

Capitolul 8

Data mining – date corelate

Reprezentarea datelor

- ◆ Vom continua să considerăm modelul de date "coșuri de produse" și vom vizualiza datele ca o matrice booleană unde:
- ◆ linii=coșuri și
- ◆ coloane=articole.

Asertiuni

1. Matricea este foarte rară; aproape peste tot 0.
2. Numărul de coloane (articole) este suficient de mic pentru a putea stoca în memoria centrală ceva per coloană dar suficient de mare astfel încât nu putem stoca ceva per pereche de articole în memoria centrală (aceeași aserțiune pe care am făcut-o până acum privind regulile de asociere).

Asertiuni

3. Numărul de linii este atât de mare încât nu putem stoca întreaga matrice în memorie chiar profitând de faptul ca e rară și comprimând-o (din nou aceeași aserțiune ca întotdeauna).
4. Nu suntem interesați de perechile sau mulțimile de coloane cu larg suport; în schimb dorim perechile de coloane puternic corelate.

Aplicatii

- ◆ În timp de aplicațiile de marketing sunt interesate doar de produsele de larg consum (nu merită să se încerce promovarea obiectelor pe care oricum nu le cumpără nimeni), există un număr de aplicații care se potrivesc cu modelul de mai sus, de interes fiind în special problema **perechilor** de coloane / articole inclusiv de consum restrâns dar puternic corelate:

Aplicatii

1. Liniile și coloanele sunt pagini de web; (r, c) = 1 înseamnă că pagina corespunzătoare liniei r conține o legătură către pagina coloanei c . Coloanele similare pot fi pagini despre același domeniu.
2. La fel ca (1) dar pagina corespunzătoare coloanei c conține legături către pagina liniei r . Acum, coloane similare pot reprezenta copii multiple (mirror) ale unei pagini.

Aplicatii

3. Linii = pagini web sau documente; coloane = cuvinte. Coloane similare reprezintă cuvinte care apar aproape mereu împreună, e.g. "frază".
4. La fel ca (3) dar liniile sunt propoziții [iar coloanele sunt pagini web]. Coloane similare pot indica copii multiple ale unei pagini sau plagiat.

Similaritate

- ◆ Am vorbit despre similaritatea coloanelor fără să dam o măsură cantitativă a acesteia.
- ◆ Definiție: Să ne gândim la o coloană ca la mulțimea liniilor pentru care coloana conține 1. Atunci *similaritatea* a două coloane $C1$ și $C2$ este $\text{Sim}(C1, C2) = |C1 \cap C2| / |C1 \cup C2|$.

Similaritate

- ◆ $|x|$ reprezintă cardinalul mulțimii x , deci:
- ◆ $|C1 \cap C2|$ reprezintă numărul de linii în care ambele coloane au valoarea 1
- ◆ $|C1 \cup C2|$ reprezintă numărul de linii în care macar una dintre coloane are valoarea 1.

Exemplu

0	1
1	0
1	1
0	0
1	1
0	1

$= 2/5 = 40\%$ similară

Problema

- ◆ Deoarece matricea conține foarte multe linii și coloane testarea similarității este laborioasă deoarece matricea nu încapă în memoria centrală.
- ◆ Ar fi preferabil ca fiecare coloană să fie reprezentată de o cantitate mult mai mică de informație având proprietatea ca testul de similaritate efectuat pe aceste informații să fie relevant pentru similaritatea coloanelor.

Signatura

- ◆ Ideia principală: Se mapează ("dispersează") fiecare coloană C într-o cantitate mică de date numită *signatură* lui C , (notă $\text{Sig}(C)$) astfel încât:
- ◆ $\text{Sig}(C)$ este suficient de mică pentru ca signaturile tuturor coloanelor să încapă în memoria centrală și să se poată efectua testul de similaritate.

Signatura

- ◆ Cand un mod de calcul pentru semnături este bun:
- ◆ Coloanele $C1$ și $C2$ sunt puternic similare dacă și numai dacă $Sig(C1)$ și $Sig(C2)$ sunt puternic similare. (dar de notat că este nevoie să definim "similaritatea" pentru semnături).

Exemplu (prost)

- ◆ O idee care însă nu funcționează: Se iau aleator 100 de linii și șirul de 100 de biți ai coloanelor pentru acele linii este semnatura fiecărei coloane.
- ◆ De ce?

Exemplu (prost)

- ◆ Motivul pentru care ideea nu funcționează este că matricea este presupusă ca fiind f. rară deci multe coloane vor avea semnături identice formate doar din 0 chiar dacă ele nu sunt deloc similare.

Convenție utilă

- ◆ Dându-se două coloane $C1$ și $C2$, ne vom referi la liniile lor ca fiind de patru tipuri – a, b, c, d – în funcție de biții lor pe aceste coloane, după cum urmează:

Tip	C1	C2
a	1	1
b	1	0
c	0	1
d	0	0

Convenție utilă

- ◆ De asemenea vom utiliza a pentru "numărul de linii de tip a ", ș.a.m.d.
- ◆ De notat că $Sim(C1, C2) = a/(a+b+c)$.
- ◆ Dar cum cele mai multe linii sunt de tip d , într-o selecție de, să spunem, 100 de linii alese aleator toate vor fi de tip d , deci similaritatea coloanelor doar pentru aceste 100 de linii nici nu este definită.

Ce vom prezenta în continuare

- ◆ Dispersia de tip Min și dispersia de tip k-min – metode de calcul semnături
- ◆ Dispersia senzitivă la localizare – o metodă prin care se minimizează numărul de perechi de semnături care se testează pentru similaritate

Dispersia de tip Min (Minhashing)

- ◆ Este o metoda de calcul signaturi
- ◆ Signatura unei coloane va fi un sir de numere intregi.
- ◆ Să ne imaginăm liniile permutate într-o ordine aleatoare. "Dispersăm" fiecare coloana C în $h(C)$, numărul primei linii în care coloana C are un 1.
- ◆ În felul acesta obținem primul intreg al signaturii

Dispersia de tip Min (Minhashing)

- ◆ Probabilitatea ca $h(C1) = h(C2)$ este $a/(a+b+c)$ deoarece valoarea de dispersie este aceeași dacă prima linie cu un 1 în vreuna din coloane este de tip a și este diferită dacă prima astfel de linie este de tip b sau c .
- ◆ De notat că această probabilitate este aceeași cu $Sim(C1, C2)$.

Dispersia de tip Min (Minhashing)

- ◆ Dacă repetăm experimentul cu o nouă permutare a liniilor de un număr mare de ori, să zicem 100, obținem o signatură constând din 100 de numere de linii pentru fiecare coloană.
- ◆ "Similaritatea" acestor liste (fracțiune a pozițiilor în care ele sunt egale) va fi foarte apropiată de similaritatea coloanelor.

Dispersia de tip Min (Minhashing)

- ◆ Observație importantă: nu trebuie să permutăm fizic liniile, ceea ce ar duce la multe treceri prin întreaga cantitate de date.
- ◆ În schimb citim liniile într-o ordine oarecare și dispersăm fiecare linie (numărul acesteia) utilizând (să zicem) 100 de funcții de dispersie diferite.

Dispersia de tip Min (Minhashing)

- ◆ Pentru fiecare coloană memorăm cea mai mică valoare a funcției de dispersie a unei linii în care acea coloană are un 1, independent pentru fiecare dintre cele 100 de funcții de dispersie.
- ◆ După parcurgerea tuturor liniilor vom avea pentru fiecare coloană primele linii în care coloana are 1 dacă liniile ar fi fost permutate în ordinea dată de fiecare dintre cele 100 de funcții de dispersie.

Exemplu

Funcții	Tabela	Signaturi
1 4 3	1 0 1 0	2 1 2 1
3 2 4	1 0 0 1	2 1 4 1
7 1 7	0 1 0 1	1 2 1 2
6 3 6	0 1 0 1	
2 6 1	0 1 0 1	
5 7 2	1 0 1 0	
4 5 5	1 0 1 0	

Exemplu

Similaritati

	1-3	2-4	1-2	3-4
Col-Col	0,75	0,75	0	0
Sig-Sig	0,67	1,00	0	0

Signaturi

2	1	2	1
2	1	4	1
1	2	1	2

Deci signaturi similare =>

Coloane similare.

Aici diferentele sunt mai mari datorita dimensiunii tabelului (prea mica)

Dispersia senzitiva la localizare

◆(eng: Locality-Sensitive Hashing - LSH)

◆Problema: avem signaturile fiecărei coloane în memoria centrală iar signaturi similare înseamnă cu mare probabilitate coloane similare,

◆Pot fi totuși atât de multe coloane încât a face ceva care este proportional cu pătratul numărului de coloane, chiar și în memoria centrală, este prohibitiv.

Dispersia senzitiva la localizare

◆Dispersia senzitivă la localizare (DSL, LSH în engleză) este o tehnică destinată a fi utilizată în memoria centrală pentru a aproxima mulțimea de perechi de coloane similare cu o complexitate mult mai mică decât cea pătratică.

◆Scopul: în timp proporțional cu numărul de coloane să se elimine cea mai mare parte a perechilor de coloane din mulțimea posibilelor perechi similare.

Etape

◆ Este deci o metoda de a micșora numărul de perechi de signaturi care se compara pentru similaritate.

◆ Etapele sunt:

1. Considerăm signatura ca fiind o coloană de întregi.
2. Partiționăm liniile signaturilor în *benzi*, să spunem r benzi de câte r linii fiecare.

Etape

3. Dispersăm coloanele din fiecare bandă în intrări ale unei tabeli de dispersie. O pereche de coloane este o pereche-candidat dacă ambele sunt dispersate în aceeași intrare în vreo bandă.
4. După identificarea candidatelor se verifică fiecare pereche candidat (C_i , C_j) examinând pentru similaritate $Sig(C_i)$ și $Sig(C_j)$.

Exemplu

- ◆Pentru a vedea efectul DSL să considerăm date cu 100.000 de coloane și signaturi constând din 100 de întregi fiecare.
- ◆Signaturile ocupă 40Mb de memorie, nu atât de mult la standardele actuale.
- ◆Să presupunem că vrem perechile care sunt 80% similare.
- ◆Vom examina signaturile în loc de coloane, deci în mod real vom identifica coloanele ale căror *signaturi* sunt 80% similare – deci nu chiar același lucru.

80%

- ◆ Dacă două coloane sunt 80% similare atunci probabilitatea ca ele să fie identice în una dintre benzile de 5 întregi este $(0,8)^5 = 0,328$.
- ◆ Probabilitatea ca ele să nu fie identice în *nici una* dintre cele 20 de benzi este $(1 - 0,328)^{20} = 0,00035$.
- ◆ Astfel toate mai puțin aproximativ 1/3000 dintre perechile cu semnături 80% similare vor fi identificate ca și candidate.

40%

- ◆ Acum, să presupunem că două coloane sunt doar 40% similare.
- ◆ Atunci probabilitatea ca ele să fie identice într-o bandă este $(0,4)^5 = 0,01$
- ◆ Probabilitatea ca ele să fie identice în cel puțin una dintre cele 20 de benzi nu este mai mare ca 0,2 ($\leq 20 \cdot 0,01$)
- ◆ Astfel, putem ignora cel puțin 4/5 dintre perechi care nu vor deveni candidate dacă 40% este similaritatea tipică a coloanelor.

Concluzie

- ◆ În fapt, cele mai multe perechi vor fi cu mult mai puțin decât 40% similare astfel încât realmente eliminăm o parte importantă a perechilor de coloane care nu sunt similare.

Dispersia k-Min

- ◆ Dispersia Min ne cere să dispersăm fiecare număr de linie de k ori dacă vrem semnături de k întregi.
- ◆ În loc de asta, în cazul *dispersiei k-min* dispersăm fiecare linie o singură dată și, pentru fiecare coloană luăm ca semnătură numerele primelor k linii în care acea coloană are un 1.

Dispersia k-Min

- ◆ Pentru a vedea de ce similaritatea acestor semnături este aproape aceeași cu similaritatea coloanelor din care derivă să examinăm figura:

	Sig1		Sig2	
100 de 1	1		0	
	1		1	
	0		1	
	
	0		1	
	1		0	
	1		1	
				100 de 1

Dispersia k-Min

- ◆ În figura sunt semnăturile *Sig1* și *Sig2* pentru coloanele *C1* și respectiv *C2*.
- ◆ S-a presupus ca liniile au fost permutate în ordinea valorilor funcției de dispersie.
- ◆ Liniile de tip d (în care nici o coloană nu are 1) sunt omise.
- ◆ Astfel, vedem doar liniile de tip a , b și c și indicăm că o linie este în semnatura printr-un 1.

Dispersia k-Min

- ◆ Să presupunem că $c \geq b$ astfel încât situația tipică (pentru $k = 100$) este cea din figura: cele 100 de linii pentru prima coloană includ unele linii care nu sunt printre cele 100 ale celei de-a doua coloane.
- ◆ Atunci o estimare a similarității lui *Sig1* și *Sig2* poate fi calculată astfel:

$$| \text{Sig1} \cap \text{Sig2} | = 100a / (a+c)$$

Dispersia k-Min

- ◆ Justificare: în medie, primele 100 de linii din *C2* care sunt și în *C1* este $a/(a+c)$.
- ◆ De asemenea:

$$| \text{Sig1} \cup \text{Sig2} | = 100 + 100c / (a+c)$$
- ◆ Motivul este că toate cele 100 de linii din *Sig1* sunt în reuniune.
- ◆ În plus, liniile din *Sig2* care nu sunt în *Sig1* sunt de asemenea în reuniune, iar acestea sunt în medie în număr de $100c/(a+c)$.

Dispersia k-Min

- ◆ Astfel, similaritatea lui *Sig1* cu *Sig2* este:

$$| \text{Sig1} \cap \text{Sig2} | / | \text{Sig1} \cup \text{Sig2} | = a / (a + 2c)$$
- ◆ Observăm că dacă c este apropiat ca valoare de b atunci similaritatea signaturilor este apropiată de similaritatea coloanelor.
- ◆ În fapt, dacă două coloane sunt foarte similare, atunci b și c sunt ambele mici comparate cu a și similaritățile signaturilor și coloanelor *trebuie* să fie apropiate.

DSL Hamming

- ◆ În cazul în care coloanele nu sunt rare ci au aprox. 50% de 1 nu avem nevoie de dispersie Min;
- ◆ O colecție aleatoare de linii servește în acest caz ca semnătură.
- ◆ *DSL Hamming* construiește o serie de matrici, fiecare având jumătate din numărul de linii ale precedentei, aplicând operatorul SAU (OR) la câte două linii succesive din matricea precedentă

DSL Hamming

- ◆ Nu exista mai mult de $\log n$ matrici, unde n este numărul de linii. Numărul total de linii în toate matricile este $2n$ și pot fi calculate toate într-o singură trecere prin matricea originală, stocându-le pe cele mari pe disc.

DSL Hamming

- ◆ În fiecare matrice, se aleg ca perechi candidat acele coloane care:
 - Au o densitate de 1 să zicem între 20% și 80%
 - E posibil să fie similare bazat pe testul DSL

DSL Hamming

- ◆ Observăm că plaja de densitate 20% - 80% ne garantează că două coloane care sunt cel puțin 50% similare vor fi considerate împreună în cel puțin o matrice, în afara cazului nefericit în care densitatea lor relativă se schimbă din cauza operației OR care combină doi de 1 într-unul singur.

DSL Hamming

- ◆ O a doua trecere prin datele originale confirmă care dintre candidate sunt întradevăr similare.
- ◆ Aceasta metodă exploatează o idee care poate fi folositoare și în alte părți: coloanele similare au un număr similar de 1, deci nu are rost compararea coloanelor al căror număr de 1 este foarte diferit

Bibliografie

- ◆ J.D.Ullman - CS345 --- Lecture Notes, Capitolul 3 (Overview of Data Mining, Association-Rules, A-Priori Algorithm)
<http://infolab.stanford.edu/~ullman/cs345-notes.html>

Sfârșitul capitolului 8