Optimisation-RO

Clément Royer

Certificat Chef de Projet IA - Université Paris Dauphine-PSL

10 octobre 2023

Intervenants

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`

Informations utiles

Ressources

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

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

Déroulé de la formation

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

Pour la séance de ce matin

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

Table des matières

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

Terminologie

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;
- ...

Terminologie

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).

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.

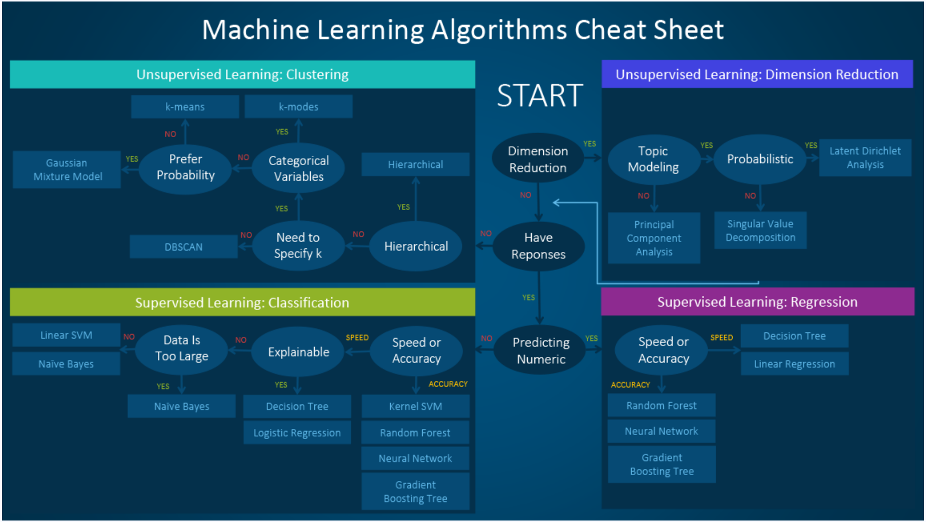
Ce que sera l'IA pour nous

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.



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

Changement de paradigme

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).

Changement de paradigme

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).

Les changements

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.

Les changements

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".

Table des matières

1 Introduction
- Un example : classification binaire

2 Concepts de base en optimisation

3 Optimisation convexe

Problème d'apprentissage statistique

Point de départ : Jeu de données $\{(\boldsymbol{a}_1, b_1), \dots, (\boldsymbol{a}_n, b_n)\}$.

- $\boldsymbol{a}_i$ vecteur d'attributs à d coordonnées;
- b_i label binaire égal à 1 ou -1.

Problème d'apprentissage statistique

Point de départ : Jeu de données $\{(\boldsymbol{a}_1, b_1), \dots, (\boldsymbol{a}_n, b_n)\}$.

- $\boldsymbol{a}_i$ vecteur d'attributs à d coordonnées;
- b_i label binaire égal à 1 ou -1.

Exemple : classification de documents

Soit un dictionnaire de d mots.

- $\boldsymbol{a}_i$ représente les mots contenus dans le document i:

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

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

Prédiction

- On cherche $h : \mathbb{R}^d \rightarrow \mathbb{R}$ telle que $h(\boldsymbol{a}_i) \approx b_i \ \forall i = 1, \dots, n$.
- On choisit h dans un ensemble $\mathcal{H}$ paramétré par un vecteur $\boldsymbol{x} \in \mathbb{R}^d$: $\mathcal{H} = \left\{ h \mid h = h(\cdot; \boldsymbol{x}), \ \boldsymbol{x} \in \mathbb{R}^d \right\}$;

Prédiction et classification

Prédiction

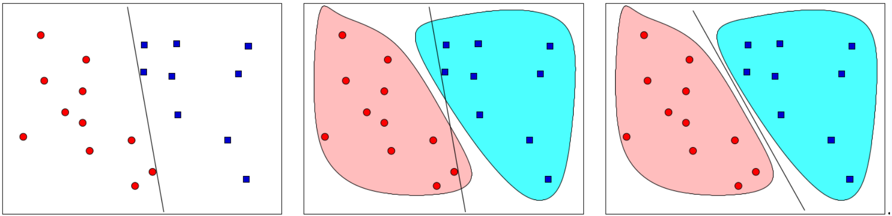
- On cherche $h : \mathbb{R}^d \rightarrow \mathbb{R}$ telle que $h(\boldsymbol{a}_i) \approx b_i \ \forall i = 1, \dots, n$.
- On choisit h dans un ensemble $\mathcal{H}$ paramétré par un vecteur $\boldsymbol{x} \in \mathbb{R}^d$: $\mathcal{H} = \left\{ h \mid h = h(\cdot; \boldsymbol{x}),\ \boldsymbol{x} \in \mathbb{R}^d \right\}$;

Modèle linéaire pour notre exemple

- On cherche un hyperplan de $\mathbb{R}^d$ qui sépare les vecteurs de label $b_i = +1$ de ceux de label $b_i = -1$;
- Cela correspond à trouver un modèle linéaire $h(\boldsymbol{a}) = \boldsymbol{a}^{\mathrm{T}} \boldsymbol{x}$, avec $\boldsymbol{x}$ tel que

$$\forall i = 1, \dots, n, \qquad \left\{ \begin{array}{ll} \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{x} \geq 1 & \text{si } b_i = +1 \\ \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{x} \leq -1 & \text{si } b_i = -1. \end{array} \right.$$

Différentes solutions



Source : S. J. Wright and B. Recht, Optimization for Data Analysis, 2022.

- Points : $\boldsymbol{a}_i$, rouges/bleus : $b_i = 1/b_i = -1$;
- 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.

Bilan de l'exemple

Problématiques d'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.

Bilan de l'exemple

Problématiques d'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.

Autres problématiques

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

Bilan de l'exemple

Problématiques d'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.

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.

Objectifs du cours

- 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.

Table des matières

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

Table des matières

1 Introduction

2 Concepts de base en optimisation
- Notations et algèbre linéaire
- Un problème d'optimisation

3 Optimisation convexe

Notations

Cadre restreint

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

Mes notations pour aujourd'hui

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

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

Opérations vectorielles

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

Algèbre linéaire (2)

Norme euclidienne sur $\mathbb{R}^d$

La norme euclidienne (ou norme ℓ_2) d'un vecteur $\boldsymbol{x} \in \mathbb{R}^d$ est donnée par

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

Produit scalaire sur $\mathbb{R}^d$

Pour tous $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^d$, défini par

$$\boldsymbol{x}^{\mathrm{T}} \boldsymbol{z} := \sum_{i=1}^{d} x_i \, z_i.$$

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

Algèbre linéaire (3)

Matrices

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

Matrice transposée

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

$$\forall i = 1, \dots, d,\ \forall j = 1, \dots, n, \qquad \left[\boldsymbol{A}^{\mathrm{T}}\right]_{ij} = \boldsymbol{A}_{ji}.$$

Matrices

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

Matrice transposée

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

$$\forall i = 1, \dots, d,\ \forall j = 1, \dots, n, \qquad \left[\boldsymbol{A}^{\mathrm{T}}\right]_{ij} = \boldsymbol{A}_{ji}.$$

Matrices carrées ($n = d$)

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

Inversion et singularité

- Une matrice $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ est dite *inversible* s'il existe $\boldsymbol{B} \in \mathbb{R}^{d \times d}$ telle que $\boldsymbol{BA} = \boldsymbol{AB} = \boldsymbol{I}_d$, avec $\boldsymbol{I}_d$ matrice identité de $\mathbb{R}^{d \times d}$.
  Dans ce cas, $\boldsymbol{B}$ est l'unique matrice vérifiant cette propriété : on l'appelle l'*inverse de la matrice* $\boldsymbol{A}$ et on la note $\boldsymbol{A}^{-1}$.

Algèbre linéaire (4)

Inversion et singularité

- Une matrice $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ est dite *inversible* s'il existe $\boldsymbol{B} \in \mathbb{R}^{d \times d}$ telle que $\boldsymbol{BA} = \boldsymbol{AB} = \boldsymbol{I}_d$, avec $\boldsymbol{I}_d$ matrice identité de $\mathbb{R}^{d \times d}$.
  Dans ce cas, $\boldsymbol{B}$ est l'unique matrice vérifiant cette propriété : on l'appelle l'*inverse de la matrice* $\boldsymbol{A}$ et on la note $\boldsymbol{A}^{-1}$.
- Une matrice $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ est *singulière* s'il existe $\boldsymbol{x} \in \mathbb{R}^d$ non nul tel que $\boldsymbol{Ax} = 0$.
  Une matrice non singulière est dite *de rang plein* $\min\{n, d\}$.

Algèbre linéaire (4)

Inversion et singularité

- Une matrice $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ est dite *inversible* s'il existe $\boldsymbol{B} \in \mathbb{R}^{d \times d}$ telle que $\boldsymbol{BA} = \boldsymbol{AB} = \boldsymbol{I}_d$, avec $\boldsymbol{I}_d$ matrice identité de $\mathbb{R}^{d \times d}$.
  Dans ce cas, $\boldsymbol{B}$ est l'unique matrice vérifiant cette propriété : on l'appelle l'*inverse de la matrice* $\boldsymbol{A}$ et on la note $\boldsymbol{A}^{-1}$.
- Une matrice $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ est *singulière* s'il existe $\boldsymbol{x} \in \mathbb{R}^d$ non nul tel que $\boldsymbol{Ax} = 0$.
  Une matrice non singulière est dite *de rang plein* $\min\{n, d\}$.

Caractère (semi)-défini positif

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

$$\forall \boldsymbol{v} \in \mathbb{R}^d, \quad \boldsymbol{v}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{v} \geq 0.$$

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

Table des matières

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

Optimisation ?

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

$\Rightarrow$ Tous ces concepts peuvent correspondre à de l'optimisation.

Optimisation ?

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

$\Rightarrow$ 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.

Du côté machine

Langages typiques des optimiseurs

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

Du côté machine

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.

Du côté mathématique

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

$$\operatorname*{minimiser}_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x}) \quad \text{s. c. } \boldsymbol{x} \in \mathcal{X}$$

Du côté mathématique

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

$$\operatorname*{minimiser}_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x}) \quad \text{s. c. } \boldsymbol{x} \in \mathcal{X}$$

- $\boldsymbol{x}$ représente 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.

Du côté mathématique

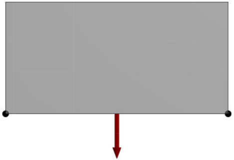
Un **problème** de minimisation sur d paramètres réels s'écrit sous la forme :

$$\operatorname*{minimiser}_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x}) \quad \text{s. c. } \boldsymbol{x} \in \mathcal{X}$$

- $\boldsymbol{x}$ représente 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 $-f$.

(a) *Design domain with dimensions* 2 × 1, *boundary conditions, and external load.*



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

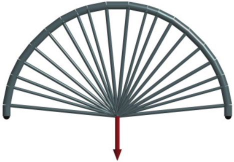


Figure 2.4. *Optimal design of (half) a wheel.*

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 × 11	71,823	10.6	22.5593
Cantilever	81 × 21	719,298	496.6	22.4726
Michell	31 × 21	199,292	48.9	9.0244
Michell	61 × 41	1,917,377	2335.6	9.0066
Wheel	21 × 11	38,906	3.9	3.1708
Wheel	41 × 21	700,623	230.2	3.1565
Wheel	61 × 31	3,097,263	2755.5	3.1499

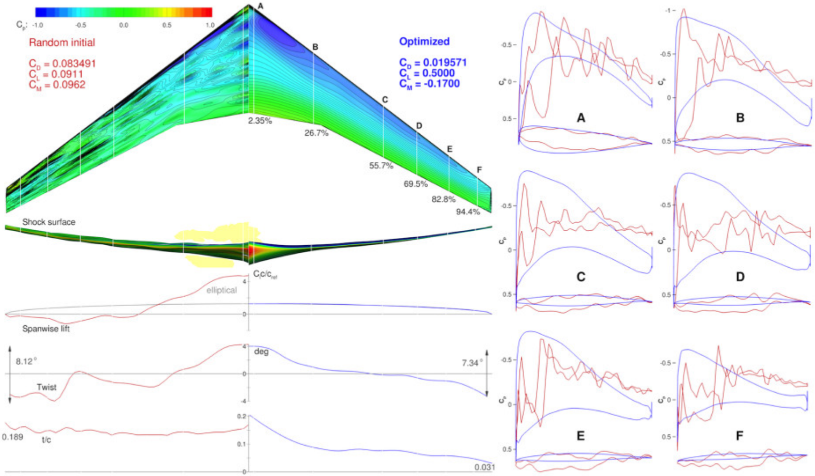


Figure 19.1. *The optimization starts from a random geometry (left/red) and converges to an optimal wing (right/blue). Originally appeared in* [1245] *and* [1053].

Premières définitions

$$\operatorname*{minimiser}_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) \quad \text{s. c.} \quad \boldsymbol{x} \in \mathcal{X}$$

Premières définitions

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

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

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

L'ensemble des solutions du problème est noté

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

Premières définitions

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

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

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

  L'ensemble des solutions du problème est noté

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{argmin}} \left\{ f(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{X} \right\}.$$

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

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

Premières définitions (2)

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

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

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

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

Premières définitions (2)

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

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

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

  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'a pas de minimum sur $\mathcal{X}$. Dans ce cas, on note

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

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

Table des matières

1 Introduction

2 Concepts de base en optimisation

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

L'échelle Linderoth-Luedtke-Leyffer (2016)

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

L'échelle Linderoth-Luedtke-Leyffer (2016)

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	5×10^7
Programme quadratique convexe	5×10^5
Programme conique	5×10^4
Programme quadratique non convexe	300
Problème non convexe	100.

L'échelle Linderoth-Luedtke-Leyffer (2016)

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	5×10^7
Programme quadratique convexe	5×10^5
Programme conique	5×10^4
Programme quadratique non convexe	300
Problème non convexe	100.

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

Quels modèles d'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 :

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

- Sous contraintes convexes :

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

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

Table des matières

1 Introduction

2 Concepts de base en optimisation

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

Programmes convexes

Forme de base

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

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

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

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

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

Exemple : Programmes quadratiques (QP)

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

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

- Extension du cas sans contraintes avec contraintes linéaires;
- Peut avoir une solution même si $\boldsymbol{H} \not\succeq 0$!

Exemples : Programmes semi-définis positifs (SDP)

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

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

[a] $\mathrm{trace}(\boldsymbol{U}\boldsymbol{V}) = \sum_{i=1}^{d} \sum_{j=1}^{d} U_{ij} V_{ij}$.

- 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).

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

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

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

où $\boldsymbol{C}, \boldsymbol{A}_1, \dots, \boldsymbol{A}_n \in \mathbb{R}^{d \times d}$, $\boldsymbol{b} = [b_i] \in \mathbb{R}^n$.
L'ensemble $\mathcal{K}$ est un cône pointé de $\mathbb{R}^{d \times d}$:

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

Exemples : Programmes coniques (CP)

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

où $\boldsymbol{C}, \boldsymbol{A}_1, \dots, \boldsymbol{A}_n \in \mathbb{R}^{d \times d}$, $\boldsymbol{b} = [b_i] \in \mathbb{R}^n$.
L'ensemble $\mathcal{K}$ est un cône pointé de $\mathbb{R}^{d \times d}$:

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

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

Sommaire

1 Introduction

2 Concepts de base en optimisation

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

Schéma de résolution

Pour les classes majeures

- LP, QP, SDP, CP;
- 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.

Cas des programmes linéaires

Problème de base (dit primal)

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

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

Cas des programmes linéaires

Problème de base (dit primal)

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

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

Problème dual (aussi LP !)

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

Cas des programmes linéaires

Problème de base (dit primal)

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

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

Problème dual (aussi LP !)

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

Conditions de KKT

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

$$\begin{array}{ll} \boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{b}, & \boldsymbol{x}^* \geq 0, \\ \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}^* + \boldsymbol{s}^* = \boldsymbol{c}, & \boldsymbol{s}^* \geq 0, \\ x_i^* s_i^* = 0\ \forall i = 1, \dots, d. & \end{array}$$

Dualité forte en programmation linéaire

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 est non borné et l'ensemble réalisable de l'autre est vide (problème irréalisable);

iii) Soit les deux problèmes sont irréalisables.

Dualité forte en programmation linéaire

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 est non borné et l'ensemble réalisable de l'autre est vide (problème irréalisable);
iii) Soit les deux problèmes sont irréalisables.

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.

Points intérieurs

Principe

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

$$\left\{ \begin{array}{lcl} \boldsymbol{A}\boldsymbol{x} & = & \boldsymbol{b} \\ \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{s} & = & \boldsymbol{c} \\ x_i s_i & = & 0 \ \forall i = 1, \dots, d. \end{array} \right.$$

- Obtenir un nouveau point "intérieur" $(\boldsymbol{x}^+, \boldsymbol{y}^+, \boldsymbol{s}^+)$ avec une valeur $[\boldsymbol{x}^+]^{\mathrm{T}}[\boldsymbol{s}^+]$ réduite (via la méthode de Newton).

Points intérieurs

Principe

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

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

- Obtenir un nouveau point "intérieur" $(\boldsymbol{x}^+, \boldsymbol{y}^+, \boldsymbol{s}^+)$ avec une valeur $[\boldsymbol{x}^+]^{\mathrm{T}}[\boldsymbol{s}^+]$ réduite (via la méthode de Newton).

Ça marche !

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

Programmation conique

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

où $\boldsymbol{C}, \boldsymbol{A}_1, \dots, \boldsymbol{A}_m \in \mathbb{R}^{d \times d}$, $\boldsymbol{b} = [b_i] \in \mathbb{R}^m$, 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`).

Table des matières

1 Introduction

2 Concepts de base en optimisation

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

Cadre

Données

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

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

Cadre

Données

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

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

But

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

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

Prédiction idéale

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

Résoudre des systèmes d'équations linéaires

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

Juste de l'algèbre linéaire ?

Un mauvais jeu de données

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

Juste de l'algèbre linéaire ?

Un mauvais jeu de données

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

Expliquer les données par un modèle linéaire

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

$$\left\{ \begin{array}{lcl} x & = & b_1 \\ x & = & b_2 \\ \vdots & & \\ x & = & b_n \end{array} \right.$$

Juste de l'algèbre linéaire ?

Un mauvais jeu de données

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

Expliquer les données par un modèle linéaire

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

$$\left\{ \begin{array}{lcl} x & = & b_1 \\ x & = & b_2 \\ \vdots & & \\ x & = & b_n \end{array} \right.$$

- 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

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimiser}} \, \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_\infty := \max_{1\le i\le n} |\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}-b_i|.$$

Pb 2) Approximation robuste/en norme ℓ_1

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimiser}} \, \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_1 := \sum_{1\le i\le n} |\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}-b_i|.$$

Pb 3) Moindres carrés linéaires

$$\underset{\boldsymbol{x}\in\mathbb{R}^d}{\text{minimiser}} \, \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|^2 := \sum_{1\le i\le n} |\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}-b_i|^2.$$

Pb 1) Approximation de Chebyshev

$$\operatorname*{minimiser}_{\boldsymbol{x}\in\mathbb{R}^d} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_\infty := \max_{1\le i\le n} |\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}-b_i|.$$

Reformulation en programme linéaire

$$\begin{array}{lll}
\operatorname*{minimiser}_{\substack{\boldsymbol{x}\in\mathbb{R}^d \\ t\in\mathbb{R},\boldsymbol{s}\in\mathbb{R}^{2n}}} & t & \\
\text{s. c.} & -t-\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}+b_i+s_i = 0, & i=1,\dots,n \\
& -t+\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}-b_i+s_{n+i} = 0, & i=1,\dots,n \\
& t \ge 0 & \\
& s_i \ge 0 & i=1,\dots,2n.
\end{array}$$

Pb 1) Approximation de Chebyshev

$$\operatorname*{minimiser}_{\boldsymbol{x}\in\mathbb{R}^d} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_\infty := \max_{1\le i\le n} |\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} - b_i|.$$

Reformulation en programme linéaire

$$\begin{array}{llll}
\operatorname*{minimiser}_{\substack{\boldsymbol{x}\in\mathbb{R}^d \\ t\in\mathbb{R},\boldsymbol{s}\in\mathbb{R}^{2n}}} & t & & \\
\text{s. c.} & -t - \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} + b_i + s_i & = 0, & i = 1,\dots,n \\
 & -t + \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} - b_i + s_{n+i} & = 0, & i = 1,\dots,n \\
 & t & \ge 0 & \\
 & s_i & \ge 0 & i = 1,\dots,2n.
\end{array}$$

- 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 ℓ_1

$$\operatorname*{minimiser}_{\boldsymbol{x} \in \mathbb{R}^d} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_1 := \sum_{1 \le i \le n} |\boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{x} - b_i|.$$

Pb 2) Approximation en norme ℓ_1

$$\operatorname*{minimiser}_{\boldsymbol{x}\in\mathbb{R}^d} \|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\|_1 := \sum_{1\le i\le n} |\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}-b_i|.$$

Reformulation en programme linéaire

$$\begin{array}{llll} \operatorname*{minimiser}_{\substack{\boldsymbol{x}\in\mathbb{R}^d \\ \boldsymbol{t}^+,\boldsymbol{t}^-\in\mathbb{R}^n}} & \sum_{i=1}^n \left(t_i^+ + t_i^-\right) & & \\ \text{s. c.} & \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} - b_i - t_i^+ + t_i^- & = 0 & i=1,\dots,n \\ & t_i^+ & \ge 0 & i=1,\dots,n \\ & t_i^- & \ge 0 & i=1,\dots,n \end{array}$$

Pb 2) Approximation en norme ℓ_1

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

Reformulation en programme linéaire

$$\begin{array}{lll} \underset{\substack{\boldsymbol{x}\in\mathbb{R}^d \\ t^+,t^-\in\mathbb{R}^n}}{\text{minimiser}} & \sum_{i=1}^n \left(t_i^+ + t_i^-\right) & \\ \text{s. c.} & \boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x} - b_i - t_i^+ + t_i^- = 0 & i=1,\dots,n \\ & t_i^+ \ge 0 & i=1,\dots,n \\ & t_i^- \ge 0 & i=1,\dots,n \end{array}$$

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

Pb 3) Moindres carrés linéaires

Formulation du problème

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

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

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

Pb 3) Moindres carrés linéaires

Formulation du problème

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

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

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

Propriété

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

Résolution des moindres carrés linéaires

$$\operatorname*{minimiser}_{\boldsymbol{x} \in \mathbb{R}^d} \frac{1}{2n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 .$$

Outil : Décomposition en valeurs singulières

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

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

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

Thèorème (Inverse de Moore-Penrose)

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

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

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

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

$$\boldsymbol{A} \in \mathbb{R}^{n \times d}, \qquad \boldsymbol{b} \in \mathbb{R}^n.$$

Théorème

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

$$\operatorname*{minimiser}_{\boldsymbol{x} \in \mathbb{R}^d} \frac{1}{2n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2.$$

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

$$\boldsymbol{A} \in \mathbb{R}^{n \times d}, \qquad \boldsymbol{b} \in \mathbb{R}^{n}.$$

Théorème

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

$$\operatorname*{minimiser}_{\boldsymbol{x} \in \mathbb{R}^d} \frac{1}{2n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2.$$

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

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

Un exemple

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \qquad \boldsymbol{A} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \ \boldsymbol{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

Une solution

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

Lien avec la régression linéaire

Qualité de la solution des moindres carrés

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

Lien avec la régression linéaire

Qualité de la solution des moindres carrés

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

En présence de données bruitées

- Approche statistique : supposer que $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* + \epsilon$ avec ϵ vecteur aléatoire de loi connue;
- Calculer l'estimateur du **maximum de vraisemblance** en résolvant un problème d'optimisation;
- Équivalent aux moindres carrés pour ϵ suivant une loi gaussienne/normale;
- Équivalent à la régression ℓ_1 pour ϵ suivant une loi de Laplace.

Conclusions

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.

L'optimisation convexe en pratique

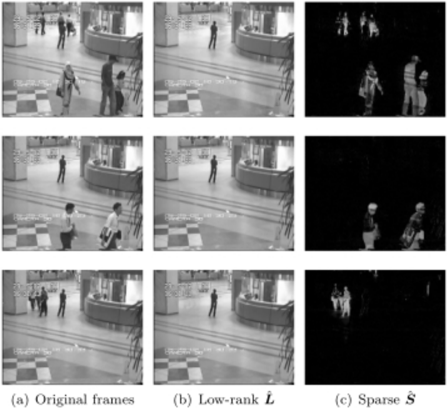


Figure 5.3 Background modeling from video. Three frames from a 200 frame video sequence taken in an airport [LHGT04]. (a) Frames of original video $\boldsymbol{Y}$. (b)-(c) Low-rank $\hat{\boldsymbol{L}}$ and sparse components $\hat{\boldsymbol{S}}$ obtained by PCP.

Source : J. Wright et Y. Ma, *High-Dimensional Data Analysis with Low-Dimensional Models*, 2022.

Références

- 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.
- J. Wright and Y. Ma, *High-Dimensional Data Analysis with Low-Dimensional Models*. Cambridge University Press, 2022.

Codes

- `SciPy/NumPy` : bibliothèque de base en calcul scientifique pour Python.
- `CVX, CVXPY` : logiciels académiques pour l'optimisation convexe (+`IPOPT` pour le cas non convexe).
- `Gurobi, CPLEX, MOSEK` : logiciels commerciaux pour les programmes linéaires (en variables continues et discrètes) et au-delà (programmes coniques).

Fin de la première partie

Aujourd'hui

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

Demain

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

Fin de la première partie

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 !