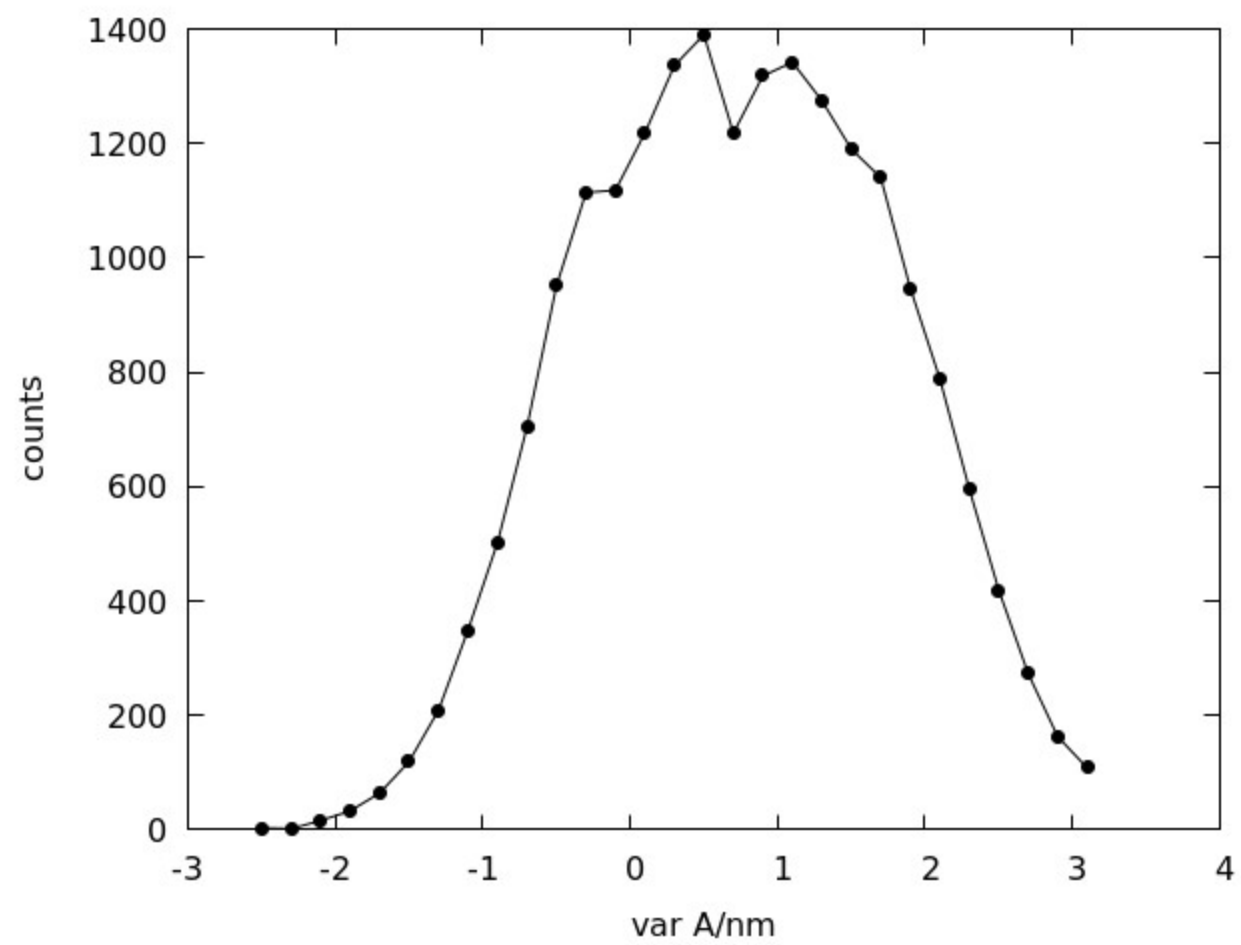
$$\begin{aligned}
 I = & \frac{3}{19} \log \left(\frac{3/19}{6/19 \cdot 6/19} \right) + \frac{1}{19} \log \left(\frac{1/19}{6/19 \cdot 7/19} \right) \\
 & + \frac{2}{19} \log \left(\frac{2/19}{6/19 \cdot 6/19} \right) + \frac{1}{19} \log \left(\frac{1/19}{5/19 \cdot 6/19} \right) + \\
 & \frac{1}{19} \log \left(\frac{1/19}{5/19 \cdot 7/19} \right) + \frac{3}{19} \log \left(\frac{3/19}{5/19 \cdot 6/19} \right) + \\
 & \frac{1}{19} \log \left(\frac{1/19}{3/19 \cdot 6/19} \right) + \frac{0}{19} \log \left(\frac{0/19}{3/19 \cdot 7/19} \right) + \\
 & \frac{2}{19} \log \left(\frac{2/19}{3/19 \cdot 6/19} \right) + \frac{1}{19} \log \left(\frac{1/19}{5/19 \cdot 6/19} \right) + \\
 & \frac{1}{19} \log \left(\frac{1/19}{5/19 \cdot 7/19} \right) + \frac{3}{19} \log \left(\frac{3/19}{5/19 \cdot 6/19} \right) \approx 0.17
 \end{aligned}$$

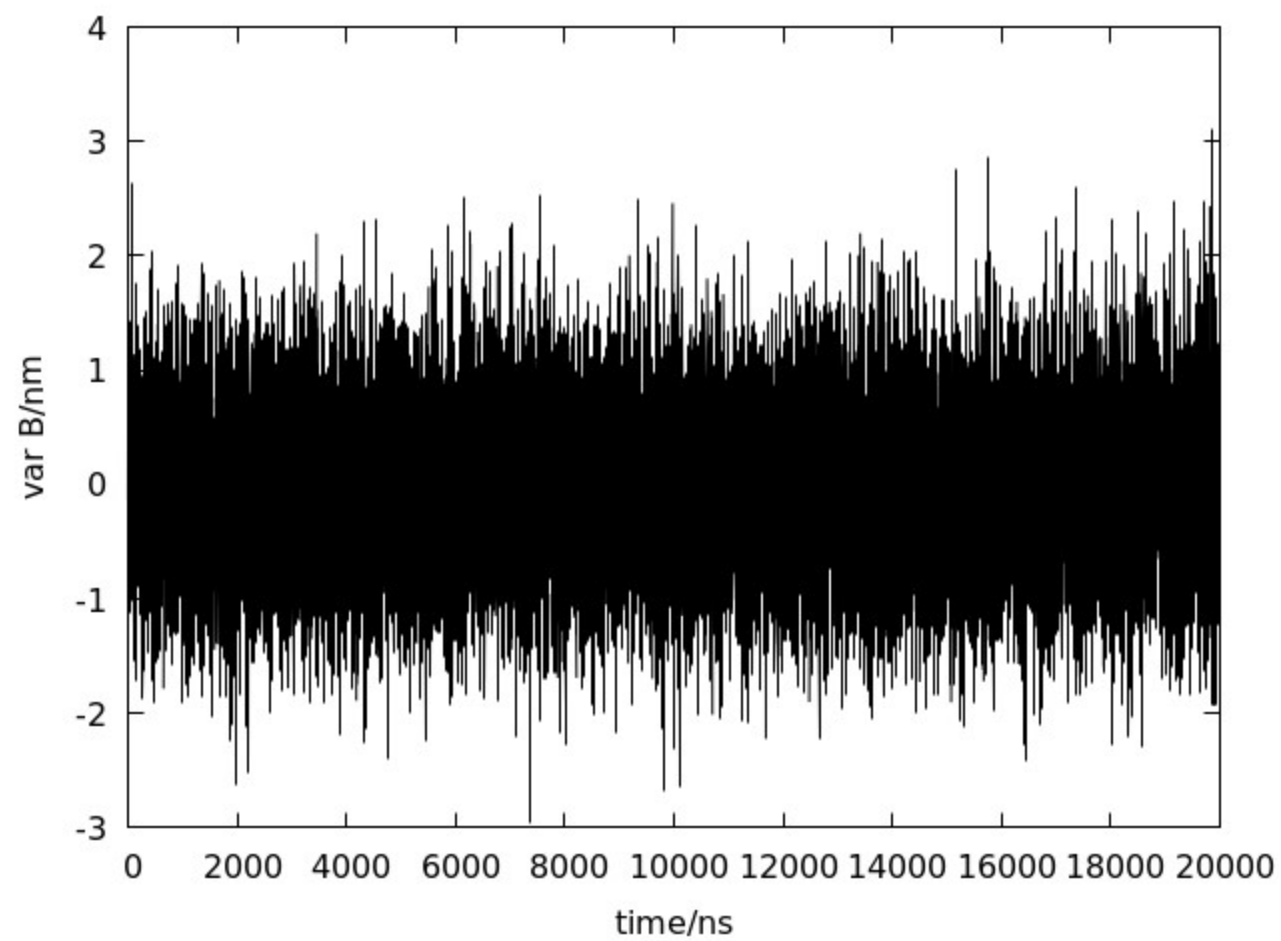
Techniques for Feature Selection (1)

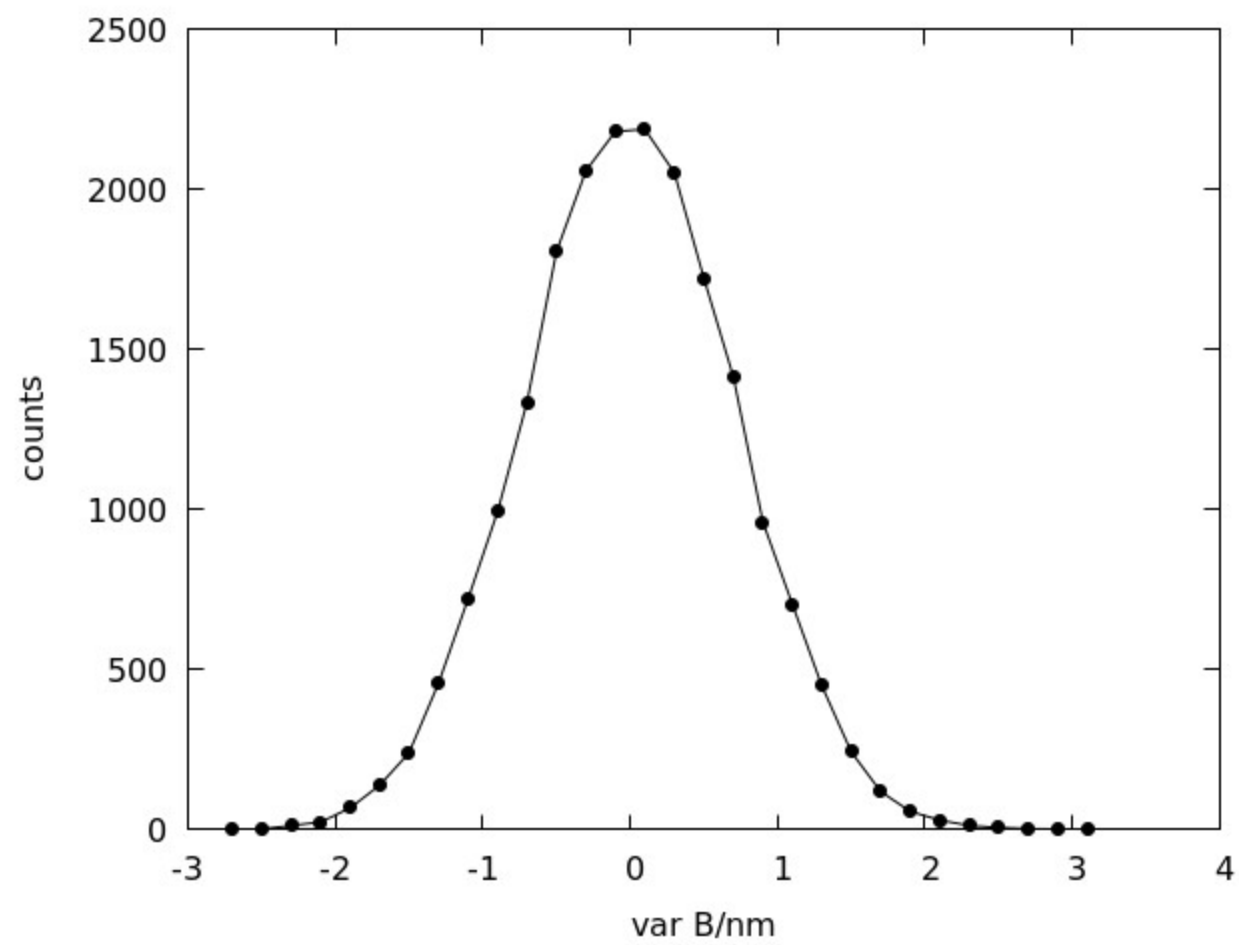
Some questions about Variable Ranking

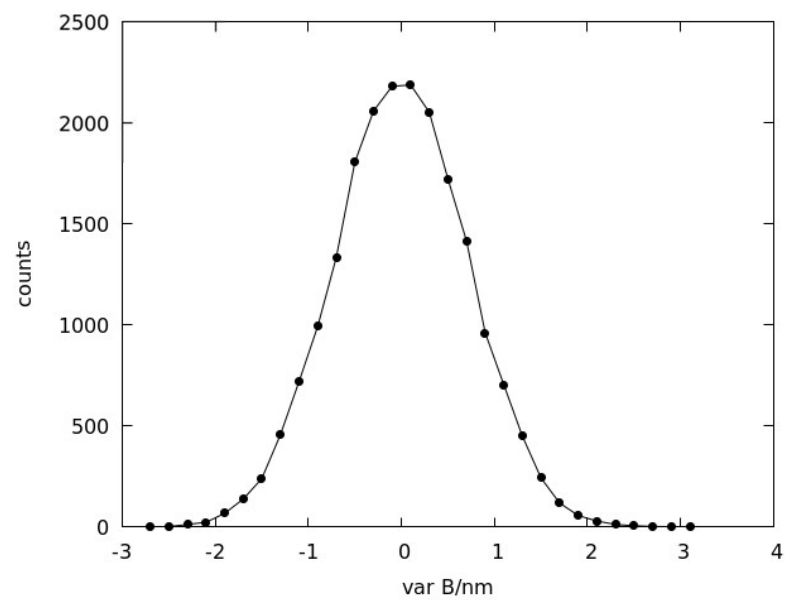
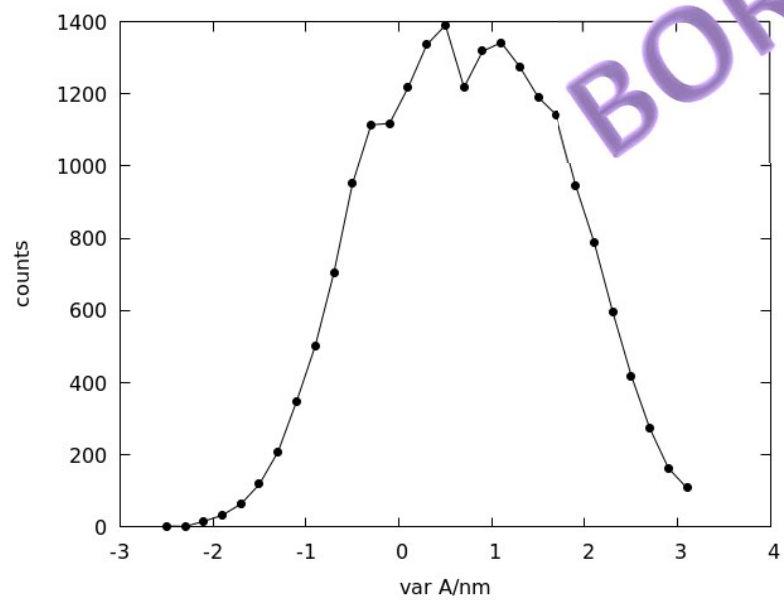
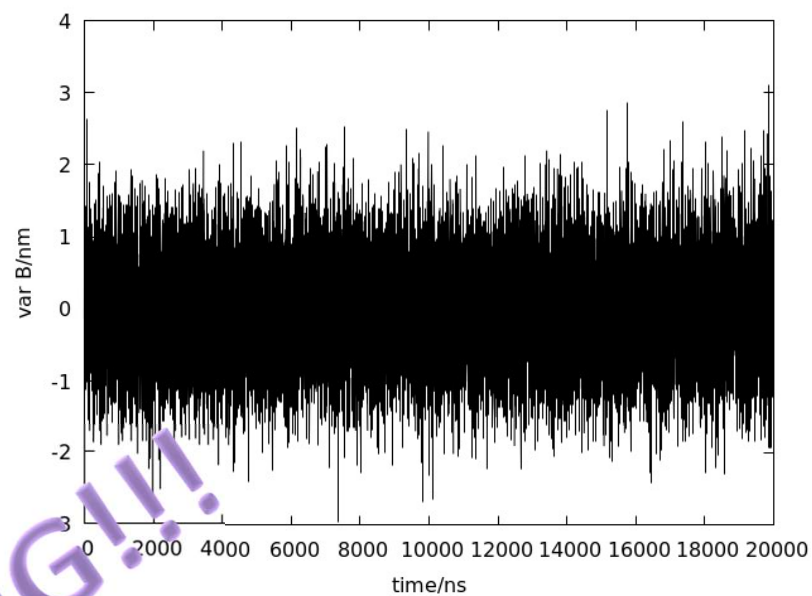
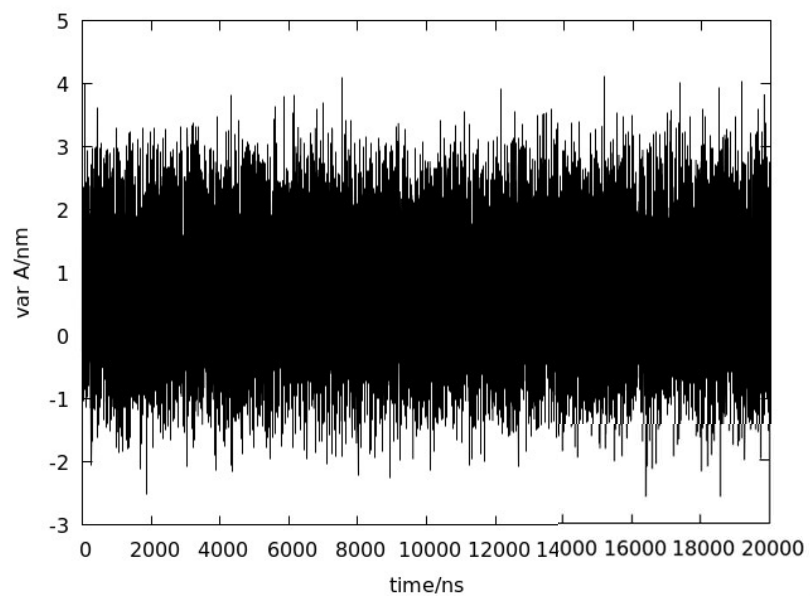
- How to treat redundant variables?
 - Redundant variables get the same information but its combination can lead to noise reduction.
 - Correlation is a measure of redundancy
- A variable useless by itself can be useful together with others



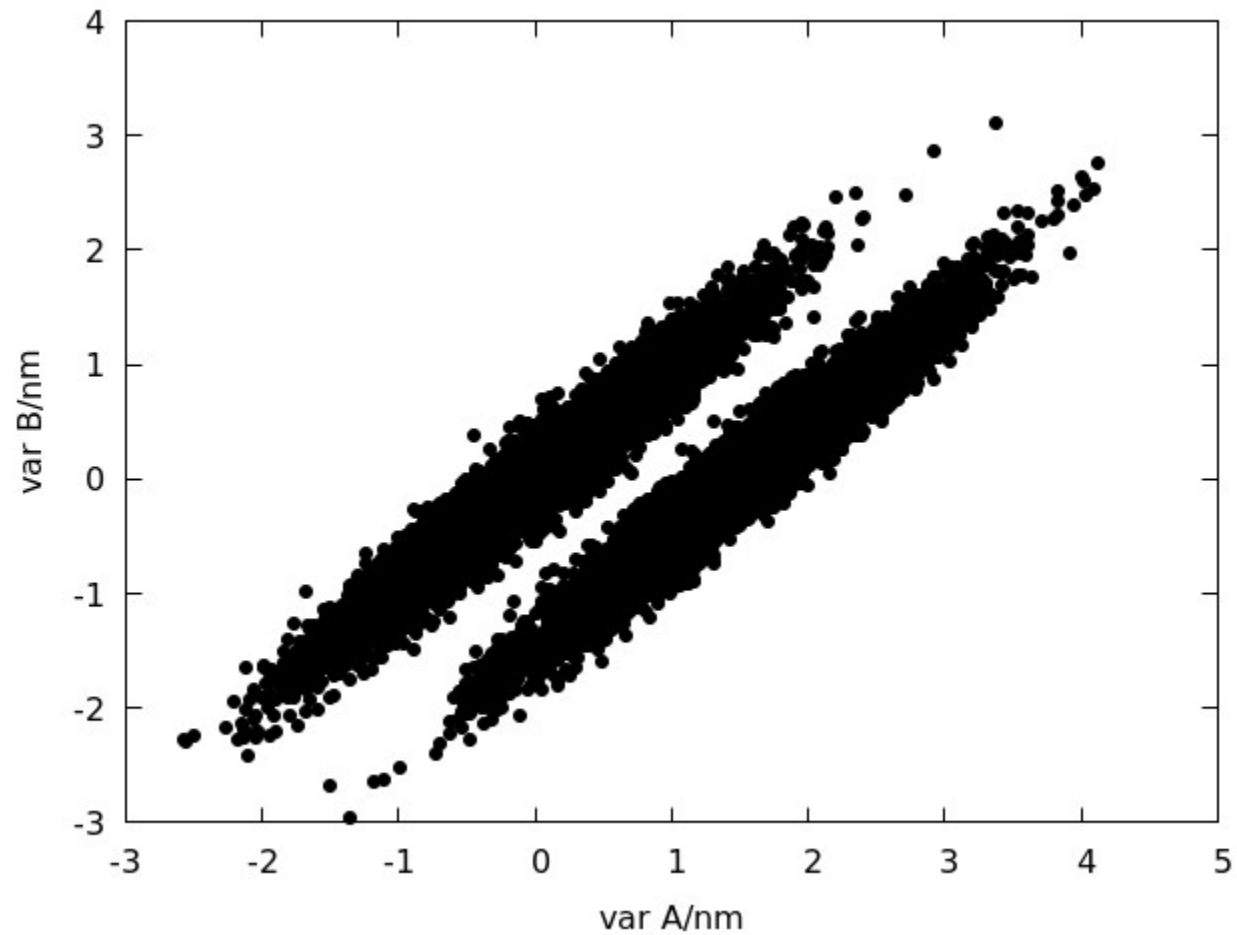




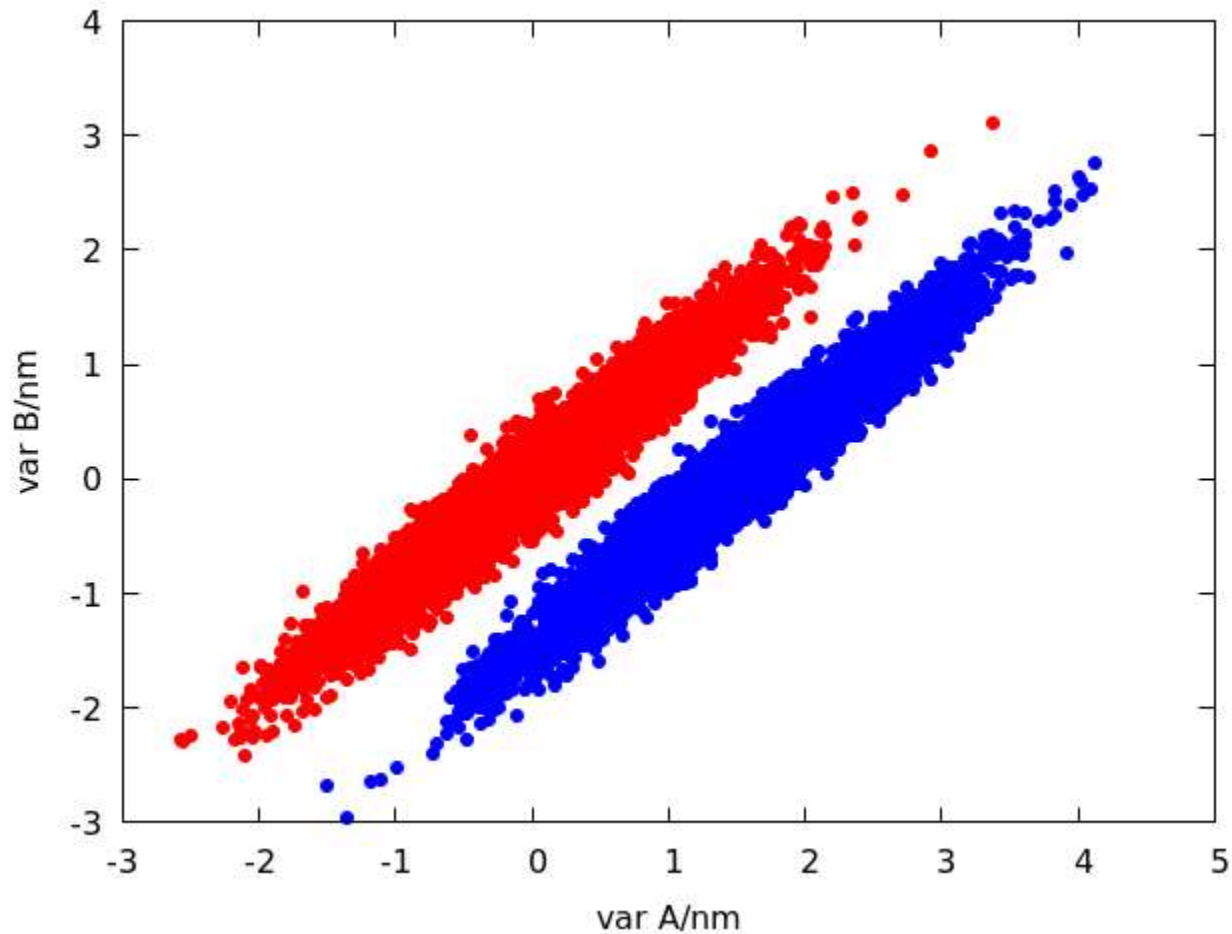




Go Multivariate! Plot 2D



Some structures appear only in
multivariate analysis



My stone collection (4)



Metálico	1.- Talco	6.- Ortosa	Azufre	Magnetita	Escamosa	Blanco	Negro	Naranja
Vítreo	2.- Yeso	7.- Cuarzo	Aragonito	Galena	Concoidea	Gris	Amarillo	Transparente
Graso	3.- Calcita	8.- Topacio	Rejalgar	Cinabrio	Lisa Yeso Especcular	Marrón	Granate	Rojo
Adamantino	4.- Fluorita	9.- Corindón	Azurita	Mercurio	Fibrosa	Dorado	Verde (opaco)	Verde (trasp.)
Anacarado	5.- Apatito	10.- Diamante	Malaquita	Oro	Rosetas	Morado	Azul	Combinado

- Weight
- Light wavelength
- Shape
- Volume
- Rugosity
- ...

Will it recover the density as an important feature for mineral recognition?
 $\text{Density} = \text{Weight} / \text{Volume}$

Techniques for Feature Selection (2)

Subset selection

- Wrappers: Use the predicting power of a given learning machine to assess the usefulness of a given subset
 - How to search the space? (Brute force is NP hard.)
 - How to assess the performance of the prediction.
 - Which learning machine use.

Similarities and distances

Master in High Performance
Computing

Similarity and Distances

- Clustering tries to separate data “*naturally*”, in such a way that *similar* elements lay in the same cluster while *dissimilar* elements belong to a different one
- *Similarity* (S_{ij}) is a pairwise function of the features of the elements i and j .
- In terms of space, it can be thought that similar elements are near while dissimilar are far. So many times it is useful to talk about “distances between elements” (D_{ij})
- Its definition depends on the nature of the features

Similarity and Distances almost the same but...

A (metric) distance must accomplish:

1. Symmetry: $d(x, y) = d(y, x)$
2. Non-negativity: $d(x, y) \geq 0$
3. Identity of indiscernibles: $d(x, y) = 0 \Leftrightarrow x = y$
4. Triangle inequality: $d(x, z) + d(z, y) \geq d(x, y)$

We have to take this into account for some clustering algorithms

Quantitative Features: *Metric Distances*

- Minkowski distance:

$$d_{ij} = \left(\sum_{l=1}^d |x_{il} - x_{jl}|^p \right)^{1/p}$$

– Special cases are:

- Euclidean (p=2)
- City-block (p=1)
- Sup (p→∞) . Eqv to $d_{ij} = \max_l |x_{il} - x_{jl}|$

- Mahalanobis distance

$$d_{ij} = (x_i - x_j)^T \mathbb{C}^{-1} (x_i - x_j)$$

Invariant with respect to any non-singular linear transformation of the coordinates. \mathbb{C} is the covariance matrix.

Quantitative Features: *Not metric distances*

- Pearson correlation:

$$d_{ij} = \frac{1-r_{ij}}{2}; r_{ij} = \frac{\sum_{k=1}^m (x_{k,i} - \bar{x}_i)(x_{k,j} - \bar{x}_j)}{\sqrt{\sum_{k=1}^m (x_{k,i} - \bar{x}_i)^2 \sum_{k=1}^m (x_{k,j} - \bar{x}_j)^2}}$$

- Point Symmetry distance:

$$d_{ij} = \min_{k \neq i} \frac{\|(x_i - x_j) + (x_k - x_j)\|}{\|(x_i - x_j)\| + \|(x_k - x_j)\|}$$

- Cosine similarity:

$$S_{ij} = \frac{x_i^T \cdot x_j}{\|x_j\| \|x_i\|}$$

Qualitative Features

- Jaccard similarity:

$$S_{ij} = \frac{|i \cap j|}{|i \cup j|}$$

$$i = \boxed{1}000\boxed{1}00\boxed{1} \quad j = 0\boxed{1}00\boxed{1}0\boxed{1}\boxed{1}; S_{ij} = \frac{2}{5}$$

- Hamming distance:

$$D_{ij} = |i \cup j| - |i \cap j|$$

i.e. minimum number of changes that you need to turn i in j .

More complicated distances

- Working in the metric can extremely simplify the clustering work.
- A good metric can dramatically improve the performance of an algorithm.
- However, usually they need to compute a simplest distance as starting point.
- Example: Geodesic distance