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 problemes de fertilitat. Volem diagnosticar els pacients segons si tenen l'esperma normal o alterat.

L'arxiu de dades fertility.csv té 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 attributes.txt conté la descripció d'aquests atributs.

Aquests arxius pertanyen al problema "Fertility" 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 per valors de k u i tres a partir de l'arxiu train.csv. És a dir, heu de construir els models: 1NN i 3NN. Un cop tingueu el model, heu de classificar els exemples de l'arxiu test.csv.

Ens proporcionen un arxiu amb dades de pacients amb problemes de fertilitat.

No hi ha valors absents, no els hem de tractar. Els atributs ens venen normalitzats, no els hem de tractar.

Per aplicar el kNN, calculem les distàncies euclidees de tots els exemples de test a tots els exemples de train:

```
          [,1]      [,2]
[1,] 1.500000 1.4810807
[2,] 2.450490 2.0042205
[3,] 1.019804 1.0215674
[4,] 0.120000 1.4154858
[5,] 1.428286 0.2088061
[6,] 1.420176 0.1900000
```

Per a 1NN, assignarem les classes 'O' a tots dos exemples, ja que els més petits (marcats en vermell) són d'aquestes classes, que correspon a una precisió del 50% ($= \frac{0+1}{2}$).

Per al 3NN els vots serien ('N', 'O', 'O') i ('N', 'N', 'O') que corresponen a les prediccions 'O' i 'N' respectivament; que equival a una precisió del 0% ($= \frac{0+0}{2}$).

b) Construïu els models de classificació basats en el k–means per a un valor de k de dos a partir de l'arxiu train.csv. Un cop tingueu el model, heu de classificar els exemples de l'arxiu test.csv.

Apliquem en primer lloc el 2-means dues vegades: una per als 4 exemples de cadascuna de les classes. Agafant com a centroides inicials els dos primers exemples en cada cas ens queden els quatre centroides:

```
  accident intervention fevers alcohol smoking sitting
N      1.0          1.0      0     0.8       0   0.880
N      0.5          1.0      0     1.0      -1   0.380
O      0.0          1.0      0     0.8       1   0.310
O      0.5          0.5      0     0.8      -1   0.375
```

Apliquem 1NN agafant com a training els quatre centroides obtinguts en el pas anterior i com a test el conjunt de test. El tractarem com en l'exercici anterior. Obtenim les distàncies següents:

```
          [,1]      [,2]
[1,] 1.5000000 1.4810807
[2,] 1.1357817 0.5418487
[3,] 2.4504897 2.0042205
[4,] 0.7071245 0.7100880
```

Que corresponen a les prediccions 'O' i 'N' respectivament. Això equival a un 0% de precisió.

c) Construïu un arbre de decisió a partir de l'arxiu train.csv i classifiqueu amb ell els exemples de l'arxiu test.csv.

Per a construir arbres de decisió no cal normalitzar, donat que no treballarem amb distancies. Els atributs inherentment nominals els tractarem com a tals, obtenim les bondats:

```
accident      67%    =(2+2)/6
intervention  67%    =(1+3)/6
fevers        50%    =(0+3)/6
alcohol       83%    =(3+2)/6
smoking       67%    =(2+1+1)/6
```

Utilitzarem els punts de tall per tractar l'atribut numèric. Els punts de tall es calculen ordenant tots els valors d'un atribut i calculant la mitjana aritmètica de cada dos valors consecutius. La taula següent mostra els valors ordenats sense repeticions, els punts de tall i la seva bondat per a l'atribut 'sitting':

```
0,25   0,28   67%    =(1+3)/6
0,31   0,345  83%    =(2+3)/6
0,38   0,44   50%    =(2+1)/6
0,5    0,69   67%    =(3+1)/6
0,88
```

El millor que obtenim és el punt de tall 0,345 de l'atribut sitting i l'atribut alcohol, que aconsegueix un 83% de bondat. Podem escollir qualsevol dels dos. Nosaltres escollim el numèric.

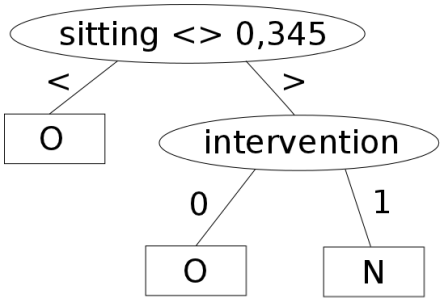
Per sota del punt de tall assignem la classe 'O' i el deixem com a node terminal. Per sobre del punt de tall hem de tornar a iterar. Les bondats dels atributs nominals són ara:

```
accident      75%    =(1+2)/4
intervention  100%   =(1+3)/4
fevers        75%    =3/4
alcohol       75%    =(1+2)/4
smoking       75%    =(2+1)/4
```

I les de l'atribut sitting:

```
0,38   0,44   75%    =(2+1)/4
0,5    0,69   75%    =(2+1)/4
0,88
```

El millor atribut és intervention que particiona correctament el conjunt d'exemples. Així, hem enllestit la construcció de l'arbre, que quedaria:



La prediccions són respectivament 'O' i 'N'. Per tant, obtindrem una predicció del 0%.

d) Apliqueu l'algorisme del PCA per reduir la dimensionalitat dels conjunts anteriors conservant el 95% de la variància. Utilitzeu el 1-NN i 3-NN sobre els conjunts reduïts de la mateixa forma que en el primer apartat. S'obtenen resultats comparables? Expliciteu les diferències en aplicar el PCA sobre el conjunt d'entrenament i sobre el de test.

Per realitzar aquest exercici, hem utilitzat el PCA des de l'R. A continuació teniu les ordres concretes:

```
dt <- read.table("train.csv",head=F,sep=",")
pca.dt <- princomp(dt[,1:6])
vars.dt <- pca.dt$sdev ^ 2
v <- vars.dt / sum(vars.dt)
v
v[1]+v[2]
v[1]+v[2]+v[3]
```

Amb la penúltima línia ens dóna un 88% i amb l'última un 98%. Per tant, hem de menester 3 components. A partir d'aquí fer el que segueix per generar l'arxiu amb aquestes components:

```
dt2 <- cbind(pca.dt$scores[,1:3],as.character(dt$V7))
write.table(dt2,"train.pca.csv", row.names=FALSE,
   col.names=FALSE, sep=",",quote=FALSE)
```

Per fer el test hem de projectar utilitzant els components principals que hem obtingut del conjunt d'entrenament. Les ordres serien:

```
dt3 <- read.table("test.csv",head=F,sep=",")
dt4 <- predict(pca.dt,dt3[,1:6])
dt5 <- cbind(dt4[,1:3],as.character(dt3$V7))
write.table(dt5,"test.pca.csv", row.names=FALSE,
   col.names=FALSE, sep=",",quote=FALSE)
```

Un cop obtingudes les dades projectades, fem el mateix que al primer apartat, calcular la matriu de distàncies:

```
           [,1]       [,2]
[1,] 1.47282243 1.47994550
[2,] 2.45028586 1.97997804
[3,] 1.01143079 0.92303897
[4,] 0.04391927 1.39992880
[5,] 1.40832696 0.03318040
[6,] 1.41112054 0.06953885
```

Per a 1NN, assignarem les classes 'O' i 'N' respectivament, ja que els més petits (marcats en vermell) són d'aquestes classes, que correspon a una precisió del 0% ($= \frac{0+0}{2}$).

Per al 3NN els vots serien ('N', 'O', 'N') i ('N', 'O', 'N') que corresponen a les prediccions 'N' i 'N' respectivament; que equival a una precisió del 50% ($= \frac{1+0}{2}$).

Exercici 2

L'objectiu d'aquest segon exercici és la construcció d'un model amb un conjunt de dades proper al que utilitzaríem en una aplicació pràctica per a un cas real. Per realitzar aquest exercici, teniu a la vostra disposició una eina anomenada Weka a la Web; la seva direcció és:

http://www.cs.waikato.ac.nz/ml/weka

L'arxiu adjunt fertility.csv conté 100 exemples de pacients amb problemes d'esperma amb dues classes. Aquest arxiu està en un dels formats d'entrada del Weka.

Se us demana l'estudi de cara a construir un classificador. Per a realitzar-los heu de:

- Provar els algorismes per validació creuada (cross-validation). És a dir, no utilitzar un sol arxiu de train i un de test. Aquesta opció la permet realitzar el Weka de forma automàtica.
- Provar almenys els algorismes: naïve bayes, algun tipus d'arbre de decisió (mostrant a l'informe l'arbre generat), AdaBoost, xarxes neuronals (perceptró multicapa) i algun altre algorisme.
- Provar les Support Vector Machines (SMO i/o libSVM al Weka) amb diferents tipus de kernel.
- Apliqueu el PCA per reduir la dimensionalitat conservant el 95% de la variància i torneu a aplicar els mateixos algorismes que al conjunt original.

Comenteu els resultats obtinguts i justifiqueu tot el que feu.

Hem escollit realitzar les proves que demana l'enunciat amb un 10-fold crossvalidation sobre els algorismes: Naïve Bayes, 1NN (IB1), 3NN (IB3), Decision Stumps, l'arbre de decisió J48 i l'AdaBoost.M1. Utilitzarem l'última columna com a classe.

La taula següent mostra els resultats sense aplicar cap mena de tractament als atributs. Concretament mostra el temps de construcció dels models, la precisió global, una mesura per cadascuna de les classes i les dades de l'estadístic Kappa (mesura la proporció de la precisió entre les diferents classes):

	NB	DecisionStump	1NN	3NN	J48	AdaBoost.M1
Temps (s)	0,01	0	0,02	0	0,03	0,05
Oks	88	88	83	90	85	88
Precision	88%	88%	83%	90%	85%	88%
F-mesure N	0,936	0,936	0,904	0,946	0,919	0,935
F-mesure O	0	0	0,261	0,375	0	0,25
Kappa stat.	0	0	0,165	0,3351	-0,0504	0,2021

A continuació es mostra l'arbre que s'ha generat amb el mètode d'arbres de decisió J48:

```
J48 pruned tree
------------------
: N (100.0/12.0)
```

i a continuació les matrius de confusió:

Naïve Bayes	88 0 | a = N 12 0 | b = O
Decision Stump	88 0 | a = N 12 0 | b = O
1NN	80 8 | a = N 9 3 | b = O
3NN	87 1 | a = N 9 3 | b = O
J48	85 3 | a = N 12 0 | b = O
AdaBoost.M1	86 2 | a = N 10 2 | b = O

A continuació tenim les dades de les Support Vector Machines:

	SMO			
	lineal	quadratic	rbf ($\gamma = 0.01$)	rbf ($\gamma = 1$)
Temps (s)	0,12	0,04	0,1	0,03
Oks	88	80	88	88
Precision	88%	80%	88%	88%
F-mesure N	0,936	0,889	0,936	0,936
F-mesure O	0	0	0	0
Kappa stat.	0	-0,1062	0	0

SMO	lineal	88 0 | a = N 12 0 | b = O
	quadratic	80 8 | a = N 12 0 | b = O
	rbf ($\gamma = 0.01$)	80 8 | a = N 12 0 | b = O
	rbf ($\gamma = 1$)	80 8 | a = N 12 0 | b = O

Ara hem de repetir tot el procès aplicant PCA.

	NB	DecisionStump	1NN	3NN	J48	AdaBoost.M1
Temps (s)	0,01	0,01	0,02	0	0,04	0,09
Oks	86	88	82	89	88	85
Precision	86%	88%	82%	89%	88%	85%
F-mesure N	0,925	0,936	0,897	0,939	0,936	0,935
F-mesure O	0	0	0,308	0,421	0	0
Kappa stat.	-0,355	0	0,2049	0,3649	0	-0,0504

A continuació es mostra l'arbre que s'ha generat amb el mètode d'arbres de decisió J48:

```
J48 pruned tree
------------------
: N (100.0/12.0)
```

i a continuació les matrius de confusió:

Naïve Bayes	86 2 | a = N 12 0 | b = O
Decision Stump	88 0 | a = N 12 0 | b = O
1NN	78 10 | a = N 8 4 | b = O
3NN	85 3 | a = N 8 4 | b = O
J48	88 0 | a = N 12 0 | b = O
AdaBoost.M1	85 3 | a = N 12 0 | b = O

A continuació tenim les dades de les Support Vector Machines:

	SMO			
	lineal	quadratic	rbf ($\gamma = 0.01$)	rbf ($\gamma = 1$)
Temps (s)	0,1	0,07	0,13	0,03
Oks	88	74	88	87
Precision	88%	74%	88%	87%
F-mesure N	0,936	0,847	0,936	0,93
F-mesure O	0	0,133	0	0
Kappa stat.	0	-0,0125	0	-0,0188

SMO	lineal	88 0 | a = N 12 0 | b = O
	quadratic	72 16 | a = N 10 2 | b = O
	rbf ($\gamma = 0.01$)	80 8 | a = N 12 0 | b = O
	rbf ($\gamma = 1$)	87 1 | a = N 12 0 | b = O

Exercici 3

Realitzeu una valoració global comparant els mètodes i els 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 apartat s'espera que extrèieu conclusions generals sobre l'exercici. Aquestes conclusions dependran dels resultats obtinguts. A mode d'exemple enumerarem algunes de les qüestions sobre les que podeu argumentar:

- La possibilitat de que sigui pràctica l'aplicació d'algun dels mètodes sobre el problema donat. En cas que no, que faltaria afegir.
- Comparativa dels diferents mètodes emprats. Enumeració dels avantatges i inconvenients en funció de: precisió, eficiència, categories, 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 i/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 que es disposen?
- Quin cost computacional té cadascun dels mètodes? Tant en el procés de training com en el de test.

En comparar els resultats, és important notar que els conjunts de dades tenen diferent número de classes.

Exercici 4

Doneu una definició intuïtiva i curta del aprenentatge incremental ("incremental learning") i digueu com creieu que s'aplica a les màquines de vectors de suport.

L'aprenentatge incremental intenta adaptar el model que s'ha construït amb els exemples entrenament a la forma dels exemples de test. És a dir, la forma dels exemples de test pot anar modelant el classificador per adaptar-lo als canvis que es vagin produint en les noves dades.

En el cas de les màquines de vectors de suport això s'aplicaria produint petits canvis en les vectors de suport, o inclús arribar a introduir-ne de nous o eliminar-ne d'existents.