Course 3: Unsupervised Learning

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Summary

Last session

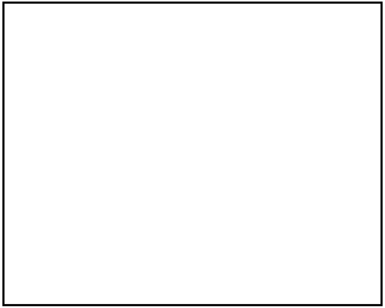
1. Supervised learning - learning from labeled examples
2. Bias/variance tradeoff
3. Overfitting and cross-validation
4. VC Dimension and curse of dimensionality

Today's session

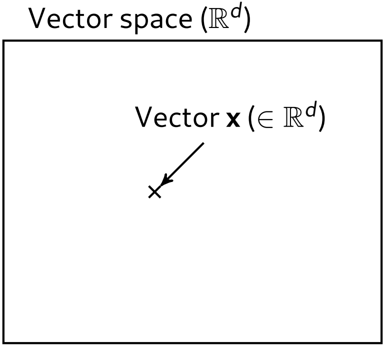
1. Learning from Unlabeled examples
2. Clustering, decomposition and dimensionality reduction

Notations

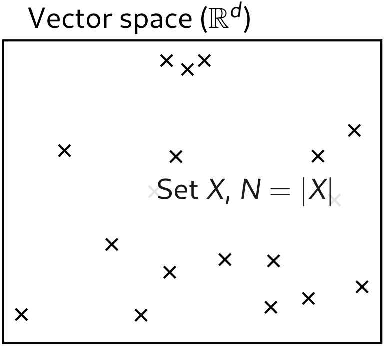
Vector space ($\mathbb{R}^d$)



Notations



Notations



Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Quantization, Dimensionality reduction
  - Visualization...

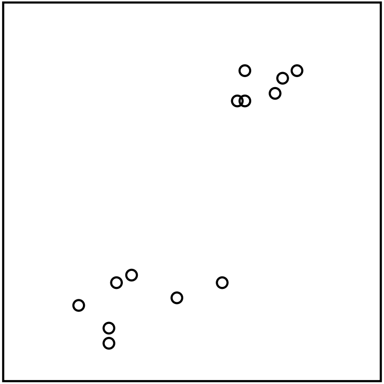
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Quantization, Dimensionality reduction
  - Visualization...



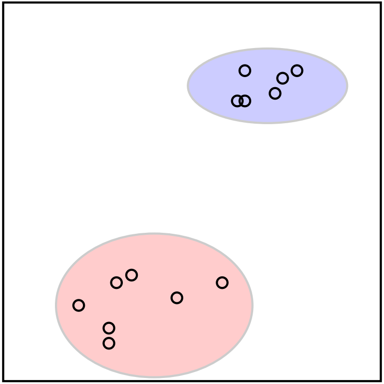
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Quantization, Dimensionality reduction
  - Visualization...



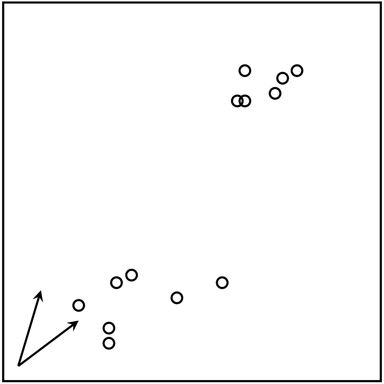
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Quantization, Dimensionality reduction
  - Visualization...

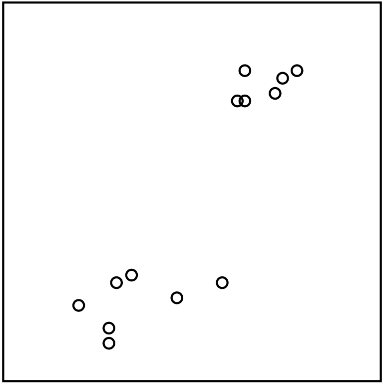


Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Quantization, Dimensionality reduction
  - Visualization...



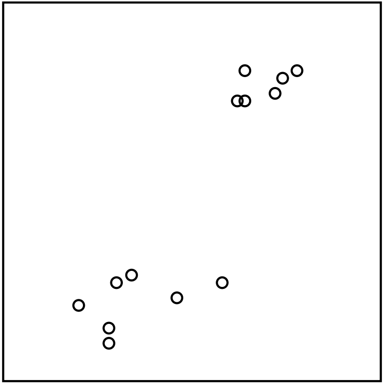
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels
- Main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
  - Manifold Learning.
- Applications :
  - Quantization, Dimensionality reduction
  - Visualization...



Example: clustering using L_2 norm (1/6)

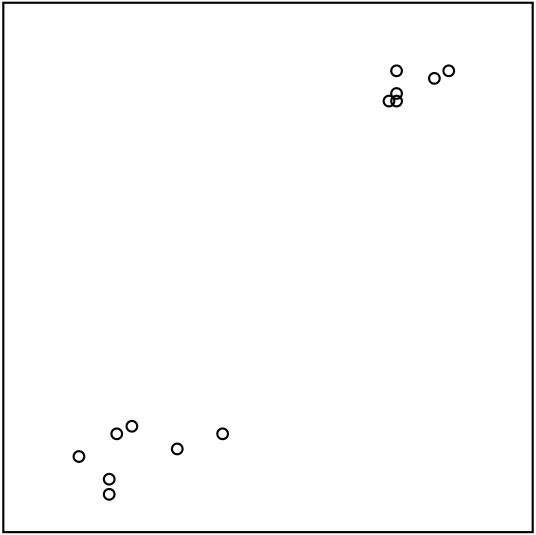
An example to perform clustering is to rely on distances to centroids. We define K *cluster centroids* $\Omega_k, \forall k \in [1..K]$

Definitions

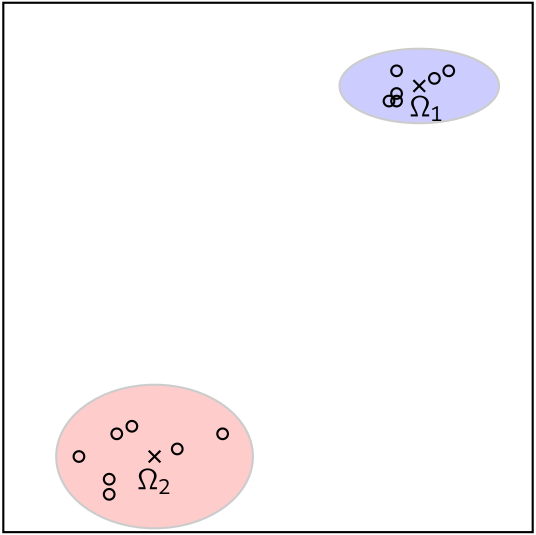
We denote $q : \mathbb{R}^d \to [1..K]$ a function that associates a vector $\mathbf{x}$ with the index of (one of) its closest centroid $q(\mathbf{x})$. Formally:

- $\forall k \in [1..K], \Omega_k \in \mathbb{R}^d$
- $\forall \mathbf{x} \in X, \forall j \in [1..K], \|\mathbf{x} - \Omega_{q(\mathbf{x})}\|_2 \leq \|\mathbf{x} - \Omega_j\|_2$
- Error $E(q) \triangleq \sum_{\mathbf{x} \in X} \|\mathbf{x} - \Omega_{q(\mathbf{x})}\|_2$
- $X = \bigcup_k \underbrace{\{\mathbf{x} \in X, q(\mathbf{x}) = k\}}_{\text{cluster } k}$

Example: clustering using L_2 norm (2/6)



Example: clustering using L_2 norm (2/6)



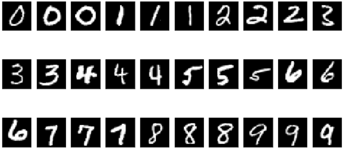
Clustering using L_2 norm (3/6)

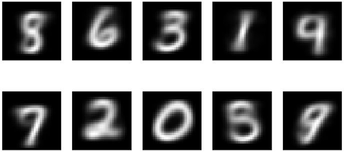
MNIST Dataset

- "Toy" dataset (=small and easy)
- 60000 + 10000 handwritten digits

Clustering MNIST

Using K-means algorithm with $K = 10$





Clustering using L_2 norm (4/6)

Quantizing MNIST

- Replace $\mathbf{x}$ by $\Omega_{k(\mathbf{x})}$
- Compression factor $\kappa = 1 - K/N$

Clustering using L_2 norm (5/6)

Optimal clustering

- Define $E_{opt_K}(q^*) \triangleq \arg \min_{q:\mathbb{R}^d \to [1..K]} E(q)$,
- Finding an optimal clustering is an NP-hard problem.

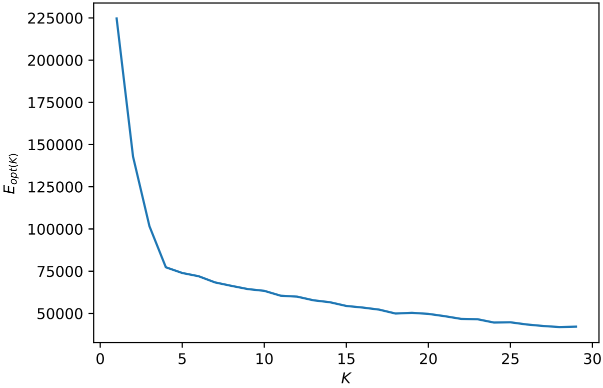
Properties

- $0 = E_{opt_N}(q^*) \leq E_{opt_{N-1}}(q^*) \leq \dots \leq E_{opt_1}(q^*) = var(X)$,
  - Proof: monotonicity by particularization, extremes with identity function (left) and variance (right).
- $0 \leq \kappa \leq \frac{N-1}{N}$.

Clustering using L_2 norm (6/6)

Choosing K

- Finding a compromise between error and compression,
- Simple practical method : "elbow".

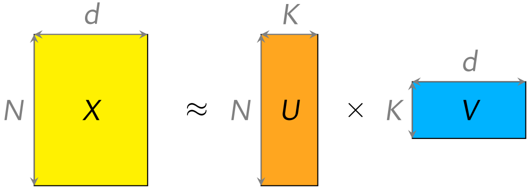


Example 2: Sparse Dictionary Learning (1/4)

Definitions

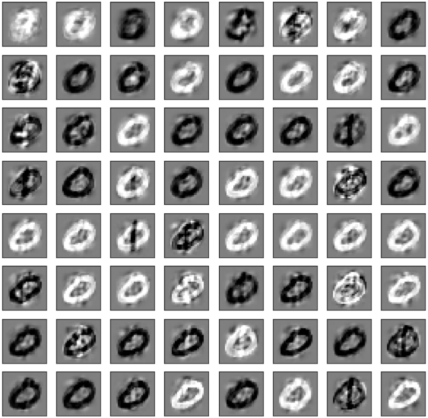
Dictionary learning solves the following matrix factorization problem:

- The set X is considered as a matrix $X \in \mathcal{M}_{N \times d}(\mathbb{R})$,
- We consider decompositions using a dictionary $V \in \mathcal{M}_{K \times d}(\mathbb{R})$ and a code $U \in \mathcal{M}_{N \times k}(\mathbb{R})$, with the lines of V being with norm 1,
- Error $E(U, V) \triangleq \|X - UV\|_2 + \alpha \|U\|_1$
- Training: find U^*, V^* that minimizes $E(U^*, V^*)$
- α is a sparsity control parameter that enforces codes with soft (ℓ_1) sparsity



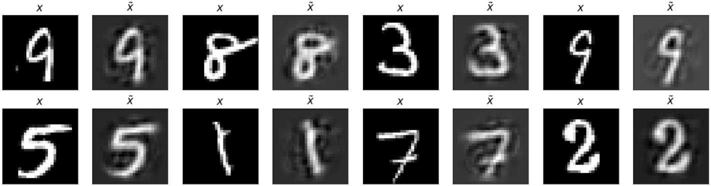
Example: Sparse Dictionary Learning (2/4)

Learning a dictionary on MNIST with $K = 64$

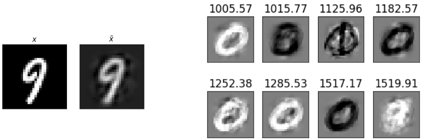


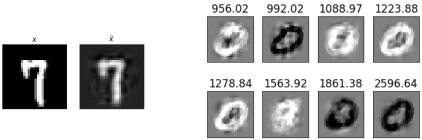
Example 2: Sparse Dictionary Learning (3/4)

Reconstruction $\tilde{\mathbf{x}} = UV$ of $\mathbf{x}$



8 atoms with largest absolute values:





Example 2: Sparse Dictionary Learning (4/4)

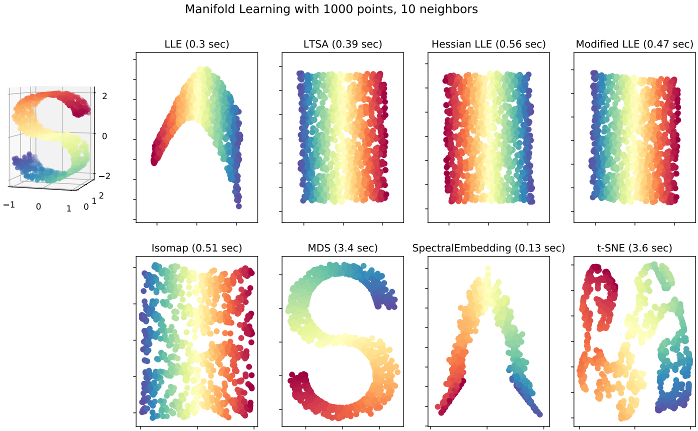
Optimal error

- $E_{opt_K}(U^*, V^*) \triangleq \arg\min_{U,V} E(U, V)$.

Some results

- For $\alpha = 0$ and $K \geq d$, $E_{opt_d}(U^*, V^*) = 0$,
  - One can choose any completion of a basis.
- For $K = N$, $\forall\alpha$, $E_{opt_K}(U^*, V^*) = \alpha N$,
  - If vectors of X are with norm 1, one can choose $V = X$ and $U = \mathbf{I}_N$.

Example 3: Manifold Learning

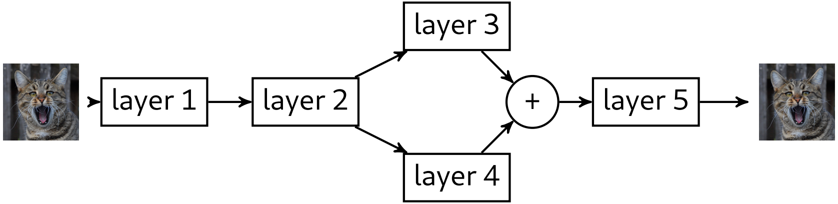


Approaches to uncover lower dimensional structure of high dimensional data. Source : Manifold module, sklearn website

Example 4: (Deep) auto-encoders

Inputs/outputs

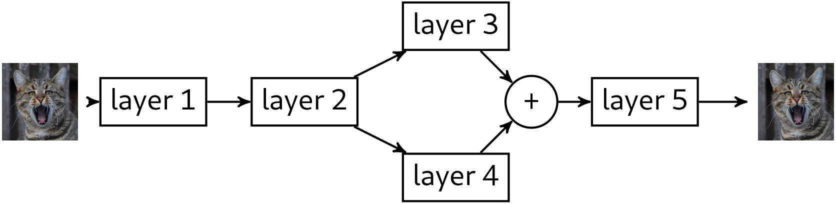
- Often: inputs are **raw signals**,
- Often: outputs are **raw signals**.



Example 4: (Deep) auto-encoders

Inputs/outputs

- Often: inputs are **raw signals**,
- Often: outputs are **raw signals**.



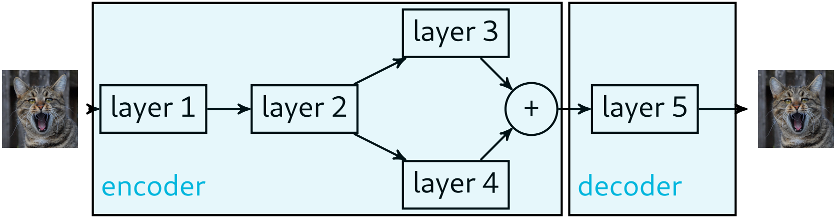
Precisions

- Parameters are trained to **reproduce the input**,
- Some (arbitrary) **intermediate representation** is interpreted as the **decomposition**,
- Loss is typically **Mean Square Error**: $\sum_i (\mathbf{y}_i - \mathbf{x}_i)^2$.

Example 4: (Deep) auto-encoders

Inputs/outputs

- Often: inputs are **raw signals**,
- Often: outputs are **raw signals**.



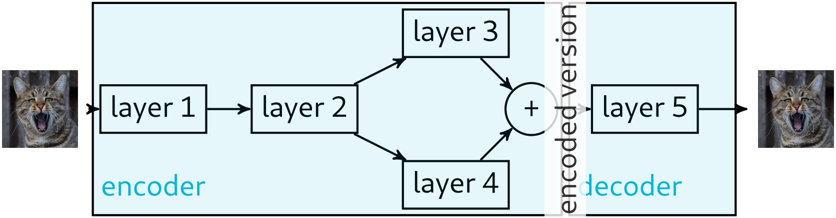
Precisions

- Parameters are trained to **reproduce the input**,
- Some (arbitrary) **intermediate representation** is interpreted as the **decomposition**,
- Loss is typically **Mean Square Error:** $\sum_i (\mathbf{y}_i - \mathbf{x}_i)^2$.

Example 4: (Deep) auto-encoders

Inputs/outputs

- Often: inputs are **raw signals**,
- Often: outputs are **raw signals**.



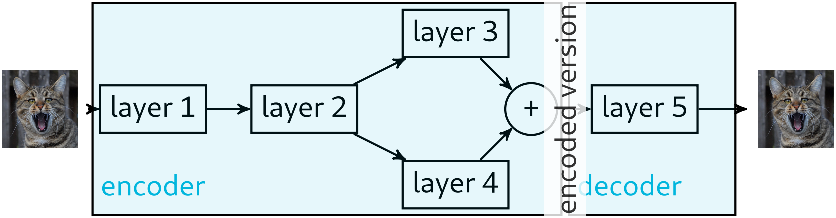
Precisions

- Parameters are trained to **reproduce the input**,
- Some (arbitrary) **intermediate representation** is interpreted as the **decomposition**,
- Loss is typically **Mean Square Error:** $\sum_i (\mathbf{y}_i - \mathbf{x}_i)^2$.

Example 4: (Deep) auto-encoders

Inputs/outputs

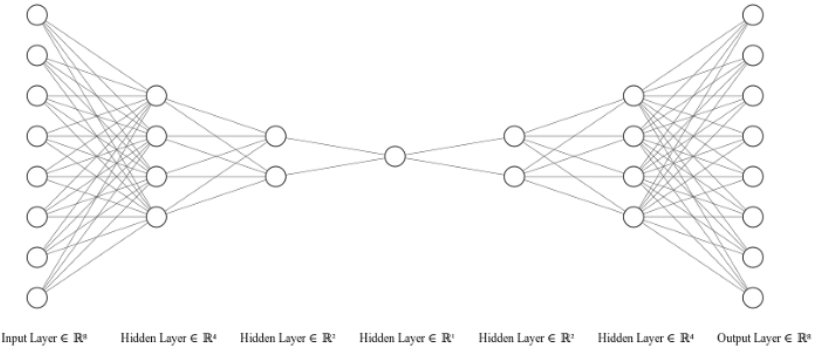
- Often: inputs are **raw signals**,
- Often: outputs are **raw signals**.



Precisions

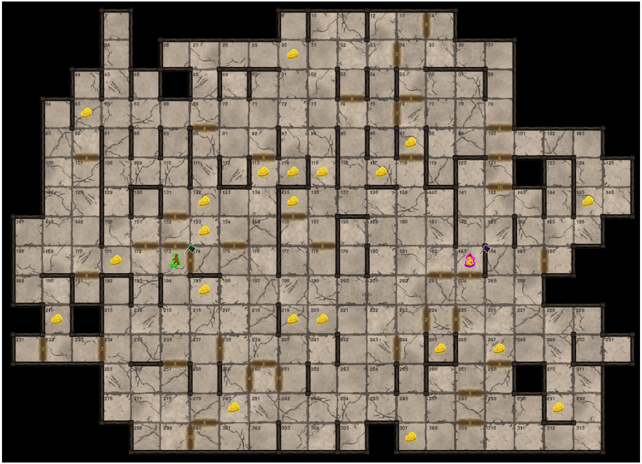
- Parameters are trained to **reproduce the input**,
- Some (arbitrary) **intermediate representation** is interpreted as the **decomposition**,
- Loss is typically **Mean Square Error**: $\sum_i (\mathbf{y}_i - \mathbf{x}_i)^2$.

Autoencoder on MNIST



Illustrated example of an autoencoder in Pytorch on MNIST
https://medium.com/pytorch/implementing-an-autoencoder-in-pytorch-19baa22647d1

Clustering on pyrat derived features



	density(rat)	density(python)	distance(rat, python)	density(cheese_0)	distance(rat, cheese_0)	distance(python, cheese_0)	density(cheese_1)	distance(rat, cheese_1)	distance(python, cheese_1)
0	3.475423	2.307948	9.0	2.266966	1.0	8.0	2.016979	4.0	11.0
1	2.401837	2.202623	1.0	2.091537	3.0	4.0	2.109535	6.0	7.0
2	2.948456	2.659309	11.0	2.474181	3.0	14.0	3.116937	5.0	16.0
3	2.655862	2.352433	13.0	2.355235	2.0	11.0	2.183774	3.0	10.0
4	2.204104	2.645298	3.0	3.629737	5.0	8.0	3.910904	6.0	9.0

Clustering on pyrat derived features

We give you 1000 initial game configurations (two players, 21 cheese pieces) with the following features (66 in total), computed using the distances as shortest path in the graph:

- Distance between the two players $d(p, r)$
- Distance between each player and each cheese
- Density of cheese around each player starting position $density(p) = \sum_c \frac{1}{d(p,c)}$
- Density of cheese around each cheese position $density(c) = \sum_{c' \neq c} \frac{1}{d(c,c')}$
- Cheese are sorted according to the ratio $o(c) = \frac{d(r,c)}{d(p,c)}$

Your task : Find clusters in this dataset, we will evaluate your cluster labels using the ground truth.
more details in the lab session 3 notebook

Working with features

N.b. : valid in unsupervised and supervised settings.

Feature preprocessing

Objective : change the statistical distribution of the features

- Scaling / Normalization
- Power transform
- Encode, discretization
- Manual feature engineering
- See more `https://scikit-learn.org/stable/modules/preprocessing.html`

Many techniques need or are greatly helped when features are on the unit sphere.

Working with features

N.b. : valid in unsupervised and supervised settings.

Feature selection

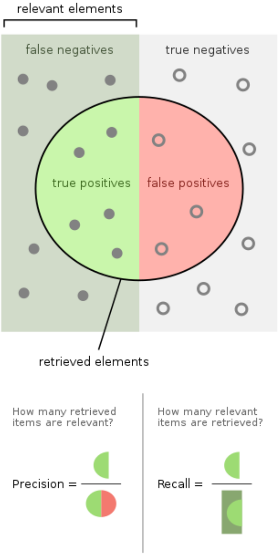
Objective : remove features

- Remove features with low variance
- Select features according to their explained variance towards labels (e.g. SelectKBest)
- See more `https://scikit-learn.org/stable/modules/feature_selection.html`

Helps to adress the dimensionality curse.

Metrics

In supervised learning : per class metric



Metrics

Clustering Metrics :

- Error defined slide 5 : similar to inertia (sum of squared distances)
- The Silhouette score (per sample) is $(b - a)/max(a, b)$, with mean intra-cluster distance (a) and the mean nearest-cluster distance (b).

Clustering metrics using labels :

- Random Index : measures the similarity of two assignments, ignoring permutations
- Homogeneity : each cluster contains only members of a single class.
- Completeness : all members of a given class are assigned to the same cluster.

See more on sklearn website and in the lab session

Metrics

Clustering Metrics :

- Error defined slide 5 : similar to inertia (sum of squared distances)
- The Silhouette score (per sample) is $(b - a)/max(a, b)$, with mean intra-cluster distance (a) and the mean nearest-cluster distance (b).

Clustering metrics using labels :

- Random Index : measures the similarity of two assignments, ignoring permutations
- Homogeneity : each cluster contains only members of a single class.
- Completeness : all members of a given class are assigned to the same cluster.

See more on sklearn website and in the lab session

Metrics

Clustering Metrics :

- Error defined slide 5 : similar to inertia (sum of squared distances)
- The Silhouette score (per sample) is $(b-a)/max(a,b)$, with mean intra-cluster distance (a) and the mean nearest-cluster distance (b).

Clustering metrics using labels :

- Random Index : measures the similarity of two assignments, ignoring permutations
- Homogeneity : each cluster contains only members of a single class.
- Completeness : all members of a given class are assigned to the same cluster.

See more on sklearn website and in the lab session

Metrics

Clustering Metrics :

- Error defined slide 5 : similar to inertia (sum of squared distances)
- The Silhouette score (per sample) is $(b - a)/max(a, b)$, with mean intra-cluster distance (a) and the mean nearest-cluster distance (b).

Clustering metrics using labels :

- Random Index : measures the similarity of two assignments, ignoring permutations
- Homogeneity : each cluster contains only members of a single class.
- Completeness : all members of a given class are assigned to the same cluster.

See more on sklearn website and in the lab session

Lab Session 3 and assignments for Session 5

TP Unsupervised Learning (TP2)

- K-means, Dictionary Learning and Manifold Learning
- Application on Digits and PyRat

Project 2 (P2)

- Find clusters in the provided dataset of pyrat games features.
- You can combine every technique you want (feature selection, decomposition, clustering, ...)
- During Session 4 you will have 7 minutes to present your work.
- We will evaluate the quality of your clustering during your presentation.