

# Optimisation-RO

Clément Royer

Certificat Chef de Projet IA - Université Paris Dauphine-PSL

11 octobre 2022





Clément Royer

- Maître de conférences @ Dauphine-PSL;
- Chaire tremplin en optimisation @ PRAIRIE;
- `clement.royer@lamsade.dauphine.fr`



Florentin Goyens

- Chercheur post-doctorant @ Dauphine-PSL;
- Membre junior optimisation @ PRAIRIE;
- `florentin.goyens@dauphine.psl.eu`

## En attendant Moodle/EduSign

<https://github.com/clementwroyer/opt-ro-psl>

- Transparents de cours;
- Matériel illustratif (notebooks Python).

## Déroulé de la formation

- Mardi 11/10 après-midi (C. Royer) : Bases de l'optimisation et convexité;
- Mercredi 12/10 matin (C. Royer) : Optimisation non convexe;
- Mercredi 12/10 après-midi (F. Goyens) : Gradient stochastique;
- Jeudi 13/10 matin (C. Royer) : Optimisation sans dérivées;
- Jeudi 13/10 après-midi (F. Goyens) : Optimisation à grande échelle.

- 1 Introduction
- 2 Concepts de base en optimisation
- 3 Optimisation convexe

- 1 Introduction
- 2 Concepts de base en optimisation
- 3 Optimisation convexe

## Ce dont tout le monde parle

- Sciences des données (data science);
- Analyse de données (data analysis);
- Fouille de données (data mining);
- Apprentissage machine/profond (machine/deep learning);
- Intelligence artificielle (IA);
- Big Data;
- ...

## Ce dont tout le monde parle

- Sciences des données (data science);
- Analyse de données (data analysis);
- Fouille de données (data mining);
- Apprentissage machine/profond (machine/deep learning);
- Intelligence artificielle (IA);
- Big Data;
- ...

## Ce dont nous allons parler

- Optimisation pour la science des données en général...
- ...et pour l'IA en particulier (ou vice-versa).

## Un ensemble de problèmes basés sur des données

- Extraction d'information à partir de la donnée :  
*statistiques, attributs principaux, structures;*
- Utilisation de cette information pour la **prédiction** du comportement de données futures.



# Ce que sera l'IA pour nous

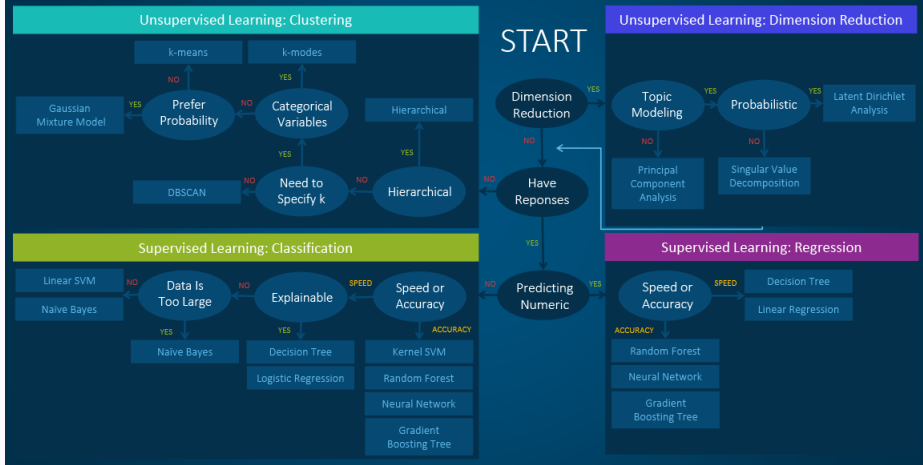
## Un ensemble de problèmes basés sur des données

- Extraction d'information à partir de la donnée : *statistiques, attributs principaux, structures*;
- Utilisation de cette information pour la **prédiction du comportement de données futures**.

## Composantes de l'IA/de la science des données

- Statistiques;
- Informatique (gestion de la donnée, calcul parallèle, etc);
- **Optimisation** pour la modélisation des problèmes et leur résolution par des algorithmes.

## Machine Learning Algorithms Cheat Sheet



Source : <https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-learning-algorithm-use/>

## Optimisation numérique

- Montée en puissance en 1970-1980;
- Succès des algorithmes en ingénierie (chimique, aéronautique, etc).
- *Pratique standard en calcul scientifique*: utiliser une méthode de points intérieurs (basée sur Newton, développée dans les années 2000s).

## Optimisation numérique

- Montée en puissance en 1970-1980;
- Succès des algorithmes en ingénierie (chimique, aéronautique, etc).
- *Pratique standard en calcul scientifique*: utiliser une méthode de points intérieurs (basée sur Newton, développée dans les années 2000s).

## Optimisation pour l'IA

- Problèmes basés sur de grands volumes de données;
- Les méthodes standard en optimisation ne sont pas les plus efficaces!

*Pratique classique en IA*: Utiliser une approche de gradient stochastique avec momentum (1950s + article théorique de 1983).

## Contexte de données massives/Big Data

- Les calculs usuels (fonction, dérivées) sont très coûteux car ils accèdent à **toute la donnée**.
- La précision souhaitée n'est pas forcément très grande en raison du bruit sur les données.

## Contexte de données massives/Big Data

- Les calculs usuels (fonction, dérivées) sont très coûteux car ils accèdent à **toute la donnée**.
- La précision souhaitée n'est pas forcément très grande en raison du bruit sur les données.

## Communauté de l'IA

- Le problème d'optimisation est souvent un moyen plus qu'une fin;
- Propriétés statistiques des solutions;
- Théorie et pratique différentes de la communauté d'optimisation "classique".

- 1 Introduction
  - Un exemple : classification binaire
- 2 Concepts de base en optimisation
- 3 Optimisation convexe

**Point de départ :** Jeu de données  $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ .

- $\mathbf{x}_i$  vecteur d'**attributs** à  $d$  coordonnées;
- $y_i$  **label** binaire égal à 1 ou  $-1$ .



**Point de départ :** Jeu de données  $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ .

- $\mathbf{x}_i$  vecteur d'**attributs** à  $d$  coordonnées;
- $y_i$  **label** binaire égal à 1 ou  $-1$ .

**Exemple :** classification de documents

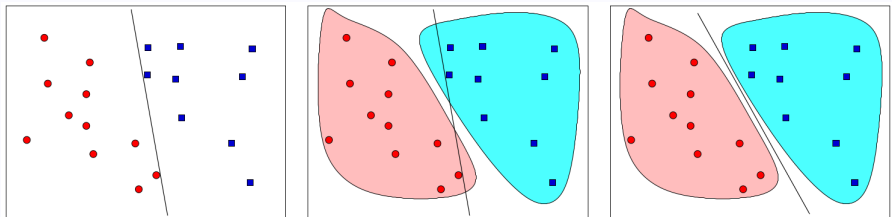
Soit un dictionnaire de  $d$  mots.

- $\mathbf{x}_i$  représente les mots contenus dans le document  $i$ :

$$[\mathbf{x}_i]_j = \begin{cases} 1 & \text{si le mot } j \text{ est dans le document } i, \\ 0 & \text{sinon.} \end{cases}$$

- $y_i$  égal à  $+1$  si le document traite de l'automobile, à  $-1$  sinon.

# Différentes solutions



Source : S. J. Wright, Optimization Algorithms for Data Analysis, 2018.

- Points :  $\mathbf{x}_i$ , rouges/bleus :  $y_i = 1/y_i = -1$ ;
- Nuages rouges/bleus : distribution des documents;
- Deux techniques de classification linéaires;
- Figure de droite : solution à marge maximale (SVM).

# Deux points de vue sur le problème

## Optimiseur

- Le problème peut être modélisé comme un programme quadratique convexe, et résolu de manière efficace;
- Potentiellement plusieurs solutions.

# Deux points de vue sur le problème

## Optimiseur

- Le problème peut être modélisé comme un programme quadratique convexe, et résolu de manière efficace;
- Potentiellement plusieurs solutions.

## Applicatif

- Le modèle doit s'appliquer à tous les documents de la distribution  $\Rightarrow$  généralisation;
- Mieux d'avoir une unique solution bien définie, qui ne varie pas trop par rapport aux données.

# Deux points de vue sur le problème

## Optimiseur

- Le problème peut être modélisé comme un programme quadratique convexe, et résolu de manière efficace;
- Potentiellement plusieurs solutions.

## Applicatif

- Le modèle doit s'appliquer à tous les documents de la distribution  $\Rightarrow$  généralisation;
- Mieux d'avoir une unique solution bien définie, qui ne varie pas trop par rapport aux données.

## Après discussion (data scientist?)

- Prise en compte de certaines problématiques dans la formulation du problème  $\Rightarrow$  Solution unique avec meilleure généralisation.
- Plus de connaissances sur le problème  $\Rightarrow$  Meilleure optimisation.

- Possible de définir des problèmes d'optimisation basés sur des données et de les résoudre efficacement;
- Potentiellement décorrélé du but originel : trouver un modèle sur la distribution des données.

- Possible de définir des problèmes d'optimisation basés sur des données et de les résoudre efficacement;
- Potentiellement décorrélé du but originel : trouver un modèle sur la distribution des données.

## Autres problématiques

- Quantité massive d'attributs (*tous les mots du dictionnaire*) ?
- Quantité massive de données (*articles Wikipedia*) ?
- Classification impossible par modèles linéaires ?

- Possible de définir des problèmes d'optimisation basés sur des données et de les résoudre efficacement;
- Potentiellement décorrélé du but originel : trouver un modèle sur la distribution des données.

## Autres problématiques

- Quantité massive d'attributs (*tous les mots du dictionnaire*) ?  
Réduction de dimension, recherche de parcimonie.
- Quantité massive de données (*articles Wikipedia*) ?  
Algorithmes stochastiques.
- Classification impossible par modèles linéaires ?  
Optimisation non linéaire.



- Fournir une boîte à outils moderne en optimisation;
- En lien avec la pratique courante en IA et sciences des données.

## Procédure

- Présenter des problématiques et des algorithmes associés;
- Les fondements théoriques;
- Des applications.

- 1 Introduction
- 2 Concepts de base en optimisation
  - Notations et algèbre linéaire
  - Un problème d'optimisation
- 3 Optimisation convexe

- 1 Introduction
- 2 Concepts de base en optimisation
  - Notations et algèbre linéaire
  - Un problème d'optimisation
- 3 Optimisation convexe

## Cadre restreint

- Optimisation sur des variables réelles;
- En dimension finie;
- On utilisera la structure d'espace classique.

## Mes notations pour aujourd'hui

- Scalaires :  $a, b, c, \dots$
- Vecteurs :  $\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$
- Matrices :  $\mathbf{A}, \mathbf{B}, \mathbf{C}, \dots$
- Ensembles :  $\mathcal{A}, \mathcal{B}, \mathcal{C}, \dots$

- $\mathbb{R}^d$  : ensemble des vecteurs à  $d \geq 1$  coordonnées réelles;
- Pour tous  $\mathbf{x} \in \mathbb{R}^n$  et  $i \in \{1, \dots, d\}$ ,  $x_i \in \mathbb{R}$  est la  $i$ -ème coordonnée de  $\mathbf{x}$  :  $\mathbf{x} = [x_i]_{1 \leq i \leq d}$ ;
- On représente  $\mathbf{x} \in \mathbb{R}^d$  en colonnes :  $\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$ ;
- On utilise des vecteurs lignes comme “transposés” de vecteurs colonnes :  $\mathbf{x}^T := [x_1 \cdots x_d]$ ;

## Opérations vectorielles

- *Addition dans  $\mathbb{R}^d$  :  $\mathbf{x} + \mathbf{z} := [x_i + z_i]_{1 \leq i \leq n}$ ;*
- *Multiplication d'un vecteur de  $\mathbb{R}^n$  par un réel :  $\lambda \mathbf{x} := [\lambda x_i]_{1 \leq i \leq n}$ .*

## Norme euclidienne sur $\mathbb{R}^n$

La norme euclidienne (ou norme  $\ell_2$ ) d'un vecteur  $\mathbf{x} \in \mathbb{R}^n$  est donnée par

$$\|\mathbf{x}\| := \sqrt{\sum_{i=1}^n x_i^2}.$$

## Produit scalaire sur $\mathbb{R}^n$

Pour tous  $\mathbf{x}, \mathbf{z} \in \mathbb{R}^n$ , défini par

$$\mathbf{x}^T \mathbf{z} := \sum_{i=1}^n x_i z_i.$$

On a ainsi  $\mathbf{x}^T \mathbf{z} = \mathbf{z}^T \mathbf{x}$  et  $\mathbf{x}^T \mathbf{x} = \|\mathbf{x}\|^2$ .

## Matrices

- $\mathbb{R}^{n \times m}$  : matrices à  $n$  lignes et  $m$  colonnes;
- $\mathbb{R}^{n \times 1} \simeq \mathbb{R}^n$ .

## Matrice transposée

Soit  $\mathbf{A} = [\mathbf{A}_{ij}] \in \mathbb{R}^{n \times m}$ . La *matrice transposée* de  $\mathbf{A}$ , notée  $\mathbf{A}^T$ , est la matrice à  $d$  lignes et  $n$  colonnes telle que

$$\forall i = 1, \dots, n, \forall j = 1, \dots, m, \quad [\mathbf{A}^T]_{ij} = \mathbf{A}_{ji}.$$

## Matrices

- $\mathbb{R}^{n \times m}$  : matrices à  $n$  lignes et  $m$  colonnes;
- $\mathbb{R}^{n \times 1} \simeq \mathbb{R}^n$ .

## Matrice transposée

Soit  $\mathbf{A} = [\mathbf{A}_{ij}] \in \mathbb{R}^{n \times m}$ . La *matrice transposée* de  $\mathbf{A}$ , notée  $\mathbf{A}^T$ , est la matrice à  $d$  lignes et  $n$  colonnes telle que

$$\forall i = 1, \dots, n, \forall j = 1, \dots, m, \quad [\mathbf{A}^T]_{ij} = \mathbf{A}_{ji}.$$

## Matrices carrées

- $\mathbf{A}^T \in \mathbb{R}^{n \times n}$ ;
- $\mathbf{A}$  est dite *symétrique* si  $\mathbf{A} = \mathbf{A}^T$ .



## Inversion et singularité

- Une matrice  $\mathbf{A} \in \mathbb{R}^{n \times n}$  est dite *inversible* s'il existe  $\mathbf{B} \in \mathbb{R}^{n \times n}$ , telle que  $\mathbf{BA} = \mathbf{AB} = \mathbf{I}_n$ , avec  $\mathbf{I}_n$  matrice identité de  $\mathbb{R}^{n \times n}$ .

Dans ce cas,  $\mathbf{B}$  est l'unique matrice vérifiant cette propriété : on l'appelle *l'inverse de la matrice  $\mathbf{A}$*  et on la note  $\mathbf{A}^{-1}$ .

## Inversion et singularité

- Une matrice  $\mathbf{A} \in \mathbb{R}^{n \times n}$  est dite *invertible* s'il existe  $\mathbf{B} \in \mathbb{R}^{n \times n}$ , telle que  $\mathbf{BA} = \mathbf{AB} = \mathbf{I}_n$ , avec  $\mathbf{I}_n$  matrice identité de  $\mathbb{R}^{n \times n}$ .

Dans ce cas,  $\mathbf{B}$  est l'unique matrice vérifiant cette propriété : on l'appelle l'*inverse de la matrice*  $\mathbf{A}$  et on la note  $\mathbf{A}^{-1}$ .

- Une matrice  $\mathbf{A} \in \mathbb{R}^{m \times n}$  est *singulière* s'il existe  $\mathbf{x} \in \mathbb{R}^n$  non nul tel que  $\mathbf{Ax} = \mathbf{0}$ .

Une matrice non singulière est dite *de rang plein*  $\min\{m, n\}$ .

## Inversion et singularité

- Une matrice  $\mathbf{A} \in \mathbb{R}^{n \times n}$  est dite *inversible* s'il existe  $\mathbf{B} \in \mathbb{R}^{n \times n}$ , telle que  $\mathbf{BA} = \mathbf{AB} = \mathbf{I}_n$ , avec  $\mathbf{I}_n$  matrice identité de  $\mathbb{R}^{n \times n}$ .

Dans ce cas,  $\mathbf{B}$  est l'unique matrice vérifiant cette propriété : on l'appelle l'*inverse de la matrice*  $\mathbf{A}$  et on la note  $\mathbf{A}^{-1}$ .

- Une matrice  $\mathbf{A} \in \mathbb{R}^{m \times n}$  est *singulière* s'il existe  $\mathbf{x} \in \mathbb{R}^n$  non nul tel que  $\mathbf{Ax} = \mathbf{0}$ .

Une matrice non singulière est dite *de rang plein*  $\min\{m, n\}$ .

## Caractère (semi)-défini positif

Une matrice symétrique  $\mathbf{A} \in \mathbb{R}^{n \times n}$  est dite *semi-définie positive* si

$$\forall \mathbf{v} \in \mathbb{R}^n, \quad \mathbf{v}^T \mathbf{A} \mathbf{v} \geq 0.$$

Elle est *définie positive* lorsque  $\mathbf{v}^T \mathbf{A} \mathbf{v} > 0$  pour tout vecteur  $\mathbf{v}$  non nul.

- 1 Introduction
- 2 Concepts de base en optimisation
  - Notations et algèbre linéaire
  - Un problème d'optimisation
- 3 Optimisation convexe

- Recherche opérationnelle;
- Prise de décision;
- Sciences de la décision;
- Programmation mathématique;
- Optimisation mathématique.

⇒ Tous ces concepts peuvent correspondre à de l'optimisation.

- Recherche opérationnelle;
- Prise de décision;
- Sciences de la décision;
- Programmation mathématique;
- Optimisation mathématique.

⇒ Tous ces concepts peuvent correspondre à de l'optimisation.

## Ma définition

*Le but de l'optimisation est de prendre la meilleure décision parmi un ensemble de possibilités.*

## Langages typiques des optimiseurs

- C/C++/Fortran (calcul à hautes performances)
- Matlab, Python, Julia (interprétés).

## Langages typiques des optimiseurs

- C/C++/Fortran (calcul à hautes performances)
- Matlab, Python, Julia (interprétés).

## Langages de modélisation

- GAMS, AMPL, CVX, Pyomo sont génériques;
- MATPOWER, PyTorch sont spécifiques à un domaine;
- La plupart peuvent être interfacés avec les langages ci-dessus.



Un **problème** de minimisation sur  $d$  paramètres réels s'écrit sous la forme :

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} f(\mathbf{x}) \quad \text{s. c. } \mathbf{x} \in \mathcal{X}$$

Un **problème** de minimisation sur  $d$  paramètres réels s'écrit sous la forme :

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} f(\mathbf{x}) \quad \text{s. c. } \mathbf{x} \in \mathcal{X}$$

- $\mathbf{x}$  représenté les **variables de décision**, supposées continues;
- $d$  est la dimension du problème (on prendra toujours  $d \geq 1$ );
- $f(\cdot)$  est la fonction **objectif/de coût/de perte**;
- $\mathcal{X}$  est l'ensemble réalisable/admissible regroupant les contraintes sur les variables de décision.

Un **problème** de minimisation sur  $d$  paramètres réels s'écrit sous la forme :

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} f(\mathbf{x}) \quad \text{s. c. } \mathbf{x} \in \mathcal{X}$$

- $\mathbf{x}$  représenté les **variables de décision**, supposées continues;
- $d$  est la dimension du problème (on prendra toujours  $d \geq 1$ );
- $f(\cdot)$  est la fonction **objectif/de coût/de perte**;
- $\mathcal{X}$  est l'ensemble réalisable/admissible regroupant les contraintes sur les variables de décision.

*Maximiser  $f$  revient à minimiser  $-f$ .*

# Example : Optimisation de charpente (Stolpe, 2017)

(a) Design domain with dimensions  $2 \times 1$ , boundary conditions, and external load.



(b) Optimal design for the wheel problem for a ground structure with  $61 \times 31$  nodes and 1,786,995 potential bars.

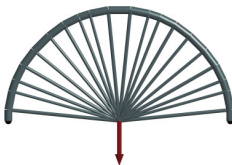
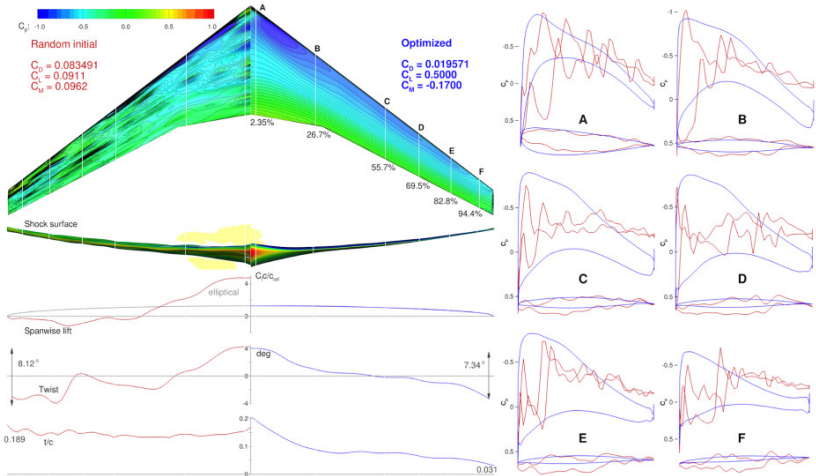


Figure 2.4. Optimal design of (half) a wheel.

Table 2.2. Numerical results obtained by the primal simplex method in IBM ILOG CPLEX for the single-load truss topology design problems listed in Table 2.1.

Problem	Number of nodes	Itn.	CPU (s)	Objective
Cantilever	$41 \times 11$	71,823	10.6	22.5593
Cantilever	$81 \times 21$	719,298	496.6	22.4726
Michell	$31 \times 21$	199,292	48.9	9.0244
Michell	$61 \times 41$	1,917,377	2335.6	9.0066
Wheel	$21 \times 11$	38,906	3.9	3.1708
Wheel	$41 \times 21$	700,623	230.2	3.1565
Wheel	$61 \times 31$	3,097,263	2755.5	3.1499

# Example : Optimisation multidisciplinaire (Martins, 2017)



**Figure 19.1.** The optimization starts from a random geometry (left/red) and converges to an optimal wing (right/blue). Originally appeared in [1245] and [1053].

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} f(\mathbf{x}) \quad \text{s. c. } \mathbf{x} \in \mathcal{X}$$

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} f(\mathbf{x}) \quad \text{s. c. } \mathbf{x} \in \mathcal{X}$$

- $\mathbf{x} \in \mathbb{R}^d$  est dit **admissible** ou **réalisable** si  $\mathbf{x} \in \mathcal{X}$ .
- $\mathbf{x}^* \in \mathbb{R}^d$  est une **solution du problème** si

$$\mathbf{x}^* \in \mathcal{X} \quad \text{et} \quad f(\mathbf{x}) \geq f(\mathbf{x}^*) \quad \forall \mathbf{x} \in \mathcal{X}.$$

L'ensemble des solutions du problème est noté

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\operatorname{argmin}} \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}.$$

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} f(\mathbf{x}) \quad \text{s. c. } \mathbf{x} \in \mathcal{X}$$

- $\mathbf{x} \in \mathbb{R}^d$  est dit **admissible** ou **réalisable** si  $\mathbf{x} \in \mathcal{X}$ .
- $\mathbf{x}^* \in \mathbb{R}^d$  est une **solution du problème** si

$$\mathbf{x}^* \in \mathcal{X} \quad \text{et} \quad f(\mathbf{x}) \geq f(\mathbf{x}^*) \quad \forall \mathbf{x} \in \mathcal{X}.$$

L'ensemble des solutions du problème est noté

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\operatorname{argmin}} \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}.$$

- La **valeur optimale du problème** est notée

$$\min_{\mathbf{x} \in \mathbb{R}^d} \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\}.$$



$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} f(\mathbf{x}) \quad \text{s. c. } \mathbf{x} \in \mathcal{X}$$

- Le problème est dit **irréalisable** si l'ensemble des contraintes est vide :  $\mathcal{X} = \emptyset$ . Dans ce cas, on a par convention

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\operatorname{argmin}} \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\} = \emptyset \quad \text{et} \quad \min_{\mathbf{x} \in \mathbb{R}^d} \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\} = +\infty.$$

Si  $\mathcal{X}$  n'est pas vide, le problème est dit réalisable.

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} f(\mathbf{x}) \quad \text{s. c. } \mathbf{x} \in \mathcal{X}$$

- Le problème est dit **irréalisable** si l'ensemble des contraintes est vide :  $\mathcal{X} = \emptyset$ . Dans ce cas, on a par convention

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\operatorname{argmin}} \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\} = \emptyset \quad \text{et} \quad \min_{\mathbf{x} \in \mathbb{R}^d} \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\} = +\infty.$$

Si  $\mathcal{X}$  n'est pas vide, le problème est dit réalisable.

- Le problème est dit **non borné** lorsque  $\mathcal{X}$  est non vide mais que  $f$  n'est pas minorée sur  $\mathcal{X}$ . Dans ce cas, on note

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\operatorname{argmin}} \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\} = \emptyset \quad \text{et} \quad \min_{\mathbf{x} \in \mathbb{R}^d} \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{X}\} = -\infty.$$

Pour un problème de minimisation (resp. de maximisation), on parlera de problème non minoré (resp. non majoré).

## 1 Introduction

## 2 Concepts de base en optimisation

## 3 Optimisation convexe

- Classes principales de problèmes
- Résolution de programmes convexes
- Application : Approximation et programmation convexe

*Pour une classe de problèmes donnés, jusqu'à quelles dimensions suis-je prêt à vous parier qu'un solveur du marché résout votre problème ?*

*Pour une classe de problèmes donnés, jusqu'à quelles dimensions suis-je prêt à vous parier qu'un solveur du marché résout votre problème ?*

Classe de problème	Nb. variables
Programme linéaire	$5 \times 10^7$
Programme quadratique convexe	$5 \times 10^5$
Programme conique	$5 \times 10^4$
Programme quadratique non convexe	300
Problème non convexe	100.

*Pour une classe de problèmes donnés, jusqu'à quelles dimensions suis-je prêt à vous parier qu'un solveur du marché résout votre problème ?*

Classe de problème	Nb. variables
Programme linéaire	$5 \times 10^7$
Programme quadratique convexe	$5 \times 10^5$
Programme conique	$5 \times 10^4$
Programme quadratique non convexe	300
Problème non convexe	100.

La **convexité** est une notion essentielle en optimisation !

**Idée principale :** Les modèles les plus utilisés sont ceux dont on sait calculer des solutions **de manière efficace**.

**Idée principale :** Les modèles les plus utilisés sont ceux dont on sait calculer des solutions **de manière efficace**.

## Cas fondamental : Programmation convexe

- Minimiser une fonction  $f$  convexe :

$$\forall (\mathbf{x}, \mathbf{y}) \in (\mathbb{R}^d)^2, \forall \alpha \in [0, 1], \quad f(\alpha \mathbf{x} + (1 - \alpha) \mathbf{y}) \leq \alpha f(\mathbf{x}) + (1 - \alpha) f(\mathbf{y})$$

- Sous contraintes convexes :

$$\forall (\mathbf{x}, \mathbf{y}) \in \mathcal{X}^2, \forall \alpha \in [0, 1], \quad \alpha \mathbf{x} + (1 - \alpha) \mathbf{y} \in \mathcal{X}.$$

**En général, possible de calculer efficacement des solutions, potentiellement pour des millions de variables !**



## 1 Introduction

## 2 Concepts de base en optimisation

## 3 Optimisation convexe

- Classes principales de problèmes
- Résolution de programmes convexes
- Application : Approximation et programmation convexe

## Forme de base

$$\left\{ \begin{array}{ll} \text{minimiser}_{\mathbf{x} \in \mathbb{R}^d} & f(\mathbf{x}) \\ \text{s. c.} & g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \\ & h_i(\mathbf{x}) = 0, \quad i = 1, \dots, \ell, \end{array} \right.$$

avec  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  convexe,  $g_i : \mathbb{R}^d \rightarrow \mathbb{R}$  convexe pour tout  $i$  et  $h_i : \mathbb{R}^d \rightarrow \mathbb{R}$  affine pour tout  $i$ .

$$\begin{array}{ll}\text{minimiser}_{\mathbf{x} \in \mathbb{R}^d} & \mathbf{c}^T \mathbf{x} \\ \text{s. c.} & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq 0,\end{array}$$

avec  $\mathbf{c} \in \mathbb{R}^d$ ,  $\mathbf{A} \in \mathbb{R}^{m \times d}$ ,  $\mathbf{b} \in \mathbb{R}^m$ .

- Problème canonique en optimisation;
- Possible de résoudre des problèmes avec des millions/milliards de variables.

## Exemple : Programmes quadratiques (QP)

$$\begin{array}{ll} \text{minimiser}_{\mathbf{x} \in \mathbb{R}^d} & \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{c}^T \mathbf{x} \\ \text{s. c.} & \mathbf{A} \mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0, \end{array}$$

avec  $\mathbf{H} \in \mathbb{R}^{d \times d}$ ,  $\mathbf{c} \in \mathbb{R}^d$ ,  $\mathbf{A} \in \mathbb{R}^{m \times d}$ ,  $\mathbf{b} \in \mathbb{R}^m$ .

- Extension du cas sans contraintes avec contraintes linéaires;
- Peut avoir une solution même si  $\mathbf{H} \succeq 0$ !

# Exemples : Programmes coniques d'ordre deux (SOCP)

$$\begin{array}{ll} \text{minimiser}_{\mathbf{x} \in \mathbb{R}^d} & \mathbf{c}^T \mathbf{x} \\ \text{s. c.} & \|\mathbf{A}_i \mathbf{x} + \mathbf{b}_i\| \leq \mathbf{c}_i^T \mathbf{x} + d_i, \quad i = 1, \dots, n. \end{array}$$

avec pour tout  $i = 1, \dots, m$ ,  $\mathbf{A}_i \in \mathbb{R}^{n_i \times d}$ ,  $\mathbf{b}_i \in \mathbb{R}^{n_i}$ ,  $\mathbf{c}_i \in \mathbb{R}^d$  et  $d_i \in \mathbb{R}$ .

- Généralisent LP et QP;
- Peuvent être résolus efficacement.

# Exemples : Programmes semi-définis positifs (SDP)

$$\begin{array}{ll}\text{minimiser}_{\mathbf{X} \in \mathbb{R}^{d \times d}} & \text{trace}(\mathbf{C}^T \mathbf{X}) \\ \text{s. c.} & \text{trace}(\mathbf{A}_i^T \mathbf{X}) = b_i, \quad i = 1, \dots, n \\ & \mathbf{X} = \mathbf{X}^T \succeq 0,\end{array}$$

avec  $\mathbf{C}, \mathbf{A}_1, \dots, \mathbf{A}_n \in \mathbb{R}^{d \times d}$  et  $\mathbf{b} = [b_i] \in \mathbb{R}^n$ .<sup>a</sup>

---

$$^a \text{trace}(\mathbf{UV}) = \sum_{i=1}^d \sum_{j=1}^d U_{ij} V_{ij}.$$

- Généralisent LP et QP;
- Possible de les résoudre en temps polynomial, mais les calculs algébriques peuvent être coûteux;
- Problèmes souvent de grandes tailles (relaxations de problèmes continus ou combinatoires).

# Exemples : Programmation conique (CP)

$$\begin{array}{ll}\text{minimiser}_{\mathbf{X} \in \mathbb{R}^{d \times d}} & \text{trace}(\mathbf{C}^T \mathbf{X}) \\ \text{s. c.} & \text{trace}(\mathbf{A}_i^T \mathbf{X}) = b_i, \quad i = 1, \dots, n \\ & \mathbf{X} \in \mathcal{K},\end{array}$$

où  $\mathbf{C}, \mathbf{A}_1, \dots, \mathbf{A}_n \in \mathbb{R}^{d \times d}$ ,  $\mathbf{b} = [b_i] \in \mathbb{R}^n$ .

$$\begin{array}{ll}\text{minimiser}_{\mathbf{X} \in \mathbb{R}^{d \times d}} & \text{trace}(\mathbf{C}^T \mathbf{X}) \\ \text{s. c.} & \text{trace}(\mathbf{A}_i^T \mathbf{X}) = b_i, \quad i = 1, \dots, n \\ & \mathbf{X} \in \mathcal{K},\end{array}$$

où  $\mathbf{C}, \mathbf{A}_1, \dots, \mathbf{A}_n \in \mathbb{R}^{d \times d}$ ,  $\mathbf{b} = [b_i] \in \mathbb{R}^n$ .

L'ensemble  $\mathcal{K}$  est un cône pointé de  $\mathbb{R}^{d \times d}$  :

- $\lambda \mathbf{X} \in \mathcal{K}$  pour tous  $\mathbf{X} \in \mathcal{K}$ ,  $\lambda > 0$ ;
- $\mathcal{K} \cap (-\mathcal{K}) = \{0\}$ .



$$\begin{array}{ll}\text{minimiser}_{\mathbf{X} \in \mathbb{R}^{d \times d}} & \text{trace}(\mathbf{C}^T \mathbf{X}) \\ \text{s. c.} & \text{trace}(\mathbf{A}_i^T \mathbf{X}) = b_i, \quad i = 1, \dots, n \\ & \mathbf{X} \in \mathcal{K},\end{array}$$

où  $\mathbf{C}, \mathbf{A}_1, \dots, \mathbf{A}_n \in \mathbb{R}^{d \times d}$ ,  $\mathbf{b} = [b_i] \in \mathbb{R}^n$ .

L'ensemble  $\mathcal{K}$  est un cône pointé de  $\mathbb{R}^{d \times d}$  :

- $\lambda \mathbf{X} \in \mathcal{K}$  pour tous  $\mathbf{X} \in \mathcal{K}$ ,  $\lambda > 0$ ;
- $\mathcal{K} \cap (-\mathcal{K}) = \{0\}$ .

- Généralise tout ce qui précède !
- Formulation utilisée par des bibliothèques spécialisées (cvxpy) et des solveurs commerciaux (MOSEK).

- 1 Introduction
- 2 Concepts de base en optimisation
- 3 **Optimisation convexe**
  - Classes principales de problèmes
  - **Résolution de programmes convexes**
  - Application : Approximation et programmation convexe

## Pour les classes majeures

- LP, QP, SDP, SOCP;
- D'autres classes non traitées ici.

## Dans la suite

- On illustre les notions dans le cadre des programmes linéaires;
- On donne l'idée derrière les méthodes dites de points intérieurs.

## Problème de base (dit primal)

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser } \mathbf{c}^T \mathbf{x}} \quad \text{s. c.} \quad \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0,$$

$$\mathbf{A} \in \mathbb{R}^{m \times d}, \mathbf{b} \in \mathbb{R}^m.$$

## Problème de base (dit primal)

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \mathbf{c}^T \mathbf{x} \quad \text{s. c.} \quad \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0,$$

$$\mathbf{A} \in \mathbb{R}^{m \times d}, \mathbf{b} \in \mathbb{R}^m.$$

## Problème dual (aussi LP !)

$$\underset{\mathbf{y} \in \mathbb{R}^m, \mathbf{s} \in \mathbb{R}^d}{\text{maximiser}} \mathbf{b}^T \mathbf{y} \quad \text{s. c.} \quad \mathbf{A}^T \mathbf{y} + \mathbf{s} = \mathbf{c}, \mathbf{s} \geq 0.$$

## Problème de base (dit primal)

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \mathbf{c}^T \mathbf{x} \quad \text{s. c.} \quad \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0,$$

$$\mathbf{A} \in \mathbb{R}^{m \times d}, \mathbf{b} \in \mathbb{R}^m.$$

## Problème dual (aussi LP !)

$$\underset{\mathbf{y} \in \mathbb{R}^m, \mathbf{s} \in \mathbb{R}^d}{\text{maximiser}} \mathbf{b}^T \mathbf{y} \quad \text{s. c.} \quad \mathbf{A}^T \mathbf{y} + \mathbf{s} = \mathbf{c}, \mathbf{s} \geq 0.$$

## Conditions de KKT

Si  $\mathbf{x}^* \in \mathbb{R}^d$  est une solution, il existe  $\mathbf{y}^* \in \mathbb{R}^m$  et  $\mathbf{s}^* \in \mathbb{R}^d$  tels que

$$\begin{aligned} \mathbf{Ax}^* &= \mathbf{b}, & \mathbf{x}^* &\geq 0, \\ \mathbf{A}^T \mathbf{y}^* + \mathbf{s}^* &= \mathbf{c}, & \mathbf{s}^* &\geq 0, \\ x_i s_i &= 0 \quad \forall i = 1, \dots, d. \end{aligned}$$

## Théorème

Pour tout programme linéaire,

- i) Soit ce problème et son dual ont chacun une solution, et l'ensemble vérifie les conditions de KKT;
- ii) Soit un des deux problèmes n'a pas de solution, et l'ensemble réalisable de l'autre est vide (problème irréalisable);
- iii) Soit les deux ensembles réalisables sont vides.

## Théorème

Pour tout programme linéaire,

- i) Soit ce problème et son dual ont chacun une solution, et l'ensemble vérifie les conditions de KKT;
- ii) Soit un des deux problèmes n'a pas de solution, et l'ensemble réalisable de l'autre est vide (problème irréalisable);
- iii) Soit les deux ensembles réalisables sont vides.

## Conséquences

- Trouver une solution revient à trouver un triplet vérifiant les contraintes de chaque problème !
- C'est ce que font les approches de points intérieurs, de manière primale-duale ou duale.



## Principe

- Partir d'un triplet  $(\mathbf{x}, \mathbf{y}, \mathbf{s})$  réalisable pour le problème et son dual avec  $\mathbf{x} > 0$  et  $\mathbf{s} > 0$ .
- Appliquer la méthode de Newton au système de KKT

$$\begin{cases} \mathbf{A}\mathbf{x}^* &= \mathbf{b} \\ \mathbf{A}^T\mathbf{y}^* + \mathbf{s}^* &= \mathbf{c} \\ x_i s_i &= 0 \quad \forall i = 1, \dots, d. \end{cases}$$

- Obtenir un nouveau point “intérieur” avec une valeur  $\mathbf{x}^T \mathbf{s}$  réduite (via la méthode de Newton).

## Principe

- Partir d'un triplet  $(\mathbf{x}, \mathbf{y}, \mathbf{s})$  réalisable pour le problème et son dual avec  $\mathbf{x} > 0$  et  $\mathbf{s} > 0$ .
- Appliquer la méthode de Newton au système de KKT

$$\begin{cases} \mathbf{A}\mathbf{x}^* &= \mathbf{b} \\ \mathbf{A}^T\mathbf{y}^* + \mathbf{s}^* &= \mathbf{c} \\ x_i s_i &= 0 \quad \forall i = 1, \dots, d. \end{cases}$$

- Obtenir un nouveau point “intérieur” avec une valeur  $\mathbf{x}^T \mathbf{s}$  réduite (via la méthode de Newton).

## Ça marche !

- Vitesses de convergence en théorie;
- Très bonnes performances en pratique;
- Implémentations très sophistiquées.

## Programmation conique

$$\begin{array}{ll}\text{minimiser}_{\mathbf{X} \in \mathbb{R}^{d \times d}} & \text{trace}(\mathbf{C}^T \mathbf{X}) \\ \text{s. c.} & \text{trace}(\mathbf{A}_i^T \mathbf{X}) = b_i, \quad i = 1, \dots, n \\ & \mathbf{X} \in \mathcal{K},\end{array}$$

où  $\mathbf{C}, \mathbf{A}_1, \dots, \mathbf{A}_n \in \mathbb{R}^{d \times d}$ ,  $\mathbf{b} = [b_i] \in \mathbb{R}^n$ , et  $\mathcal{K}$  est un cône pointé.

## Points intérieurs pour CP

- Partir des conditions de KKT;
- Rester dans l'intérieur du cône  $\mathcal{K}$ ;
- Se rapprocher progressivement de la frontière;
- Implémentations efficaces (SeduMi, SDP3).

## 1 Introduction

## 2 Concepts de base en optimisation

## 3 Optimisation convexe

- Classes principales de problèmes
- Résolution de programmes convexes
- Application : Approximation et programmation convexe

## Données

- Jeu de données à  $n$  éléments (individus, échantillons, etc);
- Chaque élément  $i$  est caractérisé par un vecteur  $\mathbf{a}_i \in \mathbb{R}^d$  d'attributs ainsi qu'un label  $b_i \in \mathbb{R}$ .

$$\Rightarrow \text{Matrice } \mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_n^T \end{bmatrix} \in \mathbb{R}^{n \times d} \text{ et vecteur } \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

## Données

- Jeu de données à  $n$  éléments (individus, échantillons, etc);
- Chaque élément  $i$  est caractérisé par un vecteur  $\mathbf{a}_i \in \mathbb{R}^d$  d'attributs ainsi qu'un label  $b_i \in \mathbb{R}$ .

$$\Rightarrow \text{Matrice } \mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_n^T \end{bmatrix} \in \mathbb{R}^{n \times d} \text{ et vecteur } \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

## But

On cherche un modèle linéaire  $h : \mathbf{a} \mapsto \mathbf{a}^T \mathbf{x}$  qui prédise correctement les  $b_i$  d'après les  $\mathbf{a}_i$ .

- Les modèles linéaires (dans le bon espace) sont souvent une bonne approximation;
- Utilisation d'algèbre linéaire (intérêt à la fois théorique et numérique).

## Prédiction idéale

- Un vecteur  $\mathbf{x} \in \mathbb{R}^d$  tel que  $\mathbf{a}_i^T \mathbf{x} = b_i$  pour tout  $i$ ;
- Ces  $n$  equations peuvent s'écrire sous la forme d'un système linéaire:  
 **$\mathbf{Ax} = \mathbf{b}$ .**

## Prédiction idéale

- Un vecteur  $\mathbf{x} \in \mathbb{R}^d$  tel que  $\mathbf{a}_i^T \mathbf{x} = b_i$  pour tout  $i$ ;
- Ces  $n$  equations peuvent s'écrire sous la forme d'un système linéaire:  
 **$\mathbf{Ax} = \mathbf{b}$ .**

## Résoudre des systèmes d'équations linéaires

- Une histoire d'algèbre linéaire;
- L'existence de solutions ne dépend que de  **$\mathbf{A}$**  et  **$\mathbf{b}$** .



## Un mauvais jeu de données

- $\mathbf{a}_1 = \mathbf{a}_2 = \dots = \mathbf{a}_n = 1$  ( $d = 1$ );
- $b_1, \dots, b_n$  sont distincts (typique de mesures bruitées).

## Un mauvais jeu de données

- $\mathbf{a}_1 = \mathbf{a}_2 = \dots = \mathbf{a}_n = 1$  ( $d = 1$ );
- $b_1, \dots, b_n$  sont distincts (typique de mesures bruitées).

## Expliquer les données par un modèle linéaire

- On cherche  $\mathbf{x} = x \in \mathbb{R}$  tel que  $\mathbf{a}_i^T \mathbf{x} = a_i x = b_i \forall i$ ;
- Système linéaire :

$$\begin{cases} x = b_1 \\ x = b_2 \\ \vdots \\ x = b_n \end{cases}$$

# Juste de l'algèbre linéaire ?

## Un mauvais jeu de données

- $\mathbf{a}_1 = \mathbf{a}_2 = \dots = \mathbf{a}_n = 1$  ( $d = 1$ );
- $b_1, \dots, b_n$  sont distincts (typique de mesures bruitées).

## Expliquer les données par un modèle linéaire

- On cherche  $\mathbf{x} = x \in \mathbb{R}$  tel que  $\mathbf{a}_i^T \mathbf{x} = a_i x = b_i \forall i$ ;
- Système linéaire :

$$\begin{cases} x = b_1 \\ x = b_2 \\ \vdots \\ x = b_n \end{cases}$$

- Ce système n'a pas de solution !
- En revanche, il existe une solution au problème de “coller aux données” (“data fitting”).

# Trois problèmes d'approximation

## Pb 1) Approximation de Chebyshev

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \|\mathbf{Ax} - \mathbf{b}\|_{\infty} := \max_{1 \leq i \leq n} |\mathbf{a}_i^T \mathbf{x} - b_i|.$$

## Pb 2) Approximation robuste/en norme $\ell_1$

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \|\mathbf{Ax} - \mathbf{b}\|_1 := \sum_{1 \leq i \leq n} |\mathbf{a}_i^T \mathbf{x} - b_i|.$$

## Pb 3) Moindres carrés linéaires

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \|\mathbf{Ax} - \mathbf{b}\|^2 := \sum_{1 \leq i \leq n} |\mathbf{a}_i^T \mathbf{x} - b_i|^2.$$

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \|\mathbf{Ax} - \mathbf{b}\|_\infty := \max_{1 \leq i \leq n} |\mathbf{a}_i^T \mathbf{x} - b_i|.$$

## Reformulation en programme linéaire

$$\begin{array}{ll} \underset{\substack{\mathbf{x} \in \mathbb{R}^d \\ t \in \mathbb{R}, \mathbf{s} \in \mathbb{R}^{2n}}}{\text{minimiser}} & t \\ \text{s. c.} & \\ & -t - \mathbf{a}_i^T \mathbf{x} + b_i + s_i = 0, \quad i = 1, \dots, n \\ & -t + \mathbf{a}_i^T \mathbf{x} - b_i + s_{n+i} = 0, \quad i = 1, \dots, n \\ & t \geq 0 \\ & s_j \geq 0 \quad i = 1, \dots, 2n. \end{array}$$

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \|\mathbf{Ax} - \mathbf{b}\|_{\infty} := \max_{1 \leq i \leq n} |\mathbf{a}_i^T \mathbf{x} - b_i|.$$

## Reformulation en programme linéaire

$$\begin{array}{ll} \underset{\substack{\mathbf{x} \in \mathbb{R}^d \\ t \in \mathbb{R}, \mathbf{s} \in \mathbb{R}^{2n}}}{\text{minimiser}} & t \\ \text{s. c.} & \\ & -t - \mathbf{a}_i^T \mathbf{x} + b_i + s_i = 0, \quad i = 1, \dots, n \\ & -t + \mathbf{a}_i^T \mathbf{x} - b_i + s_{n+i} = 0, \quad i = 1, \dots, n \\ & t \geq 0 \\ & s_j \geq 0 \quad i = 1, \dots, 2n. \end{array}$$

- Programme linéaire à  $d + 2n + 1$  variables;
- Résolution aisée via les points intérieurs;
- Donne la solution et la valeur optimale du problème d'origine !

## Pb 2) Approximation en norme $\ell_1$

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \|\mathbf{Ax} - \mathbf{b}\|_1 := \sum_{1 \leq i \leq n} |\mathbf{a}_i^T \mathbf{x} - b_i|.$$

## Pb 2) Approximation en norme $\ell_1$

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \|\mathbf{Ax} - \mathbf{b}\|_1 := \sum_{1 \leq i \leq n} |\mathbf{a}_i^T \mathbf{x} - b_i|.$$

### Reformulation en programme linéaire

$$\begin{array}{ll} \underset{\substack{\mathbf{x} \in \mathbb{R}^d \\ t^+, t^- \in \mathbb{R}^n}}{\text{minimiser}} & \sum_{i=1}^n (t_i^+ + t_i^-) \\ \text{s. c.} & \begin{array}{ll} \mathbf{a}_i^T \mathbf{x} - b_i - t_i^+ + t_i^- & = 0 \quad i = 1, \dots, n \\ t_i^+ & \geq 0 \quad i = 1, \dots, n \\ t_i^- & \geq 0 \quad i = 1, \dots, n \end{array} \end{array}$$



## Pb 2) Approximation en norme $\ell_1$

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \|\mathbf{Ax} - \mathbf{b}\|_1 := \sum_{1 \leq i \leq n} |\mathbf{a}_i^T \mathbf{x} - b_i|.$$

### Reformulation en programme linéaire

$$\begin{array}{ll} \underset{\substack{\mathbf{x} \in \mathbb{R}^d \\ t^+, t^- \in \mathbb{R}^n}}{\text{minimiser}} & \sum_{i=1}^n (t_i^+ + t_i^-) \\ \text{s. c.} & \begin{array}{ll} \mathbf{a}_i^T \mathbf{x} - b_i - t_i^+ + t_i^- & = 0 \quad i = 1, \dots, n \\ t_i^+ & \geq 0 \quad i = 1, \dots, n \\ t_i^- & \geq 0 \quad i = 1, \dots, n \end{array} \end{array}$$

- Programme linéaire à  $d + 2n$  variables;
- Résolution aisée via les points intérieurs;
- Donne la solution et la valeur optimale du problème d'origine !

### Formulation du problème

Étant donné  $\{(\mathbf{a}_i, b_i)\}_{1 \leq i \leq n}$  avec  $\mathbf{a}_i \in \mathbb{R}^d$ , on considère le problème d'optimisation :

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \frac{1}{2n} \|\mathbf{Ax} - \mathbf{b}\|^2 = \frac{1}{2}(\mathbf{Ax} - \mathbf{b})^T(\mathbf{Ax} - \mathbf{b}),$$

$$\text{avec } \mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_n^T \end{bmatrix} \in \mathbb{R}^{n \times d} \text{ et } \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \in \mathbb{R}^n.$$

### Formulation du problème

Étant donné  $\{(\mathbf{a}_i, b_i)\}_{1 \leq i \leq n}$  avec  $\mathbf{a}_i \in \mathbb{R}^d$ , on considère le problème d'optimisation :

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \frac{1}{2n} \|\mathbf{Ax} - \mathbf{b}\|^2 = \frac{1}{2}(\mathbf{Ax} - \mathbf{b})^T(\mathbf{Ax} - \mathbf{b}),$$

$$\text{avec } \mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_n^T \end{bmatrix} \in \mathbb{R}^{n \times d} \text{ et } \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \in \mathbb{R}^n.$$

### Propriété

- Programme quadratique **sans contraintes**;
- Peut être résolu sans points intérieurs.

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \frac{1}{2n} \|\mathbf{Ax} - \mathbf{b}\|_2^2.$$

## Outil : Décomposition en valeurs singulières

La matrice  $\mathbf{A} \in \mathbb{R}^{n \times d}$  peut s'écrire

$$\mathbf{A} = \mathbf{U} \left[ \begin{array}{ccc|c} \sigma_1 & 0 \cdots & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & \cdots 0 & \sigma_r & 0 \\ \hline 0 & \cdots & \cdots & 0 \end{array} \right] \mathbf{V}^T,$$

où  $\mathbf{U} \in \mathbb{R}^{n \times n}$  et  $\mathbf{V} \in \mathbb{R}^{d \times d}$  sont des matrices orthogonales, et  $\sigma_1 \geq \cdots \geq \sigma_r > 0$  sont les valeurs singulières (non nulles) de  $\mathbf{A}$ .

## Théorème (Inverse de Moore-Penrose)

Soit  $\mathbf{A} \in \mathbb{R}^{n \times d}$  et sa décomposition

$$\mathbf{A} = \mathbf{U} \left[ \begin{array}{ccc|c} \sigma_1 & 0 \dots & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & \dots 0 & \sigma_r & 0 \\ \hline 0 & \dots & \dots & 0 \end{array} \right] \mathbf{V}^T,$$

avec  $\sigma_1 \geq \dots \geq \sigma_r > 0$ . Alors, la **pseudo-inverse de  $\mathbf{A}$**  est définie par

$$\mathbf{A}^\dagger = \mathbf{V} \left[ \begin{array}{ccc|c} \frac{1}{\sigma_1} & 0 \dots & 0 & 0 \\ 0 & \ddots & 0 & \vdots \\ 0 & \dots 0 & \frac{1}{\sigma_r} & 0 \\ \hline 0 & \dots & \dots & 0 \end{array} \right] \mathbf{U}^T.$$

$$\mathbf{A} \in \mathbb{R}^{n \times d}, \quad \mathbf{b} \in \mathbb{R}^n.$$

## Théorème

Pour tout  $\mathbf{b} \in \mathbb{R}^n$ ,  $\mathbf{A}^\dagger \mathbf{b}$  est la solution du problème aux moindres carrés

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \frac{1}{2n} \|\mathbf{Ax} - \mathbf{b}\|^2.$$

**de norme minimale.** Pour tout  $\hat{\mathbf{x}} \in \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{2n} \|\mathbf{Ax} - \mathbf{b}\|^2$ , on a :

$$\mathbf{A} \in \mathbb{R}^{n \times d}, \quad \mathbf{b} \in \mathbb{R}^n.$$

## Théorème

Pour tout  $\mathbf{b} \in \mathbb{R}^n$ ,  $\mathbf{A}^\dagger \mathbf{b}$  est la solution du problème aux moindres carrés

$$\underset{\mathbf{x} \in \mathbb{R}^d}{\text{minimiser}} \frac{1}{2n} \|\mathbf{Ax} - \mathbf{b}\|^2.$$

**de norme minimale.** Pour tout  $\hat{\mathbf{x}} \in \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{2n} \|\mathbf{Ax} - \mathbf{b}\|^2$ , on a :

- $\|\mathbf{AA}^\dagger \mathbf{b} - \mathbf{b}\|^2 = \|\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\|^2$ ;
- $\|\mathbf{A}^\dagger \mathbf{b}\| \leq \|\hat{\mathbf{x}}\|$ .

$$\mathbf{Ax} = \mathbf{v}, \quad \mathbf{A} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

## Une solution

- Le problème minimiser  $\mathbf{x} \in \mathbb{R}^d$   $\frac{1}{2n} \|\mathbf{Ax} - \mathbf{b}\|_2^2$  possède une infinité de solutions;
- Parmi celles-ci  $\mathbf{x}^* = \mathbf{A}^\dagger \mathbf{b}$  est de norme minimale;
- Cette solution est la moyenne  $\mathbf{x}^* = \frac{1}{n} \sum_{i=1}^n b_i!$



## Qualité de la solution des moindres carrés

- Meilleure approximation possible en termes d'erreurs;
- Solution déterministe quand  $\{(\mathbf{x}_i, y_i)\}_i$  fixés.

## Qualité de la solution des moindres carrés

- Meilleure approximation possible en termes d'erreurs;
- Solution déterministe quand  $\{(\mathbf{x}_i, y_i)\}_i$  fixés.

## En présence de données bruitées

- Approche statistique : supposer que  $\mathbf{y} = \mathbf{X}\mathbf{w}^* + \epsilon$  avec  $\epsilon$  vecteur aléatoire de loi connue;
- Calculer l'estimateur du **maximum de vraisemblance** en résolvant un problème d'optimisation;
- Équivalent aux moindres carrés pour  $\epsilon$  suivant une loi gaussienne/normale;
- Équivalent à la régression  $\ell_1$  pour  $\epsilon$  suivant une loi de Laplace.

## Optimisation (en IA et au-delà)

- Outil de modélisation;
- Outil de résolution de problèmes;
- Outil numérique!

## Optimisation convexe

- Outil de prédilection;
- Large spectre de modélisation;
- Capacité de résolution à grande échelle.

## Les grands classiques

- Programmation linéaire;
- Moindres carrés linéaires.

## Ouvrages:

- T. Terlaky, M. F. Anjos, S. Ahmed (Eds.), *Advances and Trends in Optimization with Engineering Applications*. MOS-SIAM Series on Optimization, 2017.
- S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.
- S. Boyd and L. Vandenberghe, *Introduction to Applied Algebra: Vectors, Matrices, and Least Squares*. Cambridge University Press, 2018.

## Codes :

- `scipy/numpy` : bibliothèque de base en calcul scientifique pour Python.
- `cvxpy` : résolution de problèmes convexes en Python (existe aussi en MATLAB).

## Aujourd'hui

- Outils d'optimisation existants, problèmes que l'on peut résoudre;
- Focus : programmation convexe.

## Demain

- Matin : Optimisation non convexe.
- Après-midi : Optimisation stochastique.

## Aujourd'hui

- Outils d'optimisation existants, problèmes que l'on peut résoudre;
- Focus : programmation convexe.

## Demain

- Matin : Optimisation non convexe.
- Après-midi : Optimisation stochastique.

Merci beaucoup !