Computational Mathematics for Learning and Data Analysis: introduction

Antonio Frangioni

Department of Computer Science
University of Pisa
www.di.unipi.it/~frangio
frangio@di.unipi.it

Computational Mathematics for Learning and Data Analysis
Master in Computer Science – University of Pisa

Course Organization

- 1 course (9 CFU/ects)
- 1 program
- 1 exam
- 2 + 1 lecturers

Antonio Frangioni (Optimization)

Dipartimento di Informatica, room 381
Tel. 050 2212789, e-mail: frangio@di.unipi.it
Office hours: ??

Federico Poloni (Numerical methods)

Dipartimento di Informatica, room 343
Tel. 050 2213143, e-mail: federico.poloni@unipi.it
Office hours: ??

Leonardo Robol (angel)

ISTI – CNR, Via G. Moruzzi 1, Pisa
Room C-55, Building B, Floor 1, Gate 19
Tel. 050 6212799, e-mail: leonardo.robol@isti.cnr.it

Information

Course Schedule

- Wed 9:00 – 11:00 room L1
- Thu 11:00 – 13:00 room H-Lab
- Fri 9:00 – 11:00 room N1

Extra seminars, 2 h/w, time TDB

Possible: Wed 16:00 – 18:00 or Fri 16:00 – 18:00

Web page

`https://elearning.di.unipi.it/course/view.php?id=102`

Exam

$$\left\{ \begin{array}{c} \text{either computer-based test} \\ \text{or project (groups of 2)} \end{array} \right\} + \text{oral exam}$$

Possible (and encouraged) projects with the ML course.

Course material

- Slides prepared by the lecturers
- `Matlab` programs
- Recording of lessons
- L. N. Trefethen, D. Bau, Numerical Linear Algebra, SIAM, 1997
- J. Demmel, Applied Numerical Linear Algebra, SIAM, 1996
- S. Boyd, L. Vandenberghe, Convex optimization, 2004 (`http://web.stanford.edu/~boyd/cvxbook/`)
- M.S. Bazaraa, H.D. Sherali, C.M. Shetty, Nonlinear programming: theory and algorithms, Wiley & Sons, 2006
- J. Nocedal, S. Wright, Numerical Optimization, Springer Series in Operations Research and Financial Engineering, 2006

Syllabus

- Linear algebra and calculus background
- Unconstrained optimization and systems of equations
- Direct and iterative methods for linear systems and least-squares
- Numerical methods for unconstrained optimization
- Iterative methods for computing eigenvalues
- Constrained optimization and systems of equations
- Duality (Lagrangian, linear, quadratic, conic, Fenchel's)
- Numerical methods for constrained optimization
- Software tools for numerical computations (Matlab, Octave, ...)
- Sparse hints to AI/ML applications

Why this course

- Huge amounts of data is generated and collected, but one has to make sense of in order to use it: learn it
- Take something big and unwieldy and produce something small and nimble that can be used instead
- That's a (mathematical) model
- Word comes from "modulus", diminutive from "modus" = "measure": "small measure"
- First known uses in architecture: proving in the small that the real building won't collapse (particularly famous the models of Filippo Brunelleschi for the Cupola of the Dome of Florence)
- Countless many physical models afterwards (planes, cars, ...)
- Mathematics is cheaper than bricks
- But mathematical problems can be difficult for various reasons

Choosing a mathematical model

- How a mathematical model should be:
  1. accurate
  2. computationally inexpensive
  3. general

  Typically impossible to have all three!
- Developing general models is convenient (work once, apply many)
- The shape of the model controls the computational cost
- But how to get accuracy for a given application?

  model is parametric, learn the right values of the parameters
- In other words: within the family of infinitely many models with the given shape, find the one that better represent your phenomenon
- This is fitting, and it is clearly some sort of optimization problem
- Solving the fitting problem is typically the computational bottleneck
- However, ML $\gg$ fitting: fitting minimizes training error $\equiv$ empirical risk, but ML aims at minimizing test error $\equiv$ risk $\equiv$ generalization error!

Example 1: Linear Estimation

- Phenomena measured by one number y is believed to depend on a vector $x = [\, x_1 \,, \ldots \,, x_n \,]$ of other numbers
- Available set of observations $(y^1, x^1), \ldots, (y^m, x^m)$
- Horribly optimistic assumption: the dependence is linear, i.e.,

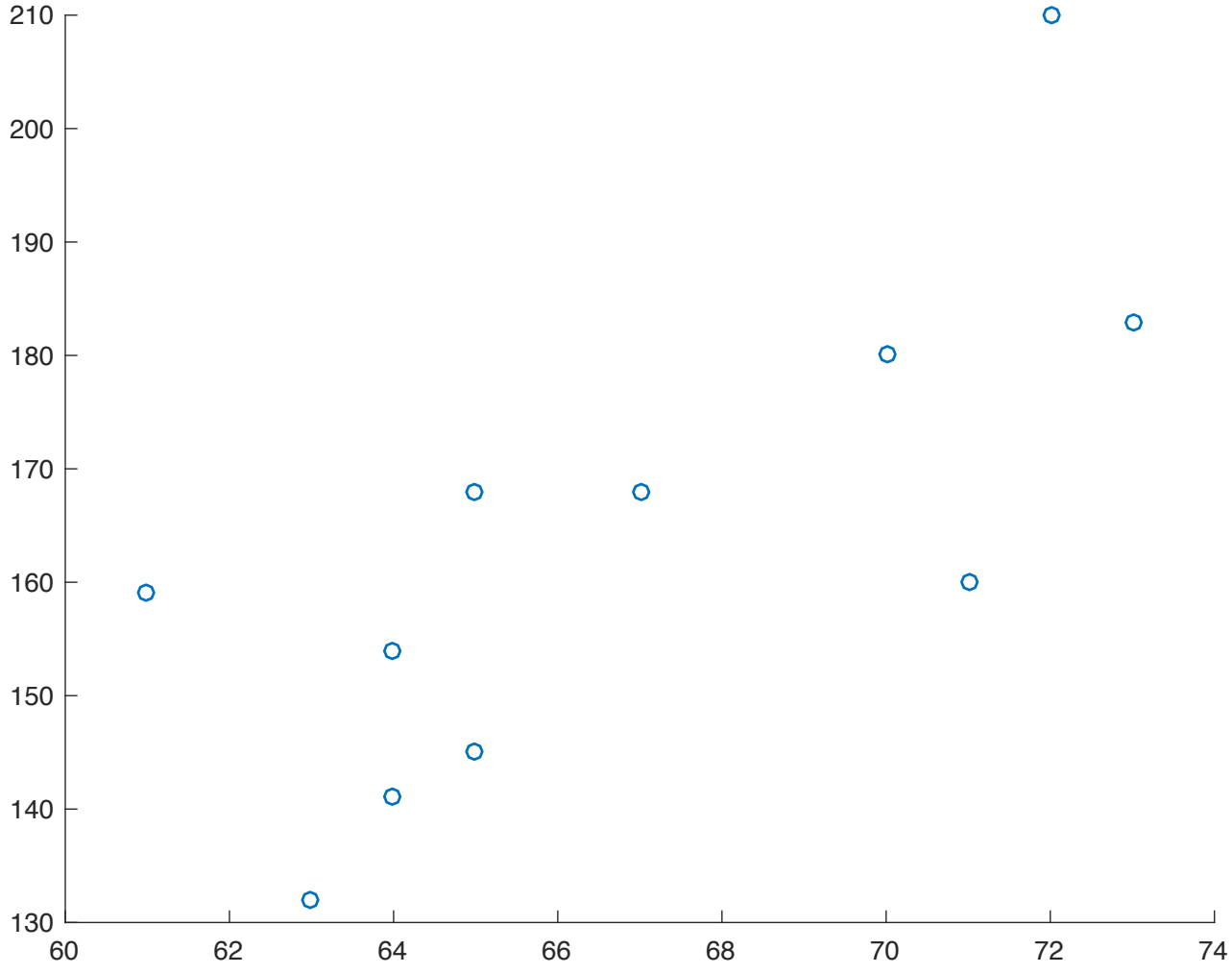
$$y = \sum_{i=1}^{n} w_i x_i + w_0 = wx + w_0$$

  for fixed $n+1$ real parameters $w = [\, w_0 \,, \; w_+ = [w_1 \,, \ldots \,, w_n \,]\,]$
- This would imply that $y^i = w_+ x^i + w_0$ for all $i = 1, \ldots, m$, which is not really true for any w and w_0
- Find the w for which it is less untrue (Linear Least Squares):

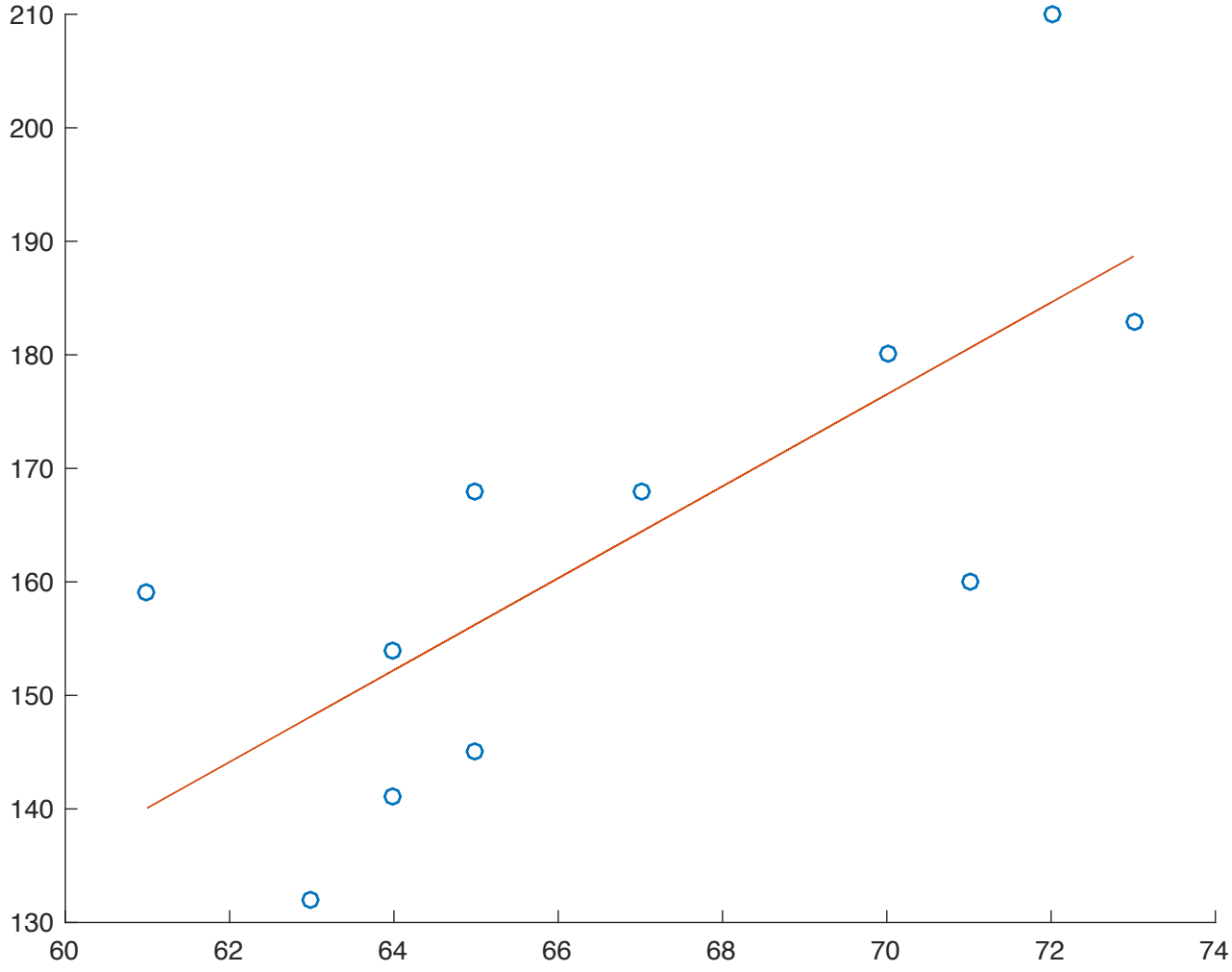
$$y = \begin{bmatrix} y^1 \\ \vdots \\ y^m \end{bmatrix} \;, \quad X = \begin{bmatrix} 1 & x^1 \\ \vdots & \vdots \\ 1 & y^m \end{bmatrix} \;, \quad \min_{w,b} \left|\left| \, y - Xw \, \right|\right|$$

- Simple closed formula: $XX^T w = X^T y \;\; \leftarrow w = (XX^T)^{-1} X^T y$

Linear Estimation (cont.d)



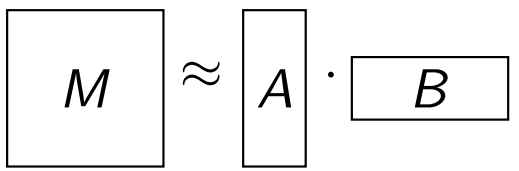
Linear Estimation (cont.d)



- In `Matlab`, this is just `w = X \ y`
- Trade-off: very simple fitting for exceedingly crude model $\Longrightarrow$ high risk
- Then, of course Nonlinear Estimation ...

Example 2: Low-rank approximation

- A (large, sparse) matrix $M \in \mathbb{R}^{n \times m}$ describes a phenomenon depending on pairs (e.g., objects chosen from customers)
- Describe $M \approx AB$ with "tall and thin" $A \in \mathbb{R}^{n \times k}$ and "fat and large" $B \in \mathbb{R}^{k \times m}$ $(k \ll n, m)$



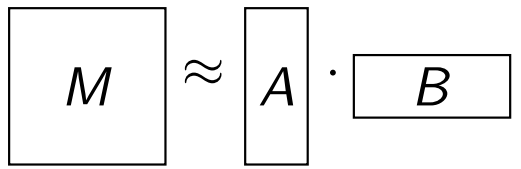
$$, \quad \min_{A,B} \left|\left| M - AB \right|\right|$$

  $\equiv$ find a few features that describe most of users' choices

- Many applications (neural networks, community analysis...)
- A, B can be obtained from eigenvectors of $M^T M$ and MM^T ...

Example 2: Low-rank approximation

- A (large, sparse) matrix $M \in \mathbb{R}^{n \times m}$ describes a phenomenon depending on pairs (e.g., objects chosen from customers)
- Describe $M \approx AB$ with "tall and thin" $A \in \mathbb{R}^{n \times k}$ and "fat and large" $B \in \mathbb{R}^{k \times m}$ $(k \ll n, m)$



$$, \quad \min_{A,B} \left|\left| M - AB \right|\right|$$

  $\equiv$ find a few features that describe most of users' choices

- Many applications (neural networks, community analysis. . .)
- A, B can be obtained from eigenvectors of $M^T M$ and MM^T . . .
  . . . but that's a huge, possibly dense matrix
- Solve this problem avoiding forming $M^T M$ and MM^T, exploiting structure of M, ensuring the solution is numerically stable . . .

Low-rank approximation for image compression

Black/white image $\equiv M$ with color intensities $\in [0, 1]$

Original (512×512)

$k = 1$

$k = 10$

$k = 25$

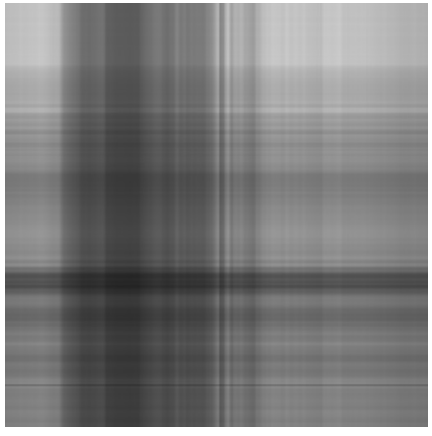
$k = 50$

$k = 100$

Low-rank approximation for image compression

Black/white image $\equiv M$ with color intensities $\in [0, 1]$

Original (512×512)



$k = 1$

$k = 10$

$k = 25$

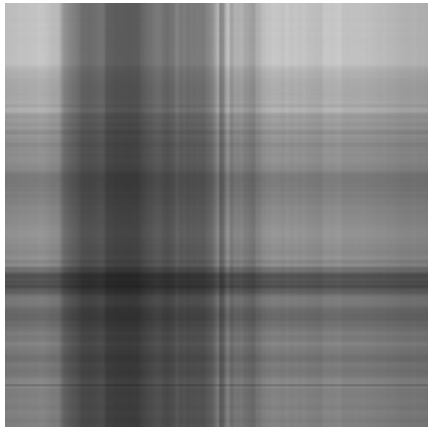
$k = 50$

$k = 100$

Low-rank approximation for image compression

Black/white image $\equiv M$ with color intensities $\in [0, 1]$

Original (512×512)



$k = 1$

$k = 10$

$k = 25$

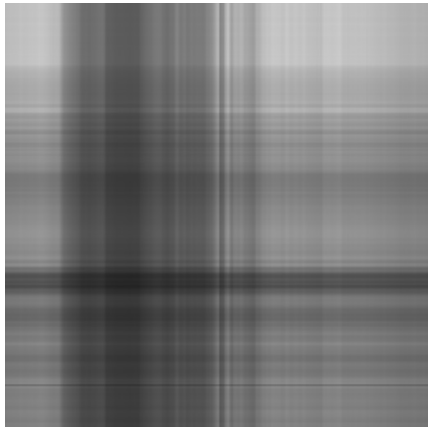
$k = 50$

$k = 100$

Low-rank approximation for image compression

Black/white image $\equiv M$ with color intensities $\in [0, 1]$

Original (512×512)



$k = 1$

$k = 10$

$k = 25$

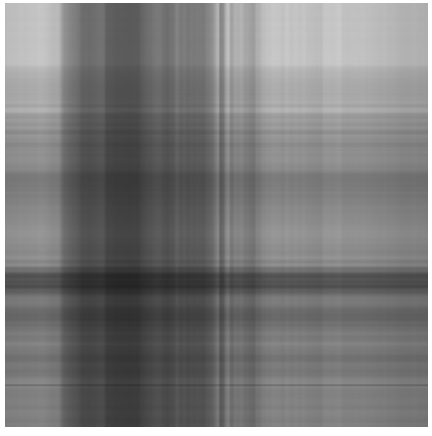
$k = 50$

$k = 100$

Low-rank approximation for image compression

Black/white image $\equiv M$ with color intensities $\in [0, 1]$

Original (512×512)



$k = 1$

$k = 10$

$k = 25$

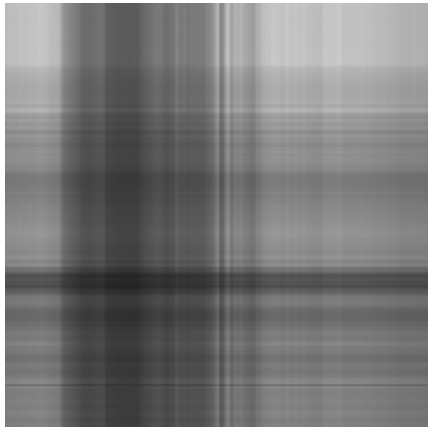
$k = 50$

$k = 100$

Low-rank approximation for image compression

Black/white image $\equiv M$ with color intensities $\in [0, 1]$

Original (512×512)



$k = 1$

$k = 10$

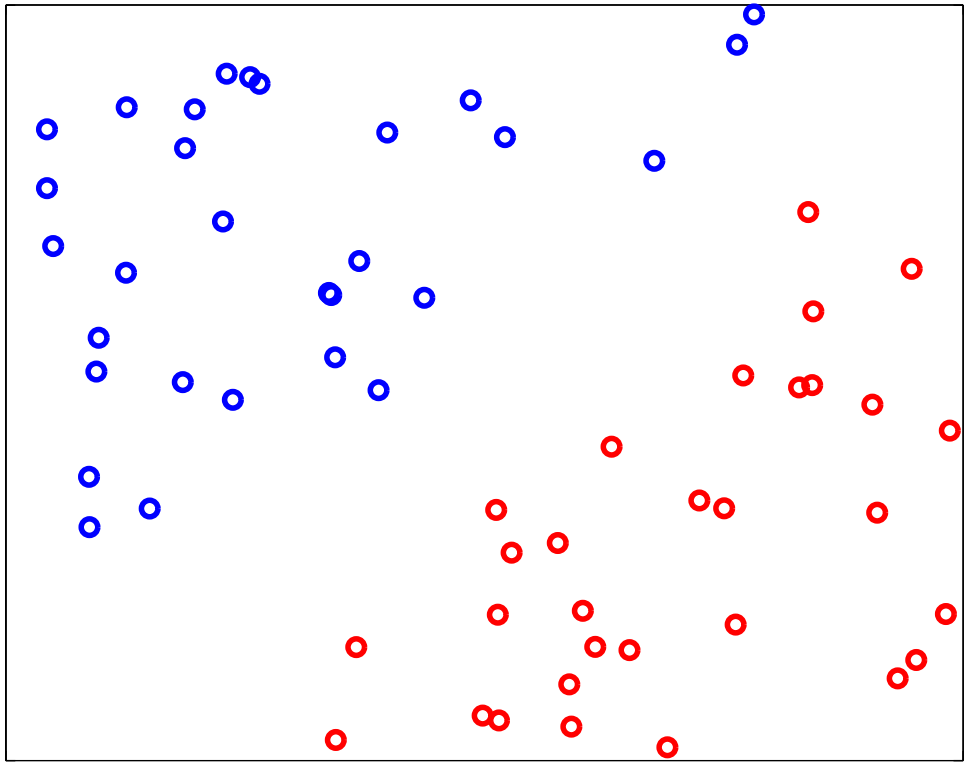
$k = 25$

$k = 50$

$k = 100$

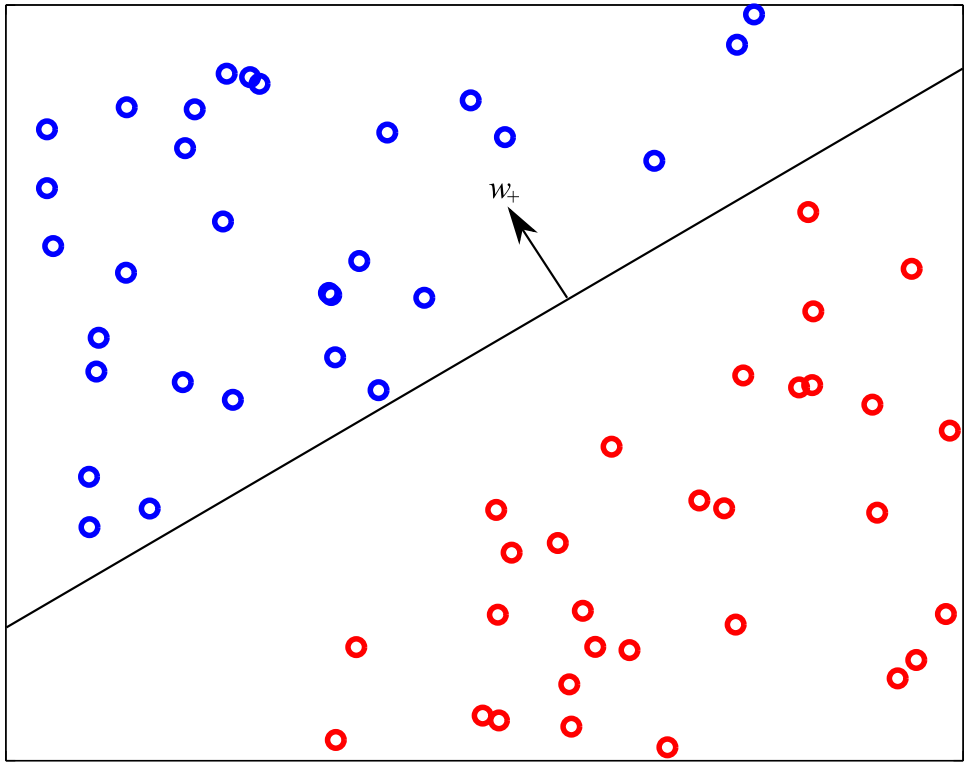
Example 3: Support Vector Machines

- Same setting as Example 1 but $y^i \in \{1, -1\}$ (have cancer or not)



Example 3: Support Vector Machines

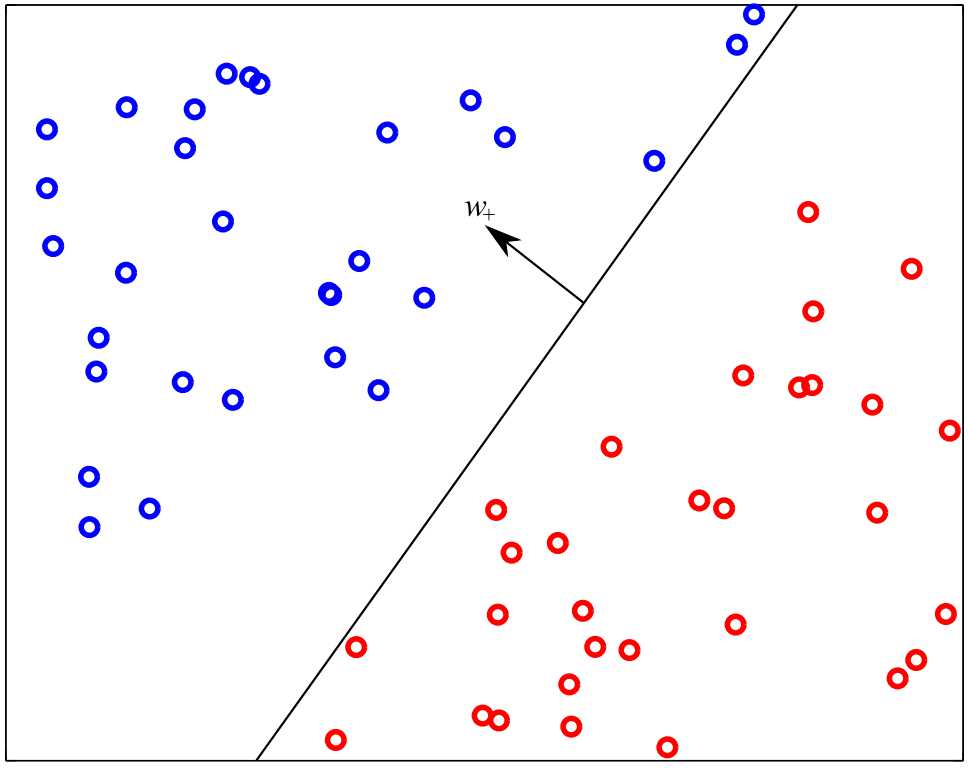
- Same setting as Example 1 but $y^i \in \{1, -1\}$ (have cancer or not)



- Want to linearly separate the two sets (diagnose the next patient)
- Countless many applications (medical diagnosis, OCR, spam filtering, fraud detection, marketing, image processing . . .)

Example 3: Support Vector Machines

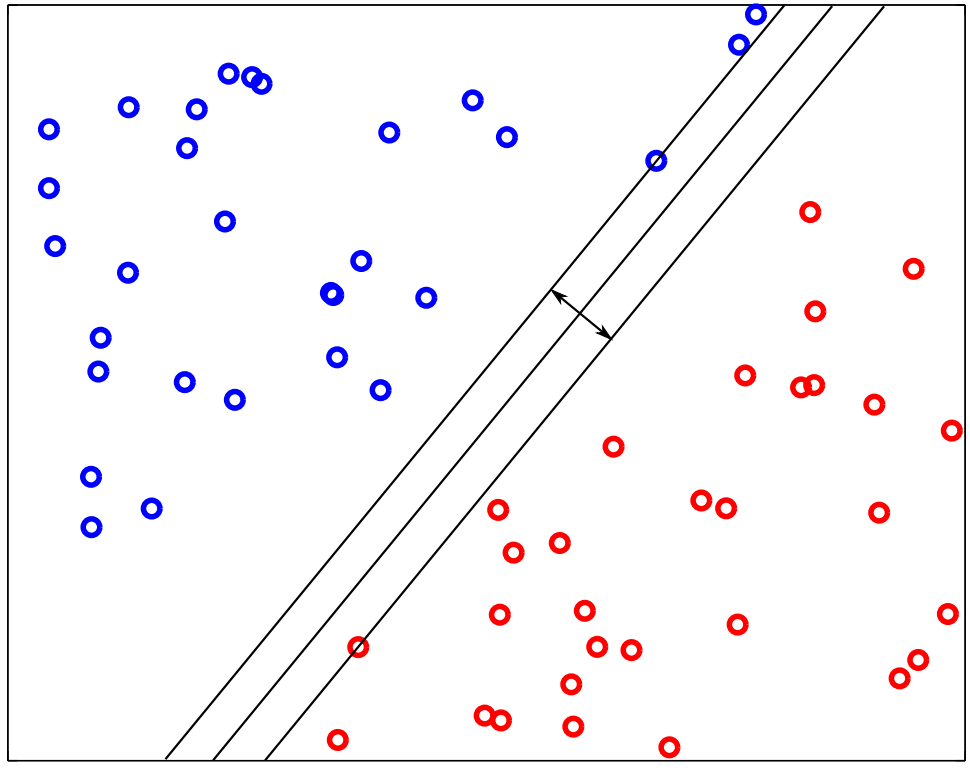
- Same setting as Example 1 but $y^i \in \{1, -1\}$ (have cancer or not)



- Want to linearly separate the two sets (diagnose the next patient)
- Countless many applications (medical diagnosis, OCR, spam filtering, fraud detection, marketing, image processing . . .)
- But which hyperplane do we choose?

Example 3: Support Vector Machines

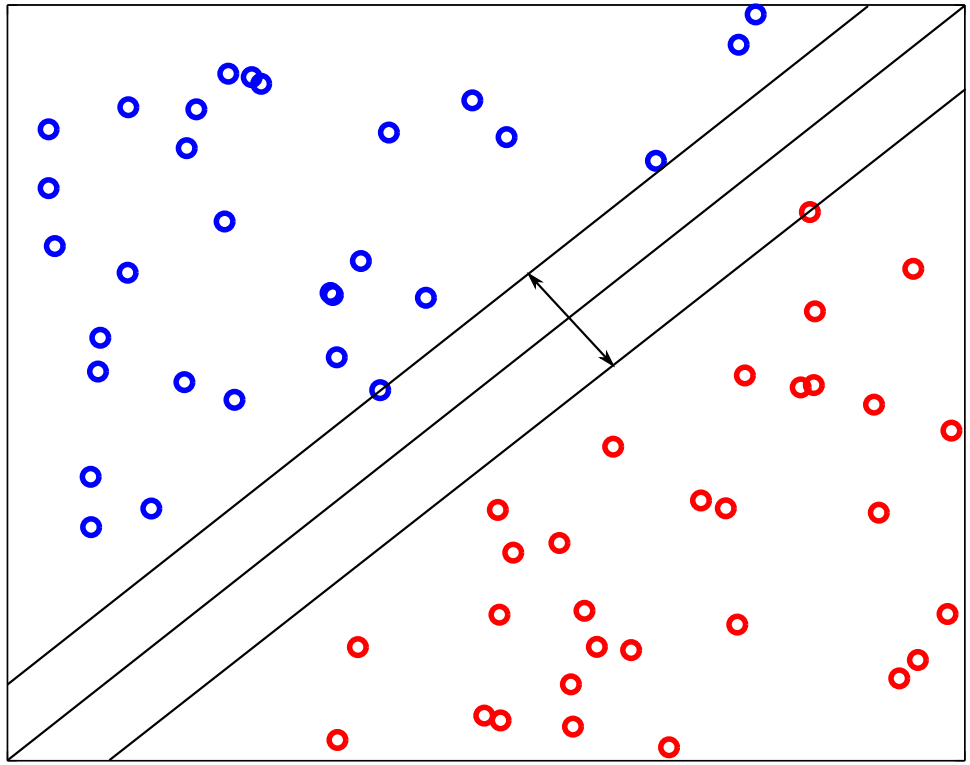
- Same setting as Example 1 but $y^i \in \{1, -1\}$ (have cancer or not)



- Want to linearly separate the two sets (diagnose the next patient)
- Countless many applications (medical diagnosis, OCR, spam filtering, fraud detection, marketing, image processing . . .)
- But which hyperplane do we choose?
- Intuitively, the margin is important

Example 3: Support Vector Machines

- Same setting as Example 1 but $y^i \in \{1, -1\}$ (have cancer or not)



- Want to linearly separate the two sets (diagnose the next patient)
- Countless many applications (medical diagnosis, OCR, spam filtering, fraud detection, marketing, image processing . . .)
- But which hyperplane do we choose?
- Intuitively, the margin is important
- We want the margin to be maximum

Support Vector Machines (cont.d)

- Distance of // hyperplanes (w_+, w_0) and (w_+, w_0') is $| w_0 - w_0' | / || w_+ ||$
- We can always take the hyperplane equidistant from the two sets
- $w_+ x^i + w_0 \geq 1$ if $y^i = 1$, $w_+ x^i + w_0 \leq -1$ if $y^i = -1$
- The maximum margin separating hyperplane is the solution of

$$\min_w \left\{ || w_+ ||^2 \ : \ y^i (w_+ x^i + w_0) \geq 1 \quad i = 1, \ldots, m \right\}$$

(margin $= 2/||w_+||$), assuming any exists

- General Support Vector Machine is

$$\begin{aligned} \min_{w,\xi} \ & || w_+ ||^2 + C \sum_{i=1}^m \xi_i \\ & y^i (w_+ x^i + w_0) \geq 1 - \xi_i \qquad i = 1, \ldots, m \\ & \xi_i \geq 0 \qquad i = 1, \ldots, m \end{aligned}$$

C weighs violation of separation against margin, "2" because yes

Support Vector Machines (cont.d)

- Equivalently, one can solve the dual problem

$$\begin{aligned} \max_{\alpha} \ & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \langle x^i, x^j \rangle \alpha_j \\ & \textstyle\sum_{i=1}^{m} y^i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C \qquad\qquad i = 1, \ldots, m \end{aligned}$$

  a convex quadratic program, hence "easy"

- Primal and dual optimal solutions are related by

$$w_+^* = \textstyle\sum_{i=1}^{m} \alpha_i^* y^i x^i$$

- Dual formulation $\Longrightarrow$ kernel trick: input space $\rightsquigarrow$ feature space

$$\langle x^i, x^j \rangle \longrightarrow \phi(x^i)\phi(x^k)$$

  where points are hopefully more linearly separable

- Efficient algorithms of many different kinds, ...

Wrapping up: what's this course about

- Learning as a computational, hence mathematical, process
- Mathematical foundations of many important learning processes
  $\equiv$ nonlinear optimization and numerical analysis techniques
- Easy problems and/or local optimal solutions, because
  size is huge (hard because large, not hard because hard)
- Besides, in ML the global optimal solution can be bad!
- Emphasis on what can be done by linear algebra
- Emphasis on "easy" optimization problems:
  (linear,) quadratic, conic, convex
- Focus on methods and software tools
- Applications to be seen in "Machine Learning" and/or "Data Mining"
  (in parallel, you can do it, we talk to each other)