

Application JAVA d'analyse de données en utilisant ACP

Réalisé par: EDDAOU M'hamed Issam
Sous la supervision: Pr. NAJI Abdelouahab



I- Présentation du sujet :

L'analyse en composante principale est une technique dont l'objectif principal est de :

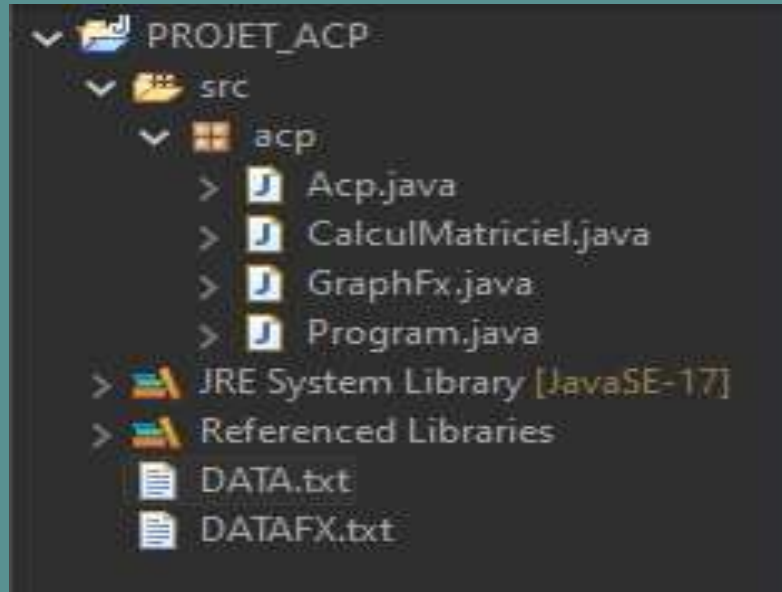
- Réduire la taille de données et dont l'objectif est d'avoir des composantes principales dont le nombre est inférieur ou égale au nombre de variables d'origine.

- Dans le cas où le nombre de composantes principales est inférieur ou égal à 3, nous avons la possibilité de visualiser les données. Dans ce projet, nous allons utiliser JAVA FX pour visualiser les composantes principales pour $n_{comp}=2$ (graphe bidimensionnel).

- L'analyse en composantes principales (ACP) présente aussi un outil extrêmement puissant de synthèse de l'information, très utile lorsque l'on est en présence d'une somme importante de données quantitatives à traiter et interpréter.

II-Conception de l'application:

Notre projet aura la structure suivante:



1-Fichier CalculMatriciel.java

Tout d'abord, Le fichier 'CalculMatriciel' présente les différents méthodes afin de traiter les matrices, on cite:

- ++L'affichage d'une matrice: `'static void printf(double[][])'`
- ++La détermination des valeurs propres et vecteurs propres d'une matrice en utilisant l'algorithme QR.
- ++Et ainsi plusieurs méthodes utiles que vous pouvez consulter la documentation de chacune en commentaires.
- ++Ce fichier déclare aussi deux classes supplémentaire:
- 'Propri_Val_Vect': Afin de structurer notre travail.
- 'MatrixException': Afin de gérer les erreurs.

2-Fichier Acp.java

Le fichier Acp.java contenant la classe ACP de son tour va utiliser les outils fournis du fichier 'CalculMatriciel.java'

Les méthodes principales de la classe ACP on cite:

- Les méthodes de charge de fichier ('uploadData' et 'uploadDataFx'), qui prennent en paramètre soit le nom du fichier(cas uploadData), ou bien un paramètre de type File qui sera choisi à l'aide de l'interface graphique javaFx(cas uploadDataFx).

-Les méthodes de centrer et réduction des données(`void centrer_reduire()` et `void centrer()`) qui vont centrer ou centrer et réduire la matrice des données selon le choix de l'utilisateur....

-La méthode (`double[][] matricecov_coor()`) qui fait appel à la méthode (`static double cov_coor(double[] a, double[] b)`) afin de calculer et trouver la matrice de variance-covariance.

-La méthode (`Propr_Val_Vect cov_corpropresget()`) qui va retourner une instance contenant les valeurs et vecteurs propres; en utilisant la méthode (`CalculMatriciel.decompositionpropres()`) fournie par la classe `CalculMatriciel`.

-La méthode (`double[][] composantsprincipalesextract(int , Propr_Val_Vect)`), qui va retourner les axes des composantes principales en prenant comme paramètre un entier qui signifie le `n_comp` qui va être retenue, ainsi une instance 'Propr_Val_Vect' qui présente les valeurs et vecteurs propres de la matrice de variance-covariance.

-La méthode (`static double[][] analysecomposantsprincipales(double[], int)`) qui va tout simplement effectuer la transformation ACP en fonction du `n_comp` retenue.

-Enfin, la méthode (`double[] quantite_dinfo(double[])`) qui va retourner le pourcentage d'information retenue en chaque axe.

3-Fichier Program.java

Pour la classe Program qui sera exécutée:

La classe va tout d'abord demander à l'utilisateur de saisir le nom du fichier, qui va être stockée comme matrice double[][] (Acp.uploadData())

Après, l'utilisateur doit aussi saisir le nombre de composantes(axes) à retenir.

Ainsi, l'utilisateur aura le choix de soit centrer la matrice ou bien centrer et réduire.

En fonction du choix, et par une boucle switch,case les données seront soit centrées ou bien centrées et réduites, et la matrice de covariance ou la matrice de corrélation sera calculée(en fonction du choix).

→D'où, les étapes de l'analyse de composantes principales seront effectuées.(les valeur propres et vecteurs propres associés/Identification des composantes principales/Projection des données sur les nouveaux composantes principales/Affichage du pourcentage d'information)



4-Fichier GraphFx.java

Concernant la partie application graphique réalisée par JAVAFX:

Tout d'abord, la première fenêtre va nous permettre à partir du bouton 'parcourir' à charger le fichier .txt des données, ainsi la méthode (Acp.uploadDataFx) nous permettra de convertir le fichier vers une matrice double[][], on trace alors le nuage de points concernant les données initiales par la méthode ScatterChart implémentée par JavaFx qui permet de manipuler des graphes bidimensionnels de points.

Alors, de la même façon que l'application console, les données seront centrée et réduites, en plus des étapes de l'analyse ACP, nous aurons le nuage de points x_PROJ qui sera tracé afin de comparer les deux nuages de points.

III-Réalisation de l'application:

On passe alors à l'exécution du programme, il existe deux interfaces d'exécution:

1-Interface console:

Il suffit de se référer à la classe
Program du package acp.

Démonstration →



2-Interface graphique:

On exécute tout simplement le fichier
GraphFx.java, et sélectionner le fichier .txt.

Démonstration →



IV- Conclusion:

Pour conclure, l'ACP présente un outil très puissant de visualisation des données, son implémentation en JAVA s'avère très intéressante.



Merci

Pour votre attention !

