Data Mining

Lab 3: Clasificarea datelor

Sumar:

- Problema clasificarii
- Clasificatori bazati pe instanțe (k- NearestNeighbor)
- Clasificatori bazati pe arbori de decizie (rpart)
- Clasificatori bazati pe reguli de decizie (OneR, JRip)
- Clasificatori probabilişti (Naïve Bayes)
- 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)

Kappa = (Accuracy – ExpectedAcc)/(1-ExpectedAcc) - ajustare a măsurii de acurateţe care ţine cont de proporţia cazurilor în care clasificatorul generează răspuns corect datorită întâmplării – este util în particular în cazul seturilor nebalansate de date.

> ExpectedAcc= frecvenţa(cazuri positive: obs şi real)+rate(cazuri negative: obs şi real)) = $(TP+FP)(TP+FN)/N^2 + (TN+FN)(TN+FP)/N^2$ (N=nr total de date)

Curba ROC: TP rate (sensitivitate) vs. FP rate (1-specificitate). 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), P_1, …,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))

Exerciţiul 1. Familiarizare cu pachetul "caret" – Classification And Regression Training (interfaţă unitare pentru peste 200 de variante de algoritmi de clasificare şi regresie implementaţi în diverse pachete R). Facilităţi principale:

- Partiţionare seturi de date în subset de antrenare şi subset de testare (function createDataPartition)
- Antrenarea unui model (funcţia train) şi definirea unei procedure de control a procesului de antrenare şi determinare a valorilor parametrilor de control (hiperparametri) (funcţia trainControl); permite specificare unei strategii de validare: bootstrapping (implicit – cu 25 de repetări), validare încrucişată etc.
- includerea unui pas de pre-procesare în faza de antrenare (e.g. parametrul preProc în funcţia train);exemple de preprocesare: centrare ("center"), scalare cu valoarea abaterii standard ("scale"), transformare PCA ("pca"), imputarea valorilor absente ("knnImpute")

(Punct de pornire: https://cran.r-project.org/web/packages/caret/vignettes/caret.pdf)

2. 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 şi condițiile de ramificare caracterizate prin cel mai mare câștig informațional in cee ace privește clasificarea datelor din setul de date
- pentru fiecare dintre ramuri şi setul de date corespunzător ramurii se aplică în mod recursiv aceeaşi 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.

Pachete R: rpart, party

Implementare în Weka: Id3, J48

Exerciţiul 2/ R: Construire, visualizare şi evaluarea performanţelor unor arbori de clasificare pentru: (a) iris dataset; (b) Titanic dataset; (c) Breast Cancer dataset; (d) [bioinfo] DNA (din pachetul mlbench). Utilizare pachete: party, rpart şi caret

Punct de pornire: DecisionTrees_PartyRpart.R, DecisionTrees_caret.R

Exerciţiul 2/Weka: 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)

3. 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")

Pachete R: caret include clasificatori bazaţi pe reguli din RWeka (JRip şi OneR)

Implementare in Weka:

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

Exerciţiul 3/ R: Construire, visualizare şi evaluarea unor clasificatori bazaţi pe regului pentru: (a) iris dataset; (b) Breast Cancer dataset; (c) Titanic dataset; (d) [bioinfo] DNA (din pachetul mlbench). Utilizre pachete: caret si RWeka

Punct de pornire: ClassificationRules.R

Exercitiul 3/Weka: 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, PART
b) Identificați cel mai bun clasificator bazat pe reguli pentru "weather.nominal.arff"

4. 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.

Pachete R: caret/ metoda knn, pachet class / metode knn, knn1

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

Exercițiul 4/ R: Construirea şi evaluarea performanţelor unui clasificator unui clasificator de tip kNN pentru: (a) Breast Cancer dataset; (b) Titanic dataset. (d) [bioinfo] DNA (din pachetul mlbench).

(i) Analizaţi tipului atributelor şi a preprocesărilor (e.g. standardization) asupra performanţei în clasificare şi asupra valorii determinate a parametrului de control (k = numărul de vecini utilizaţi în faza de predicţie)

(ii) Visualizaţi curba curba ROC (in cazul problemelor de clasificare binară) şi analizaţi valoarea ariei AUC.

Starting point: kNN.R

Exercițiul 4/ Weka: 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

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. Calculele se bazează pe formula lui Bayes: $P(C_k| a_1,a_2,\ldots,a_n)=P(a_1,a_2,\ldots,a_n | C_k)P(C_k)/ P(a_1,a_2,\ldots,a_n)$.

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.

Pachete R: pt Naïve Bayes: caret / metoda bn; pt reţele Bayesiene: pachet bnlearn

Exerciţiu 5/ R: Construiţi şi evaluaţi performanţa unui clasificator de tip naïve Bayes pt: (a) Breast Cancer dataset; (b) Titanic dataset; (d) [bioinfo] DNA (din pachetul mlbench).

Punct de pornire: NaiveBayes.R

Exercițiul 5/ Weka:

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 (invățarea 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"

R implementation: pachet nnet (inclus in caret) – rețea neuronală cu un nivel (cu diferite funcţii de activare la nivelul de ieşire – logistică (implicit), liniară, softmax) antrenată folosind un algoritm de tip Backpropagation ce minimizează MSE (mean squared error – implicit) sau entropia încrucişată (când e utilizată softmax ca funcţie de activare la nivelul de ieşire). Inclusă în Rattle.

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

Exerciţiul 6/ R: Construiţi şi evaluaţi performanţa unei rețele neuronale cu un nivel ascuns pt: (a) Breast Cancer dataset; (b) Titanic dataset; (d) [bioinfo] DNA (din pachetul mlbench).

Punct de pornire: NeuralNetworks.R

Exercițiul 6/Weka : Analiza influentei arhitecturii rețelei asupra performanțelor clasificatorului

a) Deschideti fișierul "breast-w.arff"
b) Antrenați o rețea neuronală folosind setările standard (un nivel ascuns cu număr de unități ascunse cu K=(nr atribute+nr clase)/2 unități)
c) 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)
d) Analizați influența ratei de învățare si a coeficientului corespunzător termenului "moment".
e) 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)

Exerciţiul 7/ R: Construiţi şi evaluaţi performanţa unui clasificator de tip SVM pt: (a) Breast Cancer dataset; (b) Titanic dataset; (d) [bioinfo] DNA (din pachetul mlbench). Analizaţi influenţa nucleului (metode svmRadial, svmLinear, svmPoly).

Punct de pornire: SVM.R

Exercițiul 7/Weka:

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.

Temă:

1. Utilizaţi Rattle sau R (la alegere) pentru a construi şi compara performanţele unor clasificatori bazaţi pe (i) arbori de decizie; (ii) vectori suport; (iii) reţele neuronale.

   Seturi de date (la alegere):
   - Glass (from mlbench);
   - LetterRecognition (from mlbench);
   - German credit data (GermanCredit in caret) [http://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29]
   - [bioinfo] Molecular Biology (Promoter Gene Sequences) Data Set [http://archive.ics.uci.edu/ml/datasets/Molecular+Biology+%28Promoter+Gene+Sequences%29]