



Αλγοριθμική Επιστήμη Δεδομένων

Frequent Patterns and Association Rules Mining

Επιμέλεια διαφανειών:
Δώρα Σούλιου – Άρης Παγουρτζής

ΔΠΜΣ Επιστήμη Δεδομένων και Μηχανική Μάθηση (ε.δε.μ²)
ΕΜΠ 2019

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} \rightarrow {ζάχαρη, καφές}
- **Στήριγμα (support, ή frequency)** συνόλου αντικειμένων (itemset) X : πλήθος εγγραφών της D που περιέχουν το itemset X .

Παράδειγμα

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
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

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

Έγκυροι κανόνες

- **support** του itemset X : #εγγραφών στη βάση δεδομένων D που περιέχουν το X ($\sim \text{prob}(X)$)
- **support** του κανόνα $A \rightarrow B$: support του $A \cup B$
- **confidence** του κανόνα $A \rightarrow B$: ο λόγος πλήθους εγγραφών στη βάση D που περιέχουν την ένωση $A \cup B$ ως προς αυτές που περιέχουν το A .
$$\text{confidence}(A \rightarrow B) = \text{support}(A \cup B) / \text{support}(A) = \text{prob}[B|A]$$
- Δεδομένης μιας database D ενδιαφερόμαστε να παράγουμε έγκυρους κανόνες δηλαδή κανόνες που έχουν support and confidence πάνω από κάποια δεδομένα **thresholds** t, c .

Παράδειγμα

<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>
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

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

- $\text{support}(A \rightarrow B) = 2$
- $\text{confidence}(A \rightarrow B) = 2/3$

Συχνά σύνολα αντικειμένων

- Εάν το support της ένωσης $A \cup B$ είναι πάνω από κάποιο threshold, τότε το support του A είναι σίγουρα πάνω από αυτό το threshold
- Έτσι η παραγωγή όλων των itemsets με $\text{support}(X) > t$ κάνει απλό τον υπολογισμό του $\text{confidence}(A \rightarrow B)$
- Τα itemsets αυτά ονομάζονται large ή **frequent** (συχνά)
- Το πρόβλημα εύρεσης έγκυρων κανόνων ανάγεται στο πρόβλημα εύρεσης συχνών συνόλων αντικειμένων

Εφαρμογές του προβλήματος

○ **Market Basket Data**

- **Baskets:** καλάθια αγορών S.M.
- **Items:** τα αντικείμενα που αγοράζει κανείς
- **Frequent itemsets:** Ποιά αντικείμενα αγοράζονται ταυτόχρονα με κάποια άλλα (καλύτερη τοποθέτηση)

○ **Ανάλυση κειμένων**

- **Baskets:** κείμενα
- **Items:** λέξεις
- **Frequent itemsets :** σύνολα λέξεων που εμφανίζονται συχνά μαζί δηλώνουν κείμενα με παρόμοιο περιεχόμενο

○ **Ανάλυση Web**

- **Baskets:** web pages
- **Items:** σελίδες που κάνουν link σε αυτές
- **Frequent itemsets:** σελίδες με κοινά εισερχόμενα links ίσως έχουν παρόμοιο θέμα

Απλοϊκός αλγόριθμος

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

Προβλήματα και μέθοδοι επίλυσής τους

- **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]: βασική ιδέα

- **Monotonicity property**: Έστω Σ σύνολο αντικειμένων με n στοιχεία και $J = \text{row}(\Sigma)$ 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 1:** Σχηματισμός λίστας L_1 συχνών μονοσυνόλων
- **Βήμα $k+1$** ($k=1,2,\dots$): Σχηματισμός λίστας υποψηφίων $(k+1)$ -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 L_{k+1} των υποψηφίων frequent $(k+1)$ -itemsets και υπολογίζεται η συχνότητά του. Αν είναι συχνό παραμένει, αλλιώς διαγράφεται.

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 $\{c,d,f\}$
- $\{a,b,d\}$, $\{a,b,f\} \Rightarrow \{a,b,d,f\}$
- Υποσύνολα: $\{a,b,d\}$ $\{a,b,f\}$ $\{a,d,f\}$ $\{b,d,f\}$
- $\{a,b,d,f\}$ είναι **candidate frequent**

- $\{a,b,d\}$, $\{a,d,f\} \Rightarrow \{\}$ // διαφέρουν σε 2 items

- $\{b,c,f\}$, $\{b,c,g\} \Rightarrow \{b,c,f,g\}$
- Υποσύνολα: $\{b,c,f\}$, $\{b,c,g\}$, $\{b,f,g\}$, $\{c,f,g\}$
- $\{b,c,f,g\}$ δεν είναι **candidate frequent**

Πλεονεκτήματα-μειονεκτήματα του A-priori

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

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

- Αποθήκευση της βάσης δεδομένων στη δομή FP-tree με 2 διασχίσεις
 - Υπολογισμός των συχνοτήτων όλων των singletons (1-itemsets)
 - ❖ Αφαίρεση των non-frequent items και επαναδιάταξη των εγγραφών με φθίνουσα σειρά συχνότητας των items
 - Τοποθέτηση των εγγραφών στο δέντρο
- Εξόρυξη των συχνών συνόλων αντικειμένων χωρίς candidate generation

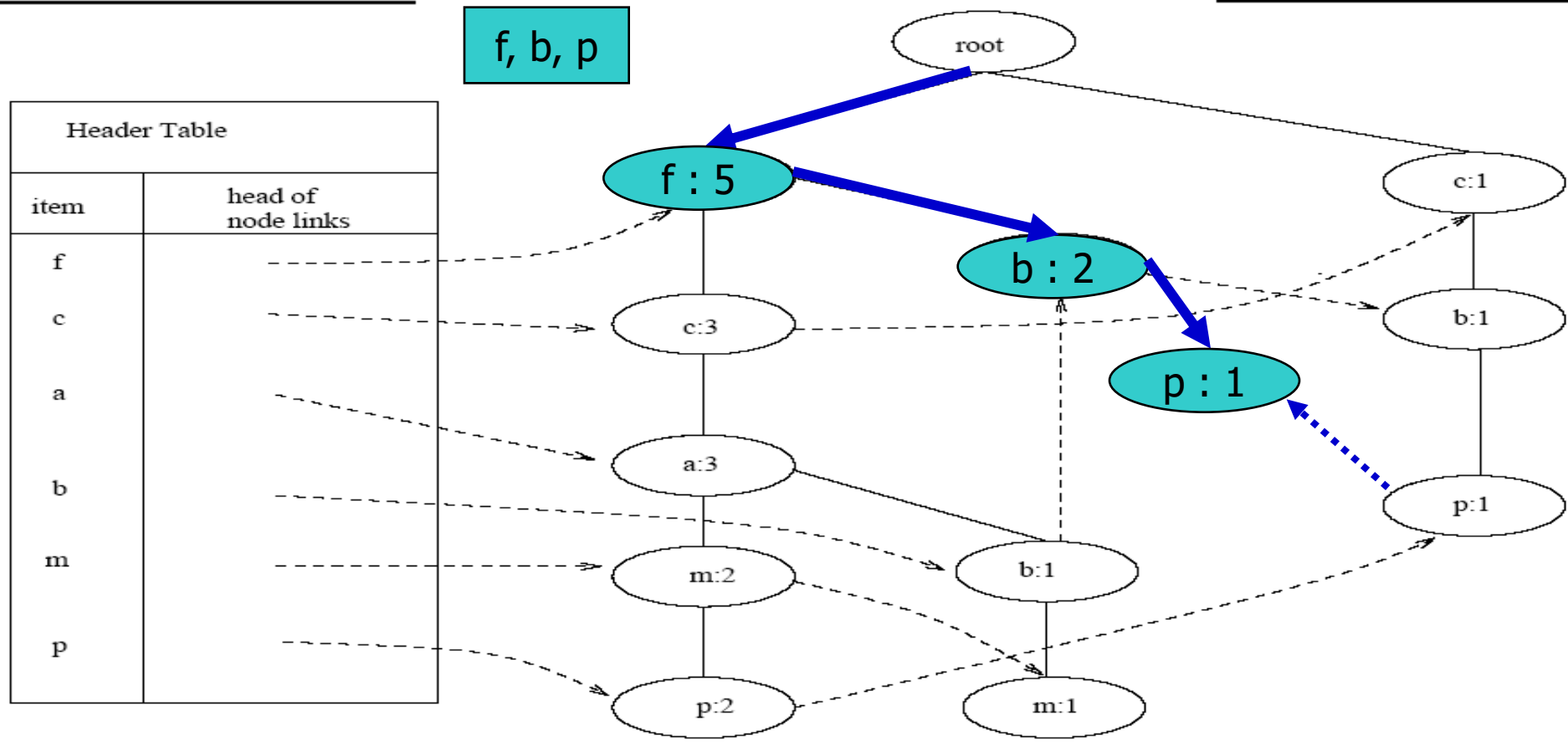
<i>TID</i>	<i>Items</i>
------------	--------------

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

<i>TID</i>	<i>Items</i>
------------	--------------

101	<i>f, c, a, m, p</i>
102	<i>f, c, a, b, m</i>
103	<i>f, b</i>
104	<i>c, b, p</i>
105	<i>f, c, a, m, p</i>

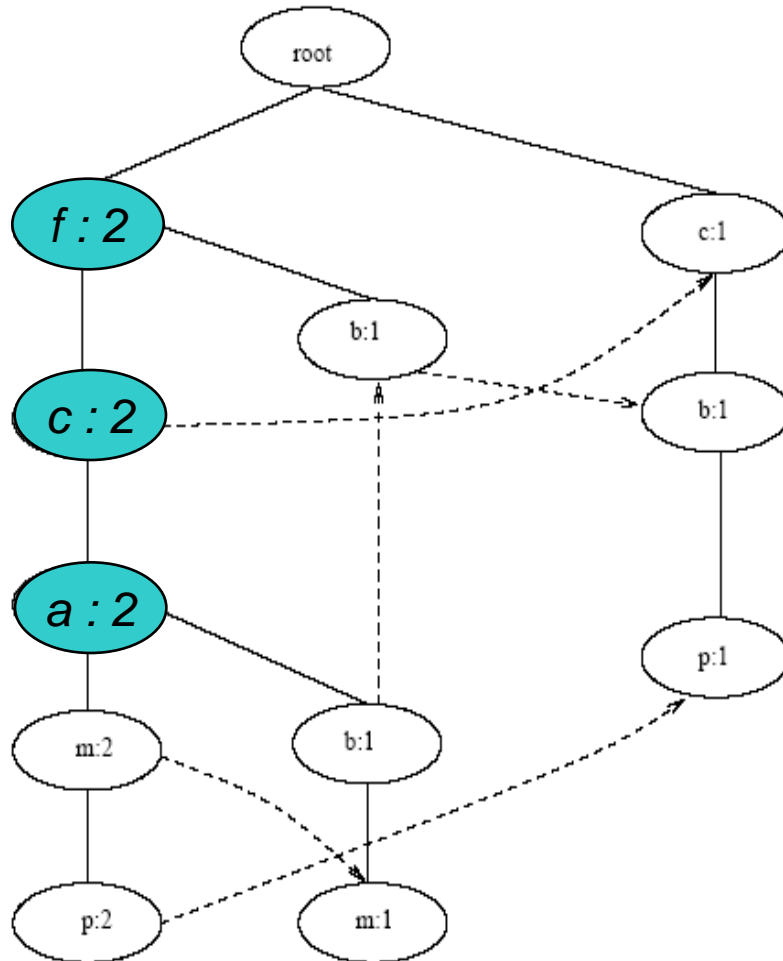
Κατώφλι συχνότητας 3



Αλγόριθμος FP-growth

- Ο αλγόριθμος 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.
- Συνεχίζει με τα δέντρα που προκύπτουν για καθένα από αυτά τα δισύνολα
- Σε κάθε τέτοιο δέντρο προσπαθεί να φτιάξει τρισύνολα κ.λπ.

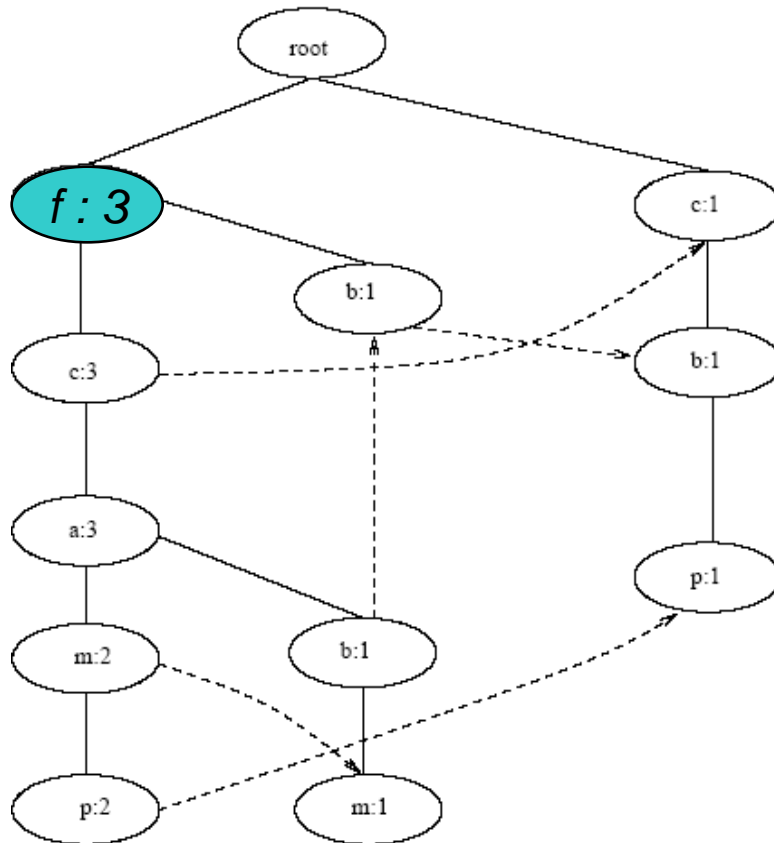
Παράδειγμα



Ο αλγόριθμος ξεκινά με το 'p'
2 μονοπάτια έχουν 'p'
Ενημέρωση συχνοτήτων
Υπολογισμός συχνοτήτων items
Συχνό το 'c'
Άρα το 'cp' είναι συχνό.

Threshold 3

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

Βελτιώσεις του A-priori

- Με χρήση hashing
 - Στο 1^ο πέρασμα:
 - Υπολογισμός συχνών 1-itemsets
 - Και όλων των 2-itemsets
 - Εισαγωγή σε hash table
 - Διαγραφή μη-συχνών «κουβάδων»
- Συρρίκνωση βάσης
 - Διαγραφή εγγραφών που δεν περιέχουν frequent 2-itemsets (υπερσύνολα: non frequent)

Παράδειγμα

TID	List of items
100	I1,I2,I5
200	I2,I4
300	I2,I3
400	I1,I2,I4
500	I1,I3
600	I2,I3
700	I1,I3
800	I1,I2,I3,I5
900	I1,I2,I3

Παράδειγμα hash function

$$h(x,y) = (\text{order } x) * 10 + (\text{order } y) \text{ mod } 7$$

Bucket address	0	1	2	3	4	5	6
Bucket count	2	2	4	2	2	4	4
Bucket contents	I1, I4	I1, I5	I2, I3	I2, I4	I2, I5	I1, I2	I1, I3
	I3, I5	I1, I5	I2, I3	I2, I4	I2, I5	I1, I2	I1, I3
			I2, I3			I1, I2	I1, I3
			I2, I3			I1, I2	I1, I3

Ερευνητικές 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