Exercise 7 : Bayes Optimal Classifier, Naïve Bayes, Random Variables and Distributions, EM Clustering

Exercise 7-1 : Bayes Optimal

We have a classification problem with two classes "+" and "−", three trained classifiers h_1, h_2, and h_3, with the following probabilities of the classifiers, given the training data D :

$$\Pr(h_1|D) = 0.5$$
$$\Pr(h_2|D) = 0.3$$
$$\Pr(h_3|D) = 0.2$$

For the three test instances o_1, o_2, o_3, the classifiers give the following class probabilities :

$$\begin{aligned}
o_1 : \Pr(+|h_1) &= 0.6 & \Pr(-|h_1) &= 0.4 \\
\Pr(+|h_2) &= 0.2 & \Pr(-|h_2) &= 0.8 \\
\Pr(+|h_3) &= 0.9 & \Pr(-|h_3) &= 0.1 \\
o_2 : \Pr(+|h_1) &= 0.6 & \Pr(-|h_1) &= 0.4 \\
\Pr(+|h_2) &= 0.6 & \Pr(-|h_2) &= 0.4 \\
\Pr(+|h_3) &= 1 & \Pr(-|h_3) &= 0 \\
o_3 : \Pr(+|h_1) &= 0.6 & \Pr(-|h_1) &= 0.4 \\
\Pr(+|h_2) &= 0.6 & \Pr(-|h_2) &= 0.4 \\
\Pr(+|h_3) &= 0 & \Pr(-|h_3) &= 1
\end{aligned}$$

We combine the three classifiers to get a Bayes optimal classifier. Which class probabilities will we get from this Bayes optimal classifier for the three test instances ?

Exercise 7-2 : Naïve Bayes

The skiing season is open. To reliably decide when to go skiing and when not, you could use a classifier such as Naïve Bayes. The classifier will be trained with your observations from the last year. Your notes include the following attributes :

The weather : The attribute `weather` can have the following three values : `sunny`, `rainy`, and `snow`.

The snow level : The attribute `snow level` can have the following two values : ≥ 50 (There are at least 50 cm of snow) and < 50 (There are less than 50 cm of snow).

Assume you went skiing 8 times during the previous year. Here is the table with your decisions :

weather	**snow level**	**ski ?**
sunny	< 50	no
rainy	< 50	no
rainy	≥ 50	no
snow	≥ 50	yes
snow	< 50	no
sunny	≥ 50	yes
snow	≥ 50	yes
rainy	< 50	yes

(a) Compute the *a priori* probabilities for both classes `ski = yes` and `ski = no` (on the training set) !

(b) Compute the distribution of the conditional probabilities for the two classes for each attribute.

(c) Decide for the following weather and snow conditions, whether to go skiing or not ! Use the Naïve Bayes classifier as trained in the previous steps for your decision.

	weather	**snow level**
day A	sunny	≥ 50
day B	rainy	< 50
day C	snow	< 50

Exercise 7-3 : Assignments in the EM-Algorithm

Given a data set with 100 points consisting of three Gaussian clusters A, B and C and the point p.

The cluster A contains 30% of all objects and is represented using the mean of all its points $\mu_A = (2, 2)$ and the covariance matrix $\Sigma_A = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$.

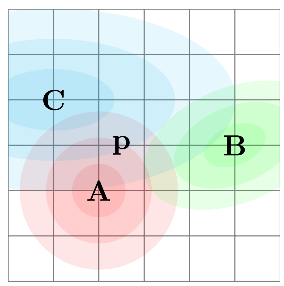
You will need the inverse : $\Sigma_A^{-1} = \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{3} \end{pmatrix}$.

The cluster B contains 20% of all objects and is represented using the mean of all its points $\mu_B = (5, 3)$ and the covariance matrix $\Sigma_B = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix}$. $\Sigma_B^{-1} \approx \begin{pmatrix} 0.571428 & -0.142857 \\ -0.142857 & 0.285714 \end{pmatrix}$.

The cluster C contains 50% of all objects and is represented using the mean of all its points $\mu_C = (1, 4)$ and the covariance matrix $\Sigma_C = \begin{pmatrix} 16 & 0 \\ 0 & 4 \end{pmatrix}$. $\Sigma_C^{-1} = \begin{pmatrix} \frac{1}{16} & 0 \\ 0 & \frac{1}{4} \end{pmatrix}$.

The point p is given by the coordinates $(2.5, 3.0)$.

The following sketch is not exact, and only gives a rough idea of the cluster locations :



Compute the three probabilities of p belonging to the clusters A, B, and C.

Exercise 7-4 : Tools : EM algorithm

Consider EM algorithm on iris dataset as bellow :

```
import matplotlib.pyplot as plt
from sklearn import datasets
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
import scipy.stats
import seaborn as sns
import numpy as np

# import some data to play with
iris = datasets.load_iris()
X = iris.data  # we only take the first two features.
y = iris.target

X = PCA(n_components=2).fit_transform(iris.data)
# --------------------
# The EM algo
# --------------------

def normal_density(X, mu, Sigma):
    L = np.linalg.cholesky(Sigma)
    Linv = np.linalg.inv(L)
    Sinv = Linv.T.dot(Linv)

    XL = X.dot(Linv)
    # P stands for precision, i.e. inverse Sigma
    xPx = (XL*XL).sum(axis=1)
    xPmu = X.dot(Sinv).dot(mu)
    muPmu = mu.dot(Sinv).dot(mu)
    mahalanobis = xPx -2*xPmu + muPmu
    twoPiPowD = (2*np.pi)**D
    sqrtDetSigma = L.diagonal().prod()
    density = 1/(np.sqrt(twoPiPowD)*sqrtDetSigma)*
                        np.exp(-0.5*(mahalanobis))
    return density

K = 2 # Cluster count
max_iter = 20
(N,D) = X.shape

# Initialize
mu = np.random.randn(K,D)
```

```
Sigma = np.zeros([K,D,D])
for k in range(K):
    #L = np.random.randn(D,D)
    Sigma[k,:,:] = np.eye(D) #+ L.dot(L.T)
cls_prob = np.zeros([N,K])
pi_k = np.ones(K)/K

list_log_lik = np.zeros([max_iter])

for iter in range(max_iter):

    # E-STEP ---------------------------------------------
    # Update cluster probabilities
    for k in range(K):
        cls_prob[:,k] = pi_k[k]*normal_density(X,mu[k,:],Sigma[k,:,:])
    cls_prob = cls_prob / np.broadcast_to(np.expand_dims
                    (cls_prob.sum(axis=1), axis=1),(N,K))

    Nk = cls_prob.sum(axis=0)
    pi_k = Nk / Nk.sum()

    # M-STEP ---------------------------------------------
    # Update means and covariances
    for k in range(K):
        clsProbMat = np.broadcast_to(np.expand_dims(cls_prob[:,k]
                                                  ,axis=1),(N,D))
        mu[k,:] = 1/Nk[k]*(X*clsProbMat).sum(axis=0)

        Z = (X - mu[k,:])*np.sqrt(clsProbMat)
        ZtZ = 1/Nk[k]*Z.T.dot(Z)
        Sigma[k,:,:] = 1/Nk[k]*ZtZ + np.eye(D)

    # Report model fit ----------------
    evidence = 0
    for k in range(K):
        evidence += pi_k[k]*normal_density(X,mu[k,:],Sigma[k,:,:])
    list_log_lik[iter] = np.log(evidence).sum()
```

(a) Rerun the algorithm for different number of clusters.

(b) Plot marginal log likelihood and cluster assignments for different values of K.

(c) Plot heatmaps for different values of K.

(d) Describe which k can better fit the model.