TP1 : Méthode des moindres carrés

La méthode des moindres carrés est utilisée lorsque l'on souhaite ajuster une fonction sur un ensemble de données. Plus précisément, on considère une famille de fonctions $\mathcal{F}$ dont les éléments f_a dépendent d'un paramètre $a \in \mathbb{R}^d$. Par exemple

$$\mathcal{F} = \left\{ f_a : \mathbb{R} \to \mathbb{R}, x \longmapsto a\,x \right\}$$

ensemble des fonctions linéaires, paramétrées par un paramètre réel $a \in \mathbb{R}$. On dispose d'un ensemble de données, dont on suppose qu'il est généré par une fonction de la famille $\mathcal{F}$, c'est-à-dire qu'on dispose d'un ensemble de points $\{(x_i, y_i)\}_{1 \leqslant i \leqslant n}$ tel que

$$\forall\, i = 1 \dots n, \qquad y_i = f_{a_0}(x_i) + \varepsilon_i$$

où $x_i \in \mathbb{R}^p$ et $y_i \in \mathbb{R}^q$, $i = 1 \dots n$ pour un certain paramètre a_0. La présence de ε_i peut être due à l'existence d'un bruit d'acquisition (arrondi numérique, erreur ou imprécision) et/ou à l'approximation de la modélisation. Dans de nombreuses applications, on cherche à trouver la *meilleure* fonction de $\mathcal{F}$ qui modélise la génération des données. Autrement dit, étant donné l'ensemble des données $u = \{(x_i, y_i)\}_{1 \leqslant i \leqslant n}$, quelle est la fonction $f_a \in \mathcal{F}$ pour laquelle le nuage de points $v_{f_a} = \{(x_i, f_a(x_i))\}_{1 \leqslant i \leqslant n}$ se rapproche le plus des données initiales u ? Ces deux ensembles de points appartenant un espace euclidien, on peut mesurer la distance entre ces deux ensembles à l'aide de la distance (appelée dans ce contexte distance de FROBENIUS)

$$\|v_{f_a} - u\|_F^2 = \sum_{i=1}^{n} \|f_a(x_i) - y_i\|_2^2.$$

Le problème d'ajustement peut donc s'écrire comme le problème d'optimisation suivant :

$$\min_{f_a \in \mathcal{F}} \sum_{i=1}^{n} \|f_a(x_i) - y_i\|_2^2.$$

ce qui revient à minimiser l'erreur au carré moyenne entre les données y_i et le modèle théorique $f_a(x_i)$; d'où le nom de méthode des *moindres carrés*. Puisque la famille $\mathcal{F}$ est paramétrée par le vecteur $a \in \mathbb{R}^d$, la forme finale du problème d'ajustement devient

$$\min_{a \in \mathbb{R}^d} \sum_{i=1}^{n} \|f_a(x_i) - y_i\|_2^2. \tag{1}$$

1 Regression linéaire

Prenons l'exemple loi d'OHM, qui stipule qu'il existe une relation linéaire entre l'intensité I (Ampère) du courant électrique qui traverse un conducteur et la tension U (Volt) à ses bornes :

$$U = R\,I.$$

Le rapport R (constant) entre la tension et l'intensité est appelé *résistance* (Ohm). Afin d'estimer la valeur de la résistance d'un conducteur donné, on peut faire passer un courant d'une intensité connue I_0 à travers le conducteur, puis mesurer la tension U_0 associée ; enfin, faire le rapport et en déduire une valeur de résistance. Si l'on réitère l'expérience avec une autre valeur d'intensité I_1 connue, on *devrait* trouver la même valeur de résistance. En pratique, ce n'est pas le cas (et ce, même si $I_0 = I_1$!). De nombreux facteurs expliquent une telle différence : le voltmètre utilisé n'est pas suffisamment précis, l'affichage tronque la valeur de la tension, la loi d'OHM n'est qu'une loi théorique qui *approche* le comportement réel du conducteur... Ainsi, si on fait une série de n mesures de tension U_i associées à n valeurs d'intensité connues I_i, on a seulement

$$\forall\, i = 1 \dots n, \qquad U_i \approx R\,I_i$$

Pour estimer la valeur de la résistance, nous allons procéder à un ajustement linéaire sur les données $\{(I_i, U_i)\}_{1 \leqslant i \leqslant n}$, pour lequel on peut appliquer la méthode des moindres carrés.

Exercice 1. Régression linéaire en dimension 1.

1. Choisir une valeur de $R_{\text{théorique}} \in \mathbb{R}$ (en ohm). Générer $n = 10$ valeurs d'intensité I_i (en ampère), puis générer n valeurs de tension U_i (en volt). Pour simuler l'imprécision des U_i, on ajoutera à chaque valeur théorique $R_{\text{théorique}}\, I_i$ un bruit blanc gaussien de variance σ à choisir.
2. Donner l'expression de la fonction à minimiser selon la méthode des moindres carrés ?
3. Donner l'expression de la résistance R_{opt} optimale selon la méthode des moindres carrés.
4. Comparer numériquement la valeur de $R_{\text{théorique}}$ et R_{opt}.
5. Reprendre les questions précédentes avec un niveau de bruit σ plus important. Qu'en pensez vous ?
6. Reprendre les questions précédentes avec moins de points de données ($n = 5$ par exemple). Qu'en pensez vous ?

2 Régression polynomiale

On suppose maintenant que l'on dispose de n couples de données $(x_i, y_i) \in \mathbb{R}^2$ satisfaisant :

$$\forall i = 1 \dots n, \quad y_i = P(x_i) + \varepsilon_i,$$

avec P un polynôme réel de degré au plus p. Nous rappelons que tout tel polynôme est défini de manière unique par un vecteur $a = (a_k)_{0 \leqslant k \leqslant p} \in \mathbb{R}^{p+1}$:

$$P(X) = \sum_{k=0}^{p} a_k X^k$$

On suppose connaître les couples de données (x_i, y_i) et on cherche un polynome qui approche au mieux ces données.

Exercice 2. Régression polynomiale.

1. Choisir un vecteur $a \in \mathbb{R}^6$. On note P le polynôme associé. Générer $n = 10$ valeurs x_i, puis n valeur $y_i = P(x_i) + \varepsilon_i$ avec $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, pour σ de votre choix.
2. Ecrire le problème d'ajustement au sens des moindres carrés. Le reformuler en introduisant la matrice $X \in \mathcal{M}_{n,p+1}(\mathbb{R})$, $X_{i,k} = x_i^k$.
3. Le problème a-t-il une solution ? Cette solution est elle unique ?
4. Réaliser l'ajustement polynomial pour des degrés entre $p = 0$ et $p = 6$.
5. Afficher sur le même graphique le nuage de points (x_i, y_i), le polynôme P théorique et les 7 ajustements polynomiaux.

Exercice 3.
Proposer une méthode et l'implémenter pour approcher les données suivantes

```
x = np.array([-2.6605, -2.5024, -2.4075, -2.0656, -1.2418, -1.1105, -1.0194,
       -0.8459, -0.8403, -0.4906,  0.7695,  0.8933,  1.0023,  1.5085, 2.3193])
y =np.array([ 97.1414, 80.8250, 72.0406, 46.0602, 12.0776, 9.4146, 7.8925,
   5.6353, 5.5751, 3.0525, -0.3093, -1.3142, -2.4258, -11.0032, -39.9284])
```

On fera varier le degré du polynôme de 1 à 6.

1. Quel est le degré du polynôme qui approche au mieux les données ? Quelle vaut l'erreur définie en (1) ?
2. Comparer votre résultat avec celui obtenu avec la fonction `polyfit` du module `numpy`. Voir l'aide de `numpy.polyfit` pour l'utilisation de cette fonction.