Aprendizagem Automática

FICHA N. 1 ENUNCIADO Semestre de Inverno 2017/2018

Nome: Rui Filipe Guimarães Dos Santos
Número: A39286

1. No ficheiro `A39286_Q001_data.p`, encontram-se um conjunto de dados bi-dimensionais divididos em 3 classes (índices de `0` a `2`). Há duas variáveis num dicionário: a chave `trueClass` contém os índices das classes dos dados, enquanto a chave `dados` contém os dados bi-dimensionais. Verificam-se as seguintes condições no conjunto de dados disponibilizado:

   (a) A média dos dados é: $\begin{bmatrix} 0.41 \\ 0.55 \end{bmatrix}$

   (b) A percentagem de pontos da classe `2` é de 28.1%

   (c) A matriz de covarância dos dados é: $\begin{bmatrix} 0.14 & 0.00 \\ 0.00 & 0.12 \end{bmatrix}$

   (d) A distância entre as médias da classe `1` e da classe `0` é de 1.40 unidades.

   (e) A matriz de covarância da classe `2` é: $\begin{bmatrix} 0.14 & 0.00 \\ 0.00 & 0.12 \end{bmatrix}$

   (f) A média da classe `0` é: $\begin{bmatrix} -0.01 \\ -0.01 \end{bmatrix}$

2. Dado um conjunto de N pontos a d dimensões, pretende-se calcular a matriz de covariância dos dados. Para tal, assuma que:

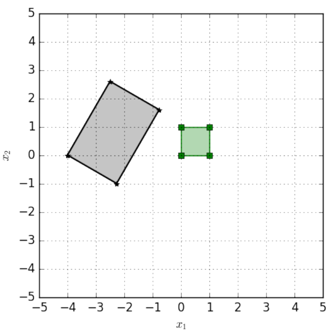
   - Já foi executado o comando:`import numpy as np`
   - Os dados estão guardados numa variável `X` – array de dimensão $d \times N$.

   As seguintes instruções em Python calculam a matriz de covariância dos dados (guardada na variável `Cx`):

   (a)
   ```
   Cx=np.cov(X,ddof=1)
   ```
   (b)
   ```
   mx=np.mean(X,axis=1)
   Xn=(X.T-mx).T
   Ctmp=Xn*Xn
   Cx=Ctmp/(X.shape[1]-1)
   ```
   (c)
   ```
   Cx=np.cov(X.T,rowvar=False,ddof=0)
   ```
   (d)
   ```
   mx=np.mean(X,axis=0)
   Xn=X-mx
   Ctmp=np.dot(Xn,Xn.T)
   Cx=Ctmp/(X.shape[1]-1)
   ```
   (e)
   ```
   mx=np.mean(X,axis=1)
   Xn=X-mx[:,np.newaxis]
   Ctmp=np.dot(Xn,Xn.T)
   Cx=Ctmp/(X.shape[1]-1)
   ```
   (f)
   ```
   mx=np.mean(X,axis=0)
   mx=mx
   Ctmp=np.dot(X,X.T)/(X.shape[1]-1)
   Cx=Ctmp-np.dot(mx,mx.T)
   ```

3. Considere uma variável aleatória 2D uniformemente distribuída na área a cinzento da figura. Considere que esta variável 2D foi obtida através de uma transformação linear de uma outra variável uniformemente distribuída no intervalo $x_1, x_2 \in [0, 1]$ (quadrado pequeno verde). A transformação consistiu em escalar a 1ª dimensão dos dados, x_1, por um factor de 2, a segunda, x_2 por um factor de 3. Depois foi aplicada um rotação contrária ao sentido dos ponteiros do relógio de -30 graus, seguido de uma translação. Em termos genéricos, esta transformação pode ser descrita segundo um produto matricial entre duas matrizes, $\mathbf{R}$ e Δ, de rotação e de escalamento dos dados, mais um vector $\mathbf{b}$ de translação.

$$\mathbf{y} = \mathbf{R}\Delta\mathbf{x}+\mathbf{b} = \mathbf{A}\tilde{\mathbf{x}} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$$



Esta transformação também pode ser obtida através de um único produto matricial com uma matriz $\mathbf{A}$, representando os dados em coordenadas homogéneas.

As seguintes afirmações são verdadeiras (considere arredondamentos até à segunda casa decimal):

(a) A matriz de transformação é: $\mathbf{A} = \begin{bmatrix} -4.00 & 1.73 & 1.50 \\ 0.00 & -1.00 & 2.60 \end{bmatrix}$

(b) O vector de translação é: $\mathbf{b} = \begin{bmatrix} -1.00 \\ 2.00 \end{bmatrix}$

(c) A matriz de rotação é: $\mathbf{R} = \begin{bmatrix} 0.87 & -0.50 \\ 0.50 & 0.87 \end{bmatrix}$

(d) A matriz de escalamento é: $\Delta = \begin{bmatrix} 2.00 & 0.00 \\ 0.00 & 2.00 \end{bmatrix}$

4. Considere uma variável aleatória 2D, cuja a distribuição consiste numa soma de duas densidades Gaussianas 2d:

$$\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \quad \text{com } \boldsymbol{\mu}_1 = \begin{bmatrix} -2 \\ 1 \end{bmatrix} \quad \text{e } \boldsymbol{\Sigma}_1 = \begin{bmatrix} 2.25 & -0.00 \\ -0.00 & 2.25 \end{bmatrix}$$

$$\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \quad \text{com } \boldsymbol{\mu}_2 = \begin{bmatrix} 2 \\ -1 \end{bmatrix} \quad \text{e } \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.25 & 0.00 \\ 0.00 & 0.25 \end{bmatrix}$$

$$p(\mathbf{x}) = p_1\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + p_2\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \quad \text{com } p_1 = 0.50$$

Pretende-se gerar 1000 pontos com esta distribuição. Para tal assuma que já se executou o comandos `import numpy as np` e `import scipy.linalg as la`. Assuma igualmente que já foram definidas as matrizes de covariância, os vetores de média, e as probabilidades a priori das Gaussianas (`np.arrays`: matrizes de covariâncias 2×2 `S1` e `S2`, vetores de média 2×1 `m1` e `m2`. `floats`: probabilidades a priori `p1` e `p2`). Os seguintes comandos geram os pontos pretendidos (guardados na variável `X`):

(a)
```
A1=la.sqrmt(S1)
A2=la.sqrtm(S2)
X1=np.dot(A1,np.random.randn(2,1000))+m1
X2=np.dot(A2,np.random.randn(2,1000))+m2
X=X1+X2
```

(b)
```
A1=la.sqrmt(S1)
A2=la.sqrtm(S2)
X1=np.dot(A1,np.random.randn(2,1000))+m1
X2=np.dot(A2,np.random.randn(2,1000))+m2
X=np.hstack((X1,X2))
```

(c)
```
A1=np.array([[0.00,1.50],[-1.50,0.00]])
A2=np.array([[0.50,0.00],[0.00,0.50]])
X1=A1*np.random.randn(2,1000*p1)+m1
X2=A2*np.random.randn(2,1000*p1)+m2
X=np.hstack((X1,X2))
```

(d)
```
A1=la.sqrmt(S1)
A2=la.sqrtm(S2)
X1=A1*np.random.randn(2,1000)+m1
X2=A2*np.random.randn(2,1000)+m2
X=np.hstack((X1,X2))
```