Optimisation-RO

Clément Royer

Certificat Chef de Projet IA - Université Paris Dauphine-PSL

15 novembre 2021

Intervenant du jour

Clément Royer

- Maître de conférences @ Dauphine-PSL;
- Chercheur permanent @ LAMSADE;
- Chaire tremplin en optimisation @ PRAIRIE;
- Ingénieur @ ENSEEIHT;
- Docteur @ Université de Toulouse.

Informations utiles

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

- Transparents de cours;
- Notebooks Python.

Déroulé de la formation

- Lundi 15/11 matin+après-midi (C. Royer) : Optimisation douce et optimisation convexe;
- Mardi 16/11 après-midi+Jeudi 18/11 matin (P. Ablin) : Méthodes stochastiques à grande échelle;
- Jeudi 18/11 après-midi (C. Royer) : Optimisation sans dérivées.

Pour la séance de ce matin

1. Introduction
2. Concepts de base en optimisation
3. Optimisation sans contraintes

Table des matières

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/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/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

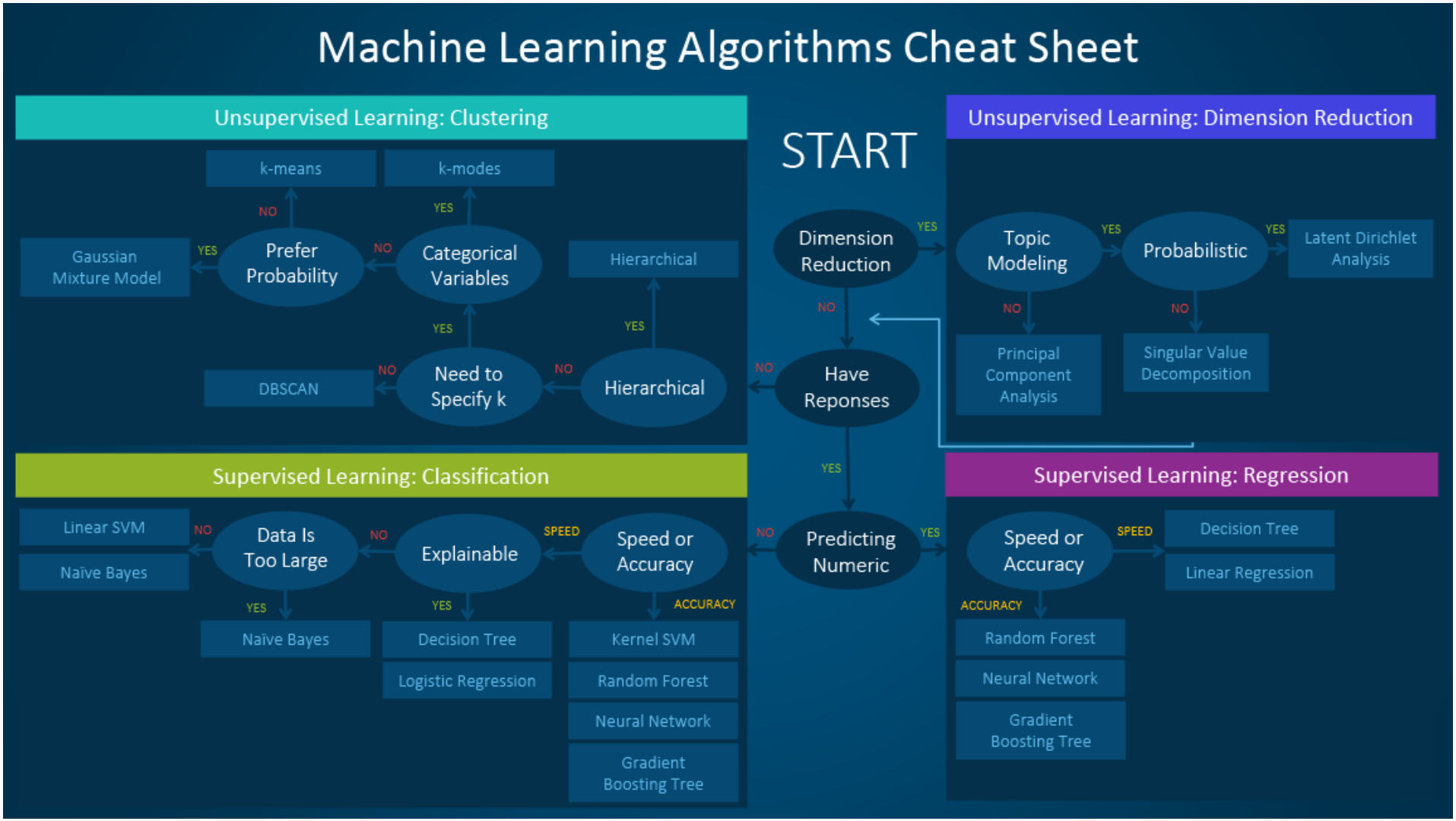
Un ensemble de problèmes basés sur des données

- 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.

Optimisation et apprentissage



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).

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

Problème d'apprentissage statistique

Point de départ : Jeu de de données $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$.

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

Problème d'apprentissage statistique

Point de départ : Jeu de de données $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$.

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

Exemple : classification de documents

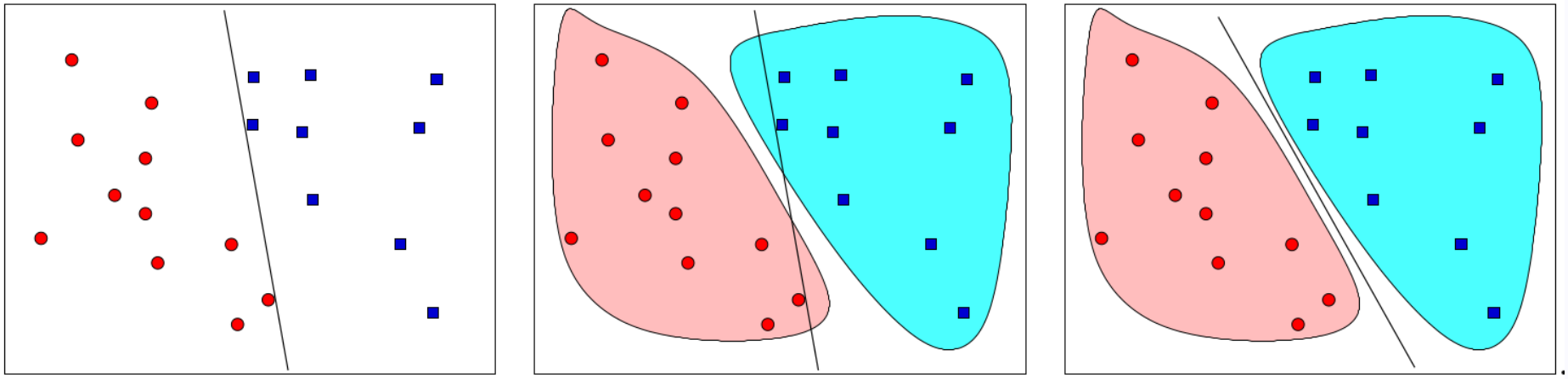
Soit un dictionnaire de d mots.

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

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

- 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 : $\boldsymbol{x}_i$, 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.

Bilan de l'exemple

- 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

- 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

- 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;
- De la théorie (mais pas tout);
- Des applications via des notebook Python.

Table des matières

Table des matières

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 : $\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$

Algèbre linéaire

- $\mathbb{R}^d$: ensemble des vecteurs à $d \geq 1$ coordonnées réelles;
- Pour tous $\boldsymbol{w} \in \mathbb{R}^d$ et $i \in \{1, \ldots, d\}$, $w_i \in \mathbb{R}$ est la i-ème coordonnée de $\boldsymbol{w}$: $\boldsymbol{w} = [w_i]_{1 \leq i \leq d}$;
- On représente $\boldsymbol{w} \in \mathbb{R}^d$ en colonnes : $\boldsymbol{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$;
- On utilise des vecteurs lignes comme "transposés" de vecteurs colonnes : $\boldsymbol{w}^{\mathrm{T}} := [w_1 \cdots w_d]$;

- $\mathbb{R}^d$: ensemble des vecteurs à $d \geq 1$ coordonnées réelles;
- Pour tous $\boldsymbol{w} \in \mathbb{R}^d$ et $i \in \{1, \ldots, d\}$, $w_i \in \mathbb{R}$ est la i-ème coordonnée de $\boldsymbol{w}$: $\boldsymbol{w} = [w_i]_{1 \leq i \leq d}$;
- On représente $\boldsymbol{w} \in \mathbb{R}^d$ en colonnes : $\boldsymbol{w} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$;
- On utilise des vecteurs lignes comme "transposés" de vecteurs colonnes : $\boldsymbol{w}^{\mathrm{T}} := [w_1 \cdots w_d]$;

Opérations vectorielles

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

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

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

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

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

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

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

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

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

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

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

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

Matrices carrées

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

Inversion de matrice

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 $\boldsymbol{BA} = \boldsymbol{AB} = \mathrm{I}_d$, avec 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 $\boldsymbol{A}^{-1}$.

Caractère (semi)-défini positif

Une matrice $\boldsymbol{A} \in \mathbb{R}^{d \times d}$ 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.

Valeurs propres et vecteur propres

Soit $\boldsymbol{A} \in \mathbb{R}^{d \times d}$. Un réel $\lambda \in \mathbb{R}$ est une *valeur propre de* $\boldsymbol{A}$ si

$$\exists \boldsymbol{v} \in \mathbb{R}^d, \|\boldsymbol{v}\| \neq 0, \qquad \boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}.$$

Le vecteur $\boldsymbol{v}$ s'appelle alors un *vecteur propre de* $\boldsymbol{A}$ (associé à λ).

Valeurs propres et vecteur propres

Soit $\boldsymbol{A} \in \mathbb{R}^{d\times d}$. Un réel $\lambda \in \mathbb{R}$ est une *valeur propre de* $\boldsymbol{A}$ si

$$\exists \boldsymbol{v} \in \mathbb{R}^d, \|\boldsymbol{v}\| \neq 0, \qquad \boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}.$$

Le vecteur $\boldsymbol{v}$ s'appelle alors un *vecteur propre de* $\boldsymbol{A}$ (associé à λ).

Une matrice symétrique de $\mathbb{R}^{d\times d}$ possède d valeurs propres réelles. Étant données deux matrices symétriques $(\boldsymbol{A}, \boldsymbol{B}) \in \mathbb{R}^{d\times d}$, on utilisera les notations suivantes :

- $\lambda_{\min}(\boldsymbol{A})/\lambda_{\max}(\boldsymbol{A})$: plus petite/plus grande valeur propre de $\boldsymbol{A}$;
- $\boldsymbol{A} \overset{n}{\succeq} \boldsymbol{B} \Leftrightarrow \lambda_{\min}(\boldsymbol{A}) \geq \lambda_{\max}(\boldsymbol{B})$;
- $\boldsymbol{A} \overset{n}{\succ} \boldsymbol{B} \Leftrightarrow \lambda_{\min}(\boldsymbol{A}) > \lambda_{\max}(\boldsymbol{B})$.

Avec ces notations $\boldsymbol{A}$ sera semi-définie positive (resp. définie positive) si et seulement si $\boldsymbol{A} \succeq 0$ (resp. $\boldsymbol{A} \succ 0$).

Calcul différentiel (2)

On considère une fonction **lisse** (ou douce, ou *smooth*) $f : \mathbb{R}^d \to \mathbb{R}$.

On considère une fonction **lisse** (ou douce, ou *smooth*) $f : \mathbb{R}^d \to \mathbb{R}$.

Dérivée à l'ordre 1

Si f est continûment dérivable sur $\mathbb{R}^d$, on définit pour tout $\boldsymbol{w} \in \mathbb{R}^d$ le **gradient de f en $\boldsymbol{w}$** par

$$\nabla f(\boldsymbol{w}) := \left[\frac{\partial f}{\partial w_i}(\boldsymbol{w})\right]_{1 \leq i \leq d} \in \mathbb{R}^d.$$

L'ensemble des fonctions continûment dérivables est noté $\mathcal{C}^1 \overset{n}{=} \mathcal{C}^1(\mathbb{R}^d, \mathbb{R})$.
On parle de fonction de classe $\mathcal{C}^1$.

Calcul différentiel (2)

On considère une fonction **lisse** (ou douce, ou *smooth*) $f : \mathbb{R}^d \to \mathbb{R}^m$.

Matrice jacobienne

$f : \mathbb{R}^d \to \mathbb{R}^m$ est dérivable en $\boldsymbol{w} \in \mathbb{R}^d$ si il existe une matrice $\boldsymbol{J}_f(\boldsymbol{x}) \in \mathbb{R}^{m \times d}$ telle que

$$\lim_{\substack{\boldsymbol{z} \to \boldsymbol{w} \\ \boldsymbol{z} \neq \boldsymbol{w}}} \frac{\|f(\boldsymbol{z}) - f(\boldsymbol{w}) - \boldsymbol{J}_f(\boldsymbol{w})(\boldsymbol{z} - \boldsymbol{w})\|}{\|\boldsymbol{z} - \boldsymbol{w}\|} = 0.$$

$\boldsymbol{J}_f(\boldsymbol{w})$ s'appelle la (matrice) **jacobienne** de f en $\boldsymbol{w}$.

Cas particuliers

- $m = 1$: la jacobienne se ramène à un vecteur $\nabla f(\boldsymbol{x}) \equiv \boldsymbol{J}_f(\boldsymbol{x})^{\mathrm{T}}$, que l'on appelle le vecteur gradient;
- $n = m = 1$: la jacobienne est équivalente à un scalaire $f'(w)\boldsymbol{J}_f(w)$, que l'on appelle la dérivée de f en $\boldsymbol{w}$.

Calcul différentiel (3)

On considère une fonction **lisse** (ou douce, ou *smooth*) $f : \mathbb{R}^d \to \mathbb{R}$.

On considère une fonction **lisse** (ou douce, ou *smooth*) $f : \mathbb{R}^d \to \mathbb{R}$.

Dérivée d'ordre 2

Si f est *deux fois* continûment dérivable sur $\mathbb{R}^d$, on définit pour tout $\boldsymbol{w} \in \mathbb{R}^d$ la **matrice hessienne de f en $\boldsymbol{w}$** par

$$\nabla^2 f(\boldsymbol{w}) := \left[\frac{\partial^2 f}{\partial w_i \partial w_j}(\boldsymbol{w})\right]_{1 \leq i,j \leq d} \in \mathbb{R}^{d \times d}.$$

Cette matrice est **symétrique**.
L'ensemble des fonctions deux fois continûment dérivables est noté $\mathcal{C}^2$ (on dira que f est de classe $\mathcal{C}^2$.

Développement de Taylor à l'ordre 1

Si $f \in \mathcal{C}^1$, pour tous $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$f(\boldsymbol{w} + \boldsymbol{h}) = f(\boldsymbol{w}) + \int_0^1 \nabla f(\boldsymbol{w} + t\, h)^{\mathrm{T}} \boldsymbol{h}\, dt.$$

Développement de Taylor à l'ordre 1

Si $f \in \mathcal{C}^1$, pour tous $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$f(\boldsymbol{w} + \boldsymbol{h}) = f(\boldsymbol{w}) + \int_0^1 \nabla f(\boldsymbol{w} + t\, h)^{\mathrm{T}} \boldsymbol{h}\, dt.$$

Développement de Taylor à l'ordre 2

Si $f \in \mathcal{C}^2$, pour tous $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$f(\boldsymbol{w} + \boldsymbol{h}) = f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}} \boldsymbol{h} + \tfrac{1}{2} \int_0^1 \boldsymbol{h}^{\mathrm{T}} \nabla^2 f(\boldsymbol{w} + t\, \boldsymbol{h}) \boldsymbol{h}\, dt.$$

Continuité de Lipschitz

Définition

Une fonction $\boldsymbol{g} : \mathbb{R}^d \to \mathbb{R}^m$ est dite L-lipschitzienne si il existe $L > 0$ telle que

$$\forall (\boldsymbol{w}, \boldsymbol{z}) \in (\mathbb{R}^d)^2, \quad \|g(\boldsymbol{w}) - g(\boldsymbol{z})\| \leq L\|\boldsymbol{w} - \boldsymbol{z}\|.$$

La valeur L s'appelle une constante de Lipschitz pour $\boldsymbol{g}$.

- Ex) Toute fonction linéaire est Lipschitzienne;
- $\mathcal{C}_L^{1,1}$: sous-ensemble de $\mathcal{C}^1$ des fonctions avec dérivée première L-lipschitzienne;
- $\mathcal{C}_L^{2,2}$: sous-ensemble de $\mathcal{C}^2$ des fonctions avec dérivée seconde L-lipschitzienne;

Approximation de Taylor à l'ordre 1

Soit $f \in \mathcal{C}_L^{1,1}$. Pour tous $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$f(\boldsymbol{w}+\boldsymbol{h}) \leq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}} \boldsymbol{h} + \frac{L}{2}\|\boldsymbol{h}\|^2.$$

Approximation de Taylor à l'ordre 1

Soit $f \in \mathcal{C}_L^{1,1}$. Pour tous $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$f(\boldsymbol{w}+\boldsymbol{h}) \leq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}} \boldsymbol{h} + \frac{L}{2}\|\boldsymbol{h}\|^2.$$

$\Rightarrow$ **Une des inégalités majeures en optimisation non linéaire.**

Approximation de Taylor à l'ordre 1

Soit $f \in \mathcal{C}_L^{1,1}$. Pour tous $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$f(\boldsymbol{w}+\boldsymbol{h}) \leq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}} \boldsymbol{h} + \frac{L}{2}\|\boldsymbol{h}\|^2.$$

$\Rightarrow$ **Une des inégalités majeures en optimisation non linéaire.**

Approximation de Taylor à l'ordre 2

Soit $f \in \mathcal{C}_L^{2,2}$. Pour tous $\boldsymbol{w}, \boldsymbol{h} \in \mathbb{R}^d$,

$$f(\boldsymbol{w}+\boldsymbol{h}) \leq f(\boldsymbol{w}) + \nabla f(\boldsymbol{w})^{\mathrm{T}} \boldsymbol{h} + \tfrac{1}{2}\boldsymbol{h}^{\mathrm{T}} \nabla^2 f(\boldsymbol{w}) \boldsymbol{h} + \frac{L}{6}\|\boldsymbol{h}\|^3,$$

Références

- De nombreuses notes de cours disponibles;
- **Annexes** de nombreux ouvrages d'optimisation (voire d'IA)!

Mes recommandations

- En français :
  https://www.lpsm.paris/pageperso/bolley/poly-cdiff.pdf
  https://www.lpsm.paris/pageperso/bolley/poly-algebre3.pdf
- En anglais :
  http://vmls-book.stanford.edu/vmls.pdf (Chapters 1-3)
  https://sebastianraschka.com/pdf/books/dlb/appendix_d_calculus.pdf.

Table des matières

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.

Formulation d'un problème d'optimisation

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

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

Formulation d'un problème d'optimisation

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

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

- $\boldsymbol{w}$ representé les **variables de décision**;
- d est la dimension du problème (on prendra toujours $d \geq 1$);
- $f(\cdot)$ est la fonction objectif/de coût/de perte;
- $\mathcal{F}$ est l'ensemble réalisable/admissible regroupant les contraintes sur les variables de décision.

Formulation d'un problème d'optimisation

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

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

- $\boldsymbol{w}$ representé les **variables de décision**;
- d est la dimension du problème (on prendra toujours $d \geq 1$);
- $f(\cdot)$ est la fonction objectif/de coût/de perte;
- $\mathcal{F}$ est l'ensemble réalisable/admissible regroupant les contraintes sur les variables de décision.

Maximiser f revient à minimiser $-f$.

Solutions locales et globales

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

Minimum local

- Un point $\boldsymbol{w}^*$ est un **minimum local** du problème s'il existe un voisinage $\mathcal{N}$ de $\boldsymbol{w}^*$ sur lequel $f(\boldsymbol{w}^*) \leq f(\boldsymbol{w})\ \forall \boldsymbol{w} \in \mathcal{N} \cap \mathcal{F}$;
- Cas sans contraintes : il existe $\epsilon > 0$, $f(\boldsymbol{w}^*) < f(\boldsymbol{w})\ \forall \boldsymbol{w}, \|\boldsymbol{w} - \boldsymbol{w}^*\| \leq \epsilon$.

Minimum global

Un point $\boldsymbol{w}^*$ est un **minimum global** du problème si $f(\boldsymbol{w}^*) \leq f(\boldsymbol{w})\ \forall \boldsymbol{w} \in \mathcal{F}$.

Solutions locales et globales (2)

- Trouver des minima globaux est difficile en général;
- Trouver et certifier des minima locaux peut aussi être difficile.

Solutions locales et globales (2)

- Trouver des minima globaux est difficile en général;
- Trouver et certifier des minima locaux peut aussi être difficile.

Cas "faciles"

- Bonnes propriétés de la fonction objectif;
- Bonne géométrie de l'ensemble des contraintes, le cas échéant.

$\Rightarrow$ L'optimisation douce conduit à certains cas faciles.

Conditions d'optimalité en optimisation douce

Problème sans contraintes minimiser$_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$,
f de classe $\mathcal{C}^1$.

Conditions d'optimalité en optimisation douce

Problème sans contraintes minimiser$_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$,
f de classe $\mathcal{C}^1$.

Condition nécessaire à l'ordre 1

Si $\boldsymbol{w}^*$ est un minimum local du problème, **alors**

$$\|\nabla f(\boldsymbol{w}^*)\| = 0.$$

Problème sans contraintes minimiser$_{\boldsymbol{w}\in\mathbb{R}^d}\, f(\boldsymbol{w})$,
f de classe $\mathcal{C}^1$.

Condition nécessaire à l'ordre 1

Si $\boldsymbol{w}^*$ est un minimum local du problème, **alors**

$$\|\nabla f(\boldsymbol{w}^*)\| \;=\; 0.$$

- Cette condition est seulement nécessaire;
- Un point tel que $\|\nabla f(\boldsymbol{w}^*)\| = 0$ peut aussi être un maximum local ou un point selle.

Conditions d'optimalité en optimisation douce (2)

Problème sans contraintes minimiser$_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$,
f de classe $\mathcal{C}^2$.

Conditions d'optimalité en optimisation douce (2)

Problème sans contraintes minimiser$_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w})$,
f de classe $\mathcal{C}^2$.

Condition nécessaire à l'ordre 2

Si $\boldsymbol{w}^*$ est un minimum local du problème, **alors**

$$\|\nabla f(\boldsymbol{w}^*)\| = 0 \quad \text{et} \quad \nabla^2 f(\boldsymbol{w}^*) \succeq 0.$$

Conditions d'optimalité en optimisation douce (2)

Problème sans contraintes minimiser$_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w})$,
f de classe $\mathcal{C}^2$.

Condition nécessaire à l'ordre 2

Si $\boldsymbol{w}^*$ est un minimum local du problème, **alors**

$$\|\nabla f(\boldsymbol{w}^*)\| = 0 \quad \text{et} \quad \nabla^2 f(\boldsymbol{w}^*) \succeq 0.$$

Condition suffisante à l'ordre 2

Si $\boldsymbol{w}^*$ vérifie

$$\|\nabla f(\boldsymbol{w}^*)\| = 0 \quad \text{and} \quad \nabla^2 f(\boldsymbol{w}^*) \succ 0,$$

alors c'est un minimum local du problème.

Table des matières

Trois approches en optimisation

- **Mathématique :** Prouver l'existence de solutions, le côté bien posé d'une formulation souvent complexe.
- **Logicielle :** Programmer un algorithme pour résoudre une classe de problèmes en pratique.
- **Numérique :** Élaborer des algorithmes, établir des garanties théoriques et valider leur implémentation.

Trois approches en optimisation

- **Mathématique :** Prouver l'existence de solutions, le côté bien posé d'une formulation souvent complexe.
- **Logicielle :** Programmer un algorithme pour résoudre une classe de problèmes en pratique.
- **Numérique :** Élaborer des algorithmes, établir des garanties théoriques et valider leur implémentation.

On traitera essentiellement de la dernière catégorie.

Comment résoudre un problème d'optimisation douce ?

Approche idéale

- Trouver les solutions de $\|\nabla f(\boldsymbol{w})\| = 0$;
- Prendre celle avec la plus faible valeur à l'objectif.

Comment résoudre un problème d'optimisation douce ?

Approche idéale

- Trouver les solutions de $\|\nabla f(\boldsymbol{w})\| = 0$;
- Prendre celle avec la plus faible valeur à l'objectif.

Problème

- Résoudre l'équation n'est pas trivial;
- Il peut y avoir une infinité de solutions;
- L'implémentation de cette procédure doit être possible.

Comment procéder

De manière itérative

- Idée de base : étant donné un point courant, se déplacer vers un point potentiellement meilleur;
- Une itération représente l'ensemble des calculs nécessaires pour ce déplacement.

Notre but dans le reste du cours

- Proposer des algorithmes;
- Décrire leurs garanties théoriques;
- Vérifier leur intérêt pratique (notebooks).

Quelles garanties

Pour résoudre minimiser$_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$, l'algorithme devrait satisfaire les propriétés suivantes :

1. Les points calculés tendent vers une solution;
2. Les valeurs de l'objectif tendent vers la valeur optimale;
3. Une condition d'optimalité est satisfaite à la limite.

Quelles garanties

Pour résoudre minimiser$_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w})$, l'algorithme devrait satisfaire les propriétés suivantes :

1. Les points calculés tendent vers une solution;
2. Les valeurs de l'objectif tendent vers la valeur optimale;
3. Une condition d'optimalité est satisfaite à la limite.

Convergence des itérés

L'algorithme génère une suite $\{\boldsymbol{w}_k\}_k$ telle que

$$\|\boldsymbol{w}_k - \boldsymbol{w}^*\| \to 0 \quad \text{lorsque } k \to \infty,$$

où $\boldsymbol{w}^* \in \text{argmin}_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w})$ est une solution globale du problème.

Quelles garanties (2)

Convergence en valeur de fonction

$$f(\boldsymbol{w}_k) \to f^* \quad \text{lorsque } k \to \infty,$$

où $f^* = \min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$.

Quelles garanties (2)

Convergence en valeur de fonction

$$f(\boldsymbol{w}_k) \to f^* \quad \text{lorsque } k \to \infty,$$

où $f^* = \min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$.

Convergence vers un point stationnaire d'ordre 1

$$\|\nabla f(\boldsymbol{w}_k)\| \to 0 \quad \text{lorsque } k \to \infty.$$

Condition plus générale.

Pourquoi ces conditions ?

Pratique $\neq$ théorie :

- Solution optimale inconnue;
- Valeur à l'optimum inconnue.

Pourquoi ces conditions ?

Pratique $\neq$ théorie :

- Solution optimale inconnue;
- Valeur à l'optimum inconnue.

Du point de vue algorithmique, on peut observer:

- le comportement des itérés;
- le comportement des valeurs de la fonction objectif;
- le comportement de la norme du gradient.

Convergence et vitesses de convergence

- En optimisation, les résultats classiques sont asymptotiques

$$Ex : \quad \|\nabla f(\boldsymbol{w}_k)\| \to 0 \quad \text{lorsque } k \to \infty.$$

Convergence et vitesses de convergence

- En optimisation, les résultats classiques sont asymptotiques

$$Ex: \ \|\nabla f(\boldsymbol{w}_k)\| \to 0 \quad \text{lorsque } k \to \infty.$$

- Les **vitesses de convergence** sont de plus en plus à la mode

$$Ex: \|\nabla f(\boldsymbol{w}_k)\| = \mathcal{O}\left(\frac{1}{k}\right) \quad \Leftrightarrow \quad \exists C > 0, \|\nabla f(\boldsymbol{w}_k)\| \leq \frac{C}{k} \ \forall k.$$

tout comme la notion équivalente de **complexité au pire cas**:

$$Ex: \text{Algorithm } \mathcal{O}\left(\epsilon^{-1}\right) \Leftrightarrow \exists C > 0, \ \|\nabla f(\boldsymbol{w}_k)\| \leq \epsilon \text{ quand } k \geq \frac{C}{\epsilon}.$$

- Garanties en temps/budget fini;
- Classiques en informatique théorique/statistiques.

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.

Conclusions : concepts de base

Modélisation

- Objectif, variables et contraintes;
- Solutions locales et globales.

Conclusions : concepts de base

Modélisation

- Objectif, variables et contraintes;
- Solutions locales et globales.

Outils d'analyse

- Dérivées et développements de Taylor;
- Conditions d'optimalité.

Conclusions : concepts de base

Modélisation

- Objectif, variables et contraintes;
- Solutions locales et globales.

Outils d'analyse

- Dérivées et développements de Taylor;
- Conditions d'optimalité.

Principe algorithmique

- Processus itératifs : trouver une suite de points qui conduit à la solution;
- Étude de la vitesse de convergence.

Révisons! Bases de l'optimisation

1. Comment s'écrit la condition d'optimalité à l'ordre 1 pour la minimisation d'une fonction de classe $\mathcal{C}^1$?
2. Quel objet caractérise la dérivée à l'ordre 2 d'une fonction de classe $\mathcal{C}^2$?
3. Quelle est la différence entre un minimum local et un minimum global ?

Table des matières

Problème

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimiser}}\, f(\boldsymbol{w}).$$

Hypothèses

- f minorée par f^*;
- f douce/lisse $\Rightarrow$ les dérivées de f peuvent être utilisées pour résoudre ce problème.

Deux classes de problèmes

Moindres carrés linéaires

- Reposent sur de l'**algèbre linéaire**;
- Caractérisation explicite des solutions possible;
- Application orientée données : Régression linéaire.

Optimisation douce sans contraintes

- Outil d'analyse : le gradient;
- But : Convergence itérative vers une solution;
- Application orientée données : Régression logistique.

Table des matières

Données

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

$$\Rightarrow \text{Matrice } \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{n \times d} \text{ et vecteur } \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

Cadre

Données

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

$$\Rightarrow \text{Matrice } \boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{n \times d} \text{ et vecteur } \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

But

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

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

Des modèles aux systèmes linéaires

Prédiction idéale

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

Résoudre des systèmes d'équations linéaires

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

Juste de l'algèbre linéaire ?

A dataset

- $\boldsymbol{x}_1 = \boldsymbol{x}_2 = \cdots = \boldsymbol{x}_n = 1$ $(d = 1)$;
- $y_1, \ldots, y_n$ sont distincts (typique de mesures bruitées).

Juste de l'algèbre linéaire ?

A dataset

- $\boldsymbol{x}_1 = \boldsymbol{x}_2 = \cdots = \boldsymbol{x}_n = 1$ $(d = 1)$;
- $y_1, \ldots, y_n$ sont distincts (typique de mesures bruitées).

Expliquer les données par un modèle linéaire

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

$$\begin{cases} w = y_1 \\ w = y_2 \\ \vdots \\ w = y_n \end{cases}$$

Juste de l'algèbre linéaire ?

A dataset

- $\boldsymbol{x}_1 = \boldsymbol{x}_2 = \cdots = \boldsymbol{x}_n = 1$ $(d = 1)$;
- $y_1, \ldots, y_n$ sont distincts (typique de mesures bruitées).

Expliquer les données par un modèle linéaire

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

$$\begin{cases} w = y_1 \\ w = y_2 \\ \vdots \\ w = y_n \end{cases}$$

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

Formulation du problème

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

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimiser}} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 = \frac{1}{2}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}),$$

avec $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{n \times d}$ et $\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$.

Formulation du problème

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

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimiser}} \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2 = \frac{1}{2}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}),$$

avec $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{x}_n^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{n \times d}$ et $\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n$.

Propriétés

- Problème sans contraintes;
- Fonction objectif minorée par 0;
- Objectif doux/lisse : polynomial en les coordonnées de $\boldsymbol{w}$.

Résolution des moindres carrés linéaires

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimiser}} \; \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 .$$

Résolution des moindres carrés linéaires

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimiser}} \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y}\|_2^2 .$$

- Si $\boldsymbol{w}^*$ est une solution du système $\boldsymbol{X}\boldsymbol{w}=\boldsymbol{y}$, c'est aussi une solution du problème d'optimisation !

Résolution des moindres carrés linéaires

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimiser}} \frac{1}{2}\|\boldsymbol{X}\boldsymbol{w}-\boldsymbol{y}\|_2^2 .$$

- Si $\boldsymbol{w}^*$ est une solution du système $\boldsymbol{X}\boldsymbol{w}=\boldsymbol{y}$, c'est aussi une solution du problème d'optimisation !
- Quid du cas où le problème n'a pas de solution ?

Résolution de systèmes linéaires : cas agréable

Système carré

$\boldsymbol{X}\,\boldsymbol{w} = \boldsymbol{y}$, avec $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ et $n = d$.

Cas 1 : $\boldsymbol{X}$ est inversible

$$\boldsymbol{X}\,\boldsymbol{w} = \boldsymbol{y} \;\Leftrightarrow\; \boldsymbol{w} = \boldsymbol{X}^{-1}\boldsymbol{y}.$$

Le système possède une unique solution $\boldsymbol{w}^* = \boldsymbol{X}^{-1}\boldsymbol{y}$, qui est aussi le minimum global du problème d'optimisation minimiser$_{\boldsymbol{w} \in \mathbb{R}^d}$ $\frac{1}{2}\|\boldsymbol{X}\,\boldsymbol{w} - \boldsymbol{y}\|_2^2$.

Exemple avec $d = n = 2$

$$\begin{cases} w_1 + w_2 &= 0, \\ 3w_1 + 2w_2 &= 1. \end{cases}$$

L'unique solution est $\boldsymbol{w} = [1 \; -1]^{\mathrm{T}}$.

Résolution de systèmes linéaires : autres cas

Système carré

$\boldsymbol{X}\,\boldsymbol{w} = \boldsymbol{y}$, avec $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ et $n = d$.

Cas 2: $\boldsymbol{X}$ non inversible

- Il peut ne pas y avoir de solution;
- Il peut en avoir une infinité.

Dans les deux cas, on peut calculer une solution au sens des moindres carrés!

Résolution de systèmes linéaires : autres cas

Système carré

$\boldsymbol{X}\,\boldsymbol{w} = \boldsymbol{y}$, avec $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ et $n = d$.

Cas 2: $\boldsymbol{X}$ non inversible

- Il peut ne pas y avoir de solution;
- Il peut en avoir une infinité.

Dans les deux cas, on peut calculer une solution au sens des moindres carrés!

Autres cas

- $\boldsymbol{X}\,\boldsymbol{w} = \boldsymbol{y}$, avec $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, $n \neq d$;
- Pas de solution, une ou une infinité !

Algèbre linéaire à la rescousse

Ce que l'on voudrait

- L'équivalent d'une inverse;
- Qui donne une solution qu'il en existe ou non.

Ce que l'on voudrait

- L'équivalent d'une inverse;
- Qui donne une solution qu'il en existe ou non.

Pseudo-inverse

Pour toute matrice $\boldsymbol{X} \in \mathbb{R}^{n \times d}$, il existe $\boldsymbol{A} \in \mathbb{R}^{d \times n}$ qui vérifie les équations de Moore-Penrose

$$\begin{cases} \boldsymbol{AXA} &= \boldsymbol{A} \\ \boldsymbol{XAX} &= \boldsymbol{X} \end{cases} \quad \text{and} \quad \begin{cases} (\boldsymbol{AX})^{\mathrm{T}} &= \boldsymbol{AX} \\ (\boldsymbol{XA})^{\mathrm{T}} &= \boldsymbol{XA} \end{cases}$$

Cette matrice s'appelle la pseudo-inverse de $\boldsymbol{X}$ et on la note $\boldsymbol{A} = \boldsymbol{X}^{\dagger}$.
Si $\boldsymbol{X}$ *est inversible,* $\boldsymbol{X}^{\dagger} = \boldsymbol{X}^{-1}$.

$$\boldsymbol{X} \in \mathbb{R}^{n \times d}, \qquad \boldsymbol{y} \in \mathbb{R}^{n}.$$

Théorème

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

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimiser}} \, f(\boldsymbol{w}) = \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2.$$

de norme minimale. Pour tout $\hat{\boldsymbol{w}} \in \operatorname{argmin}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$, on a :

$$\boldsymbol{X} \in \mathbb{R}^{n \times d}, \qquad \boldsymbol{y} \in \mathbb{R}^{n}.$$

Théorème

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

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimiser}} \, f(\boldsymbol{w}) = \frac{1}{2} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|^2.$$

de norme minimale. Pour tout $\hat{\boldsymbol{w}} \in \operatorname{argmin}_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w})$, on a :

- $f\left(\boldsymbol{X}^\dagger \boldsymbol{y}\right) = f(\hat{\boldsymbol{w}})$;
- $\|\boldsymbol{X}^\dagger \boldsymbol{y}\| \leq \|\hat{\boldsymbol{w}}\|$.

Un exemple

$$\boldsymbol{X}\boldsymbol{w} = \boldsymbol{y}, \qquad \boldsymbol{X} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}, \ \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

Une solution

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

En pratique

Points clés

- Calcul de la pseudo-inverse;
- Ou son approximation!

Utilisation de solveurs d'algèbre linéaire

- Décomposition de $\boldsymbol{X}$ (QR, LU, SVD, etc) pour faciliter le calcul;
- Utilisation d'algèbre linéaire itératif (CG, LSQR, etc) pour calculer une solution approchée, prise en compte des erreurs d'arrondi;
- **Éxécution dans des environnemments parallèles/distribués.**

Régression linéaire

- Données : $\{(\boldsymbol{x}_i, y_i)\}_i$, $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$.
- But : calculer un modèle linéaire $h(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$ tel que $h(\boldsymbol{x}_i) \approx y_i$ for $i = 1, \ldots, n$.
- Objectif : minimiser les carrés des erreurs $|h(\boldsymbol{x}_i) - y_i|$.

Régression linéaire

- Données : $\{(\boldsymbol{x}_i, y_i)\}_i$, $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$.
- But : calculer un modèle linéaire $h(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$ tel que $h(\boldsymbol{x}_i) \approx y_i$ for $i = 1, \ldots, n$.
- Objectif : minimiser les carrés des erreurs $|h(\boldsymbol{x}_i) - y_i|$.

Problème d'optimisation

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimiser}} \frac{1}{2n}\sum_{i=1}^{n}(h(\boldsymbol{x}_i) - y_i)^2 = \frac{1}{2n}\|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 .$$

- On retrouve les moindres carrés linéaires !

Régression linéaire (2)

Qualité de la solution des moindres carrés

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

Régression linéaire (2)

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 $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\epsilon}$ avec $\boldsymbol{\epsilon}$ vecteur aléatoire de loi connue;
- Calculer l'estimateur du **maximum de vraisemblance** en résolvant un problème d'optimisation;
- Équivalent aux moindres carrés pour $\boldsymbol{\epsilon}$ gaussien.

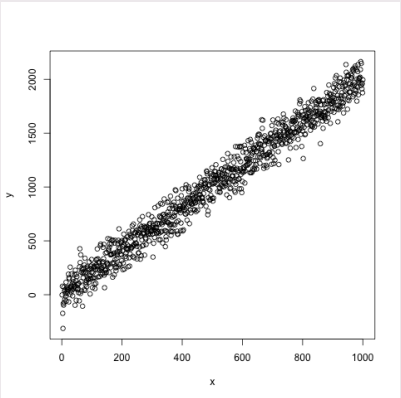
Régression linéaire (2)

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 $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}^* + \boldsymbol{\epsilon}$ avec $\boldsymbol{\epsilon}$ vecteur aléatoire de loi connue;
- Calculer l'estimateur du **maximum de vraisemblance** en résolvant un problème d'optimisation;
- Équivalent aux moindres carrés pour $\boldsymbol{\epsilon}$ gaussien.



En bref : moindres carrés linéaires

Buts

- Trouver une relation linéaire entre composantes d'un jeu de données;
- Fonctionne même en l'absence d'un modèle sous-jacent !

Techniques

- Faire le lien avec les systèmes linéaires;
- Utilisation de solveurs d'algèbres linéaires pour le calcul explicite;
- Interprétation statistique possible, comme dans le contexte de la régression linéaire.

Table des matières

Cadre

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimiser}} \; f(\boldsymbol{w}).$$

Cadre

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimiser}} \, f(\boldsymbol{w}).$$

Hypothèses

- f est minorée par f_{low} ;
- f est lisse (au moins de classe $\mathcal{C}^1$).

Exemple : Nonlinear regression

Contexte

- Jeu de données $\{(\boldsymbol{x}_i, y_i)\}_i$, $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \{0, 1\}$;
- But : Classification via un modèle linéaire $\boldsymbol{x} \mapsto \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}$.

Exemple : Nonlinear regression

Contexte

- Jeu de données $\{(\boldsymbol{x}_i, y_i)\}_i$, $\boldsymbol{x}_i \in \mathbb{R}^d$, $y_i \in \{0, 1\}$;
- But : Classification via un modèle linéaire $\boldsymbol{x} \mapsto \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$.

Fonction objectif

- Basée sur la perte sigmoïde : $\phi(\boldsymbol{x}_i; \boldsymbol{w}) = \left(1 + e^{-\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}}\right)$;
- On pénalise l'erreur $(y_i - \phi(\boldsymbol{x}_i; \boldsymbol{w}))^2$.

Example : Régression non linéaire (2)

Problème d'optimisation

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimiser}}\, f(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \frac{1}{1+e^{-\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}}}\right)^2.$$

Example : Régression non linéaire (2)

Problème d'optimisation

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimiser}}\, f(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \frac{1}{1+e^{-\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}}}\right)^2.$$

- Structure de moindres carrés **non linéaires**;
- Problème lisse : on applique la descente de gradient.

Descente et gradient

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimiser}}\, f(\boldsymbol{w}), \quad f \in \mathcal{C}^1.$$

Pour tout $\boldsymbol{w} \in \mathbb{R}^d$,

1. Soit $\boldsymbol{w}$ est un minimum local et donc $\nabla f(\boldsymbol{w}) = 0$;
2. Soit f décroît localement depuis $\boldsymbol{w}$ dans la direction de $-\nabla f(\boldsymbol{w})$.
   Preuve basée sur Taylor.

Algorithme de descente de gradient

Entrées : $\boldsymbol{w}_0 \in \mathbb{R}^d,\ \alpha_0 > 0,\ \varepsilon > 0, k_{\max} \in \mathbb{N}$.

Set $k = 0$.

1. Evaluer $\nabla f(\boldsymbol{w}_k)$; si $\|\nabla f(\boldsymbol{w}_k)\| < \varepsilon$ terminer.
2. Poser $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)$.
3. Incrémenter k de 1; si $k = k_{\max}$ terminer, sinon aller à l'étape 1.

Algorithme de descente de gradient

Entrées : $\boldsymbol{w}_0 \in \mathbb{R}^d,\ \alpha_0 > 0,\ \varepsilon > 0, k_{\max} \in \mathbb{N}$.
Set $k = 0$.

1. Evaluer $\nabla f(\boldsymbol{w}_k)$; si $\|\nabla f(\boldsymbol{w}_k)\| < \varepsilon$ terminer.
2. Poser $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)$.
3. Incrémenter k de 1; si $k = k_{\max}$ terminer, sinon aller à l'étape 1.

Critères d'arrêt

- Convergence : $\|\nabla f(\boldsymbol{w}_k)\| < \varepsilon$;
- Budget : $k = k_{\max}$.

Choix de la longueur de pas α_k

Pas constant

Si $f \in \mathcal{C}_L^{1,1}$, poser $\alpha_k = \frac{1}{L}$:

- Garantit une décroissance à chaque itération;
- Mais demande de connaître L.

Pas décroissant

Choisir α_k tel que $\alpha_k \to 0$.

- Garantit une décroissance à partir d'un certain rang;
- Mais force la valeur à décrôitre.

Choix de la longueur de pas α_k (2)

En optimisation classique

- Recherche linéaire : À chaque itération, α_k obtenue par retour arrière (*backtracking*) sur un ensemble de valeurs en ordre décroissants (ex: $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots,$).
- La value renvoyée vérifie une condition type décroissance de la valeur de l'objectif.

En apprentissage (notamment profond)

α_k =*Learning rate*

- Utiliser une valeur fixe pendant un certain nombre d'itérations;
- Diminuer progressivement cette valeur selon une règle fixée (*scheduling*).

Exemple : Minimiser une fonction quadratique

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimiser}}\, f(\boldsymbol{w}) = \tfrac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{w} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{w}$$

- Pour tout $\boldsymbol{w} \in \mathbb{R}^d$, $\nabla f(\boldsymbol{w}) = \boldsymbol{A}\boldsymbol{w} - \boldsymbol{b}$;
- Si $\boldsymbol{w}$ est un minimum local, $\nabla f(\boldsymbol{w}) = 0$...
- ...donc $\boldsymbol{w}$ est une solution de $\boldsymbol{A}\boldsymbol{w} = \boldsymbol{b}$.

- Descente de gradient : Algorithme itératif pour résoudre $\boldsymbol{A}\boldsymbol{w} = \boldsymbol{b}$;
- Bien défini même quand le problème n'a pas de solution (càd lorsque $\boldsymbol{A} \not\succeq 0$).

Analyse théorique de la descente de gradient

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}), \qquad f \in \mathcal{C}_L^{1,1}.$$

Rappels : Descente de gradient

- Itération : $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f(\boldsymbol{w}_k)$, terminer si $\nabla f(\boldsymbol{w}_k) = 0$.
- Choix de base en théorie : $\alpha_k = \frac{1}{L}$.

Résultats théoriques

- Convergence : Montrer que $\|\nabla f(\boldsymbol{w}_k)\| \to 0$;
- Vitesse de convergence : Décroissance de $\|\nabla f(\boldsymbol{w}_k)\|$.
- Complexité au pire cas : Effort requis pour obtenir $\|\nabla f(\boldsymbol{w}_k)\| \leq \epsilon$ pour $\epsilon > 0$.

Complexité pour la descente de gradient

Théorème

Si $f \in \mathcal{C}_L^{1,1}$ et $\alpha_k = \frac{1}{L}$, la descente de gradient produit $\boldsymbol{w}_k$ tel que $\|\nabla f(\boldsymbol{w}_k)\| \leq \epsilon$ en au plus

$$2L(f(\boldsymbol{w}_0) - f_{\text{low}})\epsilon^{-2} \quad \text{itérations.}$$

- Même résultat pour d'autres choix pour α_k, dont la recherche linéaire.
- On dit que la complexité de la descente de gradient est en $\mathcal{O}(\epsilon^{-2})$.

Vitesses de convergence pour la descente de gradient

Théorème

Si $f \in \mathcal{C}_L^{1,1}$ et $\alpha_k = \frac{1}{L}$, alors pour tout $K \geq 1$, si $\{\boldsymbol{w}_k\}$ est la suite des itérés produite par l'algorithme de descente de gradient, on a

$$\min_{0 \leq k \leq K-1} \|\nabla f(\boldsymbol{w}_k)\| \leq \frac{\sqrt{2L(f(\boldsymbol{w}_0) - f_{\text{low}})}}{\sqrt{K}}.$$

Interpretation

- On dit que la vitesse de convergence de la descente de gradient est $\mathcal{O}\left(\frac{1}{\sqrt{K}}\right)$.
- Il existe une fonction telle que cette vitesse correspond exactement au comportement de la méthode !

Révisons ! Optimisation sans contraintes

1. Quel est l'avantage des moindres carrés linéaires sur les systèmes linéaires ?
2. Écrire l'itération de la descente de gradient.
3. Quelle est la différence entre une garantie de complexité, une vitesse de convergence et une garantie de convergence ?

Conclusions

Optimisation

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

Optimisation en IA

- Algorithmes classique
- Problèmes structurés et spécifiques;
- Descente de gradient comme méthode de base;
- Autres méthodes (régularisation, second ordre) peuvent aussi être employées).

Références

Ouvrages:

- J. Nocedal et S. J. Wright, *Numerical Optimization, 2nd Ed.*, Springer-Verlag, 2006.
- S. J. Wright, *Optimization Algorithms for Data Analysis*. In *The Mathematics of Data*, M. Mahoney, J. C. Duchi and A. Gilbert (eds), AMS, IAS/Park City Mathematics Institute and SIAM, 2018.

Codes :

- `scipy` : bibliothèque de base en calcul scientifique pour Python.
- `MANOPT/PyMANOPT` : codes MATLAB et Python.

Fin de la première partie

Notebook

- Quelques résultats sur la descente de gradient...
- ...et les moindres carrés.

Cet après-midi : Optimisation convexe.

Fin de la première partie

Notebook

- Quelques résultats sur la descente de gradient...
- ...et les moindres carrés.

Cet après-midi : Optimisation convexe.

Merci beaucoup !

Problème lisse

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimiser}}\, f(\boldsymbol{w})$$

où f est de classe $\mathcal{C}^2$.

Descente de gradient applicable mais :

- Pas invariante par transformation linéaire du problème;
- Pas d'utilisation de la dérivée seconde;
- Pas de vitesse de convergence pour un point stationnaire d'ordre deux.

Itération de type Newton

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \boldsymbol{d}_k, \quad \nabla^2 f(\boldsymbol{w}_k)\boldsymbol{d}_k = -\nabla f(\boldsymbol{w}_k).$$

- Algorithme dit de Newton appliqué à $\nabla f(\boldsymbol{w}) = 0$;
- Pas de garanties autres qu'au voisinage d'une solution en général;
- Peut être modifié (notamment en introduisant un pas α_k) pour obtenir des garanties de complexité/des vitesses de convergence.

Résultats récents

- Même complexité que la descente de gradient pour des problèmes $\mathcal{C}^2$ génériques (ϵ^{-2}));
- Bien meilleurs résultats en pratique et sous d'autres hypothèses (voir séance suivante).

Gestion de contraintes

Problèmes sous contraintes

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

où $\mathcal{F} \subset \mathbb{R}^d$ est l'ensemble des valeurs admissibles pour $\boldsymbol{w}$.

Deux techniques modernes

- Utiliser la structure des contraintes;
- Régulariser plutôt que contraindre.

Contraintes d'égalité

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimiser}}\, f(\boldsymbol{w}) \quad \text{s. c.} \quad \boldsymbol{w} \in \mathcal{F} := \left\{ \boldsymbol{w} \in \mathbb{R}^d \mid \boldsymbol{c}(\boldsymbol{w}) = 0 \right\},$$

où $\boldsymbol{c} : \mathbb{R}^d \to \mathbb{R}^m$.

- Lorsque $\boldsymbol{c}$ est une fonction lisse/douce avec $m \leq d$, l'ensemble réalisable est typiquement une variété riemannienne de $\mathbb{R}^d$ (qui ressemble à $\mathbb{R}^{d-m}$)
  Ex) Sphère $\boldsymbol{c}(\boldsymbol{w}) = \|\boldsymbol{w}\|^2 - 1$.
- On peut définir une notions de gradient sur des variétés)
  $g_f(\boldsymbol{w}) = (\boldsymbol{I}_d - \boldsymbol{w}\boldsymbol{w}^{\mathrm{T}})\nabla f(\boldsymbol{w})$.
- De nombreux algorithmes se généralisent alors à ce contexte.