

Projet numérique de Probabilités 2021

January 12, 2021

Instructions

Ce sujet est constitué de deux parties : la partie 1 correspond au sujet donné l'an dernier, dont la correction vous est donnée, la partie 2 constitue la partie qui sera évaluée et se place dans la continuité de la partie 1.

1 Enoncé du problème (Partie 1)

L'objectif de ce projet est d'estimer la longueur de câble sous-marin nécessaire pour relier deux côtes A et B en utilisant des simulations conditionnelles.

Le câble reposera sur le fond marin dont la profondeur est inconnue. Le segment $[AB]$ est discrétisé par une séquence de $(N+1)$ points. On pose $x_0 = A$ et pour $i = 1, \dots, N$,

$$x_i = x_0 + i\Delta$$

où

$$\Delta = \frac{AB}{N}$$

de telle sorte que $x_N = B$. On note $z(x)$ la profondeur du fond marin au point x de telle sorte qu'on pourra estimer la longueur totale de câble nécessaire par la somme des longueurs sur les segments de la discrétisation :

$$l = \sum_{i=1}^N \sqrt{\Delta^2 + (z(x_i) - z(x_{i-1}))^2}.$$

Enfin, notons que l'on dispose d'un ensemble de n observations de la profondeur que l'on supposera situées sur des points de discrétisation $z(x_{j_1}), \dots, z(x_{j_n})$.

On adopte un modèle probabiliste pour la profondeur. On suppose que le vecteur des profondeurs sur les points de discrétisation $\mathbf{z} = (z(x_0), \dots, z(x_N))$ est la réalisation d'un vecteur aléatoire gaussien $\mathbf{Z} = (Z(x_0), \dots, Z(x_N))$ dont le vecteur d'espérance ne contient qu'une seule valeur μ répétée $N+1$ fois et dont la matrice de covariance Σ a pour termes σ_{ij} définis par $\sigma_{ij} = C(|x_i - x_j|)$ où C est une fonction décroissante, traduisant le fait que deux points géographiquement proches ont tendance à avoir des profondeurs plus similaires que deux points éloignés.

On supposera que la matrice de covariance ainsi générée est définie-positive (en fait, C sera choisie parmi les fonctions qui, appliquées aux termes d'une matrice de distance, produisent des matrices définie-positives).

Si on note L la variable aléatoire donnant la longueur de câble nécessaire :

$$L = \sum_{i=1}^N \sqrt{\Delta^2 + (Z(x_i) - Z(x_{i-1}))^2},$$

un bon estimateur de L est fourni par l'espérance conditionnelle

$$L^* = E[L | Z(x_{j_1}) = z(x_{j_1}), \dots, Z(x_{j_n}) = z(x_{j_n})].$$

Cependant, cette quantité est difficilement accessible par le calcul. On va donc avoir recours à des simulations conditionnelles. C'est-à-dire que l'on va simuler un nombre K de réalités (disons des réalisations du modèle probabiliste choisi), et sur chacune d'entre elle, la quantité de câble nécessaire sera évaluée. On disposera ainsi d'un échantillon $l_{(1)}, \dots, l_{(K)}$ de longueurs simulées. Puis on approchera l'espérance conditionnelle par

$$L^* = \frac{1}{K} \sum_{k=1}^K l_{(k)}.$$

L'objectif de ce projet est donc d'écrire un code permettant d'effectuer cette simulation conditionnelle, puis de l'appliquer au jeu de données fourni et d'en déduire une estimation de la longueur de câble nécessaire.

1.1 Questions théoriques

1. Quel théorème du cours nous autorise-t-il à estimer l'espérance conditionnelle par la moyenne empirique de simulations conditionnelles ?

la loi forte des grands nombres, cf [Proba IV p.16](#), en vérifiant que L est bien intégrable

2. Rappeler la loi conditionnelle du vecteur des composantes de \mathbf{Z} correspondant aux points de discrétisation sans observation, connaissant les valeurs prises par les composantes aux sites d'observation.

cf. [Proba III p.18](#),

3. Si $\mathbf{Y} = (Y_1, \dots, Y_p)$ est un vecteur de composantes gaussiennes indépendantes, toutes d'espérance nulle et de variance 1, quelle est la loi du vecteur $\mathbf{Z} = m + R\mathbf{Y}$ où R est une matrice $p \times p$ et m est un vecteur de taille p ?

cf. [Proba V p.12](#)

4. En déduire un algorithme de simulation conditionnelle.

Appliquer l'algorithme de la q.3 à la loi conditionnelle (on calculera en particulier l'espérance conditionnelle de \mathbf{Z} sachant les données et la décomposition de Cholesky de la matrice de covariance conditionnelle sachant les données).

1.2 Données du problème

Conventionnellement, A est l'origine, $B = 500$, $N = 100$.

Les données

i	$z(x_i)$
0	0
20	-4
40	-12.8
60	-1
80	-6.5
100	0

L'espérance de chaque composante du vecteur aléatoire \mathbf{Z} est donnée par $\mu = -5$.

La fonction C est définie par

$$C(h) = \sigma^2 e^{-|h|/a},$$

où $|h|$ correspond à la distance entre deux points, $a = 50$ et $\sigma^2 = 12$.

1.3 Implémentation

1.3.1 Préambule

```
[ ]: #Chargement de dépendances

import numpy as np
import matplotlib.pyplot as plt

#Discrétisation
A=0
B=500
N=101 #Nombre de points de discrétisation
Delta = (B-A)/(N-1)
discretization_indexes = np.arange(N)
discretization = discretization_indexes*Delta
#Paramètres du modèle

mu=-5
a = 50
sigma2 = 12

#Données

observation_indexes = [0,20,40,60,80,100]
depth = np.array([0,-4,-12.8,-1,-6.5,0])
```

```
#Indices des composantes correspondant aux observations et aux composantes non-
↪ observées

unknown_indexes=list(set(discretization_indexes)-set(observation_indexes))
```

1.3.2 Questions

1. Ecrire une fonction qui prend en argument la distance entre les points, le paramètre a , et le paramètre σ^2 , et qui retourne la covariance entre deux points. On pourra fournir une matrice de distance à cette fonction. Dans ce cas, la fonction renverra la matrice de covariance.

```
[ ]: #Fonction C

def Covexp(dist,rangeval,sigma):
    return sigma * np.exp(-dist/rangeval)
```

2. Calculer la matrice de distance.

```
[ ]: distmat=np.subtract.outer(discretization,discretization)
```

3. Calculer la matrice de covariance du vecteur $\mathbf{Z} = (Z(x_0), \dots, Z(x_N))$.

```
[ ]: Sigma=Covexp(distmat,a,sigma2)
```

4. Extraire les 3 matrices de covariance suivantes :

- entre les observations
- entre les observations et les inconnues
- entre les inconnues

```
[ ]: SigmaObs = Sigma[observation_indexes,:][:,observation_indexes]
SigmaObsUnknown = Sigma[observation_indexes,:][:,unknown_indexes]
SigmaUnknown = Sigma[unknown_indexes,:][:,unknown_indexes]
```

5. Calculer l'espérance conditionnelle des composantes non observées connaissant les observations et la représenter avec les données.

```
[ ]: invSigma = np.linalg.inv(SigmaObs)
Ec= mu+np.matmul(np.transpose(SigmaObsUnknown),np.matmul(np.linalg.
↪ inv(SigmaObs),depth-mu))
```

```
[ ]: allval1 = np.zeros(N)
allval1[unknown_indexes]=Ec
allval1[observation_indexes]=depth
plt.plot(discretization,allval1)
plt.plot(discretization[observation_indexes], depth, 'ro')
plt.show()
```

6. Calculer la matrice de variance conditionnelle et tracer sa diagonale (variance conditionnelle) en fonction de la position. Commenter.

```
[ ]: SigmaCond = SigmaUnknown - np.matmul(np.transpose(SigmaObsUnknown), np.matmul(np.  
    ↪ linalg.inv(SigmaObs), SigmaObsUnknown))
```

```
[ ]: allval2 = np.zeros(N)  
allval2[unknown_indexes]=np.diag(SigmaCond)  
plt.plot(discretization, allval2)  
plt.plot(discretization[observation_indexes], np.zeros(np.  
    ↪ shape(observation_indexes)[0]), 'ro')  
plt.show()
```

On voit ici que la variance est nulle aux points de données (puisque la valeur est fixée) puis qu'elle augmente lorsqu'on s'éloigne des données

7. Effectuer une simulation conditionnelle. Sur un même graphique, tracer la simulation ainsi que les données et l'espérance conditionnelle. Commenter.

```
[ ]: Cholesky = np.linalg.cholesky(SigmaCond)  
x = np.random.normal(0,1,np.shape(unknown_indexes)[0])  
simu = Ec + np.matmul(Cholesky,x)
```

```
[ ]: allval3 = np.zeros(N)  
allval3[unknown_indexes]=simu  
allval3[observation_indexes]=depth  
plt.plot(discretization, allval3)  
plt.plot(discretization, allval1)  
plt.plot(discretization[observation_indexes], depth, 'ro')  
plt.show()
```

l'espérance conditionnelle est lisse car c'est la fonction des données qui minimise la variance ; la simulation conditionnelle reproduit les fluctuations du modèle et donc l'aspect du fond marin

8. Ecrire une fonction qui calcule la longueur du câble en fonction du vecteur des profondeurs et du pas de discrétisation.

```
[ ]: def length(z,delta):  
    return sum(np.sqrt(Delta**2+(z[1:N]-z[0:-1])**2))
```

9. Utiliser cette fonction pour calculer la longueur du câble à partir de 100 simulations. Comparer l'espérance conditionnelle (estimée) de la longueur avec la longueur de l'espérance conditionnelle.

```
[ ]: K=100000  
result = np.zeros(K)  
for i in range(K):  
    x=np.random.normal(0,1,np.shape(unknown_indexes)[0])  
    allval3[unknown_indexes]=Ec + np.matmul(Cholesky,x)
```

```
result[i]=length(allval3,Delta)
```

```
[ ]: sum(result)/K
```

```
[ ]: length(allval1,Delta)
```

On donne ici directement les résultats pour 100000 simulations (comparer la vitesse d'exécution avec votre code pour voir l'intérêt de la vectorisation).

On voit ici que la longueur donnée par l'espérance conditionnelle est bien moindre que la longueur d'une simulation conditionnelle. Ceci est dû au fait que la simulation conditionnelle va reproduire les fluctuations attendues, au contraire de l'espérance conditionnelle qui minimise la variance.

NB : On remarquera que c'est le seul endroit où on a utilisé une boucle dans cette correction. On aurait pu s'en passer en utilisant notamment les outils de la librairie Pandas mais le code est plus lisible ainsi.

10. Représenter la suite M_n des moyennes des longueurs de câbles en fonction du nombre de simulations. Commenter.

```
[ ]: indice_simu = 1+np.arange(K)
plt.plot(indice_simu,np.cumsum(result)/indice_simu)
plt.show()
```

Comme attendu d'après la LFGN, on observe une stabilisation de la valeur moyenne. Les variations sont imperceptibles après quelques dizaines de milliers de simulations

11. Représenter l'histogramme des longueurs de câbles générées.

```
[ ]: plt.hist(result,50,density=True)
plt.show()
```

L'histogramme ressemble à une densité gaussienne, avec néanmoins une légère dissymétrie (plus de valeurs fortes).

12. Donner un intervalle de confiance à 95% de la longueur du câble par 2 méthodes différentes. Commenter.

1ère méthode : par approximation gaussienne. L'histogramme précédent suggère qu'une loi gaussienne peut bien représenter la loi conditionnelle de la longueur du câble. On obtient un intervalle de confiance en calculant la moyenne (milieu de l'intervalle) et l'écart type de l'échantillon de longueurs de câble puis en appliquant la formule du cours.

```
[ ]: Ln = sum(result)/K
sigman = np.std(result)
[Ln - sigman*1.96, Ln + sigman*1.96]
```

2e méthode : on extrait les quantiles à 2.5% et 97.5% de l'échantillon, c'est-à-dire les valeurs x_α pour $\alpha \in \{0.025; 0.975\}$ telles que

$$\mathbb{P}(L < x_\alpha) \approx \frac{1}{n} \sum_{i=1}^N 1_{\{L_i < x_\alpha\}}$$

```
[ ]: np.quantile(result,[0.025,0.975])
```

ATTENTION : ici l'utilisation du TCL tel qu'exposé au début du chapitre 5 fournissait un intervalle de confiance de l'espérance de la longueur de câble et non de la longueur de câble (erreur fréquente).

On voit par ailleurs que la 2e méthode est plus précise car elle est basée directement sur la loi empirique de l'échantillon

13. Donner une estimation de la probabilité que la longueur du câble dépasse 525 m.

```
[ ]: np.mean(result>525)
```

14. Reprendre les questions précédentes avec 1000, 10000 puis 100000 simulations. Commenter cf. le commentaire de la question 10. On observe une stabilisation progressive des estimateurs des différentes quantités calculées : moyennes, quantiles, probabilité de dépassement de seuil.

2 Enoncé 2021 (Partie 2)

Nous nous plaçons dans le même cadre que ci-dessus. Nous tenons pour acquis les résultats précédents et allons maintenant approfondir l'analyse.

2.1 Valeurs positives

a= 1 ,rajouter U...

fx(Y) ? quel test exactement ?

- préservé loi ? 1. A la question 7 de la partie 1, on peut s'apercevoir que certains profils de profondeur présentent des valeurs positives, ce qui n'a pas de sens si on considère qu'il n'y a pas de partie émergée entre les deux côtes. Proposer et implémenter un algorithme de type rejet qui ne génèrera que des valeurs négatives. Représenter un profil de profondeur généré.
- quoi y et quoi U ?
quelle DENSITE ?
juste tester ? 2. L'emploi de cet algorithme a-t-il un impact sur la valeur moyenne de la longueur de câble ? Sur l'histogramme des longueurs de câble ? Sur l'intervalle de confiance obtenu par la méthode des quantiles ? Sur la probabilité de dépassement du seuil de 525 m ? Donner une estimation du taux de rejet de l'algorithme. Considérer 10000 simulations.
- tout retracer et comparer, être fin...

2.2 Approche Bayésienne

taux de rejet : valeur théorique et avec expérimentation ?

Dans la première partie, on a considéré que l'ensemble des paramètres du modèle était parfaitement connu. Toutes choses égales par ailleurs, on va maintenant considérer que la moyenne μ est issue d'une estimation entachée d'une incertitude. Plus précisément, on va modéliser cette incertitude en considérant que $\mu \sim \mathcal{N}(-5, 4)$.

On va également décomposer le vecteur \mathbf{Z} en $\mathbf{Z} = (\mathbf{Z}_{\text{OBS}}, \mathbf{Z}_{\text{UNK}})$. On notera C_{OBS} et C_{UNK} les matrices de covariance respectives de \mathbf{Z}_{OBS} et \mathbf{Z}_{UNK} ainsi que $C_{\text{OBS,UNK}}$ la matrice des covariances entre ces deux vecteurs.

2.2.1 Questions théoriques

3. Montrer que $\mu|\mathbf{Z}_{\text{OBS}} = \mathbf{z}_{\text{OBS}} \sim \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)$ avec

$$\hat{\sigma}^2 = (\mathbf{1}^t C_{\text{OBS}}^{-1} \mathbf{1} + \frac{1}{4})^{-1},$$

où $\mathbf{1}$ est un vecteur de longueur 6 (soit la longueur du vecteur \mathbf{Z}_{OBS}) ne contenant que des 1 et

$$\hat{\mu} = \hat{\sigma}^2 (\mathbf{1}^t C_{\text{OBS}}^{-1} \mathbf{z}_{\text{OBS}} - \frac{5}{4})$$

Indication : écrire la densité jointe du vecteur $(\mathbf{Z}_{\text{OBS}}, \mu)$, calculer $-2 \ln$ de cette densité puis identifier la forme quadratique faisant intervenir μ .

4. Montrer que si X , Y et Z sont trois vecteurs aléatoires à densité, alors $f_{X,Y|Z=z}(x,y) = f_{X|Y=y,Z=z}(x)f_{Y|Z=z}(y)$
5. En déduire que la densité de $(\mathbf{Z}_{\text{UNK}}, \mu)|\mathbf{Z}_{\text{OBS}} = \mathbf{z}_{\text{OBS}}$ s'écrit comme le produit de $f_{\mu|\mathbf{Z}_{\text{OBS}}=\mathbf{z}_{\text{OBS}}}$ et de $f_{\mathbf{Z}_{\text{UNK}}|\mu=\mu^*, \mathbf{Z}_{\text{OBS}}=\mathbf{z}_{\text{OBS}}}$ que l'on exprimera.

Indication : $f_{\mathbf{Z}_{\text{UNK}}|\mu=\mu^*, \mathbf{Z}_{\text{OBS}}=\mathbf{z}_{\text{OBS}}}$ est identique à la densité $f_{\mathbf{Z}_{\text{UNK}}|\mathbf{Z}_{\text{OBS}}=\mathbf{z}_{\text{OBS}}}$ utilisée dans la partie 1 avec $\mu = \mu^*$

2.2.2 Simulation

On va maintenant chercher à simuler les valeurs de \mathbf{Z}_{UNK} en tenant compte de l'incertitude sur μ . En première approche, on pourrait chercher à obtenir la densité de $\mathbf{Z}_{\text{UNK}}|\mathbf{Z}_{\text{OBS}} = \mathbf{z}_{\text{OBS}}$ en calculant

$$\int_{\mathbb{R}} f_{(\mathbf{Z}_{\text{UNK}}, \mu)|\mathbf{Z}_{\text{OBS}}=\mathbf{z}_{\text{OBS}}}(\mathbf{Z}_{\text{UNK}}, \mu^*) d\mu^*$$

mais ce calcul a peu de chances d'aboutir.

On va plutôt procéder par simulations en cascade, c'est-à-dire que pour chaque simulation de \mathbf{Z}_{UNK} , on va d'abord générer une valeur μ^* de μ selon $f_{\mu|\mathbf{Z}_{\text{OBS}}=\mathbf{z}_{\text{OBS}}}$ puis on simulera \mathbf{Z}_{UNK} selon $f_{\mathbf{Z}_{\text{UNK}}|\mathbf{Z}_{\text{OBS}}=\mathbf{z}_{\text{OBS}}, \mu=\mu^*}$.

6. Calculer $\hat{\mu}$ et $\hat{\sigma}^2$.
7. Calculer la probabilité de $\{\mu|\mathbf{Z}_{\text{OBS}} = \mathbf{z}_{\text{OBS}} > 0\}$. Proposer et implémenter un algorithme de simulation de $\mu|\mathbf{Z}_{\text{OBS}} = \mathbf{z}_{\text{OBS}}, \mu < 0$.
8. Générer un couple $\mathbf{Z}_{\text{UNK}}, \mu|\mathbf{Z}_{\text{OBS}}, \mu < 0$ et représenter le profil de profondeur obtenu. On prendra soin de ne générer que des valeurs négatives de \mathbf{Z}_{UNK} en utilisant l'algorithme de la section précédente.
9. Générer 10000 profils de profondeur et calculer les longueurs de câble associées. Comparer la valeur moyenne de la longueur de câble, dont on visualisera la convergence, l'histogramme des longueurs de câble, l'intervalle de confiance obtenu par la méthode des quantiles, la probabilité de dépassement du seuil de 525 m avec les résultats obtenus précédemment. Commenter.
10. Qu'advierait-il selon vous si on rendait également aléatoires les autres paramètres du modèle σ^2 et a ?

11. (facultatif) On suppose désormais que $\alpha = \frac{1}{\sigma^2} \sim \Gamma(a_1, a_2)$, où σ^2 correspond au paramètre qui intervient dans la fonction C . Donner l'expression de la densité de $\alpha | \mathbf{Z}_{\text{OBS}} = \mathbf{z}_{\text{OBS}}$, de $\mu | \alpha, \mathbf{Z}_{\text{OBS}} = \mathbf{z}_{\text{OBS}}$, et de $\mathbf{Z}_{\text{UNK}} | \mu, \alpha, \mathbf{Z}_{\text{OBS}} = \mathbf{z}_{\text{OBS}}$. Proposer un algorithme de simulation tenant compte de l'incertitude sur le paramètre α et analyser son impact sur les différentes quantités évaluées.

Indication : Pour identifier la densité de $\alpha | \mathbf{Z}_{\text{OBS}} = \mathbf{z}_{\text{OBS}}$, exprimer d'abord la densité jointe de $(\alpha, \mu, \mathbf{Z}_{\text{OBS}})$ puis marginaliser pour obtenir la densité de $(\alpha, \mathbf{Z}_{\text{OBS}})$