Machine Learning and Optimization in Tourism and Hospitality

Roberto Battiti and Mauro Brunato

LION-lab

University of Trento

LION
intelligent-optimization.org

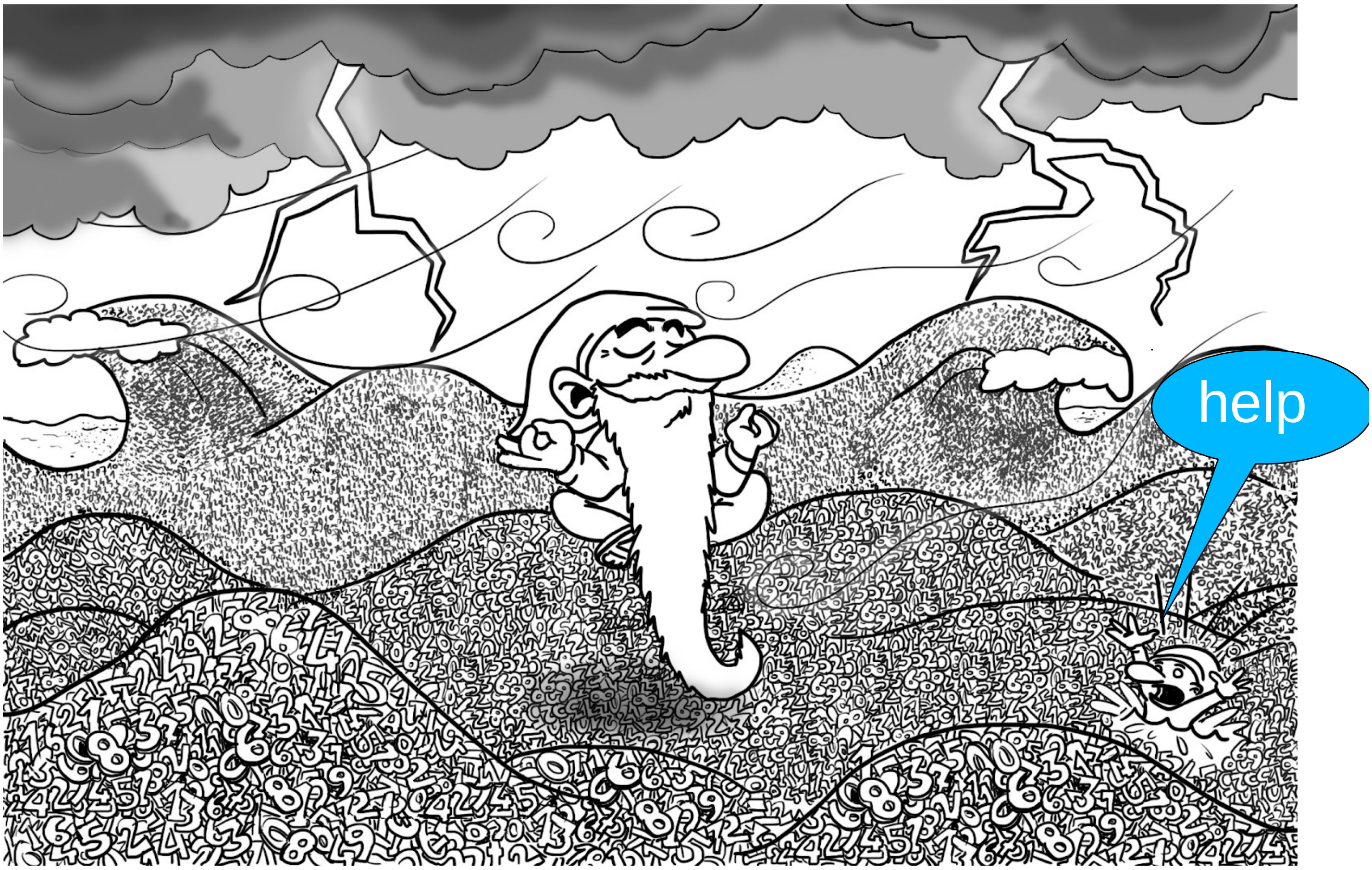
UNIVERSITY OF TRENTO - Italy

Objectives of course

1. Understand the “landscape” of **Machine Learning**
2. Understand the “landscape” of **Intelligent Optimization**
3. Disruptive innovation by **combining ML + IO** (“automated creativity”)

4. **Opportunities for tourism and hospitality**
5. **Simulation-based optimization**

Some motivation …



Price war and downward contagion

new entries with low price policies

Airbnb

commercial **intermediation**

Booking.com

Google, Facebook "Intermediating disintermediaries«

sirens?

algorithmic intermediation

transparent prices more **knowledge and power in the hands of customers**

billboard effect

reputational intermediation
UGC Portals “user-generated content”
Tripadvisor

social

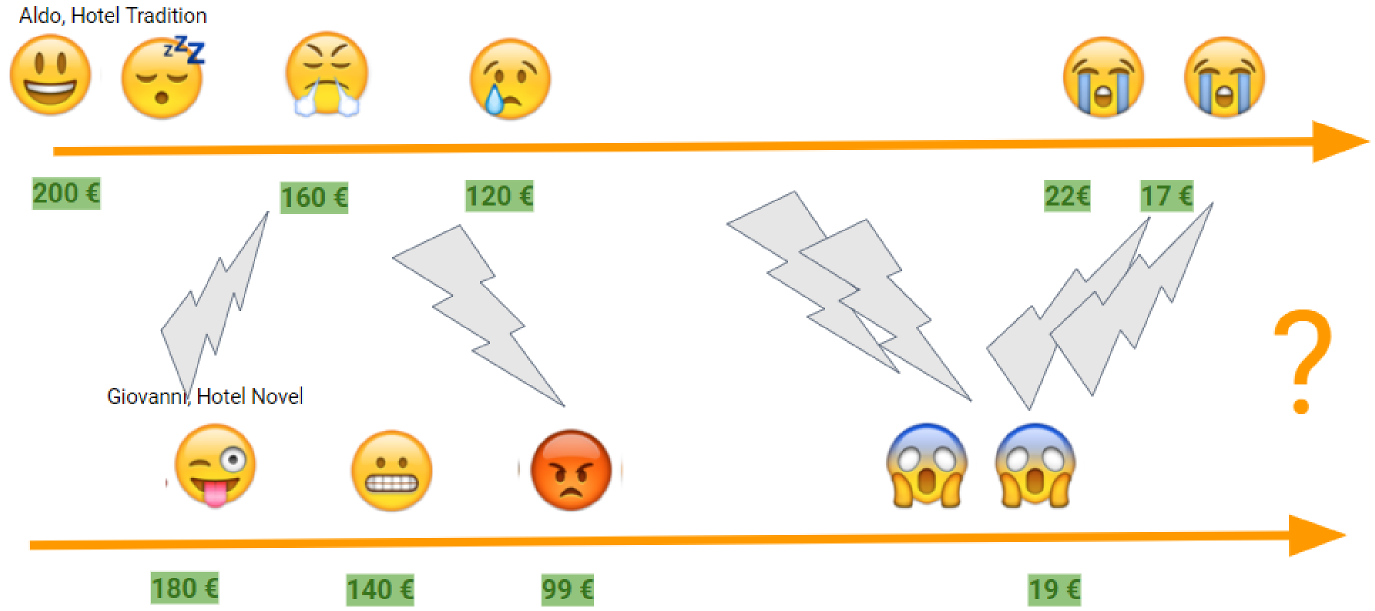
parity rate

Experiences and meanings

Pricing e revenue management

Total profit management

Data… and price wars



…marginal production cost!

- Marginal cost of water?
- H2O price/liter?

Theory or painful practice?

Airbnb (or others ...)

- "Automatic price determination"
- "Optimize my fill“

General-purpose pricing schemes based on **average price analysis** are widespread, give results in the short term, but are **dangerous** for the hotelier in the medium / long term

Exponential revolution of algorithms,

with exponential opportunities and threats

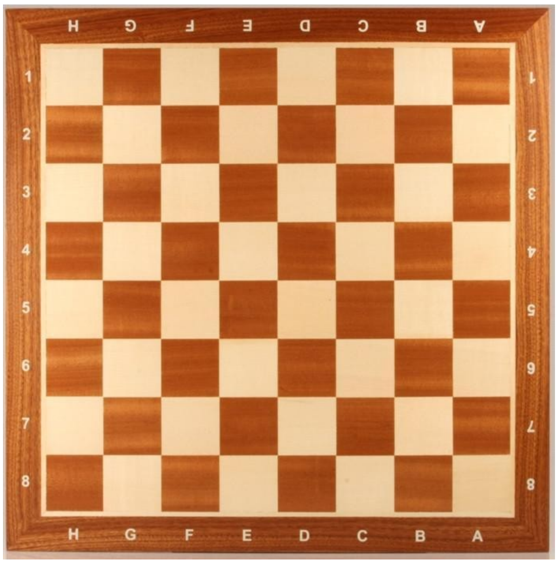
1) Computer power (**speed**)

2) Availability of **memory** (and data)

3) Progress in theory (artificial intelligence, neural networks, machine learning, data, optimization)

Exponential revolution…

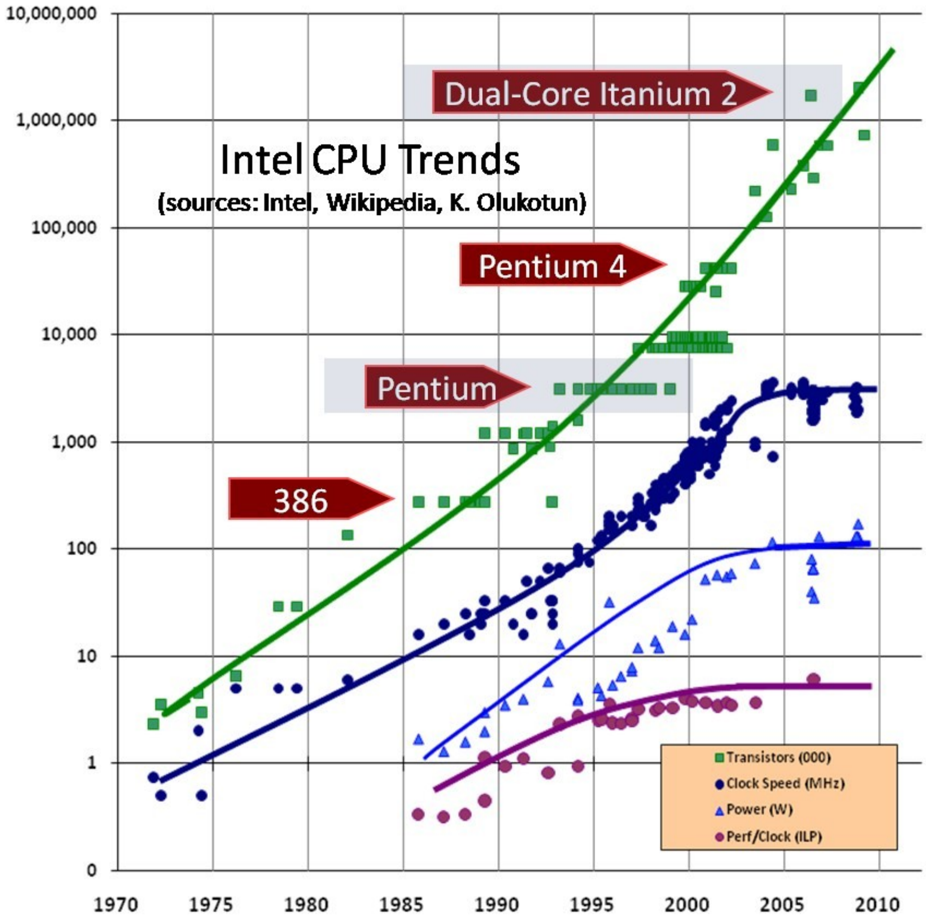
- ... we are used to thinking linearly (gradual changes)



The smart farmer…

2x2x2x2….. (64 volte)

18,446,744,073,709,551,616



2x transistor per chip

Every 18 months

10,000,000 X

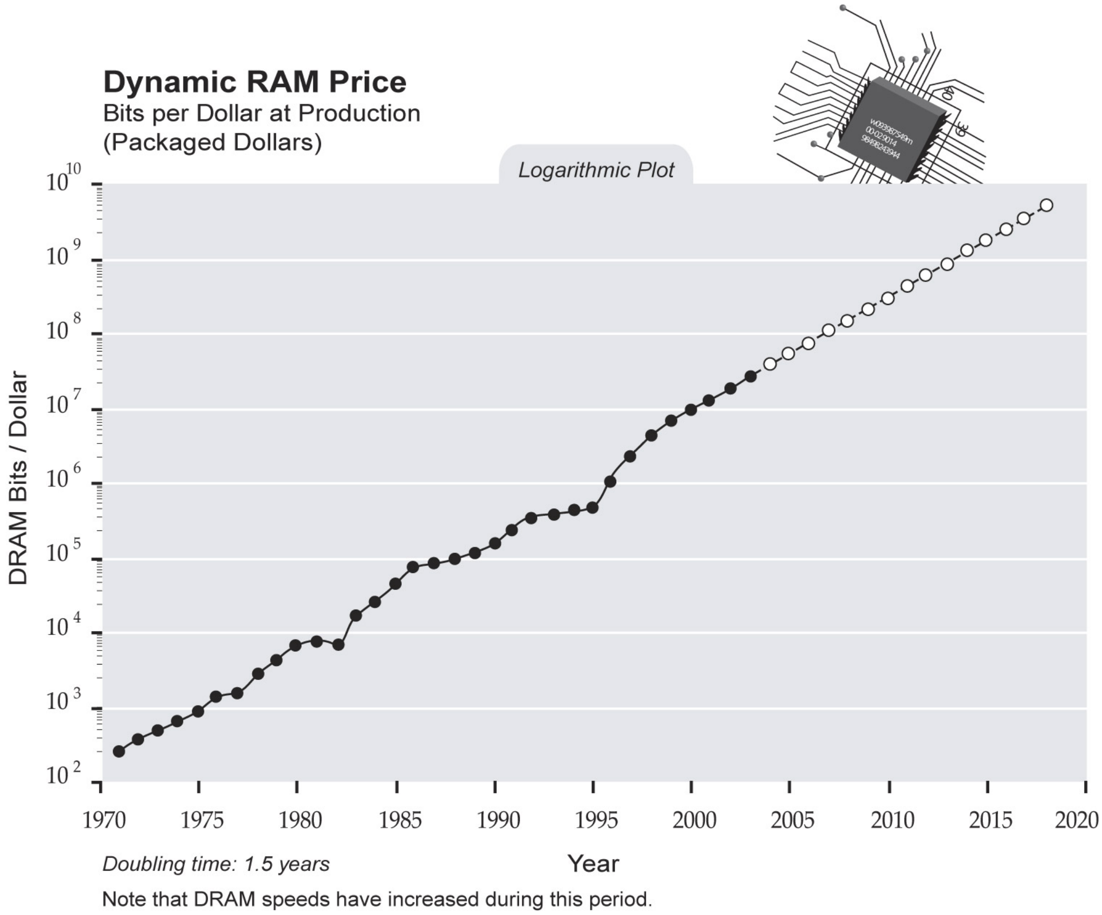
faster

in 30 years

CPU 2.0 GHz

2,000,000,000

cycles / second



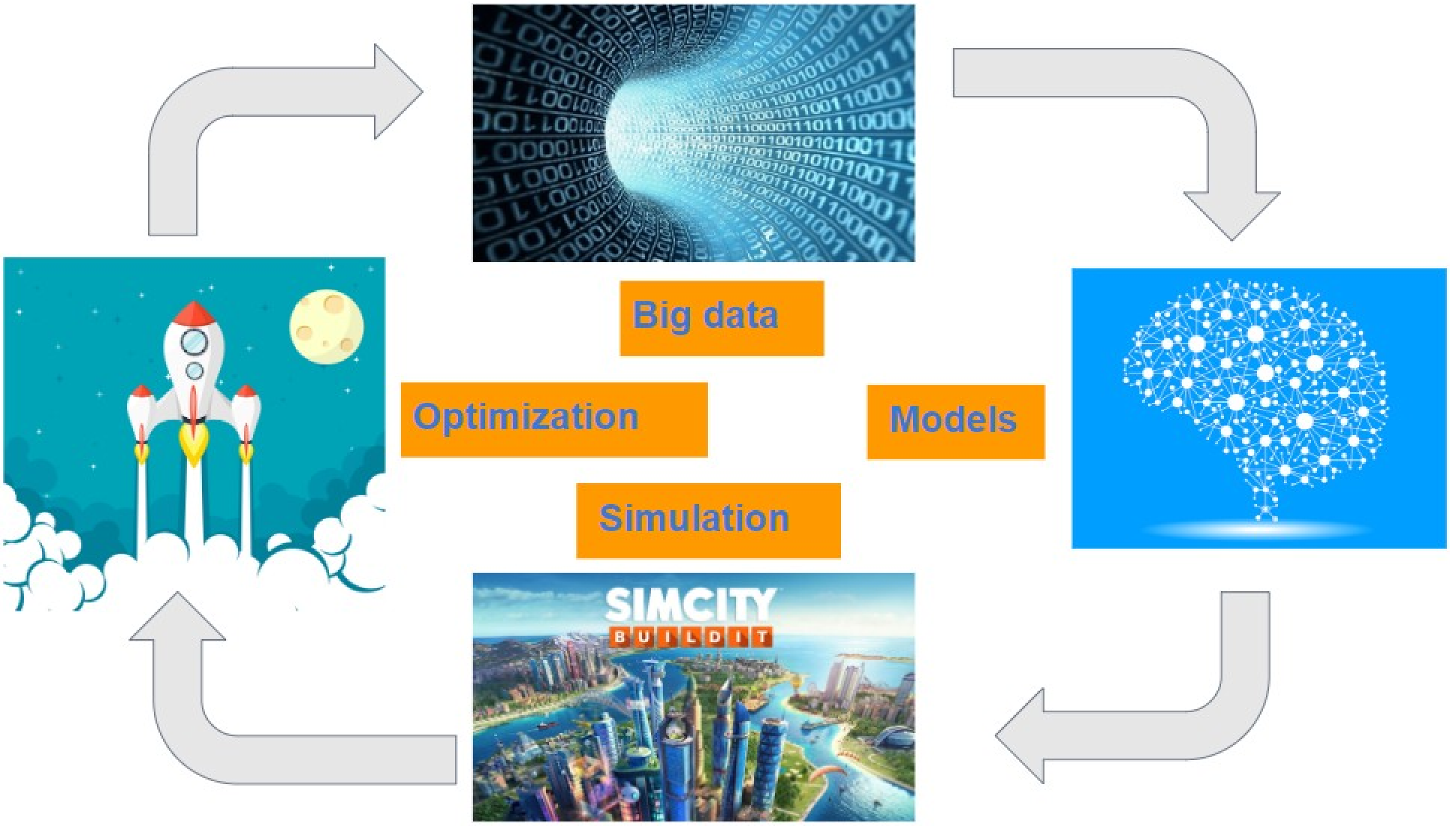
The world

“in your pocket”

2x

every 18 months

In this context…
one ring to rule them all



What’s behind

- use **data** to build models and extract knowledge

Machine learning or learning from data

- exploit **knowledge** to **automate** the discovery of improving solutions

Optimization (automated problem solving)

- connect insight to **decisions and actions**.

Prescriptive analytics (much more than BI)

ROBERTO BATTITI, MAURO BRUNATO.
The LION Way: Machine Learning plus *Intelligent Optimization*.
LIONlab, University of Trento, Italy,

http://intelligent-optimization.org/LIONbook

Part 1
The landscape of Machine Learning

A “zip” of the history of AI - NN - ML

Symbolic AI (up to 1985)	Syb-symbolic Neural nets	Statistics/Machine learning. Deep learning…
Symbols **Logic** **Expert systems** Explicit symbolic programming Inference, search algorithms AI programming languages Rules, Ontologies, Plans, Goals…	**Knowledge in parameters** **Dynamical systems** **Neural Nets / Backprop** Bayesian learning Deep learning Connectionism	

Learning from Data and Machine Learning

If you show a picture to a three-year-old and ask if there is a tree in it, you will likely get the correct answer. If you ask a thirty-year-old what the definition of a tree is, you will likely get an inconclusive answer. We didn't learn what a tree is by studying the mathematical definition of trees. We learned it by looking at trees. In other words, we learned from 'data'.

Yaser Abu-Mostafa

A zip of the history of AI - NN - ML

Symbolic AI (up to 1985) → **Syb-symbolic Neural nets** → **Statistics/Machine learning. Deep learning…**

Easier to debug
Easier to explain
Easier to control
Not so Data-based
More useful for **explaining** people's thought
Better for abstract problems
Fragile
Needs knowledge elicitation
Curse of dimensionality

More robust against noise
Faster (from inputs to outputs)
Less knowledge upfront
Easier to scale up
Data-based
More useful for connecting to neuroscience
Better for perceptual problems

Why do we need models? Why surrogates?

Three ways of building **models**

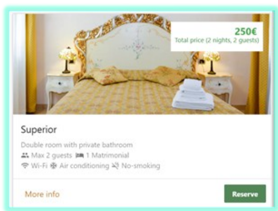
output

Buy or not

Model

- what-if questions
- predictions
- Input optimization

Room type and characteristics
Price
Context
Advertisement



input

1) Explicit and rigid models

Pressure = N k T / V

Model

(Volume, Temperature)

e.g., Physics: Boyles's law:

"For a fixed mass of gas, at a constant temperature, the product (pressure x volume) is a constant."

PV = N k T

Why do we need other models?

2) Parametric, with statistics

Ronald Fisher in 1913

Quantity demanded

Model

Price

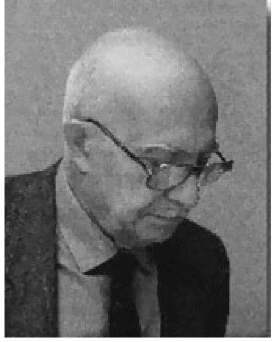
Price elasticity of demand =

$$\frac{\text{Proportionate change in quantity demanded}}{\text{Proportionate change in price}} = \frac{\frac{\Delta Q}{Q} \times 100\%}{\frac{\Delta P}{P} \times 100\%} = \frac{\frac{\Delta Q}{Q}}{\frac{\Delta P}{P}}$$

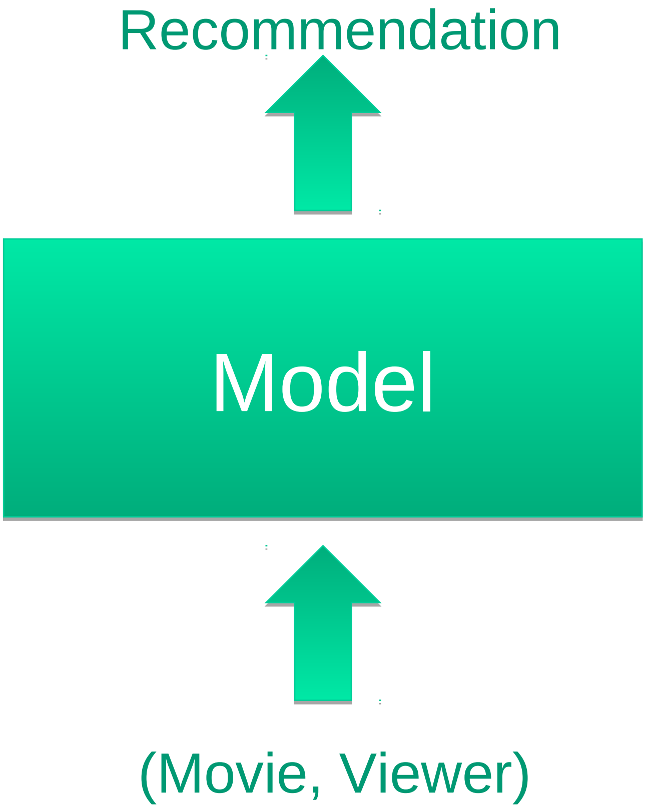
e.g., Maximum likelihood estimation

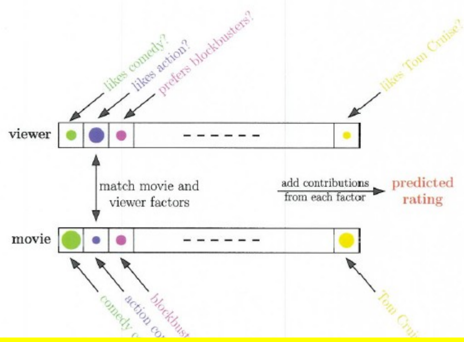
Is this related to Machine Learning?

3) *Non*-parametric models, neural nets, modern ML (1960++, 1985, 2010)



Eduardo Caianiello, 1961



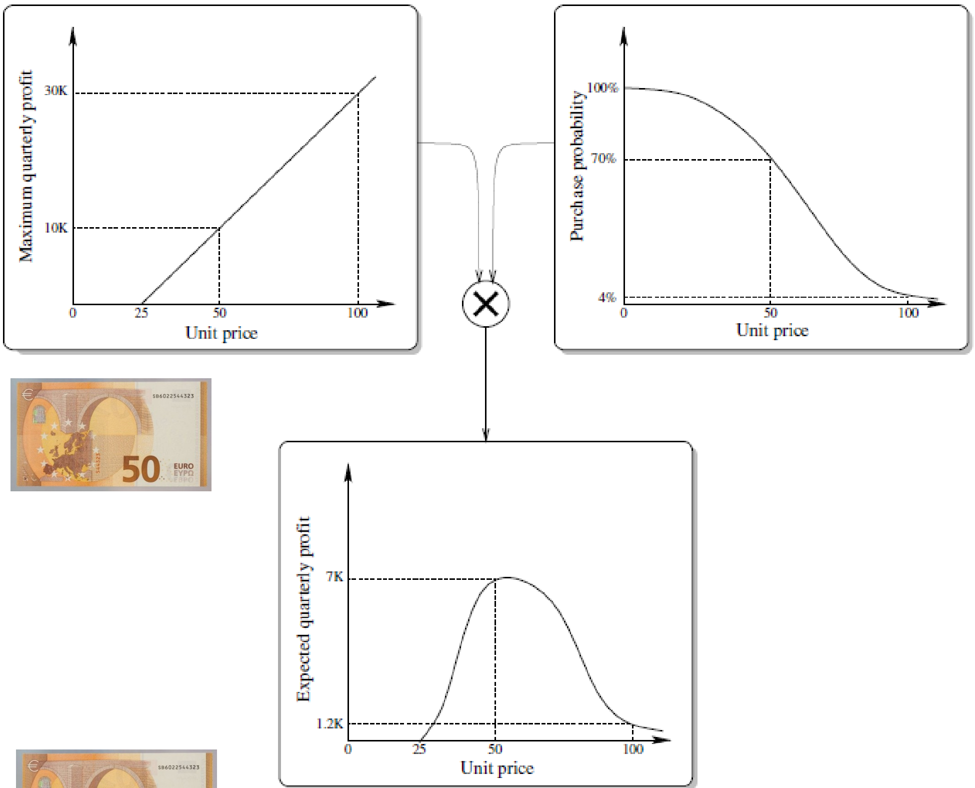


Very flexible, no rules elicitation,
Only need abundant (relevant) data

Different models for different contexts

Which kind of model?

Cost model



Purchase probability

Profit

The dream

"give computers the ability to **learn** without being explicitly programmed" (Arthur Samuel, 1959).

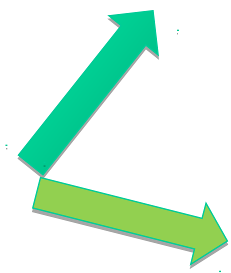
The Tool

Weights of the flexible model are determined via optimization, but aiming at generalization (learning is *mean* not *end*)

No need to be an expert to improve businesses
Business need data scientists

Refresh: vectors and scalar products

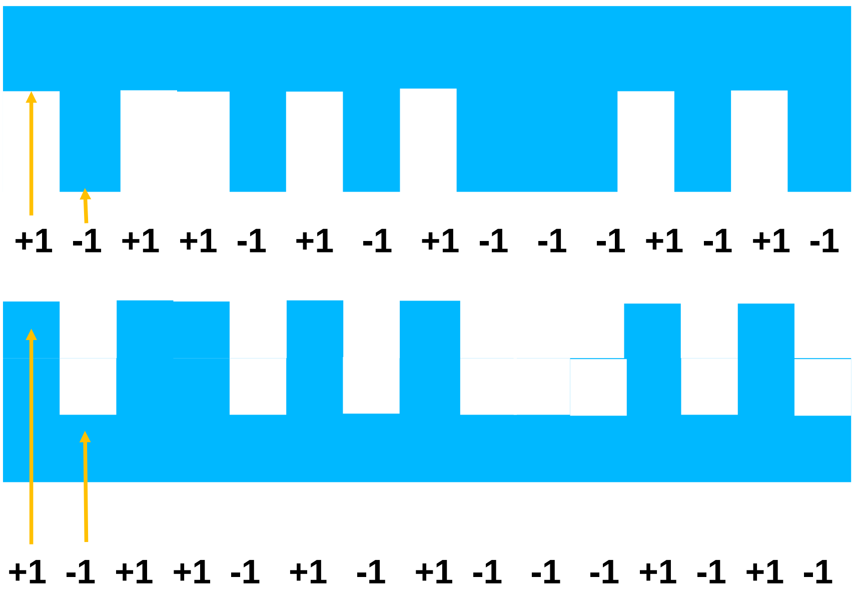
- [4.0, -3.0, 4.0, ….]   measure
- [2.0, -2.0, -3.0, …]
- 8.0   6.0   -12.0
- 8.0 + 6.0   -12.0 = 2.0



Related to linear correlation

“Keys and keyholes”

When is the key opening the keyhole? (15 dents)



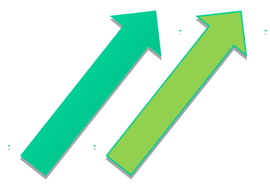
When is a customer buying my room?

Movies and Viewers (hotel rooms and customers)

- Movie1 = [1.2, 3.3, 2.1, …., …., …., …., 7.7]
- Movie2 = [3.2, 5.6, 1.2, …., …., …., …., 3.4]

- Viewer1 = [6.2, 5.6, 7.2, …. 2.1]
- …

Map to vectors of the same dimensions → **m, v**



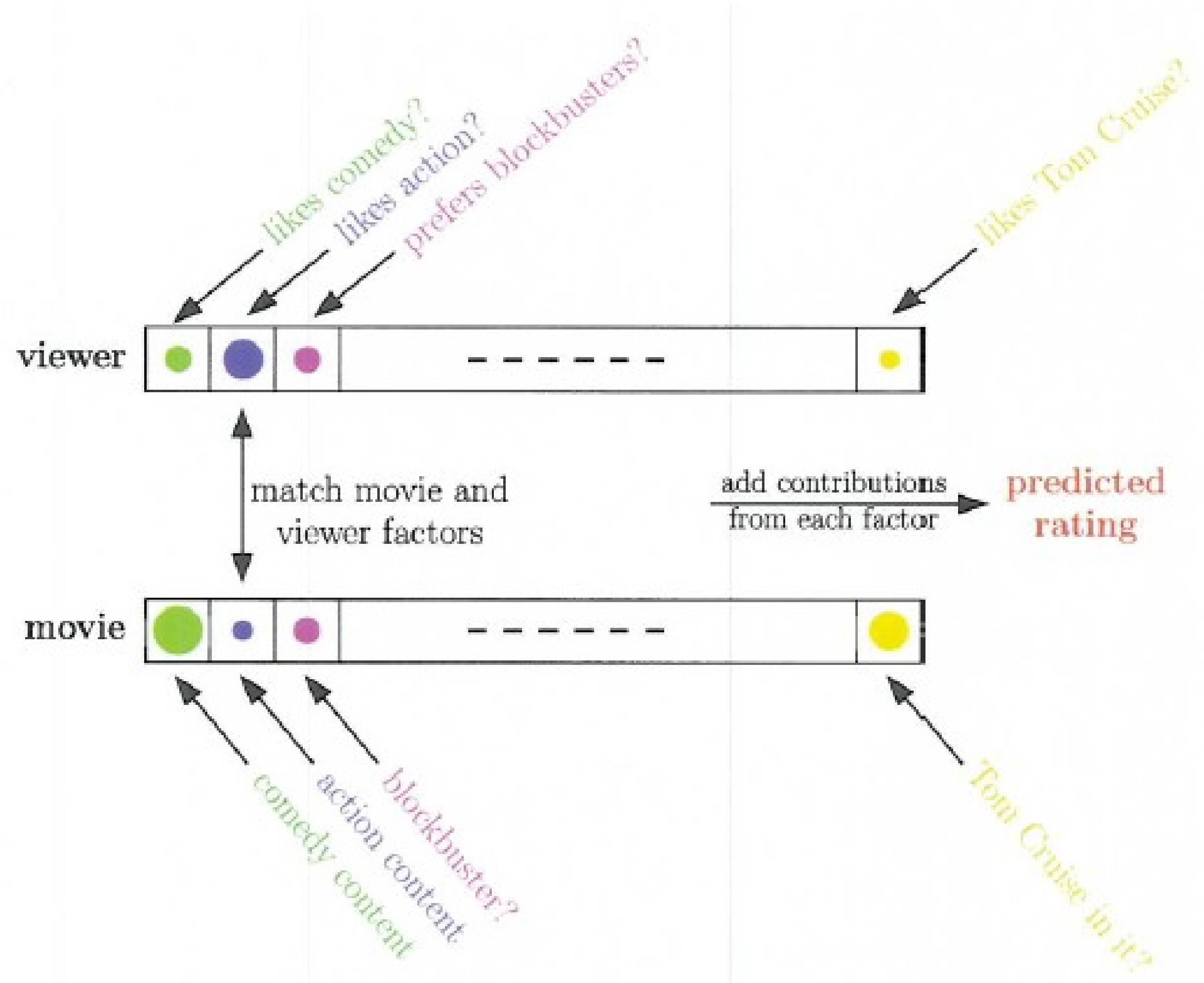
Obtain **rating** by simple scalar product (measure «degree of collinearity fo two vectors»

Measure errors

Objective = Sum_data_i (**m_i . v_i - **r_i)^2

Minimize to determine vectors!

Movies and Viewers



Is it possible? Neural networks!

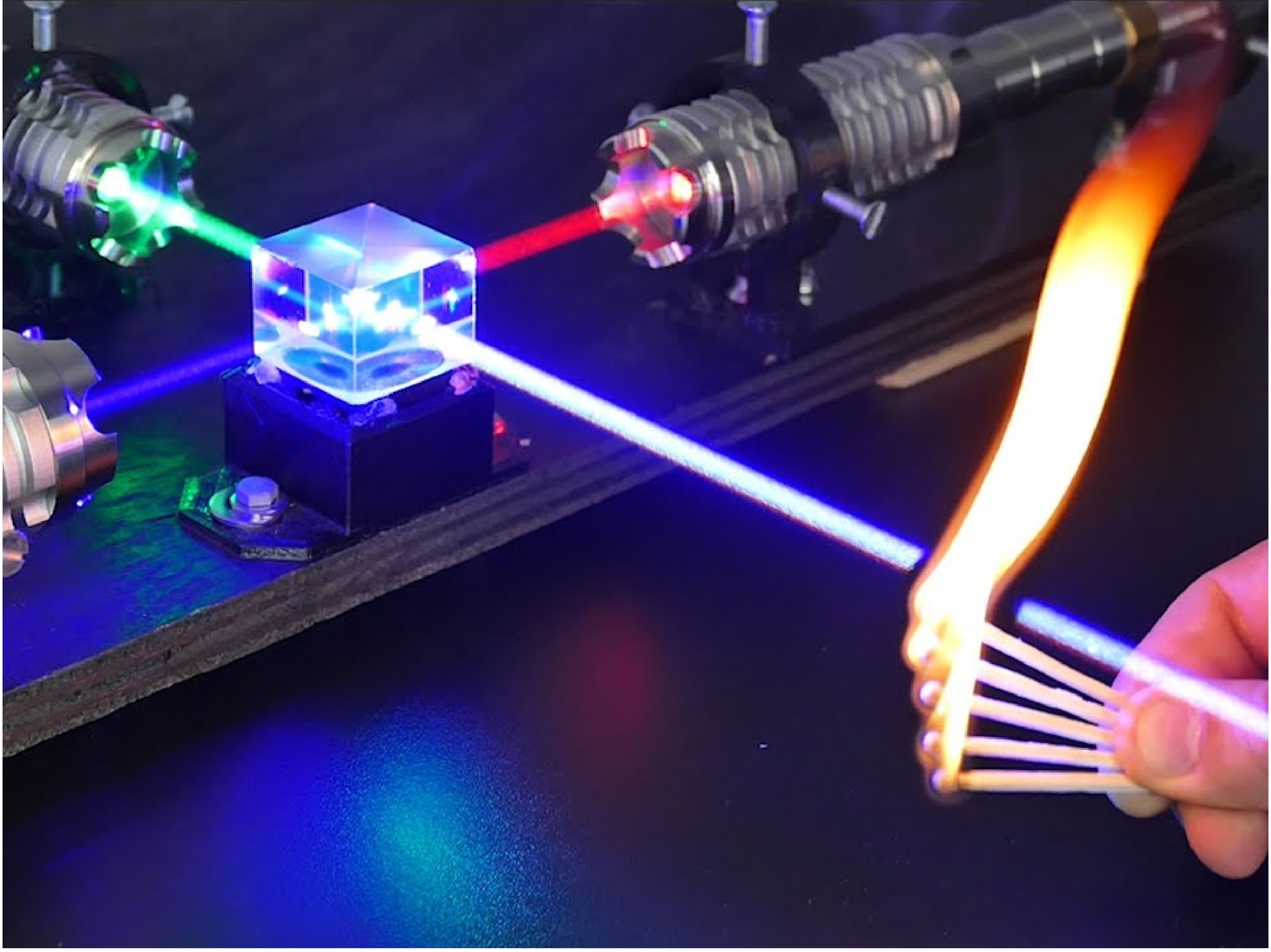
Quegli che pigliavano per altore altro che la natura, maestra de' maestri, s'affaticavano invano.
(Leonardo Da Vinci)

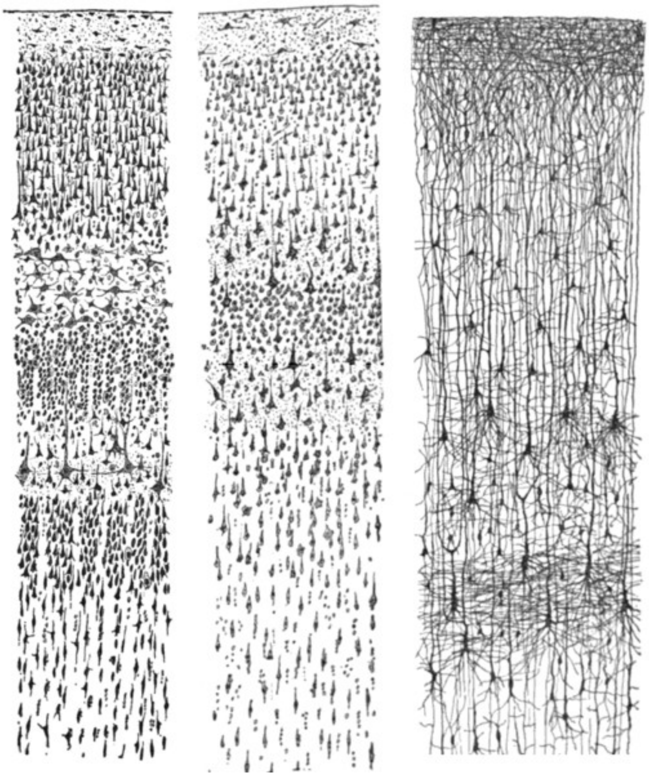
The biological metaphor

We are the living proof of learning from data

- Our neural system is composed of 100 billion computing units (neurons) and 10^{15} connections (synapses).
- How can a system composed of many simple interconnected units give rise to highly complex activities?
- Emergence: complex systems arise out of a multiplicity of relatively simple *interacting* units.

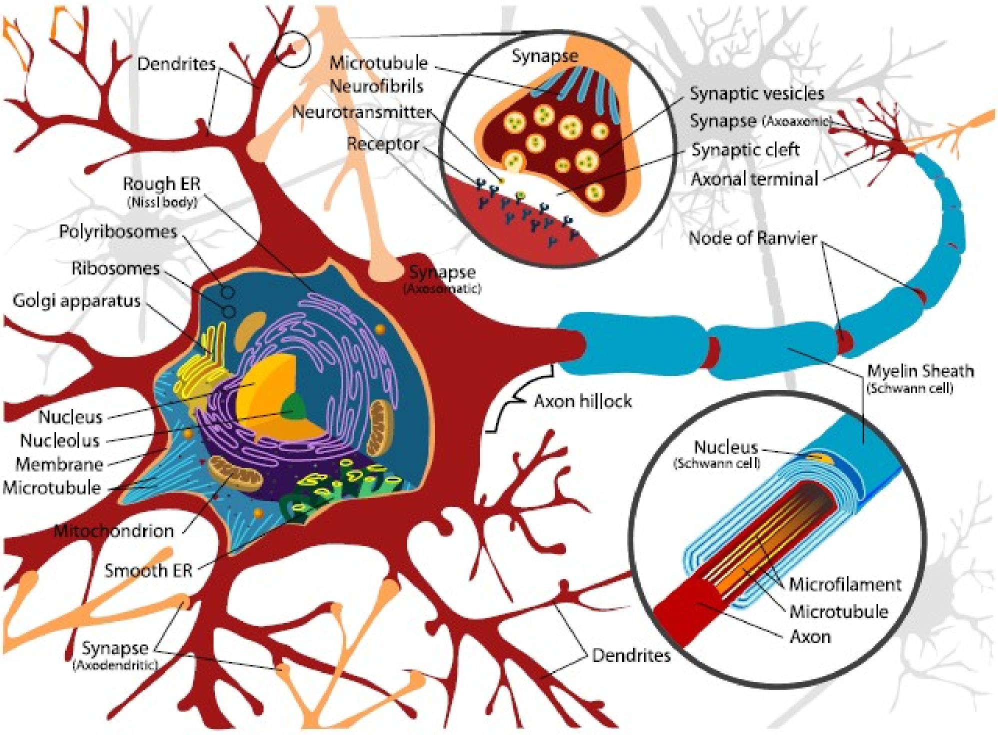
Physics!





Drawings of **cortical lamination** by Santiago Ramon y Cajal, each showing a vertical cross-section, with the surface of the **cortex** at the top. The different stains show the **cell bodies of neurons** and the **dendrites and axons** of a random subset of neurons.

Biological motivations



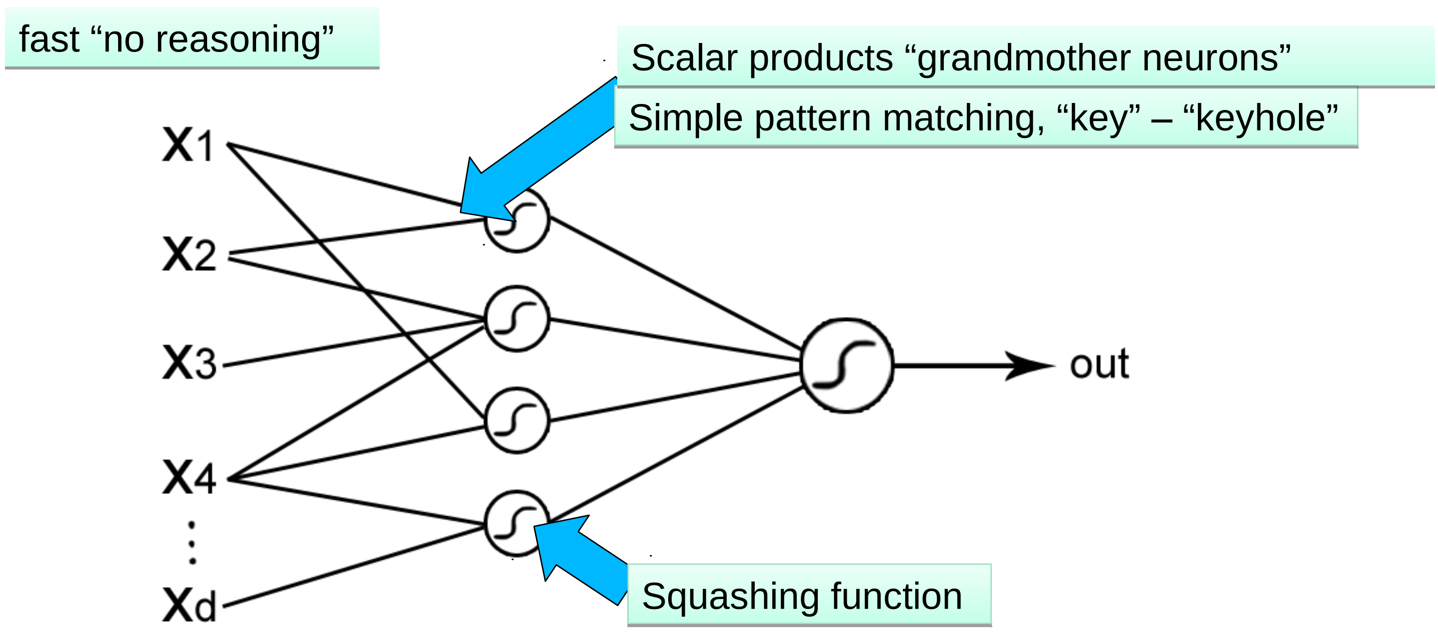
Neurons and synapses in the human brain

Artificial Neural Networks

- A neuron is modeled as a simple computing unit, a scalar product **w x** ("pattern matching") followed by a sigmoidal ("logistic") function.
- The complexity comes from having more interconnected layers of neurons involved in a complex action (if linear layers are cascaded, the system *remains* linear)
- The "squashing" functions is essential to introduce nonlinearities in the system

MLP architecture

- a large number of interconnected units working in parallel and organized in **layers** with a **feedforward** information flow.

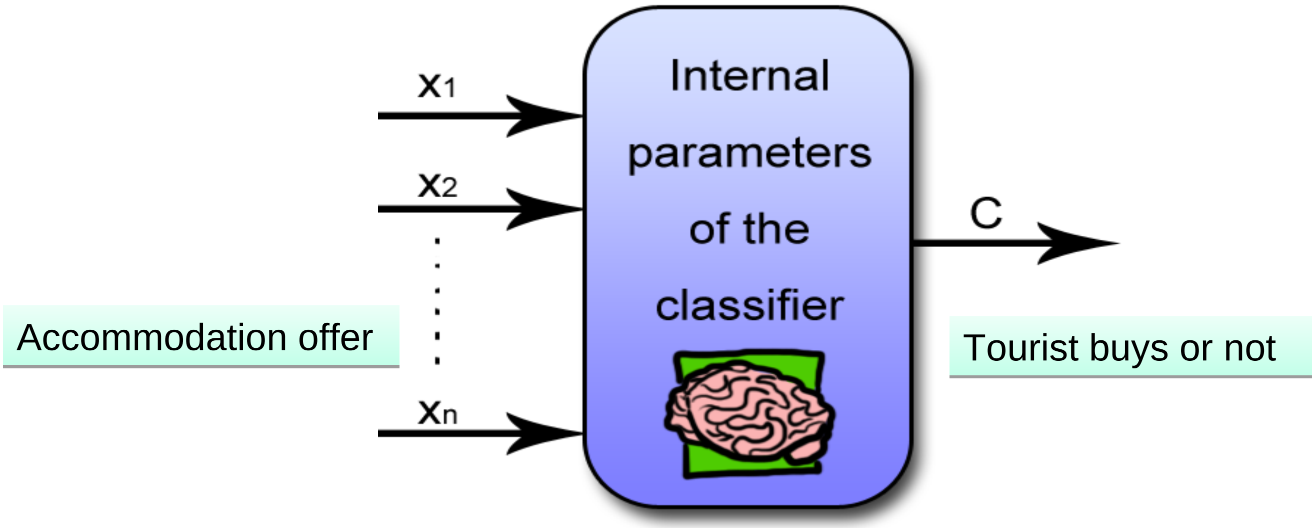


What is learning?

- Learning is *more than memorizing («learning by heart»)*
- Unifying/compressing different cases by discovering the **underlying explanatory laws**.
- Learning from examples is only a **means** to reach the real goal: ***generalization***, the capability of explaining new cases

How to learn: Supervised machine learning

a «teacher» is giving labeled examples



Given

Examples → $(\mathbf{x}_i; \; y_i), i = 1;...; L$

inputs label

- Classification
- Regression

Output can be probability

Find

- «Best» internal parameters of the system

Learning from labeled examples: minimization and generalization

- A **flexible model f(x;w),** where the flexibility is given by some **tunable parameters** (or weights) **w**

- determination of the best parameters is fully **automated**, this is why the method is called *machine* learning after all

Very flexible models

Learning from labeled examples: minimization and generalization (2)

- fix the free parameters by demanding that the **learned model works (approximately) correctly on the examples in the training set**.

- **power of optimization**:

full clarity about the objective

  - 1. define an **error measure** to be minimized,
  - 2. determine optimal parameters via (automated) **optimization**

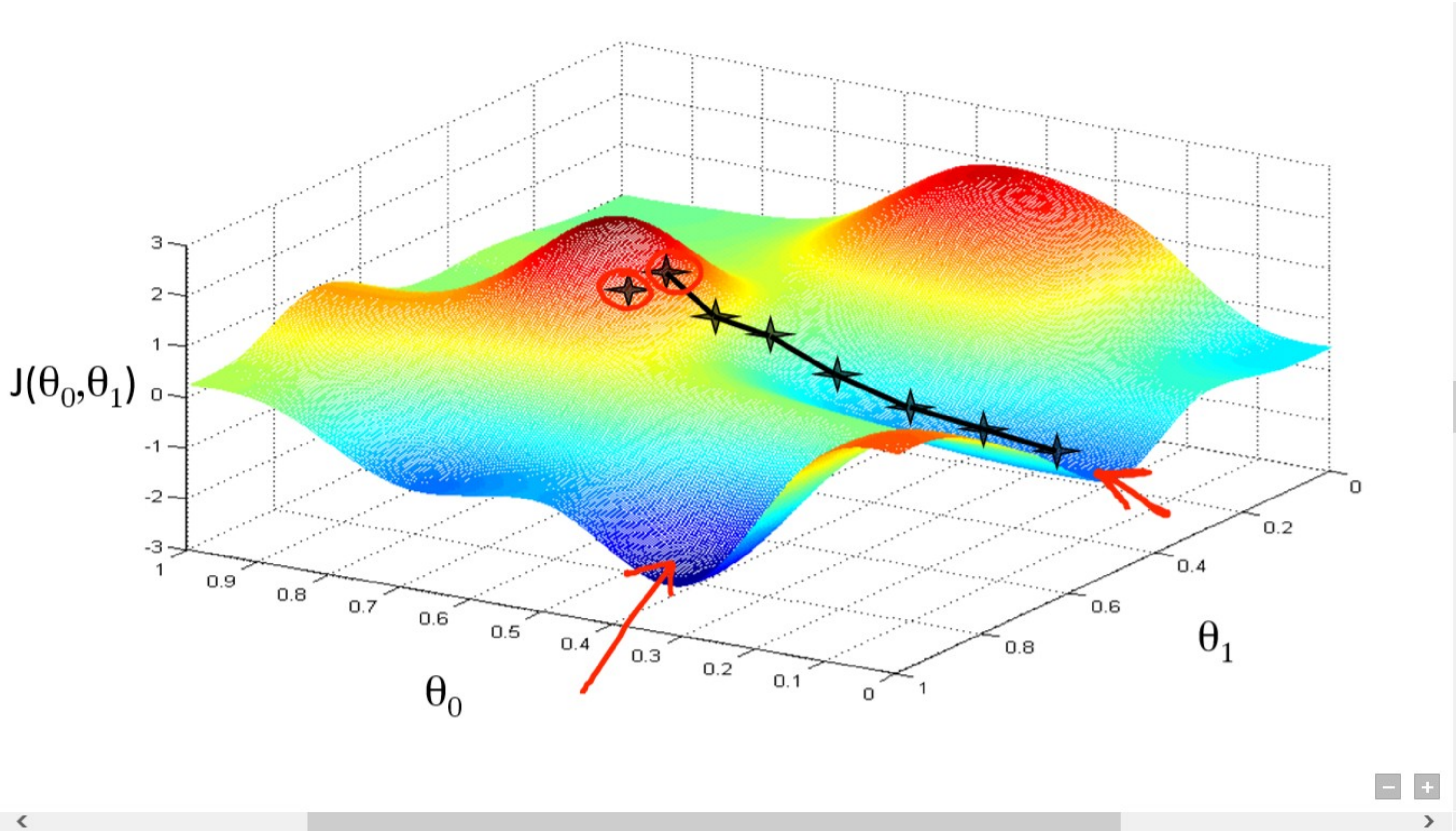
Learning from labeled examples: minimization and generalization (3)

- suitable **error measure** is the **sum of the errors** between the correct answer (given by the example label) and the outcome predicted

- if the function is smooth one can discover points of low altitude by being blindfolded and parachuted to a random initial point…

(gradient descent)

Gradient descent



RMS (root mean square) error function

- Individual errors
- Square
- Average (Sum and divide)
- Square root is optional... (optimizing sum of squares or its square root leads to the same result)

$$RMS = \sqrt{\frac{e_1^2 + e_2^2 + \cdots + e_\ell^2}{\ell}}$$

Error Backpropagation

How do we **learn** optimal MLPs from examples?

1. take a ”guiding” function to be optimized (e.g., sum-of-squared errors on the training examples)

1. Use gradient descent with respect to the weights to find the better and better weights

1. Stop the descent when results on a validation set are best (if over-learning, generalization can worsen). Learning is not an end, but a *means* for generalizing.

Batch backpropagation

skiing

- Given an MLP, define its sum-of-squared-differences energy as:

$$E(w) = \frac{1}{2}\sum_{p=1}^{P} E_p = \frac{1}{2}\sum_{p=1}^{P}(t_p - o_p(w))^2$$

1. Let the initial weights be randomly distributed
2. Calculate the gradient $g_k = \nabla E(w_k)$ Partial derivatives
3. The weights at the next iteration k + 1 are updated as follows

$$w_{k+1} = w_k - \epsilon\, g_k.$$

why small epsilon?

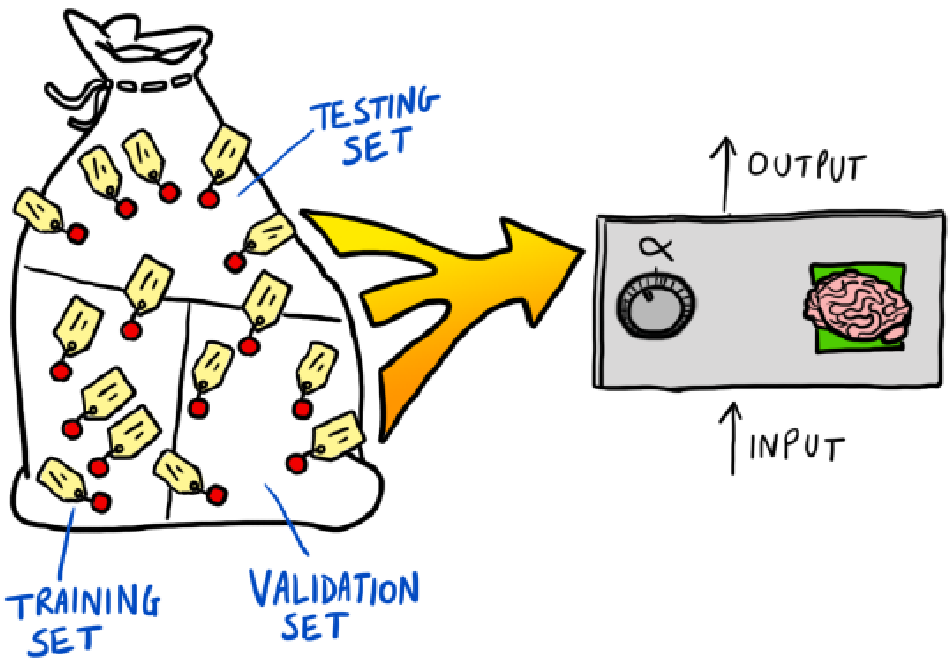
Learn, validate, test!

- careful experimental procedures to measure the effectiveness of the learning process.

- It is a terrible mistake to measure the performance of the learning systems on the same examples used for training

- **The test set is used only once** for a final measure of performance.

Learn, validate, test!



Deep neural networks

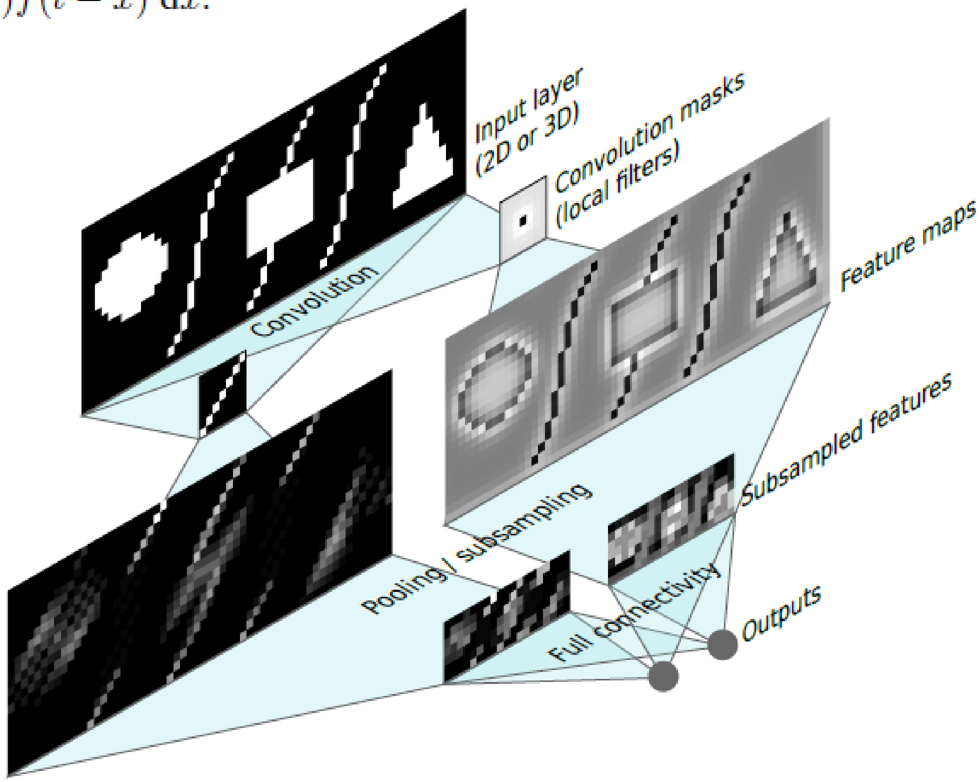
- Some classes of input-output mappings are easier to build if more hidden layers are considered.
- **The dream**: feed examples to an MLP with many hidden layers and have the MLP **automatically develop internal representations** (encoded in the activation patterns of the hidden-layers).

Deep Learning

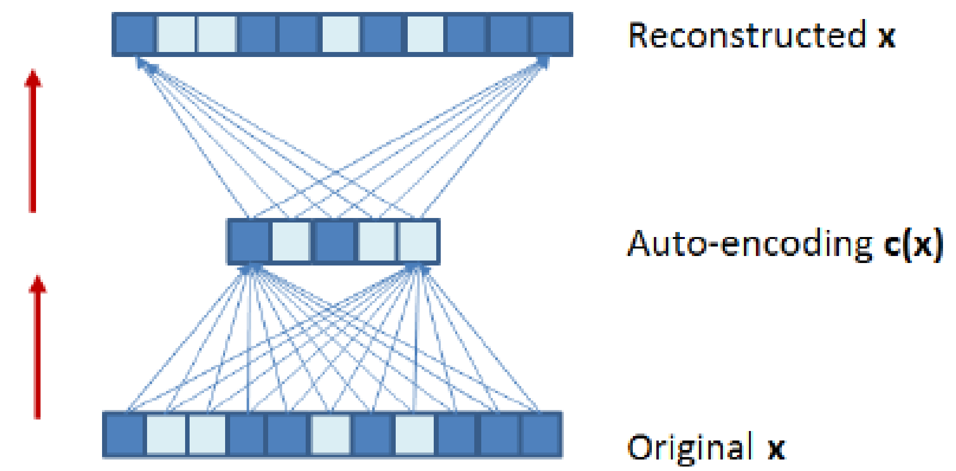
Feature detectors in a frog retina (*Bufo Bufo*) are hard-wired and **specialized to detect a fly at the distance that the frog could strike.**

Deep networks
Convolutional Neural Networks

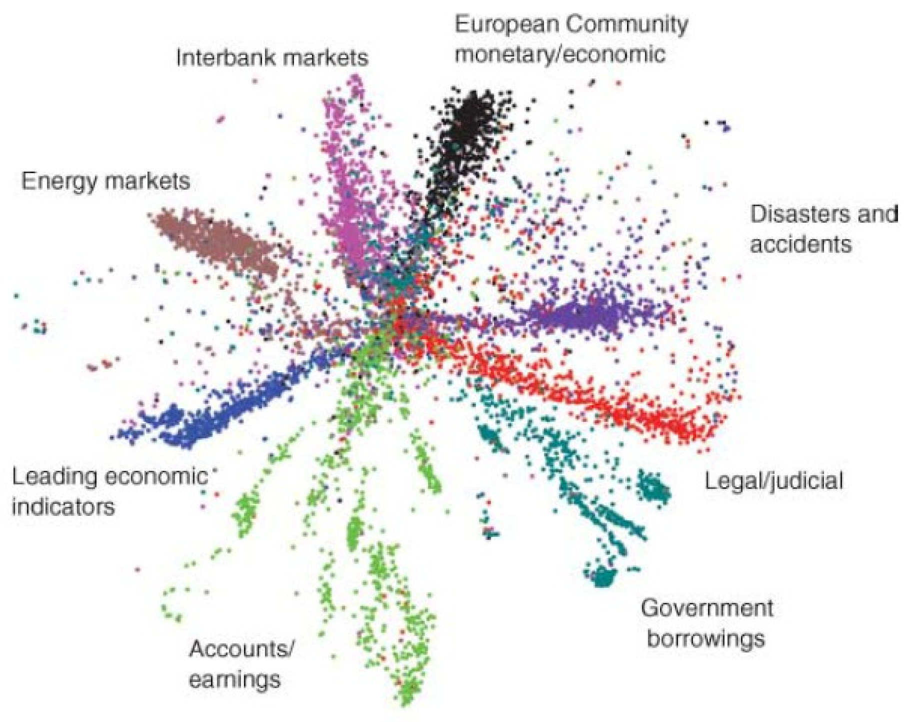
$$s * f(t) = \int_{-\infty}^{+\infty} s(x) f(t-x) \, \mathrm{d}x.$$



Deep networks: Auto-encoders



Deep Networks: Auto-encoders



The codes produced by a 2000- 500-250-125-2 autoencoder on news stories by Reuters. Clusters corresponding to different topics, with different colors, are clearly visible (details in [187]).

Unsupervised learning: What can be learnt *without* teachers and labels?

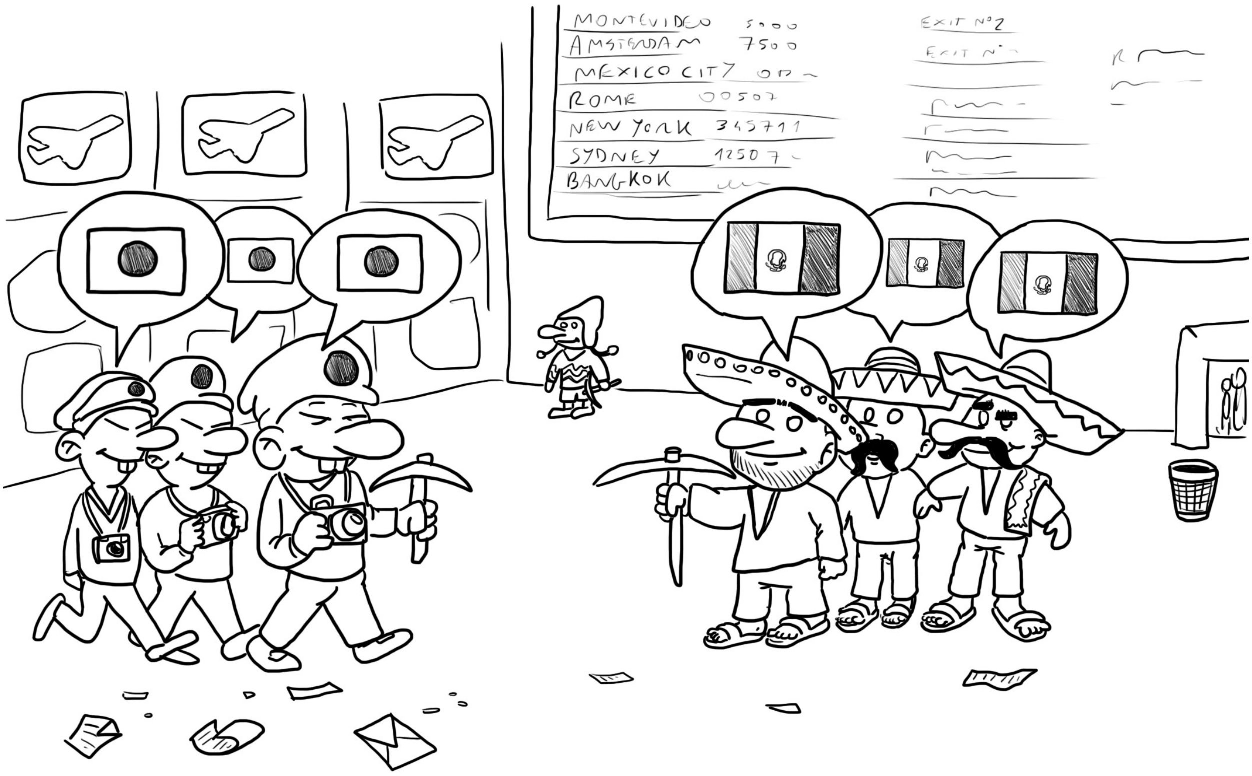
- Modeling and understanding structure is at the basis of our cognitive abilities.
- A name is a way to **group** different experiences so that we can start speaking and reasoning (think about animal species, or continent's names)

First God made heaven and earth. The earth was without form and void, …. And God said, "Let there be light"; and there was light. And God saw that the light was good; and God separated the light from the darkness. God called the light Day, and the darkness he called Night. [. . .] So out of the ground the Lord God formed every beast of the field and every bird of the air, and brought them to the man to see what he would call them; and whatever the man called every living creature, that was its name. The man gave names to all cattle, and to the birds of the air, and to every beast of the field.

(Book of Genesis)

An example

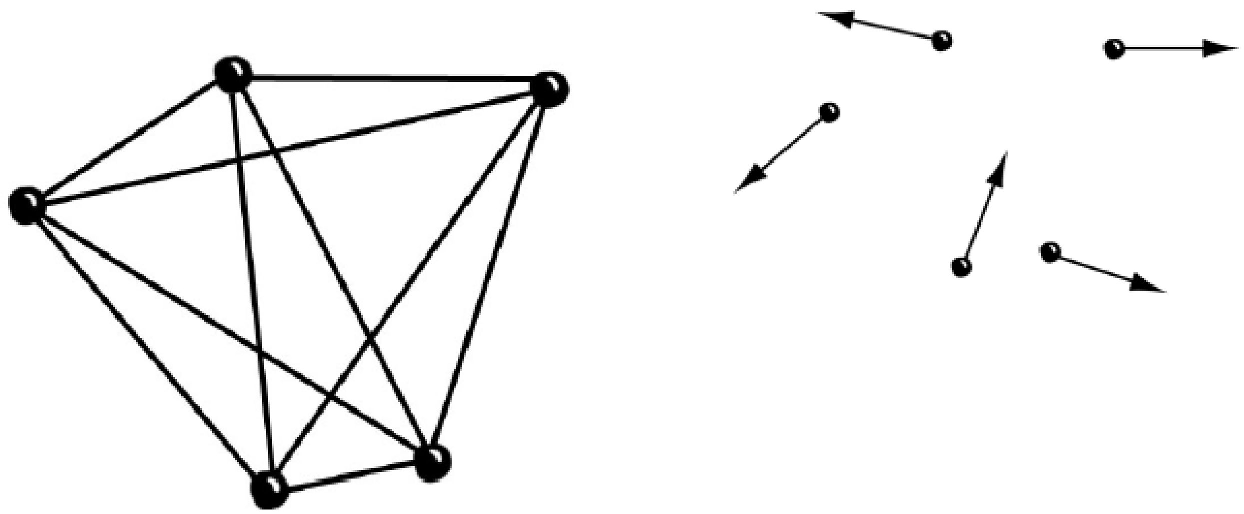
- Clustering different flowers in a meadow without knowing names



Clustering

- Clustering: grouping similar things together, then one can label the groups with names.
- Compression of information (prototypes)
- The prototype **summarizes** the information contained in the subset of cases which it represents
- When similar cases are grouped together, one can reason about groups instead of individual entities.
- Example: marketing segments

Clustering: Representation and metric



External representation by relationships (left) and **internal representation** with coordinates (right). In the first case mutual similarities between pairs are given, in the second case individual vectors.

Clustering: Representation and metric (2)

An **internal representation** is available for each entity, and mutual similarities are derived from it

dissimilarity
$$\delta_E(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \mathbf{x}\| = \sqrt{\sum_{i=1}^{M} (x_i - y_i)^2}.$$

K-means for hard clustering

- **Hard clustering** problem: partition the entities D into k disjoint subsets C = (C_1, … ,C_k) to reach the following **two objectives**:

1. Minimization of the average **intra-cluster dissimilarities**

$$\min \sum_{d_1, d_2 \in C_i} \delta(\mathbf{x}_{d_1}, \mathbf{x}_{d_2}). \qquad \min \sum_{d \in C_i} \delta(\mathbf{x}_d, \mathbf{p}_i).$$

2. Maximization of **inter-cluster distance**

Clustering is a multi-objective optimization task

K-means for hard and soft clustering(2)

- **Divisive algorithms** are very simple clustering algorithms: begin with the whole set and divide it into successively smaller clusters
- For each cluster, its **prototype** is calculated by minimizing the its quantization error:

$$\text{Quantization Error} = \sum_d \|\mathbf{x}_d - \mathbf{p}_{c(d)}\|^2,$$

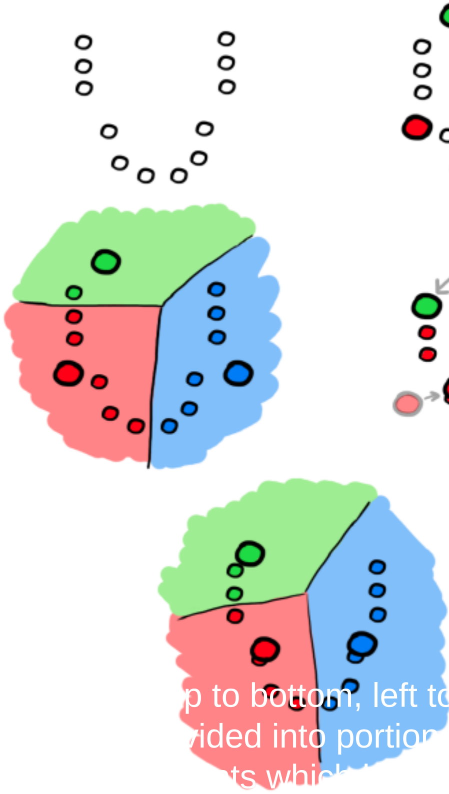
- **k-means clustering** partitions the observations into k clusters, so that each observation belongs to the cluster with the nearest **centroid**

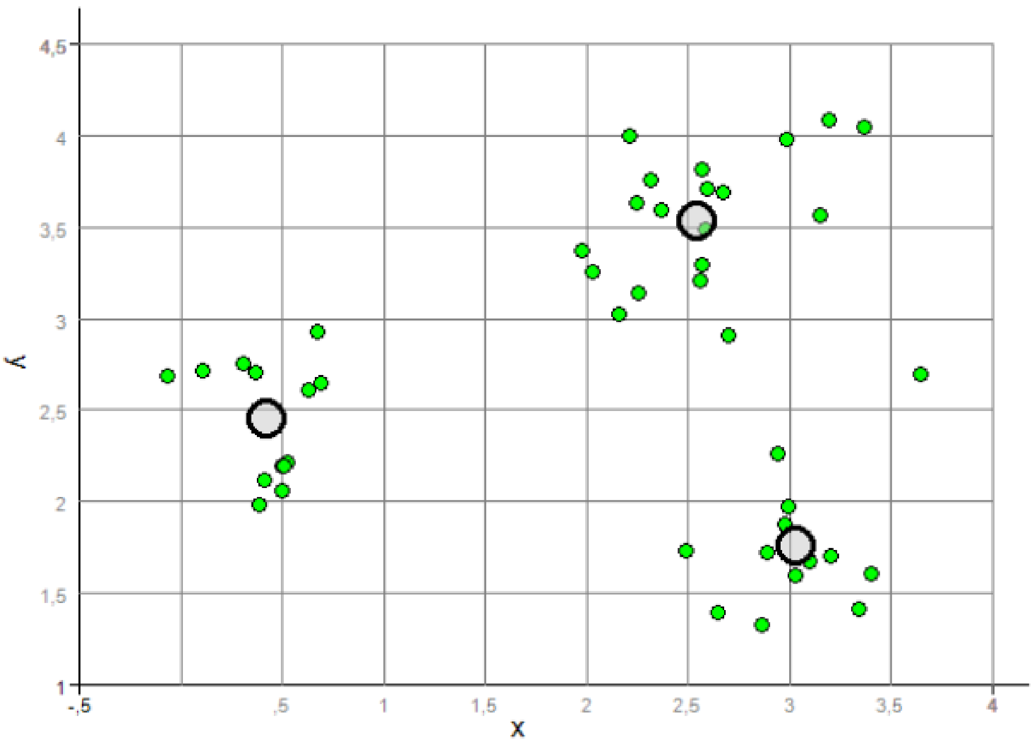
K-means: the algorithm

1. Choose the number of clusters **k**.
2. Randomly generate k clusters and determine the **cluster centroids pc**
3. Repeat the following steps until some convergence criterion is met
   - **Assign** each point x to the nearest cluster centroid
   - **Recompute** the new cluster centroid

$$\mathbf{p}_c \leftarrow \frac{\sum_{\text{entities in cluster } c} \mathbf{x}}{\text{number of entities in cluster } c}.$$

SIMPLE AND FAST!



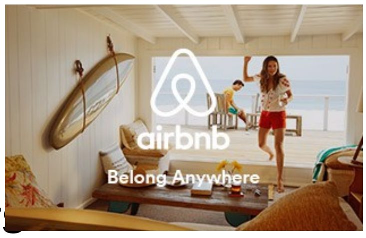


K-means clustering. Individual points and cluster prototypes are shown.

Part 2
The landscape of Intelligent Optimization

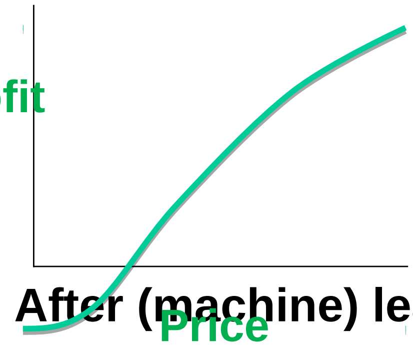
What is the meaning of optimization for you?

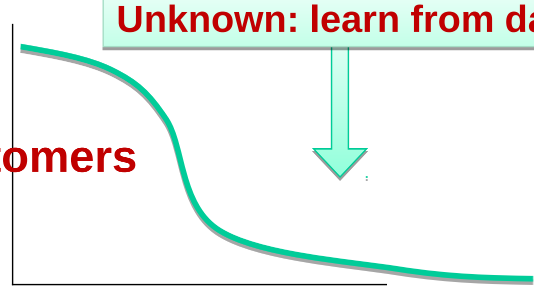
Example: determine the best price



- Profit = price paid – **costs**
- **Probability of accepting offer**
- Actual profit is **multiplication** of the two factors

Unknown: learn from data!





- **After (machine) learning… optimize!**

How to find the minimum

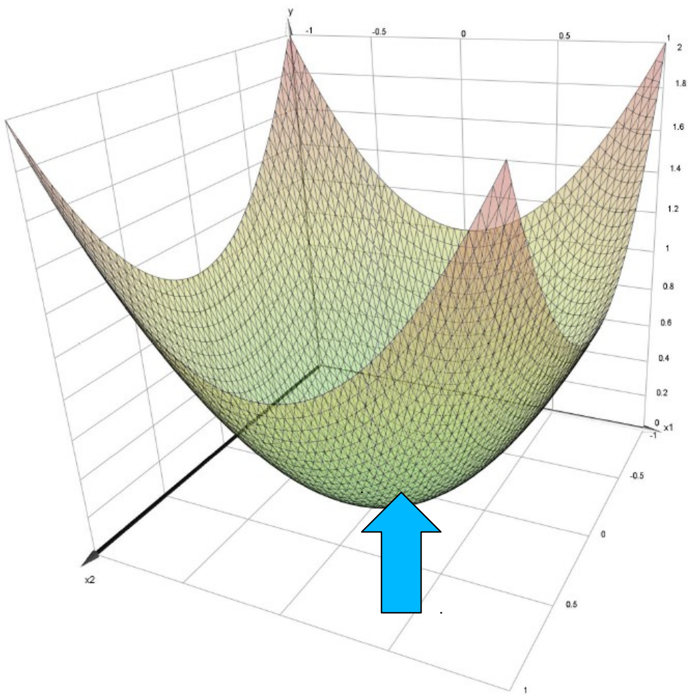


Figure 18.6: Quadratic positive definite f of two variables.

One sees it…

Try many (x,y) values…

Which values?
All possible vals?

“Local steps”

Global Optimization Problem

Global optimization problem:

$$\begin{aligned} \text{Given} \quad & f : A \to \mathbb{R} \\ \text{find} \quad & x^* \in A \\ \text{such that} \quad & f(x^*) \leq f(x) \text{ for every } x \in A. \end{aligned}$$

- X* satisfying above is called a global optimum
- **record value** (the best-so-far value) at iteration n $\hat{y}_n = \min_{i=1,\ldots,n} f(x_i),$

Two paradigmatic methods

Optimization is a very old topic…

Operations research

- **Stochastic global optimization** (memory-less, “brute force”, but very robust)
- **Local Search** and **Reactive Search Optimization** (use learning while optimizing)

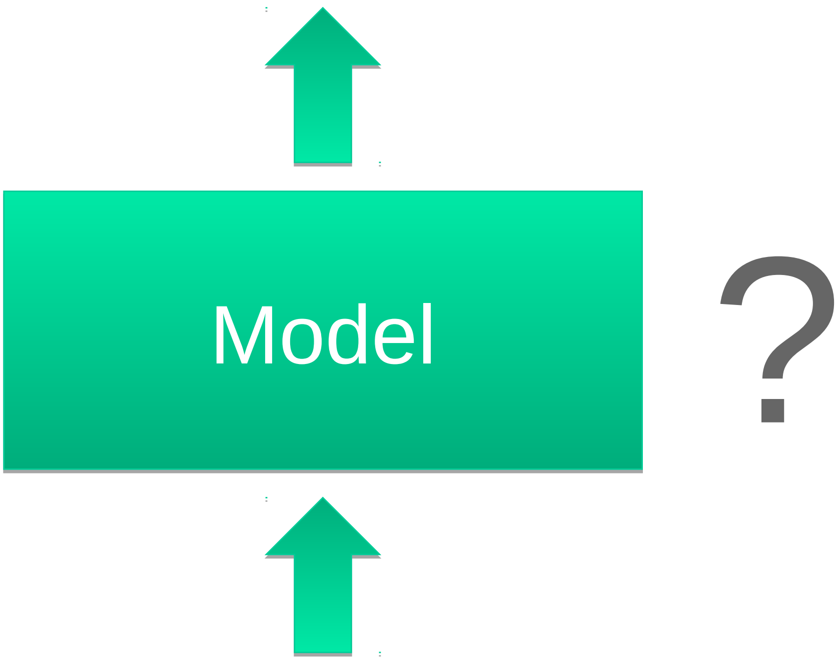
Paradigm1: Stochastic Global Optimization

Stochastic Global Optimization

- **black-box interface**: the algorithm can query the value f(x) for a sample point x, but it cannot "look inside" f
- **separation of concerns**: to be as generally applicable as possible, optimization routines do not need to know anything about the application domain;
- a computer scientist can improve profits for a financial institution or improve survivability of patients cured for cancer **without any knowledge** of economics or medicine.

Ignorance **can** bring value

Black-box optimization



John Von Neumann

“The **sciences do not try to explain**, they hardly even try to interpret, they mainly make models. By a **model** is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that **it is expected to work** - that is correctly to describe phenomena from a reasonably wide area. Furthermore, it must satisfy certain esthetic criteria - that is, in relation to how much it describes, it must be rather simple.”

Apples fall because they fall

Apple fall because of the law of gravitation

Stochastic Global Optimization

- just function evaluations
- **function of continuous (real) variables**
- one can decide where to place sample points, and one can use the information obtained to build internal models of the function and tune its own meta-parameters.
- stochasticity in the generation of sample points helps to improve robustness and avoid that some false initial assumptions lead to low-quality local optima

Convergence Rate of **Pure Random Search**

- Success with probability (1 – γ)
- In the asymptotic behavior when *d* is fixed and

$\epsilon \to 0$,number of iteration for success $n_* = O\left(\frac{1}{\epsilon^d}\right)$

- **Curse of dimensionality**

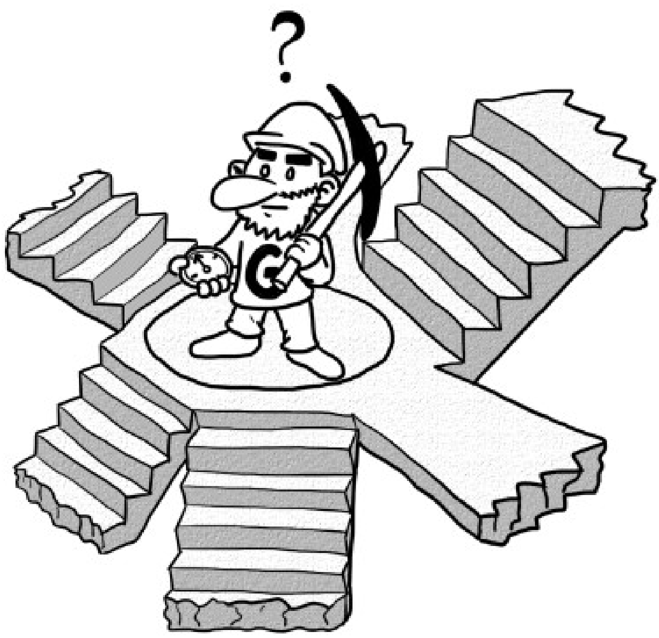
d	$\gamma = 0.1$			$\gamma = 0.05$		
	$\varepsilon = 0.5$	$\varepsilon = 0.2$	$\varepsilon = 0.1$	$\varepsilon = 0.5$	$\varepsilon = 0.2$	$\varepsilon = 0.1$
1	0	5	11	0	6	14
2	2	18	73	2	23	94
3	4	68	549	5	88	714
4	7	291	4665	9	378	6070
5	13	1366	43743	17	1788	56911
7	62	38073	$4.9 \cdot 10^6$	80	49534	$6.3 \cdot 10^6$
10	924	$8.8 \cdot 10^6$	$9.0 \cdot 10^9$	1202	$1.1 \cdot 10^7$	$1.2 \cdot 10^{10}$
20	$9.4 \cdot 10^7$	$8.5 \cdot 10^{15}$	$8.9 \cdot 10^{21}$	$1.2 \cdot 10^8$	$1.1 \cdot 10^{16}$	$1.2 \cdot 10^{22}$
50	$1.5 \cdot 10^{28}$	$1.2 \cdot 10^{48}$	$1.3 \cdot 10^{63}$	$1.9 \cdot 10^{28}$	$1.5 \cdot 10^{48}$	$1.7 \cdot 10^{63}$
100	$1.2 \cdot 10^{70}$	$7.7 \cdot 10^{109}$	$9.7 \cdot 10^{139}$	$1.6 \cdot 10^{70}$	$1.0 \cdot 10^{110}$	$1.3 \cdot 10^{140}$

Table 2.1. Values of $n_* = n_*(\gamma, \varepsilon, d)$, see (2.22), for vol$(A) = 1$, $\gamma = 0.1$ and 0.05, $\varepsilon = 0.5, 0.2$ and 0.1, for various d.

Curse of dimensionality

- "Abandon all hope, you who enter here". If dimension is large there is **no magic algorithm to rapidly approximate the global optimum for a generic function in less than exponential number of iterations**.
- There are just too many places to hide in d dimensions.
- Hope is related to **functions with special forms, so that regularities can be learnt** from an initial sampling, albeit in approximated form, and used to identify shortcuts leading rapidly to close approximations of the optimal solution (learning x optimization)
- Chance that we encounter highly-structured functions in real applications? Not negligible. **Nature doe not play dice…**
- convergence is only a theoretical fiddle

Paradigm2: Local Search and Reactive Search Optimization (RSO)



Brute force is not the solution

- Let's assume that one has to find the minimum of a discrete (combinatorial) optimization problem (for example, think about the *travelling salesman* problem)
- Evaluating all possible combinations of inputs can be computationally impossible
- One needs to resort to clever techniques to solve these problems

Local search based on perturbations

- starting from an initial tentative solution

- try to improve it through repeated small changes

- stop when no improving local change exists

(local optimum, or locally optimal point)

Local search optimization: notation

- χ is the search space
- $X^{(t)}$ is the current solution at iteration t.
- $N(X^{(t)})$ is the neighborhood of point $X^{(t)}$, obtained by applying a set of basic moves μ_0, …, μ_M to the current configuration

$$N(X^{(t)}) = \{X \in \mathcal{X} \text{ such that } X = \mu_i(X^{(t)}), i = 0, \ldots, M\}.$$

Local search optimization

- Local search starts from an admissible configuration $X^{(0)}$ and builds a trajectory $X^{(0)}$, …,$X^{(t+1)}$.
- The successor of the current point is constructed as follows

$$Y \leftarrow \textsc{Improving-Neighbor}(\ N(X^{(t)})\ )$$

$$X^{(t+1)} = \begin{cases} Y & \text{if } f(Y) < f(X^{(t)}) \\ X^{(t)} & \text{otherwise (search stops).} \end{cases}$$

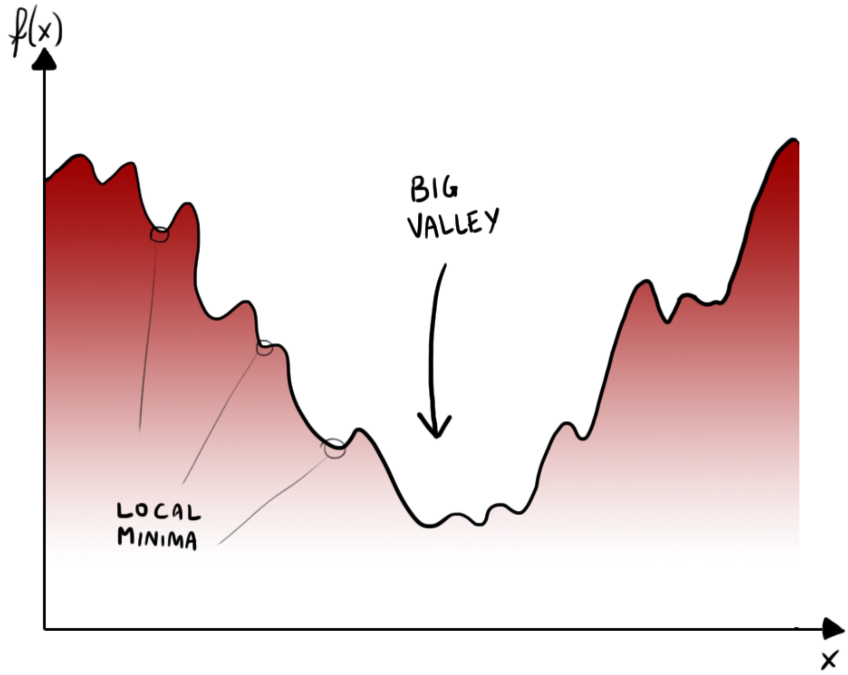
- IMPROVING -NEIGHBOR returns an improving element in the neighborhood

Local optima are not always global optima

- For many optimization problems, a closer approximation to the global optimum is required
- More complex search schemes have to be adopted to balance in an optimal way exploration and exploitation

Attraction basins

- Local minima tend to be clustered (good local minima tend to be closer to other good minima)
- The attraction basin associated with a local optimum is the set of points X which are mapped to the given local optimum by the local search trajectory
- if local search stops at a local minimum, kicking the system to a close attraction basin can be much more effective than restarting from a random configuration


f(x)
BIG
VALLEY
LOCAL
MINIMA
x

Modifications of local search based on perturbations

- local search by small perturbations is an effective technique but additional ingredients are in certain cases needed to obtain superior results

Myhts and building blocks

[341] Kenneth S¨orensen. Metaheuristics—the metaphor exposed. International Transactions in Operational Research,22(1):3–18, 2015.

In recent years, the field of combinatorial optimization has witnessed a true tsunami of “novel” metaheuristic methods, most of them based on **a metaphor of some natural or man-made process**. The behavior of virtually any species of insects, the flow of water, musicians playing together – it seems that no idea is too far-fetched to serve as inspiration to launch yet another metaheuristic. In this paper, we will argue that this line of research is threatening to lead the area of metaheuristics away from scientific rigor. ….

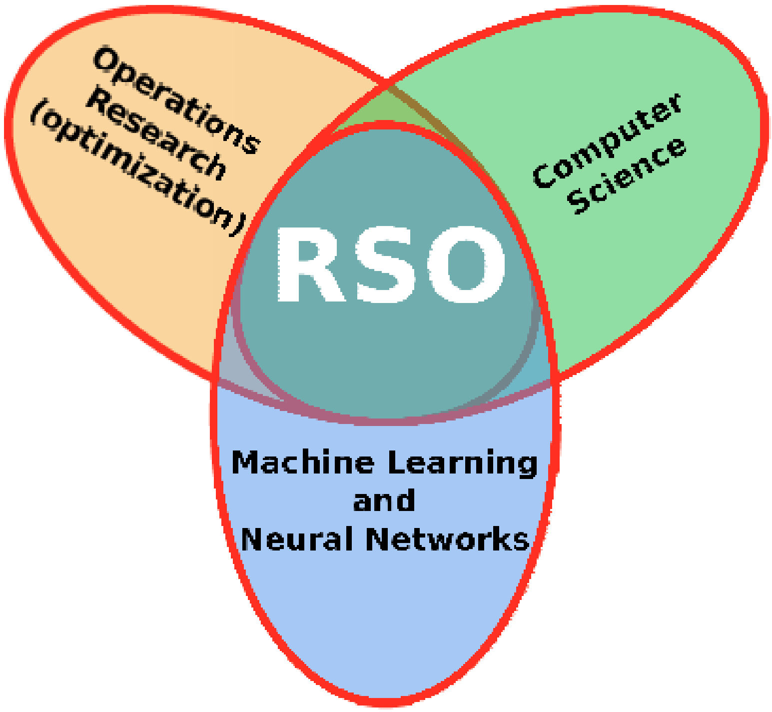
"It is a good morning exercise for a research scientist to discard a pet hypothesis every day before breakfast: it keeps him young" (Konrad Lorenz, 1903-1989).

Reactive Search Optimization (RSO): **Learning while searching**

- Many problem-solving methods are characterized by a certain number of choices and free parameters, usually manually tuned.
- Parameter tuning can be automated as a part of the optimization algorithm
- This leads to self-contained, fully automated algorithms, independent from human intervention

Reactive Search Optimization (RSO) integrates online machine learning techniques and search heuristics for solving complex optimization problems.

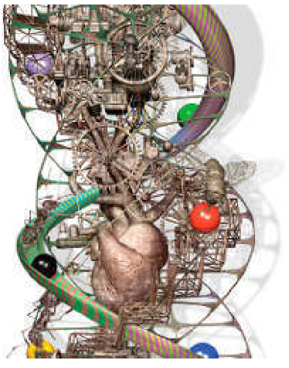
Reactive Search Optimization (RSO):



Reactive Search Optimization

integration of online machine learning techniques for local search heuristics.

The word ***reactive*** hints at a ready response to events *during* the search through an internal online feedback loop for the *self-tuning* of critical parameters.



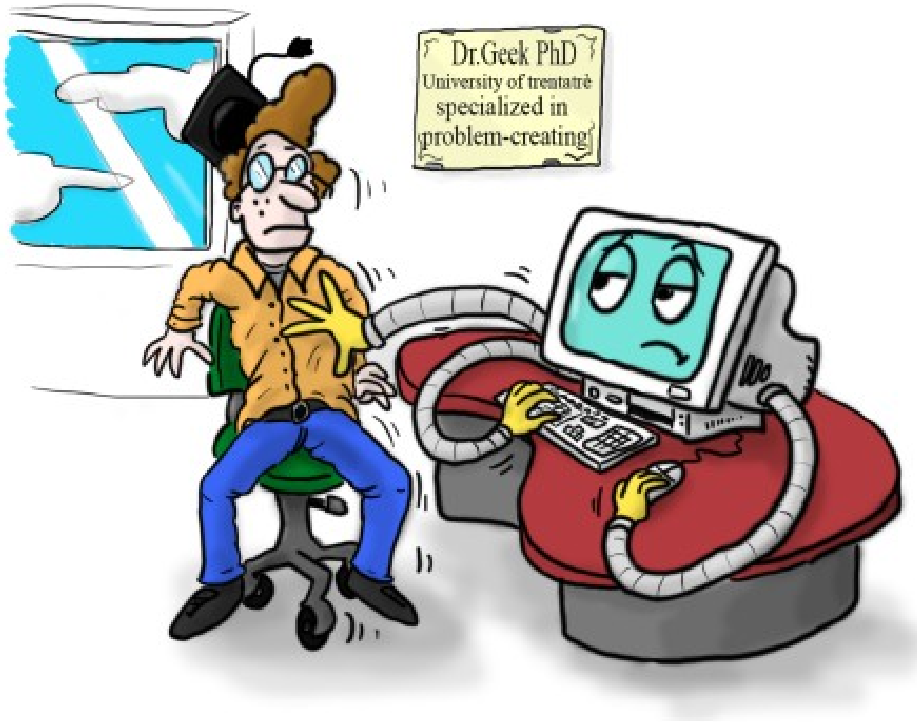
Biological systems are highly adaptive; they use signals coming in from receptors and sensors to fine-tune their functioning. Adaptivity is a facet of the **reactivity** of such systems.

Reactive Search Optimization

- RSO can be applied to systems that require to set some operating parameters to improve its functionality.
- A simple loop is performed: set the parameters, observe the outcome, then change the parameters in a strategic and intelligent manner until a suitable solution is identified
- In order to operate efficiently, RSO uses memory and intelligence to improve solutions in a directed and focused manner

Reactive Search Optimization

- While many alternative solutions are tested in the exploration of a search space, patterns and regularities appear
- The human brain quickly learns and drives future decisions based on previous observations.
- This is the main inspiration source for inserting online machine learning techniques into the optimization engine of RSO



RSO based on prohibitions: tabu search

- Basic idea: using prohibitions to encourage diversification

How?

- While constructing a trajectory for local minima search, every time a move is applied, the inverse move is temporarily prohibited

Tabu search: an example

- Let $\chi=\{0,1\}^L$
- The neighborhood is obtained by applying the elementary moves μ_i, ($i = 1,\ldots,L$) that change the i-th bit of the string $X = [x_1,\ldots, x_i,\ldots, x_L]$
- At each step, the selected move is the one that minimizes the target f in the neighborhood even if f increases, to exit from local minima.
- As soon as a move is applied, **the inverse move is temporarily prohibited**

Prohibition and diversification

- Let $H(X, Y)$ be the Hamming distance between two strings X and Y
- if only allowed moves are executed, and T satisfies $T < (n - 2)$ (at least two moves are allowed at each iteration), then

- The Hamming distance H between a starting point and successive points along the trajectory is strictly increasing for $T + 1$ steps:

$$H(X^{(t+\tau)}, X^{(t)}) = \tau \quad \text{for} \quad \tau \leq T + 1.$$

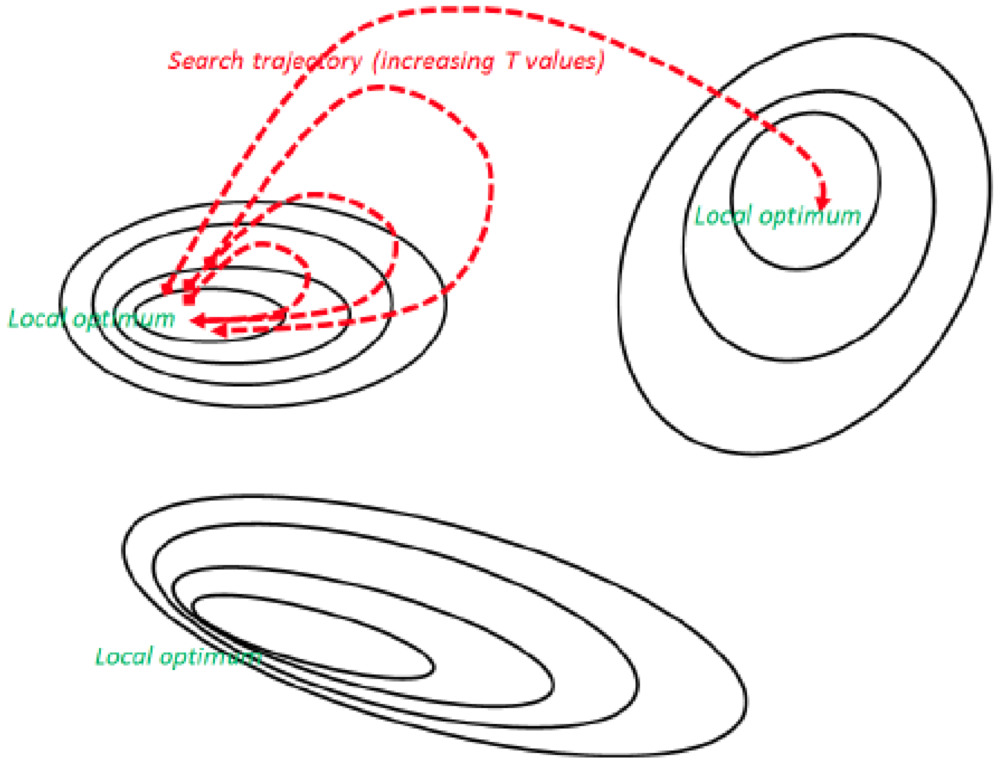
- The minimum repetition interval R along the trajectory is $2(T + 1)$:

$$X^{(t+R)} = X^{(t)} \Rightarrow R \geq 2(T + 1).$$

	Iteration t	$X^{(t)}$								$f(X^{(t)})$	$H(X^{(t)},X^{(0)})$
	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	1	1	1
	2	0	0	0	0	0	0	1	1	3	2
	3	0	0	0	0	0	1	1	1	7	3
$T+1$ →	4	0	0	0	0	1	1	1	1	15	4
	5	0	0	0	0	1	1	1	0	14	3
	6	0	0	0	0	1	1	0	0	12	2
	7	0	0	0	0	1	0	0	0	8	1
$2(T+1)$ →	8	0	0	0	0	0	0	0	0	0	0

Tuning the T parameter

- The parameter T should be tailored to the specific problem
- BUT the choice of a **fixed T** without a priori knowledge is difficult
- RSO uses a simple mechanism to change T during the search so that the value $T^{(t)}$ is appropriate to the local structure of the problem
- RSO determines the minimal prohibition value which is sufficient to escape from an attraction basin around a minimizer



RSO for tabu search

- T is equal to one at the beginning
- T **increases** if the trajectory is trapped in an attraction basin
- T **decreases** if unexplored search regions are visited, leading to different local optima

RSO: conclusions

- If the problem has a single local optimum the power of RSO is not needed, although not dangerous
- Most real-world problems are infested with many locally optimal points
- RSO is crucial to **transform a local search building block into an effective and efficient solver**.
- RSO with prohibitions has been used for problems ranging from combinatorial optimization to the minimization of continuous functions and to sub-symbolic machine learning tasks

Part 3
Disruptive innovation by
combining ML + IO
(“automated creativity”)

Optimization: a tremendous power

Tapping and musik

- Still largely unexploited in most real-world contexts: standard optimization assumes a **function f(x)** to be minimized, …and **math** knowledge.
- function f(x) (a.k.a “model”) helps people to **concentrate on goals/objectives**, not on algorithms (on policies not on processes)
- BUT static f(x) does not exist in explicit form or is extremely difficult and costly to build by hand, and math knowledge is scarce.

Try asking an hotel manager

A practical view of a «function»

Rating

Model

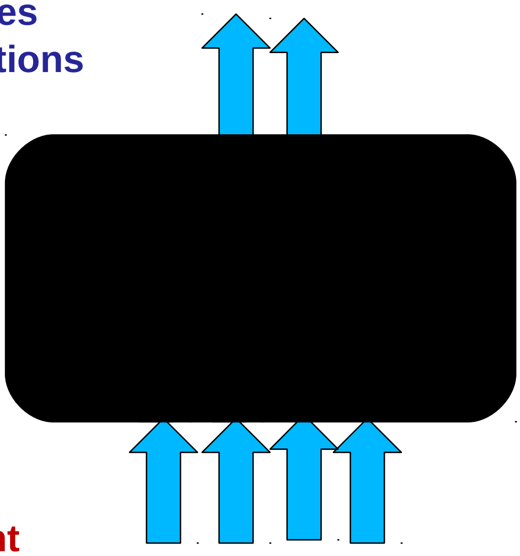
(Accommodation offer, Tourist)

Real word is dirty (black?)

Some posivite objectives (MOOP)
Combination not clear
Hidden objectives
Dynamic aspirations

Learn !

No math formula
Maybe some
high-level
knowledge
and intuition



Learn !

Many inputs,
noisy,
some irrelevant

Learn !

Machine Learning !

Optimization: a tremendous power

- Machine learning: learn f(x) from data (including from user feedback)

- **Learning and Intelligent Optimization (LION):** <u>**machine learning from data for optimization**</u> which can be applied to complex, noisy, dynamic contexts.
- **ML to approximate f(x)** but also **to guide opt. process via *self-tuning*, both *offline* and *online***

- **Autonomy: more power directly in the hands of businesses**

Optimization → for Machine Learning

Source of power

Flexible model (with parameters **w**) How to pick **w?**

ErrorFunction E(w)

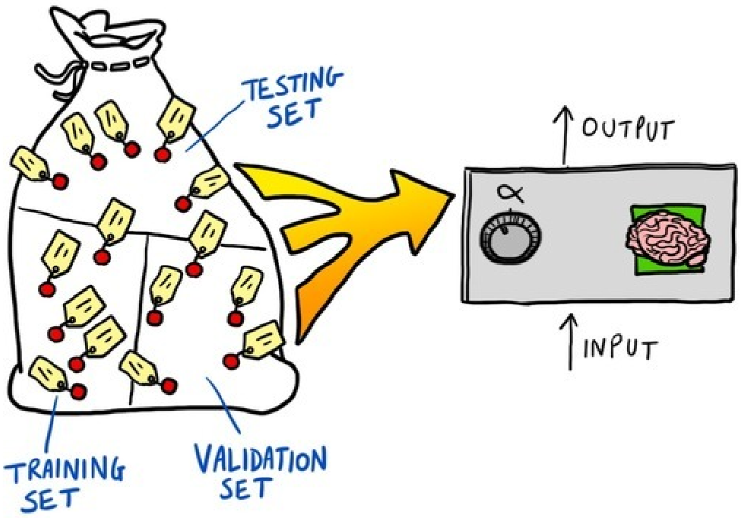
Learn by minimizing E(w) on training examples

...generalization

complicates a bit

MLP and Backpropagation

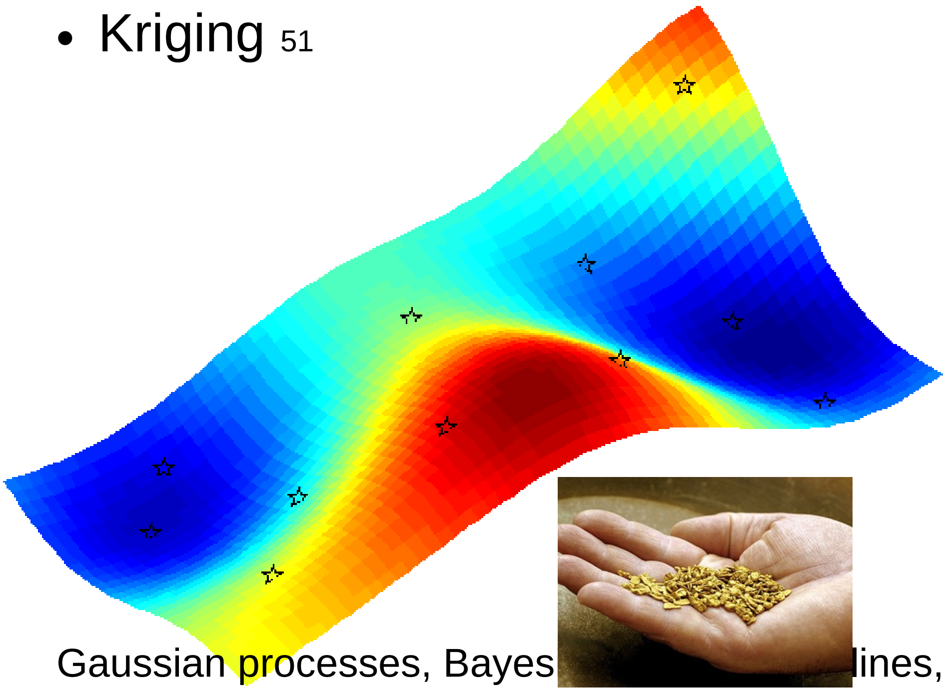
SVM ...



Machine Learning → for Optimization

Practical optimization is coslty ...f(x)

- Kriging 51



Danie Gerhardus Krige
(26 August 1919 – 3 March 2013)

Gaussian processes, Bayes … lines, local models in continous optimization....

Angela Kunoth: «adaptive multi-scale»

If f(x) not given? Learn *what* to optimize

Example: MOP: Finding a partner: *intelligence* versus *beauty*

How many IQ points for one less beauty point?

Is beauty more important than intelligence for you? By how much?

Effective optimization
as iterative process with learning

Pareto-optimality

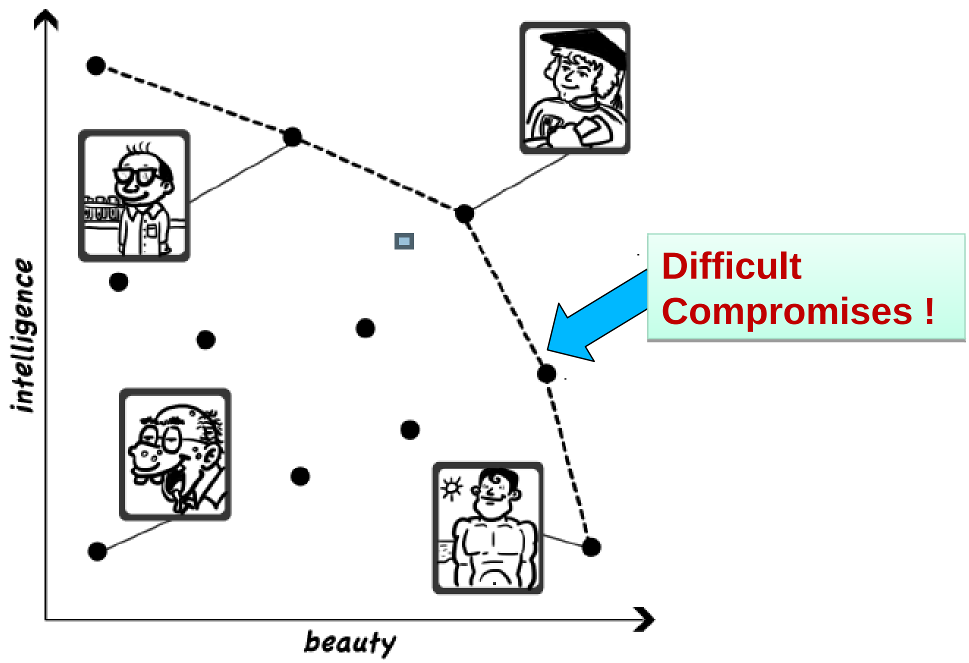


Figure 41.3: Pareto optimality. All dominated points like the persons in the middle are not considered as potential candidates for the final choice. On the Pareto frontier, shown with a dashed line, tradeoffs need to be considered.

Many hot issues are MOOPs

- **Energy production** (best mix…nuclear, oil, wind, solar)
  - Objectives: Cost / safety / pollution
- **Transportation** (cars, trains, roads, metro, taxi, uber, …)
  - Objectives: Energy consumption / speed / safety
- **Healthcare** (prevention, cure, cancer, explosion of costs..)
  - Objectives: Money / age / overall quality of life / priorities

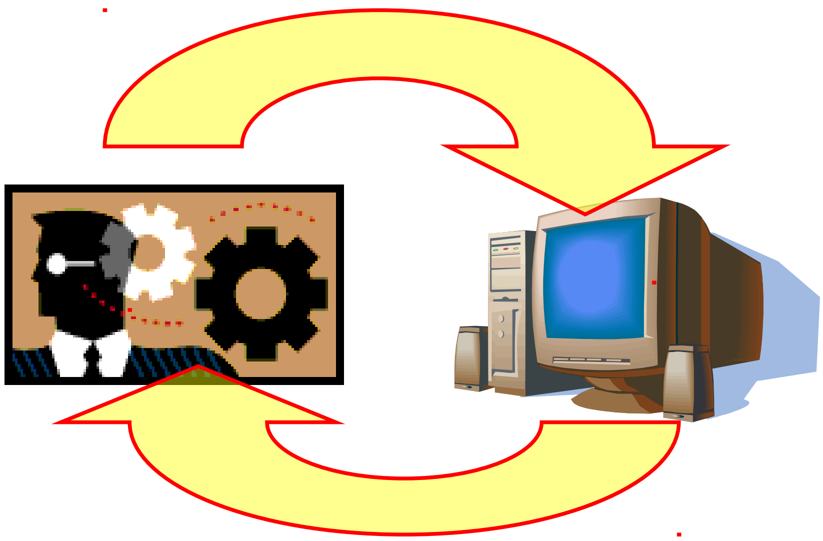
Pareto-optimality (dealing with tradeoffs) has **a huge educational impact to avoid extremism**, fanaticism, radicalism

Compromises are a necessity

Flexible and interactive decision support and problem solving

Crucial decisions depend on factors and priorities which are not always easy to describe before.

Feedback from the user in the exploration phase!

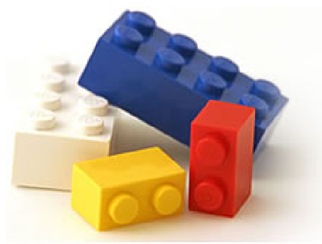


Multiobjective optimization

intermediate (classical) case of missing knowledge:

some criteria are given f1(x) f2(x) … fk(x)

but not easily **combined** into a single f(x)



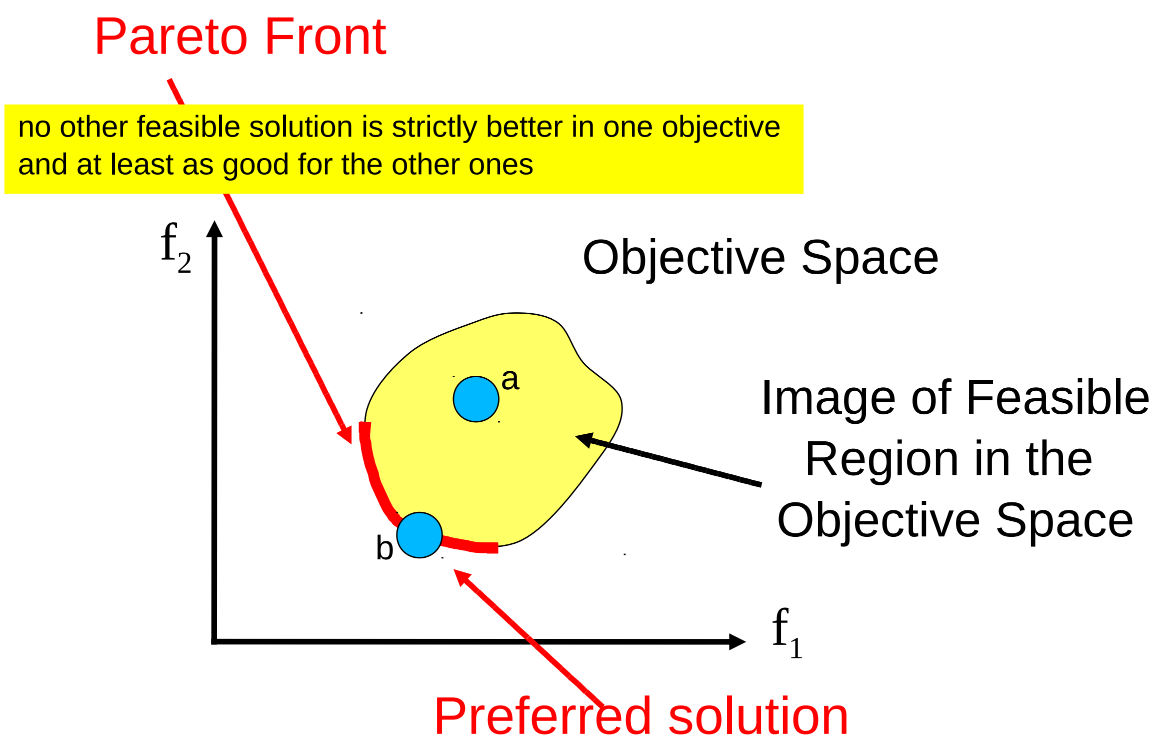
…provide efficient vector solutions (f1,…,fk)

leave to the user the possibility to *decide*

(and to *learn* about possibilities and “real”

objectives, even if not formalized)

Efficient frontier (PF)



Interactive methods

- Solutions generation phases alternated to solution evaluation phases requiring user interaction
- Effective approach
  - Only a subset of the Pareto optimal set has to be generated and evaluated by the DM
  - The DM drives the search process
  - The DM gets to know the problem better (learning on the job)

Conclusions

- **business innovation now is:**
  - **machine learning + intelligent optimization**

- most traditional business are bound to disappear…
- **the new context requires humility** (ask for help by non-experts of your field!)

Interesting area for young and open-minded researchers, challenging problems still ahead!

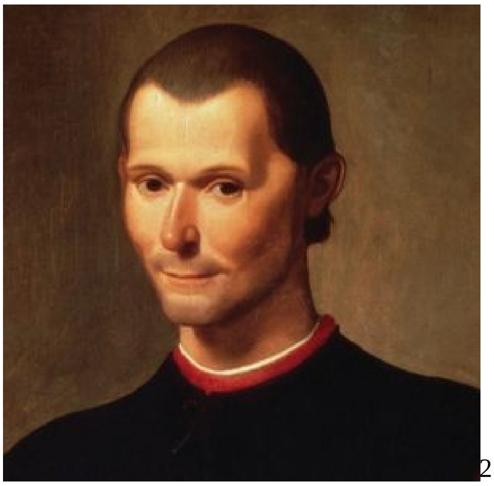
Nerds are conquering the world!

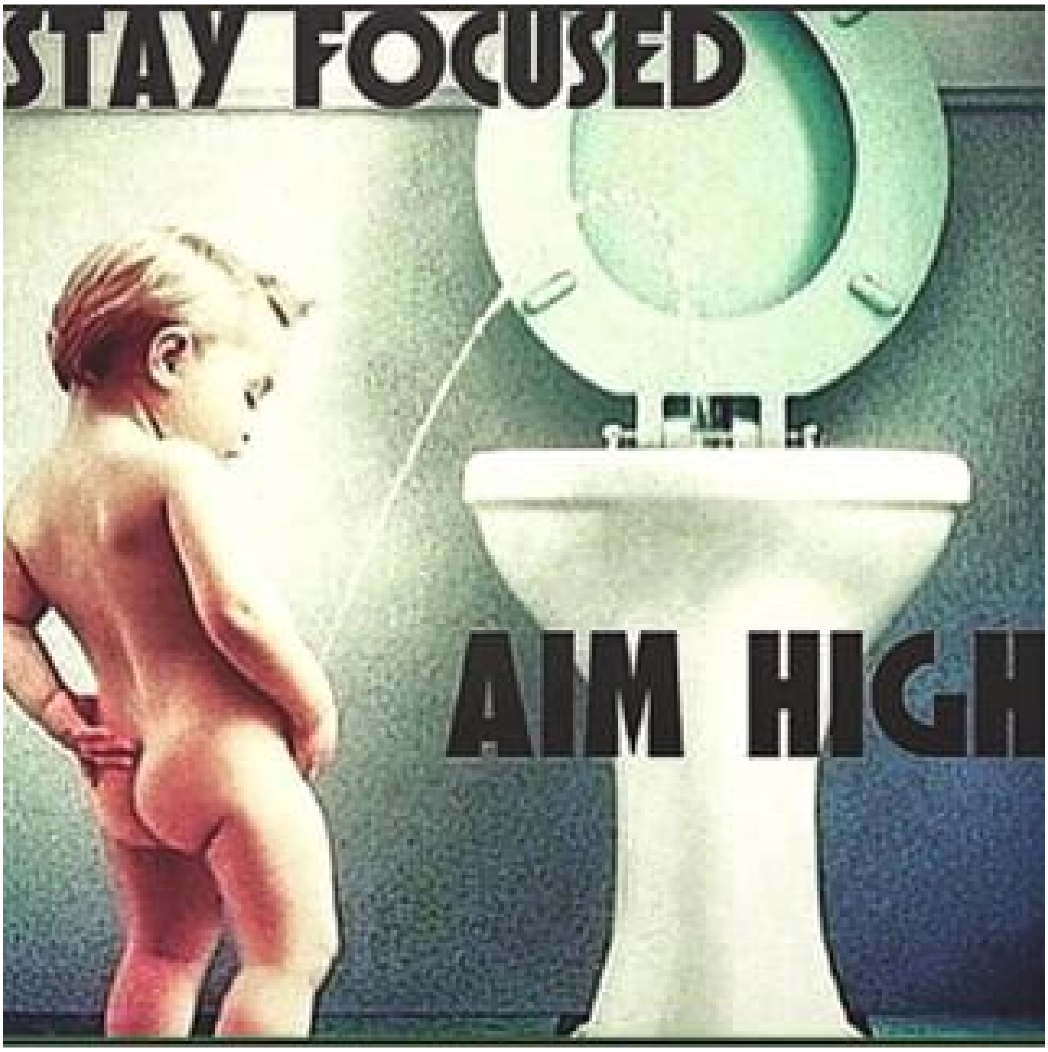
Aim high

Act like the **clever archers** (*arcieri prudenti)* who, designing to hit the mark which yet appears too far distant, and knowing the limits to which the strength of their bow attains, **take aim much higher than the mark**, not to reach by their strength or arrow to so great a height, but to be able with the aid of so high an aim to **hit the mark they wish to reach.**

Niccolò Machiavelli ,

The Prince, c.a. 1500





Thank you