

PARTEA A III-A.

CLASIFICARE

Prediction is very difficult, especially if it's about the future.
Niels Bohr

5. ALGORITMI DE CLASIFICARE. GENERALITĂȚI

5.1. Introducere

Clasificarea datelor reprezintă procesul prin care unui tuplu i se atribuie una sau mai multe etichete dintr-o mulțime de etichete dată. Clasificarea este un proces în două etape. În prima etapă se va construi un clasificator care va descrie un set predefinit de date sau concepte. Această etapă se numește și etapă de învățare sau etapă de antrenare, în care un algoritm de clasificare construiește clasificatorul învățând pe baza unor tupluri de date (set de antrenament) și etichetele asociate claselor acestora. Un tuplu X este reprezentat de un vector de atribute n -dimensional $X = (x_1, x_2, \dots, x_n)$ care reprezintă măsurători efectuate în cadrul tuplului asupra a n atribute $A_1, A_2, \dots, A_n$. Se presupune că fiecare tuplu aparține unei clase, care la rândul ei, are o etichetă de clasă. Eticheta clasei este o valoare discretă și nu este ordonată. În contextul clasificării, tuplurile se numesc exemple, instanțe sau obiecte. Deoarece eticheta clasei este dată pentru fiecare tuplu în etapa de antrenare, acest pas este un caz de învățarea supervizată.

Astfel, prima etapă a procesului de clasificare poate fi văzută ca și învățarea unei funcții de mapare $y=f(X)$ care poate prezice pentru un tuplu X dat eticheta y a unei clase. Această mapare este reprezentată sub forma unor reguli de clasificare, arbori de decizie sau formule matematice.

A doua etapă în procesul de clasificare o reprezintă utilizarea modelului elaborat în prima etapă, în clasificarea unui set de test. Pentru a măsura eficiența clasificatorului trebuie efectuată o aproximare a acurateței de clasificare a acestuia. Dacă am măsura acuratețea de clasificare pe baza setului de antrenament valorile obținute ar fi foarte optimiste iar efectul de „over-fit” poate apărea (supraînvățare – algoritmul învață unele anomalii existente în setul

de antrenament care nu pot fi aplicate cazului general). Din acest motiv se creează un set de test care constă din tupluri și clasele asociate acestora, altele decât cele prezente în setul de antrenament. Astfel, acuratețea clasificatorului este calculată ca procentaj al numărului de tupluri corect clasificate de către clasificator. Metrici pentru evaluarea acurateței clasificatorilor au fost prezentate în secțiunea 2.6.

În secțiunile următoare vom prezenta o taxonomie posibilă pentru algoritmi de clasificare, prezentând pentru fiecare categorie clasificatorii reprezentativi pentru acea categorie.

5.2. Algoritmi stohastici

Cel mai reprezentativ clasificator stohastic este clasificatorul bayesian. El prezice cu o anumită probabilitate apartenența la o clasă dată, cum ar fi de exemplu, probabilitatea ca un document să facă parte dintr-o anumită clasă dată. Clasificarea bayesiană se bazează pe teorema lui Bayes, și este descrisă în secțiunea 5.2.1. Studiile care au comparat algoritmi de clasificare au identificat clasificatorul bayesian simplu cunoscut sub numele de clasificator bayesian naiv (Naïve Bayes Classifier) ca având, în practică, performanțe bune atunci când este aplicat la seturi de date mari, cum este cazul de față, unde există multe documente, fiecare reprezentat prin multe atribute.

Clasificatorul bayesian simplu pleacă de la premisa „naivă” că atributele pentru o anumită clasă sunt independente unele de altele. Această presupunție se numește independență condițională de clasă. În cadrul acestui clasificator se pornește de la această presupunție, pentru a simplifica calculele.

5.2.1. Clasificarea bayesiană

Fie Y o variabilă pentru o clasă (categorie) care poate lua valorile $\{y_1, y_2, \dots, y_m\}$.

Fie X o instanță a unui vector cu n atribute $\langle x_1, x_2, \dots, x_n \rangle$ și x_k o valoare posibilă pentru X și x_{ij} o valoare posibilă pentru x_i . Pentru clasificarea de tip Bayes calculăm probabilitățile $P(Y = y_i | X = x_k)$ pentru $i = \overline{,m}$. Asta ar însemna calcularea tuturor probabilităților pentru fiecare categorie, pentru fiecare instanță posibilă din spațiul de instanțe – ceea ce este foarte greu de calculat pentru un set rezonabil de date.

Practic, pentru a determina categoria lui x_k , trebuie să determinăm pentru fiecare y_i probabilitatea:

$$P(Y = y_i | X = x_k) = \frac{P(Y = y_i)P(X = x_k | Y = y_i)}{P(X = x_k)} \quad (5.1)$$

Probabilitatea $P(X = x_k)$ poate fi determinată, deoarece mulțimea categoriilor este completă și disjunctă. Rezultă imediat relația de echilibru de mai jos:

$$\sum_{i=1}^m P(Y = y_i | X = x_k) = \sum_{i=1}^m \frac{P(Y = y_i)P(X = x_k | Y = y_i)}{P(X = x_k)} = 1 \quad (5.2)$$

așadar:

$$P(X = x_k) = \sum_{i=1}^m P(Y = y_i)P(X = x_k | Y = y_i) \quad (5.3)$$

Probabilitatea $P(Y = y_i)$ poate fi ușor aproximată având în vedere faptul că dacă n_i exemple din D se regăsesc în y_i atunci $P(Y = y_i) = n_i / |D|$, unde D reprezintă mulțimea documentelor din setul de antrenament.

Probabilitatea $P(X = x_k | Y = y_i)$ trebuie estimată (deoarece există 2^n posibile instanțe pentru a calcula probabilitatea). De aceea, dacă presupunem că atributele unei instanțe sunt independente (condițional independente), atunci:

$$P(X | Y) = P(x_1, x_2, \dots, x_n | Y) = \prod_{i=1}^n P(x_i | Y) \quad (5.4)$$

Astfel, trebuie să calculăm doar $P(x_i | Y)$ pentru fiecare pereche posibilă "valoare atribut"-"categorie".

Dacă Y și toate X_i sunt binare, atunci trebuie să calculăm doar $2n$ valori:

$$P(x_i = true | Y = true) \text{ și } P(x_i = true | Y = false) \text{ pentru fiecare } X_i$$

$$P(x_i = false | Y) = 1 - P(x_i = true | Y)$$

față de 2^n valori, dacă nu am presupune independența atributelor.

Practic, dacă setul de date D conține n_k exemple din categoria y_k și n_{ij} din aceste n_k exemple au a j -a valoare pentru atributul x_i pe x_{ij} atunci estimăm că:

$$P(X_i = x_{ij} | Y = y_k) = \frac{n_{ij}}{n_k} \quad (5.5)$$

TEXT MINING

Tehnici de clasificare și clustering al documentelor

Radu G. Crețulescu, Daniel I. Morariu

PARTEA I. INTRODUCERE

1. INTRODUCERE

1.1. Structura cărții

2. PROCESAREA AUTOMATĂ A DOCUMENTELOR DE TIP TEXT. GENERALITĂȚI

2.1. Data mining

2.1.1. Preprocesarea datelor

2.1.1.1. Curățirea datelor

2.1.1.1.1. Completarea valorilor lipsă

2.1.1.1.2. Netezirea zgomotului

2.1.1.2. Integrarea și transformarea datelor

2.1.1.2.1. Integrarea datelor

2.1.1.2.2. Transformarea datelor

2.1.1.3. Selectarea și reducerea datelor

2.1.2. Analiza datelor

2.1.3. Evaluarea și prezentarea pattern-urilor rezultate

2.2. Text mining

2.2.1. Analiza datelor text și regăsirea informației

2.2.2. Metode de regăsire a informației

2.2.3. Asocierea între cuvinte cheie și clasificarea documentelor

2.2.4. Alte tehnici de indexare pentru regăsirea textului

2.3. WWW mining

2.3.1. Mineritul structurii paginilor web

2.3.2. Mineritul link-urilor pentru identificarea paginilor web autoritare

2.3.3. Mineritul utilizării web

2.3.4. Construirea informațiilor de bază pe mai multe niveluri web

2.3.5. Clasificarea automată a documentelor web

2.4. Clasificare versus Clustering

2.4.1. Învățare supervizată și nesupervizată

2.4.2. Clasificare și analiza clasificării

2.4.3. Clustering și analiza clusterilor

2.4.4. Cerințe cheie pentru algoritmi de clustering

2.5. Metrice de similaritate a documentelor text

2.5.1. Structurarea datelor

2.5.1.1. Matricea de date

2.5.1.2. Matricea de disimilaritate

- 2.5.2. Disimilaritate și similaritate
- 2.5.3. Distanțe uzuale
- 2.5.4. Tipuri de variabile utilizate în clasificare/clustering
 - 2.5.4.1. Variabile scalate într-un anumit interval
 - 2.5.4.2. Variabile standardizate
 - 2.5.4.3. Variabile binare (dihotomice)
 - 2.5.4.3.1. Matricea de disimilaritate pentru variabile binare
 - 2.5.4.4. Variabile nominale
- 2.6. Evaluarea algoritmilor de clasificare/clustering
 - 2.6.1. Măsuri externe de validare a clusteringului și a clasificării
 - 2.6.2. Măsuri de validare internă a clusterilor
- 2.7. Seturi de date utilizate
 - 2.7.1. Setul de date Reuters
 - 2.7.1.1. Alegerea documentelor pentru antrenare - testare
 - 2.7.1.2. Setul A1
 - 2.7.1.3. Setul T1
 - 2.7.1.4. Setul T2
 - 2.7.2. Setul de date RSS –Web

PARTEA A II-A. CLUSTERING

3. ALGORITMI DE CLUSTERING. GENERALITĂȚI

- 3.1. O posibilă taxonomie
 - 3.1.1. Algoritmi partiționali (sau metode partiționale)
 - 3.1.1.1. Metoda k-Means
 - 3.1.1.2. Metoda k-Medoids
 - 3.1.2. Metode ierarhice
 - 3.1.2.1. Algoritmi aglomerativi ierarhici (HAC)
 - 3.1.2.1.1. Single link
 - 3.1.2.1.2. Complete link
 - 3.1.2.1.3. Average link
 - 3.1.2.1.4. Centroid link
 - 3.1.2.1.5. Metoda lui Ward
 - 3.1.2.1.6. SAHN (Sequential, Agglomerative, Hierarchical and Nonoverlapping)
 - 3.1.2.2. Algoritmul BIRCH
 - 3.1.2.3. Algoritmul CURE
 - 3.1.2.4. Algoritmi divizivi
 - 3.1.3. Metode bazate pe ordinea cuvintelor - Suffix Tree Clustering (STC)
 - 3.1.3.1. Pas 1. Construcția arborelui de sufixe
 - 3.1.3.2. Pas 2. Selectarea nodurilor de bază
 - 3.1.3.3. Pas 3. Unirea clusterilor de bază similari
 - 3.1.3.4. Pas 4. Etichetarea clusterilor

- 3.1.4. Metode bazate pe densități
- 3.1.5. Metode de tip grid-based
- 3.1.6. Metode bazate pe modele
- 3.2. Algoritmi ierarhici. HAC – implementarea AGNES
- 3.3. Algoritmi partiționali. K-Medoids

4. CLUSTERINGUL DOCUMENTELOR

- 4.1. Modele de reprezentare utilizate
 - 4.1.1. Reprezentarea utilizând modelul Vector Space Model – VSM
 - 4.1.1.1. Indexarea documentelor
 - 4.1.1.2. Tipuri de reprezentare a termenilor
 - 4.1.2. Reprezentarea utilizând modelul Suffix Tree Document Model – STDM
- 4.2. Metodologia de lucru
- 4.3. Metrice pentru calculul matricei de similaritate și metode de evaluare
- 4.4. Rezultate obținute pe seturile RSS
 - 4.4.1. Rezultatele obținute de algoritmul HAC - reprezentare VSM
 - 4.4.2. Rezultatele obținute de algoritmul HAC - reprezentare STDM
 - 4.4.3. Rezultatele obținute de algoritmul k-Medoids cu reprezentare VSM
 - 4.4.4. Rezultatele obținute de algoritmul k-Medoids cu reprezentare STDM
 - 4.4.5. Comparații între algoritmi de clustering și între modurile de reprezentare

PARTEA A III-A. CLASIFICARE

5. ALGORITMI DE CLASIFICARE. GENERALITĂȚI

- 5.1. Introducere
- 5.2. Algoritmi stohastici
 - 5.2.1. Clasificarea bayesiană
 - 5.2.2. Antrenarea clasificatorului Bayes
 - 5.2.3. Testarea clasificatorului Bayes
 - 5.2.4. Rezultate obținute cu clasificatorului Bayes
- 5.3. Algoritmi de învățare bazați pe regula Backpropagation
 - 5.3.1. Modelul neuronului artificial
 - 5.3.2. Arhitectura rețelelor neuronale
 - 5.3.3. Învățarea rețelelor neuronale
 - 5.3.3.1. Regula de învățare Boltzmann
 - 5.3.3.2. Regula de învățare Hebb
 - 5.3.3.3. Regula de învățare competitivă
 - 5.3.3.4. Reguli de învățare prin corecție a erorii (“error-correction rules”)
 - 5.3.4. Metoda Backpropagation
 - 5.3.4.1. Perceptronul [Vint07]
 - 5.3.4.2. Perceptroni multistrat cu funcție de activare neliniară
 - 5.3.4.3. Perceptronul multistrat
 - 5.3.5. Algoritmul de învățare Backpropagation

- 5.3.5.1. Pasul forward
- 5.3.5.2. Pasul backward
- 5.3.6. Cercetări privind evitarea saturării ieșirii neuronilor
- 5.4. Algoritmi evoluționiști. Algoritmi genetici
 - 5.4.1. Codificarea cromozomilor și problema de optimizare
 - 5.4.2. Metode de alegere a cromozomilor
 - 5.4.2.1. Metoda „Roulette Wheel” (ruleta)
 - 5.4.2.2. Alegerea utilizând metoda lui Gauss
 - 5.4.3. Operatorii genetici utilizați
 - 5.4.3.1. Selecția
 - 5.4.3.2. Mutația
 - 5.4.3.3. Crossover
- 5.5. Algoritmi bazați pe nuclee. Support Vector Machine
- 5.6. Clasificatori hibrizi. Metaclasificatori

6. CLASIFICAREA DOCUMENTELOR

- 6.1. Evaluarea clasificatorilor de tip SVM
 - 6.1.1. Problema limitării metaclasificatorului cu clasificatori de tip SVM
 - 6.1.2. O primă tatonare a problemei
- 6.2. Soluții explorate pentru îmbunătățirea metaclasificatorului bazat pe clasificatoare de tip SVM
 - 6.2.1. Soluția introducerii unor noi clasificatori SVM
 - 6.2.2. Soluția alegerii altei clase
 - 6.2.3. Soluția adăugării unui clasificator de alt tip
 - 6.2.3.1. Adaptarea clasificatorului Bayes pentru utilizarea în metaclasificator
 - 6.2.3.2. Compararea clasificatorului Bayes adaptat (BNA) cu clasificatorii de tip SVM
 - 6.2.3.3. Antrenarea clasificatorilor pe setul A1 și testarea pe setul T1
 - 6.2.3.4. Antrenarea pe setul A1 și testarea pe setul T2
 - 6.2.3.5. Antrenarea și testarea pe setul T2
- 6.3. Metode de selecție a clasificatorilor
 - 6.3.1. Selecția bazată pe vot majoritar (MV). Rezultate
 - 6.3.2. Selecția bazată pe distanța euclidiană (SBED). Rezultate
 - 6.3.3. Selecția bazată pe distanța cosinus (SBCOS). Rezultate
- 6.4. Arhitecturi neadaptive propuse și dezvoltate
 - 6.4.1. Metaclasificator cu ponderi predefinite. Evaluare de tip Eurovision. Rezultate obținute.
 - 6.4.1.1. Metaclasificator neadaptiv bazat pe sumă
 - 6.4.1.2. Metaclasificator neadaptiv bazat pe sumă normalizată
 - 6.4.1.3. Metaclasificator neadaptiv bazat pe sumă ponderată
 - 6.4.1.4. Cercetări privind alte variante de ponderare a elementelor vectorilor
 - 6.4.1.4.1. Înjumătățirea ponderii
 - 6.4.1.4.2. Ponderi mici descrescătoare linear

6.4.2. Metaclasificator cu ponderi calculate. Design Space Exploration cu algoritmi genetici. Rezultate obținute.

6.5. Arhitecturi adaptive propuse și dezvoltate

6.5.1. Metaclasificatoare bazate pe similaritate

6.5.1.1. Rezultate obținute în cazul selecției bazate pe distanța euclidiană

6.5.1.2. Rezultatele obținute în cazul selecției bazate pe distanța cosinus

6.5.2. Metaclasificator bazat pe algoritmul Backpropagation

6.5.2.1. Influența numărului de neuroni de pe stratul ascuns

6.5.2.2. Influența coeficientului de învățare