Data Mining

Lab 3: Clasificarea datelor

Sumar:

- Problema clasificarii
- Clasificatori bazati pe instanțe (k- NearestNeighbor)
- Clasificatori bazati pe arbori de decizie (J48)
- Clasificatori bazati pe reguli de decizie (OneR, JRip)

1. Problema clasificării

Scopul unui model de clasificare: stabilirea clasei careia ii aparțin datele de intrare

Construirea unui clasificator = extragerea unui model de clasificare pornind de la date de antrenare; procesul de construire este denumit învățare sau antrenare.

Validarea unui clasificator: tehnica validării încrucișate

Evaluarea performanței: matrice de confuzie, acuratete, senzitivitate, specificitate, regăsire (recall), precizie (precision), F-measure.

Exercițiul 1: Vizualizarea suprafețelor de decizie

a) Activați BoundaryVisualizer (Weka GUI Chooser -> Visualization -> Boundary Visualizer)

b) Deschideti fișierul "iris2D.arff" (conține atributele petalLength și petalWidth din iris.arff). Alegeți succesiv unul din clasificatorii: IBk (k=1, k=2, k=5), J48, OneR, JRip, NaiveBayes si analizati rezultatele.

2. Clasificatori bazati pe instanțe (Instance Based Learning Classifiers)

Cei mai simpli clasificatori bazat pe instanțe sunt cei din categoria "cel mai apropiat vecin" (nearest neighbor)
Mod de construire:

- modelul de clasificare constă chiar din setul de date (nu există etapă propriu-zisă de antrenare)

Mod de utilizare:

- in procesul de clasificare, pentru o nouă instanță se identifică cele mai similar instanțe din setul de antrenare și clasa dominant va fi considerate ca fiind cea corespunzătoare noii date.

Implementare Weka: IB1 (se utilizează un singur vecin), IBk (se utilizează k vecini)

Exercițiul 2: Alegerea numărului de "vecini" la k-Nearest Neighbour

a) Deschideți in Weka fișierul "breast-w.arff". Pentru IBk identificați valoarea lui k (din {1,2,3,4,5,6,7,8,9,10}) care conduce la cel mai bun rezultat (Indicație: modificați valoarea parametrului k de la IBk).

b) Deschideți in Weka fișierul "glass.arff" și efectuați aceeași prelucrare

3. Clasificatori bazați pe arbori de decizie

Un arbore de decizie conține:

- Noduri interne (fiecare nod intern are asociat un atribut); fiecare nod intern are asociate conditii de ramificare, fiecare conditie corespunzând unei ramuri în arbore)
- Noduri frunză (fiecare nod frunză are asociată o clasă)

Mod de construire:

- pentru nodul rădăcină se identifică atributul și condițiile de ramificare caracterizate prin cel mai mare câștig informațional in cee ace privește clasificarea datelor din setul de date
- pentru fiecare dintre ramuri și setul de date corespunzător ramurii se aplică în mod recursiv aceeași strategie până este îndeplinită o condiție de oprire (s-a ajuns la un set care conține date din aceeași clasă sau la un set mic de date)

Observatie: un aspect important este cel referitor la complexitatea arborelui (sunt de preferat arbori cât mai simpli); simplificarea unui arbore (tree pruning) se poate realiza:

- in etapa de construire (stopând procesul de ramificare dacă numărul de elemente din setul current este mai mic decât un prag)
- după construirea arborelui, prin înlocuirea unor subarbori cu noduri frunză (in cazul în care performanța clasificatorului nu este alterată in mod semnificativ)

Mod de utilizare: pentru o anumită dată se identifică ramura din arborele de decizie care se potrivește cu valorile atributelor iar eticheta nodului frunză la care se ajunge reprezintă clasa asociată datei.

Implementare în Weka: Id3, J48

Exercițiul 3: Compararea algoritmilor de construire a arborilor de decizie

a) Deschideți in Weka fișierul "weather_nominal.arff". Comparați performanțele următorilor clasificatori: Id3, J48 (variantele "pruned" respective "unpruned" – se setează din lista de parametric de la J48)

b) Deschideți in Weka fișierul "weather_numeric.arff" și încercați să aplicați aceleași prelucrări

c) Deschideți in Weka fișierul "glass.arff" și analizati impactul asupra performanței clasificatorului J48 a numărului minim de date corespunzător unui nod frunză (se specific ca valoare a parametrului minNumObj)

Exercitiul 4: Construire unui arbore binar de decizie prin selecția datelor

Deschideți in Weka fișierul "iris2D.arff" și parcurgeți următoarele etape:

a) Selectați "User Classifier" (grupul Tree de la Classify)

b) In fereastra care se deschide (si in care apare nodul rădăcină) treceți la Data visualizer și selectați un subset de date cât mai "pur" (dacă e posibil cu date din aceeași clasă = puncte de aceeași culoare):

- de la Select Instance se alege Rectangle,
- se incadreaza punctele de selectat
- se dă click pe Submit

Efectul este că în arbore nodul rădăcină este ramificat în două noduri: unul corespunzător subsetului de date selectat, iar celălalt datelor rămase (se poate vizualiza în panoul Tree visualizer)

c) Se selectează un nou subset de date dintre cele rămase și se continua până se epuizează setul de date.

4. Clasificatori bazați pe reguli de decizie

Regulile de decizie sunt de forma IF <antecedent> THEN clasa, unde partea de antecedent conține condiții referitoare la valorile atributelor combinate prin operator se conjuncție sau disjuncție.

Mod de construire:

- Pornind de la un arbore de decizie, pentru fiecare ramură se poate construi o regulă de clasificare (condițiile din antecedent sunt combinate prin conjuncții)
- Direct pornind de la date prin algoritmi de acoperire ("covering algorithms")

Implementare in Weka:

- Pornind de la arbori de decizie: M5rules, PART
- Covering algorithms: OneR, JRip, PRISM

Exercitiul 5: analiza algoritmilor de extragere a regulilor de decizie

a) Deschideți fisierul "weather.nominal.arff" și aplicati următoarele modele de clasificare bazate pe reguli: OneR, JRip, PRISM, PART
b) Identificați cel mai bun clasificator bazat pe reguli pentru "weather.nominal.arff"