
Aprendizagem Automática

FICHA N. 2

ENUNCIADO

SEMESTRE DE INVERNO 2017/2018

Nome: Rui Filipe Guimarães Dos Santos

Número: A39286

1. No ficheiro `A39286_Q001_data.p` estão os resultados de um classificador, num problema de duas classes (negativos e positivos). Há duas variáveis num dicionário: a chave `trueClass` contém os índices da verdadeira classe, enquanto a chave `estClass` contém o resultado da classificação (ambas as variáveis são `np.arrays` com valores inteiros 0-negativos, 1-positivos).
 - (a) A probabilidade total de respostas corretas é de 72.8%
 - (b) A taxa de re-chamada (recall) é: 0.800
 - (c) A precisão (precision) é: 0.734
 - (d) O número de Falsos Negativos é: 90
 - (e) O número de Falsos Positivos é: 269
 - (f) A probabilidade total de erro é de 19.8%
2. Dado um conjunto de N pontos a d dimensões pertencendo a uma de três classes (0, 1 e 2) considere que se quer classificar o conjunto com o método da distância ao centroide, usando uma das seguintes métricas de distância: Euclideana, “Manhattan”, ou cosseno. Pretende-se implementar o processo de classificação em Python. Para tal, assuma o seguinte:
 - Assuma que já foram executados os comandos:

```
import numpy as np
import numpy.linalg as la
from sklearn.neighbors import KNeighborsClassifier as KNC
from sklearn.neighbors.nearest_centroid import NearestCentroid as NC
```
 - Os dados estão guardados numa variável X – array de dimensão $d \times N$.
 - Os centroides estão guardados nas variáveis m_0, m_1 e m_2 – arrays de dimensão $d \times 1$.
 - O resultado da classificação é o array, `estClass`, de N entradas, com números inteiros de 0 a 2.

As seguintes instruções em Python implementam o classificador de distância ao centroide usando, ou a distância Euclideana, ou a de “Manhattan”, ou a de cosseno:

- | | |
|---|---|
| <pre>(a) kNN=KNC(n_neighbors=1, algorithm='brute', p=1) M=np.hstack((m0,m1,m2)) kNN.fit(M.T,np.arange(3)) estClass=kNN.predict(X.T)</pre> | <pre>M=np.hstack((m0,m1,m2)) kNN.fit(M.T,np.arange(3)) estClass=kNN.predict(X.T)</pre> |
| <pre>(b) kNN=KNC(n_neighbors=1, algorithm='brute') kNN.fit(X.T,np.arange(3)) M=np.hstack((m0,m1,m2)) estClass=kNN.predict(M.T)</pre> | <pre>(d) X0=X-m0; X1=X-m1; X2=X-m2 d0=np.sqrt(np.sum(X0*X0,0)) d1=np.sqrt(np.sum(X1*X1,0)) d2=np.sqrt(np.sum(X2*X2,0)) D=np.vstack((d0,d1,d2)) estClass=np.argmin(D,axis=0)</pre> |
| <pre>(c) kNN=KNC(n_neighbors=1, algorithm='brute')</pre> | |

3. No ficheiro `A39286_Q003_data.p`, encontram-se um conjunto de dados bi-dimensionais divididos em três classes (0, 1, e 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.

(a) Verificam-se as seguintes condições no conjunto de dados disponibilizado:

- | | |
|--|---|
| i. A matriz de covariância da classe 2 é: $\begin{bmatrix} 0.26 & -0.00 \\ -0.00 & 0.24 \end{bmatrix}$ | iii. A média da classe 1 é: $\begin{bmatrix} -0.20 \\ 0.90 \end{bmatrix}$ |
| ii. A matriz de covariância dos dados é: $\begin{bmatrix} 0.59 & 0.44 \\ 0.44 & 0.39 \end{bmatrix}$ | iv. A probabilidade apriori da classe 0 é de 28.6% |

(b) Considere que se pretende classificar este conjunto usando o classificador de distância ao centroide, com a métrica de distância de Mahalanobis. Dado este classificador e os respectivos resultados da classificação, verifica-se o seguinte:

- | | |
|---|--|
| i. A probabilidade de erro da classe 1 é de 1.8% | iii. A probabilidade de pontos da classe 1 serem classificados na classe 2 é de 1.8% |
| ii. A probabilidade de pontos da classe 2 serem classificados na classe 0 é de 3.1% | iv. A probabilidade total de respostas corretas é de 85.4% |

4. No ficheiro `A39286_Q004_data.p` estão os resultados de um classificador. Há duas variáveis num dicionário: a chave `trueClass` contém os índices da verdadeira classe, enquanto a chave `estClass` contém o resultado da classificação (ambas as variáveis são `np.array`s com valores inteiros de 0 a 3).

- | | |
|---|---|
| (a) A probabilidade a priori da classe 1 é de 17.0% | (c) A probabilidade de pontos da classe 0 serem classificados na classe 1 é de 4.0% |
| (b) A probabilidade de pontos da classe 1 serem classificados na classe 0 é de 4.0% | (d) A probabilidade de acertos na classe 2 é de 90.0% |