Optimisation
Algorithme de la descente du gradient
- L2 Math, IMIA -

Fonctions convexes

1. Donner la définition d'une fonction convexe et une condition suffisante pour avoir la convexité.
2. Parmi les fonctions suivantes lesquelles sont convexes ?

$$f(x) = x\sin(x) \quad \text{pour} \quad x \in [0, 2\pi]$$
$$g(x) = -log(x) + x^2 \quad \text{pour} \quad x \in \mathbb{R}^+$$
$$h(x) = x\sqrt{x} \quad \text{pour} \quad x \in \mathbb{R}^+$$
$$\ell(w) = (y - xw)^2 \quad \text{pour} \quad w \in \mathbb{R}$$

Problème de minimisation de l'erreur quadratique

Supposons qu'on dispose d'un ensemble d'observations $\{(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)\}$ avec $\mathbf{x}_i \in \mathbb{R}^p$ et $y_i \in \mathbb{R}$ pour tout i. Un problème de régression consiste à chercher une fonction f telle que, pour tout i, $f(x_i)$ est aussi proche que possible de y_i au sens de l'erreur en moyenne quadratique. Autrement dit, on cherche la fonction f solution du problème de minimisation

$$\min_{f \in \mathcal{F}} \frac{1}{2} \sum_{i=1}^{n} (y_i - f(x_i))^2$$

où $\mathcal{F}$ est une famille de fonction.

Si $\mathcal{F}$ est la famille des fonctions linéaires, le problème se ramène à chercher une constante w_0 et un vecteur de poids $\mathbf{w} \in \mathbb{R}^p$ solution du problème

$$\min_{w_0 \in \mathbb{R}, \mathbf{w} \in \mathbb{R}^p} \frac{1}{2} \sum_{i=1}^{n} (y_i - w_0 - \mathbf{x}_i^T \mathbf{w})^2 \tag{1}$$

où x^T signifie "x transposé" et $\mathbf{x}_i^T \mathbf{w} = \sum_{j=1}^{p} x_{ij} w_j$

Fonction quadratique réelle

Considérons un problème simplifié dans lequel on a une seule observation telle que $x_1 = -2$ et $y_1 = 1$.

3. Ecrire le le problème (1) dans ce cas, en ajoutant l'hypothèse que $w_0 = 0$.
4. Quel est la solution exacte de ce problème ?
5. Ecrire l'algorithme de descente de gradient pour ce problème.

6. Appliquer l'algorithme en partant du point $w_0 = 0$ avec un taux d'apprentissage $\alpha = 0.1$. Combien d'itérations sont-elles nécessaires pour converger si le critère d'arrêt est "la valeur absolue de la dérivée au point courant est inférieure à 0.1" ?
7. Calculer 4 itérations de l'algorithme en partant du point $w_0 = 0$ avec un taux d'apprentissage $\alpha = 1$. Que se passe t'il ?
8. Calculer 4 itérations de l'algorithme en partant du point $w_0 = 0$ avec un taux d'apprentissage $\alpha = 0.001$. Que se passe t'il ?
9. Recherche linéaire
   Considérer un pas constant n'est pas une très bonne solution : le choix du pas est difficile.
   On peut alors, à chaque itération de l'algorithme de descente du gradient chercher un taux d'apprentissage. On propose ici la règle du "back tracking" connue aussi sous le nom de la règle d'Amijo.

   recherche linéaire par back-tracking
   choisir deux constantes $0 \leq a \leq 1/2$ et $0 \leq b \leq 1$.
   $\tau = \alpha$
   tant que $L(w - \tau \nabla L(w)) > L(w) - a * \tau * \|\nabla L(w)\|_2^2$:
   $\tau = b * \tau$
   où ∇L est ici la dérivée de L et $\|\nabla L(w)\|_2^2 = \nabla L(w)^T \nabla L$.

   Ecrire l'algorithme de descente du gradient avec back-tracking.
10. Appliquer l'algorithme de descente du gradient avec back-tracking en partant du point $w_0 = 0$ avec un taux d'apprentissage $\alpha = 1$, $a = b = 1/2$.
    Compter le nombre de fois où on appelle la fonction L ou la fonction L'.
    Comparer le nombre d'appels à celui qu'on a quand on applique l'algorithme de descente gradient sans back-tracking en partant du point $w_0 = 0$ avec un taux d'apprentissage $\alpha = 0.1$.

Fonction quadratique dans $\mathbb{R}^{p+1}$

On rappelle que pour la régression linéaire, le problème de minimisation qui permet d'estimer les coefficients (ou poids) est

$$\min_{(\beta_0,\beta_1)\in\mathbb{R}^2} \frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

11. Vérifier que si on définit

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} \\ \vdots & \vdots \\ 1 & x_{n1} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

    alors

$$\frac{1}{2} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 = \frac{1}{2} (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) = \frac{1}{2} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 = L(\mathbf{w})$$

12. Calculer le gradient de L [1].
    Vous pourrez vérifier que $\nabla L(\mathbf{w}) = \mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w})$

13. Ecrire l'algorithme de descente de gradient dans ce cas.
14. En utilisant les conventions du langage Python, "implémenter" l'algorithme dans une fonction qui prend en entrée le point initiale, la fonction à minimiser, son gradient, le taux d'apprentissage, les constantes du back-tracking, le ϵ du critère d'arrêt et un nombre maximum d'itérations.

    ```
    def grad_desc_bt(w0,L,gL,alpha=.1,a=.5,b=.5,eps=1e-4,maxiter=10)
    ```

    La fonction doit renvoyer, la "solution" obtenue, la valeur de la fonction et de son gradient à la "solution" ainsi que le nombre d'itérations et un booléen indiquant si l'algorithme a convergé.
15. Pourquoi écrit-on "solution" entre guillements ?

1. astuce : aidez vous du TD précédent.