Université de Sherbrooke
Département d'informatique

IFT870/BIN710
Forage de données / Forage de données pour la bio-informatique

Hiver 2023

Examen final

Professeure :
Aïda Ouangraoua

Le vendredi 21 avril 2023
de 09 h 00 à 12 h 00

Remise de la première partie écrite avant de quitter la salle d'examen.
Remise de la seconde partie pratique avant le lundi 24 avril 2023 à 23h59 sur Turnin.

Toute documentation est permise.

Cet examen est à faire de façon individuelle. Lors de la correction, la note zéro (0) sera attribuée à tout travail pour lequel une preuve de plagiat est attestée. L'examen comporte quatre questions sur cinq pages (+ une annexe de quatre pages + une page additionnelle, pour un total de dix pages). Vous devez répondre à la question 1) directement sur le questionnaire que vous remettrez avant de quitter la salle d'examen. Pour soumettre le reste du travail, vos réponses aux questions 2), 3), et 4), vous devez vous connecter, dans un fureteur, au serveur **http://turnin.dinf.usherbrooke.ca** en utilisant votre CIP, puis choisir le cours **IFT870 (BIN710)** et le projet **ExamenFinal**, charger votre fichier **Examenfinal.ipynb** et le soumettre. Le nom du fichier de remise doit être exactement **Examenfinal.ipynb**.

DÉBUT

Nous avons vu en cours quatre principaux critères de qualité pour l'évaluation de clustering en utilisant des méthodes intrinsèques : cohésion, séparation, connectivité, et robustesse. Plusieurs méthodes d'évaluation intrinsèques basées sur les critères de cohésion, séparation et connectivité existent dans la bibliothèque scikit-learn, mais aucune méthode basée sur le critère de robustesse.

Question 1 : Définition de la cohésion, la séparation et la connectivité (30 pts)

(1.A) Définissez les notions de cohésion, séparation et connectivité d'un clustering.

(1.B) Donnez un exemple de mesure de cohésion, de mesure de séparation, et de mesure de connectivité pour un clustering.

(1.C) Dessinez des exemples de clusterings, pour des données représentées par des points dans le plan cartésien, qui satisfont les propriétés suivantes :

- Forte cohésion, Faible séparation, Faible connectivité

- Faible cohésion, Forte séparation, Faible connectivité

- Faible cohésion, Faible séparation, Forte connectivité

Dans la stratégie d'évaluation de clustering basée sur la robustesse, on distingue deux approches pour évaluer le résultat d'un modèle de clustering :

1. Faire varier la valeur des paramètres de l'algorithme et évaluer la similarité entre les résultats ;
2. Ajouter du bruit aux données et évaluer la similarité entre les résultats.

Vous devez implémenter des méthodes d'évaluation intrinsèques basées sur ces deux approches pour les algorithmes de clustering KMeans et DBSCAN. Pour tester vos méthodes d'évaluation, vous utiliserez l'ensemble de données d'images de visages de personnes connues (Labeled Faces in the Wild, LFW), limité à 40 images maximum par personne, soit 1916 images de dimension 87x65=5655). Les détails du jeu de données LFW sont disponibles à l'adresse http://vis-www.cs.umass.edu/lfw/. Les instruction pour récupérer le jeu de données de départ sont fournies, en annexe, à la fin de l'énoncé d'examen.

Soient $C_1,\ldots,C_N$, N clusterings obtenus en faisant varier les paramètres d'un algorithme de clustering, ou en ajoutant du bruit aux données. Chaque clustering C_i est un partionnement des données. Le score de robustesse $R=R(C_1,\ldots,C_N,)$ est calculé comme suit :

$$R = \frac{\sum_{(i,j)\in P} \#(i,j)}{|P| * N}$$

tel que P est l'ensemble des paires d'objets appartenant au même cluster dans au moins un des clustering $C_1,\ldots,C_N$; $|P|$ est la taille de P ; Pour une paire (i,j) dans P, $\#(i,j)$ est le nombre de clusterings dans lesquels i,j appartiennent au même cluster. Par exemple, si $C_1,\ldots,C_N$ sont tous égaux, alors R=1 car toutes les paires (i,j) de P appartiennent au même cluster dans tous les clusterings $C_1,\ldots,C_N$, donc $\#(i,j) = N$.

Question 2 : Robustesse aux changements de paramètres des algorithmes KMeans et DBSCAN (30 pts)

(2.A) Écrivez **une fonction prenant en paramètre une instance de la classe KMeans** et retournant la robustesse de cette instance, calculée comme suit :
Faire varier uniquement le paramètre n_clusters de l'instance en lui additionnant les valeurs [-5,-4,-3,-2,-1,0,1,2,3,4,5]. Pour chaque valeur du paramètre n_clusters, entraîner le modèle et prédire un clustering. Calculer le score de robustesse *R* correspondant aux 11 clusterings obtenus.

Calculer la robustesse de l'algorithme KMeans(n_clusters=k, random_state=0) pour k = 40, 60 et 80. Quel est le modèle le plus robuste suivant le score *R ?*

(2.B) Écrivez **une fonction prenant en paramètre une instance du modèle DBSCAN**, et retournant la robustesse de cette instance, calculée comme suit :
Faire varier uniquement le paramètre eps de l'instance en lui additionnant les valeurs [-0.5,-0.4,-0.3,-0.2,-0.1,0,0.1,0.2,0.3,0.4,0.5]. Pour chaque valeur du paramètre eps, entraîner le modèle et prédire un clustering. Calculer le score de robustesse *R* correspondant aux 11 clusterings obtenus.

Calculer la robustesse de l'algorithme DBSCAN(min_samples =3, eps=e) pour e = 7, 8 et 9. Quel est le modèle le plus robuste suivant le score *R ?*

Question 3 : Robustesse à l'ajout de bruit des algorithmes KMeans et DBSCAN (30 pts)

(3.A) Écrivez **une fonction prenant en paramètres une instance de la classe KMeans ou DBSCAN et un entier X de valeur comprise entre 0 et 100 représentant un pourcentage** et retournant la robustesse de cette instance, calculée comme suit :
Générer aléatoirement 10 ensembles de données bruitées contenant chacun X*1916/100 données de la même forme que les données initiales (vecteur de 5655 dimensions) suivant la loi normale $N(\mu, \sigma^2)$ pour chaque dimension telle que μ est la moyenne de la dimension et σ^2 sa variance (utiliser numpy.random.randn par exemple). Considérer un 11^e ensemble de bruit qui est vide, i.e. contenant 0 données. Itérer 11 fois, et pour chaque itération, ajouter aux 1916 données un des 11 ensembles de bruit générés, puis entraîner le modèle et prédire un clustering. Calculer le score de robustesse *R* correspondant aux 11 clusterings obtenus, réduits aux données initiales.

(3.B) Calculer la robustesse de l'algorithme KMeans(n_clusters=k, random_state=0) pour k = 40, 60 et 80, pour une valeur X = 5. Quel est le modèle le plus robuste suivant le score *R ?*

(3.C) Calculer la robustesse de l'algorithme DBSCAN(min_samples =3, eps=e) pour e = 7, 8 ou 9, pour une valeur X = 5. Quel est le modèle le plus robuste suivant le score *R ?*

Question 4 : Modèle pour la génération du bruit (10 pts)

Proposez une méthode pour l'ajout de bruit dans les données, différente de celle décrite à la question (3.A). Justifiez la méthode proposée.

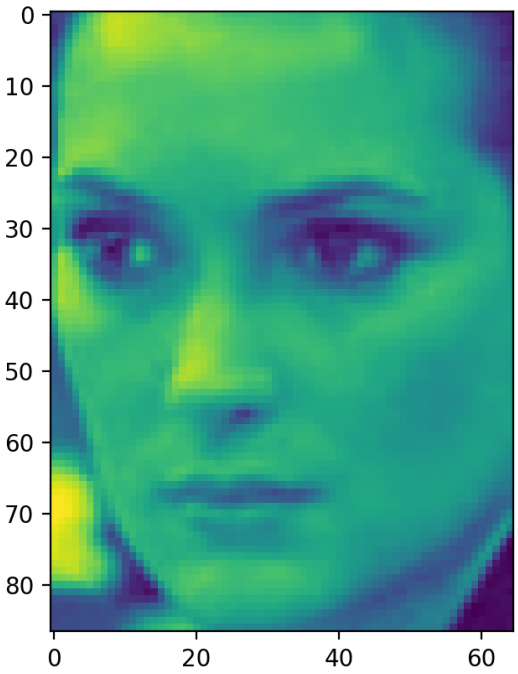
In [24]:
```
import sklearn
import numpy as np
import scipy as sp
import pandas as pd
%matplotlib notebook
import matplotlib.pyplot as plt
import seaborn as sns
```

In [25]:
```
#Chargement d'un ensemble de données de faces de personnages connus
from sklearn.datasets import fetch_lfw_people
faces = fetch_lfw_people(min_faces_per_person=20, resize=0.7)
```

In [26]:
```
#format des images, données, cibles, et nombres de clusters
print("Format des images: {}".format(faces.images.shape))
print("Format des images: {}".format(faces.data.shape))
print("Format des images: {}".format(faces.target.shape))
print("Nombre de classes: {}".format(len(faces.target_names)))
```

```
Format des images: (3023, 87, 65)
Format des images: (3023, 5655)
Format des images: (3023,)
Nombre de classes: 62
```

In [27]:
```
#Affichage de l'image, les données, et la cible pour la première en
trée
plt.imshow(faces.images[0])
print(len(faces.data[0]))
print(faces.data[0])
print(faces.target[0])
print(faces.target_names[faces.target[0]])
```




```
5655
[57.       60.333332 78.       ... 17.333334 16.666666 22.666666]
61
Winona Ryder
```

In [28]:
```
#nombre de données par classe
nombres = np.bincount(faces.target)
for i, (nb, nom) in enumerate(zip(nombres, faces.target_names)):
    print("{0:25} {1:3}".format(nom, nb), end='   ')
    if (i + 1) % 3 == 0:
        print()
```

```
Alejandro Toledo           39   Alvaro Uribe              35   Amel
ie Mauresmo                21
Andre Agassi               36   Angelina Jolie            20   Arie
l Sharon                   77
Arnold Schwarzenegger      42   Atal Bihari Vajpayee      24   Bill
Clinton                    29
Carlos Menem               21   Colin Powell             236   Davi
d Beckham                  31
Donald Rumsfeld           121   George Robertson          22   Geor
ge W Bush                 530
Gerhard Schroeder         109   Gloria Macapagal Arroyo   44   Gray
Davis                      26
Guillermo Coria            30   Hamid Karzai              22   Hans
Blix                       39
Hugo Chavez                71   Igor Ivanov               20   Jack
Straw                      28
Jacques Chirac             52   Jean Chretien             55   Jenn
ifer Aniston               21
Jennifer Capriati          42   Jennifer Lopez            21   Jere
my Greenstock              24
Jiang Zemin                20   John Ashcroft             53   John
Negroponte                 31
Jose Maria Aznar           23   Juan Carlos Ferrero       28   Juni
chiro Koizumi              60
Kofi Annan                 32   Laura Bush                41   Lind
say Davenport              22
Lleyton Hewitt             41   Luiz Inacio Lula da Silva 48   Mahm
oud Abbas                  29
Megawati Sukarnoputri      33   Michael Bloomberg         20   Naom
i Watts                    22
Nestor Kirchner            37   Paul Bremer               20   Pete
Sampras                    22
Recep Tayyip Erdogan       30   Ricardo Lagos             27   Roh
Moo-hyun                   32
Rudolph Giuliani           26   Saddam Hussein            23   Sere
na Williams                52
Silvio Berlusconi          33   Tiger Woods               23   Tom
Daschle                    25
Tom Ridge                  33   Tony Blair               144   Vice
nte Fox                    32
Vladimir Putin             49   Winona Ryder              24
```

In [29]:
```
#Affichage des images et nom pour les 10 premières entrées
fig, axes = plt.subplots(2, 5, figsize=(10, 6),
                         subplot_kw={'xticks': (), 'yticks': ()})
for nom, image, ax in zip(faces.target, faces.images, axes.ravel
()):
    ax.imshow(image)
    ax.set_title(faces.target_names[nom])
```

In [30]:
```
#Reduction du jeu de données à 40 entrées par classe
garder40 = np.zeros(faces.target.shape,dtype=bool)
for nom in np.unique(faces.target):
    garder40[np.where(faces.target==nom)[0][:40]] = 1
X_faces, y_faces, X_images = faces.data[garder40], faces.target[gar
der40], faces.images[garder40]
len(y_faces)
```

Out[30]: 1916

In [ ]: