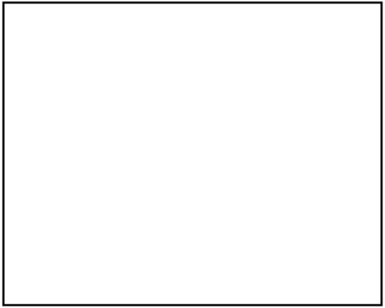
Course 2: Supervised Learning

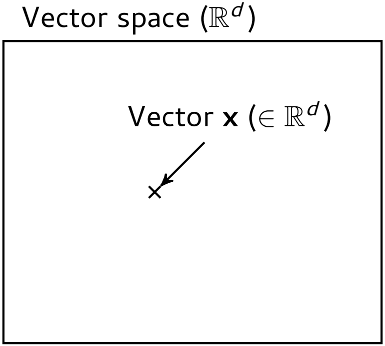
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Notations

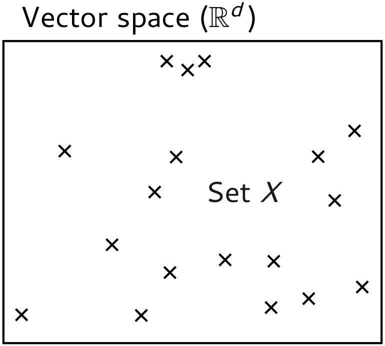
Vector space ($\mathbb{R}^d$)



Notations



Notations



Supervised learning

Machine learning

To learn is to **generalize** ($\neq$ memorize),

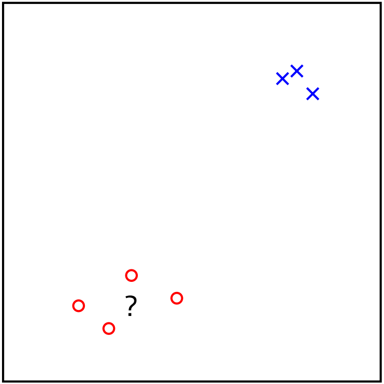
Supervised learning

Machine learning

To learn is to **generalize** ($\neq$ memorize),

Supervised learning

- Regression,
- Need of an expert,
- Tons of applications:
  - Pattern recognition,
  - Detection,
  - Prediction...



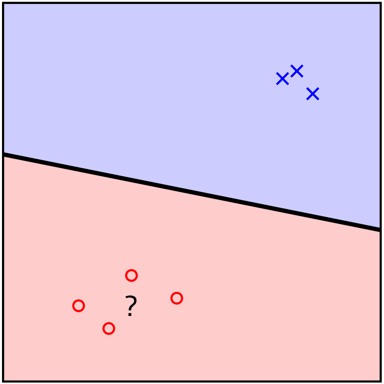
Supervised learning

Machine learning

To learn is to **generalize** ($\neq$ memorize),

Supervised learning

- Regression,
- Need of an expert,
- Tons of applications:
  - Pattern recognition,
  - Detection,
  - Prediction...



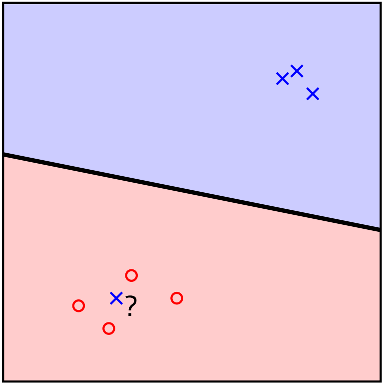
Supervised learning

Machine learning

To learn is to **generalize** ($\neq$ memorize),

Supervised learning

- Regression,
- Need of an expert,
- Tons of applications:
    - Pattern recognition,
    - Detection,
    - Prediction...



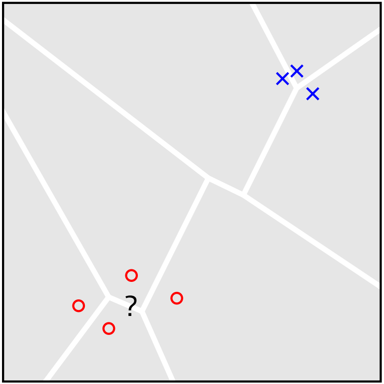
Supervised learning

Machine learning

To learn is to **generalize** ($\neq$ memorize),

Supervised learning

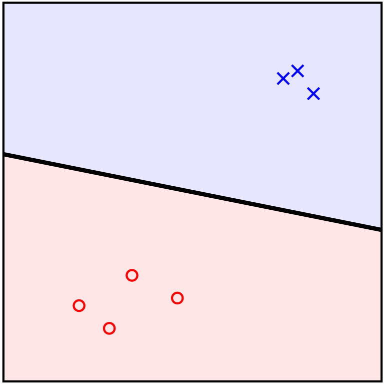
- Regression,
- Need of an expert,
- Tons of applications:
  - Pattern recognition,
  - Detection,
  - Prediction...



Challenges of supervised learning (1/5)

An ill-defined problem

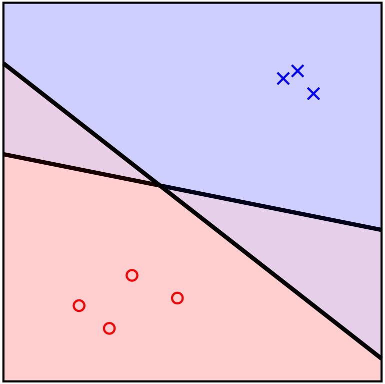
- An infinity of potential solutions, one must be the “best one” but is unreachable,
- $\Rightarrow$ requires a priori, constraints.



Challenges of supervised learning (1/5)

An ill-defined problem

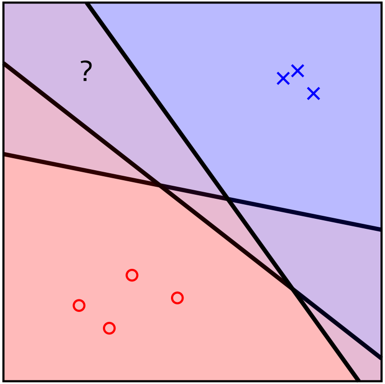
- An infinity of potential solutions, one must be the "best one" but is unreachable,
- $\Rightarrow$ requires a priori, constraints.



Challenges of supervised learning (1/5)

An ill-defined problem

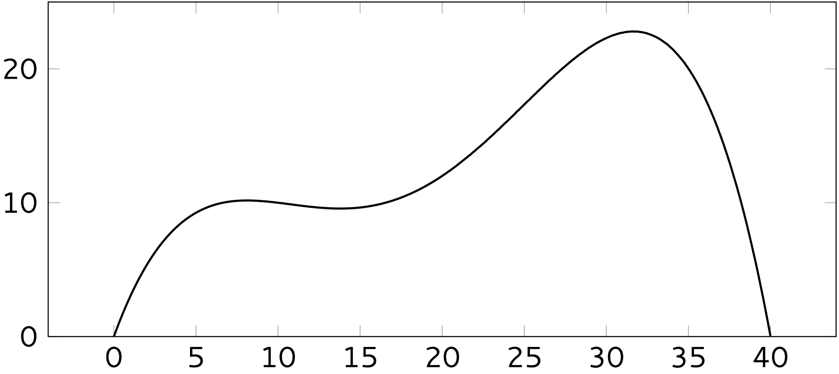
- An infinity of potential solutions, one must be the "best one" but is unreachable,
- $\Rightarrow$ requires a priori, constraints.



Challenges of supervised learning (2/5)

Bias/variance trade-off

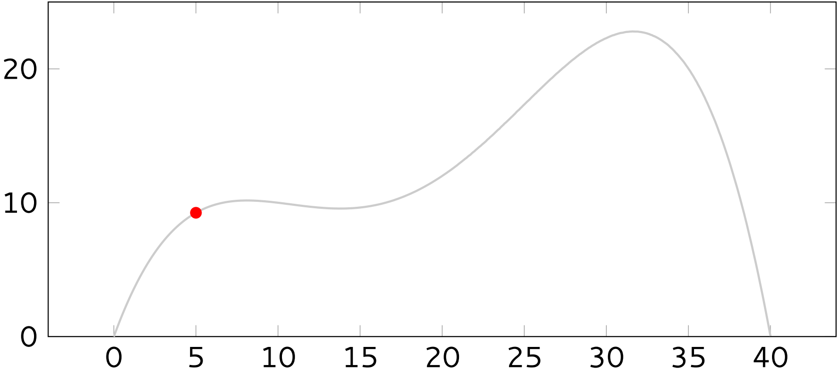
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

Bias/variance trade-off

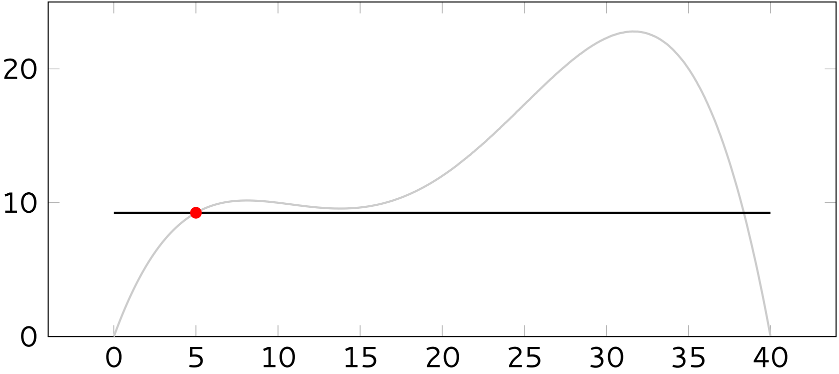
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

Bias/variance trade-off

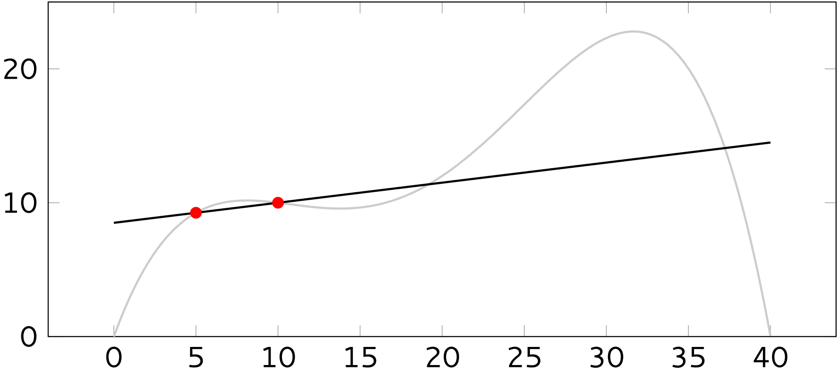
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

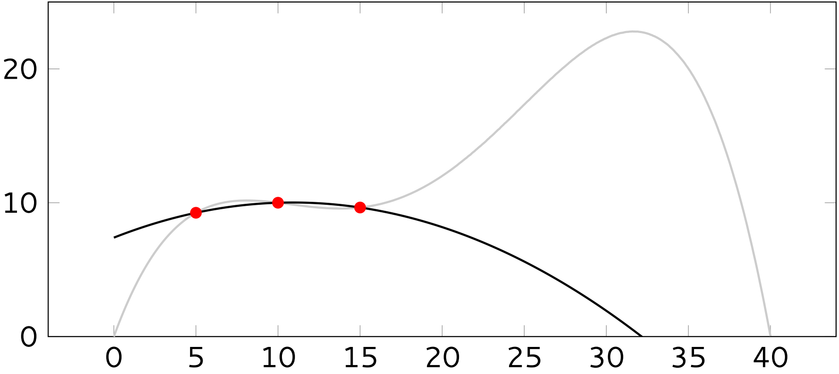
Bias/variance trade-off

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



Bias/variance trade-off

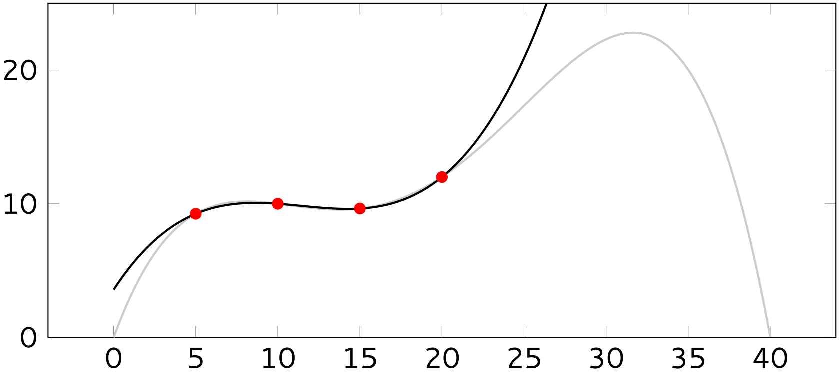
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

Bias/variance trade-off

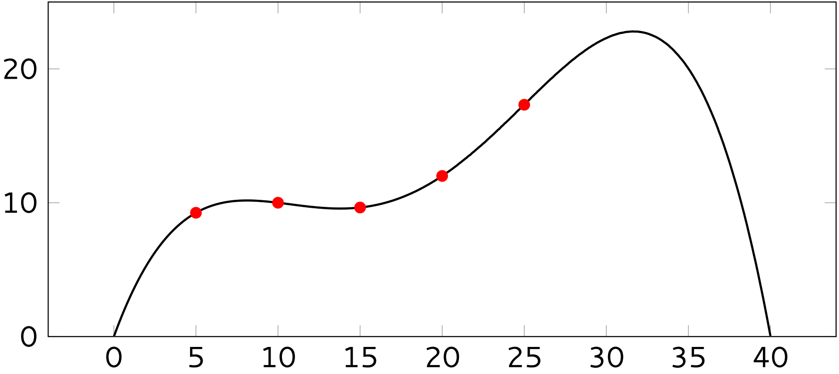
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

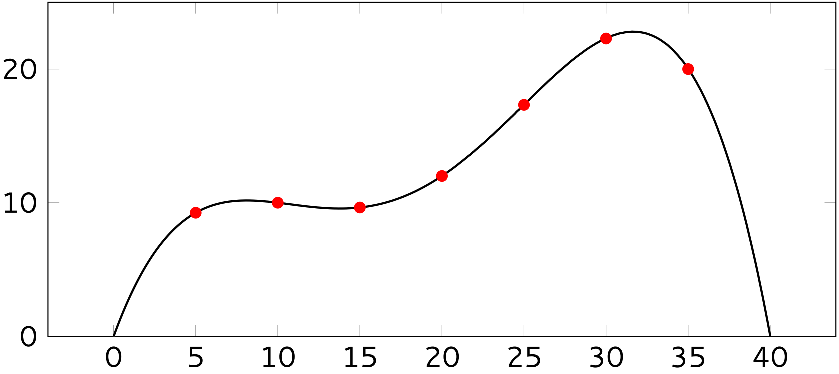
Bias/variance trade-off

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



Bias/variance trade-off

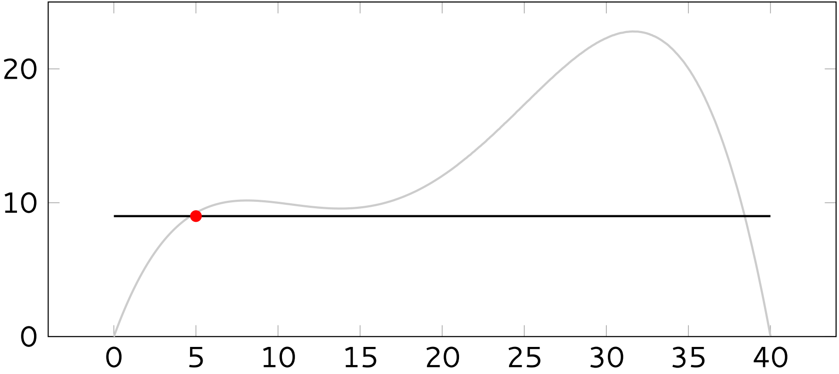
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

Bias/variance trade-off

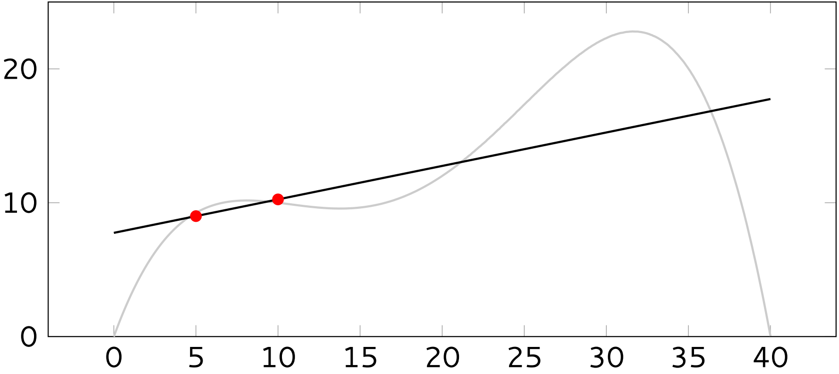
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

Bias/variance trade-off

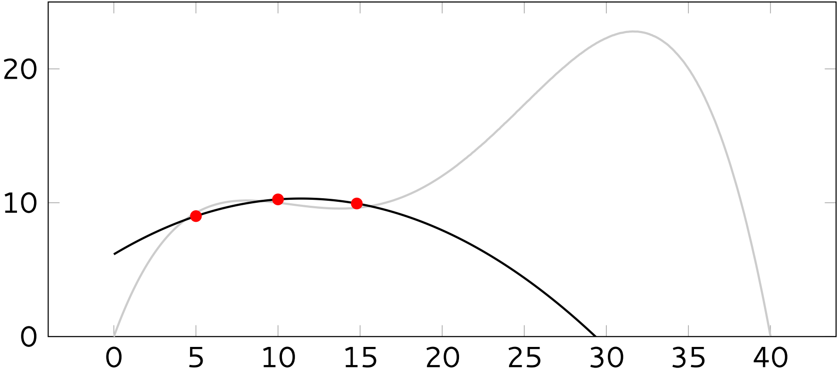
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

Bias/variance trade-off

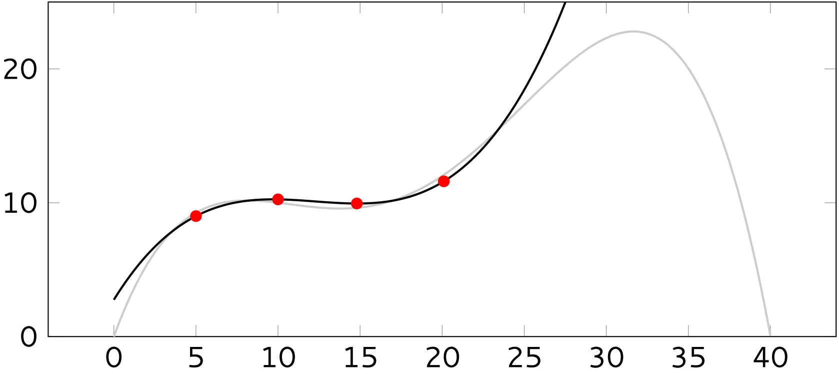
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

Bias/variance trade-off

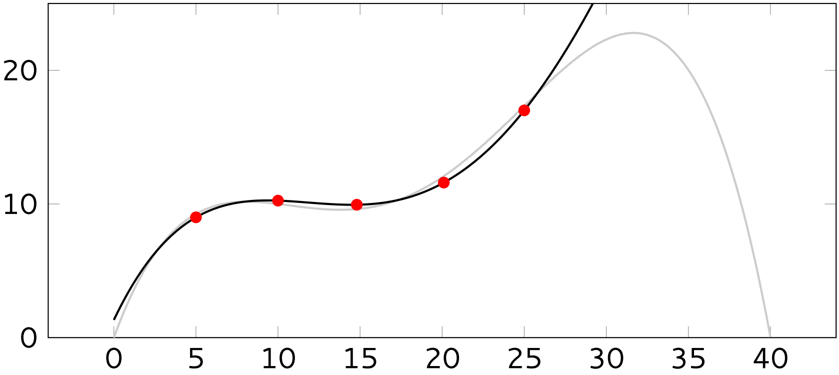
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

Bias/variance trade-off

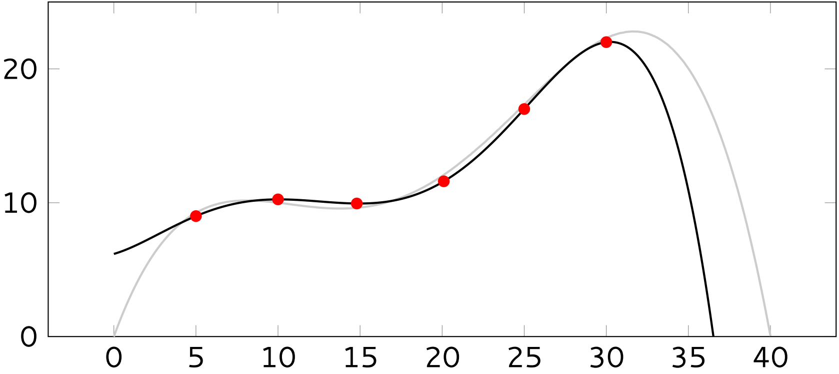
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

Bias/variance trade-off

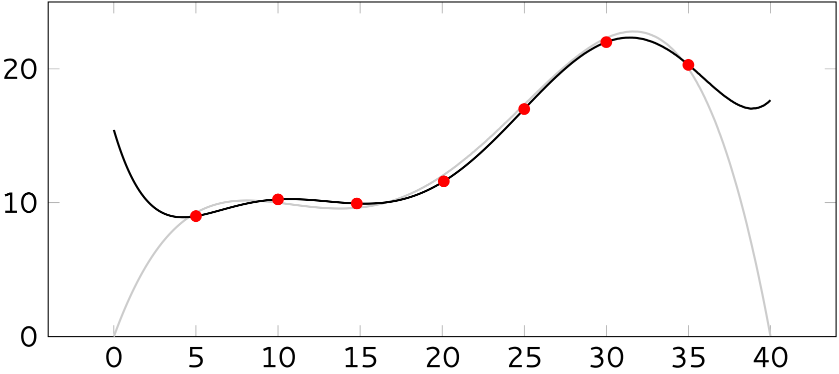
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

Bias/variance trade-off

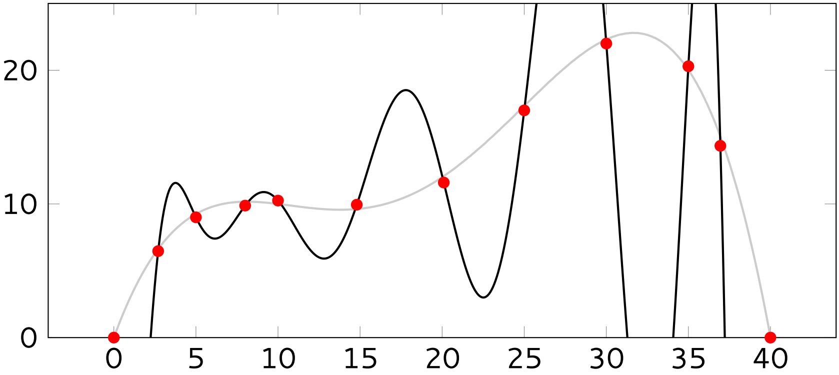
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



Challenges of supervised learning (2/5)

Bias/variance trade-off

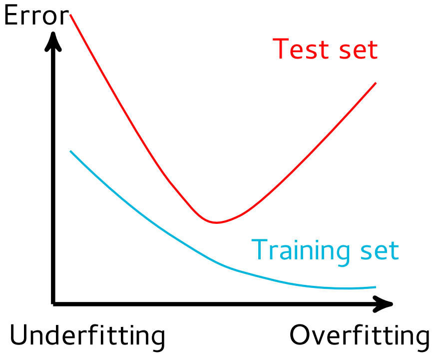
- A simple solution that almost matches is better than a complex one that fully matches,
- Mimicking is not learning: overfitting problem.



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 avoid overfitting, split training dataset in two parts:
  1. A first part is used to train,
  2. A second part is used to validate,

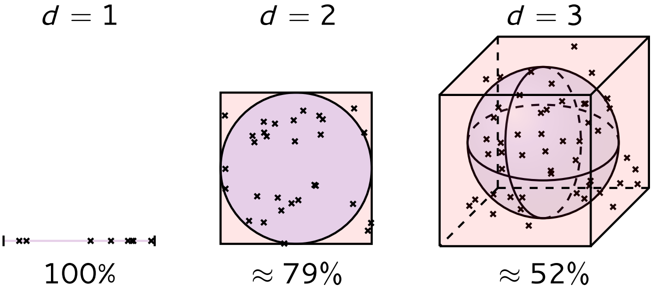
Challenges of supervised learning (3/5)

Curse of dimensionality

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

Curse of dimensionality

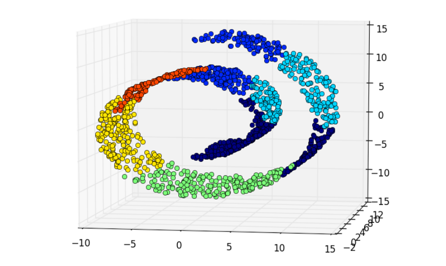
- Geometry is not intuitive in high dimension,
- Efficient methods in 2D are not necessarily still valid.

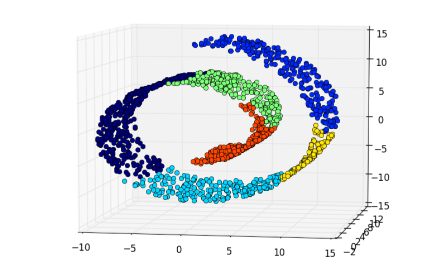


$$V_d^s = \frac{\pi^{d/2} R^d}{\Gamma(d/2+1)} \text{ versus } V_d^c = (2R)^d$$

Challenges of supervised learning (4/5)

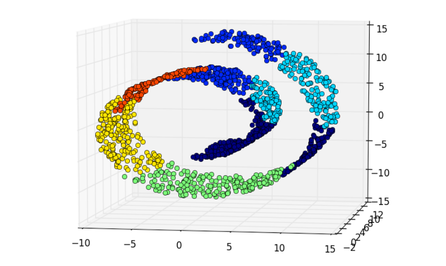
Riemannian manifolds

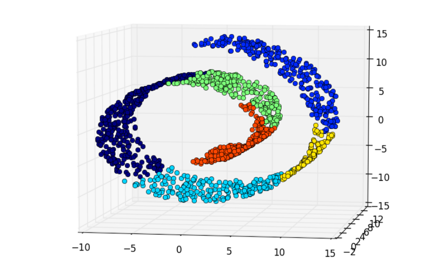




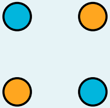
Challenges of supervised learning (4/5)

Riemannian manifolds





Linear separability and need for embedding



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.

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.

Tractability limitation

- 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.

VC dimension

Definition

- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

VC dimension

Definition

- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

Consider for example lines to shatter set of points with $d = 2$.

VC dimension

Definition

- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

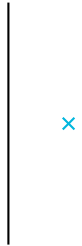
Consider for example lines to shatter set of points with $d = 2$.

VC dimension

Definition

- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

Consider for example lines to shatter set of points with $d = 2$.



VC dimension

Definition

- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

Consider for example lines to shatter set of points with $d = 2$.

VC dimension

Definition

- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

Consider for example lines to shatter set of points with $d = 2$.

VC dimension

Definition

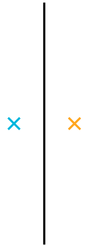
- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

Consider for example lines to shatter set of points with $d = 2$.

Definition

- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

Consider for example lines to shatter set of points with $d = 2$.



VC dimension

Definition

- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

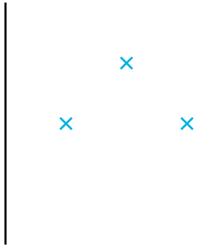
Consider for example lines to shatter set of points with $d = 2$.

VC dimension

Definition

- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

Consider for example lines to shatter set of points with $d = 2$.



VC dimension

Definition

- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

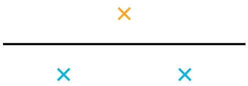
Consider for example lines to shatter set of points with $d = 2$.

VC dimension

Definition

- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

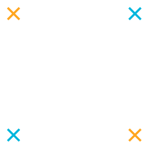
Consider for example lines to shatter set of points with $d = 2$.



Definition

- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

Consider for example lines to shatter set of points with $d = 2$.



VC dimension

Definition

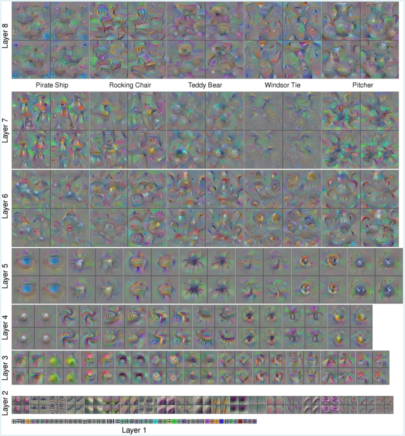
- Let us fix d,
- The VC dimension is a measure of the genericity of a method,
- It is the maximum cardinality of a set of vectors it is able to shatter in any possible way.

Consider for example lines to shatter set of points with $d = 2$.

VC is 3.

Open Questions

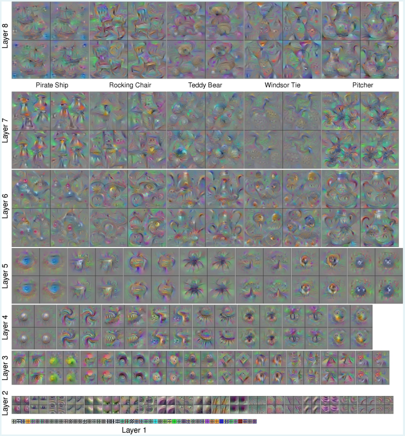
Interpretability



Source: "Understanding Neural Networks Through Deep Visualization"

Open Questions

Interpretability



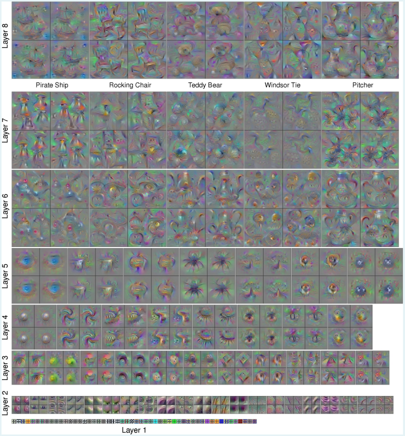
Source: "Understanding Neural Networks Through Deep Visualization"

Few examples

Source: "How to grow a mind: statistics, structure, and abstraction"

Open Questions

Interpretability



Source: "Understanding Neural Networks Through Deep Visualization"

Few examples

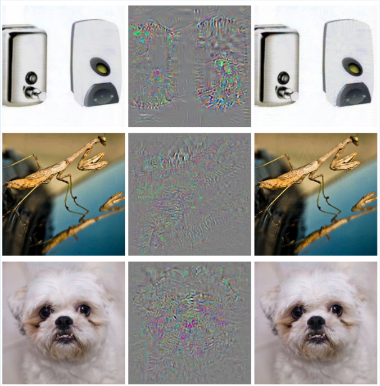
Source: "How to grow a mind: statistics, structure, and abstraction"

Other open questions

- Incremental learning
- Irregular domains
- Choice of hyperparameters...

The Problem of Overfitting

Problems with crossvalidation



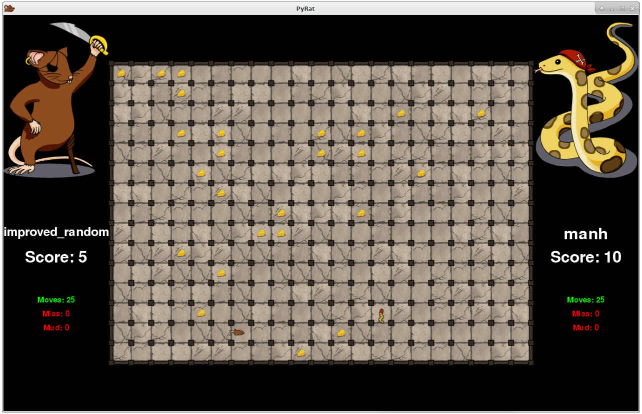
Source: "Intriguing properties of neural networks"

The Problem of Overfitting

Problems with crossvalidation

Source: "Intriguing properties of neural networks"

Non-symmetric PyRat without walls / mud



Supervised learning - predict the outcome of a game from the start configuration.

Lab Session 2 and assignments for Session 3

TP Supervised Learning (TP1)

- Generating PyRat Datasets
- Basics of machine learning using sklearn
- Tests on PyRat datasets using a naive approach

Project 1 (P1)

You will be assigned a supervised learning method. You have to prepare a Jupyter Notebook on this method, including:

- A brief description of the theory behind the method,
- Basic tests on simulated data to show the influence of parameters and hyperparameters
- Advanced tests and analysis on your own PyRat Datasets

During Session 3 you will have 7 minutes to present your notebook.