

JAVASLAT A TOP-K ELEMCSERÉK KERESÉSÉRE NAGY ONLINE KÖZÖSSÉGEKBEN

*Supporting Top-k item exchange
recommendations in large online communities*

Barabás Gábor
Nagy Dávid
Nemes Tamás

Probléma

- Cserekereskedelem nagy online közösségekben
 - Online fórumon
 - Kézi keresés nehéz
 - Hasonló értékű tárgyra
 - Újabb és újabb felmerülő igények

Igény

- Felesleges és hiányzó eszközök nyilvántartása
- Automatikus cserék ajánlása
- Hasonló értékű eszközök
- Friss lista => gyakori frissítés

- Példák: Schede, iswap, Gumtree (valódi)
FrontierVille (virtuális)

Ötlet és kihívások

- Ajánlások: lehetséges cserék csoportja
 - k legelőnyösebb (top- k)
- Mindkét fél számára megfelelő cserék legyenek
- NP-teljes probléma
- Valós idejű frissítés
 - sok frissítés lehetséges egyszerre

Módszerek

- Circular Single-item Exchange Model (CSEM)
 - 1 tárgy cseréje
 - kört határoz meg
 - Hátrány:
 - tárgyaknak nincs értéke
 - 1 tárgy cseréje nem elég hatékony
 - csere csak akkor ha mindenki elfogadja

Módszerek

- Binary Value-based Exchange Model (BVEM)
 - több tárgy
 - csere két felhasználó között akkor, ha közel azonos értékűek a felajánlott tárgyak
- Jó alapötlet
- Kiegészítése a Top-K Csere Ajánlás (Top-K Exchange Recommendation)

BVEM példa

ID	Name	Price
I ₁	Nail	\$10
I ₂	Ribbon	\$20
I ₃	Screw er	\$70
I ₄	Hammer	\$80
I ₅	Paint	\$100
I ₆	Drill	\$170

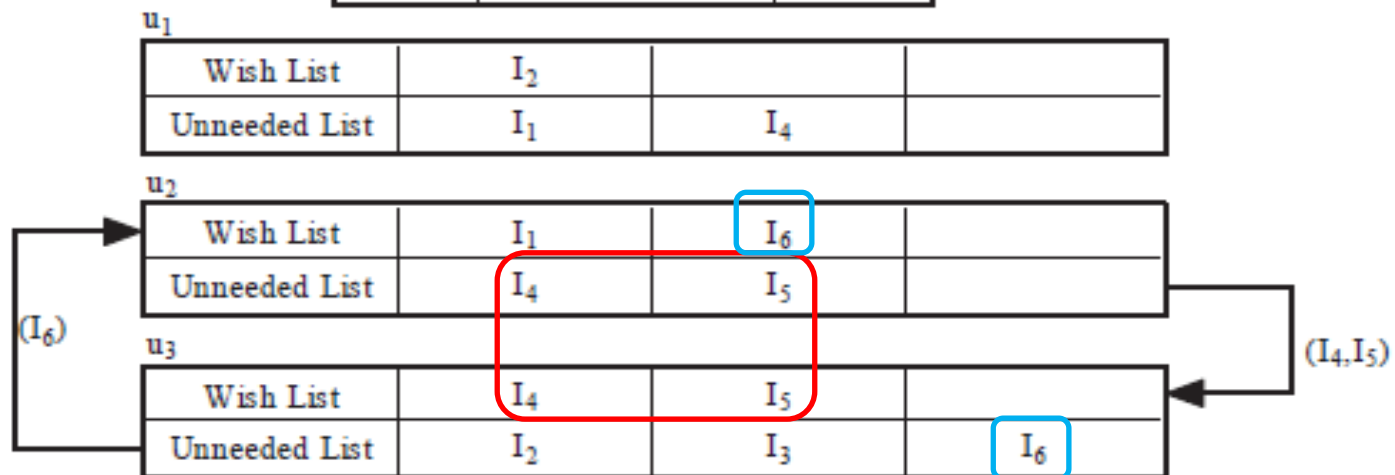


Figure 2: Example of transaction in BVEM

BVEM

- n felhasználó: $U = \{u_1, \dots, u_n\}$
- m tárgy: $O = \{I_1, \dots, I_m\}$
 - v_j : I_j ára
- 2 lista minden felhasználónak:
 - L_i : szükségtelen
 - W_i : szükséges („vágyott”)

BVEM

- $O' \subseteq O$
- $V(O')$: O' -beli tárgyak összértéke
- $E = (u_i, u_l, S_i, S_l)$: csere u_i és u_l között
($S_i \subseteq L_i$ és $S_l \subseteq L_l$)
- $G(E, u_i) = V(S_l)$: u_i nyeresége (a kapott elemek összértéke)

Megfelelő cserepár

- $E = (u_i, u_l, S_i, S_l)$ csere megfelelő, ha
 - u_i -nek szüksége van minden u_l által felajánlott elemre és fordítva
$$(S_i \subseteq W_l \text{ és } S_l \subseteq W_i)$$
 - a 2 felhasználó haszna közötti eltérés egy határon belül van
$$\beta V(S_i) \leq V(S_l) \leq \beta^{-1} V(S_i), \text{ ahol } 0 < \beta \leq 1$$

Maximális hasznú cserepár

- Tfh. mindenki maximalizálni akarja a hasznát
- E csere pár maximális, ha minden más E' cserére:
 $G(E', u_i) \leq G(E, u_i)$ és $G(E', u_l) \leq G(E, u_l)$

Top-K Cserre Ajánlás

- Top(k,i):
 u_i felhasználó k legértékesebb (maximális hasznú) $E^*(u_i, u_l)$ cserepárja k különböző felhasználóval
- Probléma: L_i vagy W_i lista módosításakor Top(k,j) listák frissítése minden u_j felhasználónál

Két felhasználó közötti csere

- u_i és u_l között a maximális hasznú cserepár meghatározása
- Top-1 két felhasználó között (TIU2)
- Bonyolultság: exponenciálisan nő a listák (L_i és W_i) növekedésével

Két felhasználó közötti csere

Algorithm 1 Brute-force algorithm for T1U2
exchange(L_i, W_i, L_l, W_l)

- 1: Clear optimal solutions S^*
 - 2: Generate subsets $\phi_L = 2^{L_i \cap W_l}$ and sort on value
 - 3: Generate subsets $\phi_R = 2^{L_l \cap W_i}$ and sort on value
 - 4: Set $m = |\phi_R|$
 - 5: **for** n from $|\phi_L|$ to 1 **do**
 - 6: **while** $m > 0$ and $\beta * |\phi_R[m]| > |\phi_L[n]|$ **do**
 - 7: $m = m - 1$
 - 8: **end while**
 - 9: **if** $\phi_L[n]$ and $\phi_R[m]$ is an eligible exchange **then**
 - 10: $S^* = (u_i, u_l, \phi_L[n], \phi_R[m])$ if $V(\phi_L[n]) \geq G(S^*, u_i)$
 and $V(\phi_R[m]) \geq G(S^*, u_l)$
 - 11: **end if**
 - 12: **end for**
 - 13: Return S^*
-

Két felhasználó közötti csere

- S^* optimális (maximális) cserepár
- $2^{L_i \cap W_i}$ és $2^{W_i \cap L_i}$ részhalmazok kiszámítása
- Minden részhalmaz párt megvizsgál és kiválasztja a maximális hasznút az S^* -ba
- Futási idő: $O(|S_i|2^{|S_i|} + |S_l|2^{|S_l|})$

ε -közelítő módszer

- A haszon nem rosszabb E^* -nál $1-\varepsilon$ -nal, ahol E^* maximális hasznú cserepár
- A brute-force TIU2 a részhalmazok kiszámítása miatt költséges



- Ötlet: csak néhány kombináció közelítő értékekkel γ (O')

ε -közelítő módszer

- $f(x)$ kerekítő függvény

v_{max}, v_{min} : maximális és minimális érték
bármilyen nem üres tárgy

kombinációnak

ε : hibatűrés paramétere

N : tárgyak maximális száma

- $$f(O') = \left\lceil \frac{\log v_{min} - \log V(O')}{\log(1 - \frac{\varepsilon}{N})} \right\rceil$$

ε -közelítő módszer

	érték	lbi	lb	ubi	ub
AVT[0]	0	0	0	0	0

$$v_1 = 2, v_2 = 2, v_3 = 3$$

m az a minimális egész, melyre $v_{\min}(1 - \frac{\varepsilon}{N})^{-1}$

$$(1 - \varepsilon/N)^{-1} = 2 \text{ és } v_{\min} = 1$$

ϵ -közelítő módszer

	érték	lbi	lb	ubi	ub
AVT[0]	0	0	0	0	0
AVT[1]	2	{11}	2	{11}	2

ϵ -közelítő módszer

	érték	lbi	lb	ubi	ub
AVT[0]	0	0	0	0	0
AVT[1]	2	{11}	2	{11}	2
AVT[2]	4	{11,12}	4	{11,12}	4

ϵ -közelítő módszer

	érték	lbi	lb	ubi	ub
AVT[0]	0	0	0	0	0
AVT[1]	2	{11}	2	{11}	2
AVT[2]	4	{11,12} {13}	4-3	{11,12}	4

ϵ -közelítő módszer

	érték	lbi	Lb	ubi	ub
AVT[0]	0	0	0	0	0
AVT[1]	2	{11}	2	{11}	2
AVT[2]	4	{11,12} {13}	4 -3	{11,12}	4
AVT[3]	8	{11,13}	5	{11,13}	5

ϵ -közelítő módszer

	érték	lbi	lb	ubi	ub
AVT[0]	0	0	0	0	0
AVT[1]	2	{11}	2	{11}	2
AVT[2]	4	{11,12} {13}	4 3	{11,12}	4
AVT[3]	8	{11,13}	5	{11,13} {11,12,13}	5 7

Cserepár keresés AVT-n

- Algoritmus: hogyan találhatunk ε -közelítő cserepárokat u_i és u_l számára egyszerre
- *Lineáris időben adja vissza az ε -közelítő optimális Top-K cserepárokat bármely két felhasználó között*

(az algoritmus komplexitása $O(N^2)$, de csökkenthetjük $O(N)$ -re ha minden bejegyzést a közelítő érték csökkenő sorrendjében és a szkennelő bejegyzéseket fentről-lefelé rendezzük)

Cserepár keresés AVT-n

- AVT_1 $W_i \cap L_l$ -en és AVT_2 $L_i \cap W_l$ -en
- Minden $AVT_1[m] \in AVT_1$ és $AVT_2[n] \in AVT_2$ párra:
 - ha $\beta \leq \frac{AVT_1[m].ub}{AVT_2[n].lb} \leq (1 - \beta)$ és $\beta \leq \frac{AVT_2[n].ub}{AVT_1[m].lb} \leq (1 - \beta)$:
generálja le a cserepárokat u_i -nek és u_i -nek

Általános Top-K csere

- Minden egyes u_i, u_l user párra két közelítő értéktábla AVT_{il} és AVT_{li} és frissül $W_i \cap L_l$ és $L_i \cap W_l$ szerint
- Folyamatos frissítés nem működik
 - Túl nagy memória igény: $|U|(|U| - 1)$ közelítő értéktáblázat

Általános Top-K csere

- Dinamikus frissítés helyett könnyű súlyú indexek
 - u_i listájának módosítása esetén csak az érintett u_l felhasználók Top-K listáinak frissítése
- ⇓
- Kritikus tárgy halmaz meghatározásával

Kritikus tárgy halmaz

- Adott u_i felhasználó W_i tárgy listája esetén a tárgyak $O' \subseteq W_i$ alcsoportja egy kritikus alcsoportot alkot, ha $V(W_i) - V(O') < G(u_i, \text{Top}(k, i))$,

azaz

O' alcsoport kritikus W_i szempontjából, ha a többi W_i -ben lévő elem összértéke nem nagyobb mint u_i jelenlegi optimális haszna

- Jelölés: K_i

Kritikus tárgy halmaz

- Csak akkor kell újra számolni a TIU2 cserét u_i és u_l között a $\text{Top}(k, i)$ –t frissítéséhez, ha u_l legalább egy kritikus elemét tartalmazza u_i –nek és fordítva
- $CL(I_j)$ tartalmazza azokat a felhasználókat, akiknél I_j a kritikus elemek között van és $UL(I_j)$ tartalmazza azokat a felhasználókat, akiknél I_j a szükségtelen elemek között van

Általános Top-K frissítés

- ha van egy frissítés a W_i listán, akkor a rendszer megkap egy lehetséges felhasználó csoportot a fordított listákról és kiszámolja a TIU2 cserét

Tárgy beszúrása

- újraszámolja a TIU2 cseréket
- I_j beillesztése u_i -hez W_i -be:
 - $UL(I_j)$ megadja CU -t:
jelöltek (akik érintettek lehetnek)
 - minden u_l eleme CU -ra:
 - ha u_i -nak van u_l egy kritikus eleme vagy fordítva,
akkor:
Cserepár keresése AVT-n

Tárgy törlése

- Törlés drága lehet
- Top- κ cserepárok megtartása minden u_i user számára ($\kappa > k$)
- Ha a cserepárok egyike $E \in Top(\kappa, l)$ eltávolítódik $I_j \in W_i$ törlése által, a cserelista nem frissül azonnal, hanem új TIU2 u_i és u_l között
- Ha az új optimális csere $Top(\kappa, l)$ marad, akkor beilleszti, egyébként csökkenti a κ -t 1-el
- $Top(\kappa, l)$ kiszámítását elhalasztja addig, amíg u_l új elemet szűr be vagy $\kappa < k$

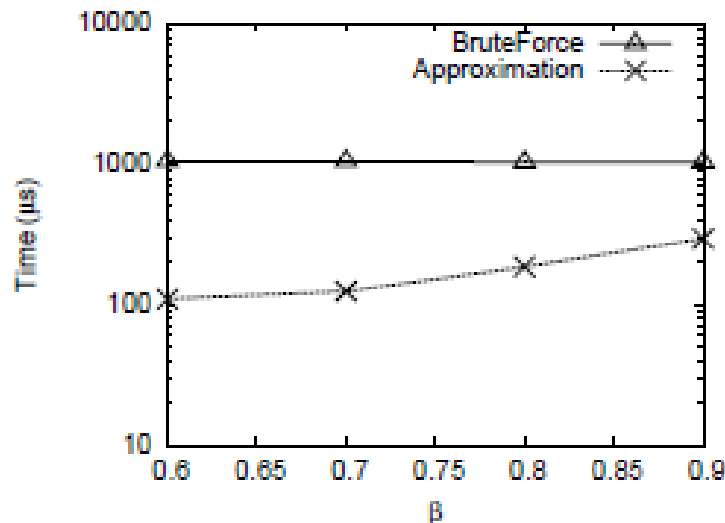
Tesztelés

- Szintetikus adatokon
- Valódi adatokon
 - eBay.com
 - 90 napig
 - 34191 felhasználó
 - 452774 tárgy
 - 1094152 tranzakció

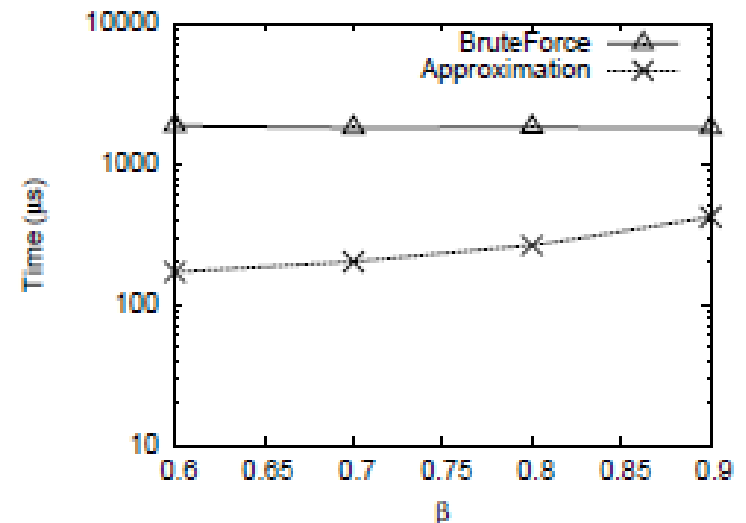
Tesztelés

- Kívánt és szükségtelen listák a vásárolt és eladott elemekből
- Kicsit alakítottak rajta, eredmény:
2458 felhasználó és 2769 tárgy

Tesztelés



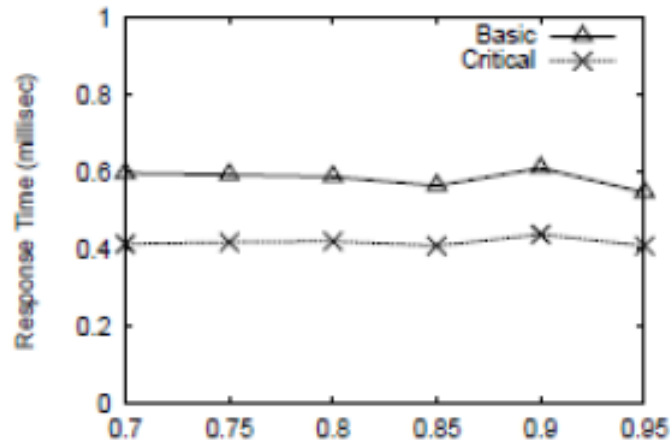
(a) Running time on exponential price distribution



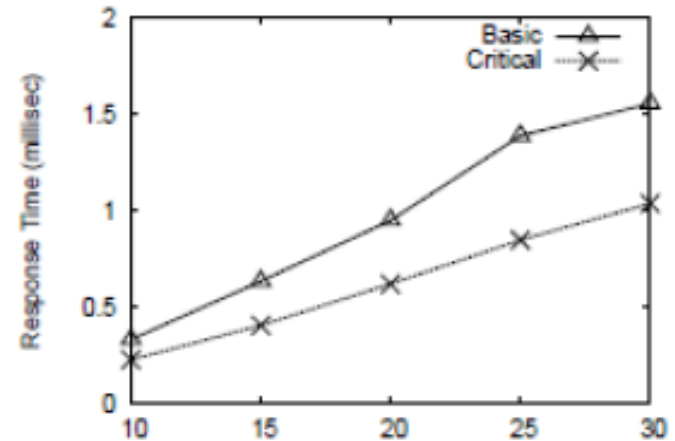
(b) Running time on Zipf price distribution

Figure 8: Impact of varying β on running time

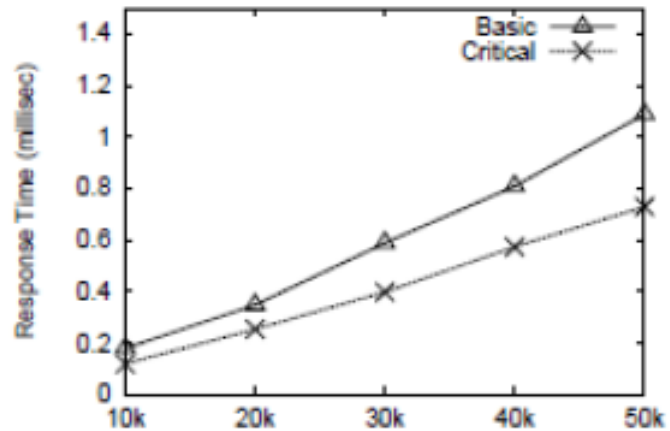
Tesztelés



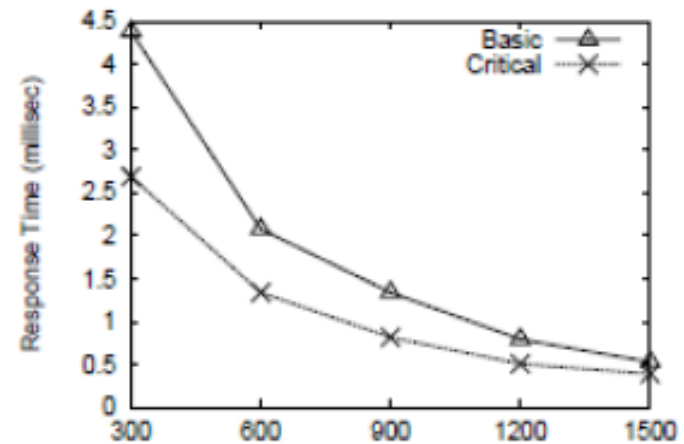
(c) Effect of relaxation factor β



(d) Effect of item list length N



(e) Effect of user number $|U|$



(f) Effect of total item number



Köszönjük a figyelmet!