SI 221

Bases de l'Apprentissage

TP "Décision Bayésienne"

Laurence Likforman-Sulem, Lucien Maman, David Perera, Morgan Buisson, Florian Angulo

Telecom Paris

Février 2022

1. Préliminaires

1.1 Librairies

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.linalg import sqrtm
from numpy.linalg import eig
import scipy
import math
from numpy.linalg import det
```

1.2 Liste des opérations ou fonctions dont vous pouvez avoir besoin

- np.array([[1, 0],[0,6]])] crée la matrice [1 0 ; 0 6]
- x.transpose() : transpose le vecteur/matrice x
-
- np.eye(2) crée une matrice identité de taille 2x2
- np.dot(x,y) produit matriciel entre 2 matrices x et y, ou produit scalaire entre 2 vecteurs x et y.
- np.mean(x) calcule la moyenne du vecteur x
- np.exp(x) calcule l'exponentielle de x
- sqrtm(s), s étant une matrice, retourne la matrice r telle que $s = r\, r^t$
- np.sqrt(s), retourne la matrice contenant les racines carrées de chaque élément de la matrice s, ou si s est une variable, sqrt renvoie sa racine carrée.
- np.random.randn(N) : crée N échantillons d'une variable gaussienne (moyenne 0, variance égale à 1)
- np.random.randn(2,N) : crée N échantillons d'un vecteur gaussienne de dimension 2 (moyenne 0, matrice de covariance égale à l'identité)
- np.linalg.inv(s) : inverse la matrice s
- det(s) : calcule le determinant de s

- la fonction np.linspace permet de créer une grille de points.

- plt.contour (X, Y, F.transpose()) ; où X (resp. Y) est un vecteur de dimension m (resp.n) et F une matrice de dimension n x m . Affiche en 2-D les lignes de contours dans les éléments de la matrice F.

1.3 Fonctions d'affichage

• plt.figure() ouvre une nouvelle fenêtre d'affichage

• plt.subplot(2,1,2) et plt.plot (x,y), x et y étant des vecteurs de même taille. Subplot crée plusieurs sous figures. Ici c'est dans la 2eme sous figure que la commande plot va s'exécuter.

- plt.plot(x[0,:],x[1,:],'ro'): affiche le graphe de la 2eme dimension du vecteur x, en fonction de sa première dimension, en couleur rouge

- plt.axis('equal') : égalise l'échelle des axes x et y

2. Génération d'une variable aléatoire gaussienne

En utilisant la fonction randn, générer N échantillons d'une variable aléatoire gaussienne de moyenne 3 et de variance 4. Etudier l'évolution de la moyenne et de la variance empiriques en fonction de N.

3. Génération de vecteurs aléatoires gaussiens

On va générer en dimension 2, un ensemble d'apprentissage constitué de trois classes d'échantillons suivant des lois normales, de vecteurs moyenne et de matrices de covariance donnés.

1) Générer un échantillon de taille N=100 d'un vecteur aléatoire gaussien défini par le vecteur moyenne m=[4 9] et la matrice de covariance égale à l'identité. Afficher les échantillons.

2) Donner l'expression permettant de générer N=100 échantillons d'un vecteur aléatoire de moyenne m=[4 9] et de matrice de covariance diagonale s= [1 0 ; 0 6]
Vérifier votre résultat en utilisant les fonctions *mean* et *cov* et afficher les échantillons.

3) Soit X un vecteur aléatoire gaussien centré de matrice de covariance égale à l'identité. Chercher une transformation linéaire X'= U X qui permette d'obtenir un vecteur aléatoire centré de matrice de covariance Σ .

4) Générer des échantillons dont la matrice de covariance est égale à s= $\begin{bmatrix} 2 & 2 \\ 2 & 5 \end{bmatrix}$

Vérifier votre résultat avec les fonctions *mean* et *cov*.
Calculer l'orientation de l'ellipsoide de Mahalanobis asscocié à s.
Vérifier la relation liant s à sa matrice diagonalisée S_d : $s=V.S_d.V^t$
avec V, matrice des vecteurs propres de s.

5.) Générer dans les matrices x1, x2, x3, trois vecteurs aléatoires gaussiens en dimension 2 (100 échantillons par vecteur). On donne:
m1 = [4 9] s1 = $[2 \quad 2, \; 2 \; 5]$
m2 = [8.5 7.5] s2 = $[2 \; -2, \; -2 \; 5 \quad]$
m3 = [6 3.5] s3 = $[7 \; -4, \; -4 \; 7 \quad]$

Afficher l'ensemble des échantillons, chaque classe dans une couleur différente.

4. Courbes d'équidensité

1). Ouvrir une autre fenêtre. Créer une grille de points X(i), Y(j) répartis régulièrement dans l'espace de taille 57x57 entre les valeurs 0.27 et 12.5 pour la première mesure et -2 et 15 pour la deuxième mesure (Utiliser la fonction *linspace*).

2) Pour la classe 1 de la question 3.5, construire la matrice dens1(i,j) contenant la densité de probabilité conditionnelle en chaque point (X(i), Y(j)) de la grille.

3) Afficher les courbes d'équidensités pour la classe 1 à l'aide de la fonction *contour*.
Quelle est la forme de ces courbes ?

4). Faire de même pour les deux autres classes. Représenter en 3-D sur la même figure les trois lois de densités conditionnelles (utiliser la fonction *mesh*).
Expliquer l'allure des amplitudes maximales des lois de densités

5. Visualisation des frontières

1) Classer les points (X(i), Y(j)) de la grille précédente. On utilisera une matrice Z(i,j) dont les éléments contiendront l'étiquette (1, 2 ou 3) de la classe obtenue pour chacun des éléments de la grille.

2) Pour faire apparaître les frontières entre classes, appliquer la fonction *contour* à la transposée de Z et aux vecteurs X et Y. Indiquez les régions de l'espace correspondant à chacune des classes ?
On pourra superposer les frontières aux points échantillons.

6. Application

Cette application est issue du Cars dataset (collecté par E. Ramos et D. Donoho et entreposé dans la bibliothèque StatLib). Cet ensemble contient environ 400 échantillons représentant des modèles de voitures et 8 variables les décrivent. La dernière variable est la classe des échantillons, c-a-d le continent d'origine du modèle : USA (1), Europe (2), Asie (3). Les modèles voitures ne sont pas équirépartis suivant les continents.

On veut prédire la classe du modèle (le continent d'origine) à partir de deux variables :

- MPG : miles per gallon relatif à la consommation du véhicule (en position 1)
- weight : le poids de la voiture (en position 5)

Déterminer le vecteur moyenne et la matrice de covariance de chacune des classes. Tracer les frontières entre classes à partir des échantillons fournis dans le fichier *voitures.mat* (*load voitures* renvoie dans la variable *cars* l'ensemble des échantillons).