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 targes de crèdit. Volem predir el seu tipus en funció de les seves propietats.

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. La primera columna de l'arxiu *credit_card.csv* correspon a l'identificador del client.

Aquests arxius pertanyen al problema "default of credit cards clients" 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 clients de targes de crèdit. No hi ha valors absents, no els hem de tractar. Els atributs són numèrics i els seus valors són molt dispersos. Decidim aplicar estandardització per tal de normalitzar. Els valors dels promitjos i desviacions són respectivament:

58750 1.5 1.8 1.6 33.6
31795 0.5 0.4 0.5 10.2

En estandarditzar el conjunt de test hem d'utilitzar els promitjos i desviacions del conjunt d'entrenament.

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

```
          [,1]     [,2]
[1,] 3.240363 2.067935
[2,] 3.204294 3.158954
[3,] 2.843961 2.453381
[4,] 3.390957 2.656223
[5,] 4.979741 4.511624
[6,] 2.428702 3.481693
[7,] 3.194316 2.635830
[8,] 2.213583 3.056635
```

Per a 1NN, assignarem la classe 1 al primer exemple i la 1 al segon, ja que els més petits (marcats en vermell) són d'aquestes classes, que correspon a una precisió del 50% (= $\frac{1+0}{2}$).

Per al 3NN els vots serien (1, 0, 0) i (1, 0, 1) que corresponen a les prediccions 0 i 1 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:

```
 0.3538265  0 -0.5773503  0.7745967  0.1845394  0
-0.2751984  0  0.5773503 -1.2909944  1.3163811  0
-1.2187358  0 -0.5773503 -0.2581989 -0.9473023  1
 1.1401076  0  0.5773503  0.7745967 -0.5536182  1
```

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,] 2.059746 2.513928
[2,] 4.022380 3.425853
[3,] 2.073632 1.846534
[4,] 2.928923 2.608890
```

Que corresponen a les prediccions 0 i 1 respectivament. Això equival a un 0% de precisió.

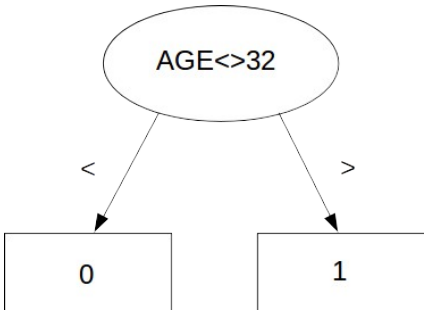
c) Construïu un arbre de decisió a partir de l'arxiu train.csv i classifiqueu amb ell els exemples de l'arxiu test.csv.

Per a construir arbres de decisió no cal normalitzar, donat que no treballarem amb distancies.

Tots els atributs són numèrics. Utilitzarem els punts de tall per tractar-los. Els punts de tall es calculen ordenant tots els valors d'un atribut i calculant la mitjana aritmètica de cada dos valors consecutius. A continuació es mostra els valors ordenats sense repeticions, els punts de tall i la seva bondat per cadascun dels atributs:

LIMIT_BAL
20000 35000 0,75 =(2+4)/8
50000 60000 0,625 =(3+2)/8
70000 80000 0,5 =(3+1)/8
90000 105000 0,625 =(4+1)/8
120000
SEX
1 1,5 0,5 =(2+2)/8
2
EDUCATION
1 1,5 0,5 =(1+3)/8
2
MARRIAGE
1 1,5 0,625 =(2+3)/8
2
AGE
24 25 0,75 =(2+4)/8
26 28 0,875 =(3+4)/8
30 32 1 =(4+4)/8
34 35,5 0,875 =(4+3)/8
37 47 0,625 =(4+1)/8
57

El millor que obtenim és el punt de tall 32 de l'atribut 'AGE', amb el que aconsegueixen un 100% de bondat. Tenim particionat correctament el conjunt d'exemples. Així, hem enllestit la construcció de l'arbre, que quedaria:



La prediccions són respectivament '0' i '0'. Per tant, obtindrem una predicció del 50%.

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.

Apliquem el PCA sobre el conjunt de l'arxiu. Obtenim els percentatges de variàncies:

```
0.39 0.34 0.21 0.05 0.01
```

I acumulant:

```
0.39 0.73 0.94 0.99 1.00
```

El resultat és:

```
 0.88 -0.77 -1.99  0.26
 1.35  2.01  0.45 -0.16
 1.06  1.12  0.45 -0.32
 1.30 -0.99 -0.61 -0.38
 0.26 -2.46  1.50  0.05
-2.04  0.12  0.46 -0.61
-0.35  0.62  0.64  1.13
-2.46  0.35 -0.91  0.03
```

I projectant el test:

```
-0.97 1.27 -1.25 -1.28
 0.22 0.62 -1.50  0.44
```

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

```
          [,1]     [,2]
[1,] 3.238839 1.627279
[2,] 3.173981 2.714071
[3,] 2.823021 2.307914
[4,] 3.386313 2.285145
[5,] 4.977751 4.314975
[6,] 2.423083 3.212744
[7,] 3.193252 2.315643
[8,] 2.211871 2.783155
```

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

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

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 "credit_card.csv" conté 30.000 exemples de clients de targes de crèdit. 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-lo 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.

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, l'AdaBoost.M1 i el perceptró multicapa. Hem eliminat la primera columna (ID), aplicat el filtre NumericToNominal a la classe i estandarditzat la resta.

En aplicar els algorismes a les dades constatem que el conjunt que ens donen té un nombre molt elevat d'exemples per a alguns algorismes del Weka. Aquests són l'arbre de decisió J48, el perceptró multicapa i les màquines de vectors de suport SMO. Si volguéssim aplicar algun d'aquests conjunts hauríem de reduir el nombre d'exemples del conjunt d'entrenament.

La taula següent mostra els resultats per al Naïve Bayes, Decision Stumps, 1NN, 3NN i AdaBoost per al conjunt; 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	AdaBoost.M1
Temps (s)	0,13	0,33	0,04	0,01	2,22
Oks	2320	24385	2307	23896	24385

			7		
Precision	67,4%	81,3%	76,9 %	79,7	81,3%
F-mesure 0	0,772	0,889	0,858	0,878	0,889
F-mesure 1	0,429	0,396	0,378	0,4	0,396
Kappa stat.	0,2165	0,3088	0,243	0,2907	0,3088

i a continuació les matrius de confusió:

Naïve Bayes	16566	6798 |	a = 0
	2972	3664 |	b = 1
Decision Stump	22541	823 |	a = 0
	4792	1844 |	b = 1
1NN	20970	2394 |	a = 0
	4529	2107 |	b = 1
3NN	21864	1500 |	a = 0
	4604	2032 |	b = 1
AdaBoost.M1	22541	823 |	a = 0
	4792	1844 |	b = 1

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

A la descripció del problema que hem tractat en aquestes dues PACs ens diu que seria interessant donar una probabilitat de pertànyer a la classe «default» enlloc de la predicció de classe. Cerqueu informació de com es podria implementar això en el kNN i doneu una descripció breu de com fer-ho donant enfasi en la seva implementació pràctica.

Existeixen moltes formes de fer el que demana l'enunciat. En aquesta solució esmentar dues de les més utilitzades.

Les dues formes que comentarem parteixen de canviar el sistema de votacions per a que els vots depenguin de les distàncies als exemples d'entrenament (com el que demana l'enunciat de la pràctica d'enguany).

La primera forma consisteix en calcular el tant per cent de votació que van a parar al valor 1 de la predicció (sumant el pes/predicció de tots els vots amb 1 dividint per la suma total). Amb això tenim un valor de confiança que equival a una «probabilitat».

La segona forma consisteix en aplicar una regressió logística també a les vots modificats per a que depenguin de les distàncies.

La regressió és un problema similar a la classificació però, en aquest cas, les classes tindrien un valor continu enlloc de discret. Un exemple de regressió seria l'intentar predir el nombre d'hectàrees cremades en un incendi en funció de les característiques de la zona i ambientals. La majoria d'algorismes de classificació tenen la seva versió per a regressió.

La regressió logística és un tipus de regressió en que el valor de les classes en el conjunt d'entrenament es discret (tipus 1 i 0); però donarà un valor continu en classificar el conjunt de test.

En aplicar aquesta segona opció tindríem dos sistemes d'aprenentatge connectats un darrera de l'altre (el kNN i la regressió). Així la regressió prendria com a valors d'entrada els pesos/prediccions generats pel kNN.