Vertikális felosztás

Definíció: Adott egy r reláció. Annak $F_r = \{r_1, r_2, \ldots, r_n\}$ a felosztása, ahol minden $r_i = \prod_{x_i \cup K} r$ (x_i azon attribútumok halmaza, amiket r_i-ben látni szeretnénk, K r elsődleges kulcsa)

s:

<u>ID</u>	név	cím	telszám

$\underbrace{\text{ID}\quad\text{név}}_{x_1}\quad\underbrace{\text{cím}\quad\text{telszám}}_{x_2}$

x_1 x_2 => $s_1 = \prod_{x1 \cup K} s = \prod_{ID, név} s$

$s_2 = \prod_{x2 \cup K} s = \prod_{cím,\ telszám,\ ID} s$

A felosztás célja, hogy olyan attribútumok kerüljenek egy töredékbe, melyeket gyakran kérdezünk le egyszerre. A kulcsok minden töredékbe bele kell tenni, hiszen az alapján tudjuk majd újra összeilleszteni a teljes relációt.

Partícionáló algoritmus

Bemenetek:

$Q = \{q_1, \ldots, q_q\}$ lekérdezések halmaza

$R = (a_1, \ldots, A_n)$ felosztandó r reláció sémája

$$use(q_j, A_j) = \begin{cases} 1, \text{ ha } q_j \text{ lekérdezés használja } A_j \text{ attribútumot} \\ 0 \text{ egyébként} \end{cases}$$

(használati indikátor)

$acc(q_k)$ = q_k futási gyakorisága (olyan attribútumokat akarunk egy halmazba rakni, amiket gyakran használunk együtt)

$aff(A_i, A_j)$ = affinitás = $\sum_{k: use(q_k, A_i)=1 \cap use(q_k, A_j)=1} acc\ (q_k)$

(összefüggőségi mérték)

milyen gyakran hivatkozzák lekérdezések a 2 attribútumot együttesen (A_i és A_j konstans időintervallumon belül hányszor vagy milyen gyakran hivatkozódik együtt)

[aff(A_i, A_j)] -> összefüggőségi mátrix

$$\begin{array}{cc} & \begin{array}{ccc} A_1 & \ldots & A_n \end{array} \\ \begin{array}{c} \text{A1} \\ \\ \text{An} \end{array} & \begin{bmatrix} aff(A_1, A_1) & \cdots & aff(A_1, A_n) \\ \vdots & \ddots & \vdots \\ aff(A_n, A_1) & \cdots & aff(A_n, A_n) \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{ccc} A_1 & \ldots & A_n \end{array} \\ \begin{bmatrix} a(1,1) & \cdots & a(1,n) \\ \vdots & \ddots & \vdots \\ a(n,1) & \cdots & a(n,n) \end{bmatrix} \end{array}$$

Algoritmus

I. **Mátrix létrehozása (affinitás mátrix)**
II. **Mátrix oszlopainak permutálása**
III. **Sorok átrendezése oszlopnak megfelelően**
IV. **Felosztási pont meghatározása**

II. lépésnél a cél a globális összefüggőségi mérték (AM = affinity measure) maximalizálása

bond(i,j) = i. és j. oszlop szorzatösszege

$AM^* = \sum_{i=1}^{n} \sum_{j=1}^{n} a(i,j)[a(i,j-1) + a(i,j+1) + a(i-1,j) + a(i+1,j)]$

a(0, j) = a(i, 0) = a(n+1, j) = a(i, n+1) = 0

A globális összefüggőségi mátrix ortogonálisan szimmetrikus, ezt kihasználva szépen le lehet vezetni, hogy:

$\boldsymbol{AM = \sum_{j=1}^{n}[bond(j,j-1) + bond(j,j+1)]}$

$$\left[\begin{array}{cc|cc} 90 & 80 & 1 & 2 \\ 80 & 70 & 3 & 4 \\ \hline 1 & 3 & 60 & 50 \\ 2 & 4 & 50 & 40 \end{array}\right]$$

„Szuboptimális” oszlopsorrendet határozunk meg, ami lehet, hogy nem a legjobb, de legalább emberi időben számítható.

Algoritmus (oszloppermutáció)

1. **Rögzítjük a mátrix 1. oszlopát**

2. **i oszlop beszúrása után a maradékból az i+1. oszlop kiválasztása, bepróbáljuk a lehetséges i+1 db pozícióba úgy, hogy beszúrással a legnagyobb javítást érjük el AM metrika mértékében**
3. **A 2. lépés ismétlése, míg kész nem vagyunk.**

Javítás

Tegyük fel, hogy a jelenlegi állapot:

$$\underbrace{A_1, \dots, A_{i-1}}_{AM'}, A_i, A_j, \underbrace{A_{j+1}, \dots, A_n}_{AM''} \quad \text{a globális összefüggőség mértéke}$$

A_k-t szeretnénk beszúrni A_i és A_j közé.

A_k beszúrása előtt $AM\ régi = AM' + AM'' + \overbrace{bond(i-1,i) + bond(i,j)}^{A_i \text{ miatt}} +$

$$\underbrace{bond(i,j) + bond(j,j+1)}_{A_j \text{ miatt}}$$

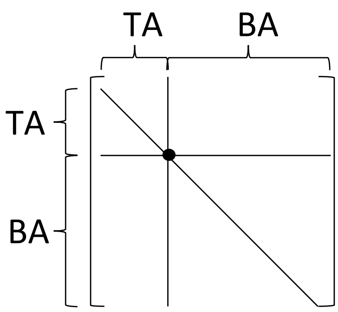
$$AM\ új = AM' + AM'' + \underbrace{bond(i-1,i) + bond(i,k)}_{} + \underbrace{bond(i,k) + bond(k,j)}_{} +$$

$$\underbrace{bond(k,j) + bond(j,j+1)}_{}$$

Javítás* = AM új – AM régi = 2*bond(i,k) + 2*bond(k,j) – 2*bond(i,j)

***Javítás* = Javítás*/2 = bond(i,k) + bond(k,j) – bond(i,j)**

Osztópontkeresés



Cél: a főátló mentén találjunk egy pontot az i. és az i+1. sor/oszlop között, amelyre a metrika maximális. (olyan osztópontot találjunk, hogy a két töredéklerációra lekérdezések diszjunkt halmazai hivatkozzanak)

$TA = \{A_1, \dots, A_i\}$

$BA = \{A_{i+1}, \dots, A_n\}$

Cél: egy töredékhez való (lekérdezések általi) hozzáférések száma -> maximális

több töredékhez való (lekérdezések általi) hozzáférések száma -> minimális

Definíciók

Adott Q lekérdezéshalmaz, $Q = \{q_1, q_2, \dots, q_q\}$

$AQ(q_i) = \{A_j \mid use(q_i,A_j) = 1\}$ (azon attribútumok halmaza, amiket használ)

$TQ = \{qi \mid AQ(qi) \subseteq TA\}$ (azon lekérdezések, amelyek attribútumai TA-ban vannak)

$BQ = \{qi \mid AQ(qi) \subseteq BA\}$ (azon lekérdezések, amelyek attribútumai BA-ban vannak)

$OQ = Q \setminus (TQ \cup BQ)$

Költségfüggvények

$\left.\begin{array}{l} CTQ = \sum_{q \in TQ} acc(q) \\ CBQ = \sum_{q \in BQ} acc(q) \end{array}\right\}$ az a jó, ha ezek minél nagyobbak

$\left.CQQ\right.$

$COQ = \sum_{q \in OQ} acc(q)$ } cél, hogy minél kisebb legyen

***Célfüggvény*: max[CTQ*CBQ – COQ²]**

Példa
(Vertikális felosztás)

Adott:

- R(<u>A</u>, B, C, D) séma, r(R) reláció
- Q = {q_1, q_2, q_3, q_4}
    - $q_1 = \sigma_{C>A}$r acc(q_1) = 20
    - $q_2 = \pi_B \sigma_{C>3}$r acc(q_2) = 10
    - $q_3 = \pi_{B,D}$r acc(q_3) = 5
    - $q_4 = \pi_{C,D}$r acc(q_4) = 15

I. Mátrix létrehozása (affinitás mátrix)

$$\begin{array}{cc} & \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{array}{c} A \\ B \\ C \\ D \end{array} & \begin{bmatrix} 20 & 0 & 20 & 0 \\ 0 & 15 & 10 & 5 \\ 20 & 10 & 45 & 15 \\ 0 & 5 & 15 & 20 \end{bmatrix} \end{array}$$

aff(A,B) = 0, mivel egyik lekérdezés sem használja A-t és B-t együtt

aff(A,A) = acc(q_1), mivel csak q_1 használja A-t

aff(B,B) = acc(q_2) + acc(q_3) = 10+5=15, mivel q_2 és q_3 használja B-t

…

II. Mátrix oszlopainak permutálása

Rögzítjük az első sort, A-t. Utána beszúrjuk mellé B-t (mindegy, hogy melyik oldalra, legfeljebb tükörképe lesz egyik megoldás a másiknak).

Ezután majd próbáljuk beszúrni C-t, ami a következő oszlop. Ezt 3 helyre tehetjük meg (a szélső oszlopoknál mindig 0-val számolunk):

$$\begin{array}{cccc} & A & B & \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 15 & 0 \\ 0 & 20 & 10 & 0 \\ 0 & 0 & 5 & 0 \end{array}$$

I II III

I

	C	A
0	20	20
0	10	0
0	45	20
0	15	0

II

A	C	B
20	20	0
0	10	15
20	45	10
0	15	5

III

B	C	
0	20	0
15	10	0
10	45	0
5	15	0

Javítás$_{I}$ = bond(0,C) + bond(C,A) – bond(0,A) = 0 + 1300 + 0 = 1300

Javítás$_{II}$ = bond(A,C) + bond(C,B) – bond(A,B) = 1300 + 675 – 200 = **<u>1775</u>**

Javítás$_{III}$ = bond(B,C) + bond(C,0) – bond(B,0) = 675 + 0 – 0 = 675

A legnagyobb javítás a II-es, 1775-tel, így C-t A és B közé szúrjuk be. Ezután jön a D oszlop.

	A	C	B	
0	20	20	0	0
0	0	10	15	0
0	20	45	10	0
0	0	15	5	0

I II III IV

I

	D	A
0	0	20
0	5	0
0	15	20
0	20	0

II

A	D	C
20	0	20
0	5	10
20	15	45
0	20	15

III

C	D	B
20	0	0
10	5	15
45	15	10
15	20	5

IV

B	D	
0	0	0
15	5	0
10	15	0
5	20	0

Javítás$_{I}$ = bond(0,D) + bond(D,A) – bond(0,A) = 0 + 300 – 0 = 300

Javítás$_{II}$ = bond(A,D) + bond(D,C) – bond(A,C) = 300 + 1025 - 1300 = 25

Javítás$_{III}$ = bond(C,D) + bond(D,B) – bond(C,B) = 1025 + 325 - 675 = **<u>675</u>**

Javítás$_{IV}$ = bond(B,D) + bond(D,0) – bond(B,0) = 325 + 0 - 0 = 325

A legnagyobb javítás a III-as, 675-tel, így D-t C és B közé szúrjuk be.

Így a végleges oszlopsorrend: A, C, D, B. Ennek megfelelően rendezzük a sorokat is.

III. Sorok átrendezése oszlopnak megfelelően

$$
\begin{array}{c c}
 & \begin{array}{cccc} A & C & D & B \end{array} \\
\begin{array}{c} A \\ C \\ D \\ B \end{array} &
\begin{bmatrix}
20 & 20 & 0 & 0 \\
20 & 45 & 15 & 10 \\
0 & 15 & 20 & 5 \\
0 & 10 & 5 & 15
\end{bmatrix}
\end{array}
$$

IV. Felosztási pont meghatározása

A végleges, átrendezett mátrixban 3 ponton lehet az osztópont:

$$
\begin{array}{c c}
 & \begin{array}{cccc} A & C & D & B \end{array} \\
\begin{array}{c} A \\ C \\ D \\ B \end{array} &
\begin{bmatrix}
20 & 20 & 0 & 0 \\
20 & 45 & 15 & 10 \\
0 & 15 & 20 & 5 \\
0 & 10 & 5 & 15
\end{bmatrix}
\end{array}
$$

I.

TA = {A}	TQ = { }	CTQ * CBQ – COQ^2 = -400
BA = {C, D, B}	BQ = {q_2, q_3, q_4}	(0 * 30 – 20^2)
	OQ = {q_1}	

II.

TA = {A, C}	TQ = {q_1}	CTQ * CBQ – COQ^2 = -525
BA = {D, B}	BQ = {q_3 }	(20 * 5 – 25^2)
	OQ = {q_2, q_4}	

III.

TA = {A, C, D}	TQ = {q_1, q_4}	CTQ * CBQ – COQ^2 = **-225**
BA = {B}	BQ = { }	(35 * 0 – 15^2)
	OQ = {q_2, q_3}	

A maximális összeget, vagyis a -225-öt úgy kapjuk meg, ha a harmadik helyre hozunk létre osztópontot.

Így végül a kapott két részséma***: R₁(A, C, D)*** és ***R₂(A, B)***.