Algorithme de descente de gradient stochastique

Romain Canelle
Lacroix Ewan
2024-2025

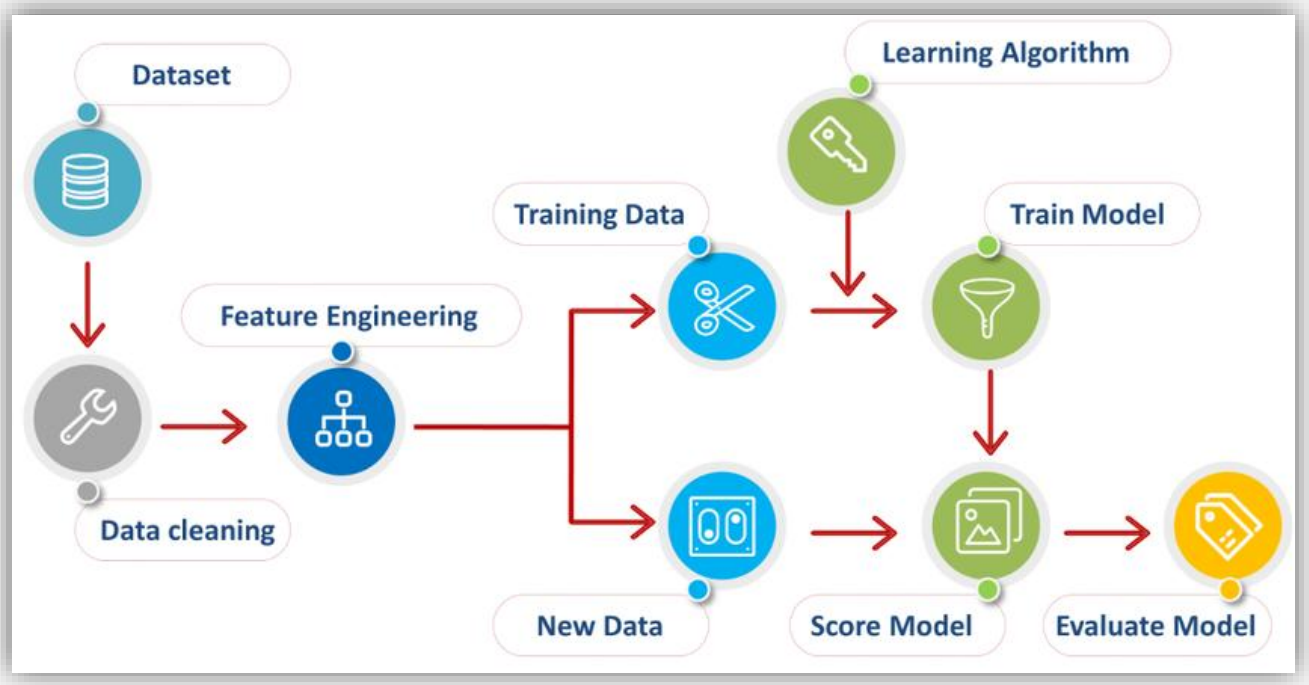
Qu’est ce que la méthode de descente de gradient stochastique (SGD) ?

Le SGD est un Algorithme d’optimisation numérique utile en Machine Learning.

Cet algorithme va chercher à minimiser une fonction Coût que l’on appelle $J(\theta)$. Avec θ les paramètres de notre modèle.

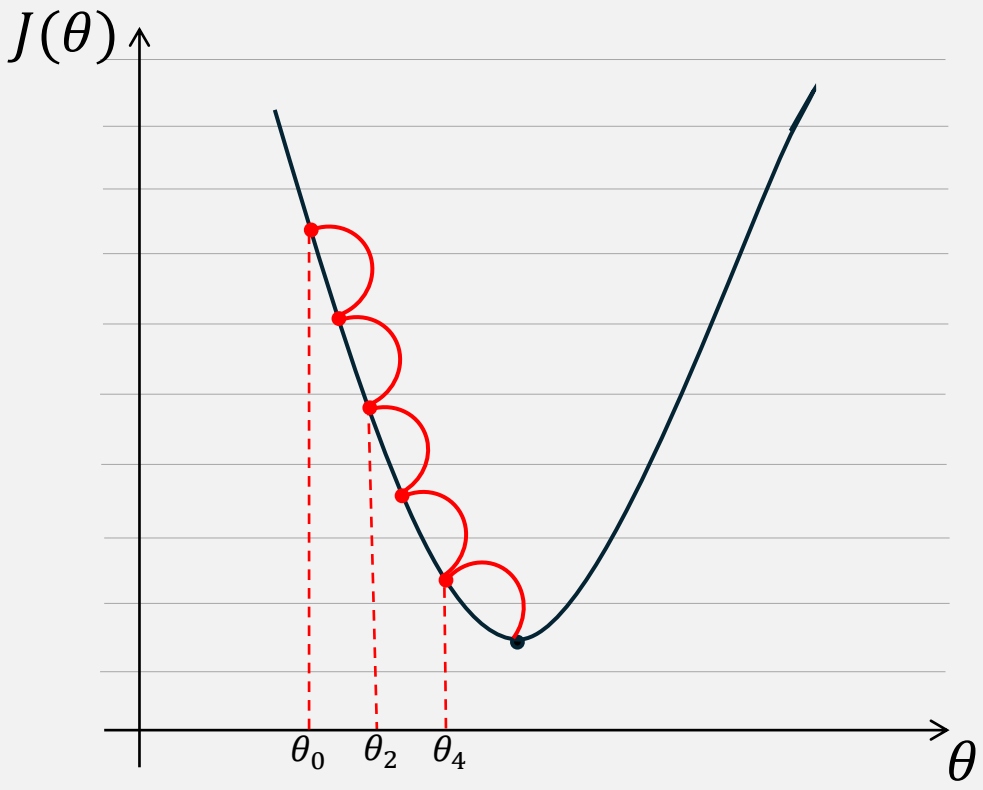
L’objectif de l’optimisation ici, sera d’ajuster les paramètres du modèle spécifié afin de maximiser ses performances et donc minimiser ses erreurs.

Il se base sur le gradient individuel, de la fonction Coût, d’une observation au hasard de l’échantillon. (vitesse de calcul)



https://www.researchgate.net/publication/373826952/figure/fig2/AS:11431281187851205@1694435004303/Fig-2-Machine-learning-process-diagram.ppm

Présentation de l'algorithme (SGD)



$\nabla J(\theta)$ Nous sert de boussole λ L'amplitude du pas

Initialisation :

On fixe une valeur initiale à nos paramètres θ_0, le learning rate λ (vitesse d'apprentissage) et le nombre d'epoch (itération).

Règle de passage :

Pour chaque itération, nous allons opérer la règle suivante : $\boldsymbol{\theta_{i+1} \leftarrow \theta_i - \lambda \nabla(\theta_i, x_j)}$ avec $\nabla(\theta_i, x_j)$ le gradient individuel de la fonction coût évalué avec une observation j tirée au hasard dans l'échantillon.

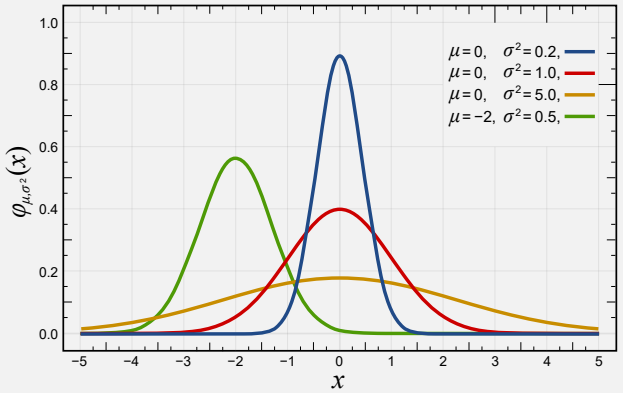
Règle d'arrêt :

L'algorithme va s'arrêter lorsque le nombre d'itération est fini ou bien lorsque la règle d'arrêt est déclenchée. Par exemple lorsqu'entre 2 itérations, les paramètres ne varient que très peu.

Méthode du maximum de vraisemblance

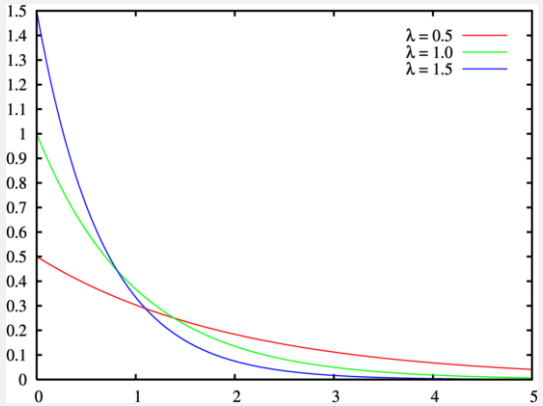
Soit un échantillon $x = (x_1, x_2, x_3, \cdots, x_n)$. L'objectif du maximum de vraisemblance est de trouver les paramètres inconnus de la distribution connue suivie par nos données.

Loi Normale : $\mathcal{N}(\mu, \sigma^2)$



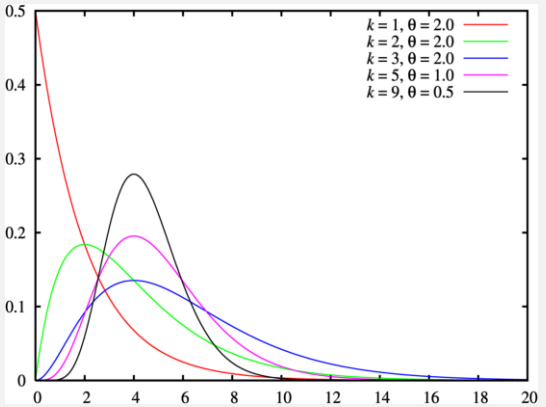
$$\theta = (\mu, \sigma^2)$$

Loi Exponentielle : $\mathcal{E}(\lambda)$



$$\theta = (\lambda)$$

Loi Gamma : $\Gamma(k, \theta)$



$$\theta = (k, \theta)$$

θ est le vecteur des paramètres à estimer propre à chaque loi.

Méthode du maximum de vraisemblance

Fonction de densité

$$f(x,\theta) = P(x_i) = L_i(\theta, x)$$

→ **Vraisemblance (likelihood)**

données iid

Probabilité jointe de l'échantillon

$$P(x_1 \cap x_2 \cap \cdots \cap x_n) = P(x_i) \times P(x_i) \times \cdots \times P(x_i)$$

$$= \prod_{i=1}^{n} f(x_i, \theta)$$

$$= L_n(\theta, x)$$

→ **Fonction de vraisemblance**

Méthode du maximum de vraisemblance

On veut trouver les paramètres qui maximisent la fonction de vraisemblance.

$$L_n(\theta, x) = \prod_{i=1}^{n} f(x_i, \theta)$$

$\hat{\theta}$ est l'estimateur du maximum de vraisemblance.

$$\hat{\theta} = \arg max\ L_n(\theta, x)$$

On note $\ln[L_n(\theta, x)]$ le logarithme de la fonction de vraisemblance.

$$ln[L_n(\theta, x)] = \ln \prod_{i=1}^{n} f(x_i, \theta) = \sum_{i=1}^{n} \ln[f(x_i, \theta)]$$

$$\arg max\ L_n(\theta, x) \quad = \quad \arg max\ \ln[L_n(\theta, x)]$$

Méthode du maximum de vraisemblance
Conditions du maximum de vraisemblance

Gradient

$$g_n(\theta, x) = \frac{\partial ln[L_n(\theta,x)]}{\partial \theta}$$

Condition du premier ordre

Le gradient évalué en $\hat{\theta}$ est nul.

$$g_n(\hat{\theta}, x) = 0$$

Hessienne

$$H_n(\theta, x) = \frac{\partial^2 \ln[L_n(\theta,x)]}{\partial \theta \partial \theta^T}$$

Condition du second ordre

La hessienne évaluée en $\hat{\theta}$ doit être définie négative.

$H_n(\hat{\theta}, x)$ définie négative

On obtient $\hat{\theta}$ l'estimateur du maximum de vraisemblance.

Application analytique

Loi normale

Soit un échantillon de 1000 individus qui suit une loi normale de paramètres μ=20 et σ=3.
On veut retrouver la valeur de ces paramètres grâce à la méthode du maximum de vraisemblance.

$$x = (x_1, x_2, x_3, \cdots, x_n) \sim \mathcal{N}(\mu, \sigma^2)$$

$$f(x, \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

$$L_n(\theta, x) = \frac{1}{(\sigma\sqrt{2\pi})^n} e^{\frac{\sum_{i=1}^{n} -(x_i-\mu)^2}{2\sigma^2}}$$

$$ln[L_n(\theta, x)] = -\frac{n}{2}\ln(\sigma^2) - \frac{n}{2}\ln(2\pi) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(x_i - \mu)^2$$

Application analytique

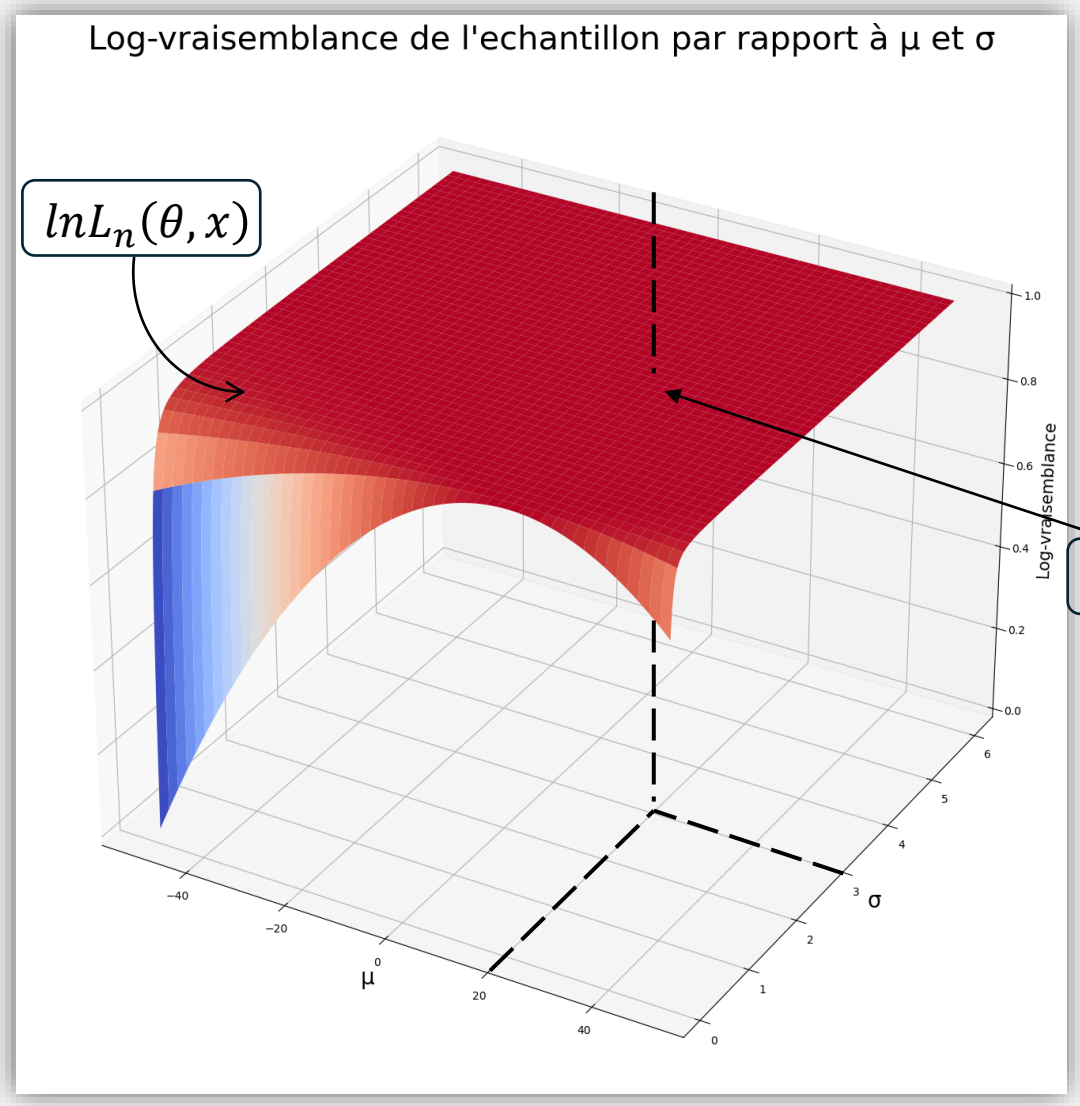
Loi normale

$$g_n(\theta, x) = \begin{pmatrix} \frac{\partial ln L_n(\theta, x)}{\partial \mu} \\ \frac{\partial ln L_n(\theta, x)}{\partial \sigma^2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Longrightarrow \begin{cases} \hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \\ \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu)^2 \end{cases}$$

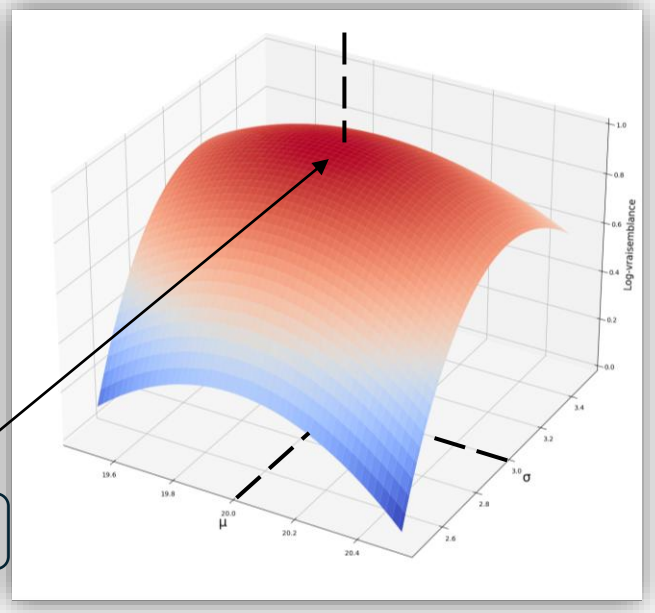
$$H_n(\theta, x) = \begin{pmatrix} \frac{\partial^2 ln L_n(\theta, x)}{\partial \mu \partial \mu} & \frac{\partial^2 ln L_n(\theta, x)}{\partial \mu \partial \sigma^2} \\ \frac{\partial^2 ln L_n(\theta, x)}{\partial \sigma^2 \partial \mu} & \frac{\partial^2 ln L_n(\theta, x)}{\partial \sigma^2 \partial \sigma^2} \end{pmatrix}$$

La matrice hessienne est bien définie négative.

Par résolution analytique, on trouve $\hat{\mu} = 19{,}94$ et $\hat{\sigma} = 3{,}0486$.



$maximum$



Représentation graphique de la fonction de log-vraisemblance de la loi normale avec les données de notre échantillon x .

Cas sans solution analytique

La plupart du temps il n'existe pas de solution analytique.

Dans la pratique, nous sommes confrontés à des distributions et fonctions plus **complexes** :

- Une fonction de vraisemblance **non-linéaire**
- Un trop **grand nombre** de paramètres ou d'observations

Ces situations exigent des méthodes **d'optimisation numérique** sous la forme d'algorithmes tels que :

- L'algorithme de descente de gradient
- L'algorithme de descente de gradient stochastique
- L'algorithme de Newton-Raphson

Application numérique

Algorithme de Newton-Raphson

Initialisation :

On fixe une valeur initiale à nos paramètres θ_0.

Règle de passage :

Pour chaque itération, nous allons opérer la règle suivante :
$\boldsymbol{\theta_{i+1} \leftarrow \theta_i - g_n(\theta_i, x)\, H_n(\theta_i, x)^{-1}}$ avec $g_n(\theta_i, x)$ le gradient individuel et $H_n(\theta_i, x)^{-1}$ l'inverse de la hessienne individuelle.

Règle d'arrêt :

L'algorithme va s'arrêter lorsque le nombre d'itération est fini ou bien lorsque la règle d'arrêt est déclenchée. Par exemple lorsqu'entre 2 itérations, les paramètres ne varient que très peu.

Présentation de l'algorithme (SGD) dans le Cadre du Maximum de vraisemblance

Dans le cadre du maximum de vraisemblance, on cherche à trouver les paramètres qui maximisent notre fonction Coût, étant la Log-vraisemblance.

$$\text{Arg}\,max\ L_n(\theta, x) \iff \text{Arg}\,min\ -L_n(\theta, x)$$

Cela revient donc à minimiser l'opposé de la fonction Coût, passant alors d'une fonction concave à convexe.

On pourrait aussi modifier la règle de passage pour aller dans le sens du gradient plutôt qu'à son opposé.

$$\boldsymbol{\theta_{i+1} \leftarrow \theta_i + \lambda \nabla(\theta_i, x_j)}$$

Contrairement à l'algorithme Newton-Raphson, nous aurons besoin ici seulement du gradient individuel d'une observation j tirée au hasard.

Avantages et inconvénients de la méthode de descente de gradient stochastique

Efficacité de calcul

Avantages

Evaluation du gradient sur une seule observation

→ Rapidité de calcul

→ S'adapte aux bases de données gigantesques

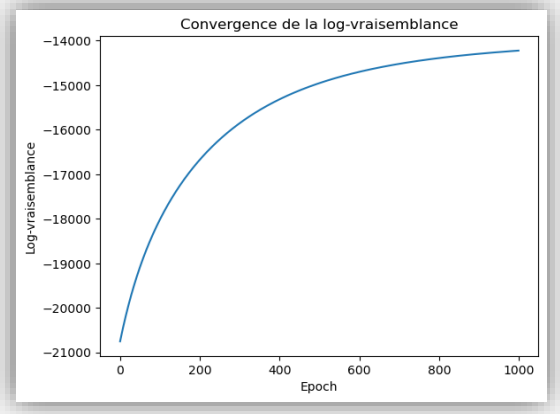
→ S'adapte aux bases avec flux de données continu

→ Faible utilisation de mémoire

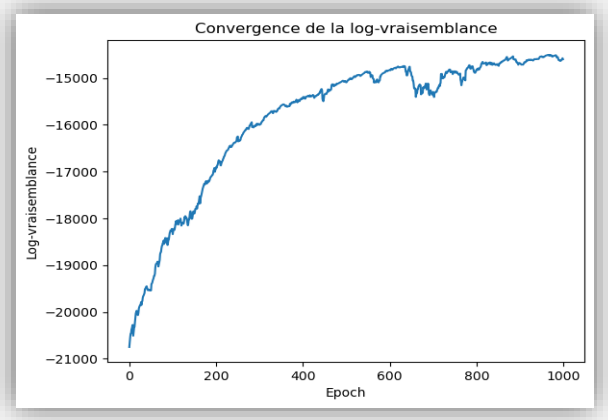
Inconvénients

La trajectoire de la convergence sera bruitée car sensible aux valeurs de l'observation choisie

→ Sensible donc au λ choisi



GD



SGD

Avantages et inconvénients de la méthode de descente de gradient stochastique

Convergence

Avantages

Va converger rapidement vers un optimum approximatif grâce à des données importantes.

Peut échapper à un minima local (stochastique).

Inconvénients

Convergence lente dépendant de la vitesse d'apprentissage et pouvant osciller autour du minimum.

Sensible à la qualité des données, non adapté aux petits échantillons.

Application de la méthode SGD sur régression logistique en MV sur python

Rappels régression logistique

Nous souhaitons modéliser une variable dichotomique Y_i prenant : 1 si défaut, 0 sinon. Soit x_i le vecteur de variables explicatives observables allant de i=1 à N et β le vecteur de paramètres inconnus allant de i=1 à K.

On a $p_i = \Lambda(x_i\beta) = \frac{1}{1+e^{-(x_i\beta)}} = \Pr(y_i = 1 \mid x_i)$

On sait que $L(y_i, \beta) = p_i^{y_i}(1-p_i)^{(1-y_i)}$ $\log L(y, \beta)$ étant notre fonction Coût.

De plus, $\hat{\beta} = \arg\max[\log L(y, \beta)] \iff \frac{\partial L(y,\beta)}{\partial \beta} = 0$

Ce système à k équations n'a pas de solution analytique. On a donc besoin d'un algorithme d'optimisation numérique, le SGD ici.

Ainsi : $\nabla(\hat{\beta}_i, y_i) = \frac{\partial L(y_i,\hat{\beta})}{\partial \hat{\beta}} = [y_i - \Lambda(x_i\hat{\beta})]x_i$ est notre gradient individuel pour l'observation i.

Notre règle de passage sera de la forme : $\beta_{t+1} = \beta_t + \lambda * \nabla(\beta_t, y_i)$

Pistes d'amélioration de la méthode de descente de gradient stochastique

Il est très utile lorsque l'on gère de grosses bases de données.

Gère les cas irrésolvables analytiquement.

Bénéficie de nombreux avantages (vitesse, mémoire, minima local,...).

Il nécessite cependant un arbitrage soigné concernant les paramètres d'initialisation et les données utilisées afin d'échapper aux divers inconvénients précédents. (bruit, oscillation,...)

Pistes d'améliorations:

- La méthode des mini-lots (mini-batch) : on ne tire plus une seule observation mais un lot n. Cela permettrait la réduction de bruit tout en gardant une vitesse de calcul correcte.

- Un taux d'apprentissage dynamique au cours des epoch (graduellement ou bien par déclencheur (changement de signe du gradient)).

- Le Momentum, on conserve une partie des gradients des itérations précédentes pour réduire l'oscillation et accélérer la convergence.

- Gradient adaptatif, Early stopping, NAG, ...

Sources

Elkan, C. (2014). Maximum likelihood, logistic regression, and stochastic gradient training. Working paper, University of California, San Diego.

Econométrie des Variables Qualitatives. Cours C. Hurlin

Statistiques et probabilités en économie-gestion 2ème édition (2022), C. Hurlin, V. Mignon

Advanced Econometrics – Statistique mathématique. Cours C. Hurlin