Music Machine Learning

I – Introduction

Master ATIAM - Informatique

Philippe Esling (esling@ircam.fr)

Maître de conférences – UPMC

Equipe représentations musicales (IRCAM, Paris)

UPMC SORBONNE UNIVERSITÉS master ATIAM ircam Centre Pompidou

Artificial intelligence

- **Artificial *intelligence* ?**
- Requires to first understand what intelligence is !
- Must have something to do with ***thinking***
- But in a broader sense, might also be about *perception* and *action*
- In terms of philosophy, we would talk about problems involving these
- However, we *try to understand the mechanisms of our thinking itself*
- *So here, we seek* ***models*** targeted at thinking, perception and actions
- Hence, here we will try to construct a *model of thinking*

This might even help you in understanding your own thinking process ☺

Models targeted at
Thinking, perception and action

Artificial intelligence

- **Thinking**
- Creation ex-nihilo can not really exist …

« *Man can only desire what he has already perceived* »
William Blake

Intelligence: « the capacity to assemble two ideas that seemed heterogeneous »

- We exist (morphologically) since ~200 kYears, and around ~50 kY …

« *Take two concepts and create a third one without impairing the two first* »
Noam Chomsky

- So thinking is finding similarity, discriminating and seeing common patterns

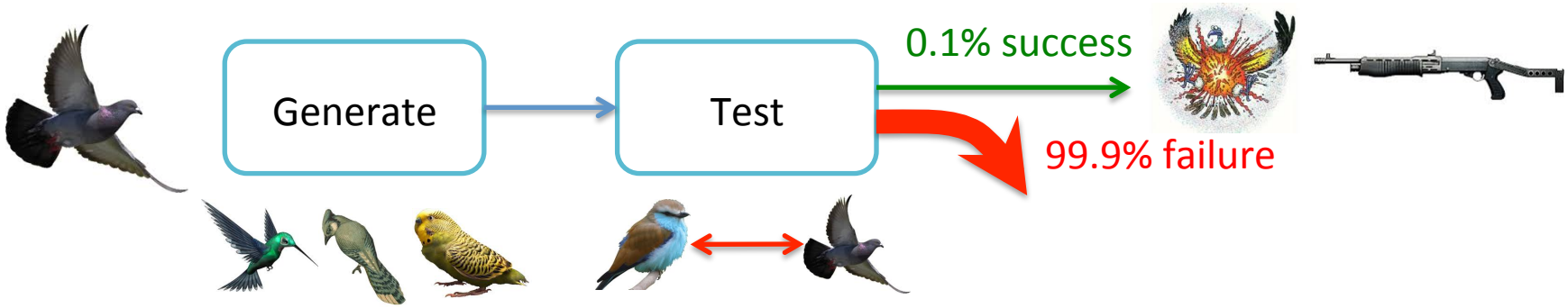
NB: *Frames of Mind: the Theory of multiple intelligence* – **Howard Gardner**
(logico-mathematic, spatial, inter-personal, corporal, linguistic, intra-personnal, musical, ecologist, existential)

- In order to have a model, you need a *representation*
- *A representation that supports a model that allows to understand thinking*

Representations supporting
Models targeted at
Thinking, perception and action

Artificial intelligence

- The typical example of **pattern recognition**
- Generate and test method (cf. trying to find a bird from a book)



Once you have a name for it, you can start talking about it, you have **power over it**

- **Symbolic labels give us power over concepts**

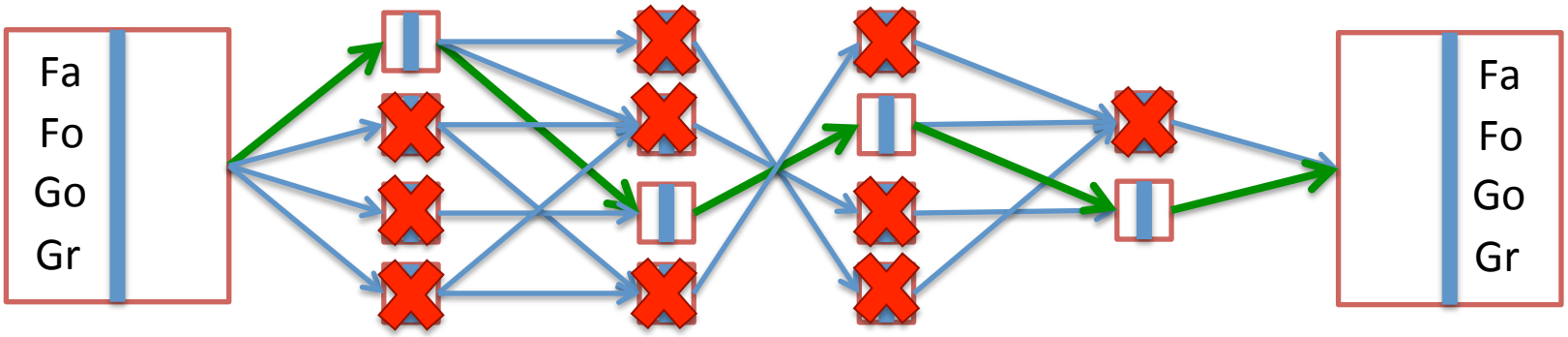
The Rumpelstiltskin tale

« *Once you can name something you have power over it* »

Representations supporting
Models targeted at
Thinking, perception and action

Artificial intelligence

- The right representation? Exemple of the farmer, fox, goose and grain

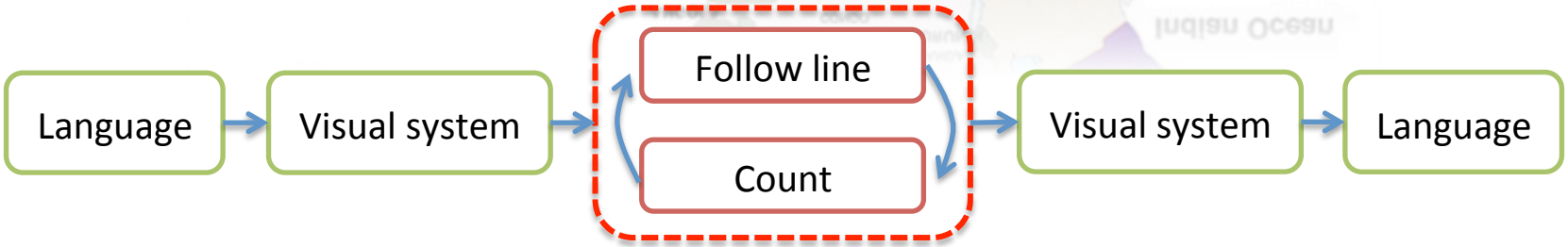


- Normally 16 possibles, but some are impossible (someone gets eaten)
- By drawing as the *right representation* (a graph), it **exposes the constraints**

Constraints exposed by
Representations supporting
Models targeted at
Thinking, perception and action

Artificial intelligence

How many countries are crossed by the equator in Africa ?



Algorithms enabled by
Constraints exposed by
Representations supporting
Models targeted at
Thinking, perception and action

Artificial intelligence

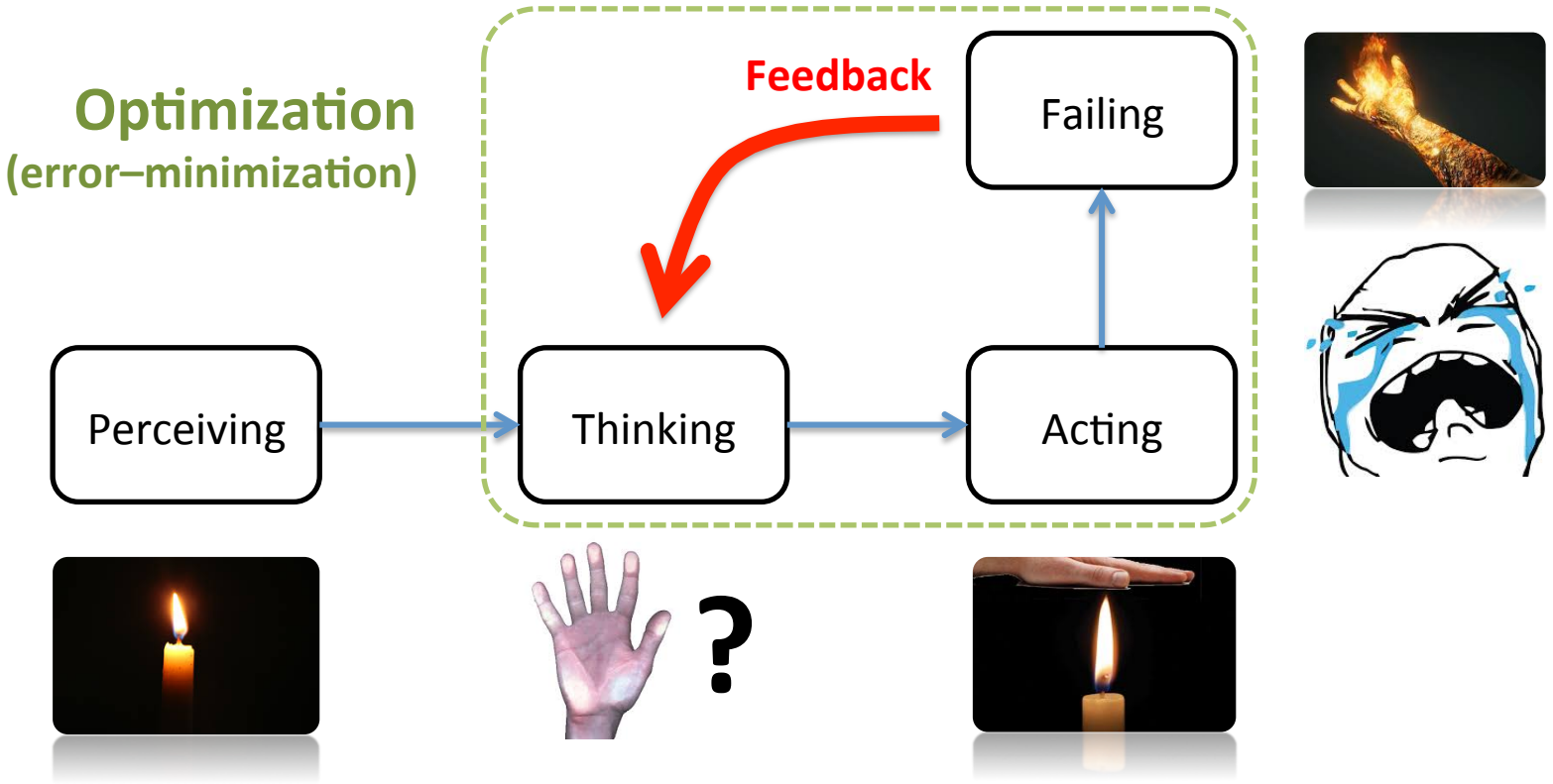
- Imagine you are running down the street with a bowling ball ?
- … What will happen to you?
- You can know the consequences, without having any prior experience !
- **Trivial is not simple.** Warning on the confusion …
- Some people think that ideas are only good if they are complicated

Simple ideas are usually the best and most powerful **(Occam's razor)**

- In the past, AI was tought solely about *reasoning*.
- As there is wide controversies about AI (weak vs. strong, feasible vs. fake)
- … we will only dwell in the world of ***machine learning***.

Algorithms enabled by
Constraints exposed by
Representations supporting
Models targeted at
Thinking, perception and action

Learning ?



- **Goal-directed learning** = acquiring, modifying or reinforcing knowledge
- Most problems can be summarized by a **trial-error process**
- Is a nickname for **optimization** (through **error minimization**)

Why learning ?

- Some things computers do …
- … That we can not

(Try beating a computer at PI decimals)

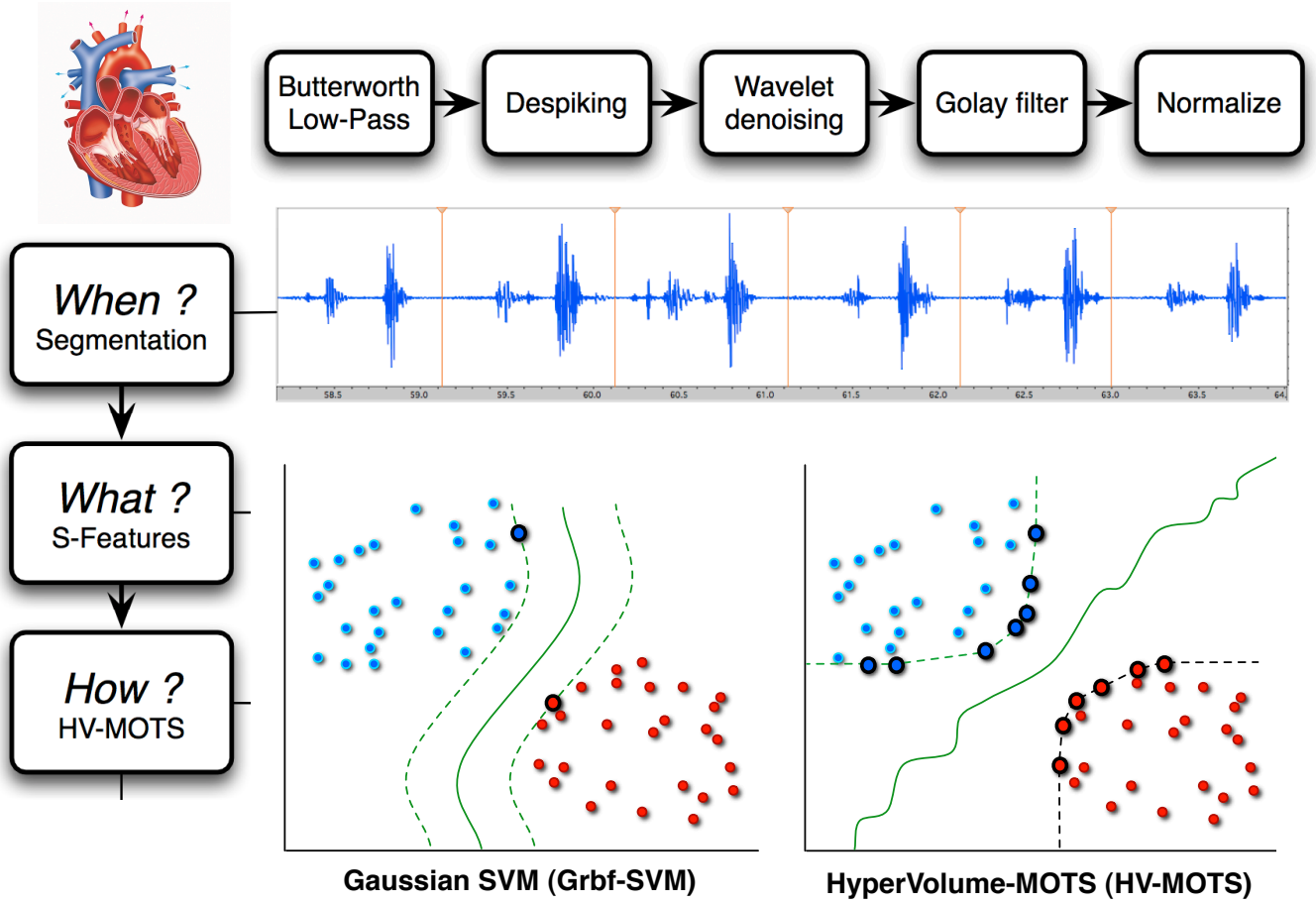
- Some things we do almost instantly …
- … That computer fails miserably

(Typical conversation with a computer)

- 60+ years of ongoing research in AI
- (face recognition, content segmentation, etc…)

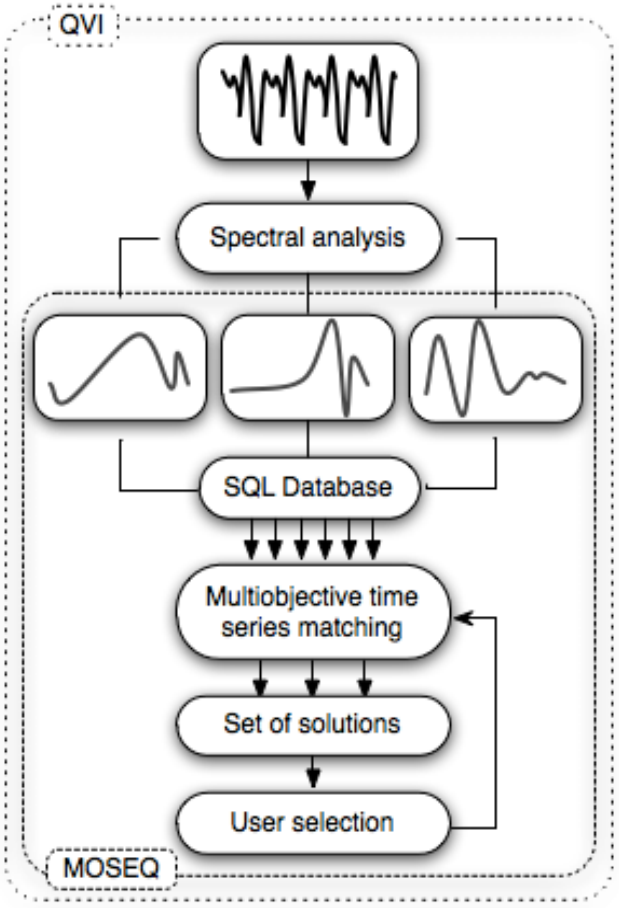
Why learning ?

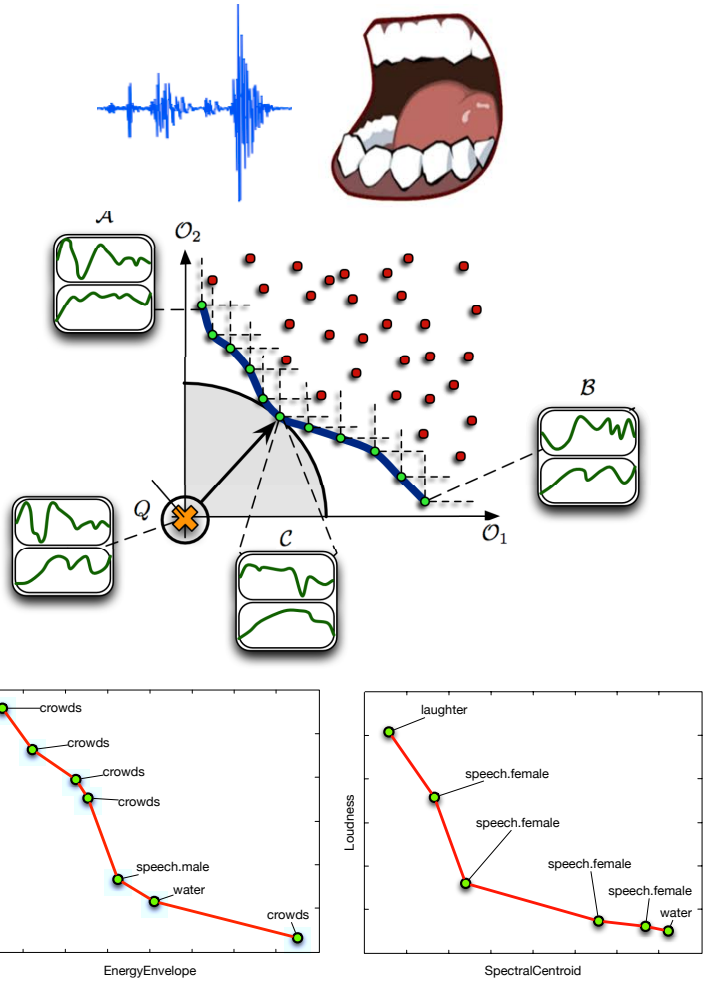
Typical learning problem: **pattern classification**



Why learning ?

Typical learning problem: **pattern classification**

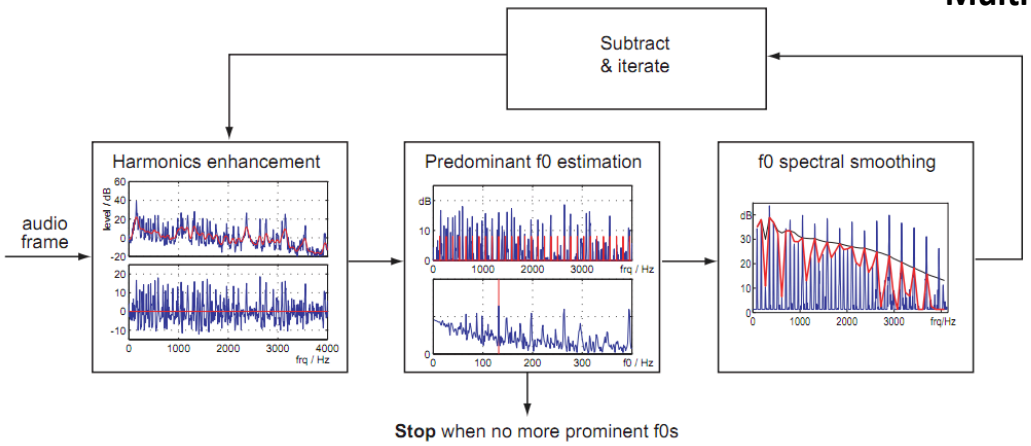


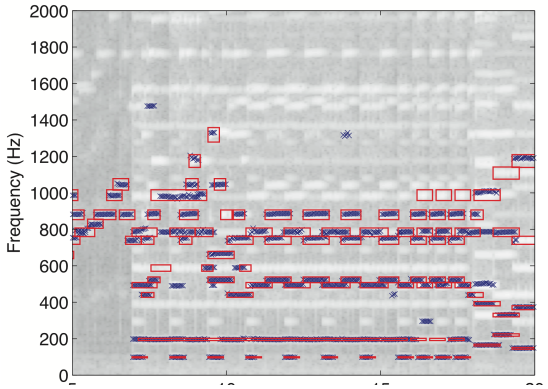


Why learning ?

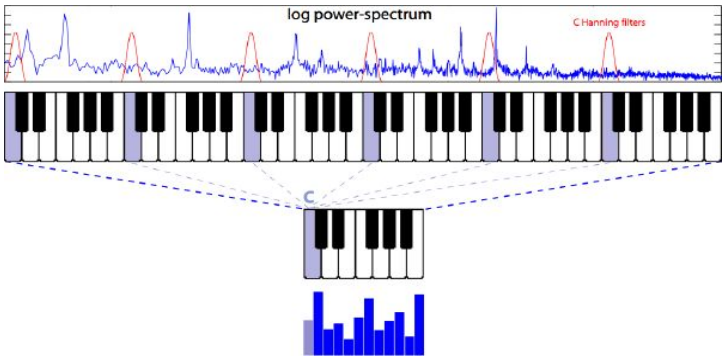
Applications of **pattern classification**

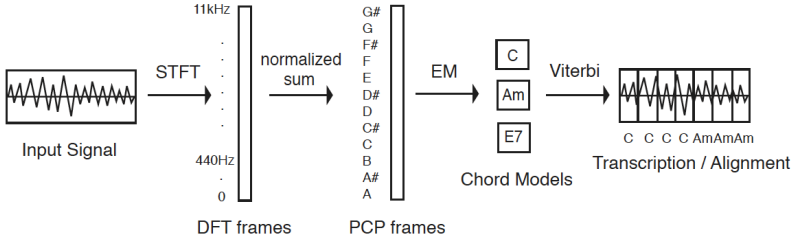
Multi-F0 estimation (figures from *Yeh, 2010*)





Chords estimation (figures from *Padapopoulos, 2007*)

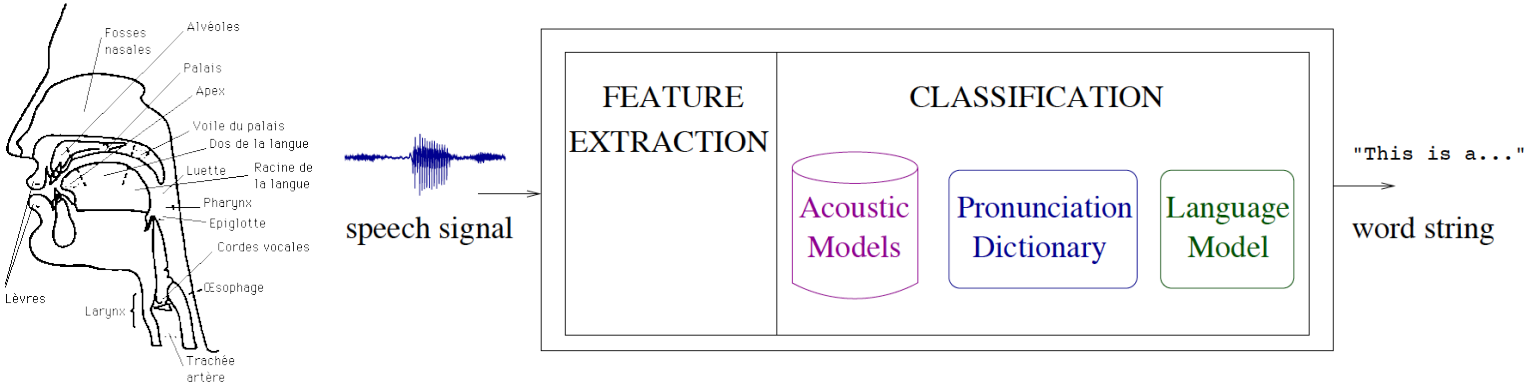




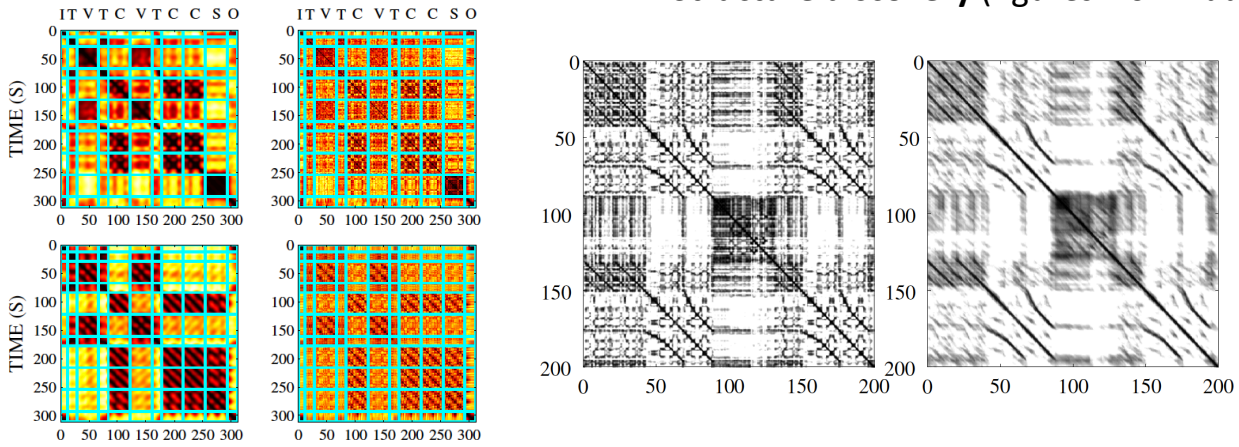
Why learning ?

Applications of **pattern classification**

Speech recognition (figures from *HTK Documentation*)

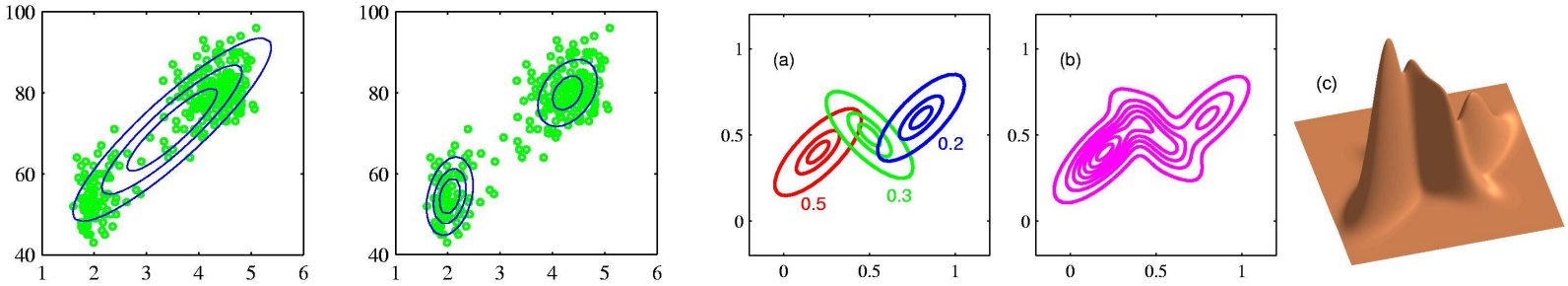


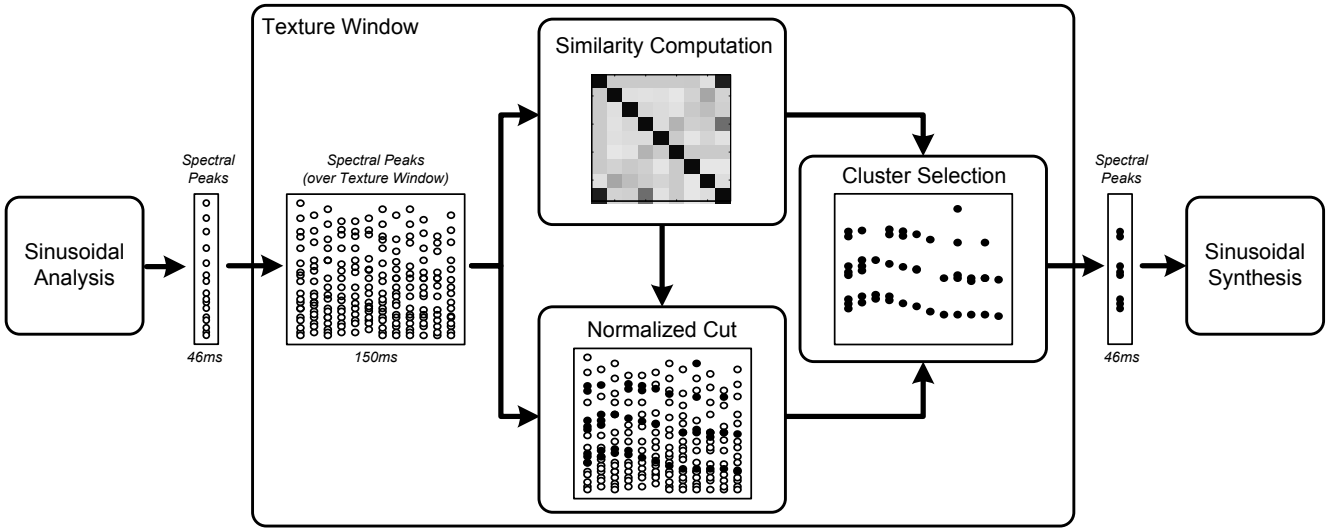
Structure discovery (figures from *Paulus, 2010*)



Why learning ?

Typical learning problem: **clustering**



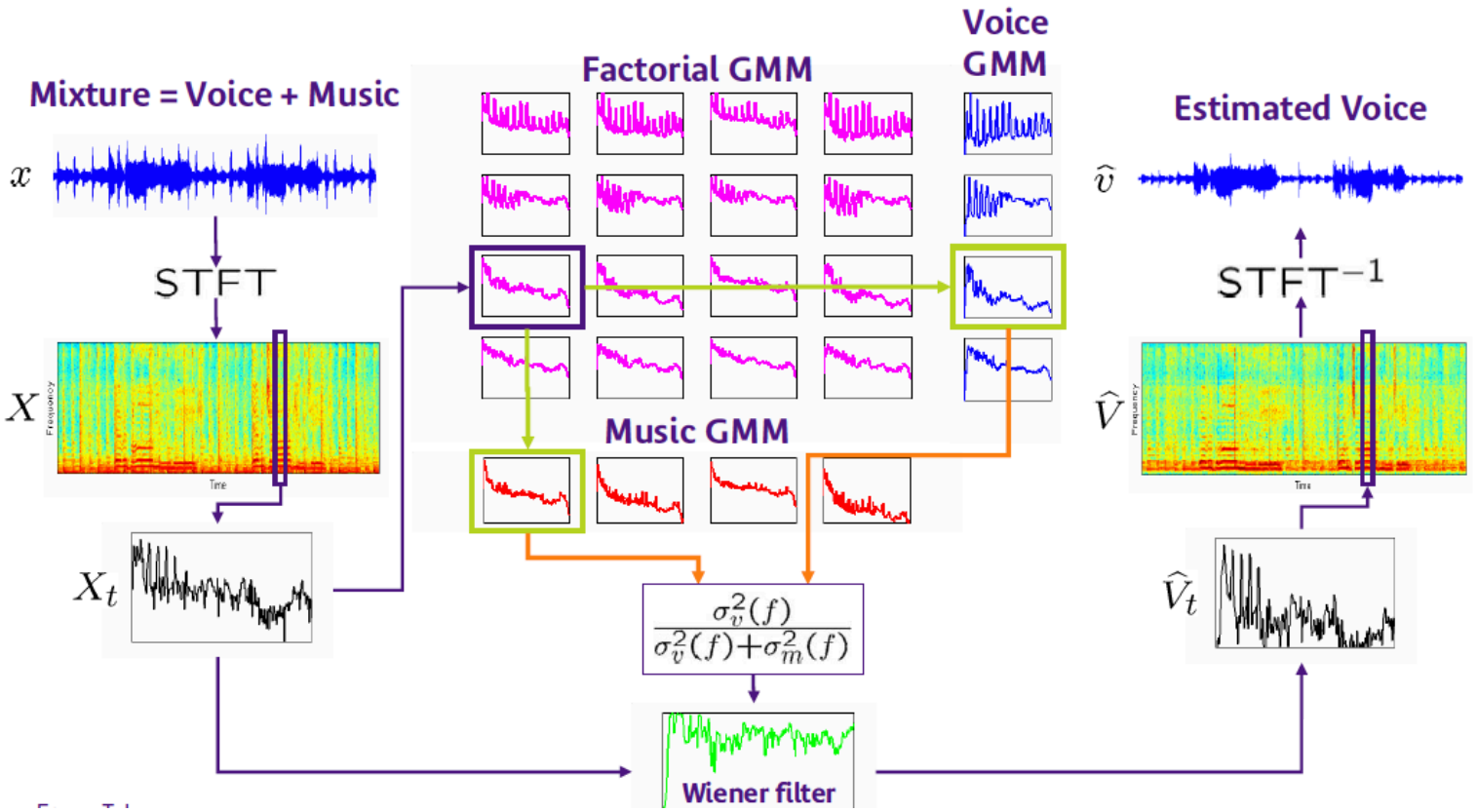


Computational Audio Scene Analysis (CASA) – *Figures from Barker, 2011*

Why learning ?

Applications of **clustering**

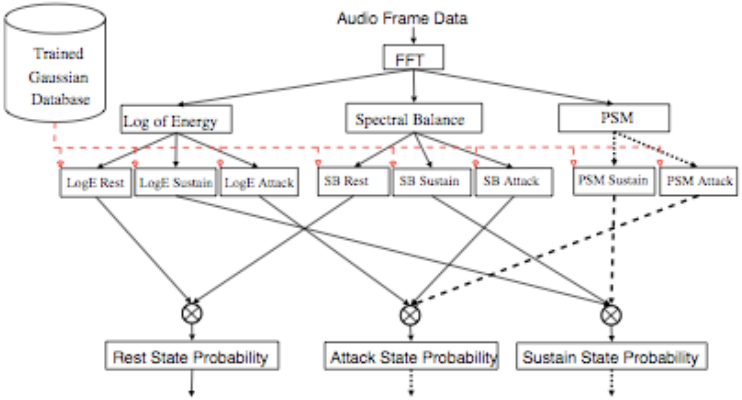
GMM-Based Sound Source Separation – *Figures from Ozerov, 2005*

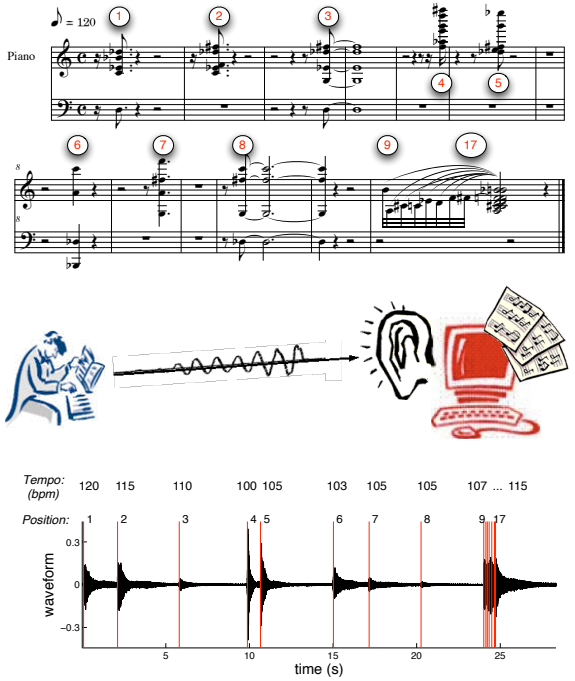


Why learning ?

Sequential learning (temporal models)

- The score following problem:
  - Need *observation models* at the front-end.
  - From audio frames to low-level state probs:

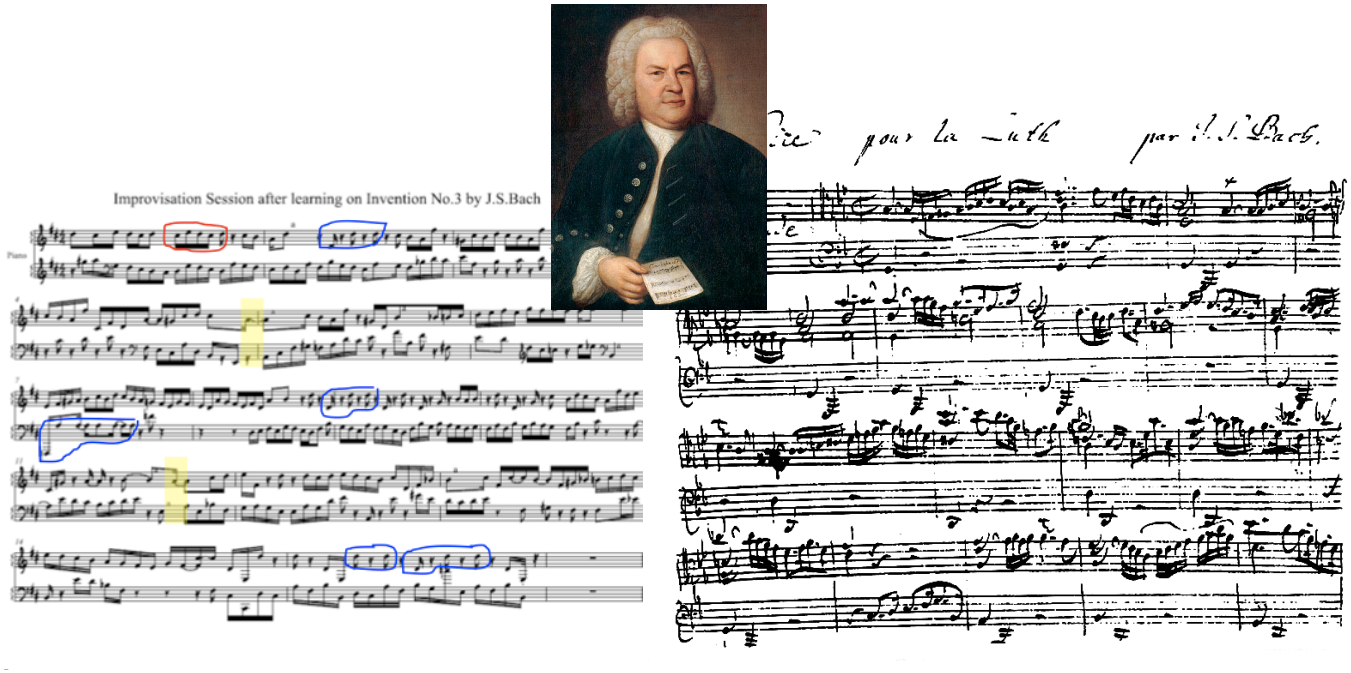


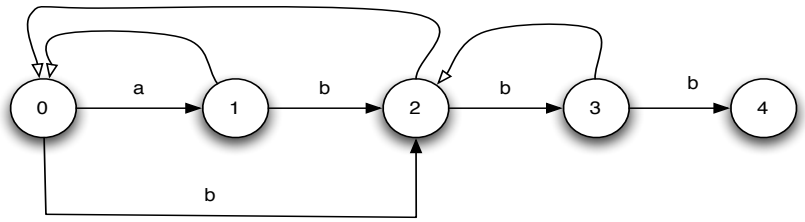


Figures from Cont, 2011

Why learning ?

Sequential learning (temporal models)





Why learning ?

Associative Memory

Artificial intelligence

So why developing and using AI in general?

Engineer « Building smarter programs »

Scientist « Build a computational account of intelligence »

1842: *Lady Ada Lovelace*, the first ever programmer
(even decades before computers even existed)
« *The analytical engine has no pretentions to originate anything, it can do whatever we know how to order it to perform* »

1950s: Alan Turing - « the Turing test »
Marvin Minski - « Steps towards artificial intelligence »

1960s: *Early dawn age* (age of speculation)
Symbolic integration
ELIZA chatbot

ELIZA Program

- The first ever implemented chatbot (1966)
- Eliza emulates a *Rogerian psychotherapist.*
- No intelligence whatsoever, but a lot of interesting tricks
- Uses string substitution and keywords canned response
- Beginning of the Natural Language Processing (NLP) field

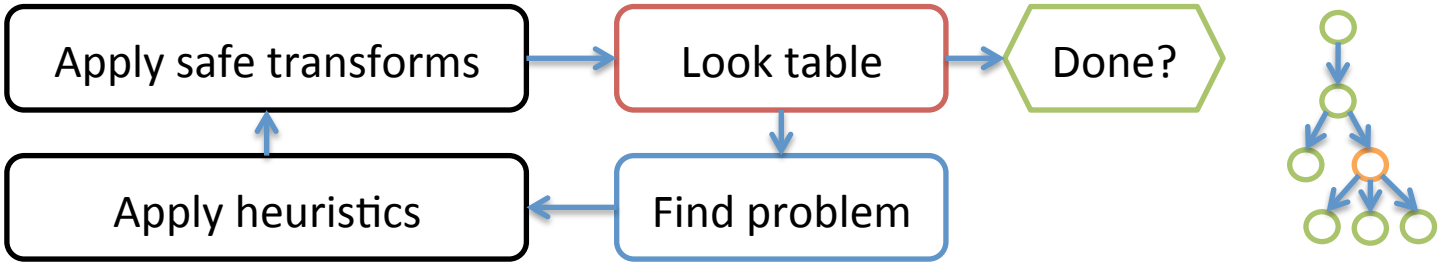
- **Live demonstration**

http://nlp-addiction.com/chatbot/eliza/

Early dawn age: symbolic integration

$$\int \frac{-5x^4}{(1-x^2)^{5/2}}dx \quad \Rightarrow \quad \int (y^2 - 1 + \frac{1}{1+y^2})dx$$

- Program of symbolic integration written by Joel Moses in 1967 at MIT



(1) $\int -f(x) = -\int f(x)$ (2) $\int cf(x)dx = c\int f(x)dx$ (3) $\int \sum_i f_i(x)dx = \sum_i \int f_i(x)dx$

(A) $\int [sin, cos, tan, ...]dx = g([sin, cos, tan, ...])$ (B) $\int f(tan(x))dx = \int \frac{f(y)}{1+y^2}dy$

- Intelligence vanish with explanation
- Knowledge is power => **Knowledge over knowledge is power**
- Problem solving by problem transformation (**problem reduction**)
- The rise of goal trees and expert systems

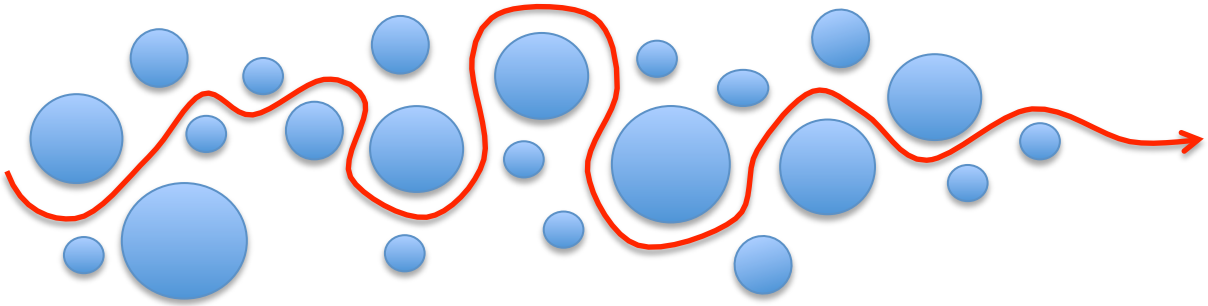
Late dawn: expert system

Ex. *Mycin* – A rule-based expert system for antibiotics diagnostic
Symptoms ? When? Type of infections? My recommendations are …
(turned out better than most practicians)

- Ex. Problem of finding an animal based on its characteristics
- « Expert » is exagerated should be more common sense
- More of a rule-based system (is in fact a form of goal tree)
- Can also go from hypothesis to facts instead of the other way around

How to construct ?
Ask question to experts then reify the rules
Warning to the lack of specificity (elicit some conditions)
⇒ New vocabulary ! New power over the corresponding domain
⇒ Knowledge over things does not mean understanding !
⇒ Prognosis system can give arsenic to a patient as it is the most efficient!

On complexity



- The path of an ant might seems extremely complicated
- You could say that the ant is extremely smart
- You look closer, there are rocks on its path that the ant is just avoiding
- Complexity of a behavior can be the consequence …
- Of the complexity of the environment not of the program

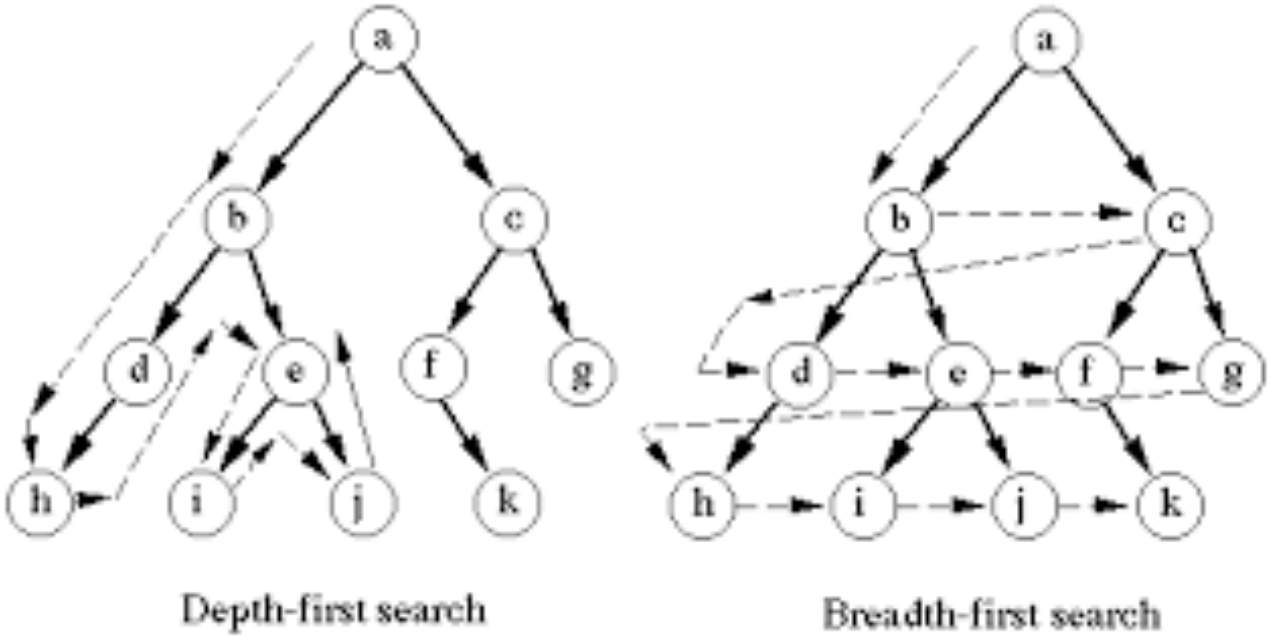
C(behavior) = max(C(program), C(environment)

We'll see a lot of complex behaviors that boil down to simple programs (but complex environments) => **Occam's razor**

Search :Depth-First, Hill-Climbing

A wide part of the optimization field is centered on search

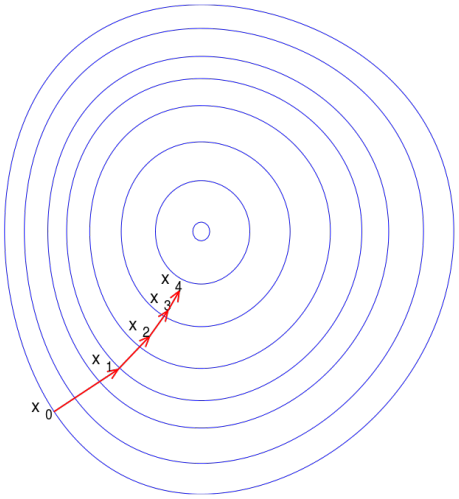
- Brute force
- Depth-first (lexical order)
- Breadth-first

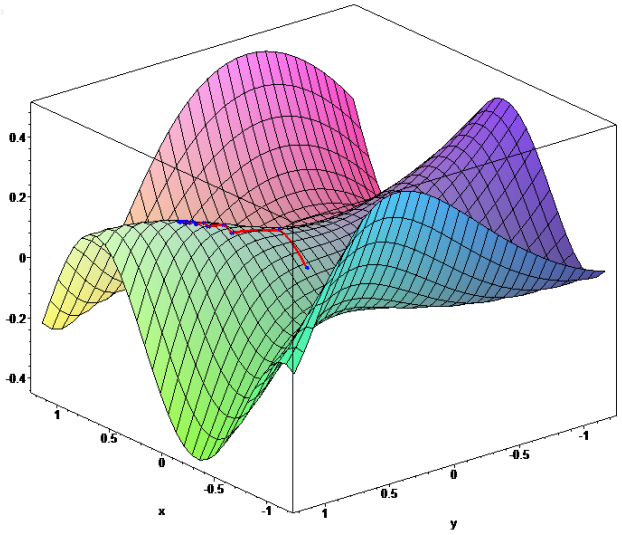


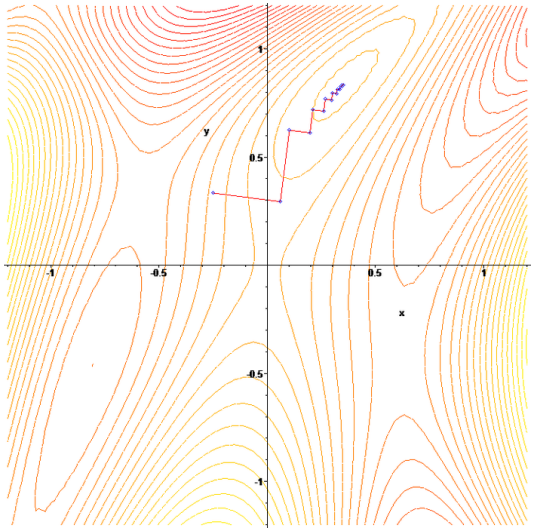
Search :Depth-First, Hill-Climbing

A wide part of the optimization field is centered on search

- Brute force
- Depth-first (lexical order)
- Breadth-first
- Hill climbing / Gradient-descent







Search :Depth-First, Hill-Climbing

A wide part of the optimization field is centered on search

- Brute force
- Depth-first (lexical order)
- Breadth-first
- Hill climbing / Gradient-descent

If you got an heuristic to tell you that you're closer to goal … **Use it**

What are the typical problems of search ?

⇒ Several local optima can impair the search

⇒ Typical « telephone-pole » as a worst search space

⇒ In fact all depends on the distance space

- Do we need search to be intelligent or to make smart algorithms?
- Yes! For instance if you have a symbolic pattern at hand
- You can search to find if you match this higher-level pattern!

(We should do a full lesson on search but think A*, Dijkstra, etc…)

Now ? The **Bulldozer** age

Replacing intelligence by mass computing (parallel and HPC)

Deep blue

Games, Minmax and alpha-beta

Ex. DeepBlue beating Kasparov

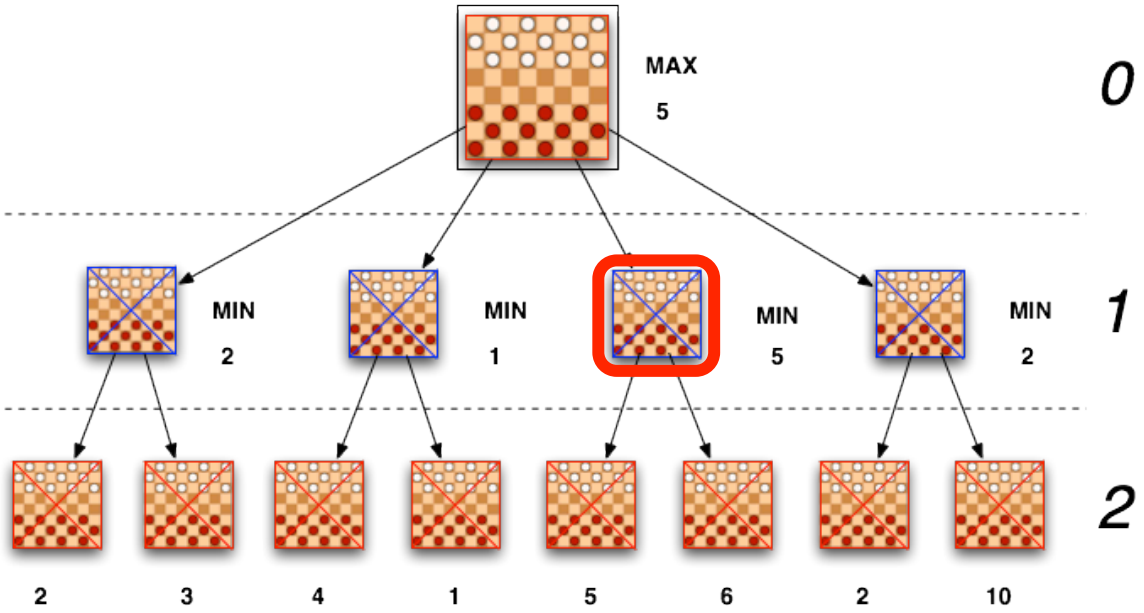
Ways to play

- Analysis/strategy/tactics ?=>? Move
- If/Then rules => Pick the highest current move possible (still lame)
- Look ahead and evaluate the situation (this requires a static value)
  - S = g(f1,f2,…,fn) = w1.f1+w2.f2+…+wn.fn = linear scoring polynomial
- Brute force approach = evaluate the whole tree of possibles
- … However 10^120 possible … leafs !
- So the best way is to look ahead the most possible
- And try to evaluate the highest number of leaves

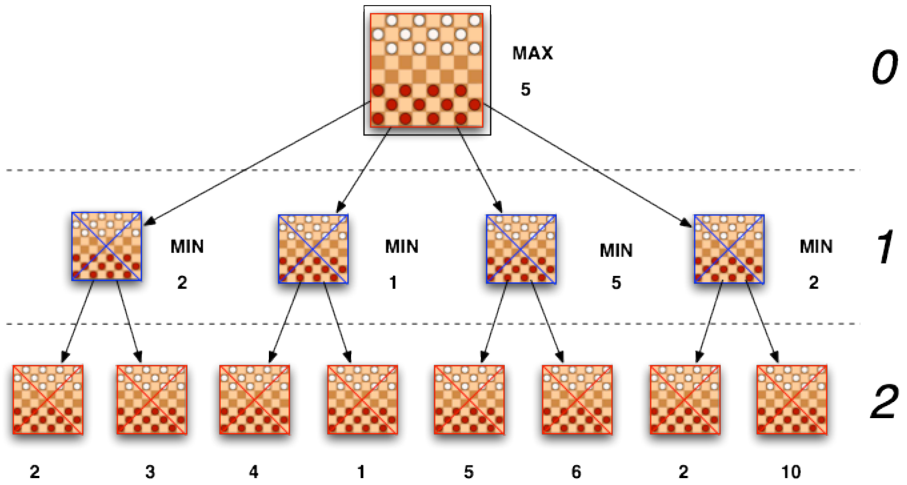
Games, Minmax and alpha-beta

A computer beats a chessmaster: **Min-max algorithm**

- We have a fitness function and compute the *chess tree*
  - We see the tree from the view of the « Maximizing player »
  - But **the minimizing player alternates**
  - So we do not choose the max … but **the max of the mins** ☺
- Need to go as far down the tree as possible
- 14 levels separates a moron from a master !



Games, Minmax and alpha-beta



- Alpha beta **is layered over** minmax to remove large parts of the tree
- The idea is as soon we find a branch lower than our expected value
- Stop evaluating the other sub-branches and try to go further down the other tree!
- Problem of how deep can we go ? Use an insurance policy on level (d-1)
- This idea is called progressive deepening

Deep blue = all this + parallel computing + books + uneven tree dev.

So deep blue intelligent ? Maybe … but mostly **bulldozer** intelligence

Remember to avoid overkill => eg. Ressource allocation = Constraints problem

Learning

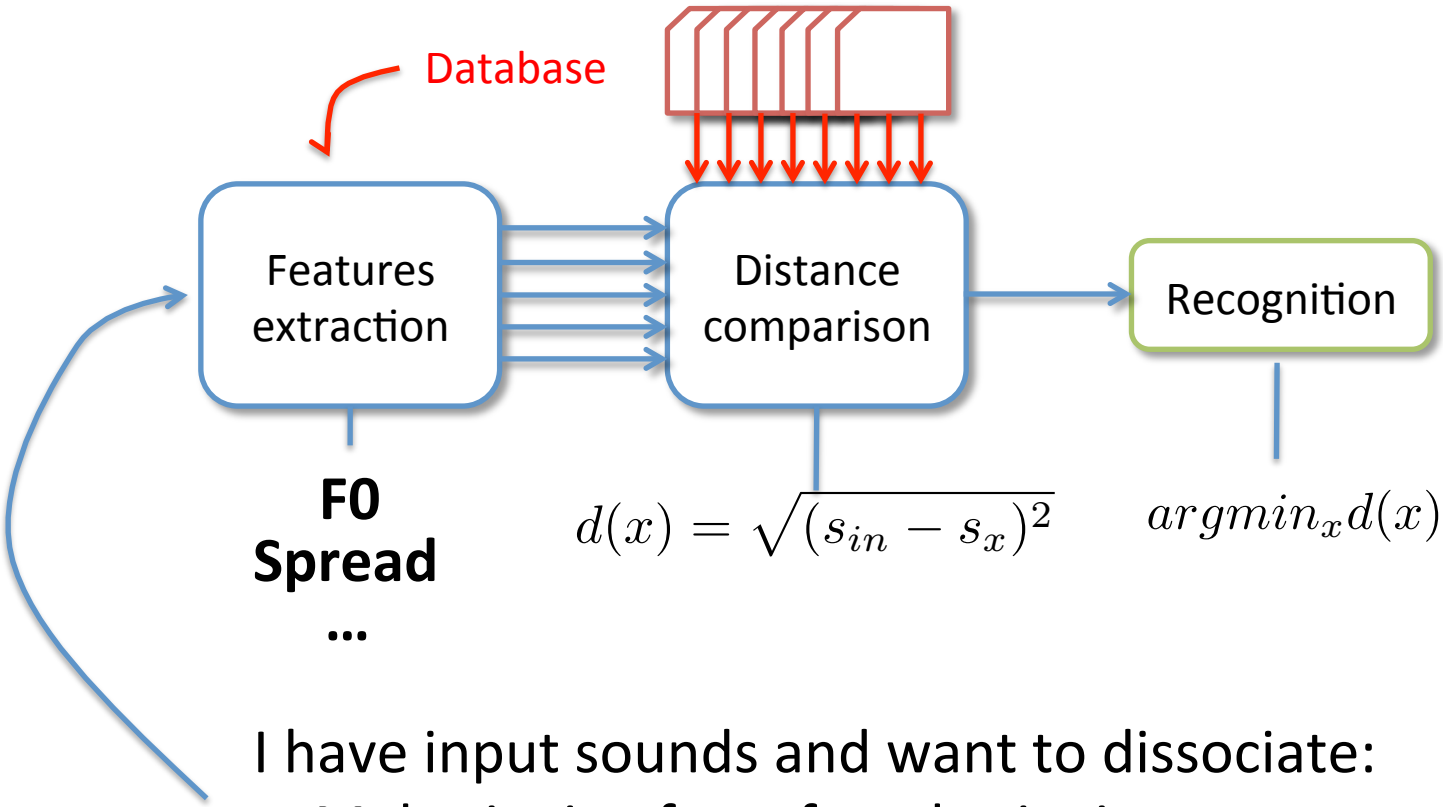
3 major types of learning

- **Supervised learning** is inferring a function from labeled training data
- **Unsupervised learning** is trying to find hidden structure in unlabeled data
- **Reinforcement learning** is acting to maximize a notion of cumulative reward

What will we see along these courses?

1. **Nearest neighbors**
2. **Neural Networks – Single and multilayers**
3. **Support Vector Machines and Kernels**
4. **Clustering (K-Means)**
5. **Probabilistic (Bayesian) inference**
6. **Expectation-Maximization (EM) algorithm**
7. **Gaussian Mixture Models (GMM)**
8. **Sequential Learning (Hidden Markov Models)**
9. **Data complexity (time series, audio)**
10. **Deep learning**

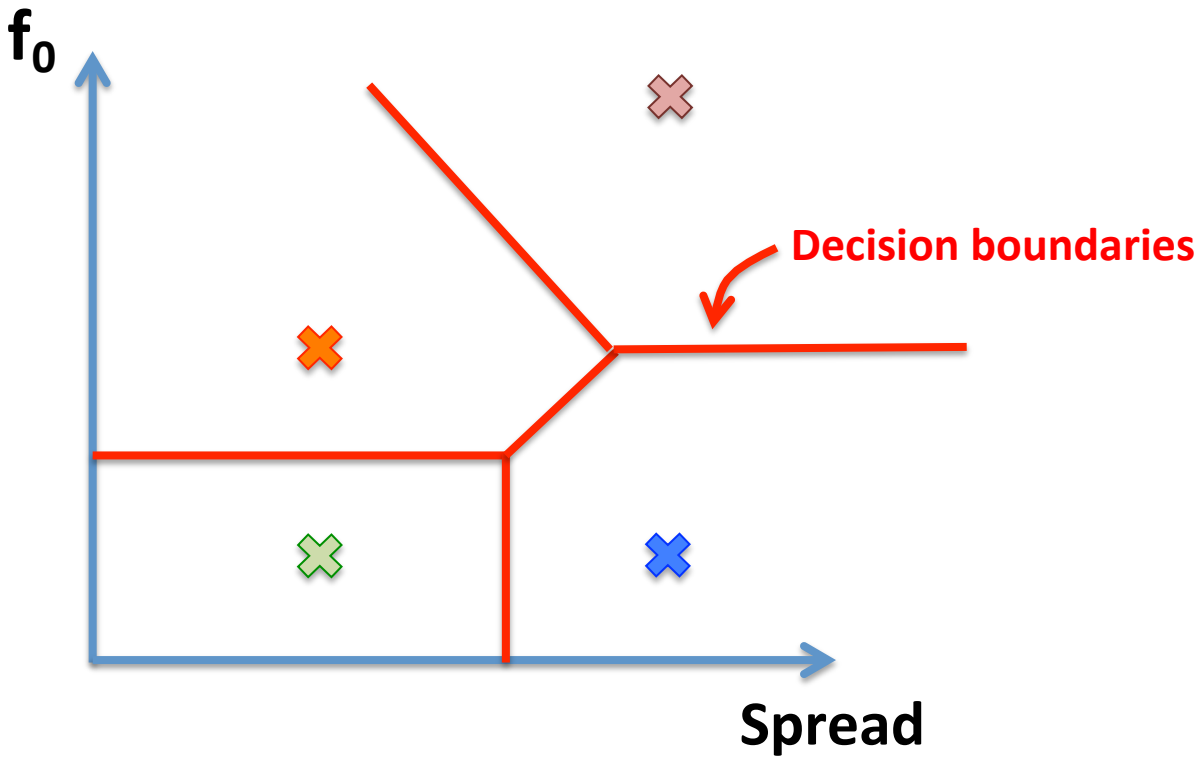
Nearest-Neighbor



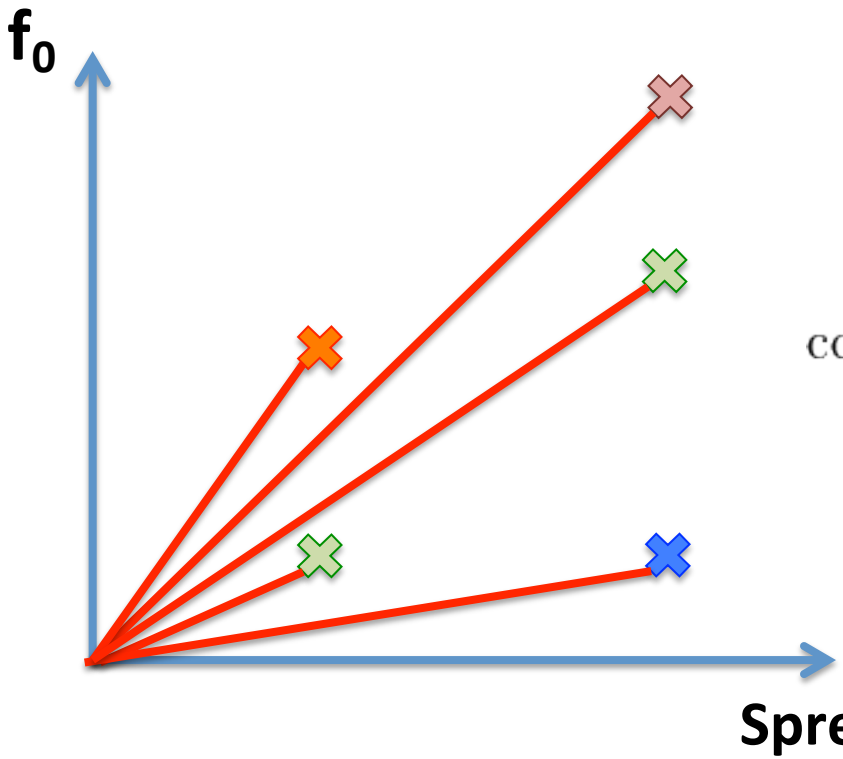
I have input sounds and want to dissociate:

- Male singing from female singing

Nearest-Neighbor

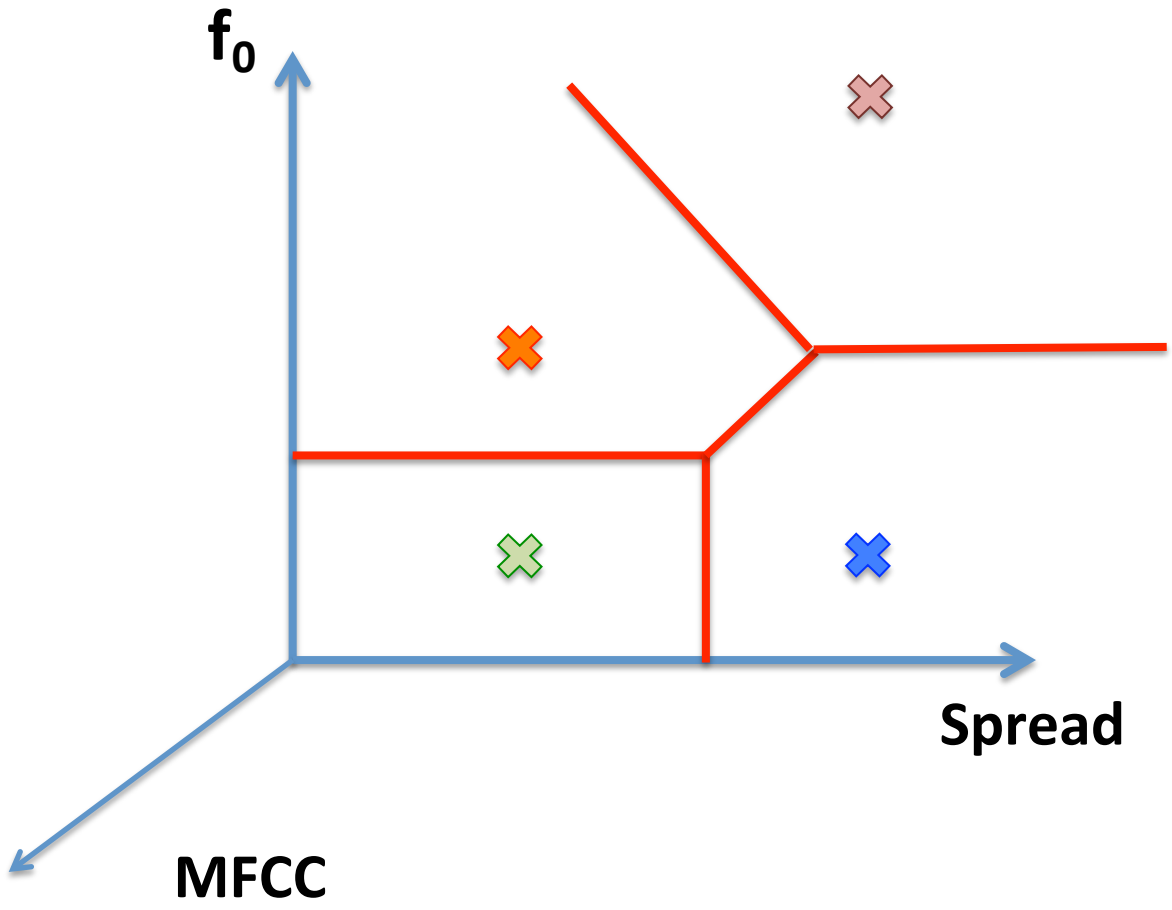


Nearest-Neighbor



$$\cos\theta = \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Nearest-Neighbor



Nearest-Neighbor

Typical turnarounds of Nearest-Neighbor

⇒ Improve your features (make them problem-specific)

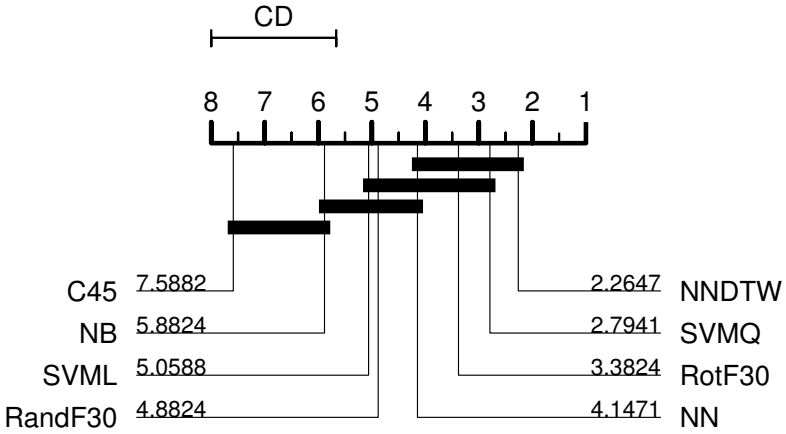
⇒ Add dimensions (without overspecifying the problem)

⇒ Rely on non-standard metrics (cosine, Bregman, …)

⇒ Go from 1-NN to k-NN (eventually with variable k)

- Nearest-Neighbor might seem like a quite dull approach
- However recent researches show that features are more important than classifier
- A dull classifier works perfectly if features are great

Cf. the ant's path and Occam's razor



Results from [Bagnall et. al 2012] on the **17** datasets from the UCR time series archive

1-NN with DTW is statistically superior to other classification methods (even though not *critical* with SVMQ and RotF30)