TP de Probabilités (1)

Génie Informatique
Licence STS Mention Electronique, année 2004-2005

Ce TP aura pour objet de familiariser les étudiants avec quelques notions de bases de probabilités et d'initier à la programmation sous Matlab. Matlab est un logiciel de programmation adapté au calcul matriciel et disposant de nombreuses routines mathématiques notamment statistiques. Ce sont ces dernières que nous exploiterons lors de ce TP.

Exercice 1 : loi Normale, densité de probabilité

Tout d'abord, lançons l'application "Matlab.exe", et ouvrons un nouveau fichier à l'aide de l'éditeur de Matlab. Ce dernier prendra l'extension ".m" lors de l'archivage. Tout script possédant l'extension ".m" peut être exécuter sous Matlab à partir de la fenêtre de dialogue. Ainsi on tapera "tp1" suivi du "retour chariot" afin d'exécuter le programme sauvegardé sous le nom "tp1.m".

Commençons par générer un processus de N variables aléaroires gausiennes de moyenne nulle et de variance unité. Ceci se fait à l'aide de la commande :

$$gaussien = randn(1, N);$$

On prendra successivement $N = 10$, 100 puis 1000. Notons par ailleurs que le ";" à la fin d'une commande Matlab aura pour effet de ne pas afficher à l'écran le contenu du vecteur "*gaussien*" lors de l'éxecution du code Matlab. A présent nous allons représenter graphiquement les valeurs de ce processus à l'aide du script suivant :

$$figure(1);$$
$$plot(gaussien);$$

Chercher à l'aide de la commande "*help*" de Matlab suivi du nom de la fonction, l'effet des deux commandes précédentes. De même nous pourrons habiller le graphe obtenu à l'aide des commandes suivantes :

$$title(\text{'Représentation graphique d un processus'});$$
$$xlabel(\text{'Nombre d échantillons'});$$
$$ylabel(\text{'Valeurs du processus'});$$

A présent, représentons l'histogramme de notre vecteur gaussien à l'aide de la commande "*hist*". Rappelons que le rôle d'un histogramme est de quantifier graphiquement la répartition d'un ensemble de nombres sur un ensemble d'intervalles, autrement dit de comptabiliser le nombre de points appartenant à chaque intervalle. Exécutons les commandes suivantes :

$$M = 10;$$
$$figure(2);$$
$$hist(gaussien, M);$$

Puis :

$$figure(3);$$
$$dx = -2.9 : 0.1 : 2.9;$$
$$hist(gaussien, dx);$$

Quelle différences observe-t-on entre les figures 2 et 3 ?

Dès lors, représentons graphiquement la courbe asociée à la densité de probabilité d'une variable aléatoire gaussienne :

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\ \mathrm{e}^{-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2}$$

où σ et m désignent respectivement l'écart type et la moyenne de la variable gaussienne. Pour cela, donnons nous un vecteur $\boldsymbol{x}$ de valeurs réelles :

$$x = -50 : pas : 50;$$

et construisons le vecteur $\boldsymbol{f}_x$ contenant les valeurs de la fonction $f(x)$ prises pour les différents points de $\boldsymbol{x}$. La fonction exponentielle existe sous Matlab sous le nom de *exp(.)*, notons qu'elle peut prendre en argument un vecteur de points. l'opération d'élévation à une certaine puissance se fait à l'aide de l'opérateur "^". Quant à la racine carrée, elle est effectuée à l'aide de la commande *sqrt(.)*. Que remarque-t-on ? quel lien peut-on faire avec l'histogramme ?

Exercice 2 : loi uniforme, estimateurs empiriques de statistiques

De même, il est possible sous Matlab de générer des variables de loi uniforme. Ceci se fait à l'aide de la fonction "*rand(.)*" qui s'utilise comme la fonction "*randn(.)*".

Ainsi dans le domaine des télécommunications radios, bien que l'information soit lors de son transport sous forme analogique, lors de la réception une conversion analogique/numérique est mise en oeuvre. Le traitement d'information qui suit cette étape se fait alors sur des signaux plus communément appelés modulations numériques. En pratique il existe un certain nombre de modulations numériques, chacune d'elle possédant des caractéristiques différentes (temporelles, spectrales, etc...). Une modulation couramment utilisée est la QPSK filtrée NRZ. Du point de vue du récepteur, elle peut être considérée comme une séquence de variables aléatoires i.i.d. (indépendantes et identiquement distribuées) prenant leur valeur de manière équiprobable dans l'ensemble $\{1, \mathrm{i}, -1, -\mathrm{i}\}$. Nous allons sous Matlab générer un tel processus.

Tout d'abord construisons un vecteur $\boldsymbol{x}$ de N variables uniformément distribuées dans l'intervalle $[0, 1]$. Soit g l'application définie par :

$$g(x_j) = \begin{cases} 1 \text{ si } x_j \in [0, 1/4[\\ \mathrm{i} \text{ si } x_j \in [1/4, 1/2[\\ -1 \text{ si } x_j \in [1/2, 3/4[\\ -\mathrm{i} \text{ si } x_j \in [3/4, 1[\end{cases}$$

Appliquons à chaque composante de $\boldsymbol{x}$ la fonction g afin d'obtenir le vecteur $\boldsymbol{y}$. On pourra utiliser le fait que le script suivant :

$$v = (x >= 0.25\ \&\ x < 0.5);$$

a pour effet d'affecter à la j-ième composante de $\boldsymbol{v}$ la valeur 1 si la j-ième composante de $\boldsymbol{x}$ appartient à l'intervalle $[1/4, 1/2[$ et la valeur 0 sinon. Ainsi le vecteur $\boldsymbol{y}$ obtenu est une réalisation du processus QPSK filtré NRZ. Représentons alors graphiquement la constellation de la dite QPSK, autrement dit les valeurs du processus $\boldsymbol{y}$.

Dans un second temps, nous allons estimer la moyenne et la variance des variables du processus précédent et les comparer aux valeurs théoriques. L'estimateur de la moyenne empirique d'une des variables du processus $\boldsymbol{y}$ nous est donné par la formule suivante :

$$\hat{m} = \frac{1}{N} \sum_{j=1}^{N} y_i$$

L'estimateur de la variance nous est donné par :

$$\hat{\sigma}^{\,2} = \frac{1}{N-1} \sum_{j=1}^{N} |y_i|^2$$

Nous pouvons alors utiliser sous Matlab la fonction "*sum(.)*" pour calculer $\hat{m}$. Quant à $\hat{\sigma}^{\,2}$, ce n'est rien d'autre que la norme au carré du vecteur $\boldsymbol{y}$ divisée par $N-1$. Matlab est un logiciel adapté au calcul matriciel. Ainsi, le produit scalaire de vecteurs $\boldsymbol{y}$ et $\boldsymbol{z}$ à composantes complexes s'écrit simplement sous Matlab[1] : $y * z'$. Notons cela dit qu'il existe sous Matlab des fonctions prédéfinies nommées "*mean(.)*" et "*var(.)*" accomplissant ce travail. Enfin, pour clore cette partie, nous allons représenter sur un même graphique l'estimateur empirique de la moyenne en fonction du nombre de variables N du processus, ainsi que la moyenne théorique. Faisons de même pour la variance empirique. Que remarque-t-on ?

Exercice 3 : transformation affine de variable aléatoire

Nous allons dans cette partie voir l'effet des paramètres de position et de dispersion sur une variable aléatoire donnée. Autrement dit, quel comportement statistique adopte la variable Y définie par :

$$aX + b$$

lorsque X suit une certaine loi. D'autre part, nous allons à titre d'exemple prendre comme variable X une variable de loi normale (d'espérance et de variance unité), notée X_1, et une variable de loi uniforme sur $[0, 1]$, notée X_2. A chacune d'elles nous associerons respectivement Y_1 et Y_2.

Dans un premier temps, nous allons nous intéresser à l'impact du paramètre a et de ce fait posons $b = 0$: représentons alors graphiquement la moyenne (moment d'ordre 1), la variance et le moment d'ordre 2 empiriques de Y_j en fonction de a. Quel est l'effet du paramètre a sur les différentes statistiques de X_j ?

Puis, intéressons-nous à l'impact du paramètre b et posons de ce fait $a = 1$. Après avoir représenter graphiquement la moyenne, la variance et le moment d'ordre 2 empiriques de Y_j en fonction de b cette fois, que peut-on dire de l'effet de b sur les différentes statistiques de X_j ?

[1] La transposée conjuguée (transposée hermitienne) du vecteur $\boldsymbol{z}$ s'effectue sous Matlab en ajoutant à $\boldsymbol{z}$ une apostrophe, la transposée non conjuguée (transposée symétrique) en ajoutant un point suivi d'une apostrophe.