Rapport de TP MA412

TP2 : Classification

Michel Kot
Mathieu Marleix

Sommaire

Dans tout le TP chaque pays est représenté par un numéro.

Pays	Numéro	Pays	Numéro
Canada	1	Pologne	13
Mexique	2	Portugal	14
USA	3	Espagne	15
Japon	4	Suède	16
Corée	5	Turquie	17
Australie	6	Royaume-Uni	18
République Tchèque	7	Brésil	19
France	8	Russie	20
Allemagne	9	Inde	21
Grèce	10	Indonésie	22
Hongrie	11	Chine	23
Italie	12	Afrique du Sud	24

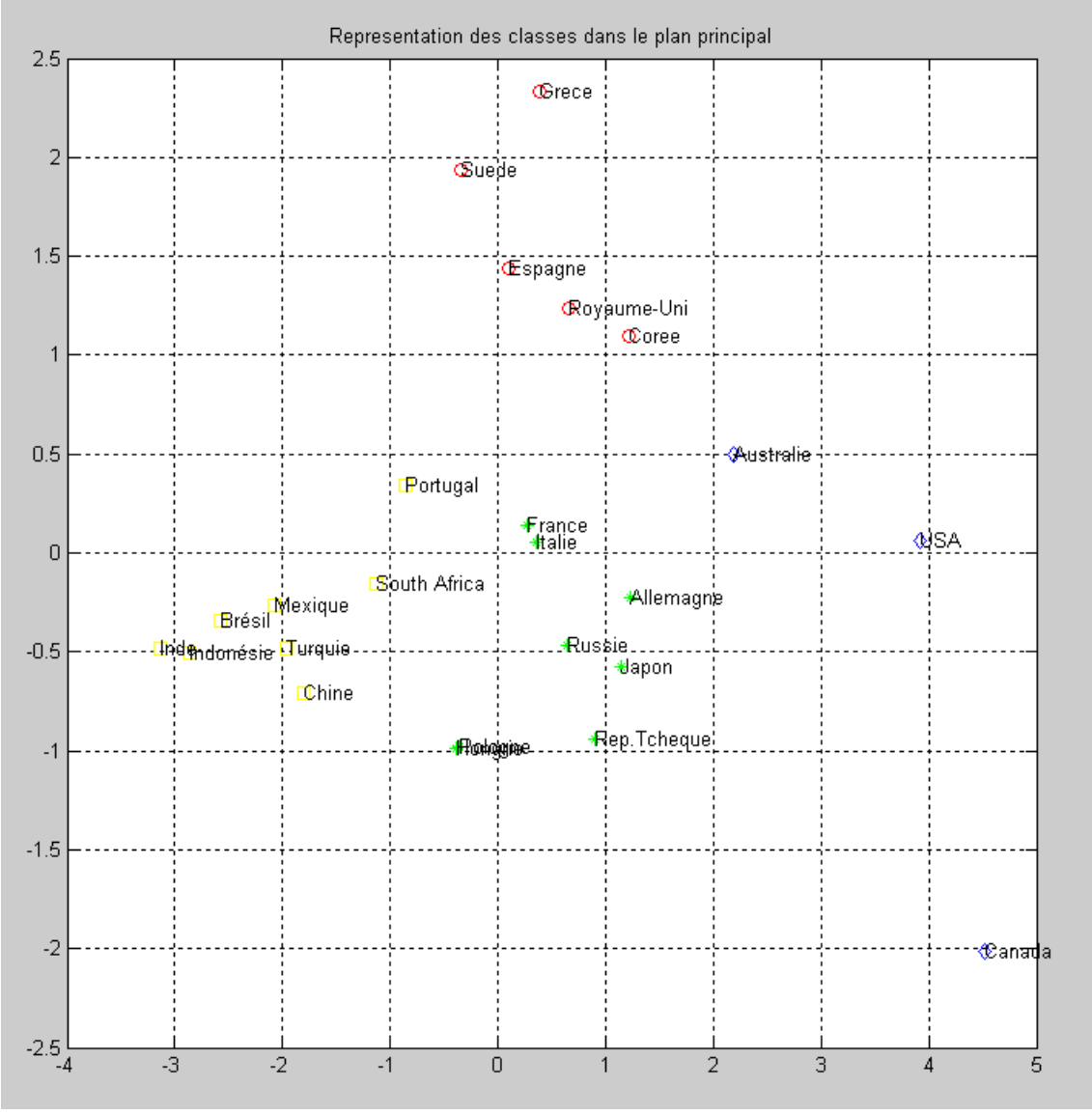
Partie 1 : Algorithme des k-means

Etude de l'algorithme

A l'aide de la fonction *kmeans* de MATLAB, on réalise une classification par l'algorithme k-means des données normalisées. La classification obtenue est la suivante :

Pays	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Classe	2	3	2	4	1	2	4	4	4	1	4	4	4	3	1	1	3	1	3	4	3	3	3	3

A l'aide de la fonction *classr*, nous avons ensuite représenté la classification dans le plan principale.



Les options de cette fonction nous permettent de spécifier la manière dont sont choisis les différents centres initiaux. Nous avons donc testé avec différents paramètres tout en affichant le critère de la somme des inerties pour pouvoir choisir la manière la plus adapté. Nous avons finalement choisi la méthode *sample* se basant sur un choix aléatoire des centres initiaux parmi les points à classifier. La fonction *kmeans* permet également d'afficher les itérations grâce au paramètre *Display* et non simplement le résultat final. De même en passant le paramètre *Replicates*, on réitère l'opération dans le cas d'un mauvais choix de centre.

iter	phase	num	sum
1	1	24	63.5385
2	1	1	61.1179
3	1	1	59.9244
4	2	6	50.2566
5	2	3	45.7642

5 iterations, total sum of distances = 45.7642

iter	phase	num	sum
1	1	24	57.2298
2	1	2	55.1407
3	2	2	46.5656
4	2	1	45.7642

4 iterations, total sum of distances = 45.7642

iter	phase	num	sum
1	1	24	65.6429
2	1	2	60.4918
3	1	2	55.8541
4	2	3	48.3274
5	2	1	48.0752

5 iterations, total sum of distances = 48.0752

iter	phase	num	sum
1	1	24	86.4898
2	1	3	69.6959
3	1	2	62.0461
4	2	4	57.6253
5	2	2	45.7642

5 iterations, total sum of distances = 45.7642

D'après les résultats obtenus, on peut conclure que le critère de la somme des inerties est 45.7642. On remarque également que ce critère est sensible au choix des centres initiaux, car une des quatre valeurs obtenues est différente.

Calcul du critère de la somme des inerties

Nous avons ensuite dût écrire une fonction permettant de calculer le critère de la somme des inerties, le résultat de cette fonction pouvait être vérifié simplement grâce aux résultats précédents renvoyés par *kmeans*.

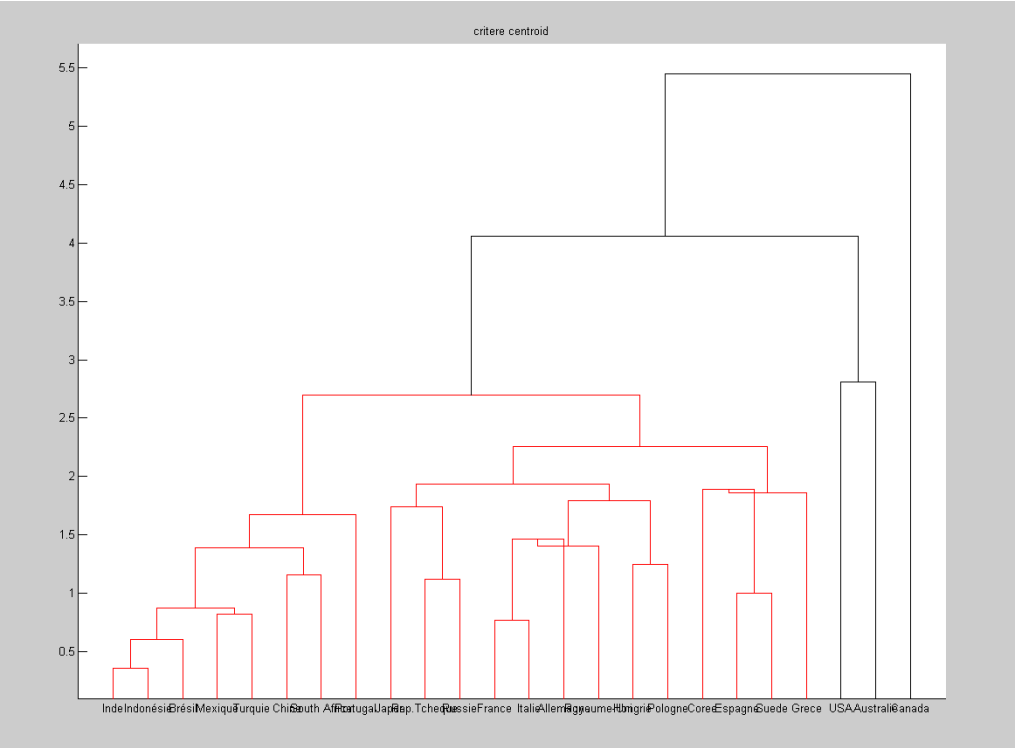
Cette fonction commence par calculer le centre de gravité des classes. Elle calcule ensuite la somme des distances entre chaque point et le centre de gravité pour chaque classe. Puis elle fait la somme de ces sommes, ce qui nous donne le critère.
(Voir annexes pour le code la fonction.)

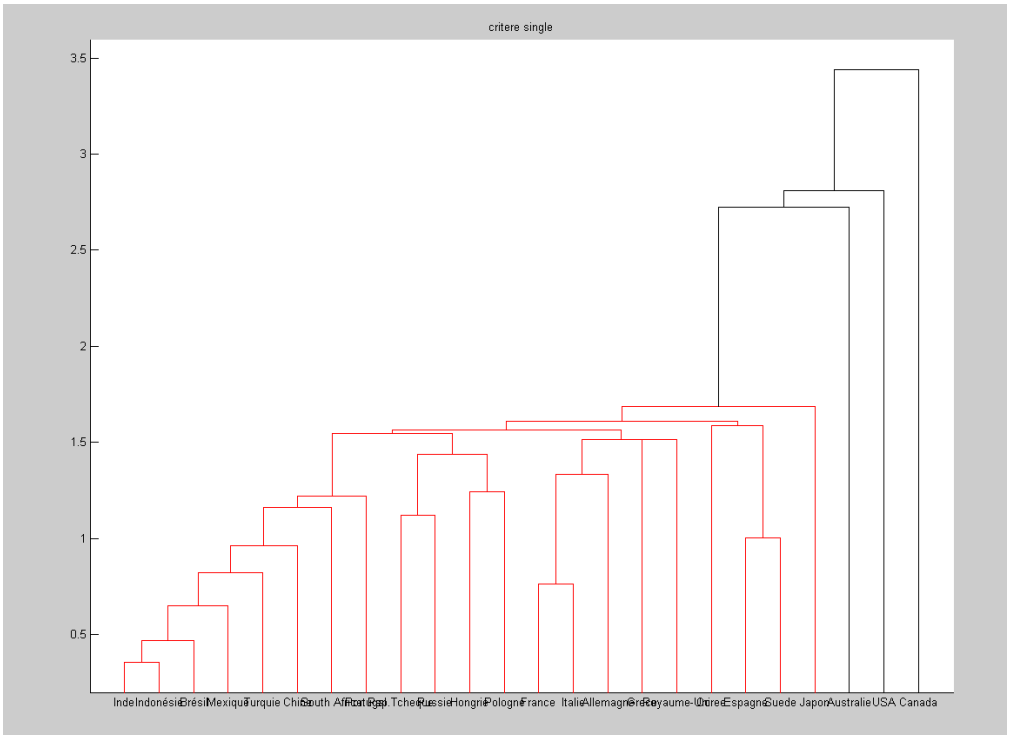
Le critère calculé par notre fonction est 45.7642, ce qui correspond bien à la valeur donnée par *kmeans*.

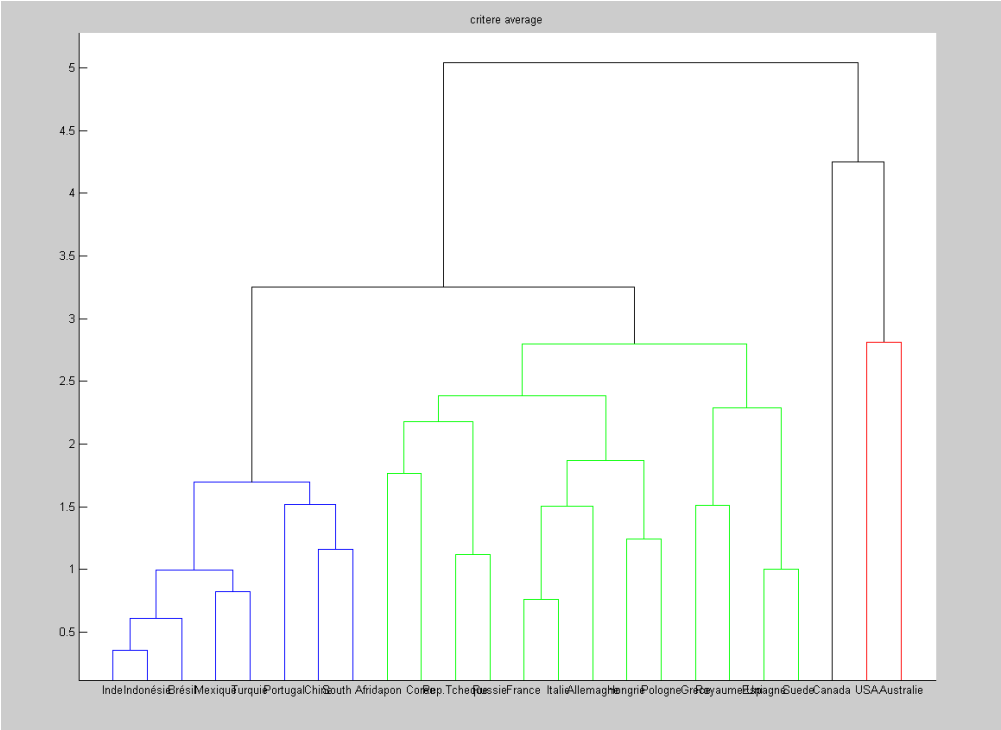
Partie 2 : Classification par construction ascendante hiérarchique

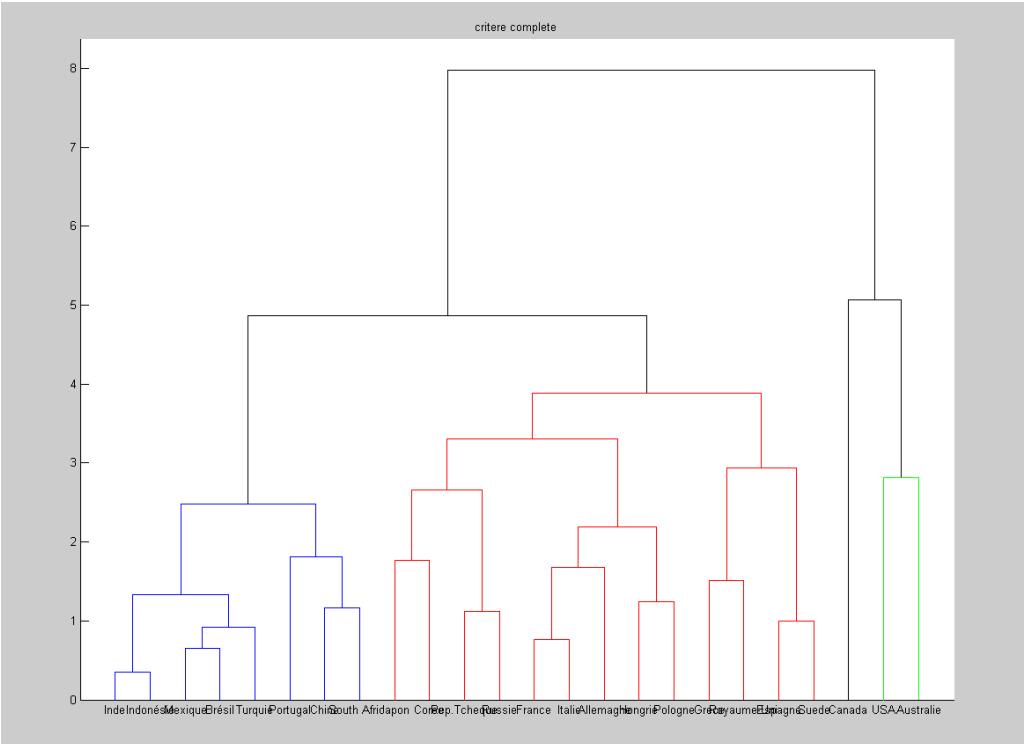
On souhaite maintenant faire une classification par construction ascendante. On commence d'abord par utiliser la fonction *pdist* qui nous donne la matrice de distance entre chaque point. On utilise ensuite la fonction *linkage* qui génère un arbre binaire représentant la classification hiérarchique. Un critère d'agrégation peut lui être spécifié en paramètre.

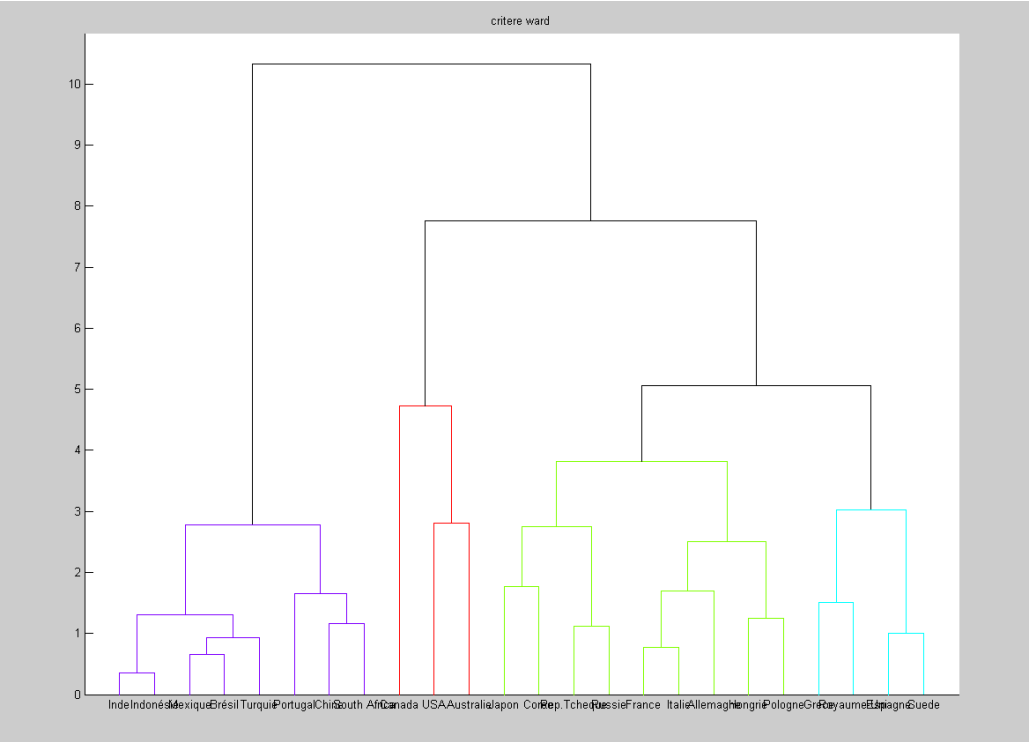
La fonction *dendrogram* nous permet de représenter les dendrogrammes des classifications hiérarchiques pour les différents critères d'agrégations.






critere average
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
Inde
Indonésie
Brésil
Mexique
Turquie
Portugal
Chine
South Afridapon
Corée
Rep.Tchèque
Russie
France
Italie
Allemagne
Hongrie
Pologne
Grèce
Royaume
Espagne
Suede
Canada
USA
Australie


critere complete
8
7
6
5
4
3
2
1
0
Inde
Indonésie
Mexique
Brésil
Turquie
Portugal
Chine
South Afridapon
Corée
Rep.Tchèque
Russie
France
Italie
Allemagne
Hongrie
Pologne
Grèce
Royaume
Espagne
Suede
Canada
USA
Australie



La fonction *cluster* nous donne la classification pour chaque critère d'agrégation.

Classe selon :	Pays																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
centroid	4	3	1	3	3	2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
single	4	2	3	2	2	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
average	3	1	4	2	2	4	2	2	2	2	2	2	2	1	2	2	1	2	1	2	1	1	1	1
complete	3	1	4	2	2	4	2	2	2	2	2	2	2	1	2	2	1	2	1	2	1	1	1	1
ward	3	4	3	2	2	3	2	2	2	1	2	2	2	4	1	1	4	1	4	2	4	4	4	4

En utilisant la fonction *critere*, on calcule la somme des inerties pour chaque critère d'agrégation.
- critère *centroid* : 104.4199
- critère *single* : 104.4199
- critère *average* : 60.9252
- critère *complete* : 60.9252
- critère *ward* : 47.7080

On remarque donc que la classification par l'algorithme k-means est meilleure, et que seule la classification par critère de Ward est presque aussi efficace.

Annexes

Script MATLAB

```
% Algorithme des k-means
% Normalisation des donnees
Xn=(X-repmat(mean(X),size(X,1),1))*inv(diag(std(X,1)));
% Classification et representation
IDX = kmeans(Xn,4)
classr(X, IDX, Pays)
% Critere de la somme des inerties
kmeans(Xn,4,'Start','sample','Replicates',4,'Display','iter');
% Calcul du critère
critere(X,IDX)

% Classification par construction ascendante hierarchique
% Calcul des distances
D = pdist(Xn,'euclidean');
% Critere centroid
Z1 = linkage(D,'centroid');
figure
dendrogram(Z1,'COLORTHRESHOLD',2.8,'LABELS',Pays)
title('critere centroid')
T1 = cluster(Z1,'MaxClust',4)
% Critere single
Z2 = linkage(D,'single');
figure
dendrogram(Z2,'COLORTHRESHOLD',2,'LABELS',Pays)
title('critere single')
T2 = cluster(Z2,'MaxClust',4)
% Critere average
Z3 = linkage(D,'average');
figure
dendrogram(Z3,'COLORTHRESHOLD',3,'LABELS',Pays)
title('critere average')
T3 = cluster(Z3,'MaxClust',4)
% Critere complete
Z4 = linkage(D,'complete');
figure
dendrogram(Z4,'COLORTHRESHOLD',4,'LABELS',Pays)
title('critere complete')
T4 = cluster(Z4,'MaxClust',4)
% Critere ward
Z5 = linkage(D,'ward');
figure
dendrogram(Z5,'COLORTHRESHOLD',5,'LABELS',Pays)
title('critere ward')
T5 = cluster(Z5,'MaxClust',4)
% Calcul des critères
C1 = critere(X,T1)
C2 = critere(X,T2)
C3 = critere(X,T3)
C4 = critere(X,T4)
C5 = critere(X,T5)
```

Fonction critere

```
function [ W ] = critere(donneesX,classificationL)
%UNTITLED Calcul le critere de la somme des inerties
%   Argument donneesX: tableau de donnees
%   Argument classificationL: classification

% Initialisation parametre de retour, matrice des points
% et matrice centre de gravite.

   [n,p]=size(donneesX);
   donneesNormaliseesX = normtab(donneesX);
   k = max(classificationL);
   G = zeros(k,p);
   D = zeros(k,1);

% Remplissage des matrices de points et centres de gravite
   for i = 1 : k
      %Vecteur colonne renvoyant les indices des points appartenant à chacune des classes
      Ik = find( classificationL == i);
      G(i,:) = mean(donneesNormaliseesX(Ik,:));
      for j = 1 : size(Ik)
         %Somme des distances au carree entre les donnees et les centres de gravite
         D(i) = D(i) + sum((donneesNormaliseesX( Ik(j) , : ) - G(i,:)  ).^ 2);
      end
   end
   W = sum(D,1);
end

function [Xn]=normtab(X)

%[Xn]=NORMTAB(X)
%Normalisation du tableau de donnees X

Xn=(X-repmat(mean(X),size(X,1),1))*inv(diag(std(X,1)'));

end
```