Université de Technologie de Compiègne
SY09:Analyse des données et Data-Mining , P17

Rendu TP2

Oumaima Talouka, Zineb Slam

28 avril 2017

Dans ce TP2 nous allons travailler sur la méthode de classification automatique qui est une méthode non supervisée, c'est a dire on a aucune connaissance a priori ni sur le nombre ni sur le nom des classes qui peuvent exister. Nous utiliserons en majorité la méthode des **kmeans** dont nous analyserons et critiquerons les résultats

1 Exercice 1 : Visualisation des données

Nous disposons de 3 jeux de données : **Iris**, **Crabs** et une jeu de données **Mutations** de dissimilarites des espèces

1.1 Iris

Les iris est un jeu de donnes de la librairie *MASS* avec **150 individus** et 4 variables (Sepal.Length, Sepal.Width, Petal.Length et Petal.Width). La variable Z de réponse est l'espèce. Le but est d'identifier l'espèce de chaque individu en fonction des 4 variables : . Pour commencer nous allons centrer et réduire les données puis effectuer une ACP pour pourvoir réduire la dimension du dataset et passer de 4 dimensions et 2 dimensions. Après avoir fait appel a la fonction *prcomp* nous affichons les données dans le premier plan factoriel sans tenir compte de l'espèce. Nous notons d'abord que les 2 premiers plan factoriels expliquent **95.8%** des données *(inertie expliquée)*. Ensuite nous utilisons la fonction *autoplot* pour colorer chaque espèce selon son appartenance a une espèce et ce dans le même plan factoriel.

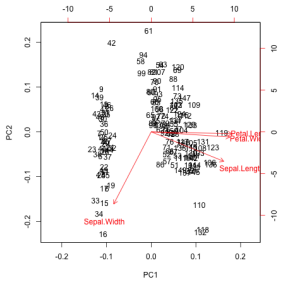

FIGURE 1 – Représentation des données Iris dans les 2 premiers plan factoriel

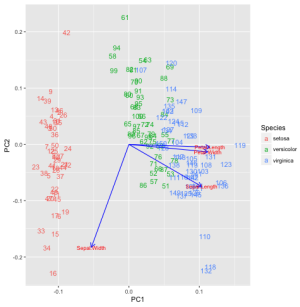

FIGURE 2 – Représentation des données Iris dans les 2 premiers plan factoriel en colorant les espèces

Nous observons donc que selon le graphe qu'il existe au moins 2 classes bien distinctes. Maintenant nous allons faire appel a la fonction *autoplot* en colorant chaque individu selon son espèce 2. On constate qu'en fait il existe 3 classes : **setosa, versicolor et virginica**. Dans la première partie nous avons confondu les classes versicolor et virginica qui sont très proches et donc indistinguable a l'œil nu. L'appartenance a la classe setosa est majoritairement expliquée par la variable Sepal.Widtth, tandis que les 2 autres classes par les 3 autres. Enfin Petal.Length et Petal.Width sont deux variables fortement corrélées On devrait donc s'attendre a 3 classes pour ce jeu de données

1.1.1 Conclusion

1.2 Crabs

Nous allons procéder de la même manière que dans la question précédente mais cette fois pour les données Crabs2. Le jeu de donnes Crabs2 compte 200 individus et 4 variables **(FL2, RW2, CL2, BD2)** et deux variables de réponses le sexe et l'espèce Nous utilisons la fonction *interaction* pour merger le sexe et l'espèce et obtenir une seule variable de réponse z. **91.2%** des données sont expliquées par les 2 premières composantes principales. Les 2 figures ci-dessous représentent les donnes Crabs dans les 2 premiers plan factoriel sans distinguer les espèces (a gauche) puis en les colorant (a droite).

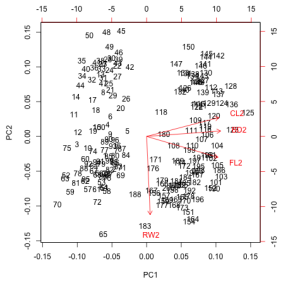

FIGURE 3 – Représentation des données Crabs2 dans les 2 premiers plan factoriel

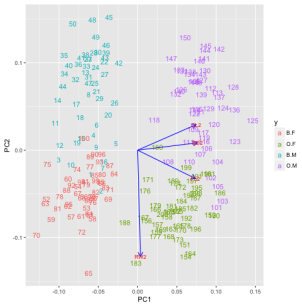

FIGURE 4 – Représentation des données Crabs2 dans les 2 premiers plan factoriel en colorant les espèces

Selon la figure 3 on peut observer qu'in existe au moins 2 classes celle de gauche et celle de droite voir 4 classes si on distingue 2 dans celle de droite et 2 dans celle de gauche. Néanmoins on peut remarquer que ces classes ne sont pas très séparées. La figure 4 montre qu'il existe bien 4 classes B.F, O.F, B.M et O.M. (B et O pour distinguer les 2 espèces alors que , M et F pour le sexe) . La deuxième composante PC2 qui est concentrée dans la variable RW2 permet de distinguer le sexe, tandis que CL2, FL2, BO2 encodées dans PC1 permettent de distinguer l'espèce. On pourrait alors mettre l'hypothèse que le fait que le sexe ne soit pas aussi distinguable que l'espèce est le fait qu'il est explique que par une seule variable tandis que l'espèce par 3.

1.2.1 Conclusion

Pour conclure la l'AFTD nous a permis de passer d'une representation de dissimilarites de dimension 20 a une autre de dimension 6 avec un poucentage d'inertie expliquee de 95%. Nous n'avons pas pu choisir 2 dimensions car la representation n'est pas fiable comme les dissimilarites calculees sont loines des initiales. On peut supposer que le fait que cette methode n'a pas ete aussi performante avec ce jeu de donnes est parce que nos dissimilartites ne sont pas des distances Euclediennes mais les differences entres les aminos acides dans les chaines chromosomique. chaque différence contribue à la distance de mutation en fonction du nombre minimum de nucléotides qui devrait être changé pour convertir l'un dans l'autre. Fitch & Margoliash ont utilisé ces données pour construire un arbre phylogénétique.

1.3 Mutations

Nous disposons dans cette étude d'une matrice de dissimilarites entre 20 individus (Homme, Singe, Kangourou, Cheval...). Grâce a la méthode **AFTD** et la fonction nous allons essayer de réduire la dimension du jeu de donnes. Nous essayerons de trouver au fur et a mesure le bon nombre de variables a choisir. Nous allons alors commencer par réduire le nombre de dimensions a 2. Pour

évaluer cette représentations nous utiliserons le pourcentage d'inertie expliquée cumulée et le Diagramme de Shepard Notons que dans le Diagramme de Shepard un axe représente la disimilaritee calculée pard AFTD alors que l'autre les disimmilarites initiales a partir de la matrice 20x20. Ainsi si la disimilaritee calculée par l'AFTD est exacte le Diagramme de Shepard sera une fonction y=x.

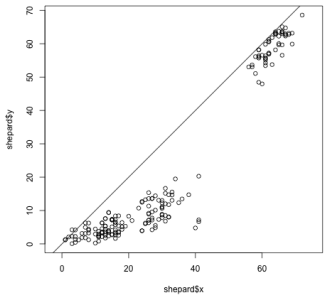

FIGURE 5 – Diagramme de Shepard des Mutations avec 2 variables

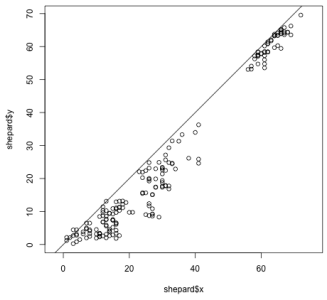

FIGURE 6 – Diagramme de Shepard des Mutations avec 3 variables

On remarque d'après les 2 premières présentations ci dessous que le choix de 2 variables n'est pas très représentatif de nos donnes, en effet les distances calculées par représentations en 2 dimensions restent relativement que celle de 20. Avec 3 variables il y'a plus de données les disimilarites calculées se rapprochent de leur valeur exacte mais il reste toujours d'autres points éloignés Dans la partie suivante nous allons améliorer cette représentation avec 4 puis 5 variables.s

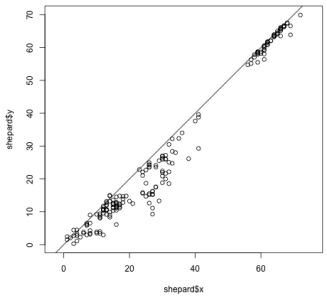

FIGURE 7 – Diagramme de Shepard des Mutations avec 4 variables

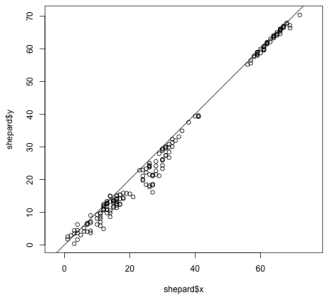

FIGURE 8 – Diagramme de Shepard des Mutations avec 5 variables

Nous remarquons que la représentation avec 4 variables le calcul de certaines disimilarites est exacte mais certaines restent loines des valeurs initiales. Avec 5 variables on se rapproche plus du cas initial ou toutes les disimilarites calculées sont plus proche de la droite y=x. Ces résultats étaient prévisibles si on calcule le pourcentage d'inertie expliquée cumule en mettant l'option $eig = TRUE$ dans la fonction *cmscale* et récupérer les valeurs propres.

1.3.1 Conclusion

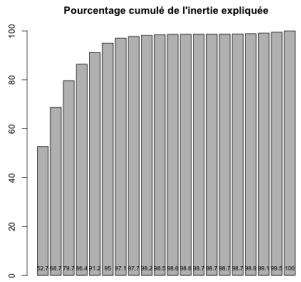

FIGURE 9 – Pourcentage d'inertie cumulee expliquee par AFTD

On remarque bien qu'avec 2 variables seulement 68% et a partir de 4 variables le pourcentage est supérieur a 91%.

2 Exercice 2

3 Exercice 3 : Méthodes des centres mobiles

Le but de cet exercice est de manipuler la méthode des kmeans sur les trois jeux de données Iris2, Crabs2 et mutations et analyser la qualité de la méthode et ses limites. On utilisera l'index de Rand ajustée ainsi que les représentations graphiques pour juger des performances des kmeans.

3.1 Iris2

Nous commençons par classifier nos méthodes en 2, 3 puis 4 classes et les représentant dans en fonction des 2 variables **Petal.Length et Sepal.Width** car comme on a vu dans lq première partie ces dernières sont bien expliquées par les 2 premières composantes principales de l'ACP. Les résultats sont représentés ci-dessous :

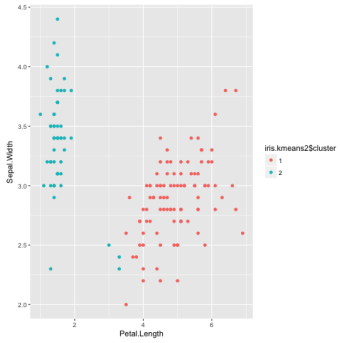

FIGURE 10 – Representation du jeu de donnees de Iris en 2 classes

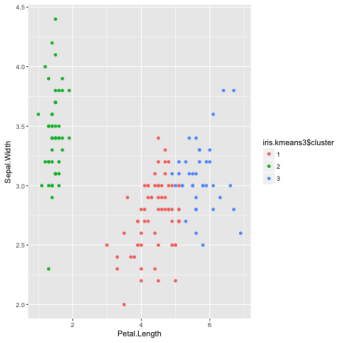

FIGURE 11 – Representation du jeu de donnees de Iris en 3 classes

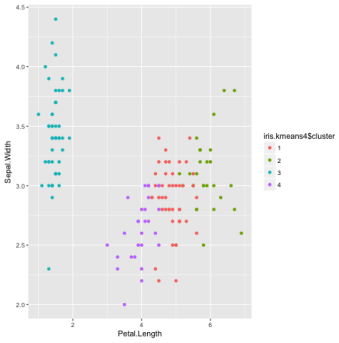

FIGURE 12 – Representation du jeu de donnees de Iris en 4 classes

On remarque qu'avec **k= 2** classes nous obtenons des individus qui appartiennent a la classe 1 alors qu'ils sont plus proche de la classe 2. Tandis qu'avec **k= 3** on a des individus qui appartiennent a une classe mais qu'ils sont très éloignés La représentation avec **k= 4** classes semble plus correcte de plus avec une connaissance a priori des donnes nous savons que c'est le nombre de classe correct.

A présent nous effectuant plusieurs opérations de kmeans avec k=3, et on remarque qu'a chaque opération nous obtenons des résultats différents En effet ceci est normal car a la première itération de l'algorithme, celui ci choisit au hasard 3 points (k=3) appartenant aux individus et va construire a partir de ces points les 3 classes. Pour toujours obtenir le même résultat on peut faire un appel au préalable a la fonction *set.seed(10)* sur R.

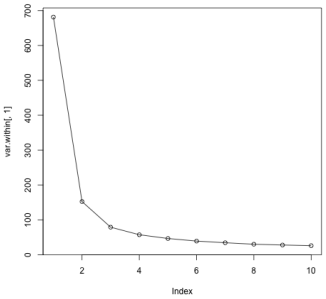

FIGURE 13