Correction – TD/TP 5
Régression, arrêt optimal et Robbins–Monro

M2 Finance – Université Gustave Eiffel

Exercice 1 – Espérance conditionnelle et choix des fonctions de régression

1) **Prédiction optimale et bruit irréductible.** On a $Y = X^3 + \varepsilon$. Le bruit ε est indépendant de X et centré, donc $\mathbb{E}(\varepsilon \mid X) = \mathbb{E}(\varepsilon) = 0$. Par linéarité,

$$\boxed{m(X) = \mathbb{E}(Y \mid X) = X^3}, \qquad m(x) = x^3.$$

Pour une autre fonction g, écrivons

$$Y - g(X) = \varepsilon + \big(X^3 - g(X)\big).$$

On développe le carré et on prend l'espérance :

$$\mathbb{E}[(Y - g(X))^2] = \mathbb{E}(\varepsilon^2) + \mathbb{E}[(X^3 - g(X))^2] + 2\mathbb{E}[\varepsilon(X^3 - g(X))].$$

Le terme croisé est nul. En effet, $X^3 - g(X)$ ne dépend que de X, et

$$\mathbb{E}[\varepsilon(X^3 - g(X))] = \mathbb{E}\big[(X^3 - g(X))\mathbb{E}(\varepsilon \mid X)\big] = 0.$$

Comme $\mathbb{E}(\varepsilon^2) = \mathrm{Var}(\varepsilon) = 1/4$,

$$\boxed{\mathbb{E}[(Y - g(X))^2] = \frac{1}{4} + \mathbb{E}[(X^3 - g(X))^2] \geq \frac{1}{4}.}$$

L'égalité a lieu en choisissant $g(x) = x^3$. Ainsi, même en connaissant parfaitement X, on ne peut supprimer le bruit indépendant ε : l'erreur résiduelle minimale est $1/4$.

2) **Projection sur les fonctions affines.** On cherche les coefficients a, b minimisant

$$Q(a, b) = \mathbb{E}[(Y - a - bX)^2].$$

Cette fonction est quadratique et convexe en (a, b). On peut dériver sous l'espérance, puisque toutes les variables ici sont de carré intégrable. Les dérivées premières sont

$$\frac{\partial Q}{\partial a} = -2\mathbb{E}(Y - a - bX), \qquad \frac{\partial Q}{\partial b} = -2\mathbb{E}[X(Y - a - bX)].$$

Les **équations normales** $\partial_a Q = \partial_b Q = 0$ s'écrivent donc

$$\boxed{\mathbb{E}(Y - a - bX) = 0, \qquad \mathbb{E}[X(Y - a - bX)] = 0.}$$

Elles disent que le résidu de la prédiction est orthogonal aux fonctions 1 et X. La loi de X étant symétrique, $\mathbb{E}(X) = \mathbb{E}(X^3) = 0$, et donc $\mathbb{E}(Y) = 0$. La première équation devient $a = 0$. Pour la seconde,

$$\mathbb{E}(XY) = \mathbb{E}(X^4) + \mathbb{E}(X\varepsilon) = \frac{1}{5}, \qquad \mathbb{E}(X^2) = \frac{1}{3}.$$

L'indépendance et le centrage donnent bien $\mathbb{E}(X\varepsilon) = 0$. Dès lors,

$$\frac{1}{5} - b\frac{1}{3} = 0 \quad \Longrightarrow \quad \boxed{b = \frac{3}{5}}, \qquad \boxed{g_{\mathrm{aff}}(x) = \frac{3}{5}x}.$$

3) **Erreur quadratique du prédicteur affine.** D'après la question 1,

$$\mathbb{E}[(Y - g_{\text{aff}}(X))^2] = \frac{1}{4} + \mathbb{E}\left[\left(X^3 - \frac{3}{5}X\right)^2\right].$$

En développant :

$$\begin{aligned}\mathbb{E}\left[\left(X^3 - \frac{3}{5}X\right)^2\right] &= \mathbb{E}(X^6) - \frac{6}{5}\mathbb{E}(X^4) + \frac{9}{25}\mathbb{E}(X^2) \\ &= \frac{1}{7} - \frac{6}{5} \times \frac{1}{5} + \frac{9}{25} \times \frac{1}{3} \\ &= \frac{1}{7} - \frac{6}{25} + \frac{3}{25} = \frac{1}{7} - \frac{3}{25} = \frac{4}{175}.\end{aligned}$$

Ainsi,

$$\boxed{\mathbb{E}[(Y - g_{\text{aff}}(X))^2] = \underbrace{\frac{1}{4}}_{\text{bruit irréductible}} + \underbrace{\frac{4}{175}}_{\text{erreur d'approximation}}}.$$

Cette décomposition montre précisément ce qui pourrait être amélioré par un meilleur choix de fonctions de régression.

4) **Pourquoi x^2 n'améliore-t-il pas la prédiction ?** Le nouvel espace est engendré par $1, x, x^2$. Comme X est symétrique, toute fonction *paire* de X est orthogonale à toute fonction *impaire* intégrable : le produit est impair et son espérance est nulle. Or x^3 est impair, tandis que 1 et x^2 sont pairs. La projection de x^3 sur les fonctions paires $1, x^2$ est donc nulle. Il ne reste que la projection sur x, déjà calculée, ce qui donne

$$\boxed{a = 0, \quad b = \frac{3}{5}, \quad c = 0.}$$

L'erreur reste égale à $1/4 + 4/175$. En revanche, si on ajoute x^3 aux fonctions autorisées, $m(x) = x^3$ appartient exactement à l'espace d'approximation : l'erreur d'approximation devient nulle, et l'erreur minimale est $1/4$.

Idée à retenir

Une famille plus grande n'est pas toujours meilleure. Ce qui compte est l'adéquation entre les fonctions proposées et la dépendance réelle de la quantité à prédire. La projection sert à quantifier cette adéquation.

Exercice 2 – Arrêt optimal sur une chaîne à deux états

1) **Valeur à la date finale.** À la date 2, l'arrêt est obligatoire. Par définition de V_2,

$$\boxed{V_2(A) = h_2(A) = 8, \qquad V_2(B) = h_2(B) = 1.}$$

Il n'y a pas de valeur de continuation, puisqu'aucune date ultérieure n'est disponible.

2) **Remontée à la date 1.** Si l'on se trouve en A à la date 1, le prochain état vaut A avec probabilité $3/4$ et B avec probabilité $1/4$. Le gain futur optimal moyen, vu de la date 1, doit être actualisé d'un pas :

$$\begin{aligned}C_1(A) &= \delta\left(\frac{3}{4}V_2(A) + \frac{1}{4}V_2(B)\right) \\ &= 0{,}9\left(\frac{3}{4} \times 8 + \frac{1}{4} \times 1\right) = 0{,}9 \times 6{,}25 = \boxed{5{,}625}.\end{aligned}$$

L'arrêt immédiat donne $h_1(A) = 6$. Comme $6 > 5{,}625$, on **s'arrête** en A, d'où $V_1(A) = 6$. En partant de B, les deux prochains états ont probabilité $1/2$:

$$C_1(B) = 0{,}9\left(\frac{1}{2} \times 8 + \frac{1}{2} \times 1\right) = 0{,}9 \times 4{,}5 = \boxed{4{,}05}.$$

L'arrêt immédiat ne donne que $h_1(B) = 3$. Comme $3 < 4{,}05$, on **continue** en B, d'où $V_1(B) = 4{,}05$.

$$\boxed{V_1(A) = 6, \qquad V_1(B) = 4{,}05.}$$

Noter que V_1 est exprimé en unités de la date 1 : on n'a pas encore ramené ces gains à la date 0.

3) **Remontée à la date 0.** On ne remplace pas V_1 par les seuls gains h_1 : V_1 tient déjà compte du meilleur choix entre arrêter et continuer à la date 1. C'est le point essentiel de la programmation dynamique. En partant de A,

$$\begin{aligned} C_0(A) &= 0{,}9\left(\frac{3}{4}V_1(A) + \frac{1}{4}V_1(B)\right) \\ &= 0{,}9\left(\frac{3}{4} \times 6 + \frac{1}{4} \times 4{,}05\right) = 0{,}9 \times 5{,}5125 = \boxed{4{,}96125}. \end{aligned}$$

L'arrêt immédiat rapporterait $h_0(A) = 4$. Il vaut donc mieux **continuer**, et $V_0(A) = 4{,}96125$. De même,

$$C_0(B) = 0{,}9\left(\frac{1}{2} \times 6 + \frac{1}{2} \times 4{,}05\right) = 0{,}9 \times 5{,}025 = \boxed{4{,}5225}.$$

Mais $h_0(B) = 5 > 4{,}5225$: en B, il faut **s'arrêter immédiatement**. Finalement,

$$\boxed{V_0(A) = 4{,}96125, \qquad V_0(B) = 5.}$$

La règle optimale est donc : en date 0, arrêter en B et continuer en A ; en date 1, arrêter en A et continuer en B ; en date 2, arrêter dans tous les cas.

4) **Comparer aux dates d'arrêt fixes (avec $X_0 = A$).** Si $\tau = 0$, on reçoit immédiatement $h_0(A) = \boxed{4}$. Si $\tau = 1$, on ignore la possibilité de décider selon l'état à la date 1 ; le gain moyen ramené en 0 est

$$\mathbb{E}(\delta h_1(X_1) \mid X_0 = A) = 0{,}9\left(\frac{3}{4} \times 6 + \frac{1}{4} \times 3\right) = 0{,}9 \times 5{,}25 = \boxed{4{,}725}.$$

Pour $\tau = 2$, il faut d'abord calculer la loi de X_2 à partir de $X_0 = A$. En deux transitions,

$$\mathbb{P}(X_2 = A \mid X_0 = A) = \frac{3}{4} \times \frac{3}{4} + \frac{1}{4} \times \frac{1}{2} = \frac{11}{16}, \quad \mathbb{P}(X_2 = B \mid X_0 = A) = \frac{5}{16}.$$

On actualise maintenant de **deux** périodes :

$$\mathbb{E}(\delta^2 h_2(X_2) \mid X_0 = A) = 0{,}9^2\left(\frac{11}{16} \times 8 + \frac{5}{16} \times 1\right) = \boxed{4{,}708125}.$$

Ainsi,

$$\boxed{V_0(A) = 4{,}96125 > \max(4; 4{,}725; 4{,}708125) = 4{,}725.}$$

La stratégie optimale fait mieux car elle **utilise l'état observé à la date** 1 pour décider : on n'impose pas la même date d'arrêt à toutes les trajectoires.

Idée à retenir

Le calcul à rebours ne consiste pas à optimiser séparément chaque date : le terme V_{k+1} contient déjà la meilleure stratégie des dates ultérieures. C'est cette information qui remonte vers k.

Exercice 3 – Une régression Longstaff–Schwartz à la main

1) **Régression quadratique : d'où viennent les moyennes par groupe ?** On cherche les coefficients a, b, c qui minimisent sur les six observations

$$S(a,b,c) = \sum_{i=1}^{6} \big(H_2^{(i)} - a - bX_1^{(i)} - c(X_1^{(i)})^2\big)^2.$$

Comme X_1 ne prend que trois valeurs distinctes $0, 1, 2$, un polynôme quadratique peut prendre **trois valeurs arbitraires** sur ces trois points : on peut donc choisir indépendamment

$$v_0 = \widehat{c}_1(0), \quad v_1 = \widehat{c}_1(1), \quad v_2 = \widehat{c}_1(2).$$

La somme des carrés devient

$$S = (2-v_0)^2 + (4-v_0)^2 + (6-v_1)^2 + (8-v_1)^2 + (3-v_2)^2 + (5-v_2)^2.$$

Chaque paire est minimisée lorsque v_x est la moyenne des deux observations :

$$\boxed{v_0 = 3, \qquad v_1 = 7, \qquad v_2 = 4.}$$

Pour récupérer les coefficients,

$$a = 3, \qquad a + b + c = 7, \qquad a + 2b + 4c = 4.$$

Les deux dernières équations donnent $b + c = 4$ et $2b + 4c = 1$, d'où $c = -7/2$ puis $b = 15/2$. Par conséquent,

$$\boxed{\widehat{c}_1^{\,\mathrm{quad}}(x) = 3 + \frac{15}{2}x - \frac{7}{2}x^2.}$$

Il s'agit d'une approximation de continuation calculée sur un échantillon, non d'une expression théorique de l'espérance conditionnelle dans tous les modèles possibles.

2) **Règle d'arrêt et gains sur l'échantillon d'apprentissage.** On compare $\widehat{c}_1(x)$ au gain immédiat constant 5 :

x	0	1	2
$\widehat{c}_1^{\,\mathrm{quad}}(x)$	3	7	4
choix	s'arrêter	continuer	s'arrêter

Sur les trajectoires $i = 1, 2$ (état 0), on reçoit 5 à la date 1 ; sur $i = 3, 4$ (état 1), on continue et reçoit 6, 8 ; sur $i = 5, 6$ (état 2), on s'arrête et reçoit 5. Le vecteur des gains réalisés *sur les six trajectoires d'apprentissage* est donc

$$(5, 5, 6, 8, 5, 5), \qquad \boxed{\overline{G}^{\mathrm{quad}}_{\mathrm{apprentissage}} = \frac{34}{6} = \frac{17}{3} \simeq 5{,}667.}$$

3) **Régression affine et changement de politique.** Pour $g(x) = a + bx$, on peut remplacer chaque paire de données par sa moyenne, car les deux observations de chaque groupe ont le même x et le même poids. On ajuste donc la droite aux trois points $(0, 3)$, $(1, 7)$, $(2, 4)$. La moyenne des abscisses est $\overline{x} = 1$, celle des ordonnées est $\overline{y} = (3 + 7 + 4)/3 = 14/3$. La pente des moindres carrés est

$$\begin{aligned} b &= \frac{\sum_{x=0}^{2}(x-1)(\overline{H}_2(x) - 14/3)}{\sum_{x=0}^{2}(x-1)^2} \\ &= \frac{(-1)(3 - 14/3) + (1)(4 - 14/3)}{1+1} = \frac{1}{2}. \end{aligned}$$

L'ordonnée à l'origine est $a = \overline{y} - b\overline{x} = 14/3 - 1/2 = 25/6$. Ainsi,

$$\boxed{\widehat{c}_1^{\text{aff}}(x) = \frac{25}{6} + \frac{x}{2}.}$$

Les valeurs prévues en $0, 1, 2$ sont respectivement $25/6 \simeq 4{,}167$, $14/3 \simeq 4{,}667$ et $31/6 \simeq 5{,}167$. La règle affine **s'arrête en** 0 **ou** 1 et **continue en** 2 : elle diffère sensiblement de la règle quadratique. Les gains sur les six trajectoires deviennent $(5, 5, 5, 5, 3, 5)$, de moyenne

$$\boxed{\overline{G}^{\text{aff}}_{\text{apprentissage}} = \frac{28}{6} = \frac{14}{3} \simeq 4{,}667.}$$

Ici, la continuation a une forme en *bosse* : la famille affine ne peut pas représenter cette variation, tandis que la quadratique le peut.

4) **Pourquoi tester la règle ailleurs ?** Les mêmes six réalisations ont servi à ajuster les coefficients, à choisir la règle, et à calculer les deux moyennes précédentes. Les résultats ne constituent donc pas des estimations indépendantes de la performance future. Une famille trop riche pourrait s'adapter au bruit de cet échantillon. Il faut **figer les coefficients** obtenus, simuler de **nouvelles trajectoires indépendantes**, appliquer la règle en n'utilisant à chaque date que l'état courant, et calculer la moyenne des gains test. Si l'on compare plusieurs familles, on peut consacrer un échantillon de validation au choix de la famille et réserver un dernier échantillon test à l'évaluation finale.

Exercice 4 – Robbins–Monro pour une équation non linéaire

1) **Calcul exact de la fonction dont on cherche un zéro.** La densité de $U \sim \mathcal{U}([0, 2])$ vaut $1/2$ sur $[0, 2]$. Pour $0 \le \theta \le 2$,

$$(\theta - u)_+ = \begin{cases} \theta - u, & 0 \le u < \theta, \\ 0, & \theta \le u \le 2. \end{cases}$$

Par conséquent,

$$F(\theta) = \frac{1}{2}\int_0^\theta (\theta - u)\,\mathrm{d}u = \frac{1}{2}\left[\theta u - \frac{u^2}{2}\right]_0^\theta = \boxed{\frac{\theta^2}{4}}.$$

Sur $[0, 2]$, la fonction est croissante, avec $F'(\theta) = \theta/2 \ge 0$ (strictement croissante sur cet intervalle). L'équation $F(\theta) = 1/4$ donne $\theta^2 = 1$, donc l'unique solution dans $[0, 2]$ est

$$\boxed{\theta_* = 1}.$$

Cette expression n'a pas besoin d'être connue pour exécuter Robbins–Monro : nous l'avons calculée pour vérifier la méthode et contrôler numériquement sa convergence.

2) **Récurrence et absence de biais conditionnel.** À l'étape $n + 1$, θ_n est calculé à partir des observations $U_1, \ldots, U_n$. On tire un nouveau U_{n+1} indépendant de ces observations et on mesure

$$Y_{n+1} = (\theta_n - U_{n+1})_+.$$

Conditionnellement à $\mathcal{F}_n = \sigma(U_1, \ldots, U_n)$, la valeur θ_n est fixée et U_{n+1} reste uniforme indépendante. Ainsi,

$$\mathbb{E}(Y_{n+1} \mid \mathcal{F}_n) = F(\theta_n).$$

L'observation $Y_{n+1} - 1/4$ est donc un estimateur conditionnellement sans biais de $F(\theta_n) - 1/4$. Si $\Pi_{[0,2]}(t) = \min(2, \max(0, t))$, la mise à jour de Robbins–Monro projetée est

$$\boxed{\theta_{n+1} = \Pi_{[0,2]}\left(\theta_n - \frac{0{,}8}{n+1}\left[(\theta_n - U_{n+1})_+ - \frac{1}{4}\right]\right).}$$

Le signe $-$ est naturel : lorsque l'observation dépasse $1/4$, on diminue θ pour diminuer $F(\theta)$, qui est croissante ; sinon on l'augmente.

3) **Trois itérations explicites.** On part de $\theta_0 = 1{,}5$. À la première étape, $\gamma_1 = 0{,}8$, $U_1 = 0{,}5$ et

$$Y_1 = (1{,}5 - 0{,}5)_+ = 1, \quad \theta_1 = 1{,}5 - 0{,}8(1 - 0{,}25) = \boxed{0{,}9}.$$

À la deuxième, $\gamma_2 = 0{,}8/2 = 0{,}4$, $U_2 = 1{,}6$:

$$Y_2 = (0{,}9 - 1{,}6)_+ = 0, \quad \theta_2 = 0{,}9 - 0{,}4(0 - 0{,}25) = \boxed{1}.$$

À la troisième, $\gamma_3 = 0{,}8/3$, $U_3 = 0{,}2$:

$$Y_3 = (1 - 0{,}2)_+ = 0{,}8, \quad \theta_3 = 1 - \frac{0{,}8}{3}(0{,}8 - 0{,}25) = 1 - \frac{0{,}44}{3} = \boxed{0{,}853\overline{3}}.$$

Aucune des trois étapes ne sort de $[0, 2]$; la projection ne change donc pas ces valeurs. La dernière correction s'éloigne de 1 parce qu'elle repose sur **une seule réalisation bruitée**, et non sur la fonction moyenne F.

4) **Pourquoi les deux conditions sur les pas ?** Pour $\gamma_n = n^{-\alpha}$, on utilise le critère des séries de Riemann : $\sum n^{-\alpha}$ diverge si $\alpha \leq 1$, tandis que $\sum n^{-2\alpha}$ converge si $2\alpha > 1$, soit $\alpha > 1/2$. Il faut donc

$$\boxed{\frac{1}{2} < \alpha \leq 1.}$$

γ_n	$\sum \gamma_n = +\infty$	$\sum \gamma_n^2 < +\infty$	Retenir ?
$n^{-0{,}5}$	oui	non	non
$n^{-0{,}7}$	oui	oui	**oui**
n^{-1}	oui	oui	**oui**
$n^{-1{,}2}$	non	oui	non

La divergence de $\sum \gamma_n$ permet que l'effort de correction ne s'épuise pas avant d'atteindre la solution. La convergence de $\sum \gamma_n^2$ amortit la variance du bruit accumulé : pour des perturbations indépendantes centrées de variance σ^2, la somme $\sum_{j=1}^{n} \gamma_j \xi_j$ a variance $\sigma^2 \sum_{j=1}^{n} \gamma_j^2$. Ces conditions se complètent, mais ne suffisent pas à elles seules à garantir la convergence de n'importe quel algorithme : il faut aussi des hypothèses sur F et sur le bruit.

Exercice 5 – TP Python : apprendre et tester une politique d'arrêt

1) **Code complet et lecture des variables.** Dans la fonction d'apprentissage, `stop_time[i]` est la date *provisoire* d'exercice pour la trajectoire i, en supposant déjà construite la politique des dates ultérieures. La variable `gain[i]` contient le montant *non actualisé* associé à cette date. Vu de la date k, le gain futur devient

$$y_i = \delta^{\texttt{stop_time[i]} - k}\, \texttt{gain[i]}.$$

On ajuste $\widehat{c}_k$ aux couples $(X_k^{(i)}, y_i)$ puis on compare ce montant moyen conditionnel estimé au gain immédiat. Voici une implémentation complète :

```
import numpy as np

M, delta = 6, 0.96
rng_train = np.random.default_rng(2026)
rng_test = np.random.default_rng(2027)
```

```
def paths(n, rng):
    x = np.zeros((n, M+1))
    for k in range(M):
        x[:, k+1] = 0.6*x[:, k] + 0.8*rng.normal(size=n)
    return x

def reward(x):
    return np.maximum(1.8-(x-0.8)**2, 0.)

def phi(x, degree):
    return np.column_stack([x**j for j in range(degree+1)])

train = paths(8000, rng_train)
test = paths(30000, rng_test)

def fit_policy(sample, degree):
    n = len(sample)
    gain = reward(sample[:, M]).copy()
    stop_time = np.full(n, M, dtype=int)
    coeffs = {}
    for k in range(M-1, 0, -1):
        x = sample[:, k]
        y = delta**(stop_time-k) * gain
        A = phi(x, degree)
        beta = np.linalg.lstsq(A, y, rcond=None)[0]
        c = A @ beta
        stop = reward(x) >= c
        gain[stop] = reward(x[stop])
        stop_time[stop] = k
        coeffs[k] = beta
    return coeffs

def evaluate(sample, coeffs, degree):
    n = len(sample)
    stopped = np.zeros(n, dtype=bool)
    payoff0 = np.zeros(n)
    for k in range(1, M):
        c = phi(sample[:, k], degree) @ coeffs[k]
        stop = (~stopped) & (reward(sample[:, k]) >= c)
        payoff0[stop] = delta**k * reward(sample[stop, k])
        stopped[stop] = True
    payoff0[~stopped] = delta**M * reward(sample[~stopped, M])
    return payoff0.mean()
```

Suite de la question 1 : pourquoi les deux boucles n'ont-elles pas le même sens ? La boucle de `fit_policy` va de $M-1$ à 1 car elle doit connaître, au moment d'étudier la date k, la stratégie déjà retenue pour les dates $k+1, \dots, M$. La régression au temps k utilise le gain que cette stratégie future produirait sur chaque trajectoire simulée. Ce calcul est possible en remontant depuis la date finale. À l'inverse, `evaluate` simule l'utilisation *réelle* de la règle. Sur une nouvelle trajectoire, on voit X_1, puis éventuellement X_2, etc. Dès qu'on décide de s'arrêter, on enregistre le gain $\delta^k h(X_k)$ et on ne prend plus de décision sur cette trajectoire. Les trajectoires encore actives à la date M reçoivent obligatoirement le payoff terminal.

2) **Comparaison numérique sur les données fixées de l'énoncé.** Le programme suivant permet d'obtenir les résultats :

```
for degree in (1, 2, 3):
    coeffs = fit_policy(train, degree)
    value = evaluate(test, coeffs, degree)
    print(f"degre {degree}: {value:.6f}")

always_1 = delta * reward(test[:, 1]).mean()
always_M = delta**M * reward(test[:, M]).mean()
```

```
print(f"toujours date 1: {always_1:.6f}")
print(f"toujours date 6: {always_M:.6f}")
```

Avec les graines fournies, on obtient, à l'arrondi près :

Politique	Gain test moyen en date 0
Régression de degré 1	1,337710
Régression de degré 2	1,337347
Régression de degré 3	1,338519
Toujours s'arrêter à la date 1	0,900533
Toujours attendre la date 6	0,679649

La politique construite par régression fait nettement mieux que les deux dates fixes choisies comme références. En revanche, les différences entre les trois degrés sont **très petites** au regard de l'échelle globale des gains. Il ne faut pas interpréter le léger avantage du degré 3 sur cet échantillon comme une supériorité générale.

3) **Pourquoi séparer apprentissage et évaluation ?** La régression a été calculée en utilisant les trajectoires `train`. Ces trajectoires influencent donc les coefficients et les dates d'arrêt décidées. Réutiliser les mêmes trajectoires pour juger la politique peut produire une estimation trop favorable, particulièrement quand la famille de régression est riche. Avec `test`, les coefficients sont déjà figés, et les gains proviennent de trajectoires indépendantes. Conditionnellement aux coefficients appris, ces gains de test sont i.i.d., ce qui autorise une moyenne Monte-Carlo classique et son erreur standard empirique. Pour une comparaison systématique des degrés, l'idéal est de choisir le degré sur un échantillon de *validation*, puis d'évaluer le choix final sur un autre échantillon *test* resté intact.

Idée à retenir

Le rôle de la base de régression est de produire une bonne **règle de décision**, pas seulement un bon ajustement statistique. Une base plus grande peut parfois améliorer la continuation, mais elle augmente aussi l'instabilité possible sur un échantillon fini.

Exercice 6 – TP Python : l'effet des pas de Robbins–Monro

1) **Code corrigé et trois choix de pas.** À l'étape k, l'observation bruitée est $y = (\theta - U_k)_+$. Pour chercher $F(\theta) = 1/4$ dans $[0, 2]$, on applique

$$\theta_k = \Pi_{[0,2]}\left(\theta_{k-1} - \gamma_k \left[(\theta_{k-1} - U_k)_+ - \frac{1}{4}\right]\right).$$

L'appel `np.clip(t, 0., 2.)` réalise précisément cette projection sur l'intervalle.

```
import numpy as np
import matplotlib.pyplot as plt

def robbins_monro(n, gamma, seed):
    rng = np.random.default_rng(seed)
    theta = 1.5
    history = np.empty(n+1)
    history[0] = theta
    for k in range(1, n+1):
        u = rng.uniform(0., 2.)
        y = max(theta-u, 0.)
        theta = np.clip(theta - gamma(k)*(y-0.25), 0., 2.)
        history[k] = theta
```

```
    return history

steps = {
    "2/k": lambda k: 2/k,
    "0.8/k**0.7": lambda k: 0.8/k**0.7,
    "constant 0.08": lambda k: 0.08,
}
results = {label: robbins_monro(20000, rule, 101)
           for label, rule in steps.items()}

for label, hist in results.items():
    print(label, hist[-1], np.std(hist[-1000:]))
    plt.plot(np.arange(len(hist)), hist, label=label)
plt.axhline(1.0, linestyle="--", color="black", label="theta*=1")
plt.xscale("symlog", linthresh=20)
plt.xlabel("iteration k")
plt.ylabel("theta_k")
plt.legend()
plt.tight_layout()
plt.show()
```

2) **Interprétation et résultats attendus.** Sur la graine 101 et 20 000 itérations, les dernières valeurs et écarts types empiriques sur les 1 000 derniers itérés sont, à l'arrondi près :

Pas	Dernière valeur	Écart type des 1000 derniers itérés
$2/k$	1,000143	0,000227
$0{,}8/k^{0{,}7}$	1,009256	0,002520
0,08 constant	0,972765	0,086701

Ces valeurs varient avec la graine aléatoire. Avec les pas décroissants, les corrections finissent par être très petites : le bruit est progressivement amorti. Un pas constant continue au contraire à injecter, à chaque itération, une perturbation de taille comparable, de sorte que la suite peut rester fluctuante même si elle oscille autour de la bonne solution. Il est donc utile de regarder à la fois la valeur finale, le chronogramme et la variabilité récente. Il serait trompeur de juger l'algorithme à partir d'une seule réalisation ou de prétendre que $2/k$ est toujours optimal : ce facteur dépend de l'échelle de F, du bruit et du nombre d'itérations.

Idée à retenir

La méthode de Robbins–Monro utilise des observations **bruitées mais correctement centrées** pour orienter une recherche récursive. Les pas doivent être assez grands au total pour corriger l'erreur, mais assez petits à long terme pour laisser le bruit s'atténuer.