Filière 3EA
1re année

Statistique

BE n°2 - Tests d'hypothèses paramétriques et régression multilinéaire

1 Détection de présence d'un signal

Un signal $\mathbf{s} = (s_1, s_2, \ldots, s_n)^\mathsf{T}$ est envoyé sur un canal de télécommunication. Lors de sa transmission, ce signal est modulé en amplitude par un facteur θ (constant sur toute la durée du signal et non-négatif) et perturbé par un bruit additif gaussien $\mathbf{b} = (b_1, b_2, \ldots, b_n)^\mathsf{T}$ de moyenne nulle et de variance σ^2, de telle sorte que le signal reçu $\mathbf{y} = (y_1, y_2, \ldots, y_n)^\mathsf{T}$ s'écrit sous la forme :

$$y_i = \theta s_i + b_i \ , \ \ 1 \leq i \leq n.$$

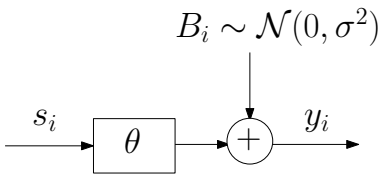

FIGURE 1 – Transmission d'un signal par un canal bruité.

En réception, il s'agit de déterminer si les observations $\mathbf{y} = (y_1, y_2, \ldots, y_n)^\mathsf{T}$ ne contiennent que du bruit, ou si elles contiennent également le signal modulé. Ce problème de détection de signal dans du bruit additif peut s'écrire sous la forme du test binaire d'hypothèses simples suivant :

$$\begin{array}{ll} H_0 : & \theta = \theta_0 = 0 \\ H_1 : & \theta = \theta_1 \neq 0 \end{array}$$

1. Montrer que la statistique de test donnée par le **théorème de Neyman et Pearson** s'écrit :

$$T\left(y_1, \ldots, y_n\right) = \mathbf{s}^\mathsf{T}\mathbf{y}.$$

Pour une probabilité de fausse alarme α, la région critique (zone de rejet de H_0) est donnée par :

$$R_\alpha = \left\{ (y_1, \ldots, y_n) \in \mathbb{R}^n \,\middle|\, \mathbf{s}^\mathsf{T}\mathbf{y} > \lambda_\alpha \right\}$$

Dans ce cas, le seuil de décision s'écrit :

$$\lambda_\alpha = \sigma\sqrt{\mathbf{s}^\mathsf{T}\mathbf{s}}\,\Phi^{-1}(1-\alpha) \ ,$$

où $\Phi(x)$ est la fonction de répartition de la loi normale centrée réduite calculée au point x, c'est-à-dire :

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$$

et $\Phi^{-1}(\cdot)$ est l'inverse de cette fonction de répartition.

La probabilité de non-détection ou risque de deuxième espèce s'écrit aors :

$$\beta = \Phi\left(\Phi^{-1}(1-\alpha) - \frac{\theta_1\sqrt{\mathbf{s}^\mathsf{T}\mathbf{s}}}{\sigma}\right).$$

On souhaite tracer les courbes théoriques de la puissance du test $\pi = 1-\beta$ en fonction de la probabilité de fausse alarme α et les comparer aux courbes obtenues par simulation (comparaison des courbes ROC théorique et empirique).

2. En utilisant les fonctions `norminv` et `normcdf`, écrire une fonction

   `piTheo=pi_theorique(signal,theta1,sigma2,alpha)` qui renvoie la puissance théorique π du test pour un vecteur de risques de première espèce `alpha`, en fonction de `signal`, `sigma2` et `n`. Afficher le vecteur `piTheo` obtenu avec $\theta_1 = 1$, $n = 20$, $s_i = \sin(2\pi \times 0.1 \times i)$, $1 \leq i \leq n$, $\sigma^2 = 1$ et $\boldsymbol{\alpha} = (0.01, 0.02, 0.03, \ldots, 0.98, 0.99)$. Superposer alors les courbes obtenues pour $\sigma^2 = 2$ et $\sigma^2 = 3$. Commenter (à l'aide de l'expression de β).

3. On cherche maintenant à retrouver ces résultats par simulation.
   Ecrire une fonction `Y = generer(theta1,signal,sigma2,K)` qui renvoie une matrice `Y` de taille $n \times K$, dont chaque colonne contient une réalisation du vecteur signal $\mathbf{y} = (y_1, y_2, \ldots, y_n)^\mathsf{T}$ générée sous l'hypothèse H_1.

4. Ecrire une fonction `piEst=pi_estimee(signal,theta1,sigma2,K,alpha)` qui renvoie la puissance estimée $\widehat{\pi}$ du test pour un vecteur de risques de première espèce `alpha`, en fonction de `signal`, `sigma2`, `n`, et du nombre de simulations `K` (cette fonction appellera la fonction `generer`). Superposer à la courbe théorique obtenue à la question 1 les résultats empiriques générés par `pi_estimee` avec $n = 20$, $s_i = \sin(2\pi \times 0.1 \times i)$, $1 \leq i \leq n$, $\sigma^2 = 1$, $K = 500$ et $\boldsymbol{\alpha} = (0.01, 0.02, 0.03, \ldots, 0.98, 0.99)$. On tracera la courbe des résultats empiriques en forme d'escalier, à l'aide de la fonction `stairs`. Commenter.

1.1 Régression linéaire multiple

On souhaite illustrer sur un exemple simple la régression multilinéaire sous Matlab.

1. Charger tout d'abord le fichier `carsmall` avec la ligne de commande `load carsmall.mat` (qui rentre en mémoire vive les diverses variables présente dans le fichier, il faudra faire attention à la casse des variables). Ce fichier regroupe des données relatives à des automobiles.

En particulier, on souhaite modéliser la consommation en carburant (variable `MPG` pour Miles per Gallon qu'on stockera dans un vecteur `y`) d'un véhicule en fonction de son poids (variable `Weight` qu'on stockera dans un vecteur `x1`) et de sa puissance (variable `Horsepower` qu'on stockera dans un vecteur `x2`).

2. Pour calculer le produit de la matrice $(\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}$ avec un vecteur $\mathbf{y}$ sous Matlab, nous pouvons utiliser l'opérateur *backslash* : `X \ y`. Effectuer le calcul nécessaire pour calculer les paramètres de la régression des moindres carrées pour le modèle

   $$\hat{y} = a_0 + a_1 x_1 + a_2 x_2,$$

   et afficher le résultat. Que se passe-t-il ? Afficher les valeurs de `X` et de `y` et donner la raison du résultat. Comment circonvenir ce problème ?

3. La fonction `regress` résout en interne ce problème. Elle vérifie
   — variables d'entrée : le vecteur contenant la consommation en carburant (la réponse `y`) et la matrice `X` dont la première colonne est constituée de 1 et les deux suivantes correspondent aux deux prédicteurs considérés `x1` et `x2`,
   — variable de sortie : le vecteur `b` contenant les coefficients du modèle.

Calculer avec cette fonction les paramètres du modèle.

4. On veut tracer sur un même graphique, les données et les valeurs estimées par le modèle.
   Pour calculer les valeurs estimées par le modèle, on prendra des valeurs de x_1 linéairement espacées entre la valeur minimale observée et la valeur maximale observée de `x1` avec un pas de 100 et pour des valeurs de x_2 linéairement espacées entre la valeur minimale observée et la valeur maximale observée de `x2` avec un pas de 10.
   Pour l'affichage, on pourra utiliser la fonction `scatter3` pour représenter les données observées sous forme de nuage de points en 3D et la fonction `mesh` pour représenter le plan obtenu à partir du modèle de régression (utiliser `hold on` pour afficher tout ensemble).

Choix du modèle :

Charger le fichier `hald`. Celui-ci regroupe des données relatives à la composition du ciment et à sa température de durcissement. En utilisant la fonction `regstat`, déterminer quel modèle est le plus approprié (linéaire, linéaire avec interaction ou purement quadratique) pour représenter la température de durcissement du ciment en fonction de ses ingrédients. Pour cela, on pourra par exemple comparer l'erreur quadratique moyenne et le coefficient de détermination obtenus pour les différents modèles.