TP4 : OPTIMISATION CONVEXE

COURS D'APPRENTISSAGE, ECOLE NORMALE SUPÉRIEURE

Raphaël Berthier
raphael.berthier@inria.fr

1. Choix des pas dans la descente de gradient pour la minimisation de quadratiques

Dans cette exercice, on s'intéresse à la minimisation d'une fonction quadratique, c'est-à-dire une fonction de la forme

$$f(x) = \frac{1}{2}x^T Hx - b^T x\,,$$

où $H \succ 0$ est une matrice $d \times d$ symmétrique définie positive et $b \in \mathbb{R}^d$. Notons x^* le minimum de f que l'on cherche à atteindre. Pour cela, on propose une méthode de descente de gradient avec pas constant γ :

$$x_{t+1} = x_t - \gamma \nabla f(x_t)\,.$$

1) Montrer l'équation de récurrence $x_{t+1} - x^* = (I_d - \gamma H)(x_t - x^*)$.

2) Pour analyser cette recurrence, on décide de diagonaliser la matrice H : notons $\lambda_1 > \cdots > \lambda_d > 0$ ses valeurs propres et $u_1, \dots, u_d$ les vecteurs propres associés. Quelles sont, dans cet exemple, les constantes μ et L telle que f est μ-fortement convexe (μ-strongly convexe) et L-lisse (L-smooth) ?

3) Montrer que

$$\langle x_t - x^*, u_i\rangle = (1 - \gamma\lambda_i)^t \langle x_0 - x^*, u_i\rangle\,.$$

4) En déduire le choix de γ qui maximise la vitesse de convergence de $\|x_t - x^*\|_2$. Comparer avec le choix proposé dans le cours.

En pratique, la vitesse de convergence décrite ci-dessus peut devenir très lente lorsque le ratio μ/L se déteriore. Pour y remédier, on propose d'utiliser une méthode de la forme

$$x_{t+1} = x_t - \gamma \nabla f(x_t) + \beta(x_t - x_{t-1})\,. \tag{1}$$

Cette itération s'interprète de la manière suivante : la vitesse $x_{t+1} - x_t$ à l'instant t est déterminée par le gradient, comme dans le cas de la descente de gradient simple, auquel on ajoute un peu de la vitesse $x_t - x_{t-1}$ à l'instant $t-1$. Ce terme supplémentaire accélère les convergences lentes et ralentit les convergences oscillantes.

Le choix des paramètres γ et β peut être optimisé comme pour la descente de gradient simple. Pour simplifier les calculs, on donne directement la solution optimale :

$$\gamma = \frac{4}{(\sqrt{L} + \sqrt{\mu})^2}\,, \qquad \beta = \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}\right)^2\,. \tag{2}$$

5) Calculer l'équation de récurrence d'ordre 2 satisfaite par $\langle x_t - x^*, u_i\rangle$ et la résoudre. En déduire la vitesse de convergence de $\|x_t - x^*\|_2$.

La méthode définie par les équation (1)-(2) s'appelle la méthode *heavy-ball* car elle peut être interprétée comme la discrétisation de l'équation différentielle satisfaite par une boule qui roulerait

sur la quadratique f. Plus généralement, il est fréquent de choisir x_{t+1} en fonction de x_t, $\nabla f(x_t)$ *et de* x_{t-1} pour accélerer une méthode d'optimisation : on parle de *méthode inertielle*. Si la méthode heavy-ball ne marche pas nécessairement sur les fonctions convexes non-quadratiques, il existe des modification simples qui fonctionnent : voir l'accélération de Nesterov.

2. Descente de gradient pour la régression ridge

Implémentons l'algorithme de la descente de gradient avec un pas constant γ sur un problème simple : celui de la régression ridge. On rappelle qu'il s'agit du problème de minimisation du risque quadratique régularisé :

$$\min_{w \in \mathbb{R}^p} \frac{1}{2n} \|y - Xw\|_2^2 + \frac{\lambda}{2} \|w\|_2^2 \,.$$

On se place dans le régime $p > n$.

6) Montrer que ce problème de minimisation est la minimisation d'une quadratique de la forme (1). Quels sont les paramètres H, b, L et μ ?

7) Rappeler l'expression de l'estimateur de la régression ridge (annulez le gradient matriciel comme dans le premier cours).

8) Commencer par générer de manière aléatoire une matrice de design $X \in \mathbb{R}^{n \times p}$ de taille $n = 50, p = 60$ dont les entrées sont des gaussiennes standard i.i.d. et un vecteur de réponses y composé aussi de gaussiennes standard i.i.d..

9) Fixer une valeur arbitraire pour λ, par exemple $\lambda = 1$. Calculer numériquement μ et L. Représentez sous la forme d'un histogramme la distribution des valeurs propres de H.

10) On va maintenant illustrer la convergence d'une méthode de descente de gradient à pas constant vers cet optimum. Implémenter la méthode de descente de gradient simple pour trouver le minimum et le vecteur réalisant ce minimum. Représenter graphiquement la vitesse de convergence.

Remarques : - Les vitesses de convergence des algorithmes de se représentent généralement sur un graphe logarithmique, il faut donc penser à utiliser les fonctions `semilogy`, `loglog` de `matplotlib.pyplot`.
- En pratique, il est souvent difficile de calculer les paramètres μ et L. On ne peut donc pas suivre de recommandation théorique pour le choix de γ. Dans ce cas, γ devient un hyper-paramètre qu'on ajuste "à la main".

11) Implementer la méthode heavy-ball et représenter graphiquement la vitesse de convergence.

12) Que se passe-t-il computationnellement si le paramètre de régularisation λ tend vers 0 ?

Ce TP montre qu'en optimisation convexe, la hessienne de la fonction à minimiser, et plus précisément son conditionnement L/μ, est un paramètre essentiel. En machine learning, ces hessiennes sont aléatoires car elles dépendent des données, et sont de grande dimension. Il existe alors des théorèmes limites qui donnent des propriétés que ces matrices satisfont avec grande probabilité : c'est l'objet du domaine des *matrices aléatoire* de décrire ces propriétés. Ces propriétés peuvent alors être exploitées par les algorithmes d'optimisation pour le machine learning.