# Seminar in Data Science

# Lecture 1: Overview

Laurent El Ghaoui

Seminar in Data Science and Information Technology, Summer 2020
TBSI – UC Berkeley

7/13/2020

# Outline

# Outline

# What is data science?

- *Descriptive* (unsupervised learning): "understand data"
  clustering, low-rank approximation, factor analysis, filling missing data, outliers removal
- *Predictive* : "forecast the future"
  regression, classification, & deep learning approaches to those
- *Prescriptive* : "make decisions"
  portfolio optimization, control & reinforcement learning for investment or construction planning / decision, automated driving, control of nuclear reactors, etc

Currently a lot of discussion is around the first two (the "machine learning" part), and the last is mostly mentioned in the context of robotics (*e.g.*, self-driving cars). This course makes the case that a lot is to be gained from a comprehensive view where all three components are included.

# Sources of data

- *structured:* polls, company data, commercial transactions, credit card, balance sheets, etc. Comes as tables of numbers.
- *unstructured:* text (press releases, news, blogs, social media, etc), graphs, satellite images, traffic data, earnings calls transcripts, videos, etc.

In practice, we may not have as much *relevant* data as often touted.

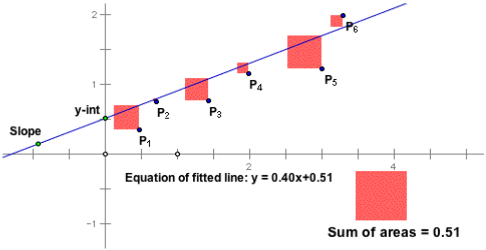
# Outline

# The four axes

- Unsupervised learning: “understand data structure”
- Supervised learning: “predict future outcomes”
- Deep learning: “learn features” in data
- Optimization & reinforcement learning: “learn decisions”

# Example

## Least-squares regression



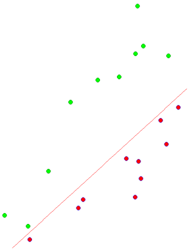

$$\min_w \|X^T w - y\|_2$$

where

- $X = [x_1, \ldots, x_m]$ is a $n \times m$ matrix of data points ($x_i \in \mathbf{R}^n$);
- $y$ is a response vector;
- $\|z\|_2 := \sqrt{z_1^2 + \ldots + z_m^2}$ is the $l_2$ (*i.e.*, Euclidean) norm of a vector $z \in \mathbf{R}^m$.
- Many variants (with *e.g.*, constraints) exist (more on this later).
- Perhaps the most popular / useful optimization problem.

# Example

## Linear classification



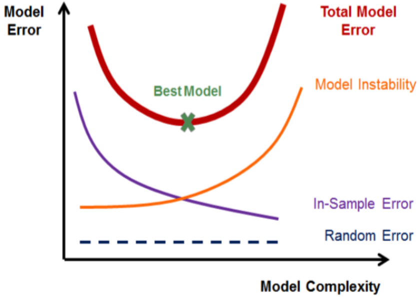
$$\min_{w,b} \sum_{i=1}^{m} \max(0, 1 - y_i(w^T x_i + b))$$

where

- $X = [x_1, \ldots, x_m]$ is a $n \times m$ matrix of data points ($x_i \in \mathbf{R}^n$);
- $y \in \{-1, 1\}$ is a *binary* response vector.
- A new data point is classified as $\hat{y}(x) = \mathbf{sign}(w^T x + b)$.

- Many variants (with *e.g.*, constraints) exist (more on this later).
- Very useful for *e.g.* sentiment analysis.

# How to evaluate results




- In supervised learning, we can reserve a part of the available data to test a model trained on the remaining part. There is a trade-off between model complexity and error.
- In unsupervised learning, there is no such "yardstick". One way is to consider the stability of the result with respect to perturbations in data. (More on this later.)

# Outline

# What is optimization?

Optimization is a field of applied mathematics also known as “mathematical programming”.

It is a *language* that allows to describe precisely how a decision should be made.

It includes as special cases:

- Machine learning problems: the decision may be about what prediction rule to use, in order to predict alpha or sentiment;
- Decision problems: Portfolio optimization.

Most machine learning problems can be viewed as a special case of an optimization problem.

- This connection allows to design algorithms (*e.g.*, stochastic gradient) to solve ML problems.
- It allows points to a better understanding of how to design models (*e.g.*, take into account prediction errors within a portfolio optimization problem).

# Optimization problem

## A standard form

An optimization problem is a problem of the form

$$p^* := \min_x \; f_0(x) \text{ subject to } f_i(x) \leq 0, \quad i = 1, \ldots, m,$$

where

- $x \in \mathbf{R}^n$ is the *decision variable* ;
- $f_0 : \mathbf{R}^n \to \mathbf{R}$ is the *objective* (or, *cost* ) function;
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$ represent the *constraints* ;
- $p^*$ is the *optimal value* .

Often the above is referred to as a “mathematical program” (for historical reasons).

# Example

## A short-term financing problem

A company faces the following *net cash flow requirements:*

| Month | Jan | Feb | Mar | Apr | May | Jun |
|---|---|---|---|---|---|---|
| Net cash flow (in $ k) | $-150$ | $-100$ | 200 | $-200$ | 50 | 300 |

Available *sources of funds:*

- Line of credit (max 100k, interest rate 1% per month);
- In any of the first 3 months it can issue 90-day commercial paper bearing a total interest of 2% for the 3-month period;
- Excess funds can be invested at 0.3% per month.

# Example

## A short-term financing problem: decision problem

*Variables* :

- Balance on the credit line $x_i$ for month $i = 1, 2, 3, 4, 5$.
- Amount $y_i$ of commercial paper issued ($i = 1, 2, 3$).
- Excess funds $z_i$ for month $i = 1, 2, 3, 4, 5$.
- $z_6$, the company's wealth in June.

*Decision problem:*

$$\text{maximize } z_6 \text{ subject to } \begin{cases} \text{Bounds on variables,} \\ \text{Cash-flow balance equations.} \end{cases}$$

# Example

## A short-term financing problem: constraints

- Non-negativity: $x_i \geq 0$, $i = 1, \ldots, 6$; $z_i \geq 0$, $i = 1, \ldots, 6$; $y_i \geq 0$, $i = 1, 2, 3$.
- Upper bounds on $x_i$'s: $x_i \leq 100$, $i = 1, \ldots, 5$.
- Cash flow balance equations.

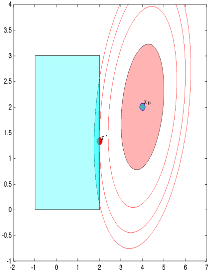
Linear programming formulation:

$$
\begin{array}{ll}
\max\limits_{x,y,z} & z_6 \\
\text{s.t.} & x_1 + y_1 - z_1 = 150, \\
& x_2 + y_2 - 1.01x_1 + 1.003z_1 - z_2 = 100, \\
& x_3 + y_3 - 1.01x_2 + 1.003z_2 - z_3 = -200, \\
& x_4 - 1.02y_1 - 1.01x_3 + 1.003z_3 - z_4 = 200, \\
& x_5 - 1.02y_2 - 1.01x_4 + 1.003z_4 - z_5 = -50, \\
& -1.02y_3 - 1.01x_5 + 1.003z_5 - z_6 = -300, \\
& 100 \geq x_i \geq 0, \quad i = 1, \ldots, 5, \\
& y_i \geq 0, \quad i = 1, 2, 3, \\
& z_i \geq 0, \quad i = 1, \ldots, 6.
\end{array}
$$

# Nomenclature

## A toy optimization problem

$$\begin{array}{ll} \min_{x} & 0.9x_1^2 - 0.4x_1x_2 + 0.6x_2^2 - 6.4x_1 - 0.8x_2 \\ \text{s.t.} & -1 \leq x_1 \leq 2, \quad 0 \leq x_2 \leq 3. \end{array}$$




- *Feasible set* in light blue.
- 0.1- *suboptimal set* in darker blue.
- *Unconstrained minimizer* : $x_0$; optimal point: $x^*$.
- *Level sets* of objective function in red lines.
- A *sub-level set* in red fill.

# Other standard forms

*Equality constraints.* We may single out equality constraints, if any:

$$\min_x \; f_0(x) \text{ subject to } \begin{array}{l} h_i(x) = 0, \quad i = 1, \ldots, p, \\ f_i(x) \leq 0, \quad i = 1, \ldots, m, \end{array}$$

where $h_i$'s are given. Of course, we may reduce the above problem to the standard form above, representing each equality constraint by a pair of inequalities.

*Abstract forms.* Sometimes, the constraints are described abstractly via a set condition, of the form $x \in \mathcal{X}$ for some subset $\mathcal{X}$ of $\mathbf{R}^n$. The corresponding notation is

$$\min_{x \in \mathcal{X}} \; f_0(x).$$

# Minimization vs. maximization

Some problems come in the form of maximization problems. Such problems are readily cast in standard form via the expression

$$\max_{x \in \mathcal{X}} f_0(x) = -\min_{x \in \mathcal{X}} : g_0(x),$$

where $g_0 := -f_0$.

- *Minimization* problems correspond to loss, cost or risk minimization.
- *Maximization* problems typically correspond to utility or return (*e.g.*, on investment) maximization.

# Penalization

A trade-off between two objectives is commonly accomplished via a *penalized* problem:

$$\min_{x} \; f(x) + \lambda g(x),$$

where $f$ and $g$ represent loss and risk functions, and $\lambda > 0$ is a risk-aversion parameter.

*Example:* penalized least-squares

$$\min_{w} \; \|X^T w - y\|_2^2 + \lambda \|w\|_2^2$$

Here, the risk term $\|w\|_2^2$ controls the variance associated with random noise in the entries of $X$.

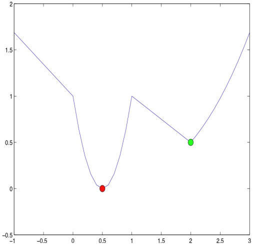
# Optimization in Finance
## Applications

- Machine learning:
  - Unsupervised learning: Market data analysis, covariance estimation and factor models, matrix completion, clustering.
  - Supervised learning: Model fitting, regression, classification, sentiment analysis.
- Decision-making:
  - Single-period: Portfolio optimization, asset allocation.
  - Multi-period: Portfolio optimization, asset liability management.
- Pricing and arbitrage detection: Static and dynamic.

# Outline

# Global vs. local minima

## The curse of optimization




- Point in red is globally optimal (optimal for short).
- Point in green is only locally optimal.
- In many applications, we are interested in global minima.

### Curse of optimization

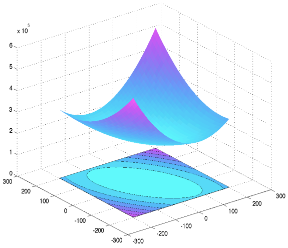
Optimization algorithms for general problems can be trapped in local minima.

# Convex function
## Definition

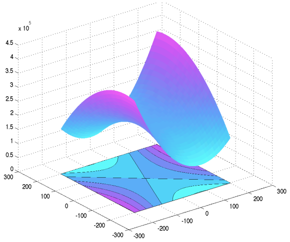
A function $f : \mathbf{R}^n \to \mathbf{R}$ is convex if it satisfies the condition

$$\forall\, x, y \in \mathbf{R}^n, \lambda \in [0, 1] \;:\; f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y)$$

Geometrically, the graph of the function is “bowl-shaped”.



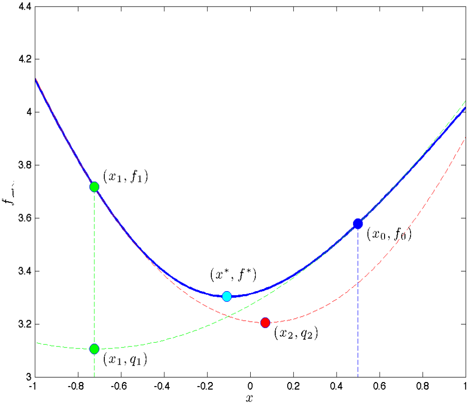
Convex function.



Non-convex function.

# Convexity and local minima

When trying to minimize convex functions, specialized algorithms will always converge to a global minimum, irrespective of the starting point, provided some (weak) assumptions on the function hold.




The Newton algorithm.

# Convex optimization
## Definition

The problem in standard form

$$p^* := \min_x \; f_0(x) \text{ subject to } f_i(x) \le 0, \quad i = 1, \dots, m,$$

is convex if the functions $f_0, \dots, f_m$ are all convex.

Examples:

- Linear programming ($f_0, \dots, f_m$ affine).
- Quadratic programming ($f_0$ convex quadratic, $f_1, \dots, f_m$ affine).
- Second-order cone programming ($f_0$ linear, $f_i$'s of the form $\|A_i x + b_i\|_2 + c_i^T x + d_i$, for appropriate data $A_i, b_i, c_i, d_i$).

# Software for convex optimization

- Free: CVX [3], Yalmip, Mosek (student version) [1].
- Really free: CVXPY [4] (in development).
- Commercial: Mosek, CPLEX, etc.

CVX syntax for cash-flow problem (assume data is in matrix $A$, vector $b$):

```
cvx_begin
variables x(5,1) y(3,1) z(6,1);
minimize( z(6) )
subject to
 A*[x;y;z] == b;
 x >= 0; x <= 100;
 y >= 0;
 z >= 0;
cvx_end
```

# Non-convex problems

## Examples

- *Boolean/integer optimization:* some variables are constrained to be Boolean or integers. Convex optimization can be used for getting (sometimes) good approximations.
- *Cardinality-constrained problems:* we seek to bound the number of non-zero elements in a vector variable. Convex optimization can be used for getting good approximations.
- *Non-linear programming:* usually non-convex problems with differentiable objective and functions. Algorithms provide only local minima. Includes as special case many machine learning prolems (*e.g.*, neural nets).

Not all non-convex problems are hard!

# Does convexity really matter?

In machine learning, convexity may not be a big deal; *e.g.*, ARIMA or neural net models are essentially non-convex, non-linear least-squares. Local minima are not usually an issue: a local minimum is “good enough”.

The main reason: *there are no constraints* in those problems.

When there are constraints, and the problem is not convex, the algorithms may not behave well (*e.g.*, may not find a feasible point, even though there exist one). Thus when it comes to portfolio optimization, convex models should be preferred.

# Outline

# Course goals

- Introduce you to the main concepts in machine learning and optimization.
- Illustrate the relevance of those concepts in (mostly) engineering.

# Course topics

- *Topic 1: Optimization models.* Basic optimization nomenclature, convex functions and sets. Linear and quadratic programming.

Next:

- *Linear algebra background.* Vectors and matrices, scalar product, mean and variance, eigenvalues and singular values, covariance matrices.
- *Unsupervised learning.* Clustering, principal component analysis, covariance matrix estimation, matrix completion, feature engineering.
- *Supervised learning.* Basics of prediction and classification. Least-squares regression, regularization, robust and quantile regression, auto-regressive and other time-series models, extensions.

# Course material and references

- *Lecture slides* (early version posted in advance). Make sure to check out the version posted after lecture.
- *Textbooks:*
  - G.C. Calafiore and L. El Ghaoui. *Optimization Models.* Cambridge, 2014.
    Introductory reference on optimization.
  - S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.
    In-depth treatment of convex models.
  - *Optimization models.* livebook available (for free) at `http://livebooklabs.com/keeppies/c5a5868ce26b8125`.
    A gentler introduction with many applications in engineering, finance, operations research, statistics.
- *Software:* we will rely on CVX (matlab toolbox for convex optimization, [3]) or its Python version, CVXPY [4].

# References (follow'd)


- T. Hastie, R. Tibshirani, J. Friedman. *The elements of statistical learning* . Springer, 2001.

  Good introduction to the fundamentals of machine learning, from a statistics viewpoint.
- I. Goodfellow , Y. Bengio and A. Courville. Deep learning.

  A reference on this hot topic.

# Homeworks

There will be a total of about three homeworks, most of which will require the use of software such as CVX [3, 4], or Mosek [1], all of which have free (student) versions.

Topics:

- *Homework 1* : Convexity; clustering; PCA, generalized low-rank models.
- *Homework 2* : Feature engineering. Kernel methods for supervised learning. Regression & classification.
- *Homework 3* : Optimization and robustness.

# Logistics

- *Instructor:* Laurent El Ghaoui (`elghaoui@berkeley.edu`)
- *TAs:* Zifeng Wang (`wangzf18@mails.tsinghua.edu.cn`)
- *Professor's office hours:* TBD.
- *Z.W.'s office hours:* TBD.
- *Grading:* 50 % homeworks, 50 % final.

*How do we communicate?* WeChat to Professor. **Always** cc Zifeng!

# Outline

# References I


E. D. Andersen and K. D. Andersen.
The MOSEK optimization package.

A. Bental, L. El Ghaoui, and A. Nemirovski.
*Robust Optimization*.
Princeton University Press, 2009.

S. Boyd and M. Grant.
The CVX optimization package, 2010.

S. Boyd and T. Tinoco.
The CVXPY optimization package, 2010.

Stephen Boyd and Lieven Vandenberghe.
*Convex optimization*.
Cambridge university press, 2004.

Giuseppe Calafiore and Laurent El Ghaoui.
*Optimization models*.
Cambridge University Press, October 2014.

G. Cornuejols and R. Tütüncü.
*Optimization Methods in Finance*.
Cambridge University Press, 2007.

Gerard Cornuejols and Reha Tütüncü.
*Optimization methods in finance*, volume 5.
Cambridge University Press, 2006.

# References II


Jerome Friedman, Trevor Hastie, and Robert Tibshirani.
*The elements of statistical learning*, volume 1.
Springer series in statistics Springer, Berlin, 2001.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
*Deep Learning*.
MIT Press, 2016.
http://www.deeplearningbook.org.