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)
- Clasificatori bazaţi pe reţele neuronale (perceptron multinivel)
- Clasificatori bazaţi pe vectori suport

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, curba receiver-operating characteristic (ROC) şi aria de sub curbă (AUC=area under curve).

Accuracy = (TP+TN)/(TP+TN+FP+FN) = nr date clasificate corect/ nr total de date
Sensitivity = TP/(TP+FN) (TP rate sau recall = rata de regăsire)
Specificity= TN/ (TN+FP) (TN rate), 1-specificitate=FP/(TN+FP) = FP rate
Precision = TP/(TP+FP) (nr cazuri real pozitive/ nr cazuri clasificate ca fiind pozitive)
F-measure=2*precision*recall/(precision+recall)

Curba ROC: TP rate (specifity) vs. FP rate (1-sensitivity). Se calculează perechi de valori (FP rate, TP rate) pentru fiecare felie din setul de antrenare (în contextual validării încrucişate) şi se unesc punctele (0,0), P1, …,Pn, (1,1) (unde Pi este punctul având a i-a valoare în ordine crescătoare a valorii FP rate). Ideal ar fi ca toate punctele să aibă coordonatele (0,1), iar aria de sub curbă să coincidă cu aria pătratului definit de (0,0) şi (1,1), adică 1.

Pentru seturile de date nebalansate (semnificativ mai multe exemple intr-o clasă în raport cu celelalte clase) se foloseşte coeficientul de corelaţie Matthew (MCC):

MCC=(TP*TN-FP*FN)/sqrt((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))

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 dominantă va fi considerată 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 curent 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" respectiv "unpruned" – se setează din lista de parametri 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 specifica 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 <eticheta clasa>, unde partea de antecedent conține condiții referitoare la valorile atributelor combinate prin operator de 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"

5. Modelele probabiliste

Permit estimarea probabilității ca o dată să aparțină unei clase ($P(C_k|(a_1,a_2,\ldots,a_n)$). Clasa corespunzătoare datei este cea pentru care probabilitatea este maximă.

Variante:

- Model Bayesian simplu (Naïve Bayes): presupune ca atributele nu sunt corelate (probabilitatea observării unei date este produsul probabilităților de a observa fiecare dintre valorile atributelor).
- Rețea cauzală (Bayesian Net sau Belief network): permite descrierea unui model sub forma unui graf orientat in care sunt specificate relații de cauzalitate

Exercițiul 6:

a) Deschideti fișierul "weather.nominal.arff". Aplicați clasificatorii Naïve Bayes și Bayes Net
b) Utilizați Bayes Network Editor (de la Weka GUI Choser -> Tools) pentru a construi o rețea bayesiană. Etape:
    - Selectare set date: Tools->Set Data
    - Plasare noduri (fiecare atribut, inclusiv cel de clasă va avea asociat un nod) folosind Edit->Add node
    - Specificare relații folosind Edit->Add arc (de exemplu: se plasează arc de la nodul corespunzător clasei la fiecare dintre nodurile associate atributelor)
    - Antrenare rețea (Tools->Learn network (invătarea structurii), Tools->Learn CPT (determinarea tabelelor de probabilitati))
    - Utilizare: se selectează valorile corespunzătoare instanței ce urmează să fie clasificată (folosind click dreapta pe fiecare nod asociat unui atribut și Set evidence); decizia se bazează pe valorile probabilităților corespunzătoare nodului asociat clasei

Obs: detalii privind utilizarea rețelelor bayesiene se găsesc la http://www.cs.waikato.ac.nz/~remco/weka_bn/

6. Rețele neuronale

Etape in proiectarea unei retele neuronale pentru clasificare:

- Stabilirea arhitecturii. In cazul rețelelor multi-nivel (multi-layer perceptrons) se fixează:
    - Număr de unitati de intrare = numarul atributelor
    - Număr de nivele/unitati ascunse – depinde de complexitatea problemei
    - Număr de unități de ieșire:
        - Clasificare binară: o unitate (rezultatul se interpretează folosind o valoare prag) sau două unități (unitatea care produce cea mai mare valoare va indica clasa)

- Stabilirea functiilor de activare. Pentru rețelele antrenate folosind algoritmul Backpropagation:
  - Unitățile ascunse au funcții de activare de tip sigmoidal (functia logistică sau tanh)
  - Unitățile de ieșire au funcții de activare de tip sigmoidal sau funcții liniare
- Alegerea algoritmului de antrenare și a parametrilor acestuia. In cazul algoritmului Backpropagation parametrii uzuali de control sunt:
  - Numărul de epoci de antrenare
  - Rata de învățare
  - Coeficientului termenului de tip „moment”

Implementare Weka: MultilayerPerceptron = rețea feedforward antrenată cu Backpropagation (varianta cu moment); arhitectura rețelei este implicit cu un nivel de unități ascunse; numărul de unitäti ascunse se poate specifica prin opţiunea hiddenLayers

Exercițiul 7: Analiza influentei arhitecturii rețelei asupra performanțelor clasificatorului

c) Deschideti fișierul “breast-w.arff”
d) Antrenați o rețea neuronală folosind setările standard (un nivel ascuns cu număr de unități ascunse cu K=(nr atribute+nr clase)/2 unitäti)
e) Comparați performanțele rețelei pentru următoarele valori ale parametrului Hidden Layers (H):
   a. ‘a’ (K=(nr atribute+nr clase)/2)
   b. ‘i’ (K=nr attribute)
   c. ‘o’ (K=nr clase)
   d. ‘t’ (K = nr atribute+nr clase)
   e. 4,2 (două nivele ascunse cu 4, respectiv 2 unități)
f) Analizați influența ratei de învățare si a coeficientului corespunzător termenului “moment”.
g) Comparați rezultatul obținut folosind MultilayerPerceptron cu cel obținut folosind o rețea de tip RBF – radial basis function (functions->RBF Network)

Obs. Prin activarea opțiunii GUI de la Multilayer Perceptron poate fi vizualizată arhitectura rețelei.

7. Clasificare folosind vectori suport (Support Vector Machines)

Varianta de clasificatori bazati pe vectori suport implementată in Weka folosește un algoritm rapid pentru rezolvarea problemei de optimizare (Sequential Minimal Optimization). In cazul clasificării in M clase problema se transformă în mai multe probleme de clasificare binară (corespunzătoare perechilor de clase)

Exercițiul 8:

c) Deschideți in Weka fișierul “breast-w.arff” si utilizati SVM (functions->SMO) ; analizați influența funcției nucleu asupra performanțelor clasificatorului (variante: PolyKernel, RBFKernel)
d) Deschideți in Weka fișierul “arrhythmia.arff” și efectuați aceeași prelucrare; incercați să rezolvați aceeași problemă utilizând o rețea neuronală.