Rapport individuel : Elisa Lescarret

Parties réalisées :

* Déterminer les valeurs propres et vecteurs propres associés
* Déterminer la base du plan principale
* Cercle de corrélation et affichage du cercle de corrélation et des projections des caractères
* Partie permettant le lancement de la deuxième ACP

# Déterminer les valeurs propres et vecteurs propres associés

Les valeurs et vecteurs propres à déterminer sont ceux de la matrice de corrélation. Il nous permettrons par la suite à déterminer les composantes.

Pour tout les ACP, la matrice de corrélation est une matrice symétrique avec des 1 sur la diagonale, elle est donc diagonalisable.

Code :

function [vals, base] = calculValeurVecteurBase (matCorr)

[B, D] = spec(matCorr)

//On trie dans l'ordre décroissant les valeurs propres //obtenu avec la fonction spec

[vals, indices] = gsort(diag(D))

//on recupère le nombre de valeurs dans vals

taille = length(vals)

// On initialise la variable base

base = B

//Comme on a trier les valeurs propres il faut mettre

//les vecteurs propres associés dans le même ordre

for i = 1:taille

base (:,i) = B(:,indices(i))

end

endfunction

On récupère donc les vecteurs propres et les valeurs propres grâce à la fonction spec de scilab. Ensuite on trie les valeurs propres dans l’ordre décroissant (l’indice 1 de vals fait référence à la plus grande valeur propre). On réorganise l’ordre des vecteurs associés selon le nouvel ordre des valeurs propres.

Exemple : avec la fonction spec on obtient λ1 , λ2, λ3 avec comme vecteurs associés u1, u2 ,u3 si on a λ1>λ3>λ2 alors on auras vals=[λ1,λ3,λ2] et base = [u1,u3,u2]

# Déterminer la base du plan principale

La base du plan principale dépend de la qualité de représentation. On détermine les valeurs propres qui représentent la variance totale du nuage de points. Ce sont les 2 valeurs propres les plus grandes, d’où l’intérêt de les triés.

# Cercle de corrélation et affichage du cercle de corrélation et des projections des caractères

function afficherCercleCor(listePoints,titre, numFigure)

scf(numFigure);

clf(numFigure);

theta=0:0.1:2\*%pi;

//Affichage des cercles

plot(1\*cos(theta),1\*sin(theta))

//Affichage des coordonnees

taille\_listePoints = size(listePoints,"r");

for i=1:taille\_listePoints

point = listePoints(i,:);

plot(point(1,1), point(1,2) ,'+r','markersize',10)

xstring(point(1,1), point(1,2),string(i));

end

plot(0.5\*cos(theta),0.5 \*sin(theta))

xtitle(titre);

xgrid

endfunction

On récupère une matrice correspondant à la liste des coordonnées des caractères, on les affiches et on affiche un cercle de 1 de rayon et un autre de 0.5 de rayon afin de déterminé plus facilement les caractères mal représentés et les corrélations.

# Lancement de la deuxième ACP

MSansIndMalRepr = [];

for i = 1 : n

normePoint = norm(Q2(i,:));

if (normePoint>=normeMinIncertitude & normePoint<=normeMaxIncertitude) then

MSansIndMalRepr = [MSansIndMalRepr; M(i,:)]

end

end

NouvellesDonnees = [];

for j = 1 : p

norme = norm(listePointCaractere(j,:));

if(norme>=normeMinIncertitude & norme<=normeMaxIncertitude) then // on ne garde que les caractères qui sont aux normes

NouvellesDonnees = [NouvellesDonnees,MSansIndMalRepr(:,j)]

end

end

On retire simplement les caractères et les individus mal représentés. Pour les données que nous avons choisies il n’est pas obligatoire d’enlever les caractères mal représentés vu qu’ils sont tous bien représenté. Cependant pour les individus il est indispensable d’en retirer.

Le fait d’enlever à la fois les individus et les caractères mal représentés, signifie qu’on peut utiliser notre programme pour d’autres données.