Kontextus-vezérelt faktorizációs módszerek implicit feedback alapú ajánlási problémákra

Tézisfüzet

Hidasi Balázs

Témavezető: Dr. Magyar Gábor
Külső konzulens: Dr. Tikk Domonkos

Budapesti Műszaki és Gazdaságtudományi Egyetem
Távközlési és Médiainformatikai Tanszék

2015. május

Tartalomjegyzék

1 Bevezetés **2**

2 Témavázlat **4**

2.1 További információ felhasználása mátrix faktorizációban, inicializálás segítségével 4

2.2 Kontextus-vezérelt faktorizációs algoritmusok 5

2.3 ALS alapú faktorizáció gyorsítása 7

2.4 GFF és preferencia modellezés kontextussal 8

3 A disszertáció felépítése **10**

4 Új eredmények összefoglalása **12**

5 Eredmények alkalmazása **16**

Tézisekhez kapcsolódó publikációk **17**

A szerző további publikációi **18**

Hivatkozások **20**

1 Bevezetés

Az ajánlórendszerek olyan információszűrő eszközök, amik segítik az információs túlterheltségben szenvedő felhasználókat azáltal, hogy számukra érdekes/megfelelő/hasznos termékeket (tartalmat, stb) javasolnak nekik. A felhasználók személyre szabott ajánlási listákat kapnak, amik tipikusan csak néhány terméket tartalmaznak; ezek a termékek ráadásul nagy valószínűséggel megfelelnek a felhasználó érdeklődésének. A termékek megfelelőségét (az egyes felhasználóknak) az ajánlóalgoritmusok jelzik előre; a felhasználónak a legmagasabb előre jelzett megfelelőségi értékkel rendelkező termékeket mutatja a rendszer.

Az ajánlórendszerek központi része az ajánlóalgoritmus, ami a megfelelőségük alapján rangsorolja a termékeket. Az ajánlóalgoritmusokat öt nagyobb csoportba szokás sorolni [13], amelyek közül a két legjelentősebb stratégia a kollaboratív szűrést (collaborative filtering – CF) és a tartalom alapú szűrést (content-based filtering – CBF) alkalmazó módszerek csoportja. A CF algoritmusok csak a felhasználó–termék interakciókat (eseményeket, tranzakciókat) használják fel a működésük során. Ezen módszerek felteszik, hogy két felhasználó hasonló, ha hasonló termékeket fogyasztottak; és két termék hasonló, ha hasonló felhasználók fogyasztották őket [15]. A CBF algoritmusok a termékek metaadatait (pl. szerző, műfaj, stb) használják. Első lépésként a termékek metaadatait a szövegbányászatból és az információ kinyerés (information retrieval) területéről ismert módszerekkel elemzik [2]. Ezután felhasználói profilokat képeznek, felhasználva, hogy a felhasználók mely termékeket kedvelték. A preferenciák előrejelzése a termékek metaadatainak és a felhasználói profilok összevetésével történik [9]. Általánosságban a CF módszerek a legpontosabbak a vegytiszta módszerek közül, pl. pontosabbak a CBF módszereknél, amennyiben elég esemény áll rendelkezésre [10].

A CF algoritmusokat két osztályba szokás sorolni, a memória alapú és modell alapú megoldások közé. Az előbbibe tartoznak a szomszéd módszerek, amik a termékek vagy a felhasználók értékelés vektorait vetik össze, és ezáltal definiálnak hasonlóságokat köztük. Az ajánlások a hasonló termékek vagy felhasználók súlyozott átlagaként állnak elő (pl. [5, 12, 7]). Az elmúlt évtizedben a modell alapú módszerek váltak népszerűvé, mivel sokkal jobban teljesítettek a Netflix Prize versenyen [4], egy 2006-ban indult nyilvános versenyen, ami hosszú ideig a legnagyobb és legjelentősebb méretű bechmarknak számított. A modell alapú módszerek általánosítanak az adatokból és olyan modelleket építenek, amik jól képesek előre jelezni a felhasználók preferenciáit. Ezek közül is a faktorizációs algoritmusok bizonyultak a legsikeresebb megoldásnak. Ezek minden terméket és felhasználót egy K dimenziós látens jellemzőtérben reprezentálnak, mint egy K hosszú jellemzővektor.

Matrix faktorizáció: A legismertebb látens jellemzőket használó algoritmusok a mátrix faktorizációs módszerek (pl. [3, 16, 11, 17, 6, 14]). Ezek a megfigyelt értékeléseket egy preferencia mátrixba (R) rendezik, aminek a két dimenziója a felhasználóknak és a termékeknek felel meg. Ha a u felhasználó r-rel értékeli az i terméket, akkor $R_{u,i} = r$. R egy nagy méretű, de igen ritka mátrix. A mátrix faktorizáció alapja, hogy ezt az R mátrixot és alacsony rangú mátrix szorzatára bontjuk fel ($\hat{R} = (M^{(U)})^T M^{(I)}$), amikre jellemzőmátrixként hivatkozunk. Az egyik jellemzőmátrix a felhasználókhoz ($M^{(U)}$) a másik a termékekhez tartozik ($M^{(I)}$). A felhasználók jellemzőmátrixának u. sora az u felhasználó (K hosszú) látens jellemzővektora. A termékek jellemzővektorait hasonló módon értelmezzük. Az u felhasználó előre jelzett értékelése/preferenciája az i termékre

a jellemzővektoraik skalárszorzata, azaz $\hat{R}_{u,i} = (M_u^{(U)})^T M_i^{(I)}$.

Interakciók típusai: A felhasználó–termék interakciók típusától függően az ajánlási problémák lehetnek explicit vagy implicit visszajelzés (feedback) alapúak. Explicit visszajelzést a felhasználók aktív hozzájárulásként biztosítanak, leggyakrabban értékelések formájában. Ez expliciten kódolja a preferenciáikat a termék(ek)kel kapcsolatban. A klasszikus explicit feedback alapú feladat az értékelés előrejelzés, azaz a cél a hiányzó értékelések minél pontosabb előrejelzése. Ezzel szemben egy ajánló feladata néhány érdekesnek/hasznosnak ítélt termék kiválasztása az egyes felhasználók számára. Ehhez a termékeket először rangsorolni kell (a felhasználók számára való megfelelőségük alapján), majd a rangsor első néhány termékét visszaadhatjuk ajánlásként. Ez a topN ajánlási feladat. Az értékelés előrejelzési feladat eredménye átalakítható topN ajánlássá úgy, hogy a legnagyobb előre jelzett értékeléssel rendelkező termékeket ajánljuk. Ugyanakkor a jó értékelés előrejelzés nem mindig eredményez jó topN ajánlást.

Implicit feedbacket a felhasználók viselkedésének passzív megfigyelésével nyerhetünk, miközben a szolgáltatást (pl. egy webshopot) használják. Mivel nincs szükség a felhasználó aktív hozzájárulására, ezért nagy mennyiségben gyűjthető. Ez kifejezetten fontos szempont a gyakorlatban. Viszont a preferenciákat ki kell következtetni az így gyűjtött interakciókból. Ha a felhasználó interakcióba lépett egy termékkel az egy zajos visszajelzés pozitív preferenciáról. A negatív visszajelzések gyűjtése még problémásabb, mivel az interakció hiányának több oka lehet, amik közül a leggyakoribb az, hogy a felhasználó nem is tud a termék létezéséről. Az implicit feedbackkel dolgozó algoritmusoknak ezért figyelembe kell venniük a "hiányzó" eseményeket is.

Kontextus-vezéreltség: A kontextus-vezérelt[1] ajánlórendszerek (context-aware recommender systems – CARS) [1] a tranzakciókon felül egyéb információt is figyelembe vesznek, amire összefoglaló néven kontextusnak nevezünk. A kontextus-vezérelt ajánlás jelentősen megnövelheti az ajánlás pontosságát: (1) Az algoritmus képes kezelni a kontextus függő hatásokat a tanulási fázisban. Például, bizonyos változások a felhasználók viselkedésében, mint a szezonális hatások csak a kontextus ismeretében érthetőek. A kontextust nem használó algoritmusoknak a hasonló hatások véletlenszerűnek tűnnek és a helytelen kezelés folyományaként hasonló helyzet áll elő, mint zajos adatokon való tanulásnál. (2) Az ajánlási listák a kontextus függvényében változnak, így jobban alkalmazkodnak a felhasználók aktuális igényeihez.

Ajánlóalgoritmusok kiértékelése: A topN ajánlók kiértékelése ajánlási pontosság szempontjából a következők szerint történik. A termékeket egy adott felhasználó számára az adott kontextuson belül rangsoroljuk az előre jelzett megfelelőségük alapján ($\hat{r}$). A kiértékelést különböző mértékek segítségével végezzük egy, a tanítóhalmazzal diszjunkt, teszt adatbázison. Egy adott felhasználó–kontextus (azaz lekérdezés) számára a teszthalmaz ezen felhasználójának ebben a kontextusban előforduló eseményeinek termékei számítanak relevánsnak. Az ajánlott termékek a lekérdezés számára generált rangsor első N darab eleme. A disszertációban általában $N = 20$-at használok; az eredmények $N = 10$ és $N = 5$ mellett erősen korrelálnak ezzel. Fontos megemlíteni, hogy a kiértékelés során az összes terméket rangsorolom.

Az ajánlási pontosságot több mértékkel lehet mérni. A disszertációban főként a re-

[1] A terület nevének a fordítása nem teljesen pontos. A terület algoritmusai figyelembe veszik a kontextust és az ajánlások is ennek megfelelően változhatnak, ugyanakkor nem közvetlenül a kontextus vezérli az ajánlásokat.

call@N mértékre támaszkodok, ami az ajánlott releváns termékek és a releváns termékek aránya. Más szavakkal, ez azon tesztesemények aránya, amiknek a terméke a megfelelő ajánlási listák első N eleme között van. A recall jól közelít bizonyos gyakorlati ajánlási környezeteket és jól korrelál az átkattintási aránnyal (click-through rate – CTR), egy gyakran használt online mértékkel.

2 Témavázlat

A kutatás központi eleme a kontextus-vezérelt, implicit feedback alapú ajánlási probléma megoldása faktorizációval; és jelentősen befolyásolják gyakorlatból vett szempontok. A kutatás célja, hogy integrálja a kontextust (valamint egyéb információkat, pl. metaadatokat) faktorizációs algoritmusokba, így növelve azok pontosságát implicit feedbackre építő topN ajánlási problémák esetén. A kontextust, mint esemény kontextust (azaz a tranzakciókhoz és nem az egyes entitásokhoz kapcsolódó információt) definiálom. Az ajánlási pontosság fő mérőszámaként a recall@20 mértéket használom.

Kétféle kontextust használok, a *szezonalitást* és a *szekvenciális kontextust*. Ezen kontextusok gyakorlati hasznossága abban rejlik, hogy könnyen létrehozhatóak szinte minden implicit adatsor esetén, mivel a származtatásukhoz csak arra van szükség, hogy az események időbélyege is rendelkezésre álljon.

Szezonalitás: A legtöbb olyan terület, ahol ajánlórendszereket alkalmazunk, mutat valamilyen szezonális viselkedési mintázatot, mivel az emberi viselkedés sok esetben periodikus. Emiatt a szezonális információ kontextusként való használata nyilvánvaló [8]. A származtatásához első lépésként a szezon hosszát kell definiálni. A szezonon belül nem számítunk szabályszerű ismétlődésekre a felhasználók aggregált viselkedésében. Két különböző szezonban ugyanazon, a kezdéstől számított relatív időpontjánál az aggregált viselkedés hasonlóságára számítunk. A szezon optimális hossza az adatoktól függ. Második lépésként *idősávokat* határozunk meg a szezonon belül, ezek lesznek a kontextus lehetséges állapotai. Az idősávok határozzák meg a szezonalitás felbontását és optimális kialakításuk szintén az adatoktól függ. Az események kontextusát az időbélyegük alapján határozzuk meg, attól függően, hogy melyik idősávba esnek.

Szekvenciális kontextus: Bizonyos alkalmazási területeken, mint például filmeknél vagy zenénél, a felhasználók hasonló termékeket fogyasztanak. Más esetekben, mint például elektronikus eszközök vásárlásánál vagy az e-kereskedelemben viszont pont hogy kerülik a hasonló termékeket és inkább kiegészítőket keresnek. Mindkét típusú területen megfigyelhetőek viszont szekvenciális mintázatok. A szekvenciális kontextust a [T2] cikkben javasoltam. Egy tranzakció szekvenciális kontextusa az eseményben résztvevő felhasználó előző eseményének terméke. Ez az információ segít az ismétlődésekhez kapcsolódó használati mintázatok és a sorrendfüggő fogyasztási szokások modellezésében.

A módszerek és algoritmusok kiértékeléséhez öt implicit feeback alapú adatbázist használok, amik közül három nyilvánosan elérhető.

2.1 További információ felhasználása mátrix faktorizációban, inicializálás segítségével

A kutatás első fázisa annak a megvizsgálása, hogy a tisztán CF faktorizációs eljárásokba bevihető-e valamilyen más információ, anélkül, hogy az algoritmusokon változtatni kell-

jen, és ezáltal megnövelhető-e ezen módszerek hatékonysága. A mátrix faktorizációs algoritmusok első lépése, hogy véletlenszerűen inicializálják a jellemzőmátrixokat, és a tanítási eljárás ezen érétkeket fogja megváltoztatni. Az elképzelés az, hogy ehelyett indítsuk a faktorizációt egy olyan pontból, ami jobban jellemzi az adatainkat, azaz a jellemzővektorok inicializálásához használjunk valamilyen külső információt. Ezáltal ezt a külső információt képesek vagyunk bevinni a modellbe. Bár ez a megoldás még nem kontextus-vezérelt, az ajánlási pontosságot megnövelheti.

Három fő inicializálási megoldást vizsgálok meg. Az inicializálás arra a megfigyelésre épít, hogy a hasonló termékek (tanított) jellemzővektorai is hasonlóak. A termékek hasonlóságát többféle módon is definiálhatjuk, és ehhez a külső információkat is felhasználhatjuk. Az első lépés mindhárom módszer esetében egy termék (vagy felhasználó) leírómátrix definiálása a külső információk (pl. metaadat, kontextus) segítségével. Ezután ezt a leírómátrixot faktorizáljuk, így megkapjuk az egyes termékek és a leíró dimenzió entitásainak (metaadat kifejezések vagy kontextus állapotok) látens jellemzővektorait. Innen többféle módon is továbbléphetünk:

- Mivel a hasonló termékek jellemzővektorai hasonlóak a faktorizálás után, használhatjuk a termék jellemzővektorokat az inicializáláshoz.
- A leíróvektorok közötti hasonlóságot is definiálhatjuk, de a teljes hasonlósági mátrix kiszámítása a gyakorlatban megvalósíthatatlan. Viszont a hasonlóságok közelíthetőek a látens jellemzővektorok segítségével. Mivel a hasonlóságmátrix a leírómátrixnak és a transzponáltjának a szorzata, olyan jellemzővektorok is kiszámíthatóak – a faktorizáció után keletkező termék és entitás jellemzővektorok segítségével – amik hasonlóságai jobban közelítik az eredeti hasonlóságokat. Ezek a vektorok használhatóak az inicializáláshoz.
- A hasonlóságot úgy is definiálhatjuk, mint a többi termékhez való hasonlóság vektorok közötti hasonlóságot. Bár a pontos hasonlóságok kiszámítása ebben az esetben sem valósítható meg a gyakorlatban, az előző pontban leírt módszerhez hasonló megoldással ezek is közelíthetőek.

Ezeket a módszereket kipróbáltam az iALS, egy gyakran használt, implicit feedback alapú mátrix faktorizáció javítására. Az ajánlási pontosság jelentősen megnőtt a véletlenszerű inicializáláshoz képest. Összehasonlítottam a metaadat és kontextus alapú inicializálásokat is, és azt találtam, hogy a kontextussal jobb eredményt lehet elérni. A két információtípus kombinálása pedig tovább növeli a pontosságot.

2.2 Kontextus-vezérelt faktorizációs algoritmusok

A kutatás következő fázisában kifejlesztettem két kontextus vezérelt faktorizációs algoritmust, amik hatékonyan képesek tanulni az implicit adatokon. Mindkét algoritmus egy N_D dimenziós tenzorban reprezentálható tenzoron (R) dolgozik. A tenzor egy-egy dimenziója megfelel a felhasználóknak (felhasználó azonosítók) és a termékeknek (termékazonosítók), a maradék $N_D - 2$ pedig a kontextus dimenzióknak. R nullákat és egyeseket tartalmaz: $r_{u,i,c_1,\cdots,c_{N_D-2}} = 1$, ha az u felhasználónak volt legalább egy eseménye az i terméken, miközben a j. kontextus állapota c_j volt. R minden eleme ismert (azaz nincsenek hiányzó "értékelések"), de az egyesek száma jelentősen alacsonyabb a nullákénál. Ez a fajta kialakítás azt feltételezi, hogy egy esemény megléte pozitív, hiánya pedig negatív preferenciát kódol. Mivel a hiányzó esemény és a negatív preferencia között sokkal gyengébb a kapcsolat, mint az esemény és a pozitív preferencia

között, egy súlyfüggvény létrehozására is szükség van. A $\mathcal{W}(i_1, \ldots, i_{N_D})$ súlyfüggvény minden egyes lehetséges entitás kombinációhoz egy valós értéket rendel. A gyakorlati szempontból jó $\mathcal{W}(\cdot)$ az adatoktól függ és befolyásolhatja a tanítás hatékonyságát is. Az egyszerűség kedvéért tegyük fel, hogy $\mathcal{W}(\cdot)$ érétke 1 a hiányzó eseményeknél és $100 \cdot \#(i_1, \ldots, i_{N_D})$ a többi helyen. Az algoritmusok a súlyozott négyzetes hibaösszegre optimalizálnak (ami ekvivalens az RMSE-re való optimalizálással), ahol a célváltozók az R beli értékek és a súlyokat $\mathcal{W}(\cdot)$ szolgáltatja. Az optimalizáláshoz az alternáló legkisebb négyzetes hibák módszerét (alternating least squuares - ALS) használom. Ez azt jelenti, hogy egy adott időpillanatban egy kivételével az összes jellemzőmátrix fix, és csak és kizárólag a nem fixált mátrix jellemzőinek értékét számítjuk ki, mint legkisebb négyzetes megoldást. Ez az eljárás iteratív módon csökkenti a hibát. Egy teljesen kitöltött tenzoron (mint amilyen az R is) a naiv ALS rosszul skálázódik. De a számítások okos szétválasztása és bizonyos statisztikák előzetes kiszámítása lehetővé teszi a hatékony tanítást.

A két algoritmus közötti különbség az eltérő preferencia modellek használata, azaz a becsült preferencia kiszámításához használt képlet eltérő.

iTALS: Az iTALS algoritmus az u felhasználó preferenciáját az i terméken adott kontextus állapotok mellett, mint a megfelelő jellemzővektorok elemenkénti (Hadamard) szorzataként előálló vektor koordinátáinak összegeként (azaz N_D vektor "skalárszorzataként") becsüli. Ez az N-utas (N-way) interakció modell. Az alábbi kifejezés formálisan leírja a modellt[2]:

$$\hat{r}_{i_1,\ldots i_{N_D}} = 1^T \left(M^1_{i_1} \circ M^2_{i_2} \circ \cdots \circ M^{N_D}_{i_{N_D}} \right) \tag{1}$$

Ez a modell azt feltételezi, hogy a preferenciák az összes dimenzió együttes interakciójaként alakulnak ki. Az ajánlások szempontjából a modell a felhasználó–termék interakció újrasúlyozása egy kontextus függő jellemzővektor által (ami $N_D > 3$ esetén több jellemzővektor elemenkénti szorzataként áll elő).

iTALSx: Ez a módszer alapvetően háromdimenziós problémákhoz (felhasználó, termék és egy kontextus) lett kifejlesztve. Az u felhasználó preferenciája az i terméken adott kontextus mellett, mint a megfelelő jellemzővektorok páronkénti skalárszorzatainak összegeként áll elő. Azaz a felhasználó és termék, a felhasználó és a kontextus, valamint a termék és a kontextus jellemző vektorok skalárszorzatainak összegével becsüljük a preferenciát. Ez a modell a páronkénti interakció modell. A modellt a következő módon fejezhetjük ki formálisan:

$$\hat{r}_{u,i,c} = 1^T \left(M^{(U)}_u \circ M^{(I)}_i + M^{(U)}_u \circ M^{(C)}_c + M^{(I)}_i \circ M^{(C)}_c \right) \tag{2}$$

Ebben a modellben a preferencia a felhasználó–termék interakció, egy kontextus függő termék és egy kontextus függő felhasználó bias összegeként áll elő. A kontextus függő felhasználó bias nem vesz részt ugyan a termékek rangsorolásban – hiszen mindig egy adott felhasználónak ajánlunk egy adott kontextus mellett, így az értéke minden termékre megegyezik – de képes csökkenteni a kontextus-függő mintázatok hatását a tanítás során, amit egy klasszikus mátrix faktorizáció számára zajként jelenne meg.

Algoritmikus komplexitás: Mindkét algoritmus esetében egy epoch (azaz minden jellemzőmátrix egyszeri újraszámolásának) komplexitása $O(N_D N^+ K^2 + \sum_{i=1}^{N_D} S_i K^3)$,

[2] Az eltolás (bias) értékek nem kerülnek feltűntetésre a könnyebb érthetőség végett.

ahol N_D, $N+$, K a dimenziók, események és (vektoronkénti) látens jellemzők száma, S_i pedig a tenzor i. dimenziójának mérete (azaz a felhasználók/termékek/kontextus állapotok száma). Az algoritmus lineárisan skálázódik az események számával. Ez a gyakorlatban kifejezetten fontos a rendelkezésre álló sok esemény és az eseményhalmaz gyors növekedése miatt. Az algoritmus elméletben köbösen skálázódik a látens jellemzők számával. Viszont mivel $N_D N^+ \gg \sum_{i=1}^{N_D} S_i$ és K érték alacsony a gyakorlatban, az első tag a domináns. Így az algoritmus négyzetesen skálázódik a gyakorlatban használt K érétkekre.

Módszerek összehasonlítása: Az iTALS-t és iTALSx-et (a) mátrix faktorizációval, (b) mátrix faktorizációk összességére építő alapvető kontextus-vezérelt eljárással és (c) egymással is összehasonlítottam. A kísérletek során mind szezonalitást, mind szekvenciális kontextust is használtam. A főbb eredmények az alábbiak:

- Mind az iTALS és az iTALSx jelentősen jobban teljesít az ajánlási pontosságot tekintve, mint a mátrix faktorizáció és a kontextus-vezérelt alap megoldás.
- A javasolt szekvenciális kontextussal jelentősen jobb pontosság érhető el, mint kontextus használata nélkül vagy a szezonalitással.
- Az ITALS tanulási kapacitása magasabb, de az N-utas modell érzékenyebb a zajokra és az alacsony faktorszám összemosó hatására. Ezért az iTALS akkor teljesít jobban, mint az iTALSx, ha a jellemzők száma kellően nagy vagy az adatbázis kellően sűrű. Az eredmények azt mutatják, hogy ritka adatok esetén és alacsony jellemzőszám esetén az iTALSx-et érdemes használni.

2.3 ALS alapú faktorizáció gyorsítása

Gyakorlati alkalmazhatóság szempontjából az algoritmusok tanítási ideje különlegesen fontos jelentőséggel bír. A gyorsabb tanítás lehetővé teszi, hogy (1) a modellezett rendszernek egy újabb állapotát modellezzük (különösen fontos akkor, ha a termékek életciklusa rövid vagy ha gyorsan jelennek meg új termékek); (2) gyakrabban tanítsuk újra a modelleket; (3) jobb kompromisszumokat találjunk a tanítás sebessége és a pontosság között azáltal, hogy magasabb faktorszámot vagy több epochot használunk. Két közelítő megoldást javasolok az ALS tanulás felgyorsítására, amik különösen magasabb faktorszám esetén hatékonyak (azaz a gyorsulás annál nagyobb, minél több jellemzőt használunk).

Az ALS esetén a számítási komplexitás amiatt magas, mert magába foglalja egy $K \times K$ méretű lineáris egyenletrendszer megoldását. A javasolt módszerek ennek a direkt megoldását kerülik el.

ALS-CD: Az első megoldás jellemzőnkénti optimalizálásra (coordinate descent - CG) épít. Ebben a módszerben egy adott pillanatban egy kivételével minden modellparaméter fix, és egyszerre mindig egy jellemző kerül kiszámításra. Így a mátrix invertálása helyett elég egy egyszerű osztást használni. Implicit feedback esetén nem egyértelmű ennek a stratégiának a használata, mivel a negatív (hiányzó) eseményeket is figyelembe kell venni. A probléma megoldása, hogy a hiányzó eseményeket $K + 1$ példába tömörítjük.

Az ALS-CD nem közelíti közvetlenül az ALS megoldását, de hasonló eredményeket ad. A módszer használatával az iTALS/iTALSx komplexitása $O(N_D K^3 + N_D N^+ N_I K + \sum_{i=1}^{N_D} S_i K^2)$ lesz – ahol N_I a belső iterációk száma – ami a gyakorlatban használt K értékek tartományán lineáris skálázódást jelent, mivel ott $N_D N^+ N_I$ elnyomja az N_D és

$\sum_{i=1}^{N_D} S_i$ szorzókat.

ALS-CG: A második megoldás konjugált gradienst használ az egyenletrendszer megoldásának közelítésére. A módszer hatékonysága attól, függ, hogy milyen hatékonyan tudjuk kiszámolni az együtthatómátrix és egy tetszőleges vektor szorzatát. Itt az együtthatómátrix egy előre kiszámolt mátrix és egy diádösszeg összegeként áll elő. Így a szorzás is hatékonyan elvégezhető.

Ha a belső iterációk száma megegyezik K-val, akkor az ALS-CG pontosan az ALS megoldását adja. Viszont már jóval alacsonyabb belső iteráció szám mellett is jó közelítések adhatóak. A módszer használatával az iTALS/iTALSx komplexitása $O(N_D N^+ N_I K + N_I K^2 \sum_{i=1}^{N_D} S_i)$, ami a gyakorlatban használt K értékek tartományán lineáris skálázódást jelent.

Összehasonlítás: Az ALS-hez képest jelentős a gyorsulás: a CG $\sim$ 10.6-szer a CD $\sim$ 2.9-szer gyorsabb, mint az ALS $K = 200$ esetén (és a különbség tovább nő, ahogy K-t növeljük). A gyakorlatban gyakrabban használt $K = 80$ esetén a gyorsulás $\sim$ 3.5-szeres CG és $\sim$ 1.3-szeres CD esetén. A három módszer pontossága nagyon hasonló és az eltérések általában jelentéktelenek, Ez azt jelenti, hogy a javasolt gyorsítások anélkül használhatóak, hogy feláldoznánk a módszer pontosságát.

Egymással összehasonlítva kijelenthető, hogy a CG-nek több előnyösebb tulajdonsága van: gyorsabb, stabilabb, egy kicsit pontosabb és az ALS direkt közelítése is előnyös.

2.4 GFF és preferencia modellezés kontextussal

Ahogy az iTALS és ITALSx összehasonlításánál is láttuk, különféle preferencia modellek jók különféle szituációkban. A faktorizáció és az adatsor bizonyos tulajdonságai (pl. faktorszám, ritkaság) az egyik vagy a másik modell számára előnyösek. A faktorizációs módszerek többsége az N-utas vagy a páronkénti modell közül használja az egyiket, miközben a lehetséges modellek száma a dimenziók számának növekedésével exponenciálisan nő. Szintén érdemes megjegyezni, hogy ezek a modellek szimmetrikusak, azaz minden dimenzió ugyanazt a szerepet tölti be; az ajánlási problémánál viszont mindig van két kitűntetett dimenzió: a felhasználóké és a termékeké. A preferencia modell hatással van a tanulás folyamatára is. Ez különösen akkor jelentős, ha a tanulás magába foglal különböző transzformációkat és a számítások okos szétválasztását, hogy a módszer komplexitása alacsony legyen. Az implicit feedback alapú problémánál pontosan ez a helyzet. Például hiába nagyon hasonló az iTALS és az iTALSx, néhány kulcsfontosságú lépés eltérő és más előre kiszámított statisztikákra van szükségük a hatékony tanításhoz.

A preferencia modellezés rendes feltérképezésének hiánya azon rugalmas eszközök hiányára vezethető vissza, amikkel a különféle modellek könnyen kipróbálhatóak, anélkül, hogy minden egyes új modellhez új algoritmust kelljen implementálni. Ez motivált az általános faktorizációs keretrendszer (General Factorization Framework - GFF) megalkotására, ami egy olyan rugalmas algoritmus, aminek az egyik bemenete a preferencia modell, és képes kiszámítani a modellnek (és az adatoknak) megfelelő látens jellemzőmátrixokat. A GFF lehetővé teszi, hogy egyszerűen kísérletezzünk különböző lineáris modellekkel a kontextus-vezérelt ajánlási feladatra, legyen szó implicit vagy explicit feedbackről. A GFF új kutatási irányokat nyit a kontextussal való modellezés területén.

Az alábbi tulajdonságok voltok fontosak a GFF megalkotásakor.

- Szabad kontextus használat: A GFF bármilyen kontextus vezérelt ajánlási problémán működik, függetlenül a dimenziók jelentésétől és számától.
- Nagy preferencia modell osztály: Az egyetlen megszorítás a preferencia modellel kapcsolatban, hogy a probléma dimenzióiban lineáris legyen.[3] Ez az intuitív korlátozás a gyakorlati problémák szempontjából nem jelentős.
- Adatfüggetlenség: a gyakorlatiasabb implicit eset mellett a GFF képes kezelni az explicit feedbacket is, amihez mindössze a súlyfüggvényt kell átállítani.
- Rugalmasság: a GFF súlyfüggvénye kellően általános ahhoz, hogy azon keresztül extra információkat vigyünk a modellbe, például csökkenthetjük a régebbi események súlyát (event decay) vagy az események nem véletlenszerű hiányára (not missing at random) vonatkozó hipotéziseinket is integrálhatjuk.
- Skálázhatóság: a GFF mind a tranzakciók számával, mind a faktorszámmal (a gyakorlatban használt tartományon) lineárisan skálázódik. Így éles ajánló szolgáltatásokban is jól használható.

A GFF segítségével könnyen kísérletezhetünk új preferencia modellekkel. A következő komponenseket definiáltam egy 4 dimenziós (felhasználó, termék, szezonalitás, szekvenciális kontextus) kontextus-vezérelt problémán, amikből a modell összeállítható.

- $\boldsymbol{UI}$**:** Interakció a felhasználó és a termék között, a klasszikus CF modell.
- $\boldsymbol{USI}$**,** $\boldsymbol{UQI}$**,** $\boldsymbol{USQI}$**:** A felhasználó–termék interakció kontextus függő újrasúlyozása, azaz a kontextus befolyásolja az interakciót. Több kontextus dimenzió is használható a súlyozáshoz. De több dimenzió használata esetén a komponens érzékenyebb a zajokra, és csak magasabb faktorszám mellett ad jó eredményt. [T4].
- $\boldsymbol{US}$**,** $\boldsymbol{UQ}$**:** Kontextus függő felhasználó bias, ami a felhasználó–kontextus interakcióból származik. Nem vesz részt a termékek rangsorolásban, de tanítás közben szűri a kontextus függő változásokat az adatban. Ebben a komponensben csak egy kontextus dimenziót engedek meg, mert több dimenzió használata azt modellezné, hogy a kontextus dimenziók is interakcióba lépnek egymással.
- $\boldsymbol{IS}$**,** $\boldsymbol{IQ}$**:** Kontextus függő termék bias, ami a termék–kontextus interakcióból származik. Mind a termékek rangsorolásában, mind a tanulás segítésében részt vesz. Ebben a komponensben is csak egy kontextus dimenzió engedélyezett.
- $\boldsymbol{SQ}$**:** Kontextus dimenziók interakciója. A tradicionális páronkénti modellhez szükséges.

A fenti komponensekből modelleket állítottam össze, amik az ajánlási probléma bizonyos speciális aspektusait is figyelembe veszik. Az általam vizsgált modellek a következők:

- **Interakció modell** $(\boldsymbol{UI + USI + UQI})$**:** Ez a modell az alap felhasználói viselkedés (UI) és a kontextus által befolyásolt viselkedések (USI és UQI) összessége. A modell feltevése, hogy a felhasználók preferenciája egy kontextus független és egy kontextus függő részből tevődik össze. Az utóbbiban a felhasználó–termék interakciókat egy kontextus függő jellemző vektorral átsúlyozzuk. Az $USQI$ a több dimenzióval való átsúlyozás zajossága miatt nem szerepel a modellben.
- **Kontextus interakció modell** $(\boldsymbol{USI + UQI})$**:** Ebben a modellben a prefer-

[3] Ez azt jelenti, hogy egy dimenzió nem léphet interakcióba önmagával a modellben.

encia kizárólag kontextus függő részekből áll. A modell azt feltételezi, hogy az preferenciát erősen befolyásolja a kontextus, és ez a befolyás a felhasználó–termék interakcióra van hatással, nem pedig az egyes entitásokra (felhasználó, termék) külön-külön.

- **Redukált páronkénti modell ($UI + US + IS + UQ + IQ$):** Ez a modell a tradicionális páronkénti modell egy variációja, amiből kihagyjuk a kontextus dimenziók interakcióját (SQ). A kontextus és a felhasználók, illetve termékek interakciója az egyes entitásokkal külön-külön történik, azaz itt a kontextus nem a felhasználó–termék interakciót befolyásolja.
- **Felhasználó bias modell ($UI+US+UQ$):** A modell feltevése, hogy kizárólag a felhasználó lép interakcióba a többi dimenzióval. Az eredmény egy olyan modell, amiben a felhasználó–termék interakciót kiegészítik kontextus függő felhasználó biasok. Bár a termékek sorrendjét ezek a biasok nem befolyásolják, a tanítást zajszűrő hatásukkal segítik.
- **Termék bias modell ($UI+IS+IQ$):** Ebben a modellben a kontextus hatását kontextus függő termék biasokkal írjuk le (pl. bizonyos termékek más és más körülmények között népszerűek). A termék biasok mind a rangsorolásban, mind a tanítás segítésében részt vesznek.
- **Egy komplex model ($UI+US+IS+UQ+IQ+USI+UQI$):** Ez a modell a redukált páronkénti és az interakció modell összessége. Szintén tekinthető egy redukált 3-utas interakciós modellnek, amiből a kontextus-kontextus interakciókat elhagytuk.

A kísérleti eredmények alapján ezek az új modellek jobban illenek az ajánlási problémához, mint a tradicionálisan használt modellek és pontosabbak is. Az interakció modell teljesít a legjobban és közelről követi a kontextus interakció modell. Ez a két modell intuitíve is illik a feladathoz. A faktorszám befolyásolja a modellek pontosságának rangsorát. Kevesebb jellemző esetén az alacsonyabb szintű interakciókat tartalmazó modellek jobban teljesítenek, miközben a több dimenzió interakcióját tartalmazó modellek több jellemző esetén jobbak. Az interakció modell végig jól teljesít a gyakorlatban használt faktorszámok tartományán.

3 A disszertáció felépítése

A disszertáció első fejezete magas szinten áttekinti az ajánlórendszerek területét, részletezi az implicit feedback alapú kontextus-vezérelt faktorizáció témaköréhez tartozó fogalmakat, specifikálja a kutatási problémát és ismerteti a kiértékelés általános menetét. A második fejezet áttekinti a kapcsolódó irodalmat.

A következő négy fejezet (a harmadik fejezettől a hatodikig) fedi le a saját kutatásom fővonalát és ismerteti a téziseimet:

- A 3. fejezet megvizsgálja, hogy a mátrix faktorizáció inicializálása miként használható további információ (pl. termék metaadatok, kontextus) felhasználására a faktorizációs algoritmusokban az alap módszerek megváltoztatása nélkül. A módszerek és a kísérleti eredmények alkotják az első téziscsoportot.
- A 4. fejezet ismerteti az általam az implicit feedback alapú ajánlási problémára kidolgozott kontextus-vezérelt faktorizációs algoritmusokat. Mind az iTALS (második téziscsoport), mind az iTALSx (harmadik téziscsoport) algoritmus jól

skálázódó módon oldja meg a feladatot. A két módszer összehasonlítása és különféle problémákra való megfelelőségük vizsgálata a negyedik téziscsoportban van összefoglalva. Szintén ez a fejezet mutatja be a szekvenciális kontextust, ami jelentősen képes javítani az ajánlási pontosságon és szinte minden implicit feedback alapú adatsorhoz könnyen létrehozható.

- Az ötödik fejezet a faktorizációs megoldásokban – mint az előbb említett két algoritmus – használt ALS tanítás gyorsítására és a látens jellemzők számában való skálázódásának javítására koncentrál, miközben nem ront jelentősen az ajánlási pontosságon. Két gyorsítási módszert javaslok és vizsgálok meg. Az egyik az elemenkénti optimalizálásra (coordinate descent), a másik a konjugált gradiensre épít. A két módszert egymással és az alap ALS-sel is összehasonlítom. A módszerek és a kísérleti megfigyelések alkotják az ötödik téziscsoportot.
- A hatodik fejezetben javaslom az általános faktorizációs keretrendszert (General Factorization Framework – GFF), egy algoritmust, ami hatékony kísérletezést tesz lehetővé különféle preferencia modellekkel az implicit feedback alapú kontextus-vezérelt ajánlási környezetben. A GFF segítségével javaslok és megvizsgálok több újszerű modellt, amik figyelembe veszik az ajánlási feladat specialitásait. Ezeket a módszereket összehasonlítom a terület tradicionális modelljeivel, az N-utas és a páronkénti interakció modellel. GFF, az új modellek és a kísérleti eredmények alkotják a hatodik téziscsoportot.

A disszertáció utolsó három fejezete összegzi az eredményeket, áttekinti az eredmények alkalmazását és felvázol több lehetséges jövőbeli kutatási irányt.

4 Új eredmények összefoglalása

1. téziscsoport: Javasoltam, hogy a mátrix faktorizációs módszerek inicializálásához használjunk fel egyéb, a termékekről (vagy a felhasználókról) rendelkezésre álló információt, hogy megnöveljük az ajánlások pontosságát. (A módszerek és az eredmények a következő cikkekben kerültek publikálása: [T1, T3].)

1.1. tézis *Javasoltam, hogy a mátrix faktorizációt a szokásos véletlenszerű kiindulási mátrixok helyett az entitások hasonlóságát kihasználó mátrixokból indítsuk. Az így kapott inicializáló séma általános és bármilyen mátrix faktorizáció esetén felhasználható. A séma két lépése a következő: (1) rendeljünk leíróvektorokat az entitásokhoz; (2) tömörítsük a leíróvektorokat, hogy a tömörített információ mérete megegyezzen a jellemzővektorokéval. Az implicit ALS módszeren és öt adatsoron alkalmazva megmutattam, hogy az inicializáló séma jelentősen megnöveli az ajánlási pontosságot (recall és MAP mérőszámok tekintetében).*

1.2. tézis *Javasoltam a SimFactor algoritmust, ami olyan jellemzővektorok előállítására képes, amelyek jobban megőrzik az entitások közötti hasonlóságokat. A SimFactor nem igényli a gyakorlatban nehezen kiszámítható teljes hasonlóságmátrix kiszámítását. Öt adatsoron megmutattam, hogy a módszer által létrehozott jellemzővektorok jobban közelíti a hasonlóságokat, mint a leíróvektorok egyszerű tömörítésével kapottak. Megmutattam, hogy az így kapott vektorok általában az inicializálás során is jobban teljesítenek.*

1.3. tézis *Javasoltam a* Sim^2*Factor algoritmust, ami olyan jellemző vektorok előállítására képes, amik az entitások egymáshoz való hasonlósága alapján definiált hasonlóság értékeket képesek közelíteni. A* Sim^2*Factor nem igényli a gyakorlatban nehezen kiszámítható teljes hasonlóságmátrix kiszámítását. Megmutattam, hogy az így kapott jellemzővektorok hasznosak az inicializáláskor.*

1.4. tézis *Javasoltam, hogy az entitások leírásához használjuk a kontextust. Megmutattam, hogy a kontextus alapú leírók jobbak az inicializáláshoz, mint a metaadat alapúak. Megmutattam, hogy a kontextus és metaadat alapú inicializálások kombinálása tovább javítja az ajánlási pontosságot.*

2. téziscsoport: Javasoltam az iTALS algoritmust az implicit feedback alapú kontextus-vezérelt ajánlási problémára. (A módszerek és az eredmények a következő cikkben kerültek publikálása: [T2].)

2.1. tézis *Kifejlesztettem az iTALS algoritmust, egy tenzor faktorizációs algoritmust, ami pontszerű preferenciabecslést végez azáltal, hogy a négyzetes hiba súlyozott négyzetösszegére optimalizál. A preferenciákat az N-utas modellel, azaz dimenziónként egy-egy jellemző vektor elemenkénti szorzatában lévő elemek összegével közelíti. Megmutattam, hogy az iTALS jól használható az implicit feedback alapú kontextus-vezérelt problémára úgy, hogy egyeseket használunk a pozitív és nullákat a hiányzó események esetén, mint preferencia értéket, miközben az előbbieket jelentősen felülsúlyozzuk.*

2.2. tézis *Megmutattam, hogy az iTALS jelentősen jobban teljesít a recallal kifejezett ajánlási pontosság szempontjából, mint a kontextust figyelembe nem vevő implicit mátrix faktorizáció, valamint egy előszűrésre építő kontextus-vezérelt módszer.*

2.3. tézis *Megmutattam, hogy az iTALS hatékonyan tanítható ALS-sel az implicit feedback alapú kontextus-vezérelt ajánlási problémán. Megmutattam, hogy az iTALS a gyakorlatban is hatékonyan tanítható, mivel lineárisan skálázódik az események számával, és négyzetesen a gyakorlatban használt tartományon a látens jellemzők számával.*

3. téziscsoport: Javasoltam az iTALSx algoritmust, mint egy alternatív megoldást az implicit feedback alapú kontextus-vezérelt ajánlási problémára. (A módszerek és az eredmények a következő cikkben kerültek publikálása: [T4].)

3.1. tézis *Kifejlesztettem az iTALSx algoritmust, egy tenzor faktorizációs algoritmust, ami pontszerű preferencia becslést végez azáltal, hogy a négyzetes hiba súlyozott négyzetösszegére optimalizál. A preferenciákat a páronkénti interakció modellel, azaz dimenziópáronként a megfelelő jellemzővektorok skalárszorzatainak összegével közelíti. Megmutattam, hogy az iTALSx jól használható az implicit feedback alapú kontextus-vezérelt problémára úgy, hogy egyeseket használunk a pozitív és nullákat a hiányzó események esetén, mint preferencia értéket, miközben az előbbieket jelentősen felülsúlyozzuk.*

3.2. tézis *Megmutattam, hogy az iTALSx jelentősen jobban teljesít a recallal kifejezett ajánlási pontosság szempontjából, mint a kontextust figyelembe nem vevő implicit mátrix faktorizáció, valamint egy előszűrésre építő kontextus-vezérelt módszer.*

3.3. tézis *Megmutattam, hogy az iTALSx hatékonyan tanítható ALS-sel az implicit feedback alapú kontextus-vezérelt ajánlási problémán. Megmutattam, hogy az iTALSx a gyakorlatban is hatékonyan tanítható, mivel lineárisan skálázódik az események számával, és négyzetesen a gyakorlatban használt tartományon a látens jellemzők számával.*

4. téziscsoport: Kísérleteket folytattam iTALS és iTALSx algoritmusokkal, összehasonlítottam őket, valamint meghatároztam egy könnyen használható kontextus dimenziót. (A módszerek és az eredmények a következő cikkekben kerültek publikálása: [T2, T4].)

4.1. tézis *Javasoltam egy újszerű kontextus dimenzió, szekvenciális kontextus, használatát ajánlási problémáknál. Egy esemény szekvenciális kontextusa az ugyanezen felhasználó előző eseményének terméke. Megindokoltam, hogy a kontextus széles körben elérhető a gyakorlatban, hiszen csak az események sorrendezhetőségére épít, ami a legtöbb esetben adott. Megmutattam, hogy a szekvenciális kontextus használatával az ajánlás pontossága széles körben (különböző adatsorok, algoritmusok, modellek, jellemző számok esetén) jelentősen megnövelhető a kontextust nem, valamint a szezonalitást használó esetekhez képest.*

4.2. tézis *Összehasonlítottam az iTALS (N-utas modell) és az iTALSx (páronkénti modell) algoritmusokat. Az N-utas modell használata megfelelőbb akkor, ha a jellemzők száma magas és/vagy az adatsor sűrűbb; egyébként pedig a páronkénti modell használatát javaslom.*

5. téziscsoport: Javasoltam két közelítő módszert az ALS tanítás felgyorsítására. (A módszerek és az eredmények a következő cikkben kerültek publikálása: [T6, T7].)

5.1. tézis *Javasoltam egy konjugált gradiens alapú ALS közelítést, ami általánosan használható ALS alapú faktorizációs algoritmusokban. Megmutattam, hogy a módszer a gyakorlatban használt értékek esetén lineárisan skálázódik a látens jellemzők számával. Megmutattam, hogy a megoldás lehetővé teszi magas számú látens jellemző használatát és jobb kompromisszumok felfedezését a futási idő és a pontosság között. Megmutattam, hogy a módszer csak minimálisan módosítja az ajánlások pontosságát az ALS-hez hasonlítva.*

5.2. tézis *Javasoltam egy jellemzőnkénti optimalizálásra építő ALS variánst, ami általánosan használható ALS alapú faktorizációs algoritmusokban. Megmutattam, hogy a módszer a gyakorlatban használt értékek esetén lineárisan skálázódik a látens jellemzők számával. Megmutattam, hogy a megoldás lehetővé teszi magas számú látens jellemző használatát és jobb kompromisszumok felfedezését a futási idő és a pontosság között. Megmutattam, hogy a módszer csak minimálisan módosítja az ajánlások pontosságát az ALS-hez hasonlítva.*

5.3. tézis *Több szempont szerint is összehasonlítottam a konjugált gradiensre és a jellemzőnkénti optimalizálásra építő közelítő megoldásokat. Megmutattam, hogy a konjugált gradiens alapú módszer jobb, mivel (a) a pontossága jobban közelíti az ALS-ét; (b) gyorsabb; (c) jobban skálázódik; és (d) stabilabb.*

5.4. tézis *Meghatároztam egy jó kompromisszumot a futási idő és az ajánlás pontossága között a közelítő módszerekhez. Ennek eléréséhez azt javasoltam, hogy a belső iterációk számát állítsuk 2-re.*

6. téziscsoport: Javasoltam egy rugalmas algoritmust (GFF), ami lehetővé teszi, hogy kísérletezzünk újszerű preferencia modellekkel. (A módszerek és az eredmények a következő cikkben kerültek publikálása: [T5, T7].)

6.1. tézis *Kifejlesztettem az általános faktorizációs keretrendszert (General Factorization Framework – GFF), egy rugalmas faktorizációs algoritmust az implicit feedback alapú kontextus-vezérelt ajánlási problémára. A GFF rugalmassága abban rejlik, hogy a preferencia modell megadható, mint az algoritmus bemenete. A modell tetszőleges számú dimenziót képes kezelni, valamint a köztük lévő lehetséges interakciók halmazának bármely részhalmazát. Bemutattam, hogy ez a rugalmasság lehetővé teszi, hogy kísérletezzünk újszerű preferencia modellekkel. A GFF az egy attribútumos MDM adatmodellre épít, ami megfelelő a gyakorlatban előforduló kontextus-vezérelt problémákhoz.*

6.2. tézis *Különféle, újszerű preferencia modelleket javasoltam a kontextus-vezérelt ajánlási problémára. Egy négydimenziós kontextus-vezérelt ajánlási problémán kiértékeltem az új modelleket ajánlási pontosság szempontjából. A felhasznált kontextus dimenziók olyanok, hogy minden a gyakorlatban előforduló adatsornál létrehozhatóak, amennyiben az események rendelkeznek időbélyeggel, és emiatt különösen fontosak. Megmutattam, hogy több újszerű modell is jobban teljesít, mint az irodalomban megtalálható tradicionális modellek.*

6.3. tézis *Megmutattam, hogy a javasolt modellek egyike, az interakció modell, általában is jól teljesít. Ez a modell a felhasználó–termék (UI) és a kontextustól függően súlyozott felhasználó–termék (UCI) relációk összessége. Öt adatsorból négy esetén ezt volt a legjobb modell, a maradék egy esetben pedig második helyen végzett. Az utóbbi adatsoron a legjobb modell a kontextus interakció modell volt, ami közeli rokonságban áll az interakció modellel.*

6.4. tézis *Összehasonlítottam a GFF-fel legjobban teljesítő újszerű modelleket a legkorszerűbb faktorizációs módszerekkel. A GFF az újszerű modellekkel jelentősen jobbnak bizonyult, mint a vizsgált módszerek öt adatsorból háromnál és hasonlóan teljesített egyen.*

6.5. tézis *Kiterjesztettem a GFF algoritmust, hogy teljesen kompatibilis legyen a többdimenziós adattér modellel (Multidimensional Dataspace Model – MDM) és így képes legyen további információt, mint például az aktuális session eseményeit vagy termék metaadatokat, is hatékonyan figyelembe venni. Előzetes kísérleteimmel megmutattam, hogy a session események figyelembe vétele jelentősen megnöveli az ajánlási pontosságot.*

5 Eredmények alkalmazása

A kutatásom egészét erősen befolyásolták gyakorlati követelmények.

- Az első téziscsoport egy relatíve olcsó megoldást javasol extra információ figyelembevételére mátrix faktorizációs megoldásokban és így jelentősen megnöveli a pontosságukat.
- A második, harmadik és negyedik téziscsoportok olyan kontextus-vezérelt faktorizációs algoritmusokat javasolnak, amik a gyakorlati, implicit feedback alapú környezetben jól használhatóak. Mind iTALS, mint iTALSx lineárisan skálázódik az események számával. Az itt használt kontextus dimenziók szintén jelentősek gyakorlati szempontból, mivel minden olyan adatsornál rendelkezésre állnak, ahol az eseményekhez időbélyeget is eltárolunk. Az általam javasolt szekvenciális kontextus kifejezetten jól teljesít és jelentősen megnöveli az ajánlási pontosságot.
- Az ötödik téziscsoport a faktorizációnál használt ALS tanítás sebességének és a látens jellemzők számában való skálázódásának javítására koncentrál, miközben nem áldoz jelentősen az ajánlási pontosságból. Ez kifejezetten fontos gyakorlati szempontból, mivel a rövidebb tanítási idők lehetővé teszik a gyakoribb tanítást, valamint jobb kompromisszumokat tesznek lehetővé a futási idő és a pontosság között (pl. azáltal, hogy több látens jellemzőt vagy több epochot használunk) és hatékonyabbá teszik a rendelkezésre álló erőforrások kihasználását.
- A hatodik téziscsoport egy olyan algoritmust javasol, ami rugalmas kísérletezést tesz lehetővé különféle kontextus-vezérelt preferencia modellekkel. Az algoritmus kifejlesztése mögötti motiváció szintén a gyakorlatból származik, mivel a tradicionális kontextus-vezérelt modellek nem veszik figyelembe az ajánlási feladat specialitásait (pl. kitűntetett dimenziók, realisztikus interakciók a dimenziók között, stb).

Az algoritmusokat és a kutatásból származó know-howt sikeresen alkalmaztam a gyakorlatban is. Az algoritmusok egy részét implementáltam a Gravity Research & Development Zrt. – különféle alkalmazási területen ajánló szolgáltatást biztosító cég, nemzetközi ügyfélkörrel – ajánlómotorjában. Az algoritmusokat sikerrel használtam az éles ajánlórendszerben, valamint különböző egyéb ajánlási projektekben, tenderekben és POC-kben. A megoldások alkalmazási köre kiterjed, a teljesség igénye nélkül, az online bevásárlásra, VoD és TV műsorok ajánlására IPTV környezetben [P3], e-kereskedelemre és apróhirdetéses oldalakra.

Emellett az eredmények nagyban hozzájárultak az Európai Unió Seventh Framework Programme (FP7/2007-2013) által támogatott CrowdRec projekthez[4]. A CrowdRec projekt célja az ajánlórendszerek új generációjának kifejlesztése, így többek között a kontextus használata, a felhasználókkal való interakció, stream jellegű ajánlások és heterogén információforrások felhasználása. A kutatásom a CrowdRec projekt kontextussal foglalkozó részéhez kapcsolódik.

[4] Azonosítószám: n°610594

Tézisekhez kapcsolódó publikációk

[T1] Balázs Hidasi and Domonkos Tikk. Enhancing matrix factorization through initialization for implicit feedback databases. In *Proceedings of the 2Nd Workshop on Context-awareness in Retrieval and Recommendation*, CaRR '12, pages 2–9, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1192-2. doi: 10.1145/2162102.2162104. URL `http://doi.acm.org/10.1145/2162102.2162104`.

[T2] Balázs Hidasi and Domonkos Tikk. Fast als-based tensor factorization for context-aware recommendation from implicit feedback. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 7524 of *Lecture Notes in Computer Science*, pages 67–82. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33485-6. doi: 10.1007/978-3-642-33486-3_5. URL `http://dx.doi.org/10.1007/978-3-642-33486-3_5`.

[T3] Balázs Hidasi and Domonkos Tikk. Initializing matrix factorization methods on implicit feedback databases. *Journal of Universal Computer Science*, 19(12): 1834–1853, jun 2013. ISSN 0948-695x. URL `http://www.jucs.org/jucs_19_12/initializing_matrix_factorization_methods`.

[T4] Balázs Hidasi. Factorization models for context-aware recommendations. *Infocommunications Journal*, VI(4):27–34, 2014. ISSN 2061-2079.

[T5] Balázs Hidasi and Domonkos Tikk. General factorization framework for context-aware recommendations. *Data Mining and Knowledge Discovery*, pages 1–30, 2015. ISSN 1384-5810. doi: 10.1007/s10618-015-0417-y. URL `http://dx.doi.org/10.1007/s10618-015-0417-y`.

[T6] Balázs Hidasi and Domonkos Tikk. Speeding up ALS learning via approximate methods for context-aware recommendations. *Knowledge and Information Systems*, pages 1–25, 2015. ISSN 0219-1377. doi: 10.1007/s10115-015-0863-2. URL `http://dx.doi.org/10.1007/s10115-015-0863-2`.

[T7] Balázs Hidasi. Context-aware preference modeling with factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, RecSys '15, pages 371–374, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3692-5. doi: 10.1145/2792838.2796543. URL `http://doi.acm.org/10.1145/2792838.2796543`.

A szerző további publikációi

[P1] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. Accepted for *International Conference on Learning Representations*, 2016.

[P2] Benjamin Kille, Fabian Abel, Balázs Hidasi and Sahin Albayrak. Using interaction signals for job recommendations. In *Mobile Computing, Applications and Services: 7th International Conference, MobiCASE2015.*, Berlin, Germany, November 12-13, 2015. Revised selected papers.

[P3] Dávid Zibriczky, Balázs Hidasi, Zoltán Petres, and Domonkos Tikk. Personalized recommendation of linear content on interactive tv platforms: beating the cold start and noisy implicit user feedback. In *UMAP Workshops. Proceedings of the International Workshop on TV and Multimedia Personalization*, 2012.

[P4] Balázs Hidasi and Domonkos Tikk. Context-aware item-to-item recommendation within the factorization framework. In *Proceedings of the 3rd Workshop on Context-awareness in Retrieval and Recommendation*, CaRR '13, pages 19–25, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1847-1. doi: 10.1145/2442670.2442675. URL `http://doi.acm.org/10.1145/2442670.2442675`.

[P5] Balázs Hidasi and Domonkos Tikk. Approximate modeling of continuous context in factorization algorithms. In *Proceedings of the 4th Workshop on Context-Awareness in Retrieval and Recommendation*, CARR '14, pages 3–9, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2723-7. doi: 10.1145/2601301.2601303. URL `http://doi.acm.org/10.1145/2601301.2601303`.

[P6] Balázs Hidasi and Csaba Gáspár-Papanek. Shifttree: An interpretable model-based approach for time series classification. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6912 of *Lecture Notes in Computer Science*, pages 48–64. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-23782-9. doi: 10.1007/978-3-642-23783-6_4. URL `http://dx.doi.org/10.1007/978-3-642-23783-6_4`.

[P7] Balázs Hidasi. Újfajta, automatikus, döntési fa alapú adatbányászati módszer idősorok osztályozására. Heszberger Zalán (szerk.), *Proceedings of the 2009. évi Végzős Konferencia.* ISBN 978-963-421-588-2.

[P8] Balázs Hidasi. Modell alapú idősor-osztályozó fejlesztése és kiterjesztése. M.Sc. Thesis, Budapest University of Technology and Economics, Department of Telecommunications and Mediainformatics.

[P9] Balázs Hidasi. Újfajta, automatikus, döntési fa alapú adatbányászati módszer idősorok osztályozására. B.Sc. Thesis, Budapest University of Technology and Economics, Department of Telecommunications and Mediainformatics.

[P10] Balázs Hidasi. Az idősor-osztályozás problémájának megoldása új, döntési fa alapú adatbányászati algoritmussal. XXIX. National Students' Scientific Conference (OTDK), Debrecen, 2009.

Hivatkozások

[1] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In *Recsys'08: ACM Conf. on Recommender Systems*, pages 335–336, 2008.

[2] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[3] R. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *ICDM'07: IEEE Int. Conf. on Data Mining*, pages 43–52, 2007.

[4] J. Bennett and S. Lanning. The Netflix Prize. In *KDD Cup Workshop at SIGKDD'07*, pages 3–6, 2007.

[5] Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, 22(1):143–177, 2004.

[6] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD'08: ACM Int. Conf. on Knowledge Discovery and Data Mining*, pages 426–434, 2008.

[7] G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[8] N. N. Liu, B. Caoand M. Zhao, and Q. Yang. Adapting neighborhood and matrix factorization models for context aware recommendation. In *CAMRa'10: Workshop on Context-Aware Movie Recommendation*, pages 7–13, 2010. ISBN 978-1-4503-0258-6.

[9] P. Lops, M. Gemmis, and G. Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. Springer, 2011.

[10] I. Pilászy and D. Tikk. Recommending new movies: Even a few ratings are more valuable than metadata. In *Recsys'09: ACM Conf. on Recommender Systems*, pages 93–100, 2009. ISBN 978-1-60558-435-5.

[11] I. Pilászy, D. Zibriczky, and D. Tikk. Fast ALS-based matrix factorization for explicit and implicit feedback datasets. In *Recsys'10: ACM Conf. on Recommender Systems*, pages 71–78, 2010. ISBN 978-1-60558-906-0.

[12] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[13] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010. ISBN 0387858199, 9780387858197.

[14] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20*. MIT Press, 2008.

[15] J Ben Schafer, Joseph A Konstan, and John Riedl. E-commerce recommendation applications. In *Applications of Data Mining to Electronic Commerce*, pages 115–153. Springer, 2001.

[16] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Major components of the Gravity recommendation system. *SIGKDD Explor. Newsl.*, 9:80–83, December 2007. ISSN 1931-0145.

[17] G. Takács, I. Pilászy, and D. Tikk. Applications of the conjugate gradient method for implicit feedback collaborative filtering. In *RecSys'11: ACM Conf. on Recommender Systems*, pages 297–300, 2011.