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:	lekt. 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 apžvelgtos socialinio tinklo duomenų taikymo galimybes sprendžiant šią problemą. Taip pat pasiūlyti trys nauji, socialinių tinklų duomenų panaudojimu besiremiantys metodai, kuriuos taikant galima spręsti šalto starto problemą.

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

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 skaitmeninėje erdvėje 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. Šis bruožas yra ypač aktualus kitai šio santykio pusei - siūlytojui (pavyzdžiui, pardavėjui) dėl akivaizdžių priežasčių - jis tampa labiau matomas, žinomesnis, galų gale jis gali gauti materialinės naudos.

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 interneto įmonių (pavyzdžiui, Amazon.com, Netflix.com, Last.fm) pritaikė BF metodus tam, kad padidinti naudotojų pasitenkinimą jų siūlomu produktu. Taikant BF daroma prielaida, kad istoriškai panašūs naudotojai išliks tokie ir ateityje. Taigi, esminė problema, kurią reikia spręsti - naudotojų panašumo vertinimas. Filtravimo procesas remiasi jau turimais duomenimis, kurie dėl problemos prigimties yra labai reti - sistemoje gali būti tūkstančiai naudotojų ir dar daugiau elementų, tačiau kiekvienas naudotojas dažniausiai būna įvertinęs tik labai mažą visų elementų dalį, taigi panašumo įvertinimas tampa iššūkiu. Negana to, kai sistemoje atsiranda naujas naudotojas, pradžioje apie jį žinoma per mažai, kad būtų galima pateikti patikimas rekomendacijas. Ši problema dar kitaip vadinama šalto starto (angl. cold start problem). Ji yra ypač svarbi ir dėl to, kad, jeigu naujas naudotojas per pakankamai trumpą laiką neįsitikins sistemos nauda, labai tikėtina, kad jis niekada ja nebesinaudos.

Ieškant šios problemos sprendimo būdų buvo atlikta nemažai tyrimų apie hibridines RS. Šių hibridinių RS esmė - taikant BF panaudoti informaciją apie elementų turinį. Turiniu pagrįstas RS nagrinėja atskira šaka, apie kurią šiame darbe nebus kalbama. Nors hibridinės RS ir išsprendžia dalį problemos, tačiau turi vieną esminį trūkumą - hibridinė RS yra labai priklausoma nuo konteksto, kuriame ji naudojama, kitaip sakant, ji yra neuniversali. Be to, kai kurioms dalykinėms sritims yra labai sudėtinga apibūdinti naudotojo susidomėjimo elemento atributus, taigi neįmanoma sukurti

tokios RS.

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 traktuojamas kaip alternatyvus dydis panašumui. Šie du dydžiai skiriasi:

- pasitikėjimas nebūtinai yra išskaičiuojamas iš duomenų - jis gali būti išreikštas tiesiogiai.
- pasitikėjimas turi kryptį - tai yra naudotojas u_1 gali pasitikėti u_2 ne tiek pat, kiek u_2 u_1.

Pasitikėjimo tinklas - grafas, kurio viršūnės vaizduoja 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 naudotoju pagrįstu ir daiktu pagrįstu metodų apibrėžimai, pristatyta šalto starto problema ir aprašyti įvairių autorių pasiūlyti metodai šiai problemai spręsti. Tyrimai apie socialinių tinklų duomenų panaudojimą bus aptarti plačiau ir pristatyti jau atlikti darbai šia problemos sprendimo kryptimi. Taip pat gilinamasi į socialinių tinklų duomenų panaudojimo galimybes siekiant panaikinti (arba sušvelninti) šalto starto problemos efektą. Kitame skyriuje pristatytas būdas, kaip galima generuoti socialinių tinklų duomenis ir pasiūlyti trys nauji metodai naudojami RS su socialinių tinklų duomenimis - bendrų kaimynų metodas, atsižvelgiantis tik į ryšių egzistavimą tarp naudotojų, sričių panašumo metodas, kuris taikomas RS su kategorijomis, ir pasitikėjimo interpoliavimo metodas, kurio esmė - prognozuoti naudojo tarpusavio pasitikėjimą remiantis ”paslėptais” RS duomenimis (juos naudojame generuodami RS duomenis). Trečiame skyriuje pateikta pasiektų rezultatų santrauka ir išvados.

1 Litratūros apžvalga

1.1 Bendradarbiavimo filtravimas

1.1.1 Bendradarbiavimo filtravimo metodas

Visų pirma, suformuluokime RS sprendžiamą problemą formaliai taip, kaip tai padaryta [**2**]. Vartotojų aibė pažymėkime U ir elementų aibę I. Be to, pažymėkime R aibę sistemoje turimų reitingų ir S – aibę galimų reikšmių, kurias gali įgyti reitingas (pvz. $S = [1,5]$). Taip pat, tarkime, kad vienas reitingas r_{ui} gali būti priskirtas vienam elementui $i \in I$ vieno naudotojo $u \in U$. Vartotojų poaibį, kuris yra įvertinęs elementą i, pažymėkime U_i. Analogiškai, I_u pažymėkime aibę elementų, kuriuos yra įvertinęs naudotojas u. Daiktų, 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. Dvi dažniausiai sutinkamos problemos – geriausios ir geriausių N rekomendacijos problema. Vienas būdų spręsti šias problemas yra įvertinti funkciją $f : U \times I- > S$, kuri nuspėja reitingą $f(u,i)$. Ši funkcija tada yra naudojama naudotojo u_a rekomendacijai elemento i^*, kuriam įvertinamas reitingas turi didžiausią reikšmę $i^* = \arg\max_{j \in \frac{I}{I_u}} f(u_a,j)$. RS galima modeliuoti dviem būdais:

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

- Bendradarbiavimo-filtravimu pagrįsti metodai rekomenduoja elementus, kurie patiko naudotojams, turintiems panašias pirmenybes. BF metodai remiasi tik naudotojų suteiktais reitingais. Jie ieško panašumų tarp naudotojų pirmenybių ir tai lemia dvi geras savybes, kuriomis nepasižymi turiniu pagrįsti metodai

  - įžvalgumas - siūlomi ne tik akivaizdūs pasiūlymai, bet ir netikėti (t.y. tokie, kokių naudotojas kitomis aplinkybėmis turbūt nerastų)
  - pritaikymas skirtingose srityse, elementu pagrįstos rekomendacijos reikalauja specifinių srities parametrų duomenų (pvz., kiek tam tikras filmas yra komedija, kiek drama)

  Bendradarbiavimo filtravimo sąvoką pirmąsyk panaudojo Goldberg [**16**]. Šis metodas remiasi artimiausių kaimynų metodu ir naudoja duomenis tiesiogiai generuojant rekomendacijas. Toliau darbe bus nagrinėjami būtent šiai klasei priklausantys metodai.

Bendradarbiavimo filtravimu pagrįsta reitingo prognozės esmė ta, kad parenkami artimiausi naudotojo kaimynai. Vartotojų tarpusavio artumas nustatomas naudojant panašumo metrikas, kurios bus aprašytos vėliau skyriuje 1.1.3. Šią prognozę galima atlikti dvejopai:

- Taikant artimiausių kaimynų regresiją, reitingas įvertinamas skaičiuojant pasvertą artimiausių kaimynų vidurkį.

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

Pagrindinis turiniu pagrįsto prieš naudotoju pagrįstą reitingo prognozavimo trūkumas yra tas, kad tokiu būdų sugeneruotos rekomendacijos yra nors ir tikslios, tačiau nelabai vertingos, nes rekomenduojami elementai pernelyg panašūs į tuos, 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 (plačiau -1.1.3), arba, kaip vėliau bus parodyta, vieno naudotojo pasitikėjimą kitu, apie kurį rašoma 1.2.2.

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

Šioje formulėje naudojamas svertinis vidurkis yra dažniausiai praktikoje taikomas, paprastas ir tikslus būdas nustatyti prognozei, 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 kaimynų su maža koreliacija keliamas triukšmas. Dar kitas būdas - atsižvelgiant į dalykinę sritį parinkti konstantą. Geriausią kaimynų parinkimo strategiją galima išsiaiškinti tiesiog paeksperimentavus su konkrečiais duomenimis, nes įprastai RS viena nuo kitos labai skiriasi tiek dėl dalykinės srities subtilybių, tiek dėl RS dalyvaujančių naudotojų.

1.1.2 Šalto starto problema

Šalto starto problema susijusiu su nepakankamu duomenų kiekiu. Šią problemą galima išskirti į dvi dalis:

- naudotojo šaltas startas
- elemento šaltas startas

Toliau bus rašoma tik apie 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 teikti rekomendacijas remiantis naudotojo profiliu

- gyvenamąja vieta, amžiumi, lytimi ir panašiai. Dar kitas būdas - įvertinti trūkstamus duomenis - ir yra šio darbo esminis tyrimo objektas.

1.1.3 Naudotojų panašumo apskaičiavimas

Jau anksčiau buvo minėta, kad norint rasti prognozuojamą naudotojo u tam tikram 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 matricos. Toliau pristatomi metodai, kurie padeda įvertinti naudotojų panašumą. Pyrsono, Spearmano koreliacija ir kosinuso panašumas detaliau aprašyti [**2**].

1.1.3.1. Pyrsono koreliacija

Pyrsono 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 susiduria su sunkumais, kai reikia paskaičiuoti panašumą tarp naudotojų, kurie bendrai yra įvertinę mažai elementų. Galima išeitis - nustatyti slenkstį, nuo kurio koreliacija būtų mažinama. Taigi panašumą $s(u,v)$ tokiu atveju reiktų dauginti iš baudos funkcijos

$$min\{|I_u \cap I_v|, 1\} \tag{4}$$

1.1.3.2. Apribota Pyrsono 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 $r_x = 4$. Tada Shardanand ir Maes pasiūlyta apribota Pyrsono koreliacija randama taip

$$s(u,v) = \frac{\sum\limits_{i \in I_u \cap I_v} (r_{u,i} - r_z)(r_{v,i} - r_z)}{\sqrt{\sum\limits_{i \in I_u \cap I_v} (r_{u,i} - r_z)^2} \sqrt{\sum\limits_{i \in I_u \cap I_v} (r_{v,i} - r_z)^2}} \tag{5}$$

1.1.3.3. Spearmano rango koreliacija

Spearmano rango koreliacija panaši į Pyrsono koreliaciją, vienintelis skirtumas toks, kad skaičiuojant Spearmano koreliaciją, naudotojo reitingai yra surūšiuojami didėjimo tvarka, jiems priski-

riami 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.

1.1.3.4. Kosinuso panašumas

Šis metodas skiriasi nuo ankstesnių tuo, kad yra į problemą žiūrima ne iš statistinio, o iš tiesinės algebros požiūrio taško. Vartotojai atvaizduojami kaip $|I|$ dimensijų turintys vektoriai, o panašumas apskaičiuojamas, kaip kosinuso atstumas tarp dviejų reitingo vektorių. Jis randamas sudauginant šiuos vektorius ir padalinant iš $L2$ (Euklido) normų sandaugos:

$$s(u,v) = \frac{\boldsymbol{r}_u \cdot \boldsymbol{r}_v}{||\boldsymbol{r}_u||_2||\boldsymbol{r}_v||_2} \tag{6}$$

1.1.3.5. Euristinis PIP panašumo matas

Euristinis panašumo matas pasiūlytas [**7**] kreipia dėmesį į šalto starto problemą. Dažniausias šalto starto problemos sprendimo būdas - naudoti hibridines RS, kurios naujiems naudotojams rekomendacijas pateikia naudodamos turinio informaciją ir tik surinkus pakankamai duomenų apie naudotoją, įjungiamas BF režimas. Ši panašumo metrika atsižvelgia į šalto starto problemą panašumą apskaičiuodama remdamasi trimis faktoriais - panašumu, poveikiu, populiarumu.

$$\boldsymbol{s(u_i, u_j) = \sum_{k \in C, j} PIP(r_{i,k}, r_{j,k})} \tag{7}$$

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}

$$PIP(r_1,r_2) = Proximity(r_1,r_2) \times Impact(r_1,r_2) \times Popularity(r_1,r_2) \tag{8}$$

Detalesnis aprašymas, kaip randamos šios reikšmės yra [**7**].

1.1.3.6. Panašumas su svoriais

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

Tokiu būdu randama Pyrsono koreliacija atrodytų taip:

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

ir kosinuso panašumas:

$$s_w(u,v) = \frac{\sum\limits_{i \in I_u \cap I_v} w_i^s \cdot r_{u,i} \cdot r_{v,i}}{\sqrt{\sum\limits_{i \in I_u} w_i \cdot r_{u,i}^2} \sqrt{\sum\limits_{i \in I_v} w_i^s \cdot r_{v,i}^2}} \tag{10}$$

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

$$w_i^{s,inf} = \log \frac{|U|}{|U_i|} \tag{11}$$

$$w_i^{s,lin} = 1 - \frac{|U_i|}{|R|} \tag{12}$$

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

Šaltinyje [**13**] parodyta, kad šis metodas geriausiai veikia vartotojams ”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 Pyrsono koreliaciją be svorių.

1.2 Socialiniai tinklai ir pasitikėjimu pagrįstos rekomendacinės sistemos

1.2.1 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 pasitikėjimo turi vienas naudotojas kitu, kai turimas pasitikėjimo tinklas (angl. web of trust). Įprastai toks įvertis randamas taikant propagavimo ir agregavimo operatorius. Propagavimo operatoriai nulemia, kaip bus elgiamasi su tranzityvumu. Kol kas nesigiliname į tai, kaip gaunami pasitikėjimo įverčiai, laikome juos duotais.

- Vienas dažniausiai naudojamų propagavimo operatorių (ypač, kai kalbame apie tikimybinį požiūrį) 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 A pasitikėjimą C.

Agregavimo operatoriai skirti susidoroti su situacijomis, kai yra keli propagavimo keliai. Šie operatoriai apjungia kelis pasitikėjimo įverčius į vieną. Žinoma, ne visi propagavimo keliai yra vienodo ilgio, tai yra, viename kelyje gali būti 1 naudotojas, kitame - 5. Verta pastebėti, 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 nesunkiai 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 du dalykai priešinguose vienos tolydžios skalės galuose, tai yra tik kai kurių tyrėjų daroma prielaida, kuri leidžia supaprastinti problemą. Kitas, įgaunantis vis daugiau paramos, požiūris teigia, kad nepasitikėjimas negali būti prilyginamas pasitikėjimo nebuvimui.

Josang [**18**] kalba apie subjektyvią logiką (angl. subjective logic), kurioje, nepasitikėjimas yra traktuojamas kaip atskiras nuo pasitikėjimo dydis. Šios teorijos branduolys - subjektyvios nuomonės (angl. subejctive opinions), kurios 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 turi tikslius apibrėžimus ir formules, jomis galima manipulioti ir gauti analitiškai pagrindžiamus rezultatus, pavyzdžiui paaiškinti populiarumo bangas.

1.2.2 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 skyla į dvi klases.

- 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 tik tame pačiame kontekste - jeigu a pasitiki b kai kalbama apie automobilius, o b pasitiki c sodininkystės klausimais, nieko negalėsime pasakyti apie a pasitikėjimą c kompiuterijos žiniomis.

1.2.2.1. TidalTrust

Ši formulė yra esminė Golbeck rekomendacijos algoritme. Algoritmo autoriai šią formulę išvedė atlikdami eilę eksperimentų, kurių metu jie ignoruodami tiesioginį naudotojo a pasitikėjimą naudotoju c tyrinėjo kelius, jungiančius šiuos du naudotojus. Lygindami taikant išskaidymą (angl. propagation) gautus įverčius su tikromis pasitikėjimo reikšmėmis jie pastebėjo, kad:

- trumpesni išskaidymo keliai leidžia apskaičiuoti tikslesnius pasitikėjimo įverčius
- keliai su didesnėmis pasitikėjimo reikšmėmis taip pat leidžia apskaičiuoti didesnius pasitikėjimo įverčius

Remiantis pirmu pastebėjimu buvo sugalvota, kad reikia apriboti kelio ilgį tarp naudotojų. Nustačius fiksuotą kelio ilgį gali atsitikti taip, kad tik maža dalis naudotojų gali būti pasiekiama. 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 - beveik 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.

$$t_{a,u} = \frac{\sum\limits_{v \in WOT^+(a)} t_{a,v} t_{v,u}}{\sum\limits_{v \in WOT^+(a)} t_{a,v}} \tag{13}$$

(13) 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 kartu su TidalTrust nebūtinai vi-

sada yra pranašesnis už BF, tačiau duoda žymiai geresnius įverčius naudotojams, kurie nesutinka su vidutiniu elemento i reitingu.

1.2.2.2. MoleTrust

$$p_{a,i} = \bar{r_a} + \frac{\sum\limits_{u \in R^T} t_{a,u}(r_{u,i} - \bar{r_u})}{\sum\limits_{u \in R^T} t_{a,u}} \tag{14}$$

(14) formulė - Massa [**11**] pasiūlyto rekomendacijų algoritmo pagrindas. Ši metrika susideda iš dviejų žingsnių:

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

Ciklų pašalinimo esmė ta, kad kiekvienas naudotojas tinkle būtų aplankytas tik kartą siekiant didesnio efektyvumo vykdant išskaidymą (angl. propagation). 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 (13). 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 kontroversiškus naudotojus, kuriuos dalis vertina kaip labai patikimus, o kita dalis - labai nepatikimus. Autoriai taip pat parodė, kad šis algoritmas išgauna tikslesnes prognozes naujiems naudotojams.

1.2.2.3. Pasitikėjimu pagrįstas svoris

Šis metodas pristatytas [**12**] naudoja vartotojo ir tiekėjo sąvokas. Reitingo prognozė skaičiuojama panašiai kaip (2):

$$c(i) = \bar{c} + \frac{\sum\limits_{p \in P(i)} (p(i) - \bar{p}) w(c,p,i)}{\sum\limits_{p \in P(i)} |w(c,p,i)|} \tag{15}$$

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

$$w(c,p,i) = \frac{2(sim(c,p))(trust(p,i))}{sim(c,p) + trust(p,i)} \tag{16}$$

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:

$$trust(p,i) = \frac{|\{(c_k, i_k) \in CorrectSet(p) : i_k = i\}|}{|\{(c_k,i_k) \in RecSet(p) : i_k = i\}|} \tag{17}$$

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

1.3 RS vertinimas

1.3.1 RS vertinimo metodai

Dažniausiai RS vertinimui naudojamas metodas vadinama vidutine absoliučia klaida (angl. Mean Absolute Error, trumpinama MAE) pagrįsta principu "išimk vieną" (angl. leave-one-out).

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

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 rea-

lybė - 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 būdų, kiekvienas naudotojas turi lygų svorį skaičiuojant vidutinę klaidą.

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

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (\hat{r}_{ui} - r_{ui})^2} \tag{19}$$

RMSE palyginus su MAE stipriau baudžia už dideles klaidas. Pavyzdžiui, duomenų aibėje su keturiais paslėptais reitingais RMSE geriau vertintų sistemą, kurios klaida lygi 2 trims reitingams ir 0 vienam reitingui, negu sistemą, kuri klysta 3 vienai reikšmei ir neklysta likusioms trims. MAE geriau vertintų antrą sistemą.

Kitas svarbus RS vertinimo matas yra - padengimas (angl. coverage). Herlocker RS vertinimo metodų apžvalgoje 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) apibūdina, kokią dalį visų RS esančių elementų RS gali rekomenduoti. Tai dar vadinama katalogo padengimu. 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 apskaičiuoti Gini indeksą:

    $$G = \frac{1}{n-1} \sum_{j=1}^{n} (2j - n - 1) p(i_j) \tag{20}$$

    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

    $$H = -\sum_{i=1}^{n} p(i) \log p(i) \tag{21}$$

- Naudotojų erdvės padengimas (angl. user space coverage) - terminas, nusakantis, kuriai da-

liai 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. Norėdami 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 trokštamas RS bruožas, ypač jeigu yra svarbios naujoviškumo ir įžvalgumo savybės.

1.3.2 RS vertinimo aspektai

[**3**] pristatyti kelios RS savybės - prognozės tikslumas, padengimas, pasitikėjimas, patikimumas, naujoviškumas, įžvalgumas, įvairumas, naudingumas, rizika, atsparumas atakoms, privatumas, pritaikomumas, praplečiamumas. Visų jų šioje apžvalgoje dėl gausos pristatyti neįmanoma, todėl toliau bus pristatytos tik įdomiausios šio darbo kontekste - buvo paminėtos anksčiau.

1.3.2.1. 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 - perskaityt aprašymą ar pan.

1.3.2.2. Pasitikėjimas

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

1.3.2.3. Naujoviškumas

Naujoviškos rekomendacijos naudotojams siūlo elementus, apie kuriuos jie nežinojo anksčiau. Paprasčiausias būdas padidinti rekomendacijų naujoviškumą - eliminuoti iš galimų elementų aibės jau vertintus ir peržiūrėtus elementus, tačiau šis metodas nepakankamas, 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. Metodo esmė tokia: nuo pasirinkto laiko taško reitingai yra paslepiami. Rekomenduojant sistema gautų taškų už kiekvieną iš tiesų įvertintą elementą ir baudžiama 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.

1.3.2.4. 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 reikia išlaikyti ir tikslumą.

Vienas būdų suprojektuoti sistemą, 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.

1.3.2.5. Tvirtumas

Tvirtumas (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.

2 Pasitikėjimu pagrįstos rekomendacinės sistemos modeliavimas ir siūlomas metodas

Šio darbo tyrimo objektas - naujos tinklinių programų kartos atstovė - socialinė RS. Ji generuoja prognozes (rekomendacijas) apie naudotojams galinčius patikti elementus iš tam tikros, paprastai labai didelės aibės, remdamosi tarpusavio naudotojų santykiu. Sihna ir Swearingen [**19**] palygino 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 (angl. aggregation) ir propagavimo (angl. propagation) 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.

Tranzityvumo prielaida yra teisinga tik tame pačiame kontekste (toliau - srityje, kategorijoje). Jeigu u_1 pasitiki u_2 kai kalbama apie automobilius, o u_2 pasitiki u_3 sodininkystės klausimais, nieko negalėsime pasakyti apie u_1 pasitikėjimą u_3 kompiuterijos žiniomis.

Šiame darbe siūlomi metodai remiasi nauju duomenų aplinkos interpretavimu. Iki šiol buvo kalbėta apie sistemas, kuriose naudotojai turi kitiems naudotojams priskyrę 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 vartotojas 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 $T_1, T_2..., T_N$.

Šiame tyrime daroma prielaida, kad pasitikėjimo įverčius naudotojai vieni kitiems priskiria rankiniu būdu, remdamiesi savo subjektyvia nuomone apie kitų naudotojų patikimumą. Nors realioje sistemoje tokia prielaida, ko gero, nepasiteisintų, ši problema galėtų būti sprendžiama tyrime 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 atsidaro kito naudotojo apžvalgą
- sistema pastebi, kad naudotojas u_1 skaito jau ne pirmą apžvalgą, kurią parašė u_2
- sistema primena anksčiau skaitytas apžvalgas ir paklausia, kiek jis pritaria naudotojui u_2
- jei naudotojas atsako, pasitikėjimo įvertis išsaugojamas

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.

2.1 RS vertinimas

2.1.1 Vertinimo metrikos

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.

$$MAE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} |\hat{r}_{ui} - r_{ui}|} \tag{22}$$

Kadangi mažai duomenų turintiems naudotojams tikslumas gali būti mažesnis, Massa ir Avesani 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), kurią pasiūlė Massa ir Avesani 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

$$RMSE = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (\hat{r}_{ui} - r_{ui})^2} \tag{23}$$

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.

2.1.2 Duomenų rinkinio skaidymas

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.

- Ryžtingi naudotojai - tie, kurie turi daugiau reitingų, tačiau jie yra pasiskirstę plačiai apie vidurkį. Tokiais laikysime naudotojus, kurių reitingų standartinis nuokrypis didesnis nei 2.

2.2 Sričių panašumo metodas

2.2.1 Rekomendacinės sistemos su pasitikėjimu kategorijose modeliavimas

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

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

Toliau bus aprašyti kiekvieno iš elementų generavimo algoritmai. Remiantis jais sudaromi 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.

2.2.1.1. Kategorijos

Kategorijų modeliavimas - pirmas algoritmo žingsnis. Juo siekiama apibrėžti ne tik kategorijas, kurioms gali priklausyti elementai bet ir kiekvieno elemento bruožus bei kiekvieno naudotojo pirmenybes. Duomenų rinkinio DS1 duomenys gali būti vertinami kaip filmų rekomendacinės sistemos duomenys. Apibrėžiame pavyzdžiui tokias kategorijas:

- X_1 - drama
- X_2 - komedija
- X_3 - siaubo
- X_4 - trileris
- X_5 - fantastika

Toliau apibrėžiame, kiek kiekviena iš šių kategorijų yra susijusi su kitomis. Euristiškai sudarome matricą 1:

1 lentelė. Kategorijų matrica SP1

Kategorijos	X_1	X_2	X_3	X_4	X_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 $X_1, .., X_5$ žymime kategorijas, o $x_1, .., x_5$ kategorijas atitinkančius požymius (toliau - charakteristikas). Taigi iš šios matricos galime teigti, kad pavyzdžiui:

- X_4 (trileris) yra gryniausias žanras, tai yra, turintis daugiausiai savo kategoriją atitinkančio požymio (kadangi turi didžiausią matricos įstrižainėje esančią reikšmę)
- X_3 (siaubo) - mažiausiai gryna kategorija (nes bruožų pasiskirstymas yra tolygiausias)
- X_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.

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.

2 lentelė. Kategorijų matrica SP2

Kategorijos	X_1	X_2	X_3	X_4	X_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

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.

2.2.1.2. Naudotojai

Naudotojas apibrėžiamas kaip vektorius $(y_1, y_2, y_3, y_4, y_5, q)$, kur $\sum_{i=1}^{5} y_i = 1$ ir $q \in [0,1]$. y_1, y_2, y_3, y_4, y_5 reiškia naudotojo pirmenybes - kiek svarbus jam yra tam tikras bruožas elemente. q - kokybės parametras rodo, kiek naudotojas yra jautrus elemento kokybei. Motyvacija kokybės parametro naudojimui tokia - 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 (naudojant tolygų skirstinį)
- 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.

2.2.1.3. Elementai

Elementas apibrėžiamas vektoriumi $(c, z_1, z_2, z_3, z_4, z_5, q)$. Čia c nurodo, kuriai kategorijai priklauso elementas, parametrai z_1, z_2, z_3, z_4, z_5 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, šiame tyrime standartinį nuokrypį prilyginame konstantai lygiai 0.3). Vidurkis parenkamas taip: pažiūrėję į c reikšmę atfiltruojame kategorijų matricoje kategoriją (stulpelį). Tada turime vidurkių, naudojamų generuojant z_1, z_2, z_3, z_4, z_5, vektorių. Kokybės parametras, kaip ir naudotojo atveju, 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.

2.2.1.4. Reitingai

Naudotojo reitingai elementams generuojami naudojant jo pirmenybes ir reiklumo kokybei parametrą bei atitinkamus produkto parametrus. Siekiama, kad jų pasiskirstymas būtų kuo artimesnis tikrovei, tai reiškia - nebūtų pasiskirstę galimų reikšmių kraštuose arba pernelyg vienodi. Sugeneruotų duomenų charakteristikos bus pateiktos kitame skyrelyje.

Kiekvienam naudotojui parenkamas atsitiktinis įvertintų elementų skaičius naudojant atsitiktinį dydį pasiskirsčiusį pagal normalųjį skirstinį su vidurkiu 30 ir standartiniu nuokrypiu lygiu 27. Parinktas didelis nuokrypis užtikrina, kad duomenys bus artimesni tikriems - Epinions.com duomenų rinkinyje vieno naudotojo įvertintų elementų skaičius svyruoja nuo 0 iki 655. Kiekvienam atsitiktinai parinktam elementui generuojamas reitingas taikant tokią formulę:

$$r_u(p) = 5 \times ((1 - q_u)\sqrt{pos(corr(X_u,Y_p))} + q_u q_p) \tag{24}$$

čia

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

Kraštutiniais atvejais, kai naudotojo reiklumas kokybei ir elemento kokybė lygi 1 arba naudotojo reiklumas kokybei lygus 0, tačiau elemento charakteristikos tobulai atitinka naudotojo pirmenybes, reitingas maksimalus (šiame tyrime lygus 5). Tyrimo eigoje pastebėta, kad koreliacijos funkcijos įtaka pernelyg maža, todėl ji padidinama naudojant pasirinktą iškilią funkciją (šiuo atveju šaknis suteikia pageidaujamą efektą).

2.2.1.5. 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 X_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:

$$t_u(v) = \max(x_1^u, x_1^v) - \min(x_1^u, x_1^v) = 0.2 - 0.1 = 0.1 \tag{25}$$

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 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.

2.2.1.6. Sugeneruotų duomenų rinkinių charakteristikos

Šalto starto sąvoka nėra vienareikšmiškai apibrėžiama - negalime iš anksto žinoti, kiek ir kokių reikia duomenų, kad situacija tenkintų apibrėžimą ir taikomas metodas veiktų kaip tikimąsi. Aplinkoje, apie kurią dabar rašoma, naudotojas gali būti šalto starto padėtyje, kai kalbame apie vieną sritį, tačiau kitoje srityje padėtis gali būti priešinga. Šio metodo tikslas - išnaudoti tokias situacijas. Beje, ši idėja yra pritaikoma ne tik šalto starto atveju, kai kalbama apie naudotoją, bet

ir naujos srities šalto starto atveju. Socialinius tinklus pagal pasitikėjimo sričių daugialypiškumą galima išskirti į du tipus:

- daugiaprofilinius - juose galimos įvairios pasitikėjimo sritys - tokios, kurias galima surikiuoti pagal panašumą ir tarp pirmos bei paskutinės nėra jokio panašumo.
- specializuotos - juose pasitikėjimo sritys yra gana artimos. Tokio tinklo pavyzdys galėtų būti kino mėgėjų socialinis tinklas, o sritys - įvairūs žanrai.

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

Elementų skaičius	300
Naudotojų skaičius	100
Naudotojo ryšių skaičiaus pasiskirstymas	N(10, 9) visose kategorijose
Naudotojo įvertintų elementų skaičiaus pasiskirstymas	N(30,27) visose kategorijose
Vertinamas bendrų elementų skaičius ieškant pasitikėjimo	12
Kategorijų matrica	SP1 1

Duomenų rinkinių charakteristikos yra tokios:

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

Reitingų pasiskirstymas:

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

2.2.2 Metodai

2.2.2.1. Pasitikėjimo propagavimo RS metodai

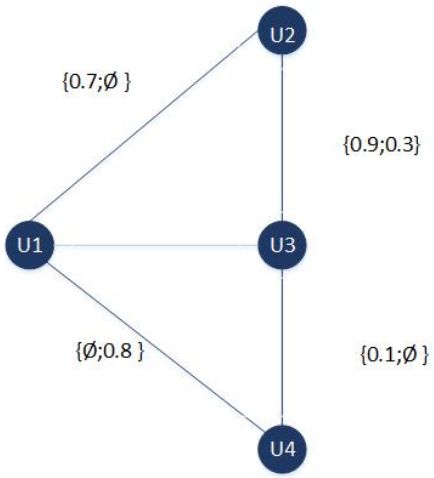
Kai kalbama apie rekomendacines sistemas su socialinių tinklų duomenimis daugiausia tyrimų [kokių] atlikta nagrinėjant agregavimo ir propagavimo metodus, kurie remiasi prielaida apie

pasitikėjimo tranzityvumą. Šiuose tyrimuose aiškinamasi, kokios yra pasitikėjimo grafo savybės ir kaip galima išskaičiuoti pasitikėjimus tarp naudotojų. Šiame tyrime minėtų metodų taikymas - tik dalis eksperimento. Dėl šios priežasties nuspręsta atlikti bandymus su paprasčiausiais metodais, priklausančiais trumpiausio kelio metodų šeimai. Esmė paprasta - randamas trumpiausias pasitikėjimo kelias, o tada taikomas propagavimo operatorius. Eksperimento metu išbandyti keli metodai:

- SHORTMULTI - daugybos operatorius
- SHORTARI - aritmetinis vidurkis

Golbeck disertacijoje parodyta, kad trumpesni keliai suteikia tikslesnę informaciją apie galimą naudotojų tarpusavio pasitikėjimą. Taip pat šioje disertacijoje kalbama apie pasitikėjimo mažėjimą (angl. trust decay) ir tai, kad metodai kreipiantys dėmesį į šį reiškinį grąžina tikslesnes prognozes. Čia tiramas SHORTMULTI metodas atsižvelgia į pasitikėjimo mažėjimo reiškinį, o SHORTARI - ne.

2.2.2.2. Sričių panašumo metodas



1 pav. Ryšių grafo fragmentas

Iš pradžių panagrinėkime paprastą pavyzdį. Tarkime, kad turime situaciją pavaizduotą grafe 1, kuriame pateikti naudotojų tarpusavio pasitikėjimai t_1, t_2, 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 $t^{T_1}_{u_1}(u_3)$

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

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). Tiriami metodai

- $GTDS$ (angl. trust-based domain similarity) esmė - turint naudotojų porų, kurios turi tarpusavio pasitikėjimą dviejose kategorijose (kurių panašumo ieškome) sąrašą, ieškoti Pyrsono 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:              trusts1.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 2. 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.

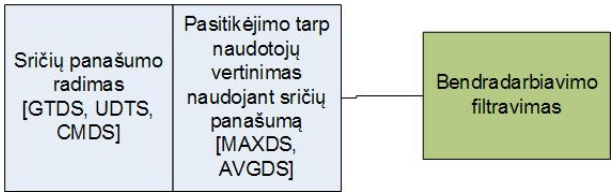
- $AVGDS$ metodu 3 siekiama panaudoti visą žinomą informaciją. Nežinomos pasitikėjimo reikšmės randamos ieškant randant žinomų pasitikėjimų sudaugintų su sričių panašumu vidurkį su svoriais. Svoriai šioje formulėje - tie patys sričių panašumai.

$$t_u^{T_i}(v) = \frac{\sum\limits_{j \in T} t_u^{T_j}(v) \times sim(T_i, T_j)^2}{\sum\limits_{j \in T} sim(T_i, T_j)} \tag{26}$$

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
```

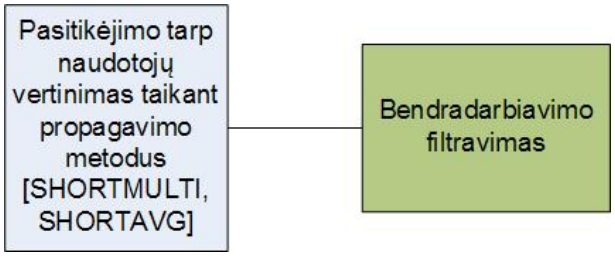
Visi aptarti metodai gali būti kombinuojami įvairiais būdais. Eksperimento metu išbandytos keturios architektūros, pavaizduotos pav. 2-5



2 pav. AR1: Sričių panašumo metodas

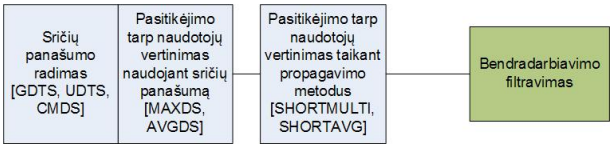
Algorithm 3 AVGDS algoritmas trūkstamų pasitikėjimų tarp dviejų naudotojų radimui

```
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
```



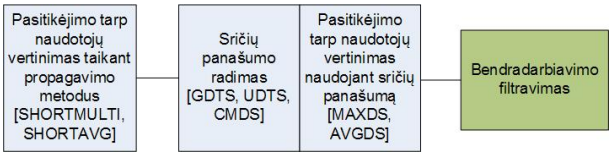
3 pav. AR2: Propagavimo metodas

Prop.jpg Prop.bb



4 pav. AR3: Sričių panašumo ir propagavimo metodas

DS.jpg DS.bb



5 pav. AR4: Propagavimo ir sričių panašumo metodas

2.2.3 Rezultatai

Šiuo eksperimentu siekiama ištirti sričių panašumo metodo efektyvumą. Tačiau be šio galutinio tikslo, taip pat galima paminėti ir tarpinius tikslus - išbandyti duomenų rinkinio generavimo algoritmą ir kitų žinomų metodų, naudojamų pasitikėjimų radimui, efektyvumą. Tolimesni skyriai

sugrupuoti pagal tai, kokia architektūra tiriama.

2.2.3.1. Bendradarbiavimo filtravimo rezultatai

Čia pateikiami rezultatai, gauti taikant paprastą bendradarbiavimo filtravimo algoritmą. Gauti tikslumo rezultatai labai panašūs į gaunamus taikant šį metodą realiems RS duomenų rinkiniams - MovieLens ir Epinions.com [papildyti - kokie tie rezultatai]. 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.

3 lentelė. BF rezultatai taikomi 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

4 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

2.2.3.2. Architektūra 1: Sričių panašumo metodas

Sugeneruotiems duomenims išbandyti $AVGDS$ 3 (sričių panašumo vidurkio) ir $MAXDS$ 2 (sričių panašumo maksimumo) metodai su slenksčiu lygiu 0.3. Panašumui tarp sričių nustatyti taikomas $GTDS$ metodas. Gauti tokie sričių panašumai pavaizduoti lentelėje 15

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

Kategorijos	X_1	X_2	X_3	X_4	X_5
X_1	1	0.73	0.84	0.24	0.61
X_2	0.73	1	0.78	0.21	0.4
X_3	0.84	0.78	1	0.36	0.56
X_4	0.24	0.22	0.36	1	0.44
X_5	0.61	0.4	0.56	0.44	1

6 lentelė. Sričių panašumo matrica taikant $CMDS$ metodą

Kategorijos	X_1	X_2	X_3	X_4	X_5
X_1	1	0.62	0.37	0.48	0.63
X_2	0.62	1	0.08	0.31	0.43
X_3	0.37	0.08	1	0.53	0.45
X_4	0.48	0.30	0.53	1	0.26
X_5	0.63	0.43	0.46	0.26	1

7 lentelė. $AVGDS$ 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

8 lentelė. $MAXDS$ 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

Iš rezultatų matosi, kad tiek tikslumas, tiek padengimas pagerėjo visų naudotojų imčiai. 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. 6 pav. ir 7 pav. matosi, kaip atrodo ryšių grafas prieš ir po sričių panašumo metodo pritaikymo.

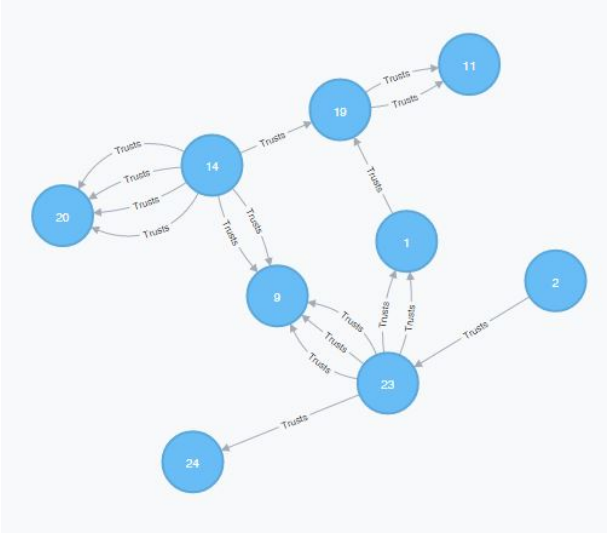
Pritaikius analogišką metodą duomenų rinkiniui DS2 gauta...

Taip pat $AVGDS$ metodas buvo išbandytas naudojant panašumus, gautus taikant $CMDS$ metodą. Šie panašumai pavaizduoti lentelėje 6.

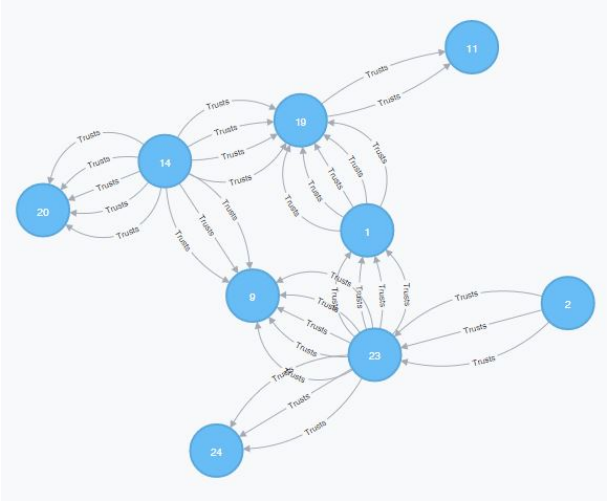
Taikant $AVGDS$ kartu su tokiais panašumais tikslumo prasme gauti prastesni rezultatai.

9 lentelė. $AVGDS$ rezultatai taikomi duomenų rinkiniui DS1 ir naudojant panašumus, išskaičiuotus iš kategorijų matricos

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



6 pav. Ryšių grafo fragmentas prieš pritaikant sričių panašumo metodą



7 pav. Ryšių grafo fragmentas pritaikius sričių panašumo metodą

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.

2.2.3.3. Architektūra 2: Propagavimo metodas

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 [kas kur taip vadina]. 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 tam, kad būtų išsaugoti, pasitikėjimai turi būti didesni.

10 lentelė. $SHORTMULTI$ rezultatai taikomi 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

11 lentelė. $SHORTARI$ rezultatai taikomi 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 [nuoroda]. $SHORTARI$ labiau padidina padengimą, tačiau tikslumas sumažėja pernelyg smarkiai, kad šie metodai būtų vertingi 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 iš propagavimų metodų naudosime tik jį.

2.2.3.4. Architektūra 3: Sričių panašumo ir propagavimo metodas

Šiame skyrelyje aprašytas eksperimentas, kai pirma taikomas sričių panašumo, o po to propagavimo metodas. Lyginant su ankstesniais eksperimentais gauti ženkliai geresni rezultatai abiems

duomenų rinkiniams tiek tikslumo, tiek padengimo prasme.

12 lentelė. $AVGDS$ + $SHORTMULTI$ (sričių panašumai gauti taikant $GTDS$), su slenksčiu 0.6, 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

13 lentelė. $AVGDS$ + $SHORTMULTI$ 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

Matome, kad $SHORTMULTI$ metodas taikomas po to, kai buvo pritaikytas sričių panašumo metodas duoda daug geresnį tikslumą ir ne prastesnį padengimą nei kitais atvejais. Dar labiau jį galima padidinti vėl pritaikius sričių panašumo metodą.

Pasitikėjimo duomenims kuriems pritaikytas $AVGDS$ + $SHORTMULTI$ metodas sričių panašumų matrica, randama taikant $GTDS$ gaunama jau stipriai iškreipta - panašumai tarp sričių tapo artimesni 0. Vis dėlto, tai netrukdo naujai gautiems pasitikėjimams dar kartą pritaikyti sričių panašumo metodo. Taikant $AVGDS$ + $SHORTMULTI$ + $AVGDS$ gaunamas kiek prastesnis tikslumas, tačiau reitingų padengimas dar labiau padidėjo (šalto starto naudotojams nepasikeitė)

14 lentelė. $AVGDS$ + $SHORTMULTI$ + $AVGDS$ rezultatai taikomi RS duomenims DS2

Duomenų rinkinio poaibis	MAE	$MAUE$	$RMSE$	RC	UC
Visi naudotojai	1.02	0.97	1.40	0.51	0.85
Šalto starto naudotojai	1	0.89	1.54	0.24	0.55
Ryžtingi naudotojai	0.98	0.87	1.68	0.51	0.67

Lyginant metodų veikimą RS su panašiomis ir skirtingomis kategorijomis tikėtasi, kad išbandyti metodai tikslumo prasme veiks geriau RS su panašiomis kategorijomis. Šis spėjimas pasitvirtino - kalbant apie geriausią metodų kombinaciją - $AVGDS$ + $SHORTMULTI$ - duomenų rinkiniui su panašiomis sritimis DS2 buvo fiksuojamas didesnis tikslumas. Tačiau verta paminėti, kad skirtumas nėra labai didelis (MAE rezultatas visiems naudotojams - 1.01 ir 0.97).

2.2.3.5. Architektūra 4: Propagavimo ir sričių panašumo metodas

Buvo atlikti du eksperimentai naudojant 6 panašumo matricą 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 papildomos informacijos galime gauti tik apie X_1 ir X_2 bei X_1 ir X_5 kategorijų panašumus (nes tik jų sandauga su žinomu pasitikėjimu, mažesniu už 1, gali viršyti 0.6).

Kategorijų panašumo matrica, gauta taikant $GTDS$ po to kai buvo įvertinti pasitikėjimai taikant $SHORTMULTI$:

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

Kategorijos	X_1	X_2	X_3	X_4	X_5
X_1	1	0.77	0.8	0.32	0.56
X_2	0.78	1	0.77	0.24	0.43
X_3	0.8	0.77	1	0.35	0.56
X_4	0.32	0.24	0.35	1	0.47
X_5	0.56	0.43	0.56	0.47	1

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.

16 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

17 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

Šis metodų kombinacija veikia prasčiau nei ankstesnė tikslumo, tačiau geriau padengimo prasme.

2.2.4 Rezultatų palyginimas

Iš visų atliktų bandymų geriausi rezultatai padengimo ir prasčiausi tikslumo prasme pasiekti taikant tik propagavimo metodą $SHORTARI$. Iš tiesų, taikant aritmetinį vidurkį pasitikėjimo propagavimui, jei tik naudotojų grafas jungus, galima pasiekti 100 proc. padengimą, tačiau jis neturės prasmės, jeigu tikslumas bus mažas.

Sričių panašumo metodas, taikomas prieš bendradarbiavimo filtravimą padidina padengimą nepablogindamas 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.

Geriausi rezultatai gauti taikant sričių panašumo metodą kartu su propagavimo metodu. Tai galima paaiškinti tuo, kad propagavimo metodas (šiuo atveju $SHORTMULTI$) gali įvertinti naudotojų pasitikėjimus kiekvienoje kategorijoje remdamasis daugiau duomenų.

Įdomu tai, kad šie du metodai taikomi atvirkščia tvarka prognozes atlieka su ženkliai mažesniu tikslumu. Tai galima paaiškinti tuo, kad propagavimo metodas taikomas pirmas gali sugeneruoti daug mažiau pasitikėjimo kelių. Tuo tarp sričių panašumo metodas taikomas antras, kai jau daug anksčiau nežinomų pasitikėjimų jau yra įvertinti ir jam nedaug lieka, ką įvertinti.

2.3 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.

3 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. Toks aplinkos transformavimas įgalina naudoti du darbe pasiūlytus metodus.

Sričių panašumo metodas leidžia įvertinti pasitikėjimą nežinomoje srityje, kai yra žinomas pasitikėjimas kitoje ir šių sričių tarpusavio panašumo įvertis. Taikant šį metodą atsiranda galimybė pasiūlyti rekomendaciją ne tik to, ką 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 - sistema apie naudotoją žino nedaug, nes metodo taikymas praplečia galimų rekomendacijų aibę.

Pasitikėjimo apskaičiavimas taikant tiesinę regresiją - kitas metodas leidžiantis įvertinti nežinomą pasitikėjimo įvertį vienoje srityje, kai yra žinomi pasitikėjimo įverčiai kitose. Šis metodas naudoja prielaidą, kad pasitikėjimas yra abipusis, tai yra, neturi krypties (ši prielaida kai kurioms dalykinėms sritims yra teisinga). Taikant tiesinę regresiją apskaičiuojamas naudotojo pasitikėjimas naudotoju, susiduriančiu su šalto starto problema ir tada jam priskiriamas pasitikėjimo įvertis.

Darbe pasiūlytas dar vienas metodas, kuris nenaudoja kelių pasitikėjimo sričių apibrėžimo. Bendrų kaimynų metodas taikomas, kai norime įvertinti vieno naudotojo pasitikėjimą kitu, tačiau jie neturi tiesioginio ryšio, o pasitikėjimo tinkle nėra jokių pasitikėjimo įverčių, tai yra, viskas, ką žinome apie konkretų naudotoją - jo ryšiai. Metodo esmė - panaudoti dviejų naudotojų bendrų ir savo ryšių skaičiaus santykį prognozuojant pasitikėjmą, o tada, remiantis prognozuojamu pasitikėjimu, įvertinti reitingų prognozę taikant bendradarbiavimo filtravimo metodą.

Atlikus tyrimą paaiškėjo, kad pasitikėjimo prognozės tiksliausios, kai yra didelės bendrų ir naudotojų ryšių skaičiaus santykio reikšmės. Galutiniai rezultatai pagal MAE ir $RMSE$ kriterijus labai panašūs į tuos, kuriuos gauname pritaikę bendradarbiavimo filtravimo algoritmą. $MAUE$ kriterijaus reikšmė kiek didesnė, o tai reiškia, kad metodas prasčiau veikia naudotojams, turintiems mažiau reitingų. Sudarant pasitikėjimų prognozę reikia sudaryti imtį iš naudotojų, turinčių pakankamai daug ryšių - tada tiek pasitikėjimo prognozė, tiek gautinės rekomendacijos būna tikslesnės. Bendrų kaimynų metodas turėtų būti naudojamas tais šalto starto atvejais, kai apie naudotoją, kuriam norime kažką rekomenduoti, yra žinomi tik jo ryšiai su kitais naudotojais. Taip pat prasminga nustatyti slenkstį, nurodantį naudotojo ryšių skaičių, nes kuo daugiau ryšių turi naudotojas, tuo prognozės tikslumas didesnis.

Pasiūlyti metodai sprendžią ne tik aptartą šalto starto problemą, bet ir kitą kertinę bėdą, su kurią

susiduria visos RS - duomenų retumo ir nepakankamumo. Jų taikymas leidžia panaudoti turimus duomenis situacijose, kai įprasti tradiciniai metodai negali veikti.

Literatūros sąrašas

[1] Pasquale Lops, Marco de Gemmis, Giovanni Semarero *Content-based Recommender Systems: State of the Art and Trends* Recommender Systems Handbook, 73-100, 2010.

[2] Christian Desrosiers, George Karypis *A Comprehensive Survey of Neighborhood-based Recommendation Methods* Recommender Systems Handbook, 101-140, 2010.

[3] Guy Shani, Asela Gunawardana *Evaluating recommender systems* Recommender Systems Handbook, 257-298, 2010.

[4] Robin Burke, Michael P. O'Mahony, Neil J. Hurley *Robust Collaborative Recommendation Systems: State of the Art and Trends* Recommender Systems Handbook, 805-836, 2010.

[5] Patricia Victor, Martine De Cock, Chris Cornelis *Trust and Recommendations Systems: State of the Art and Trends* Recommender Systems Handbook, 645-676, 2010.

[6] Paolo Massa, Paolo Avesani *Trust-aware recommender systems* Proceedings of the 2007 ACM conference on Recommender systems (2007) 17-24

[7] Hyung Jun Ahn *A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem* Information Sciences Vol 178 (2008) 37-51

[8] Jon Herlocker, Joseph A. Konstan, John Riedl *An Empirical Analysis of Design Choices in Neighborhood-Based Collaboraive Filtering Algorithms* Information Retrieval 5 178 (2002) 287-310

[9] Michael D. Ekstrand, John T. Riedl, Joseph A. Konstan *Collaborative filtering recommender systems* Foundation and trends in Human-Computer Interaction Vol. 4, No. 2 (2010) 81-173

[10] Jennifer Ann Golbeck *Computing and applying trust in web-based social networks* Dissertation

[11] Paolo Avesani, Paolo Massa, Roberto Tiella *A Trust-enhanced Recommender System application: Moleskiing* Proceedings of the 2005 ACM symposium on Applied computing (2005) 1589-1593

[12] John O'Donovan, Barry Smith *Trust in Recommender Systems* Proceedings of the 10th international conference on Intelligent user interfaces (2005) 167-174

[13] Alan Said, Brijnesh J. Jain, 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 (2012) 2035-2040

[14] Jonathan L. Herlocker, Josph A. Konstan, Al Borchers, 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 (1999) 230-237

[15] Sergio Mateo Maria *Collaborative Filtering in social Networks* (2010)

[16] David Goldberg, David Nichols, Brian M. Oki, Douglas Terry *Using collaborative filtering to weave an information tapestry* Communications of the ACM - Special issue on information filtering CACM Homepage archive Volume 35 Issue 12 (1992) 61-70

[17] Cai-Nikolas Ziegler, Georg Lausen *Propagation Models for Trust and Distrust in Social Networks* Information Systems Frontiers December 2005, Volume 7, Issue 4 Volume 35 Issue 12 (2005) 337-358

[18] Audin Josang, Stephen Marsh, Simon Pope *Exploring Different Types of Trust Propagation* Proceedings of the 4th international conference on Trust Management (2006) 179-192

[19] Sinha, Rashmi R., and Kirsten Swearingen. *Comparing Recommendations Made by Online Systems and Friends*" DELOS workshop: personalisation and recommender systems in digital libraries. Vol. 1. 2001.