Solutions

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 :

$$\begin{aligned}\Pr(h_1|D) &= 0.5\\ \Pr(h_2|D) &= 0.3\\ \Pr(h_3|D) &= 0.2\end{aligned}$$

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 ?

Suggested solution :

The Bayes optimal classifier adds the conditional class probabilities given the classifier, weighted with the conditional classifier probabilities given the data :

$$\Pr(c_j|D) = \sum_{h_i \in \mathcal{H}} \Pr(c_j|h_i) \Pr(h_i|D)$$

yielding the optimal classification :

$$\arg\max_{c_j \in C} \sum_{h_i \in \mathcal{H}} \Pr(c_j|h_i) \Pr(h_i|D)$$

The resulting probabilities are :

$$
\begin{aligned}
o_1 : \Pr(+|\text{Bayes optimal}) &= 0.54 \\
o_1 : \Pr(-|\text{Bayes optimal}) &= 0.46 \\
o_2 : \Pr(+|\text{Bayes optimal}) &= 0.68 \\
o_2 : \Pr(-|\text{Bayes optimal}) &= 0.32 \\
o_3 : \Pr(+|\text{Bayes optimal}) &= 0.48 \\
o_3 : \Pr(-|\text{Bayes optimal}) &= 0.52
\end{aligned}
$$

and the predictions therefore :

$$
\begin{aligned}
o_1 &: + \\
o_2 &: + \\
o_3 &: -
\end{aligned}
$$

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) !

Suggested solution :

$$\begin{aligned} P(ski) &= 0.5 \\ P(\neg ski) &= 0.5 \end{aligned}$$

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

Suggested solution :

$$\begin{aligned} P(weather = sunny|ski) &= \frac{1}{4} \\ P(weather = snow|ski) &= \frac{2}{4} \\ P(weather = rainy|ski) &= \frac{1}{4} \\ P(weather = sunny|\neg ski) &= \frac{1}{4} \\ P(weather = snow|\neg ski) &= \frac{1}{4} \\ P(weather = rainy|\neg ski) &= \frac{2}{4} \end{aligned}$$

$$\begin{aligned}
P(snow \geq 50|ski) &= \frac{3}{4} \\
P(snow < 50|ski) &= \frac{1}{4} \\
P(snow \geq 50|\neg ski) &= \frac{1}{4} \\
P(snow < 50|\neg ski) &= \frac{3}{4}
\end{aligned}$$

(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

Suggested solution :

$$\begin{aligned}
P(C_i|M) &\overset{\text{Bayes}}{=} \frac{P(M|C_i) \cdot P(C_i)}{P(M)} \\
&= \frac{P(M|C_i) \cdot P(C_i)}{\sum_{C_j \in C} P(C_j) \cdot P(M|C_j)}
\end{aligned}$$

A :

$$\begin{aligned}
&P(ski|weather = sunny, snow \geq 50) \\
&= \frac{P(weather = sunny|ski) \cdot P(snow \geq 50|ski) \cdot P(ski)}{P(weather = sunny, snow \geq 50)} \\
&= \frac{\frac{1}{4} \cdot \frac{3}{4} \cdot \frac{1}{2}}{P(weather = sunny, snow \geq 50)} = \frac{\frac{3}{32}}{P(weather = sunny, snow \geq 50)}
\end{aligned}$$

$$\begin{aligned}
&P(\neg ski|weather = sunny, snow \geq 50) \\
&= \frac{P(weather = sunny|\neg ski) \cdot P(snow \geq 50|\neg ski) \cdot P(\neg ski)}{P(weather = sunny, snow \geq 50)} \\
&= \frac{\frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{2}}{P(weather = sunny, snow \geq 50)} = \frac{\frac{1}{32}}{P(weather = sunny, snow \geq 50)}
\end{aligned}$$

$\Rightarrow$ Ski

B :

$$\begin{aligned}
&P(ski|weather = rainy, snow < 50)\\
&= \frac{P(weather = rainy|ski) \cdot P(snow < 50|ski) \cdot P(ski)}{P(weather = rainy, snow < 50)}\\
&= \frac{\frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{2}}{P(weather = rainy, snow < 50)} = \frac{\frac{1}{32}}{P(weather = rainy, snow < 50)}
\end{aligned}$$

$$\begin{aligned}
&P(\neg ski|weather = rainy, snow < 50)\\
&= \frac{P(weather = rainy|\neg ski) \cdot P(snow < 50|\neg ski) \cdot P(\neg ski)}{P(weather = rainy, snow < 50)}\\
&= \frac{\frac{2}{4} \cdot \frac{3}{4} \cdot \frac{1}{2}}{P(weather = rainy, snow < 50)} = \frac{\frac{6}{32}}{P(weather = rainy, snow < 50)}
\end{aligned}$$

$\Rightarrow$ do not ski

C :

$$\begin{aligned}
&P(ski|weather = snow, snow < 50)\\
&= \frac{P(weather = snow|ski) \cdot P(snow < 50|ski) \cdot P(ski)}{P(weather = snow, snow < 50)}\\
&= \frac{\frac{2}{4} \cdot \frac{1}{4} \cdot \frac{1}{2}}{P(weather = snow, snow < 50)} = \frac{\frac{2}{32}}{P(weather = snow, snow < 50)}
\end{aligned}$$

$$\begin{aligned}
&P(\neg ski|weather = snow, snow < 50)\\
&= \frac{P(weather = snow|\neg ski) \cdot P(snow < 50|\neg ski) \cdot P(\neg ski)}{P(weather = snow, snow < 50)}\\
&= \frac{\frac{1}{4} \cdot \frac{3}{4} \cdot \frac{1}{2}}{P(weather = snow, snow < 50)} = \frac{\frac{3}{32}}{P(weather = snow, snow < 50)}
\end{aligned}$$

$\Rightarrow$ do not ski

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 $\Sigma_A = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$.

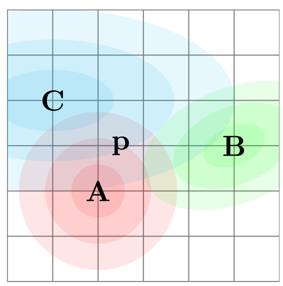
You will need the inverse : $\Sigma_A^{-1} = \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{3} \end{pmatrix}$.

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 $\Sigma_C = \begin{pmatrix} 16 & 0 \\ 0 & 4 \end{pmatrix}$. $\Sigma_C^{-1} = \begin{pmatrix} \frac{1}{16} & 0 \\ 0 & \frac{1}{4} \end{pmatrix}$.

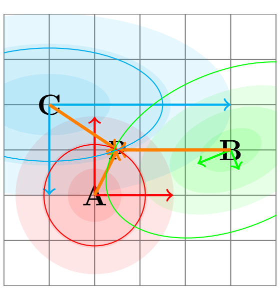
The point p is given by the coordinates $(2.5, 3.0)$.

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.

Suggested solution :



$$\Sigma_A = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix} \quad \Sigma_A^{-1} = \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{3} \end{pmatrix}$$

$$p - \mu_A = (0.5, 1)$$

$$dist^2 = (p-\mu)^T \Sigma^{-1} (p-\mu) \approx 0.41666$$

$$dens_A \approx \frac{1}{\sqrt{(2\pi)^2 9}} e^{-\frac{1}{2}0.41666}$$
$$\approx 0.04307456$$

$$\Sigma_B = \begin{pmatrix} 2 & 1 \\ 1 & 4 \end{pmatrix} \quad \Sigma_B^{-1} \approx \begin{pmatrix} 0.571428 & -0.142857 \\ -0.142857 & 0.285714 \end{pmatrix}$$

$$p - \mu_B = (-2.5, 0)$$

$$dist^2 = (p-\mu)^T \Sigma^{-1} (p-\mu) \approx 3.5714285$$

$$dens_B \approx \frac{1}{\sqrt{(2\pi)^2 7}} e^{-\frac{1}{2}3.5714285}$$
$$\approx 0.01008661$$

$$\Sigma_C = \begin{pmatrix} 16 & 0 \\ 0 & 4 \end{pmatrix} \quad \Sigma_C^{-1} = \begin{pmatrix} \frac{1}{16} & 0 \\ 0 & \frac{1}{4} \end{pmatrix}$$

$$p - \mu_C = (1.5, -1)$$

$$dist^2 = (p-\mu)^T \Sigma^{-1} (p-\mu) \approx 0.390625$$

$$dens_C \approx \frac{1}{\sqrt{(2\pi)^2 64}} e^{-\frac{1}{2}0.390625}$$
$$\approx 0.01636466$$

	A	B	C
density	0.043075	0.010087	0.016365
size	30%	20%	50%
score	0.012922	0.002017	0.008182
sum	divide by 0.023122		
weight (i.e., probability of assignment)	$\approx$ 55.9%	$\approx$ 8.2%	$\approx$ 35.4%

Point p belongs most likely to cluster A!

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.

Suggested solution :

```
# The model fit
plt.plot(list_log_lik)
plt.show()

# The cluster assignments
y_predictions = np.argmax(cls_prob, 1)
plt.scatter(X[:, 0], X[:, 1], c = y_predictions,
                cmap=plt.cm.Set1, edgecolor="k")
plt.xlabel("PC 1")
plt.ylabel("PC 2")
plt.xticks(())
plt.yticks(())
plt.show()
```

(c) Plot heatmaps for different values of K.

Suggested solution :

```
# heatmap
from scipy.stats import multivariate_normal
a, b = np.mgrid[X[:, 0].min()-.01:X[:, 0].max()+.01:.01,
                X[:, 1].min()-.01:X[:, 1].max()+.01:.01]
pos = np.dstack((a, b))
rv = multivariate_normal(mu[0], Sigma[0])
plt.contourf(a, b, rv.pdf(pos), alpha=0.25, cmap="PuRd")
rv = multivariate_normal(mu[1], Sigma[1])
plt.contourf(a, b, rv.pdf(pos), alpha=0.25, cmap="Oranges")
plt.scatter(X[:, 0], X[:, 1], c=y_predictions)
plt.xlabel("PC 1")
plt.ylabel("PC 2")
plt.show()
```

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