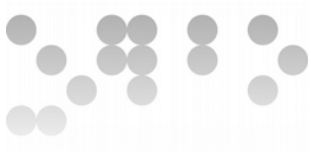


PAC 2

Presentació

Segona activitat d'avaluació continuada del curs. En aquesta PAC es practicaran els algorismes bàsics de classificació.

Objectius

L'objectiu d'aquesta prova d'avaluació és classificar les dades dels arxius adjunts relacionats amb l'origen de distints vins a partir de la seva composició i color. Volem agrupar els vins en funció del seu origen.

Els arxius de dades "csv" tenen un format tipus taula, on cada fila correspon a un exemple. L'última columna és la classe i la resta corresponen als atributs de l'exemple. L'arxiu adjunt "txt" conté la descripció d'aquests atributs.

Aquests arxius pertanyen al problema "Wine Data Set" del repositori d'aprenentatge de l'UCI:

http://archive.ics.uci.edu/ml/

Solució de la PAC

Exercici 1

a) Construïu els models de classificació basats en el veí més proper emprant k=1 i k=3 a partir de l'arxiu "train.csv". És a dir, heu de construir els models 1NN i 3NN. Amb cada un d'aquests dos models heu de classificar els exemples de l'arxiu "test.csv"

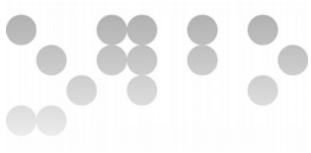
No s'han de tractar els valors absents ja que no n'hi ha. Tots els atributs són numèrics amb valors dispersos. Decidim aplicar estandardització a tots els atributs, tot i que ranging també seria adient. Els valors de la mitjana i la desviació típica per al conjunt d'entrenament són:

```
13.04, 1.71, 91.38, 2.50, 2.72, 4.50
0.93,  0.35, 9.35,  0.47, 0.58, 1.59
```

Per tal d'estandarditzar el conjunt de test hem d'utilitzar la mitjana i la desviació del conjunt d'entrenament.

El primer que fem per a aplicar kNN és calcular les distàncies euclidies de tots els exemples de test a tots els exemples d'entrenament.

	TEST1	TEST2	CLASSE
TRAIN1	5.67	6.41	0
TRAIN2	9.19	8.27	1
TRAIN3	9.25	8.07	1



TRAIN4	7.58	8.62	0
TRAIN5	7.82	8.12	0
TRAIN6	8.71	7.96	1
TRAIN7	6.45	7.83	0
TRAIN8	10.19	9.75	1

L'exemple d'entrenament més proper al primer exemple de test és TRAIN1, que és de classe 0. L'exemple d'entrenament més proper al segon exemple de test també és TRAIN1. Per tant, per a 1NN assignarem la classe 0 als dos exemples de test. Ja que els exemples de test són, realment, de classes 0 i 1 obtenim una precisió del 50%.

Els tres veïns més propers a TEST1 són TRAIN1, TRAIN7 i TRAIN4. Per tant, els vots serien (0,0,0). Els tres veïns més propera a TEST2 són TRAIN1, TRAIN7 i TRAIN6. Els vots serien (0,0,1). Per tant, 3NN assigna la classe 0 tant a TEST1 com a TEST2, resultant en la mateixa precisió que en el cas anterior: 50%.

b) Construïu el model de classificació basat en k-means per a k=2 a partir de l'arxiu "train.csv". És a dir, heu de construir el model per a 2-means. Un cop tingueu el model heu de classificar els exemples de l'arxiu "test.csv".

Comencem aplicant 2-means dues vegades: una per als exemples de cada una de les dues classes. Agafant com a centroides inicials els dos primers exemples de cada cas ens queden els següents centroides per a la classe 0:

0.07	0.97	0.55	0.54	0.93	1.22
1.41	0.18	1.14	0.78	0.45	0.22

Els centroides per a la classe 1 són:

-1.02	-0.09	-0.65	-1.21	-1.18	-1.13
0.08	-2.01	-1.43	1.00	0.80	0.50

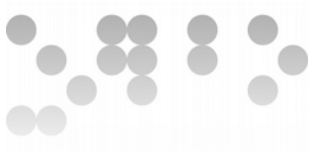
Ara hem d'aplicar 1NN al conjunt de test emprant com a dades d'entrenament aquests quatre centroides. Obtenim la següent matriu de distàncies:

	TEST1	TEST2	CLASSE
TRAIN1	6.73	7.23	0
TRAIN2	6.99	8.18	0
TRAIN3	9.00	8.04	1
TRAIN4	10.19	9.75	1

El veí més proper tant a TEST1 com a TEST2 és TRAIN1. Per tant, les classes assignades serien 0 i 0. Això suposa una precisió del 50%.

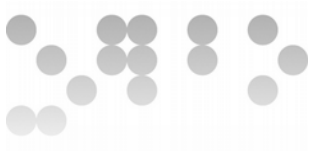
c) Construïu un arbre de decisió a partir de l'arxiu "train.csv". Un cop tingueu l'arbre, classifiqueu amb ell els exemples de l'arxiu "test.csv".

Ja que els arbres de decisió no treballen amb distàncies, no s'han de normalitzar els atributs. Per tant, treballarem directament amb les dades de "train.csv" i "test.csv".



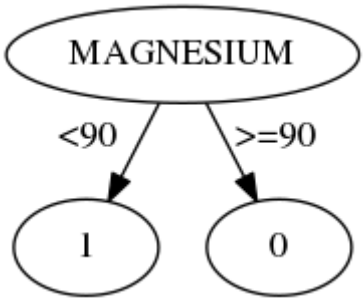
D'altra banda, essent numèrics tots els atributs treballarem amb els seus punts de tall. Aquests punts es calculen ordenant tots els valors de cada atribut i calculant la mitjana aritmètica de cada dos valors consecutius. A continuació es mostren els valors ordenats sense repeticions, els punts de tall i la seva bondat:

ATRIBUT	VALOR	TALL	BONDAT
ALCOHOL	11.65	11.95	0.625
ALCOHOL	12.25	12.31	0.75
ALCOHOL	12.37	12.71	0.875
ALCOHOL	13.05	13.08	0.75
ALCOHOL	13.11	13.135	0.875
ALCOHOL	13.16	13.515	0.75
ALCOHOL	13.87	14.35	0.625
MALIC	1.01	1.32	0.625
MALIC	1.63	1.635	0.75
MALIC	1.64	1.655	0.625
MALIC	1.67	1.7	0.75
MALIC	1.73	1.815	0.75
MALIC	1.9	2.13	0.625
MAGNESIUM	78.0	79.0	0.625
MAGNESIUM	80.0	84.0	0.75
MAGNESIUM	88.0	90.0	1.0
MAGNESIUM	92.0	94.5	0.875
MAGNESIUM	97.0	99.0	0.75
MAGNESIUM	101.0	104.0	0.625
PHENOLS	1.65	1.785	0.625
PHENOLS	1.92	2.07	0.75
PHENOLS	2.22	2.47	0.875
PHENOLS	2.72	2.76	0.75
PHENOLS	2.8	2.875	0.5
PHENOLS	2.95	2.965	0.625
FLAVANOIDS	1.61	1.82	0.625
FLAVANOIDS	2.03	2.24	0.75
FLAVANOIDS	2.45	2.71	0.875
FLAVANOIDS	2.97	2.975	0.75
FLAVANOIDS	2.98	3.08	0.625



```
FLAVANOIDS      3.18    3.21    0.75
FLAVANOIDS      3.24    3.255   0.625
COLOR           2.12    2.36    0.625
COLOR           2.6     3.0     0.75
COLOR           3.4     3.95    0.875
COLOR           4.5     4.85    0.75
COLOR           5.2     5.25    0.625
COLOR           5.3     5.49    0.75
COLOR           5.68    6.44    0.625
```

Com es pot veure, amb l'atribut MAGNESIUM i el punt de tall 90 obtenim una bondat del 100%. Així tenim el conjunt d'exemples perfectament particionat. L'arbre de decisió seria el que es mostra a continuació:



Les prediccions per als exemples de test serien 0 i 1 i, per tant, tendríem una precisió del 100%

d) Apliqueu PCA per a reduir la dimensionalitat dels conjunts anteriors conservant el 95% de la variància. Utilitzeu ara 1NN i 3NN sobre els conjunts reduïts de la mateixa forma que en el primer apartat. Compareu els resultats. Indiqueu clarament les diferències en aplicar PCA sobre el conjunt d'entrenament i sobre el de test.

Apliquem PCA sobre l'arxiu d'entrenament estandarditzat. Obtenim els següents pecentatges de variàncies:

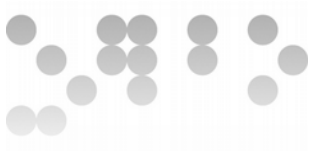
```
0.60 0.25 0.09 0.04 0.02 0.00
```

I l'acumulat:

```
0.60 0.84 0.94 0.98 1.00 1.00
```

Per tant, necessitem 4 components. El resultat de projectar el conjunt d'entrenament és:

```
-1.73  1.55  0.87 -0.25
 2.85  0.63 -0.10 -0.21
 2.55  0.12  0.44  0.75
-1.76 -0.21 -1.03  0.83
-1.45 -0.52  1.21  0.14
 1.64  0.15 -0.53 -0.51
-1.73  1.04 -0.85 -0.34
```



```
-0.37 -2.77 -0.01 -0.40
```

El resultat de projectar el test, prèviament estandarditzat amb la mateixa mitjana i la desviació típica del conjunt d'entrenament, és:

```
-3.74 6.71 0.72  0.02
-0.24 5.10 3.05 -0.89
```

Amb les dades projectades procedim a calcular la matriu de distàncies:

	TEST1	TEST2	CLASS
TRAIN1	5.55	4.47	0
TRAIN2	9.01	6.32	1
TRAIN3	9.15	6.49	1
TRAIN4	7.45	7.08	0
TRAIN5	7.60	6.12	0
TRAIN6	8.59	6.40	1
TRAIN7	6.23	5.85	0
TRAIN8	10.10	8.47	1

L'exemple més proper a TEST1 és TRAIN1 i el més proper a TEST2 també és TRAIN1. Per tant, 1NN assignaria la classe 0 als dos exemples de test. Això ens dóna una precisió del 50%.

Els tres exemples més propers a TEST1 són TRAIN1, TRAIN7 i TRAIN4. Per tant, els vots serien (0,0,0). Els tres exemples més propers a TEST2 són TRAIN1, TRAIN7 i TRAIN5. Els vots serien (0,0,0). En conseqüència, 3NN assignaria la classe 0 als dos exemples, produïnt una precisió del 50%.

Com es pot veure hem obtingut la mateixa classificació (i, per tant, precisió) tant amb PCA com sense.

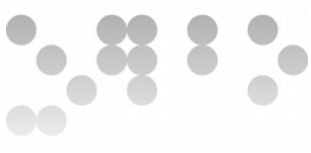
Exercici 2

L'objectiu d'aquest exercici és la construcció d'un model amb un conjunt de dades realista. Per a realitzar aquest exercici teniu a la vostra disposició la biblioteca sklearn per a Python:

http://scikit-learn.org/

L'arxiu adjunt "wine.csv" conté més de 100 exemples de vins en un format fàcilment llegible des de Python. Se us demana l'estudi de cara a construir un classificador. En particular, heu de:

- Provar els algorismes de validació creuada (*cross-validation*). És a dir, no heu d'utilitzar un sol conjunt d'entrenament i un sol conjunt de test. La biblioteca sklearn proporciona iteradors específicament dissenyats per a aquesta tasca, com ara *KFold*, *RepeatedKFold*, *GroupKFold*, *StratifiedKFold*. Llegiu-ne la documentació i concentrau-vos en un d'ells.
- Provar almenys els algorismes: KNN, algun tipus de SVM, xarxes neuronals (perceptró multicapa) i algun altre classificador de la vostra elecció.



Comenteu els resultats obtinguts i justifiqueu tot el que feu.

Hem escollit realitzar les proves sol·licitades amb un 10-fold cross validation bàsic. Per això hem emprat l'iterador KFold de sklearn. Els algorismes que hem provat són: KNN, SVC i NuSVC (ambdós són SVM), MLT (perceptró multicapa) i GaussianNB.

A continuació es mostren els resultats obtinguts:

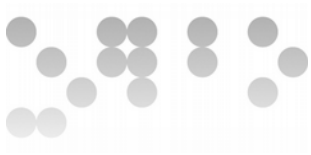
	KNN	SVC	NUSVC	MLP	GAUSSIANNB
T. MODEL(MITJANA)	0.339ms	0.717ms	0.957ms	177.3ms	0.611ms
T. MODEL(DESVIACIÓ)	0.124ms	0.129ms	0.152ms	6.669ms	0.052ms
T. CLASS (MITJANA)	0.452ms	0.112ms	0.118ms	0.157ms	1.411ms
T. CLASS (DESVIACIÓ)	0.168ms	0.022ms	0.020ms	0.005ms	2.648ms
ACCURACY	95.902%	100%	99.180%	99.180%	98.361%
PRECISION	100%	100%	98.529%	100%	97.101%
RECALL	92.537%	100%	100%	98.507%	100%
CLASS. CORRECTES	117	122	121	121	120
CLASS. INCORRECTES	5	0	1	1	2
MATRIU DE CONFUSIÓ	[[55,0], [5,62]]	[[55,0], [0,67]]	[[54,1], [0,67]]	[[55,0], [1,66]]	[[53,2], [0,67]]

Exercici 3

Realitzeu una valoració global comparant els mètodes dels diferents exercicis i redacteu unes conclusions globals sobre l'aplicació dels mètodes a aquests conjunts de dades. Els criteris de correcció de la PAC invaliden una A si tots els processos no estan ben justificats i comentats.

En aquest exercici s'espera que extreieu conclusions generals sobre els altres exercicis. Aquestes conclusions dependran dels resultats obtinguts. A mode d'exemple, enumerarem algunes de les qüestions sobre les que es podrien argumentar:

- La possibilitat de que sigui pràctica l'aplicació d'algun dels mètodes sobre el problema donat. En cas que no ho sigui, indicar què faltaria afegir.
- Comparativa dels mètodes emprats. Enumeració dels avantatges i inconvenients en funció de la precisió, l'eficiència, les categories, els models, ...
- La representació del problema. Com es comporten els atributs? És una bona representació? Com afecta el preprocés de les dades al funcionament dels algorismes?
- Avantatges dels models que generen els diferents mètodes. Comparativa dels models generats durant tot l'exercici.



- En general, intent de justificació o explicació dels resultats que es van obtenint, fixant-se no només en la precisió.
- Com es comporten els algorismes en funció del nombre d'exemples d'entrenament de que es disposen?
- Quin cost computacional té cada un dels mètodes, tant pel que fa a l'entrenament com per a la classificació?

En comparar els resultats és important notar que els conjunts de dades emprats tenen diferent número d'atributs.

Exercici 4

El *K-Fold cross validation* presenta dos principals inconvenients. D'una banda, pot requerir un elevat temps d'execució. D'altra banda, només explora un subconjunt de les possibles particions de dades entre entrenament i prova. Existeixen estratègies que pretenen alleugerir aquests problemes. Una d'elles es l'anomenada *Monte-Carlo Cross Validation*. Cerqueu informació sobre aquesta estratègia, descriviu-la i analitzeu els avantatges i inconvenients que presenta en relació al K-Fold.

En Monte-Carlo cross validation es selecciona aleatòriament un subconjunt de les dades com a conjunt d'entrenament i la resta de dades es fan servir com a dades de test. Aquest procés es repeteix un determinat nombre de vegades. Com a avantatge respecte a K-fold tenim que els conjunts seleccionats (tant de test com d'entrenament) no contenen necessàriament dades consecutives. De fet, s'aconsegueix una distribució més realista de les possibilitats amb que es podria trobar un sistema real. Es podria dir que, a l'infinit, Monte-Carlo cross validation comprovaria totes les possibles combinacions d'entrenaments i proves. També es pot considerar que proporciona un mostreig raonablement homogeni amb independència del nombre d'iteracions. Com a inconvenient, es pot argumentar que amb Monte-Carlo cross validation no tenim cap garantia d'haver fet les proves amb totes les dades disponibles, mentre que amb K-Fold sí que la tenim.