TP : Mesure empirique – Test de Kolmogorov Smirnov

Ce TP fera l'objet de la rédaction d'un compte rendu par binôme. Le compte rendu doit contenir les programmes développés, les résultats obtenus ainsi qu'une analyse de ces résultats et les éventuelles démonstrations.

Activité 1 : Fonction de répartition empirique – Théorème de Glivenko-Cantelli

Soit $X_1, \ldots, X_n$ un échantillon de loi μ sur $\mathbb{R}$ (et de fonction de répartition F). On appelle F_n sa fonction de répartition empirique.

1. Rappeler la définition de F_n.

2. Convergence de F_n
   a) Construire une fonction $Fn(Xn)$ qui représente la fonction de répartition empirique F_n déterminée à partir d'un échantillon (X_n).

   b) Construire une fonction FnGauss(n,m,s) qui simule un n-échantillon d'une loi normale de paramètres m et s2, représente F_n et la fonction de répartition exacte. Utiliser cette fonction pour un n-échantillon de 10, 100 et 1000 nombres.

   c) Même question avec une loi uniforme sur [0,1], FnUnif.

   d) Même question pour une une loi binomiale de paramètres p et n, FnBinom

Activité 1 : Loi K2 de Kolmogorov Smirnov

1. Construire une fonction DnUnif(Xn) qui détermine $D_n = \sup_{x \in \mathbb{R}} |F_{n(x)} - F(x)|$ avec X_n un n-échantillon suivant une loi uniforme sur [0,1] et F la fonction de répartition de la loi uniforme sur [0,1].

2. A l'aide de la fonction DnUnif :
   a) Construire une fonction LoiDnUnif(N,n) qui calcule N simulations de DnUnif et représente sa distribution.

   b) Pour un échantillon de $n = 10$, $n = 100$, $n = 1000$, calculer N=100 simulations de D_n et représenter la distribution de D_n obtenue.

   c) Quelles propriétés illustrent les résultats obtenus.

3. Construire une fonction DnGauss(Xn) qui détermine $D_n = \sup_{x \in \mathbb{R}} |F_{n(x)} - F(x)|$ avec X_n un échantillon suivant une loi normale centrée réduite et F la fonction de répartition de la loi normale centrée réduite puis reprendre les questions du 4.
   Quelle propriété supplémentaire illustre le résultat obtenu?

4. Comparer la loi empirique et éxacte donnée dans le cours.

Activité 3 : Test d'ajustement de Kolmogorov Smirnov

Ecrire une fonction ks pour le test de Kolmogorov-Smirnov. La fonction prend en entrée un échantillon $(x_i)_i$ et une fonction continue externe, F_0, supposée être la fonction de répartition des v.a. X_i. Les objectifs de cette fonction sont :

1. Elle trie l'échantillon par ordre croissant pour produire la série des statistiques d'ordre. Elle représente graphiquement les points d'abscisse $\frac{i}{n}$, i variant de 1 à n, et d'ordonnée les statistiques d'ordre $X_{(i)}$. Elle superpose au même graphique la représentation de F_0.

2. Elle calcule la distance D_n de Kolmogorov-Smirnov entre la distribution théorique et la distribution observée, puis la statistique K_n du test du même nom avec $K_n = \sqrt{n}\, D_n$.

3. Elle retourne la p-value du test de Kolmogorov-Smirnov, estimée par la somme limitée à n=25 :

$$\text{p-value} = 1 - \left(1 + 2\sum_{i=1}^{+\infty} (-1)^i \exp\left(-2\, i^2\, x^2\right)\right)$$

4. Test de la fonction ks sur des lois usuelles.
   a) Tester 100 n-échantillon de la loi uniforme sur [0,1] avec n=10, 100, et 1000. Calculer le risque de première espèce dans chaque cas.
   b) Tester 100 n-échantillon de la loi normale centrée réduite avec n=10, 100, et 1000. Calculer le risque de première espèce dans chaque cas.
   c) Tester 100 n-échantillon de la loi normale d'espérance m et de variance 1 avec F la fonction de répartition de la loi normale centrée réduite avec n=100. Calculer le risque de deuxième espèce pour différente valeur de m. Tracer ce risque en fonction de m.

Activité 3 : Convergence des quantiles empiriques

Soit μ une mesure de probabilité sur $\mathbb{R}$ de fonction de répartition F continue. Le quantile d'ordre p, noté k_p est défini par $F^{-1}(p)$.

Soit $X_1, \ldots, X_n$ un n-échantillon de loi μ et $X_{(1)}, \ldots, X_{(n)}$ la série des statistiques d'ordre. Le quantile empirique d'ordre p est défini par $X_{([np])}$.

1. Illustrer le fait que, pour $p \in]0; 1[$, $X_{([np])}$ converge vers k_p p.s.

2. Illustrer le fait que, pour $p \in]0; 1[$, $\sqrt{n}\left(X_{([np])} - k_p\right)$ converge en loi vers $N\left(0, \frac{p(1-p)}{f(k_p)^2}\right)$.

3. Choisissons μ égale à la loi exponentielle de paramètre 1. Illustrer la convergence p.s. de $X_{(1)}$ vers 0. A quelle vitesse a lieu cette convergence, c'est-à-dire quelle est la bonne renormalisation v telle que $v(n)\, X_{(1)}$ converge en loi vers une limite non triviale ? Illustrer ce résultat. Comment le généraliser ?

Activité 3 : Test de comparaison d'échantillons de Kolmogorov Smirnov

On utilise le théorème suivant :

Théorème d'homogénéité de Kolmogorov-Smirnov

Soit un n-échantillon (X_i) de fonction de répartition continue F et un m-échantillon (Y_j) de fonction de répartition continue G. On souhaite tester $H_0 = ' F = G '$ contre $H_1 = ' F \neq G '$. On a alors sous H_0 :

$\sqrt{\frac{nm}{n+m}}\ \sup_{\mathbb{R}} |F_{n(x)} - G_{n(x)}|_\infty$ qui converge en loi vers la loi de Kolmogorov.

---

1. Réaliser une fonction ks2(X,Y) effectuant ce test.

2. Tester votre fonction avec des 10- 100- et 1000- échantillons d'une même loi continue.

3. On a relevé les hauteurs en métres d'arbres issus de deux forêts. Tester l'homogénéité des deux forêts.

Forêt 1	23,4	24,4	24,6	24,9	25	26,2	26,3	26,8	26,9	27	27,6	27,7		
Forêt 2	22,5	22,9	23,7	24	24,4	24,5	25,3	26	26,2	26,4	26,7	26,7	26,9	27,4

Activité 4 : Fonction de densité non paramétrique

On note $K(u)$ et h le noyau et le paramètre de lissage de la fonction de densité non paramétrique f_K.

1. Construire une fonction scilab kernel(Xn,K,h,a,b) construisant la fonction de densité f_K sur [a,b] à partir d'un n échantillon.
   - Construire une subdivision de [a,b] de 101 points, $t_0 \dots t_{100}$
   - Calculer pour chaque valeur t_i la valeur de $f_K(t_i)$
   - Dans une même fenêtre, tracer le diagramme en bâton décrivant la loi empirique et sur un autre graphique la courbe représentative de f_K.

2. On note k1 le noyau uniforme, k2 le noyau gaussien.
   - Simuler un n-échantillon suivant une loi normale centrée réduite.
   - Construire une fonction MQE calculant f_K et l'écart quadratique moyen en fonction de K(u) et de h.
   - Etudier l'influence du noyau et de h sur l'estimation de f_K obtenue. Pour caractériser la qualité de réalisation, on calculera l'écart quadratique moyen et on représentera les courbes obtenues.

Correction des scripts :

```
function [D,q] = ks2 (x, y)
// test de kolmogorov Smirnov bilateral
// on s'assure que x et y sont des vecteurs colonnes
  [i,j] = size(x);
  if i==1
    x = x';
    n_1 = j;
  else
    n_1=i;
  end
  [i,j] = size(y);
  if i==1
    y = y';
    n_2 = j;
  else
    n_2 = i;
  end
  n = n_1+n_2;

  //x et y sont maintenant des vecteurs colonne
  V=[x ;y];
  [V_sort, indexes] = sort(V);
  echantillon = [ones(x); 2*ones(y)];
  echantillon = echantillon(indexes);

  // calcul de la statistique de test
  i=1;
  f_1=0;
  f_2=0;
  D=0;
  while %t
    if echantillon(i) == 1
      f_1 = f_1 + 1/n_1;
    else
      f_2 = f_2 + 1/n_2;
    end
    D = max(D, abs(f_1 - f_2));
    if (i <n_1+n_2)
      i=i+1;
    else
      break
    end
  end

  zeta = D * sqrt(n_1*n_2/(n_1+n_2));

  // calcul de la p-valeur
  // estimation du nb de termes à garder dans la somme
  // pour avoir une precision de 10^-5
  N = sqrt(-log(10^(-5))/2) /  zeta;
  k = 1:N;
  q = - 2*sum ((-1).^k .* exp(-2*k.^2*zeta^2));
endfunction
```

```
function T=ksnorm(x,alpha)
% Test de Kolmogorov-Sirmnov contre une loi N(0,1) de niveau alpha
n=length(x);
sx=sort(x);
pn=pnorm(sx);
d=max([max((1:1:n)/n-pn) max(pn-(0:1:n-1)/n)]);
% T=(pks(sqrt(n)*d)>(1-alpha))
T=(pks((sqrt(n)+0.12+0.11/sqrt(n))*d)>(1-alpha))
enfunction


function q=qksnorm2(n,alpha)
% calcul du seul de rejet pour \tilde{D}_n dans un test d'ajustement
% sur une famille gaussienne de Kolmogorov-Smirnov
% estimation sur 10000;
nexp=10000;
d=zeros(1,nexp);
for i=1:nexp
x=randn(1,n);
x=(x-mean(x))/std(x);
sx=sort(x);
pn=pnorm(sx);
d(i)=max([max((1:1:n)/n-pn) max(pn-(0:1:n-1)/n)]);
end
q=quantile(d,1-alpha);
enfunction

function [p]=pks(x)
p=[];
x2 = -2*x.^2;
factor = -2;
// the computation is performed just for x > 0.14
// else value is assumed to be 0
tag = x > 0.14;
p = bool2s(tag)
absterm = 0;
n = 0;
while  or(tag) & ( n<100 ) then
  n = n+1;
  term = factor .* exp(x2 .* (n^2)) .* bool2s(tag);
  p = p + term;
  absterm1 = abs(term);
  tag = bool2s(tag) .* (bool2s(absterm1>0.001 .* absterm)) .* (bool2s(absterm1>0.00000001 .* abs(p)));
  factor = -factor;
  absterm = absterm1;
end
p = p .* (1-tag);
// si non convergence
endfunction


function [d,p]=kstwo(x,y)
d=[];p=[];
%v = x
```

```
if min(size(%v))==1 then %v=sort(%v),else %v=sort(%v,'r'),end
x = %v($:-1:1,:);
%v = y
if min(size(%v))==1 then %v=sort(%v),else %v=sort(%v,'r'),end
y = %v($:-1:1,:);
// sort in ascending order
dx = max(size(x));
dy = max(size(y));
// samples lengths
kx = 1;
ky = 1;
fnx = 0;
fny = 0;
count = 1;
// initialise variables
while (kx<=dx)&(ky<=dy) then
  // generate cumulative distribution
  ddx = x(kx);
  ddy = y(ky);
  // functions
  if ddx<=ddy then
    fnx = kx/dx;
    kx = kx+1;
  end
  if ddy<=ddx then
    fny = ky/dy;
    ky = ky+1;
  end
  dt(1,count) = abs(fnx-fny);
  // difference between functions
  count = count+1;
end
d = mtlb_max(dt);
// maximum of difference
N = sqrt(dx*dy/(dx+dy));
//p=probks((N+0.12+0.11/N)*d);              % calculate probability
p = 1-pks((N+0.12+0.11/N)*d);
// calculate probability
endfunction
```