What is the definition of

a **good Machine Learning algorithm**?

After 60 years, is this a closed problem? And if not …

Antoine Cornuéjols

AgroParisTech – INRA MIA 518

antoine.cornuejols@agroparistech.fr

AgroParisTech

AI and ML everywhere in the medias today

Outline

1. What **does work**

2. **Limitations**

3. Learning comes with **which guarantees**?
   - Induction: how to win this game?
   - The statistical learning theory
   - A **closed case**? Not so sure

4. Other paradigms? An **historical perspective**

5. Is there **a paradigmatic change in sight**?

6. Conclusions

What does work

Object recognition in images

The ImageNet competition

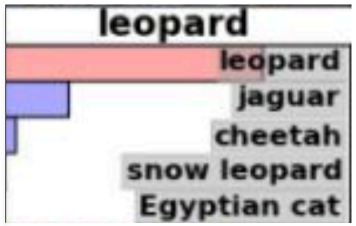
- More than **15M** high resolution **labeled images**
- Approximately **22K categories**
- Taken from the Web and labeled using Amazon Mechanical Turk

Illustration : ImageNet

The ImageNet competition

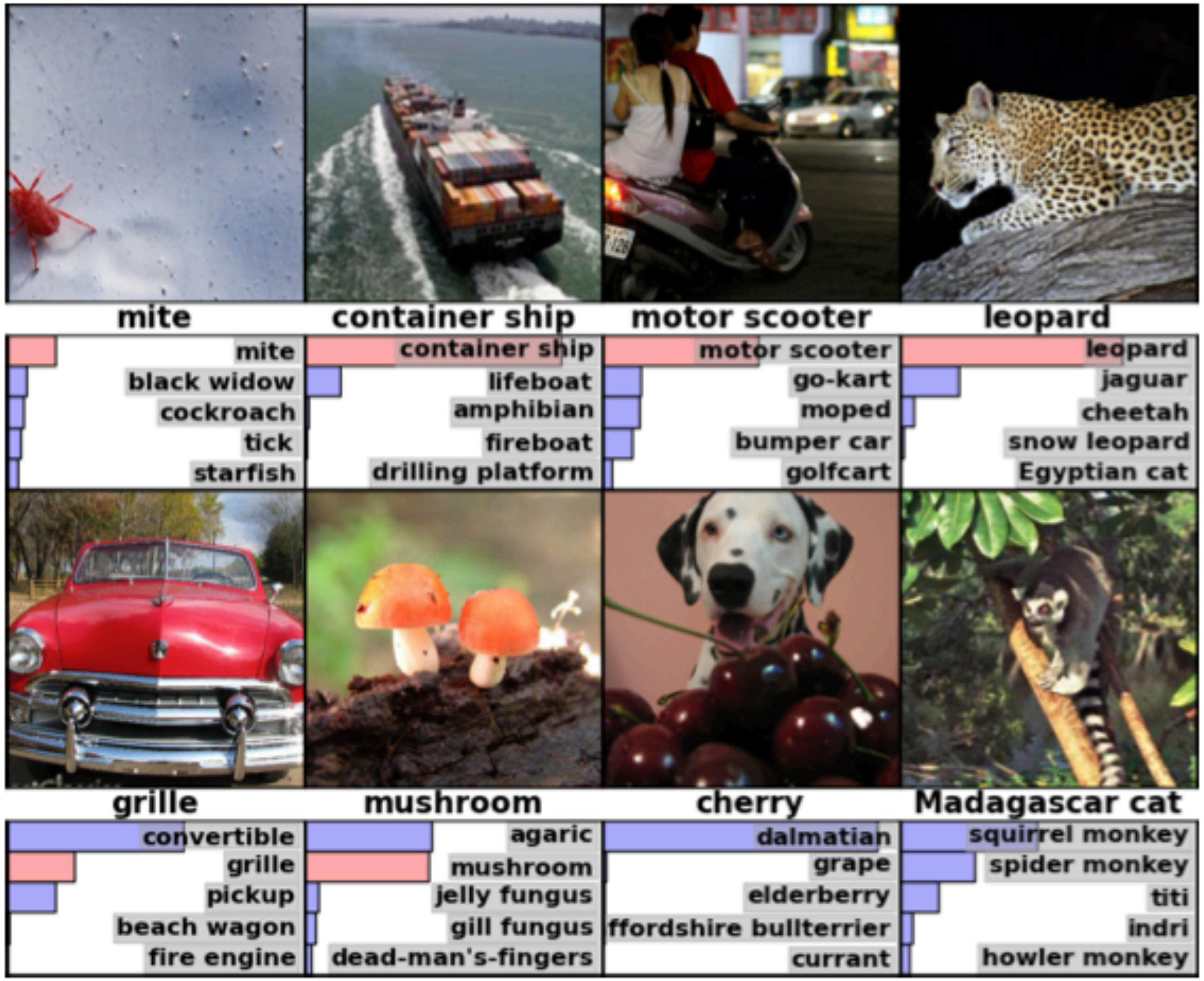
- More than **15M** high resolution **labeled images**
- Approximately **22K categories**
- Taken from the Web and labeled using Amazon Mechanical Turk

Classification

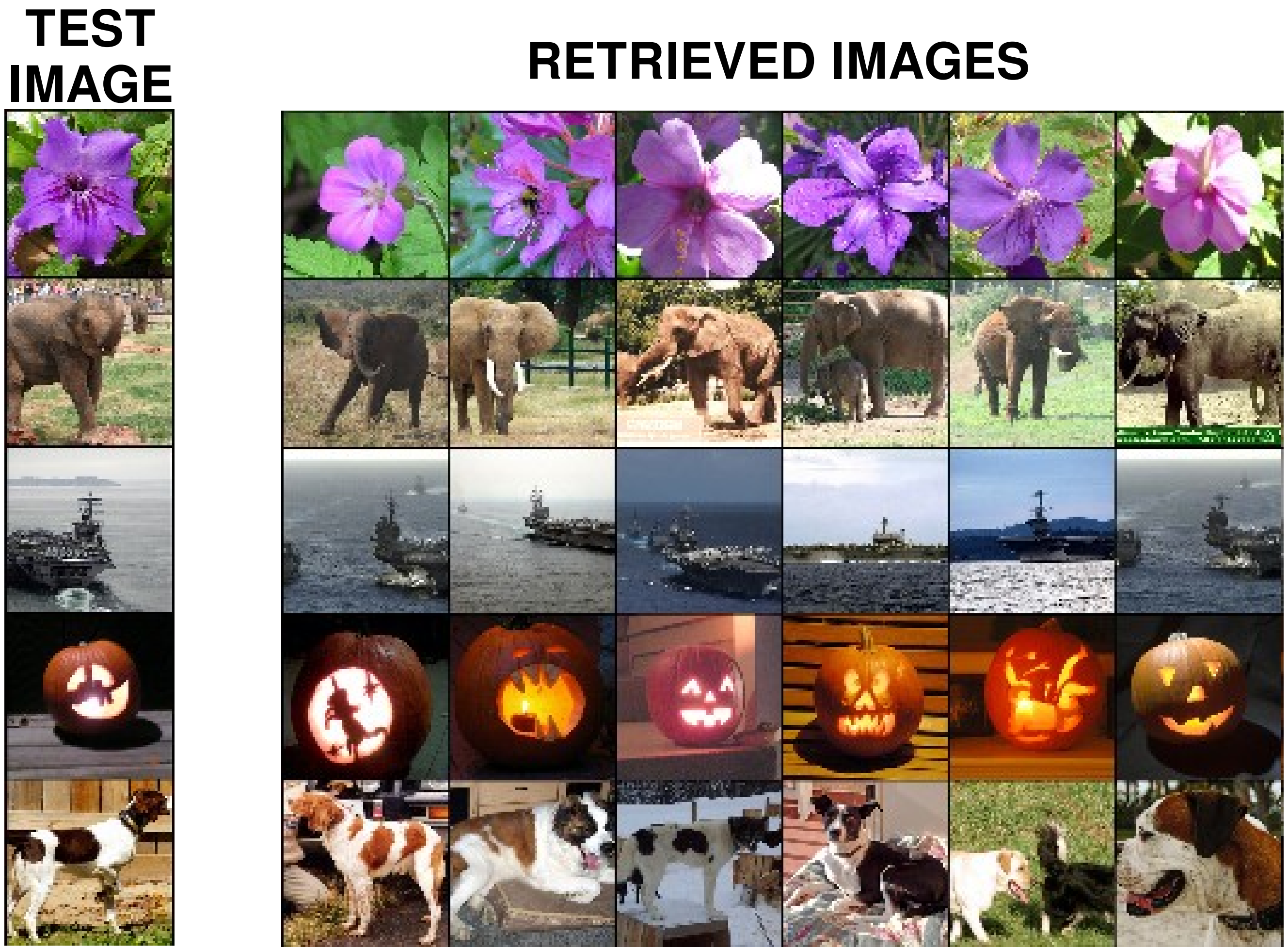


Results: 8 ILSVRC-2010 test images

- Results



Object recognition



[Krizhevsky, Sutskever and Hinton (2012)]

Image annotating

•

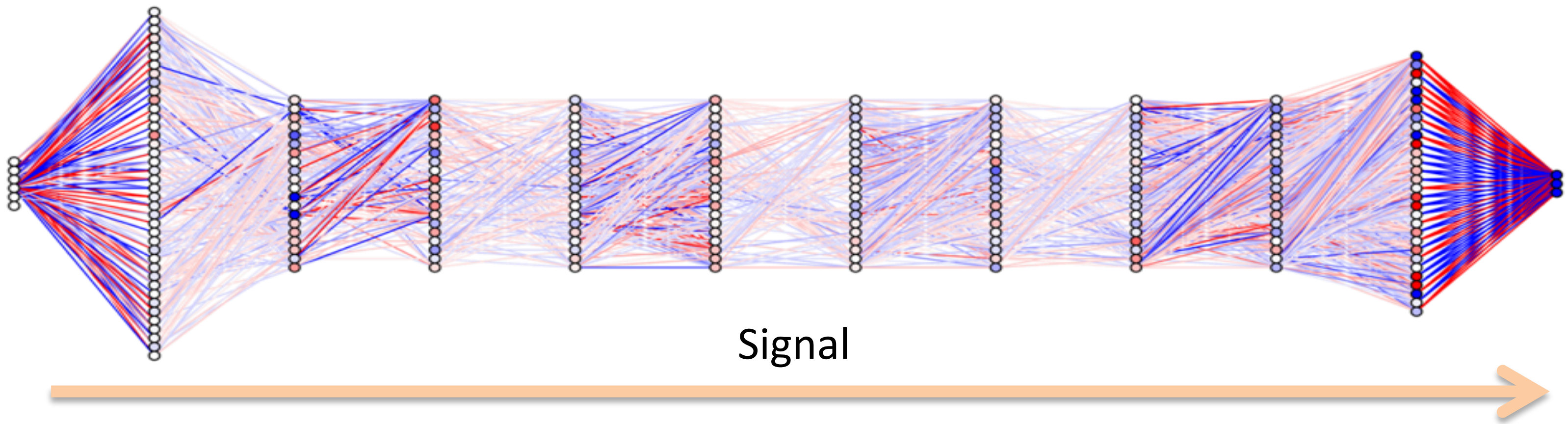
Figure 2.11: “A group of young people playing a game of frisbee”—that caption was written by a computer with no understanding of people, games or frisbees.

The SuperVision network

Image classification with deep convolutional neural networks

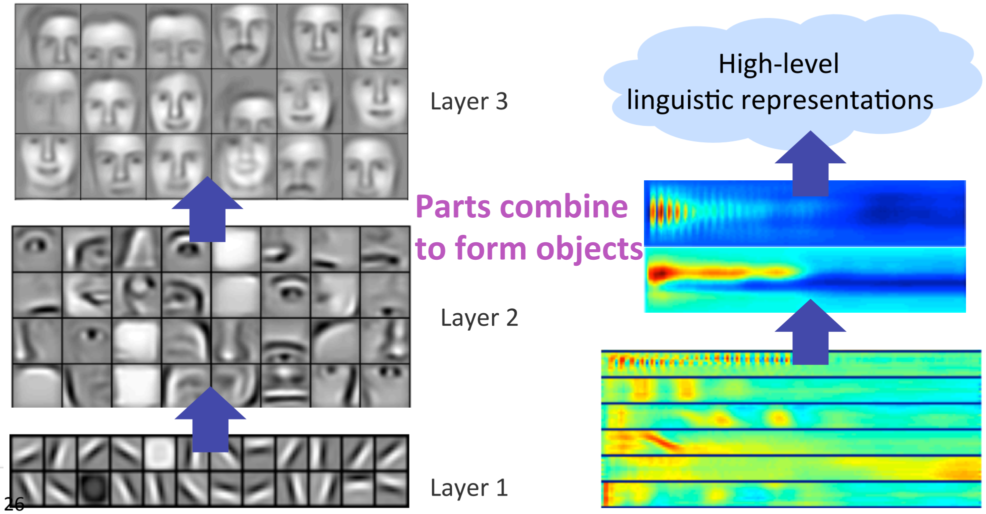
http://image-net.org/challenges/LSVRC/2012/supervision.pdf

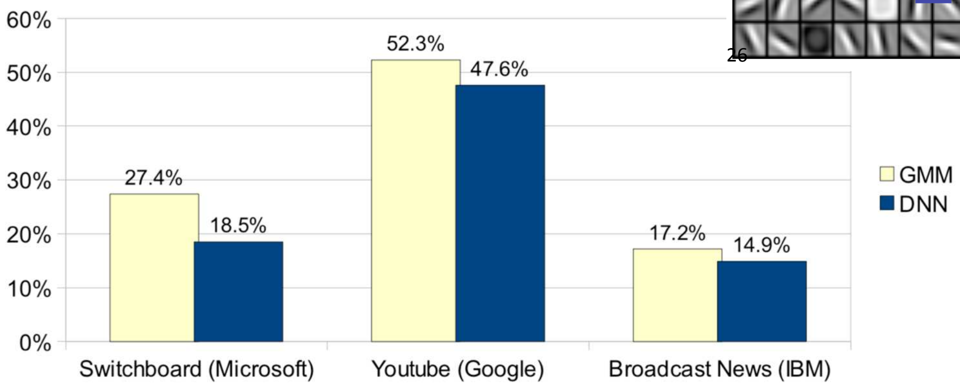
- 7 hidden “weight” layers
- 650K neurons
- **60M** parameters
- 630M connections



Speech recognition

- Works reasonably well

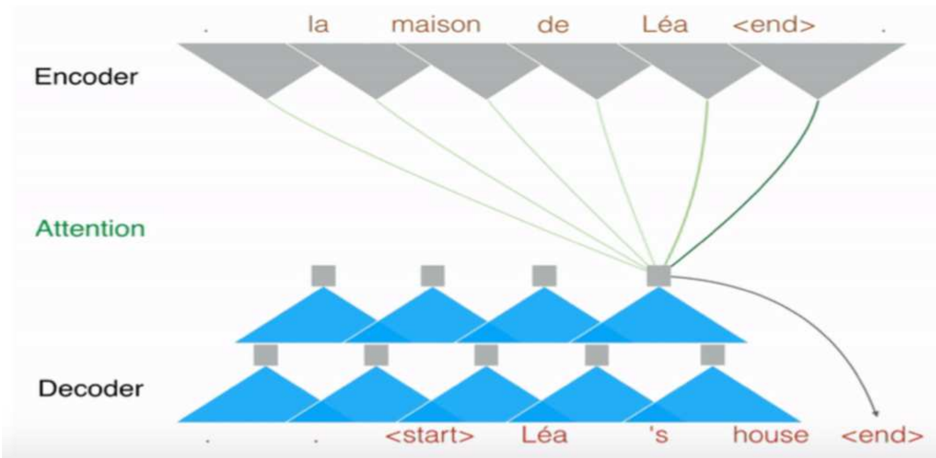




Comparison (2012) of the word error rates achieved by traditional GMMs and DNNs, reported by three different research groups on three different benchmark.

Machine translation

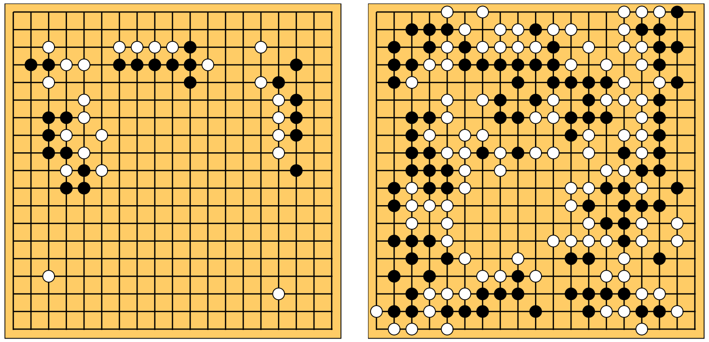
- Still far from perfect, but …

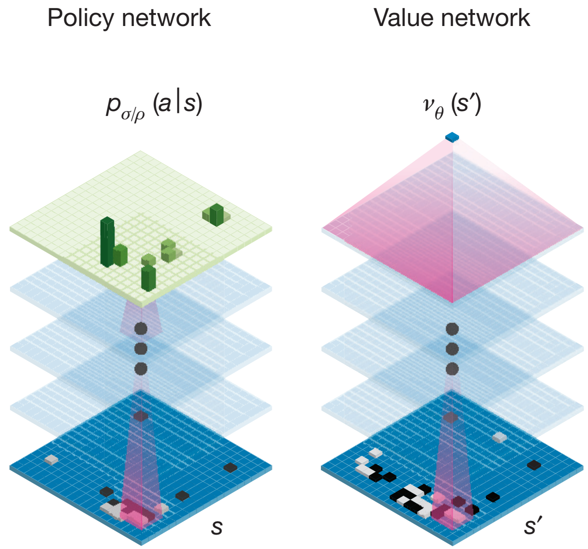


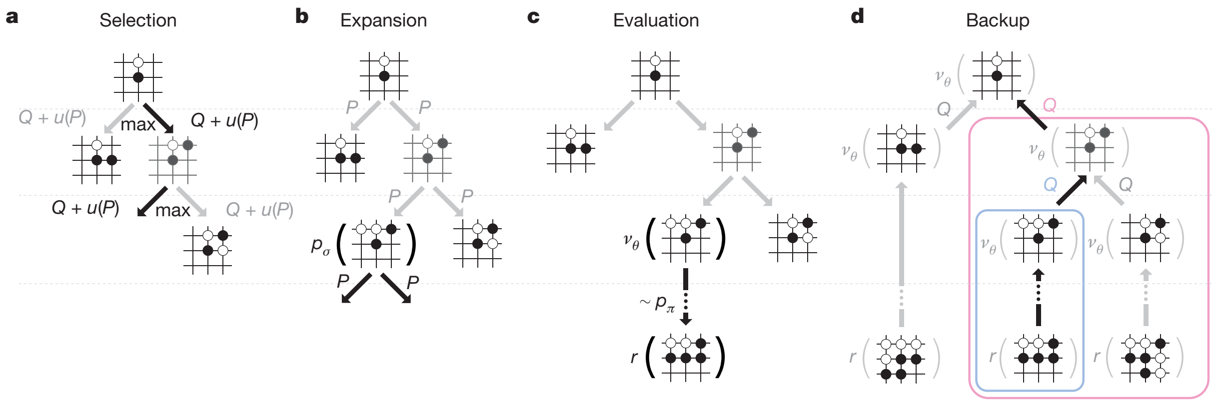
From Hofstädter (2018)

Game playing with Reinforcement Learning

- E.g. AlphaGo







Outline

1. What **does work**

2. **Limitations**

3. Learning comes with **which guarantees**?
   - Induction: how to win this game?
   - The statistical learning theory
   - A **closed case**? Not so sure

4. Other paradigms? An **historical perspective**

5. Is there **a paradigmatic change in sight**?

6. Conclusions

Limitations

Requires enormous training sets

- Image recognition
  - Object localization for 1000 categories.

    Millions of images

Requires enormous training sets

- Image recognition
  - Object localization for 1000 categories.

    Millions of images

- AlphaGo

Requires enormous training sets

- Image recognition
  - Object localization for 1000 categories.

    Millions of images

- AlphaGo
  - Training on **KGS dataset** led to overfitting
  - Self-play data (**30 million** distinct positions, each sampled from a separate game)

Requires enormous training sets

- Image recognition
  - Object localization for 1000 categories.

    Millions of images

- AlphaGo
  - Training on **KGS dataset** led to overfitting
  - Self-play data (**30 million** distinct positions, each sampled from a separate game)
  - Over the course of **millions of AlphaGo vs AlphaGo games**, the system progressively learned the game of Go from scratch, accumulating thousands of years of human knowledge during a period of just a few days. (In the first three days AlphaGo Zero played 4.9 million games against itself in quick succession.)

Exclusively focused on error rate

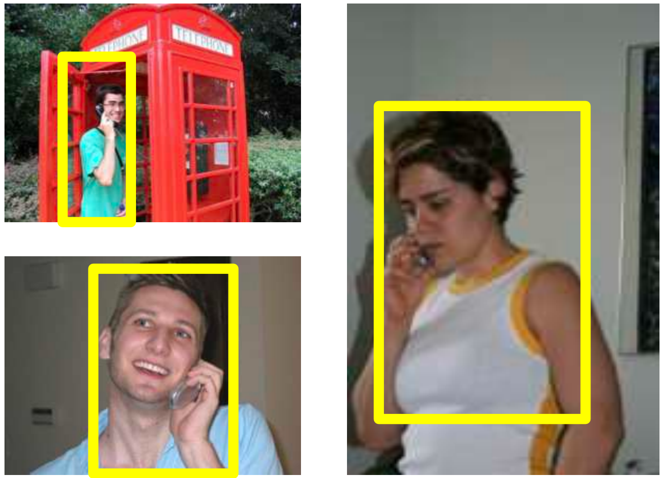
- The Netflix prize
  - The winner system was not used afterwards!!

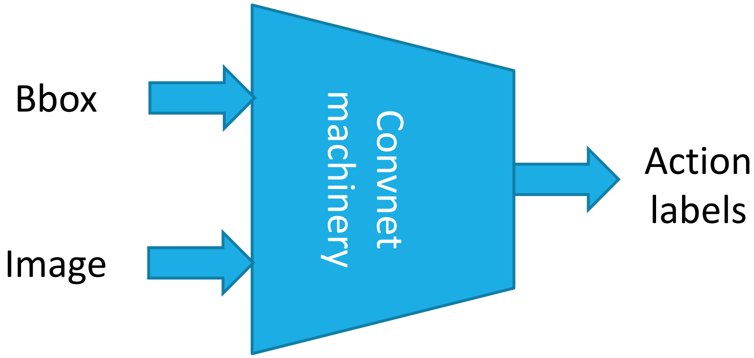
- Machine translation
  - Good on easy and mundane texts
  - Bad on interesting texts

Weak account of **the structure**

- **Texts** as *bags of words*
- **Images** as simple *correlations*

Example: detection of the action "*giving a phone call*"





[Oquab et al., CVPR (2014)] (~70% correct (SOTA))

Weak account of **the structure**

Example: detection of the action “*giving a phone call*”

Not giving a phone call.

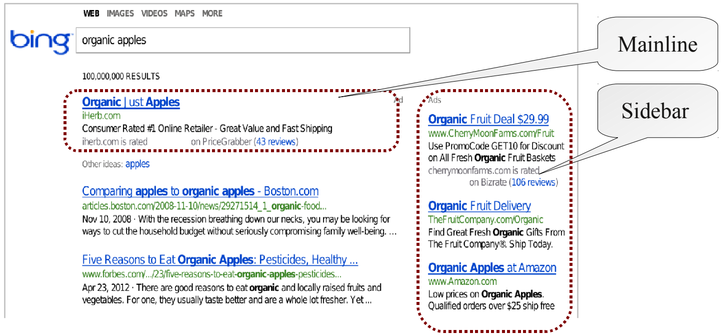
Giving a phone call ????

The learning algorithm is **statistically correct**!

In a typical image dataset, when an image shows a person near a phone (both in the same image), chances are that the person is giving a phone call

Learning systems do not work **together** flawlessly

- Two **sub-systems**
  - One locating the **ads links**
  - The other the **adds**



- That **influence each other**
  - Each takes into account the **clicks**
  - Which **depends** in part from the actions of the other sub-system
  - In addition of other **uncontrolled factors** (price, user's queries, …)

[L. Bottou et al. «*Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising* », JMLR, 14, (2013), 3207-3260]

The Simpson's paradox

- Physicians would like to know whether drug A is more or less efficient than drug B
- Two groups of 350 patients each are chosen. One is given drug A, and the other drug B

	Overall
Treatment A: Open surgery	78% (273/350)
Treatment B: Percutaneous nephrolithotomy	**83%** (289/350)

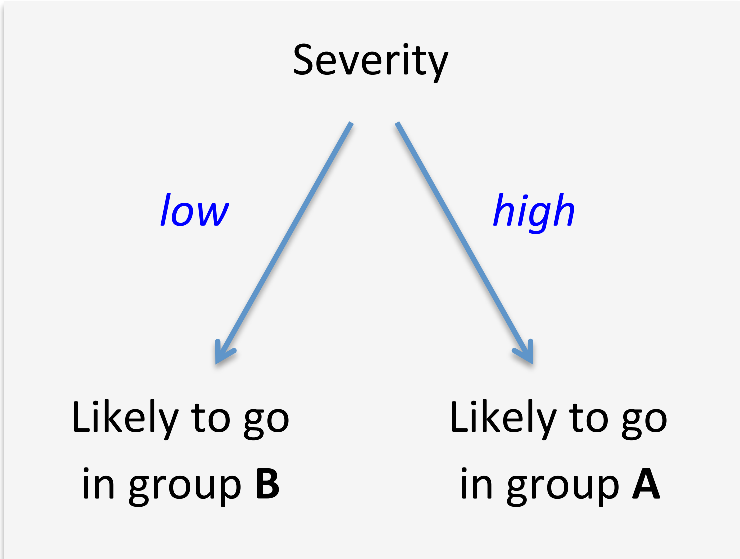
B is best?

The Simpson's paradox

	Overall	Patients with small stones	Patients with large stones
Treatment A: Open surgery	78% (273/350)	**93%** (81/87)	**73%** (192/263)
Treatment B: Percutaneous nephrolithotomy	**83%** (289/350)	87% (234/270)	69% (55/80)

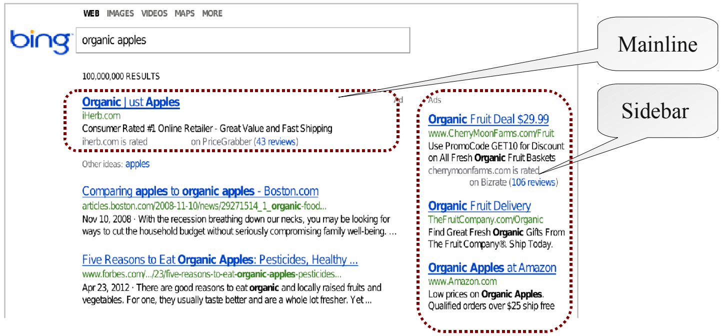
- Influencing factor

  The choice of the patients for each group was function of the severity of the pathology



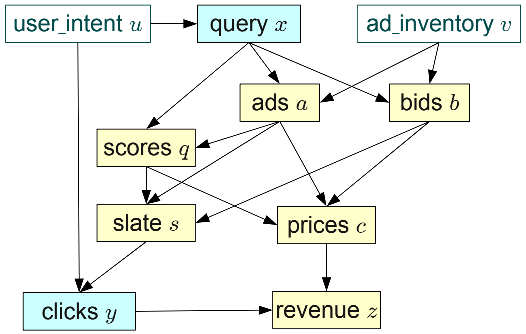
Learning systems do not work **together** flawlessly

- Two **sub-systems**
  - One locating the **ads links**
  - The other the **adds**



- That **influence each other**
  - Each takes into account the **clicks**
  - Which **depends** in part from the actions of the other sub-system
  - In addition of other **uncontrolled factors** (price, user's queries, …)

Importance of identifying the **causal graph**



[L. Bottou et al. «*Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising* », JMLR, 14, (2013), 3207-3260]

Thus, is the sky so blue?

Learning systems …

1. Require **enormous amounts** of training data

2. Are exclusively focused on **error rates**

3. Do not fully take advantage of **structures**

4. **Do not cooperate well**

   – **Software engineering** with **adaptive components** is **yet to be solved**

Outline

1. What **does work**

2. **Limitations**

3. Learning comes with **which guarantees**?
   - Induction: how to win this game?
   - The statistical learning theory
   - A **closed case**? Not so sure

4. Other paradigms? An **historical perspective**

5. Is there **a paradigmatic change in sight**?

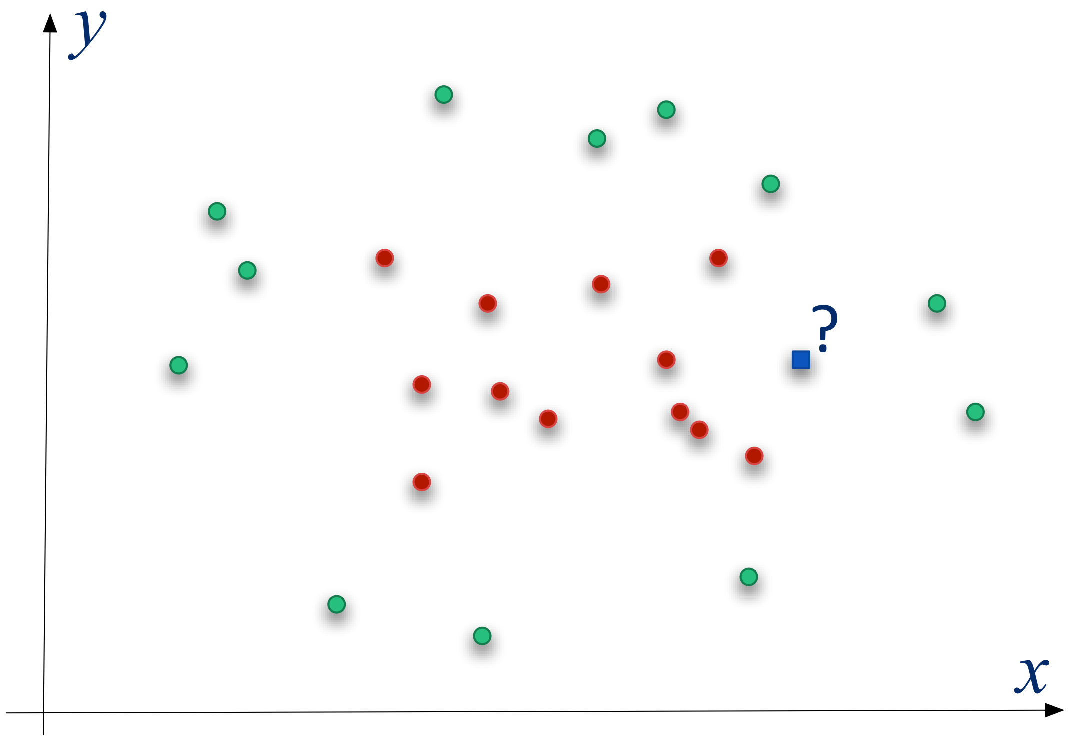
6. Conclusions

Which guarantees?

The **statistical theory** of learning

Supervised induction

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



→ A decision function

Supervised learning

Given a **training set**

$$\mathcal{S}_m \;=\; \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\overset{f}{\mathbf{x}_i}, \underset{h}{y_i}), \ldots, (\mathbf{x}_m, y_m)\}$$

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

- Hoping that it generalizes well :

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

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)

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	?	

15

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* ? $2^{2^2} = 2^4 = 16$

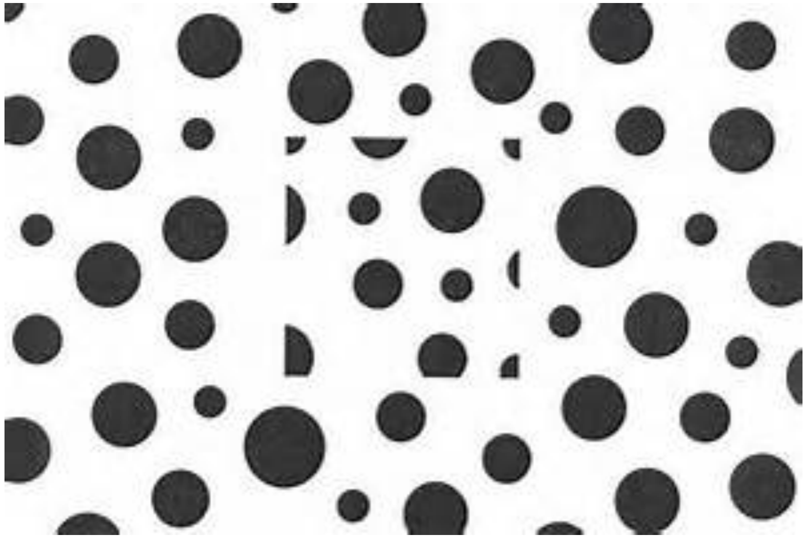
How many functions do remain after 3 ≠ training examples? $2^1 = 2$

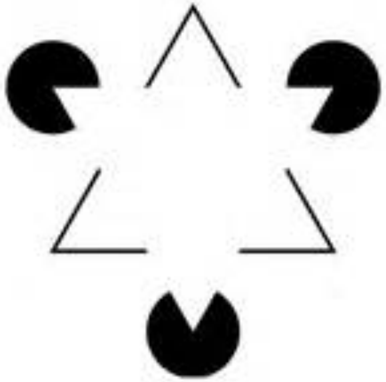
Induction: an impossible game?

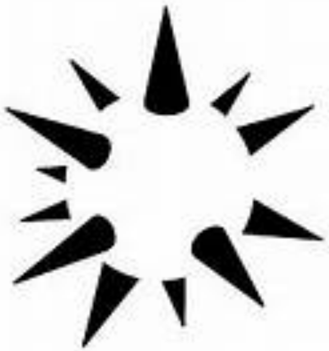
- **A bias is need**

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

Interpreting – completion of percepts

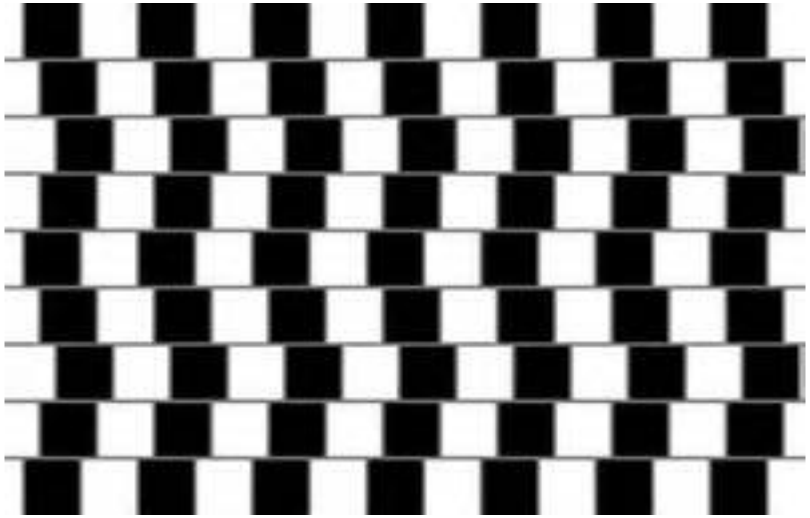




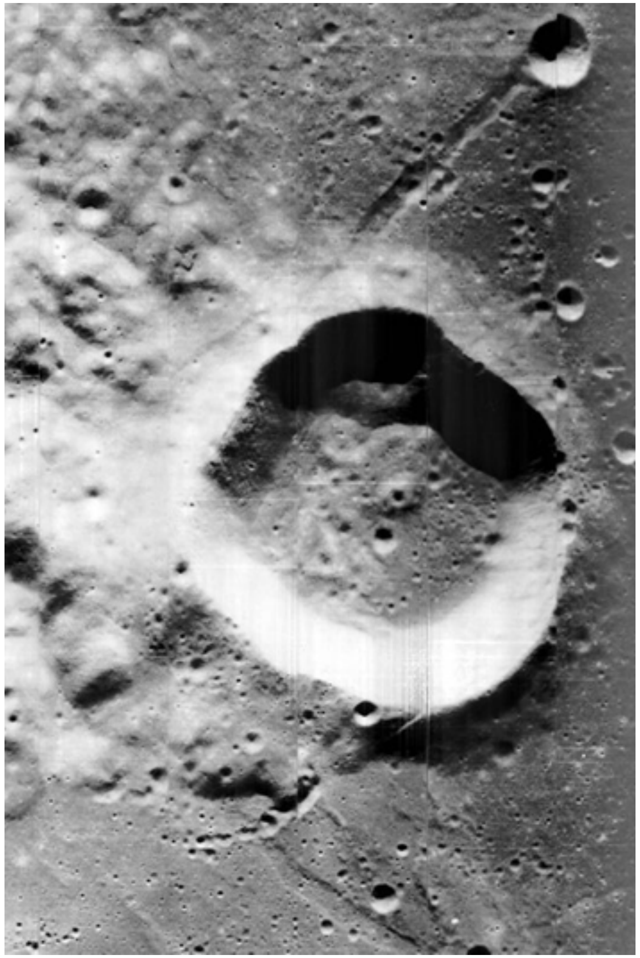


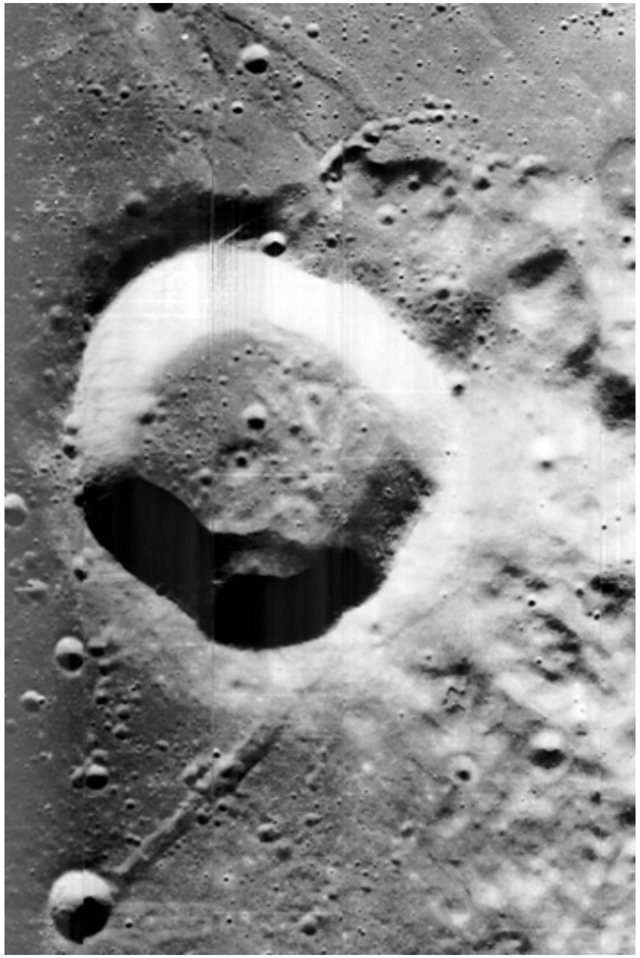
Interpreting – completion of percepts

Induction and its illusions



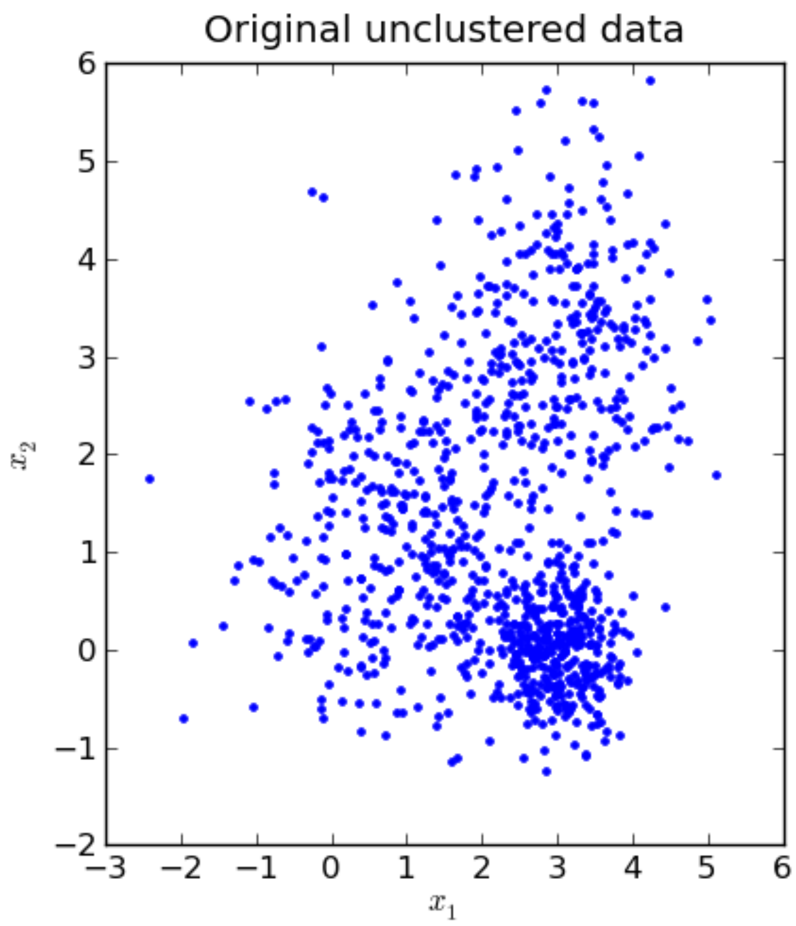
Induction and its illusions



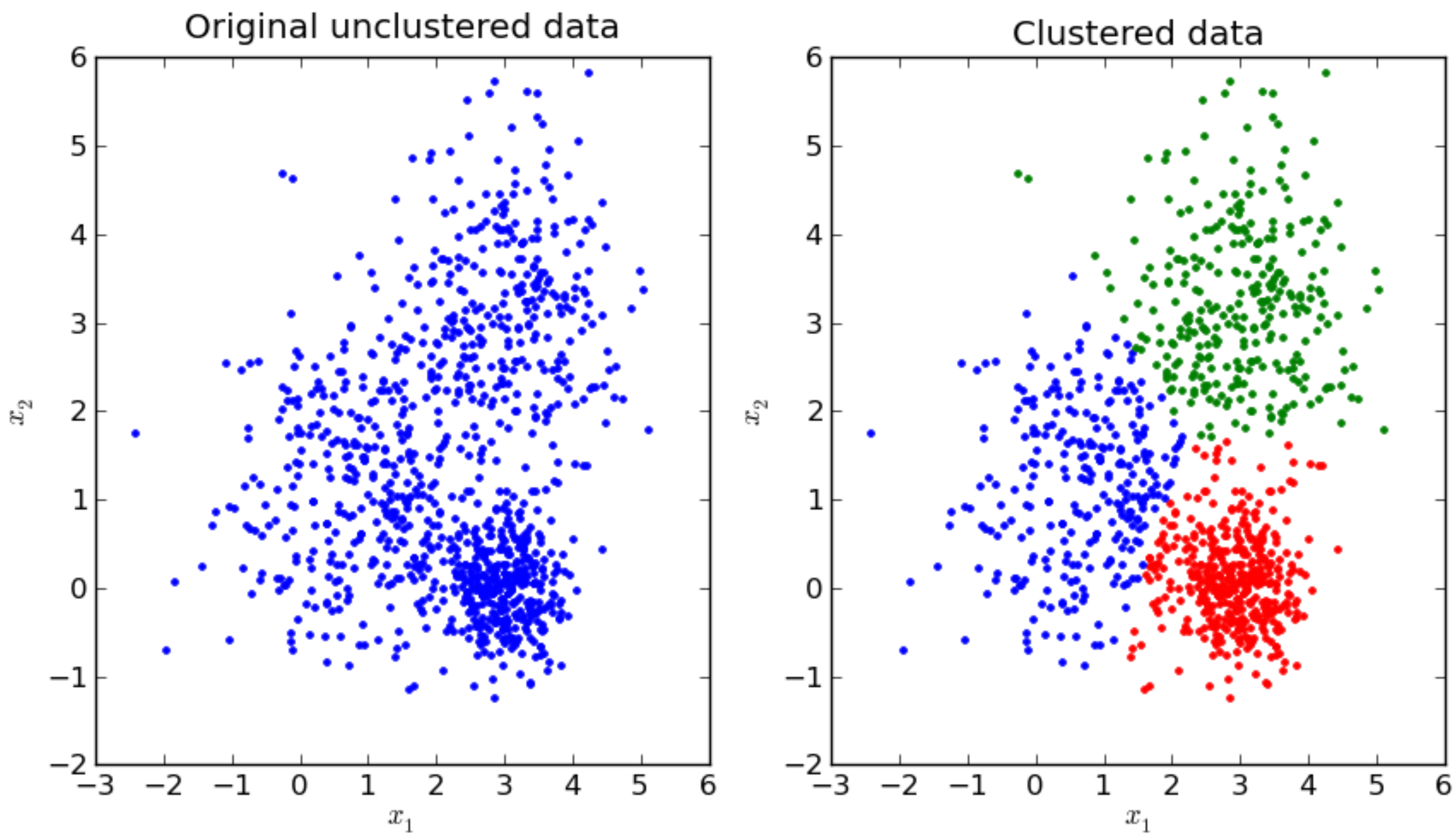


- toto

Clustering

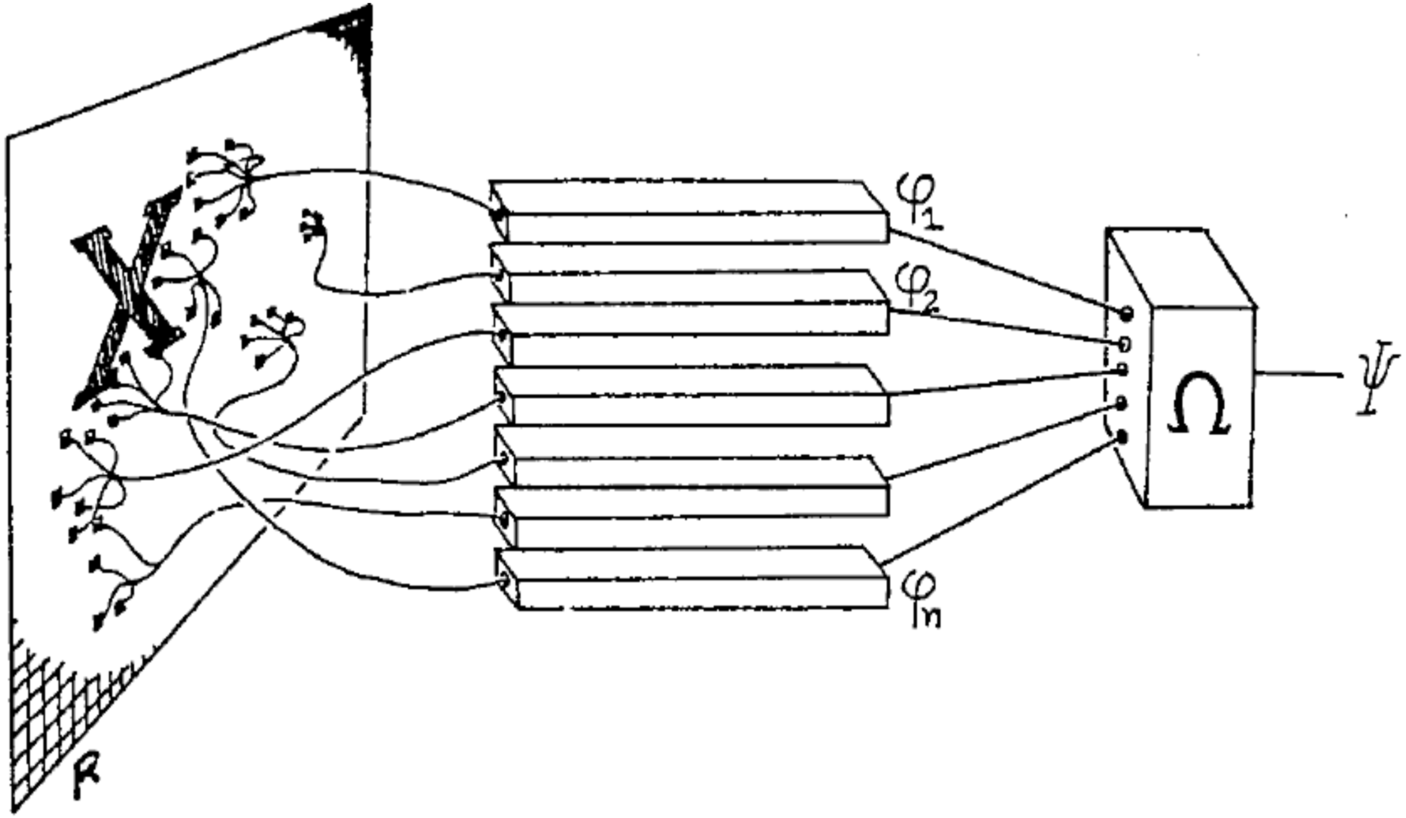


Clustering


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

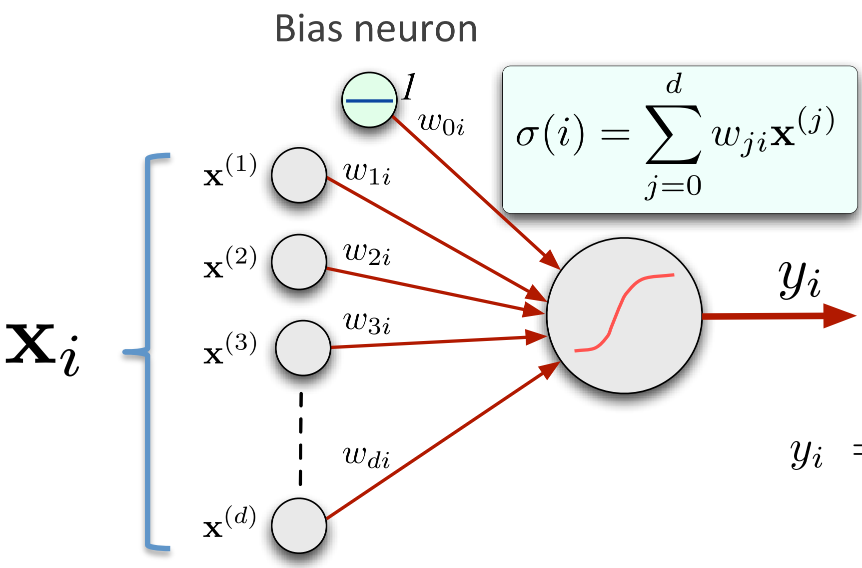
The perceptron

– Rosenblatt (1958-1962)



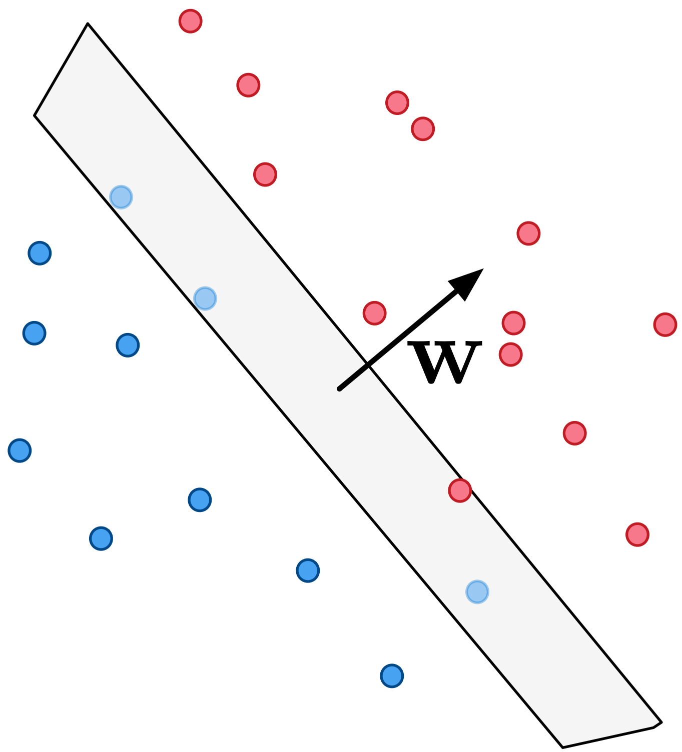
The perceptron

- Rosenblatt (1958-1962)



$$y_i = \text{sign}\left\{ g\left(\sum_{j=0}^{d} w_{ji}\, \mathbf{x}^{(j)} \right) \right\}$$

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

NO reasoning !!!

Some remarkable 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** of the input space

!!!

If there exists a linear separator of the training examples

The statistical theory of learning

Guarantees on generalization ??

- Theorems about the performance

  with respect to the **training set**

- We want guarantees about **future examples**

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 $\mathcal{H}$

The size *m* of the training set must be large enough w.r.t. to capacity 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 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

inventing **algorithms**

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

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

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

où $\mathrm{dis}_{\rho_{\mathbf{a}'}}(S_u, T_u) = \left| \mathbf{E}_{(h,h')\sim\rho_\mathbf{w}^2} \mathrm{R}_{S_u}(h,h') - \mathbf{E}_{(h,h')\sim\rho_\mathbf{w}^2} \mathrm{R}_{T_u}(h,h') \right|$ 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 :

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

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*)