

Σχεδιασμός Κατανεμημένων Βάσεων Δεδομένων

Από πάνω προς τα κάτω

- Κυρίως στο σχεδιασμό συστημάτων από την αρχή
- Κυρίως σε ομογενή συστήματα

Από κάτω προς τα πάνω

- Όταν ήδη υπάρχουν βδ σε έναν αριθμό από κόμβους

Κατάτμηση (Fragmentation) Τοποθέτηση (Allocation)

Γιατί να γίνει η κατάτμηση
Μονάδα κατάτμησης
Ορθότητα κατάτμησης

Ολόκληρη σχέση;

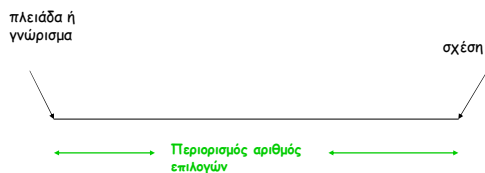
Μειονεκτήματα:

- Εφαρμογές συνήθως υποσύνολα
- Κατανομή, αντίγραφα;
- Αν τμήματα: παράλληλη εκτέλεση μιας ερώτησης
- Ταυτόχρονη εκτέλεση πολλών ερωτήσεων (αύξηση ταυτοχρονισμού, throughput)

Πλεονεκτήματα

- Έλεγχος περιορισμών
- Επιπλέον επεξεργασία και κόστος επικοινωνίας

Οριζόντια Κατάτμηση
Κάθετη Κατάτμηση
Υβριδική Κατάτμηση



Έστω η κατάτμηση της R σε R_1, R_2, \dots, R_n

ΠΛΗΡΟΤΗΤΑ (completeness)

Η κατάτμηση είναι πλήρης αν οποιοδήποτε δεδομένο της R υπάρχει σε κάποιο R_i

ΑΝΑΣΥΝΘΕΣΗ (reconstruction)

Υπάρχει κάποιο σχεσιακό τελεστής που μπορεί να δώσει την R από τις R_i : $R = \bigvee R_i$

ΞΕΝΑ ΣΥΝΟΛΑ (disjointness)

Αν το δεδομένο d_i ανήκει στο R_j δεν θα πρέπει να ανήκει σε κανένα άλλο R_k ($k \neq j$)

Που θα τοποθετηθούν τα τμήματα:

Αντίγραφα:

- Αποδοτικότητα (efficiency) και Αξιοπιστία (reliability)
- Κόστος ενημέρωσης

- **Καθόλου αντίγραφα** (no replication-partitioned): κάθε τμήμα μόνο σε έναν κόμβο
- **Πλήρης αντιγραφή** (full replication): Κάθε τμήμα σε κάθε κόμβο
- **Μερική αντιγραφή** (partial replication)

Γενικό κριτήριο: Αριθμός ερωτήσεων / Αριθμός ενημερώσεων

Πρωτεύουσα Οριζόντια Κατάτμηση: με βάση τις τιμές γνωρίσματος (γνωρισμάτων) που ανήκουν στη σχέση

Παραγόμενη Οριζόντια Κατάτμηση: με βάση τις τιμές γνωρίσματος (γνωρισμάτων) που ανήκουν σε κάποια άλλη σχέση

Έστω η κατάτμηση της R σε R_1, R_2, \dots, R_n

$$R_j = \sigma_{F_j}(R)$$

Κάθε τμήμα R_j περιέχει τις πλειάδες της R που ικανοποιούν το F_j
Τοια είναι η μορφή του F_j :

Απλά κατηγορήματα

Έστω $R(A_1, A_2, \dots, A_n)$

P_j : A_i θ Τιμή

όπου θ : $\neq, =, <, >, \leq, \geq$ και Τιμή $\in \text{Domain}(A_i)$

Για κάθε σχέση R, το σύνολο των απλών κατηγορημάτων P_r

Minterm κατηγορήματα

Έστω $R(A_1, A_2, \dots, A_n)$ και το σύνολο των απλών κατηγορημάτων $P_r = \{p_1, p_2, \dots, p_n\}$

$m_i = \bigwedge p_j^*$, όπου $p_j^* = p_j$ ή $p_j^* = \neg p_j$

M_r το σύνολο των minterm κατηγορημάτων

Οριζόντια Πρωτεύουσα Κατάτμηση

Έστω η κατάτμηση της R σε R_1, R_2, \dots, R_n

$$R_j = \sigma_{F_j}(R)$$

Ποια είναι η μορφή του F_j : **minterm κατηγορήματα**

Τα τμήματα ονομάζονται **minterm τμήματα**

Πόσες είναι οι διαφορετικές οριζόντιες πρωτεύουσες κατατμήσεις;

Οριζόντια Πρωτεύουσα Κατάτμηση

Αλγόριθμος

ΒΗΜΑ 1: Επιλογή κατάλληλου συνόλου απλών κατηγορημάτων

ΒΗΜΑ 2: Συνδυασμοί των απλών κατηγορημάτων σε **minterm** κατηγορήματα

ΒΗΜΑ 3: Απαλοιφή μη συμβατών **minterm** κατηγορημάτων

Κριτήρια για επιλογή ενός «κατάλληλου» συνόλου P_r απλών κατηγορημάτων:

Το P_r πρέπει να είναι **πλήρες** και **ελάχιστο**

Επιθυμητές Ιδιότητες Κατηγορημάτων

Ένα σύνολο P_r από απλά κατηγορήματα είναι **πλήρες** και μόνο αν η πιθανότητα προσπέλασης οποιωνδήποτε δυο πλειάδων οποιουδήποτε **minterm** τμήματος ορισμένου με βάση το P_r από κάθε εφαρμογή είναι ίση \rightarrow

Χωρίζει την R σε τμήματα με ίση πιθανότητα προσπέλασης από τις εφαρμογές

Επιθυμητές Ιδιότητες Κατηγορημάτων

Έστω η σχέση

ΠΡΟΙΟΝ(Αριθμός Προϊόντος, Προέλευση, Τιμή, Είδος)

Και οι ερωτήσεις (εφαρμογές):

- Επεξεργασία προϊόντων με βάση την προέλευση τους
- Βρες τα προϊόντα με τιμή > 100 euro

Έστω $P_r = \{\text{Προέλευση} = \text{«Ελλάδα»}, \text{Προέλευση} = \text{«ΕΕ»}, \text{Προέλευση} = \text{«Εκτός ΕΕ»}\}$

Δεν είναι πλήρες λόγω της δεύτερης ερώτησης

$P_r = \{\text{Προέλευση} = \text{«Ελλάδα»}, \text{Προέλευση} = \text{«ΕΕ»}, \text{Προέλευση} = \text{«Εκτός ΕΕ»}, \text{Τιμή} > 100, \text{Τιμή} \leq 100\}$

Επιθυμητές Ιδιότητες Κατηγορημάτων

Πότε ένα απλό κατηγορήματα *επηρεάζει* την κατάτμηση (δηλαδή, προκαλεί τη διάσπαση ενός τμήματος f σε f_i και f_j):

Θα πρέπει να υπάρχει τουλάχιστον μια εφαρμογή που προσπελαίνει το f_i και το f_j διαφορετικά.

Τότε λέμε ότι το απλό κατηγορήματα είναι σχετικό με την τμηματοποίηση

Έστω το απλό κατηγορήματα p_i και τα $m_i = p_i$ και $m_j = \neg p_i$ που ορίζουν αντίστοιχα τα τμήματα f_i και f_j . Το p_i είναι **σχετικό** αν

$$\frac{\text{acc}(m_i)}{\text{card}(f_i)} \neq \frac{\text{acc}(m_j)}{\text{card}(f_j)}$$

P_r **ελάχιστο** αν έχει μόνο σχετικά κατηγορήματα

Επιθυμητές Ιδιότητες Κατηγορημάτων

Έστω η σχέση

ΠΡΟΙΟΝ(Αριθμός Προϊόντος, Προέλευση, Τιμή, Είδος)

Και οι ερωτήσεις (εφαρμογές):

- Βρες τα προϊόντα με βάση τη προέλευση
- Βρες τα προϊόντα με τιμή > 100 euro

$P_r = \{\text{Προέλευση} = \text{«Ελλάδα»}, \text{Προέλευση} = \text{«ΕΕ»}, \text{Προέλευση} = \text{«Εκτός ΕΕ»}, \text{Τιμή} > 100, \text{Τιμή} \leq 100\}$ πλήρες και ελάχιστο

$P_r = \{\text{Προέλευση} = \text{«Ελλάδα»}, \text{Προέλευση} = \text{«ΕΕ»}, \text{Προέλευση} = \text{«Εκτός ΕΕ»}, \text{Τιμή} > 100, \text{Τιμή} \leq 100, \text{Είδος} = \text{«Επιπλο»}\}$ δεν είναι ελάχιστο

Αλγόριθμος

ΒΗΜΑ 1: Επιλογή ελάχιστου και σχετικού συνόλου απλών κατηγορημάτων

ΒΗΜΑ 2: Συνδυασμοί των απλών κατηγορημάτων σε *minterm* κατηγορήματα

ΒΗΜΑ 3: Απαλοιφή μη συμβατών *minterm* κατηγορημάτων

ΒΗΜΑ 1: Επιλογή ελάχιστου και σχετικού συνόλου απλών κατηγορημάτων

Κανόνας 1

Κάθε σχέση (ή τμήμα) χωρίζεται σε τουλάχιστον δύο τμήματα τα οποία προσπελαύνονται διαφορετικά από τουλάχιστον μια εφαρμογή

ΒΗΜΑ 1: Επιλογή P'_r ελάχιστου και σχετικού συνόλου απλών κατηγορημάτων

Αρχικοποίηση

Βρες ένα $p_i \in P_r$ που να χωρίζει την R με βάση τον Κανόνα 1

$$P'_r = p_i; P_r = P_r - p_i; F = f_i$$

Επαναληπτικά προσθέτουμε κατηγορήματα στο P'_r μέχρι το P'_r να είναι πλήρες

Βρες ένα $p_j \in P_r$ που να χωρίζει κάποιο f_k ορισμένο με *minterm* κατηγορήματα του P'_r με βάση τον Κανόνα 1

$$P'_r = P'_r \cup p_j; P_r = P_r - p_j; F = F \cup f_j$$

Αν $\exists p_k \in P_r'$ που δεν είναι σχετικό τότε

$$P'_r = P'_r - p_k$$

$$F = F - f_k$$

Αλγόριθμος

ΒΗΜΑ 1: Επιλογή ελάχιστου και σχετικού συνόλου απλών κατηγορημάτων

ΒΗΜΑ 2: Συνδυασμοί των απλών κατηγορημάτων σε *minterm* κατηγορήματα

ΒΗΜΑ 3: Απαλοιφή μη συμβατών *minterm* κατηγορημάτων

Από το Βήμα 1 έχουμε το πλήρες και ελάχιστο σύνολο απλών κατηγορημάτων P_r

- Υπολογίζουμε το σύνολο M των *minterm* κατηγορημάτων
- Καθορίζουμε το σύνολο I των συνθηκών μεταξύ των απλών κατηγορημάτων
- Απαλείφουμε όρους από το M που αντιβαίνουν τις συνθήκες

Αλγόριθμος

ΒΗΜΑ 1: Επιλογή ελάχιστου και σχετικού συνόλου απλών κατηγορημάτων

ΒΗΜΑ 2: Συνδυασμοί των απλών κατηγορημάτων σε *minterm* κατηγορήματα

ΒΗΜΑ 3: Απαλοιφή μη συμβατών *minterm* κατηγορημάτων

Παράδειγμα ...

Παράδειγμα

Έστω η τεχνική εταιρεία ΑΒΓ Α.Ε., η οποία έχει αναλάβει 2 μεγάλα έργα για την Ολυμπιάδα του 2004.

Τα έργα γίνονται στην Καστοριά και στη Ρόδο.

Τα κεντρικά γραφεία της εταιρείας βρίσκονται στην Αθήνα.

Η εταιρεία είναι χωρισμένη σε δύο μεγάλες ομάδες: αυτή που ασχολείται με τα έργα στο Βορρά και αυτή που ασχολείται με τα έργα στο Νότο. Τα έργα που γίνονται στην Αθήνα είναι ελάχιστα και εντάσσονται και αυτά στο βόρειο ή νότιο τμήμα, κατά περίπτωση.

Κάθε έργο αποτελείται από πολλά υποέργα, τα οποία και αποτελούν τη βασική οντότητα πληροφορίας. Έχουμε τρία πληροφοριακά κέντρα, ένα σε κάθε πόλη, που θέλουμε να μοιράζονται την ίδια καταγεγραμμένη βάση δεδομένων. Οι καθολικές σχέσεις που έχουμε για την εταιρεία είναι οι εξής:

```
EMP(EMPNUM, NAME, SALARY, TAX, MGRNUM, DEPTNUM)
DEPT(DEPTNUM, NAME, AREA, MGRNUM)
DEPT_TASK(DEPTNUM, TNUM, ROLE, BUDGET)
TASK(TNUM, NAME, PROGRESS, CITY)
```

Οριζόντια Πρωτεύουσα Κατάτμηση

Η σχέση **TASK** (TNUM, NAME, PROGRESS, CITY)

3 κόμβοι "KAST", "RHO", "ATH", οι 2 πρώτες πόλεις σε συχνότητα στις πλειάδες της καθολικής σχέσης.

Η πιο συχνή εφαρμογή θέλει να βλέπει το όνομα και την πρόοδο κάθε υποέργου:

```
SELECT NAME, PROGRESS
FROM TASK
WHERE TNUM = $1;
```

Η ερώτηση τίθεται σε κάθε πόλη. Στην Καστοριά, η πιθανότητα η ερώτηση να αφορά τα τοπικά έργα είναι 80%. Το αντίστοιχο ισχύει και για τη Ρόδο. Στην Αθήνα η πιθανότητα να ρωτάει κάποιος για έργα στην Καστοριά και στη Ρόδο είναι η ίδια.
ΒΗΜΑ 1: Τα εξής απλά κατηγορήματα:

$p1: CITY = 'KAST', p2: CITY = 'RHO'$

Το σύνολο $\{p1, p2\}$ είναι πλήρες και ελάχιστο.

Οριζόντια Πρωτεύουσα Κατάτμηση

ΒΗΜΑ 2: Τα σύνθετα κατηγορήματα που μπορούμε να παράγουμε είναι:

$q1: CITY = 'KAST' \text{ AND } CITY = 'RHO'$
 $q2: CITY = 'KAST' \text{ AND NOT } (CITY = 'RHO')$
 $q3: NOT (CITY = 'KAST') \text{ AND } CITY = 'RHO'$
 $q4: NOT (CITY = 'KAST') \text{ AND NOT } (CITY = 'RHO')$

Τα σύνθετα κατηγορήματα $q1$ και $q4$ εμπεριέχουν **αντιφάσεις**.

ΒΗΜΑ 3: Λογικές συνεπαγωγές

$CITY = 'KAST' \Rightarrow NOT(CITY = 'RHO')$ και $CITY = 'RHO' \Rightarrow NOT(CITY = 'KAST')$
 συνάγουμε ότι τελικά από τα $q2, q3$ προκύπτουν τα $p1, p2$ και άρα το σύνολο $\{p1, p2\}$ είναι πλήρες και ελάχιστο.

Η σχέση **TASK** διαχωρίζεται στα τμήματα:

$TASK1 = \sigma_{CITY='KAST'}(TASK)$
 $TASK2 = \sigma_{CITY='RHO'}(TASK)$

*Το ενδιαφέρον είναι ότι ενώ στην εφαρμογή δεν υπάρχει πουθενά το πεδίο **CITY**, η κατάτμηση της σχέσης γίνεται με βάση αυτό, καθώς συμμετέχει έμμεσα στη σχέση των πεδίων **TNUM** και **CITY**.*

Οριζόντια Πρωτεύουσα Κατάτμηση

Η σχέση **DEPT** (DEPTNUM, NAME, AREA, MGRNUM)

Η σχέση **DEPT** χρησιμοποιείται από δύο βασικές εφαρμογές που ψάχνουν για:

- πληροφορίες για τα υποέργα που γίνονται στην Καστοριά και στη Ρόδο. Σε κάθε μία από αυτές τις δύο πόλεις γίνονται και οι ερωτήσεις για τα τμήματα που είναι στο Βορρά (ή στο Νότο αντίστοιχα).
 - διαχειριστικές πληροφορίες για τα τμήματα. Για κάθε τμήμα, οι ερωτήσεις γίνονται τοπικά.
- Έστω ότι τα τμήματα της Καστοριάς έχουν DEPTNUM από 1 ως 10, της Αθήνας από 11 ως 20 και της Ρόδου από 21 ως 30.

ΒΗΜΑ 1:

$q1: AREA = NORTH$
 $q2: AREA = SOUTH$
 $q3: DEPTNUM \leq 10$
 $q4: 11 < DEPTNUM < 20$
 $q5: 21 < DEPTNUM < 30$

Οριζόντια Πρωτεύουσα Κατάτμηση

ΒΗΜΑ 2&3: Τα σύνθετα κατηγορήματα που μπορούμε να παράγουμε είναι:

$q1: DEPTNUM < 10$
 $q2: 10 < DEPTNUM \leq 20 \text{ AND } AREA = 'NORTH'$
 $q3: 10 < DEPTNUM \leq 20 \text{ AND } AREA = 'SOUTH'$
 $q4: DEPTNUM > 20$

Η σχέση **DEPT** διαχωρίζεται στα τμήματα:

$DEPT1 = \sigma_{DEPTNUM \leq 10}(DEPT)$
 $DEPT2 = \sigma_{10 < DEPTNUM \leq 20 \text{ AND } AREA = 'NORTH'}(DEPT)$
 $DEPT3 = \sigma_{10 < DEPTNUM \leq 20 \text{ AND } AREA = 'SOUTH'}(DEPT)$
 $DEPT4 = \sigma_{DEPTNUM > 20}(DEPT)$

Ορθότητα Κατάτμησης

Έστω η κατάτμηση της R σε R_1, R_2, \dots, R_n

ΠΛΗΡΟΤΗΤΑ (completeness)

ΑΝΑΣΥΝΘΕΣΗ (reconstruction)

Υπάρχει κάποιος σχεσιακό τελεστής που μπορεί να δώσει την R από τις $R_i: R = \cup R_i$

ΞΕΝΑ ΣΥΝΟΛΑ (disjointness)

Αν το δεδομένο d_i ανήκει στο R_j δεν θα πρέπει να ανήκει σε κανένα άλλο $R_k (k \neq j)$

Πρέπει τα κατηγορήματα να είναι αμοιβαία αποκλειστικά (mutually exclusive)

Οριζόντια Παραγόμενη Κατάτμηση

Θα δούμε μόνο τα ξένα κλειδιά

Παράδειγμα

$ΕΡΓΑΖΟΜΕΝΟΣ(ΑΡ_ΕΡΓΟΥ, ΑΡ_ΤΜΗΜΑΤΟΣ, ΜΙΣΘΟΣ)$
 $ΤΜΗΜΑ(ΑΡ_ΤΜΗΜΑΤΟΣ, ΠΟΛΗ)$

$ΤΜΗΜΑ1 = \sigma_{ΠΟΛΗ = ΠΑΤΡΑ}(ΤΜΗΜΑ)$

$ΕΡΓΑΖΟΜΕΝΟΣ1 = ΕΡΓΑΖΟΜΕΝΟΣ \text{ semijoin } ΤΜΗΜΑ1$

Έστω η κατάτμηση της R σε R_1, R_2, \dots, R_n

ΠΛΗΡΟΤΗΤΑ (completeness)

Ισχύει από τον ορισμό του ξένου κλειδιού

ΑΝΑΣΥΝΘΕΣΗ (reconstruction)

Υπάρχει κάποιο σχεσιακό τελεστής που μπορεί να δώσει την R από τις R_i : $R = \cup R_i$

ΞΕΝΑ ΣΥΝΟΛΑ (disjointness)

Αν το δεδομένο d_i ανήκει στο R_j δεν θα πρέπει να ανήκει σε κανένα άλλο R_k ($k \neq j$)

Πρέπει τα κατηγορήματα να είναι αμοιβαία αποκλειστικά (mutually exclusive)

Παρατήρηση

Τα γνωρίσματα του πρωτεύοντος κλειδιού και στα δυο τμήματα

Χρήσιμη όταν ομάδες από γνωρίσματα προσπελούνται μαζί από τις εφαρμογές

Ποιο δύσκολη από την οριζόντια γιατί υπάρχουν πιο πολύ εναλλακτικοί τρόποι

Δυο προσεγγίσεις:

- ομαδοποίηση των γνωρισμάτων σε τμήματα
- διάσπαση της σχέσης σε (ομάδες) γνωρίσματα

Έστω η σχέση

$R(A_1, A_2, \dots, A_n)$ και

τα σύνολα

$S = \{S_1, S_2, \dots, S_m\}$ από κόμβους

$Q = \{q_1, q_2, \dots, q_q\}$ από ερωτήσεις

Τι είδους πληροφορία χρειαζόμαστε;

Συνάφεια Γνωρισμάτων (attribute affinity)

- Μέτρηση του πόσο σχετίζονται μεταξύ τους τα γνωρίσματα
- Μπορεί να υπολογιστεί με βάση την κοινή τους χρήση τους

Χρησιμοποίηση Γνωρισμάτων

Δοθέντος ενός συνόλου από ερωτήσεις $Q = \{q_1, q_2, \dots, q_q\}$ πάνω σε μία σχέση $R(A_1, A_2, \dots, A_n)$

$use(q_i, A_j) = 1$, αν το γνώρισμα A_j αναφέρεται στην ερώτηση q_i
αλλιώς

$use(q_i, A_j) = 1$, αν το γνώρισμα A_j αναφέρεται στην ερώτηση q_i
αλλιώς

Παράδειγμα ...

Συνάφεια Γνωρισμάτων

Η **συνάφεια (aff)** μεταξύ δυο γνωρισμάτων A_i και A_j ορίζεται ως εξής

$$\text{aff}(A_i, A_j) = \sum_{\text{όλες οι ερωτήσεις που προσπελαίνουν και το } A_i \text{ και το } A_j} (\text{query_access})$$

$$\text{query_access} = \sum_{\text{όλοι οι κόμβοι}} \text{access frequency of a query} * \# \text{access/execution}$$

όλες οι ερωτήσεις qk που προσπελαίνουν και το A_i και το A_j : $\text{use}(q_k, A_i) = 1$ and $\text{use}(q_k, A_j) = 1$

#access/execution $\text{ref}(q_k)$: #προσπελάσεις του γνωρισματος A_i ανά εκτέλεση της ερώτησης q_k

access frequency of a query $\text{acc}(q_k)$: συχνότητα εκτέλεσης της ερώτησης q_k στη μονάδα του χρόνου στον κόμβο S_i

Συνάφεια Γνωρισμάτων

Έστω ο πίνακας
χρησιμοποίησης

	A1	A2	A3	A4
q1	1	0	1	0
q2	0	1	1	0
q3	0	1	0	1
q4	0	0	1	1

Έστω ότι η κάθε ερώτηση προσπελαίνει κάθε γνώρισμα μόνο μια φορά σε κάθε εκτέλεση, δηλαδή #access/execution = 1

Έστω 3 κόμβοι (S_i) και η συχνότητα εκτέλεσης για κάθε ερώτηση ως εξής (π.χ., $\text{acc}_1(q_1) = 20$, $\text{acc}_2(q_1) = 10$ κλπ)

	S1	S2	S3	Τότε
q1	15	20	10	$\text{aff}(A1, A3) = 15*1 + 20*1 + 10*1 = 45$
q2	5	0	0	
q3	25	25	25	
q4	3	0	0	

Συνάφεια Γνωρισμάτων

Πίνακας
Συνάφειας
Γνωρισμάτων

	A1	A2	A3	A4	
Attribute Affinity Matrix (AA)	A1	45	0	45	0
	A2	0	80	5	75
	A3	45	5	53	3
	A4	0	75	3	78

- συμμετρικός

Αλγόριθμος Ομαδοποίησης

Στόχος: με βάση τον AA πίνακα να αναδιοργανώσει τη σειρά των γνωρισμάτων έτσι ώστε να δημιουργηθούν **ομάδες** (clusters) γνωρισμάτων τέτοιες ώστε τα γνωρίσματα σε κάθε ομάδα να έχουν μεγάλη συνάφεια μεταξύ τους

Αλγόριθμος Ομαδοποίησης

Bond Energy Algorithm (BEA): γενικός αλγόριθμος ομαδοποίησης

BEA βρίσκει μια διάταξη των γνωρισμάτων που μεγιστοποιεί την ολική συνάφεια των γνωρισμάτων με τους γείτονές τους

Η **ολική συνάφεια AM** ορίζεται ως

$$AM = \sum_i \sum_j (\text{affinity of } A_i \text{ and } A_j \text{ with their neighbors}) =$$

$$\sum_i \sum_j \text{aff}(A_i, A_j) [\text{aff}(A_i, A_{j-1}) + \text{aff}(A_i, A_{j+1}) + \text{aff}(A_{i-1}, A_j) + \text{aff}(A_{i+1}, A_j)]$$

Αλγόριθμος Ομαδοποίησης

Σημείωση

$$\text{aff}(A_0, A_j) = \text{aff}(A_i, A_0) = \text{aff}(A_{n-1}, A_j) = \text{aff}(A_i, A_{n-1}) = 0$$

Είσοδος: Ο ΑΑ πίνακας

Έξοδος: Ο ομαδοποιημένος πίνακας συνάφειας γνωρισμάτων (CA)

ΒΗΜΑ 1 (Αρχικοποίηση) Τοποθέτησε μια από τις στήλες του ΑΑ στον CA

ΒΗΜΑ 2 (Επανάληψη): Τοποθέτησε μια από τις υπόλοιπες n - i στήλες στις υπόλοιπες i + 1 θέσεις του CA πίνακα. Για κάθε στήλη επέλεξε την τοποθέτηση που δίνει τη μεγαλύτερη συνεισφορά στην ολική συνάφεια

ΒΗΜΑ 3 (Διάταξη γραμμών) Αναδιάταξη των γραμμών με βάση τη διάταξη των στηλών

ΒΗΜΑ 2 (Επανάληψη): Τοποθέτησε μια από τις υπόλοιπες n - i στήλες στις υπόλοιπες i + 1 θέσεις του CA πίνακα. Για κάθε στήλη επέλεξε την τοποθέτηση που δίνει τη μεγαλύτερη συνεισφορά στην ολική συνάφεια

Ορίζουμε τη συνεισφορά τοποθέτησης A_x ως εξής:

$$\text{cont}(A_i, A_k, A_j) = 2\text{bond}(A_i, A_k) + 2\text{bond}(A_k, A_j) - 2\text{bond}(A_i, A_j)$$

όπου

$$\text{bond}(A_x, A_y) = \sum_{z=1}^n \text{aff}(A_z, A_x)\text{aff}(A_z, A_y)$$

$$\text{cont}(A_i, A_k, A_j) = 2\text{bond}(A_i, A_k) + 2\text{bond}(A_k, A_j) - 2\text{bond}(A_i, A_j)$$

$$\text{bond}(A_x, A_y) = \sum_{z=1}^n \text{aff}(A_z, A_x)\text{aff}(A_z, A_y)$$

	A1	A2	A3	A4		(1)	(2)	(3)
AA	A1	45	0	45	0	A1	45	0
	A2	0	80	5	75	A2	0	80
	A3	45	5	53	3	A3	45	5
	A4	0	75	3	78	A4	0	75

Τοποθέτηση της A3: Πιθανές θέσεις (1), (2), (3)

Θέση 1, Διάταξη (0-3-1):

$$\text{cont}(A_0, A_3, A_1) = 2\text{bond}(A_0, A_3) + 2\text{bond}(A_3, A_1) - 2\text{bond}(A_0, A_1) = 2 * 0 + 2 * 4410 - 2 * 0 = 8820$$

Θέση 2, Διάταξη (1-3-2):

$$\text{cont}(A_1, A_3, A_2) = 2\text{bond}(A_1, A_3) + 2\text{bond}(A_3, A_2) - 2\text{bond}(A_1, A_2) = 2 * 4410 + 2 * 890 - 2 * 225 = 10150$$

Θέση 3, Διάταξη (2-3-4):

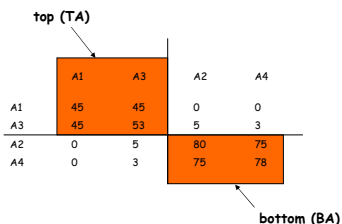
$$\text{cont}(A_2, A_3, A_4) = 1780$$

CA

	A1	A3	A2
A1	45	45	0
A2	0	5	80
A3	45	53	5
A4	0	3	75

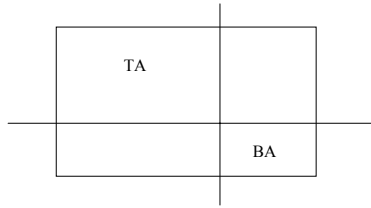
Ο τελικός (μετά την τοποθέτηση της A4 και την αναδιοργάνωση των γραμμών) CA

	A1	A3	A2	A4
A1	45	45	0	0
A3	45	53	5	3
A2	0	5	80	75
A4	0	3	75	78

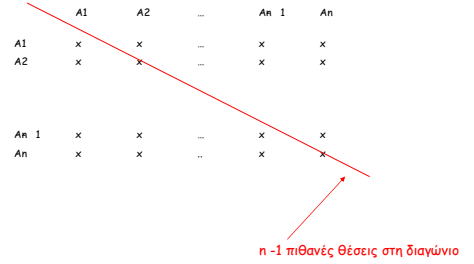


Διάσπαση του CA

Γενικά: πως μπορούμε να χωρίσουμε ένα σύνολο από ομαδοποιημένα γνωρίσματα σε δυο (ή περισσότερα) σύνολα ώστε να μην υπάρχουν καθόλου (ή να υπάρχουν πολύ λίγες εφαρμογές) που προσπελαίνουν και τα δύο)



Διάσπαση του CA



Διάσπαση του CA

TQ: εφαρμογές που προσπελαίνουν μόνο το TA

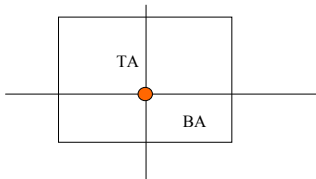
BQ: εφαρμογές που προσπελαίνουν μόνο το BA

OQ: εφαρμογές που προσπελαίνουν και το BA και το TA

CTQ = ολικός αριθμός προσπελάσεων σε γνωρίσματα από εφαρμογές που προσπελαίνουν μόνο το TA

Αντίστοιχα CBQ και COQ $\pi \cdot X \dots CQ = \sum_{q_i \in Q} \sum_{\text{για όλα τα } s_j} \text{ref}_j(q_i) \text{acc}_j(q_i)$

Εύρεση σημείου στη διαγώνιο που να μεγιστοποιεί το $CTQ * CBQ - COQ^2$



Διάσπαση του CA

Δύο προβλήματα

▪ Ομάδα στη μέση

- Μετακίνηση (shift) μια γραμμή πάνω και μια στήλη αριστερά
- Δοκίμασε όλα τις πιθανές μετακινήσεις

▪ Παραπάνω από δύο clusters:

- Δοκίμασε 1, 2, ... m-1 σημεία διάσπασης
- Κόστος $O(2^m)$

Ορθότητα Κατάτμησης

Έστω η κατάτμηση της R σε R_1, R_2, \dots, R_n

ΠΛΗΡΟΤΗΤΑ (completeness)

Θα πρέπει $A = \cup A_i$

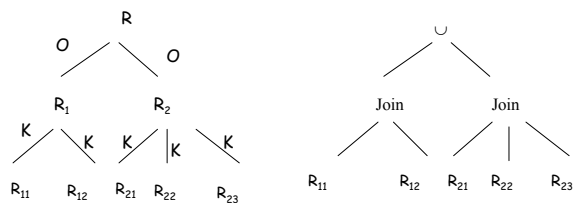
ΑΝΑΣΥΝΘΕΣΗ (reconstruction)

ΞΕΝΑ ΣΥΝΟΛΑ (disjointness)

TID

Επανάληψη κλειδιού

Υβριδική Κατάτμηση



Ορισμός προβλήματος

Δοθέντος ενός συνόλου:

$F = \{F_1, F_2, \dots, F_n\}$ από τμήματα

$S = \{S_1, S_2, \dots, S_m\}$ από κόμβους

$Q = \{q_1, q_2, \dots, q_p\}$ από ερωτήσεις

Ποια είναι η "καλύτερη" κατανομή των τμημάτων του F στους κόμβους του S ;

Ορισμός προβλήματος

Καλύτερη κατανομή:

(1) Ελάχιστο κόστος

Επικοινωνία + Αποθήκευση + Επεξεργασία (αναγνώσεις, εγγραφές)

(2) Απόδοση

Χρόνος απόκρισης και throughput

▪ Περιορισμοί

Ανά κόμβο (π.χ., μέγιστος χώρος αποθήκευσης, επεξεργασία)

Τι πληροφορίες χρειάζονται:

▪ Πληροφορίες σχετικές με τη βάση δεδομένων

- Επιλεξιμότητα των τμημάτων
- Μέγεθος των τμημάτων

▪ Πληροφορίες για την εφαρμογή

- Τοπικότητα προσπελάσεων
- Είδη και αριθμός προσπελάσεων

▪ Πληροφορίες για το δίκτυο

- Bandwidth
- Latency
- Communication overhead

▪ Πληροφορίες για τους κόμβους

- Κόστος μονάδας αποθήκευσης
- Κόστος μονάδας επεξεργασίας

Τι πληροφορίες χρειάζονται:

▪ Πληροφορίες σχετικές με τη βάση δεδομένων

- Επιλεξιμότητα των τμημάτων
- Μέγεθος των τμημάτων

▪ Πληροφορίες για την εφαρμογή

- Τοπικότητα προσπελάσεων
- Είδη και αριθμός προσπελάσεων
- αριθμός προσπελάσεων ανάγνωσης σε ένα τμήμα
- αριθμός προσπελάσεων τροποποίησης σε ένα τμήμα
- πίνακας ποιες ερωτήσεις τροποποιούν/διαβάζουν ποια τμήματα
- κόμβος υποβολής της ερώτησης

Τι πληροφορίες χρειάζονται:

▪ Πληροφορίες για το δίκτυο

- Bandwidth
- Latency
- Communication overhead

▪ Πληροφορίες για τους κόμβους

- Κόστος μονάδας αποθήκευσης
- Κόστος μονάδας επεξεργασίας

Γενική Μορφή

Min(Ολικό Κόστος)

Δοθέντων των περιορισμών

Χρόνου απόκρισης

Αποθήκευσης

Επεξεργασίας

Μεταβλητή Απόφασης

$x_{ij} = 1$ αν το τμήμα F_i αποθηκεύεται στον κόμβο S_j
 0 αλλιώς

Ολικό Κόστος

$\sum_{\text{Όλα τς ερωτήσεις}} \text{Κόστος επεξεργασίας} +$
 $\sum_{\text{Όλα οι κόμβοι}} \sum_{\text{Όλα τα τμήματα}} \text{Κόστος αποθήκευσης τμήματος σε κάποιο κόμβο}$
Κόστος αποθήκευσης (τμήματος F_j στον κόμβο S_k)
 (κόστος μονάδας αποθήκευσης στο S_k) * (μέγεθος F_j) * x_{jk}
Κόστος Επεξεργασίας Ερωτήσεων (για μια ερώτηση)
 κόστος επεξεργασίας + κόστος μετάδοσης (επικοινωνίας)

Κόστος Επεξεργασίας Ερωτήσεων (για μια ερώτηση)
 κόστος επεξεργασίας + κόστος επικοινωνίας

Κόστος Επεξεργασίας =
Κόστος Προσπέλασης +
 Κόστος Περιορισμών Ορθότητας +
 Κόστος Ελέγχου ταυτοχρονισμού
Κόστος Προσπέλασης

$\sum_{\text{Όλα οι κόμβοι}} \sum_{\text{Όλα τα τμήματα}} (\# \text{αναγνώσεων} + \# \text{ενημερώσεων}) * x_{ij} * \text{χρόνος τοπικής επεξεργασίας στον κόμβο}$

Το κόστος για τον έλεγχο των περιορισμών ορθότητας και της ταυτόχρονης εκτέλεσης μπορούν να υπολογιστούν με τον ίδιο τρόπο

Κόστος Επεξεργασίας Ερωτήσεων (για μια ερώτηση)
 κόστος επεξεργασίας + κόστος μετάδοσης

Κόστος Μετάδοσης =
Κόστος Επεξεργασίας Ενημερώσεων +
 Κόστος Επεξεργασίας Ερωτήσεων

Κόστος Ενημερώσεων
 $\sum_{\text{Όλα οι κόμβοι}} \sum_{\text{Όλα τα τμήματα}} \text{κόστος μηνύματος ενημέρωσης} + \sum_{\text{Όλα οι κόμβοι}} \sum_{\text{Όλα τα τμήματα}} \text{κόστος μηνύματος επιβεβαίωσης}$

Κόστος Ερωτήσεων
 $\sum_{\text{Όλα τα τμήματα}} \min_{\text{Όλα οι κόμβοι}} (\text{κόστος εντολής ανάγνωσης}) + \text{κόστος αποστολής του αποτελέσματος}$

Γενική Μορφή

Min(Ολικό Κόστος)
 Δοθέντων των **περιορισμών**
 Χρόνου απόκρισης
 Αποθήκευσης
 Επεξεργασίας

Περιορισμοί

Χρόνου απόκρισης
 Χρόνος εκτέλεσης ερώτησης \leq Μέγιστου Επιτρεπτού Χρόνου Απόκρισης για την Ερώτηση

Αποθήκευσης (για έναν κόμβο)
 $\sum_{\text{Όλα τα τμήματα}} \text{Απαιτηση χώρου αποθήκευσης για ένα τμήμα σε έναν κόμβο} \leq \text{Διαθέσιμος χώρος στον κόμβο}$

Επεξεργασίας (για έναν κόμβο)
 $\sum_{\text{Όλα οι ερωτήσεις}} \text{Φορτίο επεξεργασίας μια ερώτησης σε έναν κόμβο} \leq \text{Δυνατότητα επεξεργασίας στον κόμβο}$

Λύση του Προβλήματος

Γενικά NP complete!

Ευριστικοί!