

Gyakori termékhalmozok kinyerése

Lukács András

alukacs@sztaki.hu

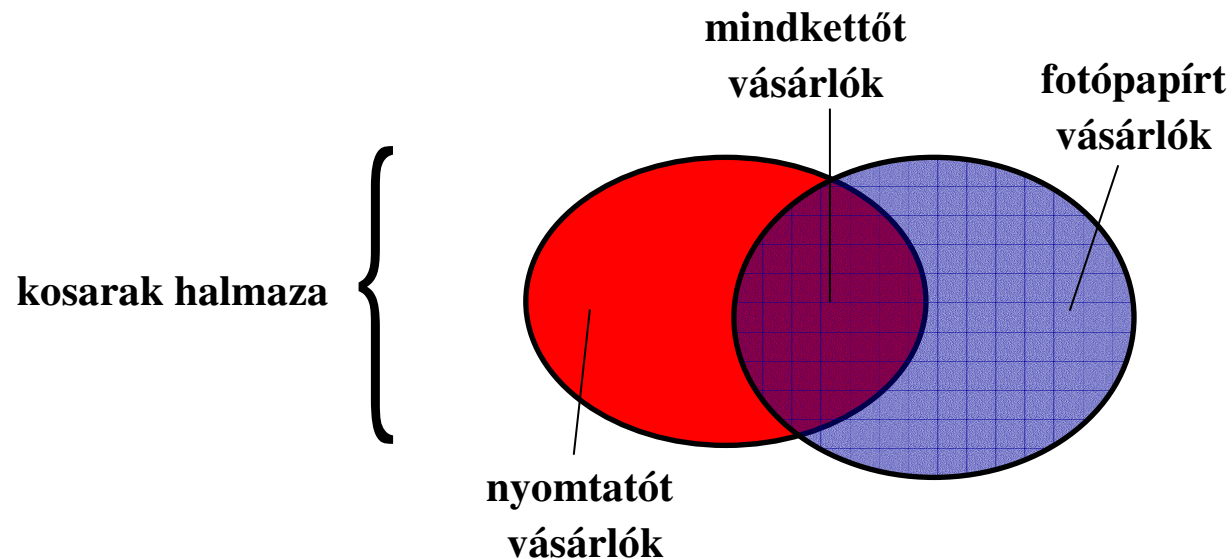
<http://www.sztaki.hu/~alukacs>

<http://www.ilab.sztaki.hu/websearch>

A vásárlói kosár modell

- termékhalmoz $T = \{t_1, \dots, t_k\}$
- kosarak $K = \{k_1, \dots, k_n\}$, $k_i = (id_i, k_i)$
- $X \subseteq T$ termékhalmozat a k kosár tartalmazza, ha $X \subseteq k$
- X termékhalmoz támogatottsága:

$$supp_k(X) = \frac{|k_i \in K : X \subseteq k_i|}{|K|}$$



Asszociációs szabályok

- Találjuk meg az összes adott minimális *támogatottsággal és bizonyossággal* rendelkező $X \rightarrow Y$ asszociációs szabályt!

- támogatottság (support): annak a valószínűsége, hogy egy kosár tartalmazza $X \cup Y$ -t

$$s = \text{supp}(X \cup Y)$$

- bizonyosság (confidence): annak a feltételes valószínűsége, hogy egy X -et tartalmazó kosár Y -t is tartalmazza

$$c = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}$$

kosár-id	kosár tartalma
10	A, B, C
20	A, C
30	A, D
40	B, E, F

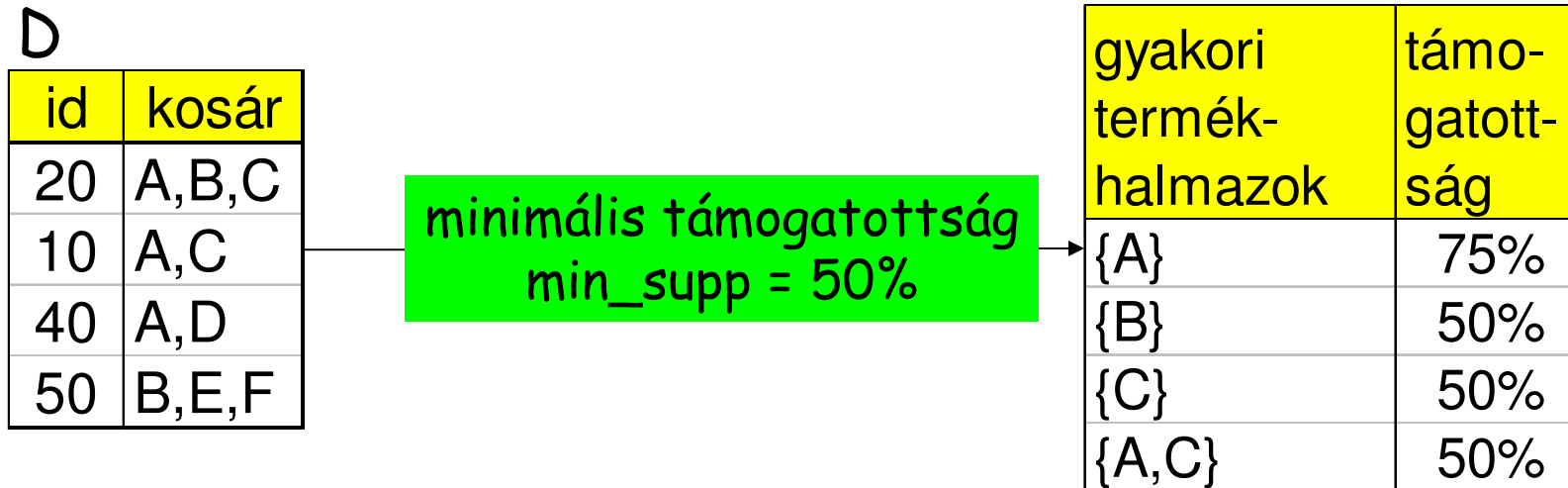
min_support := 50%

min_conf := 50%

$A \rightarrow C$ (50%, 66.7%)

$C \rightarrow A$ (50%, 100%)

APRIORI algoritmus



Cél: adott (apriori) minimális támogatottságú termék-halmazok megtalálása - ezeket gyakoriaknak hívjuk.

Az APRIORI elv: gyakori termék-halmazok *rész-halmazai* is gyakoriak (pl. ha {AC} gyakori, akkor {A} és {C} is gyakori).

Algoritmus: szétlében keresünk, *szintenként*, a gyakori termék-halmazok számossága szerint 1-től indulva a számosságot egyesével növelve haladunk addig, amíg találunk gyakori termék-halmazt.

Az APRIORI algoritmus

C_k : k méretű jelölt termékhalmozok

L_k : k méretű gyakori termékhalmozok

Input: vásárlói kosarak D listája

Algoritmus:

$L_1 = \{\text{gyakori termékek}\};$

for ($k = 1$; amíg $L_k \neq \emptyset$; $k++$) do begin

- $C_{k+1} = \text{generate}(L_k);$

- minden olyan $t \in D$ kosárra, amely tartalmaz egy $c \in C_{k+1}$ termékhalmozot növeljük meg c számlálóját eggyel;

- $L_{k+1} =$ azon C_{k+1} -beli termékhalmozok, amiknek megvan a minimális támogatottságuk;

end

Output: $\bigcup_k L_k;$

Agrawal & Srikant 1994, Mannila, et al. 1994

Az APRIORI algoritmus - jelöltállítás

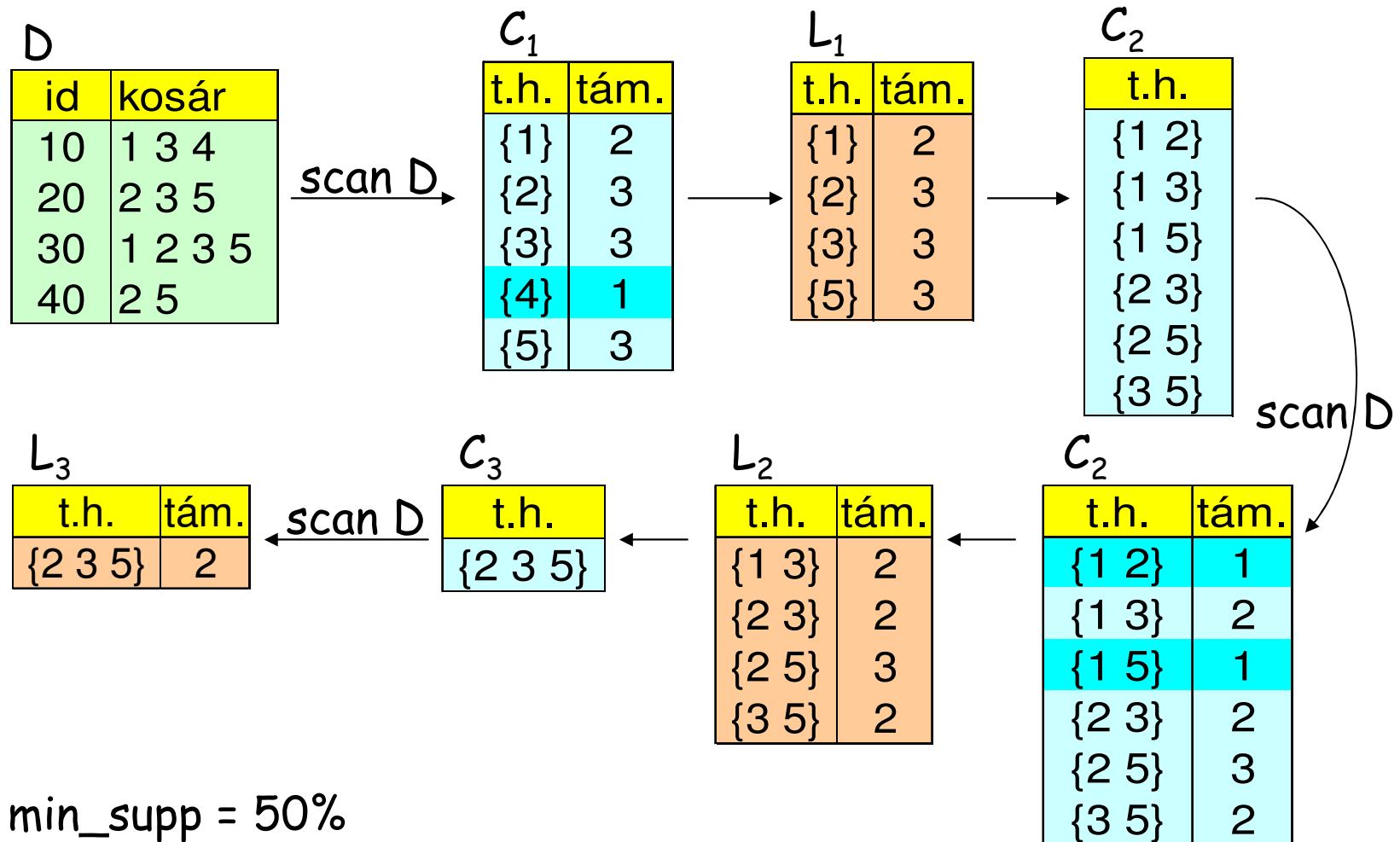
A C_k -beli k méretű jelölteket az L_{k-1} -beli $k-1$ méretű gyakori termékhalmozokból állítjuk elő. Egy k méretű jelölt minden $k-1$ méretű részhalmazának gyakornak kell lennie.

$C_k = \text{generate}(L_{k-1})$:

Legyen $<$ rendezés az L_{k-1} -ban előforduló termékeken.

- minden $p = \{p_1, \dots, p_{k-1}\}$ és $q = \{q_1, \dots, q_{k-1}\} \in L_{k-1}$ esetén, ahol $p_1 = q_1, \dots, p_{k-2} = q_{k-2}$ és $p_{k-1} < q_{k-1}$ legyen a $\{p_1, p_2, \dots, p_{k-1}, q_{k-1}\}$ termékhalmoz a C_k egy eleme;
- minden $c \in C_k$ termékhalmoz minden $s \subset c$ $k-1$ méretű részhalmazára ha $s \notin L_{k-1}$ akkor c törlendő C_k -ból.

Az APRIORI algoritmus - példa



min_supp = 50%

$L = \{\{1\}, \{2\}, \{3\}, \{5\}, \{1\ 3\}, \{2\ 3\}, \{2\ 5\}, \{3\ 5\}, \{2\ 3\ 5\}\}$

Az APRIORI algoritmus adatstruktúrái

Problémák a jelöltek támogatásának kiszámolásakor:

- túl sok jelölt
- egy kosár sok jelöltet tartalmazhat

Módszer 1:

- a jelölteket egy hash-fában tároljuk
- a hash-fa levelei tartalmazzák a jelöltek listáját és a támogatottságuk számlálóját
- a hash-fa belső pontjai hash-táblák

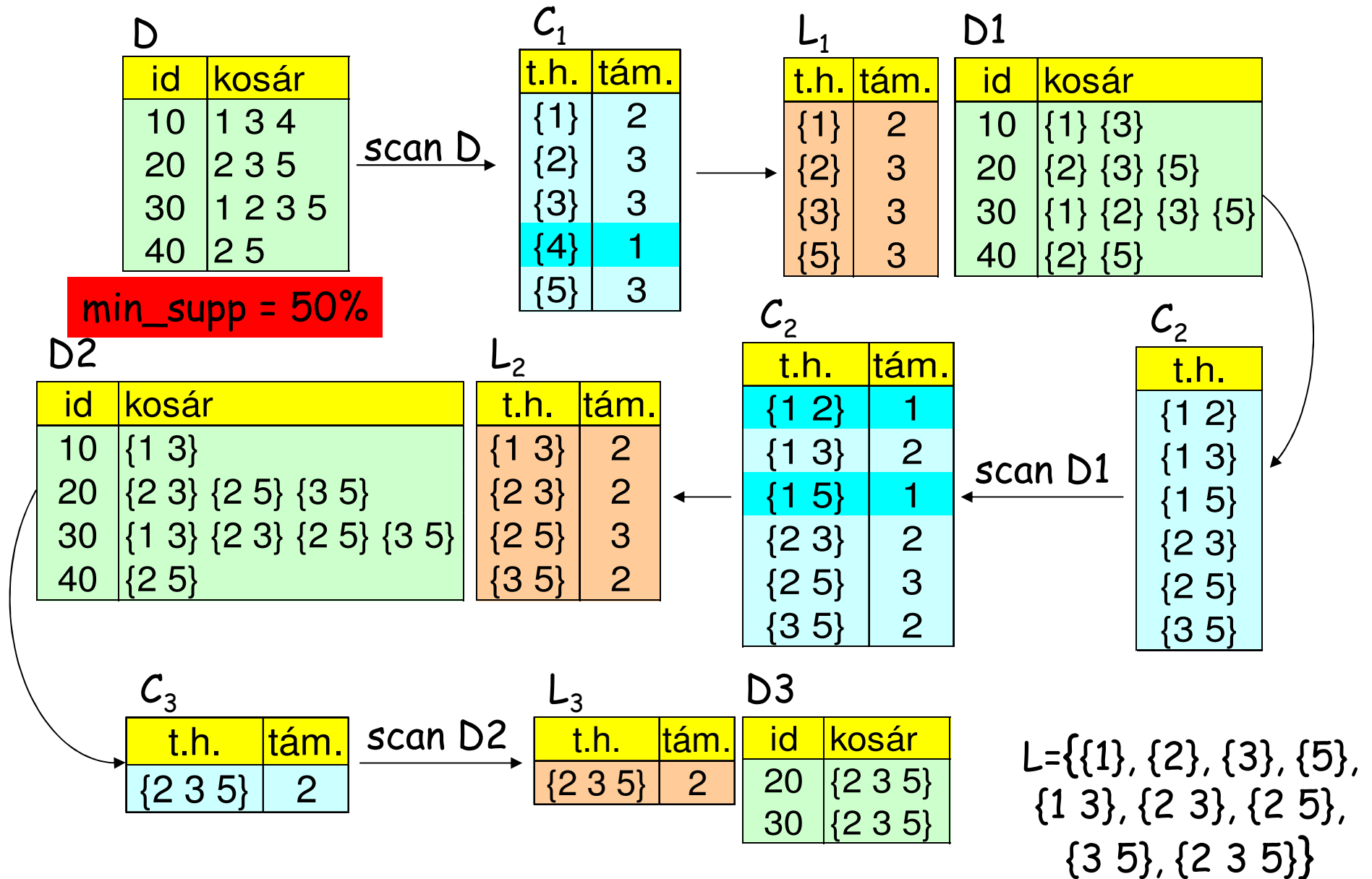
Módszer 2:

- a jelölteket egy szó-fában (tree) tároljuk

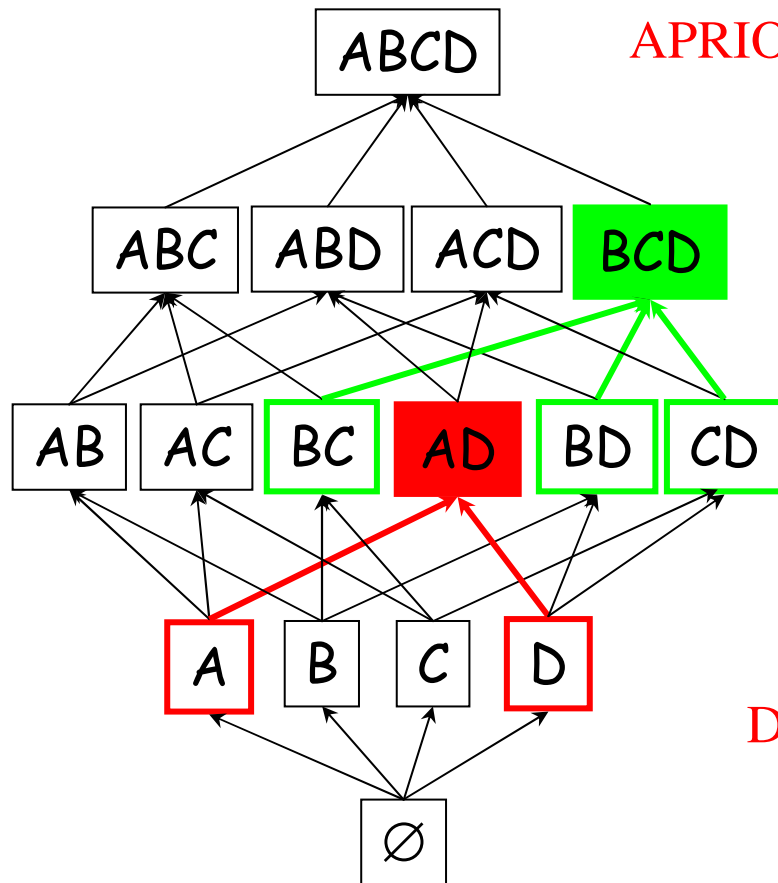
APRIORI algoritmus - gyorsítási ötletek

- **Adatbázis kivonatolása (APRIORI-TID)**: ha egy kosár nem tartalmaz k méretű gyakori termékhalmozat, akkor $k+1$ méretűt sem.
- **Dinamikus termékhalmoz számlálás** (dynamic itemset counting, **DIC**): rögtön elkezdjük egy termékhalmoz gyakoriságának számlálását, amint minden részhalmazáról kiderül, hogy gyakori.
- **Hashelés (DHP)**: egy gyakori termékhalmozat tartalmazó hash vödörben lévő termékhalmozok összes előfordulásainak száma is "gyakori".
- **Adatbázis partíciója (Partíciós- és Toivonen-algoritmus)**: minden gyakori termékhalmoz gyakori az adatbázis tetszőleges partíciójának legalább egy osztályában.

Adatbázis kivonatolása: APRIORI-TID



Dinamikus termékhalmoz számlálás: DIC algoritmus



APRIORI

Kosarak adatbázisa

1. átnézés (1 méretűekért)

2. átnézés (2 méretűekért)

...

- Amint nyilvánvaló, hogy **A** és **D** gyakori elkezdjük **AD** számlálását
- Amint **BCD** minden 2-elemű részhalmozáról kiderül, hogy gyakori elkezdjük **BCD** számlálását

DIC

1 eleműek indul

1 eleműek vége

2 eleműek indul

3 eleműek indul

2 eleműek vége

3 eleműek vége

S. Brin, R. Motwani, J. Ullman and S. Tsur: Dynamic itemset counting and implication rules for market basket data. SIGMOD'97

DIC algoritmus - technikai trükkök

- Az adatok **véletlen** megkeverése, véletlen sorrendben vesszük az adatok blokkjait.
- **Kisebb támogatottság** mint a *min_supp* legyen elég a számlálás megkezdéséhez.
- Nem minden rekord olvasása után ellenőrizzük a számlálókat, hanem csak kisebb-nagyobb rekordszám **intervallum**onként.

Hashelve leszámolás: DHP

D

id	kosár
10	1 3 4
20	2 3 5
30	1 2 3 5
40	2 5

scan D

C_1

t.h.	tám.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

H_1

termék halmazok	hash-érték	össztámogatottság	t.h.	tám.
{1 4}, {3 5}	0	3	{1 2}	1
{1 5}	1	1	{1 3}	2
{2 3}	2	2	{1 4}	1
{2 4}, {4 5}	3	0	{1 5}	1
{2 5}	4	3	{2 3}	2
{1 2}	5	1	{2 4}	0
{1 3} {3 4}	6	3	{2 5}	3
			{3 4}	1
			{3 5}	2
			{4 5}	0

min_supp = 50%

scan D
csak C_1 -et
használva

C_2

t.h.	tám.
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

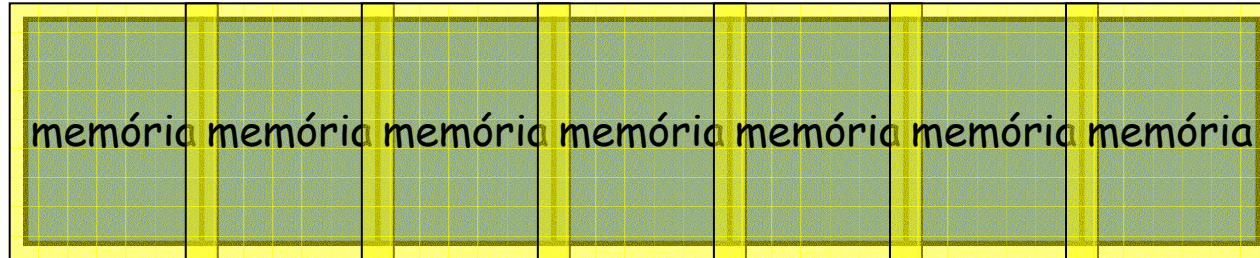
$$\text{hash}(\{t_1 \ t_2\}) = 10t_1 + t_2 \pmod{7}$$

C'_2

t.h.	tám.
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

scan D
 C_1 -et és H_1 -et
is használva

Partíciós algoritmus



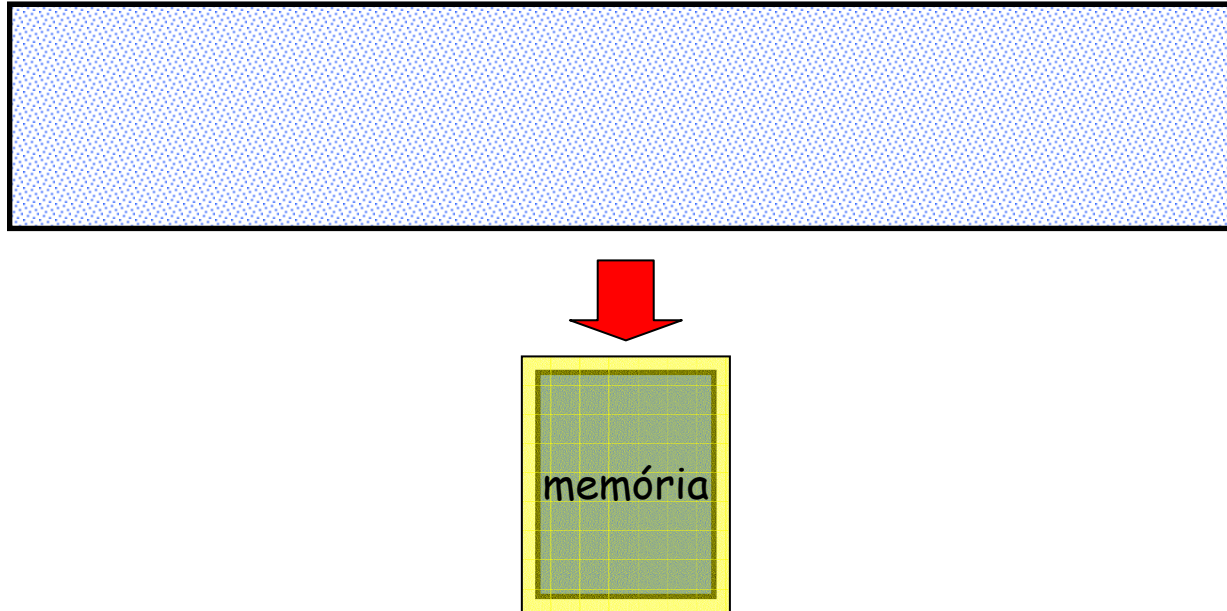
Észrevétel: minden termékhalmaz, ami gyakori egy D adatbázisban annak gyakorinak kell lennie D tetszőleges partíciójának legalább egy osztályában.

Algoritmus:

- partícionáljuk az adatbázist úgy, hogy egy osztály beleférjen a memóriába
- osztályonként megkeressük a gyakoriakat (ez most gyors)
- ezek együttesen lesznek a jelöltek a teljes adatbázisban gyakoriak megtalálásához

A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association in large databases, VLDB 1995

Mintavételezés



- Egyenletes eloszlás szerint véletlen kosarakat veszünk az adatbázisból.
- A mintaadatbázisban (gyorsan, pl. memóriában) megkeressük a gyakori termékhalmozokat.

Mintavételezés hibabecslése

Mekkorának kell lennie az $M \subseteq K$ mintának, ha egy termékhalmoz K kosarakban való előfordulásának relatív gyakoriságától legalább $1-\delta$ valószínűséggel legfeljebb ε hibával térjünk el?

$$\delta(K, M, X) := \text{Prob} \left(\left| \frac{\text{supp}_M(X)}{|M|} - \frac{\text{supp}_K(X)}{|K|} \right| \geq \varepsilon \right)$$

Ha a K -ból kivett M minta X szempontjából **egyenletes**, akkor

$$\text{supp}_M(X) \approx \text{Binom}(|M|, p), \text{ ahol } p = \frac{\text{supp}_K(X)}{|K|}$$

egy **binomiális valószínűségi változó**, azaz

$$\text{Prob}(\text{supp}_M(X) = c) \approx \binom{|M|}{c} \cdot p^c (1-p)^{|M|-c}$$

$$\begin{aligned}\delta(K, M, X) &= \text{Prob}\left(\left|\frac{\text{supp}_M(X)}{|M|} - \frac{\text{supp}_K(X)}{|K|}\right| \geq \varepsilon\right) = \\ &= \text{Prob}\left(\left|\frac{\text{Binom}(|M|, p)}{|M|} - p\right| \geq \varepsilon\right) = \text{Prob}(|\text{Binom}(|M|, p) - p|M| \geq \varepsilon|M|)\end{aligned}$$

$$p|M| = E(\text{Binom}(|M|, p)) \quad \text{várható érték}$$

$$\delta(K, M, X) = \text{Prob}(|\text{Binom}(|M|, p) - E(\text{Binom}(|M|, p))| \geq \varepsilon|M|) \leq 2e^{-2\varepsilon^2|M|}$$

Chernoff egyenlőtlenség

Ha $\geq 1-\delta$ valószínűséggel $\leq \varepsilon$ hibát akarunk elérni, akkor legyen

$$|M| \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

Mit jelent a mintavételezés hibájának ez a becslése?

Ha egy közvélemény kutatás során $|M|$ embert kérdeztek meg akkor egy adott hibakorláton legfeljebb mekkora valószínűséggel lépünk túl?

$$|M| \geq \frac{1}{2\varepsilon^2} \ln \frac{2}{\delta}$$

$ M $	hibakorlát	valószínűség
1500	3%	0,134
2000	3%	0,055
3000	3%	0,009
26500	1%	0,01

Toivonen-algoritmus

Úgy indul mint a mintavételezős algoritmus, M a minta a K kosáradatbázisból. Jelölje L_M a mintaadatbázis alapján gyakorinak talált termékhalmozok halmazát (ez viszonylag gyorsan elkészül).

Elkészítjük L_M -ből az esélyes jelöltek $EJ(L_M)$ halmazát. **Esélyes jelölt** minden olyan X termékhalmoz, ami maga nem gyakori, de minden valódi része gyakori.

$$EJ(L_M) = \{X \text{ termékhalmoz: } X \notin L_M \text{ és } \forall Y \subset X, Y \subseteq L_M\}$$

Most meghatározzuk az $L_M \cup EJ(L_M)$ -beli termékhalmozok K -beli támogatottságát. Ez a K adatbázis egyszeri átnézését jelenti.

Tétel: Ha $EJ(L_M)$ -beliek egyike sem gyakori, akkor már megtaláltunk minden gyakori termékhalmozat K -ban.

Bizonyítás: Indirekt tegyük fel, hogy $\exists X \in L_K \setminus L_M$. Ekkor $\exists Y \subset X$ legkisebb M -ben még nem gyakori termékhalmozat. Így minden Y -nál kisebb termékhalmozat L_M -beli, tehát $Y \in EJ(L_M)$. De $Y \in L_K$ is - ellentmondás.

Tétel: Legyen $L'_M := L_K \cap (L_M \cup EJ(L_M))$. Ha

$$L_M \cup EJ(L_M) = L'_M \cup EJ(L'_M),$$

akkor $L'_M = L_K$, azaz L'_M pontosan a K -ban gyakori termékhalmozatok halmaza.

Bizonyítás: Meg kell mutatnunk, hogy $L_K \subseteq L'_M$. Indirekt tegyük fel, hogy $\exists X \in L_K \setminus L'_M$. Ekkor $X \notin L_M \cup EJ(L_M)$. Tekintsük a legkisebb $Y \subset X$ halmozatot, ami még $\notin L_M \cup EJ(L_M)$. Így minden Y -nál kisebb termékhalmozat L_M vagy $EJ(L_M)$ -beli és gyakori, tehát $Y \in EJ(L'_M) = EJ(L_K \cap (L_M \cup EJ(L_M)))$. De $Y \notin L_M \cup EJ(L_M)$ - ellentmondás.