Vietnam National University - Ho Chi Minh

Optimization, Machine Learning and Kernel Methods.

Introduction to the course

Marco Cuturi - Princeton University

Some preliminary information

- Course is 5 days long
  - Saturday 12/06 7:30AM to 12:30AM room I.23
  - Monday 14/06 1:30PM to 6:30 room I.23
  - Tuesday 15/06 7:30-12:30
  - Wed 16/06 7:30-12:30
  - Thu 17/06 7:30-12:30
- Evaluation: currently speaking with TA's.

Some preliminary information

- **email**: `mcuturi@princeton.edu` **Webpage**: `www.princeton.edu/~mcuturi`
- Research interests: statistical learning, kernel methods, time-series, finance...
- My current job: Lecturer @ **Princeton University** **ORFE dept.**

PRINCETON
Operations
Research
and Financial
Engineering

- My next job (from 09/2010): Associate Prof. @ **Kyoto University** **Graduate School of Informatics**,

A master or PhD at Kyoto University?

- Want to go abroad for a Master or PhD in CS ? why not **Kyoto University**.
  - Check `http://www.g30.i.kyoto-u.ac.jp/en`
  - Google `KU profile`
- **NEW**: full curriculum in **english**.
- **Monbukagakusho** grants ≈ 1.500 USD/month, no tuition fees.
- **Deadline** to join in October is very soon: **July 5th**.
- Another enrollment in **February 2011**, maybe easier.

- Please mention this to your friends in 3rd year, and **ask me if interested**.

The course

$\boxed{\text{Three blocks in this course}}$

- **Optimization** mathematical programming
- **Machine Learning** statistics, regression, classification
- **Kernel Methods** splines, reproducing kernel Hilbert spaces

Objective: cover theoretical, computational and practical aspects to build **computer programs** that can **learn** from databases

The big picture

Some intuitions on machine learning

- Imagine you have seen this movie:

- A friend comes to you and asks you:

I feel like going to the movies tonight, do you think I will like this movie?

- How would you build your answer?

Some intuitions on machine learning

Machine learning helps industries build such **answers** *automatically*

- Imagine you are a DVD rental company.
- It is **part of your business** to recommend good movies to your customers.
- **large scale task:** for 1,000's or 1,000,000's of customers every day!

- Still the same question: would you recommend *Ironman* to customer `AD13242`?

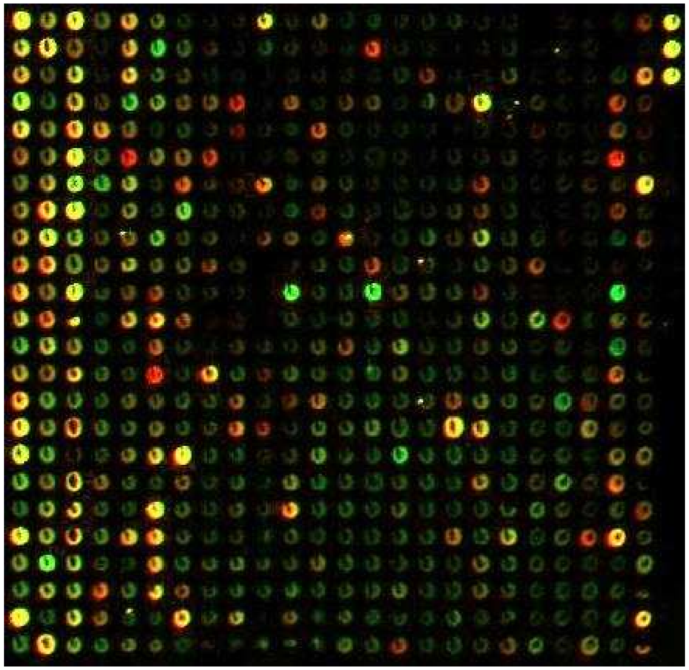
Some intuitions on machine learning

- A computer program **also needs side information**
- For instance:
  - age & background of the user → Check his inscription form.
  - Better! a few examples of movies `AD13242` has seen, with his **ratings**

  - *Lord of the rings I* (+++), *Star Wars I* (++), *Shrek 2* (-) *etc.*.
- How can we decide if we should recommend *Ironman* to `AD13242`?

A more serious problem

- Given the DNA profile of a patient...



- Can we answer (approximately) the questions:
  - What is this patient's **cancer** risk in the next years?
  - What **treatments** can be effective for this patient?

Very fast progress in last years, from theory to practice

You can do a websearch on `mammaprint` or `23andme`

Not only biology or movies.. richly structured data is everywhere

Biology : DNA chips, complex biological pathways.
Medicine : scans, 24/24 measurements of patients.
Business : commercial transactions online and offline.
Search engines : audio, video and textual contents.
Finance : electronic markets, quotes and transactions tick by tick.
Physical interactions : highway networks, mobile phones, GPS localization.
Sociological and physical interactions : social networks on internet, surveillance.
etc.

$\Downarrow$

Data *acquisition* is *cheap* $\neq$ **Data analysis is more difficult**

$\Downarrow$

Need for **data-driven algorithms**
to **fill the gap** between
storing complex data and **understanding it**

Build decision functions

- In many situations, we want to answer a question:

Given a certain situation summarized by x, what can happen/should we do?

- In mathematical terms, we want to build a function:

$$\begin{array}{rccc} f: & \mathcal{X} & \longrightarrow & \mathcal{Y} \\ & x & \longmapsto & f(x) \end{array}$$

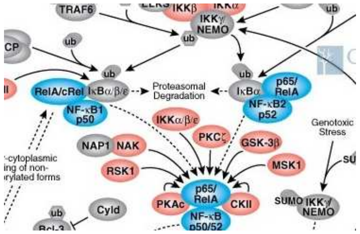
  - $\mathcal{X}$ could be: images, texts, movies, *etc.*
  - $\mathcal{Y}$ could be: "yes/no", real numbers, sentences *etc.*

Our goal: build a **computer program** that outputs a **useful** $f(x)$.

Build decision functions

A few examples in the industry

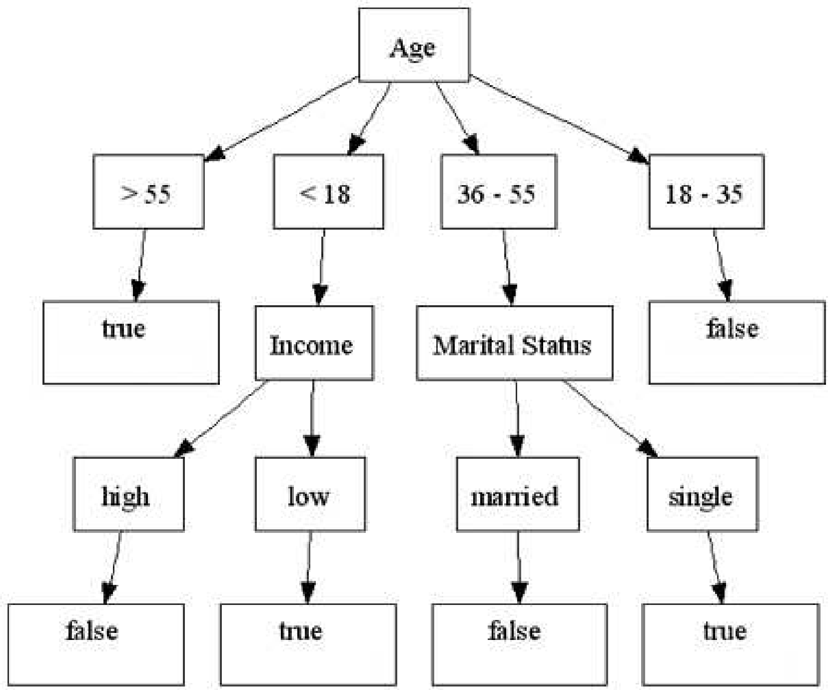
- Ranking answers to a problem,
- Learning jointly different related tasks,
- Learn maps between structured data, *e.g.* translation
- Build interaction maps, *e.g.* for proteins,



- Learn in online settings where data is provided sequentially
- Learn with very large databases: shopping.
- *etc.*

What we *will not* do

- 100% Man-made, rule-based decision trees.



- **advantages**: sometimes expertise available, just need to **rationalize** it.*etc.*
- **disadvantages**: difficult to **replicate**, unadapted for **large** systems and **new problems** (DNA) where no expertise exists by definition!

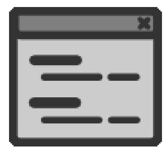
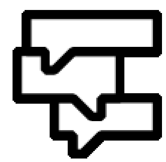
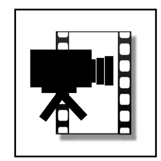
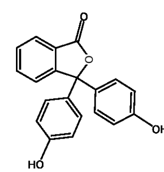
What we will do:

- Use data collected in databases as the **main ingredient** to build f.

 $\longrightarrow \quad f$

- Build architectures where **machines** can **learn** from these databases.

The kind of data we will handle

- **Random**
  - Unlike deterministic systems, we assume **randomness**.
  - **Future** requests are **not known**. Some are **more likely**.
- **Structured, complex**
  - strings, texts and sequences, 
  - images, audio and video feeds,   
  - graphs, interaction networks and 3D structures 

Statistical Inference

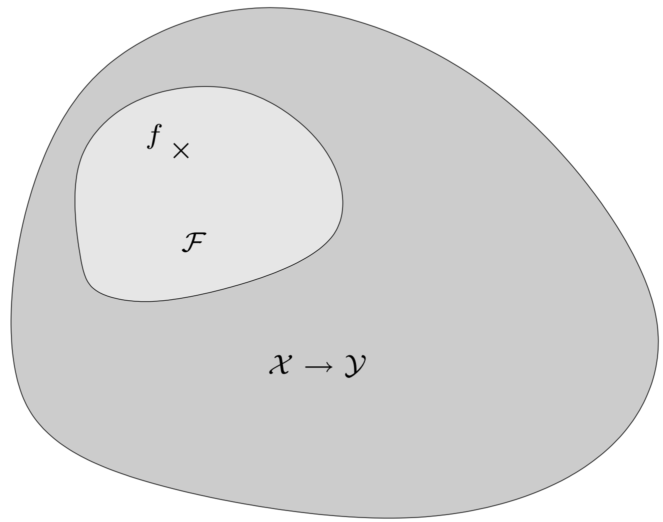
Definition

Statistical inference is the process of **making conclusions** using data that is subject to random variation, for example, observational errors or sampling variation.

- **Statistical inference** = Take decisions in a random environment based on past observations.
- **statistical**: probabilistic view of the world.
- **inference**: purpose to understand and predict better.

Ingredients to pick a good f

- A set of candidates $\mathcal{F}$.



- A way to use the database (past observations)
  - **Data-dependent** criterion $C_{\textbf{data}}$ to select f.
  - Usually given a function g, $C_{\text{data}}(g)$ big if g not accurate on the data.
- A method to find an **optimal** candidate in $\mathcal{F}$.

$$f = \operatorname{argmin}_{g \in \boldsymbol{\mathcal{F}}} \quad C_{\text{data}}(g).$$

Outline of the course

- **Optimization** (argmin).
  - Convexity & linear programming (6 hours)
  - Convex programming (4 hours)
- **Statistical Modeling** to define (C_{data}) (4 hours)
  - elementary probability,
  - study of different situations and different C.
- **Kernel Methods**, a possible choice for $\mathcal{F}$ (6 hours)