

TP du 17 octobre 2012 de 14h à 18h

TP SCILAB

1 Introduction à SCILAB

SCILAB est un langage interprété c'est-à-dire que l'on peut exécuter directement des instructions dans la fenêtre où se trouve le prompt `-->` ou exécuter une suite d'instructions depuis un fichier `toto.sci` en utilisant la commande `exec('toto.sci')`. Dans ce cas, il faut veiller à être dans le dossier où est stocké le fichier `toto.sci`. On peut s'orienter et changer de répertoire en utilisant les commandes : `pwd` y `cd`. Il y a un éditeur inclus dans SCILAB. Quelques instructions pour démarrer :

- `grand` permet la génération de réalisations indépendantes d'une loi. Par exemple, `x=grand(100,1,'nor',0,1)` génère un vecteur `x` de 100 lignes de gaussiennes standard.
- `clear` efface toutes les variables de l'espace de travail.
- `whos` donne toutes les variables de l'espace de travail.
- `help toto` donne de l'aide sur la commande SCILAB `toto`.
- `save`, `load` permet la sauvegarde et la lecture de variables (voir l'aide).

Un petit exemple : matrices aléatoires

- Distribution semi-circulaire de Wigner

```
//loi de wigner
clear;
n=input('dimension de la matrice');
// generation d'une matrice nxn entrees gaussiennes standard
a=grand(n,n,'nor',0,1);
//symetrisation
asym=(a+a')/sqrt(2);
//normalisation
nasym=asym/sqrt(n);
//calcul des valeurs propres
lam=spec(nasym);
//distribution empirique des valeurs propres
nclass=round(sqrt(n));
h=figure();
histplot(nclass,lam);
//distribution asymptotique
x=-2:0.001:2;
y=(1/(2*\%pi))*sqrt(4-x.*x);
plot(x,y);
//calcul de la moyenne et variance de la distribution empirique
m=mean(lam);
sig=variance(lam);
disp('moyenne empirique',m);
disp('variance empirique',sig);
```
- Distribution de Pastur Marchenko

```

//loi de Pastur Marchenko
clear;
m=input('nombre de ligne de la matrice');
//choisir m>n
n=input('nombre de colonnes de la matrice');
// generation d'une matrice nxp entrees gaussiennes standard
a=grand(m,n,'nor',0,1);
//tranformation en une matrice de Wishart
asym=a'*a;
//normalisation
nasym=asym/n;
//calcul des valeurs propres
lam=spec(nasym);
//distribution empirique des valeurs propres
nclass=round(sqrt(n));
h=figure();
histplot(nclass,lam);
//distribution asymptotique
l=m/n;
s1=(1-sqrt(l))^2;
s2=(1+sqrt(l))^2;
x=s1:0.01:s2;
y1=x.^(-1);
y2=sqrt(4*l-(x-(1+l)).^2);
y3=y1.*y2;
y=(1/(2*pi))*y3;
plot(x,y);
//calculo de la moyenne et variance de la distribution empirique
moy=mean(lam);
sig=variance(lam);
disp('moyenne empirique',m);
disp('variance empirique',sig)
- Loi uniforme sur le tore
//loi uniforme sur le tore
clear;
n=input('dimension de la matrice');
// generation d'une matrice nxn entrees gaussiennes standard
a=grand(n,n,'nor',0,1);
//symetrisation
asym=(a+a')/sqrt(2);
//normalizacion
nasym=asym/sqrt(n);
//calcul des valeurs propres et vecteurs propres
[x,lamb]=spec(nasym);
//calcul des valeurs propres de x
lamb=spec(x);
//partie reelle et imaginaire des valeurs propres
lambr=real(lamb);
lambi=imag(lamb);
//representation dans le plan complexe
f=figure();
plot(lambr,lambi,'+');

```

2 Choix de Modèle

On considère le modèle de régression simple :

$$Y_i := \sum_{j=1}^{10} \theta_j^* x_{ij} + \varepsilon_i, \quad i = 1 \dots N.$$

Où ε_i , $i = 1 \dots N$ sont des gaussiennes centrées indépendantes et de même variance σ_*^2 . On générera ce modèle en utilisant comme regressseurs (x_{ij}) des réalisations indépendantes de variables uniformes sur $[-1, 1]$ qui seront fixées une bonne fois pour toute (générer ces variables avec $N = 750$ puis sauver). On se placera dans le cas où très peu des paramètres θ_j^* sont non nuls (prendre juste 2, 3 ou 4 paramètres non nuls). Mettre en place les techniques de choix de modèles étudiées en cours pour trouver le *meilleur* modèle. On rappelle les différents critères :

– C_p -Mallows

$$C_p(m) = \frac{\widehat{\sigma}_{(m)}^2}{\widehat{\sigma}^2} + 2 \frac{|m|}{N},$$

– AIC corrigé

$$AIC(m) = \log \left(\widehat{\sigma}_{(m)}^2 \right) + \frac{N + |m| + 1}{N - |m| - 3},$$

– BIC

$$AIC(m) = \log \left(\widehat{\sigma}_{(m)}^2 \right) + \frac{\log N}{N} |m|.$$

Afin d'explorer facilement les 1024 modèles on construira une liste aléatoire de 1024[log 1024] modèles obtenue par la stratégie du *collectionneur*!!.

3 Dantzig contre Cramér

Soit le système linéaire $b = Ax$ où x est un vecteur de taille p et s -sparse c'est-à-dire que le nombre de composantes non nulles de x est s . b est de taille n avec $n \ll p$. Pour fixer les idées, on peut par exemple prendre $n = 20$, $p = 75$ et $s = 3, 5, 6$. D'un point de vue théorique, il est suffisant de satisfaire

$$s \leq \frac{n}{2 \log(p/n)}.$$

Il faut d'abord simuler la matrice A . Le plus simple est de prendre un matrice gaussienne avec des entrées standard et de normaliser. Remplir x avec des zéros et choisir au hasard les s composantes non nulles que l'on choisira également au hasard avec la loi gaussienne centrée de variance σ^2 . Reconstruire le vecteur x en minimisant la norme l_1 sous la contrainte $b = Ax$. Pour cela on utilisera la fonction `karmarkar`. Cette fonction minimise une forme linéaire sous une contrainte linéaire. Une astuce consiste alors à écrire, pour $i = 1, \dots, n$, $x_i = x_i^+ - x_i^-$ où x_i^+ y x_i^- sont tous les deux positifs. Ainsi la norme l_1 de x est $\sum_{j=1}^n (x_i^+ + x_i^-)$. Commandes utiles pour cet exercice :

– concatenation de deux matrices a et b en une matrice c

`c=[a' b']` ;

`c=c'` ;

– matrices constantes de uns ou de zéros

`a=ones(10,10)` ;

`b=zeros(10,10)` ;

Pour chaque valeur de s faire 100 essais et calculer le nombre de fois où la reconstruction est correcte.