TD 5
Analyse en composantes principales

L'objectif de ce TP est d'étudier l'Analyse en Composante Principale telle que vous l'avez vue en cours. Nous ferons également le lien avec les valeurs propres et vous manipulerez cette technique sur des données réelles.

Exercice 1. Observation des données

Pour cette première partie, nous allons utiliser le jeu de données `iris` qui est classiquement utilisé pour les outils d'apprentissage automatique supervisés ou non-supervisés.

Ce jeu de données contient 150 fleurs classées en trois types d'iris. Ces fleurs sont caractérisées par 4 dimensions: la longueur et largeur du sépale, ainsi que la longueur et largeur du pétale. Nous allons étudier les contributions respectives de ces dimensions pour la catégorisation des iris en types.

1. Représenter les nuages de points en 2D correspondant aux différentes caractéristiques des fleurs. Vous pourrez utiliser l'extrait de code ci-dessous en modifiant les indices correspondant aux caractéristiques.

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import datasets

#affichage du nuage de points en 2D
def nuage(x1, x2, y, dim1='dim1', dim2='dim2'):
    plt.scatter(x1, x2, c=y)
    plt.title('nuage de points')
    plt.xlabel(dim1)
    plt.ylabel(dim2)
    plt.show()

#recuperer le jeu de données iris qui contient plusieurs types d'iris (label) représenté chacu
iris = datasets.load_iris()
X = iris.data
label = iris.target
dim = ['Sepal length', 'Sepal width', 'Petal length', 'Petal width']

nuage(X[:,0], X[:,1], label, dim[0], dim[1])
```

2. En première analyse, est-ce que vous observez des relations entre les différentes caractéristiques ? est-ce que certaines semblent être plus importantes que d'autres pour discriminer les types d'iris ?

3. Déterminer les moyennes, écart-types de chaque dimensions, ainsi que la matrice de variance-covariance.

   Soit un vecteur représentant l'échantillon i à D dimensions (dans le cas des iris $D = 4$ avec $N = 150$ échantillons) $\mathbf{x_i} = (x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{iD})$.

On rappelle pour la dimension j, les définitions de la moyenne $\bar{x_j}$, de la variance $\text{Var}(x_j)$ et de la covariance $\text{Cov}(x_j, x_k)$

$$\bar{x_j} = \frac{1}{D}\sum_{i=1}^{N} x_{ij}$$

$$\text{Var}(x_j) = \left(\frac{1}{D}\sum_{i=1}^{D} x_{ij}^2\right) - \bar{x_j}^2$$

$$\text{Cov}(x_j, x_k) = \left(\frac{1}{D}\sum_{i=1}^{D} x_{ij}x_{ik}\right) - \bar{x_j}\bar{x_k}$$

4. Vous vérifierez que la diagonale de la matrice de variance-covariance, qu'on notera C correspond bien aux valeurs de variance calculées. Si ce n'est pas le cas, lire attentivement la document numpy sur les fonctions `numpy.cov()` et `numpy.var()`.

5. Déterminer la matrice de corrélation R, telle que $r_{jk} = \rho(\bar{x_j}, \bar{x_k} = \dfrac{\text{Cov}(x_j, x_k)}{\sigma_j \sigma_k}$

6. Vous vérifierez également que votre matrice R est bien symétrique et que vous avez bien des 1 sur la diagonale.

Exercice 2. Calcul de l'inertie

L'objectif de l'analyse en composante principale est de repérer et de représenter des échantillons semblables par rapport aux différentes caractéristiques.

1. Soit deux échantillons (deux fleurs) $\mathbf{x_i}$ et $\mathbf{x_k}$. Calculer la distance euclidienne entre ces deux échantillons:

$$d^2(\mathbf{x_i}, \mathbf{x_k}) = \sum_{j=1}^{N}(x_{ij} - x_{kj})^2$$

Remarque: la distance entre deux échantillons identiques ($\mathbf{x_i}$ et $\mathbf{x_i}$) est nulle !

2. On appelle centre de gravité, le vecteur contenant l'ensemble des moyennes pour les D dimensions. Nous avons déjà calculé ces moyennes et $\bar{\mathbf{x}} = (\bar{x_1}, \ldots, \bar{x_j}, \ldots, \bar{x_D})$. L'inertie totale est la moyenne des distances entre chacun des N échantillons $\mathbf{x_i}$ (chaque fleurs) et le centre de gravité.

$$I_g = \frac{1}{N}\sum_{i=1}^{N} d^2(\mathbf{x_i}, \bar{\mathbf{x}})$$

3. Que se passe-t-il si les données sont normalisées ? c'est-à-dire que chaque valeur de chaque échantillon i sera normalisé par la moyenne calculée sur la dimension j et l'écart-type $\sigma_j = \sqrt{\text{Var}(x_j)}$:

$$xn_{ij} = \frac{x_{ij} - \bar{x_j}}{\sigma_j} \tag{1}$$

Ainsi le nouveau vecteur de moyennes devrait être nul et le nouveau vecteur d'écart-type doit être égal à 1.

4. Dans le cas où les données d'entrée sont normalisées, l'inertie totale doit être égale au nombre de dimension (soit 4 dans notre cas).

Exercice 3. Inertie et valeurs propres

Nous allons à présent calculer les composantes principales. Pour cela, deux méthodes peuvent être utilisées:

- la recherche itérative des axes permettant de conserver un maximum d'inertie,
- l'utilisation des valeurs propres.

Nous allons utiliser la seconde méthode.

Rappels: Soit une matrice A, les valeurs propres de A sont les scalaires λ tels que $\det(A - \lambda I) = 0$ avec I la matrice identité.
Le vecteur propre X associé à la valeur propre λ vérifie la relation: $AX = \lambda X$

1. Soit la matrice A suivante. Déterminer ses valeurs propres.

$$A = \begin{pmatrix} 5 & -3 \\ 6 & -4 \end{pmatrix} \tag{2}$$

2. Déterminer ses vecteurs propres.

   Théorèmes: Si une matrice A carrée de dimension n possède n valeurs propres distinctes, alors elle est diagonalisable. De plus, une matrice symétrique est toujours diagonalisable.

   Soit une matrice A carrée symétrique de dimension n, elle est diagonalisable, c'est-à-dire qu'elle peut s'écrire $A = P^{-1}DP$ avec D la matrice diagonale, et P la matrice de passage. Soit λ_k les valeurs propres et X_k les vecteurs propres associés de la matrice A, alors la matrice diagonale D s'exprime en fonction des valeurs propres de A et la matrice de passage s'exprime en fonction des vecteurs propres associés.

$$D = \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \lambda_n \end{pmatrix} \quad \text{et} \quad P = \begin{pmatrix} X_1 & X_2 & \dots & X_n \end{pmatrix}$$

   Remarque: la matrice de passage permet de faire un changement de base. C'est-à-dire que les dimensions de A vont représenter par exemple les caractéristiques des fleurs, tandis que les dimensions de D (A représentée dans la nouvelle base) seront différentes.

3. Donner la matrice diagonale de A, ainsi que la matrice de passage P.

4. La matrice de variance-covariance C (et la matrice de corrélation R) est symétrique donc diagonalisable. Pour calculer ses valeurs propres, il faut résoudre le système suivant $\det(C - \lambda I_4) = 0$ qui est un système à 4 équations.

   Pour se faire, nous allons utiliser la fonction `numpy.linalg.eig(A)` qui retourne les valeurs propres et les vecteurs propres d'une matrice A

   Donner les valeurs propres et vecteurs propres de la matrice C puis de R.

5. On montre que les inerties obtenues suivant chacun des nouveaux axes sont les valeurs propres de R classées par ordre décroissant. Calculer la somme des inerties locales et comparer avec l'inertie globale.

6. Calculer les contributions de chacun des axes: rapport entre inertie locale et globale.

7. Nous obtenons 4 nouveaux axes, correspondant à des combinaisons linéaires des caractéristiques des fleurs. Selon vous quel est l'axe qui représente le mieux ces caractéristiques ?

8. Nous souhaitons à présent obtenir les coordonnées de chaque échantillon (ou fleur) sur les nouveaux axes. Nous utilisons pour cela la matrice de passage P formée de l'ensemble des vecteurs propres. Donner la matrice P.

9. Cette matrice de vecteur propres (ou matrice de passage) est très importante pour l'interprétation. En effet, elle nous indique les contributions de chaque caractéristique d'origine sur le nouvel axe.

   Par exemple, vous devriez trouver pour la première ligne de P les valeurs suivantes:

   ```
   0.52237162 -0.37231836 -0.72101681  0.26199559
   ```

   Cela veut dire que le nouvel axe $c_1 = 0.52x_1 - 0.37x_2 - 0.72x_3 + 0.26x_4$.

   Indiquer alors pour chaque nouvel axe c_j quelles sont les caractéristiques des fleurs qui contribuent le plus

10. Projetez vos données d'origine dans la nouvelle base (sur les nouveaux axes), pour cela, une simple opération matricielle sera suffisante. Et représenter les données sous forme d'un nuage de points sur les axes principaux.

Exercice 4. Réaliser une PCA avec scikit-learn

Nous allons utiliser la famille de fonctions contenue dans le paquet PCA de scikit-learn. Pour cela commencez par lire la documentation en ligne: `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`.

Ensuite importer le toolkit et tester les lignes suivantes avec `X` votre variable contenant les données.

```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
Xp = pca.fit_transform(X)
```

1. Expliquez à quoi correspond la variable `pca`.

2. A votre avis à quoi correspond la valeur de 2 (`nb_components =2`)? Ce choix est-il pertinent au vu de nos données ?

3. Représenter les données sous forme d'un nuage de points dans le plan formé par les nouveaux axes. Comparer avec la représentation obtenue précédemment, vous devriez observer la même chose.

4. Observer les valeurs obtenues avec les attributs suivants, et faites le lien avec ce que vous avez trouvé aux questions précédentes.

   ```
   pca.explained_variance_ratio_
   pca.explained_variance_
   pca.components_
   ```