TD 5 : Gradient stochastique

Optimisation pour l'apprentissage automatique, M2 Big Data

7 décembre 2021

Dauphine | PSL TUNIS

Exercice 1 : Perte de Huber

On considère un jeu de données $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, où $n \geq 1$, $\boldsymbol{x}_i \in \mathbb{R}^d$ avec $d \geq 1$ et $y_i \in \mathbb{R}$. On cherche un modèle linéaire qui prédise au mieux chaque y_i à partir du $\boldsymbol{x}_i$ correspondant. On définit donc une famille de modèles paramétrée par $\boldsymbol{w} \in \mathbb{R}^d$ comme suit :

$$\begin{array}{rccl} h_{\boldsymbol{w}} : & \mathbb{R}^d & \to & \mathbb{R} \\ & \boldsymbol{x} & \mapsto & \boldsymbol{x}^{\mathrm{T}}\boldsymbol{w} = \sum_{i=1}^d [\boldsymbol{x}]_i [\boldsymbol{w}]_i. \end{array}$$

Pour un modèle $h_{\boldsymbol{w}}$, on considèrera que ce modèle prédit parfaitement y_i à partir de $\boldsymbol{x}_i$ si on a $\ell\left(h_{\boldsymbol{w}}(\boldsymbol{x}_i) - y_i\right) = \ell\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i\right) = 0$, où $\ell : \mathbb{R} \to \mathbb{R}$ est la fonction de perte de Huber définie par :

$$\ell(t) = \begin{cases} \frac{1}{2}t^2 & \text{if } |t| < 1 \\ |t| - \frac{1}{2} & \text{otherwise.} \end{cases} \tag{1}$$

Cette fonction se comporte comme $t \mapsto \frac{t^2}{2}$ pour $|t| < 1$ et comme $t \mapsto |t|$ lorsque $|t|$ est très grand. Contrairement à ce que son expression peut suggérer, ℓ est continûment dérivable (ou de classe $\mathcal{C}^1$)

L'expression $\ell\left(h_{\boldsymbol{w}}(\boldsymbol{x}_i) - y_i\right)$ représente l'erreur du modèle en $(\boldsymbol{x}_i, y_i)$, et on cherche un modèle (c'est-à-dire un vecteur $\boldsymbol{w} \in \mathbb{R}^d$) tel que la somme de ces erreurs soit minimale. On considère donc :

$$\min_{\boldsymbol{w} \in \mathbb{R}^d} f(\boldsymbol{w}) := \frac{1}{n} \sum_{i=1}^n \ell(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i). \tag{2}$$

a) Justifier que 0 est un minorant de (2). Est-ce sa valeur minimale ?

b) Le gradient de f en $\boldsymbol{w} \in \mathbb{R}^d$ est donné par

$$\nabla f(\boldsymbol{w}) = \frac{1}{n} \sum_{i=1}^n \ell'(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i)\boldsymbol{x}_i, \tag{3}$$

avec

$$\ell'(t) = \begin{cases} 1 & \text{si} \quad t > 1 \\ t & \text{si} \quad |t| \leq 1 \\ -1 & \text{si} \quad t < -1. \end{cases}$$

Écrire (en pseudo-code) l'itération de descente de gradient avec une taille de pas constante α et en utilisant la formule (3). Que devient cette itération si le point courant est un minimum local ?

c) Une constante de Lipschitz pour ∇f est $L = \frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i\|^2$. Comment utiliser cette constante pour définir la longueur de pas ? Lorsque L est inconnue, donner deux choix possibles pour la taille de pas.

d) La fonction f s'écrit $f = \frac{1}{n}\sum_{i=1}^{n} f_i$, où $f_i(w) = \ell(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i)$. Le gradient f_i en $\boldsymbol{w}$ est

$$\nabla f_i(\boldsymbol{w}) = \ell'(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i)\boldsymbol{x}_i.$$

Écrire (en pseudo-code) l'itération du gradient stochastique pour ce problème sans choix particulier de taille de pas.

e) *On considère ici que notre unité de coût est un accès à un $\boldsymbol{x}_i$.* Quel est le coût d'une itération de descente de gradient, et celui d'une itération de gradient stochastique ?

f) Discuter de l'intérêt du gradient stochastique dans les deux cas suivants :

   i) $n >> 1$ et les échantillons $\{(\boldsymbol{x}_i, y_i)\}$ sont corrélés;

   ii) $n = d$ et les $\boldsymbol{x}_i$ sont les vecteurs coordonnées de $\mathbb{R}^n$.

g) Quand on applique le gradient stochastique avec une longueur de pas fixe, on peut parfois observer que la méthode génère des itérés de norme de plus en plus grande, ce qui conduit à un dépassement de mémoire pour l'algorithme. Fournir une justification à ce phénomène.

h) On considère une variante à lots du gradient stochastique, dans laquelle on choisit un sous-ensemble de n_b composantes dans la somme finie de (2).

   i) Écrire l'itération correspondante (en pseudo-code).

   ii) Si n_b correspond au nombre de processeurs disponibles pour les calculs, quel peut être l'intérêt de choisir n_b comme taille de lot ?

   iii) Quel est le but principal des méthodes à lots par rapport à l'algorithme du gradient stochastique basique ?

   iv) Supposons que l'on utilise plusieurs tailles de lots et que l'on observe une amélioration en termes de convergence quand n_b augmente pour $1 \leq n_b \leq \frac{n}{10}$. Supposons que l'on observe aussi qu'augmenter n_b au-delà de $n/10$ conduise à une dégradation de la performance. Comment expliquer ces observations ?

Exercice 2 : Tirage par importance

On considère un problème en somme finie de la forme

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimiser}}\, f(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{w}), \tag{4}$$

où, pour chaque $i = 1, \dots, n$, la fonction f_i est de classe $\mathcal{C}^1$et son gradient ∇f_i est L_i-lipschitzien. On suppose également que la fonction f est μ-fortement convexe.

On considère l'itération du gradient stochastique, où l'on suppose que l'indice i_k correspondant au gradient stochastique est tiré selon son importance (*importance sampling*), que l'on définit en fonction des quantités $c_i = \frac{nL_i}{\sum_{j=1}^n L_j}$. On a ainsi

$$\forall i \in \{1, \dots, n\}, \qquad \mathbb{P}\left(i_k = i\right) \;=\; \frac{c_i}{\sum_{j=1}^n c_j}. \tag{5}$$

On remplace alors l'itération du gradient stochastique telle que vue en cours par

$$\boldsymbol{w}_{k+1} \leftarrow \boldsymbol{w}_k - \frac{\alpha_k}{c_{i_k}} \nabla f_{i_k}(\boldsymbol{w}_k). \tag{6}$$

a) Montrer que

$$\mathbb{P}\left(i_k = i\right) = \frac{L_i}{\sum_{j=1}^n L_j}.$$

Interpréter alors le concept de tirage par importance en fonction de ce résultat : quelles valeurs de i ont le plus de chances d'être tirées ?

b) Montrer que l'on a $\mathbb{E}_{i_k}\left[\frac{1}{c_{i_k}} \nabla f_{i_k}(\boldsymbol{w}_k)\right] = \nabla f(\boldsymbol{w}_k)$.

c) On peut montrer que ∇f est L-lipschitzien avec $L = \frac{1}{n}\sum_{i=1}^n L_i$. Supposons que l'on se fixe une longueur de pas constante $\alpha_k = \frac{1}{L}$ pour tout k. Pour un même indice i_k, on souhaite comparer l'itération k du gradient stochastique classique à l'itération (6).

i) Montrer que $\frac{\alpha_k}{c_{i_k}} = \frac{1}{L_{i_k}}$.

ii) Quand peut-on alors avoir $\frac{\alpha_k}{c_{i_k}} \geq \alpha_k$? Que cela signifie-t-il sur l'itération (6) ?

Solutions des exercices

Solutions de l'exercice 1

a) La fonction ℓ est positive sur $\mathbb{R}$. Pour tout $\boldsymbol{w} \in \mathbb{R}^d$, on a donc

$$f(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i) \geq \frac{1}{n}\sum_{i=1}^{n} 0 = 0.$$

Cela montre que 0 est un minorant du problème (2). Cette valeur est atteinte uniquement lorsqu'il existe un point $\boldsymbol{w}$ tel que $\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w} - y_i = 0$ pour tout i. Cela n'est pas toujours le cas (prendre par exemple $n = 2, d = 1, \boldsymbol{x}_1 = 1, \boldsymbol{x}_2 = -1, y_1 = y_2 = 1$), donc 0 n'est pas nécessairement la valeur minimale du problème.

b) En un point $\boldsymbol{w}_k \in \mathbb{R}^d$, l'itération de la méthode de descent de gradient avec un pas constant α s'écrit: :

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \frac{\alpha}{n}\sum_{i=1}^{n} \ell'(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}_k - y_i)\boldsymbol{x}_i.$$

Si $\boldsymbol{w}_k$ est un minimum local, on a $\nabla f(\boldsymbol{w}_k) = 0$, et l'itération devient $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k$.

c) Si on connaît une constante de Lipschitz L pour le gradient, on peut alors choisir un pas constant égal à $\alpha = \frac{1}{L}$.
Si on ne connaît pas cette valeur, il est possible d'utiliser un pas décroissant (par exemple $\alpha_k = \frac{1}{k+1}$) ou d'effectuer une recherche linéaire pour calculer un pas approprié à l'itération.

d) En $\boldsymbol{w}_k \in \mathbb{R}^d$, l'itération du gradient stochastique (avec pas α_k) se décompose en deux parties. On tire tout d'abord un indice i_k au hasard dans $\{1, \ldots, n\}$; on calcule ensuite le nouvel itéré $\boldsymbol{w}_{k+1}$ via

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla f_{i_k}(\boldsymbol{w}_k) = \boldsymbol{w}_k - \alpha_k \ell'(\boldsymbol{x}_{i_k}^{\mathrm{T}}\boldsymbol{w}_k - y_{i_k})\boldsymbol{x}_{i_k}.$$

e) Chaque itération de descente de gradient doit accéder à toute la donnée pour calculer le gradient : comme notre unité de coût correspond à un accès à un point $\boldsymbol{x}_i$, le coût d'une itération de descente de gradient est de n. Quant à l'itération du gradient stochastique, elle a un coût de 1, puisqu'elle n'accède qu'à un point du jeu de données ($\boldsymbol{x}_{i_k}$, avec i_k tiré aléatoirement).

i) Lorsque $n >> 1$ et les éléments du jeu de données sont corrélés, on sait qu'une bonne prédiction par rapport à un élément peut aussi donner une bonne prédiction par rapport aux autres éléments corrélés avec le premier. Cela signifie que l'algorithme du gradient stochastique pourra être capable d'améliorer l'objectif tout en ayant un coût bien plus faible que celui de la descente de gradient. Il est donc intéressant d'appliquer l'algorithme dans ce contexte.

ii) Lorsque $n = d$ et $\boldsymbol{x}_i = \boldsymbol{e}_i$ (avec $\boldsymbol{e}_i$ le i-ème vecteur de la base canonique de $\mathbb{R}^n$ défini par $[\boldsymbol{e}_i]_i = 1$ et $[\boldsymbol{e}_i]_j = 0$ pour $i \neq j$), le problème s'écrit :

$$\min_{\boldsymbol{w}\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{e}_i^{\mathrm{T}}\boldsymbol{w} - y_i) = \frac{1}{n}\sum_{i=1}^{n} \ell([\boldsymbol{w}]_i - y_i).$$

On voit ainsi que la fonction objectif est une somme de n termes impliquant chacun une coordonnée différente de $\boldsymbol{w}$. Une itération de descente de gradient met à jour toutes ces coordonnées simultanément, tandis qu'une itération de gradient stochastique ne modifiera qu'une coordonnée choisie au hasard. Par conséquent, il est moins intéressant d'appliquer le gradient stochastique dans ce contexte. (Remarque : ici les éléments du jeu de données ne sont pas corrélés.)

f) L'algorithme du gradient stochastique est aléatoire par nature, car son exécution dépend d'un tirage aléatoire d'une suite d'indices : il se peut donc qu'il ne converge pas (même s'il converge en moyenne), et c'est ce qui se produit ici.

i) L'itération de gradient stochastique avec "batch" de taille n_b en $\boldsymbol{w}_k \in \mathbb{R}^d$ consiste d'abord à tirer aléatoirement un ensemble d'indices $S_k \subset \{1, \dots, n\}$ tel que $|S_k| = n_b$, puis à effectuer l'itération :

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \frac{\alpha_k}{|S_k|} \sum_{i \in S_k} \nabla f_i(\boldsymbol{w}_k),$$

avec $\alpha_k > 0$ la longueur de pas.

ii) Si n_b processeurs sont disponibles et que les gradients des f_i peuvent être calculés en parallèle, alors le coût du "batch" peut être réparti sur ces n_b processeurs.

iii) Ces méthodes utilisent un estimateur du gradient ($\frac{1}{|S_k|} \sum_{i \in S_k} \nabla f_i(\boldsymbol{w}_k)$) dont la variance est plus faible que celle de l'estimateur utilisé par le gradient stochastique ($\nabla f_{i_k}(\boldsymbol{w}_k)$).

iv) Si l'on observe une bonne convergence avec une taille de "batch", cela signifie que les données sont suffisamment corrélées pour qu'il suffise d'en considérer un petit sous-ensemble pour converger. Utiliser plus d'un point à chaque itération permet de réduire la variance des itérations, ce qui explique pourquoi $n_b = n/10$ conduit à une meilleure performance que $n_b = 1$ (qui correspond au gradient stochastique classique). En revanche, lorsque n_b se rapproche de n, le coût de la méthode se rapproche de celui d'une itération de descente de gradient, et dans le même temps la méthode est plus sensible aux redondances dans le jeu de données. Cela explique que la convergence de la méthode se dégrade, ici lorsque $n_b > n/10$.

Solutions de l'exercice 2

a) Il suffit d'utiliser la définition de c_i: on a

$$\mathbb{P}(i_k = i) = \frac{c_i}{\sum_{j=1}^n c_j} = \frac{\frac{nL_i}{\sum_{k=1}^n L_k}}{\sum_{j=1}^n \frac{nL_j}{\sum_{k=1}^n L_k}} = \frac{nL_i}{\sum_{j=1}^n nL_j} = \frac{L_i}{\sum_{j=1}^n L_j}.$$

On voit ainsi que les composantes ayant le plus de chances d'être tirées seront celles qui correspondent aux plus grandes constantes de Lipschitz, c'est-à-dire aux gradients dont la variation est la plus large. Le tirage par importance permet de donner la priorité à ces composantes.

b) Par définition, on a :

$$\begin{aligned}\mathbb{E}_{i_k}\left[\frac{1}{c_{i_k}}\nabla f_{i_k}(\boldsymbol{w}_k)\right] &= \sum_{i=1}^{n}\mathbb{P}\left(i_k=i\right)\frac{1}{c_i}\nabla f_i(\boldsymbol{w}_k)\\ &= \sum_{i=1}^{n}\frac{c_i}{\sum_{j=1}^{n}c_j}\frac{1}{c_i}\nabla f_i(\boldsymbol{w}_k)\\ &= \sum_{i=1}^{n}\frac{1}{\sum_{j=1}^{n}c_j}\nabla f_i(\boldsymbol{w}_k)\\ &= \sum_{i=1}^{n}\frac{1}{n}\nabla f_i(\boldsymbol{w}_k)=\nabla f(\boldsymbol{w}_k),\end{aligned}$$

où l'on a utilisé le fait que $\sum_{j=1}^{n}c_j=\sum_{j=1}^{n}\frac{nL_j}{\sum_{k=1}^{n}L_k}=n\frac{\sum_{j=1}^{n}L_j}{\sum_{k=1}^{n}L_k}=n$.

i) Puisque $\alpha_k=\frac{1}{L}$, on a

$$\frac{\alpha_k}{c_{i_k}}=\frac{1}{L}\frac{\sum_{j=1}^{n}L_j}{nL_{i_k}}=\frac{n}{\sum_{j=1}^{n}L_j}\frac{\sum_{j=1}^{n}L_j}{nL_{i_k}}=\frac{1}{L_{i_k}}.$$

ii) D'après ce qui précède, pour un même tirage d'indice i_k, l'itération du gradient stochastique classique s'écrit

$$\boldsymbol{w}_{k+1}=\boldsymbol{w}_k-\alpha_k\nabla f_{i_k}(\boldsymbol{w}_k)=\boldsymbol{w}_k-\frac{1}{L}\nabla f_{i_k}(\boldsymbol{w}_k),$$

tandis que l'itération (5) s'écrit

$$\boldsymbol{w}_{k+1}=\boldsymbol{w}_k-\frac{\alpha_k}{c_{i_k}}\nabla f_{i_k}(\boldsymbol{w}_k)=\boldsymbol{w}_k-\frac{1}{L_i}\nabla f_{i_k}(\boldsymbol{w}_k).$$

Par conséquent, la seconde itération fera un pas plus petit dans la direction de $-\nabla f_{i_k}(\boldsymbol{w}_k)$ si $L_i\geq\frac{1}{n}\sum_{j=1}^{n}L_j$, c'est-à-dire si la $i$ème constante de Lipschitz est plus grande que la moyenne. C'est précisément ce que le tirage par importance met en avant, et l'on voit que l'itération adapte la longueur de pas en fonction de la constante de Lipschitz, de sorte à réduire l'effet des composantes à constante de Lipschitz importante.