UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Goran Gligorin

Pregled in primerjava sistemov za priporočanje

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Igor Kononenko

Ljubljana, 2011

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01749/2011
Datum: 01.04.2011

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **GORAN GLIGORIN**

Naslov: **PREGLED IN PRIMERJAVA SISTEMOV ZA PRIPOROČANJE**
OVERVIEW AND COMPARISON OF RECOMMENDER SYSTEMS

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Tema naloge je področje algoritmov za izbiranje s sodelovanjem (angl. collaborative filtering). Le-ti so najpogostejša osnova za sisteme za priporočanje (angl. recommender systems). V grobem jih delimo na dve skupini: tiste, ki temeljijo na iskanju najbližjih sosedov, in tiste, ki temeljijo na faktorizaciji matrik. Kandidat naj naredi pregled trenutnega stanja in primerjavo sistemov za priporočanje s poudarkom na algoritmih za izbiranje s sodelovanjem. Predstavi naj značilnosti ter prednosti in slabosti obeh omenjenih pristopov. Praktični del diplomske naloge naj bo sestavljen iz implementacije vsaj dveh algoritmov (po enega iz vsake skupine) in ocenjevanja uspešnosti implementiranih in drugih javno dostopnih algoritmov na umetnih in stvarnih množicah podatkov.

Mentor:

prof. dr. Igor Kononenko

Dekan:

prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

IZJAVA O AVTORSTVU

diplomskega dela

Spodaj podpisani Goran Gligorin,

z vpisno številko 63050035,

sem avtor diplomskega dela z naslovom:

Pregled in primerjava sistemov za priporočanje

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom
  prof. dr. Igorja Kononenka
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki "Dela FRI".

V Ljubljani, dne 24.9.2011 Podpis avtorja:

Zahvala

Zahvaljujem se mentorju prof. dr. Igorju Kononenku za vodenje, vse nasvete in pomoč pri izdelavi diplomske naloge.

Zahvaljujem se Eriku Štrumblju za pomoč od začetka pa vse do konca izdelave diplomskega dela.

Nenazadnje se zahvaljujem tudi svojim staršem, ki so me podpirali in me motivirali v času izdelave diplomskega dela.

Kazalo

Seznam prevedenih izrazov

Recommender system – sistem za priporočanje

Cognitive science – kognitivna znanost

Approximation theory – teorija aproksimacije

Information retrieval – iskanje informacij

Forecasting theories – teorije napovedovanja

Management science – upravljanje

Consumer choice modeling – modeliranje želja kupcev

Item – produkt

Utility function – pomožna funkcija

Povzetek

Področje sistemov za priporočanje se običajno deli v tri kategorije: algoritmi za vsebinsko izbiranje, algoritmi za izbiranje s sodelovanjem in hibridni algoritmi, ki na različne načine združujejo elemente prvih dveh kategorij. Cilj diplomskega dela je bila implementacija dveh algoritmov iz kategorije za izbiranje s sodelovanjem, ki so dandanes najpogostejša osnova za sisteme za priporočanje, ter njuno ovrednotenje. Ta kategorija se nadalje deli v dve skupini: metode na osnovi pomnjenja in metode na osnovi modela. Za vsako izmed teh skupin smo za implementacijo izbrali po enega predstavnika. Iz metod na osnovi pomnjenja smo izbrali metodo, ki temelji na iskanju najbližjih sosedov, iz metod na osnovi modela pa metodo, ki temelji na faktorizaciji matrik. Kot dodatek smo implementirali tudi metodo, ki združuje elemente obeh izbranih metod. Za namene primerljivosti rezultatov smo implementirani tudi dve naivni metodi. Rezultati so pokazali, da je težko implementirati sistem, ki glede na izbrane mere deluje bolje od naivnih metod. Analiza je pokazala, da vzroki za to ležijo v redkosti podatkov, ki so eden izmed glavnih problemov algoritmov za vsebinsko izbiranje. Pričakovano je metoda s faktorizacijo matrik delovala boljše od drugih metod, saj je med drugim namenjena reševanju tega problema. V zaključku so predstavljene še ideje za nadaljnje delo, ki vsebujejo uporabo korekcijskih funkcij in odstranjevanje izvajalcev s premalo ocenami.

Ključne besede: sistem za priporočanje, izbiranje s sodelovanjem, k-najbližjih sosedov, faktorizacija matrike

Abstract

The field of recommender systems is most commonly classified into three main categories: content–based, collaborative filtering and hybrid recommendation algorithms which combine the features from the first two categories. The goal of the thesis was the implementation of two algorithms from the collaborative category, today the most commonly used basis for recommender systems, and their evaluation. Collaborative filtering is further divided into two groups: memory–based and model–based methods. From implementation we chose one algorithm form each group. We chose a neighborhood–based method and a method based on matrix factorization to represent each of the groups respectively. We implemented an extra method that combines the properties of the first two. The results of testing showed that building a recommender system that performs better then naive methods. The analysis showed that the main reasons lie in data sparsity problem, which is one of the main problems collaborative filtering methods face. As expected matrix factorization, which is designed to handle this problem, produced better results than other methods. In the conclusion we present some ideas for further work, which include estimate calibration and excluding unrepresentative artists.

Key words: recommender system, collaborative filtering, k-nearest neighbors, matrix factorization

Poglavje 1

Uvod

Sistemi za priporočanje skušajo priporočiti produkte kot so knjige, filmi, članki, glasba ipd., nekemu točno določenemu uporabniku. Korenine teh sistemov segajo v vede kot so kognitivna znanost (*angl. cognitive science*), teorija aproksimacije (*angl. approximation theory*), iskanje informacij (*angl. information retrieval*), teorije napovedovanja (*angl. forecasting theories*), pa tudi upravljanje (*angl. management science*) in modeliranje želja kupcev (*angl. consumer choice modeling*). Sistemi za priporočanje so se nato sredi 90. let prejšnjega stoletja, ko so se raziskovalci začeli osredotočati na probleme priporočanja, ki se zanašajo izključno na različne strukture ocenjevanja, razvili v samostojno raziskovalno področje. Problem priporočanja je tako najpogosteje formuliran kot problem napovedovanja vrednosti ocen za produkte, ki jih nek uporabnik še ni ocenil. To napovedovanje je navadno osnovano na uporabnikovih ocenah za druge produkte in nekaterih drugih podatkih. Ko je sistem sposoben napovedati vrednosti ocen za še neocenjene produkte, lahko uporabniku priporoča tiste, katerih napovedane ocene so najvišje.

Uporaba sistema za priporočanje je še posebej popularna med ponudniki fizičnih in virtualnih produktov preko interneta. V zadnjih desetih letih so se z vse večjo popularnostjo interneta zelo razširile storitve, ki uporabnikom ponujajo obilico fizičnih in virtualnih produktov. Tako velika izbira pa kaj hitro zmede uporabnike. Ponudniki tako uporabljajo sisteme za priporočanje, da bi uporabnikom ponudili prilagojeno izbiro čim bolj primernih produktov in s tem zvišali zadovoljstvo in lojalnost. Med največjimi uporabniki sistemov za priporočanje sta spletna knjigarna Amazon.com, prikazana na sliki 1.1, in največja svetovna spletna *tržnica* Ebay.com, prikazana na sliki 1.2.

Slika 1.1: Spletna knjigarna Amazon.com.

Slika 1.2: Največja svetovna spletna *tržnica* Ebay.com.

V diplomskem delu si najprej ogledamo področje sistemov za priporočanje. Predstavimo formalen zapis problema priporočanja in prikažemo primer matrike ocen. Sledi opis posameznih pristopov glede na običajno delitev v tri kategorije. Pristopi, ki temeljijo na algoritmih za vsebinsko izbiranje, pristopi, ki temeljijo na algoritmih za izbiranje s sodelovanjem in hibridni pristopi, ki združujejo lastnosti pripadnikov prvih dveh kategorij. V tretjem poglavju opišemo podatke, ki smo jih uporabljali, in domeno, iz katere izhajajo. Opišemo tudi pripravo podatkov za uporabo v implementiranih različicah sistemov za priporočanje.

Četrto poglavje obsega opis implementiranih metod. Začnemo z opisom metode na osnovi pomnjenja – k-najbližjih sosedov. Sledi opis postopkov izgradnje soseske, izračuna podobnosti med dvema uporabnikoma in izračuna ocen ter izgradnje priporočil. Opišemo tudi način uporabe vsebinskih podatkov o uporabniku za izboljšan izračun podobnosti med uporabnikoma. Nadaljujemo z opisom metode na osnovi modela – faktorizacija matrik. Tu opišemo postopek izračuna vektorjev faktorjev in izgradnje modela, ter uporabo letega za izgradnjo priporočil. Na koncu tega poglavja sledi še opis metode k-najbližjih sosedov, ki smo jo podkrepili s faktorizacijo matrik in združuje elemente prvih dveh opisanih metod.

Peto poglavje pričnemo s predstavitvijo postopka testiranja implementiranih sistemov za priporočanje. Nadaljujemo s prikazom rezultatov tako izvedenega testiranja različic implementiranih metode in jih analiziramo. V zaključku podamo še končne ugotovitve in predstavimo ideje za nadaljnjo delo.

Poglavje 2

Pregled področja sistemov za priporočanje

Formalen zapis problemov priporočanja je sestavljen iz množice vseh uporabnikov C in množice produktov, ki se lahko priporočajo, S. Prostor S vseh produktov je lahko zelo velik. Šteje lahko na sto tisoče ali tudi, v nekaterih primerih, milijone produktov. Tudi prostor uporabnikov lahko, pri največjih sistemih, številčno sega med milijone. Definirati je potrebno še funkcijo koristnosti (*angl. utility function*) u, ki meri stopnjo koristnosti produkta s uporabniku c:

$$u : C \times S \to R,$$

kjer je R popolnoma urejena množica (na primer: nenegativna cela ali realna števila v določenem intervalu). Za vsakega uporabnika $c \in C$ je cilj najti tak $s' \in S$, ki maksimizira funkcijo koristnosti u. Formalno:

$$\forall c \in C, s'_c = \arg\max_{s \in S} u(c, s)$$

Pri priporočljivostnih sistemih je koristnost produkta navadno podana z oceno, ki predstavlja, kako všeč oziroma koristen je bil določen produkt nekemu uporabniku. Na primer uporabnik Janez je dal filmu "Harry Potter"oceno 10 (od 10). Koristnost pa ni obvezno ocena, ampak je lahko poljubna funkcija, tudi funkcija profita. Glede na primer uporabe je lahko določena s strani uporabnika (ocena) ali pa izračunana (profit).

Vsak element uporabniškega prostora C je lahko predstavljen s *profilom*, ki vključuje različne uporabniške karakteristike, kot so starost, spol, dohodek ipd. Najpreprostejši profili lahko vsebujejo le en element, kot je ID uporabnika. Podobno je tudi vsak element prostora produktov S določen z množico

	Janez	Anže	Mitja	Miha
AC/DC	3	∅	2	3
Blur	3	1	3	∅
Metallica	5	5	3	5
Andrej Šifrer	∅	5	1	1

Tabela 2.1: Primer matrike ocen sistema za priporočanje glasbenih skupin.

karakteristik. Na primer v aplikaciji, ki priporoča glasbene albume, kjer je S zbirka albumov, je lahko vsak album, poleg svojega ID-ja, predstavljen z izvajalcem, naslovom, žanrom, letnico izida ipd.

Osrednja težava sistemov za priporočanje je v dejstvu, da koristnost u navadno ni definirana za celoten $C \times S$ prostor, ampak le za nek njegov podprostor. u mora biti zaradi tega ekstrapolirana, tako da pokrije celoten $C \times S$ prostor. Pri sistemih za priporočanje je koristnost navadno definirana kot ocena in jo zato v začetku predstavljajo le tiste ocene, ki so jih uporabniki za produkte že podali. Za primer si poglejmo tabelo 2.1 kjer je predstavljena majhna matrika ocen sistema za priporočanje, ki uporabnikom priporoča nove glasbene skupine. Ocene v matriki so celoštevilske, na razponu od 1 do 5. Znak ∅ predstavlja skupino, ki je tisti uporabnik še ni ocenil. Sistem za priporočanje mora biti zmožen napovedati vrednosti ocen za neocenjene skupine in na podlagi teh uporabnikom predlagati neko novo glasbeno skupino.

Ekstrapolacija od znanih k neznanim ocenam je ponavadi dobljena z 1) določitvijo *hevristike*, ki definira funkcijo koristnosti u in empirično potrdi njeno učinkovitost in 2) ocenitvijo funkcije koristnosti u, ki optimizira določene učinkovitostne kriterije, kot je srednja kvadratna napaka (*angl. mean squared error*, s kratico MSE). Enkrat, ko so vrednosti neznanih ocen napovedane, se za dejanska priporočila predmetov uporabnikom uporabijo tisti produkti katerih napovedi ocen so najvišje. Uporabniku se lahko priporoča od enega do N najboljših produktov, kjer je N lahko enako številu vseh produktov, ki jih ta še ni ocenil.

Napovedane vrednosti še neocenjenih produktov se lahko računajo na mnogo različnih načinov. Uporabijo se lahko različne metode strojnega učenja in teorije aproksimacije ter različne hevristične metode. Sisteme za priporočanje navadno razvrščamo glede na pristop, ki ga uporabljajo za napovedovanje vrednosti ocen. Najbolj pogosta je razvrstitev v naslednje tri skupine:

- *Priporočila na podlagi vsebine*: Uporabniku se bodo priporočili produkti, ki so podobni tistim, katere je visoko ocenil v preteklosti.

- *Priporočila na podlagi sodelovanja*: Uporabniku se bodo priporočili produkti, ki so bili v preteklosti visoko ocenjeni s strani uporabnikov, ki imajo podoben okus in preference kot opazovan uporabnik.
- *Hibridni pristopi*: Združujejo lastnosti obeh zgoraj omenjenih skupin sistemov za priporočanje.

2.1 Vsebinsko izbiranje

Pristop vsebinskega izbiranja (*angl. content-based filtering*) temelji na izgradnji profila za vsakega uporabnika ali vsak produkt, s čimer želi sistem označiti njegove karakteristike. Profil glasbenega albuma, na primer, bi lahko vseboval podatke o izvajalcu, naslovu, žanru, številu pesmi, uspešnosti prodaje ipd. Uporabniški profili pa lahko vsebujejo demografske lastnosti ali pa so zgrajeni na podlagi nekega, za določeno domeno izdelanega, vprašalnika. Taki profili sistemom omogočajo iskanje produktov, ki se ujemajo z uporabnikovimi lastnostmi. Uporaba pristopa vsebinskega izbiranja torej zahteva zbiranje zunanjih podatkih, ki pa niso vedno na voljo ali pa je njihovo zbiranje zelo težavno.

Sistem za priporočanje s pristopom vsebinskega izbiranja je bil uspešno uporabljen v projektu Music Genom Project, ki ga uporablja internetni radio Pandora.com (`http://www.pandora.com`) [1]. Izučeni glasbeni analitiki ocenijo vsako pesem s približno 400 *geni*. Vsak tak gen predstavlja neko ločeno glasbeno lastnost, na primer spol glavnega vokalista, vrsto stranskih vokalov, pa tudi nivo popačenja pri električni kitari. S temi geni skušajo, ne le zajeti glasbeno identiteto pesmi, ampak tudi številne lastnosti, ki so pomembne za razumevanje poslušalčevih oziroma uporabnikovih glasbenih želja [3].

Ocenitev koristnosti $u(c, s)$ produkta s za uporabnika c je v sistemih za priporočanje s vsebinskim izbiranjem osnovana na koristnosti $u(c, s_i)$, ki jih je uporabnik c določil produktu s "podobnim"produktom s_i. V aplikaciji, ki priporoča glasbene albume, bi za priporočanje le-teh uporabniku c sistem s vsebinskim izbiranjem poskušal razumeti skupne karakteristike albumov, ki jih je uporabnik c visoko ocenil. Priporočeni bi nato bili le tisti, ki imajo visoko stopnjo podobnosti z ugotovljenimi uporabnikovimi preferencami.

Veliko današnjih sistemov za priporočanje, ki uporabljajo pristop vsebinskega izbiranja, je usmerjenih v priporočanje produktov s tekstovno vsebino, kot so dokumenti, spletne strani ipd. Razlog leži v vedah iz kateri izhajajo sistemi za priporočanje. Taki sta na primer veda o iskanju informacij (*angl. information retrieval*) in veda o izbiranju informacij (*angl. information filtering*).

Izboljšave napram tem vedam izhajajo iz uporabe uporabniških profilov, ki so sestavljeni iz informacij o uporabnikovih okusih, želja, preferencah in potrebah. Te informacije se lahko pridobijo neposredno, npr. preko vprašalnikov, ali pa posredno – so naučene iz uporabnikovega uporabljanja sistema skozi čas.

Naj Vsebina(s) predstavlja profil nekega produkta s. To je torej množica atributov, ki ta produkt označujejo. Ker se metode vsebinskega izbiranja osredotočajo predvsem na tekstovne vsebine, so produkti navadno označeni s *ključnimi besedami*. Tudi uporabniki imajo svoj profil VsebinskiProfil(c), ki je zgrajen z analizo vsebin, ki jih je ta uporabnik v preteklosti videl in ocenil. Funkcijo koristnosti $u(c, s)$ navadno definiramo z:

$$u(c, s) = mera(\text{VsebinskiProfil}(c), \text{Vsebina}(s)),$$

kjer je *mera* neka funkcija podobnosti. Če uporabnik c, na primer, bere veliko člankov o glasbilih, mu bo sistem za priporočanje s vsebinskim izbiranjem predlagal druge članke na temo glasbil, saj bodo le-ti sestavljeni iz več izrazov na temo glasbil, kot so *kitara, bobni, strune, ton ipd.*, v primerjavi s članki o drugih temah. V uporabniškem profilu takega uporabnika bodo zato ti izrazi oziroma ključne besede imele veliko težo.

Poleg tradicionalnih hevristik se za sisteme za priporočanje s vsebinskim izbiranjem uporabljajo tudi druge metode, kot so Bayesov klasifikator, različne tehnike strojnega učenja, vključno z gručenjem, odločitvenimi drevesi in umetnimi nevronskimi mrežami. Te tehnike se od pristopov baziranih na iskanju informacij razlikujejo v tem, da napoved koristnosti ne temelji na hevristični formuli, temveč na modelih naučenih iz podatkov z uporabo statističnega učenja in tehnik strojnega učenja. Tako, na primer, Bayesovi klasifikatorji v praktični uporabi, kljub temu, da predpostavka o medsebojni neodvisnosti ključnih besed vedno ne drži, dosegajo visoko klasifikacijsko točnost [10].

2.1.1 Omejitve in izzivi

Omejena zmožnost analiziranja vsebin Sistemi za priporočanje, ki uporabljajo metode vsebinskega izbiranja so omejeni z značilnostmi, ki so neposredno povezane s produkti, ki jih priporočajo. Za zadostno veliko množico značilnosti morajo biti vsebine v obliki, ki jo lahko računalniki avtomatsko analizirajo ali pa jih je potrebno produktom določiti ročno. Uporaba tehnik za iskanje informacij deluje dobro pri tekstovnih dokumentih, medtem ko imajo lahko nekatere druge domene težave z avtomatskim določanjem značilnosti. To je še posebej značilno za večpredstavnostne oblike podatkov, kot so video posnetki, slike, glasba ipd.

Težave pri omejenih zmožnostih analiziranja vsebin se pojavijo tudi v primerih, ko sta z isto množico značilnosti opisana dva različna produkta. Z vidika sistema za priporočanje sta tako nerazločljiva. Taki sistemi, na primer, niso zmožni razločiti med dobro in slabo napisanima člankoma na neko temo, ker oba od teh člankov uporabljata iste izraze.

Prekomerna specializacija Sistem za priporočanje, ki lahko priporoča *le* produkte, ki rangirajo visoko glede na uporabnikov profil, bo temu uporabniku lahko priporočal le take produkte, ki so podobni tistim, ki jih je uporabnik že ocenil. Uporabnik, ki nikoli ni ocenil članka o astronomiji, tako ne bo dobil priporočila o najboljšem kraju za opazovanje zvezd. Rešitev takega problema je najpogosteje rešena z vnosom določene mere naključja v sistem.

Problem prekomerne specializacije pa ne leži le v nezmožnosti priporočanja produktov, ki so drugačni od že ocenjenih, ampak tudi produktov, ki so si med seboj *preveč* podobni, kot na primer dva članka o istem dogodku. Veliko sistemov za priporočanje zato ne izloča le vsebin, ki so preveč različne od uporabnikovih preferenc, ampak tudi preveč podobne vsebinam, ki jih je le-ta že videl.

Raznolikost priporočil je zato pogosto zaželjena lastnost sistemov za priporočanje. Uporabnik naj bi tako imel na izbiro vrsto možnosti in ne le istolično množico alternativ.

Problem novih uporabnikov Da bi si sistem za priporočanje s vsebinskim izbiranjem lahko izoblikoval dovolj dobro predstavo o uporabniku, njegovih željah in preferencah, mora ta oceniti zadostno količino produktov. Sistemi imajo zato velike težave z natančnimi priporočili novim uporabnikom z majhno množico ocenjenih vsebin.

2.2 Izbiranje s sodelovanjem

Izbiranje s sodelovanjem je drugi pristop za gradnjo sistemov za priporočanje. Za razliko od vsebinskega izbiranja, uporablja, za ocenjevanje uporabnosti produktov, ta pristop tiste produkte, ki so jih ocenili *drugi* uporabniki.

Ocena koristnosti $u(c, s)$ produkta s za uporabnika c je v sistemih izbiranja s sodelovanjem osnovana na koristnostih $u(c_j, s)$, ki so jo produktu s določili uporabniku c "podobni" uporabniki $c_j \in C$. Tako aplikacija, ki priporoča glasbene albume, za gradnjo priporočil uporabniku c, poskušala najti uporabnike, ki imajo podoben okus za glasbo (so ocenili iste glasbene albume s podobno

oceno). Kot priporočila uporabniku c nato poda le tiste albume, ki so bili visoko ocenjeni s strani takih uporabnikov.

Sisteme za priporočanje, ki temeljijo na pristopu izbiranja s sodelovanjem, nadalje delimo v dve skupini:

- metode na osnovi pomnjenja (*angl. memory-based*) in
- metode na osnovi modela (*angl. model-based*).

2.2.1 Metode na osnovi pomnjenja

Metode na osnovi pomnjenja so v svojem bistvu hevristike, ki ustvarijo napovedi na podlagi celotne zbirke, s strani uporabnikov, že ocenjenih produktov. To pomeni, da je vrednost neznane ocene $r_{c,s}$ za uporabnika c in produkt s navadno izračunana kot združek (*angl. aggregate*) ocene drugih (ponavadi N najbolj podobnih) uporabnikov za isti produkt s:

$$r_{c,s} = \operatorname*{aggr}_{c' \in \hat{C}} r_{c',s},$$

kjer je $\hat{C}$ množica N uporabnikov, ki so najbolj podobni uporabniku c in so že ocenili produkt s. N lahko sega od 1 pa vse do števila vseh uporabnikov v sistemu. Nekaj primerov funkcije za izračun združka $aggr$:

$$r_{c,s} = \frac{1}{N} \sum_{c' \in \hat{C}} r_{c',s} \tag{2.1}$$

$$r_{c,s} = k \sum_{c' \in \hat{C}} sim(c, c') \times r_{c',s} \tag{2.2}$$

$$r_{c,s} = \bar{r_c} + k \sum_{c' \in \hat{C}} simc, c' \times (r_{c',s} - \bar{r_{c'}}) \tag{2.3}$$

Normalizacijski faktor k v (2.2) in (2.3) je navadno definiran z:

$$k = 1/\sum_{c' \in \hat{C}} |sim(c, c')|,$$

povprečna ocena $\bar{r_c}$ uporabnika c v (2.3), pa z:

$$\bar{r_c} = (1/|S_c|) \sum_{s \in S_c} r_{c,s},$$

kjer

$$S_c = \{s \in S | r_{c,s} \neq \emptyset\}.$$

Združek je najenostavneje izračunati s povprečjem, kot je to pokazano v (2.1), vendar je najpogostejši pristop z uporabo uteženе vsote, razvidne iz (2.2). Mera podobnosti (*angl. similarity*) med uporabnikoma c in c', označena s $sim(c, c')$, je definirana kot mera razdalje ter se uporablja kot utež. Bolj kot sta si uporabnika c in c' podobna, večjo težo bo imela ocena $r_{c',s}$ pri napovedani vrednosti ocene $r_{c,s}$. Različne aplikacije pa lahko uporabljajo tudi lastno mero podobnosti $sim(c, c')$, vendar morajo z uporabo normalizacijskega faktorja k izračune pravilno normalizirati. To je razvidno iz (2.2). Težava z uteženo vsoto, kot je to v (2.2), je v neupoštevanju, da lahko različni uporabniki različno uporabljajo lestvico za ocenjevanje produktov. (2.3) zato uporablja popravljeno uteženo vsoto, ki se pogosto uporablja za odpravo tega problema. Namesto absolutnih vrednosti ocen, so tu sešteti odmiki od povprečne ocene posameznega uporabnika.

Sarwar je v [11] predlagal, da bi, z uporabo istih mer za podobnost, namesto med *uporabniki* računali podobnosti med *produkti* in pridobili ocene od njih. Zamisel so nadalje razvili v [12] za priporočila najboljših-N produktov. V [11, 12] so tudi empirično pokazali, da produktno-osnovani algoritmi predstavljajo manjšo računsko zahtevnost od tradicionalnih uporabniško-osnovanih, pri čemer zagotavljajo priporočila primerljive ali celo boljše kvalitete kot najboljši algoritmi iz slednje skupine.

2.2.2 Metode na osnovi modela

Za razliko od metod na osnovi pomnjenja, se pri metodah na osnovi modela, zbirka ocen uporablja za, kot nakazuje že samo ime, učenje modela. Ta se nato uporablja pri napovedovanju vrednosti neznanih ocen. [13] predlaga verjetnostni pristop k izbiranju s sodelovanjem, kjer se vrednosti neznanih ocen napovedujejo z:

$$r_{c,s} = E(r_{c,s}) = \sum_{i=0}^{n} i \times Pr(r_{c,s} = i | r_{c,s'}, s' \in S_c)$$

in se predpostavlja, da so ocene cela števila med 0 in n ter je Pr izraz za verjetnost, da bo uporabnik c podal oceno i za produkt s glede na uporabnikove že ocenjene produkte. Za ocenjevanje teh verjetnosti [13] predlaga dva alternativna verjetnostna tipa modelov:

1. *Modeli z gručenjem*: Istomišljenjske uporabnike se združi v razrede. Ob podani pripadnosti uporabnika določenemu razredu, so uporabniške ocene predpostavljeno neodvisne, torej zgradba modela predstavlja naivni Bayesov model. Število razredov in parametri modela so naučeni iz podatkov.

2. *Bayesovske mreže*: Vsak produkt v domeni predstavlja vozlišče v Bayesovski mreži, kjer vrednost vsakega vozlišča predstavlja možne vrednosti ocene za vsak produkt. Tako struktura mreže kot pogojne odvisnosti so naučene iz podatkov.

Ena od slabosti takega pristopa je pripadnost vsakega uporabnika le eni gruči. Nekaterim sistemov za priporočanje pa bi zmožnost, da uporabnik pripada več gručam hkrati, botrovala, saj večino ljudi ne zanima le eno področje. Tako bi lahko na primer nekega uporabnika zanimalo eno področje za potrebe v službi (npr. strojništvo) in neko drugo področje kot vir informacije za kakega od svojih konjičkov (npr. ribolov).

Poleg omenjenih pristopov [14] predlaga metode izbiranja s sodelovanjem v okviru strojnega učenja, kjer se lahko uporabijo različne tehnike le-tega (kot na primer nevronske mreže) skupaj s tehnikami za pridobivanje značilnosti (kot je to singularni razcep (*angl. Singular Value Decomposition*, s kratico SVD)). Tako [13] kot [14] primerjata lastne pristope na osnovi modelov z običajnimi metodami na osnovi pomnjenja in poročata, da v nekateri primerih metode na osnovi modela delujejo boljše kot tiste na osnovi pomnjenja z vidika točnosti priporočil. Vendar pa so primerjave v obeh primerih le empirične in ni podane nobene teoretične podlage, ki bi podpirala te trditve.

Tako kot pri tehnikah vsebinskega izbiranja, je glavna razlika med tehnikami izbiranja s sodelovanjem na osnovi modela in havrističnimi pristopi, v tem, da se način računanja koristnostne funkcije (ocene) osnuje na modelu, zgrajenem na podatkih z uporabo tehnik strojnega učenja in statistike, ne pa na nekem hevrističnem pravilu.

Sistemi za priporočanje, ki uporabljajo čiste (brez elementov metod drugih tipov) metode izbiranja s sodelovanjem, nimajo nekaterih slabosti s katerimi se soočajo tisti iz vsebinskega izbiranja. Tako so sistemi izbiranja s sodelovanjem, zaradi uporabe ocene drugih uporabnikov, so zmožni delovati na kakršnem koli tipu vsebin in lahko priporočajo kakršne koli produkte, tudi tiste, ki niso podobni produktom, ki so jih že obravnavali – so torej neodvisni od domene. Imajo pa sistemi, ki uporabljajo ta pristop, tudi svoje lastne omejitve.

2.2.3 Omejitve in izzivi

Redkost podatkov (*angl. data sparsity*) Komercialni sistem za priporočanje se navadno uporabljajo na bazah podatkov, ki vsebujejo veliko število tako uporabnikov, kot produktov. Ker pa večina uporabnikov oceni le peščico produktov, vsebuje matrika ocen zelo malo podatkov. To zmanjšuje učinkovitost priporočil sistemov izbiranja s sodelovanjem, ki svoja priporočila bazirajo na teh ocenah.

Redkost podatkov se lahko pojavi na več načinov:

- Problem hladnega zagona (*angl. cold start problem*) se pojavi, ko v sistem vstopajo novi uporabniki ali produkti.

- Pokritost (*angl. coverage*) je definirana kot delež produktov za katere lahko sistem poda priporočila. Problem zmanjšane pokritosti (*angl. reduced coverage*) pa se pojavi, ko je število uporabniških priporočil zelo majhno v primerjavi z velikim številom produktov v sistemu.

- Soseska tranzitivnost (*angl. neighbour transitivity*) je problem pri redkih podatkovnih bazah, kjer sistem ni sposoben zaznati podobnih uporabnikov, saj niso ocenili istih produktov.

Za odpravljanje tega problema je bilo predlaganih veliko pristopov. Tehnike za zmanjševanje števila dimenzij, kot je SVD [8], odstranijo nereprezentativne oziroma nepomembne uporabnike ali produkte, da bi neposredno zmanjšali število dimenzij matrike ocen. Hibridni sistemi za priporočanje, kot je sistem izbiranja s sodelovanjem, podkrepljen z vsebinsko informacijo, se lahko izkažejo kot uporabni tako, da z uporabo zunanjega vira informacij proizvedejo priporočila za nove uporabnika ali produkte.

Problem novih uporabnikov Kot pri vsebinskem izbiranju, se tudi pri pristopu izbiranja s sodelovanjem pojavlja problem novih uporabnikov. Da bi lahko sistem za priporočanje zgradil natančna priporočila, se mora najprej naučiti uporabnikovih potreb in želja iz ocen, ki jih je le-ta podal v preteklosti. Večina predlaganih rešitev za ta problem uporablja pristop hibridnih sistemov za priporočanje, opisan v naslednjem razdelku. Ta združuje metode vsebinskega izbiranja z metodami izbiranja s sodelovanjem. Alternativa je tudi pristop, ker se s pomočjo raznih tehnik določi najbolj reprezentativne produkte, ki naj jih oceni nov uporabnik. Te tehnike temeljijo na priljubljenosti in entropiji produktov, poosebljanju uporabnikov ter kombinaciji teh.

Problem novih produktov Novi produkti se neprestano dodajajo v podatkovno bazo sistemov za priporočanje. Ker se sistemi izbiranja s sodelovanjem zanašajo le na ocene, ki jih produktom podajo uporabniki, novih produktov ni možno priporočiti dokler le-teh ni ocenilo zadostno število uporabnikov. Tudi ta problem je, podobno kot problem novih uporabnikov, možno reševati s pomočjo hibridnih sistemov za priporočanje.

Skalabilnost (*angl. scalability*) Ko začne število uporabnikov in produktov močno rasti, začnejo tradicionalni algoritmi izbiranja s sodelovanjem podlegati problemom skalabilnosti. Računska zahtevnost namreč hitro preseže vse sprejemljive in praktične meje.

Tehnike za zmanjševanje števila dimenzij, kot je SVD, znajo obvladovati ta problem in hitro proizvesti kvalitetna priporočila, vendar pa morajo preiti zahtevne korake faktorizacije metrik. Inkrementalne različice SVD se izognejo temu problemu, saj znajo upoštevati nove ocene brez potrebe po preračunavanju nizko–dimenzijskih modelov od začetka, kar naredi take sisteme za priporočanje visoko skalabilne.

Sinonimi Sinonimi se nanašajo na nagnjenja določenega števila istih ali zelo podobnih produktov, da imajo različna imena ali vnose. Večina sistemov za priporočanje je nezmožnih najti tako prikrite povezave in zato te produkte obravnavajo kot različne. Tako sta, na primer, navidezno različna produkta *karta Slovenije* in *zemljevid Slovenije* v bistvu ista, vendar metode izbiranja s sodelovanjem na osnovi pomnjenja te povezave niso zmožne najti in ju obravnavajo ločeno. Prevladovanje sinonimov zato zmanjša učinkovitost sistemov za priporočanje.

Poskusi reševanja problema sinonimov v preteklosti so uporabljali razširjanje izrazov ali gradnjo slovarja. Slaba stran avtomatskih metod je, da imajo nekateri dodani izrazi drugačen pomen od želenega, kar vodi k zmanjšani učinkovitosti.

Tehnike SVD, še posebej metoda *Latent Semantic Indexing (LSI)*, so sposobne obvladovati probleme sinonimov. Metode SVD na podlagi velike matrike povezav izraz–dokument zgradijo semantičen prostor, kjer so izrazi in dokumenti, ki so si med bolj podobni, bližje kot ostali. To omogoča, da razporeditev prostora odraža velike povezovalne vzorce v podatkih, medtem ko manj pomembne ignorira. Učinkovitost LSI pri problemu sinonimov je impresivna, vendar pa daje le delno rešitev za problem večpomenskosti, saj ima lahko večina besed več kot le en izrazit pomen.

Sive ovce (*angl. Gray sheep*) Z izrazom *sive ovce* se označuje uporabnike, katerih mnenja oziroma okusi dosledno ne sovpadajo z nobeno od skupin ljudi in zato ne botrujejo sistemom za priporočanje na osnovi izbiranja s sodelovanjem. *Črne ovce* pa so nasprotne si skupine, katerih posebni okusi naredijo priporočanje skoraj nemogoče. Čeprav so to neuspehi sistemov za priporočanje, imajo s črnimi ovcami težave tudi ne-elektronski priporočevalci, zato so taki neuspehi sprejemljivi.

Claypool [9] je izdelal hibridni pristop, ki združuje vsebinsko izbiranje z izbiranjem s sodelovanjem, tako da osnuje priporočila na uteženem povprečju priporočil iz obeh sistemov. Pri tem pristopu se uteži določijo za vsakega posameznega uporabnika posebej in s tem sistemu omogočajo, da določi optimalno razmerje med priporočil posameznega sistema za vsakega uporabnika.

Shilling napadi Ime izvira iz angleške besede *shill*, ki pomeni *pomagač uličnega prodajalca ali krošnjarja, ki z navideznim nakupom zbudi željo pri gledalcih za nakup.* Najpogosteje se pojavljajo v javno odprtih sistemih, kot so spletne trgovine, kjer si lahko ljudje prosto ustvarijo uporabniške profile in z njimi podajajo ocene za produkte. Tako lahko nek škodoželjni uporabnik ustvari več (navadno lažnih) uporabniških profilov in z njimi poda visoke ocene za eno množico produktov in nizke za neko drugo množico. S tem želi napadalec doseči zvišanje ((*angl. push attack*)) ali znižanje ((*angl. noise attack*)) ocene določenih ciljnih produktov.

Lam in Reidl sta v [4] ugotovila, da imajo Shilling napadi veliko manjši vpliv na produktno–osnovane sisteme z izbiranja s sodelovanjem, kot na uporabniško–osnovane. Nobeni od teh sistemov pa se tem napadom ne morejo popolnoma izogniti, a vendar obstajajo različne delne rešitve [5, 6, 7].

2.3 Hibridne metode

Združevanje karakteristik sistemov za priporočanje s vsebinskim izbiranjem s tistih z izbiranjem s sodelovanjem je način kako se izogniti nekaterim slabostim enega ali drugega tipa sistemov. Za združevanje obeh pristopov obstajajo različne strategije, ki se jih lahko uvrsti v eno od naslednjih skupin:

- ločena implementacija sistemov s vsebinskim izbiranjem in sistemov z izbiranjem s sodelovanjem ter združevanje rezultatov,
- vključitev nekaterih karakteristik pristopa vsebinskega izbiranja v pristop izbiranja s sodelovanjem,

- vključitev nekaterih karakteristik pristopa izbiranjem s sodelovanjem v pristop vsebinskega izbiranja, in
- izgradnja splošnega združevalnega modela, ki vključuje tako karakteristike sistemov z vsebinskim izbiranjem kot tistih z izbiranjem s sodelovanjem.

2.3.1 Združevanje ločenih sistemov za priporočanje

Pri uporabi dveh ločenih sistemov se lahko za pridobitev končnih priporočil uporabi ena od dveh strategij:

- Izhoda obeh sistemov se združita v en skupen izhod z uporabo linearne kombinacije ocen posameznega sistema ali glasovalne sheme.
- Lahko pa se uporabi izhod le enega sistema, kjer se sistem, z uporabo neke mere *kvalitete* priporočil, vsakič posebej odloči za bolj primeren sistem v danem trenutku.

2.3.2 Dodajanje karakteristik pristopa vsebinskega izbiranja v pristop izbiranja s sodelovanjem

Taki sistemi za priporočanje so osnovani na sistemih izbiranja s sodelovanjem, poleg tega pa za vsakega uporabnika hranijo še uporabniški profil in profile za redko ocenjene produkte, kot ga imajo sistemi s vsebinskim izbiranjem. Ti profili se uporabljajo za računanje podobnosti med uporabniki. To jim omogoča premagovanje problemov povezanih z redkostjo podatkov, ki jih imajo čisti sistemi izbiranja s sodelovanjem, saj večino parov uporabnikov ne bo imela ocenjenih zadostno količino pogosto ocenjenih produktov. Dobra lastnost teh profilov je tudi, da lahko sistem, poleg produktov visoko ocenjenih s strani podobnih uporabnikov, priporoča tudi tiste produkte, katerih profili močno sovpadajo z uporabnikovim profilom samim.

2.3.3 Dodajanje karakteristik pristopa izbiranja s sodelovanjem v pristop vsebinskega izbiranja

Najbolj priljubljen pristop v tej skupini hibridnih sistemov je uporaba ene od tehnik za zmanjševanje števila dimenzij na skupini profilov iz vsebinskega izbiranja. To prinaša izboljšano učinkovitost napram navadnim sistemom s vsebinskim izbiranjem.

2.3.4 Izgradnja enega združenega modela sistema za priporočanje

Pristopi združevanja sistemov v tej skupini se med seboj močno razlikujejo, saj se meje med vsebinskim izbiranjem in izbiranjem s sodelovanjem hitro zabrišejo. [15], na primer, predlaga uporabo karakteristik obeh pristopov v gradnji kasifikatorja na osnovi enega pravila (*angl. single rule-based classifier*). V [16, 17] je predlagana uporaba poenotenih verjetnostnih metod za združevanje priporočil vsebinskega izbiranja in izbiranja s sodelovanjem, ki je osnovano na verjetnostni analizi skritih semantik (*angl. probablistic latent semantic analysis*).

Hibridni sistemi za priporočanje so lahko nadgrajeni s tehnikami domenskega znanja, s ciljem povečati natančnost priporočil in odpraviti nekatere omejitve tradicionalnih sistemov za priporočanje. Slaba stran tega je potreba po pridobivanju znanja, ki je dobro znano ozko grlo v veliko aplikacijah strojnega učenja. Take izboljšave so navadno uporabljene na področjih, kjer je pridobivanje tega znanja enostavno.

Poglavje 3

Opis in priprava podatkov

V tem poglavju predstavimo podatke, ki smo jih uporabljali kot vhod v naše sisteme za priporočanje. Začnemo s kratkim opisom domene, spletnega portala Last.fm, čemur sledi opis in statistična analiza podatkov. Na koncu si ogledamo še postopek uporabljen za transformacijo podatkov v obliko primerno za uporabo v pozneje implementiranih sistemih za priporočanje.

3.1 Množica podatkov Last.fm

Last.fm (`http://www.last.fm/`) je spletni portal usmerjen v priporočanje glasbe svojim uporabnikom. Ustanovljen je bil leta 2002 v Veliki Britaniji in je marca 2009 imel 30 milijonov aktivnih uporabnikov [18].

Slika 3.1: Logotip spletnega portala Last.fm.

Za vsakega uporabnika, ki si ustvari račun, Last.fm ustvari oseben glasbeni profil s katerim želi "razumeti"njegov glasben okus. Podatke, iz katerih ta profil zgradi, pridobiva preko dveh virov:

- **Last.fm Radio** – ta funkcija je namenjena plačljivim uporabnikom in jim omogoča poslušanje glasbe neposredno z interneta in

- **The Last.fm Scrobbler** – je majhna računalniška aplikacija, ki, poleg funkcije Last.fm Radio, omogoča beleženje pesmi, ki jih uporabnik posluša z drugimi, s strani Last.fm-a podprtimi, predvajalniki glasbe.

S vsemi tako poslušanimi in zabeleženimi pesmimi Last.fm neprestano posodablja uporabnikov glasbeni profil. Ta ji pomaga ugotoviti, katere pesmi uporabnik največ posluša, katere ima najraje, koliko uporabnik posluša določenega izvajalca skozi čas, kateri drugi uporabniki imajo podoben okus ipd. S pomočjo takega zbiranja podatkov lahko Last.fm vsakemu uporabniku tudi priporoča nove izvajalce, tako da primerja uporabnikova predvajanja s predvajanji drugih uporabnikov po celem svetu. Last.fm je v času pisanja na zabeležil že 43 milijard predvajanj pesmi in navaja dejstvo, da število vztrajno raste [19].

Last.fm poleg zbiranja poslušanj svojih uporabnikov in gradnje priporočil za le-te, ponuja razvijalcem dostop do svoje baze podatkov. Ta funkcionalnost je omogočena preko programskega vmesnika *Last.fm API* [20]. Dostopnih je veliko različnih podatkov od informacij o izvajalcih, albumih, pesmih, uporabnikih, do aktualnih podatkov, ko so najpopularnejši izvajalci.

Oscar Celma je marca 2010 na [21] preko opisanega programskega vmesnika zbral strnjeno množico podatkov, ki jo sestavljata dve glavni datoteki:

- usersha1-artmbid-artname-plays.tsv in
- usersha1-profile.tsv

Prva datoteka predstavlja seznam predvajanj in vsebuje četverice

$$(\text{uporabnik}, \text{MBID}, \text{izvajalec}, \text{število predvajanj}),$$

kjer je:

- **uporabnik** – alfa–numerični niz znakov, ki enolično določajo uporabnika,
- **MBID** – enoličen niz, ki predstavlja izvajalca v MusicBrainz podatkovni bazi izvajalcev (vrednost je lahko tudi prazna),
- **izvajalec** – ime izvajalca in
- **število predvajanj** – število predvajanj nekega izvajalca, ki jih je Last.fm zabeležil za določenega uporabnika.

Druga datoteka vsebuje podatke o uporabnikih samih in je sestavljena iz peteric:

$$(\text{uporabnik}, \text{spol}, \text{starost}, \text{država}, \text{datum registracije}),$$

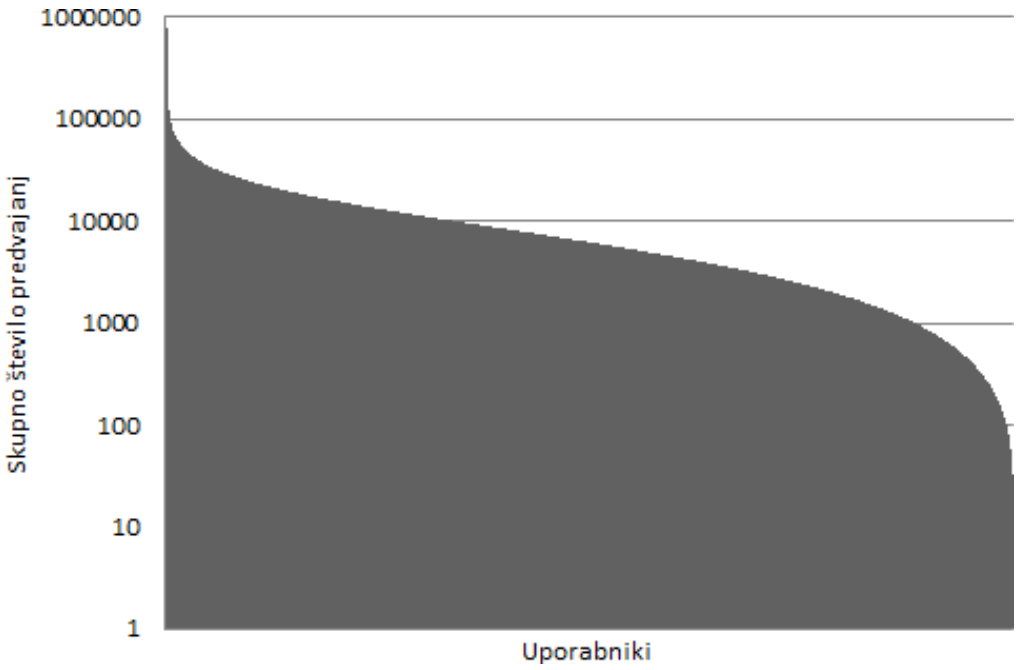
kjer je prva vrednost *uporabnik* povezuje podatke iz prve s podatki iz druge datoteke, druge pa opisujejo lastnostni posameznega uporabnika in so lahko tudi prazne.

Metrika	Vrednost
Število vnosov	17.559.337
Enoličnih uporabnikov	359.347
Izvajalcev	292.475

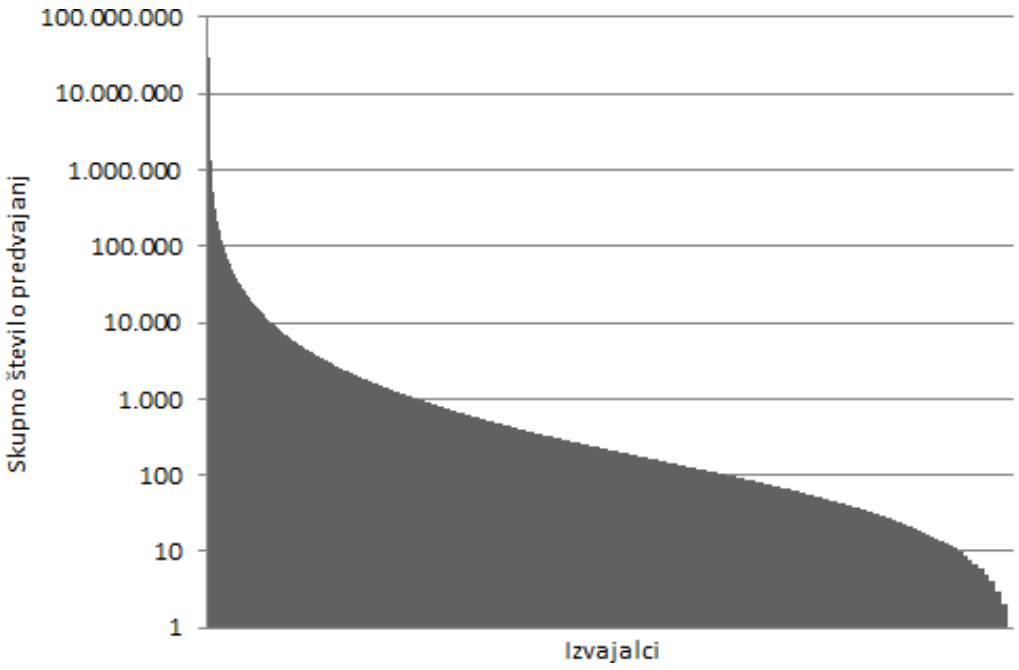
Tabela 3.1: Statistike o bazi podatkov Last.fm, pridobljene iz [21].

Osnovne metrike množice podatkov so prikazane v tabeli 3.1. Za povprečnega uporabnika vsebuje število poslušanj za 48,86 različnih izvajalcev.

Ker bomo te podatke uporabljali v sistemih za priporočanje, si splača ogledat lastnosti matrike ocen, ki jo na podlagi te množice lahko zgradimo. Polna matrika ocen bi za vsakega izmed 359.347 uporabnikov vsebovala število poslušanj za vsakega izmed 292.475 izvajalcev, torej 105.100.013.825 vrednosti. Vendar pa naša množica podatkov le 17.559.337 le-teh, kar predstavlja zgolj 0,167 odstotka. Kot smo že omenili v poglavju 2.2.3 je redkost podatkov eden izmed večjih izzivov s katerimi se sistemi za priporočanje soočajo.



Slika 3.2: Skupno število zabeleženih predvajanj posameznih uporabnikov.



Slika 3.3: Skupno število zabeleženih predvajanj po posameznih izvajalcih.

Slika 3.2 prikazuje graf skupnega števila predvajanj posameznih uporabnikov urejenem v padajočem vrstnem redu. Podobno prikazuje slika 3.3, le da so namesto po uporabnikih števila predvajanj združena po posameznih izvajalcih. Oba grafa zaradi preglednosti uporabljata logaritemsko merilo.

Tabela 3.2 prikazuje 10 najbolj predvajanih izvajalcev. Vsota njihovih predvajanj pa predstavlja kar 4,6 odstotkov vseh predvajanj v bazi podatkov. Nepresenetljivo vsebuje tabela 10 izmed najbolj poznanih izvajalcev na svetovni glasbeni sceni.

	Izvajalec	Skupno število predvajanj
1	The Beatles	30.499.140
2	Radiohead	27.452.124
3	Coldplay	16.701.858
4	Pink Floyd	15.965.959
5	Metallica	15.498.759
6	Muse	15.463.089
7	Nine Inch Nails	14.090.643
8	Red Hot Chili Peppers	13.562.637
9	Linkin Park	12.848.529
10	System of a Down	11.927.204
	skupaj	174.009.942

Tabela 3.2: 10 najbolj predvajanih izvajalcev.

Podobno kot tabela 3.2, vsebuje tabela 3.3 10 uporabnikov, ki so v naši bazi podatkov zabeležili največ predvajanj. Ta so pri uporabnikih malenkost bolje porazdeljena glede na porazdelitev pri izvajalcih. 10 uporabnikov iz omenjene tabele namreč predstavlja 0,12 odstotni delež vseh predvajanj.

3.2 Priprava podatkov za uporabo v sistemih za priporočanje

Pred dejansko uporabo podatkov smo jih morali obdelati in transformirati v obliko, ki jo uporabljajo naše implementacije sistemov za priporočanje.

Korak 1 Na začetku smo odstranili vse *MBID* vnose iz seznama predvajanj, saj jih v implementiranih sistemih ne uporabljamo. Izvoren seznam je vseboval tudi nekaj podvojenih vnosov parov uporabnik–izvajalec. Take vnose smo

	Skupno število predvajanj
1	787.884
2	568.011
3	539.942
4	474.080
5	461.744
6	436.498
7	428.354
8	420.950
9	391.406
10	389.855
skupaj	4.898.724

Tabela 3.3: 10 uporabnikov z največ zabeleženimi predvajanji.

združili v en par in za število predvajanj uporabili vsoto le-teh iz združenih vnosov. Seznam predvajanj je vseboval tudi nekaj nepravilnih vnosov, ki jih je bilo potrebno odstraniti. Izhod tega koraka je bil seznam predvajanj, sestavljen iz enoličnih trojic:

$$(\text{uporabnik}, \text{izvajalec}, \text{število predvajanj}).$$

Korak 2 V drugem koraku smo seznam predvajanj iz prvega koraka preoblikovali v seznam, ki namesto alfa–numeričnih nizov znakov za uporabnike in dejanskih imen izvajalce uporablja indekse. Indeks uporabnika odraža njegovo zaporedno številko iz seznama uporabnikov, indeks izvajalcev pa zaporedno številko iz novo–ustvarjenega seznama izvajalcev. Indeksi namesto dejanskih imen namreč izredno pohitrijo določene operacije, saj jih lahko neposredno uporabljamo za dostopanje podatkov shranjenih v poljih.

Korak 3 Kot prikazujeta grafa 3.2 in 3.3 so števila predvajanj zelo neenakomerno razporejena. V tretjem koraku smo zato seznam predvajanj preslikali v tak seznam, ki vsebuje namesto števila predvajanj celoštevilske ocene z vrednostmi od 1 do 5, s ciljem, da bodo tako dobljene ocene čim enakomerneje zastopane.

Da bi dosegli želeno preslikavo, smo morali najprej združiti vnose seznama predvajanj glede na uporabnika na katerega se nanašajo. Števila predvajanj so bila nato za vsakega uporabnika posebej preslikana v ocene. Algoritem 1

prikazuje preslikavo za enega uporabnika. N_j predstavlja število izvajalcev obravnavanega uporabnika c_i.

Algoritem 1 Preslikava števila predvajanj v ocene za uporabnika c_i.

izvajalce uporabnika c_i uredi padajoče glede na število predvajanj
for $j = 0 \rightarrow N_j - 1$ **do**
  $r_j = 5 - \lfloor j \cdot \frac{5}{N_j - 1} \rfloor$
end for

Idealen uporabnik bi po tej preslikavi imel enako število izvajalcev ocenjenih z vsako oceno od 1 do 5. Ker pa vsak uporabnik nima zabeleženih predvajanj za $5 \cdot n$ izvajalcev, to ne drži povsod.

Korak 4 V četrtem in zadnjem koraku smo podatke še ločiti na učno in testno množico. Za učno smo uporabili 70 odstotkov naključno izbranih uporabnikov, ostalih 30 odstotkov pa smo združili v testno množico. Posamezna ocena iz seznama ocen je nato pripadala tisti množici v katero je bil uvrščen uporabnik na katerega se je le-ta nanašal. Količine podatkov v posameznih množicah so prikazane v tabeli 3.4.

Ocena	Učna množica	Testna množica	Celotna množica
1	2.357.947	1.009.953	3.367.900
2	2.464.564	1.055.386	3.519.950
3	2.457.720	1.052.796	3.510.516
4	2.464.564	1.055.386	3.519.950
5	2.549.081	1.091.940	3.641.021
skupaj	12.293.876	5.265.461	17.559.337

Tabela 3.4: Pogostost pojavljanja posameznih ocen v množici podatkov.

V tabeli 3.5 so prikazani izvajalci z najvišjo povprečno oceno, za katere je v celotni množici več kot 10.000 ocen. Vidimo, da so se, glede na tabelo 3.2, *The Beatles* in še nekateri izvajalci ohranili v seznamu, nekaj pa jih je novih.

	Izvajalec	Število ocen	Povprečna ocena
1	The Beatles	76339	3,722868
2	In Flames	19805	3,655643
3	Radiohead	77347	3,631634
4	Porcupine Tree	11165	3,609763
5	Nine Inch Nails	28982	3,555483
6	Tom Waits	71006	3,548703
7	Bonobo	41689	3,538364
8	Boards of Canada	18397	3,525086
9	Rise Against	14636	3,498838
10	Dream Theater	13898	3,492229

Tabela 3.5: 10 najboljše ocenjenih izvajalcev z več kot 10.000 ocen.

Kot zanimivost si poglejmo še deset najslabše ocenjenih izvajalcev, z več kot 10.000 ocenami, predstavljenih v tabeli 3.6.

	Izvajalec	Število ocen	Povprečna ocena
1	The Postal Service	12292	2,700618
2	The Police	10487	2,705445
3	Nelly Furtado	14945	2,81994
4	Mika	11306	2,841765
5	Justin Timberlake	14374	2,84298
6	Vampire Weekend	10183	2,847098
7	Rage Against the Machine	19946	2,862429
8	Audioslave	11998	2,865227
9	The Cranberries	13674	2,871215
10	Kaiser Chiefs	13733	2,876575

Tabela 3.6: 10 najslabše ocenjenih izvajalcev z več kot 10.000 ocen.

Tako obdelani podatki so sedaj primerni za uporabo in testiranje naših sistemov za priporočanje. Zaradi primerljivosti rezultatov sta v vseh testih uporabljeni isti učna in testna množica.

Poglavje 4

Opis uporabljenih metod

V tem poglavju opišemo različice sistemov za priporočanje s pristopom izbiranja s sodelovanjem, ki smo jih implementirali.

4.1 Naivne metode

Naivne metode sistemov za priporočanje za vsak par uporabnik–izvajalec napovejo enako oceno. Vrednost te ocene se metoda nauči sama iz učnih podatkov z nekim enostavnim algoritmom. Rezultate teh metod smo uporabili kot izhodišče za ocenjevanje kvalitete bolj inteligentnih in zapletenih metod sistemov za priporočanje. Razvili smo dve naivni metodi:

- Metoda, ki vedno napove povprečno oceno:

$$r_{povprecna} = \frac{1}{N} \sum_{i=1}^{N} r_i,$$

  kjer je N število vseh ocen v učni množici in r_i i-ta zaporedna ocena v tej isti množici.

- Metoda, ki vedno napove mediana oceno. Ta je izračunana tako, da se učno množico uredi po vrstnem redu glede na vrednost ocen in se iz tako urejenega seznama vzame ocena na i-tem, kjer $i = \lfloor \frac{N}{2} \rfloor$.

Obe metodi sta se vrednosti ocene, ki jo bosta uporabljali za gradnjo priporočil, naučili z enim samim prehodom skozi učno množico. Naučene vrednosti ocen za našo učno množico so prikazane v tabeli 4.1. Zaradi "namerne" delitve na enako velike dele, sta povprečje in mediana skoraj enaki, zato pričakujemo, da ne bo bistvene razlike med obema rezultatoma.

Metoda	Ocena
Povprečna ocena	3,031094
Mediana ocena	3

Tabela 4.1: Naučene ocene, ki ju napovedujeta naivni metodi.

4.2 k-najbližjih sosedov

Kot prvega izmed dveh tipov metod sistemov za priporočanje z izbiranjem s sodelovanjem, smo v poglavju 2.2.1 opisali metode na osnovi pomnjenja. Med prevladujoče metode tega tipa spadajo metode na osnovi soseske, kamor spada tudi metoda k-najbližjih sosedov.

4.2.1 Izgradnja soseske

Soseska nekega uporabnika je sestavljena iz k njemu najbolj podobnih uporabnikov. Da bi tako sosesko lahko zgradili, moramo najprej določiti funkcijo podobnosti $w_{u,v}$ med dvema izbranima uporabnikoma u in v. Uporabljene funkcije podobnosti temeljijo na uporabi izvajalcev skupnih obema uporabnikoma

$$I = I_u \cap I_v,$$

kjer je I_u množica tistih izvajalcev, za katere je uporabnik u podal oceno. V našem sistemu za priporočanje smo implementirali dve meri podobnosti:

Pearsonov koeficient korelacije Pearsonov koeficient korelacije meri linearno sovpadanje dveh spremenljivk. Podobnost med dvema uporabnikoma u in v izračunamo z:

$$w_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r_u})(r_{v,i} - \bar{r_v})}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r_u})^2} \cdot \sqrt{\sum_{i \in I}(r_{v,i} - \bar{r_v})^2}},$$

kjer je $\bar{r_u}$ povprečna ocena izvajalcev iz množice I uporabnika u.

Kosinusna razdalja Podobnost med dvema uporabnikom lahko merimo tudi tako, da oba predstavimo kot vektorja in izračunamo kosinus kota med njima. Oba vektorja morata biti dolžine, ki je enaka številu izvajalcev, ki sta jih ocenila oba uporabnika. Vektorja zgradimo iz ocen obeh uporabnikov, tako da isto ležeče vrednosti predstavljajo ocene za istega izvajalca v vsakem izmed vektorjev.

Ko imamo tako zgrajena vektorja, podobnost izračunamo z uporabo:

$$w_{u,v} = cos(\vec{u}, \vec{v}) = \frac{\vec{u} * \vec{v}}{||\vec{u}|| \cdot ||\vec{v}||} = \frac{\sum_{i \in I} r_{u,i} r_{v,i}}{\sqrt{\sum_{i \in I} r_{u,i}^2 \cdot \sum_{i \in I} r_{v,i}^2}},$$

kjer "$*$" predstavlja skalarni produkt dveh vektorjev. $\vec{u}$ predstavlja vektor ocen uporabnika u in enako predstavlja $\vec{v}$ vektor ocen uporabnika v.

Ker obe meri uporabljata le izvajalce iz množice I, prihaja do problema, kjer sta si lahko dva uporabnika popolnoma podobna, ker sta enako ocenila le enega ali majhno število istih izvajalcev. Za odpravljanje tega problema obstaja več različnih pristopov opisanih v [23]. Naša implementacija uteži rezultate izračuna podobnosti s faktorjem

$$w_{izvajalci} = \frac{2 \cdot |I|}{|I_u| + |I_v|}.$$

Z eno izmed opisanih metod izračunamo podobnost med obravnavanim uporabnikom in vsemi uporabniki v učni množici. Kot sosesko V nato izberemo k najbolj podobnih uporabnikov.

4.2.2 Izračun ocen in izgradnja priporočil

Sistemi z izbiranjem s sodelovanjem, ki temeljijo na metodah na osnovi pomnjenja, ki uporabljajo sosesko kot osnovo za gradnjo priporočil, uporabljajo izvajalce, ki so jih ocenili obravnavanemu uporabniku podobni uporabniki, izračunani z zgoraj opisanim postopkom, in za katere obravnavani uporabnik še ni podal ocene.

Za vsakega izmed izvajalcev izračunamo oceno, pri čemer lahko uporabimo enega izmed obrazcev za izračun združka opisanih v 2.2:

- **Povprečna ocena**: $r_{ui} = \frac{1}{N} \sum_{v \in V} r_{vi}$
- **Utežena povprečna ocena**: $r_{ui} = k \sum_{v \in V} w_{u,v} \cdot r_{vi}$
- **Prilagojena utežena povprečna ocena**: $r_{ui} = \bar{r_u} + k \sum_{v \in V} w_{u,v} \cdot (r_{vi} - \bar{r_v})$

Tako dobljene pare izvajalec–ocena nato združimo v, po padajočih ocenah, urejen seznam in uporabniku priporočimo najboljših N od teh parov. Na uporabnikovo željo lahko sistem izračuna oceno tudi za izbranega izvajalca. V primeru, ko ta izvajalec ni bil ocenjen s strani uporabnikovih *sosedov*, mu bo sistem dodelil oceno 1.

Kot alternativen način smo razvili tudi lastno metodo združevanja ocen. Ta uporablja seznam izvajalcev v uporabnikovi *soseski* in ga uredi po padajočih povprečnih ocenah. Povprečna ocena za posameznega izvajalca se izračuna preko tistih k-najbližjih sosedov, ki so tega izvajalca že ocenili. Vsakemu izvajalcu se nato določi ocena glede na njegov položaj $i \in \{0, ..., M-1\}$ v tem urejenem seznamu dolžine M:

$$r_i = 5 - i \cdot interval,$$

kjer

$$interval = \frac{4}{M-1}.$$

Uporabniku priporočimo N najboljših izvajalcev tega seznama ali pa mu vrnemo oceno za izbranega izvajalca.

S to metodo smo želeli odpraviti problem, kjer sistem za priporočanje, konstantno napoveduje prenizke ocene. Primer takega sistema je prikazan v tabeli 4.2. Kot vidimo, višja kot je prava ocena, večje je odstopanje. Vsa povprečna odstopanja ocen 2, 3, 4 in 5 so negativna, kar nam pove, da sistem konstantno napoveduje prenizke vrednosti. Cilj opisane metode je tako čim bolj izenačiti napovedi preko vseh ocen.

Ocena	Pristranskost
1	0,2266129
2	-0,7220616
3	-1,680234
4	-2,612558
5	-3,423687

Tabela 4.2: Razčlenjen prikaz povprečne pristranskosti metode za posamezne prave vrednosti ocen.

4.2.3 Uporaba vsebinskih podatkov o uporabnikih

Naši podatki so poleg ocen vsebovali tudi vsebinske podatke o uporabnikih. Razvili smo različice sistemov za priporočanje, ki te podatke uporabljajo pri računanju podobnosti med uporabniki. Vsebinska podobnost med uporabnikoma u in v je izračunana po obrazcu:

$$vsebina_{u,v} = 1 - d_{spol} \cdot w_{spol} - d_{drzava} \cdot w_{drzava} - d_{starost} \cdot w_{starost},$$

kjer

- $d_{spol} = 1$, če sta uporabnika istega spola in 0, če sta različnega ali en od spolov ni podan,
- $d_{drzava} = 1$, če uporabnika prihajata iz iste države in 0, če ne ali en od uporabnikov nima določene države in
- $d_{starost} = \frac{|starost_u - starost_v|}{starost_u + starost_v}$ oziroma 0, če vsaj eden od uporabnikov nima podane starosti,

ter w_{spol}, w_{drzava} in $w_{starost}$ predstavljajo uteži za posamezen vsebinski podatek. Vrednosti uteži so določene tako, da velja

$$w_{spol} + w_{drzava} + w_{starost} = 1.$$

Vsebinsko podobnost $vsebina_{u,v}$ združimo z rezultatom izračuna podobnosti ene izmed metod opisane v poglavju 4.2.1 v novo podobnost:

$$w'_{u,v} = w_{u,v} * w_{ocene} + vsebina_{u,v} * w_{vsebina}.$$

Tudi tukaj so vrednosti uteži določene tako, da velja

$$w_{ocene} + w_{vsebina} = 1.$$

Tako izračunano novo podobnost uporabimo za izračun ocen in izgradnjo priporočil.

4.3 Faktorizacija matrik

Modeli osnovani na prikritih faktorjih (*angl. latent factors*) se uporabljajo kot eden izmed pristopov k sistemom za priporočanje z izbiranjem s sodelovanjem, ki uporabljajo metodo na osnovi modela. Taki modeli skušajo opisati tako uporabnike kot produkte, v našem primeru izvajalce, z določenim številom faktorjev naučenih iz ocenjevalnih vzorcev skritih v podatkih. Nekatere izmed najbolj uspešnih realizacij modelov na osnovi prikritih faktorjev temeljijo na faktorizaciji matrik. Le-ta v osnovi označi uporabnike in izvajalce z vektorjem sestavljenim iz faktorjev.

Modeli osnovani na faktorizaciji matrik preslikajo tako uporabnike kot izvajalce v prostor prikritih faktorjev dimenzije d, tako da so interakcije uporabnik–izvajalec predstavljene kot skalarni produkt v tem prostoru. Vsak uporabnik u je tako predstavljen z vektorjem $\vec{p_u} \in \mathbb{R}^d$ in vsak izvajalec i z vektorjem $\vec{q_i} \in \mathbb{R}^d$. Elementi vektorja $\vec{q_i}$ določajo v koliki meri izvajalec i poseduje ta

faktor. Vrednosti so lahko tako pozitivne kot negativne. Podobno elementi $\vec{p_u}$ opisujejo koliko so uporabniku u všeč izvajalci z visokimi vrednostmi soležnih faktorjev. Skalarni produkt $\vec{q_i}^T\vec{p_u}$ zajame interakcijo med uporabnikom u in izvajalcem i, torej uporabnikov skupen interes za izvajalčeve značilnosti. Ta produkt poda napoved ocene uporabnika u za izvajalca i:

$$r_{u,i} = \vec{q_i}^T\vec{p_u}. \tag{4.1}$$

Največji izziv metode s faktorizacijo matrik je preslikava uporabnikov in izvajalcev v prostor faktorjev. Singularni razcep (*angl. Singular Value Decomposition*, s kratico SVD) je dobro poznana tehnika za razpoznavo prikritih semantičnih faktorjev na področju iskanja informacij. Uporaba singularnega razcepa v sistemih za priporočanje zahteva faktorizacijo matrike ocen, kar pa lahko povzroči težave zaradi velikega števila manjkajočih vrednosti. Običajen singularni razcep namreč ni definiran za nepolne matrike.

Rane verzije so problem redkosti podatkov reševale z zapolnjevanjem manjkajočih vrednosti [24]. Tak pristop zna biti zelo drag, saj bistveno poveča količino podatkov, poleg tega pa lahko netočno zapolnjevanje izkrivi podatke. Novejši pristopi zato za modeliranje uporabljajo zgolj podane ocene in se z regularizacijo skušajo izogniti prekomernemu prilagajanju [25, 26, 27, 28].

4.3.1 Učenje faktorjev

Za učenje faktorjev $\vec{p_u}$ in $\vec{q_i}$ sistem minimizira regulariziran kvadrat napake na učni množici znanih ocen K:

$$\min_{\vec{p_u},\vec{q_i}} \sum_{(u,i)\in K} (r_{ui} - \vec{q_i}^T\vec{p_u})^2 + \lambda(||\vec{q_i}||^2 + ||\vec{p_u}||^2), \tag{4.2}$$

Konstanta λ nadzoruje regularizacijo in je navadno določena s prečnim preverjanjem (*angl. cross-validation*).

Simon Funk je leta 2006 v [25] opisal stohastično gradientno metodo (*angl. stochastic gradient descent*) za optimizacijo 4.2. Algoritem za vsako oceno v učni množici oceni vrednost ocene r_{ui}, izračuna napako

$$e_{ui} = r_{ui} - \vec{q_i}^T\vec{p_u}$$

in popravi vrednosti faktorja, ki se ga trenutno uči, v vektorju $\vec{p_u}$ uporabnika u in vektorju $\vec{q_i}$ izvajalca i:

- $\vec{p_u} \leftarrow \vec{p_u} + \gamma \cdot (e_{ui} \cdot \vec{q_i} - \lambda \cdot \vec{p_u})$

- $\vec{q_i} \leftarrow \vec{q_i} + \gamma \cdot (e_{ui} \cdot \vec{p_u} - \lambda \cdot \vec{q_i})$

Algoritem računa vrednosti istega faktorja vse dokler hitrost upadanja korena srednje kvadratne napake (*angl. Root Mean Square Error*, s kratico RMSE) ne pade pod določeno mejo β:

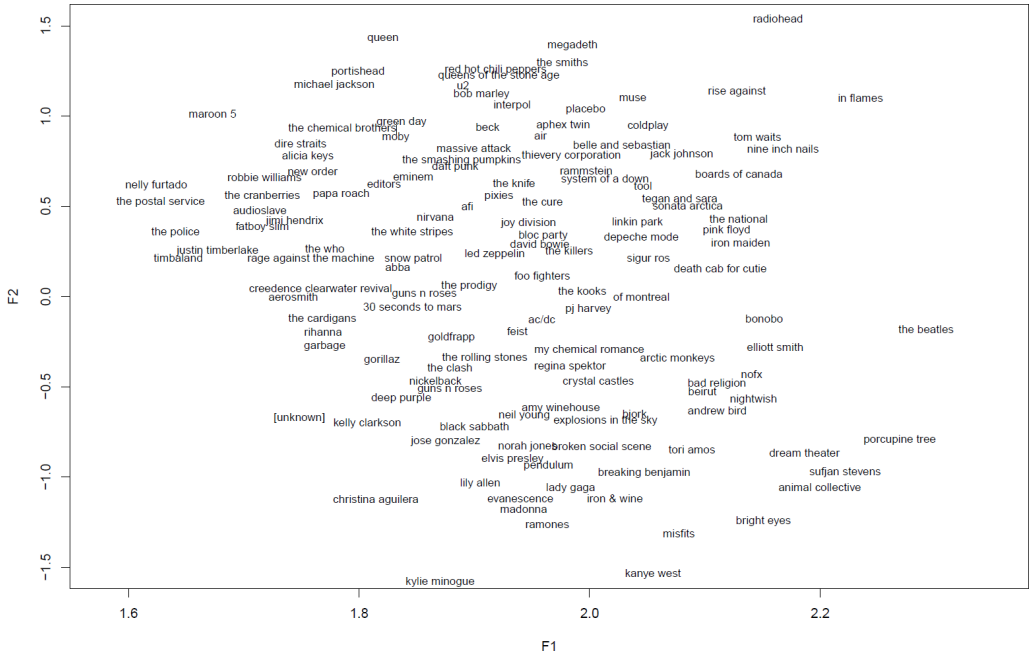
$$RMSE_{upad} = \frac{|RMSE_{trenutni} - RMSE_{prejsnji}|}{RMSE_{trenutni} + RMSE_{prejsnji}} < \beta,$$

oziroma ne doseže zgornje meje števila prehodov skozi učno množico, če mu le-to podamo.

Število ponovitev celotnega postopka je enako izbranemu številu faktorjev f. Rezultat učenja pa sta matriki U in I, katerih vrstice predstavljajo vektorje uporabnikov in izvajalcev, ki predstavljata približek singularnega razcepa matrike ocen R:

$$R \approx UI^T$$

Na grafu 4.1 so izvajalci razporejeni v prostoru glede na vrednosti prvih dveh faktorjev. Iz samega grafa ni povsem razvidno katero značilnost vsak izmed faktorjev opisuje, kar sovpada z njihovim imenom – *prikriti faktorji*.



Slika 4.1: Graf izvajalcev, razporejenih v prostoru glede na vrednosti prvih dveh faktorjev.

4.3.2 Gradnja priporočil

Za izračun napovedi ocene, ki jo bo nek uporabnik u dodelil izvajalcu i, ki ga še ni slišal, mora sistem že "poznati" tega izvajalca. Matrika I modela, zgrajenega v prejšnjem koraku, mora vsebovati vrstico, ki predstavlja izvajalca i. Enaka omejitev ne velja za uporabnike. Če sistem nekega uporabnika še ne pozna, moramo najprej izračunat njegov vektor faktorjev p_u. Vrednost vsakega faktorja je izračunana z obrazcem:

$$\vec{p_u} = \frac{1}{|I_u|} \sum_{i \in I_u} r_u i \cdot \vec{q_i},$$

kjer I_u predstavlja množico izvajalcev, ki jih je uporabnik u že ocenil. Ko imamo vektor faktorjev uporabnika u, izračunamo oceno z že omenjeno enačbo 4.1.

Priporočila, ki jih podamo uporabniku, zgradimo tako, da za vsakega izvajalca i, za katerega imamo predstavitev v matriki I, izračunamo oceno r_{ui}. Seznam priporočil nato uredimo po padajočih vrednostih tako izračunanih ocen in kot priporočila vrnemo najvišje ocenjenih N elementov tega seznama.

4.4 Metoda k-najbližjih sosedov, podkrepljena s faktorizacijo matrik

Faktorizacija matrik lahko nastopa tudi kot osnova za druge metode sistemov za priporočanje z izbiranjem s sodelovanjem, ker predvsem služi za zmanjševanje dimenzionalnosti podatkov in s tem posplošitev le-teh. Kot tretji sistem smo tako implementirali različico metode k-najbližjih sosedov, podkrepljeno s faktorizacijo matrik. Metoda uporablja za izračun podobnosti s Pearsonov koeficient korelacije ali kosinusno razdaljo med uporabnikoma u in v njuna vektorja faktorjev, kjer je v opisan z enim izmed vektorjev matrike I dobljene pri faktorizaciji matrike ocen. Nadaljnji postopek je nato identičen tistemu opisanemu v poglavju 4.2.

Poglavje 5

Rezultati in analiza

V tem poglavje najprej opišemo postopek testiranja implementiranih metod sistemov za priporočanje z izbiranjem s sodelovanjem. Sledi predstavitev rezultatov naivnih metod in osnovnih različic inteligentnih. Podrobneje si ogledamo tudi vpliv različnih parametrov na implementirane sistem. Vse rezultate tudi analiziramo.

5.1 Postopek testiranja

Za merjenje uspešnosti implementiranih sistemov za priporočanje z izbiranjem s sodelovanjem smo uporabili srednjo absolutno napako (*angl. mean absolute error*, s kratico MAE), izmerili pa smo tudi pristranskost metod. Testiranje posameznih metod in njihovih različic smo izvedli na dva načina:

Celoten prehod skozi testno množico Ta pristop smo uporabili pri naivnih metodah in osnovni metodi faktorizacije matrik. Izbranemu uporabniku iz testne množice smo odstranili izvajalca iz njegovega seznama ocen in od sistem za priporočanje zanj pridobili oceno r'. Napako za odstranjenega izvajalca smo izračunali po obrazcu:

$$e = r - r',$$

kjer r predstavlja izvirno oceno, ki jo je obravnavani uporabnik podal temu izvajalcu. Izvajalca smo nato vrnili uporabniku in postopek ponovili za vse izvajalce na njegovemu seznamu ocen in nato še za vse uporabnike v testni množici. MAE metode je tako povprečna vrednost absolutne napake:

$$MAE = \frac{1}{N} \cdot \sum_{i=1}^{N} |e_i|,$$

kjer je N število parov uporabnik–izvajalec in e_i napaka i-tega zaporednega testiranega para. Pristranskost metode pa je povprečna napaka:

$$Pristranskost = \frac{1}{N} \cdot \sum_{i=1}^{N} e_i.$$

Delni prehod skozi testno množico Zaradi časovne zahtevnosti obeh metod, ki temeljita na k-najbližjih sosedov, prej opisanega načina testiranja v praksi ni možno izvesti. Te metode smo zato testirali z delnim prehodom skozi testno množico, tako da dobimo dobre približke dejanskega MAE in pristranskosti. Postopek celotnega prehoda smo zato spremenili tako, da smo za testiranje uporabili 10.000 naključno izbranih parov uporabnik–izvajalec iz testne množice ocen. Uporabniku, na katerega se izbrani par nanaša, smo odstranili izvajalca iz izbranega para in izračunali napako para. Odstranjenega izvajalca smo nato vrnili uporabniku. Na koncu pa smo, enako kot pri celotnem prehodu izračunali MAE in pristranskost metode. Izbrano število tako testiranih parov nam zagotavlja, da so rezultati pravilni z največjim odstopanjem ± 0.025.

5.2 Testiranje in analiza osnovnih različic implementiranih metod

Osnovne različice inteligentnih metod uporabljajo parametre:

- **Metoda k-najbližjih sosedov**:
  - $k = 100$,
  - funkcija podobnosti uporablja Pearsonov koeficient korelacije,
  - izračun združka z metodo povprečne ocene in
  - brez uporabe vsebinskih podatkov o uporabnikih.
- **Metoda s faktorizacijo matrik**:
  - $d = 7$,
  - $\gamma = 0,0015$,
  - $\lambda = 0,02$,
  - $\beta = 10^{-5}$ in
  - meje prehodov so 120 in 200

- **Metoda k-najbližjih sosedov s faktorizacijo matrik** uporablja enake parametre kot metoda k-najbližjih sosedov in model zgrajen z enakimi parametri, ki jih uporablja metoda s faktorizacijo matrik.

Metoda	MAE	Pristranskost
Naivna metoda s povprečno oceno	$1,204974$	$-4,74 \cdot 10^{-5}$
Naivna metoda z mediana oceno	$1,199241$	$-0,03114143$
k-najbližjih sosedov	$1,388041$	$-0,354463$
Faktorizacija matrik	$1,217402$	$0,01136808$
k-najbližjih sosedov s faktorizacijo matrik	$1,591143$	$-0,8420151$

Tabela 5.1: Rezultati naivnih metod in osnovnih različic implementiranih metod sistemov za priporočanje z izbiranjem s sodelovanjem.

Rezultati obeh naivnih metod in osnovnih različic inteligentnih metod so prikazani v tabeli 5.1. Kot smo pričakovali, sta rezultata obeh naivnih metod skoraj enaka.

Vidimo, da se nobena od inteligentnih metod ni obnesla bolje od naivnih metod, zato smo morali najprej preveriti korelacijo med dejanskimi vrednostmi ocen in tistimi, ki jih naše metode napovedujejo. Korelacijski koeficient K in 95% interval zaupanja za vsako izmed teh metod sta prikazana v tabeli 5.2.

Metoda	K	95% int. zaup.
k-najbližjih sosedov	$0,1352986$	$0,1160159 - 0,1544794$
Faktorizacija matrik	$0,1540673$	$0,1532333 - 0,1549011$
k-najbližjih sosedov s fak. matrik	$0,1033526$	$0,089488 - 0,1226781$

Tabela 5.2: Korelacijski koeficient K in 95% interval zaupanja za osnovne različice inteligentnih metod.

Glede na podatke iz te table lahko z veliko verjetnostjo trdimo, da prava korelacija nobene izmed inteligentnih metod ni 0. Ker za vse velja $K > 0$, vemo, da so se naše metode *naučile* nekaterih zakonitosti iz podatkov, zato moramo vzroke za slabše rezultate, kot smo jih dobili pri naivnih metodah, iskati drugje.

Glavnega izmed teh vzrokov lahko najdemo v redkosti podatkov v matriki ocen in v dokaj enakem številu uporabnikov in izvajalcev. Oba razloga zmanjšujeta učinkovitost sistemov za priporočanje. Zaradi tega je tudi

razumljivo, da smo najboljši rezultat dosegli z metodo s faktorizacijo matrik, katere ena izmed poglavitnih značilnosti je usmerjenost v reševanje prav problema redkosti podatkov.

Največje presenečenje je opazna razlika med rezultatoma metode k-najbližjih sosedov in metode k-najbližjih sosedov, podkrepljene s faktorizacijo matrik. Pričakovali bi, da bo uporaba faktorizacije matrik izboljšala rezultate glede na osnovno različico, vendar nam rezultati povejo, da temu ni tako. Razloge za to gre iskati na v enem ali obeh izmed dveh vzrokov:

- **Izračun podobnosti** – izračun podobnosti med uporabniki z uporabo vektorjev faktorjev se lahko izraža v soseski, ki slabše opisuje *okus* obravnavanega uporabnika.

- **Izračun vektorja faktorjev za novega uporabnika** – izračun vektorja faktorjev za novega uporabnika je ključnega pomena za izgradnjo soseske, ki dobro opiše *okus* obravnavanega uporabnika.

V nadaljevanju si bomo pogledali, če je možno dobljene rezultate izboljšati s spreminjanjem uporabljenih parametrov.

5.3 Podrobnejša analiza metod k-najbližjih sosedov

Podrobnejše analize sistemov za priporočanje z izbiranjem s sodelovanjem, bomo pričeli s testiranjem in analiziranjem metod k-najbližjih sosedov. Obravnavali bomo tri glavne gradnike te metode: velikost soseske k, funkcija izračuna podobnosti med uporabniki in metoda za izračun ocen. Za vsakega izmed parametrov bomo vzeli nekaj različnih vrednosti in analizirali rezultate. Vrednosti ostalih gradnikov, pa bodo ostale enake, kot pri osnovi različici metode. Nato bomo najboljše rezultate za posamezne parametre skušali še izboljšati z uporabo vsebinskih podatkov o uporabnikih. Na koncu bomo testirali in analizirali, kako se obnese metoda k-najbližjih sosedov, če uporabimo parametre, ki so privedli do najboljših rezultatov.

Velikost soseske k Za testiranje smo uporabili vrednosti $k = 20$, 50, 100, 150 in 200, dodatno pa še $k = 500$ in 1000. Rezultati so prikazani v tabeli 5.3.

Velikost soseske	MAE	Pristranskost
$k = 20$	$1,529920$	$-0,6999374$
$k = 50$	$1,419502$	$-0,5015191$
$k = 100$	$1,388041$	$-0,3544630$
$k = 150$	$1,356808$	$-0,2969992$
$k = 200$	$1,327428$	$-0,2312076$
$k = 500$	$1,277412$	$-0,1280777$
$k = 1000$	$1,250307$	$-0,06573894$

Tabela 5.3: Rezultati metode k-najbližjih sosedov glede na različne velikosti soseske.

Vidimo, da se z večanjem soseske izboljšujejo rezultati, a so ti dokaj slabši od rezultatov naivnih metod. Glavni vzrok za to je razviden iz vrednosti pristranskosti, ki je vedno manjša od 0, a se ji z povečevanjem soseske približuje. Negativna pristranskost nam pove, da sistem v povprečju napoveduje prenizke ocene. Pri metodi k-najbližjih sosedov je glavni vir takih ocen pogostost dogodka, da sistem ni našel določenega izvajalca v uporabnikovi soseski in mu zato dodeli oceno 1. Z večanjem soseske se ta pogostost zmanjšuje in se MAE zmanjšuje in pristranskost približuje 0, a se povečujeta časa izračuna soseske in časa izračuna združka ocen.

Rezultata pri $k = 500$ in $k = 1000$ nam pokažeta trend izboljševanja rezultatov s povečevanjem soseske, vendar sta časovno že zelo potratna in zato neuporabna za praktično uporabo v velikih sistemih z milijoni uporabnikov in izvajalcev.

Funkcija podobnosti Za testiranje smo uporabili funkciji Pearsonov koeficient korelacije in kosinusna razdalja. Rezultati so prikazani v tabeli 5.4.

Funkcija podobnosti	MAE	Pristranskost
Pearsonov koeficient korelacije	$1,388041$	$-0,3544630$
Kosinusna razdalja	$1,353580$	$-0,3635862$

Tabela 5.4: Rezultati metode k-najbližjih sosedov glede na različne funkcije za izračun podobnosti med uporabniki.

Vidimo, da je uporaba kosinusne razdalje malenkost bolj primerna na naši množici podatkov, a pri naši metodi testiranja razlike manjše od 0.05 ne

moremo sprejeti kot statistično značilnih. Ker je kosinusna razdalja računsko manj zahtevna, tretiramo le-to kot boljšo.

Metoda izračuna ocen Za testiranje smo uporabili metode povprečna ocena, utežena povprečan ocena, prilagojena utežena povprečna ocena in naša metodo, ki smo jo poimenovali *linearno upadanje*. Rezultati so prikazani v tabeli 5.5.

Metoda za izračun ocene	MAE	Pristranskost
Povprečna ocena	$1,388041$	$-0,3544630$
Utežena povprečna ocena	$1,381835$	$-0,3456345$
Prilagojena utežena povprečna ocena	$1,395690$	$-0,3662754$
Linearno upadanje	$1,384898$	$-0,1597139$

Tabela 5.5: Rezultati metode k-najbližjih sosedov glede na različne metode izračuna ocen.

Vidimo, da je najboljši rezultat dosegla metoda z uteženo povprečno oceno. Po pričakovanjih je upoštevanje podobnosti med uporabniki, kot uteži za izračuna ocene s to metodo, privedlo do rahlo boljših priporočil.

Slabše rezultate prilagojene utežene povprečen ocene gre iskati v naši pripravi podatkov za testiranje. Ta metoda je namenjena izravnavanju različnih načinov, ki jih resnični uporabniki uporabljajo za dodeljevanje ocen. Ker so naše ocene umetno preračunane iz števila predvajanj, ta metoda v bistvu škoduje pri izračunu napovedi ocene.

Metoda linearno upadanje je delovala po pričakovanjih, kar je vidno, če primerjamo vrednosti pristranskosti, ki so več kot dvakratne kot pri ostalih metodah za izračun ocene.

Uporaba vsebinskih podatkov o uporabnikih Za testiranje smo tri metode, ki so se najbolje obnesle pri testiranju po posameznih gradnikih:

1. **Metoda 1**: $k = 200$, Pearsonov koeficient korelacije, Povprečna ocena
2. **Metoda 2**: $k = 100$, Kosinusna razdalja, Povprečna ocena
3. **Metoda 3**: $k = 100$, Pearsonov koeficient korelacije, Utežena povprečna ocena

Te metode smo vse podkrepili z uporabo vsebinskih podatkov o uporabnikih. Za uteži vsebinskih podatkov smo uporabili na enake vrednosti $w_{spol} = w_{drzava} = w_{starost} = \frac{1}{3}$. Tudi prispevke funkcije podobnosti in podobnosti uporabnikov glede na vsebinske podatke smo utežili z istima utežema $w_{ocene} = w_{vsebina} = \frac{1}{2}$. Rezultati so prikazani v tabeli 5.6.

Metoda	MAE	Pristranskost
Metoda 1	$1,314574$	$-0,2192154$
Metoda 2	$1,349001$	$-0,3252518$
Metoda 3	$1,362635$	$-0,2889099$

Tabela 5.6: Rezultati metode k-najbližjih sosedov z uporabo vsebinskih podatkov o uporabnikih.

Uporaba vsebinskih podatkov rahlo izboljša rezultate vseh treh metod, katerih parametre smo uporabili kot osnovo. Rezultate metod bi lahko potencialno tudi izboljšali s kalibracijo uteži posameznih elementov vsebinskih podatkov ter razmerja med prispevkoma funkcije podobnosti in podobnosti uporabnikov glede na vsebinske podatke.

Najboljši parametri Testirali smo še metodo, ki uporablja vrednosti parametrov metode k-najbližjih sosedov, ki so se najboljše obnesle:

- $k = 200$,
- funkcija podobnosti uporablja kosinusno razdaljo in
- izračun združka z metodo utežene povprečne ocene.

Te parametre smo testirali brez (metoda 1) in z (metoda 2) uporabo vsebinskih podatkov z enakimi utežmi, ko smo jih opisali v prejšnjem odseku. Rezultati so prikazani v tabeli 5.7.

Metoda	MAE	Pristranskost
Metoda 1	$1,287527$	$-0,2570109$
Metoda 2	$1,323881$	$-0,2327774$

Tabela 5.7: Rezultati metode k-najbližjih sosedov z uporabo parametrov, ki so se najbolje obnesli.

Metoda 1 je pričakovano izboljšala rezultate vseh do sedaj testiranih različic metode k-najbližjih sosedov. Presenetljivo pa uporaba vsebinskih podatkov v metodi 2 ni še nadalje izboljšala rezultatov sistema, kot se je to pokazalo pri vseh različicah v prejšnjem odseku. Vzorke gre iskati v utežeh, ki smo jih uporabili za vsebinske podatke, še posebej pa razmerju prispevkov funkcije podobnosti in podobnosti uporabnikov glede na vsebinske podatke. To razmerje je očitno potrebno nekoliko popraviti v prid prispevka funkcije podobnosti.

5.4 Podrobnejša analiza metod s faktorizacijo matrik

V podrobnejši analizi metod s faktorizacijo matrik si bomo natančneje ogledali tri najpomembnejše parametre za gradnjo modelov: število faktorjev d, hitrost učenja γ in regularizacija λ. Za vsak parameter bomo, podobno kot pri podrobnejši analizi metode k-najbližjih sosedov, uporabili nekaj različnih vrednosti in analizirali rezultate. Tudi tukaj bodo vrednosti ostalih parametrov ostale enake kot smo jih navedli pri osnovni različici metode.

Številko faktorjev d Za testiranje smo uporabili vrednosti $d = 5$, 7, 10 in 20. Rezultati so predstavljeni v tabeli 5.8.

Število faktorjv	MAE	Pristranskost
$d = 5$	$1,218489$	$-0,004056323$
$d = 7$	$1,217402$	$0,01136808$
$d = 10$	$1,216559$	$0,02511892$
$d = 20$	$1,216690$	$0,04476978$

Tabela 5.8: Rezultati metode s faktorizacijo matrik glede na različno število faktorjev d.

Rezultati vseh različic metod s faktorizacijo matrik so zelo blizu rezultatom naivnih metod. Kot vidimo se je glede na število faktorjev najbolje obnesla različica $d = 10$. Različica z $d = 20$ pa že kaže znake prekomernega prilagajanja učni množici, kar je pogost problem te metode. Idealno število faktorjev tako leži med tem dvema vrednostma.

Hitrost učenja γ Za testiranje smo uporabili vrednosti $\gamma = 0{,}0005$, 0,001, 0,0015, 0,002 in 0,0025. Rezultati so predstavljeni v tabeli 5.9.

Hitrost učenja	MAE	Pristranskost
$\gamma = 0,0005$	$1,503249$	$1,112318$
$\gamma = 0,001$	$1,244066$	$0,3278051$
$\gamma = 0,0015$	$1,217402$	$0,01136808$
$\gamma = 0,002$	$1,219614$	$-0,1222541$
$\gamma = 0,0025$	$1,221825$	$-0,1601878$

Tabela 5.9: Rezultati metode s faktorizacijo matrik glede na različne vrednosti hitrosti učenja γ.

Najprimernejša hitrost učenja za naše podatke je, kot vidimo, $\gamma = 0,0015$. Metoda, ki ima hitrost učenja vrednost $\gamma = 0,0005$, ki ima bistveno slabše rezultate, kot ostale različice, se vrednosti faktorjev uči prepočasi, saj jo prekine zgornja meja števila prehoda skozi učno množico za faktor. Za višje vrednosti, $\gamma = 0,002$ in $\gamma = 0,0025$, se tudi pri tem modelu kažejo znaki prekomernega prilagajanja učni množici.

Regularizacija λ Za testiranje smo uporabili vrednosti $\lambda = 0$ oziroma učenje brez regularizacije, 0,01, 0,02, 0,03 in 0,04. Rezultati so predstavljeni v tabeli 5.10.

Regularizacija	MAE	Pristranskost
$\lambda = 0$	$1,452372$	$1,019597$
$\lambda = 0,01$	$1,235929$	$0,2758972$
$\lambda = 0,02$	$1,217402$	$0,011368$
$\lambda = 0,03$	$1,218413$	$0,1007083$
$\lambda = 0,04$	$1,220388$	$-0,1489536$

Tabela 5.10: Rezultati metode s faktorizacijo matrik glede na različne vrednosti regularizacije λ.

Najboljše rezultate smo dobili pri regularizaciji $\lambda = 0,02$. Tako nižje kot višje vrednosti kažejo prekomerno prilagajanje učnim podatkom.

Najboljši parametri Najboljše se je obnesla različica metode s faktorizacijo matrik, kjer smo uporabili parametre:

- $d = 10$,
- $\gamma = 0,0015$ in
- $\lambda = 0,02$.

To različico smo uporabili že pri testiranju različnih vrednosti parametra d. Za nadaljnje izboljšave metod tega tipa je potrebno pogledati tudi parametre β ter spodnjo in zgornjo meja prehodov skozi testno množico za posamezen faktor, ki določajo kdaj sistem preneha z učenjem trenutnega faktorja in vse te še bolje kalibrirati.

Za natančnejšo analizo te metode si poglejmo še tablo 5.11 v kateri so prikazani rezultati glede na posamezno vrednost prave ocene.

Ocena	MAE	Pristranskost
1	1,961981	1,961981
2	1,033041	0,9964182
3	0,3707025	0,03567974
4	0,9183978	-0,9134018
5	1,808203	-1,808203

Tabela 5.11: Razčlenjen prikaz MAE in pristranskosti metode s faktorizacijo matrik za posamezne prave vrednosti ocen.

Vidimo, da se tako napaka in absolutna vrednost pristranskosti povečujeta čim bolj je prava ocena oddaljena od srednje ocene 3. Če pogledamo podrobneje, vidimo da je pristranskost za oceni 1 in 2 pozitivna, za oceni 4 in 5 pa negativna. To pomeni, da metoda večino ocen napove v ožjem intervalu, kot je interval vseh ocen.

5.5 Podrobnejša analiza metod k-najbližjih sosedov, podkrepljenih s faktorizacijo matrik

Za testiranje metod k-najbližjih sosedov, ki so podkrepljene s faktorizacijo matrik, smo uporabili tri metode, ki so se najbolje obnesle pri testiranju po posameznih parametrih navadnih metod k-najbližjih sosedov:

1. **Metoda 1**: $k = 200$, Pearsonov koeficient korelacije, Povprečna ocena
2. **Metoda 2**: $k = 100$, Kosinusna razdalja, Povprečna ocena
3. **Metoda 3**: $k = 100$, Pearsonov koeficient korelacije, Utežena povprečna ocena

Model, ki smo ga uporabili kot osnovo te metode, je model, ki se je najboljše odrezal pri testiranju metod s faktorizacijo matrik in je zgrajen s parametri:

- $d = 10$,
- $\gamma = 0,0015$ in
- $\lambda = 0,02$.

Rezultati so predstavljeni v tabeli 5.12.

Metoda	MAE	Pristranskost
Metoda 1	$1,493266$	$-0,6440307$
Metoda 2	$1,541864$	$-0,7531711$
Metoda 3	$1,56172$	$-0,8132488$

Tabela 5.12: Rezultati metode k-najbližjih sosedov glede na različne metode izračuna ocen.

Podobno kot pri analizi osnovnih različic vseh metod, tudi tu vidimo, da so rezultati te metode slabši, kot rezultati navadnih različic metod k-najbližjih sosedov. Vzroki za to so enaki, kot smo jih opisali že v 5.2, ko smo analizirali rezultate osnovne različice te metode.

Poglavje 6

Zaključek

V diplomskem delu smo predstavili in implementirali metode, ki so sposobne uporabniku zgraditi priporočila na podlagi izvajalcev, ki jih ta uporabnik že pozna. Ugotovili smo, da je izgradnja sistema za priporočanje z visoko učinkovitostjo težak problem. Čeprav nobena izmed inteligentnih metod ni izboljšala rezultatov, ki sta jih postavili naivni, pa imajo prve eno veliko prednost. Priporočila, ki jih zgradijo te metode, je možno urediti po padajoči vrednosti napovedane ocene. To nam omogoča uporabniku podati najboljših N izvajalcev, kar je tudi običajna naloga sistemov za priporočanje. Poleg tega pa sistemi, ki se uporabljajo v realnem svetu, vključujejo veliko dodatnih "trikov" in "zvijač" za izboljšavo priporočil.

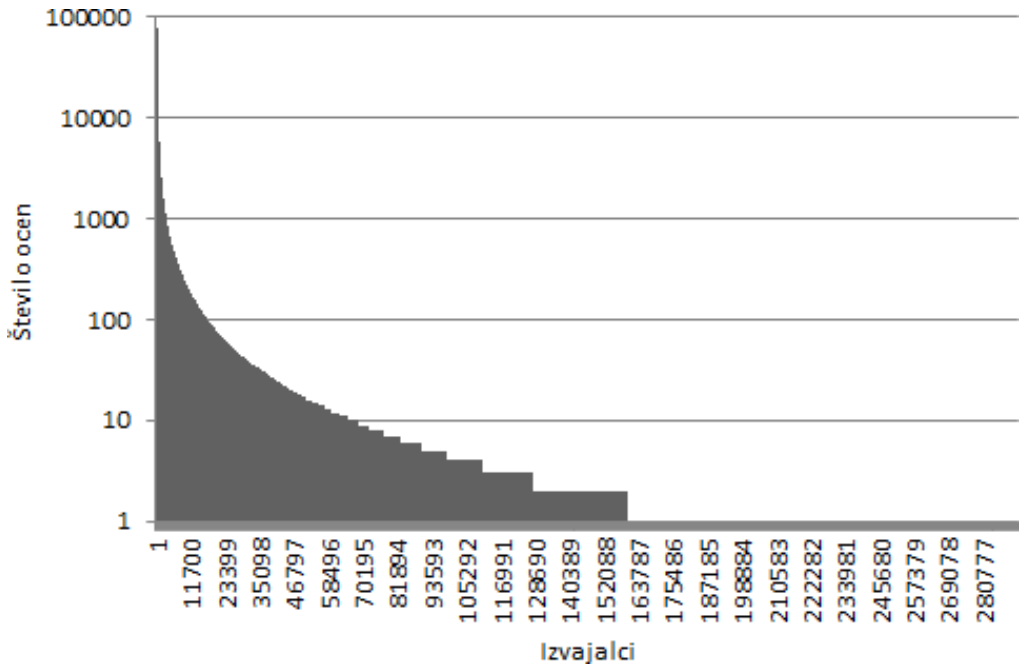
6.1 Ideje za nadaljnje delo

V tem podpoglavju predstavimo dve ideji za nadaljnje delo, ki imata lahko pozitivne učinke na učinkovitost inteligentnih metod sistemov za priporočanje z izbiranjem s sodelovanjem, ki smo jih implementirali.

6.1.1 Odstranitev izvajalcev s premalo ocenami

Pri opisu smo omenili, da so naše implementirane inteligentne metode podlegle problemu približno enakemu številu uporabnikov in izvajalcev. Podobno kot slika 3.3 v poglavju 3.1, prikaže slika 6.1 histogram števila ocen po posameznih izvajalcih.

Kot vidimo, ima veliko število izvajalcev majhen delež ocen. Take izvajalce bi lahko, z določitvijo neke spodnje meje števila ocen, ob pripravi podatkov



Slika 6.1: Število ocen po posameznih izvajalcih.

odstranili. S takim posegom želimo v množici podatkov ohraniti le tiste izvajalce, ki s svojimi ocenami doprinesejo dovolj informacije za učenje naših metod. Tako bi povečali razmerje med uporabniki in izvajalci v prid uporabnikov.

Slaba stran tega posega je v tem, da se problem razmerja med uporabniki in izvajalci sedaj delno spremeni v problem hladnega zagona, opisanega v poglavju 2.2.3, za izvajalce, ki smo jih odstranili. Teh namreč ne bomo mogli vključiti v priporočila uporabnikom in jim napovedati ocene, dokler njihovo število ocen ne bo doseglo spodnje meje.

6.1.2 Kalibracija ocen

V podrobnejši analizi najboljše metode s faktorizacijo matrik, v poglavju 5.4, smo prikazali razčlenitev rezultatov glede na ocene. Kot smo opisali, metoda očitno pogosteje napoveduje ocene znotraj manjšega intervala. Med razvojem sistemov smo za reševanje tega problema dobili idejo, da bi se metoda med učenjem naučila neke funkcije za kalibracijo napovedi ocen. Ta funkcija bi nato oceno, ki jo metoda izračuna, preslikala v neko novo, ki bi predstavljala končno vrednost napovedi.

Slike

Tabele

Literatura

[1] Y. Koren, R. Bell, C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer Society*, str. 30-37, 2009.

[2] M. Pazzani, D. Billsus, "Content-Based Recommendation Systems," *Springer Berlin / Heidelberg*, str. 325-341, 2007.

[3] About The Music Genome Project®.
Dostopno na: http://www.pandora.com/corporate/mgp

[4] S. K. Lam, J. Riedl, "Shilling Recommender Systems for Fun and Profit," v *Proceedings of the 23th International World Wide Web Conference*, str. 393-402, 2004.

[5] B. Mobasher, R. Burke, R. Bhaumik, C. Williams, "Effective attack models for shilling Item-based collaborative filtering systems," v *Proceedings of the 2005 WebKDD Workshop*, 2005.

[6] M. O'Mahony, N. Hurley, N. Kushmerick, G. Silvestre, "Collaborative Recommendation: A Robustness Analysis," *ACM Transactions on Internet Technology*, zv. 4, št. 4, str. 344-377, 2004.

[7] R. Bell, Y. Koren, "Improved Neighborhood-based Collaborative Filtering," v *Proceedings of KDD Cup and Workshop*, 2007.

[8] D. Billsus, M. Pazzani, "Learning Collaborative Information Filters," v *Proceedings of the 15th International Conference on Machine Learning*, str. 46-54, 1998.

[9] T. Miranda, M. Claypool, A. Gokhale, P. Murnikov, D. Netes, M. Sartin,"Combining Content-Based and Collaborative Filters in an Online Newspaper," v *Proceedings of the ACM SIGIR Workshop on Recommender Systems: Algorithms and Evaluation*, 1999.

[10] D. Billsus, M. Pazzani, "Learning and Revising User Profiles: The Identification of Interesting Web Sites," v *Machine Learning*, zv. 27, št. 3, str. 313-331, 1997.

[11] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Reidl, "Item-based collaborative filtering recommendation algorithms," v *Proceedings of the 10th International Conference on World Wide Web*, str. 285-295, 2001.

[12] M. Deshpande, G. Karypis, "Item-based top-N Recommendation Algorithms," *ACM Transactions on Information Systems*, zv. 22, št. 1, str. 143-177, 2004.

[13] J. S. Breese, D. Heckerman, C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," v *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, str. 43-52, 1998.

[14] D. Billsus , M. Pazzani, "Learning Collaborative Information Filters," v *Proceedings of the 15th International Conference on Machine Learning*, str. 46-54, 1998.

[15] C. Basu, H. Hirsh, W. Cohen, "Recommendation as Classification: Using Social and Content-Based Information in Recommendation," *Proceedings of the 15th National/10th Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, str. 714-720, 1998.

[16] A. I. Schein, A. Popescul, L. H. Ungar, D. M. Pennock, "Methods and Metrics for Cold-Start Recommendations", v *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002.

[17] A. Popescul, L. H. Ungar, D. M. Pennock, S. Lawrence, "Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments", v *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, str. 437-444, 2001.

[18] (2009) "Last.fm Radio Announcement." Dostopno na: http://blog.last.fm/2009/03/24/lastfm-radio-announcement.

[19] (2011) "About Last.fm." Dostopno na: http://www.last.fm/about.

[20] (2011) "Last.fm API." Dostopno na: http://www.last.fm/api.

[21] (2010) "Last.fm dataset 360K." Dostopno na: http://mtg.upf.edu/node/1671.

[22] (2011) "MusicBrainz Tag." Dostopno na: http://musicbrainz.org/doc/MusicBrainz_Identifier.

[23] X. Su , T. Khoshgoftaar, "A survey of collaborative filtering techniques", v *Advances in Artificial Intelligence*, 2009.

[24] B. M. Sarwar, G. Karypis, J. A. Konstan, J. Reidl, "Application of Dimensionality Reduction in Recommender System–A Case Study," v *Proceedings of the ACM WebKDD Workshop*, 2000.

[25] S. Funk, "Netflix Update: Try This at Home,", 2006. Dostopno na http://www.sifter.org/ simon/journal/20061211.html.

[26] Y. Koren, "Factorization Meets the Neighbourhood: A Multifaceted Collaborative Filtering Model," v *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, str. 426-434, 2008.

[27] A. Paterek, "Improving Regularized Singular Value Decomposition for Collaborative Filtering," v *Proceedings of the KDD Cup and Workshop*, str. 39-42, 2007.

[28] G. Takács , I. Pilászy , B. Németh , D. Tikk, "Major components of the gravity recommendation system", v *ACM SIGKDD Explorations Newsletter*, zv. 9, št. 2, 2007.