Classification Hiérarchique Ascendante

Maxime CHAMBREUIL
maxime.chambreuil@insa-rouen.fr

Table des matières

1 But du TP

L'objectif du TP est de donner les compositions des groupes de travaux tutorés en fonctions des choix d'UV des étudiants.

2 Explication

2.1 Les fonctions

2.1.1 GenMatInd

function matrice = genMatInd (donnees , type)

En entrée :
donnees : matrice des inscriptions aux UV
type : 1 → euclide, 2 → mahalanobis

En sortie :
matrice : Matrice des distances

Cette fonction calcule la matrice des distances entre chaque individu en fonction des UV qu'ils ont en commun et du type de distance.

2.1.2 Afusionner

function [aFusionner , hauteur] = afusionner (clusters , type)

En entrée :
clusters : matrice des distances
type : 1 → minimal, 2 → maximal

En sortie :
aFusionner : indice des clusters à fusionner
hauteur : distance entre les 2 clusters à fusionner

Suivant le type, on cherche le maximum ou le minimum dans la matrice des distances. Puis on retourne cette valeur comme hauteur et les indices de la ligne i et de la colonne j ou on a trouvé cette valeur.

2.1.3 Fusion

function level = fusion (indice , level , nbCluster)

En entrée :
indice : indice des clusters à fusionner
level : niveau courant
nbCluster : nombre de cluster avant la fusion

En sortie :
level : niveau courant avec le nouveau clustering et la nouvelle fusion

On commence par trier les indices : on va mettre le nouveau cluster dans l'indice minimum et enlever l'indice maximum. On met ses indices dans level.merged. On rassemble les clusters, puis on decale les clusters qui sont après celui qu'on doit enlever, pour ne pas avoir de trous dans notre suite de cluster. Enfin, on prend les $(nbCluster - 1)$ premiers clusters que l'on met dans level.cluster.

2.1.4 Nouvelle_distance

function distance = nouvelle_distance (matrice , indiceFusionne , type)

En entrée :
matrice : matrice original des distances
indiceFusionne : clustering
type : 1 → minimal, 2 → maximal

En sortie :
distance : matrice des distances après la fusion

A partir de notre matrice des distances de départ, on va calculer les distances entre tous nos clusters courants. On récupère les indices i des individus dans un premier cluster, puis les indices j des individus d'un deuxième cluster. On forme une matrice des distances (i, j), dans laquelle on ne garde que le maximum ou le minimum suivant le type de distance. Cette valeur est ensuite insérée dans notre matrice des distances comme la distance entre nos 2 clusters considérés. Enfin, on met des "Inf" sur la diagonale.

2.2 Déroulement de l'algorithme

Pour chaque niveau, on commence par calculer les indices à fusionner avec la fonction "aFusionner". On fixe ensuite le **level.height** avec la hauteur. Avec les indices à fusionner, on lance la fusion. On a donc mis à jour **level.merged** et **level.cluster**. Avec notre nouveau clustering, on met à jour **level.distance** à l'aide de la fonction "nouvelle_distance". Enfin, on diminue le nombre de cluster et on ajoute level à level0.

2.3 Résultats

Avec quelques astuces Matlab, nous avons obtenu ce dendrogramme :

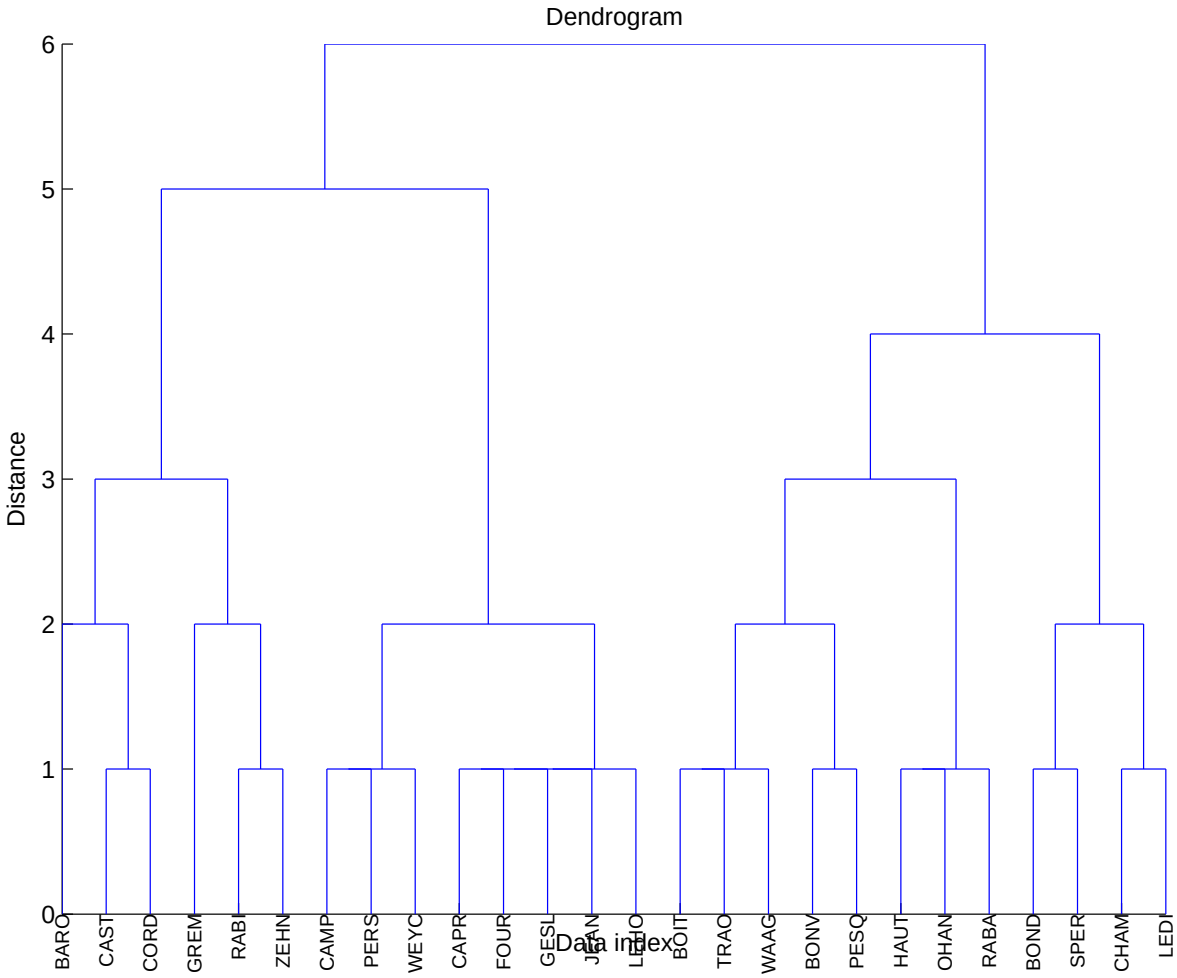

FIG. 1 – Dendrogramme sur les ASI 4

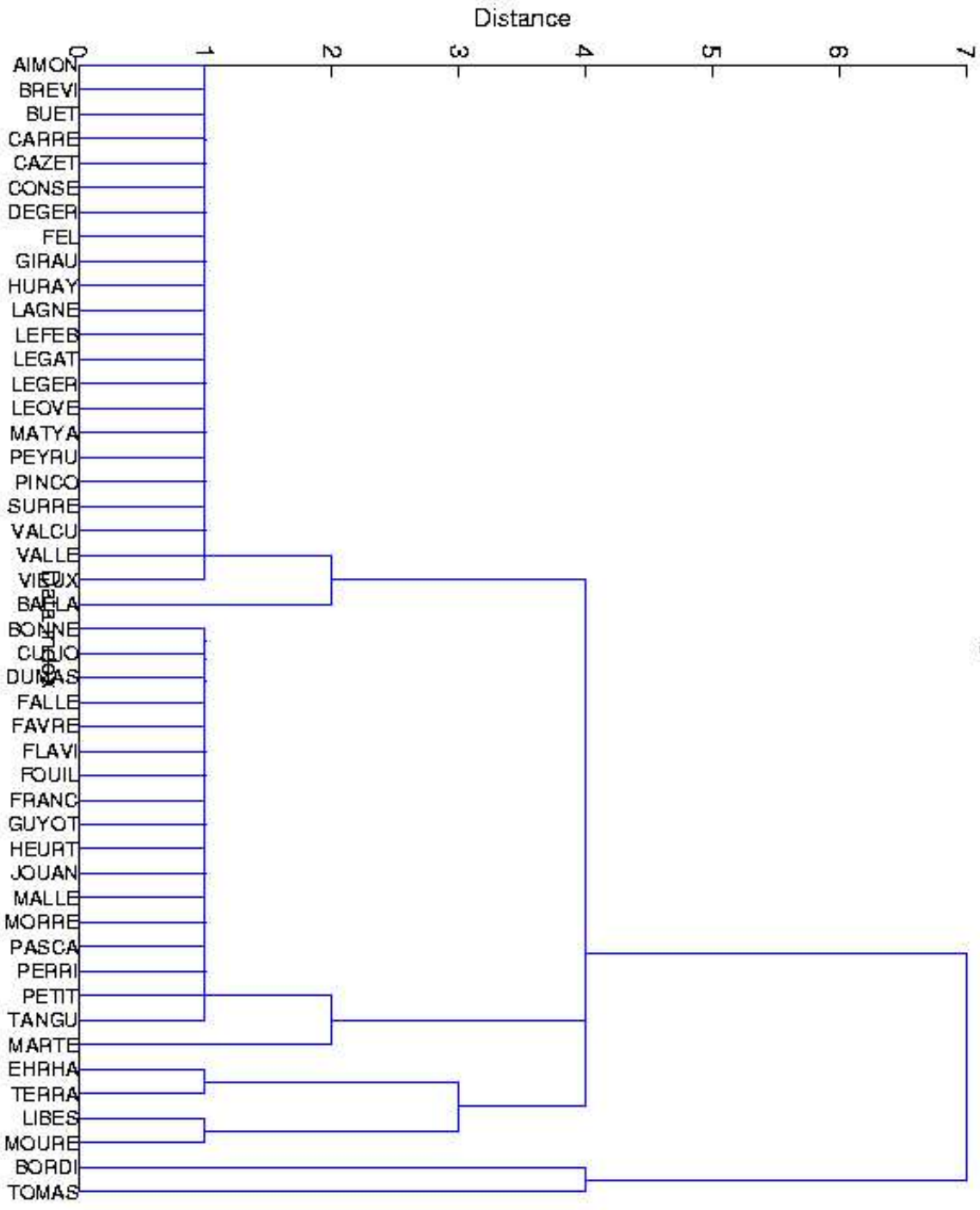

FIG. 2 – Dendrogramme sur les ASI 3

3 Interprétation

3.1 Réponse aux objectifs du TP

3.1.1 Problème du nombre de TT

Si on veut pouvoir mettre en place n'importe quel nombre de groupe de TT, on ne pourra pas se servir du dendogramme précédent : On ne pas tirer un trait horizontal qui coupe 5 clusters, on ne peut pas avoir 5 groupes de TT...

3.1.2 Solution

Lorsqu'on met à jour le level.height, j'ai mis le level.height du niveau précédent avec un petit incrément au lieu de mettre la distance entre les 2 clusters. Voila le résultat :

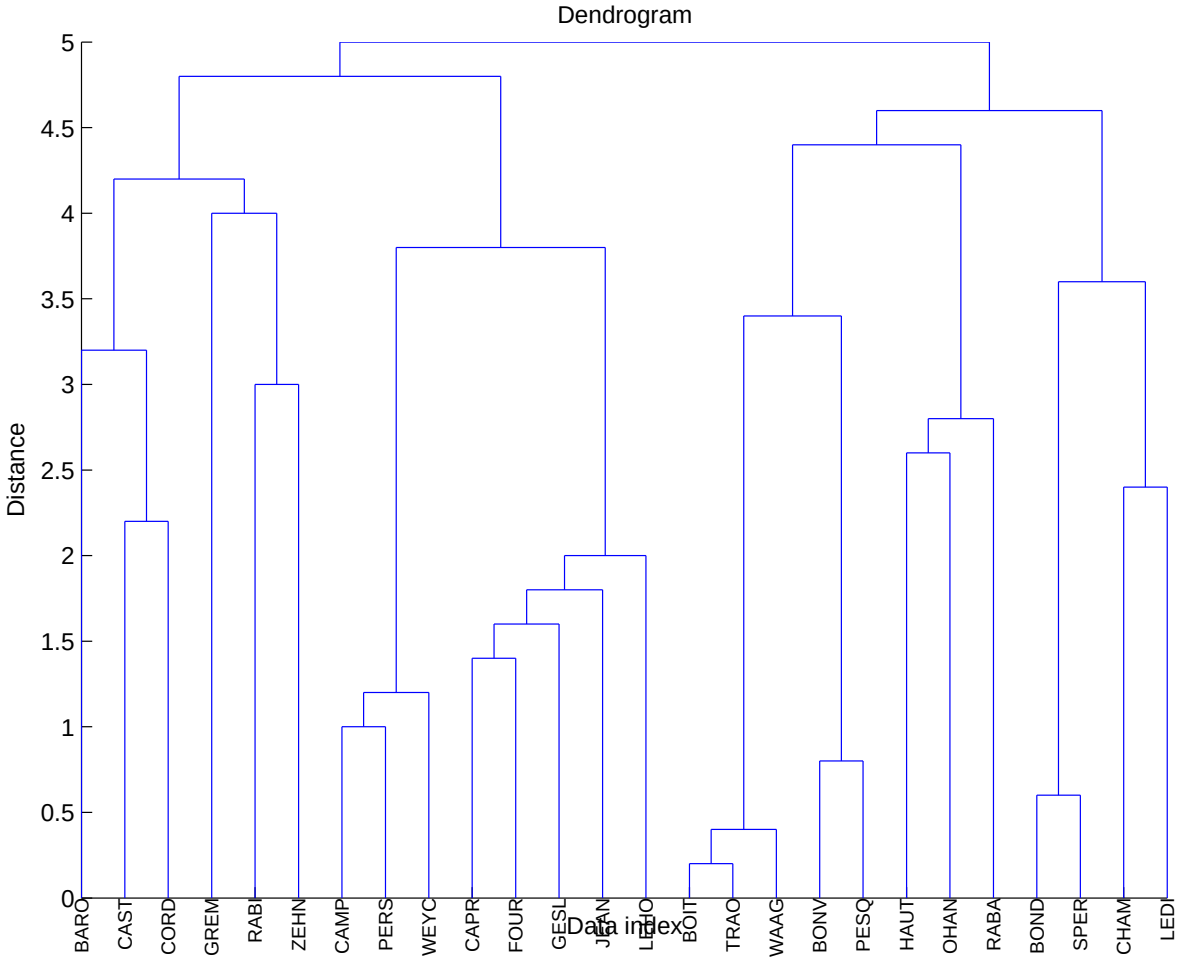

FIG. 3 – "Dendrogramme" sur les ASI 4

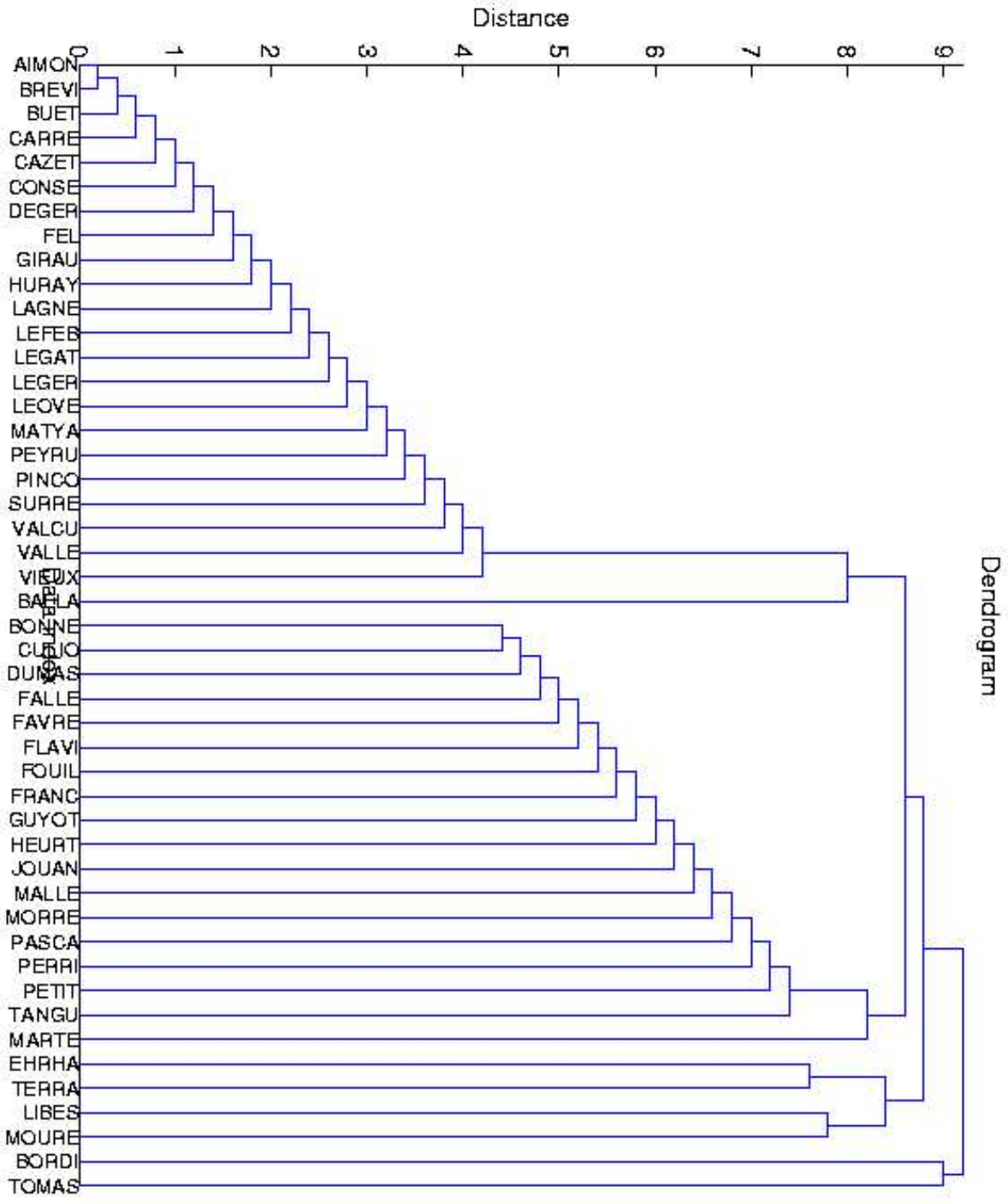

FIG. 4 – "Dendrogramme" sur les ASI 3

On peut ainsi obtenir le nombre de groupe de TT souhaité.

4 Conclusion

Ce TP était très intéressant de part ses objectifs concrets et très proches de nos préoccupations : On peut facilement comprendre et critiquer la constitution des groupes de TT. De plus, le temps de calcul est ridicule à l'échelle d'une promotion.

Cependant, il est resté très compliqué à cause de la manipulation de la structure "level" : on ne savait pas trop comment il fallait former la matrice des distances, ou à quoi correspondaient les 2 indices dans "merged".