Data Mining

Lab 5:

Reguli de asociere
Regresie neliniară. Rețele de tip RBF

---

1. Reguli de asociere

Exemplu (problema cosului de cumparaturi). Se consideră un set de tranzacţii (T_1, T_2,…, T_n), fiecare tranzacţie conţinând un set de produse achiziţionate. De exemplu:

T_1: {pâine, lapte, apă}
T_2: {pâine,carne, apă}
T_3: {pâine, unt, carne, apă}
T_4: {fructe, apă}

Se pune problema identificării produselor care sunt frecvent cumpărate împreună şi a unor reguli de asociere de forma IF pâine AND apă THEN carne.

Regulile de asociere sunt de forma IF A THEN B. Termenul A joacă rol de antecedent, iar B rol de consecinţă (totuşi nu exprimă relaţii de cauzalitate ci doar de co-ocurență).

Dintr-un set de tranzacţii pot fi extrase numeroase reguli – e necesar să poata fi evaluată relevanţa lor pentru a fi ierarhizate.

Pentru evaluarea relevanţei unei reguli se folosesc cel puţin două mărimi:

- Suport (support): supp(A->B) = numărul de tranzacţii ce conţin pe A şi B/ numărul total de tranzacţii
- Incredere (confidence): conf(A->B)=numărul de tranzacţii ce conţin pe A şi B/ numărul de tranzacţii ce conţin pe A

Exemplu: IF pâine şi apă THEN carne

A={pâine, apă}, B={carne}
Supp(A->B)=2/4=0.5
Conf(A->B)=2/3=0.6

Obs: pe lângă aceste măsuri se folosesc şi indicatori ai noutăţii regulii (cât este de interesantă sau neobişnuită regula); un exemplu de astfel de indicator este cel denumit lift:

Lift(A->B)=prob(A,B)/(prob(A)prob(B))

(probabilitatea se estimează prin frecvenţa relativă)

Regula e considerată interesantă dacă valoarea măsurii lift este mare. Daca valoarea e apropiată de 1 aceasta sugerează că A şi B nu sunt corelate deci nu poate fi extrasă o regulă utilă de forma A->B

Exemplu: R=IF pâine AND carne THEN apă
Conf(R)=2/2=1
Lift(R)=0.5/(0.5*1)=1

Algoritm de extragere a regulilor de asociere din date (algoritmul APRIORI)

Date de intrare: set de tranzacţii (fiecare tranzacţie conţine o listă de entităţi)
Parametri de control:

- Prag pentru suport minim (ex: 0.2)
- Nivel minim de încredere (ex: 0.8)

Structura generală a algoritmului Apriori:

Pas 1: identificare subseturi cu suport semnificativ (mai mare decât pragul minim) – "frequent itemsets"; identificarea acestor subseturi se bazează pe:

- Identificarea subseturilor frecvente cu un singur element (lista L_1)
- FOR k=1,K DO construirea listei L_k cu subseturi frecvente având k elemente pornind de la lista L_{k-1} (subseturi frecvente cu k-1 elemente – se combină subseturi frecvente de dimensiune k-1 care au k-2 elemente comune)

Pas 2: construirea regulilor prin partiţionarea subseturilor identificate la pasul 1 în două părţi (o parte pentru antecedentul regulii şi o parte pentru consecinţă); se reţin doar regulile care au nivelul de încredere mai mare decat pragul specificat.

Exercițiul 1/ Rattle+R. Determinați toate regulile de asociere cu support mai mare decât 0.2 și nivel de încredere mai mare decât 0.7 pornind de la setul de tranzacții folosit la curs (fișier datasets/transactions.csv).

Rattle: deschideți fișierul, marcați toate atributele ca input, utilizați Associate (setați valori pentru Support și Confidence) apoi utilizați Show Rules pentru vizualizarea regulilor.
Obs: untick Partition (nu e nevoie de separarea datelor în subseturi de antrenare/ validare/ testare întrucât e vorba de antrenare nesupervizată).

R: citiți fișierul cu tranzacții (utilizând read.transactions), extrageți cele mai frecvente itemset-uri (utilizând eclat) și regulile de asociere (utilizând apriori). Indicație: AssociationRules_ExampleLecture9.r

Exercițiul 2/R. Incărcați setul de date Groceries din pachetul arules și

a) Identificați cele mai frecvente 10 item-uri (indicație: utilizați itemFrequencyPlot)
b) Determinați itemset-urile cu suport mai mare sau egal cu 0.1 (indicație: utilizați eclat)
c) Determinați regulile de asociare cu suport mai mare sau egal cu 0.005 și coefficient de încredere 0.7 (indicație: utilizați apriori)

Exercițiul 3. Determinați reguli de asociere având support cel puțin 0.2 și nivel de încredere cel puțin 0.7 pornind de la lista de tranzacții supermarket.arff

R: Indicație: datele din fișierul de tip arff trebuie convertite într-o listă de tranzacții (e.g. TransactionData <- as(ListData, "transactions")

Weka:

a) Deschideţi în Weka setul de date supermarket.arff
b) Determinaţi reguli de asociere folosind Associate->Apriori şi valorile implicite ale parameterilor
c) Aplicaţi acelasi algoritm pentru alte valori ale pragului pentru support (lowerBoundMinSupport=0.2) si pentru incredere (minMetric=0.75).

2. Modele de regresie

2.1. Regresie liniara

In modelele liniare dependenţa dintre variabila(variabilele) prezise şi cele predictor este descrisă printr-o funcţie liniară de forma Y=WX. In funcție de numărul de componente ale lui X și Y există mai multe variante de regresie:

- Regresie simplă: o variabilă predictor (intrare) și o variabilă țintă (ieșire) (e.g. $y=w_1*x+w_0$, unde x și y sunt valori scalare)
- Regresie multiplă: mai multe variabile predictor și o variabilă țintă (e.g. $y=w_kx_k+\ldots+w_1x_1+w_0$)
- Regresie multiplă multivariată: mai multe variabile predictor și mai multe variabile țintă (e.g. Y=WX); in majoritatea cazurilor o problemă de regresie multiplă multivariate poate fi transformată în mai multe probleme de regresie multiplă univariate.

Parametrii modelului (elementele vectorului/matricii W) se determină pornind de la date folosind o tehnica de minimizare a sumei pătratelor erorilor. Funcția R destinată acestei prelucrări este lm.

2.2. Regresie neliniară

Dacă valorile de ieșire nu depend liniar de valorile de intrare atunci trebuie utilizat un model neliniar. Modelele neliniare pot fi estimate pornind de la date folosind metode de fitare (e.g. funcția nls din R), arbori de regresie (e.g. funcția rt din R), rețele RBF (e.g. funcția rbf din R).

Exercițiul 4/R. (bioinfo) Estimați dependența dintre rata de reacție enzimatică și concentrația de enzimă (ecuația Michaelis-Menten) folosind funcția nls. Exemplu de start: SimpleNonlinearRegression_Bioinfo.r

Exercițiul 5. Estimarea prețului unei mașini pornind de la caracteristicile mașinii (set de date autoPrice).

Rattle: Deschideți fișierul autoPrice.arff în Rattle, ignorați primele două attribute (symbolling și normalized-losses) și setați class ca atribut țintă (corespunde prețului mașinii, însă denumirea class e implicită pentru atributele țintă în Weka). Comparați performanța următoarelor modele de regresie:

- Regresie liniară (Model -> Linear)
- Arbori de regresie (Model -> Tree)
- Rețele neuronale (Model -> Neural Net)

Care dintre caracteristicile mașinii au impact semnificativ asupra prețului?

Exercitiul 3. (tot pentru fisierul autoPrice.arff)

a) Utilizati Class->Functions->MultilayerPerceptron cu valorile implicite ale parametrilor. Analizaţi valorile corespunăatoare coeficientului de corelaţie şi erorii Correlation Coeficient si Mean Absolute Error.
b) Identificati în categoria Class->Trees varianta care permite construirea unui arbore de regresie

Obs. Versiunile de Weka mai mici de 3.8 include si implementare a retelelor RBF (RBF Network).

Studiu de caz: predicția frecvenței diferitelor tipuri de alge din râuri pe baza caracteristicilor râurilor și a concentrației unor substanțe chimice. Exemplu de start: CaseStudy_Algae.r

Temă.

1. Identificaţi modele de regresie adecvate pentru a estima consumul de combustibil (exprimat prin "miles per gallon") în funcție de caracteristici ale mașinii (setul de date: autoMPG.arff) și comparați performanțele.
2. Implementați o rețea RBF pentru estimarea estimarea consumului de combustibil (setul de date: autoMPG.arff) pornind de la exemplul din exRBF.r (implementarea trebuie modificată pentru o problemă de regresie neliniară).