Course 2: Supervised Learning

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

Summary

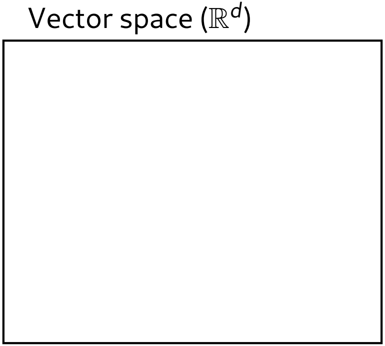
Last session

1. AI definition
2. Applications & Open Issues
3. Deep learning
4. Foundation models

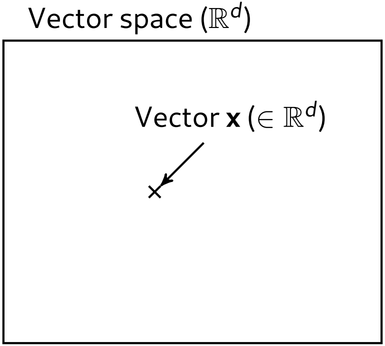
Today's session

- Learning from labeled examples
- Challenges of supervised learning

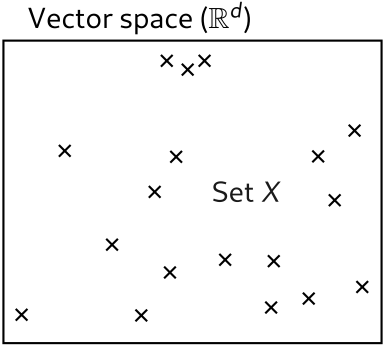
Notations



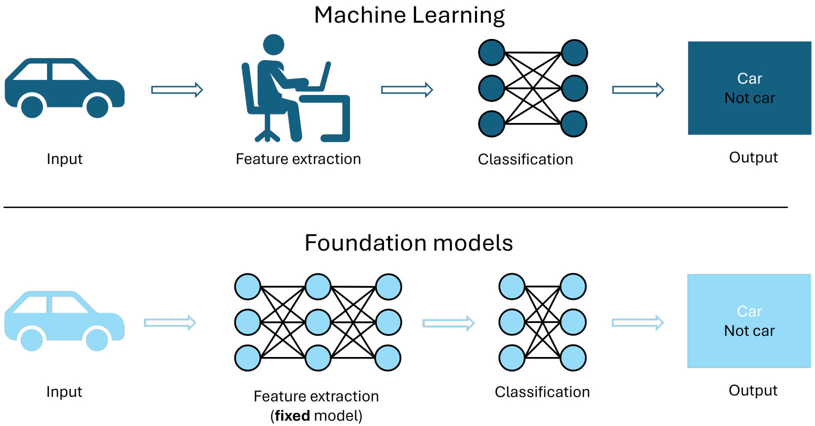
Notations



Notations



What is the vector x? (1/2)



What is the vector x? (2/2)

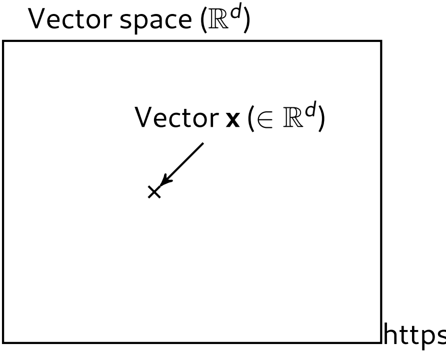
Traditional Machine Learning

x is the data, or a small transformation of the data
Ex: images, or edges in the image

The era of Foundation models

x is the projection of data in an **embedding** space

- Advantage: richer semantically than the original image
- Disadvantage: Not interpretable nor easily understandable



What is the vector x? (2/2)

Traditional Machine Learning

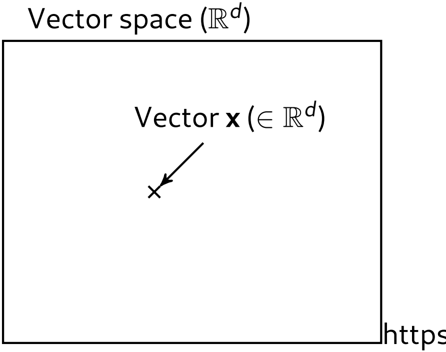
x is the data, or a small transformation of the data

Ex: images, or edges in the image

The era of Foundation models

x is the projection of data in an **embedding** space

- Advantage: richer semantically than the original image
- Disadvantage: Not interpretable nor easily understandable



Traditional Machine Learning

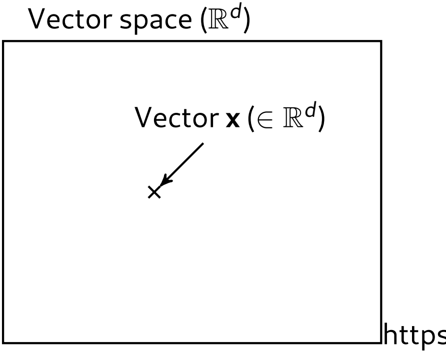
x is the data, or a small transformation of the data

Ex: images, or edges in the image

The era of Foundation models

x is the projection of data in an **embedding** space

- Advantage: richer semantically than the original image
- Disadvantage: Not interpretable nor easily understandable



In this class, we will illustrate the concepts using images... BUT in the lab, you will use embedding spaces (the future is probably there)

Supervised learning

Definition

Given:

- $\mathbf{x}$: inputs (raw signals or feature vectors (e.g. embeddings))
- $\hat{\mathbf{y}}$: **labels** (annotated by humans)

Learn:

- a function $f()$ such that $\hat{\mathbf{y}} \approx \mathrm{f}(\mathbf{x})$
  $\Rightarrow$ $f()$ is **learned** by the Machine Learning algorithm
- Ideally, $f()$ should **generalize** ($\neq$ memorize) to unlabeled examples.

$\mathrm{f}(\mathbf{x})$: 

$\hat{\mathbf{y}}$: "cat"

Supervised learning

Definition

Given:

- **x**: inputs (raw signals or feature vectors (e.g. embeddings))
- $\hat{\mathbf{y}}$: **labels** (annotated by humans)

Learn:

- a function *f*() such that $\hat{\mathbf{y}} \approx f(\mathbf{x})$
  $\Rightarrow$ *f*() is **learned** by the Machine Learning algorithm
- Ideally, *f*() should **generalize** ($\neq$ memorize) to unlabeled examples.

f(**x**):  Learn *f*() → $\hat{\mathbf{y}}$: "cat"

Supervised learning

Definition

Given:

- $\mathbf{x}$: inputs (raw signals or feature vectors (e.g. embeddings))
- $\hat{\mathbf{y}}$: **labels** (annotated by humans)

Learn:

- a function $f()$ such that $\hat{\mathbf{y}} \approx \mathrm{f}(\mathbf{x})$
  $\Rightarrow$ $f()$ is **learned** by the Machine Learning algorithm
- Ideally, $f()$ should **generalize** ($\neq$ memorize) to unlabeled examples.

f($\mathbf{x}$):

Learn $f()$ $\longrightarrow$ $\hat{\mathbf{y}}$: "cat"

Supervised learning: in practice

x: 

$$\xrightarrow{f()} \quad f(\mathbf{x}) = \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix}$$

Supervised learning: in practice

x: 

$$\xrightarrow{\quad f() \quad} \mathrm{f}(\mathbf{x}) = \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} \neq \hat{\mathbf{y}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Supervised learning: in practice

x: 

$$\xrightarrow{f()} \quad f(\mathbf{x}) = \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} \neq \hat{\mathbf{y}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Loss

- Here, labels are encoded as one-hot-bit vectors,
- We compute a **loss** $\mathcal{L}(f(\mathbf{x}), \hat{\mathbf{y}})$
- Training consist in minimizing the loss!

Supervised learning: in practice

$\mathbf{x}$: 

$$\xrightarrow{f()} \mathrm{f}(\mathbf{x}) = \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} \neq \hat{\mathbf{y}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Loss

- Here, labels are encoded as one-hot-bit vectors,
- We compute a **loss** $\mathcal{L}(\mathrm{f}(\mathbf{x}), \hat{\mathbf{y}})$
  - Euclidean distance $\mathcal{L}(\mathrm{f}(\mathbf{x}), \hat{\mathbf{y}}) = \sum_{i=1}^{D} (\mathrm{f}(\mathbf{x})_i - \hat{\mathbf{y}}_i)^2$
  - Cross-entropy: $\mathcal{L}(\mathrm{f}(\mathbf{x}), \hat{\mathbf{y}}) = -\sum_{i=1}^{D} \hat{\mathbf{y}}_i \log(\mathrm{f}(\mathbf{x})_i)$
    $\Rightarrow$ To prevent the model to classify everything as one, outputs are **softmaxed**:
    $\mathrm{f}(\mathbf{x})_i = \frac{e^{\mathrm{f}(\mathbf{x})_i}}{\sum_{j=1}^{D} e^{\mathrm{f}(\mathbf{x})_j}}$
- Training consist in minimizing the loss!

Supervised learning: in practice

$\mathbf{x}$: 

$$\xrightarrow{f()} \quad f(\mathbf{x}) = \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} \neq \hat{\mathbf{y}} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Loss

- Here, labels are encoded as one-hot-bit vectors,
- We compute a **loss** $\mathcal{L}(f(\mathbf{x}), \hat{\mathbf{y}})$
- Training consist in minimizing the loss!

  $\Rightarrow$ Here, one can use gradient descent (see class 1.)

Supervised learning: in practice

$\mathbf{x}$:

$$\xrightarrow{f()} \quad \mathrm{f}(\mathbf{x}) = \begin{bmatrix} 0.2 \\ 0.8 \end{bmatrix} \neq \hat{\mathbf{y}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Loss

- Here, labels are encoded as one-hot-bit vectors,
- We compute a **loss** $\mathcal{L}(\mathrm{f}(\mathbf{x}), \hat{\mathbf{y}})$
- Training consist in minimizing the loss!

  $\Rightarrow$ Here, one can use gradient descent (see class 1.)

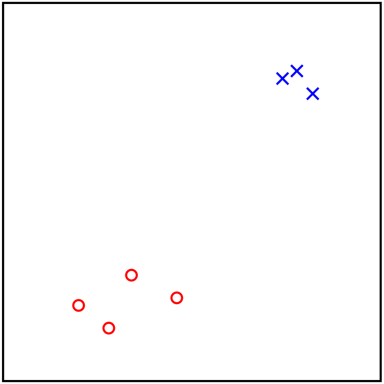
Examples

- **Classification** ($\hat{y}$ is categorical)
- **Regression** ($\hat{y}$ is scalar)
- Tons of applications:
  - Pattern recognition,
  - Prediction…

Supervised learning

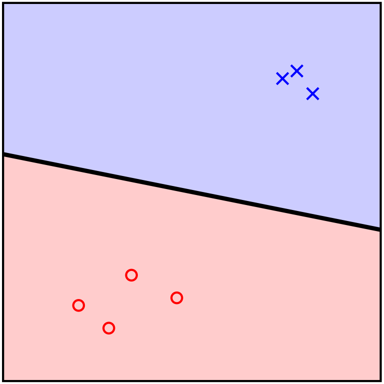
Examples

- **Classification** ($\hat{y}$ is categorical)
- **Regression** ($\hat{y}$ is scalar)
- Tons of applications:
  - Pattern recognition,
  - Prediction...



Examples

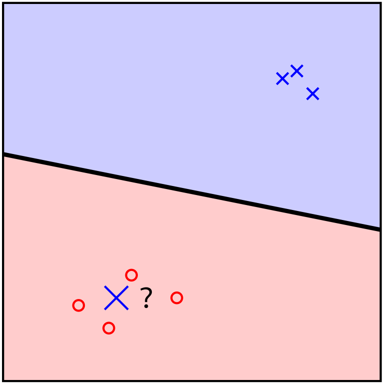
- **Classification** ($\hat{y}$ is categorical)
- **Regression** ($\hat{y}$ is scalar)
- Tons of applications:
  - Pattern recognition,
  - Prediction...



Supervised learning

Examples

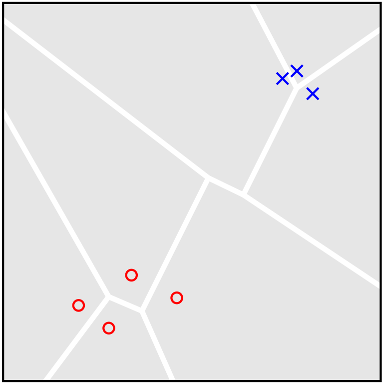
- **Classification** ($\hat{y}$ is categorical)
- **Regression** ($\hat{y}$ is scalar)
- Tons of applications:
  - Pattern recognition,
  - Prediction...



Supervised learning

Examples

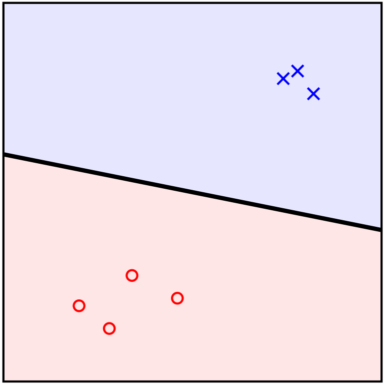
- **Classification** ($\hat{y}$ is categorical)
- **Regression** ($\hat{y}$ is scalar)
- Tons of applications:
  - Pattern recognition,
  - Prediction…



Challenges of supervised learning (1/5)

An ill-defined problem

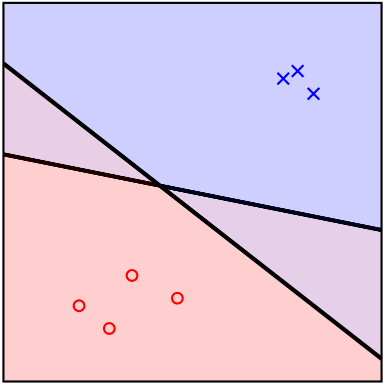
- An infinity of potential solutions, one must be the “best one” but is unreachable,
- $\Rightarrow$ requires **priors or constraints**.



Challenges of supervised learning (1/5)

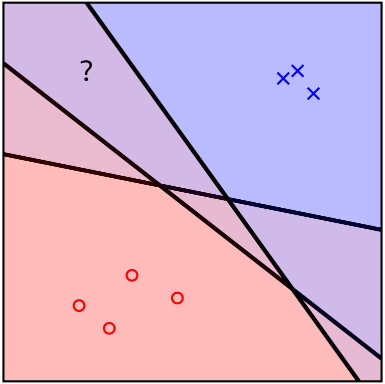
An ill-defined problem

- An infinity of potential solutions, one must be the "best one" but is unreachable,
- $\Rightarrow$ requires **priors or constraints**.



An ill-defined problem

- An infinity of potential solutions, one must be the “best one” but is unreachable,
- $\Rightarrow$ requires **priors or constraints**.

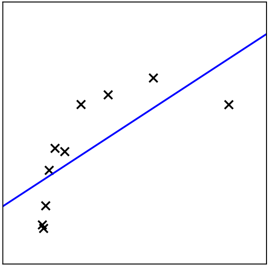


Bias/variance trade-off

- A **simple** solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: **overfitting** problem.

- Bias: Error from **erroneous assumptions in the learning algorithm**.
- Variance: Error from **sensitivity to small fluctuations** in the training set.

Polynomial regresssion,
d = 1 (under-fit; high bias)

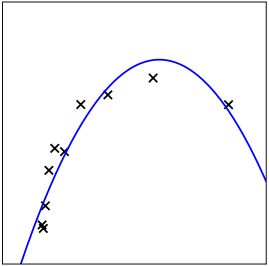


Challenges of supervised learning (2/5)

Bias/variance trade-off

- A **simple** solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: **overfitting** problem.

- Bias: Error from **erroneous assumptions in the learning algorithm**.
- Variance: Error from **sensitivity to small fluctuations** in the training set.

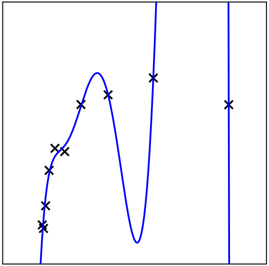


Challenges of supervised learning (2/5)

Bias/variance trade-off

- A **simple** solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: **overfitting** problem.

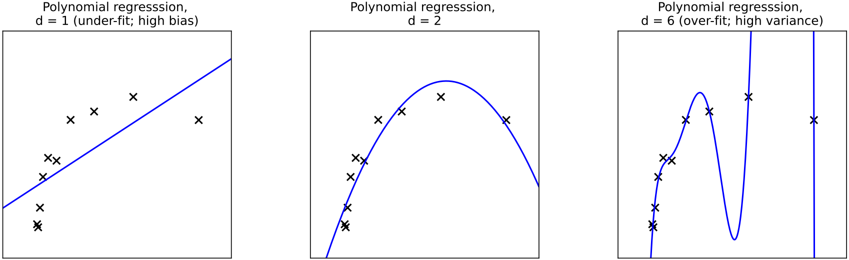
- Bias: Error from **erroneous assumptions in the learning algorithm**.
- Variance: Error from **sensitivity to small fluctuations** in the training set.



Challenges of supervised learning (2/5)

Bias/variance trade-off

- A **simple** solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: **overfitting** problem.



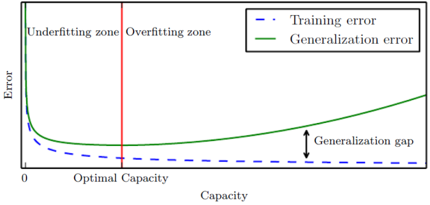
Challenges of supervised learning (2/5)

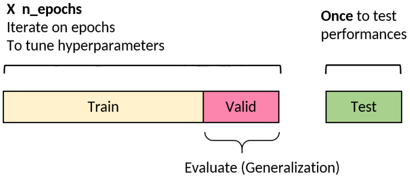
Bias/variance trade-off

- A **simple** solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: **overfitting** problem.

Crossvalidation

To detect overfitting, split training dataset in two parts, the first used to train, the second part to validate (Validation Set)





Challenges of supervised learning (3/5)

Curse of dimensionality

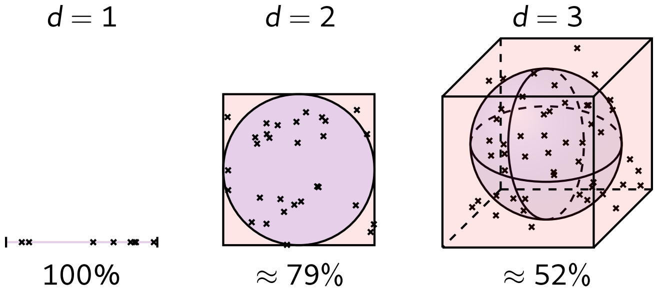
- Geometry is not intuitive in **high dimension**,
- Efficient methods in 2D are not necessarily still valid.

$$V_d^s = \frac{\pi^{d/2} R^d}{\Gamma(d/2+1)} \text{ versus } V_d^c = (2R)^d$$

see https://youtu.be/dZrGXYty3qc?t=533

Curse of dimensionality

- Geometry is not intuitive in **high dimension**,
- Efficient methods in 2D are not necessarily still valid.

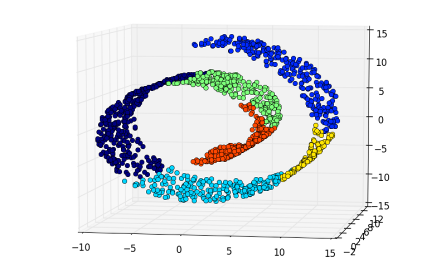


$$V_d^s = \frac{\pi^{d/2} R^d}{\Gamma(d/2+1)} \text{ versus } V_d^c = (2R)^d$$

see `https://youtu.be/dZrGXYty3qc?t=533`

Challenges of supervised learning (4/5)

Riemannian manifolds



The natural space of data may not always be suited to represent data!
$\Rightarrow$ Part of the reason why embeddings are richer semantically.

Challenges of supervised learning (5/5)

Computation time

Example on ImageNet, simply going through all images:

- $n = 10.000.000$, $d \approx 1.000.000$,
- $\approx 10^{13}$ elementary operations,
- $\approx$ 2h45 on a modern processor.

Scalability

- Finding the best solution to a problem would be feasible with unlimited computation time,
- But searching through the space of possible functions is often **untractable**,
- Solutions must be computationally reasonable, which is the true challenge today.

Challenges of supervised learning (5/5)

Computation time

Example on ImageNet, simply going through all images:

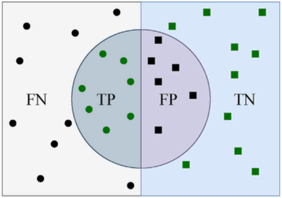
- $n = 10.000.000$, $d \approx 1.000.000$,
- $\approx 10^{13}$ elementary operations,
- $\approx$ 2h45 on a modern processor.

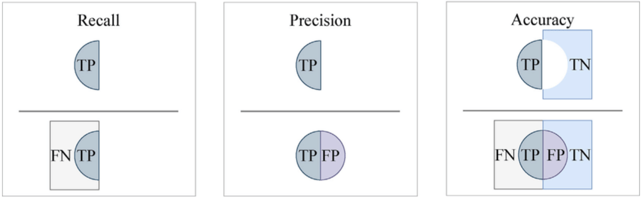
Scalability

- Finding the best solution to a problem would be feasible with unlimited computation time,
- But searching through the space of possible functions is often **untractable**,
- Solutions must be computationally reasonable, which is the true challenge today.

Metrics

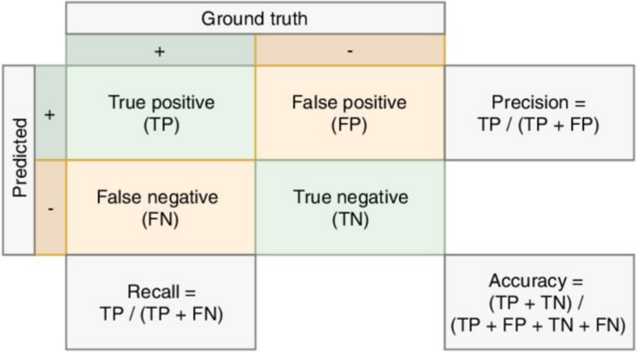
Accuracy, Precision and Recall





https://www.researchgate.net/publication/346129022_Overview_of_Machine_Learning_Part_1/figures

A useful tool: the confusion matrix



https://www.researchgate.net/publication/334840641_A_cloud_detection_algorithm_for_satellite_imagery_based_on_deep_learning/figures?lo=1

Lab Session 2 and assignments for Session 3

Lab Supervised Learning

- Basics of machine learning using sklearn (including new definitions / concepts)
- Tests on the modality chosen in Lab 1 (text, vision or audio), based on the same foundation model than in Lab 1.

Project 1 (P1)

You will choose a supervised learning method among those available (see Lab 2). You will present

- A brief description of the theory behind the method,
- Basic tests on this technique for your modality.

During Session 3 you will have 7 minutes to present.

Lab Session 2 and assignments for Session 3

List of Supervised Learning Methods

- Support Vector Machines (SVM)
- Decision Trees
- Multi-layer Perceptrons (MLP)
- Random Forest classifiers
- Logistic Regression
- Adaboost