Optimisation

Dr. Ibtissem DIDI

26 avril 2020

Plan

Introduction

Les méthodes multidimensionnelles sont consacrées à l'optimisation de fonction à un paramètre ou plus. Elles peuvent être classées selon l'information sur la fonction qu'elles utilisent. Elles sont dites *d'ordre 0*, si elles n'utilisent que la valeur de la fonction. Elles sont dites *d'ordre 1*, si elles nécessitent en plus le gradient de la fonction. Elles sont dites *d'ordre 2*, si elles utilisent le gradient et le héssien de la fonction.

Les méthodes d'ordre 0 sont en général peu précises et convergent très lentement vers l'optimum. Par contre, elles offrent l'avantage de se passer du calcul du gradient, ce qui peut être très intéressant lorsque la fonction n'est pas différentiable ou lorsque le calcul de son gradient est complexe ou représente un coût important.

Introduction

Les méthodes d'ordre 1 permettent d'accélérer la localisation de l'optimum, car le gradient donne une information sur la direction de recherche de la solution. Par contre, elles ne sont applicables qu'aux problèmes dans lesquels la fonction est continûment différentiable.

Les méthodes multidimensionnelles peuvent être divisées en deux groupes, d'une part, les méthodes analytiques ou de descente, et d'autre part, les méthodes heuristiques ou géométriques.

Les méthodes analytiques se basent sur la connaissance d'une direction de recherche, souvent donnée par le gradient de la fonction. Les exemples les plus signifcatifs de méthodes analytiques sont **les méthodes de descente (gradient à pas fixe/gradient à pas optimal), la méthode du Gradient Conjugué et les méthodes Newton/Quasi-Newton**.

Introduction

Nous allons maintenant nous intéresser aux algorithmes de calcul de minimum et plus particulièrement aux algorithmes de descente. Partant d'un point x_0 arbitrairement choisi, un algorithme de descente va chercher à générer une suite d'itérés $(x_k)_k \in \mathbb{N}$ telle que :

$$\forall k \in \mathbb{N}, f(x_{k+1}) \leq f(x_k)$$

Commençons par définir plus précisément la notion de descente.

Le gradient joue un rôle essentiel en optimisation.Dans le cadre des méthodes d'optimisation, il sera également important d'analyser le comportement de la fonction objectif dans certaines directions. Commençons pour cela par rappeler le concept de dérivée directionnelle :

Définition

Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ une application continue. Soit $x \in \mathbb{R}^n$ et $d \in \mathbb{R}^n$. La dérivée directionnelle de f en x dans la direction d est définie par :

$$df(x; d) := \lim_{t \to 0^+} \frac{f(x + td) - f(x)}{t}$$

si cette limite existe.

Proposition

Si f est différentiable en un point $x \in \mathbb{R}^n$, alors pour tout $d \neq 0$, f admet une dérivée dans la direction d en x et :

$$df(x; d) = Df(x)(d) = \nabla f(x)^T d$$

La dérivée directionnelle donne des informations sur la pente de la fonction dans la direction d, tout comme la dérivée donne des informations sur la pente des fonctions à une variable. En particulier,

- si $df(x; d) > 0$ alors f est croissante dans la direction d.
- si $df(x; d) < 0$ alors f est décroissante dans la direction d.

Dans ce dernier cas, on dira que d est une direction de descente de f.

Définition (Direction de descente)

Soient $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ et $x \in \mathbb{R}^n$. Le vecteur $d \in \mathbb{R}^n$ est une dircetion de descente pour f à partir du point x si $t \to f(x+td)$ est décroissante en $t = 0$, c'est-à-dire s'il existe $\eta > 0$ tel que :

$$\forall 0 < t \leq \eta, f(x+td) < f(x)$$

Parmi toutes les directions de descente existantes en un point x donné, il est naturel de s'intéresser à celle où la pente est la plus forte. Un résultat remarquable montre que cette direction est donnée par le gradient (ou plus exactement son opposé).

Théorème

Soit $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ différentiable. Soit $x \in \mathbb{R}^n$. Alors pour toute direction d de norme constante égale à $\|d\| = \|\nabla f(x)\|$, on a :

$$(-\nabla f(x))^T \nabla f(x) \leq d^T \nabla f(x)$$

Partant d'un point x_0 arbitrairement choisi, un algorithme de descente va chercher à générer une suite d'itérés $(x_k)_{k \in \mathbb{N}}$ telle que :

$$\forall k \in \mathbb{N}, f(x_{k+1}) \leq f(x_k)$$

D'après la caractérisation de la descente, il s'agit donc à chaque itération k, de trouver un point x_{k+1} dans une direction d vérifiant : $\nabla f(x_k)^T d < 0$.
Le schéma général d'un algorithme de descente est le suivant :

Algorithme

Données : f supposé au moins différentiable, x_0 point initial arbitrairement choisi.
Sortie : une approximation de la solution du problème : $\min\limits_{x \in \mathbb{R}^n} f(x)$.

1. k=0
2. Tant que "test de convergence" non satisfait
   1. Trouver une direction de descente d_k telle que $\nabla f(x_k)^T d < 0$.
   2. *Recherche linéaire* : Choisir un pas $\alpha_k > 0$ à faire dans la direction d_k tel que : $f(x_k + \alpha_k d_k) \leq f(x_k)$.
   3. Mise à jour : $x_{k+1} = x_k + \alpha_k d_k$; $k = k + 1$
3. Retourner x_k

Test de convergence/Test d'arrêt

Soit $\bar{x}$ un point de minimum local du critère f à optimiser. Supposons que l'on choisisse comme test d'arrêt dans l'algorithme de descente, le critère idéal : $x_k = \bar{x}$. Dans un monde idéal (i.e. en supposant tous les calculs exacts et la capacité de calcul illimitée), soit l'algorithme s'arrête après un nombre fini d'itérations, soit il construit (théoriquement) une suite infinie x_1, x_2,..., x_k,... de points de $\mathbb{R}^n$ qui converge vers $\bar{x}$.
En pratique, un test d'arrêt devra être choisi pour garantir que l'algorithme s'arrête toujours après un nombre fini d'itérations et que le dernier point calculé soit suffisamment proche de $\bar{x}$.

Soit $\epsilon > 0$ la précision demandée. Plusieurs critères sont à notre disposition : tout d'abord (et c'est le plus naturel), un critère d'optimalité basé sur les conditions nécessaires d'optimalité du premier ordre : on teste si

$$\|\nabla f(x_k)\| < \epsilon$$

auquel cas l'algorithme s'arrête et fournit l'itéré courant x_k comme solution.
En pratique, le test d'optimalité n'est pas toujours satisfait et on devra faire appel à d'autres critères (fondés sur l'expérience du numérique) :

- Stagnation de la solution : $\|x_{k+1} - x_k\| < \epsilon \|x_k\|$.
- Stagnation de la valeur courante : $|f(x_{k+1}) - f(x_k)| < \epsilon |f(x_k)|$

- Nombre d'itérations dépassant un seuil fixé à l'avance : $k < kmax$.

et généralement une combinaison de ces critères :

Critère d'arrêt = Test d'optimalité satisfait
OU (Stagnation de la valeur courante et Stagnation de la solution)
OU Nombre maximum d'itérations autorisées dépassé.

Convergence et vitesse de convergence

Étudier la convergence d'un algorithme, c'est étudier la convergence de la suite des itérés générés par l'algorithme. Un algorithme de descente selon le modèle précédent,est dit convergent si la suite de ses itérés $(x_k)_{k\in\mathbb{N}}$ converge vers un point limite $\bar{x}$, solution du problème :

$$\min_{x\in\mathbb{R}^n} f(x)$$

De plus, la convergence est dite *locale* si elle n'a lieu que pour des points initiaux x_0 dans un voisinage de $\bar{x}$. Sinon elle est dite *globale*.

En pratique, le but d'un algorithme d'optimisation ne sera que de trouver un point critique. On introduit alors la notion de convergence globale d'un algorithme d'optimisation :

Définition

Soit un algorithme itératif qui génère une suite $(x_k)_{k\in\mathbb{N}}$ dans $\mathbb{R}^n$ afin de résoudre le problème :

$$\min_{x\in\mathbb{R}^n} f(x)$$

où $f : \mathbb{R}^n \to \mathbb{R}$ est une application de classe C^1. L'algorithme est dit *globalement convergent* si quel que soit le point initial $x_0 \in \mathbb{R}^n$,

$$\lim_{k\to+\infty} \|\nabla f(x_k)\| = 0$$

Cette propriété garantit que le critère d'arrêt "$\|\nabla f(x_k)\| \leq \epsilon$" sera satisfait à partir d'un certain rang quelle que soit la précision $\epsilon > 0$ demandée.

Il est bien entendu très important de garantir la convergence d'un algorithme sous certaines hypothèses, mais la vitesse de convergence et la complexité sont également des facteurs à prendre en compte lors de la conception ou de l'utilisation d'un algorithme ; en effet, on a tout intérêt à ce que la méthode choisie soit à la fois rapide, précise et stable. Pour cela, on introduit les notions de vitesse (ou taux) de convergence qui mesurent l'évolution de l'erreur commise $\|x_k - \bar{x}\|$.

Définition

Soit $(x_k)_{k\in\mathbb{N}}$ une suite d'itérés générées par un algorithme convergent donné. On note $\bar{x}$ la limite de la suite $(x_k)_{k\in\mathbb{N}}$ et on suppose : $\forall k \in \mathbb{N}, x_k \neq \bar{x}$ (sinon l'algorithme convergerait en un nombre fini d'itérations). La convergence de l'algorithme est dite :

- **linéaire** si l'erreur $e_k = \|x_k - \bar{x}\|$ décroît linéairement i.e. s'il existe $\tau \in]0,1[$ tel que :

$$\lim_{k\to+\infty} \frac{\|x_{k+1} - \bar{x}\|}{\|x_k - \bar{x}\|} = \tau$$

- **superlinéaire** si

$$\lim_{k\to+\infty} \frac{\|x_{k+1} - \bar{x}\|}{\|x_k - \bar{x}\|} = 0$$

Définition (suite)

- d'ordre p s'il existe $\tau \geq 0$ tel que :

$$\lim_{k \to +\infty} \frac{\|x_{k+1} - \bar{x}\|}{\|x_k - \bar{x}\|^p} = \tau$$

En particulier, si $p = 2$, la convergence est dite **quadratique** (grosso modo à partir d'un certain rang, le nombre de chiffres significatifs exacts double à chaque itération).

Bien entendu, on a intérêt à ce que la convergence d'un algorithme soit la plus élevée possible afin de converger vers la solution en un minimum d'itérations pour une précision donnée.

Algorithme de descente du gradient

Soit $x_k \in \mathbb{R}^n$ l'itéré courant. Étant donnés la valeur $f(x_k)$ et le gradient $\nabla f(x_k)$, on remplace f au voisinage de x_k par son développement de Taylor au premier ordre :

$$f(x_k + d) \approx f(x_k) + \nabla f(x_k)^T d$$

On voudrait que la dérivée directionnelle $\nabla f(x_k)^T d$ soit la plus petite possible dans un voisinage de $d = 0$. On cherche donc à résoudre :

$$\min_{d \in \mathbb{R}^n} \nabla f(x_k)^T d \text{ s.c. } \|d\| = \|\nabla f(x_k)\|$$

dont la solution nous est donnée par :

$$d_k = -\nabla f(x_k)$$

Le choix de la direction de plus forte descente définit une famille d'algorithmes appelés algorithmes de descente de gradient dont le schéma est le suivant :

Algorithme

Données : f , x_0 ppremière approximation de la solution cherchée, $\epsilon > 0$ précision demandée.
Sortie : une approximation $x^\star$ de la solution de : $\nabla f(x) = 0$.

1. k=0
2. Tant que le critère d'arrêt non satisfait
   1. Direction de descente $d_k = -\nabla f(x_k)$.
   2. *Recherche linéaire* : Trouver un pas $\alpha_k > 0$ tel que : $f(x_k + \alpha_k d_k) < f(x_k)$.
   3. $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$; $k = k + 1$
3. Retourner x_k

Il reste maintenant à définir une stratégie de recherche linéaire pour le calcul du pas. Nous étudions ici en première approche une méthode à pas optimal, puis une à pas fixe.

Méthode de la plus forte pente (steepest descent)

Une idée naturelle consiste à suivre la direction de plus forte descente et à faire un pas qui rende la fonction à minimiser la plus petite possible dans cette direction. Cette méthode est appelée **méthode de gradient à pas optimal** ou encore **méthode de plus forte pente**. L'étape 2.1 de l'algorithme de descente de gradient est alors remplacée par :

Recherche linéaire exacte

2.1. Calculer un pas optimal α_k solution de :

$$\min_{\alpha > 0} f(x_k + \alpha d_k)$$

La méthode de plus forte pente est une sorte d'idéalisation : d'une part, nous ne savons pas en pratique calculer de façon exacte un point minimum α_k de l'objectif dans une direction donnée et le problème n'est en général pas trivial. D'autre part, la résolution du problème de minimisation unidimensionnel de l'étape 2.1, même de façon approchée, coûte cher en temps de calcul. Pour ces raisons, on peut lui préférer parfois l'algorithme de gradient à pas constant (ou à pas fixe).

Méthode de gradient à pas fixe

L'idée est très simple : on impose une fois pour toutes, la taille du pas effectué selon la direction de descente calculée à chaque itération. Les itérations 2.2 et 2.3 de l'algorithme de descente de gradient sont alors remplacées par :

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

Méthode de la plus forte pente avec Hessien

Si le hessien de $f(x)$ existe et peut être calculé, la valeur α qui minimise $f(x_k + \alpha d)$ peut être déterminée en utilisant un méthode analytique.
Si $\delta_k = \alpha d_k$ et en posant $\nabla f(x_k) = g_k$, alors le développement de la série de Taylor à l'ordre 2 est donnée par :

$$f(x_k + \delta_k) \approx f(x_k) + \delta_k^T g_k + \frac{1}{2}\delta_k^T H_k \delta_k$$

et si d_k est la direction de la plus forte pente, i.e.

$$\delta_k = -\alpha g_k$$

on obtient

$$f(x_k - \alpha g_k) \approx f(x_k) - \alpha g_k^T g_k + \frac{1}{2}\alpha^2 g_k^T H_k g_k$$

Méthode de la plus forte pente avec Hessien

En dérivant par rapport à α et en posant le résultat égal à zéro, on a :

$$\frac{df(x_k - \alpha g_k)}{d\alpha} \approx -g_k^T g_k + \alpha g_k^T H_k g_k = 0$$

Ainsi

$$\alpha = \alpha_k \approx \frac{g_k^T g_k}{g_k^T H_k g_k}$$

Par suite,

$$x_{k+1} = x_k - \frac{g_k^T g_k}{g_k^T H_k g_k} g_k$$

Convergence et vitesse de convergence

Si $f(x) \in C^2$ admet un minimum au moins local $\bar{x}$ et son hessien est D.P. au point $x = \bar{x}$, alors on peut montrer que si x_k est suffisamment proche de $\bar{x}$, on a :

$$f(x_{k+1}) - f(\bar{x}) \leq \left(\frac{1-r}{1+r}\right)^2 [f(x_k) - f(\bar{x})]$$

où

$$r = \frac{\text{La plus petite valeur propre de } H_k}{\text{La plus grande valeur propre de } H_k}$$

Ainsi la méthode de la plus forte pente converge linéairement avec un taux de convergence

$$\tau = \frac{1-r}{1+r}$$