Univ. Grenoble Alpes
2018–2019

Data Science Master track
MSIAM, MoSIG

Refresher course: Numerical Matrix Analysis & Optimization

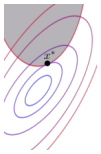
F. Iutzeler, J. Malick, and D. Grishchenko

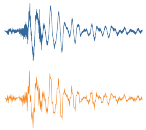
INTRODUCTION

Why matrix analysis and optimization ?

Matrix and optimization are at the **heart of computational mathematics**, With applications everywhere, e.g.

Machine Learning | Energy Management | Signal/Image Processing





	theory		**practice**
Mix between	meaning of a problem; existence/uniqueness of solution; math properties	and	Computability, speed, and use of standard libraries to solve numerically these problems.

This short course focuses on **matrix analysis and optimization in action** with exercises inspired from:

- Google PageRank, Image processing
- Machine learning applications (regression, classification)

This course is not a standard course on linear algebra or optimization

- not a math course
  basic knowledge is assumed (take a look to your undergraduate courses)
- not an algorithmic course
  basic programming skills are expected (check-out the Python tutorial in Chapter 1 at `https://github.com/iutzeler/refresher-course`))

This course is

- a review of basics of matrix analysis – from numerical perspective
- a short overview of numerical optimization
- includes quick recalls from matrix calculus and differential calculus

Contents

Organization

Teachers:

- Franck Iutzeler – Assistant Professor in Applied Maths
  `franck.iutzeler@univ-grenoble-alpes.fr`
- Dmitry Grishchenko – TA, PhD Candidate in Applied Maths and Computer Science
  `dmitry.grishchenko@univ-grenoble-alpes.fr`

Schedule:

Session	Date	Room	Topic
1.	Mon. 24th 9:45–12:45	Amphi D	Course & Tutorial on Matrix analysis
2.	Tue. 25th 9:45–12:45	Amphi D	Course & Tutorial on Matrix analysis
3.	Wed. 26th 9:45–12:45	E201	Practical session on Matrix analysis
4.	Wed. 26th 14:00–17:00	Amphi Esclangon	Course & Tutorial on Optimization
5.	Thu. 27th 9:45–12:45	Amphi D	Course & Tutorial on Optimization
6.	Fri. 28th 9:45–12:45	E100/E101	Practical session on Optimization

Tutorials: (Exercise sheet hereunder)

- to manipulate the notions on simple examples
  no fancy maths and no computations - they will be done on machines
- prepare the labs sessions...

Practical Session: (Python notebooks 2-1 and 2-2 at `https://github.com/iutzeler/refresher-course`)

- matrices and optimization in action for learning and ranking
- with Python & Numpy, by groups of 1,2 or 3
  Take your login or your own machine (with iPython installed!)

To go further

This course introduces **material** for several courses, among them:

- "Efficient methods in optimization"
  (on convex analysis & complexity and convergence of algorithms)
- "Convex and distributed optimization" (for large-scale applications & big data)
- 3 courses on machine learning !
- PDEs, inverse problems, stats...

Useful Links:

- Python/Numpy's documentation `http://docs.scipy.org/doc/numpy-1.11.0/reference/`
- Stephen's Boyd website (check the courses, quizzes, and exercises) `http://web.stanford.edu/~boyd/`

Main References:

- Horn, R. & Johnson, C.: *Matrix analysis.*
- Boyd, S. & Vandenberghe, L.: *Convex optimization.*
- Ciarlet, Ph.: *Introduction à l'analyse numérique et l'optimisation.*
- Hiriart-Urruty J.-B., Lemaréchal C.: *Fundamentals of convex analysis.*

EXERCISES

Part I – Matrix Analysis

A. Decompositions

◦ **A.1** (Rank 1 matrices)**.**

a. Justify why a rank 1 matrix A can always be written $A = uv^{\mathrm{T}}$.

b. Express matrix $B = \begin{bmatrix} 1 & 2 & 5 \\ 2 & 4 & 10 \\ 0 & 0 & 0 \end{bmatrix}$ as the product of two vectors: $B = uv^{\mathrm{T}}$.

c. Compute eigenvalues and associated eigenvectors of matrix B.

d. Justify why B is rank 1.

e. Show that $\det(I + uv^{\mathrm{T}}) = 1 + v^{\mathrm{T}}u$.
(hint: verify that $\begin{bmatrix} I & 0 \\ v^T & 1 \end{bmatrix} \begin{bmatrix} I + uv^T & u \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ -v^T & 1 \end{bmatrix} = \begin{bmatrix} I & u \\ 0 & 1 + v^T u \end{bmatrix}$*)*

f. Show that $(I + uv^{\mathrm{T}})^{-1} = I - \frac{uv^{\mathrm{T}}}{1+v^{\mathrm{T}}u}$. (This identity is called Sherman–Morrison formula)

B. Linear Systems Resolution with applications to Regression

In this example, that we will also study in the Labs, we use linear algebra to extract information from data; more precisely, we predict final notes of a group of student from their profiles[1].

Profiles include features such as student grades, demographic, social and school related features and were collected by using school reports and questionnaires. We wish to predict the final grade by a *good* linear combination of the features.

Mathematically, from the *learning matrix* A of size $n \times d$, $n \geq d$, comprising of the features values of each training student in line, and the vector of the values of the target features b; we seek a *regression vector* that minimizes the squared error between Ax and b. This problem boils down to the following least square problem:

$$\min_x \|Ax - b\|_2^2. \tag{B.1}$$

◦ **B.1.** Using basic analysis[2], one can prove that any solution of Pb (B.1) verifies the following relation:

$$A^{\mathrm{T}}Ax = A^{\mathrm{T}}b.$$

a. Assume that A has full rank, show that there is a unique solution to Pb (B.1).

b. Express this solution using the singular value decomposition of A.

In the Lab, we are going to learn a linear predictor using linear regression over a part of the data called the *learning set* and we will check our prediction by comparing the results for the rest of the data, the *testing set*.

[1] The Student Performance data can be found at `https://archive.ics.uci.edu/ml/datasets/Student+Performance`; it include secondary education students of two Portuguese schools.

[2] this will be quickly covered in the optimization part.

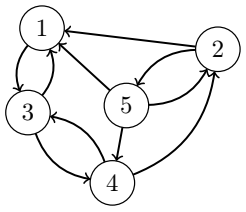
C. PageRank

The problem of ranking webpages is of the utmost importance for search engines. To this end, a very popular approach is to represent webpages as a graph where the nodes are the pages themselves and the edges are the links between them (if page i contains a links pointing toward page j, there is a directed edge from node i to node j in the graph). Then, a page/node has a high score (it is ranked high) if there are many links pointing toward it, especially coming from highly ranked pages. This approach is at the core of the PageRank algorithm developed in 1996 by Larry Page and Sergey Brin, the founders of Google. This exercise illustrates the mathematics used in this process by working on a simple example.

More formally, consider the graph of Fig. C.1. We could choose as a score the number of incoming links; node 1 is ranked first with 3, nodes $2, 3, 4$ are second with 2, 5 is last with 1. The limits of this scoring is that $2, 3, 4$ have the same score but are very different in nature as 3 is pointed by the most important page. To correct this phenomenon, the following scoring was introduced: the score x_i of page i is equal to the sum over the pages j pointing toward i of the scores (x_j) divided by their number of outgoing links n_j. Mathematically, the (implicit) score of page i is

$$x_i = \sum_{j \in \mathcal{P}_i} \frac{x_j}{n_j} \tag{C.1}$$

where $\mathcal{P}_i$ is the set of nodes pointing toward i (i itself is not in $\mathcal{P}_i$).



$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

FIGURE C.1. a simple graph and its incidence matrix

∘ **C.1** (Some algebraic graph theory)**.** The graph of Fig. C.1 can be represented by its incidence matrix A which verifies $A_{i,j} = 1$ if there is a link pointing towards i in page j, and $A_{i,j} = 0$ elsewhere.

a. Let $x \in \mathbb{R}^5$ be the vector of the pages scores. Write the score equation (C.1) as a linear equation $x = Rx$ where R is a matrix to define. Does a solution to this equation exist? Is it unique?
b. Show that matrix R is column stochastic, that is, its elements are non negative and its column sum is equal to one.
c. Deduce that 1 is an eigenvalue of R. (hint: one can show that it is an eigenvalue for R^{T}.)
d. Show that $\|R\| = 1$ for some matrix norm. Deduce that the sequence $(R^n)_n$ stays bounded and that the spectral radius of R is equal to 1.
e. Demonstrate that one can find an eigenvector for eigenvalue 1 with non-negative entries. *(hint: show that the sequence defined by $v_{k+1} = Av_k$ and $v_0 \geq 0$ give non-negative vectors)*

The above questions have led you to prove parts of a fundamental theorem in matrix analysis called the *Perron-Frobenius* theorem.

Theorem 1 (Perron-Frobenius theorem)**.** *Let A be a non-negative $n \times n$ matrix. Then,*

(i) *the spectral radius $\rho = \rho(A)$ is an eigenvalue;*
(ii) *there is a non-negative eigenvector $x \in \mathbb{R}^n_+$ such that $Ax = \rho x$;*
(iii) *if in addition A^k has all its entries (strictly) positive for some $k > 0$, then ρ is an eigenvalue of simple multiplicity and it is the only one of maximal modulus. Furthermore, there is a unique non-negative eigenvector x such that $Ax = \rho x$ and $\|x\|_1 = 1$. This vector is called the* Perron vector.

The additional condition of (iii) is often called *primitivity*.

D. Differentiability, Minima, and Convexity

◦ **D.1** (Quadratic functions)**.**

a. In $\mathbb{R}^n$, compute the gradient of the squared Euclidean norm $\|\cdot\|_2^2$ at a generic point $x \in \mathbb{R}^n$.
b. Let A be an $m \times n$ real matrix and b a size-m real vector. We define $f(x) = \|Ax - b\|_2^2$. For a generic vector $a \in \mathbb{R}^n$, compute the gradient $\nabla f(a)$ and Hessian $H_f(a)$.
c. Let C be an $n \times n$ real matrix, d a size-n real vector, and $e \in \mathbb{R}$. We define $g(x) = x^{\mathrm{T}}Cx + d^{\mathrm{T}}x + e$. For a generic vector $a \in \mathbb{R}^n$, compute the gradient $\nabla g(a)$ and Hessian $H_g(a)$.
d. Can all functions of the form of f and be written in the form of g? And conversely?

◦ **D.2** (Fundamentals of convexity)**.** This exercise proves and illustrates some results seen in the course.

a. Let f and g be two convex functions. Show that $m(x) = \max(f(x), g(x))$ is convex.
b. Show that $f_1(x) = \max(x^2 - 1, 0)$ is convex.
c. Let f be a convex function and g be a convex, non-decreasing function. Show that $c(x) = g(f(x))$ is convex.
d. Show that $f_2(x) = \exp(x^2)$ is convex. What about $f_3(x) = \exp(-x^2)$
e. Justify why the 1-norm, the 2 norm, and the squared 2-norm are convex.

◦ **D.3** (Strict and strong convexity)**.** A function $f : \mathbb{R}^n \to \mathbb{R}$ is said

- *strictly convex* if for any $x \neq y \in \mathbb{R}^n$ and any $\alpha \in]0, 1[$

$$f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y)$$

- *strongly convex* if there exists $\beta > 0$ such that $f - \frac{\beta}{2}\|\cdot\|_2^2$ is convex.

a. For a strictly convex function f, show that the problem

$$\begin{cases} \min f(x) \\ x \in C \end{cases}$$

where C is a convex set admits at most one solution.
b. Show that a strongly convex function is also strictly convex.
(hint: use the identity $\|\alpha x + (1-\alpha)y\|^2 = \alpha\|x\|^2 + (1-\alpha)\|y\|^2 - \alpha(1-\alpha)\|x-y\|^2$*.)*
c. Let f be a twice differentiable function. Show that f is strongly convex if and only if there exists $\beta > 0$ such that the eigenvalues of $\nabla^2 f(x)$ are larger than β for all x.
d. Discuss the strict and strong convexity of function f_1 and f_2 of D.2.

◦ **D.4** (Optimality conditions)**.** Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice differentiable function and $\bar{x} \in \mathbb{R}^n$. We suppose that f admits a local minimum at $\bar{x}$ that is $f(x) \geq f(\bar{x})$ for all x in a neighborhood[3] of $\bar{x}$.

a. For any direction $u \in \mathbb{R}^n$, we define the $\mathbb{R} \to \mathbb{R}$ function $q(t) = f(\bar{x} + tu)$. Compute $q'(t)$.
b. By using the first order Taylor expansion of q at 0, show that $\nabla f(\bar{x}) = 0$.
c. Compute $q''(t)$. By using the second order Taylor expansion of q at 0, show that $\nabla^2 f(\bar{x})$ is positive semi-definite.
d. Give a necessary condition on $\nabla^2 f$ for f to be convex. Deduce a condition on C for g to be convex in question b of ◦ D.3 and make the connection with question d.

[3] Formally, one would write $\forall x \in \mathbb{R}^n$ such that $\|x - \bar{x}\| \leq \varepsilon$ for $\varepsilon > 0$ and some norm $\|\cdot\|$.

E. Gradient Algorithm

◦ **E.1** (Descent lemma)**.** A function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be L-smooth if it is differentiable and its gradient ∇f is L-Lipchitz continuous, that is

$$\forall x, y \in \mathbb{R}^n, \quad \|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|.$$

The goal of the exercise is to prove that if $f : \mathbb{R}^n \to \mathbb{R}$ is L-smooth, then for all $x, y \in \mathbb{R}^n$,

$$f(x) \leq f(y) + (x - y)^{\mathrm{T}} \nabla f(y) + \frac{L}{2}\|x - y\|^2$$

a. Starting from fundamental theorem of calculus stating that for all $x, y \in \mathbb{R}^n$,

$$f(x) - f(y) = \int_0^1 (x - y)^{\mathrm{T}} \nabla f(y + t(x - y))\mathrm{d}t$$

prove the descent lemma.

b. Give a function for which the inequality is tight and one for which it is not.

◦ **E.2** (Smooth functions)**.** Consider the constant stepsize gradient algorithm $x_{k+1} = x_k - \gamma \nabla f(x_k)$ on an L-smooth function f.

a. Use the *descent lemma* to prove convergence of the sequence $(f(x_k))_k$ when $\gamma \leq 2/L$.

b. Did you use at some point that the function was convex? Conclude about the convergence of the gradient algorithm on smooth non-convex functions.