Universitat de Lleida

Universitat de Lleida
Escola Politècnica Superior

Grau en Enginyeria Informàtica

Intel·ligència Artificial

Tercera pràctica d'Intel·ligència Artificial

Autors:
Jaume Giralt Barbé
Jordi Onrubia Palacios

Professor:
Carlos Jose Ansotegui Gil
Josep Pon Farreny

20 de gener de 2017

Índex

Índex de taules

Índex de figures

1 Contingut

1.1 Contingut Principal

El contingut principal es centre en els següents fitxers:

- **Tree Predict:** Aquest és el fitxer principal de la pràctica, el que ens ha donat més feina per fer la realització. Són mètodes transparents de classificar observacions que després d'entrenar-lo amb dades amb característiques similars ho ordena fent servir sentències *if then* formant un **arbre**.
- **Bayesian Learning:** Aquest fitxer està implementat un altre mètode d'aprenentatge supervisat. Amb aquest mètode podem classificar si una paraula/frase pertany a una categoria o a una altra després d'entrenar-lo. Un exemple del seu funcionament a la vida real serie el Anti-SPAM del correu electrònic.
- **K-Means Cluster:** L'algorisme K-means és un mètode d'agrupament que té com a objectiu la partició d'un conjunt n observacions en k grups en el qual cada observació pertany al grup més proper a la mitjana. És un mètode d'aprenentatge no supervisat.

1.2 Contingut Secundari

A més a més dels fitxers esmentats anteriorment, també hem afegit un joc de proves per realitzar l'avaluació experimental. També hem afegit scripts utilitzats per extreure les dades i per crear els gràfics amb l'eina GNU-Plot.

2 Decisions de disseny

2.1 Tree Predict

En aquest document és on hem hagut d'implementar més funcions per la pràctica. En primer lloc, hem implementat les funcions **buildTree** i **buildTreeIterative**, que són les funcions encarregades de muntar l'arbre a partir de les dades d'entrada. La funció buildtree rep d'entrada un conjunt de dades i les divideix en columnes segons la impuresa, ja que a més decrement, millor són els subconjunts. Aquesta impuresa ve donada a l'utilitzar la funció amb les funcions de l'índex de Gini o l'Entropia. El resultat d'aquesta funció és un arbre amb les seves branques verdader i fals on es mostra el resultat de les preguntes realitzades. En la versió iterativa de l'algoritme hem aplicat els coneixements adquirits en l'assignatura d'Algorítmica i Complexitat on ens explicaven que un dels mètodes per passar una funció recursiva a iterativa era la utilització de piles per simular les crides recursives. És per això que hem implementat la nostra pròpia pila per fer aquesta funció. Una altra funció que hem hagut d'implementar és la funció **classifier** que consisteix en donat unes característiques i un arbre, ens dóna l'atribut que volem cercar. Aquesta funció la farem servir amb la funció de **testPerformance** que donat dos sets de característiques, entrenem un arbre amb el *trainingSet* i per cada conjunt de característiques del *testSet* cridem a l'anterior funció esmentada i comprovem si ens dóna l'argument correcte. Amb aquesta funció i amb funcions auxiliars és com hem pogut realitzar l'avaluació experimental. Per acabar, hem realitzat la funció **prune**. Aquesta funció és molt important quan tenim arbres que estan massa entrenats. Quan ens trobem en aquesta situació, pot baixar el rendiment del nostre arbre i per tant que les solucions que ens doni la funció classify sigui gairebé sempre errònia. Per fer això, per cada parell de nodes que tinguin el mateix pare, comprova si ajuntant-los obtindria un major decrement de la impuresa. Així podrem evitar que les prediccions que faci l'arbre siguin menys equivocades.

2.2 Bayesian Learning

En aquest arxiu, el qual hem realitzat la major part en les classes de laboratori de l'assignatura, està implementat un altre tipus de mètode d'aprenentatge supervisat. Aquest mètode, com he dit abans, a la vida real l'usen servidors de correu per implementar el seu filtre anti SPAM. En la pràctica, hem hagut de implementar la classe Naive Bayes i les seves funcions. Hem hagut de realitzar la funció **prob** i la funció **classify**. La funció prob la vam deixar a mitges a classe. Vam haver d'implementar el docprob que es una funció que mirava la quantitat de vegades que sortia un item en una categoria concreta. Tenint això i multiplicant-ho per catprob, funció ja implementada, ens computava la probabilitat d'una categoria sobre un item gràcies al teorema de Bayes: Màxim a Posteriori (MAP). L'altra funció que vam implementar va ser, com he dit, el classify que ens retorna la categoria on hauria d'estar aquell item o la categoria per defecte per quan no pot determinar-ho. Per fer aixo, calculavam la probabilitat d'una categoria sobre un item per totes les categories i ens quedàvem amb el millor resultat. Després comprovàvem que el valor de la categoria

més alta que ens havia donat, superes el valor de les altres multiplicat per el *threshold*. En cas que no fos així retornarem la categoria per defecte.

2.3 K-Means Cluster

Aquest tipus de mètode d'aprenentatge no supervisat ens retorna un arbre, com en els anteriors casos, però amb una serie de desavantatges com seria el alt còmput i que no acaba de separar del tot bé les dades entre si. Tenim una serie de valors representats en 2 dimensions i una serie de clusters. Aquests valors s'ajunten al cluster que tenen més aprop segons la seva distància euclideana ja implementada en el arxiu. Quan s'han afegit a un cluster, el cluster es posiciona en una posició mitja per a tots els valors. Si cap valor troba un cluster més aprop, s'estabilitza. Si no es així es van repetint els passos fins que s'arriba en una situació de calma.

3 Avaluació Experimental

Per la elaboració del procés experimental, hem creat i buscat uns jocs de proves. Tenim un joc de proves on majoritariament tenim variables de tipus categoria, després tenim un altre on majoritariament les variables són de tipus enter i un altre joc de proves amb variables de tipus boolea. També tenim un joc de proves amb multiples tipus de variables que ja l'hem estudiat a classe.

Percentatge TrainSet	Percentatge Good
0.1	19.04
0.2	37.5
0.3	44.89
0.4	35.71
0.5	40.0
0.6	53.57
0.7	47.61
0.8	57.14
0.9	57.14

Taula 1: Taula per al test de les variables de tipus categoria

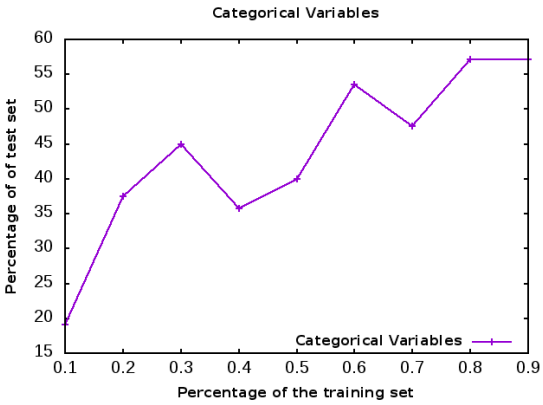

Figura 1: Gràfica per al test de les variables de tipus categoria

Percentatge TrainSet	Percentatge Good
0.1	47.12
0.2	47.74
0.3	44.85
0.4	48.27
0.5	50.51
0.6	50.0
0.7	56.89
0.8	56.41
0.9	50.0

Taula 2: Taula per al test de les variables de tipus enter

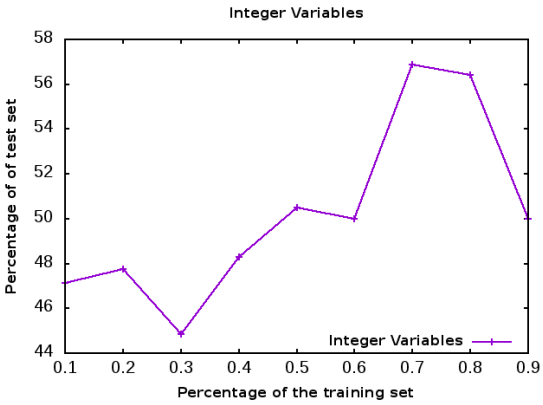

Figura 2: Gràfica per al test de les variables de tipus enter

Percentatge TrainSet	Percentatge Good
0.1	61.12
0.2	59.15
0.3	66.12
0.4	68.07
0.5	68.92
0.6	66.90
0.7	69.15
0.8	69.01
0.9	66.66

Taula 3: Taula per al test de les variables de tipus boolean

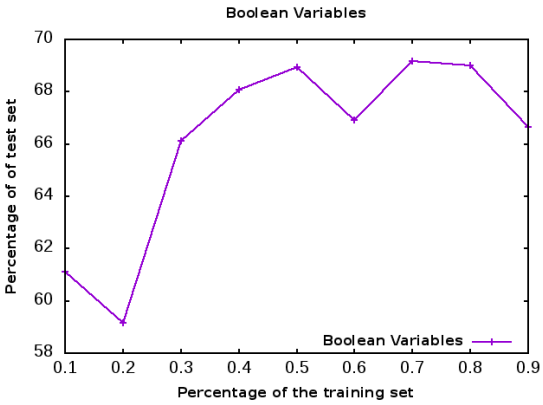

Figura 3: Gràfica per al test de les variables de tipus boolean

4 Conclusions

Després de efectuar aquesta avaluació experimental, podem analitzar amb quines variables el arbre de decisió funciona millor. En primer lloc podem veure que amb variables de categoria, la predicció millora com més informació té l'arbre incremental-ment. En canvi, podem veure que el gràfic amb variables booleanes incrementa exponencialment i és més estable que qualsevol altre. També podem extreure d'aquesta activitat empírica que els arbres amb variables de tipus enter o real, és a dir numèriques, són més difícils de preveure i per tant d'encertar. Els jocs de proves no tenen gaires valors, això també fa que les prediccions siguin més difícils i que si en falla una de predicció baixi més ràpid el rati

Percentatge TrainSet	Percentatge Good
0.1	13.33
0.2	15.38
0.3	33.33
0.4	50.0
0.5	12.5
0.6	42.85
0.7	40.0
0.8	75.0
0.9	100.0

Taula 4: Taula per al test de diferents tipus de variables

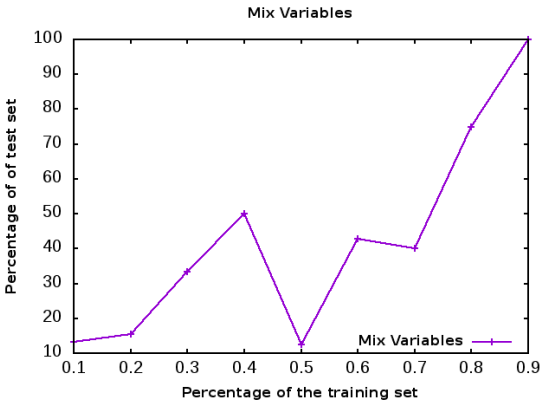

Figura 4: Gràfica per al test de diferents tipus de variables

d'encerts.També podem destacar que en el joc de proves on tenim variables de tots els tipus és molt volàtil ja que al haver poques dades ens surten resultats que no segueixen una línia però clarament, quan més entrenem a una maquina amb valors correctes i bons, la màquina preveurà millor sempre i quan els resultats no siguin sempre molts semblants ja que sinó haurem de fer prunning en el arbre.