VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

Šalto starto problemos rekomendacinėse sistemose sprendimas naudojant socialinių tinklų duomenis

Applying Social Network Data for Cold Start Problem in Recommender Systems

Magistro baigiamasis darbas

Atliko:	Andrius Juškevičius	(parašas)
Darbo vadovas:	dr. Rimantas Kybartas	(parašas)
Recenzentas:	prof. habil. dr. Antanas Žilinkskas	(parašas)

Vilnius – 2016

Santrauka

Žmonės priimdami sprendimus dažnai pasikliauja draugų ir pažįstamų rekomendacijomis. Vienas iš rekomendacinių sistemų (toliau - RS) metodų - bendradarbiavimo filtravimas (angl. collaborative filtering, toliau - BF) nors ir imituoja žmonių tarpusavio panašumą, negali identifikuoti, ką žmogus pažįsta, o ko ne. Socialinių tinklų duomenys užpildo šią spragą ir leidžia RS pateikti rekomendacijas atsižvelgiant ir į žmonių tarpusavio santykį.

Šiame darbe pateikta glausta rekomendacinių sistemų apžvalga, išnagrinėtas bendradarbiavimo filtravimo algoritmas, pristatyta šalto starto problema bei apžvelgti socialinių tinklų duomenis naudojantys metodai. Pagrindinis tyrimo objektas - RS su keliomis kategorijomis. Pasiūlytas tokios RS duomenų generavimo būdas, kurį naudojant galima sugeneruoti duomenų rinkinių, pasižyminčių skirtingomis charakteristikomis. Taip pat pasiūlytas metodas, kuriuo galima vertinti pasitikėjimą tarp naudotojų skirtingose kategorijose remiantis tuo, kad tarp kategorijų egzistuoja tam tikras panašumas. Eksperimentais parodyta, kad geriausi rezultatai prognozės tikslumo ir padengimo prasme pasiekiami taikant sričių panašumo metodą kartu su žinomais pasitikėjimo propagavimo metodais.

Raktiniai žodžiai: rekomendacinė sistema, bendradarbiavimo filtravimas, socialinis tinklas, šaltas startas, pasitikėjimas

Summary

One approach to the design of recommender systems that has wide use is collaborative filtering. It's methods are based on analyzing information on users' behaviors, activities or preferences and predicting what users will like based on their similarity to other users. Trust-based recommender systems use social network data to determine user relations and provide recommendations. In this paper a multidomain trust-based recommender system is analyzed with a focus on cold start and data sparsity problem.

One way of dealing with these problems is to apply known trust propagation methods. In this paper a domain similarity method is proposed. This method enables system to evaluate similarity between two domains and evaluate unknown trust value in one domain between two users based on known trust value in another one.

In order to be able to design and carry out experiments, a method for multidomain recommender system data generation was proposed.

It is also shown that it is possible to combine known trust propagation methods together with domain similarity methods. Effectiveness of such approach is proved by experiments which were carried out using dataset generated by applying proposed method for dataset generation.

Keywords: recommender system, collaborative filtering, social network, cold start, trust

Turinys

Įvadas

Kaskart, kai kažko ieškome, tiksliai patys nežinodami, ko - susiduriame su rekomendacijos poreikiu. Iš esmės, didžioji dalis dalykų apie kuriuos žinome, mums kažkada buvo viena ar kita forma pasiūlyta ar nurodyta. Taigi, didelė dalis pasaulio pažinimo proceso įvyksta rekomendacijų dėka. Rekomendacija, kaip reiškinys, gali įgyti įvairias, dažniausiai socialines, formas - informacijos galime gauti iš artimųjų arba tam tikrų atstovų (pavyzdžiui, finansų patarėjo arba konsultanto). Kita forma, apie kurią ir yra šis darbas, yra skaitmeninė - rekomendacinių sistemų (toliau - RS) generuojamos rekomendacijos siekia palengvinti naudotojo patirtį renkantis jį dominančius elementus iš prieinamos aibės. Šios rekomendacijos gali ne tik palengvinti paieškos procesą, bet ir pasiūlyti bei sudominti naudotoją tokiais elementais, apie kuriuos naudotojas nė nenutuokė.

RS plačiai taikomos muzikos, kino ir elektroninės prekybos platformose. Vietoj įprastos paieškos šios sistemos siūlo elementus pasiremdamos naudotojų elgesio istorija. Vienas labiausiai naudojamų metodų - bendradarbiavimo filtravimas (angl. Collaborative Filtering, toliau - BF). Aibė sėkmingų įmonių (pavyzdžiui, Amazon.com, Netflix.com, Last.fm) pritaikė BF metodus tam, kad padidintų naudotojų pasitenkinimą jų siūlomais produktais. Taikant BF daroma prielaida, kad panašūs naudotojai išliks tokie ir ateityje. Taigi, esminė sprendžiama problema - naudotojų panašumo vertinimas. Filtravimo procesas remiasi jau turimais duomenimis, kurie dėl problemos prigimties yra labai reti - sistemoje gali būti tūkstančiai naudotojų ir elementų, tačiau kiekvienas naudotojas dažniausiai būna įvertinęs tik labai mažą visų elementų dalį, taigi panašumo įvertinimas nėra paprastas uždavinys. Negana to, kai sistemoje atsiranda naujas naudotojas, apie jį žinoma per mažai, kad būtų galima pateikti patikimas rekomendacijas. Ši problema dar kitaip vadinama šalto starto problema (angl. cold start problem). Ji yra ypač svarbi ir dėl to, kad, jeigu naujas naudotojas per pakankamai trumpą laiką neįsitikins sistemos nauda, tikėtina, kad niekada ja nebesinaudos.

Ieškant šios problemos sprendimo būdų buvo atlikta nemažai tyrimų apie hibridines RS. Hibridinių RS esmė - taikant BF panaudoti informaciją apie elementų turinį ir jų savybes. Turiniu pagrįstas RS nagrinėja atskira tyrimų šaka, apie kurią šiame darbe nebus kalbama. Nors taikant hibridines RS ir galima išspręsti dalį problemos, jos turi vieną esminį trūkumą - hibridinės RS yra labai priklausomos nuo konteksto, kuriame yra naudojamos. Be to, kai kurioms dalykinėms sritims yra labai sudėtinga identifikuoti elemento atributus, taigi sukurti tokią RS yra labai sudėtinga.

Šio darbo tikslas – pasiūlyti metodą, kuriuo remiantis būtų galima išspręsti duomenų nepakankamumo problemą juos papildant duomenimis iš socialinių tinklų. Šie duomenys puikiai panaudojami pasitikėjimu pagrįstose RS. Pasitikėjimas gali būti vertinamas kaip alternatyvus dydis panašumui. Visgi šie du dydžiai skiriasi:

- pasitikėjimas nebūtinai yra išskaičiuojamas iš duomenų - naudotojai gali jį išreikšti tiesiogiai.
- pasitikėjimas tarp dviejų naudotojų turi kryptį - tai yra naudotojas u_1 gali pasitikėti u_2 ne tiek pat, kiek u_2 u_1.

Pasitikėjimo tinklas - grafas, kurio viršūnės žymi naudotojus, briaunos - santykius tarp jų, o briaunų svoriai - pasitikėjimo įverčius. Toks tinklas ir bus pamatas siūlomiems metodams, kaip spręsti šalto starto problemą, kai nepakanka duomenų naudotojų panašumui nustatyti.

Literatūros apžvalgoje suformuluoti bendradarbiavimo filtravimo metodo apibrėžimas, pristatyta šalto starto problema ir aprašyti skirtingų autorių pasiūlyti šios problemos sprendimo metodai. Taip pat plačiau aprašyti tyrimai apie socialinių tinklų duomenų panaudojimą ir pristatyti jau atlikti darbai šia problemos sprendimo kryptimi. Gilinamasi į socialinių tinklų duomenų panaudojimo galimybes siekiant panaikinti (arba sušvelninti) šalto starto problemos efektą.

Kitame skyriuje pristatytas būdas, kuriuo remiantis galima generuoti duomenis, kurių struktūra atitinka tyrimo objektą - socialinę RS, kurioje naudotojai vieni su kitais susiję pasitikėjimo ryšiais ir egzistuoja kategorijos, kurioms priklauso elementai ir kuriose naudotojai vieni kitais pasitiki. Toliau pasiūlyti du metodai skirti panašumo tarp kategorijų nustatymui bei pasitikėjimo įverčių skirtingose kategorijose radimui. Taip pat pasiūlytas būdas, kuriuo remiantis galima kombinuoti siūlomą sričių panašumo metodą kartu su žinomais pasitikėjimo propagavimo metodais. Galiausiai pateikti eksperimentų, atliktų taikant minėtus metodus, rezultatai, jie paaiškinti, pateiktos išvados ir pasiūlytos galimos tolimesnės tyrimo kryptys.

1. Bendradarbiavimo filtravimas

1.1. Bendradarbiavimo filtravimo metodas

Visų pirma, suformuluokime RS sprendžiamą problemą formaliai taip, kaip tai padaryta [DK11]. Naudotojų aibė pažymėkime U, elementų aibę I. Taip pat, pažymėkime R aibę sistemoje turimų reitingų ir S – aibę galimų reitingo reikšmių (pvz. $S = [1,5]$). Taip pat, tarkime, kad vienas reitingas r_{ui} gali būti priskirtas vienam elementui $i \in I$ vieno naudotojo $u \in U$. Naudotojų poaibį, kuris yra įvertinęs elementą i, pažymėkime U_i. Analogiškai, I_u pažymėkime aibę elementų, kuriuos yra įvertinęs naudotojas u. Elementų, kuriuos yra įvertinę abu naudotojai u ir v, aibę $I_u I_v$ pažymėkime I_{uv}. Analogiškai, U_{ij} žymi aibę naudotojų, kurie yra įvertinę tiek elementą i, tiek j. Dažniausiai sutinkama problema – geriausių N rekomendacijų problema. Vienas būdų spręsti šias problemas yra įvertinti funkciją $f : U \times I -> S$, kuria prognozuojamas reitingas $f(u,i)$. Ši funkcija tada yra naudojama naudotojui u rekomendacijai elementų aibę, kuriai įvertinti reitingai turi didžiausią reikšmę. RS galima modeliuoti dviem būdais:

- Turiniu pagrįstų metodų esmė – identifikuoti charakteristikas, kuriomis pasižymėjo elementai, kuriuos naudotojas palankiai įvertino praeityje, tada naudotojui rekomenduoti kitus elementus su panašiomis charakteristikomis.

- Bendradarbiavimo filtravimu (toliau - BF) pagrįsti metodai rekomenduoja elementus, kurie patiko naudotojams, turintiems panašias pirmenybes. BF metodai remiasi tik naudotojų suteiktais reitingais. Juos taikant ieškomi panašumai tarp naudotojų pirmenybių. Toks būdas lemia dvi geras savybes, kuriomis nepasižymi turiniu pagrįsti metodai

  - dėl įžvalgumo pasiūlymai nėra akivaizdūs, netikėti (t.y. tokie, kokių naudotojas kitomis aplinkybėmis turbūt nerastų)
  - galimas pritaikymas skirtingose srityse, kai elementu pagrįstos rekomendacijos reikalauja specifinių srities parametrų duomenų (pvz., kiek tam tikras filmas turi komedijos, kiek dramos bruožų)

  Bendradarbiavimo filtravimo sąvoką pirmąsyk panaudojo Goldberg [GNOT92]. Toliau darbe bus nagrinėjami būtent šiai klasei priklausantys metodai.

Bendradarbiavimo filtravimu pagrįstos reitingo esminis žingsnis artimiausių naudotojo kaimynų parinkimas. Naudotojų tarpusavio artumas nustatomas naudojant panašumo metrikas, kurios bus aprašytos vėliau skyriuje 1.3. Prognozę galima atlikti dvejopai:

- Taikant artimiausių kaimynų regresiją, reitingas įvertinamas skaičiuojant artimiausių kaimynų vidurkį su svoriais.

- Taikant artimiausių kaimynų klasifikaciją, elemento reitingas parenkamas toks pats, kokį jam yra suteikęs artimiausias naudotojo kaimynas.

Pagrindinis turiniu pagrįsto prieš naudotoju pagrįsto reitingo prognozavimo metodo trūkumas yra tas, kad tokiu būdu sugeneruotos rekomendacijos yra nors ir tikslios, tačiau nelabai vertingos, nes rekomenduojami elementai pernelyg panašūs į tuos, apie kuriuos naudotojas jau žino. Šią problemą galima vertinti kaip pernelyg didelio pritaikymo (angl. over-specialization) problemą arba kaip įžvalgumo (angl. serendipity) stygių. Be to, naudotoju pagrįstas metodas yra paremtas realiu žinių perdavimo iš lūpų į lūpas modeliu, todėl, tikėtina, geriau modeliuoja žinių išgavimą.

Norėdami prognozuoti naudotojo u reitingą elementui i, imame k artimiausių kaimynų $N_i(u,k)$ ir ieškome jų vidurkio.

$$\hat{r}_{ui} = \frac{1}{N_i(u,k)} \sum_{v \in N_i(u,k)} r_{vi} \tag{1}$$

Ši formulė neatsižvelgia į naudotojų panašumą. Būtų neteisinga vertinti visus kaimynus vienodai, kai kai kurie yra panašūs į naudotoją u, o kai kurie visiškai nepanašūs. Čia įtraukiame svorių sąvoką. Svoriai gali reikšti arba panašumą, arba, kaip vėliau bus parodyta, vieno naudotojo pasitikėjimą kitu.

$$\hat{r}_{ui} = \frac{\sum\limits_{v \in N_i(u,k)} w_{uv} r_{vi}}{\sum\limits_{v \in N_i(u,k)} |w_{uv}|} \tag{2}$$

Šioje formulėje naudojamas svertinis vidurkis yra dažniausiai praktikoje taikomas, paprastas ir tikslus prognozės sudarymo būdas, tačiau lieka klausimas - į kiek kaimynų reikia atsižvelgti? GroupLens sistemoje visi $U \setminus \{u\}$ laikomi kaimynais; kitose sistemose kaimynai parenkami pagal panašumo slenkstį. Tinkamas kaimynų skaičiaus parinkimas leidžia įvertinti tikslesnes prognozes, nes taip sumažinamas mažiau panašių keliamas triukšmas. Dar kitas būdas - atsižvelgiant į dalykinę sritį parinkti konstantą. Geriausią kaimynų parinkimo strategiją galima išsiaiškinti tiesiog eksperimentuojant su konkrečiais duomenimis, nes įprastai RS viena nuo kitos labai skiriasi tiek dėl dalykinės srities subtilybių, tiek dėl skirtingo RS naudotojų elgesio.

1.2. Šalto starto problema

Šalto starto problema susijusi su duomenų nepakankamumo problema. Ši problema dvejopa, galimi jos variantai:

- naudotojo šaltas startas
- elemento šaltas startas

Šiame darbe koncentruojamasi į naujo naudotojo problemą. Bendradarbiavimo filtravimu pagrįstuose metoduose, norint pateikti prasmingą rekomendaciją, visų pirma reikia suformuoti aiškų naudotojo pirmenybių vaizdą. Naujam naudotojui to padaryti faktiškai neįmanoma. Šia problemą galima spręsti visai negeneruojant rekomendacijų arba teikiant rekomendacijas remiantis naudotojo profiliu - gyvenamąja vieta, amžiumi, lytimi ir panašiai. Dar kitas būdas, kuris nagrinėjamas šiame darbe - trūkstamų duomenų įvertinimas.

1.3. Naudotojų panašumo apskaičiavimas

Jau anksčiau buvo minėta, kad norint rasti prognozuojamą naudotojo u elementui i suteikiamą reitingą, reikia žinoti svorius, kuriais matuojama kitų panašių naudotojų įtaka galutinei prognozei. Vienas šių svorių įvertinimo būdų - naudotojų panašumo išskaičiavimas iš reitingų. Toliau pristatomi metodai, kuriais galima įvertinti naudotojų panašumą. Pearson'o, Spearmano koreliacija ir kosinuso panašumas detaliau aprašyti [DK11].

1.3.1. Pearson'o koreliacija

Pearson'o koreliacija skirta statistinės koreliacijos radimui:

$$s(u,v) = \frac{\sum\limits_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum\limits_{i \in I_u \cap I_v} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum\limits_{i \in I_u \cap I_v} (r_{v,i} - \bar{r}_v)^2}} \tag{3}$$

Šis metodas veikia prasčiau, kai reikia paskaičiuoti panašumą tarp naudotojų, kurie bendrai yra įvertinę mažai elementų. Galima išeitis - nustatyti slenkstį, nuo kurio koreliacija būtų mažinama.

1.3.2. Apribota Pearson'o koreliacija

Kai kalbame apie šį metodą, pereiname nuo tolydinio prie kategorinio parametrų vertinimo. Be to, atsižvelgiama į nuokrypį ne nuo vidurkio, o nuo abejingumo įverčio. Jeigu turime reitingų skalę nuo 1 iki 7, tada 4 reiškia abejingumą. Pažymėkime $r_z = 4$. Tada Shardanand ir Maes pasiūlyta apribota Pearson'o koreliacija randama taip:

$$s(u,v) = \frac{\sum\limits_{i \in I_u \cap I_v} (r_{u,i} - r_z)(r_{v,i} - r_z)}{\sqrt{\sum\limits_{i \in I_u \cap I_v} (r_{u,i} - r_z)^2} \sqrt{\sum\limits_{i \in I_u \cap I_v} (r_{v,i} - r_z)^2}} \tag{4}$$

1.3.3. Spearman'o rango koreliacija

Spearman'o rango koreliacija panaši į Pearson'o koreliaciją, skirtumas toks, kad skaičiuojant Spearman'o koreliaciją, naudotojo reitingai yra surūšiuojami didėjimo tvarka, jiems priskiriami rangai - mažiausią reikšmę turintis reitingas gauna reikšmę 1. Tokiu būdu išvengiama reitingų normalizavimo problemos. Šis metodas veikia ne itin gerai, kai yra mažas galimų reikšmių skaičius, be to skaičiavimo požiūriu reikalaujantis daugiau resursų dėl surūšiavimo žingsnio.

1.3.4. Kosinuso panašumas

Šis metodas skiriasi nuo ankstesnių tuo, kad yra į problemą žiūrima ne iš statistinio, o iš tiesinės algebros požiūrio taško. Naudotojai atvaizduojami kaip $|I|$ dimensijų turintys vektoriai, o panašumas apskaičiuojamas, kaip kosinuso panašumas tarp dviejų reitingo vektorių. Jis randamas

sudauginant šiuos vektorius ir padalinant iš $L2$ (Euklido) normų sandaugos:

$$s(u,v) = \frac{\boldsymbol{r}_u \cdot \boldsymbol{r}_v}{||\boldsymbol{r}_u||_2||\boldsymbol{r}_v||_2} \tag{5}$$

1.3.5. Euristinis PIP panašumo matas

Euristinis panašumo matas pasiūlytas [Ahn08] kreipia dėmesį į šalto starto problemą. Taikant šią panašumo metriką į šalto starto problemą atsižvelgiama panašumą apskaičiuojant remiantis trimis faktoriais - panašumu, poveikiu, populiarumu.

$$s(u_i, u_j) = \sum_{k \in C,j} \textbf{PIP}(r_{i,k}, r_{j,k}) \tag{6}$$

čia r_{ik} ir r_{jk} reitingai elementui k nuo naudotojų i ir j atitinkamai, $PIP(r_{ik}, r_{jk})$ - PIP reikšmė reitingams r_{ik} ir r_{jk}

$$PIP(r_1,r_2) = Proximity(r_1,r_2) \times Impact(r_1,r_2) \times Popularity(r_1,r_2) \tag{7}$$

Detalesnį aprašymą, kaip randamos šios reikšmės galima rasti [Ahn08].

1.3.6. Panašumas su svoriais

Said [SJA12] pastebėjo, kad dažniausiai naudojami panašumo matai (Pearson'o koreliacija, kosinuso panašumas) turi tokį trūkumą, kad jie neatsižvelgia į bendrai įvertintų elementų populiarumą - bendrai įvertinti populiarūs (įvertinti daugelio naudotojų) elementai vertinamam panašumui daro mažesnę įtaką negu retai vertinami. Šį trūkumą siūloma spręsti panašumo matuose įvedant populiarumo svorius.

Tokiu būdu randama Pearson'o koreliacija atrodytų taip:

$$s_w(u,v) = \frac{\sum\limits_{i \in I_u \cap I_v} w_i^s (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum\limits_{i \in I_u \cap I_v} w_i^s (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum\limits_{i \in I_u \cap I_v} w_i^s (r_{v,i} - \bar{r}_v)^2}} \tag{8}$$

ir kosinuso panašumas:

$$s_w(u,v) = \frac{\sum\limits_{i \in I_u \cap I_v} w_i^s \cdot r_{u,i} \cdot r_{v,i}}{\sqrt{\sum\limits_{i \in I_u} w_i \cdot r_{u,i}^2} \sqrt{\sum\limits_{i \in I_v} w_i^s \cdot r_{v,i}^2}} \tag{9}$$

o svoriai w_i^s gali randami būti randami tokiais būdais:

$$w_i^{s,inf} = \log \frac{|U|}{|U_i|} \tag{10}$$

$$w_i^{s,lin} = 1 - \frac{|U_i|}{|R|} \tag{11}$$

Čia $|U|$ - naudotojų skaičius, $|U_i|$ - naudotojų, įvertinusių elementą i skaičius, $|R|$ reitingų skaičius.

[SJA12] parodyta, kad šis metodas geriausiai veikia naudotojams ”po šalto starto” (angl. post cold start users), kai reitingų skaičius yra tarp 20 ir 80, kitiems rėžiams rezultatai buvo labai panašūs į tuos, kurie buvo gauti naudojant Pearson’o koreliaciją be svorių.

2. Socialiniai tinklai ir pasitikėjimu pagrįstos rekomendacinės sistemos

2.1. Socialiniai tinklai ir pasitikėjimo sąvoka

Socialinis tinklas - virtuali bendruomenė, kurios nariai bendrauja ir dalinasi tarpusavyje informacija. Žmonės tokiose bendruomenėse būna susiję - arba abipusiu (draugų), arba vienpusiu (pasekėjų) ryšių.

Pasitikėjimu pagrįstų RS tikslas - įvertinti, kiek vienas naudotojas pasitiki kitu, kai žinomas pasitikėjimo tinklas (angl. web of trust). Įprastai toks įvertis randamas taikant propagavimo ir agregavimo operatorius. Propagavimo operatoriai lemia, kaip žinant pasitikėjimo kelią randami pasitikėjimo įverčiai. Kol kas nesigiliname į tai, kaip gaunami pasitikėjimo įverčiai, laikome juos žinomais.

- Vienas dažniausiai naudojamų propagavimo operatorių yra daugyba. Pavyzdžiui, u_1 pasitiki u_2 0.8, o u_2 pasitiki u_3 0.5, tada u_1 pasitiki u_3 $0.8 \times 0.5 = 0.4$.
- Kitas operatorius - silpniausios grandies. Anksčiau pateikto pavyzdžio atveju u_1 pasitikėjimas u_3 būtų lygus 0.5.
- Konjunkcijos operatorius - $max(t_1 + t_2 - 1)$ ankstesniame pavyzdyje grąžintų 0.3 u_1 pasitikėjimą u_3.

Agregavimo operatoriai skirti situacijomis, kai egzistuoja keli propagavimo keliai. Šie operatoriai apjungia kelis pasitikėjimo įverčius į vieną. Žinoma, ne visi pasitikėjimo keliai yra vienodo ilgio, tai yra, viename kelyje gali būti 1 naudotojas, kitame - 5. Verta paminėti, kad [Gol05] buvo parodyta, kad svarbesni yra trumpesni keliai, ir kuo ilgesnis kelias - tuo mažiau informacijos jis suteikia. Taip yra dėl to, kad kiekvienas pasitikėjimo įvertis turi tam tikrą paklaidą - triukšmą, ir ilgesniame kelyje šio triukšmo yra daugiau. Ši problema sprendžiama taikant agregavimo operatorių. Galimi variantai - trumpiausio kelio operatorius, matematinis vidurkis, vidurkis su įvairiomis, atsižvelgiančiomis į kelio ilgį, schemomis.

Nors gali pasirodyti, kad nepasitikėjimas ir pasitikėjimas yra dvi viena kitai priešingos sąvokos, tai yra tik prielaida, kuri leidžia supaprastinti problemą. Kitas požiūris teigia, kad nepasitikėjimas negali būti prilyginamas pasitikėjimo nebuvimui.

Josang [JMP06] rašo apie subjektyvią logiką (angl. subjective logic), kurioje nepasitikėjimas yra vertinamas kaip atskiras nuo pasitikėjimo dydis. Šios teorijos branduolys - subjektyvios nuomonės (angl. subejctive opinions), kurios formaliai užrašomos taip: $w_x^A = (b,d,u,a)$, kur b, d ir u apibūdina pasitikėjimą, nepasitikėjimą ir neužtikrintumą. Pastebima, kad $b,d,u \in [0,1]$ ir $b + d + u = 1$. Parametras $a \in [0,1]$ nurodo, kokį svorį nustatant tikėtiną nuomonės įvertį (angl. opinion's probability expectation value) turi neužtikrintumas - $E(w_x^A) = b + au$. Šis modelis remiasi tiksliais apibrėžimais ir formulėmis, jomis galima manipuliuoti ir gauti analitiškai pagrįstus rezultatus, kuriais, pavyzdžiui, paaiškinamos populiarumo bangos.

2.2. Pasitikėjimo apskaičiavimas

Pasitikėjimo tinkle dauguma naudotojų vienas kito nepažįsta. Nepaisant to, reikia nustatyti sąryšius tarp jų. Tam yra naudojamos pasitikėjimo metrikos, kurios remdamosi naudotojų santykiais nustato, kiek vienas naudotojas pasitiki kitu. Pasitikėjimo metrikos b8na lokalios ir globalios.

- Lokalios metrikos įvertina pasitikėjimą kiekvienam naudotojui individualiai - dėl to jos gali būti tikslesnės ir reikalauja daugiau skaičiavimo resursų. Toliau bus pristatyti lokalių metrikų pavyzdžiai - TidalTrust, MoleTrust.
- Globalios metrikos įvertina bendrą elemento reitingą visoje pasitikėjimo sistemoje. Apie jas toliau kalbama nebus, žymiausias pavyzdys - PageRank algoritmas naudojamas Google paieškos sistemoje.

Kaip minėta, pasitikėjimo skaičiavimui svarbi tranzityvumo prielaida, tačiau, ji teisinga viename kontekste - jeigu a pasitiki b kai kalbama apie automobilius, o b pasitiki c sodininkystės klausimais, nieko negalėsime pasakyti apie a pasitikėjimą c kompiuterijos žiniomis.

2.2.1. TidalTrust

Ši formulė yra esminė Golbeck rekomendacijos algoritme. Algoritmo autoriai šią formulę išvedė atlikdami eksperimentus, kurių metu jie ignoruodami tiesioginį naudotojo u pasitikėjimą naudotoju v tyrinėjo kelius, jungiančius šiuos du naudotojus. Lygindami taikant propagavimą gautus įverčius su tikromis pasitikėjimo reikšmėmis jie pastebėjo, kad:

- remiantis trumpesniais pasitikėjimo keliais randami tikslesni pasitikėjimo įverčiai
- remiantis keliais su didesnėmis pasitikėjimo reikšmėmis taip pat randami tikslesni pasitikėjimo įverčius

Atsižvelgiant į pirmą pastebėjimą buvo sugalvota, kad reikia apriboti ieškomo pasitikėjimo kelio ilgį tarp naudotojų. Nustačius fiksuotą kelio ilgį gali atsitikti taip, kad gali būti pasiekiama tik maža dalis naudotojų. Dėl šios priežasties nustatytas kintamas galimas kelio ilgis - ilgiausias kelias, reikalingas sujungti tikslinį naudotoją su naudotoju, įvertinusiu elementą i.

Atsižvelgdami į kitą pastebėjimą (apie didesnes pasitikėjimo reikšmes vedančias prie tikslesnių įverčių) autoriai siūlo apriboti informaciją taip, kad ji būtų gaunama tik iš patikimiausių naudotojų. Tačiau čia vėl reikia pastebėti, kad skirtingi žmones turi skirtingas pasitikėjimo skales - vienas gali pasitikėti visais, kitas - niekuo. Be to, dažnai būna taip, kad mažai kelių turi tokią pačią pasitikėjimo reikšmę. Dėl šių priežasčių Golbeck nusprendė įvesti reikšmę, atspindinčią kelio stiprumą (t.y. mažiausią pasitikėjimo reitingą kelyje) ir apskaičiuoti maksimalų kelio stiprumą max (iš visų kelių, vedančių prie elementą vertinusių naudotojų), kuris po to naudojamas kaip slenkstis dalyvavimui algoritme.

$$t_{a,u} = \frac{\sum\limits_{v \in WOT^+(a)} t_{a,v} t_{v,u}}{\sum\limits_{v \in WOT^+(a)} t_{a,v}} \tag{12}$$

(12) pateikta TidalTrust formulė. Joje $WOT^+(a)$ atspindi naudotojų aibę, kuriems naudotojo a pasitikėjimo jais reikšmė viršija slenkstį max.

Šis algoritmas yra rekursinis - $t_{a,u}$ rekursiškai skaičiuojamas, kaip svertinis pasitikėjimo reikšmių $t_{v,u}$ vidurkis. Šis algoritmas priklauso laipsniškų pasitikėjimo algoritmų klasei ir yra lokalios pasitikėjimo metrikos pavyzdys.

Golbeck parodė, kad pasitikėjimu pagrįstas svertinis vidurkis taikomas kartu su TidalTrust metodu nebūtinai visada yra pranašesnis už BF, tačiau grąžina tikslesnes prognozes naudotojams, kurie nesutinka su vidutiniu elemento reitingu.

2.2.2. MoleTrust

$$p_{a,i} = \bar{r_a} + \frac{\sum\limits_{u \in R^T} t_{a,u}(r_{u,i} - \bar{r_u})}{\sum\limits_{u \in R^T} t_{a,u}} \tag{13}$$

(13) formulė - Massa [AMT05] pasiūlyto rekomendacijų algoritmo pagrindas. Ši metrika susideda iš dviejų žingsnių:

- pirmame žingsnyje pašalinami pasitikėjimo tinkle esantys ciklai
- antrame žingsnyje atliekamas pasitikėjimo apskaičiavimas

Ciklų pašalinimu siekiama, kad kiekvienas naudotojas tinkle būtų aplankytas tik kartą. Taip siekiama efektyviau atlikti propagavimo žingsnį. Ciklų pašalinimu transformuojame pradinį tinklą į kryptinį beciklį grafą. Tuomet pasitikėjimo prognozę $t_{a,u}$ galime rasti atlikdami paprastą grafo apėjimą - visų pirma, randamas pasitikėjimas naudotojais, iki kurių atstumas lygus 1, tada pasitikėjimas tais, iki kurių atstumas 2 ir taip toliau. Verta pastebėti, kad pasitikėjimo naudotoju, esančių atstumu x priklauso nuo anksčiau apskaičiuotų pasitikėjimo reikšmių naudotojams esantiems atstumu $x - 1$.

Pasitikėjimas naudotojais, esančiais atstumu didesniu nei 1 skaičiuojamas panašiu būdu, kaip (12). TidalTrust naudotojas yra pridedamas prie $WOT^+(a)$ tada ir tik tada, jeigu jis yra trumpiausiame kelyje nuo naudotojo a iki elemento i. MoleTrust atveju $WOT^+(a)$ apima visus naudotojus, kurie įvertino tam tikrą elementą ir gali būti pasiekti pasitikėjimo tinklu per ne daugiau kaip d žingsnių. Parametras d vadinamas išskaidymo horizontu. Kitas MoleTrust parametras - pasitikėjimo slenkstis, kuris TidalTrust algoritme buvo apibrėžtas kaip dinamiška max reikšmė. MoleTrust pasitikėjimo slenkstis - fiksuotas dydis.

MoleTrust taip pat priklauso laipsniškų lokalių pasitikėjimo metrikų klasei. Algoritmo autoriai eksperimentu parodė, kad MoleTrust randa geresnius pasitikėjimo įverčius nei globalios pasitikėjimo metrikos, tokios kaip naudojamos pavyzdžiui eBay, ypač kai kalba eina apie kontraversiškus naudotojus, kuriuos dalis naudotojų vertina kaip labai patikimus, o kita dalis - labai nepatikimus. Autoriai taip pat parodė, kad šis algoritmas išgauna tikslesnes prognozes naujiems naudotojams.

2.2.3. Pasitikėjimu pagrįstas svoris

Šis metodas pristatytas [OS05] naudoja vartotojo ir tiekėjo sąvokas. Reitingo prognozė skaičiuojama panašiai kaip (2):

$$c(i) = \bar{c} + \frac{\sum\limits_{p \in P(i)} (p(i) - \bar{p}) w(c,p,i)}{\sum\limits_{p \in P(i)} |w(c,p,i)|} \tag{14}$$

$w(c,p,i)$ yra panašumo ir pasitikėjimo harmoninis vidurkis

$$w(c,p,i) = \frac{2(sim(c,p))(trust(p,i))}{sim(c,p) + trust(p,i)} \tag{15}$$

čia c - vartotojas (angl. consumer), p - gamintojas (angl. producer), i -elementas, $sim(c,p)$ - panašumas tarp vartotojo ir gamintojo. $trust(p,i)$ matuoja kiek c gali pasitikėti p elemento i vertinimu ir yra randamas taip:

$$trust(p,i) = \frac{|\{(c_k, i_k) \in CorrectSet(p) : i_k = i\}|}{|\{(c_k,i_k) \in RecSet(p) : i_k = i\}|} \tag{16}$$

Šis reiškinys rodo, kokia dalis naudotojo p rekomendacijų būna teisinga. Taip randamas pasitikėjimas vadinamas profilio lygio pasitikėjimu (angl. profile-level trust).

3. RS vertinimas

3.1. RS vertinimo metodai

Dažniausiai RS vertinimui naudojamas vidutinės absoliučios klaidos metodas (angl. Mean Absolute Error, trumpinama MAE) pagrįstas principu "išimk vieną" (angl. leave-one-out).

$$MAE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} |\hat{r}_{ui} - r_{ui}|} \tag{17}$$

Šio metodo esmė - iš duomenų rinkinio išimti vieną reitingą ir atlikti jo prognozę. Prognozė tada lyginama su tikru reitingu ir taip randama prognozės klaida. Šis metodas tokį trūkumą, kad kiekvieną klaidą ieško vienodu būdu. Pavyzdys, iliustruojantis, kodėl tai yra negerai toks: tarkime, turime 101 naudotoją 1 yra įvertinęs 300 elementų, o 100 - po 3. Tokiu atveju aptariamas duomenų rinkinys turi 600 reitingų. Testuodami RS "išimk vieną" principu, slėptume iš eilės visus reitingus ir bandytume juos nuspėti. Bėda ta, kad RS kur kas geriau veikia naudotojams, turintiems daug reitingų ir prasčiau naujiems (arba nelinkusiems reitinguoti) naudotojams. MAE atveju vienas daug reitingų suteikęs naudotojas turi tokį patį svorį, kaip likę 300. Akivaizdu, kad taip iškreipiama realybė - 300 nepatenkintų naudotojų prieš 1 patenkintą reiškia, kad sistema nėra tokia gera. Tam, kad išspręsti šią problemą buvo pasiūlytas patobulintas metodas - vidutinė absoliuti naudotojo klaida (angl. MAUE - Mean Average User Error). Jo esmė paprasta - randame MAE kiekvienam naudotojui ir tada randame visų naudotojų MAE vidurkį. Tokiu būdu kiekvienas naudotojas skaičiuojant vidutinę klaidą turi vienodą įtaką.

Alternatyvus MAE metodas yra vidutinės kvadratinės klaidos šaknis (angl. RMSE- Root Mean Squared Error):

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (\hat{r}_{ui} - r_{ui})^2} \tag{18}$$

RMSE lyginant su MAE stipriau baudžia už dideles klaidas. Pavyzdžiui, duomenų aibėje su keturiais paslėptais reitingais RMSE geriau vertintų sistemą, kurioje klaida lygi 2 trims reitingams ir 0 vienam reitingui, negu sistemą, kurioje klaida vienai reikšmei lygi trims, bet ji neklysta likusioms trims. MAE geriau vertintų antrą sistemą.

Kitas svarbus RS vertinimo matas yra - padengimas (angl. coverage). Herlocker RS vertinimo metodų apžvalgoje [HKR02] pabrėžia, kaip svarbu yra žiūrėti ne tik į tikslumą, bet ir į padengimą bei nurodo į kelis darbus tyrusius šią sritį. Padengimas - sąvoka skirta apibūdinti keliems skirtingiems aspektams:

- Elementų erdvės padengimas (angl. item space coverage), dar vadinamas katalogo padengimu, apibūdina, kokią dalį visų RS esančių elementų RS gali rekomenduoti. Paprasčiausias būdas apskaičiuoti šį rodiklį - rasti procentą elementų, kurie gali būti rekomenduoti. Kitas katalogo padengimo matas - pasiūlymų įvairumas - matuoja kaip nevienodai pasirenkami skirtingi elementai, kai naudojama tam tikra RS. Šį matą galima apskaičiuoti keliais būdais:
  - jeigu kiekvienas elementas i sudaro $p(i)$ dalį naudotojo pasirinkimų, galime apskai-

čiuoti Gini indeksą:

$$G = \frac{1}{n-1} \sum_{j=1}^{n} (2j - n - 1) p(i_j) \tag{19}$$

Kai visi elementai pasirenkami vienodai dažnai indekso reikšmė lygi 0, kai visada pasirenkamas vienas elementas - 1.

– Shannon'o entropija lygi 0, kai tas pasirenkamas tas pats elementas ir $\log n$ kai visi elementai pasirenkami vienodai dažnai

$$H = -\sum_{i=1}^{n} p(i) \log p(i) \tag{20}$$

- Naudotojų erdvės padengimas (angl. user space coverage) - terminas, nusakantis, kuriai daliai naudotojų RS gali sugeneruoti rekomendaciją. Kartais RS negali nieko rekomenduoti tam tikram naudotojui dėl mažo pasitikėjimo prognozės tikslumu. Toks padengimas gali būti įvertintas matuojant naudotojo profilio turiningumą, reikalingą, kad jam būtų sugeneruota rekomendacija. BF atveju tai galėtų būti mažiausias reitingų skaičius, kuri naudotojas privalo suteikti tam, kad gautų rekomendaciją.

Šaltas startas gali būti laikomas padengimo problemos dalimi. Norint spręsti šalto starto problemą, galima nustatyti slenkstį, apibrėžiantį, kada elementai yra "šalti". Pavyzdžiui, galima laikyti elementą "šaltu", jeigu jis neturi nė vieno reitingo arba, jeigu elementas sistemoje yra trumpiau nei nustatytą laiko tarpą.

Gali būti, kad sistema geriau rekomenduos "šaltus" elementus, "karštų" elementų rekomendacijos kaina. Tai gali būti teigiamas RS bruožas, ypač jeigu yra svarbios naujoviškumo ir įžvalgumo savybės.

3.2. RS vertinimo aspektai

[SG11] pristatytos kelios RS savybės - prognozės tikslumas, padengimas, pasitikėjimas, patikimumas, naujoviškumas, įžvalgumas, įvairumas, naudingumas, rizika, atsparumas atakoms, privatumas, pritaikomumas, praplečiamumas. Aptarsime kelias jų plačiau.

3.2.1. Patikimumas

Patikimumą (angl. confidence) geriausia apibūdinti pavyzdžiu. Jeigu sistema pasiūlo naudotojui du elementus su vienodais prognozuojamais reitingais, tačiau vienos rekomendacijos patikimumas yra mažesnis nei kitos, tai naudotojui gali būti pravartu ją atidžiau patikrinti - perskaityti aprašymą ar panašiai.

3.2.2. Pasitikėjimas

Pasitikėjimas (angl. trust) skiriasi nuo patikimumo tuo, kad pasitikėjimas matuoja sistemos pasitikėjimą reitingais, o pasitikėjimas šiuo atveju nurodo į naudotojo santykį su reitingais. Siste-

ma, siekdama padidinti pasitikėjimą gali pasiūlyti kelis elementus, kuriuos naudotojas jau žino ir mėgsta. Kitas būdas, kaip padidinti pasitikėjimą - paaiškinti naudotojui, kodėl jam siūlomas vienas ar kitas elementas.

3.2.3. Naujoviškumas

Naujoviškos rekomendacijos susideda i6 elementų, apie kuriuos naudotojai nežinojo anksčiau. Paprasčiausias būdas padidinti rekomendacijų naujoviškumą - pašalinti iš galimų elementų aibės jau vertintus ir peržiūrėtus elementus, tačiaun to nepakanka, jeigu norime iš rekomendacijų pašalinti visus elementus, apie kuriuos naudotojas jau žino.

Norint ištirti RS naujoviškumą paprasčiausia tai padaryti "on-line" eksperimentu. Vis dėlto, tai gali būti brangu, todėl buvo sugalvotas ir "off-line" eksperimento metodas, aprašytas [SG11]. Metodo esmė tokia: nuo pasirinkto laiko taško reitingai yra paslepiami. Rekomenduojant sistema gautų taškų už kiekvieną iš tiesų įvertintą elementą ir atimami taškai už kiekvieną elementą, kuris buvo rekomenduotas iki pasirinkto laiko taško.

Tarkime, norime įvertinti rekomendacijų naujoviškumą. Darydami prielaidą, kad naudotojai įvertina elementus, po to kai jais pasinaudoja, padaliname reitingus. Kiekvienam testuojamam naudotojui atsitiktinai parenkame laiko tašką, nuo kurio reitingai paslepiami. Tyrimai parodė, kad žmonės labiau linkę įvertinti elementus, kurie jiems arba labai patiko, arba labai nepatiko. Taigi, slepiame reitingus esančius prieš nukirpimo tašką su tikimybe $1 - \frac{|r-3|}{2}$, kur $r \in \{1,2,3,4,5\}$ galimų elemento reitingų aibė, o 3 yra neutralus reitingas. Siekiama, vengti paslėptų elementų prognozavimo, nes naudotojas apie juos jau žino. Tada kiekvienam naudotojui sugeneruojamos 5 rekomendacijos ir skaičiuojamas jų tikslumas atmetant rekomendacijas elementų, rekomenduotų iki pasirinkto laiko taško. RS su didesniu tikslumu laikomos pranašesnėmis.

3.2.4. Įžvalgumas

Įžvalgumu matuojama, kiek stebinančios yra sėkmingos rekomendacijos. Pavyzdžiui, kalbant apie filmų RS, jeigu naudotojas įvertino daug filmų su tam tikru aktoriumi, pasiūlytas filmas su tuo pačiu aktoriumi gali būti naujoviškas, tačiau vargu ar galėsime šią rekomendaciją vadinti netikėta. Iš kitos pusės, atsitiktinės rekomendacijos gali būti labai stebinančios, tačiau tikslumo kaina.

Vienas būdų suprojektuoti RS taip, kad jos pasiūlymai būtų įžvalgesni yra toks - nustačius atstumo matą tarp elementų, remiantis jų turiniu, sėkmingą rekomendaciją galime vertinti labiau, jeigu ji yra "toliau" nuo jau anksčiau teigiamai įvertintų elementų. Pavyzdžiui, turime knygų RS ir norime naudotojui rekomenduoti knygas autorių, kurių jis nežino. Tuomet turime sukonstruoti metriką tarp knygos b ir anksčiau perskaitytų knygų aibės B. Tarkime $c_{B,w}$ - autoriaus w knygų skaičius aibėje B. Tarkime $c_B = \max_w C_{B,w}$ - maksimalus autoriaus w knygų skaičius aibėje B. Tada $d(b, B) = \frac{1+c_B-c_{B,w(b)}}{1+c_B}$, kur $w(b)$ - b knygos autorius.

Dabar galime atlikti "off-line" eksperimentą, kuriu galime nustatyti, kuris iš galimų algoritmų generuoja įžvalgesnes rekomendacijas. Kiekvieno naudotojo profilį padaliname į dvi dalis - stebimų knygų B_i^o ir paslėptų knygų B_i^h. Naudodami B_i^o duomenis, užklausiame RS 5 rekomendacijų. Už

kiekvieną paslėptą knygą $b \in B_i^h$, kuri pasirodė tarp rekomendacijų, RS gauna $d(b, B_i^o)$ taškų. Tokiu būdų RS yra apdovanojama už sėkmingas mažiau žinomų autorių knygų rekomendacijas.

3.2.5. Atsparumas

Atsparumas (angl. robustness) reiškia sistemos atsparumą atakoms. Atakos rengiamos norint iškreipti reitingus tam tikrų elementų naudai arba nenaudai (pavyzdžiui, kai norima pakenkti konkurentams). Tai galima padaryti sukuriant netikrų profilių, kurie suteiktų elementams fiktyvius reitingus. Kadangi sukurti visiškai atsparią atakoms RS yra neįmanoma, tinkamiausias būdas įvertinti sistemos tvirtumą yra rasti, kiek informacijos reikia tam, kad iškreipti reitingus.

Tarkime U_T ir I_T - naudotojų ir elementų rinkinių aibė testiniuose duomenyse. Kiekvienai naudotojo elemento porai (u,i) prognozės pokytis matuojamas taip $\delta_{u,i} = p\prime_{u,i} - p_{u,i}$, kur p ir $p\prime$ yra prognozės prieš ir po atakos atitinkamai. Tarkime, kad pokytis yra didelis, tačiau elementas vis tiek nepatenka į rekomenduojamų elementų sąrašą. Čia gali padėti kita metrika - pataikymo santykis (angl. hit ratio). Tarkime R_u - geriausių N rekomendacijų naudotojui u aibė. Jeigu elementas pasirodo R_u, H_{ui} įgyja reikšmę 1, priešingu atveju 0. Pataikymo santykis elementui i - $HitRatio_i = \sum_{u \in U_T} H_{ui} \setminus |U_T|$. Vidutinis pataikymo santykis tada yra pataikymo santykių kiekvienam elementui suma padalinta iš elementų skaičiaus.

4. Pasitikėjimu pagrįstos rekomendacinės sistemos modeliavimas

Šio darbo tyrimo objektas - naujos tinklinių programų kartos atstovė - socialinė RS. Ji generuoja prognozes apie naudotojams galinčius patikti elementus iš tam tikros, paprastai labai didelės aibės, remdamosi tarpusavio naudotojų santykiu. Sihna ir Swearingen [SS01] palygino reakcijas į RS ir draugų suteiktas rekomendacijas ir parodė, kad žmonės labiau pasitiki rekomendacijomis gautomis iš pažįstamų žmonių nei iš sistemos, veikiančios juodos dėžės (angl. black box) principu. Žinant, kad socialiniai tinklai vis populiarėja, o besinaudojančiųjų skaičius viršija milijardą, nesunku suprasti, kodėl RS kartu su socialiniais tinklais yra populiarus tyrimų objektas.

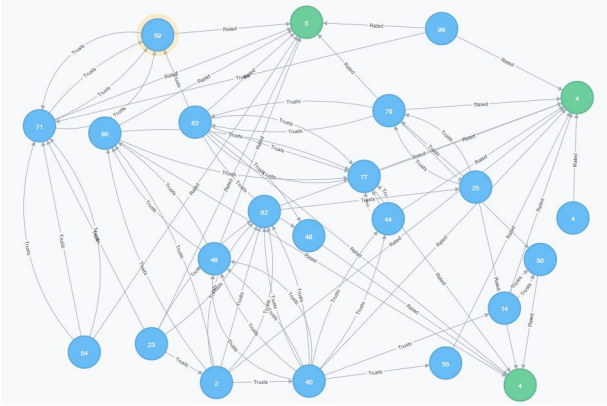
Tokiose sistemose naudotojas gauna rekomendaciją elemento, turinčio aukštą įvertinimą naudotojo WOT - pasitikėjimo tinkle (angl. web of trust). Pagrindiniai tokių sistemų įrankiai yra agregavimo ir propagavimo operatoriai. Propagavimo operatorius taiko pasitikėjimo tranzityvumo prielaidą - jeigu naudotojas u_1 pasitiki naudotoju u_2, o u_2 pasitiki u_3, tai u_1 pasitiki u_3. Agregavimo operatorius apjungia kelis pasitikėjimo įverčius į vieną.

Tikimybiniu požiūriu pasitikėjimas gali įgyti tik dvi reikšmes - arba kitu naudotoju galima pasitikėti (su tikimybe p), arba ne. Kitas, labiau įtikinantis ir panašesnis į realybę, yra laipsniškas požiūris, teigiantis, kad galima pasitikėti arba nepasitikėti tik iš dalies. Šiuo požiūriu pasitikėjimas nėra vertinamas kaip tikimybė, didesnė reikšmė tiesiog reiškia didesnį pasitikėjimą. Čia galima pastebėti ir analogiją su realiu gyvenimu - vienais žmonėmis pasitikime daugiau, kitais mažiau.

Šiame darbe siūlomas sričių panašumo metodas remiasi nauju duomenų aplinkos interpretavimu. Iki šiol buvo kalbėta apie sistemas, kuriose naudotojai vieni kitiems priskiria tam tikrus skaitinius pasitikėjimo įverčius. Šiame darbe siūloma praplėsti šį apibrėžimą iki bendresnio atvejo, kuriame galimos kelios pasitikėjimo sritys. Taigi vienas naudotojas kitam gali priskirti kelis įverčius pagal pasitikėjimo sritis, kitaip tariant, vienas naudotojas kitam priskiria pasitikėjimo vektorių. Taip pat, tinklo dalyviai gali būti tarpusavyje susiję ir be išreikšto pasitikėjimo įverčio, tai yra pasitikėjimas traktuojamas kaip neprivalomas esamo santykio atributas. Tada santykį tarp bet kurių u_1 ir u_2, galime užrašyti kaip $r_{u_1}(u_2) = (e_{u_1}(u_2), \boldsymbol{t}_{u_1}(u_2)), e_{u_1}(u_2) \in \{0,1\}, t^k_{u_1}(u_2) \in [0,1]$, kur $k = 1,..,N$, o N - pasitikėjimo sričių skaičius. e rodo ar tinklo dalyviai turi ryšį, o $t_{u_1}(u_2)$ rodo naudotojo u_1 pasitikėjimą naudotoju u_2, kuris, kai $e = 0, t_{u_1}(u_2) = \emptyset$. Pačias pasitikėjimo sritis žymėsime $T_1, T_2..., T_N$.

Pasitikėjimo įverčius naudotojai vieni kitiems priskiria rankiniu būdu, remdamiesi savo nuomone apie kitų naudotojų patikimumą. Realioje sistemoje problema, kaip sužinoti apie naudotojų tarpusavio pasitikėjimą, galėtų būti sprendžiama iš žmogaus ir kompiuterio sąveikos projektavimo požiūrio taško. Toks projektavimas, be abejo, priklausytų nuo aplinkos, kurioje norime įgalinti naudotojus išreikšti vienų kitais pasitikėjimą. Sprendimas galėtų būti pavyzdžiui toks:

- naudotojas pažymi, kad jam patinka kito naudotojo nuomonė
- sistema įvertina, kad naudotojas u_1 matė n naudotojo u_2 vertinimų ir jam patiko m jų
- remiantis šia informacija sudaromas ir išsaugomas pasitikėjimo įvertis



1 pav. RS grafo su skirtingomis kategorijomis fragmentas

Kitas scenarijus yra, kai norime priskirti pasitikėjimą ne apžvalgininkui, o kitam asmeniui (pavyzdžiui, draugui). Tuomet galima tiesiog nueiti į to asmens anketą ir joje užpildyti pasitikėjimo įvertį (skalėje nuo 1 iki 5).

Jeigu žinomas panašumo tarp naudotojų u ir v įvertis $sim(u,v)$, galima inicializuoti pasitikėjimą šiuo įverčiu ir esant progai paklausti naudotojo, ar jo pasitikėjimas naudotoju v yra lygus $sim(u,v)$. Toks metodas ypač aktualus, kai kalbama apie kelių pasitikėjimo sričių RS ir norime žinoti pasitikėjimus kiekvienoje jų. Šių ir kitų duomenų išgavimo būdų efektyvumo patvirtinimas arba paneigimas neįeina į šio darbo apimtį.

4.1. Rekomendacinės sistemos duomenų generavimo metodas

Šiuo metu nėra tokio prieinamo duomenų rinkinio, tinkančio atliekamam tyrimui apie RS, kurioje elementai priklauso kategorijoms ir naudotojai išreiškia pasitikėjimą kategorijose. Dėl šios priežasties dalis tyrimo skirta RS duomenų modelio sudarymui ir duomenų generavimui. Siekiama sukurti duomenų struktūrą, turinčią penkis elementus. Šie elementai yra:

- Kategorijos
- Naudotojai
- Elementai (vertinami produktai) priklausantys kategorijoms
- Naudotojo tarpusavio pasitikėjimai kategorijose (tolydi reikšmė tarp 0 ir 1)
- Naudotojų reitingai, priskirti elementams

Pav. 1 pavaizduotas RS duomenų fragmentas. Žalios viršūnės žymi elementus, priklausančius kategorijoms, mėlynos - naudotojus, briaunos tarp naudotojų žymi pasitikėjimą skirtingose kategorijose, o briaunos tarp žalių ir mėlynų viršūnių - naudotojų vertinimus apie elementus. Toliau aprašyti kiekvieno iš elementų generavimo algoritmai.

4.1.1. Kategorijos

Kategorijų modeliavimas - pirmas algoritmo žingsnis. Juo siekiama apibrėžti ne tik kategorijas, kurioms gali priklausyti elementai bet ir nusakyti kiekvienos kategorijos charakteristikas, kas lems kiekvieno elemento bruožus bei naudotojų reitingus elementams. Visų pirma, sudaroma kategorijų matrica. Sudarydami kategorijų matricą, turime įvertinti kiek kategorijų yra RS ir iš kiek charakteristikų jos susidaro.

1 lentelė. Kategorijų matrica bendru atveju

Kategorijos	T_1	T_2	T_3	...	T_n
x_1	c_{11}	c_{12}	c_{13}	...	c_{1n}
x_2	c_{21}	c_{22}	c_{23}	...	c_{2n}
x_3	c_{31}	c_{32}	c_{33}	...	c_{3n}
...					
x_m	c_{m1}	c_{m2}	c_{m3}	...	c_{mn}

čia

$$\sum_{i=1}^{m} c_{ij} = 1, \forall j = 1,..,n \text{ ir } c_{ij} \leq 1, \forall j = 1,..,n, i = 1,..,m \tag{21}$$

T_1,...T_n žymi kategorijas, o x_1,.., x_m - charakteristikas. Parinktos charakteristikos gali atitikti kategorijas, kaip kad kitame skyriuje generuojamų duomenų rinkinio atveju, bet nebūtinai.

Kategorijų matrica bus naudojama generuojant elementus. Nuo to, kokie yra elementai priklauso tai, kaip juos vertina naudotojai, o nuo to priklauso ir tai, kaip jie vertina vienas kito patikimumą. Taigi, ši matrica - RS duomenų generavimo pagrindas.

4.1.2. Naudotojai

Naudotojas apibrėžiamas kaip vektorius $(y_1, y_2,..., y_m, q)$, kur $\sum_{i=1}^{m} y_i = 1$ ir $q \in [0,1]$. $y_1, y_2,..., y_m$ reiškia naudotojo pirmenybes - kiek svarbus jam yra tam tikras bruožas elemente, m čia - charakteristikų kiekvienoje kategorijoje skaičius. Kokybės parametras q rodo, kiek naudotojas yra jautrus elemento kokybei. Dėl kokybės parametro naudojimo net jei naudotojui apskritai nepatinka tam tikrai kategorijai priklausantys elementai, bet jis yra jautrus kokybei ir elementas turi aukštą kokybės koeficientą, tikėtina, kad naudotojas gerai vertins tą elementą.

Praktiškai algoritmas realizuojamas taip:

- sugeneruojame 5 atsitiktinius skaičius tarp 0 ir 1 (skirstinys priklauso nuo to, kokį duomenų rinkinį siekiama sugeneruoti, šiame tyrime naudojamas tolygusis skirstinys)
- randame jų sumą
- kiekvienam bruožui priskiriame reikšmę lygią pirmame žingsnyje sugeneruotai reikšmei padalintai iš visų reikšmių sumos
- kokybės parametrui priskiriame atsitiktinę reikšmę tarp 0 ir 1

Taip užtikriname, kad naudotojai yra tikrai atsitiktiniai ir įvairūs pirmenybių prasme - naudotojui gali patikti elementai iš įvairių, tarpusavyje nepanašių kategorijų.

4.1.3. Elementai

Elementas apibrėžiamas vektoriumi $(c, z_1, z_2, .., z_m, q)$. Čia c nurodo, kuriai kategorijai priklauso elementas, parametrai $z_1, z_2,.., z_m$ rodo, kiek elementas pasižymi kiekvienu bruožu, o q - kokybės parametras. Generuojant elementus negalime taikyti tokio paties metodo, kaip naudotojo atveju, nes elementas priklauso tik vienai kategorijai, o tai reiškia, kad bruožų reikšmės negali būti visiškai atsitiktinės. Jas generuojame pasinaudodami normaliuoju skirstiniu su vidurkiu lygiu reikšmei gautai iš kategorijų matricos, aprašytos skyrelyje apie kategorijas ir parinktu standartiniu nuokrypiu (tokiu, kad duomenys būtų panašūs į realius - parinkus per didelį rezultatai gaunasi labai triukšmingi). Vidurkis parenkamas taip: pažiūrėję į c reikšmę atfiltruojame kategorijų matricoje kategoriją (stulpelį). Tada turime vidurkių, naudojamų generuojant $z_1, z_2,.., z_m$, vektorių.

4.1.4. Reitingai

Naudotojo reitingai elementams generuojami naudojant jo pirmenybes ir reiklumo kokybei parametrą bei atitinkamus produkto parametrus. Siekiama, kad reitingų pasiskirstymas būtų kuo artimesnis tikrovei, tai reiškia - nebūtų pasiskirstę galimų reikšmių kraštuose arba pernelyg vienodi.

Kiekvienam naudotojui parenkamas atsitiktinis įvertintų elementų skaičius naudojant normalųjį atsitiktinį dydį. Kiekvienam atsitiktinai parinktam elementui generuojamas reitingas taikant tokią formulę:

$$r_u(p) = (1 - q_u)\sqrt{pos(corr(Z_u,Y_p))} + q_u q_p \tag{22}$$

čia

- $r_u(p)$ - naudotojo u reitingas elementui p
- q_u - naudotojo u kokybės reiklumo parametras
- q_p - elemento p kokybės parametras
- Z_u - naudotojo u pirmenybių rinkinys
- Y_p - elemento p bruožų rinkinys
- $pos(x)$ - $f[-1,1] -> [0,1]$

Ši formulė sukurta atsižvelgiant į tokius reikalavimus:

- jeigu elementas idealiai atitinka naudotojo pirmenybes ($z_i = y_i, \forall i$), tai naudotojas priskiria maksimalų reitingą
- jeigu naudotojo jautrumas kokybei lygus 1 ir elemento kokybė lygi 1, tai priskiriamas maksimalus reitingas

- priskirtas reitingas negali viršyti maksimalaus reitingo
- didėjant koreliacijai tarp naudotojo pirmenybių ir elemento charakteristikų reitingas turi didėti

Tyrimo eigoje pastebėta, kad koreliacijos funkcijos įtaka pernelyg maža, todėl ji padidinama naudojant pasirinktą iškilią funkciją (šiuo atveju šaknis suteikia pageidaujamą efektą).

4.1.5. Pasitikėjimai

Pasitikėjimo reikšmės - svarbiausios prognozuojant reitingus, parodančios kokį svorį suteikti patikėtinio nuomonei apie elementą. Šiame tyrime naudotojai vieni kitais pasitiki kategorijos lygmenyje. Buvo išbandyti du pasitikėjimo reikšmių generavimo būdai.

Taikant pirmąjį būdą pasitikėjimas tarp dviejų naudotojų tam tikroje kategorijoje generuojamas lyginant naudotojų tarpusavio pirmenybes tos kategorijos atžvilgiu. Taigi pasitikėjimas kategorijoje T_1 tarp naudotojų $u(0.1, 0.2, 0.2, 0.5, 0, q_u)$ ir $v(0.2, 0.2, 0.2, 0.2, 0.2, q_v)$ randamas taip:

$$t_u(v) = \max(x_1^u, x_1^v) - \min(x_1^u, x_1^v) = 0.2 - 0.1 = 0.1 \tag{23}$$

Tokiu būdu rasti pasitikėjimai tenkina šias savybes:

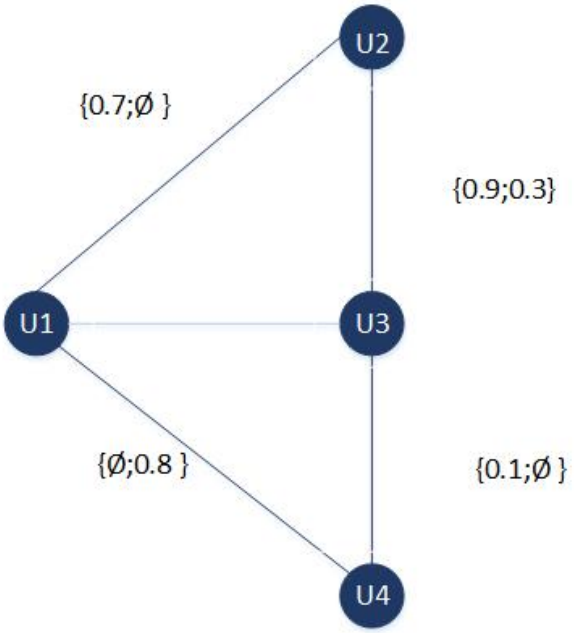
- yra intervale tarp 0 ir 1
- nepriklauso nuo kategorijų skaičiaus

Tolimesnis tyrimas parodė, kad šis būdas nėra pakankamai geras. Pagrindinė to priežastis ta, kad vertinant pasitikėjimą tam tikroje kategorijoje naudojamas tik vienas (tą kategoriją atitinkantis) bruožas, o kategorijos savaime nėra vienalytės - jos turi įvairių bruožų, kurie aprašyti kategorijų matricoje. Taigi, jei kategorijos vienareikšmiškai atitiktų charakteristikas ir kategorijų matrica būtų vienetinė - šis būdas būtų efektyvesnis.

Kitas būdas geresnis - jis, nors ir netiesiogiai, atsižvelgia į kategorijų matricą. Naudotojų, kurie pasitiki vienas kitu, poros ir kategorijos, kurioms generuojamas pasitikėjimas, parenkami atsitiktinai, kaip ir ankstesnio būdo atveju. Naudotojų porai pasitikėjimas generuojamas taip:

- parenkami n atsitiktinių elementų iš atitinkamos kategorijos ir jiems generuojami abiejų naudotojų reitingai (kaip aprašyta ankstesniame skyrelyje)
- turint abiejų naudotojų reitingų vektorius, galime rasti panašumą tarp jų taikant vieną iš panašumo metrikų
- rastas panašumas transformuojamas taip, kad priklausytų intervalui tarp 0 ir 1, o tada prilyginamas pasitikėjimui

Taikant tokį metodą atsižvelgiama į visas kategorijų charakteristikas. Tai labai svarbu tolimesniam tyrimui, ypač panašumo tarp sričių įvertinimui, kuris nagrinėjamas tolimesniuose skyriuose.



2 pav. Pasitikėjimo grafo fragmentas

5. Sričių panašumo metodas

Iš pradžių panagrinėkime paprastą pavyzdį. Tarkime, kad turime situaciją pavaizduotą pav. 2, kuriame pateikti naudotojų tarpusavio pasitikėjimai dviejose kategorijose, ir norime žinoti, kiek u_1 pasitiki u_3 srityje T_2. Tiesioginio kelio nėra, nes abiejuose galimuose keliuose - $u_1 - u_2 - v$ ir $u_1 - u_4 - v$ yra trūkstamų duomenų - pirmu atveju nežinome $t^2_{u_1}(u_2)$, antru - $t^2_{u_4}(u_3)$, tačiau matome, kad egzistuoja kelias $u_1 - u_2 - u_3$, pagal kurį galime įvertinti $t^1_{u_1}(u_3)$

$$t^1_{u_1}(u_3) = t^1_{u_1}(u_2) \times t^1_{u_2}(u_3) = 0.7 \times 0.9 = 0.63$$

Žinodami, kad sričių panašumas $sim(T_1, T_2) = 0.9$, gauname

$$t^2_{u_1}(u_3) = t^1_{u_1}(u_3) \times sim(T_1, T_2) = 0.63 \times 0.9 = 0.6048$$

Iš tiesų, šis pavyzdys nėra labai paprastas - jis susideda iš dviejų žingsnių. Pirmo žingsnio metu įvertinamas naudotojų u_1 ir u_3 tarpusavio pasitikėjimas srityje T_1 taikant propagavimo (daugybos) operatorių, aptartą ankstesniame skyrelyje, o po to pritaikytas sričių panašumo metodas. Aptarkime šį metodą formaliau.

Siūlomas metodas susideda iš dviejų etapų. Pirmas etapas skirtas panašumo tarp sričių radimui. Panašumas tarp sričių gali būti randamas globaliai - visai sistemai, arba kiekvienam naudotojui atskirai (jei tik naudotojas turi pakankamai duomenų). Šiame darbe siūlomi metodai:

- $GTDS$ (angl. global trust-based domain similarity) esmė - turint naudotojų porų, kurios turi tarpusavio pasitikėjimą dviejose kategorijose (kurių panašumo ieškome) sąrašą, ieškoti Pearson'o koreliacijos tarp pasitikėjimų abiejose kategorijose.

- $UTDS$ (angl. user-level trust-based domain similarity) randamas panašiai, kaip ir $GTDS$.

Skirtumas toks, kad koreliacijos ieškome ne tarp visų naudotojų esančių sistemoje, o tik tarp tų, kuriais pasitiki naudotojas, kuriam norime įvertinti jo asmeninį kategorijų panašumo suvokimą.

- $CMDS$ (angl. category matrix domain similarity) panašumą tarp kategorijų randa, ieškant koreliacijos tarp sričių charakteristikų kategorijų matricoje. Šis metodas įdomus teorine prasme - realiose RS kategorijų matrica nežinoma.

Algorithm 1 $GTDS$ metodas panašumo tarp sričių radimui

```
1:  procedure GetCategorySimilarity
2:      float[] trusts1;
3:      float[] trusts2;
4:      users ← GetAllUsers();
5:      foreach(var user in users):
6:          trustees ← user.GetTrusteesWithTrustInCategories(T1, T2);
7:          foreach(var trustee in trustees):
8:              trusts1.Add(trustee.T1);
9:              trusts2.Add(trustee.T2);
10:     similarity ← Correlation.Pearson(trusts1, trusts2);
11: end procedure
```

Kai panašumas tarp sričių jau žinomas, lieka atsakyti į klausimą - kaip ši informacija gali padėti įvertinti pasitikėjimą tarp naudotojų. Tyrime bus išbandyti du metodai:

- $MAXDS$ metodas. Tarkime, kad turime naudotojų porą su žinomais pasitikėjimais n sričių ir nežinomais m. Norėdami įvertinti nežinomus pasitikėjimus, parenkame tą žinomą pasitikėjimo reikšmę, kuri yra didžiausia ir naudodami ją kaip pagrindą, nežinomas randame sudauginę ją su atitinkamos kategorijos panašumu. Šio metodo trūkumas tas, kad atsižvelgiama ne į visą žinomą informaciją.

- $AVGDS$ metodu siekiama panaudoti visą žinomą informaciją. Nežinomos pasitikėjimo reikšmės randamos ieškant randant žinomų pasitikėjimų sudaugintų su sričių panašumu vidurkį su svoriais. Svoriai šioje formulėje - tie patys sričių panašumai.

$$t_u^{T_i}(v) = \frac{\sum\limits_{j \in T} t_u^{T_j}(v) \times sim(T_i, T_j)^2}{\sum\limits_{j \in T} sim(T_i, T_j)} \tag{24}$$

Labai svarbus dydis, kurį reikia nustatyti taikant šiuos metodus - slenkstis, nuo kurio saugojami pasitikėjimo įverčiai. Jei vienas naudotojas pasitiki kitu vienoje srityje ir norime prognozuoti pasitikėjimą kitoje, kuri yra visiškai nepanaši, siūlomi metodai grąžins mažą pasitikėjimo įvertį. Tačiau jeigu sritys nepanašios, tai remdamiesi pasitikėjimo reikšme vienoje, nieko negalime pasakyti apie pasitikėjimą kitoje. Taigi, slenksčio esmė - atfiltruoti nereikalingą informaciją.

Algorithm 2 MAXDS algoritmas trūkstamų pasitikėjimų tarp dviejų naudotojų radimui

```
1: procedure GetMissingTrust
2:     threshold ← 0.6;
3:     trusts ← GetTrusts(user1, user2);
4:     allCategories ← GetAllCategories();
5:     maxTrust ← (category, TrustValue);
6:     foreach(trust in trusts):
7:         If trust.TrustValue > maxTrust.TrustValue Then
8:             maxTrust ← trust;
9:         EndIf
10:    categoriesWithMissingTrust = allCategories.Except(trusts.categories);
11:    foreach(category in categoriesWithMissingTrust):
12:        categorySimilarity ← GetCategorySimilarity(maxTrust.category, category);
13:        newTrust ← (category, maxTrust × categorySimilarity);
14:        If newTrust ≥ threshold Then
15:            newTrust.Save();
16:        EndIf
17: end procedure
```

Algorithm 3 AVGDS algoritmas trūkstamų pasitikėjimų tarp dviejų naudotojų radimui

```
1: procedure GetMissingTrust
2:     threshold ← 0.6;
3:     trusts ← GetTrusts(user1, user2);
4:     allCategories ← GetAllCategories();
5:     maxTrust ← (category, TrustValue);
6:     categoriesWithMissingTrust = allCategories.Except(trusts.categories);
7:     foreach(category in categoriesWithMissingTrust):
8:         numerator ← 0
9:         denominator ← 0
10:        foreach(trust in trusts):
11:            categorySimilarity ← GetCategorySimilarity(trust.category, category);
12:            numerator ← numerator + categorySimilarity × categorySimilarity ×
    trust.TrustValue;
13:            denominator ← denominator + categorySimilarity;
14:            newTrust ← numerator / denominator;
15:            If newTrust ≥ threshold Then
16:                newTrust.Save();
17:            EndIf
18: end procedure
```

6. Eksperimentas

6.1. Problemos ir iššūkiai

Didžiausia problema šio tyrimo srityje yra realių duomenų nebuvimas ir negalėjimas praktiškai įvertinti šių metodų tinkamumo. Nėra žinomo socialinio tinklo, kuriame naudotojai išreikštų pasitikėjimą vienas kitu tolydžioje skalėje ir pasitikėjimai galėtų būtų priskirti skirtingose kategorijose. Artimiausias šiems reikalavimams Epinions.com duomenų rinkinys naudotas šiame tyrime netenkina šių dviejų reikalavimų - tai yra viena priežasčių, kliudžiusių atlikti išsamesnį tyrimą su realiais duomenimis. Dėl šios priežasties, nemaža tyrimo dalis skirta duomenų rinkinio generavimui.

Kita problema susijusi su RS vertinimu. Negalima vienareikšmiškai apibrėžti, kokia RS yra gera. Egzistuoja nemažai kriterijų, pagal kuriuos galime vertinti RS - tiek tikslumas ir kriterijai, kuriuos jis apima (vidutinė absoliuti klaida, vidutinė kvadratinė klaida, normalizuoti šių matų atitikmenys), tiek ir tam tikrų savybių tenkinimas (naujoviškumas, įžvalgumas, tikslumas, atsparumas atakoms, padengimas), tačiau RS kūrėjai turi apsispręsti, kurie kriterijai yra svarbesni, o kurie mažiau svarbūs. Kitaip sakant, reikia atsakyti į tokius klausimus kaip: ar geriau sistema generuotų tikslias rekomendacijas net jeigu naudotojas jau žino apie visus elementus iš anksčiau ar jau verčiau kartais suklysta, bet dažnai pasiūlo kažką naujo? Priimant sprendimą būtina atsižvelgti į dalykinę sritį. Vis dėlto, parinkti tinkamus reikalavimus yra didelis iššūkis analitikams, nes reikia atsižvelgti ne tik sistemos tikslumą, bet ir žmonių reakcijas į rekomendacijas. Kadangi šio tyrimo tikslas - ištirti metodus, siekiančius padėti sudaryti rekomendacijas mažai duomenų turintiems naudotojams, buvo koncentruotasi ties dviem RS vertinimo aspektais - tikslumu ir padengimu.

Trečia problema - technologinė. Darbas su dideliais grafais reikalauja technologijų optimizuotų tokiems duomenims. Dėl šios priežasties tyrimas buvo atliktas su nedidelės apimties imtimi. Algoritmus realizuojantis kodas buvo parašytas .NET aplinkoje C# ir F# kalbomis, duomenys saugomi ir kai kurios grafų operacijos (pavyzdžiui, trumpiausio kelio radimas) atliekamos NoSql neo4j grafų duomenų bazėje. Tinkamų technologijų parinkimas ir architektūros sudarymas šiame tyrime nagrinėtiems uždaviniams spręsti - potenciali tolimesnė šio tyrimo dalis.

6.2. Duomenų rinkinio sudarymas

Remiantis ankstesniame skyriuje pateiktu algoritmu, čia bus sudaryti du duomenų rinkiniai, su kuriais bus atliekami eksperimentai. Pirmą duomenų rinkinį vadinsime DS1, jame kategorijos skirtingos. Kitas rinkinys - DS2 sugeneruotas taip, kad jo kategorijos būtų panašios. Lygindami rezultatus, gautus taikant metodus abiem duomenų rinkiniams, galėsime įvertinti kaip veikia sričių panašumo metodai skirtingiems duomenų rinkiniams.

Duomenų rinkinio DS1 duomenys gali būti vertinami kaip filmų rekomendacinės sistemos duomenys. Apibrėžiame, pavyzdžiui, tokias penkias kategorijas:

- T_1 - drama
- T_2 - komedija

- T_3 - siaubo
- T_4 - trileris
- T_5 - fantastika

Toks žanrų priskyrimas kategorijoms yra tik pavyzdys skirtas tam, kad būtų lengviau suvokti panašumą tarp sričių. Toliau apibrėžiame, kiek kiekviena iš šių kategorijų yra susijusi su kitomis. Euristiškai, remdamiesi subjektyviu suvokimu apie kategorijas, sudarome matricą pavaizduotą lent. 2:

2 lentelė. Kategorijų matrica SP1

Kategorijos	T_1	T_2	T_3	T_4	T_5
x_1	0.55	0.2	0.2	0.2	0.3
x_2	0.2	0.6	0.05	0.05	0.1
x_3	0.05	0.05	0.35	0.1	0.05
x_4	0.1	0.05	0.2	0.65	0.05
x_5	0.1	0.1	0.2	0	0.5

Čia $T_1, .., T_5$ žymime kategorijas, o $x_1, .., x_5$ kategorijas atitinkančius požymius (toliau - charakteristikas). Taigi iš šios matricos galime teigti, kad pavyzdžiui:

- T_4 (trileris) yra gryniausias žanras, tai yra, turintis daugiausiai savo kategoriją atitinkančio požymio (kadangi turi didžiausią matricos įstrižainėje esančią reikšmę)
- T_3 (siaubo) - mažiausiai gryna kategorija (nes bruožų pasiskirstymas yra tolygiausias)
- T_4 kategorija neturi x_5 bruožo (trileris neturi fantastikos bruožų)

Akivaizdu, kad šie teiginiai yra subjektyvūs. Didesnio objektyvumo galima pasiekti, pavyzdžiui, sudarant kategorijų matricą remiantis apklausų duomenimis, o ne savo subjektyvia nuomone.

Bendru atveju RS gali turėti n kategorijų ir kiekviena jų gali susidaryti iš m charakteristikų. Šiame tyrime $n = m = 5$ pasirinkta siekiant sukurti ne pernelyg paprastą, tačiau tuo pat metu ir ne per sudėtingą modelį, artimą realiam scenarijui.

Analogiškai sudarome ir DS2 duomenų rinkinio kategorijų matricą, kurioje kategorijos yra tarpusavyje panašios. Tuo galėtų pasižymėti, pavyzdžiui, elektronikos prekių RS su kategorijomis - nešiojami kompiuteriai, planšetiniai kompiuteriai, išmanieji telefonai ir panašiai.

3 lentelė. Kategorijų matrica SP2

Kategorijos	T_1	T_2	T_3	T_4	T_5
x_1	0.6	0.5	0.4	0.6	0.3
x_2	0.2	0.1	0.2	0	0.3
x_3	0.1	0.1	0.1	0.1	0.1
x_4	0.1	0.1	0.1	0.2	0.1
x_5	0	0.1	0.2	0.1	0.2

Čia generuojamo duomenų rinkinio atveju naudotojas yra vektorius $(y_1, y_2, y_3, y_4, y_5, q)$, elementas - $(c, z_1, z_2, z_3, z_4, z_5, q)$. Generuojant naudotojus, kokybės parametras parenkamas atsitiktinai pagal normalųjį skirstinį su vidurkiu 0.8 ir standartiniu nuokrypiu lygiu 0.3. Generuojant elementus, kokybės parametras parenkamas atsitiktinai pagal normalųjį skirstinį su vidurkiu 0.6 ir standartiniu nuokrypiu lygiu 0.4. Jei sugeneruota reikšmė didesnė už 1 arba mažesnė už 0, ji priskiriama 1 arba 0 atitinkamai. Abiems duomenų rinkiniams generuojame po 100 naudotojų ir 300 elementų.

Kiekvieno naudotojo aktyvumas (reitingų skaičius) generuojamas naudojant normalųjį skirstinį. Abiejų rinkinių atveju, atsižvelgiant į naudotojų ir elementų skaičių, parinktas normaliojo skirstinio vidurkis lygus 30 ir standartinis nuokrypis lygus 27. Vėlgi, parametrai parinkti tokie, kad duomenys būtų artimesni tikriems - didelis nuokrypis užtikrina, kad naudotojai pasižymės dideliu aktyvumo skirtumu - bus tiek labai aktyvių, tiek šalto starto naudotojų su mažai reitingų. Palyginimui, Epinions.com duomenų rinkinyje vieno naudotojo įvertintų elementų skaičius svyruoja nuo 0 iki 655. Didžiausias įmanomas reitingas $r_{max} = 5$.

Vertindami panašumą tarp naudotojų, kiekvienam įvertiname reitingus 12-ai elementų. Parinkdami tokį gana didelį parametrą, užtikriname, kad pasitikėjimo įvertis bus pakankamai objektyvus ir nenulemtas mažos imties. Žinoma, šis parametras parinktas atsižvelgiant į produktų, esančių RS skaičių, daugėjant galimų produktų aibei, būtų prasminga parinkti didesnę parametro reikšmę.

Tyrime bus aiškinamasi, kiek tiriami metodai yra efektyvūs skirtingiems socialinių tinklų tipams. Tam, kad būtų aišku, koks duomenų rinkinys naudotas konkrečiu atveju, gali būti pateikta generavimo parametrus nusakanti lentelė. Šiuo atveju abu duomenų rinkiniai generuojami naudojant vienodus parametrus, skiriasi tik kategorijų matrica.

Parametras	DS1	DS2
Elementų skaičius	300	300
Naudotojo kokybės parametro pasiskirstymas	$\mathcal{N}(0.7,0.3)$	$\mathcal{N}(0.7,0.3)$
Naudotojų skaičius	100	100
Elemento kokybės parametro pasiskirstymas	$\mathcal{N}(0.6,0.4)$	$\mathcal{N}(0.6,0.4)$
Naudotojo ryšių skaičiaus kategorijose pasiskirstymas	$\mathcal{N}(10, 9)$	$\mathcal{N}(10, 9)$
Naudotojo įvertintų elementų skaičiaus kategorijose pasiskirstymas	$\mathcal{N}(30,27)$	$\mathcal{N}(30,27)$
Vertinamas bendrų elementų skaičius ieškant pasitikėjimo	12	12
Kategorijų matrica	SP1 (lent. 2)	SP2 (lent. 3)

4 lentelė. Duomenų rinkinio DS1 charakteristikos

Charakteristika	DS1	DS2
Naudotojų, įvertinusių bent vieną elementą, skaičius	87	87
Šalto starto naudotojų skaičius	18	17
Ryžtingų naudotojų skaičius	6	6
Reitingų standartinis nuokrypis	1.5	1.47

5 lentelė. Reitingų pasiskirstymas duomenų rinkiniuose DS1 ir DS2

Duomenų rinkinys	1	2	3	4	5
DS1	612	345	370	730	839
DS2	532	339	373	786	849

Tam, kad galėtume ištirti metodo efektyvumą skirtingiems naudotojų tipams. Išskiriame du įdomius naudotojų tipus:

- Šalto starto naudotojai - tie, kurie yra įvertinę mažiau nei 15 elementų. Turint omenyje, kad RS egzistuoja 5 kategorijos, tai reiškia, kad vidutiniškai kiekvienoje kategorijoje, kurioje atliekamos prognozės, šalto starto naudotojai turi iki 3 reitingų
- Ryžtingi naudotojai - tie, kurie turi daugiau reitingų, tačiau jie yra pasiskirstę plačiai apie vidurkį. Tokiais laikysime naudotojus, kurių reitingų standartinis nuokrypis didesnis nei 2.

Rezultatai bus vertinami tiek visų naudotojų imčiai, tiek minėtoms naudotojų imtims.

6.3. Vertinimo kriterijai

Šio tyrimo tikslas – ištirti pasiūlytų metodų efektyvumą ir tikslumą sprendžiant šalto starto problemą rekomendacinėse sistemose. Tikslumas vertinamas naudojant "išimk vieną" metodą, kurio esmė tokia - iš duomenų išimamas vienas reitingas ir tada bandoma jį prognozuoti remiantis likusiais sistemos duomenimis. Tada vertinamas tikslumas ir padengimas. Tikslumas matuojamas taikant šias metrikas:

- MAE - vidutinė absoliuti klaida (angl. mean absolute error) skaičiuoja visų prognozės klaidų vidurkį. Ši metrika ne visiškai atspindi RS tikslumą, nes taip pat vertina ir daug duomenų turinčius ir šalto starto naudotojus.

$$MAE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} |\hat{r}_{ui} - r_{ui}|} \tag{25}$$

  Kadangi mažai duomenų turintiems naudotojams tikslumas gali būti mažesnis, Massa ir Avesani [AMT05] pasiūlė kitą metriką, kuri suvienodina vieno naudotojo reikšmę vertinant vidutinę klaidą - vidutinę absoliučią naudotojo klaidą.

- $MAUE$ - Vidutinė absoliuti naudotojo klaida (angl. mean absolute user error) skaičiuojama kiekvienam naudotojui atskirai, o tada randamas tų klaidų vidurkis. Ji skiriasi nuo MAE tuo, kad prognozės tikslumas kiekvienam naudotojui turi vienodą svorį, o MAE labiau atsižvelgia į aktyvesnius naudotojus

- $RMSE$ - kvadratinė vidutinė klaida (angl. root mean squared error) - viena populiariausių metrikų, panaši į vidutinę absoliučią klaidą

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (\hat{r}_{ui} - r_{ui})^2} \tag{26}$$

Kitas vertinimo kriterijų grupė ypač svarbi, kai kalbama apie šaltą startą. Padengimą (angl. coverage) vertinsime dviem būdais.

- RC - reitingų padengimo esmė - palyginti reitingų, kuriuos algoritmas sugebėjo įvertinti taikant "išimk vieną" metodą, skaičių su visų sistemoje esančių reitingų skaičiumi.
- UC - naudotojo padengimas lygina keliems naudotojams algoritmas sugebėjo prognozuoti bent vieną reitingą su skaičiumi naudotojų, kurie yra priskyrę reitingą bent vienam elementui.

Vertindami metodus skyriuje apie sričių panašumą naudosime tokį vertinimo kriterijų rinkinį - MAE, $MAUE$, $RMSE$, reitingų padengimą - RC, naudotojų padengimą - UC.

6.4. Tiriami metodai

Šiuo tyrimu siekiama nustatyti, kiek efektyviai sričių panašumo metodas padeda susidoroti su duomenų retumo ir šalto starto problema rekomendacinėse sistemose su skirtingomis kategorijomis. Buvo pristatyti žinomi metodai, kurie naudoja socialinių tinklų duomenis tam, kad taikydami propagavimo ir agregavimo operatorius nustatytų, kokie yra ryšiai tarp sistemos naudotojų. Taip pat buvo pasiūlytas sričių panašumo metodas, kurį taikant galima sužinoti, kiek sritys sistemoje yra tarpusavyje panašios, ir, žinant sričių panašumą bei pasitikėjimo įvertį tarp naudotojų vienoje srityje, įvertinti pasitikėjimą kitoje srityje. Abu metodai gali būti realizuojami skirtingais būdais.

Tyrime eksperimentai atlikti su dviem pasitikėjimo propagavimo metodais:

- $SHORTMULTI$ metodas kaip agregavimo metodą taiko trumpiausio kelio algoritmą, o propagavimui naudojamas daugybos operatorius
- $SHORTARI$ metodas kaip agregavimo metodą taiko trumpiausio kelio algoritmą, o propagavimui naudojamas vidurkio operatorius

Šie metodai parinkti atsižvelgiant į jų paprastumą ir loginį pagrįstumą. Golbeck disertacijoje [Gol05] parodyta, kad trumpesni keliai suteikia tikslesnę informaciją apie galimą naudotojų tarpusavio pasitikėjimą, taigi trumpiausio kelio panaudojimas čia pagrįstas. Taip pat šioje disertacijoje kalbama apie pasitikėjimo mažėjimą (angl. trust decay) ir tai, kad metodai kreipiantys dėmesį į šį reiškinį grąžina tikslesnes prognozes. Čia tiriamas $SHORTMULTI$ metodas atsižvelgia į pasitikėjimo mažėjimo reiškinį, o $SHORTARI$ - ne.

Tiriamo sričių panašumo metodo struktūra tokia:

- panašumas tarp sričių randamas vienu iš šių būdų:
  - $GTDS$
  - $UTDS$
  - $CMDS$
- trūkstami pasitikėjimo įverčiai gaunami taikant:

- $MAXDS$
- $AVGDS$

Visi aukščiau paminėti metodai gali būti kombinuojami įvairiais būdais. Eksperimento metu išbandytos keturių tipų kombinacijos, prognozuojančios naudotojų reitingus

- Sričių panašumas + BF
- Propagavimas + BF
- Sričių panašumas + propagavimas + BF
- Propagavimas + sričių panašumas + BF

6.5. Rezultatai

6.5.1. Bendradarbiavimo filtravimas

Lent. 6 pavaizduoti rezultatai, gauti taikant naudotojo vidurkio metodą - reitingas prognozuojamas tiesiog imant naudotojo reitingų vidurkį. Lent. 7 pateikiami rezultatai, gauti taikant paprastą bendradarbiavimo filtravimo algoritmą. Palyginę šiuos rezultatus tarpusavyje - nors ir matome tikslumo ryžtingiems naudotojams pagerėjimą, tikslumas visiems naudotojams beveik identiškas. Taikant BF nei vienam iš 19 naudotojų, kurie buvo identifikuoti, kaip šalto starto naudotojai, nepavyko prognozuoti nei vieno reitingo. Tai tik dar kartą parodo, kokia aktuali yra šalto starto problema.

6 lentelė. Naudotojo vidurkio rezultatai RS duomenims DS1

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.10	0.96	1.44	1	1
Šalto starto naudotojai	1.40	1.82	1.99	1	1
Ryžtingi naudotojai	1.63	1.4	1.98	1	1

7 lentelė. BF rezultatai RS duomenims DS1

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.04	0.98	1.5	0.22	0.65
Šalto starto naudotojai	-	-	-	0	0
Ryžtingi naudotojai	1.17	1.17	1.84	0.06	0.17

8 lentelė. BF rezultatai taikomi RS duomenims DS2

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.04	0.91	1.51	0.21	0.63
Šalto starto naudotojai	-	-	-	0	0
Ryžtingi naudotojai	1.41	1.41	2.40	0.09	0.17

Abiejų tiriamų duomenų rinkinių DS1 ir DS2 rezultatai beveik vienodi. Tai vertinga žinoti, turint omenyje tai, kad bus tiriamas sričių panašumo metodas abiems duomenų rinkiniams.

6.5.2. Sričių panašumo metodas + BF

Pirmo sričių panašumo metodo žingsnio metu nustatomas panašumas tarp sričių. Tai galime padaryt dviem metodais - $GTDS$ ir $CMDS$. Gauti sričių panašumai taikant šiuos metodus duomenų rinkiniui DS1 pavaizduoti lent. 9 ir lent. 10.

9 lentelė. Panašumo tarp kategorijų matrica gauta taikant $GDTS$ metodą

Kategorijos	T_1	T_2	T_3	T_4	T_5
T_1	1	0.73	0.84	0.24	0.61
T_2	0.73	1	0.78	0.21	0.4
T_3	0.84	0.78	1	0.36	0.56
T_4	0.24	0.22	0.36	1	0.44
T_5	0.61	0.4	0.56	0.44	1

10 lentelė. Sričių panašumo matrica taikant $CMDS$ metodą

Kategorijos	T_1	T_2	T_3	T_4	T_5
T_1	1	0.62	0.37	0.48	0.63
T_2	0.62	1	0.08	0.30	0.43
T_3	0.37	0.08	1	0.53	0.46
T_4	0.48	0.30	0.53	1	0.26
T_5	0.63	0.43	0.46	0.26	1

Nors gauti sričių panašumai šiek tiek skiriasi, juose galima įžvelgti daugiau panašumų nei skirtumų.

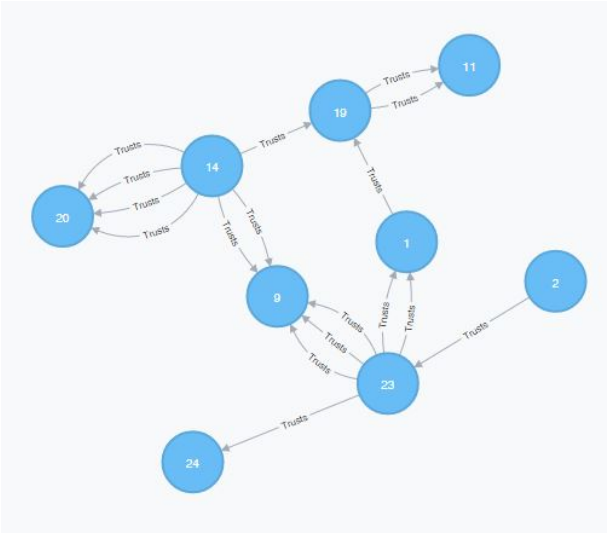
Sugeneruotiems duomenims išbandyti $AVGDS$ (sričių panašumo vidurkio) ir $MAXDS$ (sričių panašumo maksimumo) metodai su slenksčiu lygiu 0.6. Slenkstis nurodo, nuo kokios įvertintos pasitikėjimo reikšmės, pasitikėjimas yra išsaugomas.

11 lentelė. $AVGDS$ su slenksčiu 0.6 naudojant $GDTS$ sričių panašumus rezultatai taikomi duomenų rinkiniui DS1

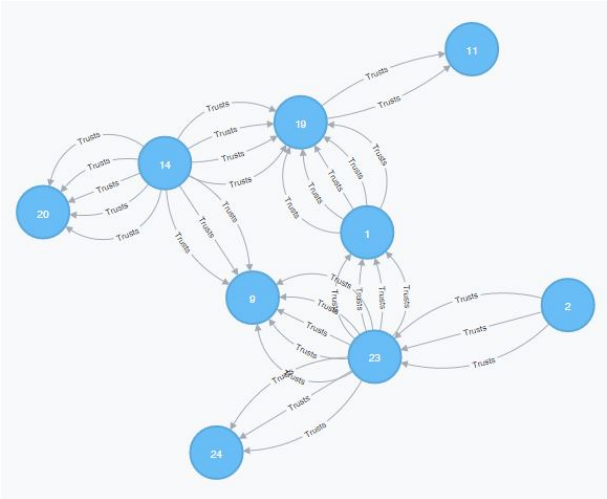
Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.04	0.98	1.49	0.33	0.69
Šalto starto naudotojai	-	-	-	0	0
Ryžtingi naudotojai	0.96	0.67	1.43	0.11	0.5

12 lentelė. $MAXDS$ su slenksčiu 0.6 naudojant $GDTS$ sričių panašumus rezultatai taikomi duomenų rinkiniui DS1

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.03	0.95	1.46	0.33	0.69
Šalto starto naudotojai	-	-	-	0	0
Ryžtingi naudotojai	1.03	0.69	1.59	0.11	0.5



3 pav. Pasitikėjimo pagal sritis grafo fragmentas prieš pritaikant sričių panašumo metodą



4 pav. Pasitikėjimo pagal sritis grafo fragmentas pritaikius sričių panašumo metodą

Lyginant rezultatus, gautus taikant $AVGDS$ su slenksčiu 0.6 (lent. 14) ir $MAXDS$ su slenksčiu 0.6 (lent. 12) su BF rezultatais lent. 7 matosi, kad visų naudotojų imčiai tikslumas nepasikeitė, o padengimas nežymiai pagerėjo. Didelis tikslumo padidėjimas ryžtingų naudotojų atveju iš dalies gali būti paaiškintas atsitiktinumu dėl mažos imties, tačiau atlikus daugiau eksperimentų pastebėta, kad tikslumas beveik visada nežymiai keičiasi į gerąja pusę, o reitingų padengimas didėja vidutiniškai apie 50% kiekvienai imčiai. 3 pav. ir 4 pav. matosi, kaip atrodo ryšių grafas prieš ir po sričių panašumo metodo pritaikymo.

13 lentelė. $AVGDS$ su slenksčiu 0.6 naudojant $GDTS$ sričių panašumus rezultatai taikomi duomenų rinkiniui DS2

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.03	0.92	1.49	0.24	0.66
Šalto starto naudotojai	-	-	-	-	-
Ryžtingi naudotojai	1.41	1.41	2.4	0.09	0.16

Pritaikius $AVGDS$ metodą duomenų rinkiniui DS2 gauti rezultatai lent. 13 parodo, kad sričių panašumo metodas duomenims su panašiomis kategorijomis veikia panašiai kaip ir duomenų rinkiniui DS1.

Taip pat $AVGDS$ metodas buvo išbandytas naudojant panašumus, gautus taikant $CMDS$ metodą. Taikant $AVGDS$ kartu su tokiais panašumais tikslumo prasme gauti prastesni rezultatai.

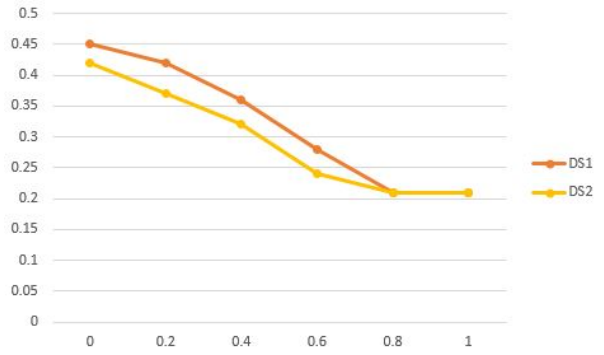
14 lentelė. $AVGDS$ rezultatai taikomi duomenų rinkiniui DS1 ir naudojant panašumus, gautus taikant $CMDS$ metodą su slenksčiu 0.6

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.12	1.1	1.69	0.37	0.69
Šalto starto naudotojai	-	-	-	0	0
Ryžtingi naudotojai	1.16	1.19	1.86	0.13	0.33

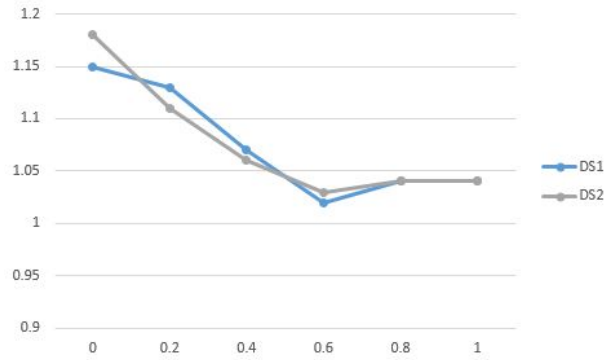
Trečio sričių panašumo metodas $UTDS$ įprastiems RS duomenims taikyti negalime, nes joks naudotojas neturi pakankamai duomenų, kad būtų galima jam įvertinti jo asmeninį sričių panašumą). Šį metodą taikysime kitame skyrelyje duomenims, kuriems pritaikytas propagavimo metodas.

Nepastebėta reikšmingo skirtumo tarp rezultatų, gautų taikant $MAXDS$ ir $AVGDS$ metodus, dėl to toliau bus taikomas tik $AVGDS$ metodas (nes jis atsižvelgia į daugiau informacijos). Taip pat, įvertinus tai, kad iš sričių panašumo metodų geriausiai veikia $GTDS$ metodas, tolimesniame tyrime taikysime tik jį.

Pav. 5 ir pav. 6 matomos tikslumo ir padengimo priklausomybės nuo slenksčio, kai taikomas $AVGDS$ metodas. Kaip ir galima tikėtis, mažinant slenkstį didėja padengimas ir, atitinkamai, mažėja tikslumas. Rezultatai rodo, kad tiek duomenų rinkiniui su panašiomis, tiek su skirtingomis kategorijomis algoritmas veikia beveik vienodai, tai yra, reikšmingo skirtumo nepastebėta. Šis pastebėjimas leidžia daryti išvadą, kad sričių panašumo metodą galima taikyti duomenų rinkiniams nepriklausomai nuo to, ar kategorijos panašios, ar ne. Iš tiesų, jei kategorijos skirtingos, tai pasitikėjimo įverčiai, sudaryti remiantis tuo mažu panašumu taip pat gaunami maži ir sudarant galutinę prognozę turi mažą svorį. Tai paaiškina, kodėl kategorijoms su skirtingomis kategorijomis galime



5 pav. Reitingų padengimo priklausomybė nuo slenksčio duomenų rinkiniams DS1 ir DS2



6 pav. Tikslumo (MAE) priklausomybė nuo slenksčio duomenų rinkiniams DS1 ir DS2

taikyti sričių panašumo metodą ir tikėtis efektyvių rezultatų. Kadangi rezultatų, gautų taikant metodą skirtingiems duomenų rinkiniams, reikšmingo skirtumo nepastebėta, toliau eksperimentai bus atliekami su DS1 duomenų rinkiniu.

6.5.3. Propagavimo metodas + BF

Abu duomenų rinkiniai buvo sugeneruoti parinkus tokius parametrus, kad juose egzistuotų duomenų retumo problema. Beje, RS su kategorijomis, ji dar opesnė. Anksčiau minėta, kad šalto starto naudotojai šiame tyrime yra tie, kurie turi mažiau kaip 15 reitingų. Vertinant įprastas RS šalto starto naudotojais vadinami tie, kurie turi 5 ar mažiau reitingų. Tačiau tiriama RS turi 5 kategorijas, taigi vienoje kategorijoje, šalto starto naudotojas turi vidutiniškai ne daugiau kaip 3 reitingus. Tai atsispindi mažose RC ir UC reikšmėse. Egzistuojantis problemos sprendimo būdas - taikyti metodus, vertinančius naudotojų tarpusavio pasitikėjimą. Tam pačiam duomenų rinkiniui išbandyti du trumpiausio kelio metodai, aprašyti ankstesniame skyrelyje ($SHORTMULTI$, $SHORTARI$). Siekiant vertinti tik svarbias pasitikėjimo reikšmes, nustatome slenkstį lygų 0.9, už kurį didesnius pasitikėjimo įverčius saugome.

15 lentelė. Rezultatai gauti taikant $SHORTMULTI$ su slenksčiu 0.9 RS duomenims DS1

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.05	0.99	1.62	0.51	0.87
Šalto starto naudotojai	1	0.83	1.55	0.23	0.55
Ryžtingi naudotojai	1.06	1.03	1.75	0.55	1

16 lentelė. Rezultatai gauti taikant $SHORTARI$ su slenksčiu 0.9 RS duomenims DS1

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.15	1.09	1.88	0.65	0.92
Šalto starto naudotojai	1.10	1.04	1.81	0.35	0.67
Ryžtingi naudotojai	1.27	1.31	2.33	0.65	1

Atlikus eksperimentus paaiškėjo, kad geriausiai turimiems duomenims veikia $SHORTMULTI$ metodas. Tai paaiškinama tuo, kad jis atsižvelgia į pasitikėjimo mažėjimą (angl. trust decay) esant ilgesniems pasitikėjimo keliams ir patvirtina tai, ką Golbeck įrodė savo tyrime [Gol05]. $SHORTARI$ labiau padidina padengimą, tačiau tikslumas sumažėja pernelyg smarkiai, kad toks metodas būtų vertingas praktikoje.

Kaip minėta anksčiau, sričių panašumo metodas gali būti taikomas nepriklausomai nuo metodų, prognozuojančių naudotojų tarpusavio pasitikėjimą naudojant propagavimo ir agregavimo operatorius. Jau parodyta, kaip globalus sričių panašumas veikia su baziniais RS duomenimis. Dabar bus siekiama ištirti, kaip veikia sričių panašumo ir agregavimo bei propagavimo metodų kombinacijos. Jau parodyta, kad geriausiai iš tiriamų agregavimo ir propagavimo metodų veikia $SHORTMULTI$ metodas, todėl toliau naudosime tik jį.

6.5.4. Sričių panašumo + propagavimo metodai + BF

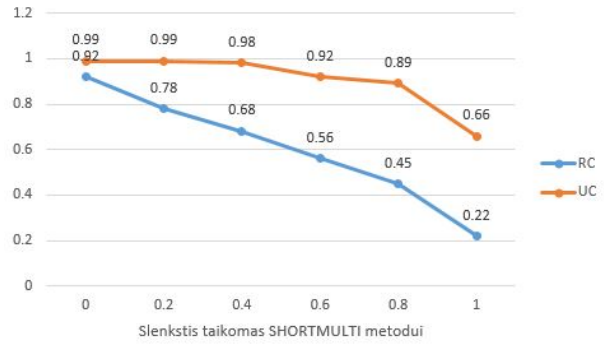
Šiame skyrelyje aprašytas eksperimentas, kai pirma taikomas sričių panašumo, o po to propagavimo metodas. Lyginant su ankstesniais eksperimentais gauti geresni rezultatai abiems duomenų rinkiniams tikslumo prasme. Padengimas šiuo atveju gautas mažesnis nei tiesiog $SHORTMULTI$ atveju dėl nustatyto slenksčio lygaus 0.9. Kadangi trumpiausiuose keliuose egzistuoja reikšmės įvertintos taikant sričių panašumą (o jos visos yra mažesnės nei 0.9), šį slenkstį viršija kur kas mažiau rastų kelių. Problemą galime spręsti mažindami slenkstį taikomą $SHORTMULTI$. Tada šalto starto naudotojams MAE lygus 1.05, o RS - 0.95. Prognozės generuojamos beveik visiems naudotojams, o tikslumas mažėja nežymiai.

17 lentelė. $AVGDS$ (sričių panašumai gauti taikant $GTDS$, su slenksčiu 0.6) + $SHORTMULTI$ (su slenksčiu 0.9), rezultatai taikomi RS duomenims DS1

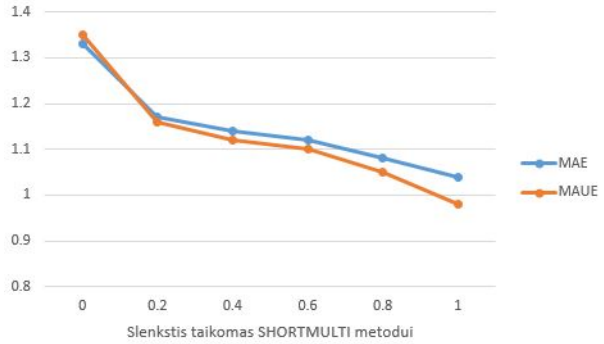
Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.01	0.99	1.46	0.43	0.86
Šalto starto naudotojai	0.87	0.96	1.88	0.2	0.5
Ryžtingi naudotojai	1.03	0.98	1.75	0.47	1

18 lentelė. $AVGDS$ (sričių panašumai gauti taikant $GTDS$, su slenksčiu 0.6) + $SHORTARI$ (su slenksčiu 0.9), rezultatai taikomi RS duomenims DS2

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	0.97	0.86	1.40	0.34	0.85
Šalto starto naudotojai	0.76	0.68	0.95	0.15	0.55
Ryžtingi naudotojai	0.96	0.4	1.61	0.38	0.67



7 pav. Padengimo priklausomybė nuo $SHORTMULTI$ metodui taikomo slenksčio, taikant sričių panašumo metodą $AVGDS$ kartu su pasitikėjimo propagavimo metodu $SHORTMULTI$



8 pav. Tikslumo priklausomybė nuo $SHORTMULTI$ metodui taikomo slenksčio, taikant sričių panašumo metodą $AVGDS$ kartu su pasitikėjimo propagavimo metodu $SHORTMULTI$

Nesudėtinga paaiškinti, kodėl padengimas taikant šiuos metodus sumažėjo - taikant pasitikėjimo propagavimo metodą su slenksčiu 0.9 duomenims, kuriems jau pritaikytas sričių panašumo metodas, randami keliai, su įvertintais pasitikėjimais, kurie visada mažesni už panašumą tarp sričių - taigi visi keliai, kuriuose egzistuoja bent vienas įvertintas pasitikėjimas sričių panašumo metodu, yra ignoruojami. Geriau atspindintys metodų veikimą rezultatai gaunami taikant pasitikėjimo propagavimo metodai su mažesniu slenksčiu.

19 lentelė. $AVGDS$ (sričių panašumai gauti taikant $GTDS$, su slenksčiu 0.6) + $SHORTMULTI$ (su slenksčiu 0.5), rezultatai taikomi RS duomenims DS1

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.09	1.08	1.53	0.72	0.99
Šalto starto naudotojai	1.05	1.07	1.54	0.45	0.94
Ryžtingi naudotojai	1.26	1.25	2.24	0.74	1

Matome, kad $SHORTMULTI$ metodas taikomas po to, kai buvo pritaikytas sričių panašumo metodas, priklausomai nuo parinktų slenksčių, grąžina daug geresnį padengimą nei atvejais kai metodai taikomi atskirai, beveik nepaveikiant tikslumo duomenų rinkiniui DS1. Duomenų rinkiniu su panašiomis kategorijomis propagavimo metodo nauda nėra tokia akivaizdi, nes taikant sričių panašumo metodą vieną patį gaunamas maksimalus padengimas išlaikant panašų tikslumą.

Pav. 7 ir pav. 8 matomi padengimo ir tikslumo rezultatų, gautų taikant sričių panašumo metodą su slenksčiu 0.6 kartu su $SHORTMULTI$ metodu, priklausomybė nuo slenksčio parametro,

taikomo pasitikėjimo propagavimo metodui. Iš esmės, šie rezultatai patvirtina vieną iš Golbeck [Gol05] darbo išvadų - pasitikėjimo keliai su mažomis pasitikėjimo reikšmėmis, grąžina prastesnius rezultatus tikslumo prasme ir todėl yra prasminga parinkti slenkstį, nuo kurio pasitikėjimo keliai turėtų būti ignoruojami. Kita vertus, pav. 7 rodo, kad kuo mažesnis slenkstis, tuo pasiekiamas didesnis padengimas. Šiuo konkrečiu atveju, prasmingiausia parinkti slenkstį lygų tarp 0.4 ir 0.6, nes, lyginant su rezultatais, gautais taikant didesnius slenksčius, tikslumas dar nėra suprastėjęs, o padengimas stipriai pagerėjęs.

6.5.5. Propagavimo + sričių panašumo metodai + BF

Buvo atlikti du eksperimentai naudojant panašumo matricą (lent. 10) trūkstamai informacijai apie pasitikėjimą užpildyti kartu su pasitikėjimo slenksčiu lygiu 0.6 ir 0.3. Pastebime, kad turint tokias panašumo reikšmes slenksčio reikšmė lygi 0.6 yra labai didelė - iš tiesų, taikydami šį metodą duomenų rinkiniui DS1, papildomos informacijos galime gauti tik apie T_1 ir T_2 bei T_1 ir T_5 kategorijų panašumus (nes lent. 10 matosi, kad tik šių kategorijų panašumų sandauga su žinomu pasitikėjimu, mažesniu už 1, gali viršyti 0.6).

Gauti tokie rezultatai rodo, kad sumažinus slenkstį padengimas padidėja, tačiau tikslumas sumažėja atitinkamai. Taigi, norint sužinoti, koks slenksčio parametras geriausias, reikia išsiaiškinti, kiek tikslumo galima paaukoti dėl didesnio padengimo.

20 lentelė. $SHORTMULTI$ + $AVGDS$ (su sričių panašumais, gautais naudojant $GTDS$, su slenksčiu 0.6), rezultatai taikomi RS duomenims DS1

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.07	0.96	1.56	0.55	0.61
Šalto starto naudotojai	1.02	1.05	1.79	0.3	0.56
Ryžtingi naudotojai	1.09	1.02	1.67	0.75	1

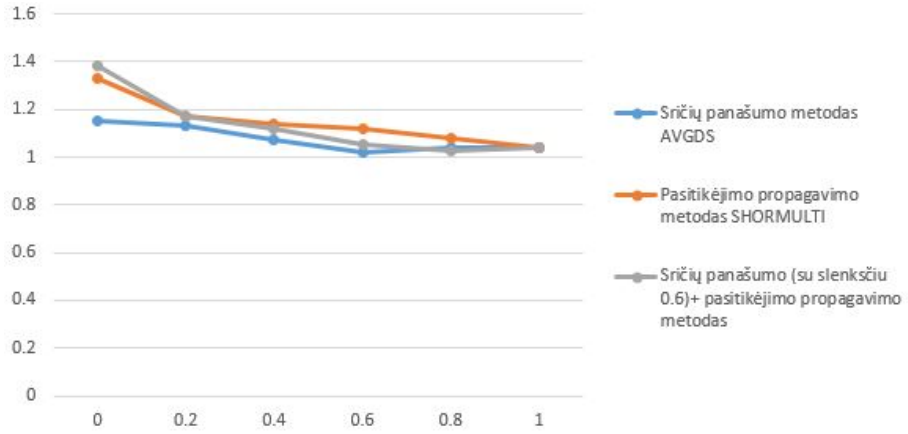
21 lentelė. $SHORTMULTI$ + $AVGDS$ (su sričių panašumais, gautais naudojant $GTDS$, su slenksčiu 0.3), rezultatai taikomi RS duomenims DS2

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.12	1.12	1.8	0.76	0.88
Šalto starto naudotojai	1.07	1.12	1.72	0.57	0.61
Ryžtingi naudotojai	1.13	1.12	1.95	0.9	1

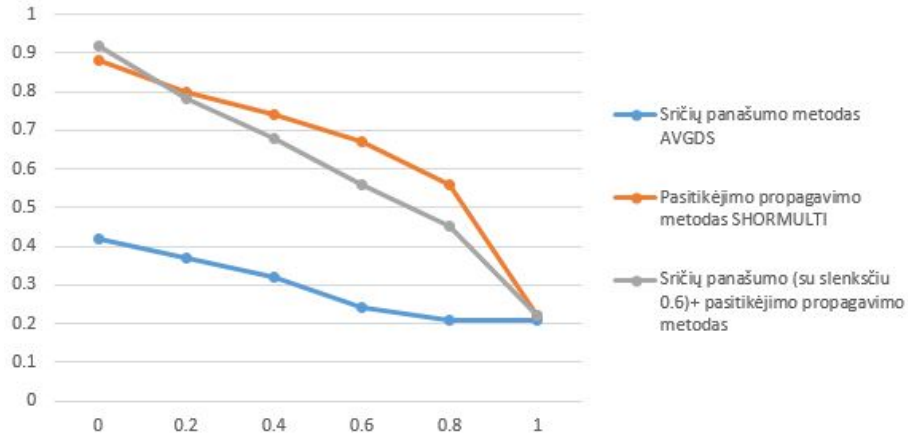
Pastebėta, kad tiek tikslumo, tiek padengimo prasme, efektyvesnė yra kombinacija, kai pirma taikomas sričių panašumo, o po to pasitikėjimo propagavimo metodas.

6.5.6. Rezultatų vertinimas

Iš visų atliktų bandymų geriausi rezultatai padengimo ir prasčiausi tikslumo prasme pasiekti taikant tik propagavimo metodą $SHORTARI$. Iš tiesų, taikant propagavimą ir parinkus mažą slenkstį, nuo kurio saugomi rasti pasitikėjimai, jei tik naudotojų grafas jungus, galima pasiekti 100 proc. padengimą, tačiau jis neturės prasmės, jeigu prognozių tikslumas bus mažas.



9 pav. Skirtingų metodų tikslumo priklausomybė nuo jiems taikomų slenksčių



10 pav. Skirtingų metodų padengimo priklausomybė nuo jiems taikomų slenksčių

Sričių panašumo metodas padidina padengimą nelabai paveikdamas tikslumo. Deja, tokiu būdu taikomas, jis tik padidina reitingų padengimą naudotojams, kuriems ir taip galime atlikti prognozę. Kitaip sakant, didėja reitingų padengimas, o naudotojų padengimas didėja nestipriai.

Eksperimentas buvo atliktas su dviem duomenų rinkiniais, kurie buvo sugeneruoti taip, kad juose egzistuotų šalto starto problema ir reitingai juose būtų pasiskirstę tolygiai. Didėjant naudotojų aktyvumui mažėja šalto starto naudotojų skaičius, didėja padengimas ir tikslumas. Sumažinus naudotojo kokybės parametro vidurkį, atitinkamai sumažėja ir vidutinis reitingas bei padidėja tikslumas. Daugėjant kategorijų RS, šalto starto problema darosi vis aktualesnė, nes netgi jei naudotojas įvertino daug elementų, to gali nepakakti tikslios rekomendacijos sudarymui, jeigu tie elementai priklauso skirtingoms kategorijoms.

Atlikus eksperimentus su dviem duomenų rinkiniais (su panašiomis ir skirtingomis kategorijomis), nepastebėtas, kad sričių panašumo metodas veiktų efektyviau vienam iš jų. Tai leidžia daryti išvadą, kad metodo rezultatai nepriklauso nuo to, kiek panašios sritys RS.

Galutiniams rezultatams tikslumo prasme yra svarbi kokybės parametro įtaka. Du skirtingi pirmenybių prasme naudotojai gali elementą įvertinti vienodai - vienas dėl to, kad yra jautrus kokybei ir to elemento kokybė yra aukšta, kitas - dėl to, kad elemento charakteristikos atitinka jo pirmenybes. Jeigu kokybės parametro (tiek naudotojui, tiek elementui) standartinis nuokrypis būtų mažas - naudotojų tarpusavio pasitikėjimo įvertis būtų aukštas tik tuo atveju, jei jų pirmenybės sutaptų,

o tai savo ruožu darytų teigiamą įtaką prognozių tikslumui. Ir priešingai - jei nuokrypis didelis - duomenys taptų chaotiškesni ir tikslumas mažėtų.

Pav. 9 ir pav. 10 matomi tikslumo (MAE) ir padengimo (RC) rezultatai su skirtingais slenkščiais duomenų rinkiniui DS1 taikant sričių panašumo metodą, pasitikėjimo propagavimo metodą ir taikant šiuos metodus kartu. Šis palyginimas geriausiai iliustruoja metodų veikimą - negalime vienareikšmiškai pasakyti, kurie rezultatai geriausi, nes tai priklauso nuo to, ko siekiama. Pasitikėjimo propagavimo metodas padengimo problemą sprendžia efektyviau, nei sričių panašumo, tačiau grąžina mažesnį tikslumą. Taikomi kartu šie metodai vienas kitą neutralizuoja - lyginant vien propagavimo grąžinamais rezultatais, tikslumas didėja ir atitinkamai padengimas mažėja. Šie rezultatai parodo, kad nors nėra vieno teisingo problemos sprendimo, tačiau šiame darbe siūlomas sričių panašumo metodas gali būti taikomas šalto starto problemos sprendimui.

Rezultatai ir išvados

Didžioji dauguma tyrimų apie RS, BF ir pasitikėjimu pagrįstas RS buvo atlikta vienmatėje aplinkoje - daroma prielaida, kad RS dalykinė sritis yra vienalytė ir naudotojų tarpusavio panašumas arba pasitikėjimas yra vienalytis. Šiame darbe siūloma RS padalinti pagal pasitikėjimo sritis ir taip pakeisti pasitikėjimo įvertį iš skaliaro į vektorių. Tokia RS su skirtingomis kategorijomis ir yra šio tyrimo objektas.

Šalto starto problema (duomenų apie naudotoją arba elementą nepakankamumas norint rekomenduoti elementą naudotojui) yra aktuali visoms RS, tačiau RS su skirtingomis kategorijomis ji yra ypač opi. Taip yra dėl to, kad tradicine prasme kiekviena kategorija formuoja atskirą RS savyje, kurioje BF metodą galima taikyti tik remiantis elementais toje kategorijoje. Taigi RS su skirtingomis kategorijomis bet kurį naudotoją galima pavadinti šalto starto naudotoju kategorijoje, kurioje jo aktyvumas yra pakankamai žemas.

Pirma problema su kuria susidurta siekiant atlikti eksperimentus - realių tinkamų pasitikėjimu pagrįstų RS su kategorijomis duomenų nebuvimas. Yra žinomos rekomendacinės sistemos su naudotojų tarpusavio ryšiais, tačiau su tolydžiais pasitikėjimais skirtingose kategorijose - ne. Populiariausias viešai prieinamas Epinions.com duomenų rinkinys pateikia naudotojų tarpusavio pasitikėjimą binarinėje skalėje ir elementų priklausymą kategorijoms. Dėl šios priežasties buvo pasiūlytas metodas, kurį taikant galima generuoti norimus RS su skirtingomis kategorijomis duomenis. Net jei duomenų rinkinys, tenkinantis norimus reikalavimus, ir būtų prieinamas, duomenų generavimas neprarastų prasmės, nes tokiu būdų galima išbandyti algoritmus skirtingomis charakteristikomis pasižyminčiais duomenims (naudotojai elementus vertina panašiai, yra nenuoseklūs, aktyvūs, neaktyvūs, jautrūs kokybei; kategorijos panašios, skirtingos). Šiame tyrime eksperimentai buvo atlikti naudojant du duomenų rinkinius - vieną su skirtingomis kategorijomis, o kitą - su panašiomis. Taip pat buvo siekiama sugeneruoti duomenis tokius, kad reitingai būtų pasiskirstę tolygiai.

Darbe buvo pasiūlytas sričių panašumo metodas. Šis metodas susideda iš dviejų etapų. Pirmojo etapo metu įvertinamas panašumo tarp sričių (kategorijų) įvertis, o antro etapo tikslas - įvertinti, kiek vienas naudotojas pasitiki kitu tam tikroje kategorijoje, jei žinomas pasitikėjimas kitoje arba keliose kitose kategorijose. Taikant šį metodą atsiranda galimybė pasiūlyti rekomendaciją ne tik to elemento, kurį palankiai įvertino naudotojai, kuriais pasitikime tam tikroje srityje, bet ir tai ką jie gerai įvertino ir kitoje srityje. Tai yra ypač aktualu esant šaltam startui - nors sistema apie naudotoją žino nedaug, galbūt tik apie jo pomėgius vienoje srityje, ji gali įvertinti jam patinkančius elementus kitoje. Nors eksperimento metu nebuvo tirta, kiek tokios rekomendacijos gali būti naujoviškos ir įžvalgios (dėl sudėtingo vertinimo), yra pagrindo manyti, kad RS taikanti sričių panašumo metodą gali pasižymėti šiomis savybėmis. Ši hipotezė galėtų lemti tolimesnę tyrimų kryptį.

Galiausiai, remiantis pastebėjimu, kad sričių panašumo ir pasitikėjimo propagavimo metodai gali būti taikomi nepriklausomai vienas nuo kito, buvo pasiūlyta taikyti šių metodų kombinacijas - pirma taikyti sričių panašumo, po to pasitikėjimo propagavimo metodus, ir atvirkščiai. Eksperimentais vertinami metodų prognozės tikslumai ir padengimai. Parodyta, kad labiausiai padengimą didina ir tikslumą labiausiai mažina pasitikėjimo propagavimo metodas. Sričių panašumo metodas

savo ruožtu mažiau didina padengimą ir mažiau paveikia tikslumą. Taikant šiuos metodus kartu gaunamas rezultatas tiek tikslumo, tiek padengimo prasme patenka tarp anksčiau minėtų rezultatų.

Įdomu tai, kad taikomi atvirkščia tvarka, šie metodai grąžina prastesnius rezultatus tikslumo prasme. Tai galima paaiškinti tuo, kad sričių panašumo metodas randa pasitikėjimus tarp naudotojų visose kategorijose ir po to taikomas pasitikėjimo propagavimas remiasi daug daugiau duomenų. Tuo tarpu, jei propagavimo metodas taikomas pirmas, jis sukuria daug pasitikėjimų tarp naudotojų, kurie nieko nežino vienas apie kitą, ir remiantis tokia informacija sudėtinga atlikti tikslias prognozes (nes jie patys yra prognozuoti).

Visi eksperimentai buvo atlikti tiek visų naudotojų imčiai, tiek šalto starto naudotojų imčiai. Jeigu bendradarbiavimo filtravimas pradiniams duomenims nesugebėjo atlikti nė vienos prognozės, tai taikant sričių panašumą kartu su pasitikėjimo propagavimu, išlaikant panašų tikslumą, pavyko pasiekti naudotojų padengimą lygų 94 procentų ir reitingų padengimą lygų 45 procentų. Šie rezultatai rodo, kad siūlomi metodai gali efektyviai išspręsti šalto starto naudotojų problemą. Visgi reikia turėti omenyje, kad socialinio tinklo charakteristikos turi didžiulę įtaką parenkant optimaliausius algoritmus ir jų parametrus.

Viena galimų tolimesnių tyrimų kryptis - labiau pažangių sričių panašumo metodų vystymas bei jų kombinavimas su efektyvesniais pasitikėjimo propagavimo metodais. Atliekant tyrimą pastebėta, kad tobulėjant pasitikėjimo propagavimo metodams, kurie taikomi didžiulėse žinomose rekomendacinėse sistemose, kuriose naudotojų skaičius skaičiuojamas dešimtimis milijonų itin aktuali ir resursų bei greitaveikos problemos. Taigi kita galima tyrimų kryptis - technologinė, susijusi su tuo, kaip galima efektyviai pritaikyti žinomus metodus realiose sistemose.

Literatūra

[Ahn08] Hyung Jun Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information sciences*, 178(1):37–51, 2008.

[AMT05] Paolo Avesani, Paolo Massa ir Roberto Tiella. A trust-enhanced recommender system application: moleskiing. *Proceedings of the 2005 acm symposium on applied computing*. ACM, 2005, p.p. 1589–1593.

[BOH11] Robin Burke, Michael P O'Mahony ir Neil J Hurley. Robust collaborative recommendation. *Recommender systems handbook*, p.p. 805–835. Springer, 2011.

[DK11] Christian Desrosiers ir George Karypis. A comprehensive survey of neighborhood-based recommendation methods. *Recommender systems handbook*, p.p. 107–144. Springer, 2011.

[ERK11] Michael D Ekstrand, John T Riedl ir Joseph A Konstan. Collaborative filtering recommender systems. *Foundations and trends in human-computer interaction*, 4(2):81–173, 2011.

[GNOT92] David Goldberg, David Nichols, Brian M Oki ir Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the acm*, 35(12):61–70, 1992.

[Gol05] Jennifer Ann Golbeck. Computing and applying trust in web-based social networks, 2005.

[HKBR99] Jonathan L Herlocker, Joseph A Konstan, Al Borchers ir John Riedl. An algorithmic framework for performing collaborative filtering. *Proceedings of the 22nd annual international acm sigir conference on research and development in information retrieval*. ACM, 1999, p.p. 230–237.

[HKR02] Jon Herlocker, Joseph A Konstan ir John Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information retrieval*, 5(4):287–310, 2002.

[JMP06] Audun Jøsang, Stephen Marsh ir Simon Pope. Exploring different types of trust propagation. *Trust management*, p.p. 179–192. Springer, 2006.

[LDGS11] Pasquale Lops, Marco De Gemmis ir Giovanni Semeraro. Content-based recommender systems: state of the art and trends. *Recommender systems handbook*, p.p. 73–105. Springer, 2011.

[MA07] Paolo Massa ir Paolo Avesani. Trust-aware recommender systems. *Proceedings of the 2007 acm conference on recommender systems*. ACM, 2007, p.p. 17–24.

[OS05] John O'Donovan ir Barry Smyth. Trust in recommender systems. *Proceedings of the 10th international conference on intelligent user interfaces*. ACM, 2005, p.p. 167–174.

[SG11] Guy Shani ir Asela Gunawardana. Evaluating recommendation systems. *Recommender systems handbook*, p.p. 257–297. Springer, 2011.

[SJA12] Alan Said, Brijnesh J Jain ir Sahin Albayrak. Analyzing weighting schemes in collaborative filtering: cold start, post cold start and power users. *Proceedings of the 27th annual acm symposium on applied computing*. ACM, 2012, p.p. 2035–2040.

[SS01] Rashmi R Sinha ir Kirsten Swearingen. Comparing recommendations made by online systems and friends. *Delos workshop: personalisation and recommender systems in digital libraries*. Tom. 1, 2001.

[VDCC11] Patricia Victor, Martine De Cock ir Chris Cornelis. Trust and recommendations. *Recommender systems handbook*, p.p. 645–675. Springer, 2011.

[ZL05] Cai-Nicolas Ziegler ir Georg Lausen. Propagation models for trust and distrust in social networks. *Information systems frontiers*, 7(4-5):337–358, 2005.