

Méthodes d'optimisation pour le Machine Learning

1 – Introduction à l'optimisation

Stéphane Canu
stephane.canu@litislab.eu

ÉCOLE MATHÉMATIQUE AFRICAINE :
Bases mathématiques de l'intelligence artificielle

July 8, 2019

Plan

1 Introduction à l'optimisation

- Quelques exemples
 - Variables, contraintes, objectifs
 - L'exemple des moindres carrés
- Formulation générale
 - Objectifs du cours

Optimisation (mathématiques)

L'**optimisation** est une branche des **mathématiques**, cherchant à analyser et à résoudre analytiquement ou numériquement les problèmes qui consistent à déterminer le *meilleur* élément d'un ensemble, au sens d'un *critère* quantitatif donné. Ce mot vient du **latin** *optimum* qui signifie *le meilleur*.

L'optimisation joue un rôle important en **recherche opérationnelle** (donc en **économie** et **microéconomie**), dans les mathématiques appliquées (fondamentales pour l'**industrie** et l'**ingénierie**), en **analyse** et en **analyse numérique**, en **statistique** pour l'estimation du **maximum de vraisemblance** d'une **distribution**, pour la recherche de **stratégies** dans le cadre de la **théorie des jeux**, ou encore en **théorie du contrôle** et de la **commande**.

Aujourd'hui, tous les systèmes susceptibles d'être décrits par un modèle mathématique sont optimisés. La qualité des résultats et des prédictions dépend de la pertinence du modèle, de l'efficacité de l'**algorithme** et des moyens pour le **traitement numérique**.

Pourquoi un cours sur l'optimisation ?

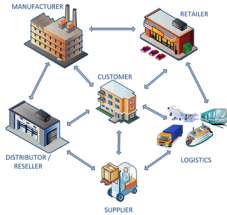
- frontières applicatives → IA
- méthodologique → reformulation
- mathématique → fonction de plusieurs variables
 - ▶ convexité
 - ▶ différentiabilité
- informatique → la question de la complexité
 - ▶ les algorithmes

Que faut-il savoir ?

- ▶ convexité : reconnaître une fonction convexe
- ▶ différentiabilité : savoir calculer un gradient et les KKT
- ▶ reformulation : connaître les principales classes de problème
- ▶ algorithmes : **savoir résoudre pratiquement des problèmes**

Exemples de problèmes d'optimisation

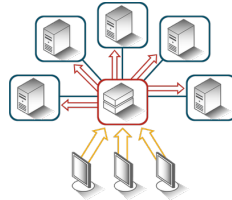
chaîne logistique (*supply chain*)



planification des vols (emplois du temps) *scheduling*



équilibrage d'un réseau (électricité, informatique. . .) (*load balancing*)



calcul de trajectoire



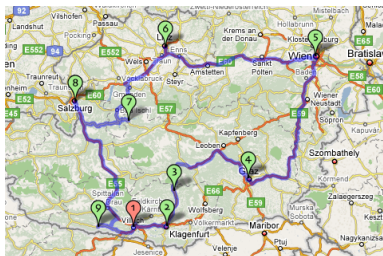
Formalisation des problèmes

Classification des problèmes

Optimisation discrète et sous contraintes

Quelques problème types :

- calcul d'une trajectoire de rentrée dans l'atmosphère d'une navette
- déménageur de Piano, sac à dos, bibliothécaire,
- emploi du temps
- voyageur de commerce. . .

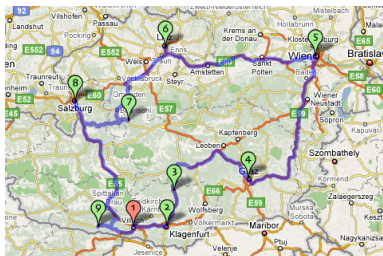


$$\left\{ \begin{array}{l} \min_{x \in \{0,1\}^{n^2}} \sum_{i=1}^n \sum_{j=1}^n x_{ij} d_{ij} \\ \text{avec} \quad \sum_i x_{ij} = 1 \\ \text{et} \quad \sum_j x_{ij} = 1 \\ \text{plus une contrainte éliminant les sous tours...} \end{array} \right.$$

Optimisation discrète et sous contraintes

Quelques problème types :

- calcul d'une trajectoire de rentrée dans l'atmosphère d'une navette
- déménageur de Piano, sac à dos, bibliothécaire,
- emploi du temps
- voyageur de commerce. . .



$$\left\{ \begin{array}{l} \min_{\mathbf{x} \in \{0,1\}^{n^2}} \sum_{i=1}^n \sum_{j=1}^n x_{ij} d_{ij} \\ \text{avec} \quad \sum_i x_{ij} = 1 \\ \text{et} \quad \sum_j x_{ij} = 1 \\ \text{plus une contrainte éliminant les sous tours...} \end{array} \right.$$

Variables, contraintes, objectifs

Dans tout problème d'optimisation on se doit de définir :

- les variables $V \in \mathbb{R}^p$ $x_{ij} \in \{0, 1\}$ toutes les relations entre les villes
- le ou les objectifs $O : V \mapsto \mathbb{R}^{(q)}$ minimiser la distance globale
- les contraintes $C : V \mapsto \mathbb{R}^m$ une entrée et une sortie par ville

Composant électronique

- V = la disposition des composants,
- C = la taille du composant, les couts,
- O = la consommation électrique

Google

- V = paramètres d'un modèle,
- C = les temps de réponse les informations à priori,
- O = maximum du pages pertinentes

optimiser un programme : V = le code..., C = toujours calculer le bon résultat, O = réduire l'utilisation de la mémoire, les temps de calculs, améliorer sa stabilité (robustesse)

...multi critère avec des critères contradictoires

L'exemple des moindres carrés

$$\min_{\mathbf{x} \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \quad \left\{ \begin{array}{l} V = \mathbf{x} \\ O = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 \\ \text{pas de } C \end{array} \right.$$

Généralisation :

$$\min_{\mathbf{x}} \sum_{i=1}^n J(r_i) \quad r_i = \mathbf{a}_i^\top \mathbf{x} - b_i$$

- $J_1(r) = |r|$;
- $J_2(r) = |r|^2$;
- $J_3(r) = \alpha^2 \log(1 - \frac{r^2}{\alpha^2})$

Plan

1 Introduction à l'optimisation

- Quelques exemples
 - Variables, contraintes, objectifs
 - L'exemple des moindres carrés
- Formulation générale
 - Objectifs du cours

Formulation abstraite

Les points qui vérifient les contraintes appartiennent au **domaine de faisabilité** Ω

Soit $\Omega \subseteq \mathbb{R}^p$

$$\min_{\mathbf{x} \in \Omega} J(\mathbf{x}) \quad J : \Omega \mapsto \mathbb{R}$$

On appelle *domaine* de J l'ensemble

$$\text{dom}(J) = \{\mathbf{x} \in \Omega \mid J(\mathbf{x}) < +\infty\}$$

Une fonction J est dite *impropre* si son domaine est vide. Le problème est donc lié à la gestion des **fonctions de plusieurs variables**

Les problèmes liés à l'optimisation

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \Omega} J(\mathbf{x}) \quad J : \Omega \mapsto \mathbb{R}$$

- **existence** et unicité de la solution

→ **convexité**

- conditions nécessaires et suffisante d'optimalité (**caractérisation** de $\mathbf{x}^*$)
→ **différentiabilité**

- **calcul** de $\mathbf{x}^*$ (les aspects algorithmiques et informatiques)

→ **CVX**

- analyse et **reformulation** du problème

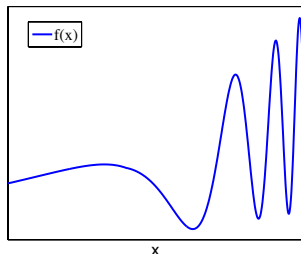
→ **LP, QP, SDP, ...**

Minimisation d'une fonction continue

- le problème : $\min_{x \in \mathbb{R}} f(x)$
- existence et unicité de la solution :
 - convexe ?
- caractérisation mathématique de la solution :

$$x^* = \arg \min_{x \in \mathbb{R}} f(x) \Rightarrow f'(x^*) = 0$$

- calcul de la solution :
 - solution exacte analytique de $f'(x^*) = 0$
 - solution approchée : construction d'une suite x_k qui converge



- le problème multidimensionnel : l'exemple des moindres carrés –
Trouver le polynôme qui approche au mieux la fonction ci-dessus

$$\min_{x \in \mathbb{R}^p} f(x) = \frac{1}{2} \|Ax - b\|^2 \qquad \nabla_x f(x^*) = 0$$

Réécriture des domaines

$$\min_{\mathbf{x} \in \Omega} J(\mathbf{x}) \quad J : \Omega \mapsto \mathbb{R}$$

$$\Omega = \{\mathbf{x} \in \mathbb{R}^p \mid H_i(\mathbf{x}) \leq 0; i = 1, m\}$$

Forme standard

$$\begin{cases} \min_{\mathbf{x} \in \mathbb{R}^p} J(\mathbf{x}) & J : \mathbb{R}^p \mapsto \mathbb{R} \\ \text{avec } H(\mathbf{x}) \leq 0 & H : \mathbb{R}^p \mapsto \mathbb{R}^m \end{cases}$$

Réécritures

- égalité vers inégalité $G(\mathbf{x}) = 0 \quad \Leftrightarrow \quad G(\mathbf{x}) \leq 0 \text{ et } G(\mathbf{x}) \geq 0$.
- égalité vers inégalité : à l'aide de variables d'écart $\mathbf{e} \in \mathbb{R}^m$:

$$H(\mathbf{x}) \leq 0 \quad \Leftrightarrow \quad H(\mathbf{x}) + \mathbf{e} = 0; \text{ avec } \mathbf{e} \geq 0$$

Formulation « pratique »

$$\left\{ \begin{array}{lll} \min_{\mathbf{x} \in \Omega \subseteq \mathbb{R}^p} & J(\mathbf{x}) & J : \mathbb{R}^p \mapsto \mathbb{R} \\ \text{avec} & G(\mathbf{x}) = 0 & G : \mathbb{R}^p \mapsto \mathbb{R}^n \\ \text{et} & H(\mathbf{x}) \leq 0 & H : \mathbb{R}^p \mapsto \mathbb{R}^m \end{array} \right.$$

Reformulation

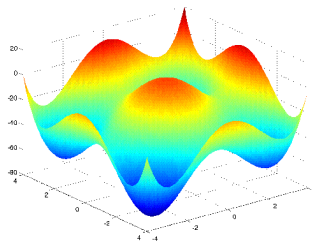
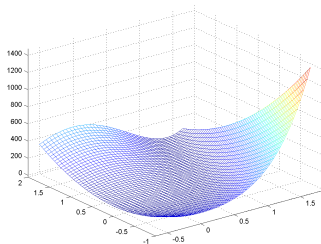
$$\min_{\mathbf{x} \in \Omega} J(\mathbf{x}) \quad \Leftrightarrow \quad \max_{\mathbf{x} \in \Omega} -J(\mathbf{x})$$

Exemple (LP) :

$$\left\{ \begin{array}{lll} \min_{\mathbf{x} \in \mathbb{R}^p} & \mathbf{c}^\top \mathbf{x} & J \\ \text{avec} & A\mathbf{x} = \mathbf{b} & G \\ \text{et} & 0 \leq \mathbf{x} & H \end{array} \right.$$

Solutions des Problèmes d'optimisation

$$\mathbf{x} \in \mathbb{R}^2 \quad J : \mathbb{R}^2 \mapsto \mathbb{R}$$



- solution (minimum) globale : $\mathbf{x}^* = \arg \min_{\mathbf{x} \in \Omega} J(\mathbf{x})$
- minimum local : $\mathbf{x}^\ell = \arg \min_{\mathbf{x} \in V_\varepsilon(\mathbf{x}^\ell)} J(\mathbf{x})$ $V_\varepsilon(\mathbf{x})$: un voisinage de $\mathbf{x}$
- solution approchée : $\mathbf{x}^a \in V_\varepsilon(\mathbf{x}^\ell)$ $V_\varepsilon(\mathbf{x}) = \{\mathbf{x} \in \mathbb{R}^p \mid \|\mathbf{x} - \mathbf{x}^a\| < \varepsilon\}$

Problèmes d'optimisation **convexes**

$$\begin{cases} \min_{\mathbf{x} \in \mathbb{R}^p} J(\mathbf{x}) & J : \mathbb{R}^p \mapsto \mathbb{R} \\ \text{avec } H(\mathbf{x}) \leq 0 & H : \mathbb{R}^p \mapsto \mathbb{R}^m \end{cases}$$

Definition

Problème d'optimisation convexe [Boyd and Vandenberghe, 2004]

- J fonction convexe
- $\Omega = \{\mathbf{x} | H(\mathbf{x}) \leq 0\}$ ensemble convexe

Problème convexe

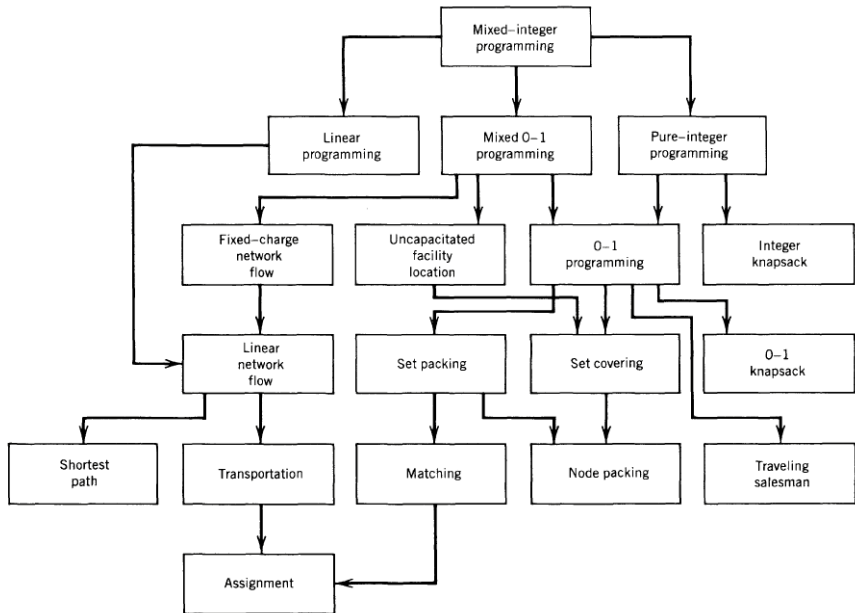
- peut être résolu efficacement
- superbe théorie (très pratique)
 - ▶ $\mathbf{x}^\ell = \mathbf{x}^\star$ (toute solution locale est aussi globale)
- souvent le cas des problèmes d'ingénierie
- ... parfois non reconnus !

Classification d'après [Minoux, 1983]

- entier vs. continue
- linéaires vs. non linéaires
- mono objectifs vs. multi objectifs
- convexes vs. non convexes

objectif J	contraintes G	domaine Ω	nature du problème
linéaire	linéaires	$\mathbb{R}^p$	programmation linéaire
linéaire	linéaires	$\mathbb{N}^p$	programmation linéaire en nombre entiers
quadratique	linéaires	$\mathbb{R}^p$	programmation quadratique
non linéaire	$\emptyset$	$\mathbb{R}^p$	optimisation continue sans contraintes
non linéaire	$\emptyset$	$\mathbb{N}^p$	optimisation discrète sans contraintes
convexe	convexe	$\mathbb{R}^p$	optimisation convexe
non linéaire	$\forall$	$\mathbb{R}^p$	optimisation (programmation mathématique)
multivoque	$\forall$	$\mathbb{R}^p$	optimisation multicritères

Différent types de problèmes



Résolution des problèmes d'optimisation

En général :

- modélisation réductrice
- très difficile à résoudre
- nécessite des compromis entre précision et ressources (temps de calcul)

Exceptions

- moindres carré (optimisation sans contrainte)
- programmation linéaire
- problèmes convexes
 - ▶ solution unique (belles maths)
 - ▶ difficile à reconnaître
 - ▶ des astuces pour les transformer (réécrire)
 - ▶ algorithmes efficaces

Évolutions algorithmiques

Le cas de la programmation linéaire :

$$\left\{ \begin{array}{ll} \min_{\mathbf{x} \in \mathbb{R}^p} & J(\mathbf{x}) = \sum_{i=1}^p c_i x_i \\ \text{avec} & \sum_{i=1}^p a_{ij} x_i = b_j; \quad j = 1, m \\ \text{et} & 0 \leq x_i; \quad i = 1, m \end{array} \right.$$

avant force brute = énumération : $\mathcal{O}(p!)$

très petits problèmes $p < 10$

1947 simplexe : énumération à cout décroissant : $\leq \mathcal{O}(p!)$

petits problèmes $p < 100$

1984 point intérieur : approximation : $\mathcal{O}(p^3)$

problèmes moyens $p < 10\,000$

2009 méthodes de gradient : approximation : $\mathcal{O}(p)$

grands problèmes

Big Data

Objectifs du cours

Modélisation reconnaître et (re)formuler les problèmes

- ▶ classer les problèmes d'optimisation (LP, QP ...)
[Nocedal and Wright, 2006]

Théorie caractérisation de la solution

- ▶ convexité
- ▶ conditions d'optimalité
- ▶ calcul de gradients
- conditions de convergence

Méthodes présenter quelques algorithmes de base

- ▶ gradient
- ▶ programmation linéaire
- ▶ programmation quadratique

Mise en œuvre savoir aborder leur résolution (en python) (pour les problèmes de taille moyenne)

 Boyd, S. P. and Vandenberghe, L. (2004).
Convex optimization.
Cambridge university press.

 Minoux, M. (1983).
Programmation mathématique: théorie et algorithmes, volume 1.
Dunod Paris.

 Nocedal, J. and Wright, S. J. (2006).
Numerical Optimization.
Springer, New York, 2nd edition.