Computational Mathematics for Learning and Data Analysis: introduction to the course

Antonio Frangioni

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

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

A.Y. 2024/25

Outline

- 1 course (9 CFU/ECTS)
- 1 program
- 1 exam
- 2 related but $\neq$ areas of computational mathematics $\Longrightarrow$ 2 lecturers:

Federico Poloni (Numerical methods)

Dipartimento di Informatica, room 343
050 2213143, mailto:federico.poloni@unipi.it
https://www.di.unipi.it/~fpoloni
Office hours (ricevimento): upon request

Antonio Frangioni (Optimization)

Dipartimento di Informatica, room 327
050 2212789, mailto:frangio@di.unipi.it
https://www.di.unipi.it/~frangio
Office hours (ricevimento): Tuesday 9:00 – 11:00

- Course Schedule
  - Wed 16:00 – 18:00 (Fib. C1)
  - Thu 11:00 – 13:00 (Fib. C)
  - Fri 11:00 – 13:00 (Fib. M1)
- Web page: `https://elearning.di.unipi.it/course/view.php?id=990`
- Team for lectures: `https://teams.microsoft.com/l/team/19%3AXKHW23QFLHIctHJWqDjeYPQVhmqbXwhkVG_jRiwl96o1%40thread.tacv2/conversations?groupId=cb04d09e-0aae-419a-a3d2-9be2e9afe1c5&tenantId=c7456b31-a220-47f5-be52-473828670aa1`
- Exam: project (groups of 2) + oral exam
  Projects either "ML" or "no-ML", but no difference in work and grading

Outline

Logistic

Motivation

Contents

Wrap up

- Huge amounts of data is generated and collected, but one has to make sense of it in order to use it: that's what learning is
- Take something big (data) and therefore unwieldy and produce something small and nimble that can be used in its stead ("actionable")
- That's a (mathematical) model
- Word comes from "modulus", diminutive from "modus" = "measure": "small measure", "measure in the small" (small is good)
- Known uses in architecture: proving beforehand that the real building won't collapse (e.g., Filippo Brunelleschi for the Cupola of the Cathedral of Florence)
- Countless many physical models afterwards (planes, cars, ...), but mathematics is cheaper than bricks / wood / iron ...
- Yet, mathematical problems can be difficult, too, for various reasons (and, of course, only truly viable after computers)
- Most of them will (likely) remain difficult for quantum computers `https://www.smbc-comics.com/comic/the-talk-3`

Choosing a mathematical model

- How a mathematical model should be:

Choosing a mathematical model

- How a mathematical model should be:
  1. accurate (describes well the process at hand)

Choosing a mathematical model

- How a mathematical model should be:
  1. accurate (describes well the process at hand)
  2. computationally inexpensive (gives answers rapidly)

Choosing a mathematical model

- How a mathematical model should be:
  1. accurate (describes well the process at hand)
  2. computationally inexpensive (gives answers rapidly)
  3. general (can be applied to many different processes)

Choosing a mathematical model

- How a mathematical model should be:
  1. accurate (describes well the process at hand)
  2. computationally inexpensive (gives answers rapidly)
  3. general (can be applied to many different processes)

  Typically impossible to have all three $\Longrightarrow$ choice crucial!

- How a mathematical model should be:
  1. accurate (describes well the process at hand)
  2. computationally inexpensive (gives answers rapidly)
  3. general (can be applied to many different processes)

  Typically impossible to have all three $\Longrightarrow$ choice crucial!
- Two fundamentally different model building approaches:
  1. analytic: model each component of the system separately + their interactions, ($\approx$)accurate but hard to construct (need system access + technical knowledge)
  2. data-driven / synthetic: don't expect the model to closely match the underlying system, just to be simple and to ($\approx$)accurately reproduce its observed behaviour

- How a mathematical model should be:
  1. accurate (describes well the process at hand)
  2. computationally inexpensive (gives answers rapidly)
  3. general (can be applied to many different processes)

  Typically impossible to have all three $\Longrightarrow$ choice crucial!
- Two fundamentally different model building approaches:
  1. analytic: model each component of the system separately + their interactions, ($\approx$)accurate but hard to construct (need system access + technical knowledge)
  2. data-driven / synthetic: don't expect the model to closely match the underlying system, just to be simple and to ($\approx$)accurately reproduce its observed behaviour
- All models are approximate (the map is not the world), but for different reasons
- Analytic models: flexible shape, (relatively) few "hand-chosen" parameters
- Synthetic models: rigid shape, (very) many automatically chosen parameters
- Fitting: find the parameters of the model that best represents the phenomenon, clearly some sort of optimization problem (often a 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!

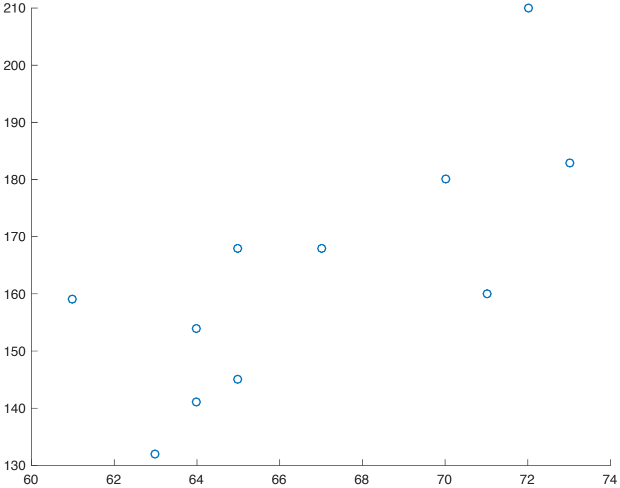
- A phenomenon measured by one number y is believed to depend on a vector $x = [\, x_1 \,, \, \dots \,, \, x_n \,]$ of other numbers
- Available (hopefully, large) set of observations $(\, y^1 \,, \, x^1 \,), \dots, (\, y^m \,, \, x^m \,)$
- Horribly optimistic assumption: the dependence is linear, i.e.,

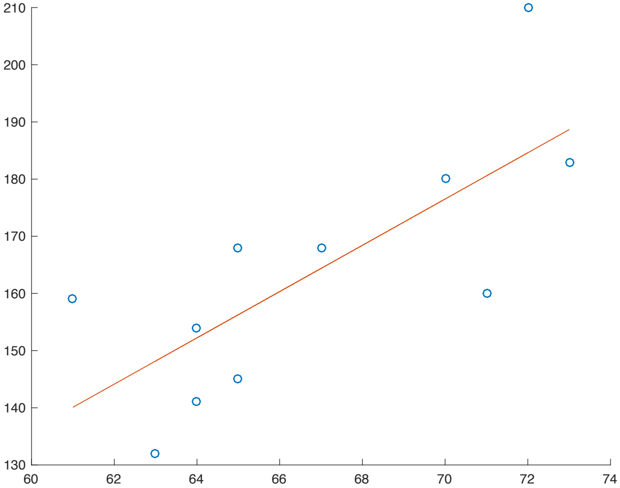
  $y = \sum_{i=1}^{n} w_i x_i + w_0 = wx + w_0$

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

$$y = \begin{bmatrix} y^1 \\ \vdots \\ y^m \end{bmatrix} \;,\; X = \begin{bmatrix} 1 & x^1 \\ \vdots & \vdots \\ 1 & x^m \end{bmatrix} \quad , \quad \min_{w} \mathcal{L}(\, w \,) = \|\, y - Xw \,\|$$

- Minimize loss function $\mathcal{L}(\, w \,) = \|\, y - Xw \,\| \equiv$ empirical risk $\equiv$ how much the model fails the predict the phenomenon on the available observations
- Simple closed formula: $X^T X w = X^T y \implies w = (X^T X)^{-1} X^T y$

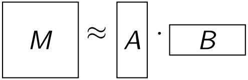

210
200
190
180
170
160
150
140
130
60
62
64
66
68
70
72
74



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

Example 2: Low-rank approximation

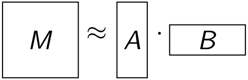
- A (large, sparse) matrix $M \in \mathbb{R}^{n \times m}$ describes a phenomenon depending on pairs (e.g., objects chosen from customers)
- Find "tall and thin" $A \in \mathbb{R}^{n \times k}$ and "fat and large" $B \in \mathbb{R}^{k \times m}$ $(k \ll n, m)$ s.t. $M \approx AB$ $\equiv$ find a few features that describe most of users' choices



$$\min_{A,B} \mathcal{L}(A, B) = ||\, M - AB \,||$$

- Minimize loss $\mathcal{L}(A, B) = ||\, M - AB \,|| \equiv$ "amount of unexplained choices"
- Many applications (neural networks, community analysis, ...)
- A, B can be obtained from eigenvectors of $M^T M$ and MM^T ...

- A (large, sparse) matrix $M \in \mathbb{R}^{n \times m}$ describes a phenomenon depending on pairs (e.g., objects chosen from customers)
- Find "tall and thin" $A \in \mathbb{R}^{n \times k}$ and "fat and large" $B \in \mathbb{R}^{k \times m}$ ($k \ll n, m$) s.t. $M \approx AB$ $\equiv$ find a few features that describe most of users' choices



$$\min_{A,B} \mathcal{L}(A, B) = \| M - AB \|$$

- Minimize loss $\mathcal{L}(A, B) = \| M - AB \| \equiv$ "amount of unexplained 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
- Efficiently solving this problem requires:
  1. low-complexity computation (of course)
  2. avoiding ever explicitly forming $M^T M$ and MM^T (too much memory)
  3. exploiting structure of M (sparsity, similar columns, ...)
  4. ensuring the solution is numerically stable

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

Original (512×512)

$k = 1$

$k = 10$

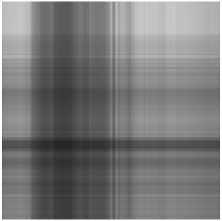
$k = 25$

$k = 50$

$k = 100$

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

Original (512×512)



$k = 1$

$k = 10$

$k = 25$

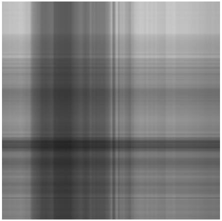
$k = 50$

$k = 100$

Low-rank approximation for image compression

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

Original (512×512)



$k = 1$

$k = 10$

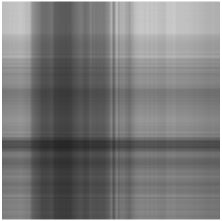
$k = 25$

$k = 50$

$k = 100$

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

Original (512×512)



$k = 1$

$k = 10$

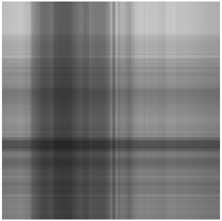
$k = 25$

$k = 50$

$k = 100$

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

Original (512×512)



$k = 1$

$k = 10$

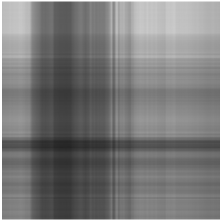
$k = 25$

$k = 50$

$k = 100$

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

Original (512×512)



$k = 1$

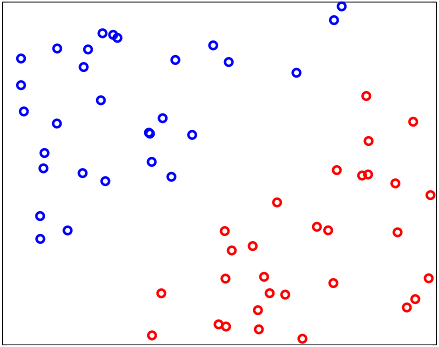
$k = 10$

$k = 25$

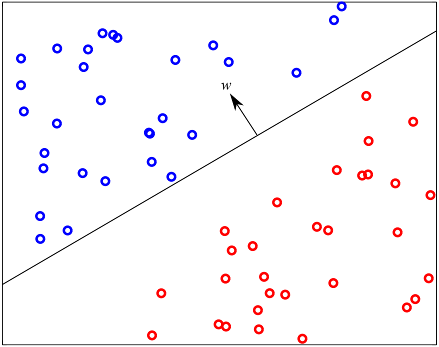
$k = 50$

$k = 100$

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

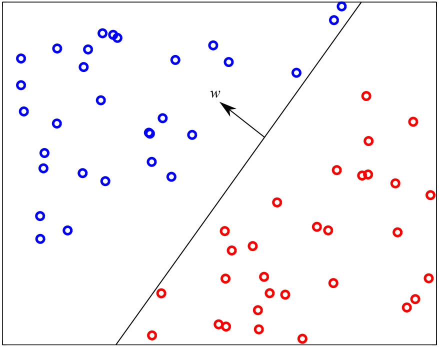


- Same setting as Example 1 but $y^h \in \{1, -1\}$ (have cancer or not)



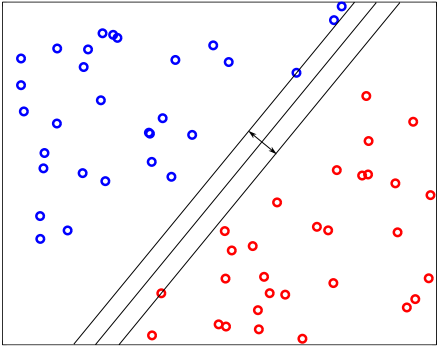
- Want to linearly separate the two sets (diagnose the next patient)
- Countless many applications (medical diagnosis, OCR, spam filtering, fraud detection, marketing, image processing ...)

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



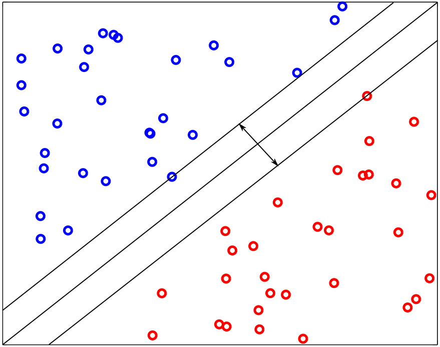
- Want to linearly separate the two sets (diagnose the next patient)
- Countless many applications (medical diagnosis, OCR, spam filtering, fraud detection, marketing, image processing ...)
- But which hyperplane do we choose?

- Same setting as Example 1 but $y^h \in \{ 1 , -1 \}$ (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 (and theory supports the intuition)

- Same setting as Example 1 but $y^h \in \{1, -1\}$ (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 (and theory supports the intuition)
- Larger margin $\Longrightarrow$ more "robust" classification

- Distance of // hyperplanes $(\, w_+ \,,\, w_0 \,)$ and $(\, w_+ \,,\, w_0' \,)$ is $|\, w_0 - w_0' \,| \,/\, \|\, w_+ \,\|$
- Can always take the hyperplane in "the middle" + scale w

  $\Longrightarrow\;\; w_+ x^h - w_0 \geq 1$ if $y^h = 1$, $w_+ x^h - w_0 \leq -1$ if $y^h = -1$
- The maximum margin separating hyperplane is the solution of

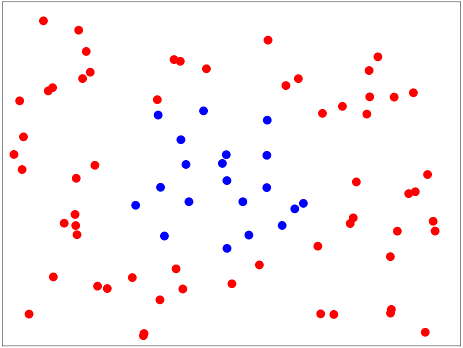
  $$\min_w \{\, \|\, w_+ \,\|^2 \;:\; y^h(w_+ x^h - w_0) \geq 1 \quad h = 1, \ldots, m \,\}$$

  (margin $= 2 \,/\, \|\, w_+ \,\|$, "2" because I say so), assuming any exists
- What if it does not? Support Vector Machine

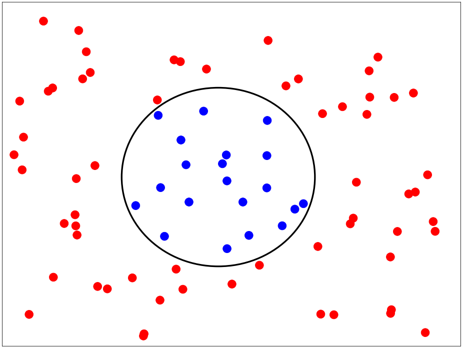
  (SVM-P) $\min_w \{\, \|\, w_+ \,\|^2 + C\mathcal{L}(\, w \,) = \sum_{h=1}^{m} \max\{\, 1 - y^h(w_+ x^h - w_0) \,,\, 0 \,\} \,\}$
- $\|\, w_+ \,\| \approx$ model complexity, the less the more chances it generalises well

  $\Longrightarrow$ C weighs $\mathcal{L}() =$ loss (of separation) on current data w.r.t. (hopefully) on future data: bias/variance dilemma (not really our business)
- $\mathcal{L}$ convex but nondifferentiable: reformulation with (many linear) constraints

  $$\text{(SVM-P)} \quad \min_{w,\xi} \|\, w_+ \,\|^2 + C \sum_{h=1}^{m} \xi_h$$
  $$y^h(w_+ x^h - w_0) \geq 1 - \xi_h \;\;,\;\; \xi_h \geq 0 \qquad h = 1, \ldots, m$$

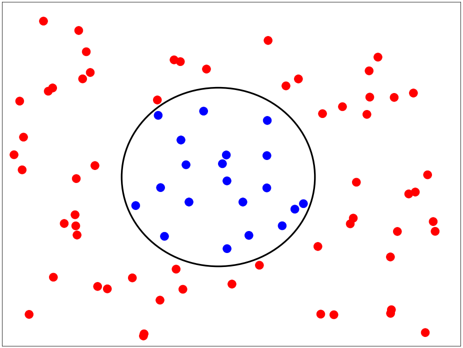


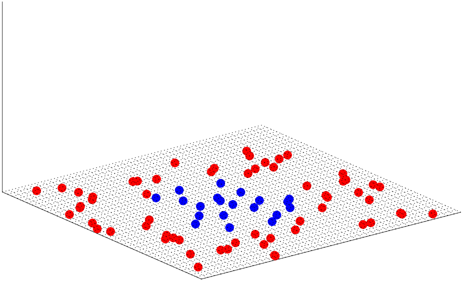
- (Approximate) linear separability



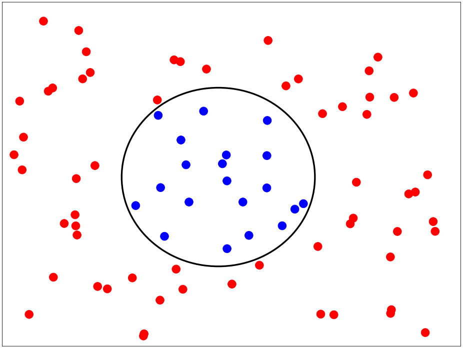
- (Approximate) linear separability rare, (approximate) linear regression weak

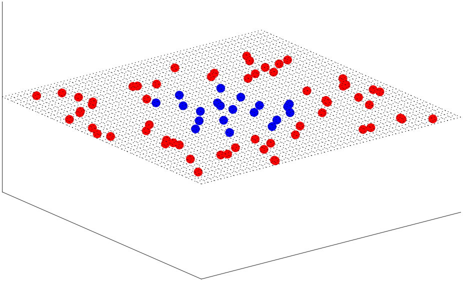
SVM: the problem, and the solution



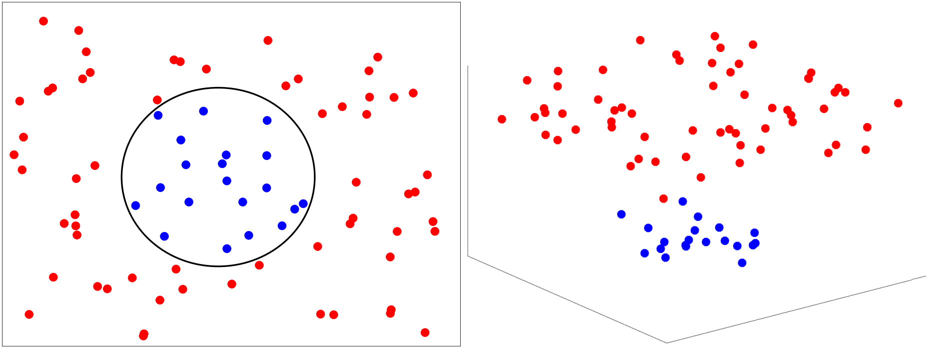


- (Approximate) linear separability rare, (approximate) linear regression weak
- Idea: embed in larger space

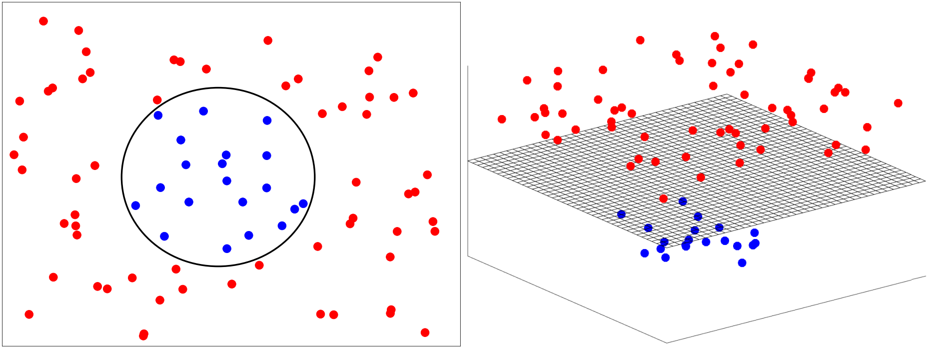




- (Approximate) linear separability rare, (approximate) linear regression weak
- Idea: embed in larger space

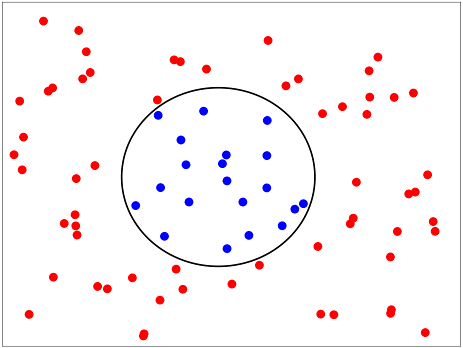


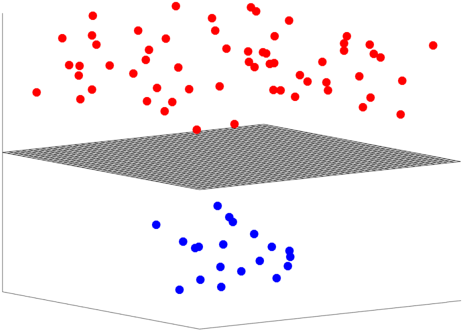
- (Approximate) linear separability rare, (approximate) linear regression weak
- Idea: embed in larger space nonlinearly, then



- (Approximate) linear separability rare, (approximate) linear regression weak
- Idea: embed in larger space nonlinearly, then linear function may work

SVM: the problem, and the solution





- (Approximate) linear separability rare, (approximate) linear regression weak
- Idea: embed in larger space nonlinearly, then linear function may work
- Doing this effectivey (how to embed) and efficiently nontrivial

- Equivalently, one can solve the dual problem (??? what ???)

$$\text{(SVM-D)} \quad \max_{\alpha} \; \sum_{h=1}^{m} \alpha_h - \tfrac{1}{2} \sum_{h=1}^{m} \sum_{k=1}^{m} \alpha_h \langle\, x^h \,,\, x^k \,\rangle \alpha_k$$
$$\sum_{i=1}^{m} y^h \alpha_h = 0$$
$$0 \leq \alpha_h \leq C \qquad h = 1, \ldots, m$$

  a convex constrained quadratic program, but with "simple constraints"

- Solve one problem by solving an apparently different one:
  α^* optimal for (SVM-D) $\implies$ $w_+^* = \sum_{h=1}^{m} \alpha_h^* y^h x^h$ optimal for (SVM-P)

- Dual formulation $\implies$ kernel trick: input space $\rightsquigarrow$ (larger) feature space

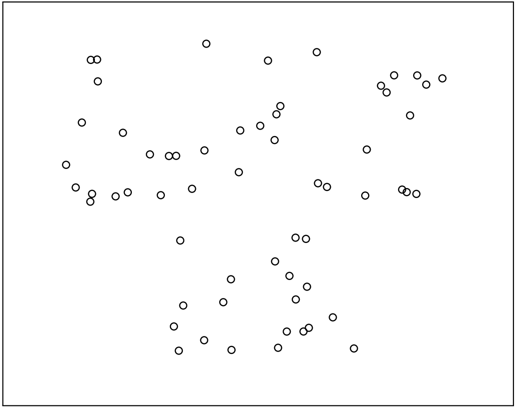
$$\langle\, x^h \,,\, x^k \,\rangle \;\rightsquigarrow\; \langle\, \phi(x^h) \,,\, \phi(x^k) \,\rangle$$

  where points are hopefully "more linearly separable"

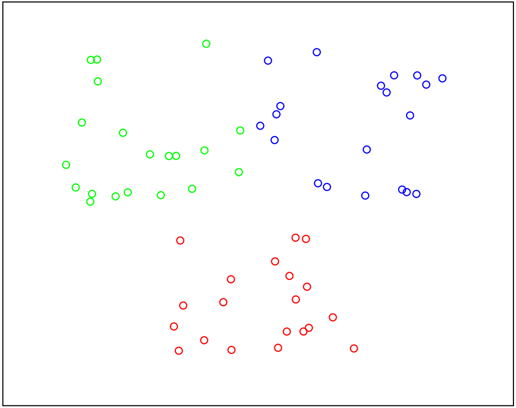
- Feature space can be infinite-dimensional, provided that scalar product can be (efficiently) computed

- Efficient algorithms: (SVM-P) or (SVM-D) (or both), complexity, ...

- $X = [\, x^i \in \mathbb{R}^h \,]_{i \in I}$ inputs, no outputs available $\equiv$ each x^i "looks the same"



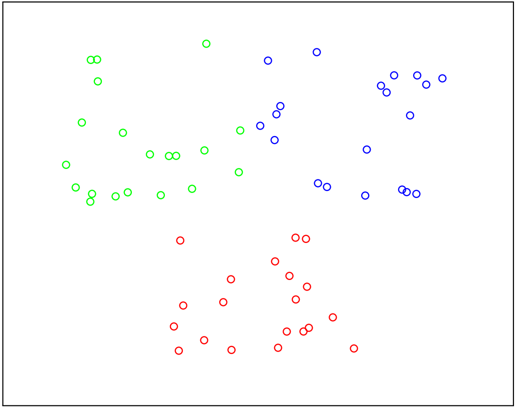
- $X = [\, x^i \in \mathbb{R}^h \,]_{i \in I}$ inputs, no outputs available $\equiv$ each x^i "looks the same"



- Given $k \in \mathbb{N}$ ($K = \{\, 1 \,, \,\ldots\, , \, k \,\}$), find $X = \bigcup_{p \in K} X^p$ $\equiv$ partition of X in clusters s.t. X^i that are homogeneous (??) and well separated (??)

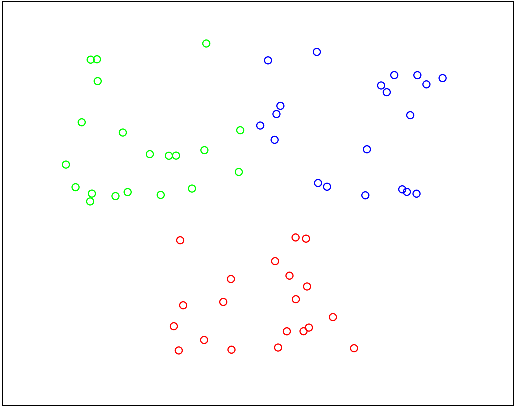
Example 4: clustering

- $X = [\, x^i \in \mathbb{R}^h \,]_{i \in I}$ inputs, no outputs available $\equiv$ each x^i “looks the same”



- Given $k \in \mathbb{N}$ ($K = \{\, 1 \,, \, \ldots \,, \, k \,\}$), find $X = \bigcup_{p \in K} X^p \;\equiv$ partition of X in clusters s.t. X^i that are homogeneous (??) and well separated (??)
- Crucial problem in unsupervisioned ML: automatically figure out the labels from the data, ill-defined by definition (many $\neq$ ways to label the same stuff)

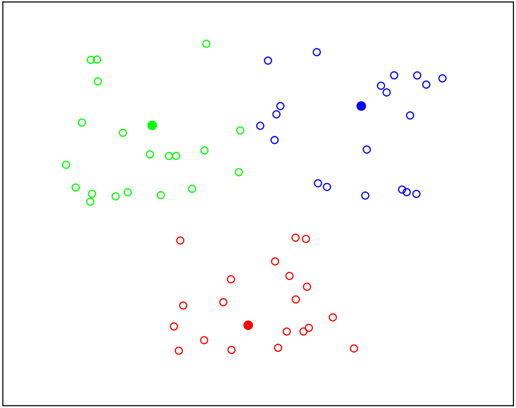
- $X = [\, x^i \in \mathbb{R}^h \,]_{i \in I}$ inputs, no outputs available $\equiv$ each x^i "looks the same"



- Many $\neq$ possible variants
- Simplest:

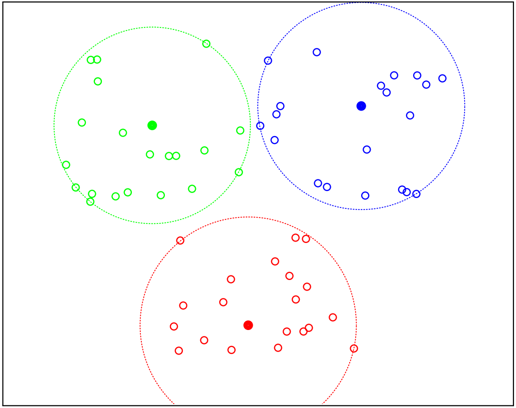
- Given $k \in \mathbb{N}$ ($K = \{\, 1 \,, \, \dots \,, \, k \,\}$), find $X = \bigcup_{p \in K} X^p \,\equiv$ partition of X in clusters s.t. X^i that are homogeneous (??) and well separated (??)
- Crucial problem in unsupervisioned ML: automatically figure out the labels from the data, ill-defined by definition (many $\neq$ ways to label the same stuff)

- $X = [\, x^i \in \mathbb{R}^h \,]_{i \in I}$ inputs, no outputs available $\equiv$ each x^i "looks the same"



- Many $\neq$ possible variants
- Simplest: define k centroids c^p $\equiv$ "archetypes" of each $x^i \in X^p$
- Given $k \in \mathbb{N}$ ($K = \{\, 1 \,, \,\ldots\, , \, k \,\}$), find $X = \bigcup_{p \in K} X^p$ $\equiv$ partition of X in clusters s.t. X^i that are homogeneous (??) and well separated (??)
- Crucial problem in unsupervisioned ML: automatically figure out the labels from the data, ill-defined by definition (many $\neq$ ways to label the same stuff)

- $X = [\, x^i \in \mathbb{R}^h \,]_{i \in I}$ inputs, no outputs available $\equiv$ each x^i "looks the same"

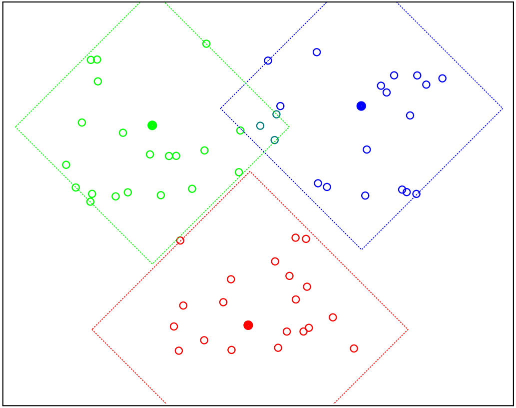


- Many $\neq$ possible variants
- Simplest: define k centroids c^p $\equiv$ "archetypes" of each $x^i \in X^p$
- $X^p = \{\, x^i \; : \;$ closer to c^p than to any other $c^q \,\}$
- Clusters (may) depend on the chosen norm $\equiv$ topology of $\mathbb{R}^h$
- Clusters in L_2

- Given $k \in \mathbb{N}$ ($K = \{\, 1 \,,\, \ldots \,,\, k \,\}$), find $X = \bigcup_{p \in K} X^p$ $\equiv$ partition of X in clusters s.t. X^i that are homogeneous (??) and well separated (??)

- Crucial problem in unsupervisioned ML: automatically figure out the labels from the data, ill-defined by definition (many $\neq$ ways to label the same stuff)

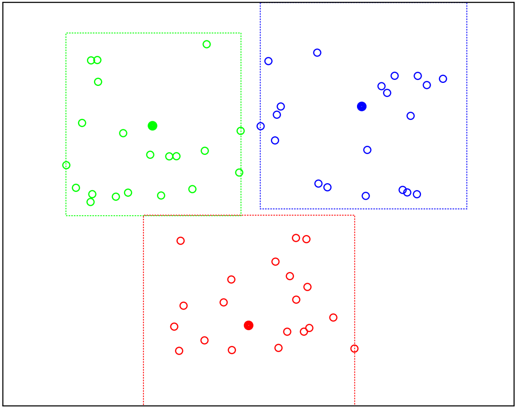
- $X = [\, x^i \in \mathbb{R}^h \,]_{i \in I}$ inputs, no outputs available $\equiv$ each x^i "looks the same"



- Many $\neq$ possible variants
- Simplest: define k centroids c^p $\equiv$ "archetypes" of each $x^i \in X^p$
- $X^p = \{\, x^i \;:\;$ closer to c^p than to any other $c^q \,\}$
- Clusters (may) depend on the chosen norm $\equiv$ topology of $\mathbb{R}^h$
- Clusters in $L_2 \neq$ in L_1

- Given $k \in \mathbb{N}$ ($K = \{\, 1 \,,\, \ldots \,,\, k \,\}$), find $X = \bigcup_{p \in K} X^p$ $\equiv$ partition of X in clusters s.t. X^i that are homogeneous (??) and well separated (??)
- Crucial problem in unsupervisioned ML: automatically figure out the labels from the data, ill-defined by definition (many $\neq$ ways to label the same stuff)

- $X = [\, x^i \in \mathbb{R}^h \,]_{i \in I}$ inputs, no outputs available $\equiv$ each x^i "looks the same"



- Many $\neq$ possible variants
- Simplest: define k centroids c^p $\equiv$ "archetypes" of each $x^i \in X^p$
- $X^p = \{\, x^i \;:\;$ closer to c^p than to any other $c^q \,\}$
- Clusters (may) depend on the chosen norm $\equiv$ topology of $\mathbb{R}^h$
- Clusters in $L_2 \neq$ in $L_1 \neq$ in L_∞

- Given $k \in \mathbb{N}$ ($K = \{\, 1 \,,\, \dots \,,\, k \,\}$), find $X = \bigcup_{p \in K} X^p$ $\equiv$ partition of X in clusters s.t. X^i that are homogeneous (??) and well separated (??)
- Crucial problem in unsupervisioned ML: automatically figure out the labels from the data, ill-defined by definition (many $\neq$ ways to label the same stuff)

- $c = [\, c^p \,]_{p \in K} \in \mathbb{R}^{hk}$, nonconvex and nonsmooth unconstrained model
  $\min\{\, f(\, c\,) = \sum_{i \in I} \min_{p \in K} \|\, c^p - x^i \,\|_2^2 \;:\; c \in \mathbb{R}^{hk} \}$

- Reformulation I: nonconvex, smooth, combinatorial, constrained model

$$
\begin{array}{lll}
\min & \sum_{i \in I} \sum_{p \in K} z_{ip} \|\, c^p - x^i \,\|_2^2 & \\
 & \sum_{p \in K} z_{ip} = 1 & i \in I \\
 & z_{ip} \in \mathbb{N} \;\; [\equiv \; \{\, 0 \,,\, 1 \,\}\,] & p \in K \;,\; i \in I
\end{array}
$$

  z_{ip} “logical” variables: 1 if x^i “assigned” to cluster p, 0 otherwise

- Two sources of nonconvexity: products zc in objective, integrality constraints

- But perfect structure for alternating minimization approaches:
  convex ($\equiv$ easy) in z if c fixed, convex in c if z fixed

- z fixed, $I(\, z \,,\, p\,) = \{\, i \in I \;:\; z_{pi} = 1 \,\} \implies (c^p)^* = \sum_{i \in I(\, z \,,\, p\,)} x^i \,/\, \# I(\, z \,,\, p \,)$
  optimal centroid $\equiv$ mean of the points in the cluster

procedure $c = k\text{-}means\,(\, X\,,\, c\,,\, \varepsilon\,)$ // note: k implicit from size of c
for($v \leftarrow \infty$; ;) **do**
foreach($p \in K$) **do** $I(\,p\,) \leftarrow \emptyset$;
foreach($i \in I$) **do** $\bar{p} \leftarrow \text{argmin}\{\, \|\, c^p - x^i \,\|_2^2 \,:\, p \in K \,\}$; $I(\,\bar{p}\,) \leftarrow I(\,\bar{p}\,) \cup \{\, i \,\}$;
foreach($p \in K$) **do** $c^p \leftarrow \sum_{i \in I(\,p\,)} x^i \,/\, \# I(\,p\,)$; // note: $I(\,p\,) = \emptyset$ happens
$\bar{v} \leftarrow \sum_{p \in K} \sum_{i \in I(\,p\,)} \|\, c^p - x^i \,\|_2^2$;
if($v - \bar{v} \leq \varepsilon$) **then break**; **else** $v \leftarrow \bar{v}$;

- Special case of (block) Gauss-Seidel approach: $f(\, x^1\,,\, x^2\,,\, \ldots\,,\, x^k\,)$, iteratively optimize over each individual (group of) variable(s) x^p keeping the other variables fixed $\Longrightarrow$ can work in parallel
- Convenient if f convex over each x^p individually but not jointly on all x
- Can be proven to "work" (converge), ends in finitely many iterations
- Local approach to nonconvex problem $\Longrightarrow$ no guarantee of global optimality $\Longrightarrow$ initial centroids relevant issue in practice (attraction basin)

Outline

- …of Computational Mathematics for Learning and Data Analysis

- . . . of Computational Magic for Learning and Data Analysis

- There are two main quests in the course:
  1. get a general understanding of several different classes of numerical algorithms and their underlying mathematical principles
  2. be able to actually implement, debug, and tune a few of them
- Algorithms are mathematical objects $\Longrightarrow$ reasoning about algorithms often is proving theorems (+ some hand-waving)
- All the more when the algorithms deal with nontrivial mathematical objects
- This is (mostly) done in the optional "Mathematically speaking" slides
- Learning theorems' proofs by heart is not a subject of the exam, not even the few (very simple) ones we'll actually see in details during lectures
- But you will have a lot more fun if you face side quests seriously
- Exercises are there for the same reason

- 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, ...)
- Numerical methods for constrained optimization
- Software tools for numerical computations (Matlab, Octave, ...)
- Sparse hints to AI/ML applications

- Slides prepared by the lecturers + recording of lectures
- `Matlab` programs + data
- L.N. Trefethen, D. Bau *Numerical Linear Algebra*, SIAM, 1997
- J. Demmel *Applied Numerical Linear Algebra*, SIAM, 1996
- S. Boyd, L. Vandenberghe *Convex optimization*, Cambridge Un. Press, 2008 (http://web.stanford.edu/~boyd/cvxbook/)
- L. Eldén *Matrix Methods in Data Mining and Pattern Recognition*, SIAM, 2007
- M.S. Bazaraa, H.D. Sherali, C.M. Shetty *Nonlinear programming: theory and algorithms*, Wiley & Sons, 2006
- D.G. Luenberger, Y. Ye *Linear and Nonlinear Programming*, Springer International Series in Operations Research & Management Science, 2008
- J. Nocedal, S. Wright *Numerical Optimization*, Springer Series in Operations Research and Financial Engineering, 2006
- Lecture notes for the optimization forthcoming, at least partly available soon

Outline

Logistic

Motivation

Contents

Wrap up

- Learning as a computational, hence mathematical, process
- Mathematical foundations of many important learning processes
  $\equiv$ nonlinear optimization and numerical analysis techniques
- Easy problems (linear, quadratic, conic, convex) or local optima,
  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
- Focus on methods and software tools, theory only as needed to understand
- Applications to be seen in "Machine Learning" and/or "Data Mining"
  (in parallel, you can/are supposed to do it, we talk to each other)