

Ajánlórendszerek

Engedy Balázs

Budapesti Műszaki és Gazdaságtudományi Egyetem
Számítástudomány szakirány

„Nagy adathalmazok kezelése”
című tárgy előadása

2010. április 21.

Tartalom

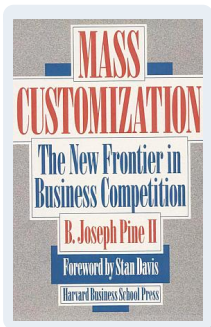
- 1 **Bevezetés**
 - Történelem
 - Gazdasági előnyök
 - Formális leírás
 - Csoportosítás
- 2 **Tartalomalapú rendszerek**
 - Áttekintés
 - Dokumentumprofil készítése
 - Felhasználói profil készítése
 - Hasznosságbecslés
 - Előnyök és hátrányok
- 3 **Kollaboratív rendszerek**
 - Áttekintés
 - Felhasználók hasonlósága
 - Hasonló ízlésű felhasználók
 - Értékelések aggregálása
 - Előnyök és hátrányok
- 4 **Kitekintés**
 - Hibrid rendszerek
 - Gradiensmódszer
 - Faktormátrixok
 - Források, vége!

Dehomogenizálódó piacok

Joseph Pine: “*Mass Customization*” (1993)

A '90-es években végbemenő piaci változásokat tárgyalja, miszerint:

- a szabványos, egyenvásárlóra tervezett *tömegtermékek* ideje lejárt,
- többféle vásárló többféle igényét kielégítő *heterogén (sokféle) termékek* gyártásának irányába kell elmozdulni.



Ajánlórendszerek

Jeff Bezos (CEO, Amazon)

„Ha 2 millió vásárlóm van a Weben, akkor 2 millió boltom kell hogy legyen a Weben!”

- A heterogenitás dilemmája
 - Megvan a sokféle termék, változatos kínálat
 - De így a vásárlónak rengeteg lehetőségből kell *választani*
 - Információs túlterhelés ⇒ ☹️
- A kínálatot meg kell szűrni!
- Egy lehetséges megoldás: *ajánlórendszerek*
 - A vásárlónak csak *releváns* termékeket mutatunk
 - Egyénre- avagy testreszabott vásárlás ⇒ 😊

Ajánlórendszerek

Jeff Bezos (CEO, Amazon)

„Ha 2 millió vásárlóm van a Weben, akkor 2 millió boltom kell hogy legyen a Weben!”

- A heterogenitás dilemmája
 - Megvan a sokféle termék, változatos kínálat
 - De így a vásárlónak rengeteg lehetőségből kell *választani*
 - Információs túlterhelés ⇒ ☹️
- A kínálatot meg kell szűrni!
- Egy lehetséges megoldás: *ajánlórendszerek*
 - A vásárlónak csak *releváns* termékeket mutatunk
 - Egyénre- avagy testreszabott vásárlás ⇒ 😊

Gazdasági előnyök az eladók számára

- 1 Böngészőkből vásárlók → **Vásárlói kosár létrejön**
 - Ha a webshop böngészése közben *érdekes* termékekkel találkozik a vásárló, talán vásárolni is fog!
- 2 Járulékos eladások → **Kosár-átlagmértet növekszik**
 - Ha vásárlás közben további *érdekes* termékekkel találkozik a vásárló, talán azokból is vásárolni fog!
- 3 Hűséges vásárlók → **Vásárlások száma növekszik**
 - Ha a konkurenciához képest nálunk *érdekesebb* termékekkel találkozik a vásárló – mert a régebbi ténykedéseiből épített vásárlói profil szerint ajánlunk kedvére valót –, talán emiatt már legközelebb is inkább hozzánk tér be vásárolni!

Gazdasági előnyök az eladók számára

- 1 Böngészőkből vásárlók → **Vásárlói kosár létrejön**
 - Ha a webshop böngészése közben *érdekes* termékekkel találkozik a vásárló, talán vásárolni is fog!
- 2 Járulékos eladások → **Kosár-átlagmértet növekszik**
 - Ha vásárlás közben további *érdekes* termékekkel találkozik a vásárló, talán azokból is vásárolni fog!
- 3 Hűséges vásárlók → **Vásárlások száma növekszik**
 - Ha a konkurenciához képest nálunk *érdekesebb* termékekkel találkozik a vásárló – mert a régebbi ténykedéseiből épített vásárlói profil szerint ajánlunk kedvére valót –, talán emiatt már legközelebb is inkább hozzánk tér be vásárolni!

Gazdasági előnyök az eladók számára

- 1 Böngészőkből vásárlók → **Vásárlói kosár létrejön**
 - Ha a webshop böngészése közben *érdekes* termékekkel találkozik a vásárló, talán vásárolni is fog!
- 2 Járulékos eladások → **Kosár-átlagméret növekszik**
 - Ha vásárlás közben további *érdekes* termékekkel találkozik a vásárló, talán azokból is vásárolni fog!
- 3 Hűséges vásárlók → **Vásárlások száma növekszik**
 - Ha a konkurenciához képest nálunk *érdekesebb* termékekkel találkozik a vásárló – mert a régebbi ténykedéseiből épített vásárlói profil szerint ajánlunk kedvére valót –, talán emiatt már legközelebb is inkább hozzánk tér be vásárolni!

Gazdasági előnyök az eladók számára

- 1 Böngészőkből vásárlók → **Vásárlói kosár létrejön**
 - Ha a webshop böngészése közben *érdekes* termékekkel találkozik a vásárló, talán vásárolni is fog!
- 2 Járulékos eladások → **Kosár-átlagméret növekszik**
 - Ha vásárlás közben további *érdekes* termékekkel találkozik a vásárló, talán azokból is vásárolni fog!
- 3 Hűséges vásárlók → **Vásárlások száma növekszik**
 - Ha a konkurenciához képest nálunk *érdekesebb* termékekkel találkozik a vásárló – mert a régebbi ténykedéseiből épített vásárlói profil szerint ajánlunk kedvére valót –, talán emiatt már legközelebb is inkább hozzánk tér be vásárolni!

Az ajánlási feladat formális leírása

Input

- C vásárlóhalmaz, $|C| = m$
- S termékhalmoz, $|S| = n$
- $u : C \times S \rightarrow \mathbb{R}$ hasznosságfüggvény (pl. rating 1–10-ig)
 - ☹ Kevés függvényérték ismert (pl. amely filmeket értékelte)
 - ☹ Az m és n pedig nagyon nagy (milliók)

Output

- $\forall c \in C$ vásárló számára a legérdekesebb/leghasznosabb új termék(ek) megadása, azaz:

$$s_c \in S, \text{ hogy } u(c, s_c) \text{ maximális}$$

Ajánlórendszerek csoportosítása

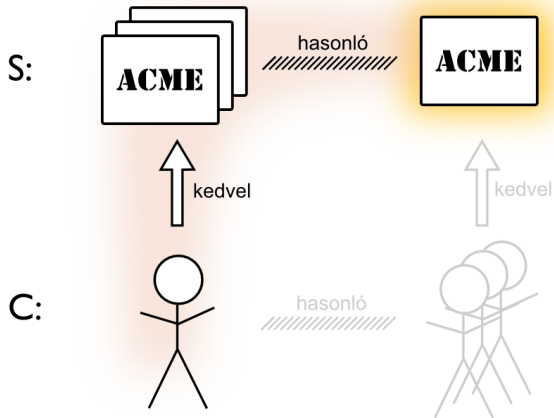
Az ajánlandó termékekhez eljuthatunk a kérdéses felhasználó által kedvelt termékeken keresztül, azokhoz hasonlót keresve; vagy a kérdéses felhasználóhoz hasonló ízlésű felhasználókon keresztül, azok által közösen kedvelt terméket keresve. Így tehát:

Tartalomalapú ajánlás A felhasználónak olyan termékeket ajánlunk, melyek tartalmukban hasonlóak azokhoz, amelyeket a múltban magasra értékelt.

Kollaboratív ajánlás A felhasználónak olyan termékeket ajánlunk, melyeket más, hasonló ízlésű felhasználók magasra értékelték.

Hibrid módszerek A fenti két módszert együttesen alkalmazzuk, vagy nem is különítjük el kategorikusan a két irányt.

Tartalomalapú ajánlórendszerek



Tartalomalapú ajánlórendszerek

- Egy adott felhasználó számára egy termék hasznosságát a hasonló termékek ugyanazon felhasználó által megadott hasznosságaiból becsüljük¹:

$$\hat{u}(c, s) \leftarrow \forall u(c, s_i) \text{ alapján, ahol: } s \sim s_i$$

- Klasszikusan szöveges dokumentumokra alkalmazzuk
 - A dokumentumokat tartalmilag pl. kulcsszavak gyakoriságával, fontosságával jellemezhetjük, a továbbiakban e skalárjellemzőkből álló ennes képviseli a dokumentumot
 - Két dokumentum hasonló, ha tartalmuk, így jellemző kulcsszavaik hasonlóak
 - Feltételezzük, hogy tartalmában a már olvasott (és kedvelt) dokumentumokhoz hasonlóra vágyik a felhasználó

¹és a legnagyobb becsltet ajánljuk

Az ajánlás menete

- 1 Az egyes dokumentumokhoz tartalmi profilt készítünk:

$$\forall s \in \mathcal{S} : s \rightarrow \text{Content}(s)$$

- 2 Az értékelt dokumentumok alapján felhasználói profil készül:

$$\forall c \in \mathcal{C} : \{(Content(s), u(c, s)) \mid s \in \mathcal{S}_c\} \rightarrow \text{ContentBasedProf}(c)$$

- 3 Ezek alapján az ismeretlen hasznosságokat becsüljük:

$$\hat{u}(c, s) = \text{score}(Content(s), \text{ContentBasedProfile}(c))$$

Dokumentumprofil készítése (egy lehetséges heurisztika)

- Legyenek a rendszerben nyilvántartott kulcsszavak:

$$\sigma_1, \sigma_2, \dots, \sigma_k$$

- Az s_i dokumentumban σ_j kulcsszó gyakorisága:

$$f_{i,j}$$

- Egy dokumentum profilja legyen az a vektor, mely az összes kulcsszavak $w_{i,j}$ *fontosságait* tartalmazza erre a dokumentumra:

$$\text{Content}(s_i) = \vec{w}_{s_i} = (w_{i,1}, w_{i,2}, \dots, w_{i,k})$$

Dokumentumprofil készítése (egy lehetséges heurisztika)

- Egy kulcsszó fontossága sokféleképp definiálható, de pl. mondhatjuk, hogy arányos a normalizált előfordulási gyakoriságával:

$$TF_{i,j} = \frac{f_{i,j}}{\max_z f_{i,z}}$$

- A túl sok dokumentumban szereplő kulcsszavak viszont nem túl informatívak, ezek súlyát hivatott csökkenteni az *inverse document-frequency* tényező:

$$IDF_j = \log \frac{n}{n_j}, \text{ ahol } n_j = \left| \left\{ i \mid f_{i,j} > 0 \right\} \right|$$

- Így a fontosság:

$$w_{i,j} = TF_{i,j} \cdot IDF_j$$

Felhasználói profil készítése (egy lehetséges heurisztika)

- A felhasználó profilja lehet pl. szintén egy vektor, mely a múltban elolvasott dokumentumok alapján az adott felhasználó egyes kulcsszavakra vonatkozó $v_{i,j}$ preferenciáit tartalmazza:

$$ContentBasedProfile(c_i) = \vec{v}_{c_i} = (v_{i,1}, v_{i,2}, \dots, v_{i,k})$$

A preferenciák meghatározására történhet a már értékelt dokumentumok \vec{w}_{s_i} vektorainak valamilyen átlagolásával.

- Így a \vec{v} preferenciavektor szintén tekinthető egy TF-IDF vektornak

Hasznosságbecslés (egy lehetséges heurisztika)

- A hasznosságot pedig definiálhatjuk pl. a \vec{v} és \vec{w} vektorok által bezárt szög koszinuszaként:

$$\begin{aligned} \hat{u}(c_i, s_l) &= \text{score}(\text{Content}(s_l), \text{ContBasedProfile}(c_i)) = \\ &= \cos(\vec{v}_{c_i}, \vec{w}_{s_l}) = \frac{\vec{v}_{c_i} \cdot \vec{w}_{s_l}}{|\vec{v}_{c_i}| \cdot |\vec{w}_{s_l}|} = \frac{\sum_{j=1}^k v_{i,j} \cdot w_{l,j}}{\sqrt{\sum_{j=1}^k v_{i,j}^2} \cdot \sqrt{\sum_{j=1}^k w_{l,j}^2}} \end{aligned}$$

Hasznosságbecslés (modellalapú módszerek)

- Az imént bemutatott vektoros becslés a *heurisztikus módszerek* családjába tartozik, és természetesen csak egy a sok lehetséges algoritmus közül
- Vannak még ún. *modell-alapú módszerek*
 - *Content* (s) itt is valamilyen statisztika a dokumentumokról
 - *ContentBasedProfile* (c) viszont valamilyen bonyolultabb modell
 - A hasznosságbecslés a modell kimenetén jelenik meg, az adott dokumentum profilját a bemenetére adva:

$$\hat{u}(c_i, s_l) = Model_{c_i}(Content(s_l))$$

Pl. Naiv bayesi osztályozó felépítése a kulcsszógyakoriságokból

Pl. Rosta (*winnow*) algoritmus, neurális hálózatok

Előnyök

- Függetlenség a többi felhasználótól
 - Egyáltalán nem használjuk a becsléshez a többi felhasználóról tárolt információkat, csak a kérdéses felhasználó preferenciáit
 - Így akár egyetlen felhasználó esetén is életképes a rendszer
- A hasznosságfüggvény opcionális
 - A felhasználói profil akár egy u hasznosságfüggvény nélkül létrehozható
 - A rendszer ekkor is életképes, olyan értelemben, hogy a már megtekintett termékekhez hasonlókat akkor is tud ajánlani

Hátrányok

- Termékjellemzők előállításának problémája
 - Szöveges dokumentumok esetén jól automatizálható
 - Más termékek (pl. filmek, képek) esetén viszont manuálisan kéne megadni (műfaj, szereplők. . .) → sok munka ☹
 - Megint más termékeknél esetleg egyáltalán nem tudunk olyan (skalár) jellemzőket megadni, melyen keresztül értelmesen összehasonlíthatnánk őket
- Megkülönböztethetlenség problémája
 - Ha két termék jellemzői azonosak (a két ennes megegyezik), a rendszer számára megkülönböztethetetlenek

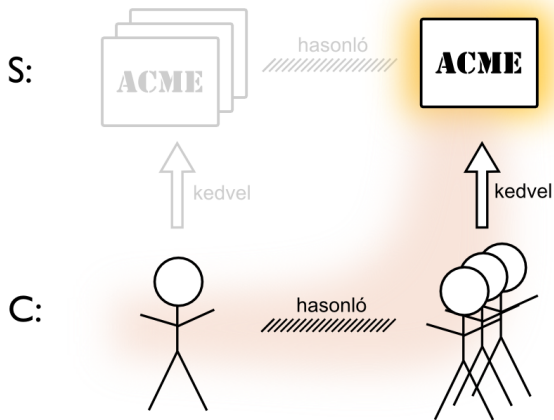
Pl. Egy jól és egy rosszul megírt cikk ugyanabban a témában

Pl. Egy jó és egy rossz 22 collos, TN-paneles,
1680 × 1050 felbontású LCD-monitor

Hátrányok

- Túlspecializáció
 - Elképzelhető, hogy nem mindig a már (magasan) értékeltékhez leginkább hasonló termékek ajánlása lenne az ideális
 - Megoldás: mesterségesen zajt kell az eredményhez keverni
- Pl. Étteremajánló-rendszer: amennyiben a felhasználó még csak kínai és magyar ízekeket kóstolt, soha nem fogunk neki görög éttermet ajánlani (még ha az is a legjobb a városban), hiszen túlságosan különböző az eddig értékelt „termékektől”
- Új felhasználó problémája
 - Egy új felhasználónak egy darabig nem tudunk értelmesen ajánlani, hiszen túlságosan kevés értékelt termékből kéne a profilját felépítenünk

Kollaboratív ajánlórendszerek



Kollaboratív ajánlórendszerek

- A továbbiakban egy termék (pl. film) hasznosságán mindig egy adott skálán mozgó értékelést (*rating*) értünk, azaz, hogy egy felhasználó milyen magasra pontozta, értékelte (volna) az adott terméket. Ennek megfelelően a következő jelölést vezetjük be:

$$r_{c,s} \equiv u(c, s)$$

- Egy adott felhasználó számára egy adott (még nem értékelt) termék értékelését most a hasonló ízlésű felhasználók által a termékre adott értékelések összesítéséből becsüljük²:

$$\hat{r}_{c,s} \leftarrow \forall r_{c_i,s} \text{ alapján, ahol: } c \sim c_i$$

²és itt is a legnagyobb becsltet ajánljuk

Az ajánlás menete

- 1 Felhasználópárok hasonlósági faktorának megállapítása:

$$\forall x, y \in \mathcal{C} : \text{sim}(x, y) \text{ meghatározása}$$

- 2 Adott felhasználóhoz hasonló ízlésű felhasználók megkeresése:

$$\forall c \in \mathcal{C} : c \rightarrow \hat{\mathcal{C}} = \{c' \mid c' \in \mathcal{C}, \text{sim}(c, c') \text{ nagy}\}$$

- 3 A hasonló ízlésű felhasználók által adott értékek aggregálásával az ismeretlen értékelés becslése:

$$\hat{r}_{c,s} = \text{aggr}_{\forall c' \in \hat{\mathcal{C}}} (r_{c',s})$$

Felhasználópárok hasonlósága

- Általában a felhasználók által közösen értékelt S_{xy} termékhalmoz alapján dolgozunk:

$$S_{xy} = \{s \in \mathcal{S} \mid r_{x,s} \neq \emptyset, r_{y,s} \neq \emptyset\}$$

- A mindkettejük által értékelt termékek pontszámait egy-egy \vec{x} és \vec{y} vektorként tekintve itt is definiálhatjuk a hasonlóságot az ezek által bezárt szög koszinuszaként:

$$\text{sim}(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|} = \frac{\sum_{s \in S_{xy}} r_{x,s} \cdot r_{y,s}}{\sqrt{\sum_{s \in S_{xy}} r_{x,s}^2} \cdot \sqrt{\sum_{s \in S_{xy}} r_{y,s}^2}}$$

Felhasználópárok hasonlósága

- Egyes felhasználók szigorúbban pontozhatnak, mint mások, ezt hivatott orvosolni a *Pearson-féle korrelációs együttható*:

$$\text{sim}(x, y) = \cos(\vec{x}_n, \vec{y}_n) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x) \cdot (r_{y,s} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{x,s} - \bar{r}_x)^2} \cdot \sqrt{\sum_{s \in S_{xy}} (r_{y,s} - \bar{r}_y)^2}}$$

ahol \vec{x}_n, \vec{y}_n az \vec{x}, \vec{y} vektorok nulla-átlagúra normálva, azaz:

$$\begin{aligned}\vec{x}_n &= \vec{x} - (\bar{r}_x, \dots, \bar{r}_x) \\ \vec{y}_n &= \vec{y} - (\bar{r}_y, \dots, \bar{r}_y)\end{aligned}$$

ahol \bar{r}_x, \bar{r}_y az x, y felhasználók értékeléseinek átlagai

Hasonló ízlésű felhasználók megkeresése

- Adott egy c felhasználó, a hozzá leginkább hasonló ízlésű néhány másikat szeretnénk megtalálni
- Kiszámíthatjuk minden x, y felhasználópárra $sim(x, y)$ -t, majd rendezzük \rightarrow lassú lehet nagy adathalmazok esetén, mivel az adatok nem férnek el a operatív memóriában
- Gráfot építhetünk a felhasználókból, ahol a hasonló ízlésűek valamilyen szempontból „közel” helyezkednek el egymáshoz
- A következő fejezetben látni fogunk egy módszert, mellyel a $sim(x, y)$ értékeket csak közelítjük, de így legalább elférünk a memóriában

Értékelések aggregálása

- Valamilyen átlag, például:
 - (a) Egyszerű átlag

$$\hat{r}_{c,s} = \frac{1}{N} \sum_{\forall c' \in \hat{C}} r_{c',s}, \text{ ahol } N = |\hat{C}|$$

- (b) Felhasználók hasonlóságával súlyozott átlag

$$\hat{r}_{c,s} = k \cdot \sum_{\forall c' \in \hat{C}} \text{sim}(c, c') \cdot r_{c',s}, \text{ ahol } k = \frac{1}{\sum_{\forall c' \in \hat{C}} |\text{sim}(c, c')|}$$

- (c) Súlyozott átlag, normalizált értékelésekkel

$$\hat{r}_{c,s} = \bar{r}_c + k \cdot \sum_{\forall c' \in \hat{C}} \text{sim}(c, c') \cdot (r_{c',s} - \bar{r}_{c'})$$

Modellalapú módszerek

- Az imént bemutatott becslés szintén a *heurisztikus módszerek* családjába tartozik
- Kollaboratív esetben is vannak *modell-alapú módszerek*
 - Az értékelésekből egy bonyolultabb modellt építünk
 - A modell kimenetén becsli $r_{c,s}$ nem ismert értékeit
- Pl. Klaszterezésen alapuló módszerek: a hasonló ízlésű felhasználókat egy-egy osztályba soroljuk, ezen belül naiv bayesi osztályozót használunk a becslésre
- Pl. Bayes-hálók: a csomópontok a termékek, lehetséges állapotaik az egyes értékelések (pl. 1–10-ig)
- Pl. Egyéb gépi tanulási módszerek

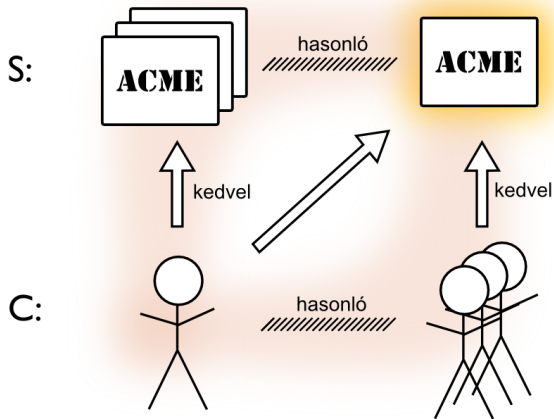
Előnyök

- Tetszőleges típusú termékek
 - A felhasználóink értékrendjük szerint lényegében bármilyen típusú terméket képesek pontozni, így mindenféle termékhez rendelhető egy skalár (a hasznosság)
 - A termékeknek klasszikus értelemben nem is kell összehasonlíthatónak lenniük
- Minden megkülönböztethető
 - A felhasználók értékeléskor a teljes tartalmat nézik, így nem lehet olyan probléma, mint a tartalomalapú esetben, hogy a kivont jellemzők megegyeznek, így egy jó és egy rossz filmet ne tudnánk megkülönböztetni

Hátrányok

- Új felhasználó problémája (mint előbb)
- Új termék problémája
 - Amíg egy termékre nem érkezik elég sok értékelés, nem fogjuk tudni ajánlani (sokáig nem lesz olyan felhasználó lesz, aki hasonló ízlésű a kérdéses felhasználóhoz és értékelte)
 - Megoldás: hibrid módszerek
- Ritkaság
 - Általában csak kevés $r_{c,s}$ érték ismert, ezek alapján kell jól ajánlanunk
 - A megfelelő működéshez még így is kell egy kritikus tömeg (felhasználókból), hogy:
 - (1) minden felhasználóhoz legyen elég sok hasonló ízlésű,
 - (2) akik ráadásul értékelték is a kérdéses terméket

Hibrid ajánlórendszerek



Hibrid ajánlórendszerek

A becslés javítható, ha az egyes megközelítéseket kombináljuk:

- 1 A külön alkalmazott tartalomalapú, illetve kollaboratív ajánlók eredményeit kombináljuk
- 2 Tartalomalapú ajánlót kollaboratív elemekkel egészítjük ki
- 3 Kollaboratív ajánlót tartalomalapú elemekkel egészítjük ki
- 4 Egységes modellt dolgozunk ki, mely egyszerre dolgozik a két megközelítés szerint, nem választja szét a két irányt

Hasonlóságbecslés gradiensmódszerrel

- Alapötlet:
 - Az előző fejezetben a felhasználók hasonlóságának kiszámítására manuálisan adtunk képletet (koszinusz, Pearson)
 - Bízunk ezt is az ajánlórendszerre: tanuljon alkalmas hasonlóságértékeket úgy, hogy a végeredmény jó legyen
- Az egyes felhasználók hasonlóságát leíró $sim(i, j)$ függvényt rendezzük egy \mathbf{H} mátrixba:

$$\mathbf{H} = [h_{i,j} = sim(i, j)]$$

- Ekkor az alábbi módon becsült hasznosságok összes (négyzetes) hibáját akarjuk minimalizálni:

$$\hat{r}_{c,s} = \frac{\sum_{c' \in C_s} h_{c,c'} \cdot r_{c',s}}{\sum_{c' \in C_s} |h_{c,c'}|}, \text{ ahol } C_s = \{c \mid c \in C, r_{c,s} \neq \emptyset\}$$

Hasonlóságbecslés gradiensmódszerrel

- A minimalizálandó hibafüggvény tehát :

$$E(\mathbf{H}, \mathcal{L}) = \frac{1}{2} \sum_{r_{c,s} \in \mathcal{L}} (\hat{r}_{c,s}(\mathbf{H}) - r_{c,s})^2 + \gamma \sum_{i < j} h_{i,j},$$

ahol \mathcal{L} a tanítóhalmaz, γ a regularizációs konstans

- Az ún. sztochasztikus gradiensmódszert használva az egyes tanítópontokkal egyenként javítjuk egy adott $r_{c,s}$ példában érintett $h_{c,i}$ paramétereket:

$$h_{c,i}^{\text{új}} = h_{c,i}^{\text{rég}} - \eta \cdot \text{sign} \left((\hat{r}_{c,s} - r_{c,s}) \frac{\partial \hat{r}_{c,s}}{\partial h_{c,i}} \right) - \eta \gamma h_{c,i}^{\text{rég}},$$

ahol η a bátorsági tényező

Faktormátrixok (a gradiensmódszerhez)

- A gradiensmódszer problémája, hogy kis számú tanítópontból akarunk rengeteg $h_{i,j}$ hasonlóságértéket meghatározni, így hajlamos a túltanulásra (és a memóriában sem fér el!)
- Csökkentsük a szabadsági fokot, a paraméterek számát!
- Erre egy lehetséges megoldás, hogy a \mathbf{H} mátrixot két ún. faktormátrix szorzataként írjuk fel:

$$\begin{matrix} n \times n & n \times k & k \times n \\ \mathbf{H} = & \mathbf{P} \cdot & \mathbf{Q} \end{matrix}$$

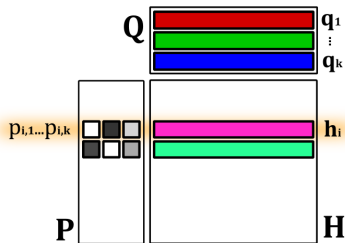
- Így az eredeti mátrix egy k -rangú approximációját kaptuk
- A becsült $\hat{r}_{c,s}$ értékek az így kapott $p_{i,j}$, $q_{i,j}$ paraméterek szerint továbbra is deriválhatók, így ezek betanulására a gradiensmódszer hasonlóan alkalmazható

A faktormátrixok szemléletes jelentése

- A $\mathbf{P} \cdot \mathbf{Q}$ mátrixszorzásra úgy is tekinthetünk, hogy a \mathbf{H} i -edik sorát \mathbf{Q} egyes sorainak $p_{i,1}, \dots, p_{i,k}$ -együtthatós lineáris kombinációjával közelítjük:




$$\mathbf{h}_i = \sum_{j=1}^k p_{i,j} \cdot \mathbf{q}_j$$

A faktormátrixok szemléletes jelentése



- Ez szemléletesen azt jelenti, hogy Q soraiban az egyes felhasználó-sztereotípiákat leíró vektorok jelennek meg, P soraiban pedig az, hogy az adott felhasználó mennyire tartozik bele az adott sztereotípiába

Források

-  Gediminas Adomavicius and Alexander Tuzhilin.
Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions.
IEEE Trans. on Knowl. and Data Eng., 17(6):734–749, 2005.
-  J. Ben Schafer, Joseph Konstan, and John Riedi.
Recommender systems in e-commerce.
In *EC '99: Proceedings of the 1st ACM conference on E-commerce*, pages 158–166, New York, USA, 1999. ACM.
-  Andreas Töscher, Michael Jahrer, and Robert Legenstein.
Improved neighborhood-based algorithms for large-scale recommender systems.
In *NETFLIX '08: Proceedings of the 2nd KDD Workshop*, pages 1–6, New York, NY, USA, 2008. ACM.

Vége!

