Data Mining

Lab 2: Pre-procesarea datelor

Sumar: Pregătirea datelor pentru aplicarea tehnicilor de analiză:

1. Curățare (e.g. tratarea erorilor şi a valorilor absente)
2. Transformare
    a. Conversii intre diferite tipuri de atribute (e.g. discretizare, binarizare)
    b. Scalare
    c. Standardizare
3. Reducerea dimensiunii
    a. Selecția atributelor
    b. Selectia instanțelor

1. Curățarea datelor

Datele pot conține valori eronate sau absente cauzate fie de disfuncționalități ale dispozitivelor de măsurare/înregistrare, erori umane, refuzul de a completa anumite informații (în cazul informațiilor colectate pe bază de chestionare). In unele situații erorile pot fi detectate şi corectate în mod automat:

a) Valori care sunt in afara domeniilor de valori valide (de exemplu, vârstă negativă, valori ale temperaturii corporale, tensiunii arteriale etc in afara plajei de valori plauzibile). In astfel de situații există mai multe variante:
    - Eliminarea atributelor ce conțin astfel de valori (ex Weka: Filters->Unsupervised->attribute->NumericCleaner)
    - Eliminarea instanțelor ce conțin valori care nu sunt valide (ex Weka: Filters->Unsupervised->Instance->SubsetByExpression)
    - Eliminarea valorii eronate (şi interpretarea ei ca valoare absentă)

b) Valorile absente pot fi tratate in diferite moduri:
    - Se consider valoarea absentă ca un caz specific (de exemplu "?" in Weka files) – în acest caz simbolul utilizat pentru a marca o valoare absent poate fi tratat ca o valoare distinctă şi poate sa apară în modelele construite pe baza datelor (de exemplu în regulile de clasificare). Aceasta este abordarea implicită în Weka.
    - Eliminarea instanțelor ce conțin valori absente (ex Weka: Filters->Unsupervised->Instance->SubsetByExpression; specificând expresia not ismissing(ATT2) se vor elimina toate instanțele în care atributul 2 are valori absente)
    - Completarea valorilor absente cu valori estimate pe baza celor existente în setul de date (de exemplu cu media tuturor valorilor existente pentru atributul corespunzător sau cu media valorilor din instanțele similar celei care conține valoarea absent – similaritatea se măsoară folosind valorile asociate celorlalte attribute). Această abordare este cunoscută sub numele de tehnica imputării.

Exercițiul 1:

a) Deschideți in Weka fişierul "autos.arff" şi eliminați toate instanțele pentru care valoarea atributului 25 (price) este mai mare decât 10000 (Indicație: utilizați Filters->Unsupervised->Instance->SubsetByExpression cu expresia ATT25<=10000)

b) Deschideți in Weka fişierul "breast-w.arff". Identificați atributele ce conțin valori absente şi eliminați instanțele care conțin astfel de valori. Indicație: utilizați Filters-

I>Unsupervised->Instance->SubsetByExpression și o expresie de forma not ismissing(ATT<nr atribut> pentru a păstra doar instanțele complete.

2. Transformări

a) Conversii între tipuri

- Discretizare: transformarea atributelor care iau valori într-un domeniu continuu (atribute de tip real) în atribute care iau valori într-o mulțime discretă (de tip întreg, nominal sau ordinal). Cea mai simplă abordare este de a diviza intervalul de valori [min, max] in r subintervale de aceeași lungime (de exemplu [min, min+h), [min+h, min+2h), …[min+(r-1)h,max], unde h=(max-min)/r) și fiecare interval va fi asociat cu o valoare discretă (de exemplu 1, 2, 3,….r).
- Binarizare: permite transformarea unui atribut nominal într-un set de attribute binare (sau logice) sau transformarea unui subset de itemi (corespunzatori unei tranzactii) într-un vector binar. De exemplu, dacă un atribut A poate lua valori din mulțimea {v1,v2,v3} atunci el va fi înlocuit cu 3 atribute binare A1, A2 și A3.

b) Scalare

- Este utilă când atribute diferite iau valori în domenii care diferă semnificativ (de exemplu un atribut ia valori in [-0.1, 0.2] și altul ia valori in [10000, 20000])
- Cea mai simplă variantă de scalare este cea liniară prin care un atribut A având valoarea v din [mina,maxA] este transformat într-un atribut care ia valori in [0,1]:

  $s(v) = \frac{v - \min_A}{\max_A - \min_A}$, unde min și max corespund valorii minime respectiv celei maxime

c) Standardizare

- E utilă când interesează măsurarea abaterii valorilor față de medie în unități proporționale cu valoarea abaterii standard a datelor
- Prin standardizare, o valoare v este transformată după cum urmează:
- $st(v) = \frac{v - avg(A)}{stdev(A)}$, unde avg(A) reprezintă media valorilor atributului A iar stdev(A) este abaterea standard a valorilor atributului A

Exercițiu 2: Deschideți in Weka fișierul "breast-w.arff"

a) Transformati toate atributele numerice prin discretizare în: (i) 10 subintervale; (ii) 5 subintervale și salvați rezultatele în fișierele "breast-w-discrete10.arff" și "breast-w-discrete5.arff". Indicație: Utilizați Filter->Unsupervised->Attribute->Discretize (setați numărul de subintervale ca valoare a parametrului bins (-B)

b) Utilizati Weka-Experimenter pentru a analiza acuratețea următorilor clasificatori: ZeroR, OneR, J48, NaiveBayes pentru seturile de date "breast-w.arff", "breast-w-discrete10.arff" și "breast-w-discrete5.arff"

Exercițiu 3: Deschideți in Weka fișierul "car.arff".

a) Transformati toate atributele nominale prin binarizare (Filter->Unsupervised->Attribute->NominalToBinary) si salvați datele transformate în fișierul carBinary.arff

b) Utilizați Weka-Experimenter pentru a analiza acuratețea următorilor clasificatori: ZeroR, OneR, J48, NaiveBayes aplicați datelor din car.arff și carBinary.arff

Reminder Lab 1: cum se utilizează Weka-Experimenter

1. Selectati Experimenter din panoul Weka (Weka chooser)
2. Click pe butonul New pentru a crea un nou experiment
3. Adaugati seturile de date - Add datasets (e.g. car.arff and carBinary.arff
4. Adaugati algoritmii - Add algorithms: zeroR, oneR, J48, naiveBayes (oneR şi zeroR sunt din grupul "Rules", naiveBayes este din grupul "Bayes" iar J48 este din grupul "Tree")
5. Rulaţi experimentul (Run)
6. Analizaţi rezultatele (Analyze):
    a. Click pe Experiment
    b. Aplicati testul statistic (Perform the statistical test)
    c. Interpretati rezultatele testului statistic (primul algoritm specificat, zeroR , va fi considerat ca metodă de referinta)
        i. v/ /* se interpretează după cum urmează: v= număr de cazuri (datasets) pentru care metoda curentă este semnificativ mai bună decât cea de referinţă; / /= număr de cazuri (datasets) pentru care metoda curentă este la fel de bună ca cea de referinţă; *= număr de cazuri (datasets) pentru care metoda curentă este semnificativ mai slabă decât cea de referinţă;;

Exerciţiu 4. Open in Weka the file "iris.arff" and "breast-w.arff"

a) Aplicaţi scalarea asupra atributelor (Indicatie: utilizati Filters->Unsupervised->Attribute->Normalize) şi salvaţi fişierele ca "iris_scaled.arff" şi "breast-w_scaled.arff"
b) Aplicaţi standardizarea supra atributelor (Indicatie: utilizati Filters->Unsupervised->Attribute->Standardize) şi salvaţi fişierele ca "iris_standardized.arff" and "breast-w_standardized.arff"
c) Analizaţi impactul scalării şi standardizării asupra următorilor clasificatori: ZeroR, OneR, J48, NaiveBayes, IB1, MultilayerPerceptron (din grupul Functions). Obs: schimbaţi pe False opţiunea NormalizeAttributes de la MultilayerPerceptron.

Exerciţiu 5 (temă). Inmultiti toate valorile primului atribut din "breast-w.arff" cu 100 (Hint: use Filters->unsupervised->attribute->MathExpression) dupa care efectuaţi din nou toate prelucrările specificate la Exerciţiul 4.

3. Reducerea dimensiunii datelor

Motivaţie: anumite atribute sau instante pot fi irelevante sau redundante. De exemplu un atribut care are aceeasi valoare pe intregul set de date ar trebui ignorat. Două atribute puternic correlate (e.g. A1=2*A2) sunt redundante şi ar fi sufficient să se reţină doar unul dintre ele.

- Reducerea numărului de attribute poate reduce costul de calcul dar poate să şi conducă la o imbunătăţire a performanţei modelului construit pe baza datelor (e.g. clasificator)
- Reducerea numărului de instanţe din setul de antrenare poate reduce timpul necesar antrenării (sau clasificării, in cazul clasificatorilor "leneşi") dar poate să şi îmbunătăţească performanţele modelului (în special în cazul seturilor de date dezechilibrate)

Reducerea numărului de attribute poate fi realizată prin:

- Selecţia atributelor:
    - Selector de tip "filtru": selecţia se bazează doar pe analiza proprietăţilor datelor fiind independent de prelucrarea care va fi ulterior aplicată datelor.

- Selector de tip "wrapper": selecția se realizează in contextul utilizării datelor intr-un proces specific de analiză (calitatea subsetului selectat este evaluată prin prisma performanței unui model de analiză extras din date)

- Proiectia datelor pe un spațiu de dimensiune mai mică (e.g. Analiza componentelor principale - Principal Component Analysis)

Reducerea instanțelor poate fi realizată prin:
- Eliminarea unor instanțe (pe baza unor reguli specifice)
- Selecția instanțelor (utilizând selectie aleatoare cu sau fără revenire)

3.1. Selector de tip filtru

Selectia atributelor poate fi realizată in două moduri:
- Căutând în spațiul submulțimilor de atribute (in cazul a n atribute spațiul de căutare are cardinalul 2^n-2 – se ignoră mulțimea vidă și întreg setul de atribute)
- Prin ierarhizarea atributelor în concordanță cu relevanța lor și selectarea celor mai relevante (în cazul a n attribute ierarhizate $A_1,A_2,\ldots A_n$, sunt n-1 variante de analizat: {A1}, $\{A_1,A_2\}$, $\{A_1,A_2,A_3\}\ldots\{A_1,A_2,\ldots,A_{n-1}\}$

Pentru fiecare dintre variante trebuie specificate:
- O măsură a relevanței unui atribut sau subset de atribute
  - *Cazul nesupervizat:* informația privind clasa atributului nu este folosită pentru a evalua relevanța (analiza se bazează in principal pe corelațiile sau similaritățile dintre attribute)
  - *Cazul supervizat*: informația privind clasa este utilizată (analiza se bazează pe corelația dintre valorile atributelor si eticheta clasei)
- O tehnică de căutare:
  - Căutare exhaustivă (toate subseturile de atribute sunt analizate)
  - Căutare greedy (forward/backward): se adaugă/elimină cel mai bun/slab atribut identificat la momentul curent
  - Cautare aleatoare
  - Cautare bazată pe o ierarhizare (atributele sunt selectate în ordinea dictate de o ierarhie stabilită anterior)

Exercițiu 6. Deschideți in Weka fisierul iris.arff, dupa care:

a) Aplicati o tehnica de selectie a atributelor (e.g. Select attributes -> CfsSubsetEval, GreedyStepwise) și salvați setul redus de date (click dreapta și Save reduced date)
b) Aplicați o metodă de ierarhizare a atributelor (e.g. Select attributes -> InfoGainAttributeEval, Ranker). Comparați rezultatul cu cel obținut la a).
c) Analizați impactul reducerii datelor asupra performanțelor clasificatorilor (utilizând Experimenter la fel ca in exercițiile 2-3).

3.2. Proiecția datelor – analiza componentelor principale

Scop: se transformă datele (schimbând sistemul de axe de coordonate) astfel încât să fie eliminată corelația dintre atribute și se păstrează doar acele atribute care conservă cât mai mult din variabilitatea datelor. Această transformare poate fi realizată prin analiza în componente principale (Principal Component Analysis) parcurgând următoarele etape:

- Se calculează matricea de covarianță C a setului de date (daca datele au n atribute atunci matricea va avea dimesniunea nxn); in practică se obișnuiește să se centreze datele inainte de a construe matricea (se scade din fiecare instanță media setului de date)
- Se calculează valorile și vectorii proprii ai matricii C
- Se ordonează descrescător valorile proprii
- Se selecteaza m<n vectori proprii (cei corespunzători celor mai mari valori proprii); m se alege astfel incât datele transformate să conserve cât mai mult din variabilitatea datelor inițiale (e.g. 95%)
- Se proiectează datele pe spațiul definit de cei m vectori proprii

Exercise 7. Deschideți în Weka fisierul iris.arff

a) Aplicați Principal Component Analysis (PCA) asupra datelor (e.g. Select attributes -> Principal Components, Ranker și salvați setul redus de date (click dreapta pe rezultat și Save transformed data)
b) Visualizați datele transformate și comparați-le cu cele inițiale
c) Analizați impactul transformării PCA asupra performanțelor clasificatorilor folosind Experimenter (vezi Exercitii 2-3).

Exercitiu 8. Aplicați transformarea PCA asupra datelor "arrhythmia.arff" și analizați rezultatele.