From Complexity to Intelligence

Machine Learning and Complexity

Pierre-Alexandre Murena

TELECOM ParisTech

Table of contents

Table of contents

A basic approach of learning

A definition (T. Mitchell, 1997)

A computer program is said to learn from experience $\mathcal{E}$ with respect to some class of tasks $\mathcal{T}$ and performance measure $\mathcal{P}$, if its performance at tasks in $\mathcal{T}$, as measured by $\mathcal{P}$, improves with experience $\mathcal{E}$.

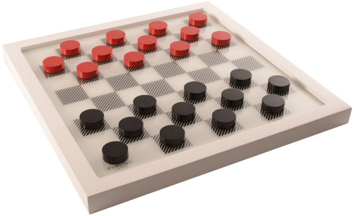
Examples

Handwriting recognition

- **Task :** recognize and label handwritten words in images
- **Performance measure :** percentage of words successfully labeled
- **Experience :** database of manually labeled handwritten words

Examples

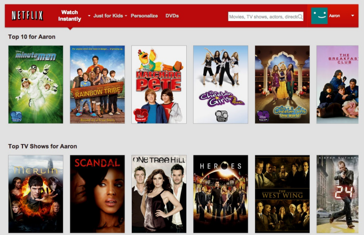
Checkers



- **Task :** play checkers
- **Performance measure :** percentage of victories
- **Experience :** practice games against itself

Examples

Video recommendation



- **Task :** recommend to any user videos he might like
- **Performance measure :** percentage of recommendation success
- **Experience :** list of videos liked by a set of users

A formal model

- **Input space :** a set $\mathcal{X}$
- **Output space :** a set $\mathcal{Y}$
- **Training data :** $\mathcal{D}_S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
- **Decision function :** a function $h : \mathcal{X} \mapsto \mathcal{Y}$

Knowing the data $\mathcal{D}_S$, the system aims at learning the function h.

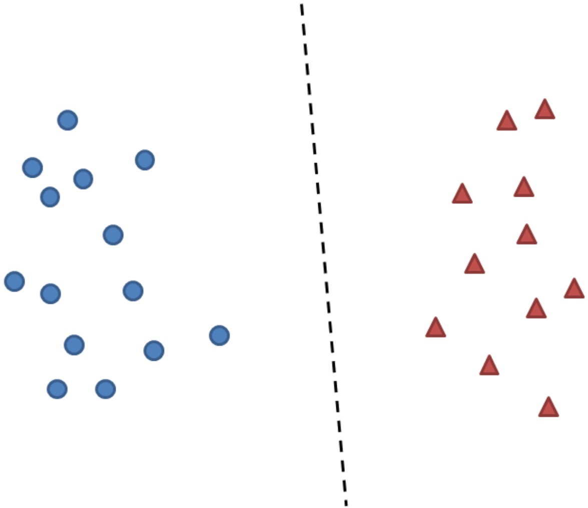
Table of contents

Supervised vs Unsupervised

- In **Supervised Learning**, the labels $y \in \mathcal{Y}$ are given. The goal is to estimate a correct labelling function $h : \mathcal{X} \mapsto \mathcal{Y}$.
- In **Unsupervised Learning**, the labels are unknown. The purpose is to group *similar* points.
- In **Semi-Supervised Learning**, some labels are unknown. The purpose is to estimate a correct labelling function h, exploiting information brought by non labelled points.

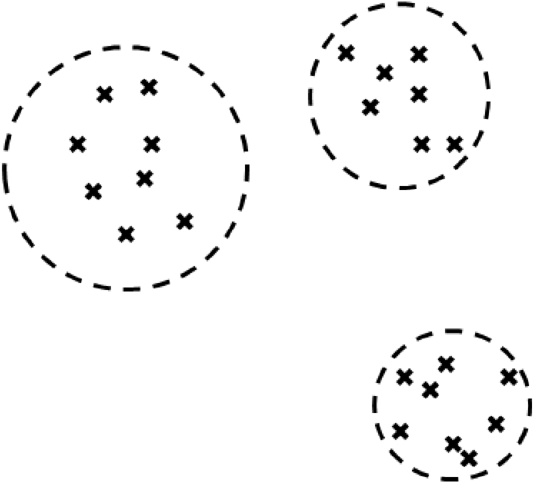
Supervised vs Unsupervised

Supervised Learning

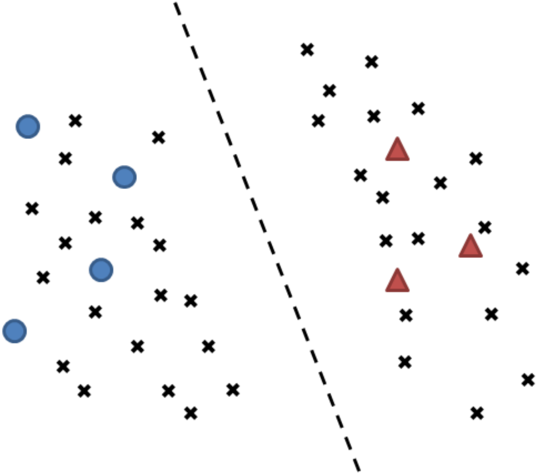


Supervised vs Unsupervised

Unsupervised Learning



Supervised vs Unsupervised
Semi-Supervised Learning

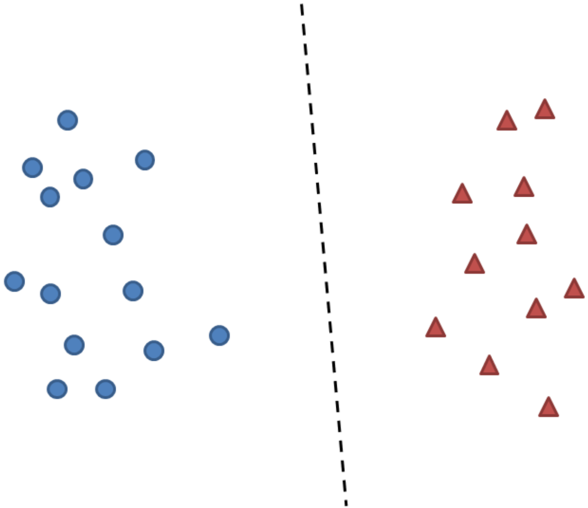


Classification vs Regression

- In **classification**, the output set $\mathcal{Y}$ is discrete (and finite).
- In **regression**, the output set $\mathcal{Y}$ is continuous.

Classification vs Regression

Classification



Classification vs Regression
Regression

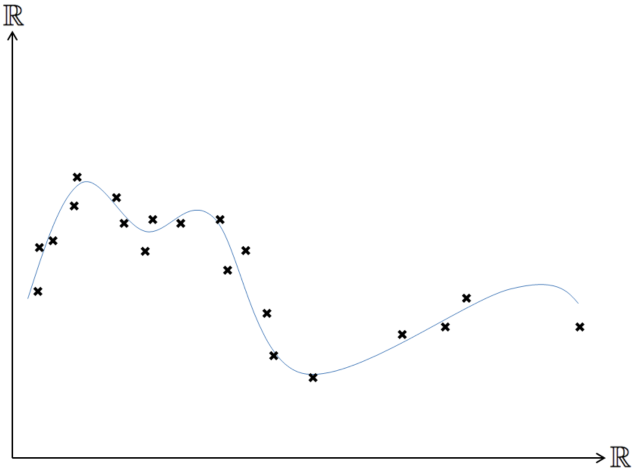


Table of contents

What is Unsupervised Learning ?

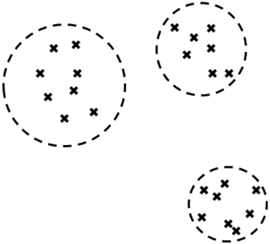
Reminder

In Unsupervised Learning, the learner receives unlabeled input data and aims at *finding a structure* for these data.

Tasks in Unsupervised Learning

- **Clustering :** grouping a set of objects such that similar objects end up in the same group and dissimilar objects are separated into different groups.
- **Anomaly detection :** identifying objects which do not conform to the global behavior.

Clustering

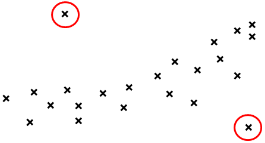


Basic idea : Points which are close are similar ; Points which are far are dissimilar.

Applications :

- *Marketing :* detect groups of users with similar behaviors
- *Medicine :* detect mutations of a virus
- *Visualization :* find similar land-use on a satellite picture

Anomaly Detection



Basic idea : Find a general rule describing data and isolate points which do not obey this rule.

Applications :

- *Fraud detection*
- *Networks :* intrusion detection, event detection...

Unsupervised learning = Compression

Idea

In both Clustering and Anomaly Detection, the problem is to find regularities / structure.

Finding structure = Compressing the description of data

Hence, Unsupervised Learning = Compression

Besides, unsupervised learning is just a redescription of data, so is not directly a problem of induction.

K-Means algorithm

Inputs : Dataset $X = \{X_1, \ldots, X_n\}$; Number of clusters k

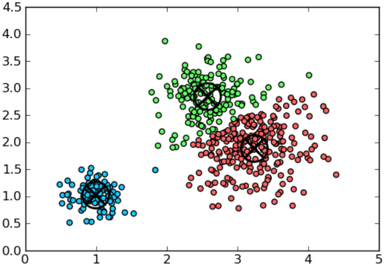
Initialization : Randomly choose initial centroids $\mu_1, \ldots, \mu_k$

Repeat until convergence :

- For all $i \leq k$, set $C_i = \{x \in X; i = \text{argmin}_j \|x - \mu_j\|\}$
- For all $i \leq k$, update $\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$

Compression in Clustering

Prototype models



The data points are not described by their **absolute position** but by their **relative position to the closest prototype**.

Compression in Anomaly Detection

Applying MDL principle : find a model M minimizing $C(M) + C(D|M)$

x is an anomaly if $C(x|M) \approx C(x)$

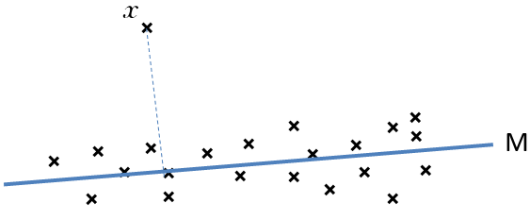


Table of contents

A well-known principle : Bayesianism

Bayesianism is based on Bayes rule :

$$P(M|D) = \frac{P(M) \times P(D|M)}{P(D)}$$

- **Maximum A Posteriori (MAP)** :

$$\widehat{h}_{MAP} = \text{argmax}_h \quad \{P(h|D) \times P(h)\}$$

- **Maximum Likelihood (ML)** :

$$\widehat{h}_{ML} = \text{argmax}_h \quad P(D|h)$$

MDL Principle

The best theory to describe observed data is the one which minimizes the sum of the description length (in bits) of :

- the theory description
- the data encoded from the theory

$$\widehat{h} = \text{argmin}_h \quad K(h) + K(D|h)$$

or

$$\widehat{h} = \text{argmin}_h \quad C(h) + C(D|h)$$

MDL and Bayesianism

Using the prefix complexity K, MDL principle is equivalent to Bayes rule :

$$K(h) + K(D|h) = -\log P(h) - \log P(D|h)$$

Thus :

$$\text{argmin}_h\{K(h) + K(D|h)\} = \text{argmax}_h\{\log P(h) + \log P(D|h)\}$$

Model selection : penalization

An even more general principle !

$$K(M) + K(X|M) + K(\beta|X, M) + K(Y|\beta, X, M)$$

Table of contents

Table of contents

A probabilistic notation

- Suppose that data $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ are generated according to a probability distribution $\mathbb{P}_{X \times Y}$.
- Consider a *loss function* $l : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ which quantifies the “cost” of misclassification
- We define the risk of a classifier $h : \mathcal{X} \mapsto \mathcal{Y}$ as :

$$R(h) = \int_{\mathcal{X} \times \mathcal{Y}} l(h(x), y) d\mathbb{P}_{X \times Y}(x, y)$$

- **Question :** can we find an algorithm which will *always* infer good hypotheses ?

The no-free-lunch theorem
Wolpert's answer

No !

The no-free-lunch theorem
[Wolpert, 1996]

For any two learning algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ with posterior distributions $p_1(h|\mathcal{S})$ and $p_2(h|\mathcal{S})$ (where $\mathcal{S}$ is a data set), for any distribution $\mathbb{P}_{\mathcal{X}}$ of data and for any number m of data, the following propositions are true :

1. In uniform average over all target functions $f \in \mathcal{F}$: $\mathbb{E}_1[R|f, m] - \mathbb{E}_2[R|f, m] = 0$
2. For any given learning set $\mathcal{S}$, in uniform average over all target functions $f \in \mathcal{F}$: $\mathbb{E}_1[R|f, \mathcal{S}] - \mathbb{E}_2[R|f, \mathcal{S}] = 0$
3. In uniform average over all possible distributions $P(f)$: $\mathbb{E}_1[R|f] - \mathbb{E}_2[R|f] = 0$
4. For any given learning set $\mathcal{S}$, in uniform average over all possible distributions $P(f)$: $\mathbb{E}_1[R|\mathcal{S}] - \mathbb{E}_2[R|\mathcal{S}] = 0$

The no-free-lunch theorem
[Wolpert, 1996]

Consequences of the no-free-lunch theorem

- A “good” classification algorithm will have **in average** the same performance as a “bad” classification algorithm (*average over the space of problems*) if all target functions f are equiprobable.
- For any region of the space of problems where an algorithm $\mathcal{A}$ is good, there exists a region where $\mathcal{A}$ is bad.

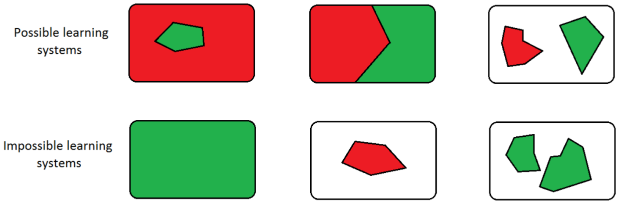


Table of contents

Reminder : the ERM principle

Given a loss function $l : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$ and a classifier h, we can define :

- The risk of h :

$$R(h) = \int_{\mathcal{X} \times \mathcal{Y}} l(h(x), y) d\mathbb{P}_{X,Y}(x, y)$$

- The empirical risk of h :

$$\widehat{R_n}(h) = \frac{1}{n} \sum_{i=1}^{n} l(h(x_i), y_i)$$

ERM principle : $\widehat{h} = \arg\min_h \widehat{R_n}(h)$

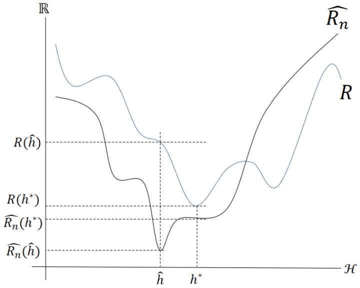
Is ERM legit ?

1. Is the hypothesis $\widehat{h}$ good in the real risk ?

$$\widehat{R_n}(\widehat{h}) \overset{?}{\longleftrightarrow} R(\widehat{h})$$

2. Am I far from the real optimum ($h^* = \arg\min_h R(h)$) ?

$$R(\widehat{h}) \overset{?}{\longleftrightarrow} R(h^*)$$



Probabilities help us answer these questions.

PAC learning

Leslie Valiant (1949-...)

The purpose of PAC learning is to select **with high probability** (*probably)* a hypothesis **with low generalization error** (*approximately correct*).

PAC = Probably Approximately Correct

Is ERM legit ?

Let's choose our hypothesis in a finite set $\mathcal{H}$. Then for all $h \in \mathcal{H}, \delta \in [0,1]$:

$$P^m \left[R(h) \leq \widehat{R_m}(h) + \frac{\ln|\mathcal{H}| + \ln\frac{1}{\delta}}{m} \right] > 1 - \delta$$

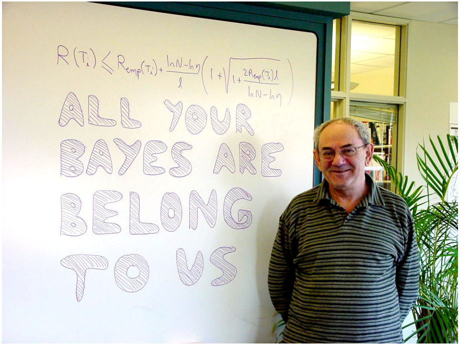
Oracle inequality :

For any $\delta \in [0,1]$:

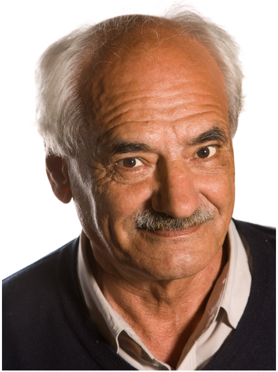
$$P^m \left[R(\widehat{h_m}) \leq R(h^*) + \sqrt{\frac{2}{n} \ln\left(\frac{2|\mathcal{H}|}{\delta}\right)} \right] > 1 - \delta$$

Is ERM legit ?

What if the hypothesis space is infinite ?



Vladimir Vapnik (1936-...)



Alexei Chervonenkis (1938-2014)

Is ERM legit ?

Step 3 : What if the hypothesis space is infinite ?

Vapnik-Chervonenkis theory

Let $\mathcal{H}$ be a Vapnik-Chervonenkis class. Then for any $\delta \in [0,1]$:

$$P\left[R(\widehat{h_m}) \leq R(h^*) + 4\sqrt{\frac{2(V_{\mathcal{H}}\ln(m+1)+\ln 2)}{m}} + \sqrt{\frac{2\ln\frac{1}{\delta}}{m}}\right] > 1-\delta$$

and :

$$P\left[|R(\widehat{h_m}) - \widehat{R_n}(\widehat{h})| \leq 2\sqrt{\frac{2(V_{\mathcal{H}}\ln(m+1)+\ln 2)}{m}} + \sqrt{\frac{\ln\frac{1}{\delta}}{2m}}\right] > 1-\delta$$

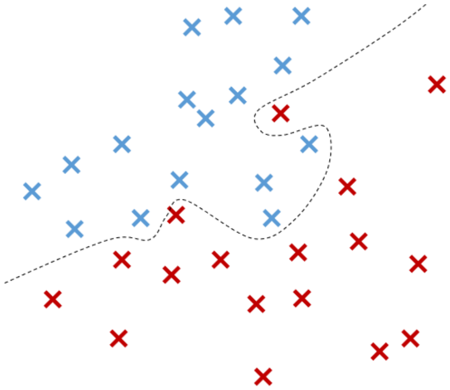
A similar result for MDL

Theorem

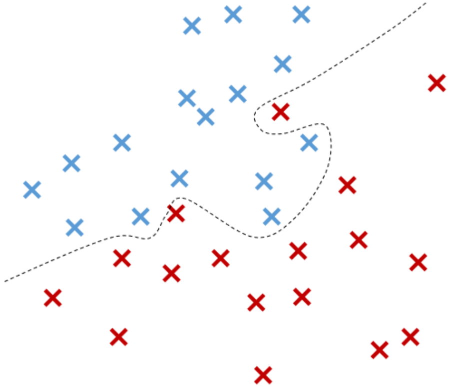
Let $\mathcal{H}$ be a hypothesis class and let $d : \mathcal{H} \to \{0,1\}^*$ be a prefix-free description language for $\mathcal{H}$. Then, for every sample size m, every confidence parameter $\delta > 0$ and every probability distribution $\mathcal{D}$, with probability greater than $1 - \delta$ over the choice of $S \sim \mathcal{D}^m$, we have that :

$$\forall h \in \mathcal{H}, L_{\mathcal{D}}(h) \leq L_S(h) + \sqrt{\frac{|h| + \ln(2/\delta)}{2m}}$$

MDL and overfitting

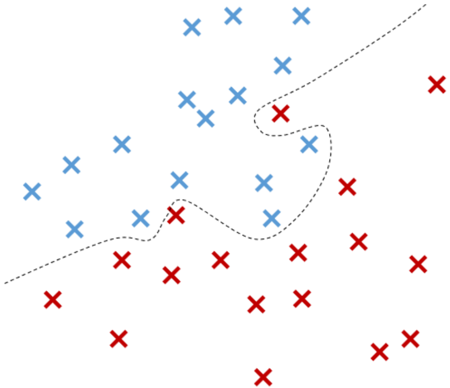


MDL and overfitting



MDL naturally prevents overfitting !

MDL and overfitting



MDL naturally prevents overfitting !

But was it intended... ?

Table of contents

From particular to general
Answered questions ?

Isn't this question of generalization already answered by PAC learning, VC theory etc... ?

From particular to general

Answered questions ?

Isn't this question of generalization already answered by PAC learning, VC theory etc... ?

Yes and no !

These theories are valid only for the limit case of i.i.d. data **and i.i.d. questions**

From particular to general

Toward new principles ?

1. **The learner is not indifferent to the future question** : the *priors* over the future are my only guarantee of generalization ?
2. **All previously encountered data, problems and knowledge have a maximal pertinence** : Asymptotic results in statistical learning and Solomonoff's induction theories ? Creation of knowledge by one-shot learning ?

From particular to particular

Back to Analogy Reasoning

$$\textbf{ABC} \Longrightarrow \textbf{ABD}$$
$$\textbf{IJK} \Longrightarrow ?$$

From particular to particular

Back to Analogy Reasoning

$$\textbf{ABC} \Longrightarrow \textbf{ABD}$$
$$\textbf{IJK} \Longrightarrow ?$$

The problem can be formulated with the machine learning notations :

$$X_{learn} \Longrightarrow Y_{learn}$$
$$X_{test} \Longrightarrow ?$$

This problem has a name : **transfer learning**

From particular to particular

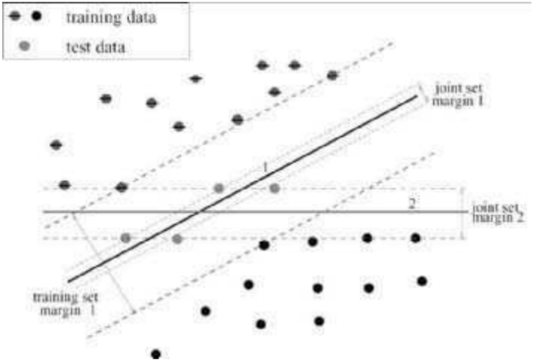
Transductive Learning

Solving a problem of interest, do not solve a more general (and therefore worse-posed) problem as an intermediate step. Try to get the answer that you really need but not a more general one.

- Do not estimate a density if you need to estimate a function. *(Do not use classical generative models ; use ML predictive models.)*
- Do not estimate a function if you need to estimate values at given points. *(Try to perform transduction, not induction)*
- Do not estimate predictive values if your goal is to act well. *(A good strategy of action can rely just on good selective inference.)*

From particular to particular
Transductive Learning

Transduction = Transfer with i.i.d. hypothesis



From particular to particular

An equation (with familiar terms...)

$$C(M_S) + C(X_S|M_S) + C(\beta_S|M_S, X_S) + C(Y_S|M_S, X_S, \beta_S)$$
$$+ C(M_T|M_S) + C(X_T|M_T)$$

From particular to particular

An equation (with familiar terms...)

$$C(M_S) + C(X_S|M_S) + C(\beta_S|M_S, X_S) + C(Y_S|M_S, X_S, \beta_S) \\ + C(M_T|M_S) + C(X_T|M_T)$$

- $C(M)$: prior
- $C(X|M)$: likelihood
- $C(Y|M, X, \beta)$: risk
- $C(M_T|M_S)$: transfer term (related to a prior ?)

From particular to general

An intimidating gap

In many problems, I don't know the future test data ! Transduction is not possible... And our equation is not valid anymore...

- What does it mean *to generalize well* from a complexity point of view ?
- Is it enough to write that $X_T = \langle\rangle$?
- Our equation seems still valid (the individual terms are used in classical inductive principles.)

Table of contents

What to remember ?

- Induction is **definitely not** a simple problem !
- Compression is closely related to learning
- The no-free-lunch theorem : no miracle classifier !
- MDL is hidden **everywhere** in Machine Learning
- New principles are necessary to formalize the transition from the particular to the general

What to remember ?

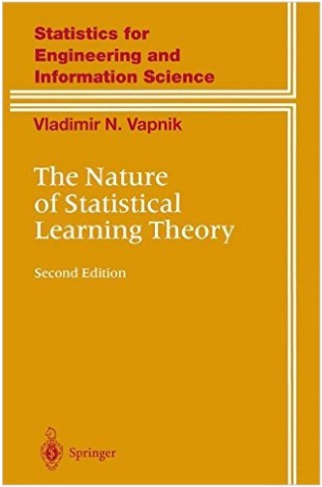
- Induction is **definitely not** a simple problem !
- Compression is closely related to learning
- The no-free-lunch theorem : no miracle classifier !
- MDL is hidden **everywhere** in Machine Learning
- New principles are necessary to formalize the transition from the particular to the general

But...

- Most of these questions are never addressed in ML courses
- Most people prefer focusing on algorithms
- Most people ignore that such problems exist

That's why...

If you are interested...



Licence de droits d'usage

Contexte public } sans modifications