Nagyméretű adathalmazok kezelése

Ajánló rendszerek

Gubek Andrea

BME–SZIT

1. Ajánló rendszerek
   - Tartalom alapú ajánló rendszerek
   - Együttműködés alapú ajánló rendszerek
2. The Movie Genome és a Jinni
   - Movie Genome

Megjegyzések:

- user = felhasználó = vásárló
- item = tétel = áru = dokumentum

Ajánló rendszer

- Az **ajánló rendszerek** olyan szoftveres technikák és eszközök, amelyek javaslatokat nyújtanak a felhasználó számára hasznos tételekről, árukról (item). Ezek a javaslatok arra hivatottak, hogy segítsék a felhasználót különféle döntésekben.
- Fejlesztéshez szakemberek több területről

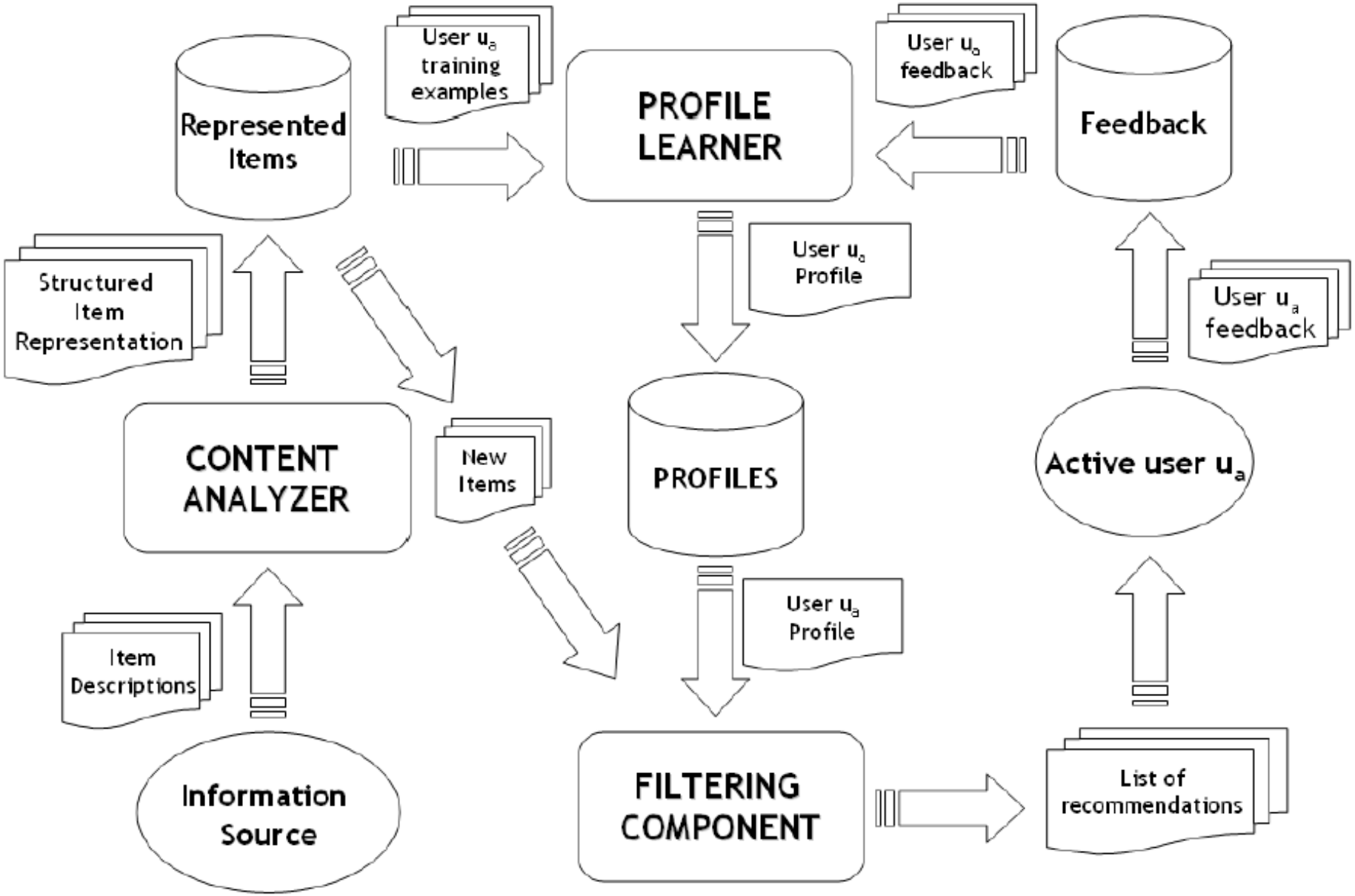
Tartalom alapú ajánló rendszerek

- Olyan árukat próbálnak ajánlani, amelyek hasonlóak a vásárló által már régebben kedveltekhez
- Információk tömkelege + dinamikusság és heterogén természet $\Rightarrow$ egyre nehezebb megtalálni, amire szükségünk van
- Szükségünk van segítségre a hatalmas adathalmazokban való keresgéléshez

- Az ajánló algoritmusok a vásárló érdeklődési köreit használják (amazon)
- Két irányvonal:
  - Szomszédság alapú (vásárló eddigi választásai alapján)
  - Együttműködő (hasonló véleményeken levő felhasználók választásai alapján)

Működés

- **Content analyzer**: előfeldolgozó lépés, amikor az információknak nincs struktúrájuk. Komponens fő feladata, hogy az adatokat az árukról megfelelő formába hozza a következő lépés számára.
- **Profile learner**: begyűjti a felhasználó érdeklődési körének adatait, általánosítja őket, a user profile létrehozásának érdekében (általában gépi tanuló technikákon keresztül)
- **Filtering component**: feldolgozza a user profile-t, annak reprezentációit összehasonlítja az ajánlandó tételekkel


Represented Items
User u_a training examples
PROFILE LEARNER
User u_a feedback
Feedback
Structured Item Representation
User u_a Profile
User u_a feedback
CONTENT ANALYZER
New Items
PROFILES
Active user u_a
Item Descriptions
User u_a Profile
Information Source
FILTERING COMPONENT
List of recommendations

Explicit feedback

- **Like/dislike**: binárisan osztályozódnak a tételek
- **Ratings**: értékelés, egy diszkrét numerikus skálán (pl. 1-től 5-ig)
- **Szöveges megjegyzések**: segítenek a többi vásárlónak megítélni a termék „jóságát”

Jelölések

- u_a: aktív user profilja
- TR_a: tanító halmaz
- r_k: az u_a user által adott értékelés az I_k árucikkre vonatkoztatva
- $TR_a = \langle I_k, r_k \rangle$
- L_a: ajánlások listája,

- $TR_a \xrightarrow{\text{Profile learner}}$ user profile (prediktív modell)
- User profile $\xrightarrow{\text{Filtering component}} L_a$ (list of recommendations)
- Profil legyen up-to-date
- Feedback

Előnyök

- **User független**: csak a user saját értékeléseit használja fel a profil kiépítésére
- **Transzparens**: működésükről magyarázatot lehet adni, szimplán felsorolva a tulajdonságokat és leírásokat, amely egy elemnél arra vezetett, hogy bekerüljön az L_a-ba.
- **Új elem**: a tartalom alapú rendszerek képesek olyan tételeket is ajánlani, amelyeket még senki nem értékelt (nem szenvednek a first-rater problémától)

Hátrányok

- **Korlátolt tartalom analízis**: tételekhez rendelt tulajdonságok száma és típusa terén határoltak, domain információk is kellhetnek
- **Túlrészletezés**: nincs megfelelő metódusuk arra, hogy ha valami nem várt dologgal kerülnek szembe (serendipity probléma)
- **Új felhasználó**: megfelelő mennyiségű értékelést be kell gyűjteni, mielőtt a rendszer rendesen felmérheti a felhasználó által előnyben részesített tételeket, és pontos ajánlásokat tud tenni

Item reprezentáció

- Tételek – attribútumok és tulajdonságok
- Általában a tétel leírások szövegesek
- Nincsenek jól definiált értékkel rendelkező attribútumok
- Szöveges leírások → bonyodalom (kétértelműség)
- Szöveg illesztés problémái:
  - Többjelentésűség (polysemy)
  - Rokonértelműség (synonymy)

Szemantikus analízis

Kétféle módon:

- Ontológiák használatával
- Enciklopédikus tudásforrások használatával

Szemantikus analízis 2

A főbb stratégiák szempontjai:

- a tudásbázis típusa (lexikon, ontológia stb.)
- a tételek reprezentálására választott módszerek
- a user profilban előforduló tartalom típusa
- tétel-profil illesztési stratégia

Módszerek user profilok tanulására

- Gépi tanulási módszerek jól alkalmazhatók szöveg kategorizálásra
- Tanuló dokumentumok $\xrightarrow{\text{következtető eljárás}}$ szöveg osztályozó
- User profilok tanulása $\sim$ bináris szöveg osztályozás
- Kategóriák halmaza: $C = \{c_+, c_-\}$
- c_+: a pozitív osztály (user-likes)
- c_- a negatív osztály (user-dislikes)

Valószínűségi módszerek, naiv Bayes

- Tanulmányozott adatok alapján kialakítják a valószínűségi modellt
- A modell megbecsüli a $P(c \mid d)$ valószínűséget
- $P(c)$: annak a valószínűsége, hogy egy c-beli dokumentummal van dolgunk
- $P(d \mid c)$: annak a valószínűsége, hogy d dokumentummal van dolgunk c-n belül
- $P(d)$: annak a valószínűsége, hogy d-vel van dolgunk

- Bayes tétel alapján: $P(c \mid d) = \dfrac{P(c)P(d \mid c)}{P(d)}$
- $c = \text{argmax}_{c_j} \dfrac{P(c_j)P(d \mid c_j)}{P(d)}$
- $P(d)$-vel egyszerűsíthetünk
- $P(d \mid c)$ és $P(c)$ értékét kikövetkeztetjük
- Gond: az adat kevés a jó valószínűségi értékek meghatározásához $\rightarrow$
- Feltételezés: dokumentumban található szavak egymástól feltételesen függetlenek az osztályon belül
- Szavak valószínűségei egyenként kerülnek meghatározásra

Két modell

- Dokumentum = vektor a szótár teste (V) felett
- Vektor elemei = előfordult-e a dokumentumban
- Többváltozós Bernoulli (multivariate): szavakat binárisan kódolja
- Többtagú (multinomial) esemény modell: megszámlálja az előfordulást is
- Különbség csak nagy szótárak esetén vehető észre

Relevancia feedback, Rocchio algoritmusa

- Alapelv: felhasználók visszajeleznek a rendszernek → user profil finomítás
- Vektoros reprezentáció
- Vektor elemei = dokumentum elemei (szavak)
- Súlyozás TF-IDF-fel

TF-IDF

- a ritka szavak nem kevésbé fontosak, mint a gyakoriak (IDF feltétel)
- a szavak többszöri előfordulása egy dokumentumban nem kevésbé fontos, mint az egyszeriek (TF feltétel)
- a hosszú dokumentumok nem részesülnek előnyben a rövidebbekhez képest (normalizációs feltevés)

- $\forall C$-beli osztályhoz prototípus vektor
- Új d dokumentum osztályozásához hasonlósági érték számítása
- Abba az osztályba sorol, ahol ez maximális

Együttműködés alapú ajánló rendszerek

- CF - Collaborative filtering
- Felhasználó specifikus ajánlások
- Nincs szüksége különleges információkra az ajánlott tételekről vagy a felhasználókról

Netflix

- 2006 októberben kezdődött
- Először jutottak hozzá nagyméretű adathalmazhoz:
  - 100480507 szavazatot tartalmazott
  - 480189 felhasználótól
  - 17770 filmre vonatkozóan

Netflix 2

- **Tanító osztályzások**: <felhasználó, film, értékelés dátuma, értékelés>
- Ismert filmek címe és megjelenési éve
- **Teszt halmaz**: 2817131 db <felhasználó, film, értékelés dátuma>
- **Cél**: eredmények RMSE-ének minimalizálása (Root Mean Squared Error, négyzetes középérték hiba)

$$RMSE(\hat{\theta}) = \sqrt{\mathbb{E}((\hat{\theta} - \theta)^2)}$$

Netflix 3

- 2009 júliusában ért véget
- Két csapat is megütötte a kellő mércét (Netflix algoritmusánál min. 10%-kal jobb teljesítmény)
- A nyertes a korábban beküldő lett
- Újabb forduló nem indult

- Sokféle bemenet
- Legjobb: explicit feedback
- Gond: nem mindig elérhető → sok ajánló rendszer az implicit feedbacket használja (browsing history, keresési minták)
- Ajánlások létrehozásához: felhasználók és tételek összefüggésbe hozása

Két fő módszer

- Szomszédsági megközelítés (neighborhood approach)
- látens (rejtett) faktor modellek (latent factor models)

A feladat

Adott:

- m felhasználó (vásárló) értékelése
- n tétel (termék)
- Indexelések:
  - u, v: felhasználók
  - i, j, l: termékek
- r_{ui}: az u felhasználó ismert értékelése az i termékre
- $\hat{r}_{ui}$: jósolt értékelések
- t_{ui}: az értékelés időpontja

A feladat 2

- Általában az értékelések nagy %-a hiányzik (Netflixnél 99%)
- Jelölések:
  - $\kappa = \{(u, i) \mid r_{ui} \text{ ismert}\}$
  - R(u): u felhasználó ismert értékelései
  - R(i): az i termék értékei
  - (N(u): amelyet u impliciten véleményezett)
- **Cél**: ismert adatok alapján ismeretlen felhasználói értékelések megjóslása

Baseline predictorok

- Olyan effektusokat gyűjtői, amelyek nem tartalmaznak felhasználó-tétel kölcsönhatásokat
- $b_{ui} = \mu + b_u + b_i$
- μ: az átlagos osztályozás
- b_u és b_i változók jelzik az u felhasználó és az i termék megfigyelt eltéréseit az átlagostól
- Leon, a profi - Bob

- b_u és b_i becsléséhez meg kell oldani a legkisebb négyzet problémát:

$$\min \sum_{(u,i)\in\kappa} (r_{ui} - \mu - b_u - b_i)^2 + \lambda_1 \left(\sum_u b_u^2 + \sum_i b_i^2 \right)$$

Szomszédsági modellek

- $\mathcal{U}$: a felhasználók halmaza
- $\mathcal{I}$: a tételek halmaza
- $\mathcal{R}$: az ismert értékelések halmaza
- $\mathcal{S}$: az értékelések lehetséges értékeinek a halmaza ($\mathcal{S} = [1, 5]$ vagy $\mathcal{S} = \{\text{like}, \text{dislike}\}$)
- $\mathcal{U}_i$: az i terméket értékelt felhasználók halmaza (feltesszük, hogy egy $u \in \mathcal{U}$ user egy $i \in \mathcal{I}$ termékre csak egy értékelést adhat, r_{ui})
- $\mathcal{I}_u$: az u user által értékelt tételek halmaza
- $\mathcal{I}_{uv} = \mathcal{I}_u \cap \mathcal{I}_v$
- $\mathcal{U}_{ij} = \mathcal{U}_i \cap \mathcal{U}_j$

Problémák

- Legjobb tétel (best item): feladat egy u felhasználó számára megtalálni azt az új $i \in \mathcal{I} \setminus \mathcal{I}_u$ elemet, ami a legjobban érdekelné őt
- Top-N

Best item

- Ha az elemekről elérhetőek értékelések → gyakran regresszióként vagy osztályozásként definiálják
- Cél: $f : \mathcal{U} \times \mathcal{I} \to \mathcal{S}$ függvény megtanulása
- f megjósolja az u felhasználó $f(u, i)$ értékelését az i tételre
- f-et felhasználják $i*$ ajánlására, amire a jósolt érték a legmagasabb
- $i* = \text{argmax}_{j \in \mathcal{I} \setminus \mathcal{I}_u} f(u_a, j)$

- Ajánló rendszerek értékelése pontosságuk alapján
- Tipikusan az értékelések $\mathcal{R}$ halmazát szétvágják $\mathcal{R}_{\text{train}}$ és $\mathcal{R}_{\text{test}}$ halmazokra
- Pontosság mérésére
  - átlagos abszolút hiba (Mean Absolute Error, MAE)
  - négyzetes középérték hiba (Root Mean Squared Error, RMSE)

- $\mathrm{MAE}(f) = \frac{1}{|\mathcal{R}_{\text{test}}|} \sum_{r_{ui} \in \mathcal{R}_{\text{test}}} |f(u,i) - r_{ui}|$
- $\mathrm{RMSE}(f) = \sqrt{\frac{1}{|\mathcal{R}_{\text{test}}|} \sum_{r_{ui} \in \mathcal{R}_{\text{test}}} (f(u,i) - r_{ui})^2}$

Ha nem ismertek a tételek értékelései:

- Nem lehet megállapítani a jóslás pontosságát
- Legjobb item megtalálás $\rightarrow$ $L(u_a)$ lista ajánlása, amely N olyan tételt tartalmaz, amely feltehetőleg érdekelni fogja
- Tételek $\mathcal{I}$ halmazát kettébontják egy tanuló és teszthalmazra
- $T(u)$: azon elemek halmaza, amelyeket az adott felhasználó relevánsnak tartott
- Adott tételek listája használható $T(u)$-ként
- Nagy hátrány: minden tétel ugyanolyan fontosnak értékel a user számára

Pontosság:

- $\text{Precision}(L) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|L(u) \cap T(u)|}{|L(u)|}$
- $\text{Recall}(L) = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|L(u) \cap T(u)|}{|T(u)|}$

Előnyök

- **Egyszerűség**: könnyű implementálni, a legegyszerűbb esetben csak egy paraméternek van szüksége finomításra
- **Igazolhatóság**: a módszerek biztosítanak ellenőrzést is a jósolt eredményekhez
- **Stabilitás**: kevésbé gyakorolnak rájuk hatást az újonnan felvett felhasználók, itemek vagy értékelések.

Előnyök 2

- **Hatékonyság**: nem igényel költséges tanuló fázisokat. Ugyan az ajánlási lépés költségesebb, a legközelebbi szomszédok egy offline lépésben előre kiszámolhatóak, tárolásuk kis memóriát igényel, milliós nagyságrendű felhasználókra és tételekre is használható

Movie Genome

- Human Genome Project: összes emberi gén feltérképezése
- Movie Genome – filmek „génjei”
- Hasonló kezdeményezés: Music Genome Project (Pandora)
- Pl. hangulat, árnyalat, cselekmény, felépítés (mood, tone, plot, structure)

Jinni

Genome alapvetően két részre:

- **Experience**: a hangulata a tartalomnak
- **Story**: cselekmény elemek (one man army), felépítés (nemlineáris, story-within-a-story), flagek (erőszak)
- + sok külső tényező

Jinni 2

- Minden filmnek kb. 50 „génje”
- Több ezer lehetséges érték
- Jinni algoritmusa automatikusan besorolja az új filmeket
- Konzisztencia, egyesít több véleményt

Jinni 3

- ún. Movie Personality alapján ajánl filmeket/sorozatokat
- Ezt hasonlítja össze a többi film génjeivel
- Felhasználóként szomszédainktól is kapunk ajánlásokat
- Minden felhasználóhoz gének egy klaszterját rendeli
- Ezeket a géneket folyamatosan módosítja, finomítja
- Egyszerűsített változata a Movie Personality Sketch

Köszönöm a figyelmet!