Αλγοριθμική Επιστήμη Δεδομένων

Frequent Patterns and Association Rules Mining

Επιμέλεια διαφανειών:
Δώρα Σούλιου – Άρης Παγουρτζής

σημμυ – ε.δε.μ^2 – εμε – αλμα
ΕΜΠ 2024

Διάγραμμα παρουσίασης

- Association rules
- Frequent itemset mining
- Αλγόριθμος A-priori
- Αλγόριθμος FP-Growth
- Βελτιώσεις

Κανόνες Συσχέτισης

- Δίνεται μία βάση δεδομένων D με εγγραφές που αποτελούνται από διάφορα αντικείμενα. Για παράδειγμα:
  {ψωμί, γάλα, καφές}
  {ψωμί, ζάχαρη, καφές, τυρί}
- **Κανόνας συσχέτισης (association rule)**: μία συνεπαγωγή της μορφής $A \rightarrow B$ όπου A και B είναι σύνολα από αντικείμενα, π.χ.
  {ψωμί, γάλα} → {ζάχαρη, καφές}
- Στήριγμα (support, ή frequency) συνόλου αντικειμένων (itemset) X: πλήθος εγγραφών της D που περιέχουν το itemset X.

Παράδειγμα

a	b	c	d	e	f	g	h
1	1	0	1	1	0	1	1
1	1	0	0	1	1	1	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
1	1	1	0	0	1	1	0
1	1	0	1	1	0	1	0
0	1	0	1	1	1	1	1
1	0	0	1	0	1	1	1
1	0	0	1	1	0	0	1
1	0	0	1	0	1	1	0

- support({a,h}) = 3
- support({a,d,h}) = 3
- support({a,g,h}) = 2
- support({a,f,g,h}) = 1

Έγκυροι κανόνες

- support του itemset X: #εγγραφών στη βάση δεδομένων D που περιέχουν το X (~ prob(X))
- support του κανόνα A → B: support του AUB
- confidence του κανόνα A → B: ο λόγος πλήθους εγγραφών στη βάση D που περιέχουν το AUB προς αυτές που περιέχουν το A.

  confidence(A → B) = support(AUB) / support(A) = prob[B|A]

- Δεδομένης μιας database D ενδιαφερόμαστε να παράγουμε έγκυρους κανόνες δηλαδή κανόνες που έχουν support and confidence πάνω από κάποια δεδομένα thresholds t, c.

Παράδειγμα

a	b	c	d	e	f	g	h
1	1	0	1	1	0	1	1
1	1	0	0	1	1	1	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
1	1	1	0	0	1	1	0
1	1	0	1	1	0	1	0
0	1	0	1	1	1	1	1
1	0	0	1	0	1	1	1
1	0	0	1	1	0	0	1
1	0	0	1	0	1	1	0

A = {a, h}, B = {g}

- support(A → B) = 2
- confidence(A → B) = 2/3

Συχνά σύνολα αντικειμένων

- Εάν το support της ένωσης AUB είναι πάνω από κάποιο threshold, τότε το support του A είναι σίγουρα πάνω από αυτό το threshold (anti-monotonicity)
- Έτσι η παραγωγή όλων των itemsets με support(X) >= t κάνει απλό τον υπολογισμό του confidence(A → B)
- Τα itemsets αυτά ονομάζονται large ή **frequent** (συχνά)
- Εύρεση έγκυρων κανόνων ανάγεται στην εύρεση συχνών συνόλων αντικειμένων

Εφαρμογές του προβλήματος

- **Market Basket Data**
  - Baskets: καλάθια αγορών super market
  - Items: τα αντικείμενα που αγοράζει κανείς
  - Frequent itemsets: Ποιά αντικείμενα αγοράζονται ταυτόχρονα με κάποια άλλα (εφαρμογή: καλύτερη τοποθέτηση / προσφορές)
- **Ανάλυση κειμένων**
  - Baskets:κείμενα
  - Items: λέξεις
  - Frequent itemsets : σύνολα λέξεων που εμφανίζονται συχνά μαζί δηλώνουν κείμενα με παρόμοιο περιεχόμενο

Εφαρμογές του προβλήματος

- **Ανάλυση Web**
  - Baskets: web pages
  - Items: σελίδες που κάνουν link σε αυτές
  - Frequent itemsets: σελίδες με κοινά εισερχόμενα links ίσως έχουν παρόμοιο θέμα
- **Recommendation Systems**
  - Baskets: ταινίες που έχει δει κάποιος χρήστης
  - Items: όλες οι ταινίες
  - Frequent itemsets: χρήστες με πολλές κοινές ταινίες πιθανόν να έχουν παρόμοια γούστα

Εξόρυξη συχνών υποσυνόλων: ένας απλοϊκός αλγόριθμος

- Παραγωγή όλων των δυνατών υποσυνόλων
- Διάσχιση της βάσης δεδομένων και ενημέρωση των συχνοτήτων των υποσυνόλων που εμφανίζονται
- Πολυπλοκότητα;

Προβλήματα και μέθοδοι επίλυσης

- **Μεγάλο πλήθος αντικειμένων** (πάρα πολλοί συνδυασμοί)
  - Τεχνικές μείωσης του εκθετικού πλήθους υποσυνόλων (2^n)
- **Μεγάλο πλήθος εγγραφών** (χρονοβόρα η διαδικασία προσπέλασής τους)
  - Τεχνικές μείωσης των διασχίσεων (passes) της βάσης δεδομένων

Στρατηγικές frequent itemsets mining

- With candidate generation

  (e.g. A-priori) [Agrawal, Srikant ’94]

- Without candidate generation

  (e.g. FP-growth) [Han, Pei, Yin ’00]

Candidate generation

Παραγωγή frequent itemsets *κατά επίπεδα* ανάλογα με την *πληθικότητα*, και υπολογισμός συχνότητας εμφάνισης.

Έστω π.χ. I={a,b,c,d} σύνολο διακριτών αντικειμένων της D

- Επίπεδο 1: {a},{b},{c},{d}
- Επίπεδο 2: {a,b}, {a,c}, {a,d}, {b,c}, {b,d}, {c,d}
- Επίπεδο 3: {a,b,c}, {a,b,d}, {a,c,d}, {b,c,d}
- Επίπεδο 4: {a,b,c,d}

Αλγόριθμος A-priori [Agrawal, Srikant '94]
η βασική ιδέα

- Monotonicity property: Έστω Σ σύνολο αντικειμένων με n στοιχεία και J=pow(Σ) με 2^n-1 στοιχεία. Μία συνάρτηση f είναι **μονότονη** εάν

$$\forall X, Y \in J : (X \subseteq Y) \Rightarrow f(X) \leq f(Y)$$

- Η συνάρτηση f είναι αντι-μονότονη εάν

$$\forall X, Y \in J : (X \subseteq Y) \Rightarrow f(Y) \leq f(X)$$

- Η συνάρτηση support είναι αντι-μονότονη:

*Ένα (k+1)-itemset μπορεί να είναι frequent μόνο αν **όλα τα υποσύνολά του με k items είναι frequent***

Αλγόριθμος A-priori

- Βήμα 0: Σχηματισμός λίστας L_1 συχνών μονοσυνόλων
- Βήμα k (k=1, 2, …): Σχηματισμός λίστας L_{k+1} συχνών (k+1)-itemsets από λίστα L_k συχνών k-itemsets
  - Για κάθε itemset στην λίστα L_k διάσχιση της υπόλοιπης λίστας ώσπου να βρεθεί itemset που διαφέρει στο τελευταίο item μόνο.
    - Παραγωγή (k+1)-itemset := ένωση των δύο k-itemsets που διαφέρουν στο τελευταίο item.
    - Αναζήτηση στην υπόλοιπη λίστα L_k όλων των άλλων υποσύνολων του νέου itemset.
    - Αν υπάρχουν, το itemset προστίθεται στη λίστα C_{k+1} των υποψηφίων συχνών (k+1)-itemsets και υπολογίζεται η συχνότητά του.
  - L_{k+1} := τα συχνά itemsets της C_{k+1} .

Παράδειγμα εκτέλεσης του A-priori

- Έστω η λίστα με τα frequent 3-itemsets

  {a,b,d}, {a,b,f}, {a,d,f}, {b,c,f}, {b,c,g}, {b,c,k}, {b,d,f}, {b,d,l}, και {c,d,f}

- {a,b,d} , {a,b,f} => {a,b,d,f}
- Υποσύνολα: {a,b,d} {a,b,f} {a,d,f} {b,d,f}
- {a,b,d,f} *είναι* **candidate frequent**

- {a,b,d} , {a,d,f} => {} // διαφέρουν σε 2 items

- {b,c,f} , {b,c,g} => {b,c,f,g}
- Υποσύνολα: {b,c,f}, {b,c,g}, {b,f,g}, {c,f,g}
- {b,c,f,g} *δεν είναι* **candidate frequent**

A-priori: πλεονεκτήματα-μειονεκτήματα

Υπέρ:

- Για κάθε συχνό υποσύνολο ελέγχονται το πολύ n μη-συχνά υποσύνολα
- Πολυπλοκότητα: πολυωνυμική *ως προς την έξοδο* (γιατί όχι και ως προς είσοδο;)

Κατά:

- Αν μεγαλύτερο frequent είναι στο επίπεδο k, απαιτούνται k διασχίσεις της βάσης
- Οι λίστες μπορεί να γίνουν πολύ μεγάλες

Βελτιώσεις του A-priori

- Χρήση hashing
  - Στο 1ο πέρασμα:
    - Υπολογισμός συχνών 1-itemsets
    - και όλων των 2-itemsets
    - Εισαγωγή σε hash table
    - Διαγραφή μη-συχνών «κάδων»
- Συρρίκνωση βάσης
  - Διαγραφή εγγραφών που δεν περιέχουν frequent 2-itemsets (υπερσύνολα των υπολοίπων: *μη-συχνά*)

Παράδειγμα

TID	List of items
001	I1,I2,I5
002	I2,I4
003	I2,I3
004	I1,I2,I4
005	I1,I3
006	I2,I3
007	I1,I3
008	I1,I2,I3,I5
009	I1,I2,I3
010	I3,I5

Παράδειγμα (συν.)

Bucket number h(x,y)	0	1	2	3	4	5	6
Bucket count	3	2	4	2	2	4	4
Bucket content	I1, I4 I3, I5 I3, I5	I1, I5 I1, I5	I2, I3 I2, I3 I2, I3 I2, I3	I2, I4 I2, I4	I2, I5 I2, I5	I1, I2 I1, I2 I1, I2 I1, I2	I1, I3 I1, I3 I1, I3 I1, I3

*h(x,y) = index(x)*10 + index(y)* ***mod*** *7*

Παράδειγμα (συν.)

TID	List of items
001	I1,I2,I5
002	I2,I4
003	I2,I3
004	I1,I2,I4
005	I1,I3
006	I2,I3
007	I1,I3
008	I1,I2,I3,I5
009	I1,I2,I3
010	I3,I5

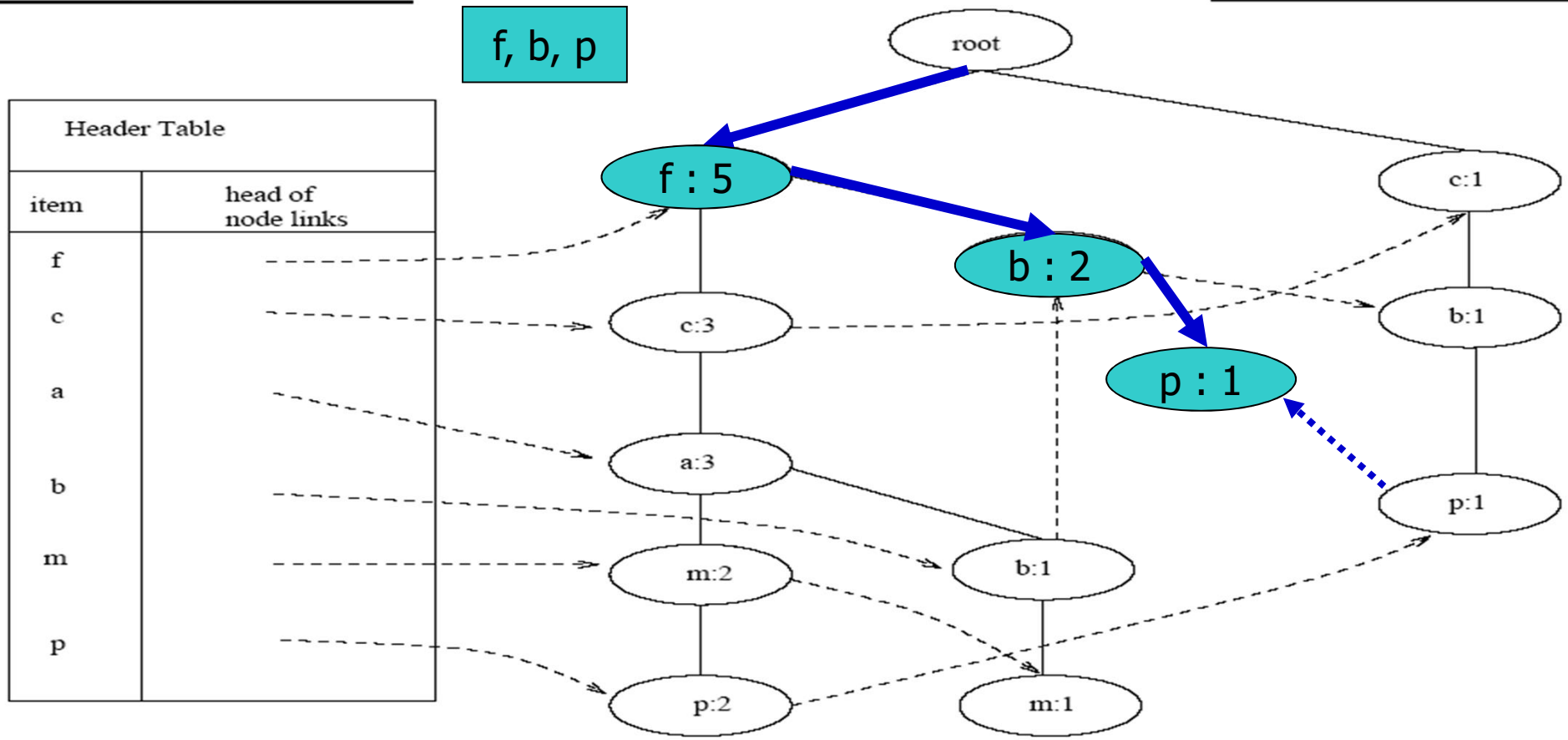
Αλγόριθμος FP-growth [Han,Pei,Yin '00]

- Αποθήκευση της βάσης δεδομένων στη δομή FP-trie με 2 διασχίσεις
  - Υπολογισμός των συχνοτήτων όλων των singletons (1-itemsets)
    - Αφαίρεση non-frequent items και ταξινόμηση εγγραφών κατά φθίνουσα σειρά συχνότητας των items
  - Απεικόνιση εγγραφών σε δέντρο
- Εξόρυξη συχνών συνόλων αντικειμένων χωρίς candidate generation

TID	Items
101	f, a, c, d, g, i, m, p
102	a, b, c, f, l, m, o
103	b, f, h, j, o
104	b, c, k, s, p
105	a, f, c, e, l, p, m, n

Κατώφλι συχνότητας 3

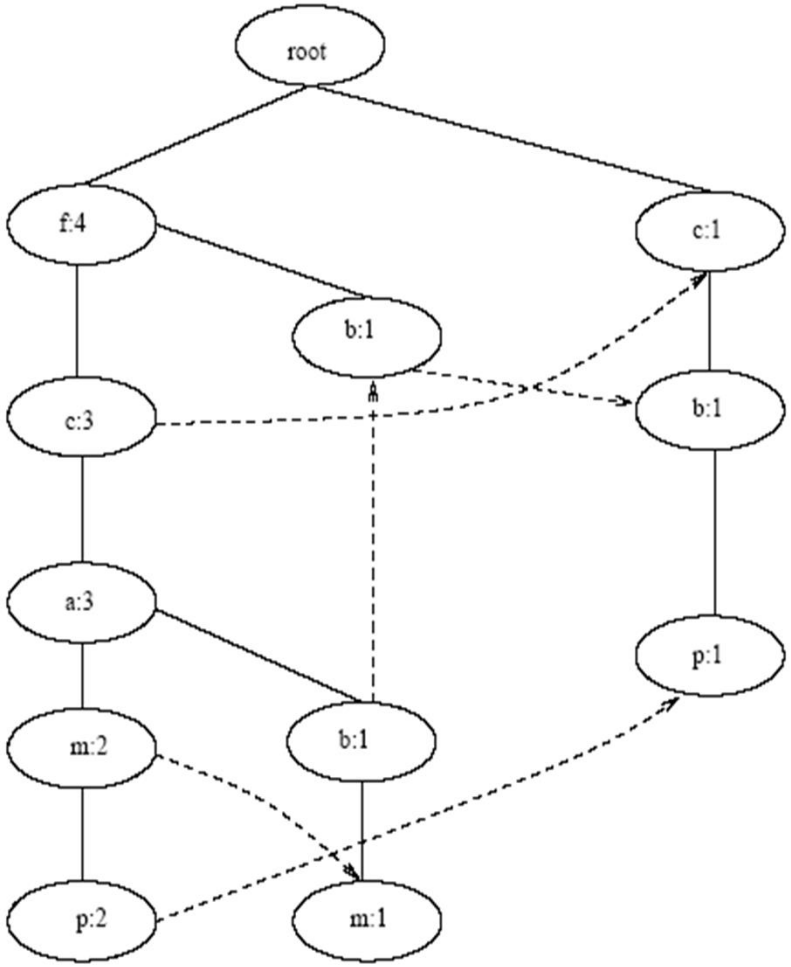
TID	Items
101	f, c, a, m, p
102	f, c, a, b, m
103	f, b
104	c, b, p
105	f, c, a, m, p



Αλγόριθμος FP-growth

- Ξεκινά από το item με την μικρότερη συχνότητα. Φτιάχνει το υποδέντρο με τα μονοπάτια που καταλήγουν σε αυτό.
- Στο νέο δέντρο ενημερώνει συχνότητες και βρίσκει όλα τα frequent 2-itemsets, πάλι ξεκινώντας από το item με τη μικρότερη συχνότητα
- Συνεχίζει με τα δέντρα που προκύπτουν για καθένα από αυτά τα δισύνολα
- Σε κάθε τέτοιο δέντρο προσπαθεί να φτιάξει τρισύνολα κ.λπ.

Παράδειγμα



Threshold 3

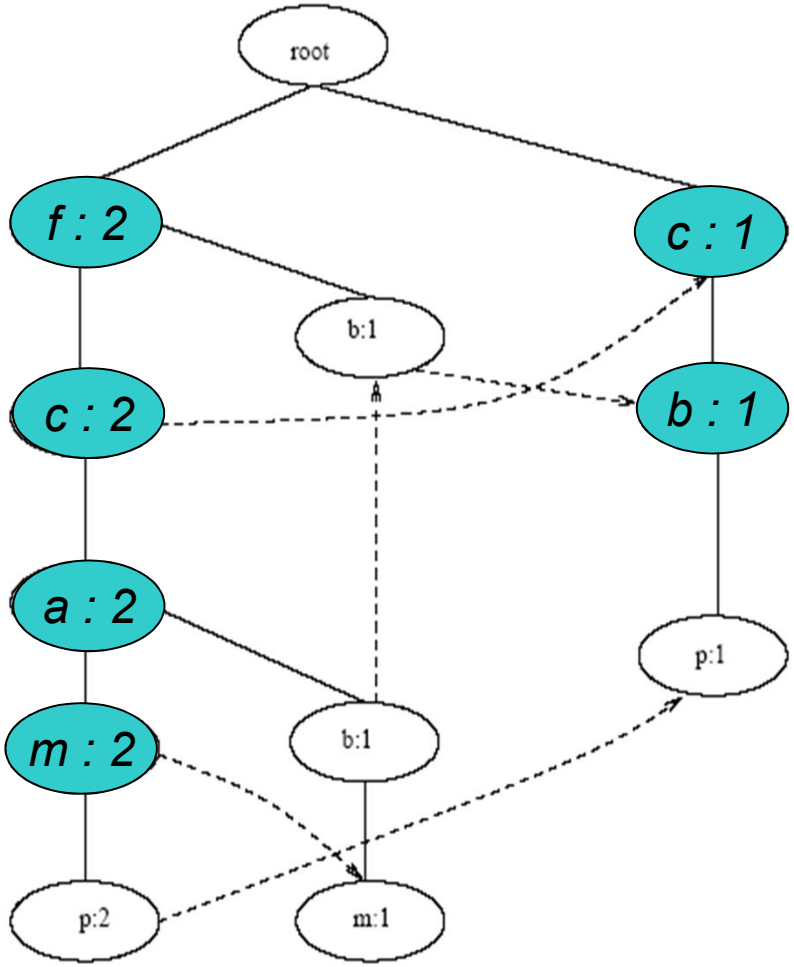
Ο αλγόριθμος ξεκινά με το ‘p’

2 μονοπάτια καταλήγουν σε ‘p’

Ενημέρωση συχνοτήτων

Υπολογισμός συχνοτήτων items

Παράδειγμα



Threshold 3

Ο αλγόριθμος ξεκινά με το ‘p’

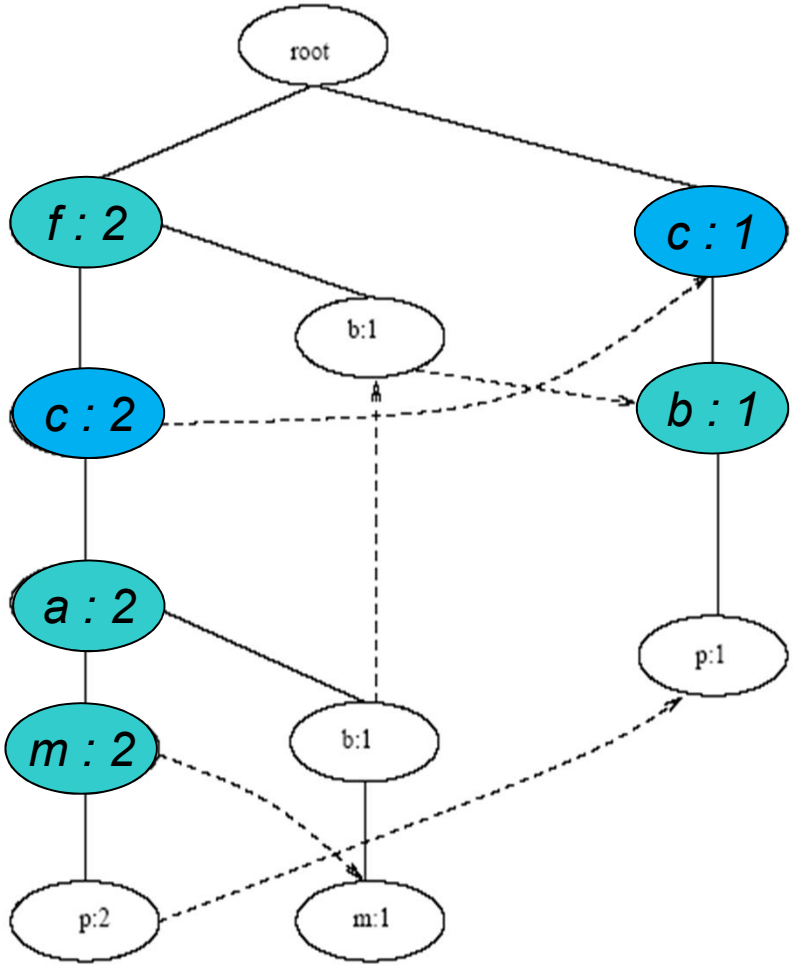
2 μονοπάτια καταλήγουν σε ‘p’

Ενημέρωση συχνοτήτων

Υπολογισμός συχνοτήτων items

Παράδειγμα

Threshold 3



Ο αλγόριθμος ξεκινά με το ‘p’

2 μονοπάτια έχουν ‘p’

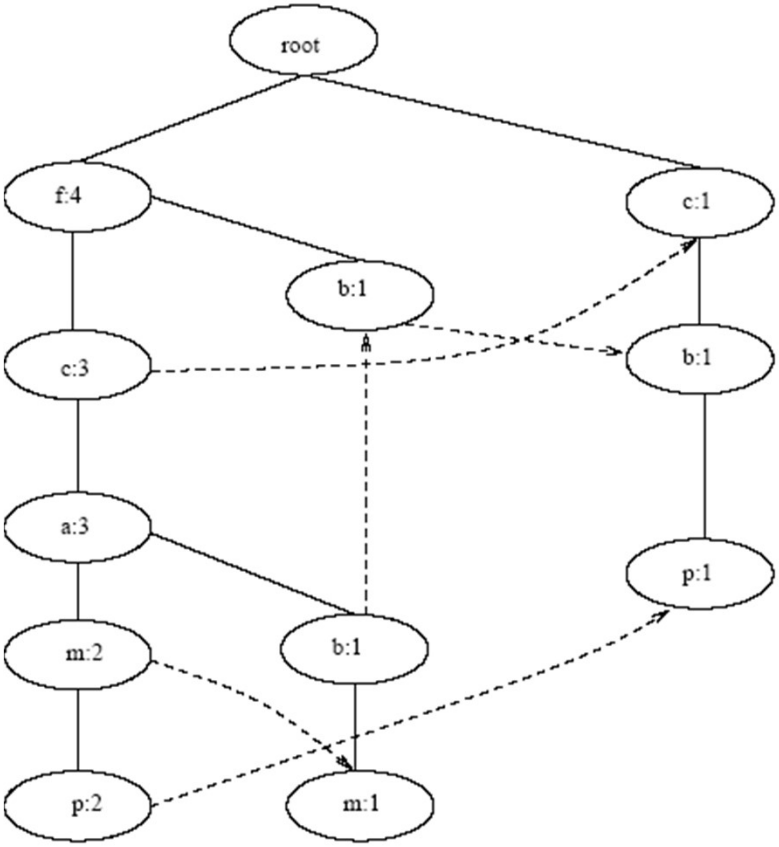
Ενημέρωση συχνοτήτων

Υπολογισμός συχνοτήτων items

Συχνό το ‘c’

Άρα το ‘cp’ είναι συχνό.

Παράδειγμα



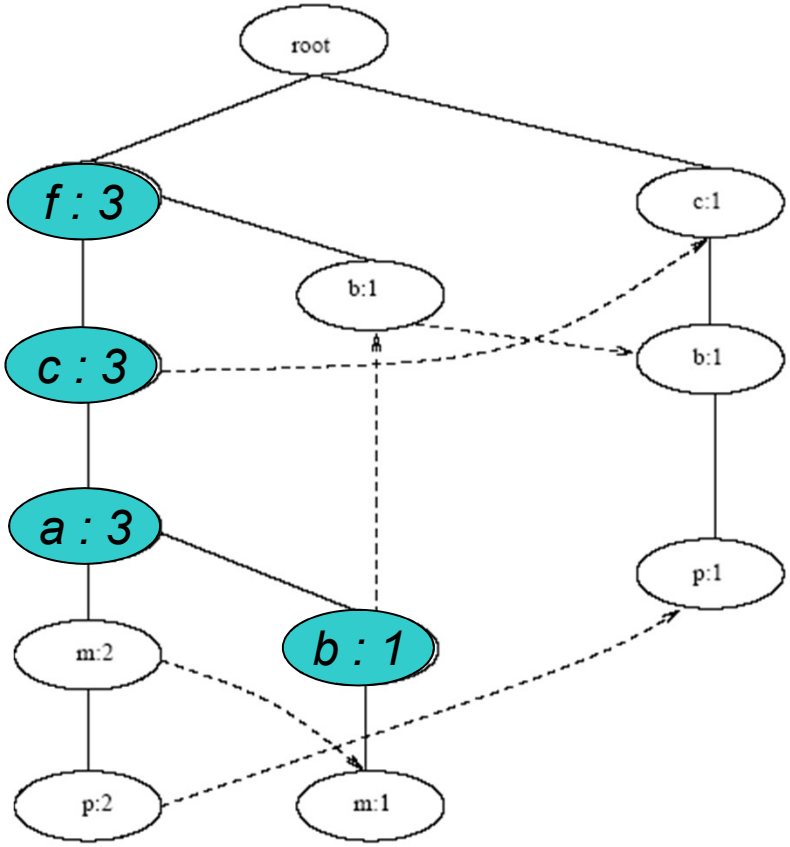
Επόμενο item ‘m’

Μονοπάτια του νέου υποδέντρου f,c,a και f,c,a,b

Ενημέρωση συχνοτήτων

Υπολογισμός συχνοτήτων των items

Παράδειγμα



Επόμενο item ‘m’

Μονοπάτια του νέου υποδέντρου f,c,a και f,c,a,b

Ενημέρωση συχνοτήτων

Υπολογισμός συχνοτήτων των items

Συχνά τα f, c, a

Άρα συχνά 2-itemsets τα (f,m), (c,m), (a,m)

Ξεκινώντας από (a,m) φτιάχνει υποδέντρο με μονοπάτια που καταλήγουν σε αυτό - στο παράδειγμα είναι μόνο τα f, c

Ενδιαφέρουσες κατευθύνσεις

- Παραλληλοποίηση αλγορίθμων

  Πρόβλημα: ανταλλαγή μηνυμάτων για μέτρηση συχνοτήτων ανά επίπεδο

- Δυναμικές βάσεις δεδομένων

  Πρόβλημα: χώρος μνήμης και για μη συχνά itemsets που πιθανόν να εμφανιστούν αργότερα