

Reflections on INDUCTION-S

A. Cornuéjols

AgroParisTech – INRA MIA 518

Learning by heart

Outline

1. Learning by heart
2. Transduction
3. Analogy
4. The statistical Theory of Learning
5. Limits
6. A theory of semi-supervised learning
7. Case study: a theory for EBL
8. What theory for analogy?



Course « InductionS » (A. Cornuéjols)

2 / 122

When there are few data points

Exemple	x ₁	x ₂	x ₃	x ₄	Etiquette
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

■ Learning a table

When there is a huge number of data points

- Learning a function $f: x \rightarrow y$

But how ?

Which function ?

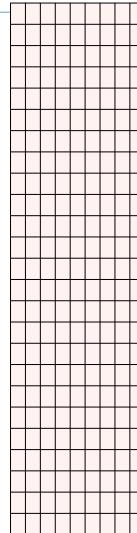


Illustration : un problème d'apprentissage

- Combien de fonctions possibles de 4 entrées booléennes et une sortie booléenne ?

- Combien de fonctions compatibles avec la donnée de **7 exemples** ?

Laquelle choisir ?

Exemple	x_1	x_2	x_3	x_4	Etiquette
3	0	0	1	0	0
4	0	0	1	1	1
5	0	1	0	0	0
6	0	1	0	1	0
7	0	1	1	0	0
10	1	0	0	1	1
13	1	1	0	0	0

Illustration : un problème d'apprentissage

- Combien de fonctions possibles de 4 entrées booléennes et une sortie booléenne ?
- Combien de fonctions encore envisageables quand on connaît **7 exemples** ?

Apprendre est-il possible ?

Exemple	x_1	x_2	x_3	x_4	Etiquette
1	0	0	0	0	?
2	0	0	0	1	?
3	0	0	1	0	0
4	0	0	1	1	1
5	0	1	0	0	0
6	0	1	0	1	0
7	0	1	1	0	0
8	0	1	1	1	?
9	1	0	0	0	?
10	1	0	0	1	1
11	1	0	1	0	?
12	1	0	1	1	?
13	1	1	0	0	0
14	1	1	0	1	?
15	1	1	1	0	?
16	1	1	1	1	?

Encore un autre exemple

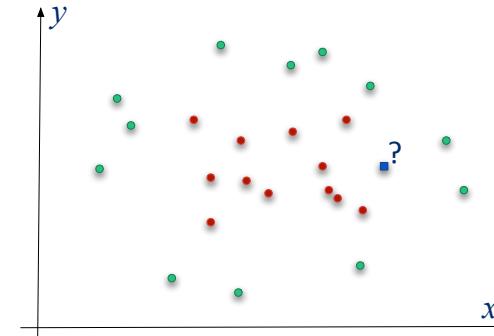
- Exemples décrits par :
 - *nombre* (1 ou 2); *taille* (petit ou grand); *forme* (cercle ou carré); *couleur* (rouge ou vert)
- Les objets appartiennent soit à la classe + soit à la classe -

Nb	Taille	Forme	Couleur	Étiquette
1	grand	carré	rouge	-
1	grand	carré	vert	+
2	petit	carré	rouge	+
2	grand	cercle	rouge	-
1	grand	cercle	vert	+
1	petit	cercle	rouge	+
1	petit	carré	vert	-
1	petit	carré	rouge	+

I will be questioned on
one new point
(Transductive learning)

Transductive learning

- I know **in advance** where I will be queried



Transduction (1)

Vous connaissez la question à l'avance.

- Quelle est l'étiquette pour la question ?

Nb	Taille	Forme	Couleur	Étiquette
1	grand	carré	rouge	-
1	grand	carré	vert	+
2	petit	carré	rouge	+
2	grand	cercle	rouge	-
1	grand	cercle	vert	+
1	petit	cercle	rouge	+
1	petit	carré	vert	-
1	petit	carré	rouge	+
2	petit	cercle	rouge	?

Transduction (2)

Vous connaissez la question à l'avance.

- Quelle est l'étiquette pour la question ?

Nb	Taille	Forme	Couleur	Étiquette
1	grand	carré	rouge	-
1	grand	carré	vert	+
2	petit	carré	rouge	+
2	grand	cercle	rouge	-
1	grand	cercle	vert	+
1	petit	cercle	rouge	+
1	petit	carré	vert	-
1	petit	carré	rouge	+
2	petit	cercle	vert	?

Transduction (3)

Vous connaissez la question à l'avance.

Nb	Taille	Forme	Couleur	Étiquette
1	grand	carré	rouge	-
1	grand	carré	vert	+
2	petit	carré	rouge	+
2	grand	cercle	rouge	-
1	grand	cercle	vert	+
1	petit	cercle	rouge	+
1	petit	carré	vert	-
1	petit	carré	rouge	+
1	-	cercle	-	?

Transduction (3)

Vous connaissez la question à l'avance.

I am going to be queried **there**, so this is the **important aspect**

Nb	Taille	Forme	Couleur	Étiquette
1	grand	carré	rouge	-
1	grand	carré	vert	+
2	petit	carré	rouge	+
2	grand	cercle	rouge	-
1	grand	cercle	vert	+
1	petit	cercle	rouge	+
1	petit	carré	vert	-
1	petit	carré	rouge	+
1	-	cercle	-	?

Which principle should guide transduction?

- Should we feel more certain about the induced answer if
 - 1. the query is **close** to some data points?
 - 2. the **answer does not change** when **the query point is changed** a little bit?
 - 3. the **answer does not change** when **the data points are changed** a little bit?
 - 4. ...

Which principle should guide transduction?

1- Proximity to data points

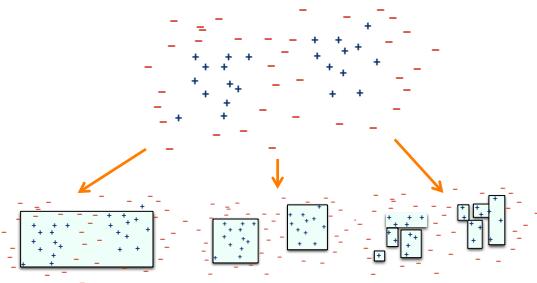
$$\hat{h}(\mathbf{x}_{m+1}) = \sum_{i=1}^m \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}_{m+1}) y_i$$

$$\hat{h}(\mathbf{x}_{m+1}) = \text{sign} \left\{ \sum_{i=1}^m \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}_{m+1}) y_i \right\}$$

How to choose κ ?

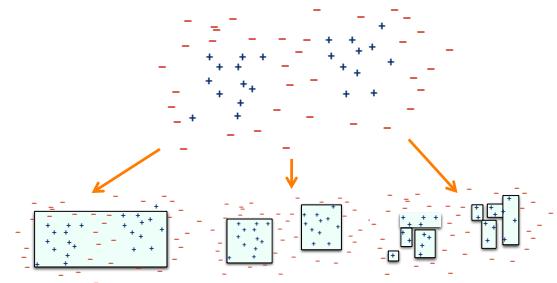
Which principle should guide transduction?

- 2- the answer does not change when the query point is changed a little bit?
- Capacity of H
- Regularization on the hypotheses



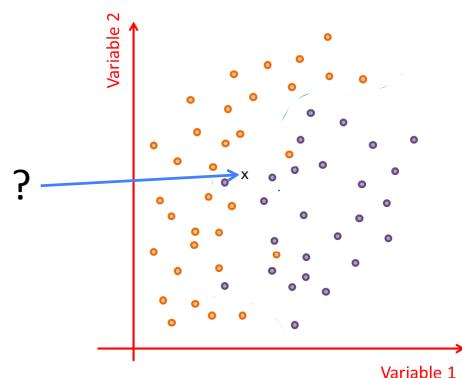
Which principle should guide transduction?

- 3- the answer does not change when the training data points are changed a little bit?
- Capacity of H
- Regularization on the hypotheses



Which principle should guide transduction?

- 1-2-3- closeness or robustness to small changes in the test or training data points



Which principle to guide transduction?

How to formally translate:

- If this is the question, I know this is
 - Important
 - Irrelevant

Transduction

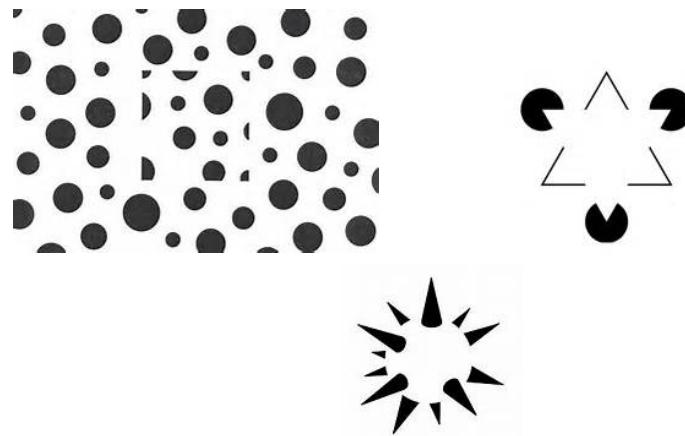
- Later other scenarios:

- Semi-supervised learning

- On-line transduction <-> tracking

Induction everywhere

Interprétation – compléTION de percepts



Interprétation – compléTION de percepts

A B C

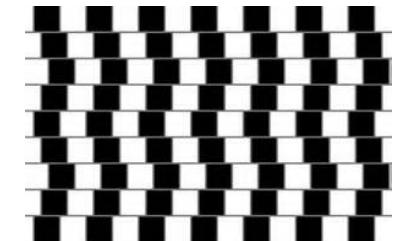
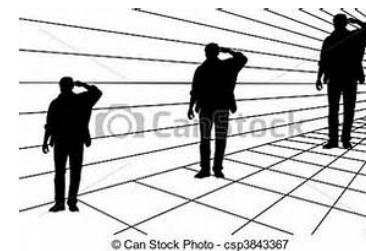
12
B
14

12
A B C
14

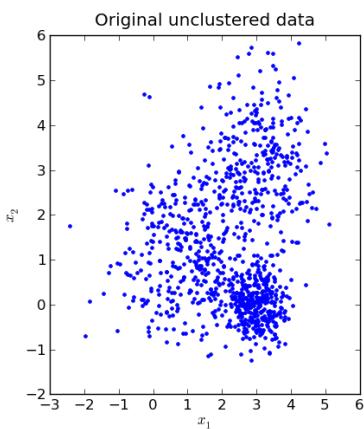
Interprétation – complémentation de percepts



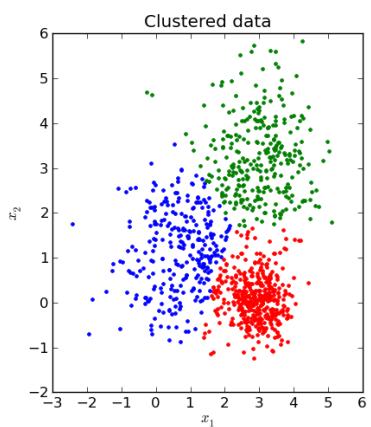
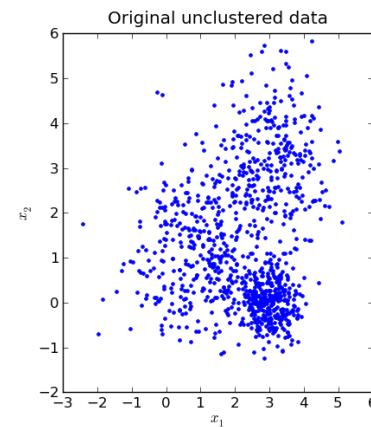
Illusions d'optique



Clustering



Clustering



Le rôle de l'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. »

Le rôle de l'induction

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

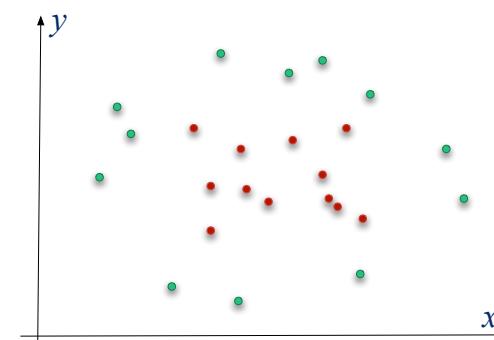
« We are hardly able to get through one waking hour without facing some situation (e.g. *will it rain or won't it?*) where we **do not have enough information to permit deductive reasoning**; but still we must decide immediately.
In spite of its familiarity, the formation of plausible conclusions is a **very subtle process**. »

Sequences

- 1 1 2 3 5 8 13 21 ...
- 1 2 3 5 ...
- 1 1 1 2 1 1 2 1 1 1 1 2 2 1 3 1 2 2 1 1 ...

Induction supervisée

- Comment choisir la fonction de décision ?



Interrogations

À chaque fois :

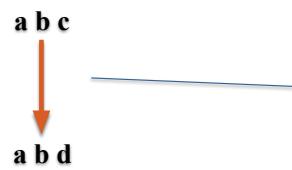
Cas particuliers => loi générale ou adaptation à nouveau cas

1. Qu'est-ce qui autorise ce passage ?

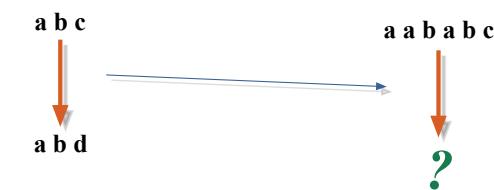
2. Est-ce que l'on peut garantir quelque chose ?

Induction by analogy

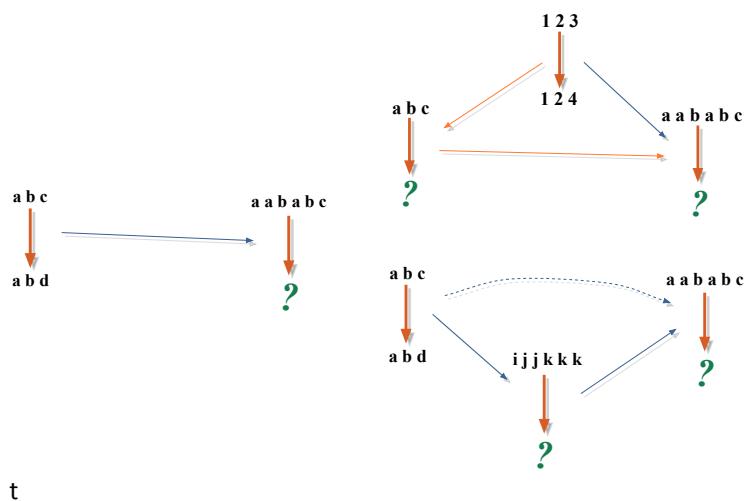
Transfer and analogy



- a b d
- i i j j k d
- i i j j k l
- i i j j k k
- ?

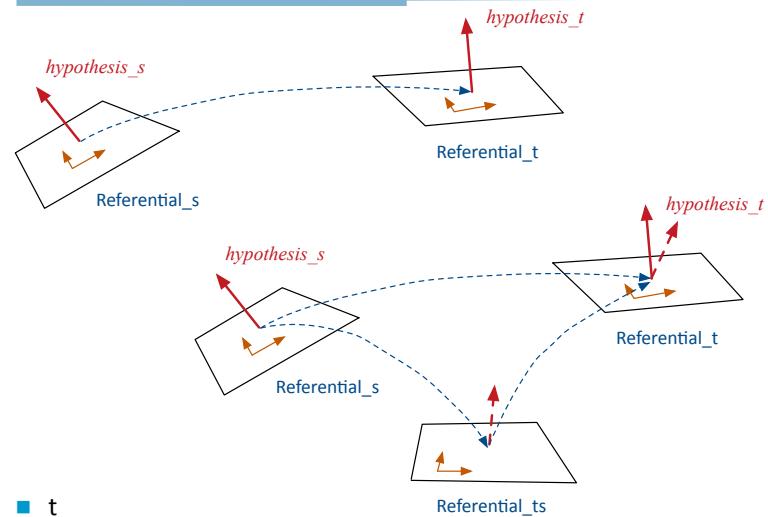


Analogy, transfer, and sequence effects



■ t

Transfer and sequence effects

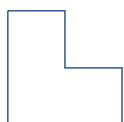


■ t

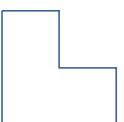
Effets de séquences

- Consigne : découper la figure suivante en n parties superposables

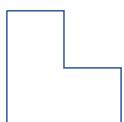
En 2 :



En 3 :



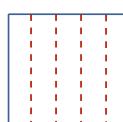
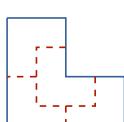
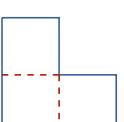
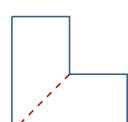
En 4 :



En 5 :



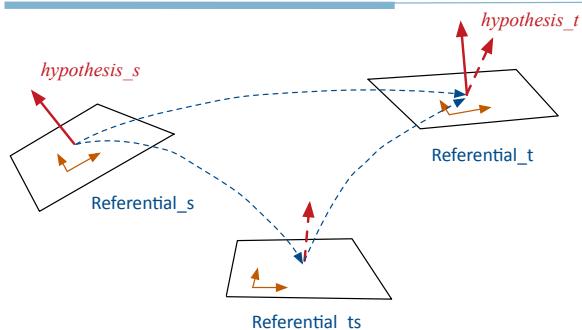
Covariant learning



No learning per se

But change of hypotheses through changes of referential

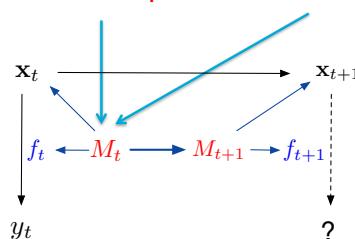
Transfer and sequence effects



1. Quelles équations pour les changements de référentiels et le transfert d'hypothèses ?
2. Comment montrer que ces équations sont optimales ?

Une formalisation

- Complexité de Kolmogorov
 - Repose sur un codage
 - Qui dépend de la connaissance a priori et de l'utilisation passée



$$K(M_t) + K(\mathbf{x}_t|M_t) + K(\mathbf{y}_t|M_t) + K(M_{t+1}|M_t) + K(\mathbf{x}_{t+1}|M_{t+1}) + K(f_{t+1}|M_{t+1})$$

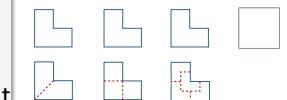
[A. Cornuéjols (1996) « Analogie, principe d'économie et complexité algorithmique »]

Nouveau scénario ...

... Nouveaux principes inductifs

1. Principe de pertinence maximale

- des situations rencontrées juste avant
- des connaissances mobilisées juste avant



2. Principe de non indifférence à la « question à venir »



Une formalisation

```
'abc' = Chaîne
    orientation : ->
    1er='A', 2ème='B', 3ème='C'
    TOTAL (longueur) : 21 bits
```

```
'abc' = Ensemble
    {'A', 'B', 'C'}
    TOTAL : 20 bits
```



```
'abc' = Séquence
    orientation : ->
    type d'éléments = lettres
    loi de succession :
        successeur(elt(lettre=x)) = elt(succ(lettre,1,x))
        L(lettre) + L(ler succ) + L(x) = 1/1/2 . 1/6 . 1
        = 1/1/12 = 4 bits
    longueur = 3
    commençant avec l'élément(lettre='A')
    TOTAL : 17 bits
```

Une formalisation

• Descripteurs utilisés dans la définition des structures :	
- orientation (\rightarrow / \leftarrow)	1 bit
- cardinalité ou nombre d'éléments : n	$\log_2(n) + 1$ bits (voir en-dessous)
- type d'éléments	$\log_2(1) + 1$ bits
- longueur : l	$L(x)$ bits
- commençant ou se terminant par l'élément = x	$(1/2) \rightarrow 1$ bit
• Lettre	$(1/2.26) \rightarrow 6$ bits
Une lettre particulière (e.g. 'd')	
• Chaîne (orientation, éléments)	$L = 3 + L(\text{orientation}) + \sum L(\text{éléments})$ e.g. $L('abbd'$ avec orientation $\rightarrow) = 3 + 1 + \log_2((1/2.26)^3) + L(3)$ $= 3 + 1 + 18 + 3 = 25$ bits
• Ensemble (type d'éléments, cardinalité, éléments)	$(1/8) \rightarrow 3$ bits
• Groupe (type d'éléments, nombre d'éléments, éléments)	$L = 3 + L(\text{type}) + L(\text{nb él.}) + \sum L(\text{éléments})$ $L = 3 + L(\text{type}) + L(\text{nb él.}) + \sum L(\text{éléments})$
• Séquence (orientation, type d'éléments, loi de succession ou nombre d'éléments, longueur, commençant ou se terminant par)	$(1/8)$
• Description et longueur d'une loi de succès	$L = 3 + L(\text{orient.}) + L(\text{type}) + L(\text{loj}) \text{ or } L(\text{nb él.}) + L(\text{long}) + L(\text{début/fin})$
succ(type-of-el., n, x) = le nième successeur de l'élément x du type type-of-el.	
$L = L(\text{type}) + L(n \text{ (voir ci-dessous)}) + L(x)$	
$L(n) = L(1/6)$	si $n=1$ ou -1 (ier successeur ou précédent)
$L(n) = L(1/3)$	si $n=0$ (même élément)
$L((1/3). (1/2)^p)$	sinon (avec $p=n$ si $n>0$, $p=-n$ sinon)
• Premier / Dernier (par rapport à l'orientation définie)	1 bit
nième	n bits

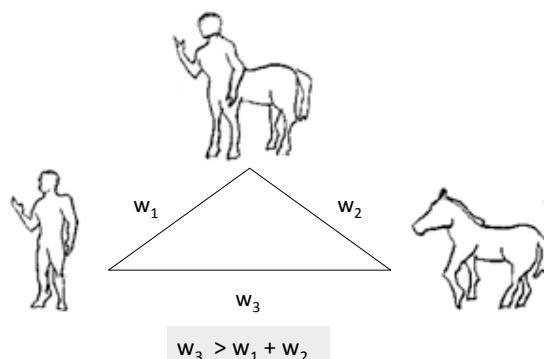
Table 1 : Liste des primitives de représentation et de leur longueur de description associée.

[A. Cornuéjols (1996) « Analogie, principe d'économie et complexité algorithmique »]



Course « InductionS » (A. Cornuéjols) 45 / 122

Need for non-symmetrical similarity



Adapted from: D.W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with non-metric distances: Image retrieval and class representation. PAMI 2000.

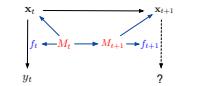
Questions

1. Comment construire une théorie de l'analogie ?

- Le choix du **critère inductif**
- Sa **déclinaison** en problème d'optimisation
- Algorithme

Cadre de l'analogie proportionnelle

Principe de pertinence maximale
+ Principe de non indifférence à la question



2. Comment valider ?

Pas de validation absolue

- Pourquoi une analogie serait jugée meilleure qu'une autre ?
- Compatible à la limite i.i.d. avec l'apprentissage statistique
- Permet d'obtenir de « meilleurs résultats » quand dérive de concept (i.e. conforme à nos attentes) ?
- Produit des **conséquences inattendues** (e.g. sur l'éducation) ?

[Murena & Cornuéjols (2016) « Minimum Description Length Principle applied to structure adaptation for classification under concept drift », IJCNN-2016.]



Course « InductionS » (A. Cornuéjols)

46 / 122

Analogy and dependency on the path followed

■ Sequencing effects



Course « InductionS » (A. Cornuéjols)

46 / 122



Course « InductionS » (A. Cornuéjols)

47 / 122

Analogy and dependency on the path followed

■ What kind of theoretical analysis

1. Would we like
2. Can we perform

What kind of theoretical guarantees
on induction can we get?

Le no-free-lunch theorem

Le no-free-lunch theorem

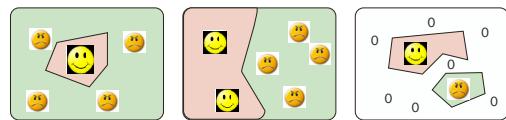
Théorème 2.1 (No-free-lunch theorem (Wolpert, 1992))

Pour tout couple d'algorithmes d'apprentissage \mathcal{A}_1 et \mathcal{A}_2 , caractérisés par leur distribution de probabilité a posteriori $\mathbf{p}_1(h|\mathcal{S})$ et $\mathbf{p}_2(h|\mathcal{S})$, et pour toute distribution d_X des formes d'entrées \mathbf{x} et tout nombre m d'exemples d'apprentissage, les propositions suivantes sont vraies :

1. En moyenne uniforme sur toutes les fonctions cible f dans \mathcal{F} :
$$\mathbb{E}_1[R_{\text{Réel}}|f, m] - \mathbb{E}_2[R_{\text{Réel}}|f, m] = 0.$$
2. Pour tout échantillon d'apprentissage \mathcal{S} donné, en moyenne uniforme sur toutes les fonctions cible f dans \mathcal{F} :
$$\mathbb{E}_1[R_{\text{Réel}}|f, \mathcal{S}] - \mathbb{E}_2[R_{\text{Réel}}|f, \mathcal{S}] = 0.$$
3. En moyenne uniforme sur toutes les distributions possibles $\mathbf{P}(f)$:
$$\mathbb{E}_1[R_{\text{Réel}}|m] - \mathbb{E}_2[R_{\text{Réel}}|m] = 0.$$
4. Pour tout échantillon d'apprentissage \mathcal{S} donné, en moyenne uniforme sur toutes les distributions possibles $\mathbf{p}(f)$:
$$\mathbb{E}_1[R_{\text{Réel}}|\mathcal{S}] - \mathbb{E}_2[R_{\text{Réel}}|\mathcal{S}] = 0.$$

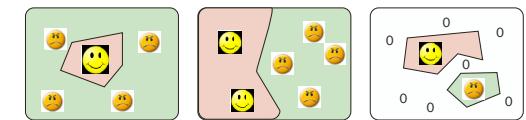
Le no-free-lunch theorem

Possible

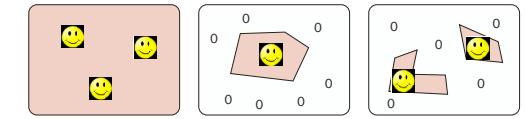


Le no-free-lunch theorem

Possible



Impossible



Déduction !

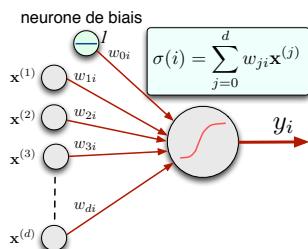
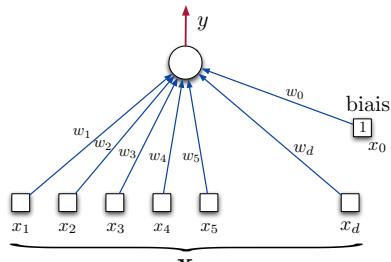
1. Tous les algorithmes inductifs se valent
2. Il ne peut y avoir aucune garantie sur les inductions réalisées

Analysis of the perceptron

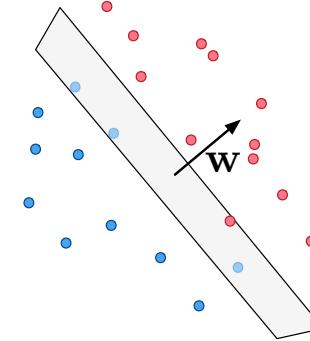
Allons à la plage !!

Le perceptron

- Rosenblatt (1958-1962)



Le perceptron : un discriminant linéaire



Le perceptron

■ Apprentissage des poids w_i

- Principe (règle de Hebb) : en cas de succès, ajouter à chaque connexion quelque chose de proportionnel à l'entrée et à la sortie

Règle du perceptron : apprendre seulement en cas d'échec

Algorithme 1 : Algorithme d'apprentissage du perceptron

tant que non convergence faire

```
    si la forme d'entrée est correctement classée alors
        | ne rien faire
    sinon
        |  $w(t+1) = w(t) + \eta x_t y_t$ 
    fin
    Passer à la forme d'apprentissage suivante
fin
```

Des propriétés remarquables !!

■ Convergence en un nombre fini d'étapes

- Indépendamment du **nombre** d'exemples
- Indépendamment de la **distribution** des exemples
- Indépendamment de la **dimension** de l'espace d'entrée



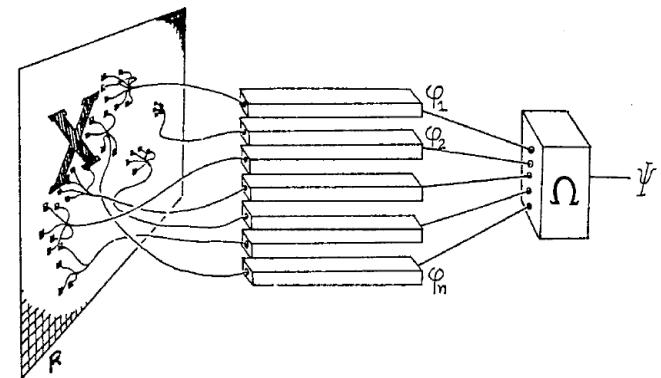
Si il existe au moins une séparatrice linéaire des exemples

Garantie de généralisation ??

- Théorèmes sur la performance par rapport à l'échantillon d'apprentissage
- Mais qu'en est-il pour des **exemples à venir** ?

Le Perceptron

- Rosenblatt (1958-1962)



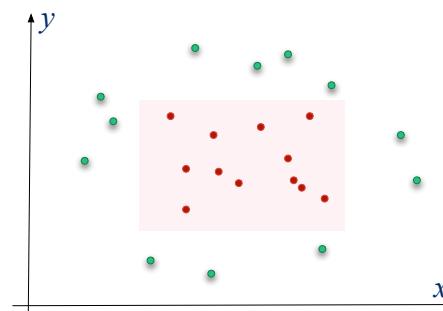
Target class: rectangles in \mathbb{R}^2

■ Sample

- Positive instances $P_{\mathcal{X}}^+$
- Negative instances $P_{\mathcal{X}}^-$

$$P_{\mathcal{X}}^+$$

$$P_{\mathcal{X}}^-$$

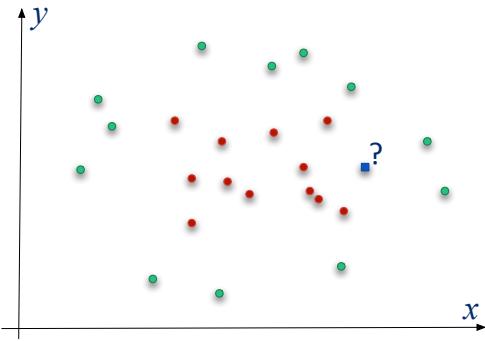


PAC learning

Probably Approximately Correct

Target class: unknown

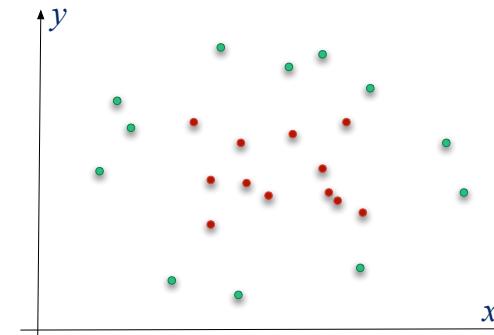
- What do we want to learn?



→ A decision function (prediction)

Target class: unknown

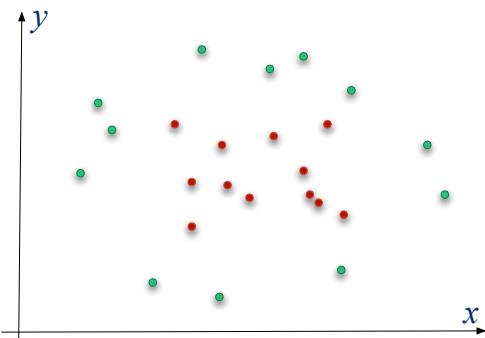
- How to learn?



Target class: rectangles in \mathbb{R}^2

- How to learn?

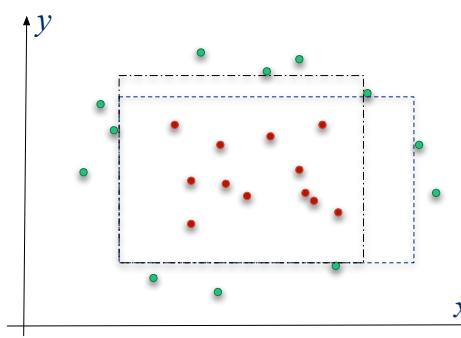
- If I know that the target concept is a rectangle



Target class: rectangles in \mathbb{R}^2

- How to learn?

- If I know that the target concept is a rectangle

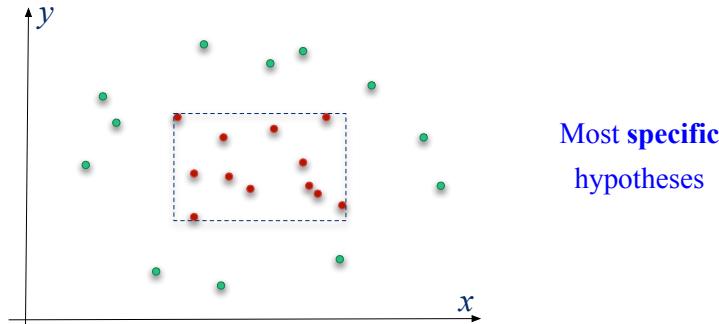


Most general
hypotheses

Target class: rectangles in \mathbb{R}^2

■ How to learn?

- If I know that the target concept is a rectangle



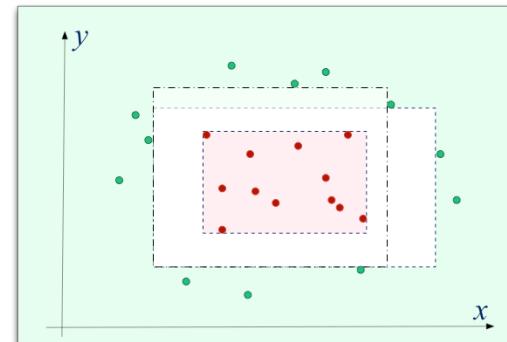
Most specific
hypotheses

Version
space

Target class: rectangles in \mathbb{R}^2

■ How to learn?

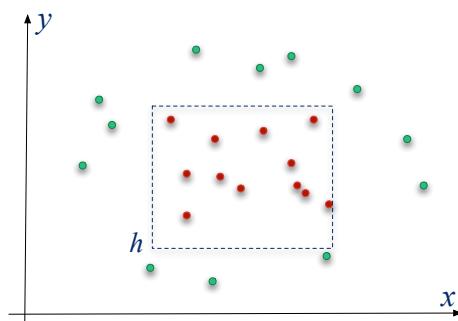
- Choice of one hypothesis h



Target class: rectangles in \mathbb{R}^2

■ Learning: choice de h

- Which performance to expect?



The statistical theory of learning

Which performance ?

Cost for a prediction error

- The *loss function*

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

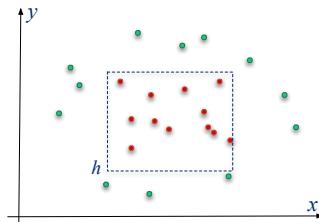
Which expected cost if I choose h ?

- The « real risk » (or true risk)

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

The statistical theory of learning

- Which **expected cost** when h is chosen?
 - Assuming that there is **no training error** on S

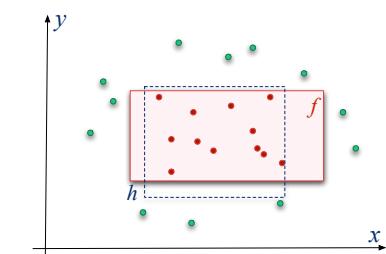
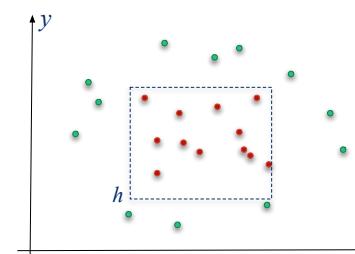


The « empirical risk »

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

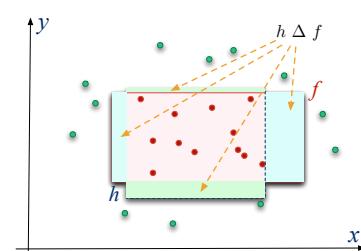
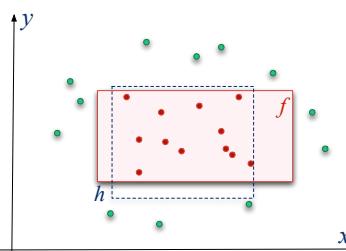
Statistical theory of learning: the ERM

- **Learning strategy:**
 - Select an **hypothesis with null empirical risk** (**no training error**)
 - Which generalization performance to expect for h ?



Statistical theory of learning: the ERM

- Select an **hypothesis with null empirical risk** (**no training error**)
- Which generalization performance to expect for h ?
- What is the **risk of getting error $R(h) > \epsilon$** ?



Question centrale : le principe inductif

- Le principe de **minimisation du risque empirique (ERM)**
... est-il sain ?

– Si je choisis h telle que $\hat{h} = \operatorname{ArgMin}_{h \in \mathcal{H}} \hat{R}(h)$

– Est-ce que h est bonne relativement au risque réel ?

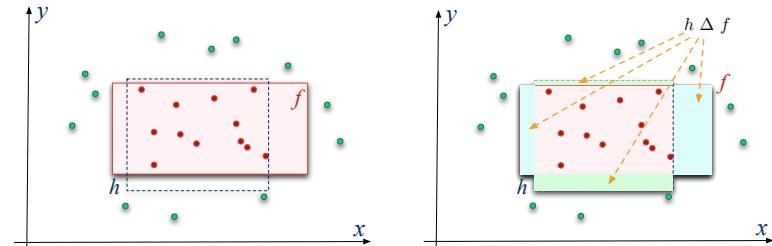
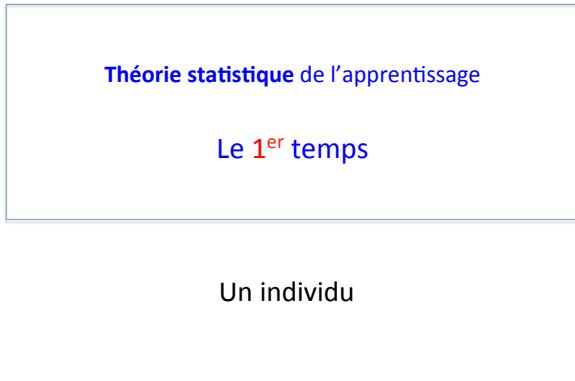
$$\hat{R}(\hat{h}) \xleftrightarrow{?} R(\hat{h})$$

– Est-ce que j'aurais pu faire beaucoup mieux ? $h^* = \operatorname{ArgMin}_{h \in \mathcal{H}} R(h)$

$$R(h^*) \xleftrightarrow{?} R(\hat{h})$$

Étude statistique pour UNE hypothèse

- choix d'une **hypothèse de risque empirique nul** (pas d'erreur sur l'échantillon d'apprentissage S)
- Quelle performance attendue pour h ?
- Quel est le risque d'avoir une erreur $R(h) > \varepsilon$?



Étude statistique pour UNE hypothèse

- Supposons h tq. $R(h) \geq \varepsilon$ (h « mauvaise »)
- Quelle est la probabilité que pourtant h ait été sélectionnée ?

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

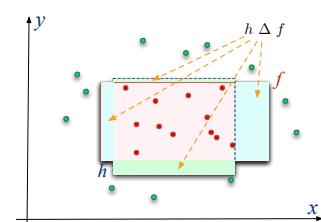
Après un exemple : $p(\hat{R}(h) = 0) \leq 1 - \varepsilon$

« tombe » en dehors de $h \Delta f$

Après m exemple (i.i.d.) :

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

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



Étude statistique pour UNE hypothèse

- On cherche : $\forall \varepsilon, \delta \in [0, 1] : p^m(R(h) \geq \varepsilon) \leq \delta$

Soit :

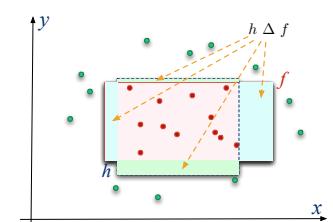
$$(1 - \varepsilon)^m \leq \delta$$

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

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

D'où :

$$m \geq \frac{\ln(1/\delta)}{\varepsilon}$$



Étude statistique pour $|\mathcal{H}|$ hypothèses

- Quelle est la probabilité que je choisisse une hypothèse h_{err} de risque réel $> \varepsilon$ et que je ne m'en aperçoive pas après l'observation de m exemples ?
- Probabilité de survie de h_{err} après 1 exemple : $(1 - \varepsilon)$
- Probabilité de survie de h_{err} après m exemples : $(1 - \varepsilon)^m$
- Probabilité de survie d'au moins une hypothèse dans \mathcal{H} : $|\mathcal{H}|(1 - \varepsilon)^m$
 - On utilise la probabilité de l'union $\mathbf{P}(A \cup B) \leq \mathbf{P}(A) + \mathbf{P}(B)$
- On veut que la probabilité qu'il reste au moins une hypothèse de risque réel $> \varepsilon$ dans l'espace des versions soit bornée par δ :

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

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

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

Théorie statistique de l'apprentissage

Le 2^{ème} temps

Quel individu dans la Foule

L'analyse « PAC learning »

- On arrive à :

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \mathbf{P}^m \left[R_{\text{Réel}}(h) \leq R_{\text{Emp}}(h) + \underbrace{\frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{m}}_{\varepsilon} \right] > 1 - \delta$$

Le principe de minimisation du risque empirique

n'est **sain que si** il y a des **contraintes sur l'espace des hypothèses**

PAC learning: definition

[Valiant, 1984]

Given $0 < \delta, \varepsilon < 1$, a *concept class* C is *learnable* by a polynomial time algorithm A if, for any distribution P of samples and any concept $c \in C$, there exists a polynomial $p(\cdot, \cdot, \cdot)$ such that A will produce with probability at least $1 - \delta$ a hypothesis $h \in C$ whose error is $\leq \varepsilon$ when given at least $p(m, 1/\delta, 1\varepsilon)$ independent random examples drawn according to P .

- **Worst case analysis**

- Against all distributions P
- For any target hypothesis in a class of hypotheses

- Notion of *computational complexity*

The statistical theory of learning

Uniform convergence bounds

Théorème 1 (Inégalité de Hoeffding). Si les ξ_i sont des variables aléatoires, tirées **indépendamment** et selon une **même distribution** et prenant leur valeur dans l'intervalle $[a, b]$, alors :

$$P\left(\left|\frac{1}{m} \sum_{i=1}^m \xi_i - \mathbb{E}(\xi)\right| \geq \varepsilon\right) \leq 2 \exp\left(-\frac{2m\varepsilon^2}{(b-a)^2}\right)$$

Appliquée au risque empirique et au risque réel, cette inégalité nous donne :

$$P(|R_{\text{Emp}}(h) - R_{\text{Réel}}(h)| \geq \varepsilon) \leq 2 \exp\left(-\frac{2m\varepsilon^2}{(b-a)^2}\right) \quad (1)$$

si la fonction de perte ℓ est définie sur l'intervalle $[a, b]$.

« **H** fini »

$$\begin{aligned} P^m[\exists h \in \mathcal{H} : R_{\text{Réel}}(h) - R_{\text{Emp}}(h) > \varepsilon] &\leq \sum_{i=1}^{|\mathcal{H}|} P^m[R_{\text{Réel}}(h^i) - R_{\text{Emp}}(h^i) > \varepsilon] \\ &\leq |\mathcal{H}| \exp(-2m\varepsilon^2) = \delta \end{aligned}$$

en supposant ici que la fonction de perte ℓ prend ses valeurs dans l'intervalle $[0, 1]$.

Bounding the true risk with the empirical risk + ...

■ \mathcal{H} finite, realizable case

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

■ \mathcal{H} finite, non realizable case

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

To sum up: for $|\mathcal{H}|$ finite

■ Non realizable case

$$\varepsilon = \sqrt{\frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{2m}} \quad \text{and} \quad m \geq \frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{2\varepsilon^2}$$

■ Realizable case

$$\varepsilon = \frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{m} \quad \text{and} \quad m \geq \frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{\varepsilon}$$

$|\mathcal{H}|$ infinite !!

- Effective dimension of \mathcal{H} = the **Vapnik-Chervonenkis dimension**

- **Combinatorial criterion**
- Size of the largest set of points (in general configuration) that can be labeled in any way by hypotheses drawn from \mathcal{H}

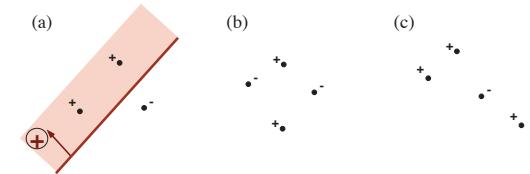
$$d_{VC}(\mathcal{H}) = \max\{m : \Pi_{\mathcal{H}}(m) = 2^m\}$$

Bound on the true risk

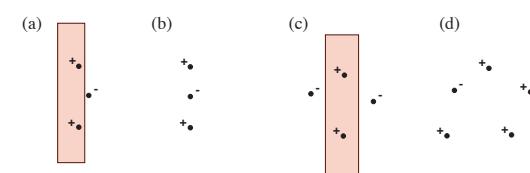
$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : P^m \left[R_{\text{Réel}}(h) \leq R_{\text{Emp}}(h) + \sqrt{\frac{8 d_{VC}(\mathcal{H}) \log \frac{2em}{d_{VC}(\mathcal{H})} + 8 \log \frac{4}{\delta}}{m}} \right] > 1 - \delta$$

VC dim: illustrations

- $d_{VC}(\text{linear separator}) = ?$



- $d_{VC}(\text{rectangles}) = ?$

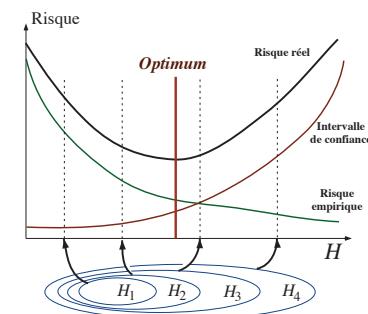


SRM : Structural Risk Minimization

Théorie statistique de l'apprentissage
Le 3^{ème} temps

Quelle Foule ?

- **Stratification** des espaces d'hypothèses
 - Faite *a priori* (indépendamment des données)
 - Par exemple en utilisant la d_{VC}



L'analyse « PAC learning » ou statistique

$$\forall h \in \mathcal{H}, \forall \delta \leq 1 : \mathbf{P}^m \left[R_{\text{Réel}}(h) \leq \underbrace{R_{\text{Emp}}(h)}_{\text{Risque empirique}} + \frac{\log |\mathcal{H}| + \log \frac{1}{\delta}}{m} \right] > 1 - \delta$$

Théorie statistique de l'apprentissage

Le 4^{ème} temps

■ Nouveau critère inductif :

- Le **risque empirique régularisé**
 1. Satisfaire les contraintes posées par les **exemples**
 2. Choisir le meilleur **espace d'hypothèses** (capacité de H)

Mais si l'espace des Foules
dépend des exemples ?

The « luckiness framework »

■ Principe : définir un **ordre sur H** qui **dépend des données** (≠ SRM)

- Si nous avons de la chance
- Alors, il n'y aura pas trop d'hypothèses mauvaises aussi compatibles avec la cible que les bonnes

[Shawe-Taylor et al., 1998], [Mendelsson & Philips, 2003]

L'apprentissage devient ...

1. **Le choix de l'espace des hypothèses H**
 - Nécessairement contraint
2. **Le choix d'un critère inductif**
 - Risque empirique nécessairement régularisé
3. **Une stratégie d'exploration de H** pour minimiser le risque empirique régularisé
 - Faire ce qu'il faut pour que l'exploration soit efficace
 - Rapide
 - Si possible un seul optimum

Un paradigme triomphant

Apprentissage = choix de normes + optimisation
(~ 1995 - ~20??)

Nouvelle perspective

■ Poser un problème d'apprentissage, c'est :

1. L'exprimer sous forme d'**un critère inductif** à optimiser

- **Risque empirique**

- avec une **fonction d'erreur** adéquate

- Un **terme de régularisation**

$$h^* = \underset{h \in \mathcal{H}}{\text{ArgMin}} [R_{\text{Emp}}(h) + \lambda \text{reg}(h)]$$

- exprimant les contraintes
 - et **connaissances a priori**
 - si possible conduisant à problème convexe

2. Trouver un **algorithme d'optimisation** adapté

Cadre séduisant

- **Algorithme d'apprentissage**
 - Générique : **minimisation du risque empirique régularisé**
 - Apprentissage = optimisation
- **Faible a priori sur le monde**
 - Suppose données (et questions) **i.i.d.**
 - $f \in H$ ou $f \notin H$
 - **Valable dans le pire cas** : contre toute distribution cible
- **Bornes en généralisation**
 - Formalisation mathématique **supportant son bien-fondé**

Un paradigme général

- Boosting
- Arbres de décisions (random forests)
- Régression logistique
- Réseaux de neurones
- Séparateurs à Vastes Marges (SVM)
- ...

« Traduction » : préférence pour les hypothèses parcimonieuses

■ Recherche d'hypothèse linéaire parcimonieuse

$$h^* = \underset{h \in \mathcal{H}}{\text{ArgMin}} \left[\frac{1}{m} \sum_{i=1}^m \ell(h(\mathbf{x}_i), y_i) + \lambda \text{reg}(h) \right]$$

$$h^* = \underset{h \in \mathcal{H}}{\text{ArgMin}} \left[\frac{1}{m} \sum_{i=1}^m \ell(h(\mathbf{x}_i), y_i) + \lambda \|h\|_1 \right]$$

Norme ℓ_1 : $\|\mathbf{w}\|_1 = \sum_{j=1}^p |w_j|$

■ Méthodes de type LASSO

« Traduction » : apprentissage multi-tâches

■ T tâches de classification binaire définies sur $X \times Y$

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

$$h_j(\mathbf{x}) = \mathbf{w}_j \cdot \mathbf{x} \quad \text{Hypothèses linéaires}$$

Partage entre tâches $\mathbf{w}_j = \mathbf{w}_0 + \mathbf{v}_j$

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

[3] 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 $S_u = \{\mathbf{x}_i^t\}_{i=1}^m$ l'échantillon S privé de ses étiquettes, on veut minimiser :

$$\min_{\mathbf{w}} c m \text{R}_S(G_{\rho_w}) + a m \text{dis}_{\rho_w}(S_u, T_u) + \text{KL}(\rho_w \parallel \pi_0), \quad (7.5)$$

où $\text{dis}_{\rho_w}(S_u, T_u) = \left| \mathbb{E}_{(h, h') \sim \rho_w} \text{R}_{S_u}(h, h') - \mathbb{E}_{(h, h') \sim \rho_w} \text{R}_{T_u}(h, h') \right|$ est le désaccord empirique entre S_u et T_u spécialisé à une distribution ρ_w sur l'espace \mathcal{H} des classificateurs 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 $\ell_{\text{dis}}(x) = 2\ell_{\text{Erf}}(x)\ell_{\text{Erf}}(-x)$ (illustrée sur la figure [7.1]), pour toute distribution D sur X , on a :

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

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

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

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

$\ell_{\text{Erf}_{\text{conv}}}(\cdot)$ (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 :

Quelles garanties exactement ?

Apprentissage statistique : quelles garanties ?

- Lien entre risque empirique et risque réel
 - Coût d'usage de h (e.g. taux d'erreur)

Ne dit rien sur :

- Seulement si
 - Intelligibilité
 - Fécondité
 - Insertion dans une théorie du domaine
 - Monde stationnaire
 - Données i.i.d.
 - Questions i.i.d. !!!

Limites

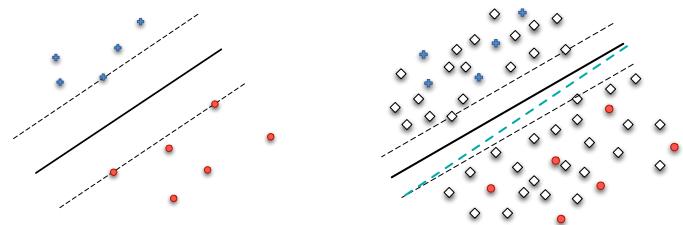
- Apprentissage passif et données et questions i.i.d.
 - Agents situés : le monde n'est pas i.i.d.
- Requiert beaucoup d'exemples
 - Nous sommes beaucoup plus efficaces
 - « Producteurs de théories », théories que nous testons ensuite
- Pas adapté à la recherche de causalités
- Pas intégré avec un raisonnement

Ces machines apprenantes ne sont pas des machines pensantes

Semi-supervised learning
And its theoretical analysis

Semi supervised learning

- General principle
 - The decision function does not cut through high density regions of X
 - P_X is related to $P_{Y|X}$
 - The S3VM algorithm



How to derive guarantees for semi-supervised learning?

[Balcan & Blum (2006). "An augmented PAC model for semi-supervised learning"]

- Let's assume that it is reasonable that the frontier between two classes does not cut through high density regions of the input space X
 - Then the unlabeled data points bring constraints on the possible decision functions -> gain of information
- Formally: let's define a compatibility function $\chi : \mathcal{H} \times X \rightarrow [0,1]$
 - E.g. $\chi(h, x)$ could be an increasing function of the distance of x to the decision function (separator) h

$$\chi(h, \mathcal{D}) = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} [\chi(h, \mathbf{x})] \quad \text{Compatibility between } h \text{ and } \mathcal{D}$$

$$\chi(h, S) = \frac{1}{m} \sum_{i=1}^m \chi(h, \mathbf{x}_i) \quad \text{Empirical compatibility measured on } S$$

How to derive guarantees for semi-supervised learning?

Theorem:

If we see m_u unlabeled examples and m_l labeled examples, where

$$m_u \geq \frac{1}{\varepsilon} \left[\ln |\mathcal{H}| + \ln \frac{2}{\delta} \right] \quad \text{and} \quad m_l \geq \frac{1}{\varepsilon} \left[\ln |\mathcal{H}_{\mathcal{D}, \mathcal{X}}(\varepsilon)| + \ln \frac{2}{\delta} \right]$$

then, with probability $\geq 1 - \delta$, any $h \in \mathcal{H}$ with $\widehat{err}(h) = 0$

and $\widehat{err}_{unl}(h) = 0$ has $err(h) \leq \varepsilon$

How to derive guarantees for semi-supervised learning?

■ Incompatibility $er_{unl}(h) = 1 - \chi(h, \mathcal{D})$

$$\widehat{err}_{unl}(h) = 1 - \chi(h, S)$$

- Let's define the set of hypotheses whose incompatibility is at most some given value τ

$$\mathcal{H}_{\mathcal{D}, \mathcal{X}}(\tau) = \{h \in \mathcal{H} : err_{unl}(h) \leq \tau\}$$

$$\mathcal{H}_{S, \mathcal{X}}(\tau) = \{h \in \mathcal{H} : \widehat{err}_{unl}(h) \leq \tau\}$$

How to derive guarantees for semi-supervised learning?

Proof:

The probability that a given hypothesis h with $err_{unl}(h) > \varepsilon$ has $\widehat{err}_{unl}(h) = 0$

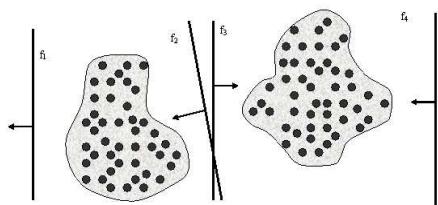
is at most $(1 - \varepsilon)^{m_u} < \frac{\delta}{2|\mathcal{H}|}$ for the given value of m_u .

Therefore, by the union bound, the number of unlabeled examples is sufficient to ensure that, with probability $1 - \delta/2$, only hypotheses in $\mathcal{H}_{\mathcal{D}, \mathcal{X}}(\varepsilon)$ have $\widehat{err}_{unl}(h) = 0$.

Similarly, the number of labeled examples ensures that with probability $1 - \delta/2$, none of those hypotheses whose true error is $\geq \varepsilon$ have an empirical error of 0, yielding the theorem.

How to derive guarantees for semi-supervised learning?

Note: Notice that for the setting of Example 1, in the worst case (over distributions D) this will essentially recover the standard margin sample-complexity bounds for the number of labeled examples. In particular, $C_{S,\chi}(0)$ contains only those separators that split S with margin $\geq \gamma$, and therefore, $s = |C_{S,\chi}(0)[2m_l, \bar{S}]|$ is no greater than the maximum number of ways of splitting $2m_l$ points with margin γ . However, if the distribution is helpful, then the bounds can be much better because there may be many fewer ways of splitting S with margin γ . For instance, in the case of two well-separated “blobs” illustrated in Figure 2.1, if S is large enough, we would have just $s = 4$.



How to derive guarantees for semi-supervised learning?

The theorem assumes

The data is i.i.d.

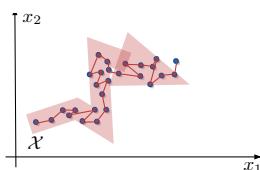
- Probability of each hypothesis to obey the criteria and still be in error
- Union bound

The true target functions obey the compatibility criterion

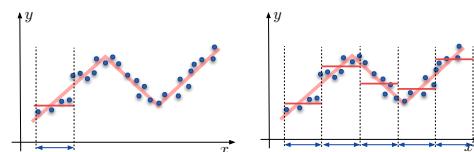
What if Nature does not obey these assumptions?

E.g. the interesting decision functions cut through high density regions of X

Le tracking [Sutton et al., 2007]



In tracking, the learning agent receives inputs that are driven by a time dependent process. It therefore encounters different parts of the environment at different times.



Even though the world involves a piecewise linear law, the learning agent may perform well by maintaining a very simple model, a constant, over its local environment.

Sutton, Koop & Silver (2007) « On the role of tracking in stationary environments ». ICML-07.

Tracking [Sutton et al., 2007]

Temporal Consistency

Small memory
Simple \mathcal{H}

i.i.d. data

Large memory
“Complex” \mathcal{H}

Conclusions

Conclusions

1. L'induction est au **centre** de l'apprentissage et est un problème **sous-constraint**.
2. Il ne peut y avoir de validation absolue de l'induction
3. On ne peut garantir une performance en induction qu'en faisant des **méta-présupposés** sur le monde
 - E.g. données i.i.d.

- Une **théorie de l'induction** vise à
 - Proposer des **méta-présupposés** « raisonnables »
 - Offrir un **cadre formel** dans lequel obtenir des « **théorèmes du lampadaire** »
 - *Si les pré-supposés sont vérifiés par les données alors je peux garantir que ...*

Un guide :
La **compatibilité**
des théories
à leurs interfaces

Leçon

■ (Quasi) garantie de bon résultat

- Si le signal présente les propriétés supposées a priori
- Alors la méthode assure que le signal cible (d'origine) sera (quasi) reconstruit

Théorème « du lampadaire »



Conclusions : la théorie statistique de l'apprentissage

- **Performance** visée : l'**espérance de coût d'usage**
(i.e. mais **pas causalité, pas intelligibilité, pas articulation à raisonnement, ...**)

- Valable si **monde stationnaire + données i.i.d. + questions i.i.d.**

■ Même le “big data” va présenter des **défis sortant du cadre**

- Même si on stocke tout : il faut **indexer la mémoire** => **choix** (pb de l'utilité)
- Objectif : aider à la **décision** => il faut **articuler au raisonnement**
- **Systèmes** d'apprentissage collaborant entre eux
=> gros problèmes de **spécification** des **entrées** et **sorties**
[Léon Bottou, ICML-2015]

Conclusions : de « nouveaux » scénarios

- Assez **peu de données**
 - on apprend (très souvent) avec très peu
- L'**histoire passée compte** : **éducation**
 - Effets de séquence
- Apprendre pour **construire des théories**
 - Nous construisons constamment des théories micro et macro

Conclusion (fin)

Comment faire ?

Conclusion (fin)

Comment faire ?

La construction de nouveaux paradigmes est difficile

Surtout quand le **paradigme dominant**

Apprentissage statistique

- Semble très bien fonctionner
- Semble être parfaitement adapté aux besoins (e.g. « big data »)
- Fait appel à des mathématiques sophistiquées (valorisées, intimidantes et forcément objectives)

Intuition des bons problèmes + intrépidité + rigueur

Qu'est-ce qu'une bonne induction ?

- [Van Frassen (1979) « The scientific image »]

Une nouvelle **théorie scientifique** vise à accroître les connaissances.

- Pas seulement pour « expliquer » les phénomènes connus
- Mais aussi pour suggérer de nouvelles règles ou prédictions sur des observations et pour en corriger des anciennes

Un peu d'histoire

IA et résolution automatique de problèmes

- Arch [Winston, 1972]
 - Stratégie de recherche guidée dans un espace de descriptions structurées
- [Simon & Lea (1979) « *Problem-solving and rule induction: a unified view* »]
 - Se focalisent sur les mécanismes de raisonnement (generate_and_test, heuristic search, hypothesis_and_match)
 - Au lieu de chercher à résoudre un problème, on cherche à « couvrir » des exemples, mais mêmes types de procédures
 - GPS -> GRI (Generalized Rule Induction)
- [Tom Michell (1980, 1982) « *Generalization as Search* », « *The need for biases in learning generalizations* »]
 - Comment organiser la recherche d'une (bonne) hypothèse
 - Si pas de biais, l'apprentissage ne peut pas faire mieux que l'apprentissage par cœur
- [David Haussler (1988) « *Quantifying inductive bias: AI learning algorithms and Valiant's learning* »]
 - Quantification du biais (par la dimension de Vapnik-Cervonenkis) de classes d'expressions logiques

L'apprentissage ...

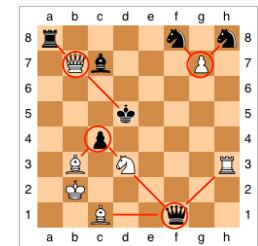
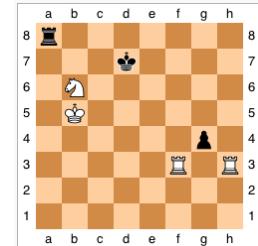
... comme

l'amélioration de l'efficacité d'un résolveur de problème

Apprendre à partir d'un exemple

Explanation-Based Learning

1. Un exemple unique
2. Recherche de la preuve de la « fourchette »
3. Généralisation



Explanation-Based Learning

Ex : apprendre le concept `empilable(Objet1, Objet2)`

■ Théorie :

```
(T1) : poids(X, W) :- volume(X, V), densité(X, D), W is V*D.  
(T2) : poids(X, 50) :- est-un(X, table).  
(T3) : plus-léger(X, Y) :- poids(X, W1), poids(Y, W2), W1 < W2.
```

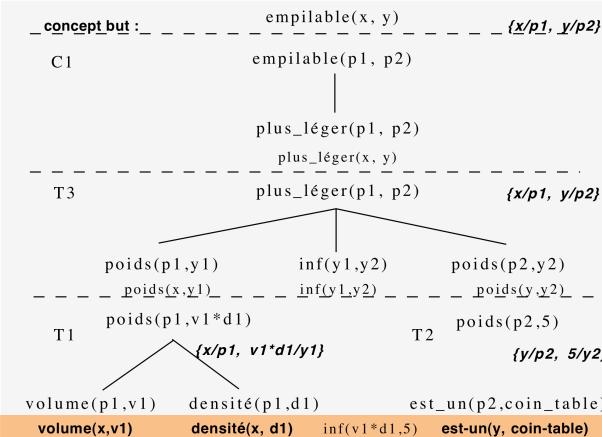
■ Contrainte d'opérationnalité :

- Concept à exprimer à l'aide des prédictats *volume*, *densité*, *couleur*, ...

■ Exemple positif (solution) :

```
sur(objet1, objet2).           volume(objet1, 1).  
est_un(objet1, boîte).          volume(objet2, 0.1).  
est_un(objet2, table).          propriétaire(objet1, frederic).  
couleur(objet1, rouge).         densité(objet1, 0.3).  
couleur(objet2, bleu).          matériau(objet1, carton).  
matériau(objet2, bois).         propriétaire(objet2, marc).
```

Explanation-Based Learning



Arbre de preuve généralisé obtenu par régression du concept cible dans l'arbre de preuve en calculant à chaque étape les littéraux les plus généraux permettant cette étape.

Explanation-Based Learning

■ Induction à partir d'un seul exemple

- ... et d'une théorie forte du domaine

■ Langage de la logique

■ Opérateurs de raisonnement (déduction, ...)

■ Maintenant utilisées dans les « solveurs » de problèmes SAT.

Explanation-Based Learning

■ Que cherche-t-on à prouver ?

■ Qu'est-ce qui est une bonne (moins bonne) théorie / méthode ?

Explanation-Based Learning

- Que cherche-t-on à prouver ?
 - Qu'est-ce qui est une bonne (moins bonne) théorie / méthode ?
1. Méthode améliorant les performances de résolution de problème
 - [Steve Minton (1990) « Quantitative results concerning the utility of Explanation-Based Learning »]
 2. Méthode « reproduisant » les performances (et limites) d'un agent cognitif naturel (animal ou humain)
 - [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 »]

Explanation-Based Learning

1. On ne s'interroge pas directement sur la validité des hypothèses induites (i.e. espérance de coût)
2. « Utility » ~ espérance d'utilité en termes de situations de résolution de problèmes

Explanation-Based Learning

- Questions traitées dans les publications
 - Quel type d'induction en fonction de la notion de conséquence logique utilisée ?
 - Comment utiliser la théorie du domaine ?
 - Que faire si la théorie du domaine est incomplète ou erronée ?
 - Comment utiliser des contre-exemples ?
 - Quel est le rôle du critère d'opérationnalité ?
 - Que faire si on obtient plusieurs arbres de preuves ?

Explanation-Based Learning

- Est-ce de l'induction ?

Déduction guidée par des critères d'opérationnalité

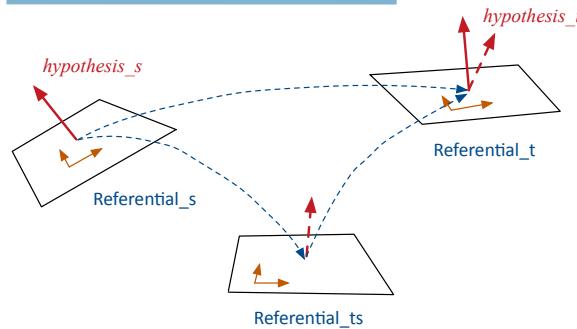
How can we prove the **validity**
of a new inductive principle?

How to obtain guarantees

in analogical reasoning?

And **which** type of guarantees?

Transfer and sequence effects

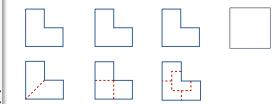


1. Quelles équations pour les changements de référentiels et le transfert d'hypothèses ?
2. Comment montrer que ces équations sont optimales ?

Nouveau scénario ...

... Nouveaux principes inductifs

1. Principe de **pertinence maximale**
 - des situations rencontrées juste avant
 - des connaissances mobilisées juste avant



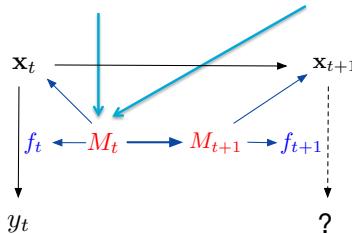
2. Principe de **non indifférence à la « question à venir »**



Une formalisation

■ Complexité de Kolmogorov

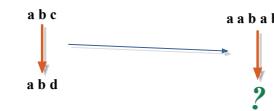
- Repose sur un codage
- Qui dépend de la connaissance a priori et de l'utilisation passée



$$K(M_t) + K(\mathbf{x}_t|M_t) + K(y_t|M_t) + K(M_{t+1}|M_t) + K(\mathbf{x}_{t+1}|M_{t+1}) + K(f_{t+1}|M_{t+1})$$

[A. Cornuéjols (1996) « Analogie, principe d'économie et complexité algorithmique »]

Une formalisation



```
'abc' = Chaîne          (1/8)
orientation : ->
1er='A', 2ème='B', 3ème='C'
TOTAL (longueur) : 21 bits
```

```
'abc' = Ensemble        (1/8)
{'A', 'B', 'C'}
TOTAL : 20 bits
```

```
'abc' = Séquence        (1/8)
orientation : ->
type d'éléments = lettres
loi de succession :
successeur(elt(lettre=x)) = élé(succ(lettre,1,x))
L(lettre) + L(1er succ) + L(x) = L(1/2 . 1/6 . 1)
= 1/(12) = 4 bits
longueur = 3
3 bits
commençant avec l'élément(lettre='A')
TOTAL : 17 bits
```

[A. Cornuéjols (1996) « Analogie, principe d'économie et complexité algorithmique »]

Une formalisation

• Descripteurs utilisés dans la définition des structures :	
- orientation (-> / <-)	1 bit
- cardinalité ou nombre d'éléments : n	$\log_2(n) + 1$ bits
- type d'éléments	(voir en-dessous)
- longueur : l	$\log_2(l) + 1$ bits
- commençant ou se terminant par l'élément = x	L(x) bits
• Lettre	(1/2) → 1 bit
Une lettre particulière (e.g. 'd')	(1/2.26) → 6 bits
• Chaîne (orientation,éléments)	(1/8) → 3 bits
$L = 3 + L(\text{orientation}) + \sum L(\text{éléments})$	
e.g. L('abcd' avec orientation =>) = $3 + 1 + \log_2((1/2.26)^3) + L(3)$	
$= 3 + 1 + 18 + 3 = 25$ bits	
• Ensemble (type d'éléments, cardinalité, éléments)	(1/8) → 3 bits
$L = 3 + L(\text{type}) + L(\text{cardinalité}) + \sum L(\text{éléments})$	
• Groupe (type d'éléments, nombre d'éléments, éléments)	(1/8) → 3 bits
$L = 3 + L(\text{type}) + L(\text{nb él.}) + \sum L(\text{éléments})$	
• Séquence (orientation, type d'éléments, loi de succession ou nombre d'éléments, longueur, commençant ou se terminant par)	(1/8)
$L = 3 + L(\text{orient.}) + L(\text{type}) + L(\text{loi}) \text{ or } L(\text{nb él.}) + L(\text{long}) + L(\text{début/fin})$	
• Description et longueur d'une loi de succession	
$\text{succ}(\text{type-of-el.}, n, x) = \text{le } n\text{ème successeur de l'élément } x \text{ du type type-of-el.}$	
$L = L(\text{type}) + L(n \text{ (voir ci-dessous)}) + L(x)$	
$L(n) = L(1/6) \quad \text{si } n=1 \text{ ou } -1$	(1er successeur ou précédent)
$L(1/3) \quad \text{si } n=0$	(même élément)
$L((1/3) \cdot (1/2)^p) \quad \text{sinon (avec } p=n \text{ si } n>0, p=-n \text{ sinon)}$	
• Premier / Dernier (par rapport à l'orientation définie)	1 bit
• nième	n bits

Table 1 : Liste des primitives de représentation et de leur longueur de description associée.

[A. Cornuéjols (1996) « Analogie, principe d'économie et complexité algorithmique »]

Questions

1. Comment construire une théorie de l'analogie ?

- Le choix du critère inductif
- Sa déclinaison en problème d'optimisation
- Algorithme

Cadre de l'analogie proportionnelle

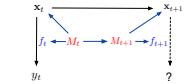
Principe de pertinence maximale

+ Principe de non indifférence à la question

2. Comment valider ?

Pas de validation absolue

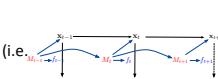
- Pourquoi une analogie serait jugée meilleure qu'une autre ?



- Compatible à la limite i.i.d. avec l'apprentissage statistique

- Permet d'obtenir de « meilleurs résultats » quand dérive de concept (i.e. conforme à nos attentes) ?

- Produit des conséquences inattendues (e.g. sur l'éducation) ?



[Murena & Cornuéjols (2016) « Minimum Description Length Principle applied to structure adaptation for classification under concept drift », IJCNN-2016.]