Capitolul 8

Data mining – date corelate

Reprezentarea datelor

- Vom continua să considerăm modelul de date "coşuri de produse" şi vom vizualiza datele ca o matrice booleană unde:
- linii=coşuri şi
- 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 (aceeaşi aserţiune 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 şi comprimând-o (din nou aceeaşi aserţiune ca întotdeauna).
4. Nu suntem interesaţi de perechile sau mulţimile de coloane cu larg suport; în schimb dorim perechile de coloane puternic corelate.

Aplicatii

- În timp de aplicaţiile 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 aplicaţii 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 şi coloanele sunt pagini de web; *(r, c)* = 1 înseamna că pagina corespunzatoare liniei *r* conţine o legătură către pagina coloanei *c*. Coloanele similare pot fi pagini despre acelaşi domeniu.
2. La fel ca (1) dar pagina corespunzătoare coloanei *c* conţine 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 împreuna, e.g. “fraze”.
4. La fel ca (3) dar liniile sunt propoziţii [iar coloanele sunt pagini web]. Coloane similare pot indica copii multiple ale unei unei pagini sau plagiat.

Similaritate

- Am vorbit despre similaritatea coloanelor fara sa dam o masura cantitativa a acesteia.
- Definitie: Să ne gândim la o coloană ca la multimea liniilor pentru care coloana conţine 1. Atunci *similaritatea* a două coloane C1 şi C2 este

$$Sim(C1, C2) = |C1 \cap C2| / |C1 \cup C2|.$$

Similaritate

- |x| reprezinta cardinalul multimii x, deci:
- $|C1 \cap C2|$ reprezinta numarul de linii in care ambele coloane au valoarea 1
- $|C1 \cup C2|$ reprezinta numarul de linii in care macar una dintre coloane are valoarea 1.

Exemplu

0	1	
1	0	
1	1	= 2/5 = 40% similare
0	0	
1	1	
0	1	

Problema

- Deoarece matricea contine foarte multe linii si coloane testarea similaritatii este laborioasa deoarece matricea nu incape in memoria centrala.
- Ar fi preferabil ca fiecare coloana sa fie reprezentata de o cantitate mult mai mica de informatie avand proprietatea ca testul de similaritate efectuat pe aceste informatii sa fie relevant pentru similaritatea coloanelor.

Signatura

- Ideia principală: Se mapează ("disperseaza") fiecare coloană *C* într-o cantitate mică de date numita *signatura lui C, (notatie Sig(C))* astfel încât:
- *Sig(C)* este suficient de mică pentru ca signaturile tuturor coloanelor să încapă în memoria centrală si sa se poata efectua testul de similaritate.

Signatura

- Cand un mod de calcul pentru signaturi este bun:
- Coloanele *C1* şi *C2* sunt puternic similare dacă şi numai dacă *Sig(C1)* şi *Sig(C2)* sunt puternic similare. (dar de notat că este nevoie să definim “similaritatea” pentru signaturi).

Exemplu (prost)

- O idee care însă nu funcţionează: Se iau aleator 100 de linii şi şirul de 100 de biti ai coloanelor pentru acele linii este signatura fiecărei coloane.
- De ce?

Exemplu (prost)

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

Conventie utila

- Dându-se două coloane *C1* şi *C2*, ne vom referi la liniile lor ca fiind de patru tipuri – *a, b, c, d* – în funcţie de biţii lor pe aceste coloane, dupa cum urmează:

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

Conventie utila

- De asemenea vom utiliza *a* pentru "numărul de linii de tip *a*", ş.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 selecţie 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 in continuare

- Dispersia de tip Min si dispersia de tip k-min – metode de calcul signaturi
- Dispersia senzitiva la localizare – o metoda prin care se minimizeaza numarul de perechi de signaturi care se testeaza 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.
- In felul acesta obtinem 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 aceeasi dacă prima linie cu un 1 în vreuna din coloane este de tip *a* şi este diferită dacă prima astfel de linie este de tip *b* sau *c*.
- De notat că această probabilitate este aceeaşi cu *Sim(C1, C2)*.

Dispersia de tip Min (Minhashing)

- Dacă repetăm experimentul cu o noua permutare a liniilor de un număr mare de ori, să zicem 100, obţinem o signatură constând din 100 de numere de linii pentru fiecare coloană.
- “Similaritatea” acestor liste (fracţiune a poziţiilor în care ele sunt egale) va fi foarte apropiată de similaritatea coloanelor.

Dispersia de tip Min (Minhashing)

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

Dispersia de tip Min (Minhashing)

- Pentru fiecare coloană memorăm cea mai mică valoare a funcţiei de dispersie a unei linii în care acea coloană are un 1, independent pentru fiecare dintre cele 100 de funcţii 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 data de fiecare dintre cele 100 de functii de dispersie.

Exemplu

Functii

1	4	3
3	2	4
7	1	7
6	3	6
2	6	1
5	7	2
4	5	5

Tabela

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

Signaturi

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

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 tabelei (prea mica)

Dispersia senzitiva la localizare

- (eng: Locality-Sensitive Hashing - LSH)
- Problema: avem signaturile fiecarei coloane în memoria centrală iar signaturi similare inseamna cu mare probabilitate coloane similare,
- Pot fi totuşi atât de multe coloane încât a face ceva care este proportional cu pătratul numărului de coloane, chiar şi 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 mulţimea de perechi de coloane similare cu o complexitate mult mai mică decat cea pătratică.
- Scopul: în timp proporţional cu numărul de coloane să se elimine cea mai mare parte a perechilor de coloane din mulţimea posibilelor perechi similare.

Etape

- Este deci o metoda de a micsora numarul de perechi de signaturi care se compara pentru similaritate.
- Etapele sunt:

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

Etape

3. Dispersăm coloanele din fiecare banda în intrări ale unei tabele de dispersie. O pereche de coloane este o pereche-candidat dacă ambele sunt dispersate în aceeaşi intrare în vreo bandă.
4. Dupa identificarea candidatelor se verifică fiecare pereche candidat (*Ci, Cj*) examinând pentru similaritate *Sig(Ci)* şi *Sig(Cj)*.

Exemplu

- Pentru a vedea efectul DSL să considerăm date cu 100.000 de coloane şi 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 caror *signaturi* sunt 80% similare – deci nu chiar acelaşi lucru.

80%

- Dacă doua 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 ele să nu fie identice în *nici una* dintre cele 20 de benzi este (1-0,328)20 = 0,00035.
- Astfel toate mai puţin aproximativ 1/3000 dintre perechile cu signaturi 80% similare vor fi identificate ca şi candidate.

40%

- Acum, să presupunem că doua 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 puţin una dintre cele 20 de benzi nu este mai mare ca 0,2 (<=20*0,01)
- Astfel, putem ignora cel putin 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 puţin 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 signaturi de *k* întregi.
- În loc de asta, în cazul *dispersiei k-min* dispersăm fiecare linie o singură data şi, pentru fiecare coloana luăm ca signatură numerele primelor *k* linii în care acea coloană are un 1.

Dispersia k-Min

- Pentru a vedea de ce similaritatea acestor signaturi este aproape aceeaşi cu similaritatea coloanelor din care derivă să examinam figura:

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

Dispersia k-Min

- In figura sunt signaturile *Sig1* şi *Sig2* pentru coloanele *C1* şi respectiv *C2.*
- S-a presupus ca liniile au fost permutate în ordinea valorilor funcţiei de dispersie.
- Liniile de tip *d* (în care nici o coloana nu are 1) sunt omise.
- Astfel, vedem doar liniile de tip *a*, *b* şi *c* şi indicăm că o linie este în signatura printr-un 1.

Dispersia k-Min

- Să presupunem că $c \geq b$ astfel încât situaţia 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 similarităţii lui *Sig1* şi *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 şi în *C1* este a/(a + c).
- De asemenea:

  $$| Sig1 \cup Sig2 \ | = 100 + 100c \ / \ (a + c)$$

- 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:

$$| \mathbf{Sig1} \cap \mathbf{Sig2} \; | \; / \; | \mathbf{Sig1} \cup \mathbf{Sig2} \; | =$$

$$\mathbf{a / (a + 2c)}$$

- 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* şi *c* sunt ambele mici comparate cu *a* şi similarităţile signaturilor şi 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 colecţie aleatoare de linii serveşte in acest caz ca signatură.
- *DSL Hamming* construieşte 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 şi pot fi calculate toate într-o singură trecere prin matricea originală, stocandu-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 intre 20% şi 80%
  - E posibil să fie similare bazat pe testul DSL

DSL Hamming

- Observam ca plaja de densitate 20% - 80% ne garantează că doua coloane care sunt cel puţin 50% similare vor fi considerate împreună în cel puţin o matrice, în afara cazului nefericit în care densitatea lor relativă se schimbă din cauza operaţiei 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 exploateaza o idee care poate fi folositoare şi în alte parţi: coloanele similare au un număr similar de 1, deci nu are rost compararea coloanelor al caror 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

Sfârşitul capitolului 8