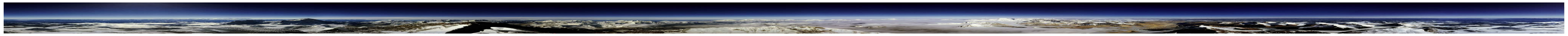
Induction and Machine Learning

What the **second** tells about the **first**
and **is induction finally a closed problem?**



Antoine Cornuéjols

AgroParisTech – INRA MIA 518

antoine.cornuejols@agroparistech.fr

Role of induction

- [Leslie Valiant, « *Probably Approximately Correct. Nature's Algorithms for Learning and Prospering in a Complex World* », Basic Books, 2013]

« From this, we have to conclude that **generalization** or **induction** is a **pervasive phenomenon** (…). It is as routine and reproducible a phenomenon as objects falling under gravity.

It is **reasonable to expect a quantitative scientific explanation** of this highly reproducible phenomenon. »

Role of induction

- [Edwin T. Jaynes, « *Probability theory. The logic of science* », Cambridge U. Press, 2003], p.3

« We are hardly able to get through one waking hour without facing some situation (e.g. *will it rain or won't it?*) where we **do not have enough information to permit deductive reasoning**; but still we must decide immediately.

In spite of its familiarity, the formation of plausible conclusions is a **very subtle process**. »

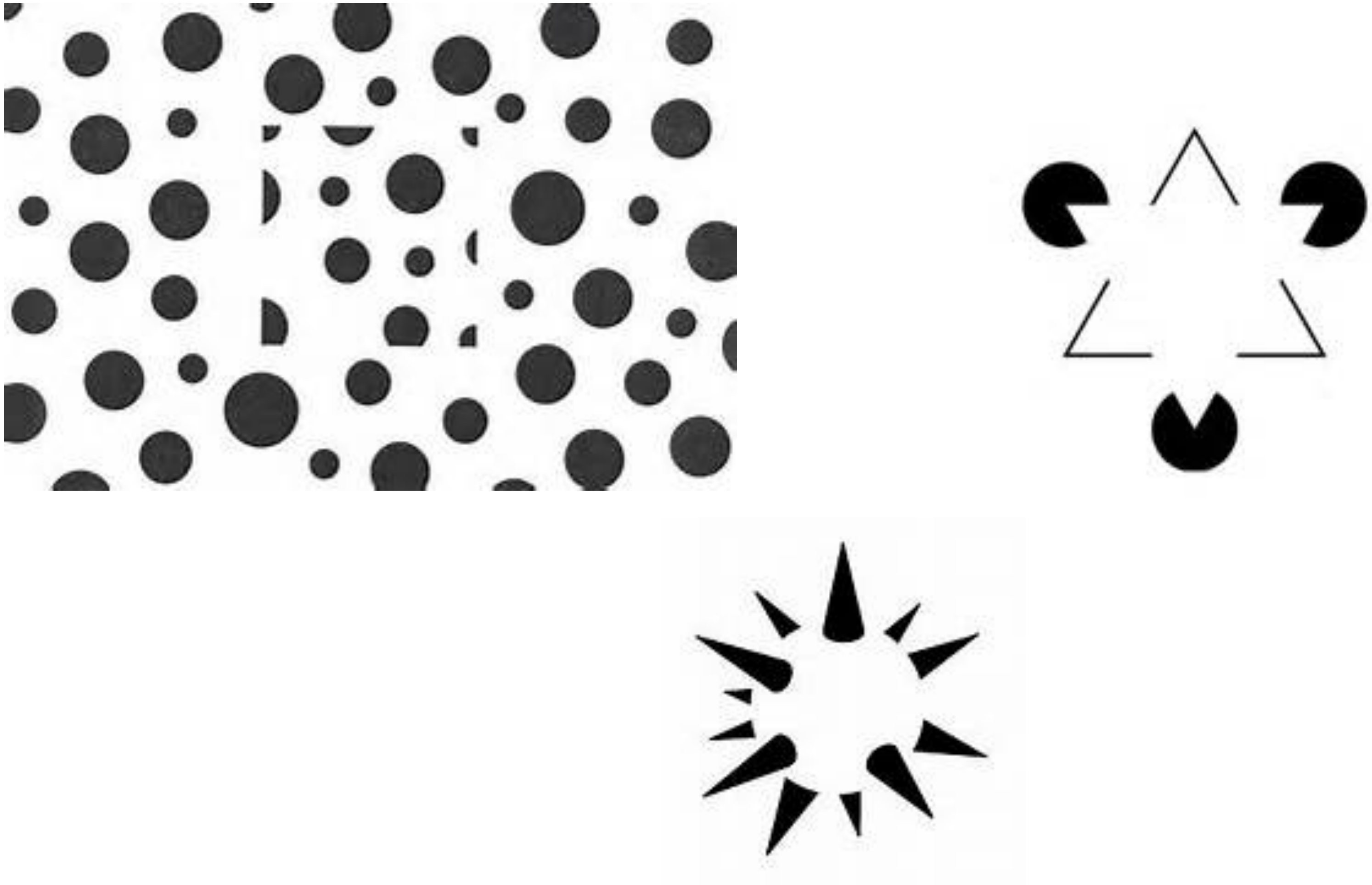
Outline

1. **Induction** and the **problem(s)** of induction
2. The **first AI approach** to induction
3. The **statistical learning** approach
    - The **Perceptron**: a principle and an algorithm
    - **Justifying induction**. The advent of statistical learning
    - The dominant **paradigm**
    - A **closed case**?
4. What about **the revolution(s)** in ML?
    - Does **deep learning** mean big troubles?
    - **New learning tasks** => in need of new learning paradigms?
5. Conclusion

Induction(s) : Illustrations

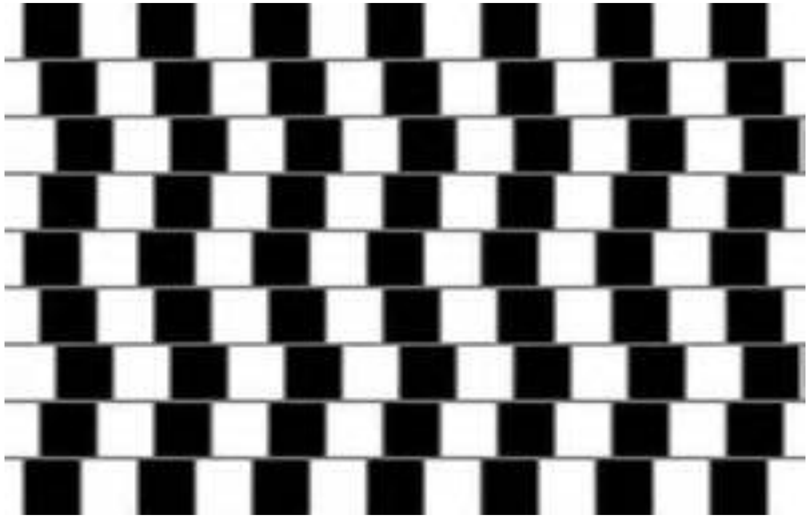
and **questions**

Interpreting – completion of percepts

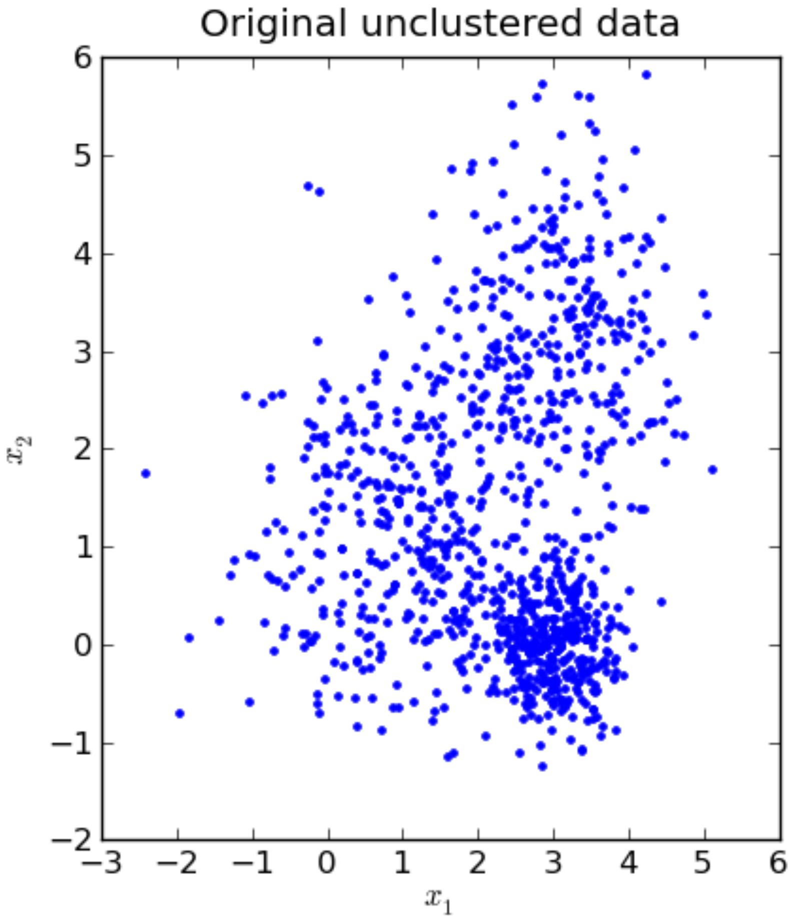


Interpreting – completion of percepts

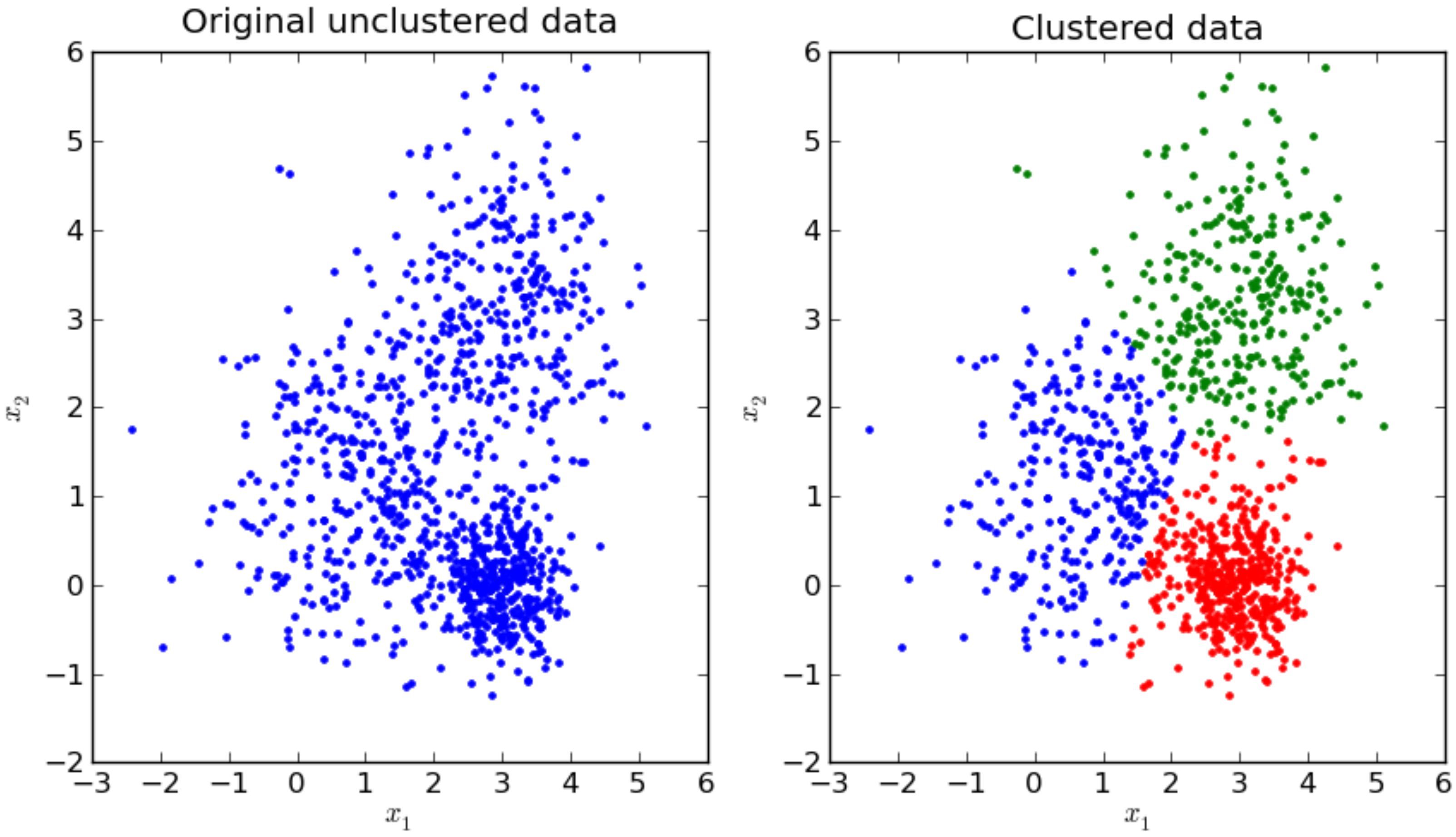
Induction and its illusions



Clustering

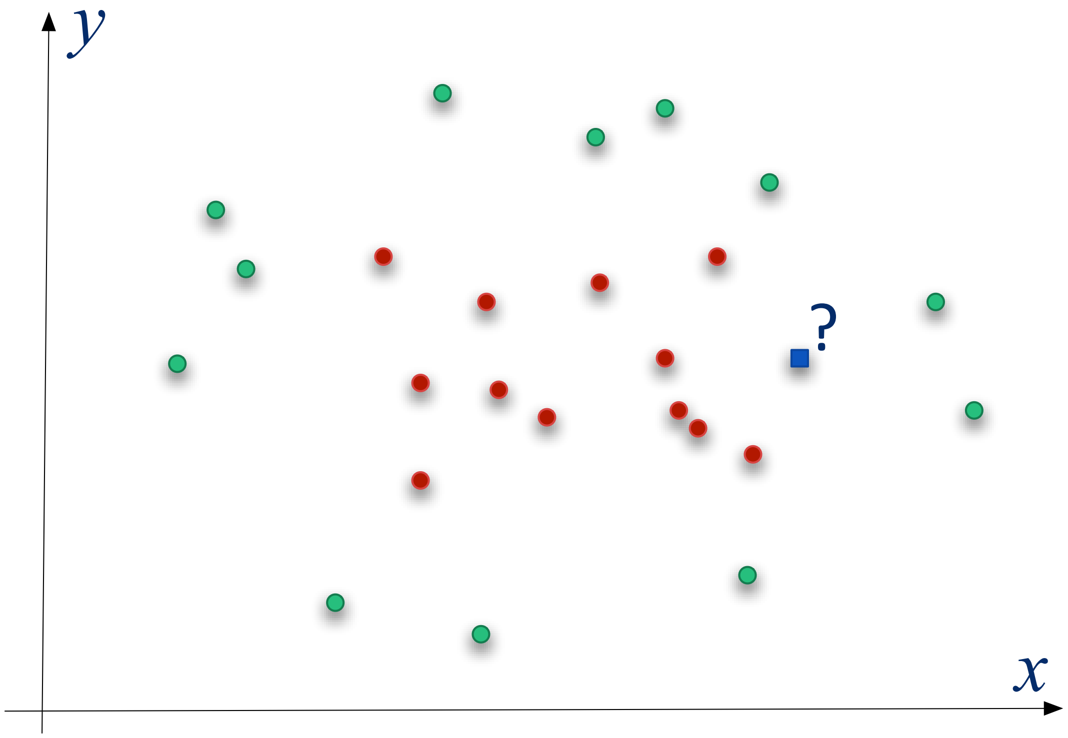


Clustering


Original unclustered data
Clustered data
x_1
x_2
6
5
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
4
5
6

Supervised induction

- We want to be able to predict the class of unseen examples



→ A decision function

Induction: a double question

Some green emeralds => **all emeralds are green**

In each case:

observations => **laws / general rules** or ways to **adapt to new situations**

1. How to **find such rules**? The problem of **invention**

2. Can we **guarantee** something about those “generalizations”?

   The problem of **justification**

Outline

1. **Induction** and the **problem**(s) of induction
2. The **first AI approach** to induction
3. The **statistical learning** approach
   - The **Perceptron**: a principle and an algorithm
   - **Justifying induction**. The advent of statistical learning
   - The dominant **paradigm**
   - A closed case?
4. What about **the revolution**(s) in ML?
   - Does **deep learning** mean big troubles?
   - **New learning tasks** => in need of new learning paradigms?
5. Conclusion

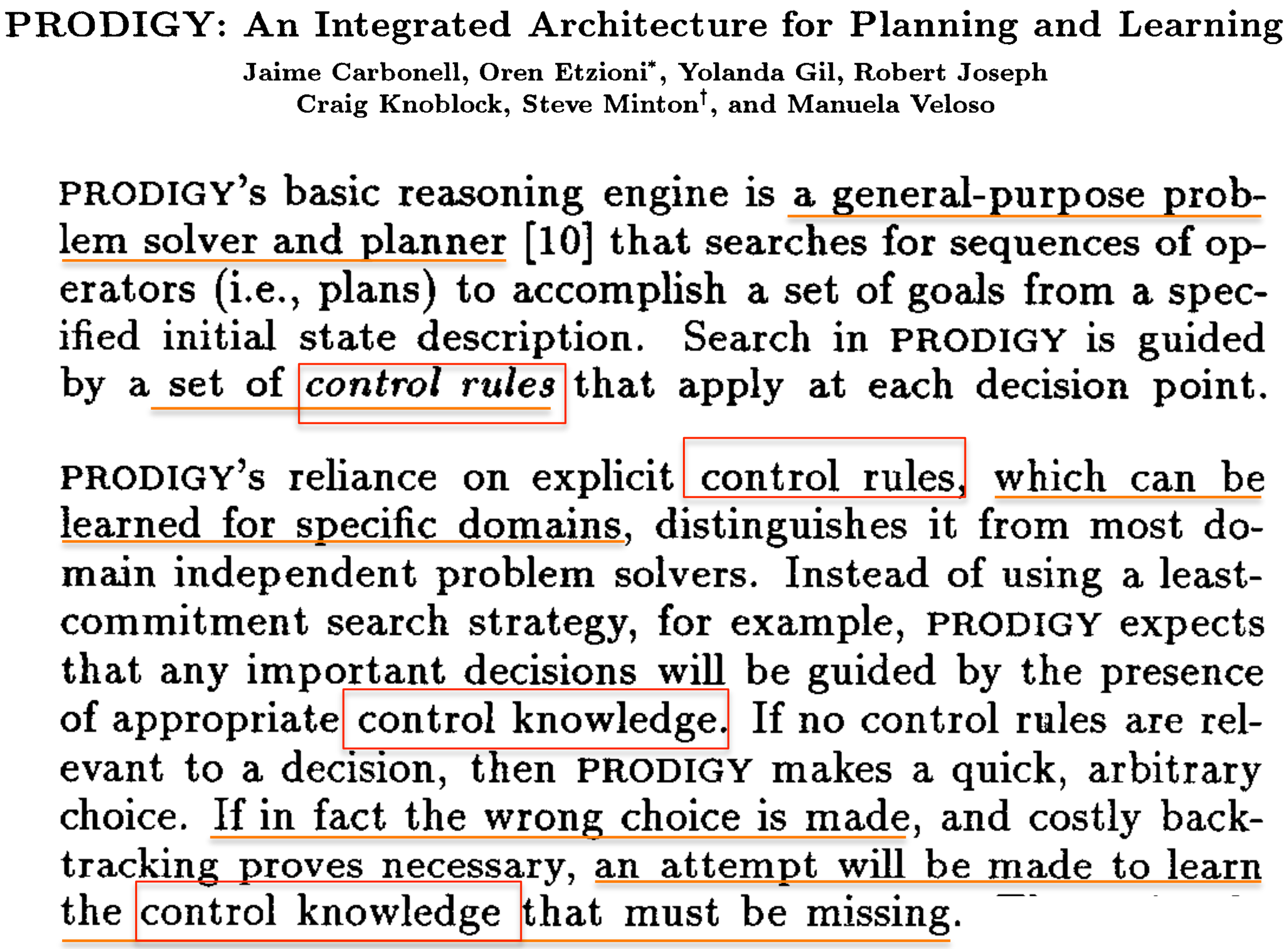
Learning …

… as

a means to improve the efficiency of a problem solver

E.g. The PRODIGY system

ACM SIGART Bulletin, 1991, vol. 2, no 4, p. 51-55



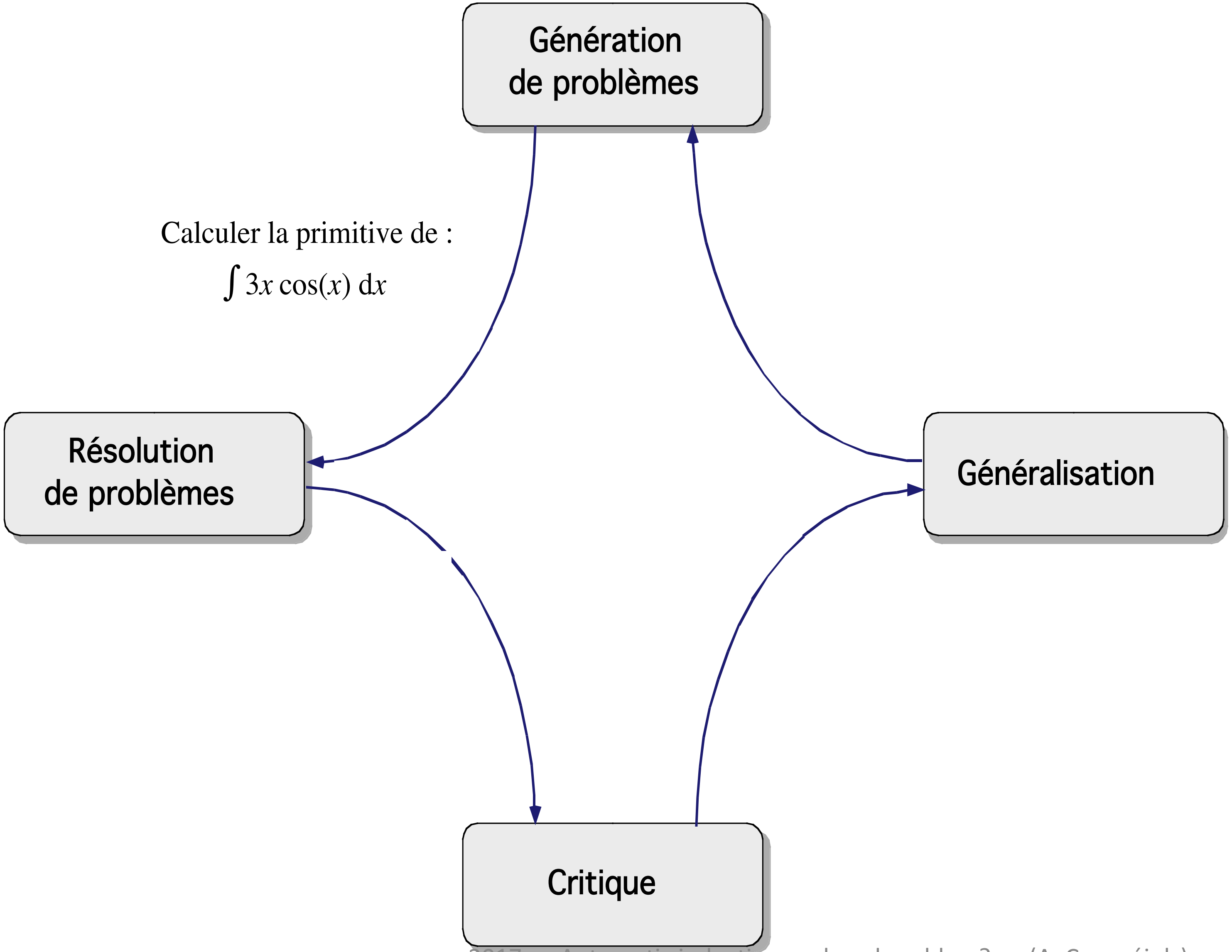
PRODIGY: An Integrated Architecture for Planning and Learning

Jaime Carbonell, Oren Etzioni[*], Yolanda Gil, Robert Joseph
Craig Knoblock, Steve Minton[†], and Manuela Veloso

PRODIGY's basic reasoning engine is a general-purpose problem solver and planner [10] that searches for sequences of operators (i.e., plans) to accomplish a set of goals from a specified initial state description. Search in PRODIGY is guided by a set of *control rules* that apply at each decision point.

PRODIGY's reliance on explicit control rules, which can be learned for specific domains, distinguishes it from most domain independent problem solvers. Instead of using a least-commitment search strategy, for example, PRODIGY expects that any important decisions will be guided by the presence of appropriate control knowledge. If no control rules are relevant to a decision, then PRODIGY makes a quick, arbitrary choice. If in fact the wrong choice is made, and costly backtracking proves necessary, an attempt will be made to learn the control knowledge that must be missing.

Illustration: LEX (Tom Mitchell)



Génération de problèmes

Résolution de problèmes

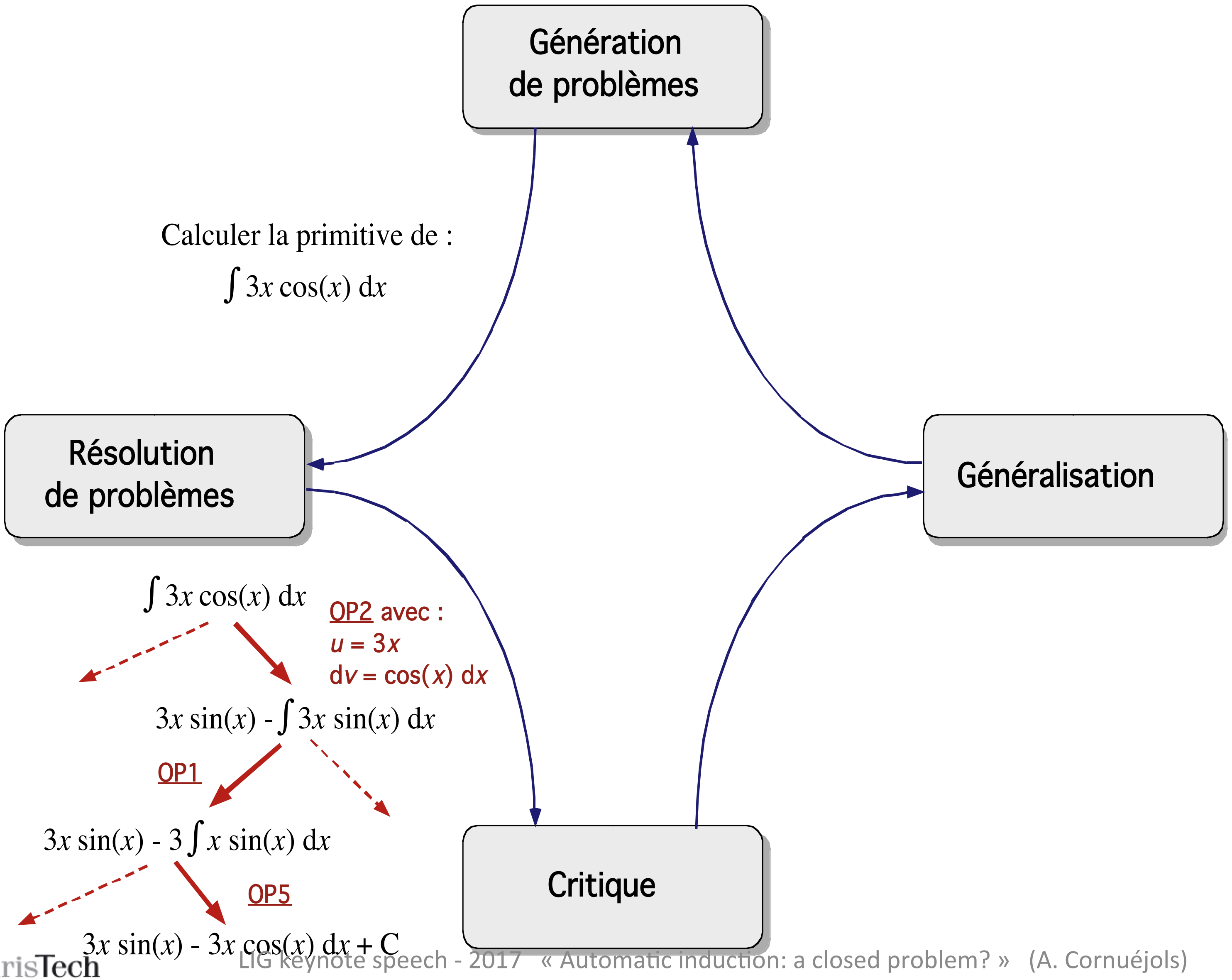
Critique

Généralisation

Calculer la primitive de :

$$\int 3x \cos(x)\, \mathrm{d}x$$

Illustration: LEX (Tom Mitchell)



Génération de problèmes

Calculer la primitive de :

$\int 3x \cos(x)\, dx$

Résolution de problèmes

Généralisation

Critique

$\int 3x \cos(x)\, dx$

OP2 avec :
$u = 3x$
$dv = \cos(x)\, dx$

$3x \sin(x) - \int 3x \sin(x)\, dx$

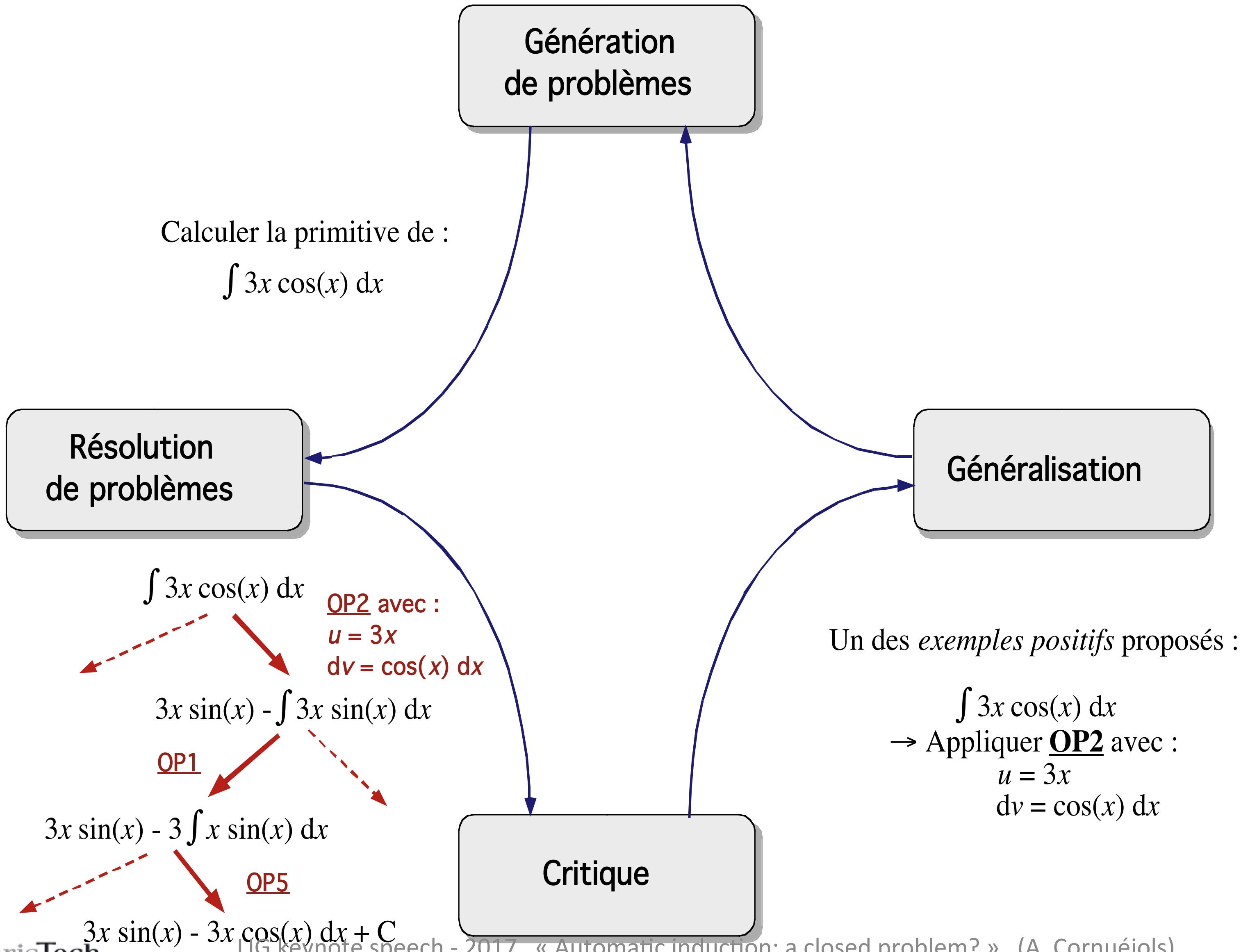
OP1

$3x \sin(x) - 3 \int x \sin(x)\, dx$

OP5

$3x \sin(x) - 3x \cos(x)\, dx + C$

Illustration: LEX (Tom Mitchell)



Génération
de problèmes

Calculer la primitive de :
$\int 3x \cos(x)\, dx$

Résolution
de problèmes

Généralisation

Critique

$\int 3x \cos(x)\, dx$

OP2 avec :
$u = 3x$
$dv = \cos(x)\, dx$

$3x \sin(x) - \int 3x \sin(x)\, dx$

OP1

$3x \sin(x) - 3 \int x \sin(x)\, dx$

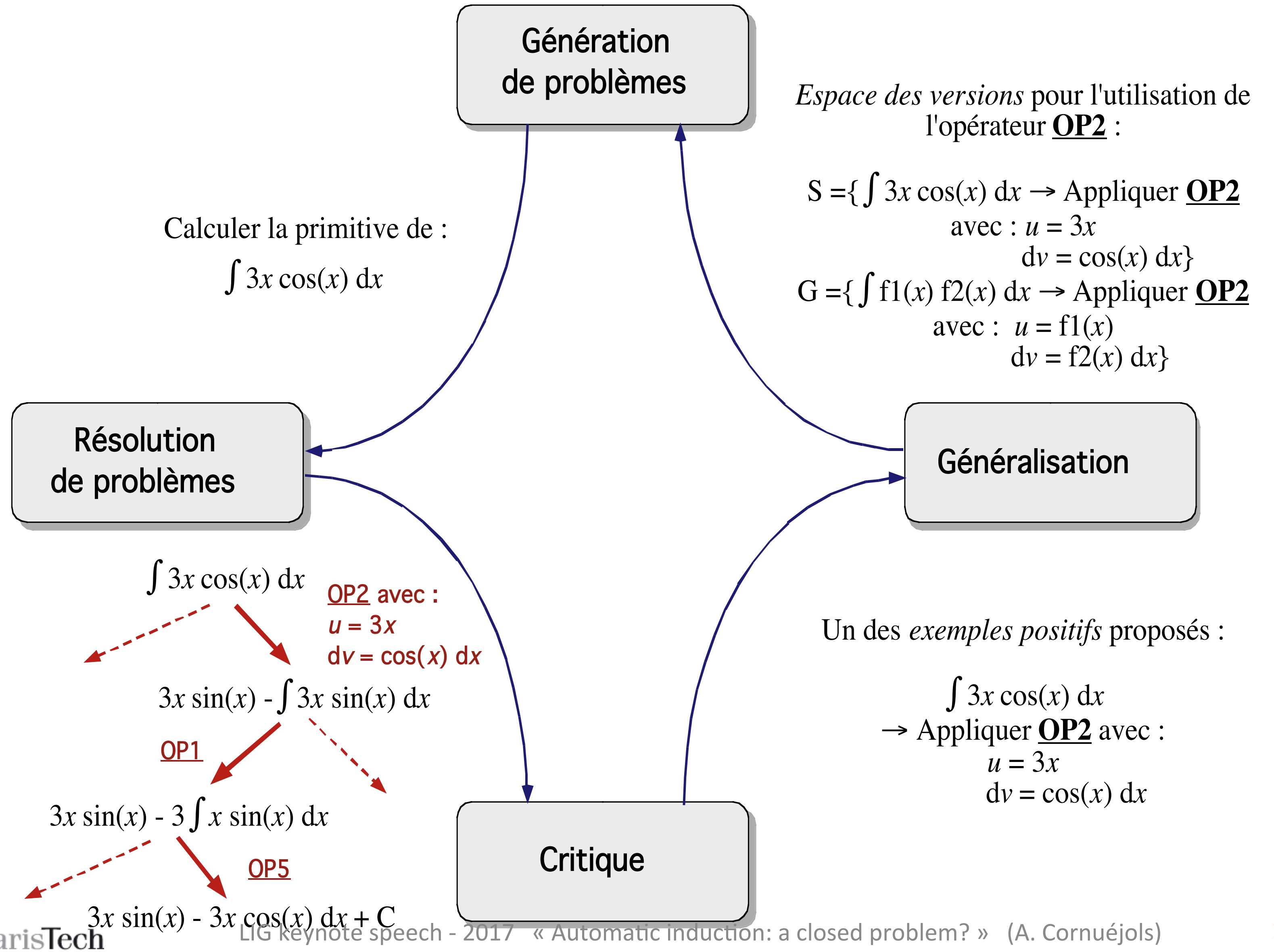
OP5

$3x \sin(x) - 3x \cos(x)\, dx + C$

Un des *exemples positifs* proposés :

$\int 3x \cos(x)\, dx$
→ Appliquer **OP2** avec :
$u = 3x$
$dv = \cos(x)\, dx$

Illustration: LEX (Tom Mitchell)



Génération de problèmes

Résolution de problèmes

Généralisation

Critique

Calculer la primitive de :

$\int 3x \cos(x)\, dx$

Espace des versions pour l'utilisation de l'opérateur **OP2** :

S ={$\int 3x \cos(x)\, dx$ → Appliquer **OP2**
avec : $u = 3x$
$dv = \cos(x)\, dx$}

G ={$\int f1(x)\, f2(x)\, dx$ → Appliquer **OP2**
avec : $u = f1(x)$
$dv = f2(x)\, dx$}

$\int 3x \cos(x)\, dx$

OP2 avec :
$u = 3x$
$dv = \cos(x)\, dx$

$3x \sin(x) - \int 3x \sin(x)\, dx$

OP1

$3x \sin(x) - 3 \int x \sin(x)\, dx$

OP5

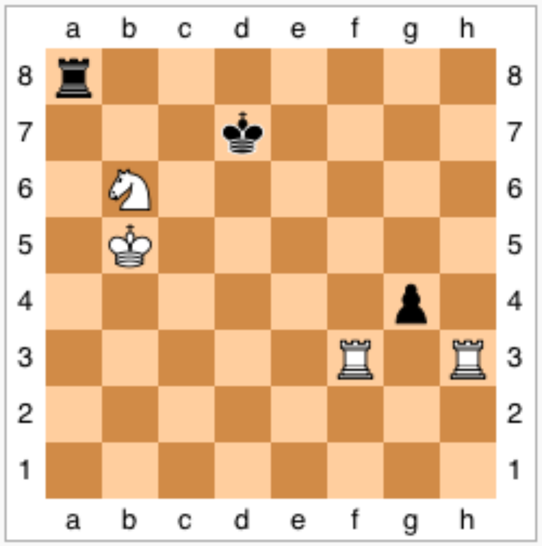
$3x \sin(x) - 3x \cos(x)\, dx + C$

Un des *exemples positifs* proposés :

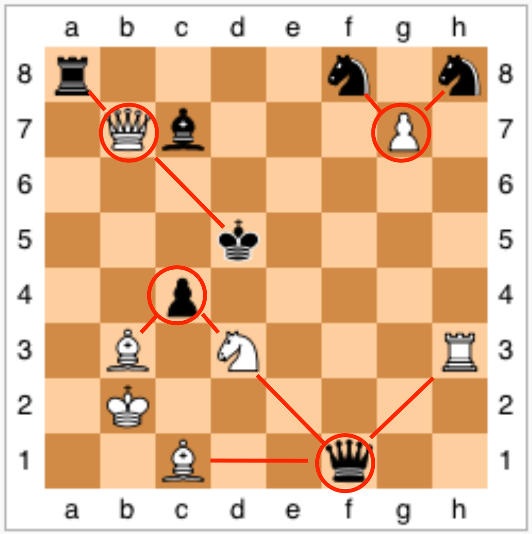
$\int 3x \cos(x)\, dx$
→ Appliquer **OP2** avec :
$u = 3x$
$dv = \cos(x)\, dx$

Learning from **one example**

Explanation-Based Learning



1. From a **single example**
2. **Try to prove** the “fork”
3. **Generalize**



Explanation-Based Learning

Ex : **learn the concept** `stackable(Object1, Object2)`

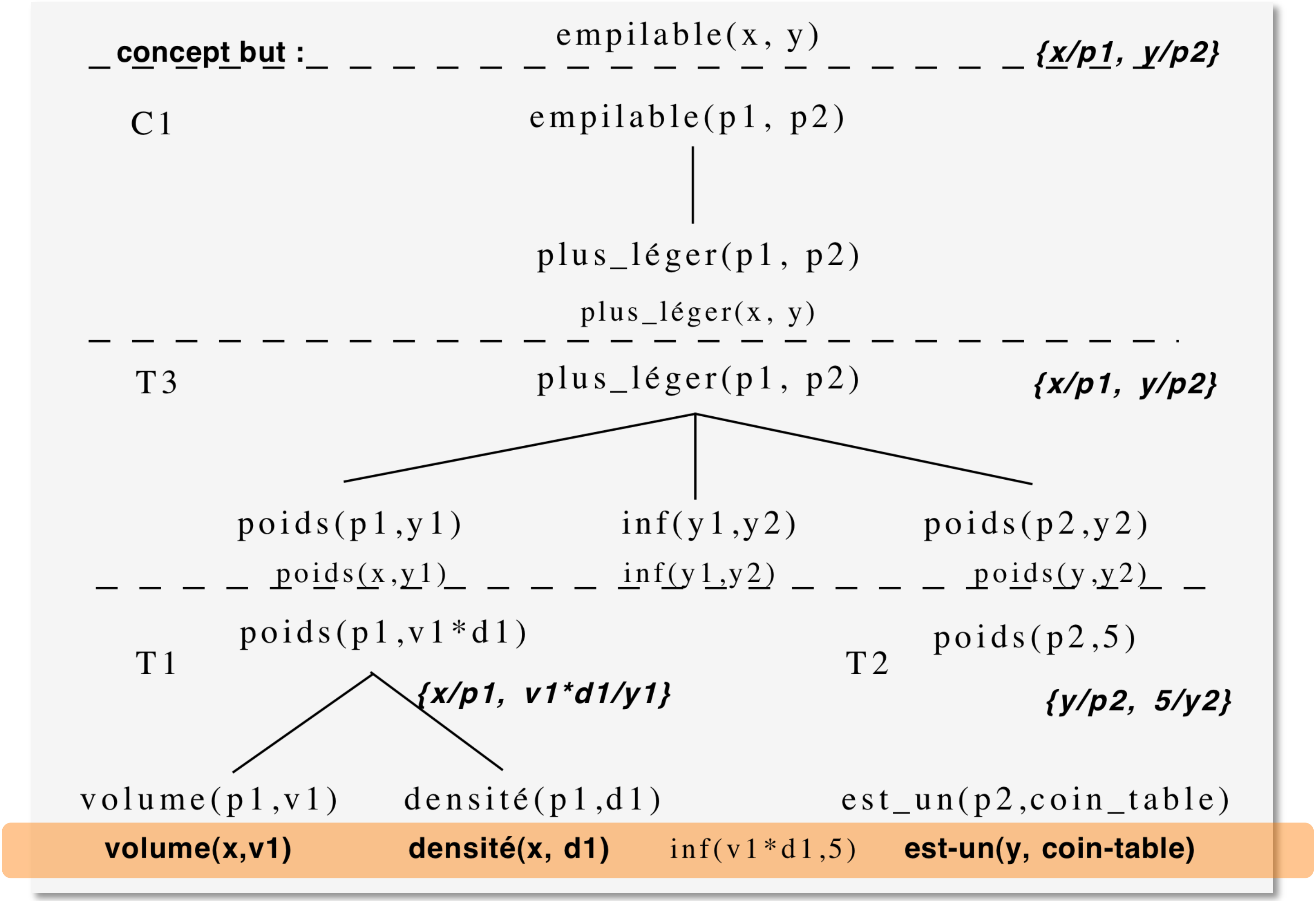
- **Domain theory** :

```
(T1) : weight(X, W) :- volume(X, V), density(X, D), W is V*D.

(T2) : weight(X, 50) :- is_a(X, table).

(T3) : lighter_than(X, Y) :- weight(X, W1), weight(X, W2), W1 < W2.
```

- **Operationality constraint**:
  - Concept should be expressible using *volume, density, color, …*
- **Positive example** (solution) :

```
on(obj1, obj2).                        volume(object1, 1).

is_a(object1, box).                    volume(object2, 0.1).

is_a(object2, table).                  owner(object1, frederic).

color(object1, red).                   density(object1, 0.3).

color(object2, blue).                  Made_of(object1, cardboard).

made_of(object2, wood).                owner(object2, marc).
```

Explanation-Based Learning



concept but : empilable(x, y) {x/p1, y/p2}

C1 empilable(p1, p2)

plus_léger(p1, p2)

plus_léger(x, y)

T3 plus_léger(p1, p2) {x/p1, y/p2}

poids(p1,y1) inf(y1,y2) poids(p2,y2)

poids(x,y1) inf(y1,y2) poids(y,y2)

T1 poids(p1,v1*d1) {x/p1, v1*d1/y1}

T2 poids(p2,5) {y/p2, 5/y2}

volume(p1,v1) densité(p1,d1) est_un(p2,coin_table)

volume(x,v1) **densité(x, d1)** inf(v1*d1,5) **est-un(y, coin-table)**

Generalized search tree resulting from regression of the target concept in the proof tree by computing at each step the most general literals allowing this step.

Explanation-Based Learning

- Induction **from a single example**
  - … plus a strong domain theory

- Based on
  - **Logic-based** knowledge representation
  - **Reasoning Operators** (deduction, goal regression in a proof tree, …)

Now used in SAT "solvers"

Explanation-Based Learning

- What was the **aim** of learning?
- What was a **good theory/ method** of learning ?

Explanation-Based Learning

- What was the **aim** of learning?
- What was a **good method** of learning ?

1. Method **improving** the **problem solving performances**
   - [Steve Minton (1990) « *Quantitative results concerning the **utility** of Explanation-Based Learning* »]
2. Method that **simulates** the performances (and limits) of a **natural cognitive agent** (human or animal)
   - [Laird, Rosenbloom, Newell (1986) « *Chunking in SOAR: The anatomy of a general learning mechanism* »]
   - [Anderson (1993) « *Rules of the mind* » ;
     Taatgen (2003) « *Learning rules and productions* »]

Learning and reasoning

Papers like

- Stephen José Hanson (1990). Conceptual clustering and categorization: **bridging the gap between induction and causal models**.

  Machine Learning journal, 1990, pp.235-268.

But

No measure of generalization performance **independent of the problem-solver**

Difficulties to **scale up** and to face **noisy data**

… when data started to pour down

Outline

1. **Induction** and the **problem**(s) of induction

2. The **first AI approach** to induction

3. The **statistical learning** approach
   - The **Perceptron**: a principle and an algorithm
   - **Justifying induction**. The advent of statistical learning
   - The dominant **paradigm**
   - A **closed case**?

4. What about **the revolution**(s) in ML?
   - Does **deep learning** mean big troubles?
   - **New learning tasks** => in need of new learning paradigms?

5. Conclusion

Supervised learning

Given a **training set**

$$\mathcal{S}_m = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, \left(\begin{tikzcd}\mathbf{x}_i, \arrow[r, bend left, "f"] \arrow[r, bend right, "h"'] & y_i\end{tikzcd}\right), \ldots, (\mathbf{x}_m, y_m)\}$$

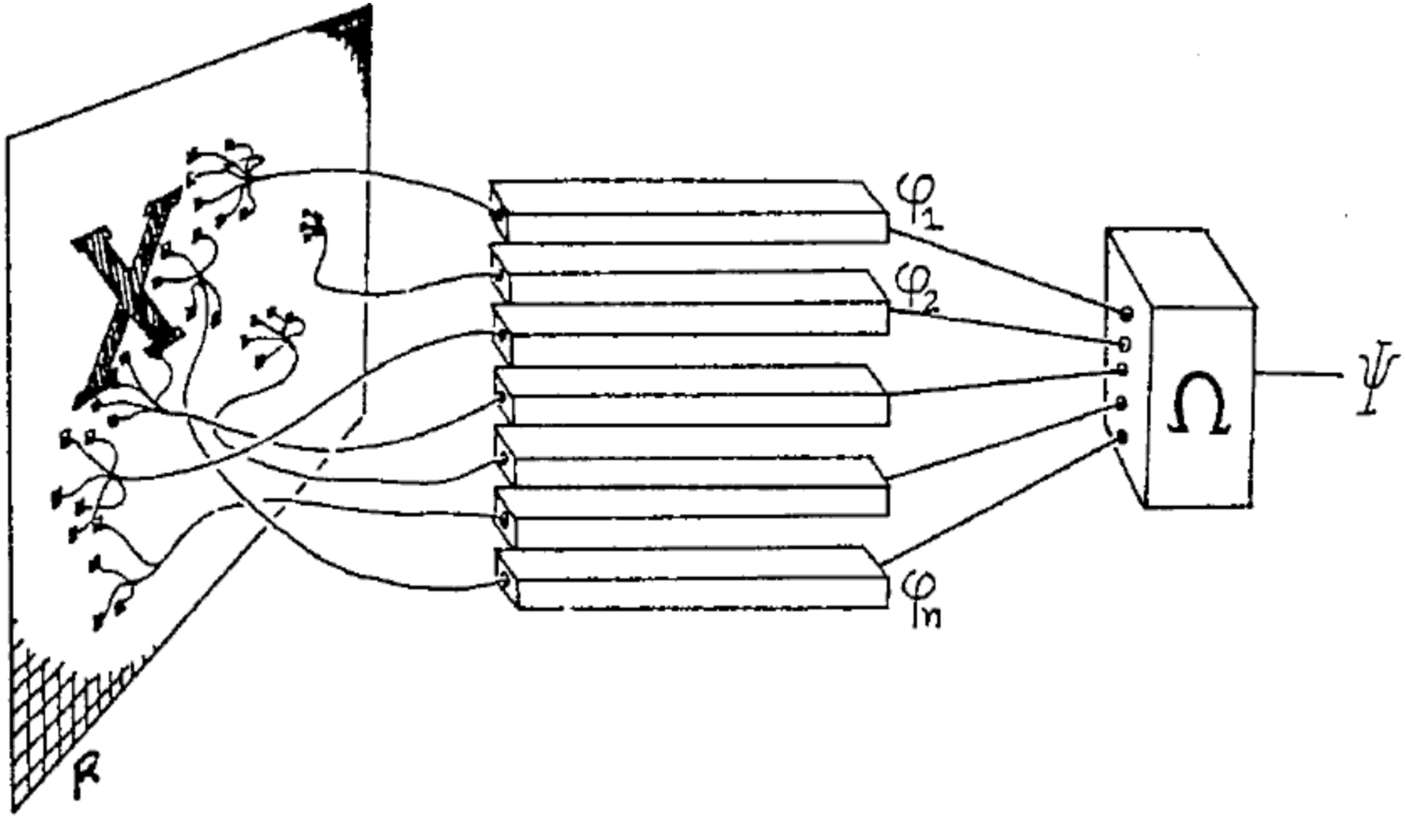
- **Find** an hypothesis $h \in \mathcal{H}$ such that $h(\mathbf{x}_i) \approx y_i$

- Hoping that it generalizes well :

$$\forall \mathbf{x} \in \mathcal{X} : \quad h(\mathbf{x}) \approx y$$

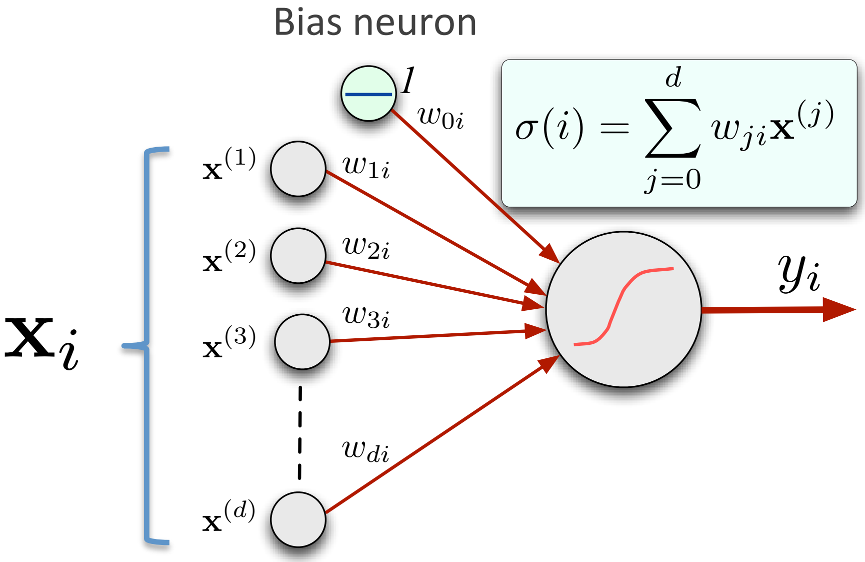
The perceptron

- Rosenblatt (1958-1962)



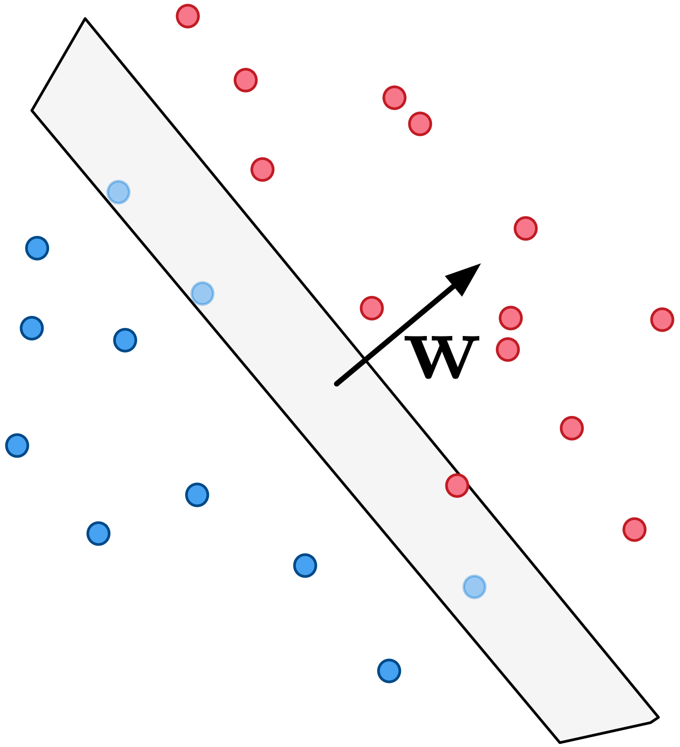
The perceptron

- Rosenblatt (1958-1962)



$$\sigma(i) = \sum_{j=0}^{d} w_{ji} \mathbf{x}^{(j)}$$

The perceptron: a linear discriminant



The perceptron learning rule

- **Adjustments of the weight** w_i

Principle (*Perceptron's rule*): learn only in case of prediction error

Algorithm 1: The perceptron learning algorithm

```
Data: A training sample: $\mathcal{S}_m = \{(\mathbf{x}_i, y_i)\}_{1 \leq i \leq m}$
Result: A weight vector $\mathbf{w}$
while not convergence do
    if the randomly drawn $\mathbf{x}_i$ is st. $sign(\mathbf{w} \cdot \mathbf{x}_i) = y_i$ then
        do nothing
    else
        $\boxed{\mathbf{w}(t+1) = \mathbf{w}(t) + \eta \, \mathbf{x}_i \, y_i}$
    Randomly select next training example $\mathbf{x}_i$
```

The perceptron: illustration

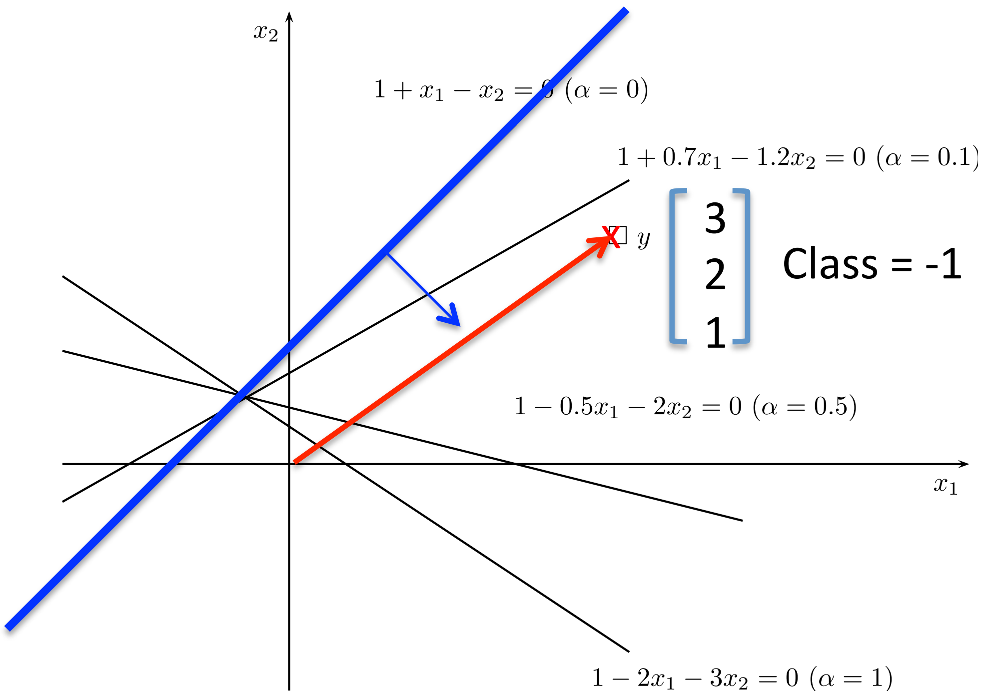
$$\mathbf{w}_t = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta \, \mathbf{x}_i \, y_i$$

$$\text{if } \eta = 0.1 : \quad \mathbf{w}_{t+1} = \begin{bmatrix} 0.7 \\ -1.2 \\ 0.9 \end{bmatrix}$$

$$\text{if } \eta = 0.5 : \quad \mathbf{w}_{t+1} = \begin{bmatrix} -0.5 \\ -2 \\ 0.5 \end{bmatrix}$$

$$\text{if } \eta = 1 : \quad \mathbf{w}_{t+1} = \begin{bmatrix} -2 \\ -3 \\ 0 \end{bmatrix}$$



The perceptron

NO reasoning !!!

Some remarquable properties !!

- **Convergence** in a finite number of steps
  - Independently of the **number** of examples
  - Independently of the **distribution** of the examples
  - Independently of the **dimension** de input space

!!!

If there exists a linear separator of the training examples

Guarantees on generalization ??

- Theorems about the performance

  with respect to the training set

- But what about **future examples**?

The statistical theory

of learning

(illustration)

One example that tells a lot …

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

One example that tells a lot …

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

- They belong either to class '**+**' or to class '**-**'

Description	**Your** prediction	**True** class
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		+
1 small red circle		+

One example that tells a lot …

- Examples described using:

  Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

Description	**Your** prediction	**True** class
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		+
1 small red circle		+

How many possible functions altogether from *X* to *Y* ? $2^{2^4} = 2^{16} = 65,536$

How many functions do remain after 6 training examples? $2^{10} = 1024$

One example that tells a lot …

- Examples described using:

Number (1 or 2); ***size*** (small or large); ***shape*** (circle or square); ***color*** (red or green)

15

Description	**Your** prediction	**True** class
1 large red square		-
1 large green square		+
2 small red squares		+
2 large red circles		-
1 large green circle		+
1 small red circle		+
1 small green square		-
1 small red square		+
2 large green squares		+
2 small green squares		+
2 small red circles		+
1 small green circle		-
2 large green circles		-
2 small green circles		+
1 large red circle		-
2 large red squares	?	

How many remaining functions?

← ?

One example that tells a lot …

- Examples described using:

~~***Number***~~ (1 or 2); ***size*** (small or large); ~~***shape***~~ (circle or square); ***color*** (red or green)

Description	**Your** prediction	**True** class
~~1~~ large red ~~square~~		-
~~1~~ large green ~~square~~		+
~~2~~ small red ~~squares~~		+
~~2~~ large red ~~circles~~		-
~~1~~ large green ~~circle~~		+
~~1~~ small red ~~circle~~		+

How many possible functions with 2 descriptors from *X* to *Y* ? $2^{2^2} = 2^4 = 16$

How many functions do remain after 3 ≠ training examples? $2^1 = 2$

Induction: an impossible game?

- **A bias is need**

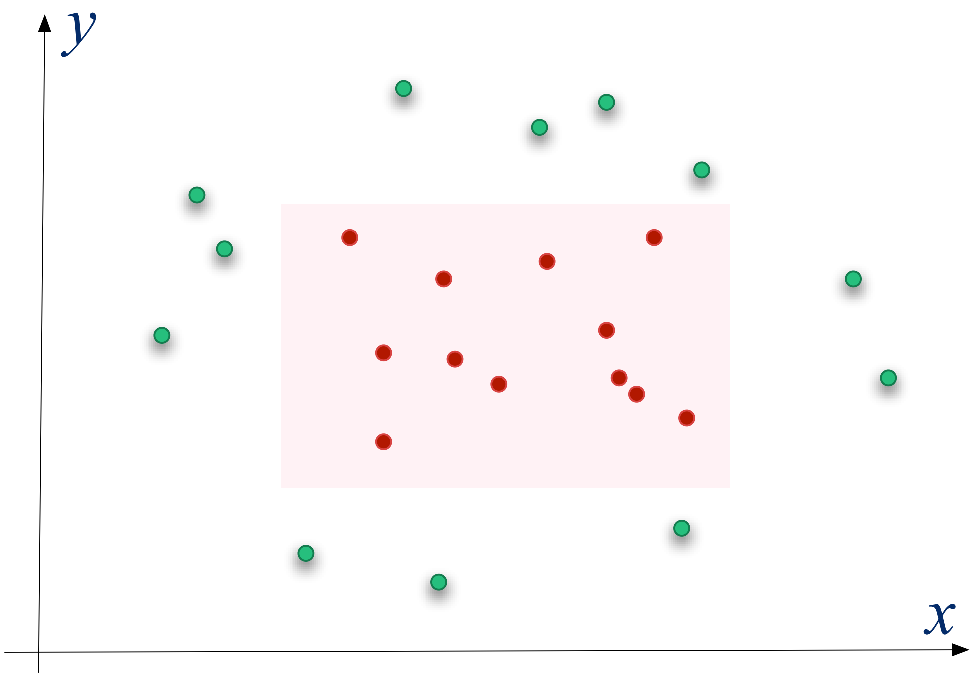
- **Types** of bias
  - **Representation** bias (declarative)
  - **Research** bias (procedural)

Learning the class of 2D points

Training sample

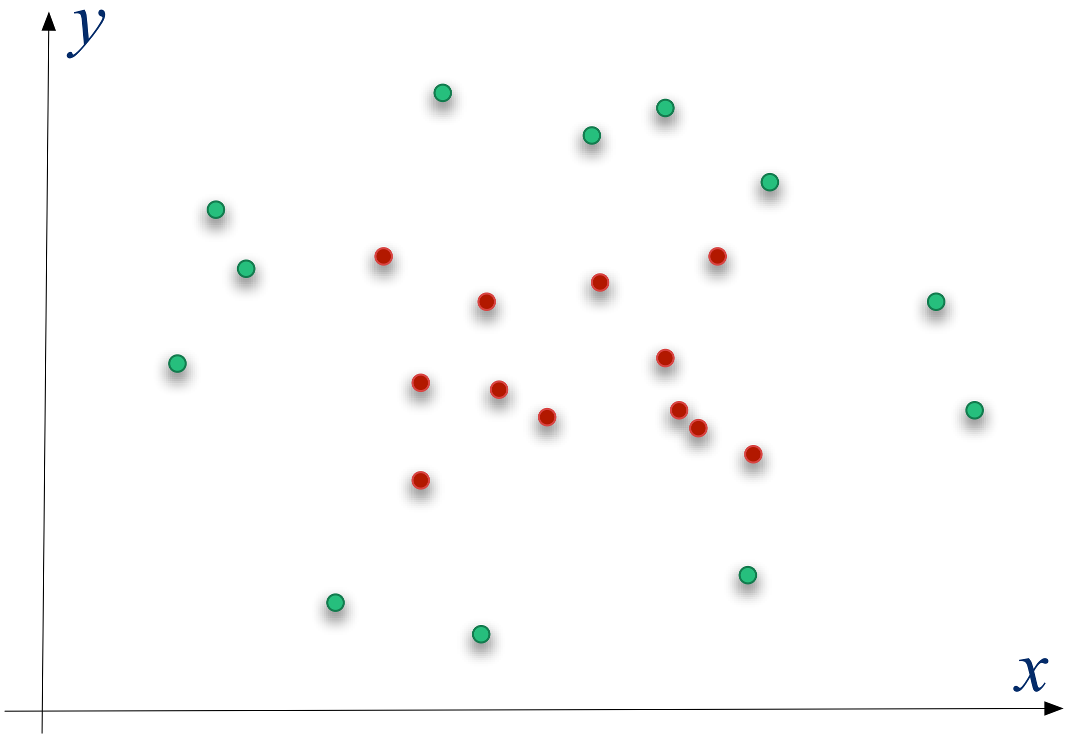
- Positive examples $\mathbf{P}_{\mathcal{X}}^{+}$
- Negative examples $\mathbf{P}_{\mathcal{X}}^{-}$

Hidden concept = rectangle



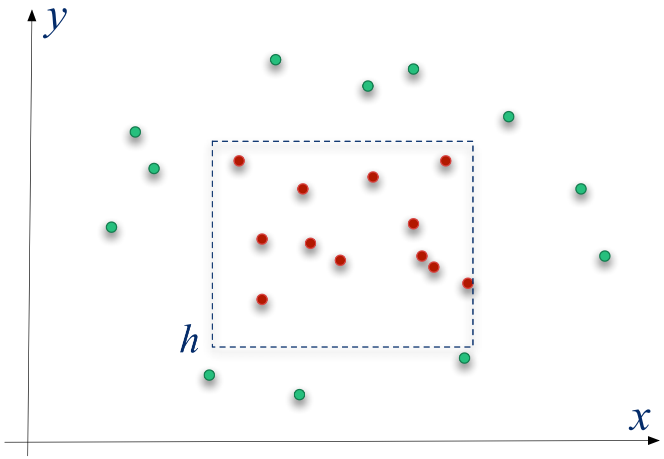
Learning the class of 2D points

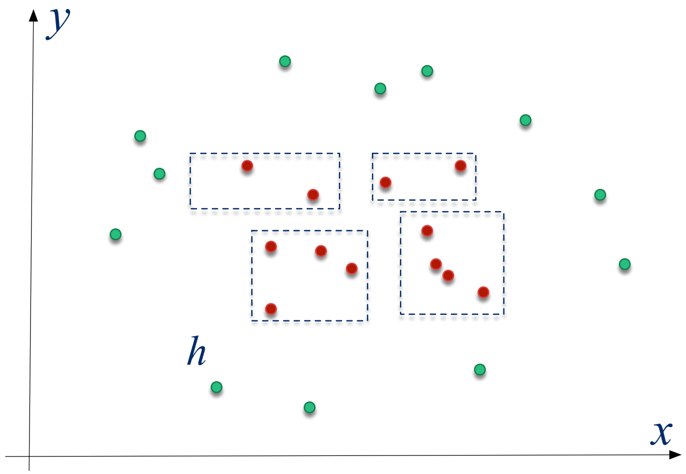
- How can we do that?

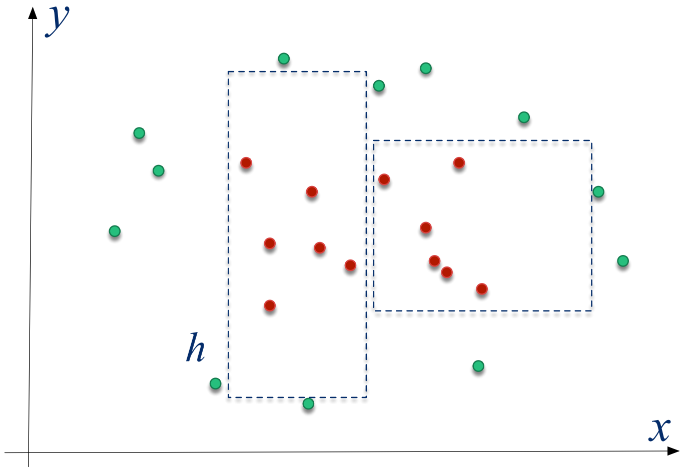


Learning the class of 2D points

- Choice of the hypothesis space $\mathcal{H}$







The statistical theory of learning

in two key steps

A statistical theory of induction

What performance do we aim at?

- Cost of a prediction error
  - The ***loss function***

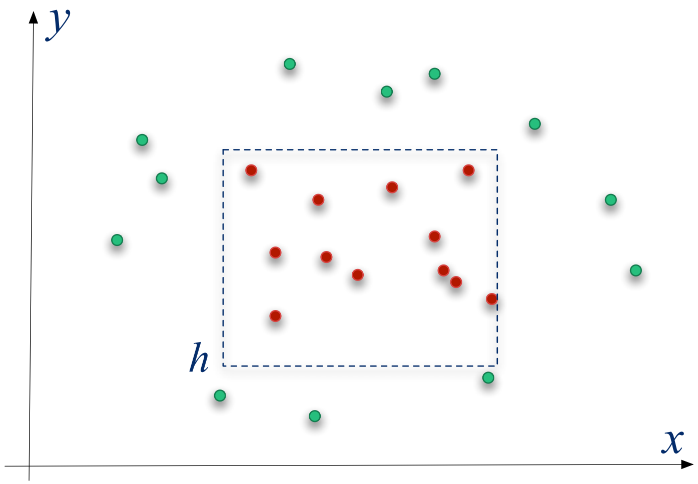
$$\ell\big(h(\mathbf{x}), y\big)$$

- What is the expected cost if I choose *h*?
  - Expected cost: the "**true risk**"

$$R(h) \;=\; \int_{\mathcal{X}\times\mathcal{Y}} \ell\big(h(\mathbf{x}), y\big) \; \mathbf{p}_{\mathcal{X}\mathcal{Y}}(\mathbf{x}, y) \; d\mathbf{x} \, dy$$

A statistical theory of induction

- The empirical performance of *h*
  - E.g. No prediction error on the training sample *S*



The "empirical risk"

$$\hat{R}(h) \;=\; \frac{1}{m} \sum_{i=1}^{m} \ell\big(h(\mathbf{x}_i), y_i\big)$$

Central question: the inductive principle

- Is the Empirical Risk Minimization (ERM) principle

  … sound?

If I choose $\hat{h}$ such that $$\hat{h} \;=\; \operatorname*{ArgMin}_{h \in \mathcal{H}} \; \hat{R}(h)$$

- Will $\hat{h}$ be good as well with respect to the true risk?

$$\boxed{\hat{R}(\hat{h}) \;\overset{?}{\longleftrightarrow}\; R(\hat{h})} \qquad (1)$$

- Could I have done much better?

$$h^* \;=\; \operatorname*{ArgMin}_{h \in \mathcal{H}} \; R(h) \qquad \boxed{R(h^*) \overset{?}{\longleftrightarrow} R(\hat{h})} \qquad (2)$$

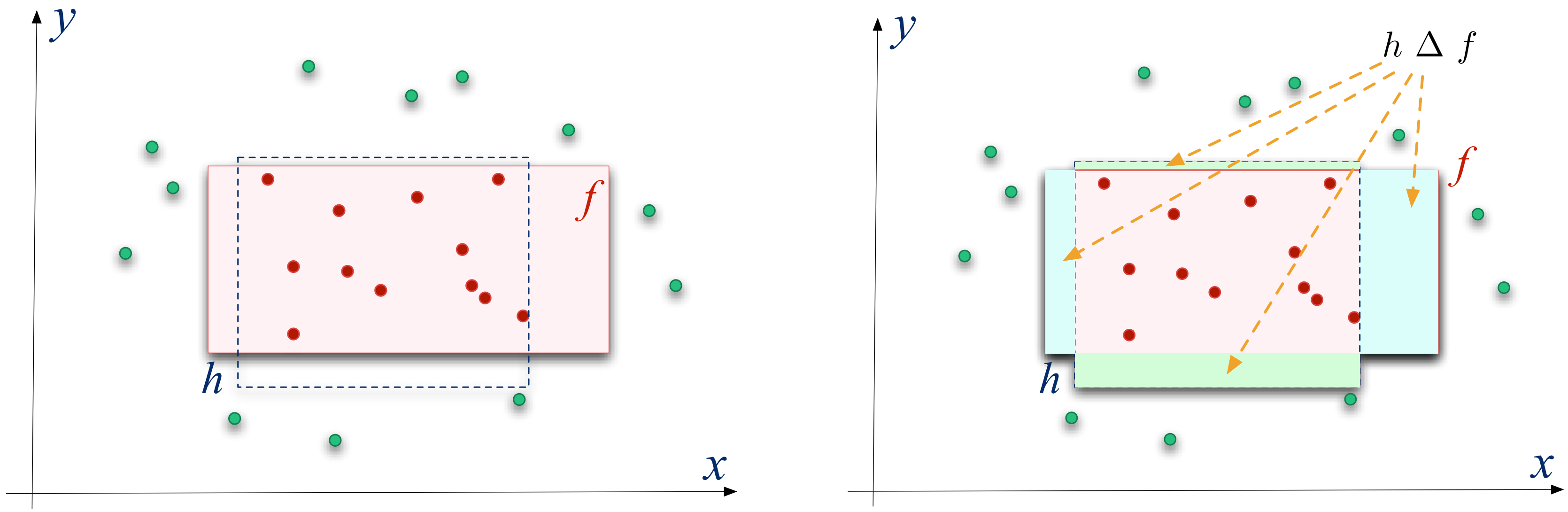
The statistical theory of learning

The 1st step (1)

One **hypothesis**

Statistical study for ONE hypothesis

- An hypothesis of null empirical risk is chosen
  (no prediction error on the training sample *S*)
- What is the expected error of *h*?
- What is the **risk of ending with** ***R*(*h*) > ε** ?



Statistical study for ONE hypothesis

- Assume *h* st. $R(h) \;\geq\; \varepsilon$ (*h* is "bad")
- What is the probability that ***h* has been selected nonetheless**?

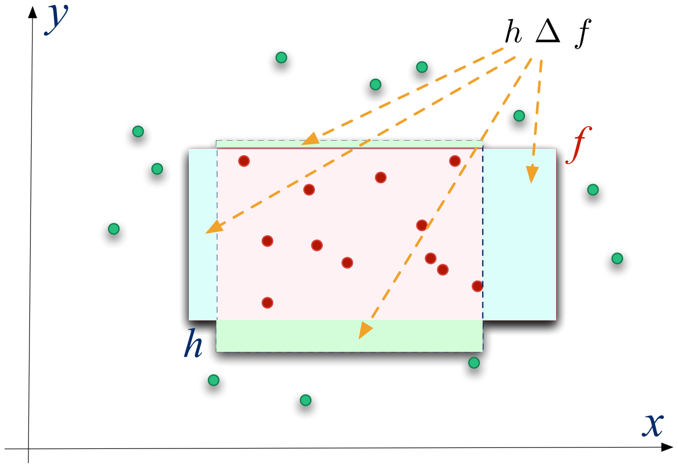
$$R(h) \;=\; \mathbf{p}_{\mathcal{X}}(h \,\Delta\, f)$$

After one example : $p(\hat{R}(h) = 0) \;\leq\; 1 - \varepsilon$

that h "falls" out of $h \,\Delta\, f$

After *m* examples (**i.i.d.**):

$$p^m(\hat{R}(h) = 0) \;\leq\; (1 - \varepsilon)^m$$



error rate

confidence

We want :

$$\forall\, \varepsilon, \delta \in [0,1] : \quad p^m\big(R(h) \geq \varepsilon\big) \;\leq\; \delta$$

Statistical study for ONE hypothesis

- We want: $\forall\, \varepsilon, \delta \in [0,1] : \quad p^m\big(R(h) \geq \varepsilon\big) \;\leq\; \delta$

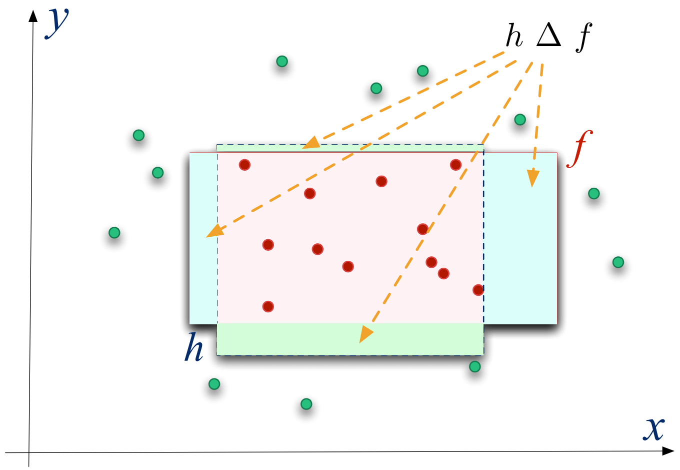
Or:

$$(1 - \varepsilon)^m \;\leq\; \delta$$

$$e^{-\varepsilon\, m} \;\leq\; \delta$$

$$-\varepsilon\, m \;\leq\; \ln(\delta)$$

Hence : $$\boxed{m \;\geq\; \frac{\ln(1/\delta)}{\varepsilon}}$$



The statistical theory of learning

The 2nd step (2)

The very **best hypothesis** in $\mathcal{H}$

Statistical study for $|\mathcal{H}|$ hypotheses

- What is the probability that I select an hypothesis h_{err} with true risk $> \varepsilon$ **and that I do not realize it after** *m* examples ?
  - Probability of "survival" of **h_{err} after 1 example** : $(1 - \varepsilon)$
  - Probability of "survival" of **h_{err} after *m* examples** : $(1 - \varepsilon)^m$

(1)

- Probability of survival of **at least one hypothesis in $\mathcal{H}$** : $|\mathcal{H}| \, (1 - \varepsilon)^m$
  - From a bound on the probability of the union $\mathbf{P}(A \cup B) \; \leq \; \mathbf{P}(A) + \mathbf{P}(B)$

We want the probability that there remains **at least one hypothesis of risk > ε** in the version space be bounded by δ :

$$|\mathcal{H}| \, (1 - \varepsilon)^m \;\; < \;\; |\mathcal{H}| e^{(-\varepsilon \, m)} \;\; < \;\; \delta$$

$$\log |\mathcal{H}| - \varepsilon \, m \;\; < \;\; \log \delta$$

$$m \;\; > \;\; \frac{1}{\varepsilon} \; \log \frac{|\mathcal{H}|}{\delta}$$

Statistical study for $|\mathcal{H}|$ hypotheses

It leads to:

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R(h) \ \leq \ \widehat{R}(h) \ + \ \overbrace{\frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{m}}^{\varepsilon} \right] \ > \ 1 - \delta$$

The Empirical Risk Minimization principle

is sound **only if** there exists a limit (a bias) on the expressivity of $\mathcal{H}$

Bounds on the difference between the true risk and the empirical risk

- $\mathcal{H}$ finite, realizable case

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R(h) \ \leq \ \widehat{R}(h) \ + \ \frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{m} \right] \ > \ 1 - \delta$$

- $\mathcal{H}$ finite, **non** realizable case

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R(h) \ \leq \ R(h) \ + \ \sqrt{\frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{2\, m}} \right] \ > \ 1 - \delta$$

Statistical study for $\mathcal{H}$ **not finite**

- Non realizable case and $\mathcal{H}$ not finite

  How to do that?

  – General principle:

    1. **Reduce** the **infinite case to a the case of an finite set of hypotheses**

    2. Estimate how much it is possible, **for any given training set $\mathcal{S}$,** to find an **hypothesis in $\mathcal{H}$ that fits the data**

Statistical study for $\mathcal{H}$ not finite

- The **_Rademacher complexity_**

  1. **Shuffle randomly the labels** of the training examples

     Each y_i is replaced by **a random label** σ_i = -1 or +1

  2. Then it is possible to estimate how it is possible

     to **find an hypothesis in $\mathcal{H}$ that fits the data** (whichever the data):

$$R_{\mathcal{S}}(\mathcal{H}) \;=\; \mathbb{E}_{\sigma}\left[\underset{h \in \mathcal{H}}{\mathrm{Max}} \; \frac{1}{m} \sum_{i=1}^{m} \sigma_i \, h(\mathbf{x}_i)\right]$$

We can get the bound:

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R(h) \;\leq\; \hat{R}(h) \;+\; R_{\mathcal{S}}(\mathcal{H}) \;+\; 3\sqrt{\frac{\log \frac{2}{\delta}}{2\,m}} \right] \;>\; 1 - \delta$$

Statistical theory of learning as a theory of justification

Use of **the ERM principle** (fitting the data) **is justified** as long as the expressiveness (or capacity) of $\mathcal{H}$ is controlled (and limited)

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R(h) \; \leq \; \hat{R}(h) \; + \; \boxed{R_{\mathcal{S}}(\mathcal{H})} + 3\sqrt{\frac{\log \frac{2}{\delta}}{2\,m}} \right] \; > \; 1 - \delta$$

From **a theory of justification**

to THE recipe for

producing **algorithms of invention**

A powerful paradigm

HOW TO … devise learning algorithms

1. Define an appropriate **regularized** (inductive) **criterion**
    1. Translate the cost of errors of prediction in the domain into a loss function
    2. Define a **regularization term** that expresses *assumptions about the underlying regularities of the world*
    3. If possible, make the resulting **optimization** problem a **convex** one

$$h_{opt} \;=\; \underset{h \in \mathcal{H}}{\text{ArgMin}} \left[\underbrace{\frac{1}{m} \sum_{i=1}^{m} l(h(\mathbf{x}_i), y_i)}_{\text{empirical risk}} \;+\; \lambda \;\; \underbrace{reg(\mathcal{H})}_{\text{bias on the world}} \right]$$

2. Use or develop an **efficient optimization solver**

Learning **sparse linear** approximator

- The **hypothesis** is of the form $h(\mathbf{x}) \;=\; \mathbf{w} \,\cdot\, \mathbf{x}$
- A priori assumption: few non zero coefficients

Ridge regression $$\mathbf{w}^*_{\text{ridge}} \;=\; \underset{\mathbf{w}}{\text{Argmin}}\left\{\sum_{i=1}^{m}(y_i - \mathbf{w}\,\mathbf{x}_i)^2 \;+\; \lambda\,||\mathbf{w}||_2^2\right\}$$

Lasso regression $$\mathbf{w}^*_{\text{lasso}} \;=\; \underset{\mathbf{w}}{\text{Argmin}}\left\{\sum_{i=1}^{m}(y_i - \mathbf{w}\,\mathbf{x}_i)^2 \;+\; \lambda\,||\mathbf{w}||_1\right\}$$

Multi-task learning

- *T* binary classification tasks on $\mathcal{X}$ x $\mathcal{Y}$

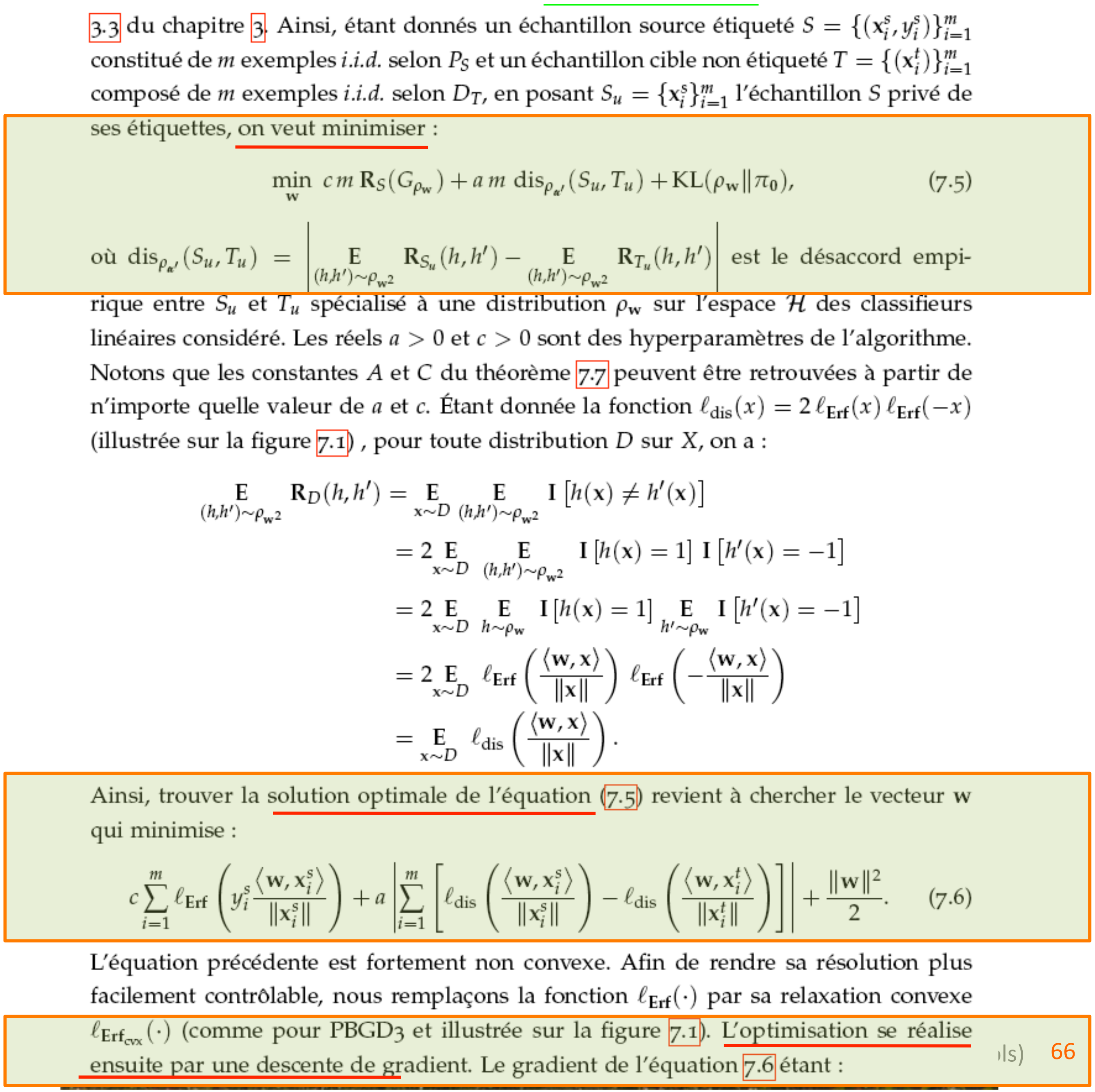
$$\mathcal{S} = \{\{(\mathbf{x}_{11}, y_{11}), (\mathbf{x}_{21}, y_{21}), \ldots, (\mathbf{x}_{m1}, y_{m1})\}, \ldots, \{(\mathbf{x}_{1T}, y_{1T}), (\mathbf{x}_{2T}, y_{2T}), \ldots, (\mathbf{x}_{mT}, y_{mT})\}\}$$

$$h_j(\mathbf{x}) = \mathbf{w}_j \cdot \mathbf{x}$$

Assumption (1) : **linear** hypotheses

Assumption (2) : tasks are related by $\mathbf{w}_j = \mathbf{w}_0 + \mathbf{v}_j$

$$h_1^\star, \ldots, h_T^\star = \underset{\mathbf{w}_0, \mathbf{v}_j, \xi_{ij}}{\text{Argmin}} \left\{ \sum_{j=1}^{T} \sum_{i=1}^{m} \xi_{ij} + \frac{\lambda_1}{T} \sum_{j=1}^{T} ||\mathbf{v}_j||^2 + \lambda_2 \, ||\mathbf{w}_0||^2 \right\}$$



3.3 du chapitre 3. Ainsi, étant donnés un échantillon source étiqueté $S = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^m$ constitué de m exemples *i.i.d.* selon P_S et un échantillon cible non étiqueté $T = \{(\mathbf{x}_i^t)\}_{i=1}^m$ composé de m exemples *i.i.d.* selon D_T, en posant $S_u = \{\mathbf{x}_i^s\}_{i=1}^m$ l'échantillon S privé de ses étiquettes, on veut minimiser :

Regularized empirical risk

$$\min_{\mathbf{w}} \; c\, m\, \mathrm{R}_S(G_{\rho_\mathbf{w}}) + a\, m\, \mathrm{dis}_{\rho_{\mathbf{w}'}}(S_u, T_u) + \mathrm{KL}(\rho_\mathbf{w} \| \pi_0), \tag{7.5}$$

où $\mathrm{dis}_{\rho_{\mathbf{w}'}}(S_u, T_u) = \left| \underset{(h,h')\sim\rho_\mathbf{w}^2}{\mathbf{E}} \mathrm{R}_{S_u}(h,h') - \underset{(h,h')\sim\rho_\mathbf{w}^2}{\mathbf{E}} \mathrm{R}_{T_u}(h,h') \right|$ est le désaccord empirique entre S_u et T_u spécialisé à une distribution $\rho_\mathbf{w}$ sur l'espace $\mathcal{H}$ des classifieurs linéaires considéré. Les réels $a > 0$ et $c > 0$ sont des hyperparamètres de l'algorithme. Notons que les constantes A et C du théorème 7.7 peuvent être retrouvées à partir de n'importe quelle valeur de a et c. Étant donnée la fonction $\ell_{\mathrm{dis}}(x) = 2\,\ell_{\mathrm{Erf}}(x)\,\ell_{\mathrm{Erf}}(-x)$ (illustrée sur la figure 7.1) , pour toute distribution D sur X, on a :

$$\begin{aligned}
\underset{(h,h')\sim\rho_\mathbf{w}^2}{\mathbf{E}} \mathrm{R}_D(h,h') &= \underset{\mathbf{x}\sim D}{\mathbf{E}} \; \underset{(h,h')\sim\rho_\mathbf{w}^2}{\mathbf{E}} \mathbf{I}\left[h(\mathbf{x}) \neq h'(\mathbf{x})\right] \\
&= 2 \underset{\mathbf{x}\sim D}{\mathbf{E}} \; \underset{(h,h')\sim\rho_\mathbf{w}^2}{\mathbf{E}} \mathbf{I}\left[h(\mathbf{x}) = 1\right] \mathbf{I}\left[h'(\mathbf{x}) = -1\right] \\
&= 2 \underset{\mathbf{x}\sim D}{\mathbf{E}} \; \underset{h\sim\rho_\mathbf{w}}{\mathbf{E}} \mathbf{I}\left[h(\mathbf{x}) = 1\right] \underset{h'\sim\rho_\mathbf{w}}{\mathbf{E}} \mathbf{I}\left[h'(\mathbf{x}) = -1\right] \\
&= 2 \underset{\mathbf{x}\sim D}{\mathbf{E}} \; \ell_{\mathrm{Erf}}\left(\frac{\langle \mathbf{w}, \mathbf{x}\rangle}{\|\mathbf{x}\|}\right) \ell_{\mathrm{Erf}}\left(-\frac{\langle \mathbf{w}, \mathbf{x}\rangle}{\|\mathbf{x}\|}\right) \\
&= \underset{\mathbf{x}\sim D}{\mathbf{E}} \; \ell_{\mathrm{dis}}\left(\frac{\langle \mathbf{w}, \mathbf{x}\rangle}{\|\mathbf{x}\|}\right).
\end{aligned}$$

Surrogate expression of the regularized empirical risk

Ainsi, trouver la solution optimale de l'équation (7.5) revient à chercher le vecteur $\mathbf{w}$ qui minimise :

$$c \sum_{i=1}^{m} \ell_{\mathrm{Erf}}\left(y_i^s \frac{\langle \mathbf{w}, \mathbf{x}_i^s\rangle}{\|\mathbf{x}_i^s\|}\right) + a \left| \sum_{i=1}^{m} \left[\ell_{\mathrm{dis}}\left(\frac{\langle \mathbf{w}, \mathbf{x}_i^s\rangle}{\|\mathbf{x}_i^s\|}\right) - \ell_{\mathrm{dis}}\left(\frac{\langle \mathbf{w}, \mathbf{x}_i^t\rangle}{\|\mathbf{x}_i^t\|}\right)\right]\right| + \frac{\|\mathbf{w}\|^2}{2}. \tag{7.6}$$

L'équation précédente est fortement non convexe. Afin de rendre sa résolution plus facilement contrôlable, nous remplaçons la fonction $\ell_{\mathrm{Erf}}(\cdot)$ par sa relaxation convexe

Optimization

$\ell_{\mathrm{Erf_{cvx}}}(\cdot)$ (comme pour PBGD3 et illustrée sur la figure 7.1). L'optimisation se réalise ensuite par une descente de gradient. Le gradient de l'équation 7.6 étant :

A very alluring framework

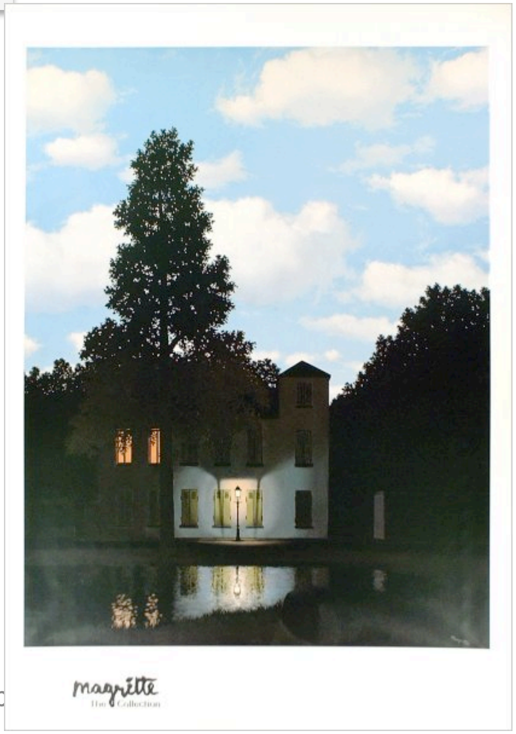
1. Based on a **justification theory**
    - **Bounds** on the generalization error **can be claimed** (very important for having paper accepted)
    - **Valid for the worst case**: against any possible distribution of the data
2. Seemingly **very benign assumptions** on the world
    - Data (and future questions) supposedly **i.i.d.**
    - $f \in H$ or $f \notin H$
3. Provides **a recipe** to produce learning algorithms
    - Very generic applicability: ***minimization of a regularized empirical risk***
    - Learning **=** optimization

A lot of "Lamppost theorems"

Theorems that guarantee that:

- **If** the world obeys **my a priori assumptions**
- **Then** the learning algorithm will end up with a good hypothesis (closed to the "real" one)

- **Otherwise** learning can lead to very bad hypotheses
  (e.g. *If the world is not sparse*)



But, may be *we cannot do any better*!?

The no-free-lunch theorem

The no-free-lunch theorem

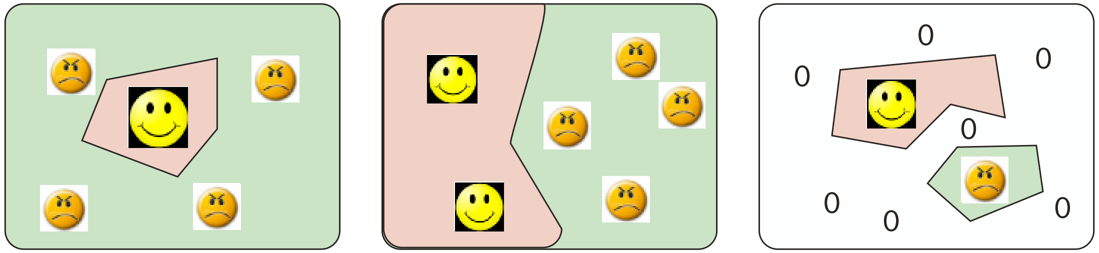
Théorème 2.2 (No-free-lunch theorem (Wolpert, 1992))

For any pair of learning algorithms $\mathcal{A}_1$ *and* $\mathcal{A}_2$*, characterized by their* a posteriori *probability distribution* $\mathbf{p}_1(h|\mathcal{S})$ *and* $\mathbf{p}_2(h|\mathcal{S})$*, and for all distribution* $d_{\mathcal{X}}$ *on the input space* $\mathcal{X}$*, and all numbers of training examples, the following claims are true :*

1. *On average* ***on all target functions*** f *in* $\mathcal{F}$ *:*
   $\mathbb{E}_1[R_{\text{Rel}}|f, m] \; - \; \mathbb{E}_2[R_{\text{Rel}}|f, m] \; = \; 0.$
2. *For* ***any given training sample*** $\mathcal{S}$*, on average* ***on all the target functions*** f *in* $\mathcal{F}$ *:*
   $\mathbb{E}_1[R_{\text{Rel}}|f, \mathcal{S}] \; - \; \mathbb{E}_2[R_{\text{Rel}}|f, \mathcal{S}] \; = \; 0.$
3. *On average* ***on every possible distributions*** $\mathbf{P}(f)$ *:*
   $\mathbb{E}_1[R_{\text{Rel}}|m] \; - \; \mathbb{E}_2[R_{\text{Rel}}|m] \; = \; 0.$
4. *For* ***any given training sample*** $\mathcal{S}$*, on average* ***on all possible distributions*** $\mathbf{p}(f)$ *:*
   $\mathbb{E}_1[R_{\text{Rel}}|\mathcal{S}] \; - \; \mathbb{E}_2[R_{\text{Rel}}|\mathcal{S}] \; = \; 0.$

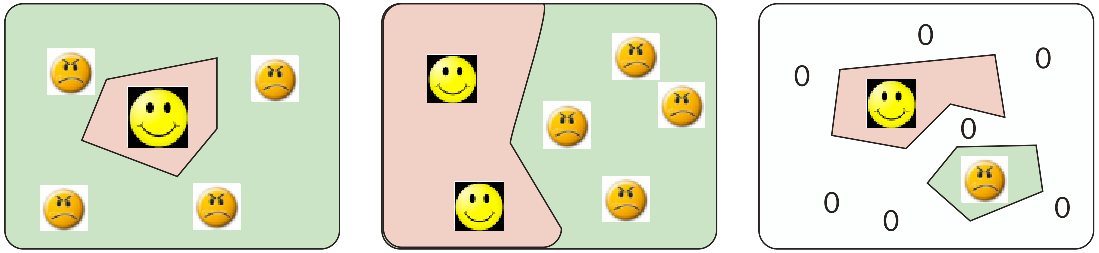
The no-free-lunch theorem

Possible

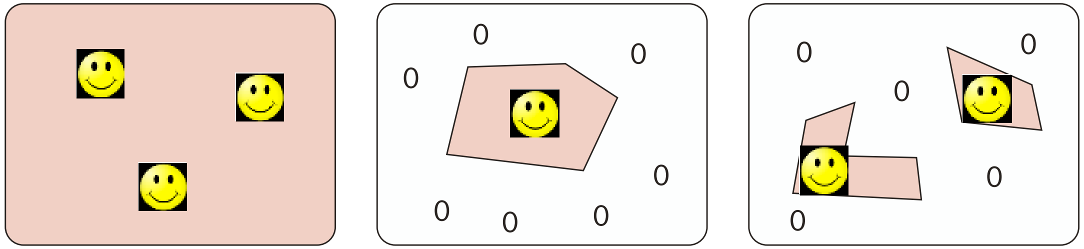


The no-free-lunch theorem

Possible



Impossible



Deduction !

1. **All** inductive **algorithms** have **equal performance** on average

2. There **cannot be any a priori guarantee** on the induction results

Case closed: we cannot do any better

The end!

Case closed: we cannot do any better

This is **the end of history** for the science of induction

- **Only** lamppost theorems **are possible**
- And **we know how to find** (all of) **them**

  except for a few constants that could be sharpened, everything is known

Outline

1. **Induction** and the **problem**(s) of induction
2. The **first AI approach** to induction
3. The statistical learning approach
   - The Perceptron: a principle and an algorithm
   - Justifying induction. The advent of statistical learning
   - The dominant paradigm
   - A closed case?
4. What about **the revolution**(s) in ML?
   - Does deep learning mean big troubles?
   - New learning tasks => in need of new learning paradigms?
5. Conclusion

Does deep learning

mean big trouble (for the theory of induction)?

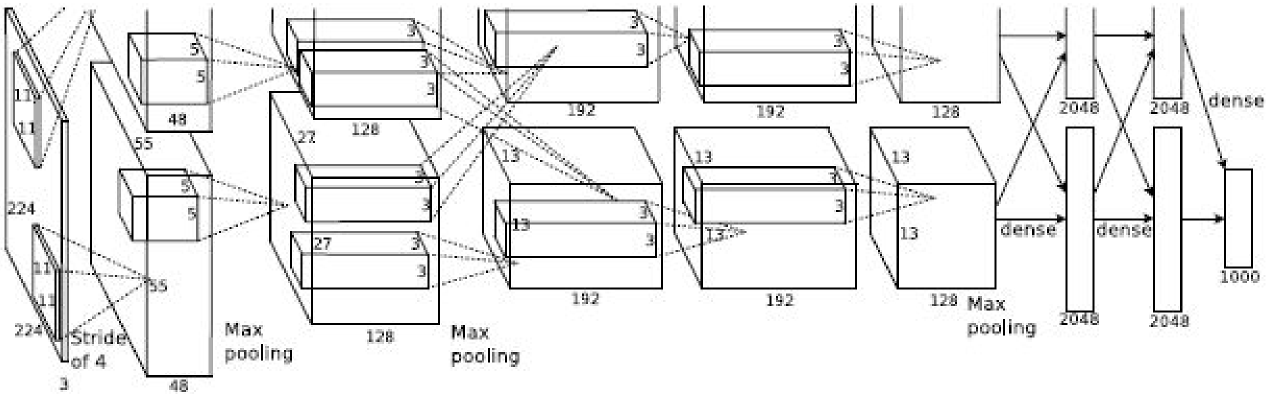
Troubling findings

A paper

- C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals (ICLR, May 2017). “Understanding deep learning requires rethinking generalization”

Extensive experiments on the classification of images

- The AlexNet (**> 1,000,000 parameters**) + 2 other architectures



- The **CIFAR-10 data set**:
  - 60,000 images categorized in 10 classes (50,000 for training and 10,000 for testing)
  - Images: 32x32 pixels in 3 color channels

Troubling findings

Experiments

1. **Original dataset** without modification
   - Results ?
     - **Training** accuracy = 100% ; **Test** accuracy = 89%
     - Speed of convergence ~ 5,000 steps

Troubling findings

Experiments

1. **Original dataset** without modification
   - Results ?
     - **Training** accuracy = 100%   ;   **Test** accuracy = 89%
     - Speed of convergence ~ 5,000 steps

Expected behavior if the **capacity** of the hypothesis space is **limited**

i.e. the system **cannot** fit any (arbitrary) training data

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R(h) \;\leq\; \widehat{R}(h) \;+ \boxed{2\, \widehat{Rad}_m(\mathcal{H})} + \; 3 \sqrt{\frac{\ln(2/\delta)}{m}} \right] \;>\; 1 - \delta$$

Troubling findings

Experiments

1. **Original dataset** without modification
   - Results ?
     - **Training** accuracy = 100% ; **Test** accuracy = 89%
     - Speed of convergence ~ 5,000 steps

2. Random **labels**

!!!

   - **Training** accuracy = 100% !!?? ; **Test** accuracy = 9.8%
   - Speed of convergence = similar behavior (~ 10,000 steps)

Troubling findings

Experiments

1. **Original dataset** without modification
   - Results ?
     - **Training** accuracy = 100% ; **Test** accuracy = 89%
     - Speed of convergence ~ 5,000 steps

2. Random **labels**
   - **Training** accuracy = 100% !!?? ; **Test** accuracy = 9.8%
   - Speed of convergence = similar behavior (~ 10,000 steps)

3. Random **pixels**
   - **Training** accuracy = 100% !!?? ; **Test** accuracy ~ 10%
   - Speed of convergence = similar behavior (~ 10,000 steps)

Now, we are in trouble!!

Troubling findings

- Deep NNs can accommodate ANY training set

Can grow without limit!!

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \quad P^m \left[R(h) \;\leq\; \widehat{R}(h) \;+\; 2\,\widehat{Rad}_m(\mathcal{H}) \;+\; 3\sqrt{\frac{\ln(2/\delta)}{m}} \right] \;>\; 1-\delta$$

But then,

> *why are deep NNs* ***so good on image classification tasks?***

New learning scenarios

=> In need of new learning paradigms?

Transfer learning

Definition [Pan, TL-IJCAI'13 tutorial]

- Ability of a system to **recognize** and **apply** knowledge and skills learned in **previous domains/tasks** to **novel domains/tasks**

Example

- We have **labeled images** (person / no person) from a **web corpus**
- Novel task: **is there a person** in unlabeled images from a **video corpus**?

Web corpus

Video corpus

Transfert learning: questions

- What can be **the basis** of transfer learning?

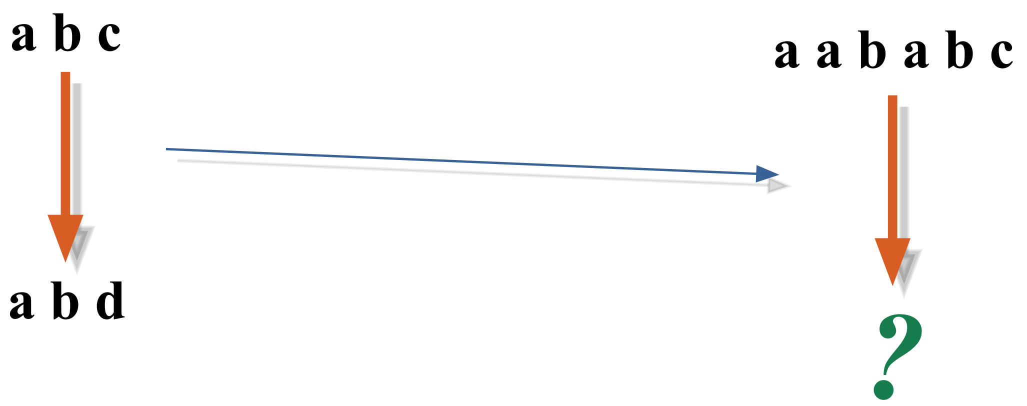
Not i.i.d. anymore

How to translate formally :

*"the target domain **is like** the source domain"*?

- What **determine a good transfer**?
  - A "good source"?
  - A high "similarity" between source and target?
- What **formal guarantees** can we have on the transferred hypothesis?

Transfer and analogy



a b c

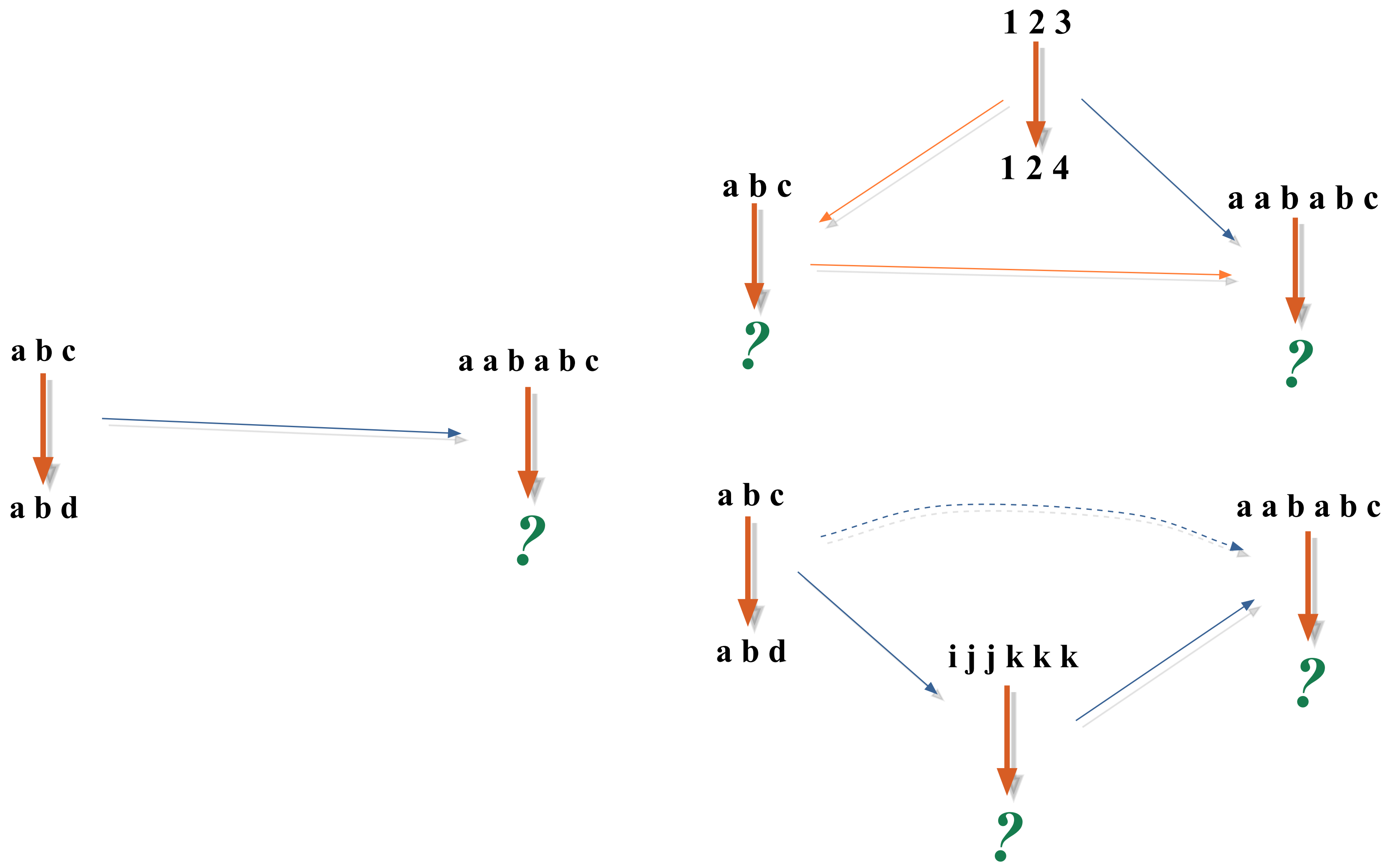
a b d

a a b a b c

?

Why should ‘**a a b a b c d**’ be any better than ‘**a b d**’?

Transfer and sequence effects



1 2 3

1 2 4

a b c

?

a a b a b c

?

a b c

a b d

a a b a b c

?

a b c

a b d

i j j k k

?

a a b a b c

?

- t

Long-life learning

- Learning organized in **a sequence of tasks**
  - Very **far from the i.i.d. scenario**

→ Learning will be affected by the **history of the system**

- We need a theory of the **dynamics of learning**
  1. Which **sequence effects** can we expect?
  2. How to **best organize the curriculum** of a learning system?

Outline

1. **Induction** and the **problem**(s) of induction

2. The **first AI approach** to induction

3. The statistical learning approach
   - **The Perceptron: a principle and an algorithm**
   - **Justifying induction. The advent of statistical learning**
   - **The dominant paradigm**
   - A **closed case?**

4. What about **the revolution(s)** in ML?
   - Does **deep learning** mean **big troubles**?
   - **New** learning **tasks** => in need of **new** learning **paradigms**?

5. Conclusions

Conclusion (1)

The *statistical learning theory* is a theory of **justification**

1. Valid in the **stationary environment** assumption
2. Says that **induction needs bias**:
   a priori assumptions on the world that limit the search space
3. Provides a **general strategy** to develop new algorithms
   1. Translate a **learning task** into a priori on the world,
      therefore into **regularization terms**
   2. Find an **efficient optimization** scheme

But

Even in the i.i.d. scenario

- **Not able to explain** the efficiency of deep learning
- **Not adapted** to new learning scenarios

Conclusions: “new” scenarios

- **Limited** data sources
  - We often learn from (very) few examples

- The past **history** of learning affects learning: Education
  - Sequence effects

- We learn in order to **build “theories”**
  - All the time: small and large theories

For instance, what would you like to ask?

Conclusion (2)

Pendulum movements in the science of induction

1. **Invention** (first AI)
   - General Problem Solvers ; heuristic reasoning
   - First **connectionism** ; **cognitively based** learning systems

2. **Justification**
   - Inventing logics to account for "imperfect" reasoning
   - **Statistical theory of learning**

3. **Invention** again
   - **Deep learning**
   - **New learning scenarios**:

     transfer/analogy ; **long-life** learning ; learning from **very few examples** ; …