Asszociációs-szabályok, 2. rész

Csima Judit

BME, VIK,
Számítástudományi és Információelméleti Tanszék

2014. április 30. és május 7.

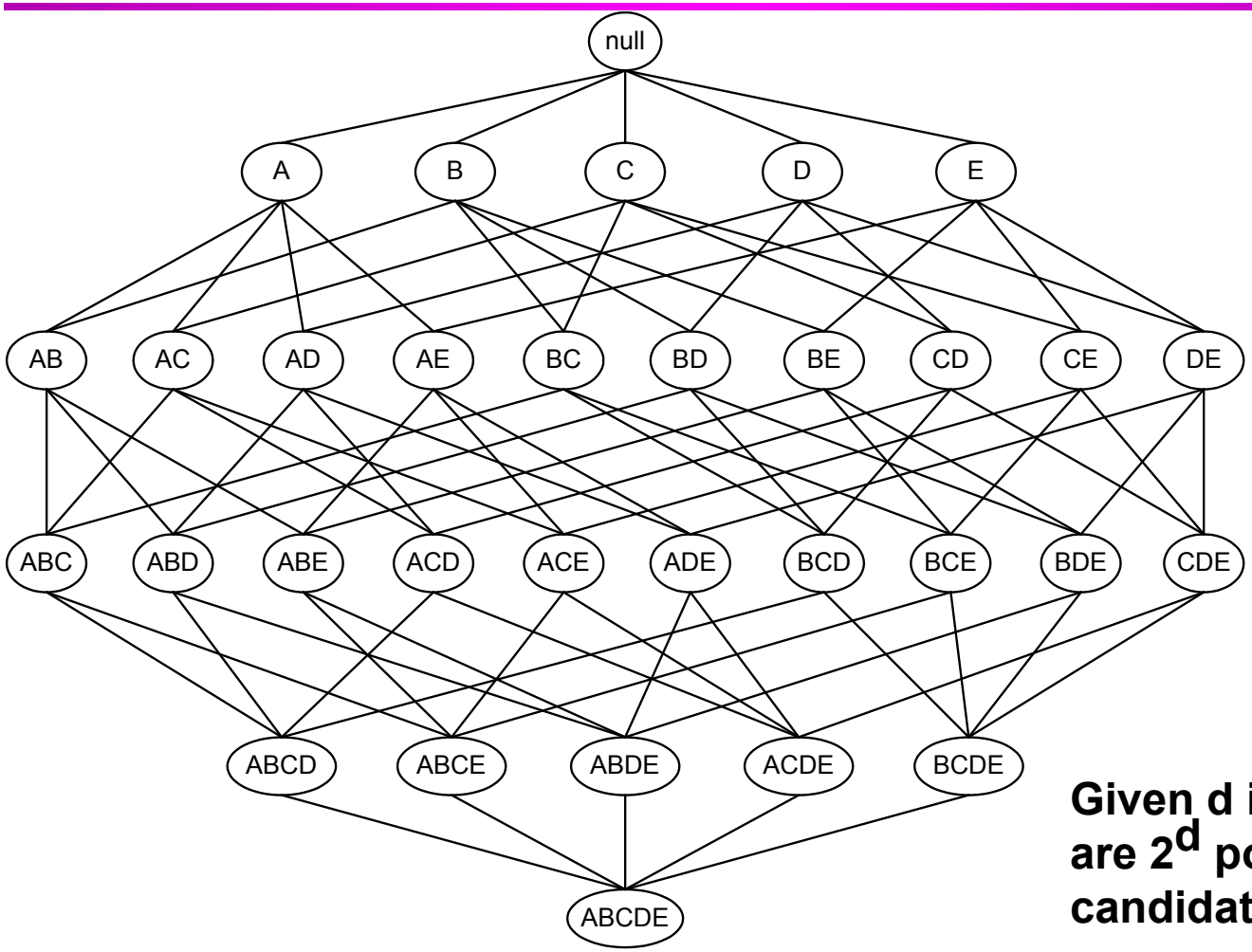
Eddig mi volt?

- Apriori-algoval gyakori elemhalmazok generálása
- a zárt gyakoriak és a hozzájuk tartozó tárolt σ értékekből az összes gyakori és ezek σ-jának meghatározása
- gyakoriak elemhalmazokból a nagy megbízhatóságú szabályok előállítása

Most mi lesz?

- Apriori algo helyett más módszerek a gyakori elemhalmazok megtalálására:
  - általános stratégiák az elemhalmazok hálójának bejárására
  - FP-fa építő algo
  - Eclat algo

Frequent Itemset Generation



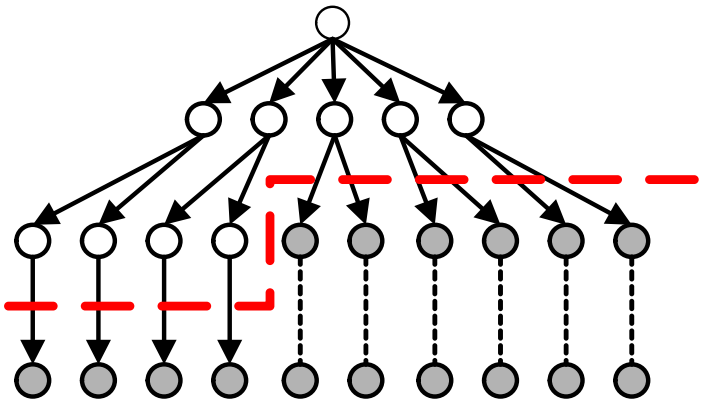
Given d items, there are 2^d possible candidate itemsets

Általános stratégiák a háló bejárására

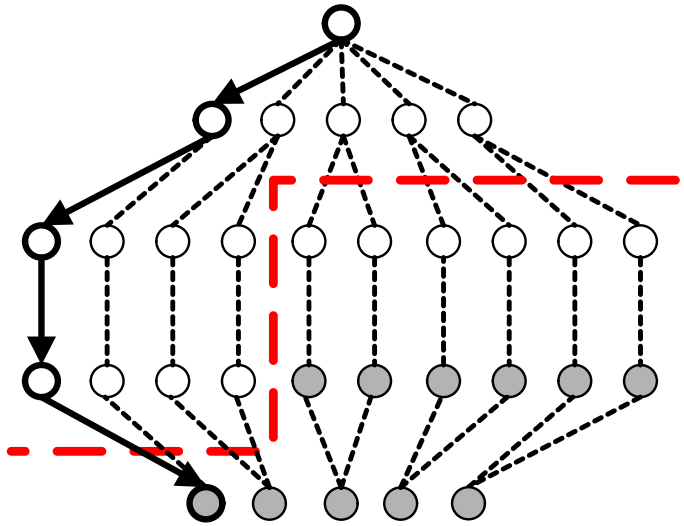
- az Apriori algo lényegében egy szélességi bejárást valósít meg
- más stratégiák:
  - mélységi bejárás
  - ekvivalencia-osztályok szerinti bejárás
- mindegyik esetben alkalmazzuk az Apriori-elvet: ha egy EH nem gyakori, akkor egyetlen olyan halmaz sem gyakori, aki őt tartalmazza vagy (ami ugyanez): ha egy elemhalmaz gyakori, akkor minden része is az

Alternative Methods for Frequent Itemset Generation

- Traversal of Itemset Lattice
  - Breadth-first vs Depth-first



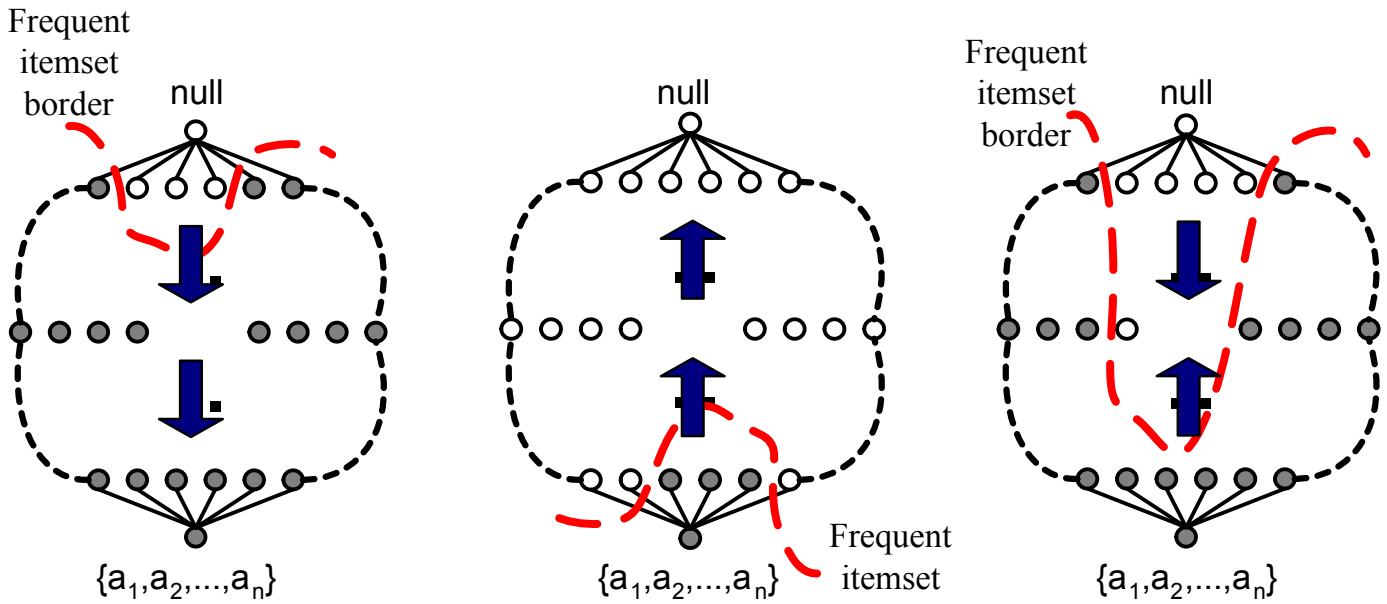
(a) Breadth first



(b) Depth first

Alternative Methods for Frequent Itemset Generation

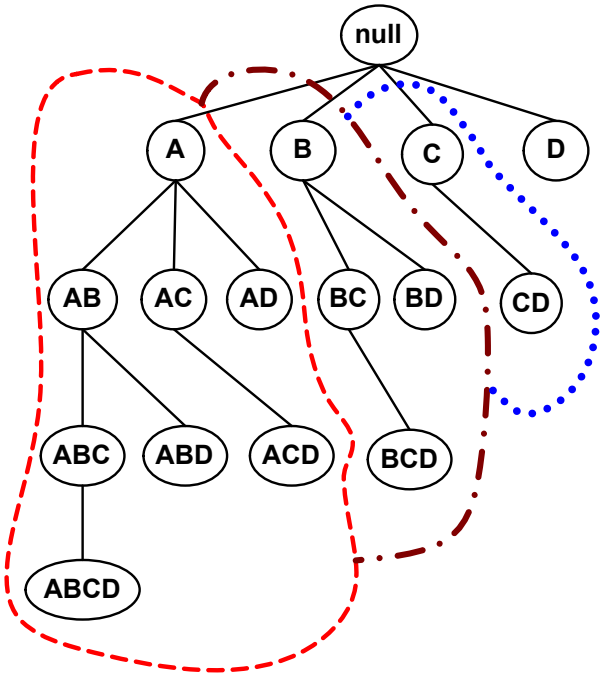
- Traversal of Itemset Lattice
  - General-to-specific vs Specific-to-general



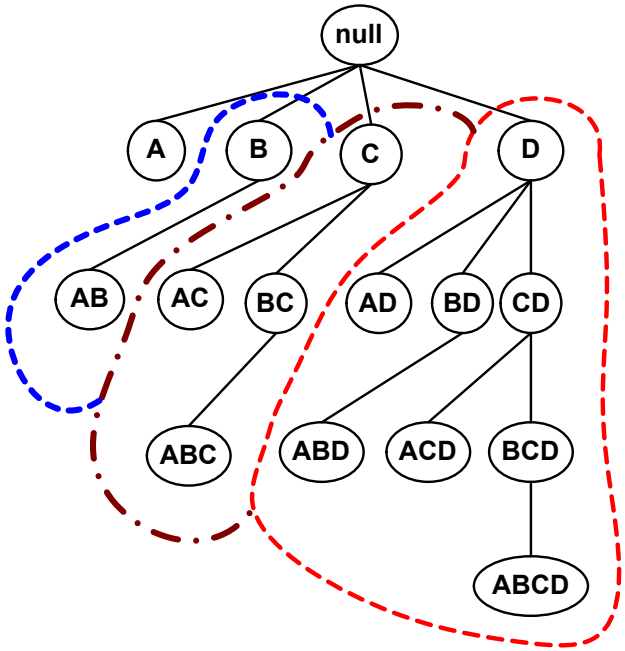
(a) General-to-specific (b) Specific-to-general (c) Bidirectional

Alternative Methods for Frequent Itemset Generation

- Traversal of Itemset Lattice
  - Equivalent Classes



(a) Prefix tree



(b) Suffix tree

FP-fa építő algo

- az Apriori-algo minden k esetén F_k számolásakor újra és újra végignézte a tranzakciókat
- FP-algo: először egy preprocesszálással egy szófa-jellegű struktúrát hotunk létre
- ezután ekvivalenciaosztályok szerint, az osztályokon belül valami bejárást használva végignézzük az elemhalmazokat, kiválogatjuk a gyakoriakat
- ehhez a válogatáshoz csak a felépített FP-fát használjuk, az eredeti tranzakciókat nem

FP-fa építés, előkészítés

- meghatározzuk az egyes elemek gyakoriságát
- a nem gyakori elemeket kidobjuk minden tranzakcióból
- a gyakori elemeket gyakoriság szerint csökkenő sorrendbe rendezzük, minden tranzakciót átrendezünk ezen sorrend szerint
- az így kapott (lerövidített és átsorrendezett) tranzakciókkal fogunk dolgozni

FP-fa

- olyan szó-fát akarunk építeni, ahol
  - minden gyökértől különböző csúcs egy elemmel van címkézve
  - a csúcsok mellett egy-egy számláló is van
- egy nem-gyökér csúcs útcímkéje az odáig vezető úton levő csúcsok címkéiből álló szó
- a csúcshoz tartozó számláló azt az értéket adja meg, hogy hány olyan tranzakció van, ami a csúcs útcímkéjének megfelelő elemhalmazzal kezdődik
- azt is akarjuk, hogy minden tranzakció minden lehetséges kezdőrésze reprezentálva legyen a fában egy csúccsal

Először nézzük, hogy ezt a fát hogy állítjuk elő, aztán nézzük majd, hogy mire lesz jó.

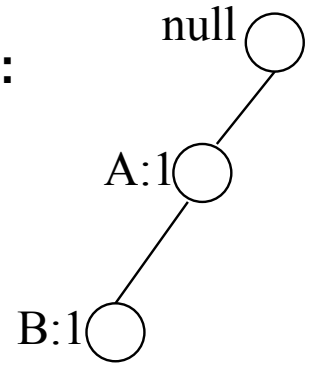
FP-fa építése adott tranzakcióhalmazhoz

- első tranzakció: egy utat hozunk létre, a tranzakciónak megfelelő sorrendben használva az elemeket, minden számláló 1
- újabb tranzakciók:
  - a tranzakcióban szereplő elemekből adódó szót követjük a fában, ha új elágazás kell, akkor létrehozzuk
  - az újonnan létrehozott csúcsok számlálója 1
  - a régi csúcsok számlálóit eggyel növeljük
- ezt csináljuk, amíg el nem fogynak a tranzakciók
- a végén minden tranzakció minden kezdőszelete reprezentálva lesz és a számláló éppen azt mutatja, hogy ez a kezdőszelet hányszor szerepelt
- lesznek még pointerek is, amik összekötik az azonos csúcs-címkéjű csúcsokat

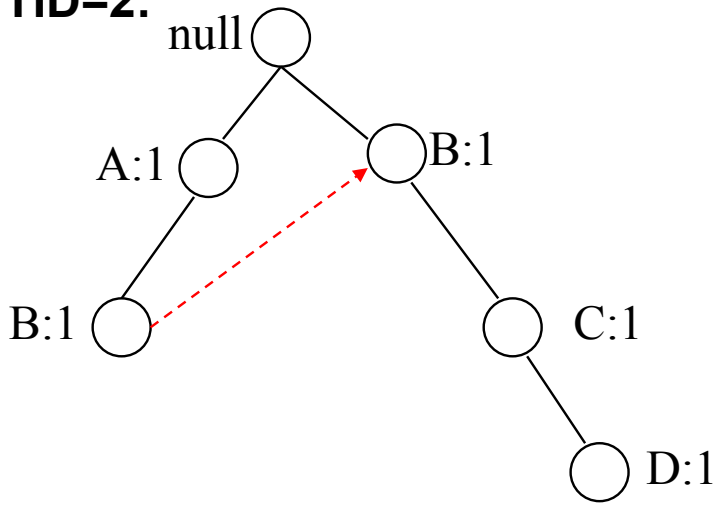
FP-tree construction

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

After reading TID=1:



After reading TID=2:



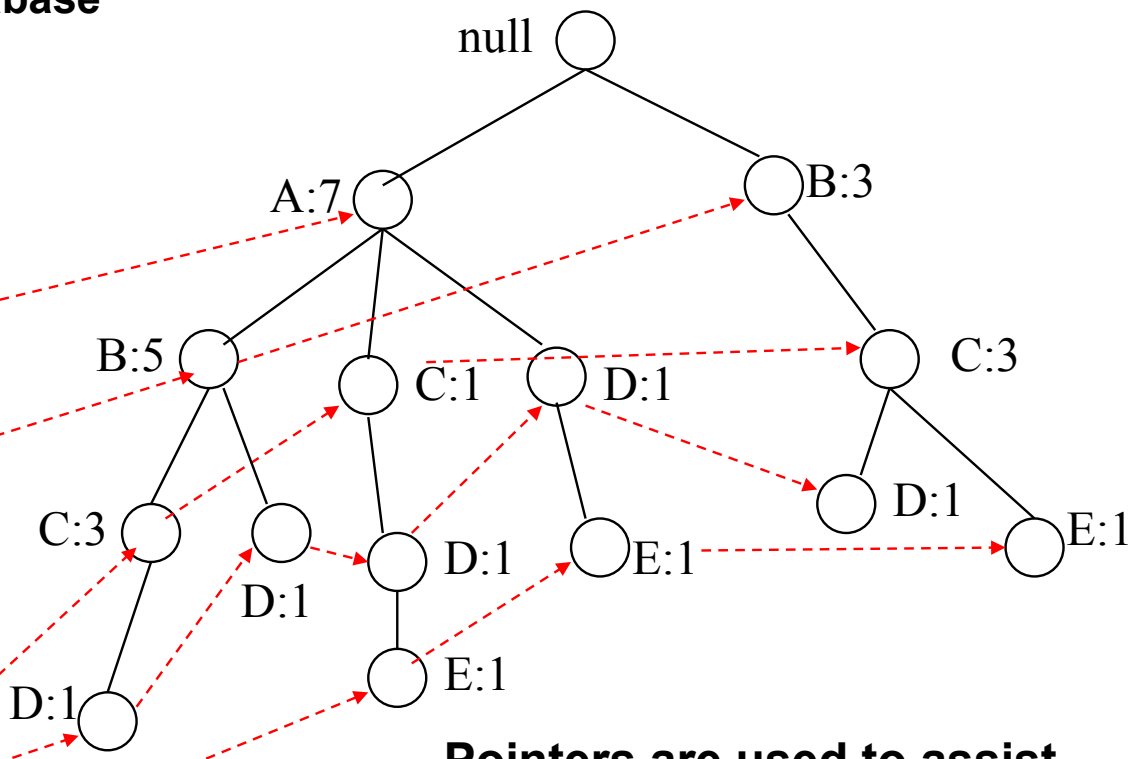
FP-Tree Construction

TID	Items
1	{A,B}
2	{B,C,D}
3	{A,C,D,E}
4	{A,D,E}
5	{A,B,C}
6	{A,B,C,D}
7	{B,C}
8	{A,B,C}
9	{A,B,D}
10	{B,C,E}

Transaction Database

Header table

Item	Pointer
A	
B	
C	
D	
E	



Pointers are used to assist frequent itemset generation

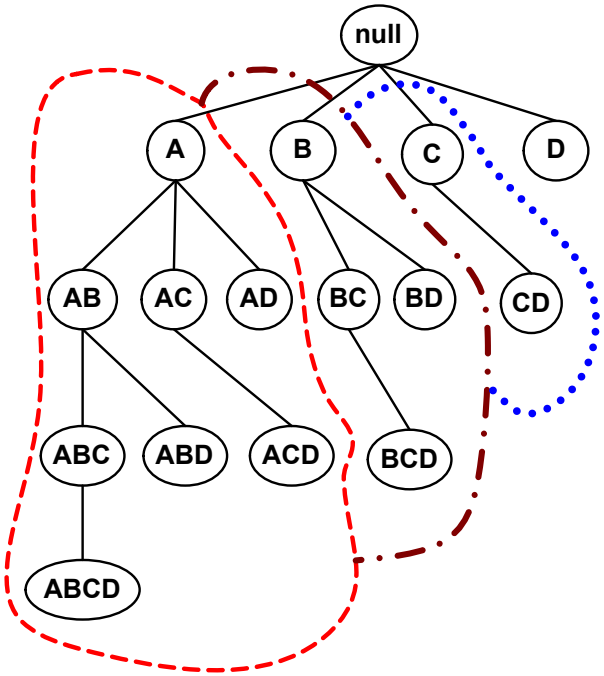
Gyakori elemhalmazok felkeresése

- ekvivalencia-osztályonként járjuk be a részhalmazok hálóját
- Ekvivalencia-osztályok:
  - azok az elemhalmazok, amikben van a legritkább (a példában e)
  - amikben nincs e, de van d
  - amikben nincs se d, se e, de van c
  - ...
  - amikben nincs se e, se d, se c, se b, de van a

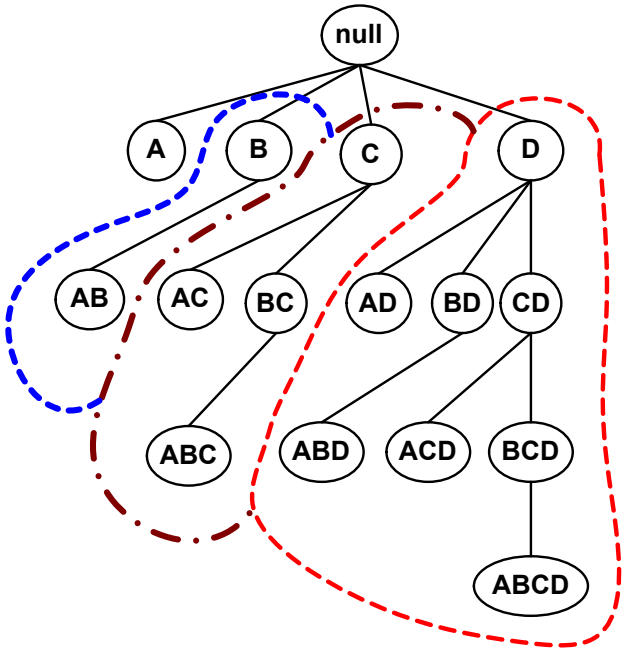
  Az egyes ekvivalenciaosztályokon belül pedig elemszám szerint növekvő sorrendben haladunk, az egyeleműtől indulva.

Alternative Methods for Frequent Itemset Generation

- Traversal of Itemset Lattice
  - Equivalent Classes



(a) Prefix tree



(b) Suffix tree

FP-fa felhasználása a bejárás során: első ekvivalencia osztály, amikben van e

- keressük azokat a gyakori elemhalmazokat, amikben van e
- nézzük meg először, hogy a csak e-t tartalmazó elemhalmaz gyakori-e (ez mindig az lesz)
- hogy nézzük ezt meg?
  - az e-hez tartozó pointereket követve összeadjuk az összes e csúcs-címkéjű fabeli csúcs számlálóját, ez éppen az e elemhalmaz abszolut gyakoriságát adja meg
- ezután megnézzük az e-t tartalmazó egyik kételemű halmazt (a következő legkevésbé gyakori elemmel bővítve e-t): de-t vizsgáljuk
  - a felépített FP-fából akarjuk kinyerni de gyakoriságát
  - ehhez elkészítjük az eredeti fából az e szerinti feltétele fát

Feltételes fa e szerint

- egy olyan FP-fa, amiben e már nem szerepel
- azt mutatja a szereplő útcímkékhez, hogy hány olyan tranzakció van, ami
  - az adott útcímkével kezdődik és
  - van benne a végén e (az útcímkében szereplő elemeken kívül)
- minden olyan elemhalmaz szerepel útcímkeként a fában, ami előfordul olyan tranzakció elején, amiben van e

Az e szerint feltételes fa elkészítése

- az eredeti fából indulunk ki
- minden olyan ágat (csúcsot) elhagyunk, amiben (aminek folytatásában) nincs e
- a megmaradó csúcsok új számlálója az e címkéjű leszármazottjaik számlálóinak összege lesz
- elhagyjuk az ágak végéről az e-ket

Így éppen azt kapjuk, amit akartunk: a megmaradó csúcsok útcímkéi éppen azok az elemhalmazok lesznek, amikhez van velük kezdődő tranzakció, ami e-t is tartalmazza; a számlálók értéke pedig az ilyen előfordulások számát mutatja.

Hogyan döntöm el az e szerinti feltételes fa alapján, hogy kik a kételemű, e-t tartalmazó gyakoriak?

- minden, a fában szereplő egyeleműre megnézem, hogy mennyi az ilyen csúcs-címkéjű csúcsok számlálóinak összege
- ha ez nagyobb, mint a küszöb, akkor az adott elem e-vel együtt gyakori

Hogyan tovább?

- ha megvannak a kételemű, *e*-t tartalmazók, akkor nézzük meg a háromelemű, *e*-t tartalmazókat
- ezek csak a gyakorinak talált kételeműekből jöhetnek egy új elem hozzáadásával
- potenciális 3-eleműek: egy, a rendezés szerint korábbi elemmel bővítünk (pl. *ce*-t csak *b* vagy *a*-val, *d*-vel nem)
- a potenciális 3-eleműek előfordulási gyakoriságai az *e*-re feltételes fából kaphatók
- ha pl. *ce* potenciális bővítéseit vizsgáljuk, akkor elkészítjük az *e*-re feltételes fából a *c*-re feltételes fát
  - *d*-ket elhagyom
  - utána ugyanazt csinálom, mint korábban, az *e*-re feltételes fa elkészítésénél, csak most *e* helyett *c*-vel

e-t tartalmazók végignézése: összefoglalás

- növekvő elemszám szerint végignézem az elemhalmazokat, hogy gyakoriak-e
- a gyakoriságot az aktuális (feltételes) FP-fából olvasom le
- ha bővítem az elemhalmazt, akkor mindig csak sorrendben előbb levő elemmel próbálok bővíteni
- a bővítéskor feltételes fár készítek

Következő fázis: *e*-t nem, de *d*-t tartalmazó elemhalmazok

- elhagyok minden *e*-s levelet
- az így kapott fával ugyanazt csinálom, amit az előbbi ekvivalencia-osztálynál tettem, csak most *e* szerepét *d* játssza

További fázisok

- egyre kisebb és kisebb fákat nézek (minden olyan csúcsot levágok, ami nem szerepel az ekvivalencia-osztályhoz tartozó elemhalmazokban)
- az így kapott csonkolt fával az előbbi algot futtatom:
  - növekvő elemszám szerint végignézem az ekvivalencia-osztály elemhalmazait

FP-algo jellemzői

- a tranzakciókat csak a fa építése során kell végignéznem
- utána már csak a fát alakítom, ebből olvasom le az egyes elemhalmazok gyakoriságait
- persze közben alkalmazom az Apriori-elvet: ha valakiről kiderül, hogy nem gyakori, akkor a nála bővebbeket nem kell néznem
- az egész eljárás akkor is megy, ha nem gyakoriság szerint csökkenően vannak rendezve az elemek a tranzakciókon belül, csak akkor lassabb
- az viszont szükséges, hogy legyen valami sorrend és mind a tranzakciókban, mind a keresett gyakori elemhalmazokban ezen sorrend szerint legyenek az elemek

ECLAT algo

- más szisztéma
- nem azt írjuk fel, hogy melyik tranzakciókban mik az elemek, hanem azt, hogy írjuk fel az egyes elemekről, hogy melyik tranzakciókban vannak benne
- ezt vertikális felírásnak is nevezik

ECLAT

- For each item, store a list of transaction ids (tids)

Horizontal Data Layout

TID	Items
1	A,B,E
2	B,C,D
3	C,E
4	A,C,D
5	A,B,C,D
6	A,E
7	A,B
8	A,B,C
9	A,C,D
10	B

Vertical Data Layout

A	B	C	D	E
1	1	2	2	1
4	2	3	4	3
5	5	4	5	6
6	7	8	9	
7	8	9		
8	10			
9				

TID-list

ECLAT algo

- DFS-sel járjuk be az elemhalmazok hálóját
- a példában legyen a gyakorisági-küszöb 2
- ekkor E gyakori
- nézzük meg E gyerekeit: DE, CE, BE, AE gyakoriságai mik?
- pl. DE gyakorisága D és E oszlopának metzsetének magassága
- hasonlóan kapható a többi kételemű gyakorisága is

Továbblépés DFS-sel

- amelyik elemhalmazról éppen kiderült, hogy gyakori, arról tudom az őt tartalmazó tranzakciók halmazát
- az egy elemű bővítések gyakorisága ezen oszlop és a bővítő elem oszlopának metszetéből számolható

ECLAT összefoglalás

- nem gyakori egy-eleműek kidobálása
- vertikális felírás elkészítése
- DFS a fenti módon, a hálót reprezentáló gráfban az éllistában a csúcsok gyakoriság szerint csökkenően (ez gyorsítja a nem gyakoriak felismerését)
- bővülő elemhalmazok gyakorisága oszlopmetszet alapján

Milyen szabályokat akarok?

- eddig: supp és conf legyen magas
- ezekhez min_sup és min_conf küszöbök
- ezek beállítása nehéz
  - ha magasak, akkor esetleg érdekes szabályok is kiesnek
  - ha alacsonyak, akkor túl sok szabály marad bent, nehéz válogatni a tényleg jókat

Érdekes szabályok keresése

- a sok szabály közül, amire supp és conf elég nagy kiválogatni azokat, amik tényleg érdekesek:
  - váratlanok
  - hasznot hozhatnak
- ezek (mechanikus algoval) megfoghatatlan fogalmak
- megoldások:
  - valami ember válogassa ki az előszűrt szabályokból az érdekeseket (ez nem járható úz igazán)
  - valami szakértő előszűri, hogy milyen szabályokat keresünk: pl. A és B termékscsoport között van-e valami asszociációs összefüggés)
- supp és conf-on kívül valami más, ami méri valahogyan az érdekességet

Computing Interestingness Measure

- Given a rule $X \rightarrow Y$, information needed to compute rule interestingness can be obtained from a contingency table

Contingency table for $X \rightarrow Y$

	Y	$\overline{Y}$	
X	f_{11}	f_{10}	f_{1+}
$\overline{X}$	f_{01}	f_{00}	f_{o+}
	f_{+1}	f_{+0}	$\|T\|$

f_{11}: support of X and $\overline{Y}$
f_{10}: support of $\overline{X}$ and $\overline{Y}$
f_{01}: support of $\overline{X}$ and $\overline{Y}$
f_{00}: support of X and Y

Used to define various measures

- support, confidence, lift, Gini, J-measure, etc.

Drawback of Confidence

	Coffee	$\overline{\text{Coffee}}$	
Tea	15	5	20
$\overline{\text{Tea}}$	75	5	80
	90	10	100

Association Rule: Tea → Coffee

Confidence= P(Coffee|Tea) = 0.75

but P(Coffee) = 0.9

⇒ Although confidence is high, rule is misleading

⇒ P(Coffee|$\overline{\text{Tea}}$) = 0.9375

Lift-mutató, motiváció

- az előző fólia mutatja, hogy a conf és supp nem elég
- lehet, hogy egy elég jó támogatottságú, nagyon magas megbízhatóságú szabály teljesen butaság
- próbáljuk valahogy kizárni az előző fólián látható jelenséget
- hasonlítsuk össze az $X \to Y$ szabály conf-ját a Y relatív gyakoriságával (gyakoribb-e X mellett Y, mint általában?)

Lift-mutató

- $Lift(X \to Y) = \frac{conf(X \to Y)}{\frac{\sigma(Y)}{n}}$, ahol n a tranzakciók száma
- ez uaz, mint $\frac{\sigma(X \cup Y)}{\sigma(X)} \cdot \frac{n}{\sigma(Y)} = \frac{supp(X \cup Y)}{supp(X) \cdot supp(Y)}$
- ez igazából X és Y előfordulásának függetlenségét méri
- ha $Lift(X \to Y) = 1$ az azt jelenti, hogy függetlenek
- ha $Lift(X \to Y) > 1$ az azt jelenti, hogy Y gyakoribb X mellett, mint általában, ez érdekel minket

Mindenféle mérőszámok

- persze Lift sem mindenható, simán lehet olyan szabály, amire supp, conf és Lift is jó, de mégis butaság
- sok más mérőszám szabályok jóságára (következő fólia, de csak illusztráció!)
- általában sup, conf és vmi Lift-szerű, függetelenséget mérő mérték

There are lots of measures proposed in the literature

Some measures are good for certain applications, but not for others

What criteria should we use to determine whether a measure is good or bad?

What about Apriori-style support based pruning? How does it affect these measures?

#	Measure	Formula
1	ϕ-coefficient	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-\max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i)\log P(A_i), -\sum_j P(B_j)\log P(B_j))}$
8	J-Measure (J)	$\max\left(P(A,B)\log(\frac{P(B\|A)}{P(B)}) + P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),\right.$ $\left. P(A,B)\log(\frac{P(A\|B)}{P(A)}) + P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\right)$
9	Gini index (G)	$\max\Big(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A})[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]$ $-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B})[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]$ $-P(A)^2-P(\overline{A})^2\Big)$
10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B\|A), P(A\|B))$
12	Laplace (L)	$\max\left(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2}\right)$
13	Conviction (V)	$\max\left(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})}\right)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B)-P(A)P(B)$
17	Certainty factor (F)	$\max\left(\frac{P(B\|A)-P(B)}{1-P(B)}, \frac{P(A\|B)-P(A)}{1-P(A)}\right)$
18	Added Value (AV)	$\max(P(B\|A)-P(B), P(A\|B)-P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})} \times \frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\|A)-P(B), P(A\|B)-P(A))$