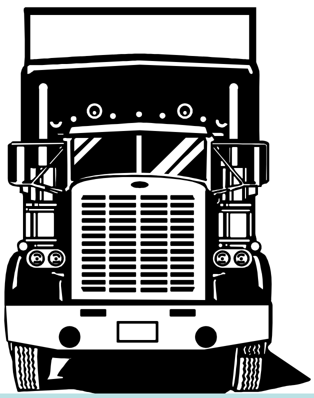
Statistical Approaches used in Machine Learning

B. Apolloni & D. Malchiodi
Neural Networks Laboratory
Dept. of Computer Science
University of Milano, Italy
http://laren.dsi.unimi.it/en

A Computational Learning tale

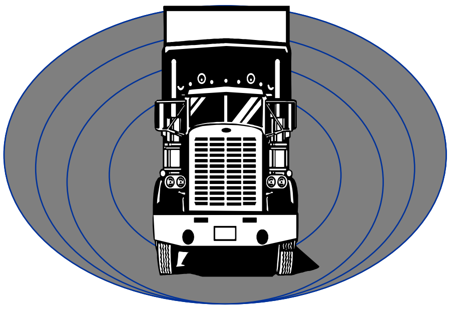


Task:
learning a Boolean function discriminating polluted from non polluted regions

A Computational Learning tale

A tank truck runs in a no inhabitants land

A Computational Learning tale

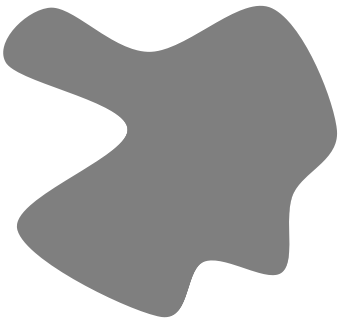


At a certain point the tank breaks and a polluting fluid spreads on the ground

A Computational Learning tale

The driver gives a look on the left, then on the right: as no people can see him, he restarts the engine and goes away fast….

A Computational Learning tale



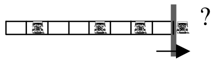
The Major of the neighbouring city wants to discover where is located and how is extended the polluted region

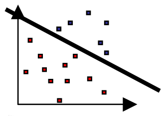
LEARNING!!

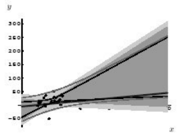
LEARNING!!
LEARNING!!

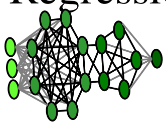
Outline

1. Statistical basics
   - Algorithmic inference
   - Inferring a Boolean variable
   - Learning a Boolean function
2. Learning tools
   - Symbolic →Boolean: Decision trees, SVM
     →Continuous: Linear Regression
   - Non symbolic → Neural Networks
     → Genetic Algorithms







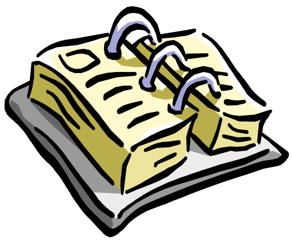


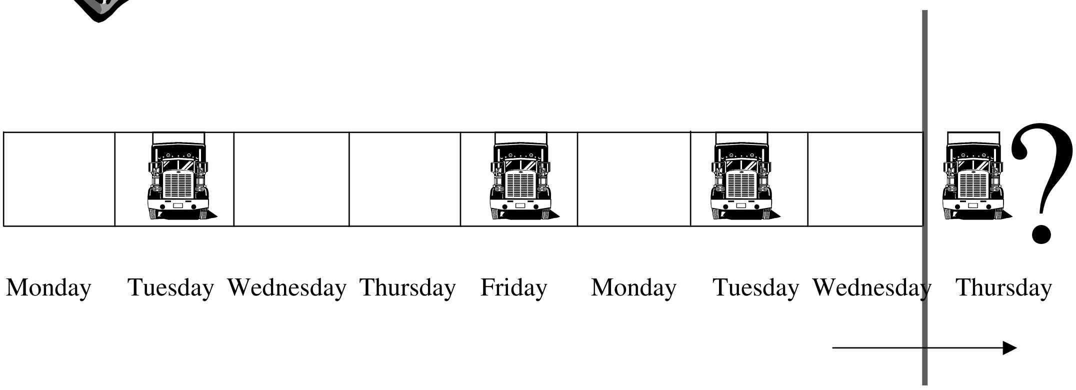
TATTAGATATTTTCTTTATTTACATTTCAAA
TATTAGATATTTTCTTTATTTACATTTCAAA

Outline

1. Statistical basics
   - Algorithmic inference
   - Inferring a Boolean variable
   - Learning a Boolean function
2. Learning tools
   - Symbolic →Boolean: Decision trees, SVM
     →Continuous: Linear Regression
   - Non symbolic → Neural Networks
     → Genetic Algorithms

First step: did the truck pass yesterday?

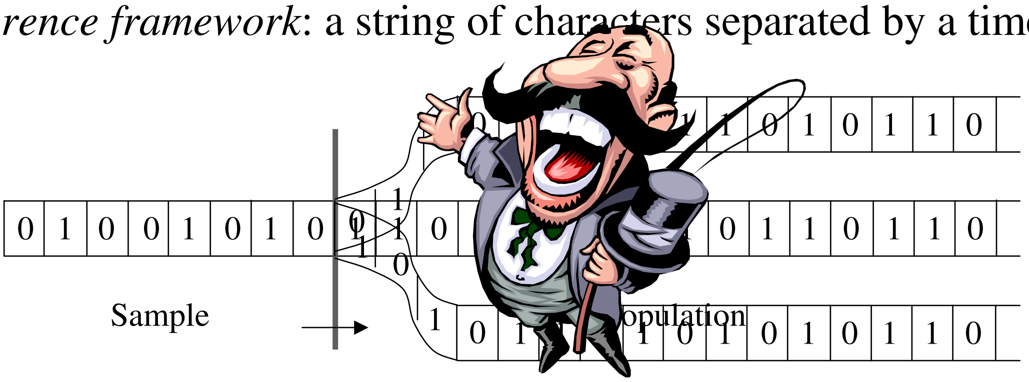




A predictive approach

Inference: a way of well organizing the observed data

Inference framework: a string of characters separated by a time pointer



- On the basis of what we have observed we want to predict properties on what we will observe in the future [Laplace 1868, Fisher 1948, De Finetti 1975, Geisser 1993]

Why predictive inference?

→ Kolmogorov framework born when:

•Computing was costly → take sum and divide by m

•Collecting data was difficult → manage ten to hundred data

•Philosophy was still Aristotelic → God tosses dice

→ Now we can:

•Commonly make complex computations

•Automatically gather a lot of data

•Mind at computable functions underlying structured data

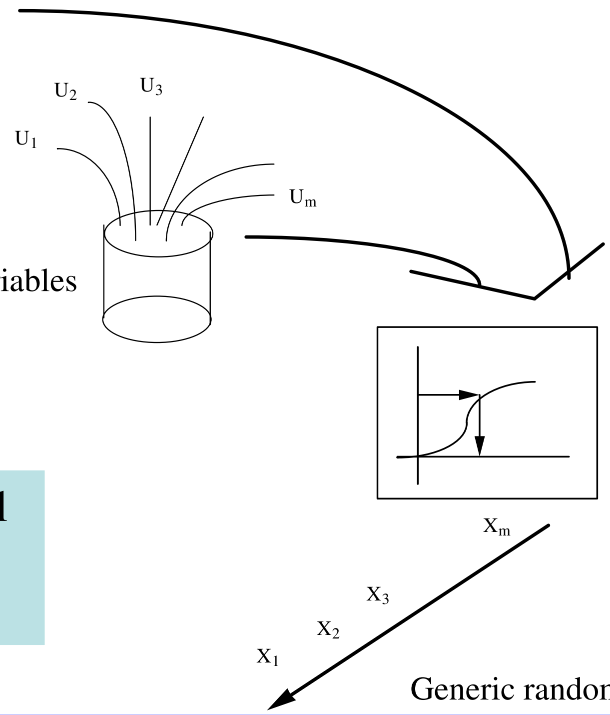
LEARNING!!

LEARNING!!

LEARNING!!

The mother of all samples

Uniform random variables



The integral transform theorem

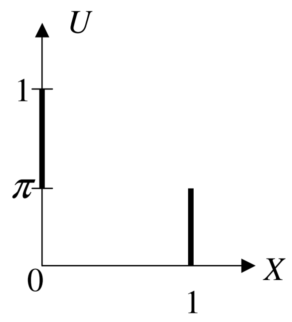
Generic random variables

Example: a Bernoulli variable

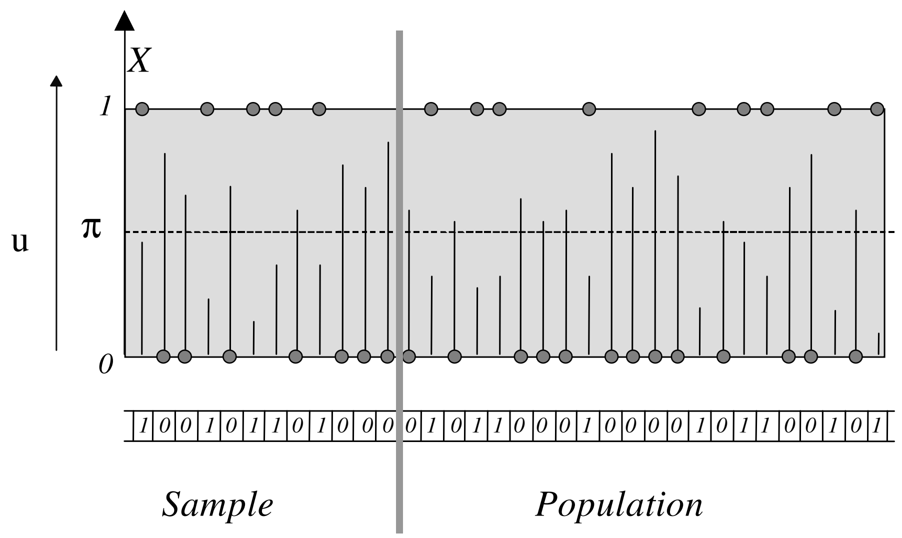
$$X = \begin{cases} 1 & P(1) = \pi \\ 0 & P(0) = 1 - \pi \end{cases}$$

•A Bernoulli random variable *X* of mean *p* can be described through a [0,1] uniform distribution *U* coupled with the transformation

$$X = \begin{cases} 1 & \text{if } U \le \pi \\ 0 & \text{elsewhere} \end{cases}$$

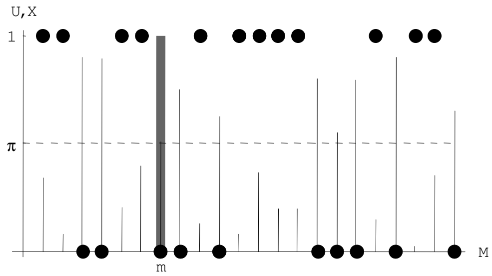


A suitable interpretation



- We don't assume anything, apart the fact that we are observing a same phenomenon

An inference problem



How is π large?

π **is**:

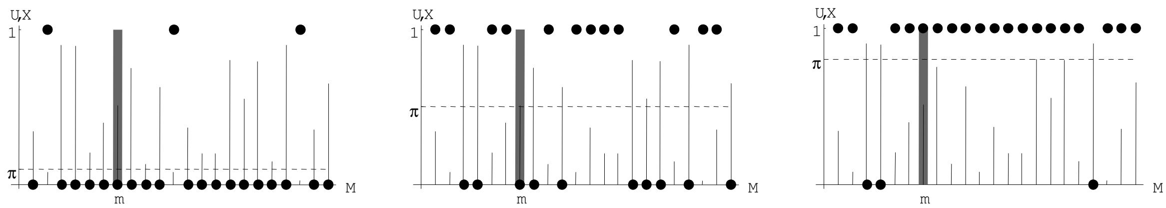
- the threshold of the sample generation mechanism

  then must be argued by the sample

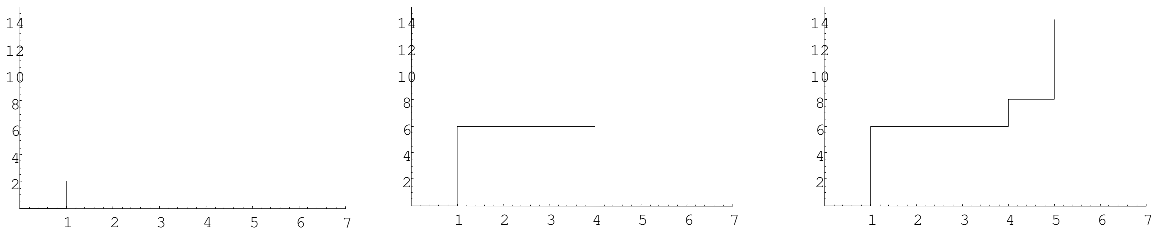
- the asymptotic frequency of 1's in the sample suffix

  then is a r.v. specification

A twisting argument



X axis: sample realizations indices; Y axis:realizations of U (lines) and X (bullets)



X axis: # 1's in sample; Y axis: # 1's in population

A twisting argument

Denoting: k the number of observed 1's
k_π the value for k if $\Pi = \pi$

Logical implication $(k_\pi \geq k) \Leftarrow (p < \pi) \Leftarrow (k_\pi \geq k+1)$

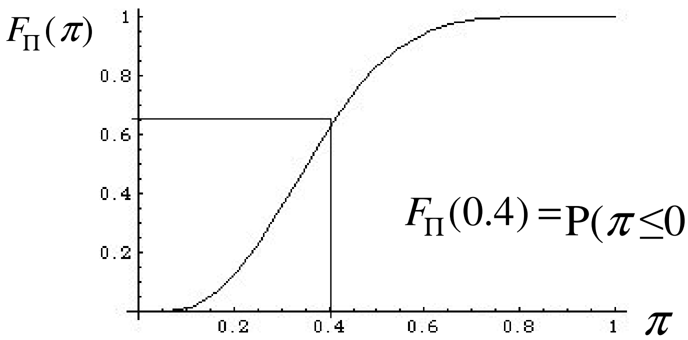
Events inclusion $(K_\pi \geq k) \supseteq (\Pi < \pi) \supseteq (K_\pi \geq k+1)$

Probability inequality $\mathrm{P}(K_\pi \geq k) \geq \mathrm{P}(\Pi < \pi) \geq \mathrm{P}(K_\pi \geq k+1)$

Note: we need one sample point to expressely recognize that $\pi > \Pi$

Distribution law of Π

$$1 - F_{K_\pi}(k-1) \geq F_\Pi(\pi) \geq 1 - F_{K_\pi}(k)$$



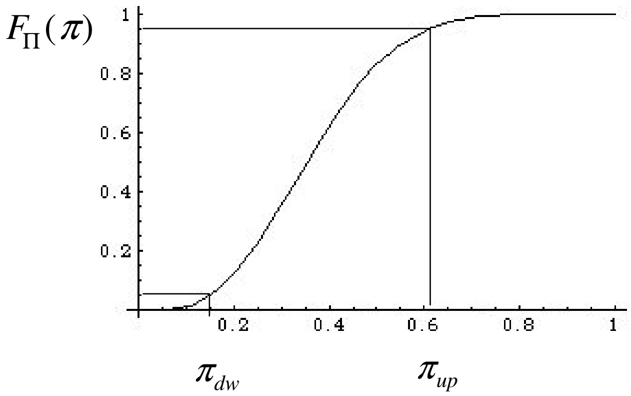
$F_\Pi(0.4) = \mathrm{P}(\pi \leq 0.4) = 0.618$

Confidence interval for Π

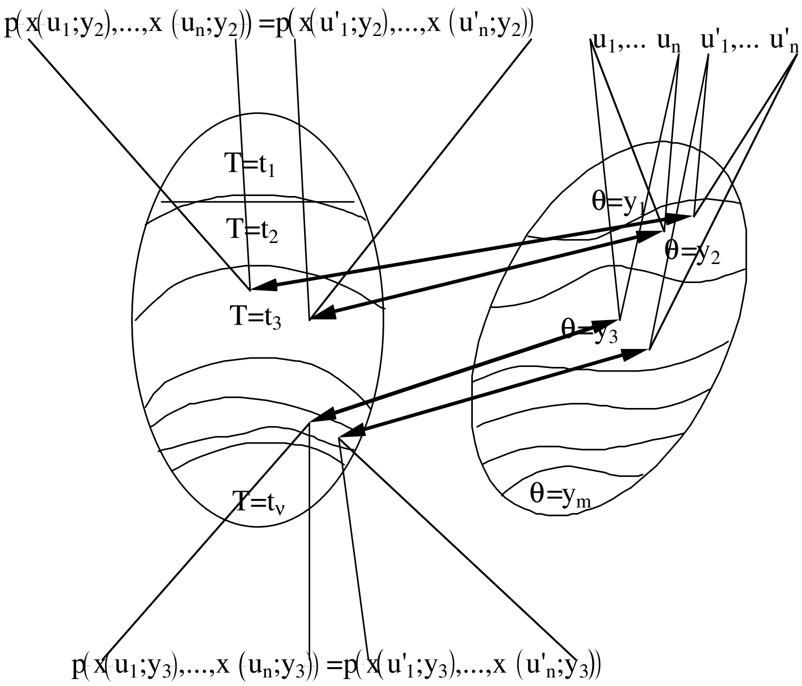
$$P(\pi_{dw} < \Pi < \pi_{up}) = 1 - \delta$$

$$P(\pi_{dw} < \Pi < \pi_{up}) = F_\Pi(\pi_{up}) - F_\Pi(\pi_{dw})$$

Tomorrow I will come at the conference btween 9 and 9.30 a.m.

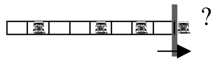


Favorite pivots: sufficient statistics



A statistic T=g($X_1,\ldots X_m$), inducing on X^m the partition $\prod$ is *sufficient* with reference to the parameter θ of X if the ratio between the density functions of two samples does not depend on θ when the samples belong to a same element of $\prod$. A sufficient statistic is minimal if whenever two samples are have the same probability, they must belong to a same element of $\prod$ [Zacks 1971].

Outline

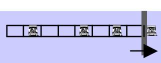
1. Statistical basics
   - Algorithmic inference
   - Inferring a Boolean variable 
   - Learning a Boolean function
2. Learning tools
   - Symbolic →Boolean: Decision trees, SVM
     →Continuous: Linear Regression
   - Non symbolic → Neural Networks
     → Genetic Algorithms

Solving the inverse problem

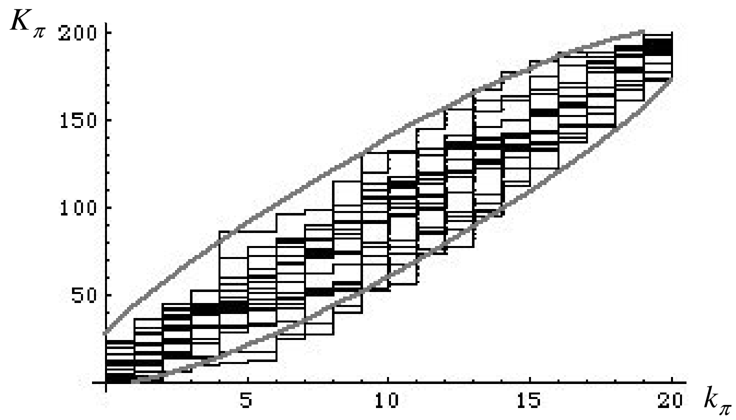
$$\mathrm{P}\left(\pi_{dw} < \Pi < \pi_{up}\right) = F_{\Pi}(\pi_{up}) - F_{\Pi}(\pi_{dw}) \geq$$

$$\geq \sum_{i=k+1}^{m} \binom{m}{i} \pi_{up}^{i} (1-\pi_{up})^{m-i} - \sum_{i=k}^{m} \binom{m}{i} \pi_{dw}^{i} (1-\pi_{dw})^{m-i}$$

$$\begin{cases} \sum_{i=k+1}^{m} \binom{m}{i} \pi_{up}^{i} (1-\pi_{up})^{m-i} = 1 - \delta/2 \\ \sum_{i=k}^{m} \binom{m}{i} \pi_{dw}^{i} (1-\pi_{dw})^{m-i} = \delta/2 \end{cases}$$



An experiment: 0.9 confidence region



The trajectories resume 10 simulations of the process with sample and population of 20 and 200 elements, respectively.

Point estimator

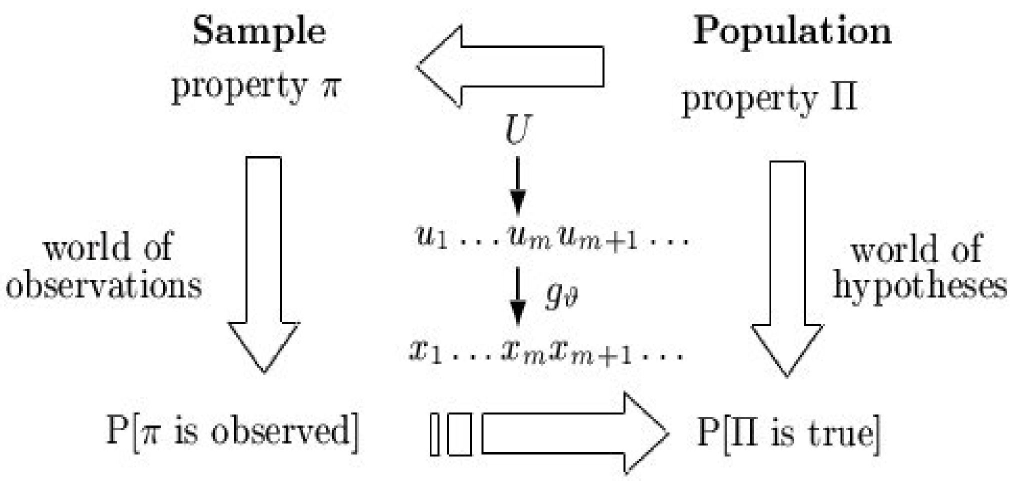
$$\min_a \left[\lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} (x_i - a)^2 \right] = \lim_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} x_i \equiv E[X]$$

$$E[X] = -\int_{-\infty}^{0} F[x]dx + \int_{0}^{+\infty} (1 - F[x])dx$$

$$\frac{k}{m+1} \leq E[\Pi] \leq \frac{k+1}{m+1}$$

the Laplace rule of succession

Algorithmic inference



A general issue for twisting arguments

$$(T_\pi \geq t) \Leftarrow (\Pi \leq \pi) \Leftarrow (T_\pi \geq t + \mu)$$

Where

- T is a statistic on the observed sample
  (**possibly a minimal sufficient statistic**),
- t is its corresponding realization,
- μ is a complexity index (*detail*) for the considered class of problems

Example: an exponential variable

$$F(x) = 1 - e^{-\lambda x}$$

Explaining function

$$x_i = \frac{-\log(1-u_i)}{\lambda}$$

$$s_\lambda = \sum_{i=1}^{m} x_i$$

- is a statistic (a function of the sole observed data)
- is monotone with respect to λ.

$$(\Lambda \leq \lambda) \Leftrightarrow (s \geq S_\lambda)$$

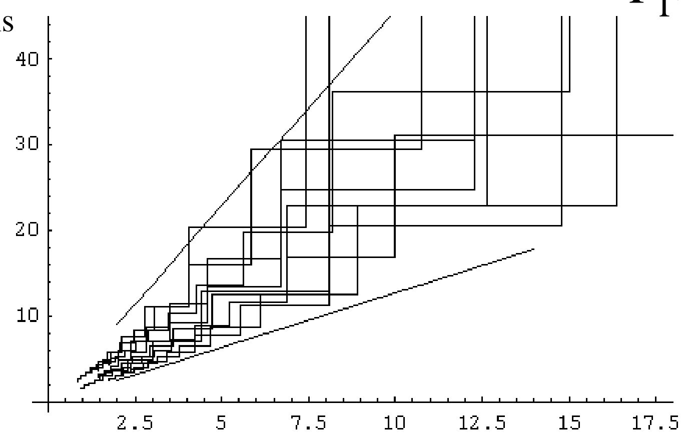
$$P(\Lambda \leq \lambda) = P(S_\lambda \leq s)$$

Confidence interval

$$F_{\Gamma(m,\lambda_{up})}(s) = 1 - \gamma/2$$

$$F_{\Gamma(m,\lambda_{dw})}(s) = \gamma/2$$

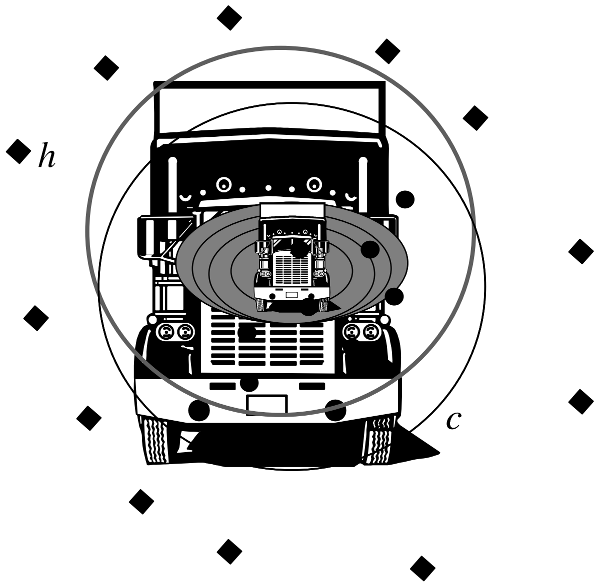
Values for S
in populations



Values for S
in samples

0.9 confidence intervals for *S*

Reviewing the tale

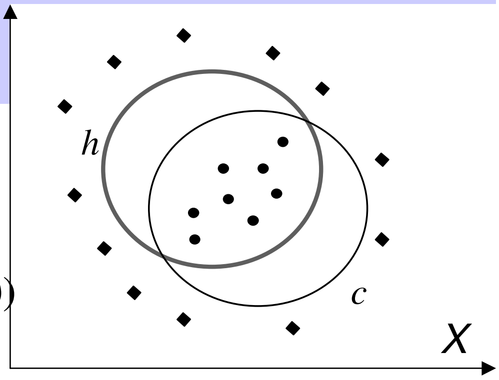


Outline

1. Statistical basics
   - Algorithmic inference
   - Inferring a Boolean variable
   - Learning a Boolean function

2. Learning tools
   - Symbolic →Boolean: Decision trees, SVM
     →Continuous: Linear Regression
   - Non symbolic → Neural Networks
     → Genetic Algorithms

The PAC model



[Valiant 1984] For a fixed space X, consider

- a concept class C on X (i.e. a set $C \subseteq \wp(X)$)
- a labelled sample $z_m = \{x_i, c(x_i)\}$, drawn from X with an uknown P and labelled according to a $c \in C$

A function $A: \{z_m\} \Rightarrow \wp(X)$ is a learning algorithm for C if for each $\varepsilon, \delta \in (0,1)$ there exists $m_0 \in \mathrm{N}$ such that, denoted $h = A(z_m)$, for a generic sample with $m > m_0$

$$P[\mathbf{E}[h \div c] < \varepsilon] > 1 - \delta$$

Err(C)=$\mathbf{E}[h \div c]$ is the property we want to estimate.

Framing into Algorithmic Inference

Assume that,

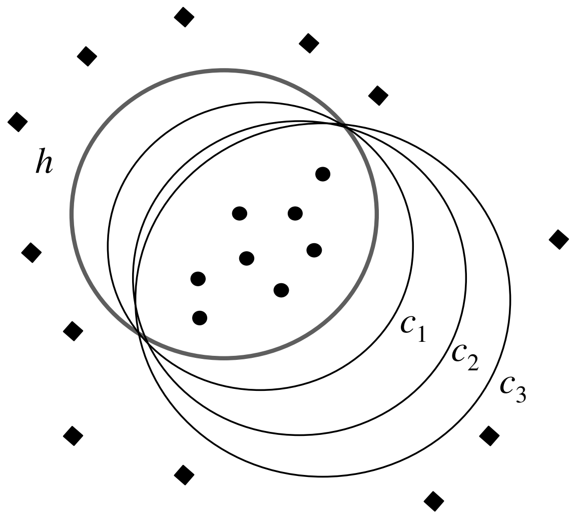
- starting from a labeled sample $z_m = \{x_i, b_i\}$
- for every suffix z_M a c exists such that

$$z_{M+m} = \{x_i, c(x_i)\},$$

Then, we are interested in the symmetric difference between an h=A(z_m) and any such c. Denote U its random measure

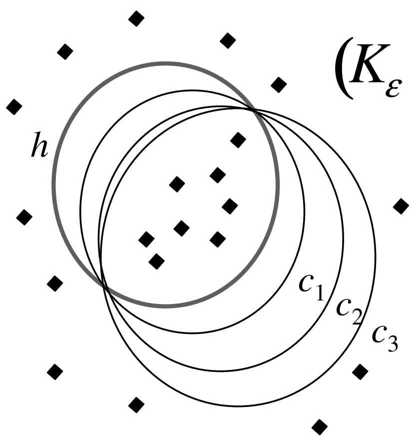
A family of hypotheses for a family of suffixes

the circle h describes the sample and possible circles c_i describe the population.



• positive points, ◆ negative points

Twisting argument



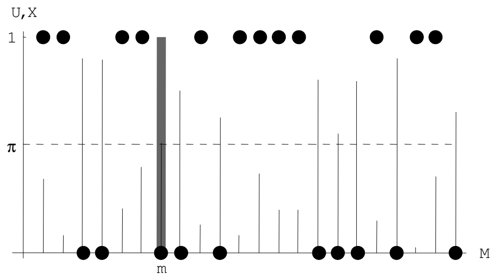
$$(K_\varepsilon \geq k+1) \Leftarrow (U_{c \div h} \leq \varepsilon) \Leftarrow (K_\varepsilon \geq k+\mu)$$

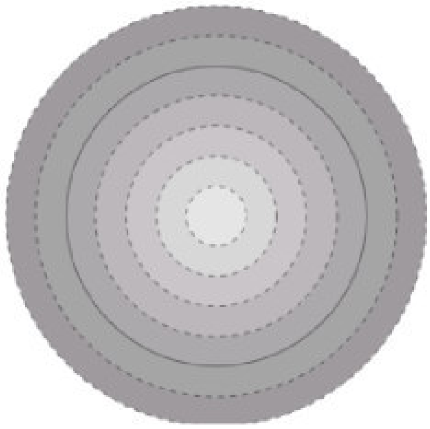
Problem:

Who tells me that

$(U_{c \div h} \leq \varepsilon)$?

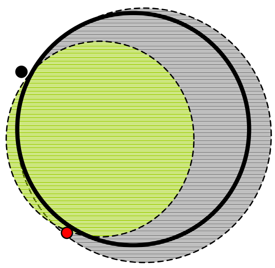
Where is the difference?





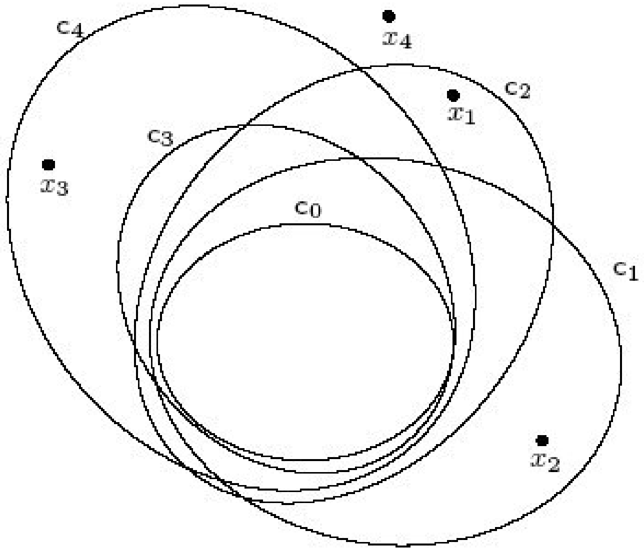
No any if we know the center of the circles, otherwise we need more expansion whitnesses

Sentry points



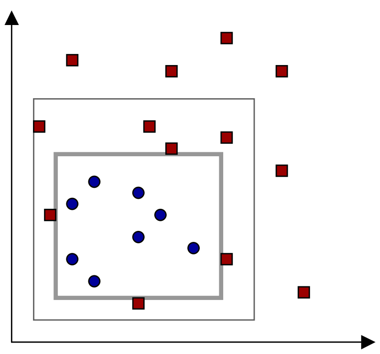
- Sentinels are outside the sentineled concept
- Sentinels are inside the invading concept
- The sentinels set is minimal
- Sentinels are honest watchers

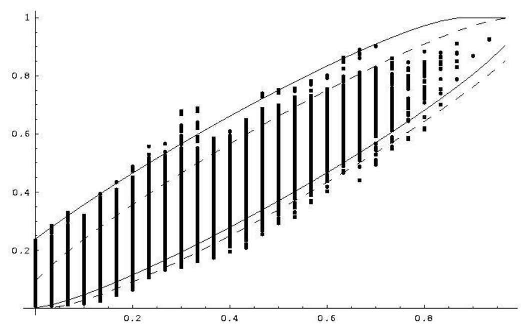
For a general concept class



Example: Bounding a rectangle in R^n

$$C = \{[a,b]\times[c,d],\ a,b,c,d \in \mathrm{R}\} \subseteq \mathrm{R}^2 \Rightarrow \mu = 4$$

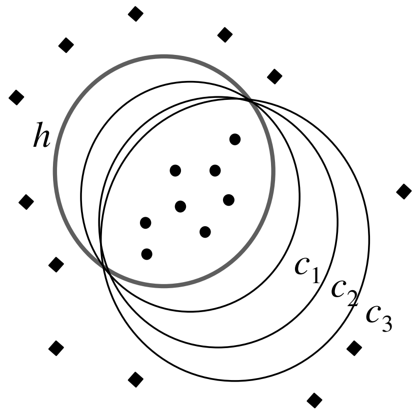




X= R^2, points and squares: labelled sample.

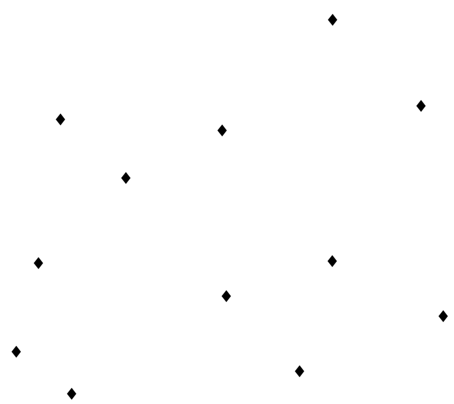
X and Y axis: percentage of points in Err(C) for sample and population, respectively.

Learning a concept



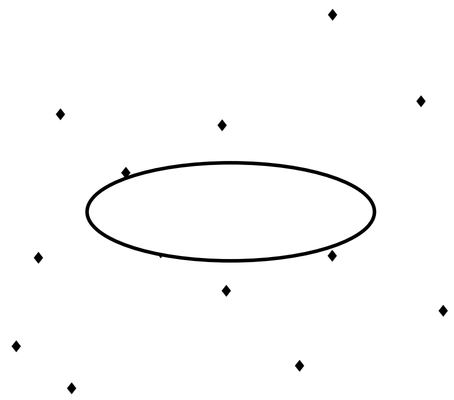
All sampled points are outside the symmetric difference

Learning a concept



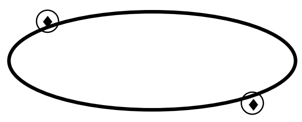
Starting from $\mathbf{S}_m$

Learning a concept



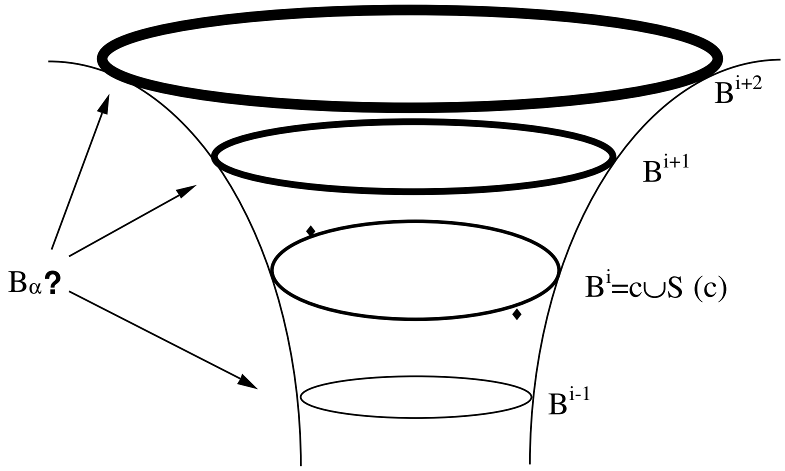
let us draw a consistent (maximal) hypothesis

Learning a concept



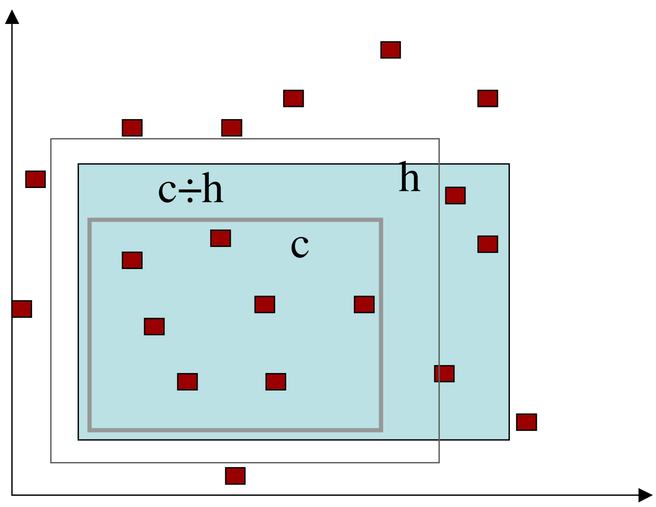
let us isolate the sentry points

is c ∪ S(c) included in B_α?



$$(U_{c\div h} \leq \alpha) \Leftarrow (c\div h \subseteq B_\alpha) \Leftarrow (S(c\div h) \subseteq B_\alpha) \Leftarrow (K_\alpha \geq \#S(c\div h))$$

Example: Learning a rectangle in R^n



Definition: The detail of C is the maximum number of points needed to sentinel a generic c∈C:

$$D_C = \sup_{S,c} \#S(c)$$

Theorem: If $D_{C,C}=\mu$, and h is an hypothesis misclassifying at least t' and at most t points of total probability not greater than π, then for each $\beta \in (\pi,1)$

$$I_\beta(1+t', m-t') \geq \mathrm{P}(U_{c\div h} \leq \beta) \geq I_\beta(\mu+t, m-(\mu+t)+1) \quad (°)$$

where

$$I_\beta(\mu+t, m-(\mu+t)-1) = 1 - \sum_{i=0}^{\mu+t-1} \binom{m}{i} \beta^i (1-\beta)^{m-i}$$

is the incomplete beta function.

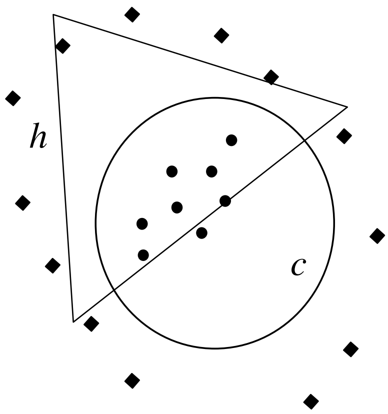
A first corollary: sample complexity

$$I_\varepsilon(\mu+1, m-\mu) = 1-\delta$$

Given a concept class C with detail μ, every consistent function A is a learning algorithm such that (°) is verified if

$$m \geq \max\left\{\frac{2}{\varepsilon}\mathrm{Log}\frac{1}{\delta}, \frac{5.5(\mu-1)}{\varepsilon}\right\} \qquad (*)$$

Usual extension: H≠C



If A is allowed to misclassify at most t points, (*) is verified if we replace μ with $\mu + t$.

$$I_\varepsilon(\mu+t, m-(\mu+t)+1) = 1-\delta$$

$$m \geq \max\left\{\frac{2}{\varepsilon}\mathrm{Log}\frac{1}{\delta}, \frac{5.5(\mu+t-1)}{\varepsilon}\right\}$$

Working on the single hypothesis

$\left(D_{C\div H}\right)_h$ = the worst case number of sentry points for sentinelling $c\div h$ against expansions due to whatever $h'\in H$ for whatever explanation $c\in C$

t_h= the numberof sample points actually mislabeled by h

$$m \geq \max\left\{\frac{2}{\varepsilon}\mathrm{Log}\frac{1}{\delta}, \frac{5.5\left(\left(D_{C\div H}\right)_h + t_h - 1\right)}{\varepsilon}\right\}$$

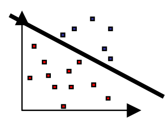
What is random what is not

PAC approach: fixed P and $c \in C$, the asymptotic frequency with which samples z_m from P and c have $|\phi[h \div c] - \mathbf{E}[h \div c]| < \varepsilon$ is high enough…..

Algorithmic Inference approach: given C, the asymptotic frequency with which samples z_m from any P and whatever $c \in C$ have $|\phi[h \div c] - u_{c \div h}| < \varepsilon$ is high enough…..

Outline

1. Statistical basics
   - Algorithmic inference
   - Inferring a Boolean variable
   - Learning a Boolean function
2. Learning tools
   - Symbolic →Boolean: Decision trees, SVM
     →Continuous: Linear Regression
   - Non symbolic → Neural Networks
     → Genetic Algorithms



Two widespread classes of concepts

- Canonical Boolean forms
- Support Vector Machines

Canonical forms

Conjunctive Normal Form (CNF)

$$(v_1 \vee v_2) \wedge (v_3 \vee v_5 \vee v_7) \wedge (v_1 \vee v_4 \vee v_5)$$

Diusjunctive Normal Form (DNF)

$$(v_1 \wedge v_2 \wedge v_5) \vee (v_4 \wedge v_5 \wedge v_7 \wedge v_8) \vee (v_1 \wedge v_4)$$

Learning atomic formulas

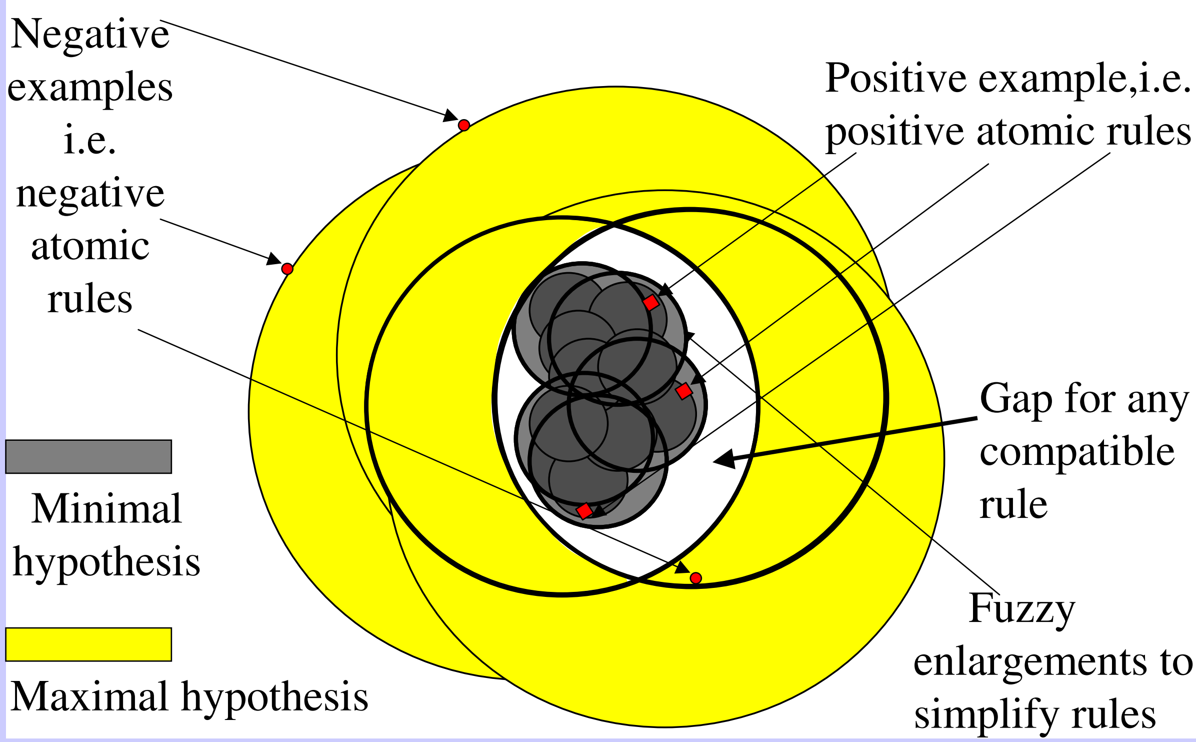
given $\mathbf{X}_n$ and set $E^+$of positive examples you get monomial m

$$\text{for each } i \in \{1,\ldots,n\}, \quad v_i \begin{cases} \in \mathrm{set}(m) & \text{if } x_i = 1 \\ \notin \mathrm{set}(m) & \text{otherwise} \end{cases}$$

given $\mathbf{X}_n$ and set $E^-$of negative examples you get clause c

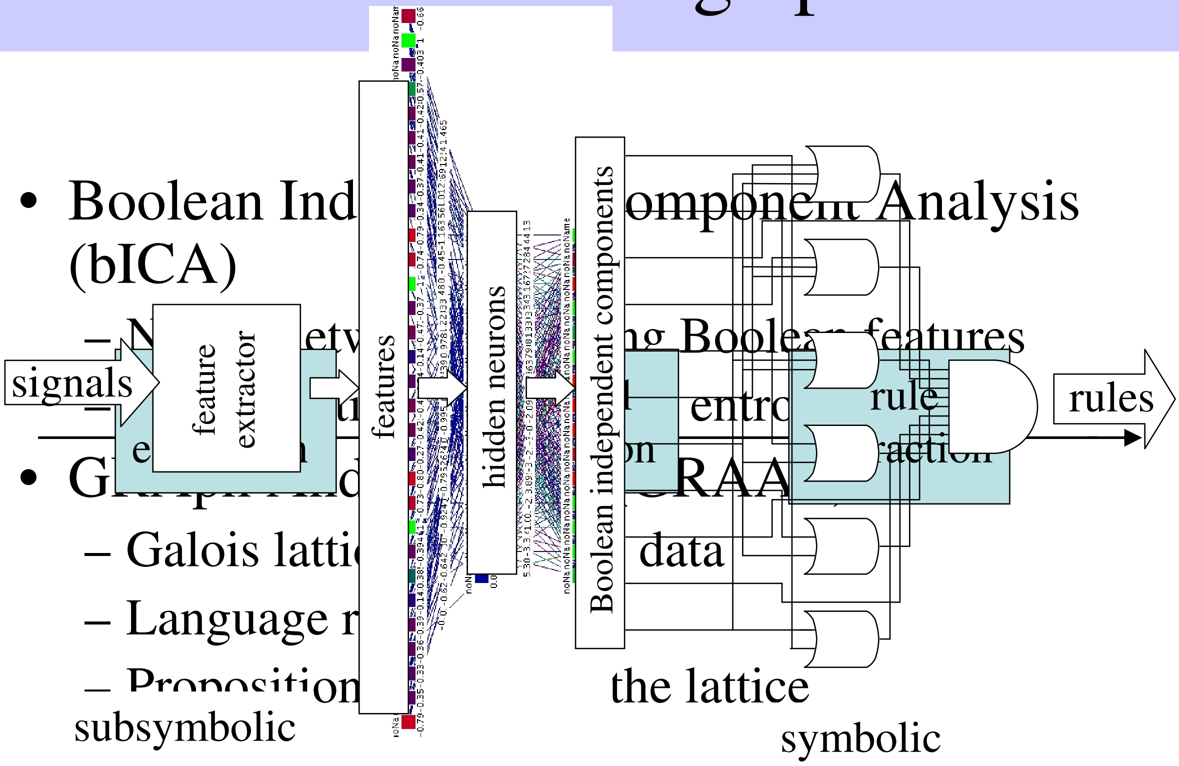
$$\text{for each } i \in \{1,\ldots,n\}, \quad v_i \begin{cases} \in \mathrm{set}(c) & \text{if } x_i = 0 \\ \notin \mathrm{set}(c) & \text{otherwise} \end{cases}$$

Building rules



Which are the boiling up variables?

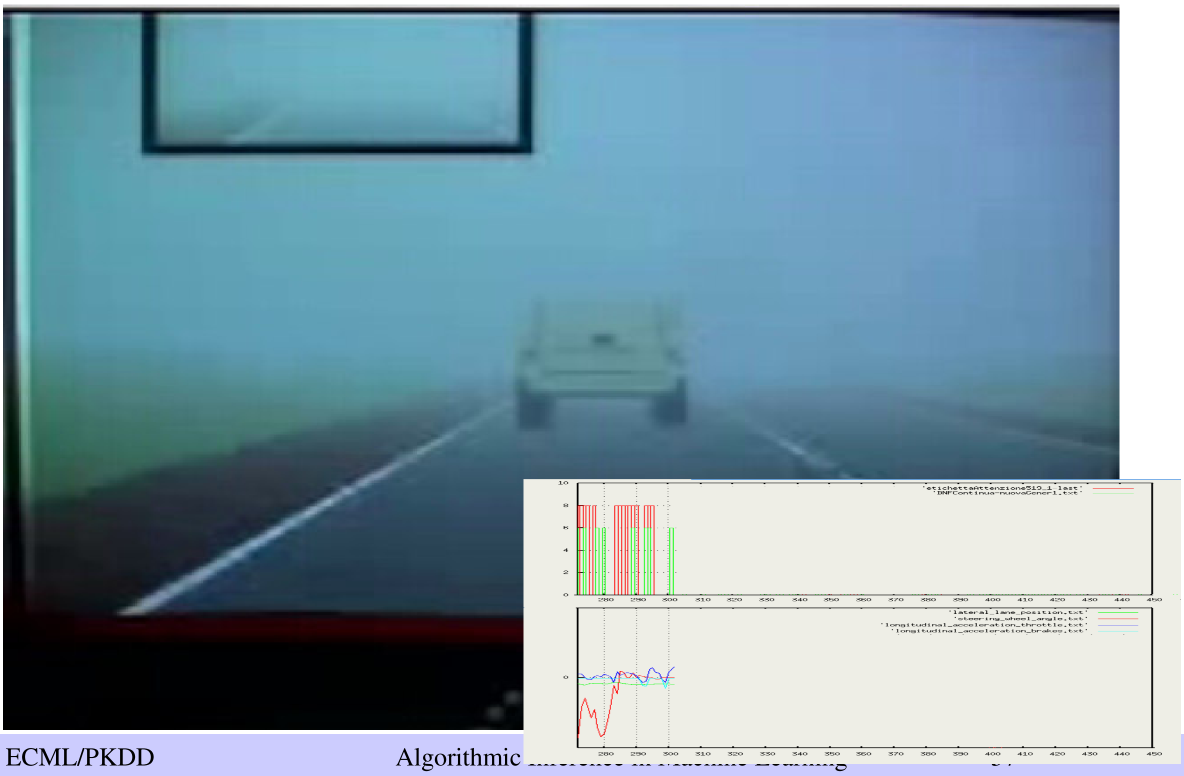
- Boolean Ind… omponent Analysis (bICA)
  – N… etw… ng Boolean features
  – … entro…
  e… n
- G… RAA
  – Galois latti… data
  – Language r…
  – Proposition… the lattice



Comparison with other methods

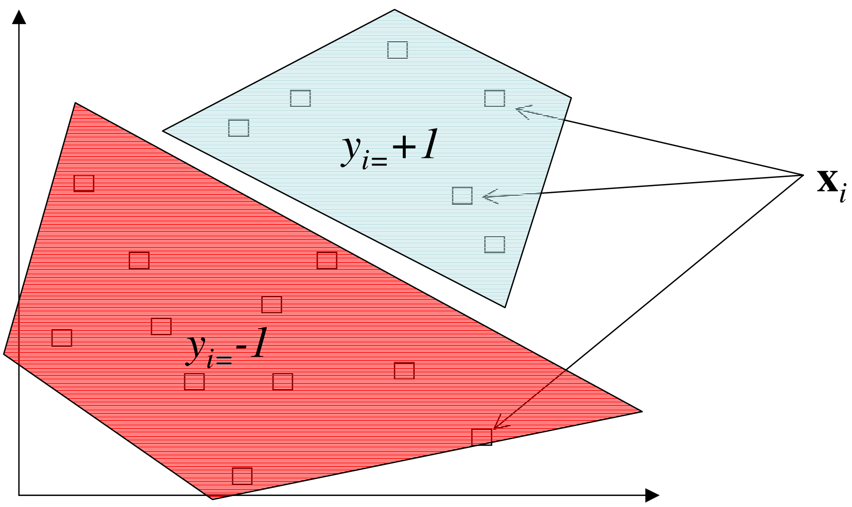
	MONK1	MONK2		MONK3	
FR - Mean Values	99,991%		72,57%		80,47%
FR - STD DEV	1,30%		2,19%		2,84%
Best Of 50	100%		77,45%		84,49%
STATEX1.0	100%	65,42%	71,61%	80,86%	87,78%
Best Of 50	100%	68,75%	81,48%	86,11%	94,91%
STATEX2.0	97,14%	72,69%	71,84%	86,53%	83,33%
Best Of 50	100%	83,33%	78,94%	93,21%	95,37%
AQ17-DCI	100,00%	100,00%		94,20%	
AQ15-GA	100,00%	86,80%		100,00%	
Assistant Professional	100,00%	81,30%		100,00%	
mFOIL	100,00%	69,20%		100,00%	
ID5R	81,70%	61,80%			
IDL	97,20%	66,20%			
ID5R-bat	90,30%	65,70%			
TDIDT	75,70%	66,70%			
AQR	95,90%	79,70%		87,00%	
CN2	100,00%	69,00%		89,10%	
CLASSWEB 0.10	71,80%	64,80%		80,80%	
CLASSWEB 0.15	65,70%	61,60%		85,40%	
CLASSWEB 0.20	63,00%	57,20%		75,20%	
PRISM	86,30%	72,70%		90,30%	
ECOBWEB l.f.	71,80%	67,40%		68,20%	
ECOBWEB l.f. & i.u.	82,70%	71,30%		68,00%	
Backpropagation	100,00%	100,00%		93,10%	
BackProp weigth decay	100,00%	100,00%		97,20%	
Cascade Correlation	100,00%	100,00%		97,20%	

Example: monitoring car driver awareness

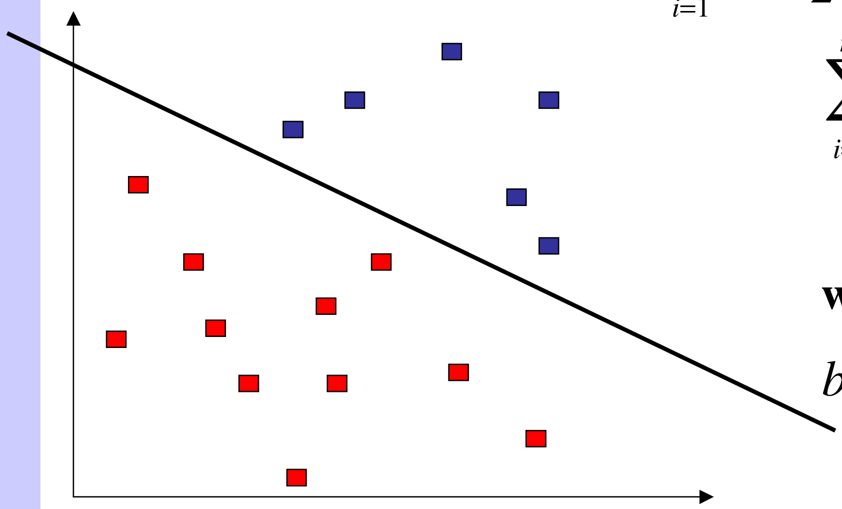


Support Vector Machines

$$\{(\mathbf{x}_i, y_i), i = 1, \ldots, m\}$$



Support Vector Machines



$$\min \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

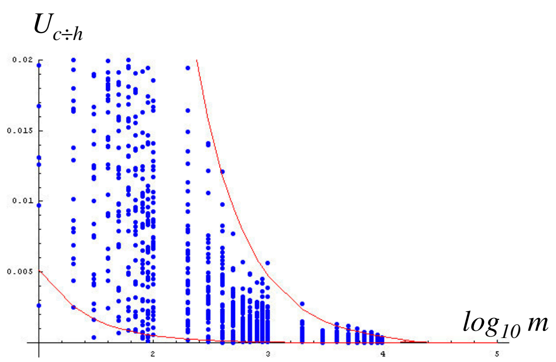
$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

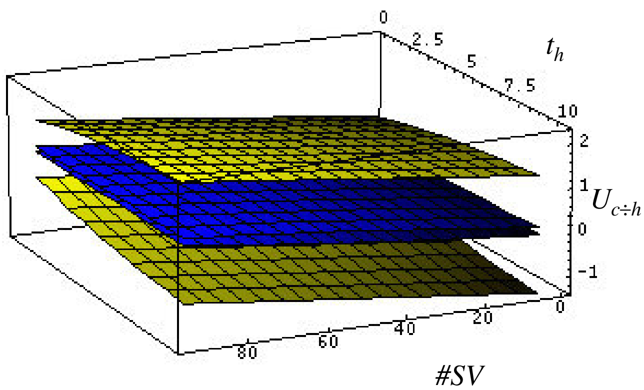
$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$b = y_i - \mathbf{w} \cdot \mathbf{x}_i$$

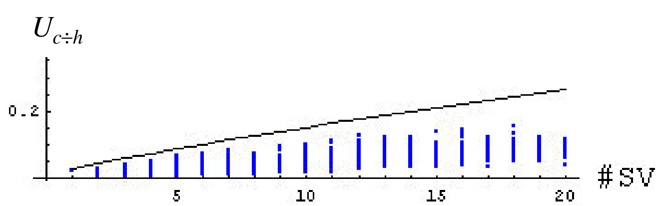
(i s.t. $\alpha_i > 0$)

The confidence region for mislabeling probability



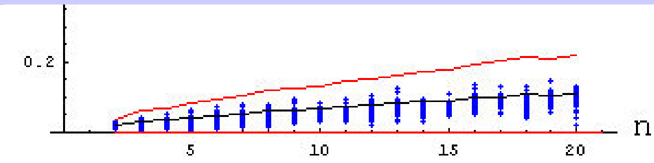


Coverages with Algorithmic Inference

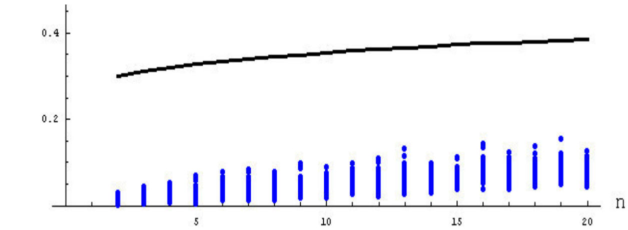


Yellow: Vapnik Approach
Blue: Algorithmic Inference

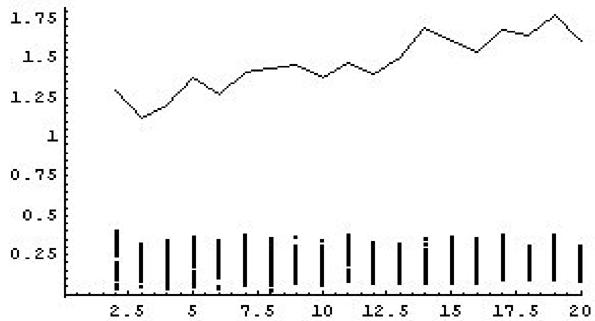
Intervals based on margin measures



Normalized bounds and their coverage using sentry points



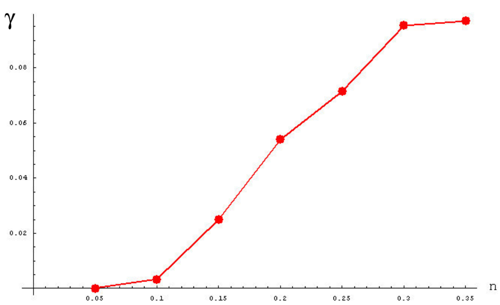
Optimistic bounds using Rademacher complexity



A realistic though unfeasible bound using Rademacher complexity

Sample versus computational complexity

Theorem: the number of sentry points of a hyperplane inferred through a SVM is less than the number of its support vectors

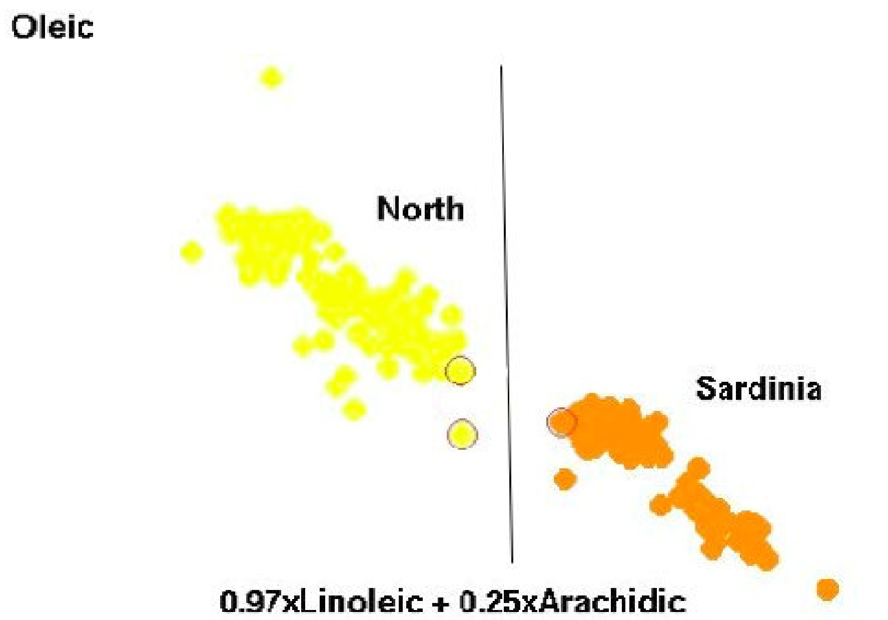


Fact: the number of sentry points ranges from

- 1, in case of exact search algorithm, to
- the number of sentry points, with degrading algorithm accuracy

Claim: the **less** you pay in computational accuracy
the **more** you spend in sample size

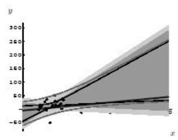
Example: Olive oil recognition



Source: D. Cook, D. Caragea and V Honavar, Visualization for Classification Problems, with Examples Using Support Vector Machines, KDD2004

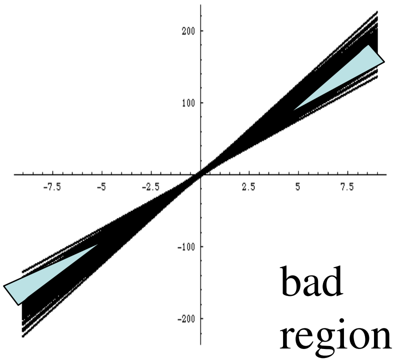
Outline

1. Statistical basics
   - Algorithmic inference
   - Inferring a Boolean variable
   - Learning a Boolean function
2. Learning tools
   - Symbolic →Boolean: Decision trees, SVM
     →Continuous: Linear Regression
   - Non symbolic → Neural Networks
     → Genetic Algorithms

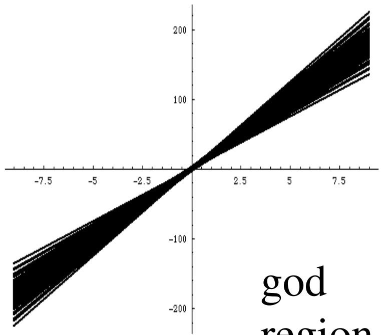


Learning a function

i.e. learning a *dense* confidence region for this function according to a contiguity notion



bad region



god region

Learning a straight line

The labelled sample

$$\mathbf{z}_m = \{(x_i, y_i) : x_i \in \mathfrak{X},\ y_i \in \mathfrak{Y},\ i = 1, \ldots, m\} \subseteq (\mathfrak{X} \times \mathfrak{Y})^m$$

The sampling mechanism

$$y_i = a + b(x_i - \overline{x}) + \varepsilon_i \; i = 1, \ldots, m$$

The class of concepts

$$\mathsf{C} = \{a' + bx : \ a', b \in \mathbb{R}\}$$

we assume that a function c exists within a class C such that, for any suffix $\mathbf{z}_M$ of $\mathbf{z}_m$ (i.e. any continuation of the observed data), and for any (x_i, y_i) belonging to the concatenated sequence $\mathbf{z}_{m+M}$

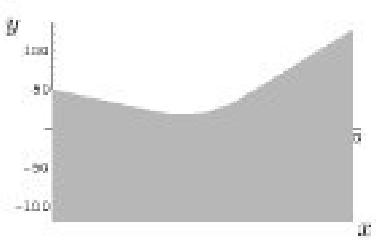
$$y_i = c(x_i) + \varepsilon_i, \quad i = 1, \ldots, m + M \qquad (2)$$

Twisting arguments

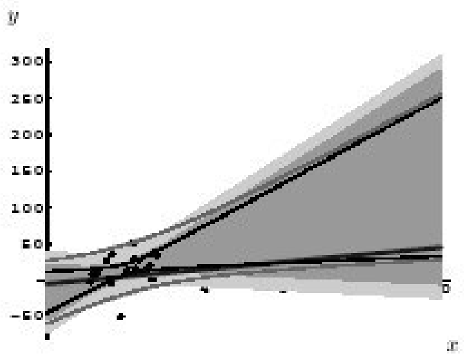
$$(a \le \tilde{a}) \Leftrightarrow \left(\sum_{i=1}^{m} y_i \le \sum_{i=1}^{m} \tilde{y}_i\right)$$

$$\left(b \le \tilde{b}\right) \Leftrightarrow \left(\sum_{i=1}^{m} y_i (x_i - \bar{x}) \le \sum_{i=1}^{m} \tilde{y}_i (x_i - \bar{x})\right)$$

And for the *whole* stright line



Confidence region



$$1 - F_{S''_{E}}(-\delta a - \delta b) = 1 - \frac{\delta}{4}$$

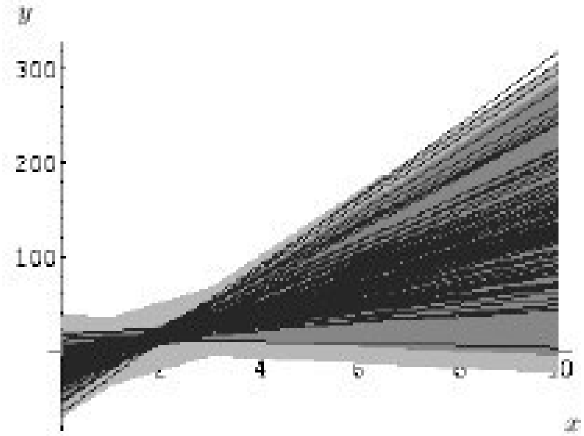
$$1 - F_{S''_{E}}(-\delta a + \delta b) = 1 - \frac{\delta}{4}$$

$$1 - F_{S''_{E}}(-\delta a - \delta b) = \delta/4$$

$$1 - F_{S''_{E}}(-\delta a + \delta b) = \delta/4$$

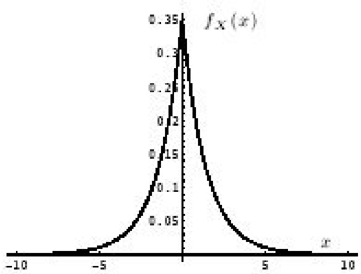
$$\Delta b = |z''_{1-\gamma/4} - \Delta a|$$

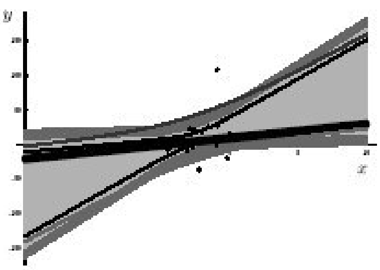
The coverage



A more complex case

$$f_{E_i}(e;\lambda) = \frac{1}{2}\lambda e^{-\lambda|e|},$$



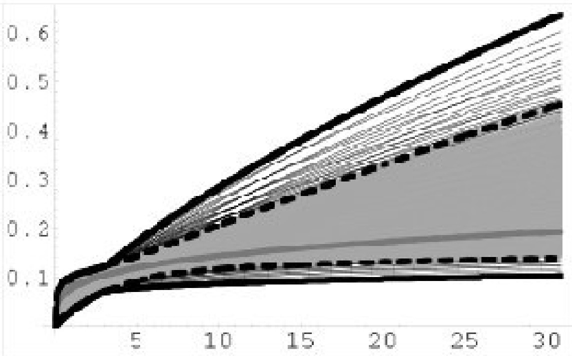


A still complex one

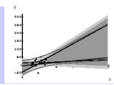
$$F(t) = 1 - e^{-t/(\beta_0 \beta_1^{-\log t})}$$

$$h(t) = \frac{1}{\beta_0} \beta_1^{\log t}(1 + \log \beta_1)$$ ← goal function

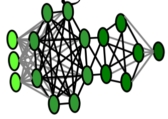
with $\beta_0 > 0, \beta_1 > 1$



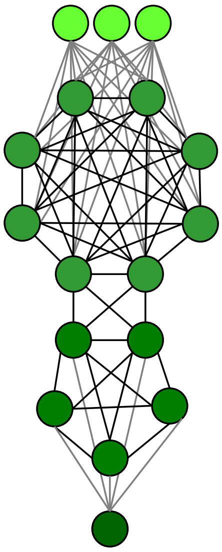
from breast cancer survival data



Outline

1. Statistical basics
   - Algorithmic inference
   - Inferring a Boolean variable
   - Learning a Boolean function
2. Learning tools
   - Symbolic →Boolean: Decision trees, SVM
     →Continuous: Linear Regression
   - Non symbolic → Neural Networks 
     → Genetic Algorithms

A very vague concept class: Neural Networks



•A composition of generic non linear functions

•A lot of free parameters for computing any real function

Only some directions for driving the network

•Twisting argument

•Cost function

•Monotony

•Stopping rule

The twisting argument

$$(T_\pi \geq t) \Leftarrow (\Pi \leq \pi) \Leftarrow (T_\pi \geq t + \mu)$$

•We expect that minimizing cost function T we minimize goal function ∏ as well (right implication)

•We check ∏ minimization b y checking T minimization (left implication)

•No robust probabilistic companion results

Cost function

- Suitable to pivoting a twisting argument, hence
  - Monotonically decreasing with the goal function
  - Well defined in the whole parameter space

Example:

- Kullback distance is well suited for learning probabilities;

$$I(\phi, \pi) = \sum_{i=1}^{m} \varphi(s_i) \log\left(\frac{\varphi(s_i)}{\pi(s_i)}\right)$$

- MSE could bring to unfeasible values of the probabilities (e.g. not adding to one)

$$MSE = \sum_{i=1}^{m} (s_i - \tau_i)^2$$

Goal function

- A function that may be computed by a sagacious agent in feasible time
- Monotonic game against nature
  - For instance solve wisely a knapsack problem with an approximate algorithm whose accuracy grows with the running time

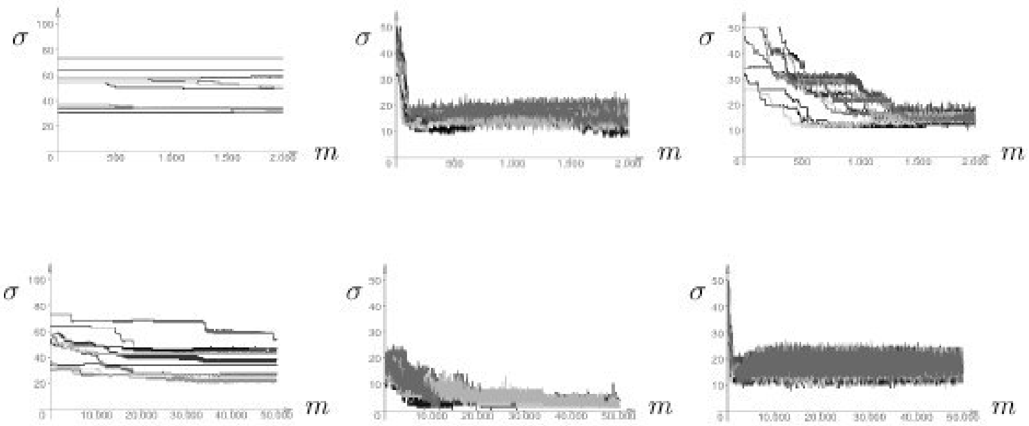
Stopping rule

Rule 1 *Starting from a set of m examples to learn a function f by minimizing a cost function σ as good pivot for minimizing a goal function τ:*

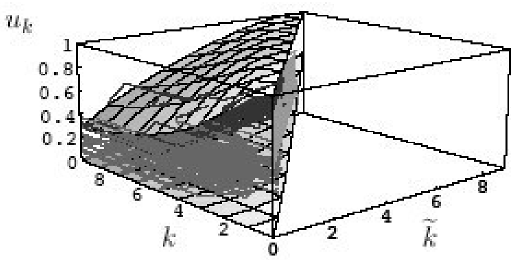
use all the examples as a training set;
if *σ goes satisfactorily fast to 0 with training iterations*
  then *you are OK,*
  else *having reached at a given iteration t a suspicious almost stationary trend of σ,*
    repeat *the training algorithm many times starting from a different random initialization of the parameters to be learnt and stopping each time at same iteration t,*
    if *the variance of the last few values of σ merged between the iterations is moderately large,*
      then *continue the training,*
      else *stop the training.*

Example

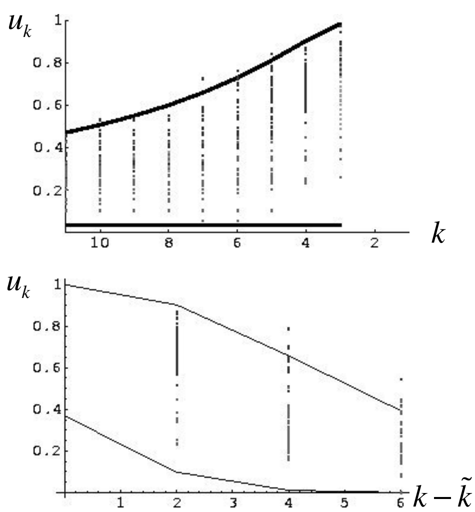
Which learning process is worth to continue?



The sails' diagram



(a)



Outline

1. Statistical basics
   - Algorithmic inference
   - Inferring a Boolean variable
   - Learning a Boolean function
2. Learning tools
   - Symbolic →Boolean: Decision trees, SVM
     →Continuous: Linear Regression
   - Non symbolic → Neural Networks
     → Genetic Algorithms

TATTAGATATTTTCTTTATTTACATTTCAAA
TATTAGATATTTTCTTTATTTACATTTCAAA

Another way for tuning ∏ Genetic Algorithms

```
ACTCATTTGGTTAAAGTGCTGTCCC
GAATAACCCTTTACACGATACTAAC
!!!!=!!!!!===!=!!!!!!=!!=
```

Optimization rule:

given a target function on phenotype (fitness),

change and select the genotype

TATTAGATATTTTCTTTATTTACAT
TATTAGATATTTTCTTTATTTACAT

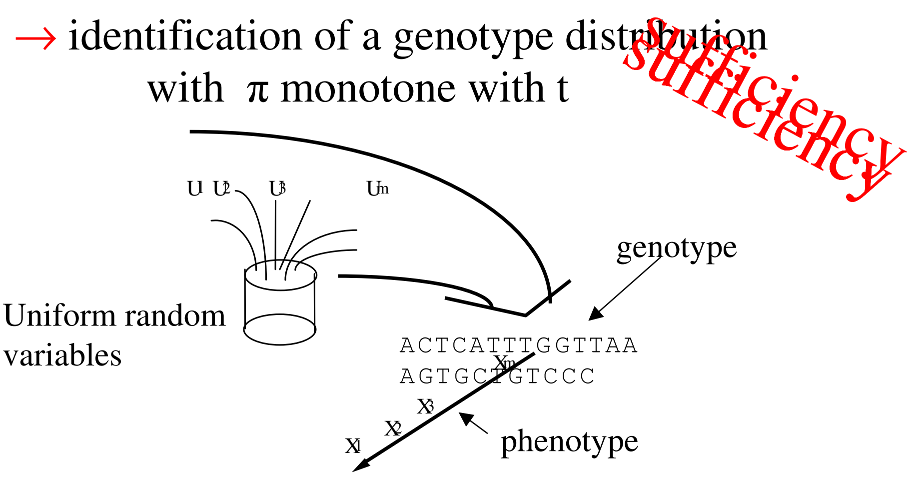
The twisting argument

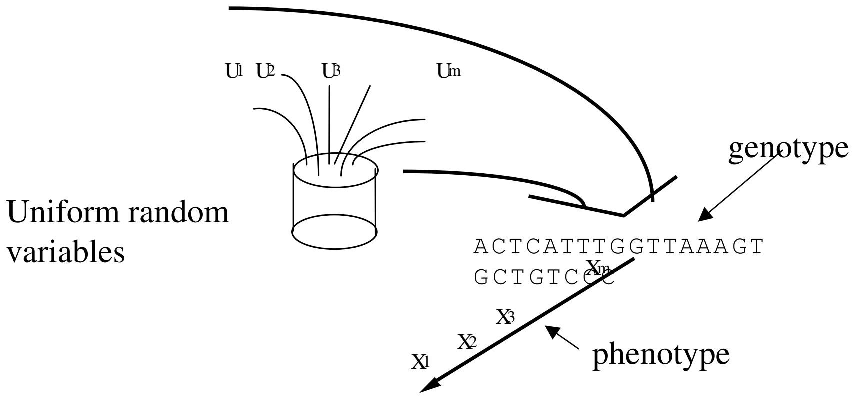
$$(T_\pi \geq t) \Leftarrow (\Pi \leq \pi) \Leftarrow (T_\pi \geq t + \mu)$$

- π → genotype
- t → phenotype
- we expect that optimizing the phenotype fitness of a string we also optimize its genotype, hence the phenotype of the whole population

Typical learning tools

- Crossover → exchange of independent sequneces of underlying u's
- Mutation → different extractions from U
- Selection → identification of a genotype distribution with π monotone with t





Bibliography

- D. Angluin and P. D. Laird, Learning from noisy examples, *Machine Learning*, 2(2):343-370, 1988.
- B. Apolloni, A. Esposito, D. Malchiodi, C. Orovas, G. Palmas and J. G. Taylor, A General Framework for Learning Rules From Data, *IEEE Trans. on Neural Networks*, 2004, to appear.
- B. Apolloni and S. Chiaravalli,PAC learning of concept classes through the boundaries of their items, *TCS* 172: 91-120.
- B. Apolloni and D. Malchiodi, Gaining degrees of freedom in subsymbolic learning, *TCS* 255: 295-321.
- B. Apolloni, D. Malchiodi and S. Gaito, *Algorithmic Inference in Machine Learning*, Adelaide: Advanced Knowledge International, 2003.
- P. L. Bartlett and S. Mendelson, Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463-482, 2002.
- C. M. Bishop, *Neural networks for pattern recognition*, Oxford: Clarendon Press, 1995.
- A. Blumer, A. ehrenfreucht, D. Haussler and M. Warmuth, Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36:929-965, 1989.
- N . Cristianini and J. Shawe-Taylor, *An introduction to Suypport Vector Machines and other kernel-based learning methods*, Cambridge: Cambridge University Press, 2000.

Bibliography

- B. De Finetti, *Theory of Probability*. Vol. 2, New York: John Wiley & Sons, 1975.
- M. A. Fisher, The fiducial argument in statistical inference, *Ann. Of Eugenics*, 6:391-398, 1935.
- S. Geisser, *Predictive Inference: An Introduction*. New York: Chapman & Hall, 1993.
- L. G. Valiant, A Theory of the Learnable, *Communications of the ACM* 11(27):1134-1142, 1984.
- V. K. Rohatgi, *An Introduction to Probability Theory and Mathematical Statistics*, New York: John Wiley & Sons, 1976.
- A. Roth and I. Erev, Learning in extensive from games: experimental data and simple dynamic models in the intermediate term, *Games and Economic Behavior* 8:164-212, 1995.
- V. Vapnik, *Statistical Learning Theory*, New York: John Wiley & Sons, 1998.
- S. Zacks, *The Theory of Statistical Inference*, New York: John Wiley & Sons, 1971.