

BitTorrent felhasználók értékeléseinek következtetése a viselkedésük alapján

Hegedűs István

Ajánló rendszerek

- Napjainkban egyre népszerűbb az ajánló rendszerek alkalmazása
- A cégeket is hasznos információval látja el
- Amazon.com
 - Felhasználók értékelik a megvett/megrendelt termékeket
 - Értékelések alapján ajánlat tétele
 - Keresések felgyorsítására
 - Üzleti modell készítése
 - Ajánló rendszerek használata

Ajánló rendszerek (2)

- Adott:
 - Felhasználók halmaza
 - Termékek halmaza
 - Valamit a felhasználók bizonyos termékekre vonatkozó értékelései (mátrix)
- Különféle módszerek az ajánlásra:
 - Collaborative filtering
 - Mátrix faktorizáció
 - Propagációs algoritmusok
 - Gépi tanuló módszerek

Probléma

- Ha adottak az értékelések → megfelelő algoritmus alkalmazása ✓
- Ha NEM adottak az értékelések???
 - Felhasználók nem szeretnek információt közzé tenni a weben
 - Nincs lehetőségük az értékelésekre
 - „régí” az adatbázis (nincs benne a megfelelő információ)
 - A felhasználó nem foglalkozik az értékeléssel
 - Erősen torzított (random, csak rossz érték)

Értékelések következtetése

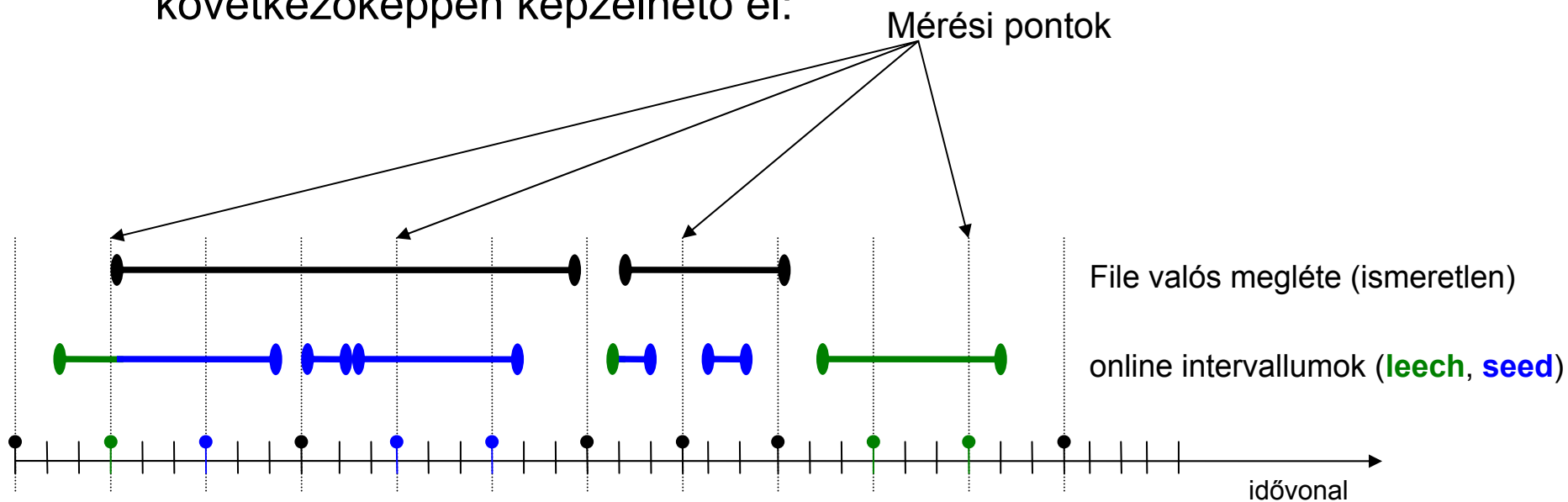
- Nem adott kellő információ
- Esetleg értékelések pontosítása
- Egy BitTorrent tracker-ből kinyert információkat használtunk
- Próbáljuk meg valahogy „kitalálni” a felhasználó tevékenysége alapján, a letöltött file tetszett-e vagy nem
- Cél: A file megléte alapján következtetni a file értékelésére (szereti/nem szereti)

A BitTorrent trace

- Filelist.org privát bittorrent oldalról származik a megfigyelés
- 93 nap, ~5 percenként, 91745 felhasználó, 3068 torrent
- Az adatbázisban lévő információk:
 - Időbélyeg
 - Felhasználó azonosító
 - Torrent (file) azonosító
 - File töltésének állapota (%-ban)

A BitTorrent trace (2)

Egy konkrét felhasználó – file pár a
következésképpen képzelhető el:



Mérési pontok:

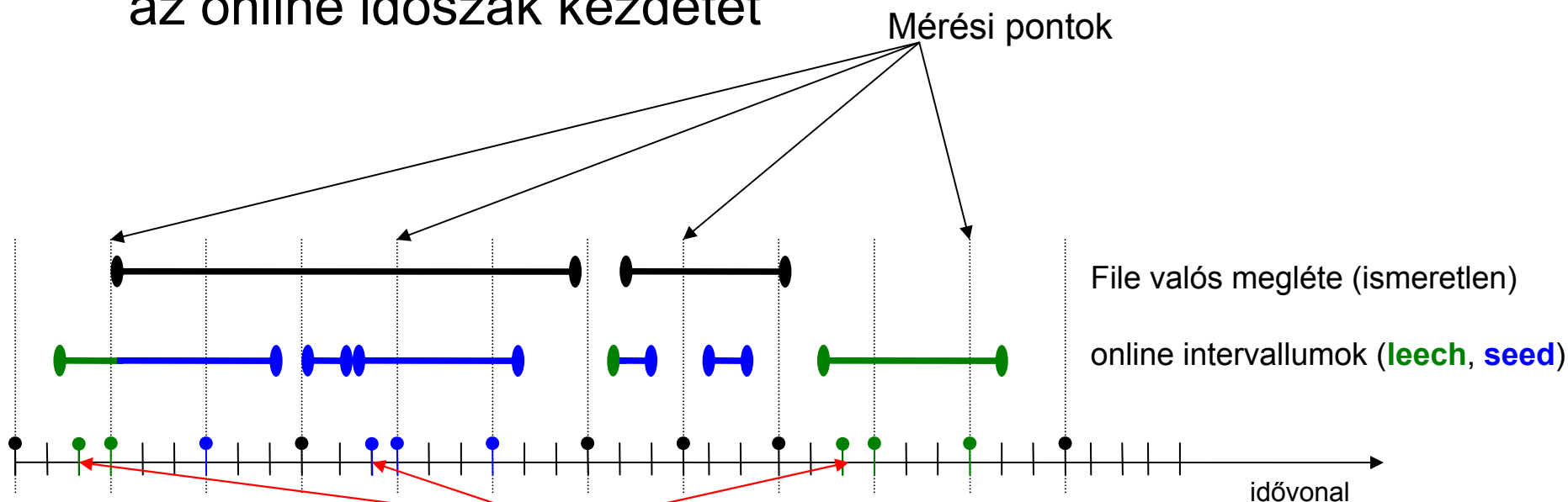
kék = file letöltve

zöld = letöltés < 100%

fekete = nincs információ

Előfeldolgozás – kezdeti időpontok színezése

- Nem ismert a file tényleges megléte, de közelíthető
- Feladat: töltsük ki a definiálatlan időpontokat
- Minden online szakaszban lévő mérési pont tartalmazza az online időszak kezdetét



**A kezdőpontokat ki tudjuk színezni
(1.lépés)**

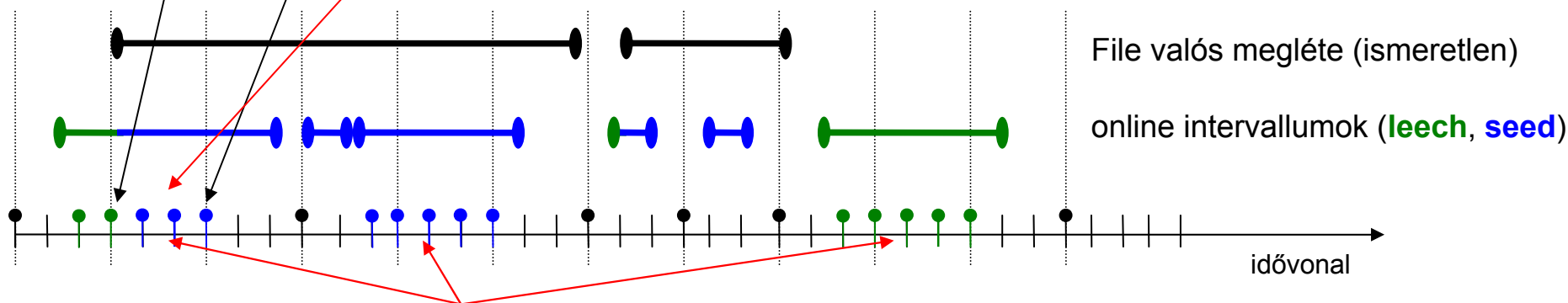
Mérési pontok: **kék** = file letöltve, **zöld** =
letöltés < 100%, **fekete** = nincs információ

Előfeldolgozás – belső időpontok színezése

- Az alábbi heurisztika alkalmazása a belső pontokra:

Online sessions		
front	end	fill
0	0	0
0	1	1
1	0	0
1	1	1

- A fennmaradó pontokat offline szakasznak tekintjük
- Rövid szakaszok elveszhetnek



**Belső pontok színezése
(2. lépés)**

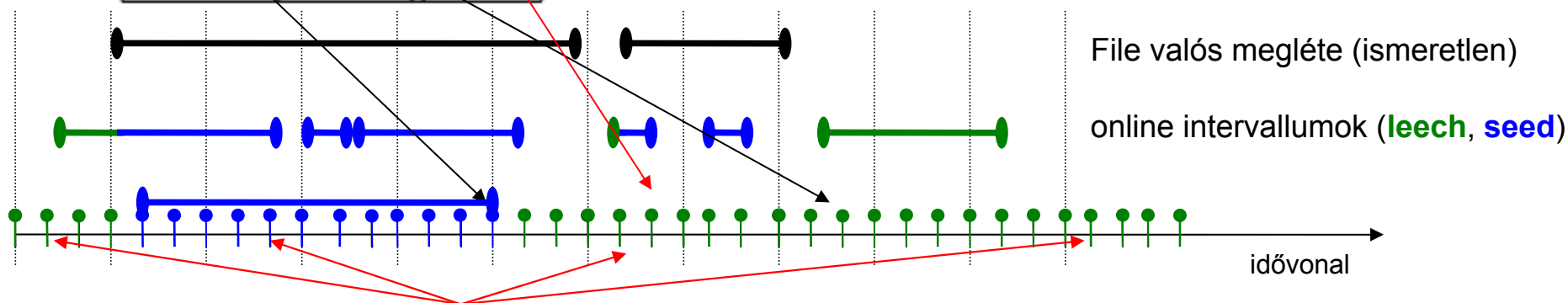
Mérési pontok: **kék** = file letöltve, **zöld** = letöltés < 100%, **fekete** = nincs információ

Előfeldolgozás – fennmaradó időpontok színezése

- A fennmaradó pontok az alábbi heurisztika alapján lett közelítve:

Offline sessions		
front	end	fill
0	0	0
0	1	0
1	0	0
1	1	1

- Ezzel a végső lépéssel közelíthető a letöltött file megléte, minden egyes felhasználó – file párra.



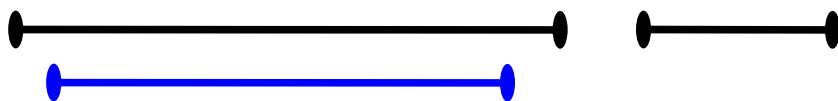
**Fennmaradó pontok színezése
(3. lépés)**

Mérési pontok: **kék** = file letöltve, **zöld** = letöltés < 100%, **fekete** = nincs információ

File tartalmazás közelítése

- Az előfeldolgozó lépések elvégzése után minden egyes felhasználó – file párra tudunk közelítést adni a file meglétére minden egyes „időpillanatban”
- Ezt az információt mátrixok sorozataként képzelhetjük el (minden egyes időponthoz tartozik egy mátrix, mely jelöli melyik felhasználónál melyik file van letöltve)
- A célunk ezen mátrix sorozatok alapján a **tetszik/nem tetszik** (like/dislike) értékek közelítése

	i1	i2	i3
u1	1	0	0
u2	1	0	1
u3	0	1	0

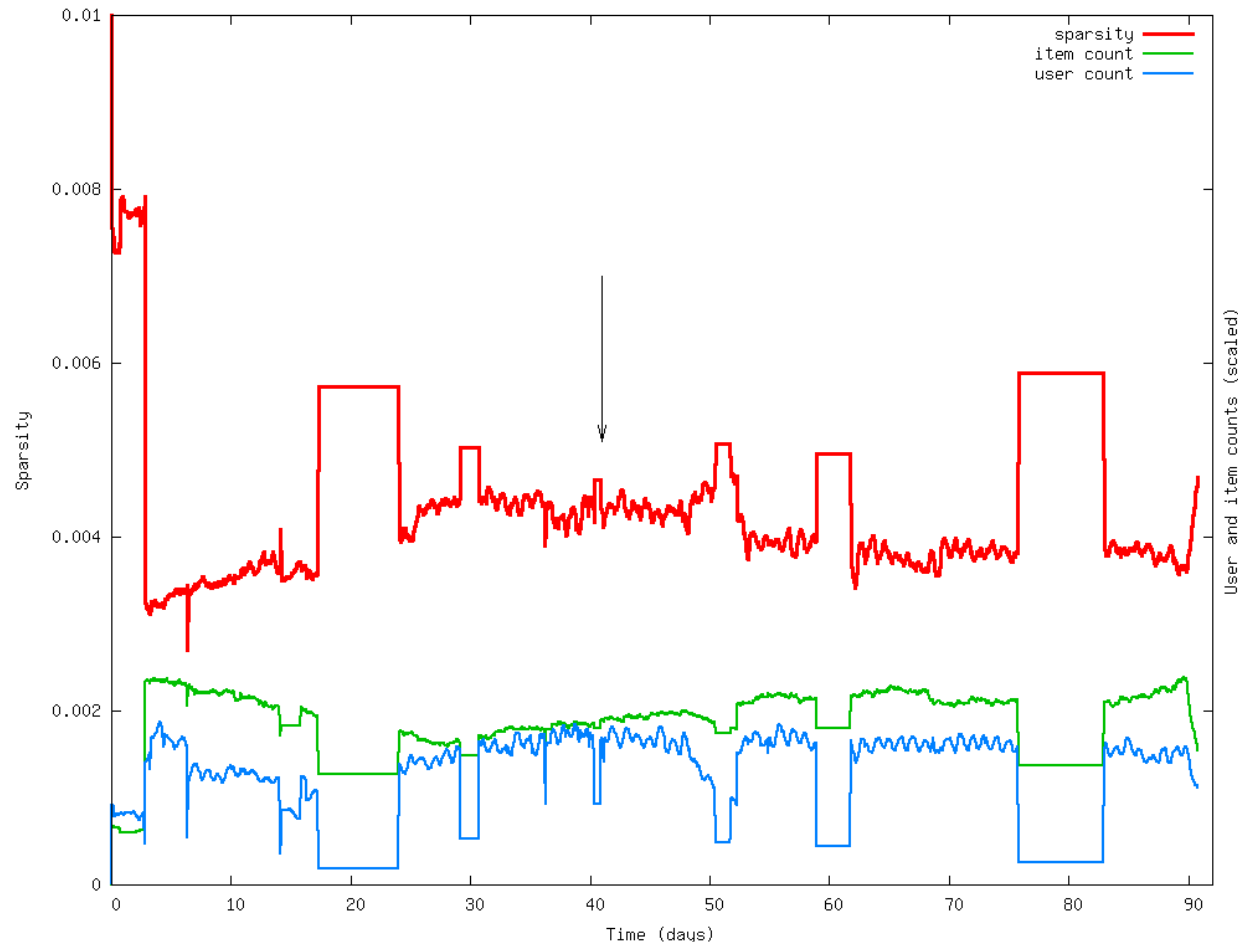


File valós megléte (ismeretlen)

File meglétének közelítése

Az adatbázis ritkasága

- Mielőtt a „tetszés” közelítésére rátérnénk vizsgáljuk meg az adatbázis ritkaságát
 - Ritkaság: egy adott időponthoz tartozó mátrixban lévő 1-esek száma osztva a cellák számával
 - Ajánlások számításának a nehézségét/pontosságát befolyásolja



Értékelések közelítése az előfeldolgozott adatokból

- Előfeldolgozó lépések elvégzése után a feladat a felhasználók értékelésének közelítése a viselkedésük alapján
- Az időpontokhoz tartozó mátrixsorozatok alapján a **tetszik/nem tetszik** értékek becslése
- Két módszert alkalmaztunk:
 - Naiv megközelítés
 - „időcsúsztatásos” megközelítés

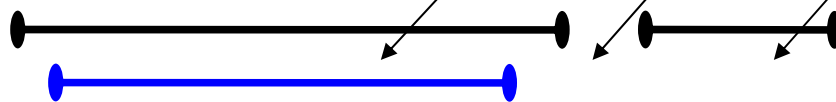
Értékelések közelítése – naiv megközelítés

- Vegyünk egy konkrét időponthoz tartozó mátrixot
 - Ha a felhasználónak le van töltve a file, akkor pozitív értékelés
 - Ha nincs, akkor hiányzó értékelés
- Azaz a naiv megközelítésben nincs negatív értékelés, mivel nincs róla információ
- Ezt a módszert tekintjük baseline-nak

Értékelések közelítése – „időcsúsztatásos” megközelítés

- Vegyünk három időpontot és a hozzá tartozó mátrixokat
- Az időcsúsztatásos eljárás az alábbi heurisztikán alapul:

Dataset labeling			
before	actual	after	inference
0	0	0	unspecified
0	0	1	unspecified
0	1	0	0 (dislike)
0	1	1	1 (like)
1	0	0	0 (dislike)
1	0	1	unspecified
1	1	0	unspecified
1	1	1	1 (like)



File valós megléte (ismeretlen)

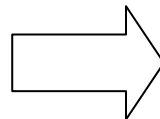
Értékelés közelítése: 0

Értékelések közelítése – „időcsúsztatásos” megközelítés (2)

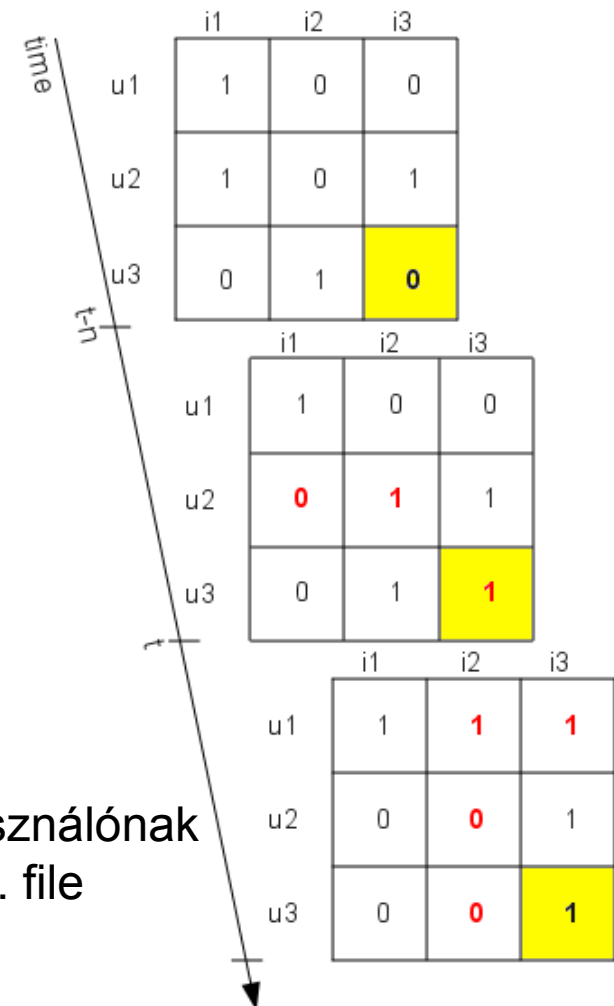
- Egy másik, részletesebb példa:

Dataset labeling			
before	actual	after	inference
0	0	0	unspecified
0	0	1	unspecified
0	1	0	0 (dislike)
0	1	1	1 (like)
1	0	0	0 (dislike)
1	0	1	unspecified
1	1	0	unspecified
1	1	1	1 (like)

- $t-n$ időpontban a 3. felhasználónak nem volt letöltve a 3. file
- t időpontban le volt töltve
- $t+n$ időpontban szintén le volt töltve



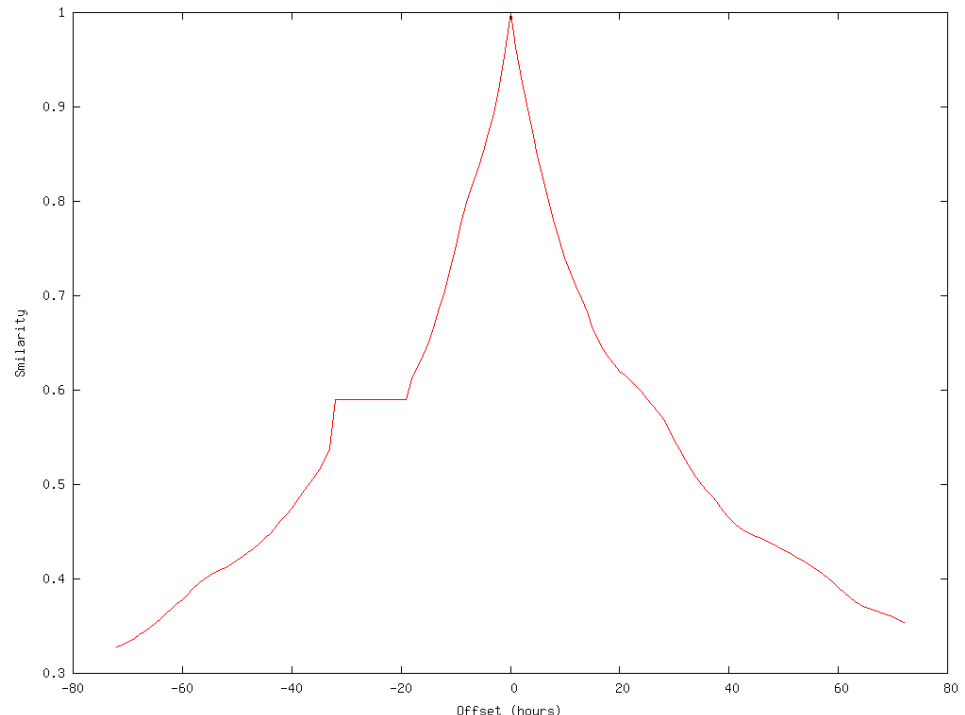
A 3. felhasználónak tetszik a 3. file



Értékelések közelítése – „időcsúsztatásos” megközelítés (3)

- A kérdés: Hogyan válasszuk ki a Δt időt, azaz a 3 konkrét időponthoz tartozó mátrixot?
 - Ha túl kicsi a Δt , nincs nagy változás
 - Ha túl nagy a Δt , elveszhet információ

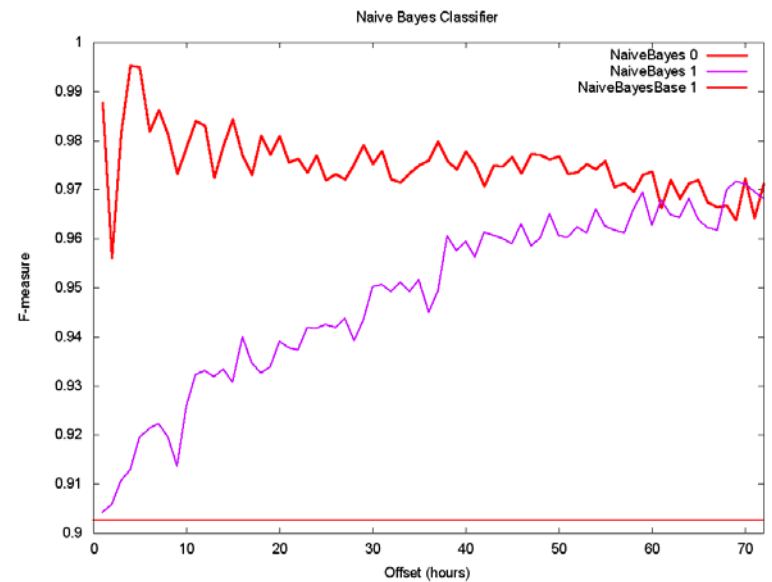
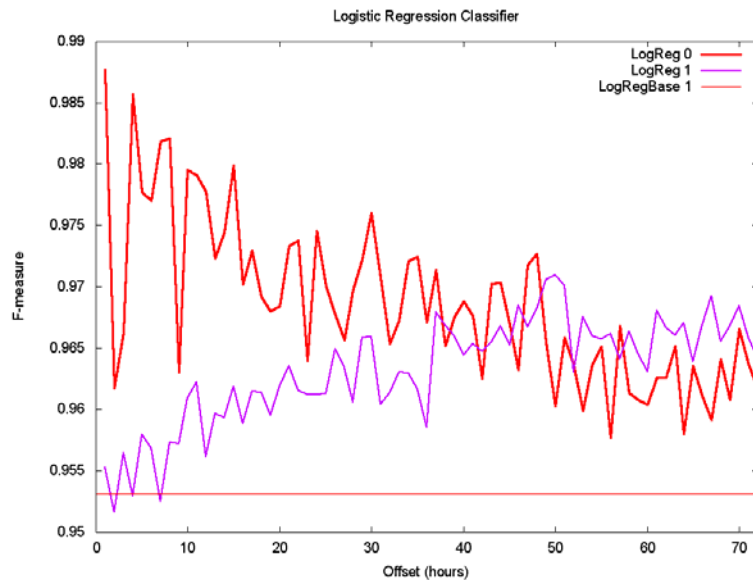
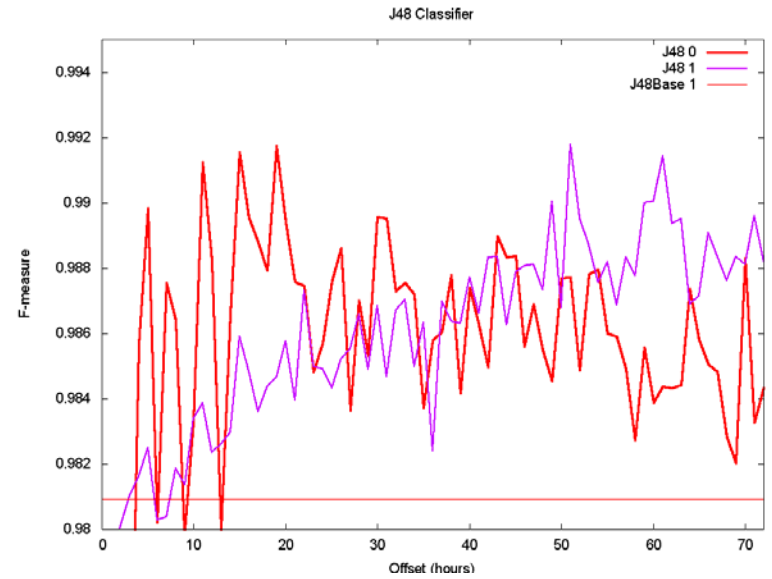
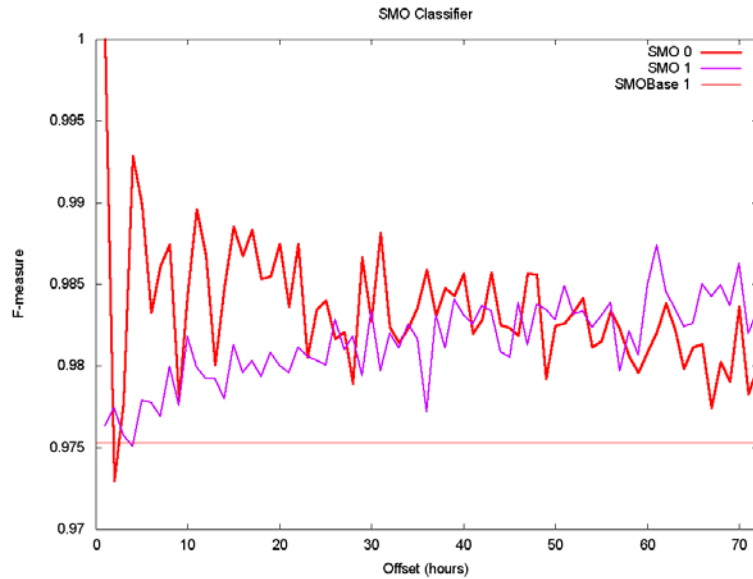
Az ábrán a Δt időkhöz tartozó mátrixok hasonlósága látható



Kiértékelés

- Mivel nem tudjuk az „igazságot” indirekten teszteltük a módszert
- Megvizsgáltuk a különböző címkézések tanulhatóságát (2 osztályos tanítások)
 - „Base 1”: naiv értékelés közelítés, a pozitív és random jelöletlen példák
 - „0”: időcsúsztatásos módszer, negatív és random jelöletlen példák
 - „1”: időcsúsztatásos módszer, pozitív és random jelöletlen példák
- Kiértékelési mérték: F-mérték
- Használt tanuló algoritmusok:
 - SMO
 - J48
 - Logistic Regression
 - Naive Bayes
- Jellemző kinyerés [1]

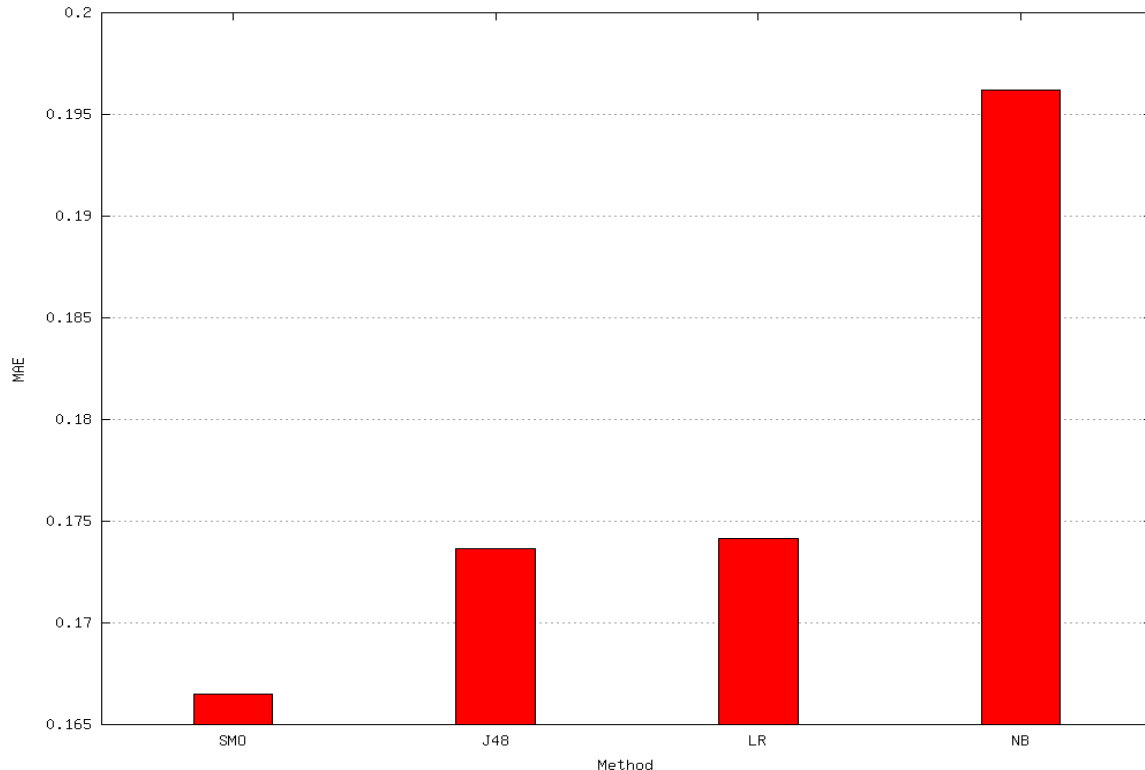
Eredmények



Publikált adatbázis

- A méréseink alapján közzétettünk egy általunk ajánlott ajánló adatbázist
 - Véletlenül kiválasztott időpillanat
 - $\Delta t = 60$ intervallum alapján

- Tanulók teljesítménye az ajánlott adatbázison (MAE)



Konklúzió

- Mutattunk egy módszert az értékelések következtetésére
- Heurisztika alapú, így nem feltétlenül pontos
- Viszont a tanuló algoritmusok eredményességéből valószínűsíthetjük a helyességét
- Készítettünk egy ajánló rendszer adatbázist a felhasználók viselkedése alapján
- Valamint publikáltuk az eredményeinket [2] a COPS 2010 konferencián, ahol megkaptuk a „Best Paper” díjat.

Hivatkozások

- [1] Park Yoon-Joo, Tuzhilin Alexander. The long tail of recommender systems and how to leverage it. Proceedings of the 2008 ACM conference on Recommender systems
- [2] Róbert Ormándi, István Hegedűs, Kornél Csernai, Márk Jelasity. Towards inferring ratings from user behavior in BitTorrent communities. 2010 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, 2010