Unsupervised Learning

Machine Learning 2019

Cesare Alippi, Jürgen Schmidhuber, Michael Wand, Paulo Rauber

TAs: Róbert Csordás, Krsto Proroković, Xingdong Zuo, Francesco Faccio, Louis Kirsch

Unsupervised learning

Supervised learning:

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n)$$

predict y using x

Unsupervised learning:
- no label/response *y* available
- discover properties of given data

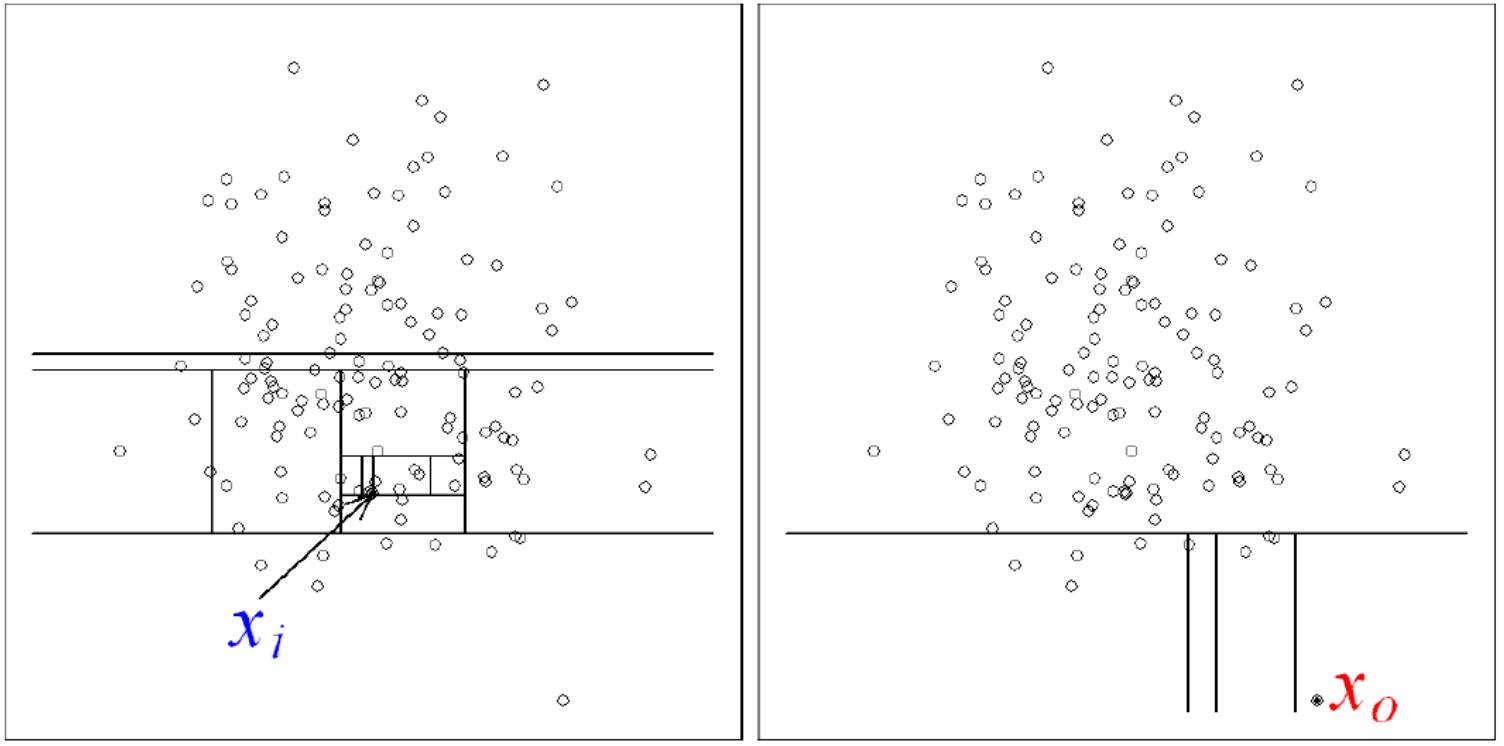
$$x_1, x_2, x_3, \ldots, x_n$$

Isolation Forest

- **Isolation Forest** (IFOR) is an interesting tree-based method designed to **explicitly isolate anomalies**
- The assumption is that anomalies are rare events that differ from the nominal class one

IFOR

- It is easier to isolate an anomalous point x_0 w.r.t a genuine one x_i when a k-dimensional tree is generated from data, with a random uniform splitting criterion.
- This means that anomalies lie in leaves with **shallow depth.**



(a) Isolating x_i (b) Isolating x_o

IFOR: The algorithm

Training

Build a forest of trees from a given dataset $\boldsymbol{X}$

For each training set generate a tree:

- Randomness: Uniformly select an axis to split
- Randomness: Determine a split point along the axis (uniform splitting)
- The selection/splitting procedure stops when
  - The tree reaches a limit on depth or
  - Number of points in the leaf equal 1 or
  - points in the leaf share the “same” values.

IFOR: The algorithm

Operational phase

- Compute the average path length $E\big(h(\boldsymbol{x})\big)$ among all the trees in the forest for the given test point $\boldsymbol{x}$.
- Point $\boldsymbol{x}$ is identified as **anomalous** if anomaly score

$$s(\boldsymbol{x}, n) = 2^{-\frac{E(h(\boldsymbol{x}))}{c(n)}} > T$$

where

- n: number of points in $\boldsymbol{X}$.
- $c(n)$: average path length of searches in generic binary trees given n data instances.

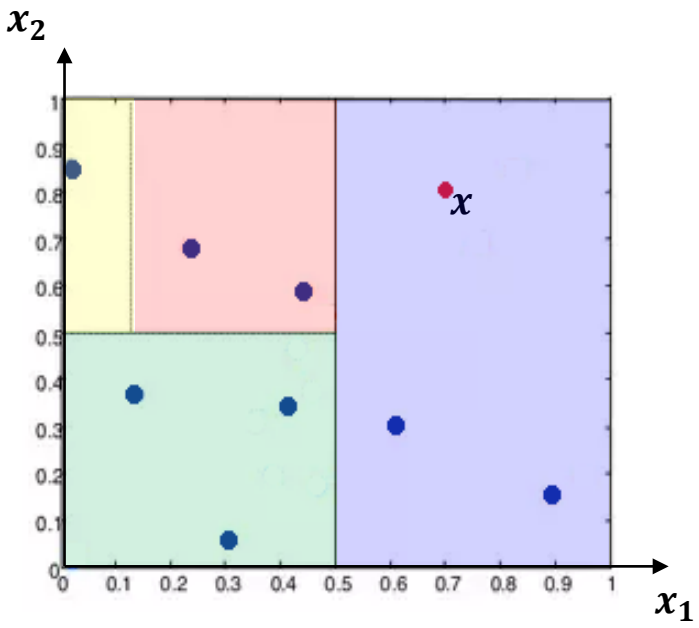
$$c(n) = 2(\ln(n-1) + \gamma_E) - 2\frac{n-1}{n}$$

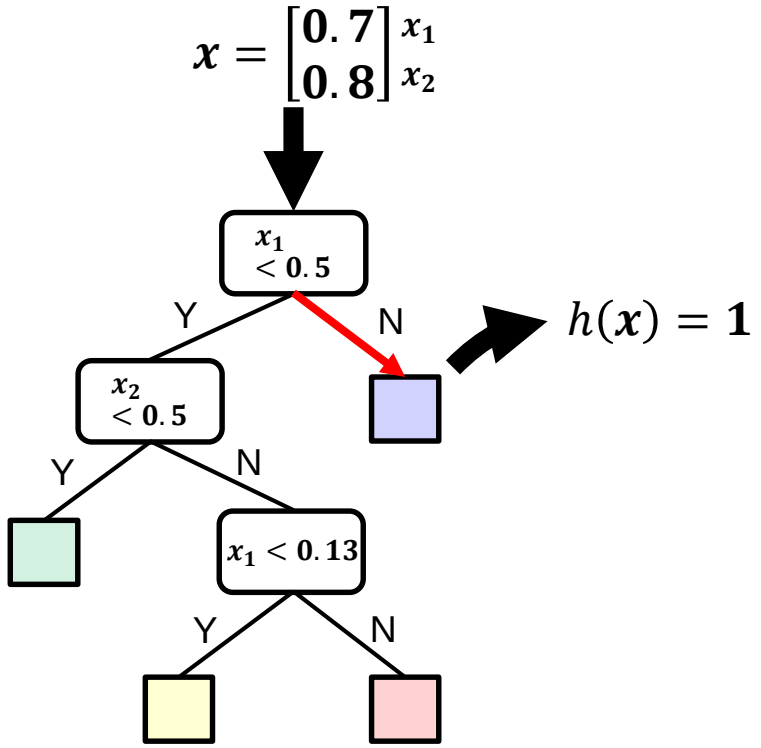
- With γ_E the Euler-Mascheroni constant
- T is a threshold

T may introduce false positiveness

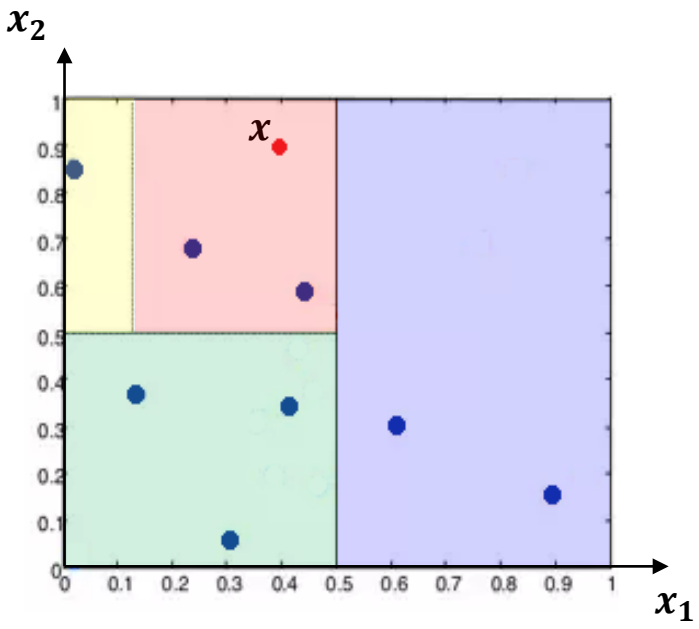
We can actually construct a distribution of s(x,n) to tell how confident we are to tell if some x is an anomaly

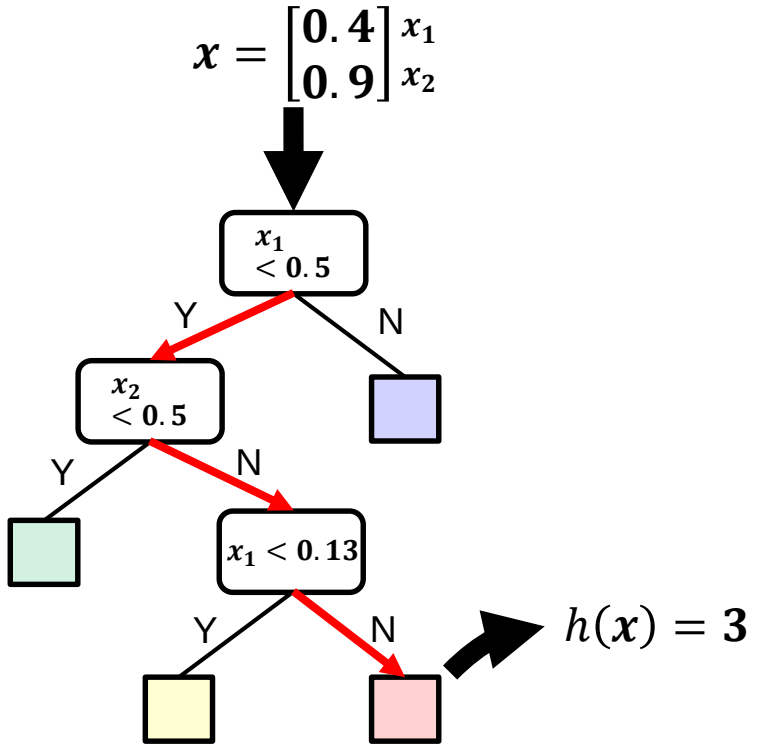
IFOR: The algorithm





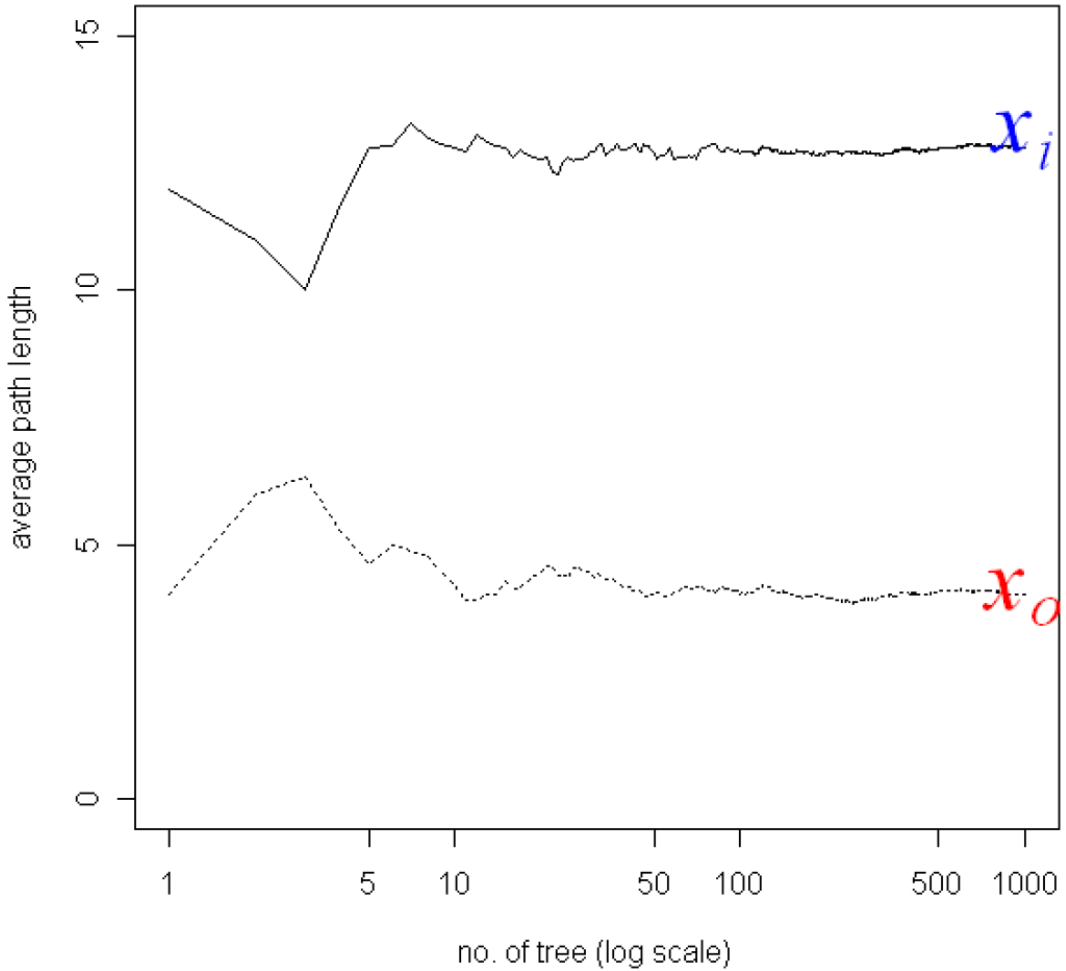
IFOR: The algorithm



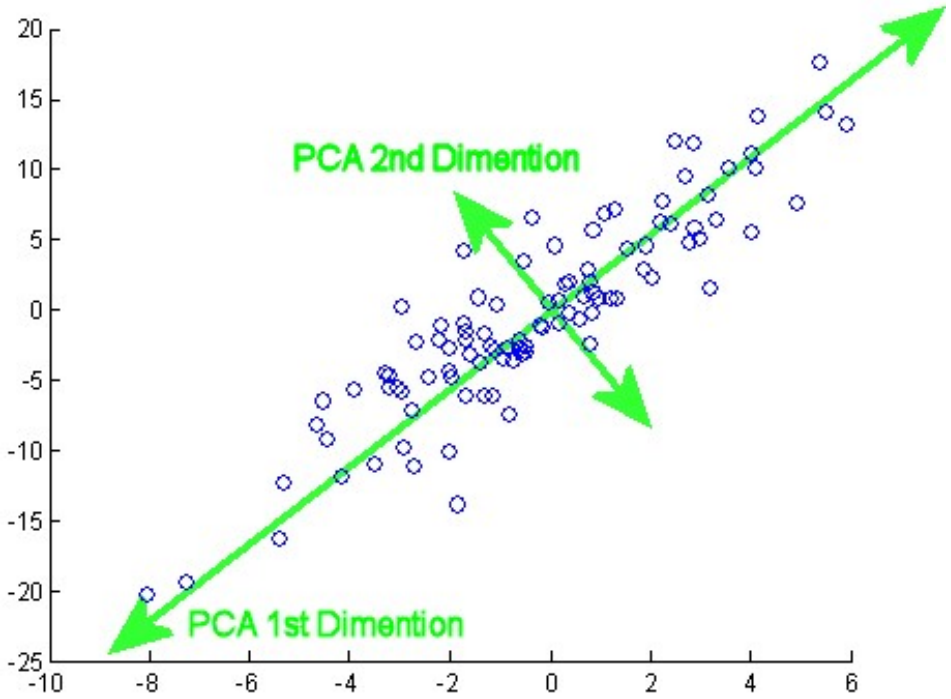


IFOR

- Averaged path lengths of x_i and x_0 **converge** when the number of trees in the forest increases.



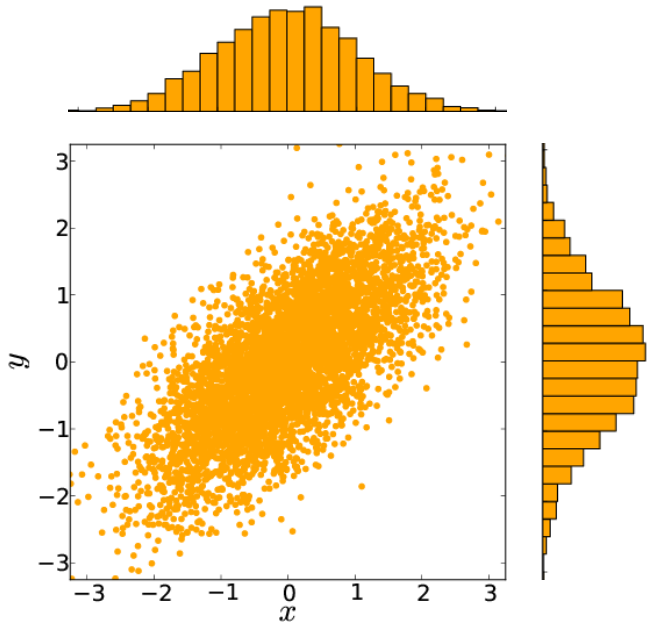
Principal Component Analysis (PCA)



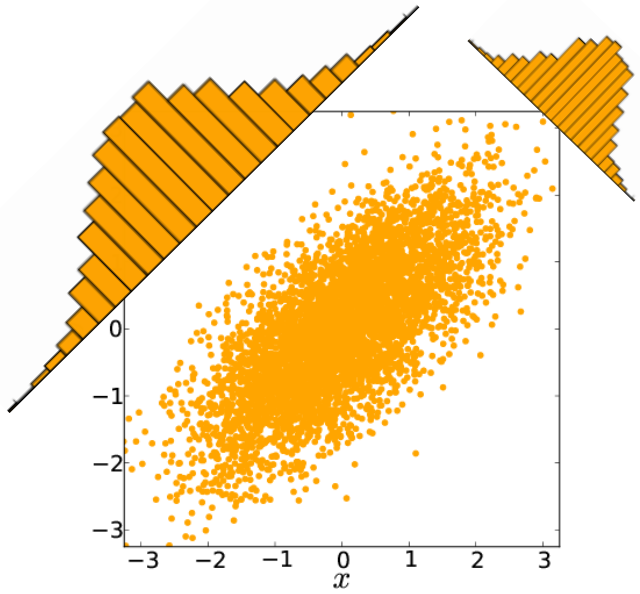
PCA changes the reference system of the data through a rotation operator

The axis of the new reference system shows the largest data scattering (one dimension at time)

Principal Component Analysis (PCA)



Principal Component Analysis (PCA)



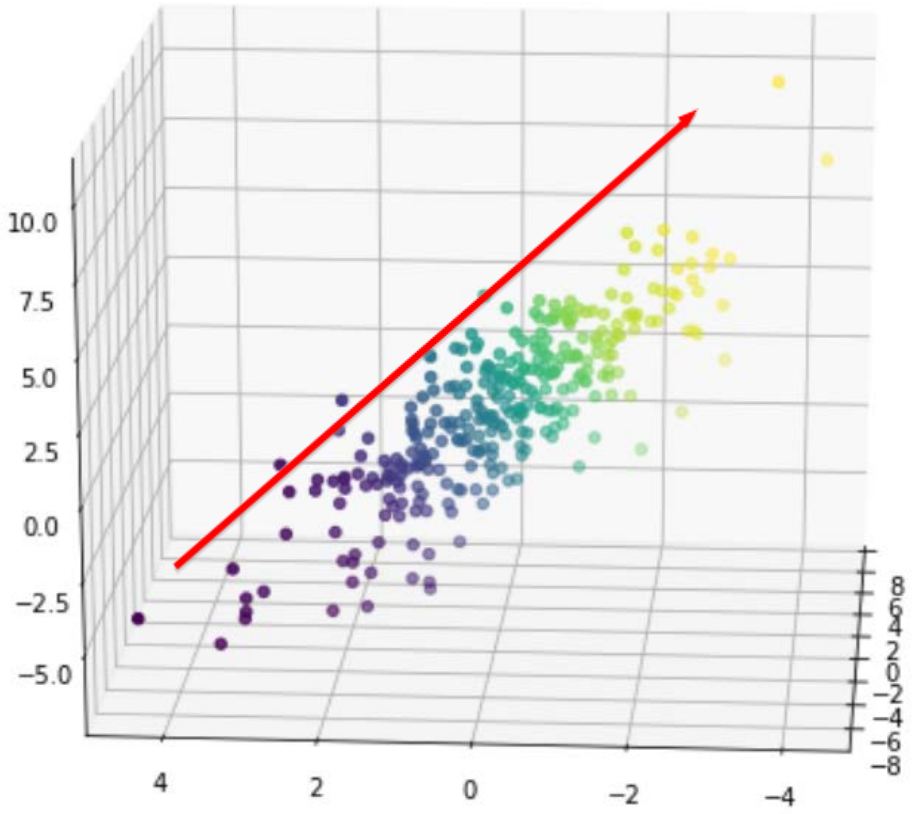
Principal Component Analysis

d-dimensional data points

$$x_1, x_2, x_3, \ldots, x_n$$

Grouped as per linear regression in

$$X = [x_1 | x_2 | \cdots | x_n]$$



PCA: data variability

Center the data (unless centered)

$$x_i' = x_i - \frac{1}{n}\sum x_j$$

- Compute the sample covariance matrix

$$\hat{\Sigma} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})(x_i - \overline{x})^{\top} = \frac{1}{n-1}\sum_{i=1}^{n} x_i x_i^{\top} = \frac{1}{n-1}X^{\top}X$$

- Consider the symmetrical semidefinite positive matrix

$$H = X^T X = U \Lambda U^T$$

characterized by eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$$

PCA: rotating the reference axes

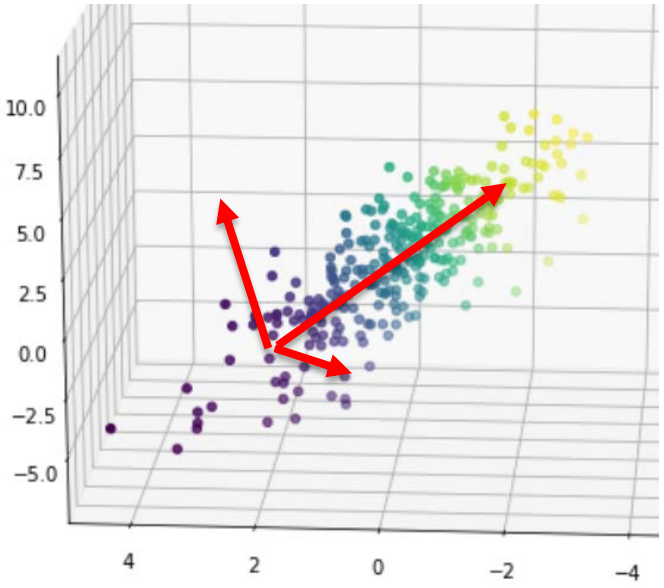
$$U = [U_1 | U_2 | \cdots | U_d]$$

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & & \dots & 0 \\ 0 & \lambda_2 & & & \\ & & \lambda_3 & & \\ \vdots & & & & \\ \vdots & & & \ddots & \\ 0 & & & & \lambda_d \end{bmatrix}$$

- Each component of

$$\tilde{x} = U^T x$$

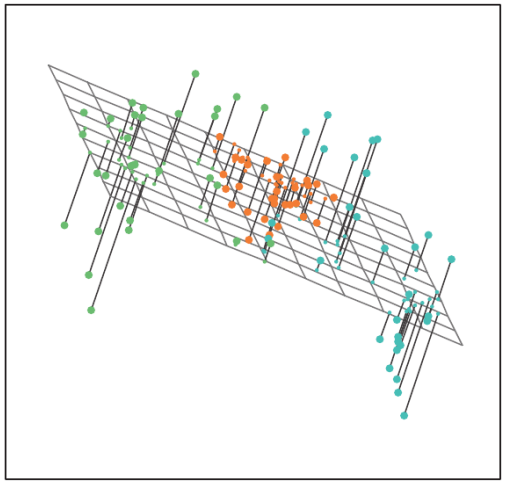
is called **principal component**



PCA: dimensionality reduction (and visualization)

- Remove the smallest *l* eigenvalues and group the remaining eigenvectors as

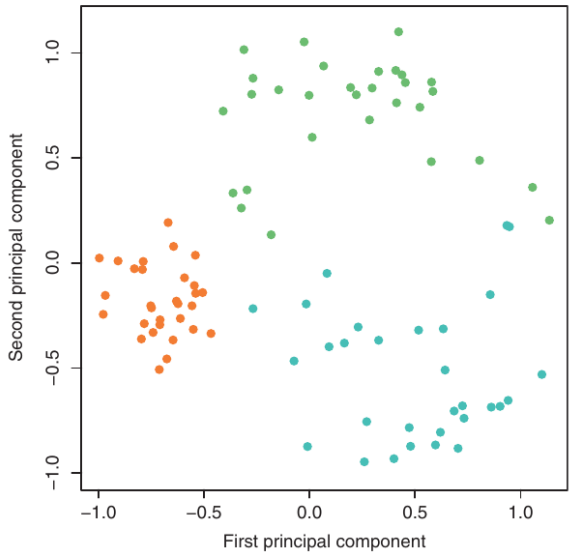
$$\tilde{U} = [U_1 | U_2 | \cdots | U_{l+1}]$$



- Projection (dimensionality reduction) is achieved by means of

$$\tilde{x} = \tilde{U}^T x$$

- as we embed vectors from dimensionality *d* into a lower *d-l* dimensional space



PCA: principal component analysis

- How much information do we loose?
- The spectral decomposition theorem grants that

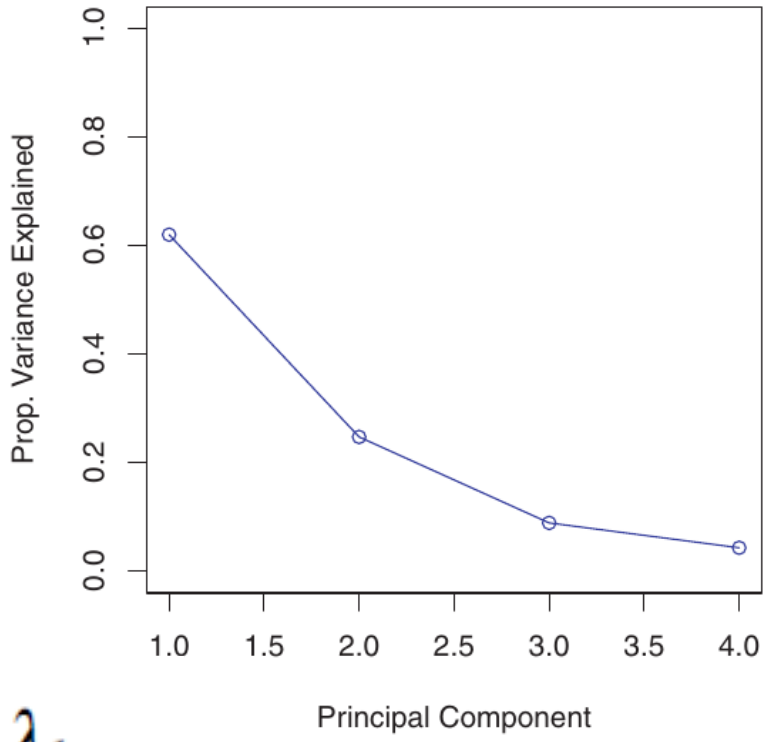
$$H = \sum_{i=1}^{d} \lambda_i U_i U_i^T$$

if we remove the smallest l eigenvalues we generate

$$\tilde{H} = \sum_{i=l+1}^{d} \lambda_i U_i U_i^T$$

- So that (worst case)

$$\|H - \tilde{H}\|_2 = \lambda_l$$



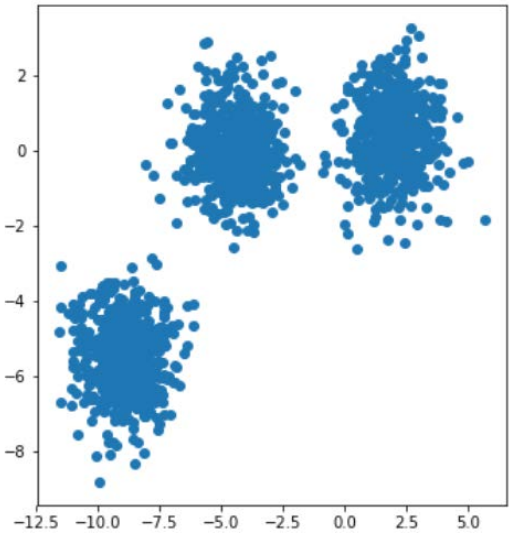
Clustering

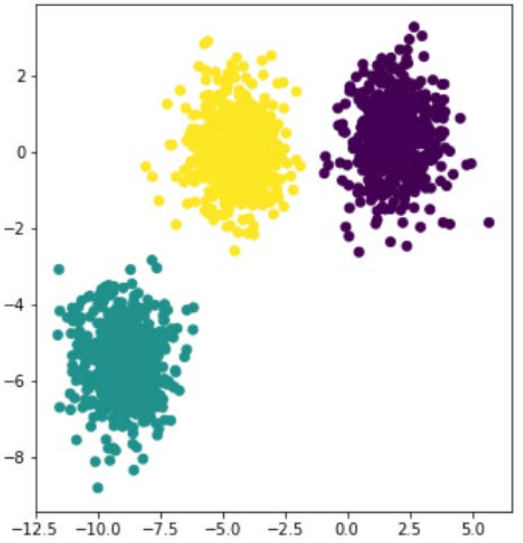
Sometimes the data group together in certain regions of the feature/input space.

Clustering: group data so that

- Points in the same cluster stay close each other

How to define the concept of closeness...





K-Means clustering

Consider data points $x_1, x_2, x_3, \ldots, x_n$

- Cluster: a set of indices $C_k \subseteq \{1, 2, 3, \ldots, n\}$
- Clustering: a collection of clusters so that

$$C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$$

$$C_k \cap C_{k'} = \emptyset \text{ for all } k \neq k'$$

K-means clustering searches for K points

$$\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^d$$

called means, and assigns the class as

$$i \in C_{\hat{k}} \iff \hat{k} = \underset{k \in \{1,2,\ldots,K\}}{\operatorname{argmin}} \|x_i - \mu_k\|_2$$

K-Means clustering

The K clusters are identified so as to minimise the scattering of the data within each cluster

$$\underset{C_1,\ldots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$

$$W(C_k) = \sum_{i,i' \in C_k} \|x_i - \mu_k\|_2^2$$

Where the $\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^d$ are the cluster means

$$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$$

K-Means clustering

Input: data $x_1, x_2, x_3, \ldots, x_n$, ***number K of clusters***
Output: clusters $C_1, \ldots, C_K$

1. Extract $\mu_1, \mu_2, \ldots, \mu_K$ randomly from $x_1, x_2, x_3, \ldots, x_n$

Introduce Markov Process to account for time-dependence

2. Create clusters: for k = 1, … , K

$$C_k = \{i \in \{1, \ldots, n\} \; s.t. \; \|x_i - \mu_k\|_2 \leq \|x_i - \mu_j\|_2, \; j = 1, \ldots, K\}$$

3. Compute means: for k = 1, … , K $\quad \mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i$

4. If cluster changed, go to step 2, otherwise exit.