



PAC4: Introducció a l'aprenentatge computacional

Presentació

Quarta PAC del curs d'Intel·ligència Artificial

Competències

En aquesta PAC es treballaran les següents competències:

Competències de grau:

- Capacitat d'analitzar un problema amb el nivell d'abstracció adequat a cada situació i aplicar les habilitats i coneixements adquirits per abordar-lo i solucionar-ho.

Competències específiques:

- Conèixer els diferents aspectes d'aprenentatge computacional (aprenentatge supervisat, no supervisat i per reforç).
- Conèixer els fonaments teòrics dels mètodes més representatius.
- Conèixer el tractament dels conjunts de dades per a la correcta validació dels sistemes d'aprenentatge.

Objectius

Aquesta PEC pretén avaluar diferents aspectes d'aprenentatge supervisat i no supervisat.

Descripció de la PAC a realitzar

Pregunta 1)

Un sistema d'estabilització d'avions necessita encendre o apagar segons dos sensors x i y . Aquests sensors tenen valors reals que van de $-\infty$ a ∞ , i el control es pot encendre (on) o apagar (off). A causa de l'alt cost d'accedir als sensors, només s'han obtingut 16 instàncies amb la predicció de control i els valors dels sensors. L'objectiu del sistema és entrenar un algoritme per a tots els valors de x i y .

Inst	Classe	X	Y
------	--------	---	---



1	off	-1	2
2	off	-2	2
3	on	2	2
4	on	1	2
5	on	-3	-1
6	on	-5	-1
7	off	4	-1
8	off	2	-1
9	off	-2	1
10	off	-4	1
11	on	4	4
12	on	6	7
13	on	-9	-9
14	on	-2	-1
15	off	3	-3
16	on	-2	1

Apartat 1 (punts 3)

Primer de tot construirem un arbre de decisió amb les característiques x i y . Aquestes característiques són nombres reals, per això hem d'aplicar una funció per binaritzar x i y . D'aquesta manera podrem calcular la bondat i decidir la millor binarització per separar les instàncies. Per tant, calcular la binarització de les dades amb les següents funcions i construir un arbre de decisió a partir de la bondat.

$$\begin{cases} 0, \text{if } x \leq -1 \\ 1, \text{if } x > -1 \end{cases} \quad \begin{cases} 0, \text{if } y \leq 0 \\ 1, \text{if } y > 0 \end{cases}$$

$$\begin{cases} 0, \text{if } x \leq 0 \\ 1, \text{if } x > 0 \end{cases} \quad \begin{cases} 0, \text{if } y \leq 1 \\ 1, \text{if } y > 1 \end{cases}$$

Solució:

Primer de tot, es calculen els 4 atributs nous, un per cadascuna de les funcions d'entrada. Un cop es calculin els nous atributs, es pot calcular la bondat per decidir els millors atributs per separar les instàncies.



$$X0 \quad \begin{cases} 0, if \ x \leq -1 \\ 1, if \ x > -1 \end{cases}$$

$$X1 \quad \begin{cases} 0, if \ x \leq 0 \\ 1, if \ x > 0 \end{cases}$$

$$Y0 \quad \begin{cases} 0, if \ y \leq 0 \\ 1, if \ y > 0 \end{cases}$$

$$Y1 \quad \begin{cases} 0, if \ y \leq 1 \\ 1, if \ y > 1 \end{cases}$$

Inst	Classe	X	Y	X0	X1	Y0	Y1
1	off	-1	2	0	0	1	1
2	off	-2	2	0	0	1	1
3	on	2	2	1	1	1	1
4	on	1	2	1	1	1	1
5	on	-3	-1	0	0	0	0
6	on	-5	-1	0	0	0	0
7	off	4	-1	1	1	0	0
8	off	2	-1	1	1	0	0
9	off	-2	1	0	0	1	0
10	off	-4	1	0	0	1	0
11	on	4	4	1	1	1	1
12	on	6	7	1	1	1	1
13	on	-9	-9	0	0	0	0
14	on	-2	-1	0	0	0	0
15	off	3	-3	1	1	0	0
16	on	-2	1	0	0	1	0

Càlcul Bondat



X0:

- X0=1 off 3 **on 4**
- X0=0 off 4 **on 5**
- Bondat: $(5+4)/16 = 9/16$

X1:

- X1=1 off 3 **on 4**
- X1=0 off 4 **on 5**
- Bondat: $(5+4)/16 = 9/16$

Y0:

- Y0=1 off 4 **on 5**
- Y0=0 off 3 **on 4**
- Bondat: $(5+4)/16 = 9/16$

Y1:

- Y1=1 off 2 **on 4**
- Y1=0 off 5 **on 5**
- Bondat: $(5+4)/16 = 9/16$

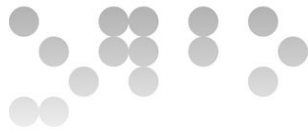
Al tenir tots els atributs la mateixa bondat, podem utilitzar qualsevol per dividir les dades.

Agafem X0 per dividir les dades tot i que es podria dividir les dades amb X1, Y0 o Y1.

Node Arrel X0:

- X0<=-1

Inst	Clase	X	Y	X1	Y0	Y1
1	off	-1	2	0	1	1
2	off	-2	2	0	1	1
5	on	-3	-1	0	0	0
6	on	-5	-1	0	0	0
9	off	-2	1	0	1	0
10	off	-4	1	0	1	0
13	on	-9	-9	0	0	0
14	on	-2	-1	0	0	0
16	on	-2	1	0	1	0



Ara calcularem la bondat amb els tres atributs que falten:

X1:

- X1=1 off 0 **on 0**
- X1=0 off 4 **on 5**
- Bondat: $(5+0)/9 = 5/9$

Y0:

- Y0=1 off 4 **on 1**
- Y0=0 off 0 **on 4**
- Bondat: $(4+4)/9 = 8/9$

Y1:

- Y1=1 **off 2** on 0
- Y1=0 off 2 **on 5**
- Bondat: $(5+2)/9 = 7/9$

Les instàncies restants es divideixen amb Y0. Sent la sortida de les fulles la següent:

Y0=0: Prob(on) = 1 y Prob(off)=0

Y0=1: Prob(on) = 0.2 y Prob(off)=0.8

- **X0>-1**

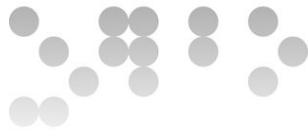
Inst	Clase	X	Y	X1	Y0	Y1
3	on	2	2	1	1	1
4	on	1	2	1	1	1
7	off	4	-1	1	0	0
8	off	2	-1	1	0	0
11	on	4	4	1	1	1
12	on	6	7	1	1	1
15	off	3	-3	1	0	0

X1:

- X1=1 off 3 **on 4**
- X1=0 off 0 **on 0**
- Bondat: $(4+0)/7 = 4/7$

Y0:

- Y0=1 off 0 **on 4**
- Y0=0 **off 3** on 0



- Bondat: $(4+3)/7 = 1$

Y1:

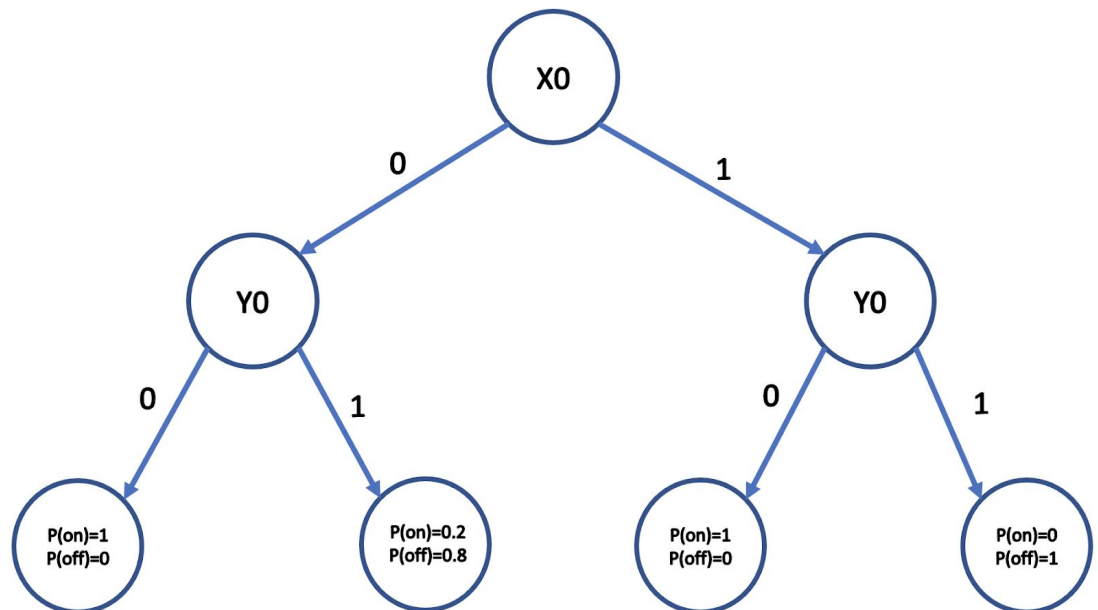
- Y1=1 off 0 on 4
- Y1=0 off 3 on 0
- Bondat: $(5+2)/7 = 7/7$

Las instancias restantes se pueden dividir con Y0 o Y1, escogiendo Y0 como referencia. Siendo la salida de las hojas la siguiente:

Les instàncies restants es poden dividir amb Y0 o Y1, escollint Y0 com a referència. Sent la sortida de les fulles la següent:

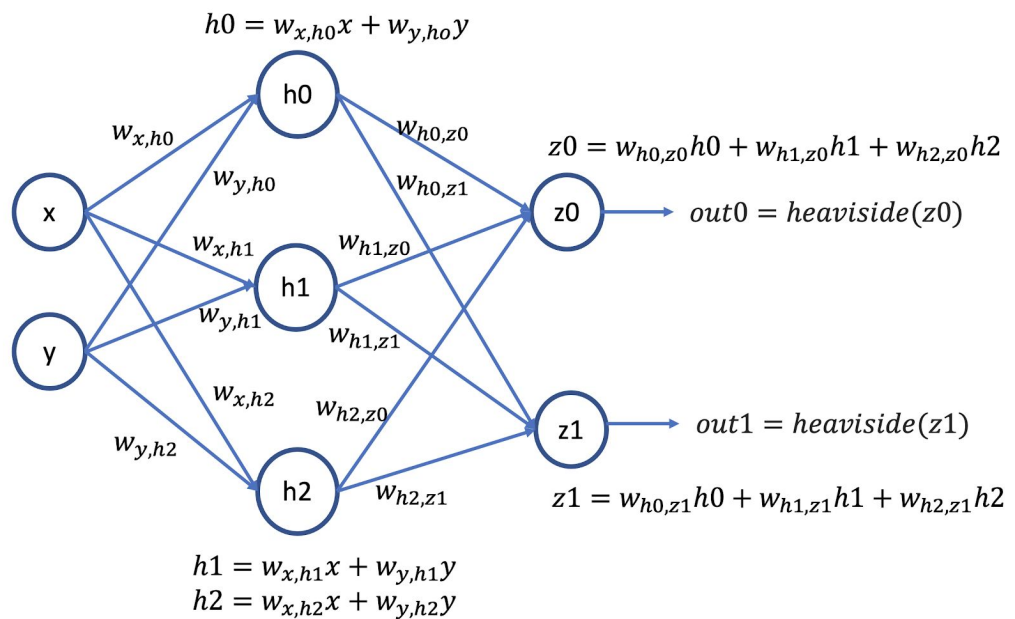
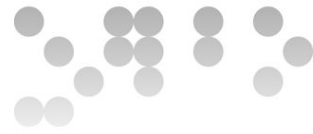
Y0=0: Prob(on) = 1 y Prob(off)=0

Y0=1: Prob(on) = 0 y Prob(off)=1



Apartat 2 (punts 2)

Enlloc d'utilitzar directament les característiques x i y , obtindrem unes noves característiques $out0$ i $out1$ a partir d'una xarxa neuronal. Així doncs, en primer lloc, s'han de calcular la sortides $out0$ i $out1$ de cadascuna de les instàncies (les 16 instàncies de la taula anterior) usant la xarxa neuronal de sota amb els següents pesos:



$$\begin{array}{llll}
 w_{x,h0} = 0 & w_{y,h0} = -1 & & \\
 w_{x,h1} = -1 & w_{y,h1} = 0 & w_{h0,z0} = 0 & w_{h1,z0} = -1 & w_{h2,z0} = 0 \\
 w_{x,h2} = 0 & w_{y,h2} = -1 & w_{h0,z1} = -1 & w_{h1,z1} = 0 & w_{h2,z1} = -1
 \end{array}$$

i la següent funció heaviside que depèn del valor d'entrada.

$$heaviside(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ 1, & \text{if } x > 0 \end{cases}$$

Solució:

Per a cada sortida out0 i out1, s'agafa cada instància d'entrada i es calcula les variables latents, després els valors previs a la sortida z, i finalment s'aplica la funció Heaviside per obtenir els variables out.

Inst	Classe	X	Y	out0	out1
1	off	-1	2	0	1
2	off	-2	2	0	1
3	on	2	2	1	1
4	on	1	2	1	1
5	on	-3	-1	0	0
6	on	-5	-1	0	0



7	off	4	-1	1	0
8	off	2	-1	1	0
9	off	-2	1	0	1
10	off	-4	1	0	1
11	on	4	4	1	1
12	on	6	7	1	1
13	on	-9	-9	0	0
14	on	-2	-1	0	0
15	off	3	-3	1	0
16	on	-2	1	0	1

Apartat 3 (punts 2)

Un cop calculats els valors per a totes les instàncies, hem de construir un arbre de decisió a partir de les sortides d'aquest conjunt de dades calculant la bondat de les particions de les dues sortides *out0* i *out1*. Així doncs, en lloc d'entrenar un arbre de decisió sobre les característiques *x* i *y*, ho farem sobre les característiques *out0* i *out1*.

Solució:

Càlcul Bondat

out0:

- *out0*=1 off 3 **on 4**
- *out0*=0 off 4 **on 5**
- Bondat: $(5+4)/16 = 9/16$

out1:

- *out1*=1 off 4 **on 5**
- *out1*=0 off 3 **on 4**
- Bondat: $(5+4)/16 = 9/16$

Al tenir tots els atributs la mateixa bondat es pot usar qualsevol per dividir les dades.

Agafem *out0* per dividir les dades tot i que es podria dividir les dades amb *out1*.



Node Arrel out0:

- out0=0

Inst	Classe	X	Y	out0	out1
1	off	-1	2	0	1
2	off	-2	2	0	1
5	on	-3	-1	0	0
6	on	-5	-1	0	0
9	off	-2	1	0	1
10	off	-4	1	0	1
13	on	-9	-9	0	0
14	on	-2	-1	0	0
16	on	-2	1	0	1

Ara calcularem la bondat amb l'atribut que falta:

out1:

- out1=1 off 4 **on 1**
- out1=0 off 0 **on 4**
- Bondat: $(4+4)/9 = 8/9$

Les instàncies restants es divideixen amb out1, on la sortida de les fulles es la següent:

out1=0: Prob(on) = 1 y Prob(off)=0

out1=1: Prob(on) = 0.2 y Prob(off)=0.8

- out0=1

Inst	Classe	X	Y	out0	out1
3	on	2	2	1	1
4	on	1	2	1	1
7	off	4	-1	1	0
8	off	2	-1	1	0
11	on	4	4	1	1



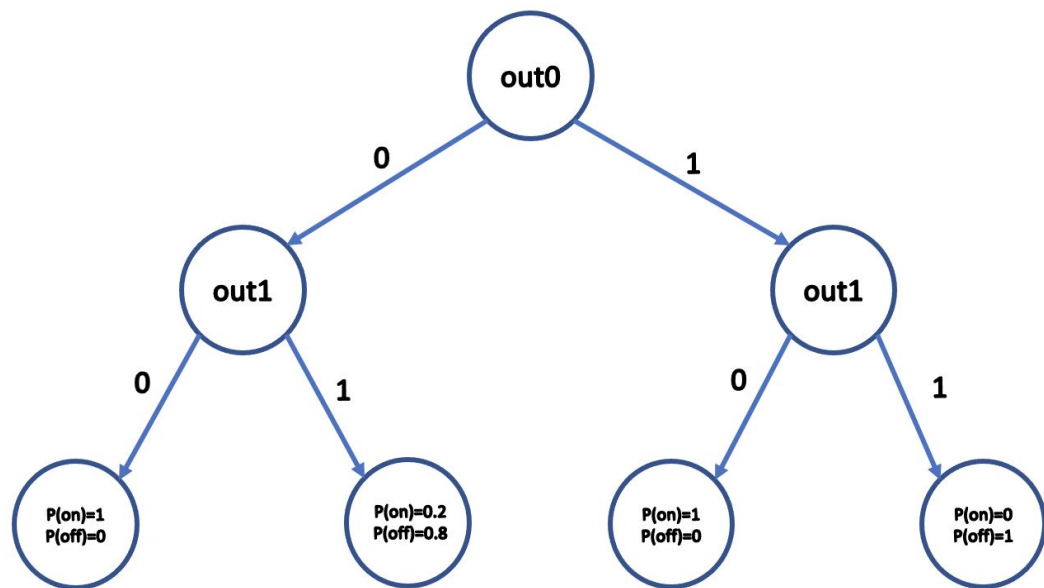
12	on	6	7	1	1
15	off	3	-3	1	0

out1:

- out1=1 off 0 **on 4**
- out1=0 **off 3** on 0
- Bondat: $(4+3)/7 = 1$

Les instàncies restants es divideixen amb out1, on la sortida de les fulles es la següent:

out1=0: Prob(on) = 1 y Prob(off)=0
 out1=1: Prob(on) = 0 y Prob(off)=1



Apartat 4 (punts 1)

Utilitzant el notebook de Python adjunt s'ha de completar el codi per poder entrenar una multilayer Perceptron (MLP) amb una única sortida. Incloure la visualització donada per la funció `plt.contourf()` i interpretar els resultats obtinguts d'aquesta visualització.

Solució:



Només era necessari crear el classificador amb els paràmetres per defecte, encara que en la solució s'afegeix el nombre màxim de passos i un random state per tenir els mateixos valors. Finalment, s'entrena el classificador amb els valors de X i y.

```
clf_mlp = MLPClassifier(random_state=1, max_iter=300)
clf_mlp.fit(X,y)
```

Pregunta 2) (2 punts)

Utilitzant el notebook de Python adjunt divideix la base de dades per a una validació simple per comparar el KNearestNeighbors Classifier amb totes les dades o reduint el nombre de dades d'entrada utilitzant algoritme k-means. Implementar les parts necessàries per saber el comportament de l'algoritme k-means i el nombre de punts propers. Com afecta el nom de clústers utilitzats per K-means i el nombre de veïns propers considerats pel classificador KNN?

Solució:

El primer pas era implementar la divisió de dades en training i test usant la funció `train_test_split` indicant un percentatge necessari per a l'entrenament.

```
# implement simple validation
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=1-percentage_training, random_state=42)
```

Un cop tenim les dades d'entrenament, l'objectiu era entrenar una `KNeighborsClassifier` com s'indica a continuació:

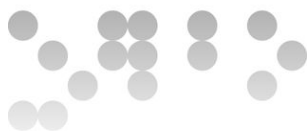
```
#implement kNeighbors classifier
neigh = KNeighborsClassifier(n_neighbors=3)
neigh.fit(X_train, y_train)
```

Per reduir el nombre d'elements del conjunt de train es procedeix a fer un clustering usant algoritme k-means i predir en què clúster pertany.

```
kmeans = KMeans(n_clusters=n_clusters, random_state=0).fit(X_train)
```

```
kmeans = KMeans(n_clusters=i_cluster, random_state=0).fit(X_train)
y_pred_train = kmeans.fit_predict(X_train)
```

Finalment, l'objectiu era entrenar el `KNeighborsClassifier` usant els clústers entrenats i el valor del clúster en comptes d'usar totes les instàncies.



```
neigh = KNeighborsClassifier(n_neighbors=i_nn)
neigh.fit(X_train_kmeans, y_train_kmeans)
y_pred = neigh.predict(X_test)
```

Recursos

Per fer aquesta PAC el material és el mòdul 5 i per executar els notebooks en Python de l'exercici va recomanar utilitzar Google Col·lab que és gratuït (<https://colab.research.google.com>).

Criteris de valoració

Les puntuacions es mostren en cada pregunta de l'enunciat.

Format i data de lliurament

Per a dubtes i aclariments sobre l'enunciat, adreceu-vos al consultor responsable de la vostra aula.

Cal lliurar la solució en un fitxer PDF fent servir una de les plantilles lliurades conjuntament amb aquest enunciat. Adjunteu el fitxer a un missatge a l'apartat Lliurament i Registre d'AC (RAC).

El nom del fitxer ha de ser *CognomsNom_IA_PAC4* amb l'extensió .pdf (format PDF).

La data límit de lliurament és el: **20/12/2020** (a les 24 hores).

Raoneu la resposta en tots els exercicis. Les respostes sense justificació no rebran puntuació.

Nota: Propietat intel·lectual

Sovint és inevitable, en produir una obra multimèdia, fer ús de recursos creats per terceres persones. És per tant comprensible fer-ho en el marc d'una pràctica dels estudis d'Informàtica, sempre i això es documenti clarament i no suposi plagi en la pràctica.

Per tant, en presentar una pràctica que faci ús de recursos aliens, s'ha de presentar juntament amb ella un document en què es detallin tots ells, especificant el nom de cada recurs, el seu autor, el lloc on es va obtenir i el seu estatus legal: si l'obra està protegida pel copyright o s'acull a alguna altra llicència d'ús (Creative Commons, llicència GNU, GPL ...). L'estudiant haurà d'assegurar-se que la llicència que sigui no impedeix específicament el seu ús en el marc de la pràctica. En cas de no trobar la informació corresponent haurà d'assumir que l'obra està protegida pel copyright.

Hauran, a més, adjuntar els fitxers originals quan les obres utilitzades siguin digitals, i el seu codi font si correspon.

