VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
KOMPIUTERIJOS KATEDRA

Baigiamasis magistro darbas

Geografinio ir socialinio konteksto užklausų taikymai

Atliko:
Simonas Stonys parašas

Vadovas:
dr. Agnė Brilingaitė

Vilnius
2018

Turinys

Santrauka **3**

Summary **4**

Įvadas **5**

1. Susijusių darbų analizė **7**
1.1. Geografinio-socialinio konteksto užklausos 7
1.2. Rekomendacijų teikimas 8
1.3. Vietovės spėjimas 9

2. Geografinio-socialinio konteksto modelis **11**
2.1. Apibrėžimai 11
2.2. Markovo grandinės 14

3. Sistemos modelis **16**
3.1. Modelio architektūra 16
3.2. Atstumo funkcijos 16
3.3. Algoritmai 17
3.3.1. Geografinio-socialinio konteksto užklausos 17
3.3.2. Rekomendacijų teikimas 19

4. Modelio įgyvendinimas **22**
4.1. Naudoti duomenys 22
4.2. Užklausų vykdymas 25

5. Užklausų taikymai **27**
5.1. Vietovės spėjimas pagal parametrus 27
5.1.1. Pagal vietovę 28
5.1.2. Pagal draugus 30
5.1.3. Pagal vietovę ir draugus 32
5.1.4. Pagal vietovę ir laiką 34
5.1.5. Metodų palyginimas 36
5.2. Rekomendacijų teikimas 38

Išvados ir rekomendacijos **42**

Ateities tyrimų planas **43**

Literatūros šaltiniai **44**

Santrauka

Socialinių tinklų vartotojai, naudodami mobilius įrenginius atlieka įsiregistravimus. Geografinio-socialinio konteksto užklausų pagalba šie duomenys yra analizuojami ir gauti rezultatai gali būti panaudoti įvairiose srityse. Šiame darbe pristatomas socialinio tinklo vartotojo būsimo įsiregistravimo vietovės spėjimo modelis, kuris remiasi geografinio-socialinio konteksto užklausomis. Šios užklausos leidžia įvertinti būsimą vartotojo vietovę atsižvelgiant į istorinius duomenis bei vartotojų ryšius socialiniame tinkle. Siūlomos modelio variacijos ir naudojant realius socialinio tinklo duomenis vertinamas vietovės spėjimo tikslumas. Sukuriama vietovės rekomendacijų teikimo sistema paremta vietovės spėjimo modeliu. Tyrimo rezultatai rodo, kad pagal vartotojo įsiregistravimų charakteristikas galima vartotoją priskirti vienai iš grupių ir parinkti tinkamą spėjimo modelio variaciją.

Raktiniai žodžiai: Geografinės-socialinės užklausos, duomenų klasterizavimas, socialinis grafas, vietovės spėjimas, rekomendacijų teikimas, duomenų tyryba.

Summary

Application of Geo-social Queries

There is a huge amount of data generated on social networks on daily basis. Data from social networks and GPS data collected from users' mobile devices makes a Geo-Social Network. In this paper Geo-Social queries are implemented to analyze data from Geo-Social network. This data can be used for finding target audiences for advertisements or selecting rescue teams for disaster recovery.

A model for executing Geo-Social queries and algorithms for query processing system is presented in this paper. Functions for evaluating geographical and social distances are defined. For evaluation of social distance between users, structural equivalence measure is used in this paper. With the help of Geo-Social queries, location prediction model of next user check-in is created. Quality threshold clustering algorithm is used to create clusters of check-ins. Check-ins are clustered together to find familiar regions for each user in Geo-Social network. Finding familiar regions for each user allows to classify check-ins into performed near familiar areas and performed in new ares. In this paper user's movement among clusters is modeled with Markov Chains model.

An implementation of query processing system is presented by using real data from Gowalla social network. Location predictions are performed by di erent methods. First method uses user's location to predict location of next check-in. In another method location data of user's friends is used to make predictions. Third method combines user's location and his friends location data for prediction making. Finally, user's location and time of check-ins is used to make predictions. These methods are compared showing, that di erent method allows to achieve di erent prediction success rate. Finally, algorithms to make location recommendations for users in social networks are presented. Recommendations providing is tested on di erent types of users.

Results described in this paper show, that Geo-Social queries are a useful tool for looking for insights in Geo-Social data. Implementation of Markov chains model in Geo-Social querying gives good results for predicting location of user's next check-in. Also Analysis of data from Gowalla social network show that users are making enough check-ins for using this data in Geo-Social queries. Results of simulations of recommendations making model show that accuracy of recommendations varies when di erent model parameters values are used.

Įvadas

Socialiniai tinklai yra neatsiejama šiuolaikinio žmogaus gyvenimo dalis. Telefonuose, planšetiniuose kompiuteriuose bei kituose mobiliuosiuose įrenginiuose esanti programinė įranga socialinius tinklus padaro prieinamus bet kuriuo metu. Jų naudotojai, atliekantys įvairius veiksmus socialiniame tinkle, generuoja didelius duomenų srautus. Per pastarąjį dešimtmetį vykusi sparti socialinių tinklų plėtra ir nuolat didėjantis mobilių įrenginių skaičius leidžia kiekvieną sekundę surinkti daugybę informacijos apie jų naudotojus. Pavyzdžiui vien socialinis tinklas Facebook 2014 metais surinkdavo po 600 terabaitų vartotojų duomenų per dieną [20].

Vienas iš galimų socialiniame tinkle vartotojo atliekamų veiksmų – įsiregistravimas. Lankydamasis kurioje nors vietovėje, naudodamas mobilių įrenginį vartotojas gali atsidaryti savo socialinio tinklo paskyrą ir atlikti įsiregistravimą. Tuomet socialinio tinklo duomenų bazėje išsaugomi įsiregistravimo duomenys. Apjungus mobilių įrenginių fiksuojamus geografinius duomenis su informacija apie vartotojus socialiniuose tinkluose gaunami geografinio-socialinio konteksto duomenys.

Didėjant socialinių tinklų skaičiui ir jų populiarumui, atsiranda poreikis apdoroti geografinius-socialinius (GS) duomenis ir iš jų išgauti naudingos informacijos apie vartotojus. Tačiau didelis duomenų kiekis kelia iššūkius susijusius tiek su duomenų apdorojimu, tiek su jų saugojimu ir analizavimu. Šie iššūkiai tampa dar didesniais, kai apdorojimui ir analizei naudojami ir istoriniai duomenys. Skirtingu laiku socialiniame tinkle atlikti įvairūs vartotojo veiksmai apie jį suteikia daugiau informacijos ir padeda geriau suprasti jo įpročius.

GS tinklų duomenys tai grafai, kurių viršūnės yra socialinių tinklų vartotojai, retkarčiais įsiregistruojantys tam tikrose vietovėse, o briaunos – draugystės ryšiai tarp atitinkamų vartotojų. GS užklausos gali būti naudojamos teikti tikslinę reklamą mobiliųjų įrenginių naudotojams, masinėms gelbėjimo misijoms vykdyti, sparčiai vykdyti užklausas socialiniuose tinkluose. Daugumoje geografinio-socialinio konteksto duomenis nagrinėjančių darbų esantys algoritmai nevertina draugystės stiprumo tarp vartotojų. Pritaikius struktūrinio ekvivalentumo vertinimo metodus galima nustatyti ryšių tarp vartotojų lygį socialiniame tinkle ir taip patobulinti jau esamus geografinių-socialinių užklausų algoritmus.

Analizuojant vartotojų įsiregistravimus, svarbu atkreipti dėmesį ir į istorinius duomenis. Žinant vartotojo įsiregistravimų tendencijas pagal tai galima vertinti ir naujų įsiregistravimų charakteristikas. Vertinant ryšius tarp vartotojų, kartais neužtenka žinoti ar vartotojai susiję ar ne. Svarbus yra ir ryšio tarp jų stiprumas. Teikiant specializuotos reklamos paslaugas, pasiūlymai turėtų būti teikiami toms vartotojų grupėms, kuriose vartotojai tarpusavyje pakankamai glaudžiai susiję. Pavyzdžiui, du asmenys gali būti draugais socialiniame tinkle, tačiau nelabai pažįstami. Siekiant gauti kuo tikslesnius GS užklausų rezultatus, tokiems ryšiams priskiriama mažesnė svarba.

Taigi, GS užklausos naudingos srityse, kur reikalingos žmonių grupės tenkinančios šias sąlygas:

- Žmonės tarpusavyje nutolę per tam tikrą atstumą;
- Žmonės tarpusavyje glaudžiai susiję socialiniais ryšiais.

Viena iš GS užklausų panaudojimo sričių – vartotojo vietovės spėjimas. Žmogaus judėjimas dažnai yra rutiniškas, nes įprastai mums lankomos vietovės yra darbas, namai, mokymosi įstaigos. Vietovės spėjimą apsunkina nereguliariai atliekamos kelionės. Jos kartais gali būti įtakojamos draugų, todėl atsižvelgiant į vartotojų draugų judėjimo įpročius šiuos spėjimus galima atlikti tiksliau.

Įvertinus tai, kaip tikėtina, kad vartotojas sekantį įsiregistravimą atliks tam tikroje vietovėje, vartotojui gali būti pateikiamos rekomendacijos. Rekomendacijų turinys tai konkrečios vietovės

arba vietovių grupės. Sėkmingai pateikiant rekomendacijas, vartotoją gali pasiekti tikslinei auditorijai skirta reklama, o įmonės naudojančios šias rekomendavimo paslaugas galėtų sumažinti reklamai skirtas išlaidas.

Darbo tikslas – sukurti geografinio bei socialinio konteksto užklausų apdorojimui skirtos sistemos modelį. Darbo uždaviniai:

- Sukurti skirtingo pobūdžio GS užklausas;
- Sukurti algoritmus, reikalingus GS užklausoms vykdyti;
- Įgyvendinti GS užklausų vykdymo sistemą naudojant socialinio tinklo duomenis.

2 skyriuje apžvelgiami susiję darbai, aprašomas sistemos modelis, apibrėžiamos funkcijos reikalingos užklausoms atlikti. 3 skyriuje nusakomi algoritmai, naudoti geografinio-socialinio konteksto užklausoms vykdyti, atlikti vietovės spėjimams ir atlikti rekomendacijų teikimą. 4 skyriuje analizuojami naudoti socialinio tinklo duomenys, nagrinėjama modelio parametrų įtaka vietovės spėjimams bei įvertinamas rekomendacijų teikimo tikslumas.

Šis darbas yra mokslo tiriamojo darbo tęsinys. Iš mokslo tiriamojo darbo paimti, pataisyti ir praplėsti 3.3.1 skirsnis, 1.1, 2.1, 3.1, 3.2 poskyriai bei 4 skyrius.

1. Susijusių darbų analizė

Šiame skyriuje apžvelgiami susiję darbai. Darbai, nagrinėjantys geografinio-socialinio konteksto užklausas, jų taikymus, problemas kylančias iš užklausų įgyvendinimo, aprašomi 1.1 poskyryje. Darbai, aprašantys rekomendacijų teikimo sistemas, apžvelgiami 1.2 poskyryje. Susiję darbai, kuriose tiriamas socialinių tinklų vartotojų judėjimo modeliavimas, bandymai nuspėti įsiregistravimų vietoves, aprašomi 1.3 poskyryje.

1.1. Geografinio-socialinio konteksto užklausos

Armenatzoglou, Papadopoulos ir Papadias [2] pristato architektūrą reikalingą atlikti geografinio-socialinio konteksto užklausoms. Joje išskiriami geografiniai, socialiniai ir užklausų apdorojimo moduliai. Geografiniai ir socialiniai moduliai tarpusavyje nesąveikauja. Užklausų apdorojimo modulis socialinio pobūdžio duomenis gauna per primityvias funkcijas kontaktuodamas su socialiniu moduliu. Atitinkamai duomenis turinčius vartotojų įsiregistravimų informaciją, užklausų apdorojimo modulis pasiekia per geografinius primityvus kontaktuodamas su geografiniu moduliu. Taip pat straipsnyje pateikiamos primityvios užklausos, kurios naudojamos kaip pagrindas tiek paprastiems, tiek sudėtingesniems duomenų apdorojimo algoritmams. Straipsnyje analizuojami pagrindiniai algoritmai reikalingi užklausų įgyvendinimui. Autoriai parodo, kad efektyviai paskirstant skaičiavimo resursus ir panaudojant skirtingas duomenų saugojimo technologijas geografinio bei socialinio konteksto užklausas galima naudoti realiu laiku veikiančioje programoje. Tuo tarpu šiame magistriniame darbe naudojama straipsnyje aprašyta architektūra. Taip pat įgyvendinama keletas straipsnyje nurodytų primityvių funkcijų skirtų sąveikauti užklausų apdorojimo moduliui su geografiniu bei socialiniu moduliu.

Chow, Bao ir Mokbel [8] pristato sistemą, kuri pagal vartotojo buvimo vietą teikia tris su naujienų srautu socialiniame tinkle susijusias paslaugas. Naujienų srautu laikomos socialiniame tinkle skelbiamos tekstinės žinutės, vaizdo įrašai, komentarai. Kiekvienas naujienų srauto elementas turi mobilaus įrenginio jam priskirtą vietovę. Pirmoji paslauga vartotojui parodo tam tikru spinduliu paskelbtas draugų žinutes. Antroji paslauga paskelbtas žinutes reitinguoja pagal vietovę, laiką bei socialiniame tinkle paskelbtus vartotojo pomėgius. Trečioji vartotojui teikia rekomendacijas pagal socialiniame tinkle draugų paskelbtas žinutes bei draugų pomėgius. Taip pat straipsnyje aptariamos problemos kylančios iš sistemos praktinių taikymų, jos nepertraukiamo veikimo. Šiame magistriniame darbe tiriamas naujienų srauto atitikmuo – vartotojų atliekami įsiregistravimai, tam tikru spinduliu nuo vartotojo įsiregistravimo bandoma aptikti jo draugų įsiregistravimų grupių. Taip pat skirtingai nuo straipsnyje aprašytos rekomendacijų sistemos, šiame darbe magistriniame rekomendacijos sudaromos pagal socialiniame tinkle draugų atliktus įsiregistravimus.

Emrich ir kiti [9] aprašo geografinio ir socialinio konteksto užklausoms skirtą algoritmą, kur ryšiai tarp vartotojų nustatomi pasitelkus atsitiktinio klaidžiojimu su perkrovimu (angl. *Random Walk with Restart*) metodą. Šis metodas leidžia nustatyti atstumą tarp dviejų grafo viršūnių A ir B, atsižvelgiant į trumpiausią kelią iš A į B bei visus galimus kelius iš A į B. Autoriai apibrėžia geografinio-socialinio konteksto užklausą, kuri pasirinktai vartotojo ir vietovės porai grąžina Pareto optimalią vartotojų aibę, t.y. kiekvienam vartotojui iš grąžintos aibės negalima surasti nė vieno kito tokio vartotojo, kurio ir socialinis ir geografinis atstumas tarp pradžioje pasirinktos vartotojo ir vietovės poros būtų mažesnis, nei nuo atitinkamo vartotojo iš grąžintos aibės. Straipsnyje analizuojamas algoritmų veikimo laikas pasitelkus socialinio tinklo duomenis. Skirtingai nei Emrich ir kiti aprašytame straipsnyje, šiame magistriniame darbe socialiniai ryšiai tarp vartotojų vertinami

naudojant struktūrinio ekvivalentumo metodą, kuris veikia greičiau nei atsitiktinio klaidžiojimu su perkrovimu metodas. Taip pat šiame magistriniame darbe optimali vartotojų aibė parenkama ne tik pagal dabartines vartotojo draugų vietoves, tačiau ir pagal istorinius jų įsiregistravimų duomenis.

Shi ir kiti [25] pristato geografinio-socialinio tinklo duomenų grupavimo algoritmą, kuris atsižvelgia į vartotojų vietovės bei socialinių ryšių duomenis. Autoriai apibrėžia geo-socialinę atstumo funkciją, kurią naudoja vartotojų įsiregistravimų klasterizavimui. Taip pat straipsnyje aprašomas socialinio atstumo matas tarp dviejų vietovių. Šis matas įvertina socialinių ryšių stiprumą tarp vartotojų, kurie lankėsi tose vietovėse. Autoriai pademonstruoja modelio veikimą naudojant socialinio tinklo duomenis ir parodo, kad toks klasterizavimo būdas leidžia aptikti įvairias tendencijas gautose duomenų grupėse. Tuo tarpu šiame magistriniame darbe vartotojų įsiregistravimai klasterizuojami remiantis tik jų vietovės duomenimis. Taip pat šiame magistriniame darbe atskiriamos geografinio ir socialinio atstumo funkcijos. Prireikus, jas galima sujungti, tačiau turint atskiras funkcijas taupomi skaičiavimo resursai, kai reikia atsižvelgti tik į socialinį arba tik į geografinį atstumą. Panašiai kaip ir Shi ir kiti straipsnyje, šiame magistriniame darbe atsižvelgiama į socialinių ryšių stiprumą tarp vartotojų, kurie lankėsi analizuojamose vietovėse.

Heyer, Kruglyak ir Yooseph [11] aprašo minimalios kokybės ribos (angl. *Quality Threshold*) klasterizavimo algoritmą, kurį naudoja genų ekspresijos šablonų grupavimui. Algoritmo tikslas – surasti pakankamai didelius klasterius, atitinkančius nustatytus kokybės parametrus. Minimalios kokybės ribos klasterizavimo algoritmas sudaro klasterius kandidatus ir vėliau iš jų parenka geriausią. Taip sudarant klasterius nėra svarbu, kokia tvarka išdėstyti pradiniai duomenys. Skirtingai nei k-vidurkių klasterizavimas (angl. *k-means clustering*) ar saviorganizuojantis neuroninis tinklas, minimalios kokybės ribos algoritmas nebūtinai kiekvieną elementą priskiria vienam iš klasterių. Jei k-vidurkių klasterizavimo metodui k parenkamas per mažas, tuomet nepanašūs elementai gali patekti į vieną klasterį. Keičiant kokybės ribą pateiktame algoritme, klasterių dydis kinta, tačiau kiekvienas iš klasterių turės garantuotą reikiamą kokybę ir nepanašūs elementai nepateks į vieną klasterį. Dar vienas šio klasterizavimo būdo pranašumas – algoritmo pradžioje nereikia nustatyti, kiek klasterių bus sudaroma. Straipsnyje minimalios kokybės ribos klasterizavimo algoritmas pritaikomas realiems duomenims, analizuojami gauti rezultatai. Šiame magistriniame darbe naudojamas straipsnyje aprašomas algoritmas. Algoritmo savybės tinka vartotojų įsiregistravimų grupėms sudaryti, kai svarbus minimalus grupės dydis ir maksimalus atstumas tarp įsiregistravimų vietovių.

1.2. Rekomendacijų teikimas

Viena iš geografinio-socialinio konteksto užklausų sprendžiamų problemų yra lankytinų vietų rekomendacijų teikimas vartotojams. Rekomendavimo sistemos – tai programinės įrangos įrankiai ir technikos siūlančios tam tikrus objektus, naudingus vartotojui. Siūlomi objektai gali būti naujienos, viešbučiai, vietovės, prekės ir t.t. [23]. Rekomendacijos dažniausiai būna specializuotos kiekvienam vartotojui – skirtingi vartotojai ar vartotojų grupės gauna skirtingas rekomendacijas. Paprasčiausiu atveju rekomendacijos – tai išrikiuotas objektų sąrašas. Rekomendacijų sistemos sudarinėja tokius sąrašus remiantis vartotojų pateiktais reitingais arba vartotojų atliekamais veiksmais. Nuolatinis interneto plėtimasis ir prieinamų duomenų šaltinių skaičiaus augimas leidžia pateikti tikslesnes rekomendacijas ir tuo pačiu sukelia iššūkių apdorojant duomenis [23].

Dažnai susijusiuose darbuose pasitaikantis rekomendacijų parinkimo būdas – grupinis filtravimas (angl. *collaborative filtering*). Grupinio filtravimo metodas – tai bandymas nuspėti kaip vartotojui patiks tam tikras objektas, kurio jis dar nėra reitingavęs, remiantis kitų vartotojų pateiktais reitingais [10]. Pagrindiniai kriterijai pateikiant vietovės rekomendacijas: vartotojo pomėgiai,

socialinė bei geografinė įtaka [29][3]. Vartotojo pomėgiai ir geografinė įtaka turi didžiausią svarbą teikiant rekomendacijas [29], tačiau socialiniai faktoriai leidžia patikslinti rekomendacijas. Šiame magistriniame darbe rekomendacijos sudaromos remiantis vartotojo mėgstamomis lankyti vietovėmis. Taip pat vartotojo draugų lankytomis vietovėmis, kuriose minėtasis vartotojas nėra lankęsis, t.y. taikant grupinio filtravimo metodo idėją, tik čia atsižvelgiama ne į reitingus, o į lankytas vietoves.

Pateikiant rekomendacijas vartotojui, laikas yra vienas iš faktorių, į kurį reikia atsižvelgti. Skirtingos vietovės turi skirtingus įsiregistravimų pasiskirstymus laike. Vietovės susijusios su darbu dažniausiai lankomos ryte, tuo tarpu vietovės susijusios su pramogomis – vakare. Taip pat pirmoje dienos pusėje vartotojų judėjimas yra lengviau nuspėjamas. Taigi tikimybė įsiregistruoti tam tikroje vietovėje priklauso nuo laiko [12]. Kai kuriuose darbuose naudojami ir kiti faktoriai padedantys vartotojams teikti rekomendacijas, pavyzdžiui oras ar vartotojo nuotaika [21]. Šiame magistriniame darbe laikas tarp įsiregistravimų padeda įvertinti maksimalų atstumą, kurį vartotojas gali įveikti per tą laiko tarpą.

Pateikti vietovių siūlymus tiems vartotojams, kurie turi mažai įsiregistravimų yra viena iš problemų teikiant rekomendacijas. Vienas iš būdų apeiti šią problemą – rekomendacijas teikti tik tose vietovėse, kurias vartotojas pasirenka pats [3]. Tokiu būdu stipriai sumažėja analizuojamų vietovių aibė ir padidėja teikiamų rekomendacijų tikslumas tiek mažai įsiregistravimų, tiek daug įsiregistravimų atliekantiems vartotojams. Šiame darbe rekomendacijas galima pateikti ir tiems vartotojams, kurie patys yra atlikę nedaug įsiregistravimų, nes vienas iš būdų teikti rekomendacijas – remiantis draugų lankytomis vietovėmis. Kita problema yra rekomendacijų pateikimas realiu laiku. Kadangi dažniausiai galimų rekomenduojamų vietovių aibė yra didelė, reikalingi būdai, kaip pagreitinti sistemos atliekamus skaičiavimus. Rekomendacijas sudarant tik iš netoliese esančių vartotojų grupių ar tų vartotojų pomėgių sumažėja aibė vietovių, iš kurių reikia atrinkti rekomendacijas, todėl sistema gali veikti realiu laiku [28]. Taip pat siekiant pagreitinti skaičiavimus, prieš analizuojant vartotojų įvertinimus kiekvienai iš vietovių galima išskirti vartotojų ekspertų grupes. Tuomet teikiant rekomendacijas atsižvelgiama tik į ekspertų nuomones. Taip išvengiama nereikalingų skaičiavimų ir gaunami tikslesni įvertinimai [3]. Tuo tarpu šiame darbe dažnai nagrinėjama ne kiekvienas įsiregistravimas, o įsiregistravimo grupių centrai, todėl skaičiavimai atliekami greičiau nei tuo atveju, kai analizuojamas kiekvienas iš įsiregistravimų. Taip pat prieš analizuojant atliekamus įsiregistravimus istoriniai vartotojų duomenys yra apdorojami ir reikalinga informacija išsaugoma duomenų bazėje, todėl įsiregistravimų vykdymo metu sumažėja atliekamų skaičiavimų laikas.

1.3. Vietovės spėjimas

Bandymas nuspėti būsimą žmogaus vietovę yra sudėtingas uždavinys. Jį sėkmingai išsprendus, galima gauti naudingos informacijos apie žmogaus pomėgius, jo įpročius. Ši informacija gali būti panaudojama tiek reklamos teikimui, tiek transporto srautų valdymui ar renginių organizavimui. Sėkmingas vartotojo vietovės nuspėjimas gali padėti vykdant gelbėjimo misijas įvairių nelaimių atveju. Pavyzdžiui įvykus žemės drebėjimui, naudojant vartotojų vietovės spėjimo modelius, galima įvertint kiek žmonių galėjo būti paveikti žemės drebėjimo ir pagal tai paskirstyti gelbėjimo komandų resursus. Socialiniuose tinkluose vartotojai sugeneruoja daug duomenų, todėl galima gana tiksliai nuspėti būsimą vartotojo vietovę. Pagrindinis požymis leidžiantis atlikti spėjimus yra konkrečios GPS modulio telefone užfiksuotos koordinatės. Egzistuoja modelių, kurie apdoroja istorinius vartotojų duomenis socialiniuose tinkluose ir pateikia galimas vietoves, kur vartotojas galėtų lankytis ateityje [15][27][19]. Kai kuriuose straipsniuose parodoma, kad spėjimai būna tikslesni,

kai atsižvelgiama ir į vartotojų draugų duomenis socialiniuose tinkluose [27][17][19]. Dar vienas svarbus parametras bandant nuspėti būsim vartotojo buvimo vietą – laikas. Jis padeda įžvelgti tendencijų, kurių analizuojant kitus parametrus galima nepastebėti [15][27]. Kompleksinė vartotojų socialiniame tinkle generuojamų duomenų bei vietovių charakteristikų analizė leidžia tiksliau nuspėti vartotojų įsiregistravimus [19]. Šiame darbe vartotojo sekančio įsiregistravimo vietovė taip pat spėjama atsižvelgiant į istorinius įsiregistravimų vietovių duomenis, į draugų įsiregistravimų duomenis bei įsiregistravimų kitimą laike.

Viena iš problemų, su kuria susiduriama kuriant vartotojo vietovės spėjimo modelius yra didelė galimų vietovių aibė. Socialiniame tinkle gali būti milijonai skirtingų vietovių, kuriose buvo įsiregistruota [6]. Dažnai pasitelkiami klasterizavimo algoritmai, leidžiantys sugrupuoti vietoves ir taip sumažinti galimų įsiregistravimų aibę [15][27], arba bandoma nuspėti tam tikrą plotą o ne konkrečią vietovę [17]. Šiame darbe galimos įsiregistravimų vietovės yra apibrėžiamos kaip tam tikri plotai, arba vietovių rinkiniai. Pasitelkus klasterizavimo algoritmą įsiregistravimų vietovės sugrupuojamos, išskiriami klasterių centrai.

Vienas iš metodų modeliuoti įsiregistravimus – naudojant Markovo grandines. Lei ir kiti [15] Markovo grandinių pagalba analizuoja kiekvieno vartotojo įsiregistravimų trajektorijas, kurios sudaromos iš vartotojų dažnai lankomų regionų. Tuo tarpu Wang ir kiti [27] vietoves suskirsto į vienodo dydžio laukus ir skaičiuoja tikimybes pereiti iš vieno lauko į kitą. Perėjimo tikimybės tarp laukų kinta laike, nes labiau tikėtina, kad vartotojas apsilankys restorane vakare, o ne ryte. Ye, Zhu ir Cheng [6] bando nuspėti sekančios vartotojo veiklos tipą ir tuomet nuspėti įsiregistravimo vietovę. Čia modeliuoti vartotojų judėjimui naudojamas paslėptas Markovo modelis (angl. *hidden Markov Model*). Šiame darbe taip pat naudojamas Markovo grandinių modelis modeliuoti vartotojo judėjimą tarp regionų, tačiau čia regionų dydis nėra pastovus ir nėra vienodas skirtingiems vartotojams. Regionai sudaromi pagal pasirinktą regiono diametrą. Žinant įsireigistravimų vietovių charaketristikas, judėjimą tarp vietovių būtų galima nuspėti tiksliau, tačiau šiame darbe tyrimai atliekami naudojant duomenis, kur vietovę nusako tik duotas GPS koordinačių rinkinys.

2. Geografinio-socialinio konteksto modelis

Šiame skyriuje pristatomas geografinio-socialinio konteksto modelis. Modelyje naudojami apibrėžimai pristatomi 2.1 poskyryje. Vietovės spėjimams modeliuoti naudojamos Markovo grandinės aprašomos 2.2 poskyryje.

2.1. Apibrėžimai

Šiame darbe nurodytuose apibrėžimuose $v \in V$ žymėsime socialinio tinklo vartotoją, kur V yra visų vartotojų socialiniame tinkle aibė. Analizuojant vartotojų duomenis daroma prielaida, kad vienas vartotojas v atitinka vieną realų asmenį. Socialinio ryšio egzistavimą tarp vartotojų v_i, $v_j \in V$ žymėsime $e = (v_i, v_j)$, $v_i \neq v_j$, o visus socialinius ryšius nusako aibė $e \in E$. Visų įsiregistravimų aibę žymėsime C. Vietovę $l \in L$ nusako GPS koordinačių rinkinys, čia L yra visų galimų vietovių aibė. Laiką $t \in T$, kur T yra visų galimų laikų aibė, nusako kartu su įsiregistravimas fiksuojamas laiko momentas. Šiame darbe naudojamos [4] šaltinyje aprašytos grafų teorijos sąvokos ir žymėjimai.

1 apibrėžimas. Įsiregistravimas. Įsiregistravimu vadinsime rinkinį c, kur $\forall c \in C, c = (v, l, t)$, $v \in V$, $l \in L$, $t \in T$. Vartotojo $v \in V$ visų įsiregistravimų aibę žymėsime C_v.

Geografinį-socialinį (GS) tinklą sudaro socialiniame tinkle identifikuojami draugystės ryšiai tarp vartotojų ir jų mobilių įrenginių pateikiami vietovės duomenys.

2 apibrėžimas. GS tinklas. Grafas $G = (V, E)$, kurio viršūnes $v \in V$ sudaro vartotojai, o briaunas $e \in E$ – socialiniai ryšiai tarp jų, kartu su įsiregistravimų aibe C sudaro GS tinklą: $GS = (G, C)$.

Vienas pagrindinių GS užklausų elementų yra geografinis atstumas tarp dviejų įsiregistravimų. Kiekvieno įsiregistravimo metu mobilaus įrenginio GPS modulis užfiksuoja vartotojo koordinates. Žinant dviejų įsiregistravimų GPS koordinates, galima nustatyti geografinį atstumą tarp įsiregistravimų.

3 apibrėžimas. Geografinis atstumas. Funkciją $d_{geo} : L \times L \to \mathbb{R}$ vadinsime geografinio atstumo matu. Čia $d_{geo}(l_i, l_j) \geq 0$, $\forall l_i, l_j \in L$. $d_{geo}(l_i, l_j) = d_{geo}(l_j, l_i)$, $\forall l_i, l_j \in L$. $d_{geo}(l_j, l_i) = 0 \iff l_i = l_j$, $\forall l_i, l_j \in L$.

Norint GS pagalba nustatyti tinkamą vartotojų grupę, svarbu įvertinti socialinių ryšių stiprumą. Socialiniuose tinkluose esantys draugai ne visada yra draugai ir realiame pasaulyje. Pavyzdžiui du žmonės gali būti draugais „Facebook“ tinkle, tačiau vienas kito realiame pasaulyje gali būti net nematę, arba ryšį socialiniame tinkle palaiko tik su darbu susijusiais motyvais. Todėl yra svarbu ne tik žinoti ar du vartotojai socialiniame tinkle yra draugais, tačiau ir žinoti kiek stiprus yra jų ryšys. Ši informacija yra naudinga kai norime išrūšiuoti vartotojo draugus pagal ryšių stiprumą, arba GS užklausoms grąžintiems vartotojams nustatyti minimalią socialinio ryšio stiprumo ribą.

4 apibrėžimas. Socialinis atstumas. Funkciją $d_{soc} : V \times V \to \mathbb{R}$ vadinsime socialinio atstumo matu GS tinkle $GS = (G, C)$. Čia $d_{soc}(v_i, v_j) \geq 0$, $\forall v_i, v_j \in V$. $d_{soc}(v_i, v_j) = d_{soc}(v_j, v_i)$, $\forall v_i, v_j \in V$.

Šiame darbe funkcija $d_{soc}(v_i, v_j)$ įgyja reikšmes intervale $[0, 1]$. Atstumo reikšmė arti 0 nurodo, kad vartotojai socialiniame tinkle yra geri draugai, o atstumo reikšmė 1 parodo, kad vartotojai

socialiniame tinkle nėra geri draugai. Socialinis atstumas šiame darbe dar vadinamas socialinio ryšio stiprumu.

GS užklausos padeda išskirti daugiau naudingos informacijos, jei galima identifikuoti regionus, kurie vartotojui yra žinomi. Šiais regionais galėtų būti mikrorajonas kuriame vartotojas gyvena, tam tikru spinduliu aplink darbą esanti teritorija, ar mėgstamos lankyti vietos. Turint šią informaciją vartotojas galėtų iš restorano gauti specialų pietų pasiūlymą skirtą keletui žmonių, kai šalia jo darbo apsilanko vienas arba daugiau draugų. Darant prielaidą, kad visi kiti regionai nėra vartotojui žinomi, galima įsiregistravimus skirstyti į tokius, kurie atlikti žinomose vietovėse bei į tokius, kurie atlikti nežinomose vietovėse. Įsiregistravimų skirstymas yra reikalingas, kai GS užklausų pagalba ieškomi vartotojai, kurie atliko įsiregistravimą nepažįstamoje vietovėje. Tokiam vartotojui nustačius draugų grupę, kuri atitinkamą vietovę žino gerai, būtų galima pasiūlyti įsigyti netoliese esančių lankytinų vietų, kuriose yra lankęsi jo draugai, bilietą.

5 apibrėžimas. *ArPažįsta*(v, l) Funkcija *ArPažįsta*: $V \times L \to \{true, false\}$ grąžina $true$, jei $\exists$ toks įsiregistravimas $c_0 \in C_v$, kad $d_{geo}(c_0, l) \leq r$ ir kuriam $\exists$ tokia įsiregistravimų aibė $C_0 \subset C_v$, kad $\forall c_i \in C_0$ $d_{geo}(c_i, c_0) \leq r$.

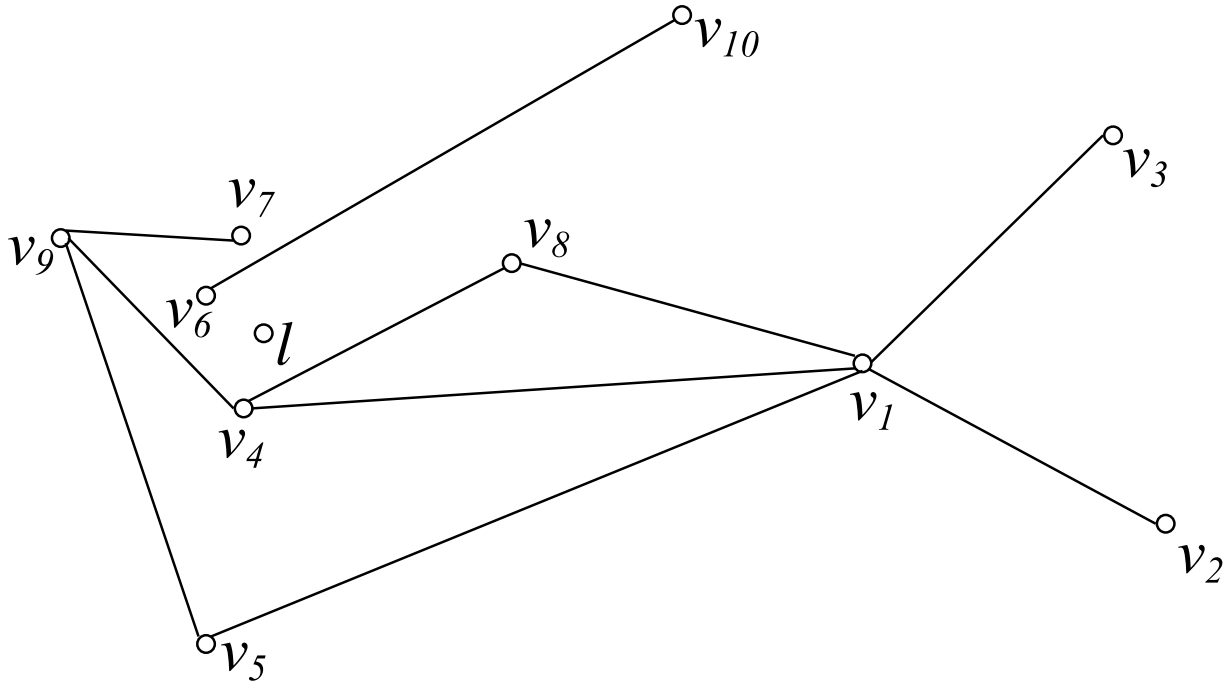
Vietovės $l_i \in c_i, \forall c_i \in C_0$ šiame darbe vadinamos pažįstamomis vietovėmis. Įsiregistravimų grupė C_0 minima 5 apibrėžime, šiame magistriniame darbe dar vadinama įsiregistravimų klasteriu.

6 apibrėžimas. Įsiregistravimų klasteris Įsiregistravimų klasteriu $kl = \{c_1, c_2, \ldots, c_k\} \in KL$ vadinsime tokią įsiregistravimų aibę, kuriai $\exists c \in kl$, toks, kad $d_{geo}(c, c_i) \leq r, \forall c_i \in kl$. Parametras k nusako reikalaujamą minimalų klasterio dydi, o parametras r – maksimalų klasterio spindulį.

Dažnai pasitaikantis GS užklausos tipas – iš anksto parinktoje vietovėje identifikuoti pažįstamų žmonių grupes. Priklausomai nuo to, kam bus naudojami užklausos duomenys, galima parinkti įvairias pradines sąlygas bei parametrus.

1 pavyzdys. Gelbėtojų komandos sudarymas. Įvykus žemės drebėjimui, reikia greitai surasti gelbėtojų komandą. Jie nelaimės vietą galės pasiekti greičiau, jei bus pakankamai arti jos, o dirbti galės efektyviau, jei turės patirties dirbant kartu. Čia socialinio ryšio atitikmuo gali būti požymis, nurodantis ar du gelbėtojai $v_i, v_j \in V$, $v_i \neq v_j$ yra dirbę kartu. Šiuo atveju pradinis parametras galėtų būti minimalus sudarytos komandos dydis. Gelbėjimo komandos parinkimo schema vaizduojama 1 pav. Tarkime l žymi vietovę, kur įvyko nelaimė, o mūsų ieškomą gelbėtojų grupę sudaro 3 asmenys. Atkarpos tarp gelbėtojų čia nurodo požymį, kad jie yra dirbę vienoje komandoje. Atstumai tarp taškų, paveikslėlyje žyminčių gelbėtojų vietoves, yra proporcingi atstumams tarp gelbėtojų. Nors arčiausiai l yra rinkinys $\{v_7, v_6, v_4\}$, šie asmenys nėra susiję socialiniais ryšiais. Įvykdžius GS užklausą, gautą gelbėtojų komandą sudarytų $\{v_7, v_9, v_4\}$. Taigi GS užklausos gali padėti optimizuoti gelbėjimo komandų veiklą.

Kartais gali nutikti taip, kad vartotojo v atžvilgiu vykdant GS užklausą, t.y. vietovėje l ieškant k draugų, GS užklausos rezultatui tinkamų vartotojų bus daugiau nei k. Vadinasi, tokiu atveju galimi keli GS užklausos rezultatai. Jeigu tinkamų vartotojų skaičius yra n, tuomet egzistuos $\frac{n!}{(n-k)!k!}$ galimų vartotojų kombinacijų. Dideliems n GS užklausos gali tapti neinformatyvioms ir prireiks naujų algoritmų apdoroti visų galimų rezultatų aibę. Todėl yra svarbu nustatyti, kuris rezultatas yra optimalus. Naudojant 4 apibrėžime nurodytą socialinio atstumo funkciją d_{soc} galima įvertinti kiekvieno iš vartotojų tinkamumą užklausos rezultatui. Tinkamų vartotojų aibę $\{v_1, \ldots, v_n\} \subset \Gamma_v$, išrikiuojame mažėjimo tvarka pagal $d_{soc}(v, v_i), i = 1, \ldots, n$ ir grąžiname pirmus k vartotojų.



1 pav. Vartotojų išsidėstymas erdvėje bei juos siejantys ryšiai.

Remiantis Armenatzoglou, Papadopoulos ir Papadias [2] straipsniu, apibrėžiamos primityvios funkcijos, reikalingos GS užklausų vykdymui. Siekiama sukurti lanksčią GS užklausų apdorojimo sistemą, todėl procedūros suskirstomos į socialines ir geografines. Visų pirma apibrėžiamos socialinės funkcijos.

7 apibrėžimas. *ArDraugai*(v_i, v_j). Funkcija *ArDraugai*: $V \times V \to \{true, false\}$ grąžina $true$, kai $(v_i, v_j) \in E$, grąžina $false$, kai $(v_i, v_j) \notin E$, $v_i, v_j \in V$, $v_i \neq v_j$.

Funkcija *GautiDraugus*(v) iš visos vartotojų aibės padeda atrinkti tik tuos vartotojus, kurie su vartotoju v siejasi draugystės ryšiais.

8 apibrėžimas. *GautiDraugus*(v). Funkcija *GautiDraugus*: $V \to V^n$ duotam $v \in V$ grąžina visus $v_i \in V$, kuriems *ArDraugai*$(v, v_i) = true$.

Vartotojo $v \in V$ draugų aibę žymėsime Γ_v. Toliau apibrėžiame geografines funkcijas.

9 apibrėžimas. *GautiVartotojoVietovę*(v). Funkcija *GautiVartotojoVietovę*: $V \to L$ grąžina paskutinę žinomą vartotojo v vietovę l_c, kur $c = (v, l_c, t_0)$, kai $t_0 = \max\{t | \forall c = (v, l, t) \in C_v\}$.

Paskutinė žinoma vartotojo vietovė padeda nustatyti, ar paskutinis įsiregistravimas vyko vartotojui v pažįstamoje vietovėje.

10 apibrėžimas. *ArtimiausiVartotojai*(l, r). Funkcija *ArtimiausiVartotojai*: $L \times \mathbb{R} \to V^n$ grąžina ne toliau kaip per r nuo vietovės l nutolusių vartotojų aibę $V^0 \subset V$, kur $d_{geo}(l, l_i) \leq r$, $\forall v_i \in V^0$, čia $l_i =$ *GautiVartotojoVietovę*(v_i).

Funkcija *ArtimiausiVartotojai* padeda identifikuoti vartotojus, kurių paskutinis įsiregistravimas vyko ne daugiau kaip r km nuo vietovės l. Funkcija naudojama kai svarbu ne tai, ar vartotojas lankėsi tam tikroje vietovėje, o tai, ar jo paskutinis įsiregistravimas buvo atliktas būtent toje vietovėje.

11 apibrėžimas. *kArtimiausiųVartotojų*(l, k). Funkcija *kArtimiausiųVartotojų*: $L \times \mathbb{N} \to V^k$ grąžina k arčiausiai vietovės l paskutinį įsiregistravimą atlikusių vartotojų sąrašą $(v_1, \ldots, v_k) \subset V$, kur $d_{geo}(l, l_1) \leq d_{geo}(l, l_2) \leq \ldots \leq d_{geo}(l, l_k)$ ir $d_{geo}(l, l_k) \leq d_{geo}(l, l_j), \forall v_j \in V \backslash \{v_1, \ldots, v_k\}$, čia $l_i =$ *GautiVartotojoVietovę*(v_i).

kArtimiausiųVartotojų leidžia tam tikros vietovės l atžvilgiu atrinkti k arčiausiai paskutinį įsiregistravimą atlikusių vartotojų. Svarbu atkreipti dėmesį į tai, kad ši funkcija nevertina laiko skirtumo nuo paskutinio įsiregistravimo įvykdymo iki užklausos atlikimo laiko.

Funkcija *GautiĮsiregistravimus* leidžia atrinkti vartotojo v įsiregistravimus įvykusius po tam tikro laiko momento.

12 apibrėžimas. *GautiĮsiregistravimus*(v, t_0). Funkcija *GautiĮsiregistravimus*: $V \times T \to C^n$ grąžina visus vartotojo v įsiregistravimus įvykusius po laiko momento t_0. T.y. $(c_i, c_{i+1}, \ldots, c_{i+n-1}) \subset C_v : t_c > t_0, \forall c \in (c_i, c_{i+1}, \ldots, c_{i+n-1})$.

2.2. Markovo grandinės

Vienas iš būdų modeliuoti žmogaus judėjimą – naudoti diskretaus laiko Markovo grandines. Šiame darbe nagrinėjant Markovo grandines, kalbama apie vartotojo įsiregistravimų vietovių modeliavimą, kai vartotojas gali būti vienoje iš jam pažįstamų vietovių grupių, sunumeruotų skaičiais $1, 2, \ldots$. Pradiniu laiko momentu 0 jis su tam tikra tikimybe gali būti k-ojoje pažįstamoje vietovėje. Laiko momentais $1, 2, \ldots$ jis su tam tikromis tikimybėmis, gali pereiti iš vienos vietovių grupės į kitas. Šiame darbe daroma prielaida, kad tikimybė laiko momentu n patekti į k-ąją pažįstamą vietovę, kai žinoma visa ankstesnė perėjimų iš vieno regiono į kitą evoliucija, priklauso tik nuo to, kokioje vietovėje vartotojas buvo $n-1$ laiko momentu. Papildoma informacija apie ankstesnę evoliuciją nekeičia tos tikimybės.

Remiantis Kubiliaus aprašyta Markovo grandinių teorija [14], nagrinėsime atsitiktinį procesą – vartotojo v judėjimą tarp jam pažįstamų vietovių $KL_v = \{kl_0, kl_1, \ldots\}$. Laikysime, kad $T = \{0, 1, \ldots\}$ yra aibė laiko momentų, kuriais atliekami vartotojo v įsiregistravimai $c \in C_v$, o būsenų erdvė B – baigtinė. Šiame darbe B – įsiregistravimo vietovių regionus nusakantys indeksai. Būsenas žymėsime natūraliaisiais skaičiais. Sakysime, kad procesas yra diskrečiojo laiko Markovo grandinė, jei bet kuriam natūraliajam skaičiui n ir bet kuriems $k, j_0, j_1, \ldots, j_{n-2}, j \in B$ teisingos lygybės

$$\begin{aligned}\mathbf{P}(X(n) = k | X(0) = j_0, X(1) = j_1, \ldots, X(n-2) = j_{n-2}, X(n-1) = j) = \\ = \mathbf{P}(X(n) = k | X(n-1) = j).\end{aligned} \tag{2.1}$$

Šaime darbe $\mathbf{P}(X(n) = k | X(n-1) = j)$ nusako tikimybę, kad sekanti įsiregistravimo vietovė $X(n)$ bus k, su sąlyga, kad praeito įsiregistravimo vietovė $X(n-1)$ buvo j. Remdamiesi sąlyginės tikimybės sąvoką, šias lygybes galime užrašyti taip:

$$\mathbf{P}(X(n) = k | X(0), \ldots, X(n-1)) = \mathbf{P}(X(n) = k | X(n-1)). \tag{2.2}$$

2.1 tikimybę vadinsime perėjimo iš j-ojo įsiregistravimo regiono į k-ąjį regioną tikimybe ir žymėsime $\mu_{jk}^{(n)}$. Matrica

$$\pi^{(n)} = \begin{pmatrix} \mu_{11}^{(n)} & \mu_{12}^{(n)} & \dots \\ \mu_{21}^{(n)} & \mu_{22}^{(n)} & \dots \\ \dots & \dots & \dots \end{pmatrix} \tag{2.3}$$

vadinama perėjimo matrica. Žinoma,

$$\sum_k \mu_{jk}^{(n)} = 1,$$

kai sumuojama pagal visus galimus regionus.

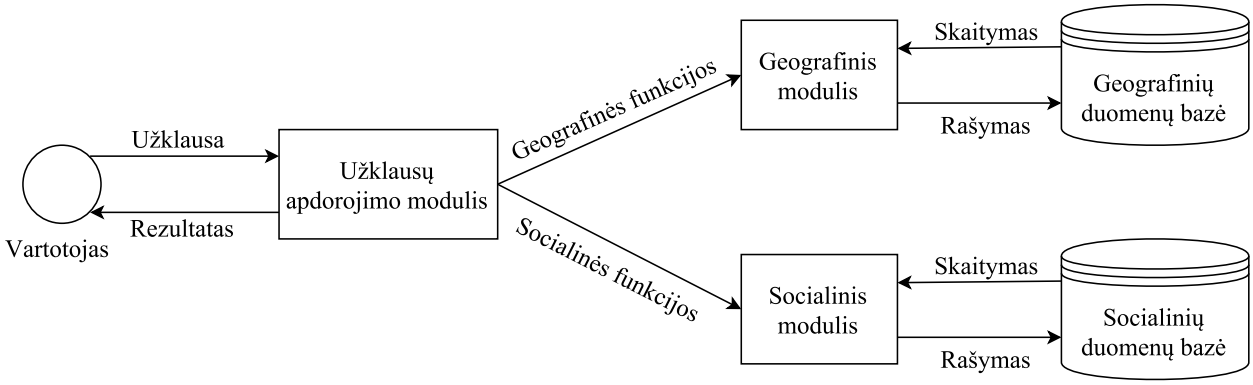
Homogenine vadinama tokia Markovo grandinė, kurioje tikimybės $\mu_{jk}^{(n)} = \mu_{jk}$ nepriklauso nuo n, t.y. perėjimo tikimybės nekinta laike. Jei būsenų skaičius yra baigtinis, tai grandinė vadinama baigtine. Šiame darbe bus naudojamos nehomogeninės, baigtinės diskretaus laiko Markovo grandinės.

Markovo grandinių modelis leidžia prognozuoti, kuriame klasteryje vyks sekantis vartotojo įsiregistravimas. Pagrindinis šio modelio trūkumas yra tai, kad įsiregistravimų klasteris gali apimti didelę teritoriją. Tuo tarpu kitos rekomendacijų teikimo sistemos vartotojams siūlo konkrečias vietas [28][30][29]. Tačiau net ir pateikiant didelę teritoriją kaip rekomenduojamą apsilankyti vietovę, toks pasiūlymas tiek vartotojui tiek reklamos užsakovui gali būti naudingas. Turint informaciją apie vartotojo ar jo draugų pomėgius, iš rekomenduojamos vietovės galima atrinkti konkrečias vietas, kurios vartotojams būtų siūlomos kaip reklama. Taip pat galima laisvai keisti parametrą, kuris nusako spėjamos vietovės spindulį. Norint tikslesnių rekomendacijų, šis parametras gali būti parenkamas pakankamai mažas, pavyzdžiui $0,5 - 3$km. Tuo atveju, kai norima sužinoti labiau tikėtiną, bet gana didelę teritoriją, kurioje vartotojas gali lankytis, šį parametrą galima parinkti didesnį, pavyzdžiui $3 - 10$km.

3. Sistemos modelis

3.1. Modelio architektūra

Geografinio-socialinio konteksto užklausas vykdanti sistema turi būti lanksti ir pritaikoma įvairaus tipo socialiniams tinklams. Tikėtina, kad realioje programoje vartotojų įsiregistravimų duomenys ir vartotojų draugų sąrašai bus laikomi atskirose duomenų bazėse. Armenatzoglou, Papadopoulos ir Papadias [2] siūlo atskirti geografines užklausas nuo socialinių užklausų. Tokiu būdu algoritmus įvykdyti galima greičiau, nei visas užklausas atliekant nuosekliai. Pagrindinis tokio užklausų apdorojimo per skirtingus modulius trūkumas – kiekvienam iš modulių reikia sukonfigūruoti atskirus serverius. Kita vertus, kiekvieną iš serverių galima pritaikyti atitinkamam moduliui. Pavyzdžiui socialines užklausas vykdyti grafo tipo duomenų bazėje, kuri leidžia greičiau atlikti užklausas nei reliacinėje duomenų bazėje [18]. Šiame darbe įgyvendinama Armenatzoglou, Papadopoulos ir Papadias [2] siūloma užklausų apdorojimo sistemos architektūra. Sistemos architektūra vaizduojama 2 pav. Vartotojui atliekant veiksmus socialiniame tinkle, pavyzdžiui atidarius socialinio tinklo programą telefone, jo atžvilgiu įvykdoma GS užklausa. Tarkime, kad šiuo atveju programa turi pateikti dvi vietovės rekomendacijas. Tuomet geografinių funkcijų pagalba, kreipdamasis į geografinį modulį, užklausų apdorojimo modulis išanalizuoja ankstesnius vartotojo įsiregistravimus ir suformuoja pradines rekomendacijas. Toliau per socialines funkcijas kreipiasi į socialinį modulį, kuris savo ruožtu kreipiasi į socialinių duomenų bazę ir užklausų apdorojimo moduliui pateikia informaciją apie vartotojo draugus socialiniame tinkle. Įvertindamas socialinio modulio pateiktą informaciją apie draugus, iš pradinių rekomendacijų užklausų apdorojimo modulis pateikia dvi rekomendacijas programai, kuri jas pateikia vartotojui.



2 pav. GS užklausų apdorojimo sistemos architektūra.

Algoritmai buvo įgyvendinti naudojant programavimo kalbą Python (*Python Software Foundation*, 3.6.1 versija). Modulių sąveika atitinka 2 pav. Gowalla socialinio tinklo duomenys buvo atskirti į dvi duomenų bazes. Socialinių duomenų bazėje laikomi visi vartotojų draugystės sąryšiai. Geografinių duomenų bazėje patalpinti visi vartotojų įsiregistravimai. Abi duomenų bazės įgyvendintos Microsoft SQL Server [24] reliacinėje duomenų bazių valdymo sistemoje, naudojant Python biblioteką pyodbc.

3.2. Atstumo funkcijos

Siekiant nustatyti socialinio ryšio stiprumą tarp vartotojų $d_{soc}(v_i, v_j)$, $v_i, v_j \in V$, šiame darbe naudojamas *struktūrinio ekvivalentumo* matas. Dvi grafo viršūnės laikomos struktūriškai ekviva-

lenčiomis, kai jos turi tuos pačius kaimynus, t.y. $v_i \cong v_j \iff \Gamma_{v_i} = \Gamma_{v_j}$, kur Γ_{v_i} žymi grafo viršūnės v_i kaimynų aibę. Kuo didesnis struktūrinis ekvivalentumas, tuo panašesnės yra grafo viršūnės. Panašiai vertinti ryšius galima ir socialiniame tinkle: kuo daugiau bendrų draugų turi du vartotojai, tuo labiau tikėtina, kad jie yra geri draugai ir realiame gyvenime. Tačiau pasitaiko tokių vartotojų, kurie turi vos keletą draugų. Dažniausiai šiems vartotojams struktūrinio ekvivalentumo dydis bus mažesnis nei tiems, kurie draugų turi daug. Todėl taikant šį matą svarbu atsižvelgti ne tik į tai, kiek grafe, vaizduojančiame socialinį tinklą, viršūnės turi bendrų kaimynų, tačiau ir į atitinkamų viršūnių laipsnį. Leich, Holme ir Newman [16] struktūrinį ekvivalentumą siūlo matuoti taip:

$$\sigma = \frac{|\Gamma_u \cap \Gamma_v|}{|\Gamma_u||\Gamma_v|}, \tag{3.1}$$

čia:
$\sigma \in [0; 1]$ – struktūrinio ekvivalentumo dydis,
u, v – grafo viršūnės.

Šis normalizuotas matas leidžia tiksliau įvertinti ekvivalentiškumą tarp viršūnių, turinčių skirtingą kiekį kaimynų.

Taigi, kuriant GS užklausas vykdančią sistemą socialiniam atstumui d_{soc} skaičiuoti buvo naudojamas modifikuotas struktūrinio ekvivalentumo matas:

$$d_{soc}(v_i, v_j) = 1 - \sigma = 1 - \frac{|\Gamma_{v_i} \cap \Gamma_{v_j}|}{|\Gamma_{v_i}||\Gamma_{v_j}|}, \tag{3.2}$$

čia:
$\Gamma_{v_i} = \textit{GautiDraugus}(v_i)$,
$v_i, v_j \in V$.

Atlikti pakeitimai užtikrina, kad kuo labiau susiję du vartotojai socialiniame tinkle, tuo mažesnis socialinis atstumas d_{soc} tarp jų. Kadangi $\sigma \in [0; 1]$, tai $d_{soc} \in [0; 1]$.

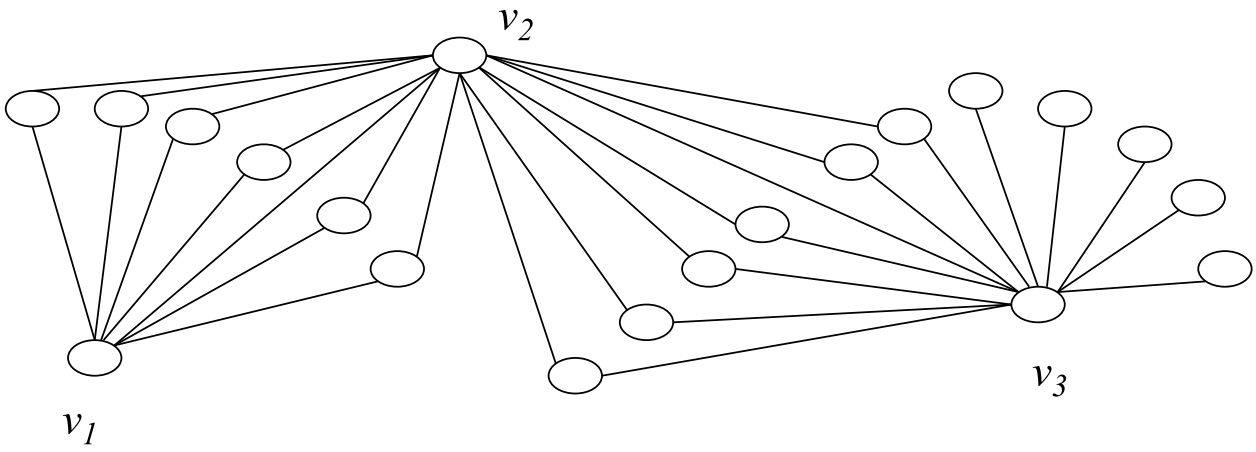
Grafas atitinkantis vartotojų ryšius socialiniame tinkle pateikiamas 3 pav. Lyginant vartotojų struktūrinį ekvivalentumą, (v_1, v_2) atrodo panašesni nei (v_2, v_3). Schemoje galima pastebėti, kad dauguma vartotojo v_1 draugų sutampa su v_2 draugais: $|\Gamma_{v_1} \cap \Gamma_{v_2}| = 6$, kai $|\Gamma_{v_1}| = 7, |\Gamma_{v_2}| = 14$. Tuo tarpu v_2 ir v_3 turi apie pusę bendrų draugų: $|\Gamma_{v_2} \cap \Gamma_{v_3}| = 6$, kai $|\Gamma_{v_2} = 14|, |\Gamma_{v_3}| = 12$. Tai, kad (v_1, v_2) yra panašesni nei (v_2, v_3) parodo suskaičiuoti socialiniai atstumai tarp vartotojų: $d_{soc}(v_1, v_2) \approx 0,9388$, $d_{soc}(v_2, v_3) \approx 0,9643$.

Skaičiuoti geografiniam atstumui tarp dviejų įsiregistravimų vietovių l_1 ir l_2 buvo naudotas Vincenty algoritmas [26]. Šis algoritmas pasirinktas dėl pakankamai mažos $0,5$ mm skaičiavimo paklaidos.

3.3. Algoritmai

3.3.1. Geografinio-socialinio konteksto užklausos

GS užklausos gali būti tikslesnės, jei vertinant vartotojo įsiregistravimus bus atsižvelgiama ir į jo istorinius įsiregistravimų duomenis. Pavyzdžiui vartotojui įsiregistravus ten, kur jis nėra



3 pav. Schema vaizduojanti vartotojų v_1, v_2, v_3 ryšius socialiniame tinkle.

linkęs lankytis dažnai, būtų naudinga surasti tokią draugų grupę, kuri atitinkamą vietovę žino gerai (dažnai ten įsiregistruoja). Ši draugų grupė jam galėtų aprodyti apylinkes, o tokią grupę žinoti pravartu ir reklamą teikiančioms programoms.

Nagrinėjant vartotojo $v \in V$ įsiregistravimų tendencijas, iš turimų duomenų sudaromi įsiregistravimų klasteriai $KL_v \subset C_v$. Klasteriams sudaryti naudotas Minimalios Kokybės Ribos (angl. Quality Threshold) klasterizavimo algoritmas [11]. Šio algoritmo tikslas – nustatyti klasterius su iš anksto parinktais kokybės parametrais r ir m, kur r žymi atstumą kilometrais, o m – nurodo minimalų įsiregistravimų kiekį klasteryje. Iš pradžių atsitiktinai parenkamas vartotojo v įsiregistravimas $c_0 \in C_v$ ir pridedamas prie klasterio kandidato kl_0^*. Toliau prie kl_0^* pridedamas kitas įsiregistravimas $c_i \in C_v$, $c_i \notin kl_0^*$, toks, kad $\exists c_j \in kl_0^* : d_{geo}(c_i, c_j) \leq r$. Taip įsiregistravimai pridedami tol, kol $\nexists c \in C_v$, $c \notin kl_0^*$ su kuriuo $\exists c_j \in kl_0^* : d_{geo}(c_i, c_j) \leq r$. Šis procesas pakartojamas kiekvienam įsiregistravimui $c \in C_v$ ir taip gaunama klasterių kandidatų aibė $\{kl_1^*, \ldots, kl_n^*\}$, $n = |C_v|$. Tuomet parenkame $kl^* \in \{kl_1^*, \ldots, kl_n^*\}$ turintį daugiausiai elementų. Jei $|kl^*| < m$, tuomet algoritmas sustabdomas. Kitu atveju įsiregistravimų klasteris $kl_1 = kl^*$ yra pirmasis tikras C_v klasteris. Procedūra kartojama, tačiau šį kartą kandidatai klasteriai formuojami iš $C_v \setminus kl_1$.

1 algoritmas. Ar žino vietovę

Įvestis: $v \in V$, $l \in L$, $r \in \mathbb{R}$
Išvestis: $x \in \{$*true, false*$\}$

```
for all kl ∈ KL_v do
  for all c ∈ kl do
    if d(l_c, l) ≤ r then
      return true
    end if
  end for
end for
return false
```

Pagal 5 apibrėžimą, sudarius įsiregistravimų klasterius buvo tariama, kad vartotojas v gerai žino vietovę l, jei ši vietovė nutolusi ne daugiau kaip r kilometrų nuo artimiausio vartotojo $v \in V$ įsiregistravimų klasterio (žr. algoritmą 1). Algoritmas naudingas ne tik nustatant, kurie vartotojai turėtų būti įtraukti į GS užklausos rezultatą, tačiau ir sprendžiant, ar reikia atlikti užklausą įvykus

naujam įsiregistravimui. Šis požymis suteikia naudingos informacijos apie vartotojo įpročius ir gali būti pritaikomas įvairiose GS užklausų modifikacijose. Pavyzdžiui [8] aprašytoje naujienų srauto generavimo sistemoje vartotojui galėtų būti rekomenduojama apsilankyti vietovėse, kurias žino jo draugai.

Analizuojant istorinius vartotojo įsiregistravimų duomenis svarbu atkreipti dėmesį į įsiregistravimų laiką. Tarkime vartotojas v_i apsilanko mieste, kurio apylinkės vartotojui nėra žinomos. Jam įvykdžius įsiregistravimą c_{v_i} vykdoma GS užklausa: tikrinama, ar galima surasti klasterį kl_{v_j}, kur $v_j \in \Gamma_v$ ir $\exists c \in kl_{v_j}$, toks, kad $d(c, c_{v_i}) \leq r$. Tačiau net ir suradus tokį klasterį reikia atkreipti dėmesį į tai, kaip seniai paskutinį kartą klasteryje buvo įsiregistruota. Galima situacija, kai klasteris yra sudarytas vien tik iš seniai vykusių įsiregistravimų (pavyzdžiui visi įsiregistravimai yra metų senumo). Todėl tikėtina, kad dabartinė įsiregistravimus atlikusio vartotojo vietovė pasikeitė ir jis neturėtų būti įtrauktas į GS užklausos rezultatą.

GS užklausų apdorojimo sistemai veikiant realiu laiku, turi būti analizuojamas kiekvienas naujas įsiregistravimas. Įvykus naujam vartotojo v įsiregistravimui c, atliekamos GS užklausos, įsiregistravimas c įtraukiamas prie istorinių duomenų ir perskaičiuojami v klasteriai. Pagal tai, kokiam tikslui bus naudojami GS užklausų rezultatai, galima išskirti tris ieškomų įsiregistravimų tipus:

1. Įsiregistravimai naujose, draugams pažįstamose vietovėse. Naujas vartotojo v įsiregistravimas c toks, kad $d_{geo}(c, c_v) > r$, $\forall c_v \in kl_v$ ir $\exists$ tokia k vartotojų aibė $V^0 \subset \Gamma_v$, $|V^0| = k$, kad $\forall v_i \in V^0$ $\exists c_{v_i} \in kl_{v_i}$ su kuriuo $d_{geo}(c, c_{v_i}) \leq r$. Aptikus tokius įsiregistravimus yra žinoma, kad vartotojas lankosi jam nepažįstamoje vietovėje. Kadangi pavyko aptikti k draugų, kurie toje vietovėje lankėsi bent m kartų, vadinasi galima atrasti k draugų socialiniame tinkle, kuriems minėtoji vietovė yra pažįstama. Čia m yra minimalaus klasterio dydžio parametras, naudojamas sudarant vartotojų įsiregistravimų klasterius. Tokia užklausa vaizduojama 4 pav. Vartotojui $id = 112282$ atlikus naują įsiregistravimą ir įvykdžius 1 tipo GS užklausą su parametru $k = 3$, buvo aptikti 3 vartotojai ($id = 21201, id = 112281, id = 21200$), kurių įsiregistravimų klasteriai yra ne toliau kaip 6 km nuo naujojo įsiregistravimo.

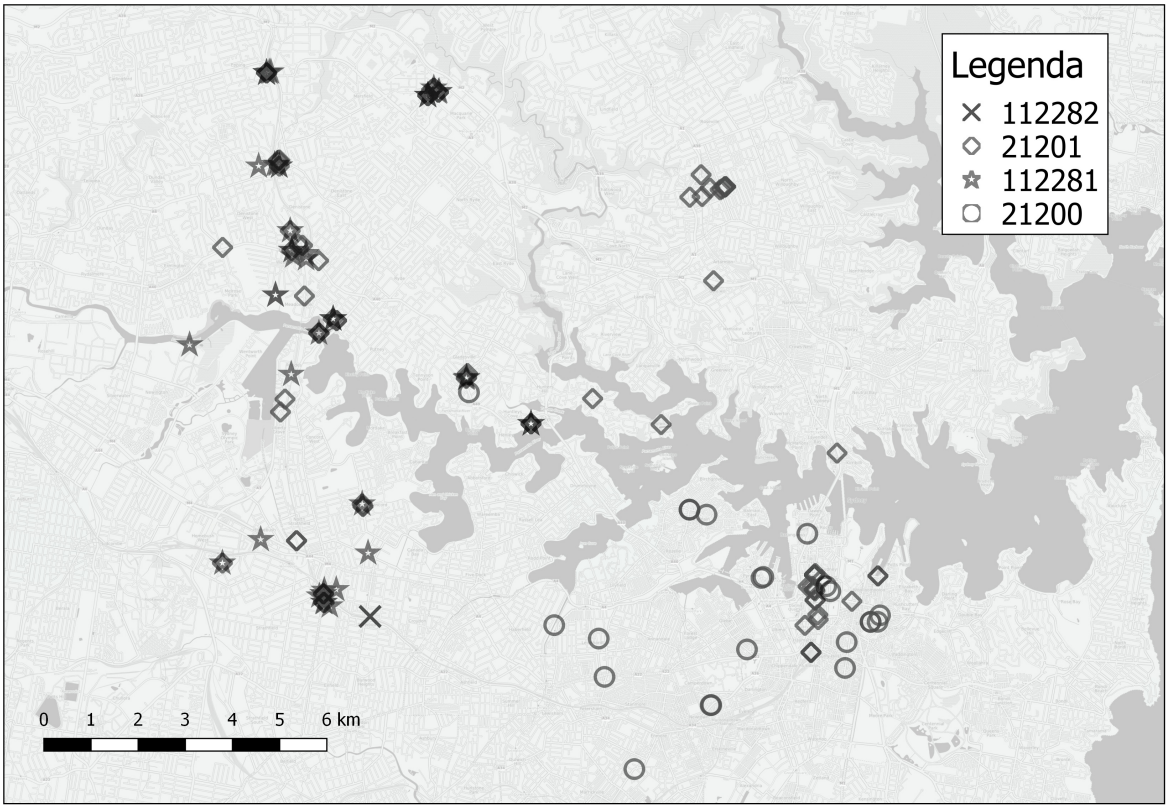
2. Įsiregistravimai naujose, draugams nepažįstamose vietovėse. Naujas v_i įsiregistravimas c toks, kad $d_{geo}(c, c_{v_i}) > r$, $\forall c_{v_i} \in kl_{v_i}$ ir $d_{geo}(c, c_{v_j}) > r$, $\forall c_{v_j} \in kl_{v_j}$, $\forall kl_{v_j} \in KLv_j$, $\forall v_j \in \Gamma_{v_i}$. Tokių įsiregistravimų ieškančios užklausos rezultatas – įsiregistravimai, kurie įvyksta toliau nuo vartotojui žinomų vietovių. Skirtingai nei pirmuoju atveju, čia nepavyksta aptikti nė vieno vartotojo v_i draugo socialiniame tinkle, kuris būtų atlikęs pakankamą kiekį įsiregistravimų atitinkamoje vietovėje.

3. Įsiregistravimai pažįstamose vietovėse. Naujas v įsiregistravimas c toks, kad $\exists c_v \in kl_v$, su kuriuo $d_{geo}(c, c_v) \leq r$. Šie įsiregistravimai parodo, kad vartotojas įsiregistravo jam pažįstamoje vietovėje. Tokių įsiregistravimų ieškančios užklausos yra mažiausiai imlios skaičiavimo resursams, nes čia nedalyvauja socialinis modulis.

3.3.2. Rekomendacijų teikimas

Žmogaus gali keliauti gana laisvai, tačiau jo judėjime galima įžvelgti tam tikras tendencijas atsirandančias dėl geografinių ir socialinių apribojimų [7]. Galima aptikti darbų, kuriuose žmogaus judėjimas modeliuojamas Markovo grandinėmis [5].

GS užklausų pagalba išanalizavus įsiregistravimus, vartotojams galima teikti rekomendacijas. Žinant vietoves, kuriose vartotojai yra linkę lankytis dažnai galima bandyti nuspėti, kurioje vieto-



4 pav. Vaizduojama GS užklausa, kurios pagalba identifikuojamas naujas vartotojo v įsiregistravimas jam nepažįstamoje vietovėje, kuri yra pažįstama vartotojo v draugams.

vėje įvyks sekantis įsiregistravimas. Šie spėjimai galėtų būti pateikiami kaip rekomendacijos ten apsilankyti.

Kiekvienam vartotojui $v \in V$, Minimalios Kokybės Ribos algoritmo pagalba, sudaroma n įsiregistravimų klasterių $kl_1, kl_2, \ldots, kl_n$, kur $kl_i \subset C_v, i = 1, \ldots, n$. Tuos įsiregistravimus, kurie nepriklauso nė vienai iš grupių, priskirsime grupei kl_{none}, šiame darbe šis klasteris dar gali būti žymimas kl_0. Taigi kiekvienas iš įsiregistravimų $c \in C_v$ priklauso vienam iš klasterių:$kl_{none} \cup kl_1 \cup \ldots \cup kl_n = C_v$. Kadangi žinomas kiekvieno iš įsiregistravimų laikas, galima sudaryti įsiregistravimų tarp klasterių seką. Toliau suskaičiuojame kiek kartų po įsiregistravimo klasteryje kl_i sekė įsiregistravimas klasteryje kl_j , šį dydi pažymėkime $|(c_i, c_j)|, c_i \in kl_i, c_j \in kl_j$. Šį skaičių padalijus iš visų įsiregistravimų klasteryje kl_i skaičiaus $|kl_i|$ gauname empirinę tikimybę įvykti įsiregistravimui klasteryje kl_j, su sąlyga, kad praeitas įsiregistravimas buvo klasteryje kl_i, t.y. $P(v_k \in kl_j | v_{k-1} \in kl_i)$. Taigi, kiekvienam $v \in V$ galima sudaryti perėjimo iš vieno klasterio į kitą tikimybių matricą

$$M = \begin{pmatrix} \mu_{00} & \mu_{01} & \ldots & \mu_{0n} \\ \mu_{10} & \mu_{11} & \ldots & \mu_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{n0} & \mu_{n1} & \ldots & \mu_{nn} \end{pmatrix}$$

Kadangi

$$\sum_j \mu_{ij} = \sum_j \frac{|(c_i, c_j)|}{|kl_i|} = \frac{1}{|kl_i|} \sum_j |(c_i, c_j)| = \frac{1}{|kl_i|} |kl_i| = 1$$

tai matrica M yra Markovo grandinės perėjimo tikimybių matrica.

Kiekvieną kartą įvykus naujam įsiregistravimui galima perskaičiuoti klasterius ir iš naujo sudaryti perėjimo tikimybių matricą M. Kadangi tikimybės nuolat atnaujinamos, tai leidžia modeliui tiksliau nuspėti sekantį klasterį, kuriame bus atliktas įsiregistravimas. Šis modelis įvertina ir tuos įsiregistravimus, kurie nebuvo atlikti nė viename iš nustatytų vartotojo v įsiregistravimų klasterių $c \notin KL_v$. Priskirti visus ne klasteriuose vykusius įsiregistravimus vienai grupei yra gana didelė abstrakcija. Ši informacija yra mažai reikalinga reklamą teikiančiai sistemai. Kita vertus modeliui nustačius, kad sekantis vartotojo įsiregistravimas turėtų vykti ne vienoje iš vartotojui pažįstamų vietovių, vartotojui galėtų būti siūloma apsilankyti jo draugų mėgstamose vietovėse.

4. Modelio įgyvendinimas

4.1. Naudoti duomenys

Algoritmų veikimui vertinti naudojami Gowalla socialinio tinklo duomenys. Šis socialinis tinklas sėkmingai veikė iki 2012 metų. Gowalla socialiniame tinkle programėlės pagalba vartotojai galėjo viešinti savo buvimo vietą įsiregistruojant. Taip pat vartotojams buvo siūloma lankytis populiariose vietose, keliauti programėlės siūlomais maršrutais. Socialiniai ryšiai tarp vartotojų šiame tinkle yra neorientuotas grafas. Naudojant Gowalla puslapio pateiktą viešą API, nuo 2009 vasario iki 2010 spalio buvo surinkta $6.442.890$ įsiregistravimų. Duomenys užfiksuoti įsiregistravimų metu vaizduojami 1 lentelėje. Kiekvienam vartotojui išsaugomas jo eilės numeris, tikslus atlikto įsiregistravimo laikas, įsiregistravimui atlikti naudotame įrenginyje esančio GPS modulio užfiksuota ilguma bei platuma ir įsiregistravimo vietos identifikacinis numeris. Kadangi jokie duomenys apie užfiksuotą vietos ID duomenų šaltinyje nėra pateikiami, šiame magistriniame darbe ši informacija nebuvo naudojama. Per nurodytą laikotarpį surinkta 196.591 vartotojų tarpusavio ryšių informacija išsaugota kaip grafo duomenys, nurodant kurias viršūnes jungia kiekviena grafo briauna. Iš viso surinkta 950.327 draugystės ryšių [1].

vartotojas	įsiregistravimo laikas	platuma	ilguma	vietos id
0	2010-10-19T23:55:27Z	30,235909	-97,795140	22847
0	2010-10-18T22:17:43Z	30,269103	-97,749395	420315
0	2010-10-17T23:42:03Z	30,255731	-97,763386	316637
0	2010-10-17T19:26:05Z	30,263418	-97,757597	16516
0	2010-10-16T18:50:42Z	30,274292	-97,740523	5535878
0	2010-10-12T23:58:03Z	30,261599	-97,758581	15372

1 lentelė. Gowalla duomenų rinkinio įsiregistravimo pavyzdys

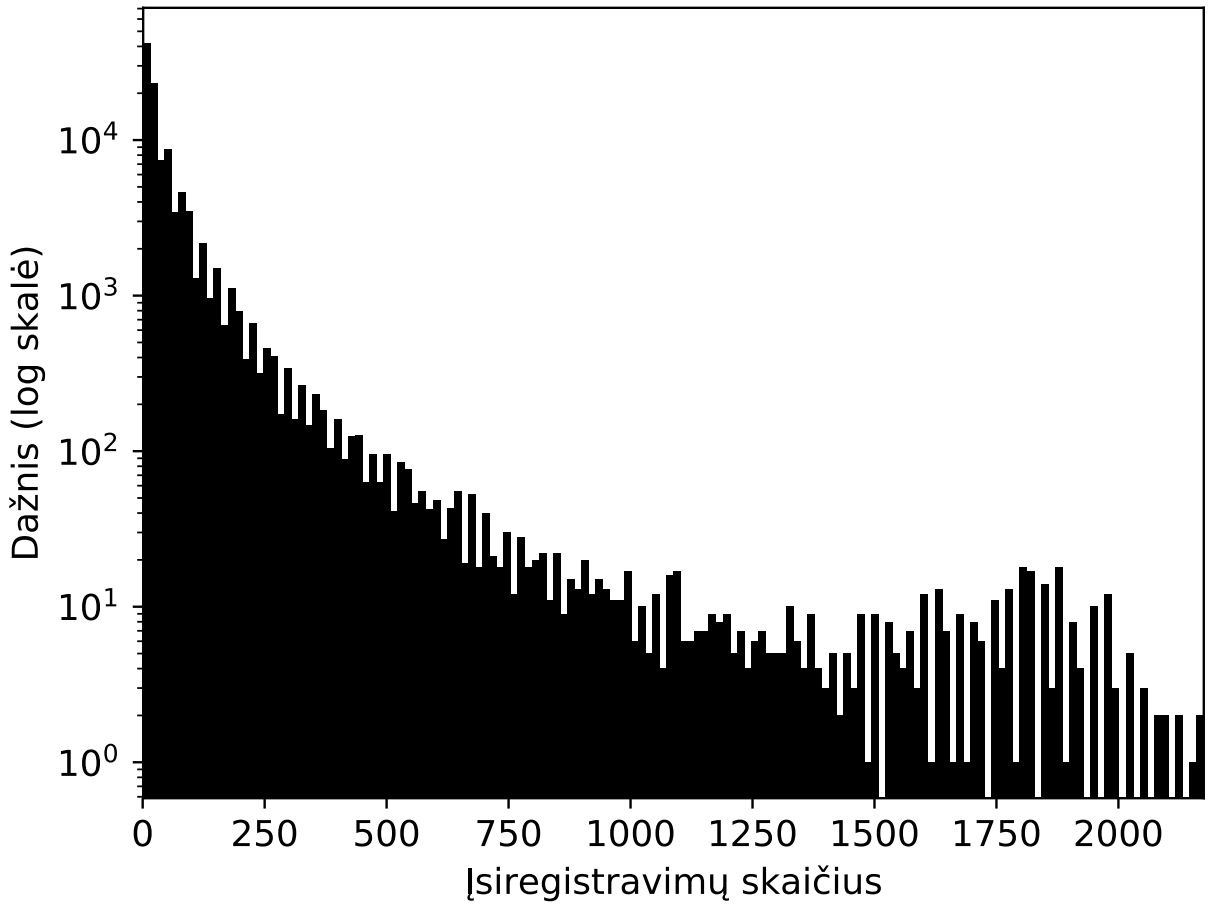
Atliekant GS užklausas svarbu, kad vartotojai įsiregistruotų pakankamai dažnai. Gowalla duomenų rinkinyje atmetus tuos vartotojus, kurie nėra atlikę nė vieno įsiregistravimo (89.499 vartotojai) buvo įvertintos įsiregistravimų statistinės charakteristikos. Rezultatai vaizduojami 2 lentelėje. Vidutinis vartotojas atlieka 60 įsiregistravimų, tikėtina kad toks įsiregistravimų skaičius tenkantis vienam vartotojui bus pakankamas, vykdant GS užklausas atlikti vietovės spėjimus ar pateikti rekomendacijas. Lyginant su vidurkiu, maža moda bei mediana ir didelis standartinis nuokrypis rodo, kad įsiregistravimų skaičiaus pagal vartotoją pasiskirstymo funkcija turi ilgą uodegą, todėl yra daug vartotojų, atliekančių didelį kiekį įsiregistravimų.

Vidurkis	Standartinis nuokrypis	Moda	Mediana
60,16	136,18	25	25

2 lentelė. Gowalla įsiregistravimų statistinės charakteristikos

Remiantis 5 pav. bei 3 lentele, vos 5% vartotojų atliko $2.644.771$ įsiregistravimų. Yra didelis kiekis vartotojų (apie 10.000) kurie atlieka daug įsiregistravimų, todėl galima tikėtis, kad Gowalla duomenų rinkinyje bus pakankamai įsiregistravimų, kad būtų galima sėkmingai atlikti GS užklausas. Vartotojų įsiregistravimų skaičiaus histograma 5 pav. patvirtina tai, kad maža dalis vartotojų

atlieka didžiąją dalį įsiregistravimų. Šioje iliustracijoje patvirtinama, kad vienam vartotojui tenkančių įsiregistravimų skaičiaus pasiskirstymas yra heterogeniškas: pasiskirstymo funkcija turi ilgą uodegą, t.y. labai daug vartotojų turi mažai įsiregistravimų.



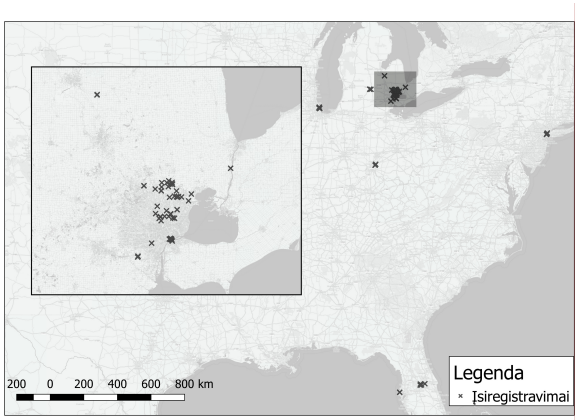
5 pav. Gowalla vartotojų įsiregistravimų skaičiaus histograma.

Procentilis	50	75	90	95
Įsiregistravimų skaičius	482.323	1.443.726	2.866.038	3.798.119

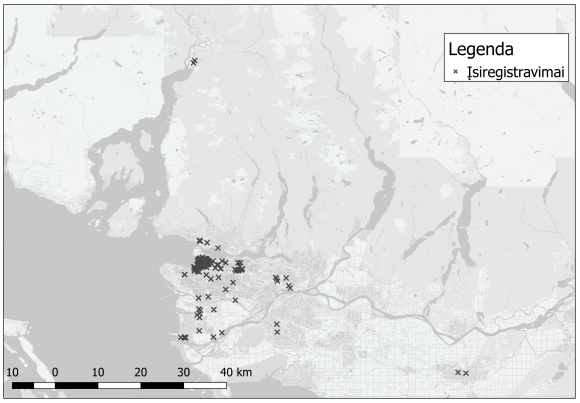
3 lentelė. Gowalla vartotojui tenkančių įsiregistravimų skaičiaus procentiliai

Vartotojų įsiregistravimų pavyzdžiai vaizduojami 6 paveikslėlyje. Iliustracijoje 6a dauguma įsiregistravimų vienoje vietovėje, tačiau yra ir toliau esančių, pavienių įsiregistravimų. Įsiregistravimų kiekis $n = 116$. Toks įsiregistravimų pasiskirstymas rodo, kad GS užklausų pagalba turėtume identifikuoti vartotojui pažįstamą regioną. Naudingos informacijos čia galima gauti ir iš atokesnių, pavienių įsiregistravimų. Nors 6b iliustracijoje vaizduojamų įsiregistravimų yra gerokai daugiau ($n = 275$) visi įsiregistravimai atlikti vienoje vietovėje. GS užklausomis čia galima identifikuoti vieną ar keletą regionų kurie vartotojui yra pažįstami.

Šiame darbe aprašytas metodas įsiregistravimų grupėms sudaryti remiasi atstumu tarp įsiregistravimų. GS užklausos nebūtų tokios naudingos jei visi atstumai tarp dviejų vienas po kito einančių to paties vartotojo įsiregistravimų būtų labai dideli arba labai maži. 7b iliustracijoje atsispindi, kad 63% vienas po kito sekančių įsiregistravimų atliekami ne toliau kaip 5km vienas nuo kito. Tačiau

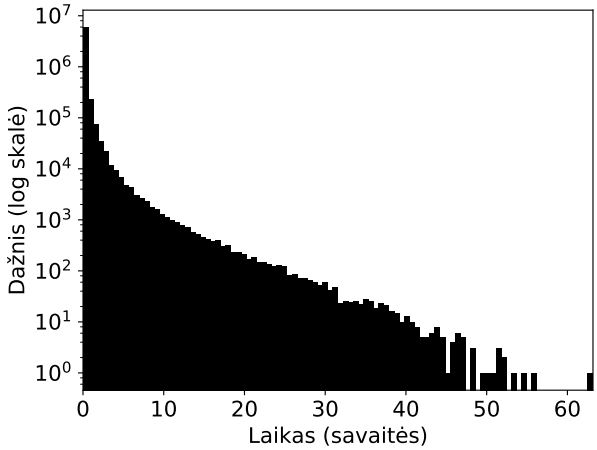


(a) Vartotojo $id = 2200$ įsiregistravimai.

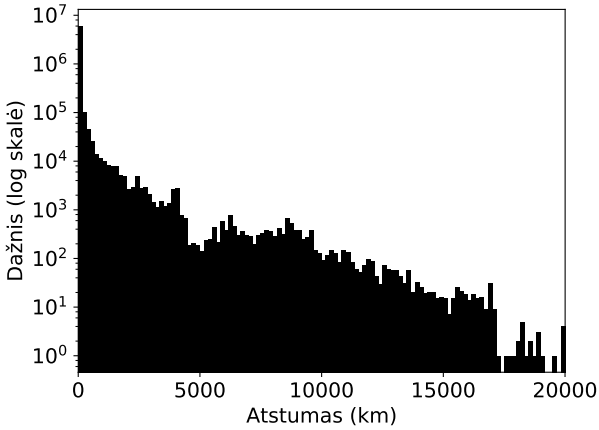


(b) Vartotojo $id = 1992$ įsiregistravimai.

6 pav. Vartotojų įsiregistravimų pavyzdžiai.



(a) Laiko skirtumų tarp gretimų įsiregistravimų



(b) Atstumo skirtumų tarp gretimų įsiregistravimų

7 pav. Įsiregistravimų charakteristikų histogramos

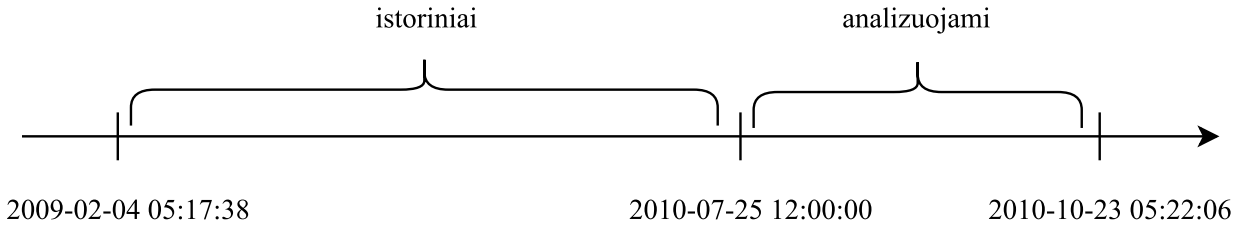
išlieka ir tokių, kurie atliekami dideliais atstumais. Analizuojant laiko skirtumus tarp įsiregistravimų, 7a iliustracijoje pastebima panaši tendencija kaip ir geografinių atstumų atveju. 60% vienas po kito sekančių įsiregistravimų įvyko 12 valandų laiko tarpu.

Prieš atliekant vartotojų įsiregistravimų vietovių spėjimus bei rekomendacijų teikimą, vartotojai buvo sugrupuoti į 3 kategorijas. Naudojant K-vidurkių klasterizavimo algoritmą iš Python bibliotekos scikit-learn [22], klasteriai buvo sudaromi remiantis šiomis vartotojo charakteristikomis:

- istorinių įsiregistravimų kiekis;
- vidutinis atstumas tarp istorinių įsiregistravimų;
- atstumo tarp istorinių įsiregistravimų standartinis nuokrypis;
- istorinių įsiregistravimų klasterių kiekis;
- vidutinis atstumas tarp istorinių įsiregistravimų klasterių;
- vidutinis laiko skirtumas tarp vienas po kito einančių įsiregistravimų;

- laiko skirtumo tarp vienas po kito einančių įsiregistravimų standartinis nuokrypis;
- didžiausias laiko skirtumas tarp dviejų vienas po kito einančių įsiregistravimų.

Kaip vaizduojama 8 paveikslėlyje, istoriniais įsiregistravimais čia laikomi tie įsiregistravimai, kurie buvo atlikti iki 2010 liepos 25 dienos. Kadangi matavimo vienetai tarp charakteristikų nėra vienodi, duomenys buvo normalizuoti charakteristikų atžvilgiu. Taip pat pašalinti tie vartotojai, kurie: turėjo vieną arba mažiau įsiregistravimų, arba neturėjo įsiregistravimų klasterių tarp istorinių duomenų. Gautų grupių charakteristikos vaizduojamos 4 lentelėje.



8 pav. Įsiregistravimų suskirstymo į istorinius bei analizuojamus diagrama.

Grupė	Vartotojų skaičius	Atstumas tarp klasterių (km)	Atstumas tarp įsireg. (km)	Laikas tarp įsireg. (h)	Įsireg. kiekis	Klasterių kiekis	Draugų kiekis
0	28 142	269,93	53,09	55,01	74,88	3,85	15,79
1	2 030	3052,92	476,12	77,97	65,45	4,08	26,28
2	1 085	809,11	30,25	8,69	655,55	17,18	51,57

4 lentelė. Vartotojų grupių charakteristikų vidurkiai

Remiantis gautomis grupių charakteristikomis, galima apibūdinti kiekvieną iš grupių. 0 ir 1 grupei priklausantys vartotojai yra panašiausi. Tačiau 0 grupės vartotojai mažiau keliauja, jų įsiregistravimų klasteriai ir vienas po kito einantys įsiregistravimai yra arčiau vienas kito, taip pat jie vidutiniškai atlieka daugiau įsiregistravimų nei 0 grupės vartotojai. Didžioji dalis vartotojų patenka į 0 grupę. Tarp turimų duomenų aktyviausi yra 2 grupės vartotojai. Jie atlieka daugiausiai įsiregistravimų, iš kurių pavyksta sudaryti daugiau klasterių nei 0 ar 1 grupės vartotojams. 2 grupės vartotojų yra mažiausiai.

4.2. Užklausų vykdymas

Atliekant užklausų vykdymą, Minimalios Kokybės Ribos algoritmu grupuojant įsiregistravimus minimalus klasterio dydis buvo parinktas $k = 3$. Tokia parametro reikšmė leidžia daryti prielaidą, kad vartotojas bent 3 kartus įsiregistravo toje vietovėje ir yra susipažinęs su jos apylinkėmis. Maksimalaus atstumo tarp bent vieno elemento klasteryje (klasterio centro) parametras buvo parinktas $r = 6$ km. Toks parametro dydis buvo parinktas, nes maždaug tokį atstumą žmogus gali įveikti per valandą pėsčiomis. Vadinasi vartotojas savo, ar savo draugų klasterį gali pasiekti per valandą. Taigi, klasterį apibrėžia bent 3 įsiregistravimai nutolę nuo arčiausiai esančio įsiregistravimo ne daugiau kaip per 6 km.

Tikrinant sistemos veikimą iš eilės buvo imama po vieną įsiregistravimą įvykusį nuo 2010 liepos 25 dienos 12:00. Kiekvienam įsiregistravimui buvo bandoma atlikti visų tipų, aprašytų 3.3.1

skirsnyje, GS užklausas. Tuomet įsiregistravimas buvo įtraukiamas į istorinių duomenų bazę ir vartotojui atlikusiam įsiregistravimą iš naujo sudaromi klasteriai, su minimaliu klasterio dydžiu $k = 3$ ir atstumo parametru $r = 6$ km. Pavykus aptikti įsiregistravimą naujoje, draugams pažįstamoje vietovėje buvo atrinkti tie draugai, kurie toje vietovėje lankėsi per paskutines 5 dienas. Laikoma, kad tie draugai ir įsiregistravimą atlikęs vartotojas sudaro užklausos rezultatą. Apdorojus 500 įsiregistravimų pavyko aptikti:

- 12 įsiregistravimų naujose, draugams pažįstamose vietovėse;
- 63 įsiregistravimus naujose, draugams nepažįstamose vietovėse;
- 425 įsiregistravimus pažįstamose vietovėse.

Iš įsiregistravimų naujose, draugams pažįstamose vietovėse buvo atrinkti tie draugai, kurie atitinkamuose klasteriuose įsiregistravo per paskutines 5 dienas. Iš atrinktų draugų ir vartotojų, atlikusių naujus įsiregistravimus buvo sudarytos 3 vartotojų ir jų draugų grupės, tinkamos reklamos pasiūlymų teikimui.

Taip pat verta paminėti, kad paskutinis analizuotas įsiregistravimas įvyko 2010 liepos 25 dieną 12:31. Todėl naudojant Gowalla socialinio tinko duomenis per 30min pavyktų gauti apie 500 įsiregistravimų. Įvykdžius GS užklausas ir jų rezultatus panaudojant reklamos transliavimui, gaunamas naudingas įrankis, kurio pagalba reklama gali pasiekti tikslinę auditoriją.

5. Užklausų taikymai

5.1. Vietovės spėjimas pagal parametrus

Šiame poskyryje, pagal 3.3.1 skirsnyje pateiktus algoritmus sudarius vartotojų įsiregistravimų klasterius, modeliuojamas vartotojų įsiregistravimų atlikimas ir bandoma nuspėti sekantį vartotojo įsiregistravimą. Skirsnyje 5.1.1 pateikiami modeliavimo rezultatai, kai remiamasi tik vartotojo vietovės duomenimis. 5.1.2 skirsnyje vartotojų įsiregistravimų vietoves bandoma nuspėti jau egzistuojančiu modeliu, kai spėjimai atliekami remiantis tik draugų įsiregistravimų informacija. 5.1.3 skirsnyje prie modelio prijungiama vartotojų socialinių ryšių informacijas. Skirsnyje 5.1.4 įsiregistravimams nuspėti atsižvelgiama į vietovę ir laiką tarp įsiregistravimų.

2 algoritmas. Vietovės spėjimų modeliavimas

Įvestis: $v \in V, t_0 \in T$

Išvestis: Įsiregistravimų vietovių l spėjimo rezultatai

$c_0 \leftarrow$ *SekantisĮsiregistravimas*(v, t_0)
$C_{nauji} \leftarrow$ *GautiĮsiregistravimus*$(v, t \in c_0)$
$C_{istoriniai} \leftarrow C_v \setminus C_{nauji}$
$KL_v \leftarrow$ *MinKokybėsRibosKlasterizavimas*$(C_{istoriniai})$
perėjimo_tikimybės $\leftarrow$ *SkaičiuotiPerėjimoTikimybes*(KL_v)
sėkmingi_spėjimai $\leftarrow \perp$
for all $c \in C_{nauji}$ **do**
  $\widehat{l} \leftarrow$ *ParinktiVietovę*(*perėjimo_tikimybės*, c_0)
  if $\widehat{l} == l_{c_0}$ **then**
    sėkmingi_spėjimai.add(1)
  else
    sėkmingi_spėjimai.add(0)
  end if
  $C_{istoriniai}.add(c)$
  $KL_v \leftarrow$ *MinKokybėsRibosKlasterizavimas*$(C_{istoriniai})$
  perėjimo_tikimybės $\leftarrow$ *SkaičiuotiPerėjimoTikimybes*(KL_v)
  $c_0 \leftarrow c$
end for
return *sėkmingi_spėjimai*

Vietovės spėjimai kiekvienu iš atvejų buvo atliekami pagal 2 algoritmą. Pradinio laiko momento t_0 reikšmė buvo parinkta pagal laiką, nuo kurio įsiregistravimai laikomi analizuojamais, t.y. $t_0 =$ 2010-07-25 12:00. Pagal t_0 parenkamas sekantis vartotojo v įsiregistravimas. Toliau gaunami visi vartotojo įsiregistravimai, vykę po įsiregistravimo c_0. Minimalios kokybės ribos algoritmo pagalba iš istorinių įsiregistravimų sudaromi įsiregistravimų klasteriai $kl \in KL_v$. Tuomet įvertinamos perėjimo tarp klasterių tikimybės. Skirtingiems sekančio įsiregistravimo vietovės spėjimo modeliavimo būdams buvo naudojamos skirtingos funkcijos skaičiuoti perėjimo tarp klasterių tikimybėms. Metodas *ParinktiVietovę* parenka vietovę sugeneruojant standartinį tolygų skirstinį turintį atsitiktinį dydį $x \sim U(0, 1)$ ir pagal turimas perėjimo tikimybes parenkamas sekantis klasteris kl_p:

$$kl_p = kl_j : \mu_{i0} + \cdots + \mu_{ij-1} < x \leq \mu_{i0} + \cdots + \mu_{ij}, \tag{5.1}$$

čia i žymi indeksą klasterio, kuriame vyko praeitas įsiregistravimas. Parinkus vietovę tikrinama, ar spėjimas buvo atliktas sėkmingai. Atlikus spėjimą šis įsiregistravimas pridedamas prie istorinių ir klasteriai bei perėjimo tikimybės sudaromos iš naujo.

Funkcija *SekantisĮsiregistravimas* reikalinga atliekant įsiregistravimų simuliavimą. Jos pagalba, žinant praeito įsiregistravimo laiką, galima gauti sekantį įsiregistravimą. Taip pat ši funkcija reikalinga skaičiuojant laiką tarp istorinių vartotojo įsiregistravimų.

13 apibrėžimas. *SekantisĮsiregistravimas*(v, t_0). Funkcija *SekantisĮsiregistravimas*: $V \times T \to C$ grąžina anksčiausiai vykusį vartotojo įsiregistravimą c_i iš sekos $(c_i, c_{i+1}, \ldots, c_{i+n-1})$ = *GautiĮsiregistravimus*(v, t_0).

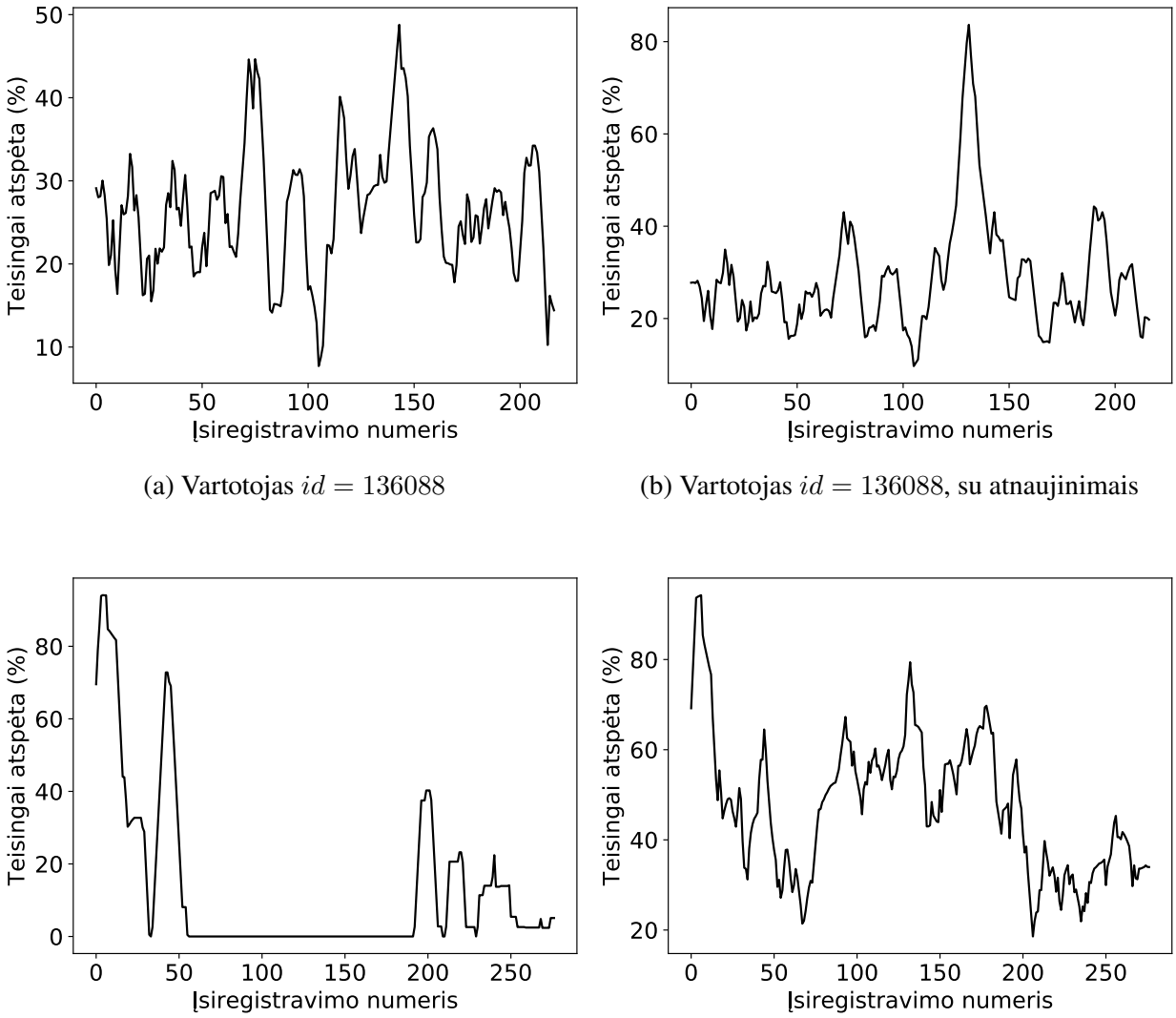
5.1.1. Pagal vietovę

Pradinės vartotojų įsiregistravimų grupės buvo sudaromos Minimalios Kokybės Ribos klasterizavimo algoritmo pagalba, su klasterio diametro parametru $r = 3$ ir minimalaus klasterio dydžio parametru $k = 3$. Skirsnyje 3.3.2 aprašytu metodu iš turimų klasterių sudaromos perėjimo tikimybių matricos kiekvienam vartotojui. Tuomet tikrinama, kuriame iš klasterių buvo paskutinis įsiregistravimas ir pagal tai Markovo grandinių pagalba, gaunamos tikimybės pereiti į kiekvieną iš galimų klasterių arba likti tame pačiame klasteryje. Tolimesnė tyrimo dalis buvo atliekama dviem būdais. Pirmuoju atveju, pabandžius nuspėti įsiregistravimą, imamas sekantis įsiregistravimas ir vėl atliekamas spėjimas. Antruoju atveju, po kiekvieno spėjimo, naujas įsiregistravimas įtraukiamas prie istorinių vartotojo duomenų ir įsiregistravimų klasteriai sudaromi iš naujo. Turint naujus įsiregistravimo klasterių duomenis perskaičiuojamos Markovo grandinės perėjimo tikimybės. Taigi, pirmuoju atveju perėjimo tikimybės nekinta įvykstant naujiems įsiregistravimams. Antruoju atveju tikimybės kaskart perskaičiuojamos.

Įsiregistravimų vietovės spėjimai buvo atlikti dviems vartotojams. Parenkant vartotojus, buvo siekiama modeliavimus atlikti vartotojams, kurių įsiregistravimo įpročiai būtų nevienodi. Praktiškai visi vartotojo v_1 įsiregistravimai buvo atliekami pažįstamose vietovėse. Panaudojus GS užklausas, jam pavyko sudaryti 5 pažįstamų vietovių klasterius. Vidutinis atstumas tarp įsiregistravimo grupių centrų 218 km, minimalus atstumas tarp dviejų klasterių centrų 7 km, maksimalus: 364 km. Tai rodo, kad klasteriai yra išsidėstę gana toli vienas kito. Iš viso šis vartotojas iš viso atliko 450 įsiregistravimų. Atskyrus istorinius įsiregistravimus liko 224 modelio analizuojami įsiregistravimai.

Vidutinis atspėtų įsiregistravimų skaičiaus kitimas vartotojui v_1 ($id = 136088$) vaizduojamas 9a pav. Grafike matoma, kad įsiregistravimo vietovių sėkmingų spėjimų rezultatai svyruoja tarp 10% ir 50%. Spėjimų procentas pagerėjo, kai po kiekvieno atlikto įsiregistravimo buvo iš naujo sudaromi įsiregistravimų klasteriai. 9b paveikslėlyje pastebima, kad spėjimų rezultatai gana stipriai varijavo. Maždaug ties c_140 įsiregistravimu buvo pasiektas 80% teisingai atspėtų vietovių rezultatas. Klasterių ir tikimybių atnaujinimas po kiekvieno įsiregistravimo leidžia tiksliau nuspėti būsimą vartotojo vietovę.

Kitokia situacija matoma ir 9c bei 9d paveikslėliuose, čia vaizduojami įsiregistravimo spėjimų procentai vartotojui v_2 ($id = 3478$). Šis vartotojas turi tik 2 įsiregistravimų klasterius bei 367 įsiregistravimus, iš kurių 80 priskirtų prie istorinių duomenų. Atstumas tarp klasterių centrų: 20 km. Modelis ypač sėkmingai identifikavo sekančias vietoves pirmiesiems įsiregistravimams. Šiam vartotojui pirmieji spėjimai buvo atliekami sėkmingiau nei vartotojui v_1, nes dauguma vartotojo v_2 pirmųjų įsiregistravimų vyko vietovėse kurios buvo identifikuotos tarp istorinių duomenų. Tačiau



(a) Vartotojas $id = 136088$

(b) Vartotojas $id = 136088$, su atnaujinimais

(c) Vartotojas $id = 3478$

(d) Vartotojas $id = 3478$, su atnaujinimais

9 pav. Vartotojų įsiregistravimų vietovės spėjimai

vėlesnių įsiregistravimų procentas ėmė kristi. Viena iš galimų priežasčių – nauji įsiregistravimai buvo atliekami naujose vietovėse $L_0 \subset L$, kurios anksčiau nepateko į nė vieną iš vartotojo v_2 klasterių. Atliekant spėjimus su tikimybių perskaičiavimais po kiekvieno įsiregistravimo gauti geresni spėjimų rezultatai. Lyginant su modeliavimo rezultatais gautais vartotojui v_1, įsiregistravimai buvo spėjami sėkmingiau. Tikėtina, kad vartotojas v_2 įsiregistravimus atliko jam pažįstamose vietovėse.

Svarbu atkreipti dėmesį į tai, kad visi įsiregistravimai, kurie nepatenka į nė vieną iš vartotojo klasterių, yra priskiriami vienai nežinomų vietovių grupei, t.y. $c \in C_v$, kurie nėra atliekami pažįstamose vietovėse $\nexists kl \in KL_v : c \in kl$ priskiriami vienai įsiregistravimų grupei. Tai yra gana didelė abstrakcija, nes visų galimų nežinomų vietovių aibė yra labai didelė. Šis modelio trūkumas šiek tiek ištaisomas, kai vartotojo įsiregistravimų klasteriai yra perskaičiuojami. Tuomet iš naujų įsiregistravimų nežinomose vietovėse galima sudaryti naujus klasterius ir spėjimai bus tikslesni. Tačiau modelis vis tiek niekada nesugebės įvardinti konkrečios tikėtinos įsiregistravimo vietovės, jei ten nebus sudarytas įsiregistravimų klasteris.

5.1.2. Pagal draugus

Modeliuojant galimas vartotojų įsiregistravimų vietoves, buvo išbandytas straipsnyje [29] nurodytas metodas nuspėti sekančio įsiregistravimo vietovę remiantis draugų lankytomis vietovėmis. Kadangi šiame darbe nagrinėjamos ne konkrečios vietovės o tam tikros teritorijos, apibrėžtos vartotojo įsiregistravimų, [29] aprašytas metodas buvo pritaikytas nuspėti vartotojo įsiregistravimų klasterį, kuriame bus atliekamas sekantis įsiregistravimas.

Įvykus naujam vartotojo v įsiregistravimui c_m, apskaičiuojamas tikimybių vektorius, kurio kiekvienas elementas nurodo tikimybę sekančiam įsiregistravimui įvykti klasteryje kl_i:

$$(\mathbf{P}(c_{m+1} \in kl_1), \ldots, \mathbf{P}(c_{m+1} \in kl_n)) \tag{5.2}$$

Tikimybė vartotojui v įsiregistruoti klasteryje kl_j $\mathbf{P}(c_{m+1} \in kl_j)$ apskaičiuojama

$$\mathbf{P}(c_{m+1} \in kl_j) = \frac{\sum_{v_k \in \Gamma_v} d_{soc}(v, v_k) \cdot \mathbb{1}_{v_k,j}}{\sum_{v_k \in \Gamma_v} d_{soc}(v, v_k)} \tag{5.3}$$

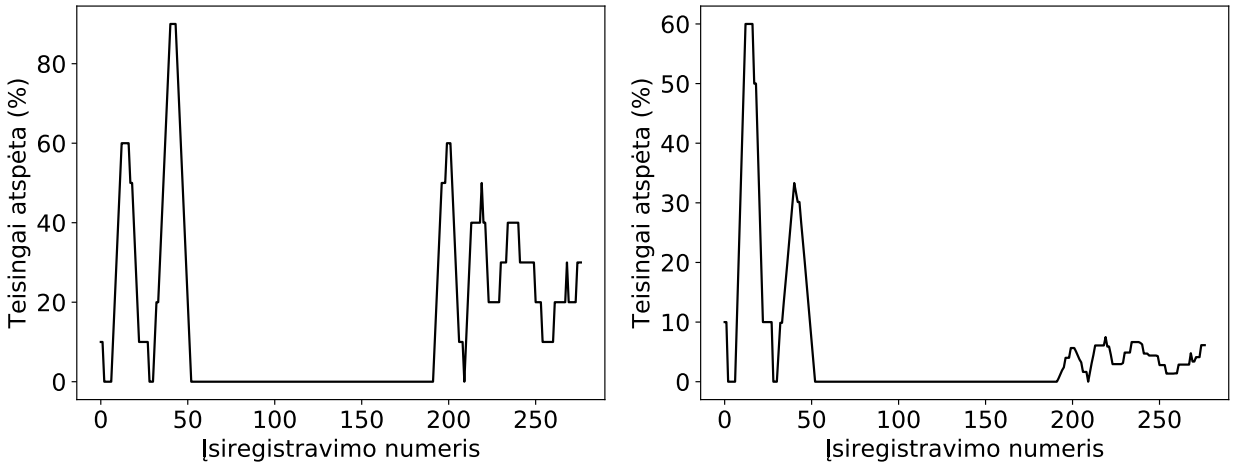
$\mathbb{1}_{v_k,j}$ reikšmė yra 1 jei vartotojas v_k turi įsiregistravimą klasteryje kl_j ir 0 kai įsiregistravimo ten nėra:

$$\mathbb{1}_{v_k,j} = \begin{cases} 1\,, \text{kai } \exists c_m \in C_{v_k} \text{ ir } \exists c_n \in kl_j : d_{geo}(c_m, c_n) \leq 3, \\ 0\,, \text{kitu atveju.} \end{cases} \tag{5.4}$$

Atstumo tarp įsiregistravimo ir įsiregistravimų klasterių centrų riba buvo parinkta 3 km, taip siekiant modelius lyginti panašiomis sąlygomis, kaip ir 5.1.1 skirsnyje.

Atliekant įsiregistravimų vietovių spėjimus, tikimybės buvo skaičiuojamos naudojant šį modelį. Vartotojo v atžvilgiu atliekant vietovės spėjimus, kiekvienam jo klasteriui $kl_j \in KL_v$ remiantis 5.3 formule paskaičiuota tikimybė $\mathbf{P}(c_i \in kl_j)$, kad sekantis įsiregistravimas c_i bus atitinkamame klasteryje kl_j. Nors kiekviena iš tikimybių $\mathbf{P}(c_i \in kl_j)$, $j = 1, \ldots, n$, kur $n = |KL_v|$ yra intervale $[0, 1]$, tačiau jų suma $\sum_{j=1}^{n} \widehat{c}_{i,j}$ gali viršyti 1. Todėl kiekviena iš tikimybių buvo sumažinta $\sum_{j=1}^{n} \mathbf{P}(c_i \in kl_j)$ kartų. Taigi, kiekvienam klasteriui paskaičiavus tikimybę, kad sekantis vartotojo v įsiregistravimas bus tame klasteryje, pagal ją atsitiktinai buvo parenkamas vienas iš klasterių ir spėjama, kad įsiregistravimas vyks būtent tame klasteryje.

Šis modeliavimas buvo atliktas vartotojui v_2 ($id = 3478$). Kaip ir 5.1.1 skirsnyje, buvo bandoma nuspėti 286 šio vartotojo įsiregistravimų vietovių. Pirmuoju atveju atliekant spėjimus tikimybės ir vartotojo klasteriai nebuvo atnaujinami. Iš 286 įsiregistravimų, jų klasterius pavyko atspėti 36 kartus (apie 13% atvejų). 10a pav. galima pastebėti, kad sėkmingiausiai nuspėti įsiregistravimų vietoves modeliui pavyko pradžioje, vėliau procentas krito ir tik ėmė šiek tiek kilti ties paskutiniais įsiregistravimais. Panaši situacija matoma ir 10b pav. Čia dauguma sėkmingų spėjimų atlikta su pirmaisiais ir paskutiniais įsiregistravimais. Paskutinius įsiregistravimus spėti pavyko šiek tiek geriau. Iš viso sėkmingai nuspėta 46 vietovės (apie 16% atvejų). Šiuo atveju spėjimai buvo atliekami sėkmingiau, nes po kiekvieno įsiregistravimo atlikus spėjimą, vartotojo v_2 įsiregistravimų klasteriai ir atitinkamai tikimybės įsiregistruoti kiekviename iš klasterių buvo perskaičiuojamos įtraukiant naują įsiregistravimą.

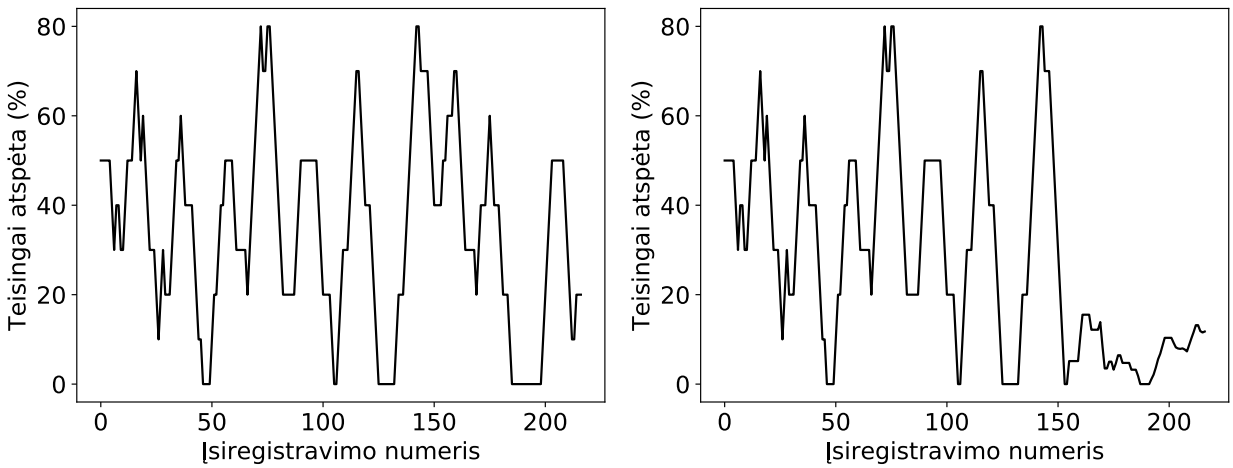


(a) Be atnaujinimų

(b) Su atnaujinimais

10 pav. Vartotojui $id = 3478$ atliktų įsiregistravimų vietovės spėjimų, remiantis draugų įsiregistravimais, rezultatai

Įsiregistravimų vietovių spėjimai buvo atliekami ir vartotojui v_1 ($id = 136088$). 11 pav. galima pastebėti, kad spėjimai buvo atliekami sėkmingiau nei vartotojui v_2. Šiam vartotojui įsiregistravimų vietoves nuspėti lengviau, nes jis turi tik du įsiregistravimų klasterius. Atliekant modeliavimą be atnaujinimų, sėkmingai atspėta 35% įsiregistravimų. Šis dydis sumažėjo iki 27% kai tikimybės buvo atnaujinamos įvykus kiekvienam įsiregistravimui. Pagrindinė to priežastis – dauguma naujų įsiregistravimų vyko naujose vietovėse, kur buvo sudaryti nauji klasteriai. Tose vietovėse lankėsi ir vartotojo v_2 draugai, todėl ir perskaičiuojant tikimybes buvo tikėtasi, kad sekantys vartotojo įsiregistravimai vyks būtent ten.



(a) Be atnaujinimų

(b) Su atnaujinimais

11 pav. Vartotojui $id = 136088$ atliktų įsiregistravimų vietovės spėjimų, remiantis draugų įsiregistravimais, rezultatai

Gauti rezultatai parodo, kad vien draugų įsiregistravimų tendencijų neužtenka siekiant sėkmingai nuspėti vartotojų įsiregistravimus. Šiuo atveju bendri spėjimų procentai panašūs, kaip ir

skirsnyje 5.1.1 atliktuose modeliavimuose. Skirtumai išryškėja vertinant šiuos vartotojus atskirai. Vartotojui v_1 įsiregistravimų vietoves sėkmingiau nuspėti sekėsi tuo atveju, kai buvo nagrinėjama jo draugų įsiregistravimų informacija. Tuo tarpu vartotojui v_2 modelis veikė geriau, kai buvo analizuojami tik jo paties įsiregistravimų duomenys. Taigi vartotojo draugų įsiregistravimų vietovės bei draugystės ryšių stiprumo įvertis socialiniuose tinkluose suteikia naudingos informacijos bandant atspėti įsiregistravimų vietoves. Tikėtina, kad įvertinus praeityje lankytas vartotojo vietoves bei jo draugų lankytas vietoves, bus galima atlikti tikslesnius spėjimus nei vertinant šiuos elementus atskirai.

5.1.3. Pagal vietovę ir draugus

Modeliuojat galimas vartotojų įsiregistravimų vietoves pagal draugus socialiniame tinkle ir istorinius įsiregistravimų duomenis buvo išbandyti du metodai. Pirmasis istorinius įsiregistravimų ir draugų įsiregistravimų duomenis vertina atskirai. Tuo tarpu antrasis metodas šiuos duomenis vertina kartu.

Pirmuoju atveju buvo naudoti vietovės spėjimų modeliai aprašyti 5.1.2 bei 5.1.1 skirsniuose. Įvykus naujam vartotojo $v \in V$ įsiregistravimui $c_m \in C_v$, pagal praeito įsiregistravimo $c_{m-1} \in C_v$ vietovę, skirsnyje 5.1.1 aprašytu būdu, sudaromas tikimybių įsiregistruoti viename iš klasterių $kl_i \in KL_v$ vektorius. Tarkime praeito įsiregistravimo klasteris yra kl_i. Tuomet sekančio įsiregistravimo vietovės tikimybių vektorius remiantis istoriniais vartotojo v įsiregistravimų duomenimis yra

$$(\mu_{i0}, \mu_{i1}, \ldots, \mu_{in}). \tag{5.5}$$

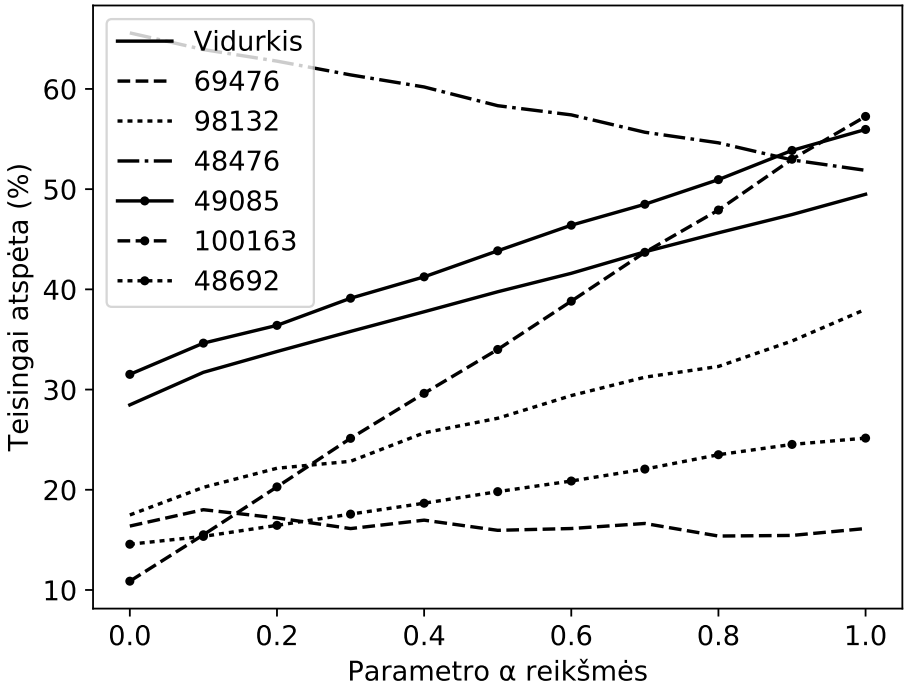
Panašų sekančio įsiregistravimo vietovės tikimybių vektorių galima sudaryti ir iš draugų įsiregistravimų duomenų. 5.1.2 skirsnyje aprašomas metodas vartotojo v įsiregistravimo vietovei nuspėti pagal tai, ar jo draugai $v_j \in \Gamma_v$ lankėsi jam pažįstamose vietovėse kl_i. Atsižvelgiant į draugystės ryšio socialiniame tinkle stiprumą, po kiekvieno įsiregistravimo sudaromas tikimybių vektorius, nusakantis tikimybes sekančiam įsiregistravimui įvykti viename iš vietovių klasterių kl_i:

$$(\mathbf{P}(c_m \in kl_0), \mathbf{P}(c_m \in kl_1), \ldots, \mathbf{P}(c_m \in kl_n)) \tag{5.6}$$

$\widehat{c}_0$ nusako tikimybę vartotojui v įsiregistruoti jam nežinomose vietovėse kl_0. T.y. tose vietovėse, kurios nepatenka į nė vieną iš šio vartotojui klasterių. Šis dydis būtų sudaromas remiantis tuo, ar vartotojo v draugai lankėsi tose vietovėse. Į vartotojui v mažai pažįstamas vietoves gali patekti daugybė, dažnai tarpusavyje nesusijusių vietovių. Kadangi šiuo atveju informacija apie draugų įsiregistravimus vartotojui v nežinomose vietovėse nėra prasminga, laikysime, kad visada $\widehat{c}_0 = 0$.

Toliau bendrai įvertinsime tikimybes gautas remiantis vartotojo v įsiregistravimais bei tikimybes gautas remiantis vartotojo v draugų Γ_v įsiregistravimais. Pažymėkime $M_i = (\mu_{i0}, \mu_{i1}, \ldots, \mu_{in})$, $P = (\mathbf{P}(c_m \in kl_0), \mathbf{P}(c_m \in kl_1), \ldots, \mathbf{P}(c_m \in kl_n))$. Tuomet bendrą tikimybių vektorių galima užrašyti

$${}_{L\Gamma}\widehat{C}_i = \alpha M_i + (1 - \alpha)P. \tag{5.7}$$



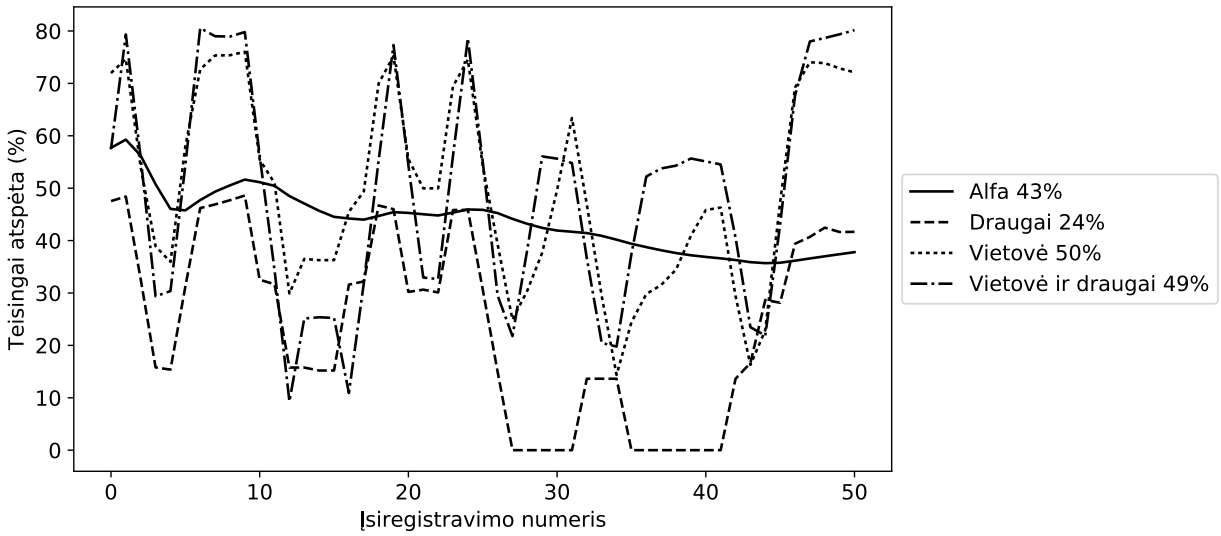
12 pav. Spėjimų rezultatai pagal istorinius įsiregistravimus ir draugų įsiregistravimų duomenis, naudojant skirtingas α reikšmes.

Čia $\alpha \in [0; 1]$ yra parametras nusakantis kiek įtakos spėjimams turės istoriniai vartotojo v įsiregistravimų duomenys ir kiek draugų įsiregistravimų vietovių tendencijos. Kadangi ne visada $0 \leq \alpha\widehat{\mu}_{ij} + (1-\alpha)\widehat{c}_j \leq 1, \forall j = 1, \ldots, n$, todėl prieš naudojant ${}_{LT}\widehat{C}_i$ tikimybių vektorių sekančiai įsiregistravimo vietovei nuspėti šis dydis turi būti normalizuojamas. Normalizavimas atliekamas kiekvieną iš tikimybių padalijant iš visų tikimybių sumos, taip užtikrinama kad lygybė $0 \leq \alpha\widehat{\mu}_{ij} + (1-\alpha)\widehat{c}_j \leq 1$ bus teisinga su $\forall j = 1, \ldots, n$.

Atliekant modeliavimus su skirtingomis α reikšmėmis, galima nustatyti kuri reikšmė geriausiai tinka kiekvienam iš vartotojų. Galimas toks atvejis, kai vartotojas turi vos keletą draugų, arba jo draugai beveik niekada nesilanko jam pažįstamose vietovėse. Tuomet prasminga parametrą α parinkti kuo didesnį. Tačiau kai kuriems vartotojams sekančio įsiregistravimo vietovė nuspėjama tiksliau, kai α reikšmė parenkama maža.

Atsitiktinai parinkus 30 vartotojų, turinčių bet 1 draugą ir bent 1 įsiregistravimą iš imties buvo pašalinti 2 vartotojai, kurių draugai nėra atlikę nė vieno įsiregistravimo. Gautai 28 vartotojų imčiai buvo atliekami sekančio įsiregistravimo vietovės spėjimai. Rezultatai buvo agreguoti – susumuotas visiems vartotojams sėkmingai atliktų spėjimų skaičius padalintas iš visų spėjimų bandymų skaičiaus. Rezultatai pateikiami 12 paveikslėlyje. Galima pastebėti, kad sąryšis tarp parametro α ir spėjimų rezultatų yra tiesinis. Taigi bendru atveju spėjimai atliekami sėkmingiau, kai atsižvelgiama tik į istorinius vartotojo įsiregistravimų duomenis. Kai kuriems vartotojams iš šios imties, spėjimai buvo atliekami sėkmingiau, su mažomis α reikšmėmis.

Dar vienas būdas atlikti sekančios įsiregistravimo vietovės spėjimus – sudaryti įsiregistravimų aibę iš vartotojo ir jo draugų įsiregistravimų ir gautą aibę suklasterizuoti. Klasteriai buvo sudaromi su minimaliu klasterio dydžiu $k = 3$ ir maksimaliu atstumu iki centro $r = 3km$. Toliau 3.3.2 skirsnyje aprašytu metodu iš gautų klasterių sudaryti perėjimo tikimybių matricas ir remiantis perėjimo tikimybėmis atsitiktinai parinkti sekančio įsiregistravimo vietovę. Šis metodas buvo



13 pav. Vartotojo $id = 49085$ sekančio įsiregistravimo vietovės spėjimų rezultatai naudojant skirtingus metodus.

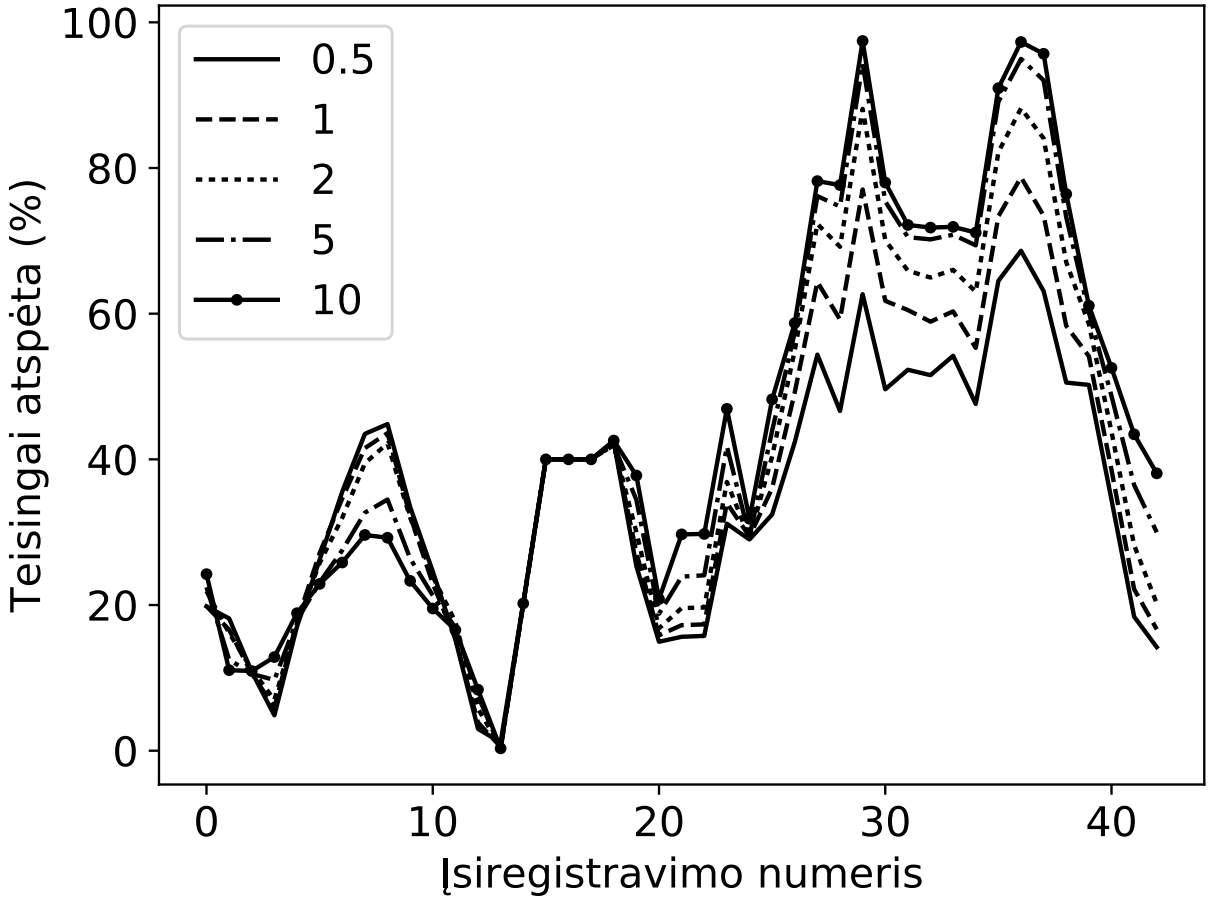
išbandytas vartotojui turinčiam 101 istorinį įsiregistravimą, tuo tarpu 16 jo draugų atliko 550 įsiregistravimų. Simuliacijos rezultatai, kartu su kitų vietovės spėjimo metodų bandymais pateikiami 13 paveikslėlyje. Sudarant klasterius iš šio vartotojo bei jo draugų įsiregistravimų pavyko atspėti 49% įsiregistravimų vietovių, taip pat kai kuriems vietovių spėjimams šis metodas buvo pranašiausias. Tačiau geriausiai veikė metodas, kai vietovė spėjama tik pagal vartotojo istorinių įsiregistravimų duomenis (atspėta 51% įsiregistravimų vietovių).

5.1.4. Pagal vietovę ir laiką

Norint nuspėti sekančio įsiregistravimo vietovę žinoti tik jų laiko neužtenka. Šiame darbe laikas buvo naudojamas apriboti bei pakeisti jau turimas tikimybes. Prieš pradedant sekančio įsiregistravimo vietovės spėjimus, vartotojui v sudaromi istorinių įsiregistravimų klasteriai $\{kl_1, \ldots, kl_n\} = KL_v$. Tuomet imamas sekantis įsiregistravimas c_{m+1} ir užfiksuojamas laikas t_Δ valandomis, praėjęs nuo paskutinio įsiregistravimo. Pagal gautą laiką nustatomas maksimalus atstumas, kurį vartotojas galėjo įveikti nuo paskutinio įsiregistravimo: $d_{max} = t_\Delta g_{max}$. Čia g_{max} didžiausias atstumas, kurį vartotojas gali įveikt per valandą. Pagal tai sudaromos tikimybės, sekančiam įsiregistravimui įvykti klasteryje kl_i ($\mathbf{P}(c_{m+1} \in kl_1), \ldots, \mathbf{P}(c_{m+1} \in kl_n)$):

$$\mathbf{P}(c_{m+1} \in kl_i) = \begin{cases} \frac{h}{r\sqrt{1+t_\Delta}} & \text{, kai } d_{geo}(l_{c_m}, kl_i) \leq r, \\ \frac{1}{d_{geo}(l_{c_m}, kl_i)} & \text{, kai } r < d_{geo}(l_{c_m}, kl_i) \leq d_{max}, \\ 0 & \text{, kai } d_{max} < d_{geo}(l_{c_m}, kl_i). \end{cases} \tag{5.8}$$

Kadangi gauti dydžiai gali būti didesni nei 1, kiekviena iš tikimybių buvo sumažinta $\sum_{i=1}^{n} \mathbf{P}(c_{m+1} \in kl_i)$ kartų. Šis tikimybių skaičiavimo būdas pagrįstas tuo, kad labiausiai tikėtina, jog sekantis įsiregistravimas c_{m+1} vyks vyks tame pačiame klasteryje kaip ir paskutinis įsiregistravimas c_m, jei nuo c_m iki c_{m+1} praėjo nedaug laiko, t.y. tokiu atveju $\sqrt{1+t_\Delta}$ reikšmė yra artima 1 ir $\frac{h}{r\sqrt{1+t_\Delta}} \approx \frac{h}{r}$. Jei laiko tarpas nuo paskutinio įsiregistravimo pakankamai didelis, tuomet šio klasterio svoris tikimybių skaičiavime mažėja. Likusių klasterių tikimybių svoriai priklauso tik nuo atstumo tarp praeito įsiregistravimo ir klasterio. Gali nutikti taip, kad $d_{max} < d_{geo}(l_{c_m}, kl_i), \forall kl_i \in KL_v$, t.y.

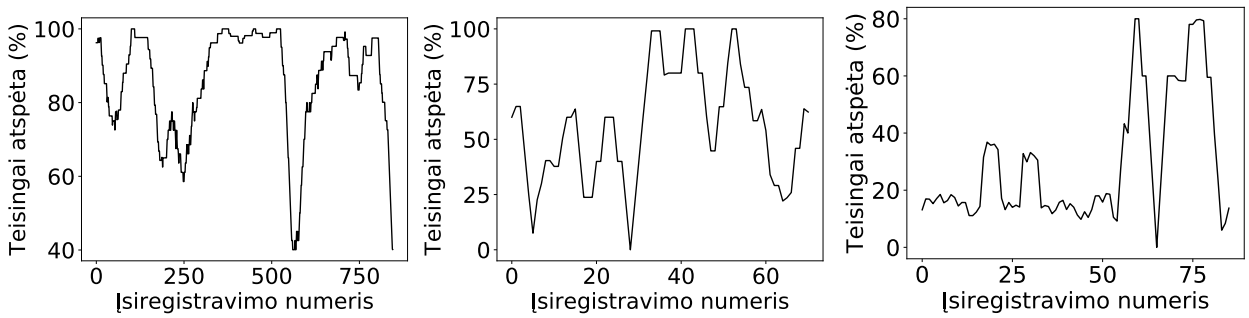


14 pav. Vartotojo $id = 34049$ sekančio įsiregistravimo vietovės spėjimai skirtingiems h.

paskutinis vartotojo v įsiregistravimas c_m buvo per toli nuo kiekvieno iš klasterių $kl_i \in KL_v$, todėl $\mathbf{P}(c_{m+1} \in kl_i) = 0$, $\forall kl_i \in KL_v$. Tokiu atveju spėjama, kad sekantis įsiregistravimas bus klasteryje kl_0, arba kitaip tariant nė viename iš istorinių įsiregistravimų klasterių.

Parametras h, atliekant šio tipo spėjimus, reguliuoja svorį, tenkantį tikimybei įsiregistruoti tame pačiame klasteryje. Šio parametro įtaka vartotojo įsiregistravimų vietovių spėjimams vaizduojama 14 paveikslėlyje. Maždaug nuo c_2 iki c_{12} įsiregistravimo, sėkmingiau veikė modeliai su mažesnėmis h reikšmėmis. Tai nutiko dėl to, kad čia įsiregistravimai buvo atliekami ne toje pat vietovėje kaip paskutinis įsiregistravimas, t.y. $c_m \in kl_i$, tuo tarpu $c_{m+1} \notin kl_i$. Tačiau nuo 20 įsiregistravimo, vietovės buvo spėjamos tiksliau tuo atveju, kai h reikšmės buvo didesnės – čia įsiregistravimai vyko tose pačiose vietovėse $c_s, c_{s+1} \in kl_j$. Parenkant didesnę h reikšmę sėkmingiau atspėjamos tos įsiregistravimų vietovės, kai dviejų vienas po kito sekančių įsiregistravimų vietovė lieka nepakitusi, kitaip tariant abu įsiregistravimai priklauso tam pačiam klasteriui. Didesnė h reikšmė mažina tikimybę, kad vartotojas sekančiu įsiregistravimu išeis iš paskutinio lankyto klasterio. Tais atvejais, kai vartotojas keliauja tarp klasterių, vietovės atspėjamos ne taip sėkmingai. 14 paveikslėlyje matoma, kad tarp atliekant spėjimus nuo c_5 iki c_{10} įsiregistravimų, h reikšmės didėjimas sumažino sėkmingų spėjimų atvejus 10%. Tačiau nuo c_{30} iki c_{35} įsiregistravimo sėkmingų spėjimų atvejų padaugėjo 10%.

Vietovių spėjimo rezultatai pateikiami 15 paveikslėlyje. Modeliavimai buvo atlikti trims vartotojams, patenkantiems į skirtingas vartotojų grupes. Sėkmingiausiai vietoves nuspėti pavyko vartotojui $id = 753$, vidutiniškai atspėta 85% atvejų. Šiam vartotojui vietovės buvo sėkmingai



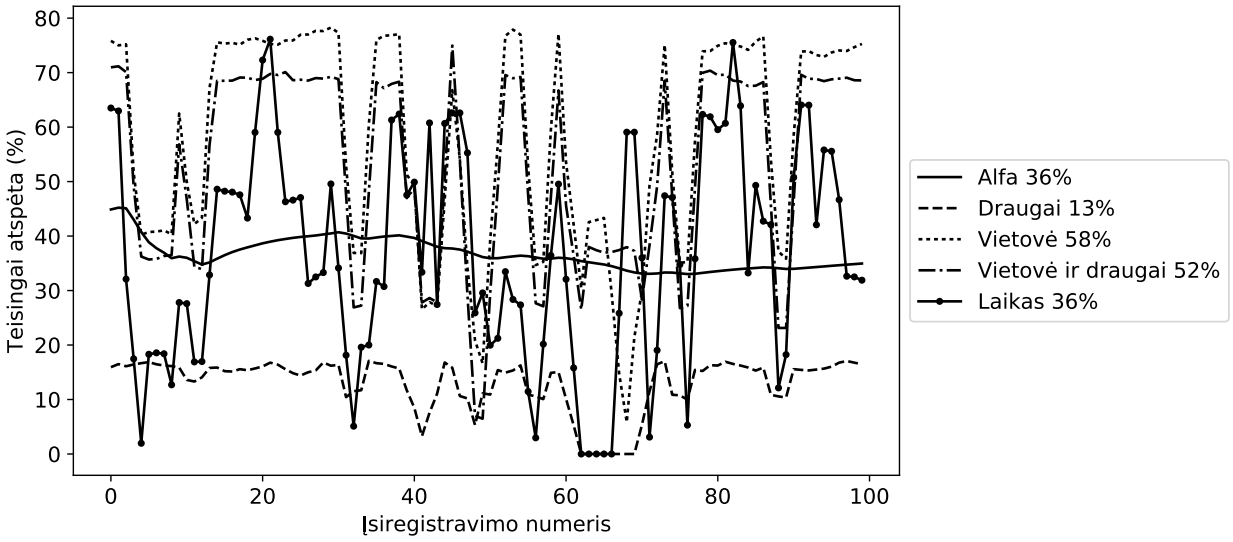
(a) $id = 753$ (b) $id = 3145$ (c) $id = 50405$

15 pav. Vietovės spėjimo pagal laiką rezultatai skirtingiems vartotojų tipams.

spėjamos, nes 98% vartotojo naujų įsiregistravimų buvo įvykdyti ne toliau kaip 3km nuo paskutinio įsiregistravimo. Dėl to kai įsiregistravimas įvykdavo viename iš klasterių, modelis sėkmingai nuspėdavo sekančio įsiregistravimo vietovę, kadangi dažniausiai ji būdavo tame pačiame klasteryje. Vartotojui $id = 3145$ atspėti pavyko 55% įsiregistravimų vietovių. 76% vartotojo $id = 3145$ įsiregistravimų buvo atlikti per 3km nuo prieš tai atlikto įsiregistravimo. Prasčiausiai nuspėti vietoves sekėsi vartotojo $id = 50405$ įsiregistravimams. Lyginant su anksčiau nagrinėtais vartotojais, šis keliavo toliausiai, nes tik 46% įsiregistravimų buvo atliekami per 3km nuo prieš tai atlikto įsiregistravimo.

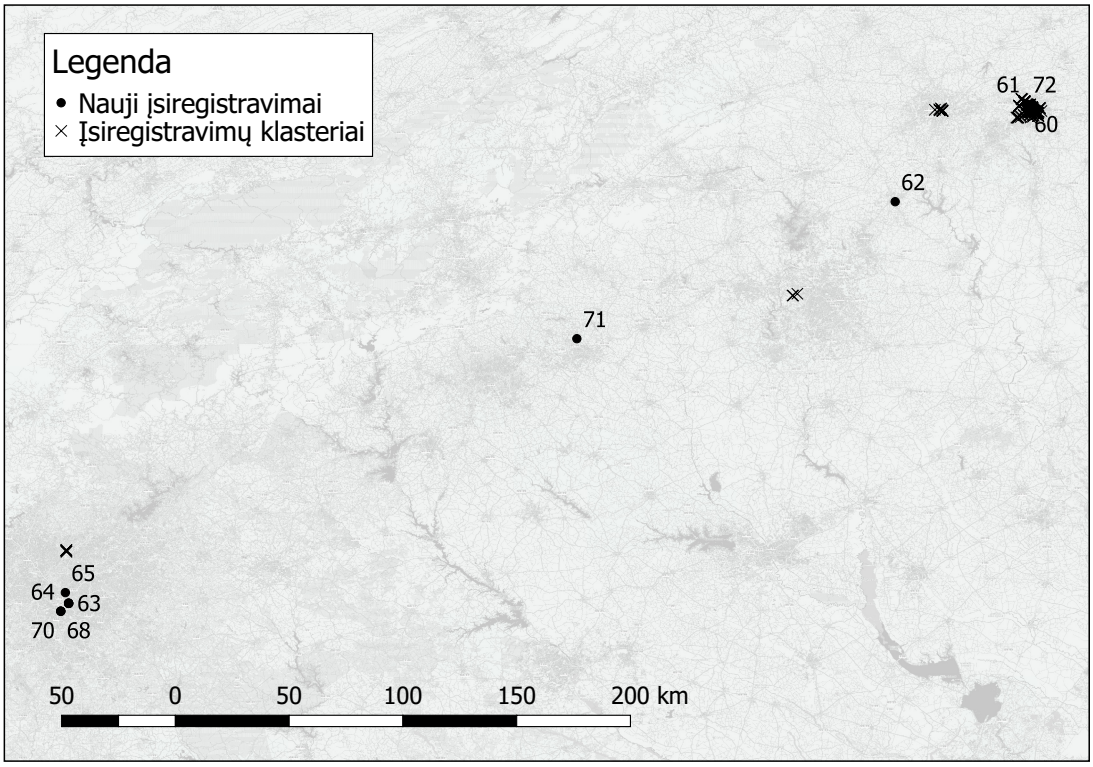
5.1.5. Metodų palyginimas

Atliekant vietovės spėjimus skirtingi metodai ne vienodai sėkmingai veikia skirtingiems vartotojams. Vartotojus išskyrus į 3 grupes, kiekvienai grupei galima įžvelgti tam tikras metodų veikimo tendencijas. Pavyzdžiui atliekant vietovės spėjimo pagal laiką modeliavimą, geriausi rezultatai pasiekiami vartotojams iš 2 grupės, nes jie atlieka įsiregistravimus mažiausiais laiko ir atstumo intervalais. Skirtingų metodų veikimas vartotojui iš 0 grupės vaizduojamas 13 pav.



16 pav. Vietovės spėjimo rezultatai naudojant skirtingus metodus.

Detaliau panagrinėsime sekančio įsiregistravimo vietovės spėjimų rezultatus, kai spėjimai atliekami vartotojui iš 2 grupės. Modeliavimo rezultatai vaizduojami 16 pav. Modelis veikė sėkmingiausiai, kai spėjimai buvo atliekami tik pagal vartotojo istorinius įsiregistravimus. Šiuo atveju atspėta 58% įsiregistravimų vietovių. Grafike galima pastebėti, kad beveik viso modeliavimo metu, šis metodas leido vidutiniškai atspėti apie 75% atvejų sėkmingiausio veikimo metu. Įsiregistravimai buvo vykdomi tame pačiame klasteryje kl_1, kai spėjimai buvo atliekami geriausiai. Tai pastebima nuo įsiregistravimo c_{14} iki c_{32}: $c_{14}, \ldots, c_{32} \in kl_1$. Prieš atliekant įsiregistravimą c_{14}, paskutinis įsiregistravimas buvo klasteryje kl_4. Remiantis istoriniais įsiregistravimais, tikimybė pereiti į klasterį kl_1 buvo $\mathbf{P}(c_{14} \in kl_1 | c_{13} \in kl_4) = 0,5$. Atitinkamai atlikus 1000 spėjimų šiam įsiregistravimui, modelis sėkmingai nuspėjo vietovę 476 kartus. Toliau vykę įsiregistravimai iki c_{32} buvo atliekami klasteryje kl_1. Kadangi atliekant spėjimus pagal vietovę, įvykus kiekvienam iš įsiregistravimų jis įtraukiamas prie istorinių duomenų, tai kiekvieną kartą kai atliekama įsiregistravimų tame pačiame klasteryje seka, tikimybė pasilikti tame pačiame klasteryje vis didėja. Šiuo atveju tikimybė padidėjo nuo $\mathbf{P}(c_{15} \in kl_1 | c_{14} \in kl_1) = 0,75$ iki $\mathbf{P}(c_{32} \in kl_1 | c_{31} \in kl_1) = 0,76$. Šiam vartotojui tikimybė per 18 įsiregistravimų pakilo tik 1 procentiniu punktu, nes tarp istorinių duomenų buvo daug įsiregistravimų, atliktų po apsilankymo klasteryje kl_1.



17 pav. Vartotojo $id = 15291$ nauji įsiregistravimai ir istorinių įsiregistravimų klasteriai.

Vietovės spėjimo tik pagal vartotojo istorinius duomenis metodas blogiausiai veikė, kai formavosi nauji klasteriai, arba kai buvo atliekami perėjimai tarp klasterių, kurių nebuvo aptikta tarp istorinių duomenų. 16 pav. nuo c_{60} iki c_{70} įsiregistravimo vidutinis teisingai atspėtų vietovių procentas nukrito iki 30%. c_{60} bei c_{61} įsiregistravimai vyko kl_1 klasteryje. Kadangi dauguma šio vartotojo įsiregistravimų atliekami šiame klasteryje, abiejų įsiregistravimų vietovės 75% atvejų buvo atspėjamos teisingai. Tačiau c_{61} įsiregistravimas buvo atliktas naujoje vietovėje, kuri nepatenka į nė vieną iš klasterių. Kadangi tarp istorinių duomenų iš visų įsiregistravimų vykusių po įsiregistravimo kl_1 klasteryje tik 4% buvo atliekami naujose vietovėse, modelis teisingai nuspėjo

tik $4,4\%$ įsiregistravimų vietovių. Sekantys įsiregistravimai $c_{62}, \ldots, c_{67}$ taip pat buvo atliekami nežinomose vietovėse. Tai galima pastebėti ir 17 pav. Įvykus c_{68} įsiregistravimui iš istorinių duomenų pavyko suformuoti naują įsiregistravimo klasterį kl_{11}. c_{69} įsiregistravimas buvo atliktas taip pat kl_{11} klasteryje, tačiau tarp istorinių duomenų nebuvo užfiksuotas nė vienas atvejis kad įsiregistravimas kl_{11} klasteryje būtų atliktas po kito įsiregistravimo tame pačiame klasteryje. Todėl šiuo atveju nepavyko nė karto teisingai atspėti įsiregistravimo vietovės. Tuo tarpu c_{70} taip pat įvyko kl_{11}, bet tarp istorinių duomenų jau buvo aptiktas atvejis, kad įsiregistravimas buvo atliekamas iš kl_{11} į kl_{11}. Todėl įsiregistravimui c_{70} vidutiniškai buvo atspėta $19,2\%$ vietovių.

Atliekant 16 pav. vaizduojamus vartotojo vietovės spėjimus tik pagal laiką, buvo parinkta didelė parametro h reikšmė. Dėl šios priežasties, modelis vietoves spėjo sėkmingiausiai, kai vartotojas įsiregistravimus atliko mažais laiko intervalais ir netoli klasterių centrų. Tačiau sekančio įsiregistravimo vietovės nebuvo atspėjamos, kai įsiregistravimai buvo atliekami nepažįstamose vietovėse, arba per didelį atstumą nuo paskutinio įsiregistravimo. Todėl nuo c_{62} iki c_{68} įsiregistravimų nebuvo atspėta nė viena vietovė. Po c_{68} įsiregistravimo susiformavo naujas klasteris kl_{11}. c_{69}, c_{70} įsiregistravimai buvo atliekami klasteryje kl_{11} apie 10h laiko intervalais. Dėl gana nedidelių laiko intervalų tarp įsiregistravimų ir dėl nedidelio atstumo nuo paskutinio įsiregistravimo iki klasterio kl_{11}, modeliuojant vietovės spėjimus įsiregistravimams c_{69} ir c_{70} vidutiniškai buvo atspėta $77,6\%$ ir $99,7\%$ vietovių. Sėkmingą modelio veikimą šiems įsiregistravimams taip pat lėmė dideli atstumai nuo jų iki kitų klasterių centrų, vidutinis atstumas 421km, tuo tarpu atstumai tarp c_{69}, c_{70} ir klasterio kl_{11} yra atitinkamai $0,2$km ir $2,7$km. Lyginant su kitais vietovės spėjimo būdais, vietovės spėjimas pagal laiką sėkmingiausiai identifikavo įsiregistravimų c_{69} ir c_{70} vietoves.

Šiam vartotojui blogiausiai veikė vietovės spėjimo metodas, kai atsižvelgiama tik į draugų įsiregistravimus. Vartotojas turi 17 draugų, kurie iš viso atliko 1107 įsiregistravimų. Jo draugai gana tolygiai atliko įsiregistravimus vartotojo klasteriuose. Viso modeliavimo metu tikimybės įsiregistruoti kuriame nors iš klasterių svyravo nuo 5% iki 15%. Dėl šios priežasties modeliui vidutiniškai pavyko atspėti 13% įsiregistravimų vietovių.

Antras geriausiai veikęs metodas buvo vietovės spėjimas pagal vartotojo ir jo draugų istorinius įsiregistravimus. Vartotojas 74% įsiregistravimų atliko kl_1 klasteryje. Sudarius perėjimo tikimybių matricą iš vartotojo bei jo draugų įsiregistravimų, tikimybė, kad sekantis įsiregistravimas bus kl_1 klasteryje, jei praeitas buvo atliktas tame pačiame klasteryje, beveik viso modeliavimo metu buvo $\mathbf{P}(c_i \in kl_1 | c_{i-1} \in kl_1) = 0,7$. Tai leido gana tiksliai atspėti daugumą vietovių, tačiau modelis taip pat nesugebėjo sėkmingai identifikuoti tų vietovių, kurios buvo atliekamos toliau nuo kl_1 klasterio.

Modeliuojant vietovės spėjimus pagal vartotojo ir jo draugų istorinius duomenis naudojant 5.1.3 skirsnyje aprašytą metodą buvo parinkta parametro reikšmė $\alpha = 0,5$. Šio metodo rezultatai viso modeliavimo metu buvo gana stabilūs. Svyravo tarp 35% – 45%.

5.2. Rekomendacijų teikimas

Teikiant rekomendacijas vartotojams svarbu ne tai ar modeliui pavyko sėkmingai atspėti sekančio įsiregistravimo vietovę, o sugebėti pasiūlyti vietovių rinkinį, kuriame vartotojas norėtų apsilankyti. Rekomendacijų teikimas šiame darbe buvo vykdomas kaip ir vietovės spėjimas. Iš pradžių iš istorinių duomenų sudaromi įsiregistravimų vietovių klasteriai. Suskaičiavus perėjimo tarp klasterių tikimybes, prieš vartotojui atliekant įsiregistravimus, kaip rekomendacijos jam pateikiama x vietovių. Įvykus įsiregistravimui patikrinama ar šis įsiregistravimas vyko bent vienoje iš vartotojui pasiūlytų vietovių. Jei įsiregistravimas vyko siūlytose vietovėse, laikoma, kad rekomendacija pateikta sėkmingai.

Vietovių klasteriai buvo sudaromi pasitelkiant Minimalios Kokybės Ribos algoritmą [13]. Sudarant įsiregistravimų klasterius buvo naudotos įvairios minimalaus klasterio dydžio parametro k bei maksimalaus atstumo iki klasterio centro r reikšmės. Sukonstravus įsiregistravimų klasterius, 3.3.2 skyrelyje aprašytu metodu sudaryta perėjimo tarp klasterių tikimybių matrica. Kadangi prieš vartotojui v atliekant naują įsiregistravimą yra žinomas paskutinio įsiregistravimo klasteris kl_i, remiantis perėjimo tikimybių matrica, gaunamas sekančio įsiregistravimo vietovės tikimybių vektorius

$$(\widehat{\mu}_{i0}, \widehat{\mu}_{i1}, \ldots, \widehat{\mu}_{in}). \tag{5.9}$$

Tuomet sudaromas sekančio įsiregistravimo vietovės tikimybių vektorius remiantis vartotojo v draugų įsiregistravimų duomenimis. 5.1.2 skyrelyje aprašytu metodu sudaromos tikimybės sekantį įsiregistravimą $c_{m+1} \in C_v$ vartotojui v atlikti klasteryje kl_i. Šiuo atveju sudarant tikimybes pagal vartotojo draugų įsiregistravimų duomenis, buvo imami tik dviejų geriausių vartotojo draugų įsiregistravimai. Čia geriausia draugai yra vartotojai v_0, v_1, kurių socialinis atstumas iki atitinkamo vartotojo v yra mažiausias, t.y. $v_0, v_1 \in \Gamma_v : d_{soc}(v, v_0) \leq d_{soc}(v, v_1) \leq d_{soc}(v, v_i), \forall v_i \in \Gamma_v, v_i \neq v_0, v_i \neq v_1$.

Sudarius tikimybių pagal draugų įsiregistravimus vektorių $(\widehat{c}_0, \widehat{c}_1, \ldots, \widehat{c}_n)$, šis vektorius sudedamas su perėjimo matricos tikimybių eilute ir gaunami kiekvieno iš klasterių pateikimo rekomendacijoms svoriai:

$$(rc_0, rc_1, \ldots, rc_n) = (\widehat{\mu}_{i0}, \widehat{\mu}_{i1}, \ldots, \widehat{\mu}_{in}) + (\widehat{c}_0, \widehat{c}_1, \ldots, \widehat{c}_n) \tag{5.10}$$

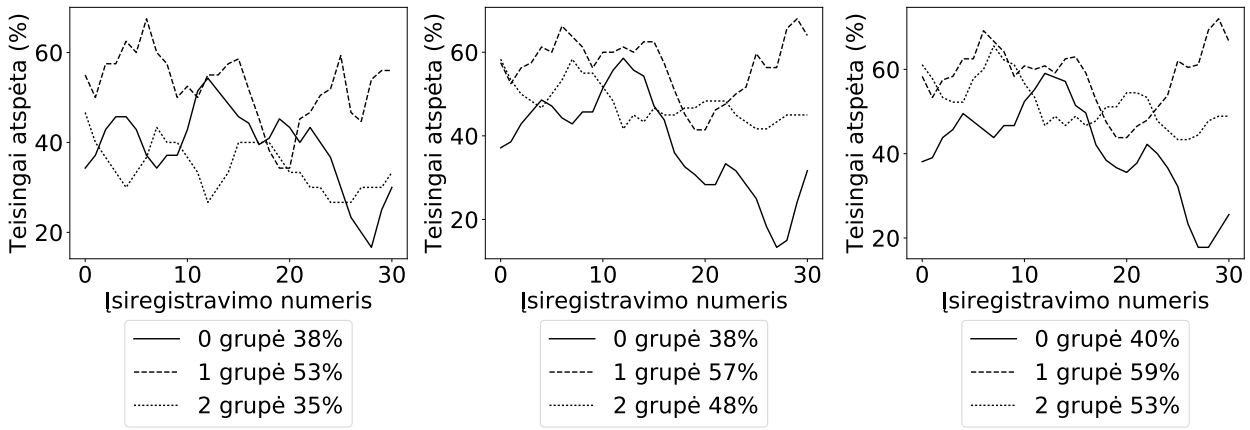
Prieš parenkant įsiregistravimų vietovių rekomendacijas, įvertinama, ar vartotojui įmanoma pasiekti kiekvieną iš įsiregistravimų klasterių, jei paskutinis jo įsiregistravimas buvo atliktas prieš $t_\Delta = t_{c_m} - t_{c_{m-1}}$ laiko. Remiantis 5.1.4 skyrelyje aprašytu metodu klasterio pateikimo rekomendacijai svoriui rc_i priskiriamas 0, kai per duotą laiko tarpą t_Δ vartotojas negali pasiekti klasterio kl_i:

$$rc_i = 0, \text{ kai } d_{max} < d_{geo}(l_{c_m}, kl_i).$$

Čia $d_{max} = t_\Delta g_{max}$, kur g_{max} yra prieš modeliavimą parenkamas maksimalaus greičio per valandą parametras. Teikiant rekomendacijas buvo parinkta $g_{max} = 200km/h$. Galiausiai parenkamos 2 didžiausius svorius turinčios vietovės ir jos vartotojui pateikiamos kaip rekomendacijos.

Atsitiktinai parinkus 21 vartotoją, turintį bent 30 įsiregistravimų jiems buvo atliekami rekomendacijų teikimai. Paveikslėliuose pateikiami agreguoti spėjimų rezultatai, suskaičiuojant sėkmingų spėjimų vidurkį pagal grupes, aprašytas 4.1 poskyryje.

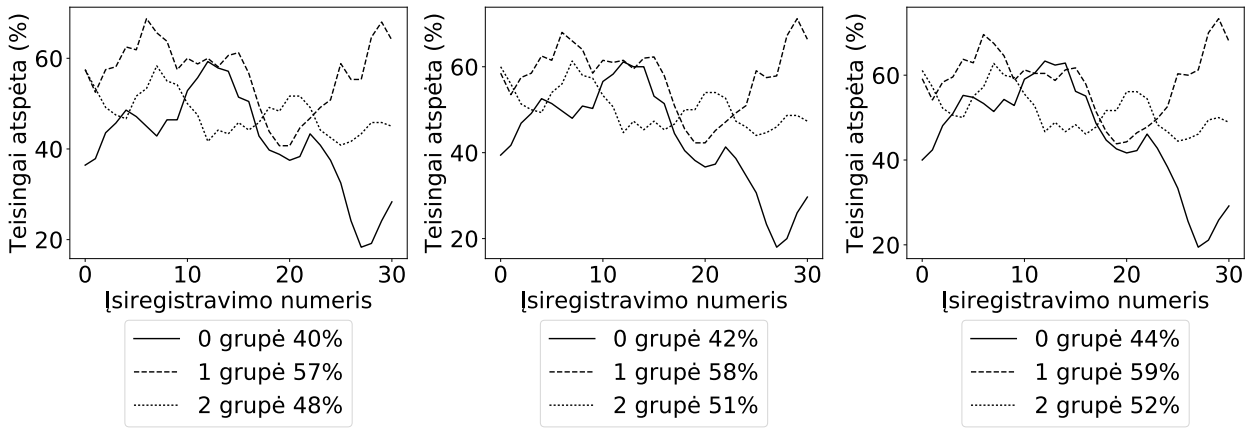
Rekomendacijų teikimo modeliavimo rezultatai su minimalaus klasterio dydžio parametru $k = 2$ pateikiami 18 paveikslėlyje. Mažiausiai sėkmingų rekomendacijų buvo pateikta tuo atveju, kai parametro r reikšmė buvo mažiausia. Maža parametro r reikšmė lemia tai, kad klasterio apimama teritorija bus mažesnė ir rekomendacijoje bus siūloma konkretesnė vietovė. Dėl šios priežasties didinant parametro r reikšmę visi rekomendacijų teikimo rezultatai buvo geresni. Mažiausią įtaką parametras r turėjo 0 grupės vartotojams. Iš visų vartotojų grupių, šiai grupei priklausantys



(a) $r = 0,5$ (b) $r = 3$ (c) $r = 6$

18 pav. Rekomendacijų teikimo rezultatai skirtingoms parametro r reikšmėms, naudojant parametrą $k = 2$.

vartotojai turėjo mažiausiai klasterių tarp istorinių duomenų ir buvo mažiausiai keliaujantys (žr. 4 lentelę). Todėl didinant klasterio spindulį rekomendacijų teikimo rezultatai keitėsi mažiausiai. Didžiausią įtaką rekomendacijų teikimo rezultatams parametras r turėjo 2 grupės vartotojams. Šios grupės vartotojai atliko daugiausiai įsiregistravimų, todėl jiems rekomendacijų teikimo rezultatai buvo jautriausi parametro r pokyčiams.

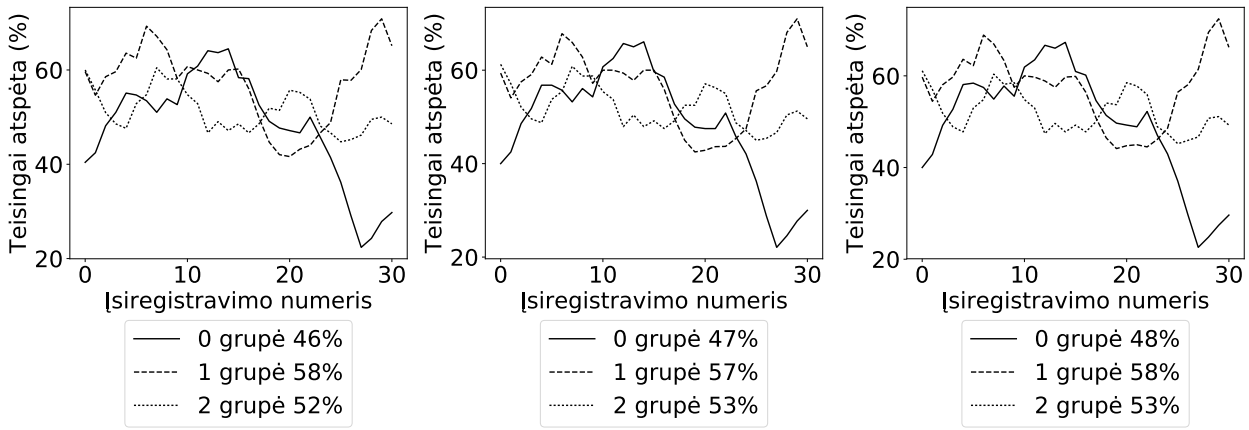


(a) $r = 0,5$ (b) $r = 3$ (c) $r = 6$

19 pav. Rekomendacijų teikimo rezultatai skirtingoms parametro r reikšmėms, naudojant parametrą $k = 3$.

Rekomendacijų teikimo modeliavimo rezultatai su minimalaus klasterio dydžio parametru $k = 3$ pateikiami 19 paveikslėlyje. Didesnė parametro k reikšmė nei 18 paveikslėlyje vaizduojamu atveju lėmė geresnius rekomendacijų teikimo rezultatus. Padidinus reikalaujamą minimalų klasterio dydį, gauti klasteriai tiksliau atspindėjo vartotojams pažįstamas vietoves. Vartotojams iš 0 grupės atliekant rekomendacijų teikimą, didesnė parametro k reikšmė lėmė ir didesnį jautrumą parametro r pokyčiams.

Rekomendacijų teikimo modeliavimo rezultatai su minimalaus klasterio dydžio parametru $k =$



(a) $r = 0,5$ (b) $r = 3$ (c) $r = 6$

20 pav. Rekomendacijų teikimo rezultatai skirtingoms parametro r reikšmėms, naudojant parametrą $k = 5$.

5 pateikiami 20 paveikslėlyje. Šiuo atveju gauti geriausi rekomendacijų teikimo rezultatai. Taip pat čia pasiektas mažiausias jautrumas parametro r svyravimams. Keičiant r reikšmes kiekvienai iš grupių gauti rezultatai svyravo per 1 procentinį punktą. Šiuo atveju pasiekti geriausi rezultatai su mažiausia r reikšme. Skirtingoms grupėms sėkmingai pateiktų rekomendacijų procentas svyravo nuo 46% iki 52%, kai klasterio spindulys buvo $r = 0,5$km.

Išvados ir rekomendacijos

Šiame darbe apibrėžiamos geografinio-socialinio konteksto užklausos, pristatomos funkcijos, reikalingos atlikti užklausoms bei pademonstruojami užklausų taikymo būdai. Pristatomi kitų autorių darbai, nagrinėjantys geografinio-socialinio konteksto užklausas ir jų taikymus. Geografinio-socialinio konteksto užklausų pagalba sukuriamas vietovės rekomendacijų teikimo modelis.

Struktūrinis ekvivalentumas yra tinkamas matas vertinti socialinius ryšius tarp vartotojų socialiniame tinkle. Dėl iš anksto parenkamų reikalaujamų klasterio kokybės parametrų, Minimalios Kokybės Ribos algoritmas yra tinkamas klasterizavimo algoritmas vartotojo pažįstamų vietovių identifikavimui. Taip pat šiame darbe parodoma, kad Markovo grandinių modelis gali būti naudojamas vartotojo įsiregistravimų vietovių spėjimams ir vietovės rekomendacijų teikimui.

Atlikus turimų duomenų analizę buvo pastebėta, kad vartotojai vidutiniškai per metus atlieka po 40 įsiregistravimų. Tokio įsiregistravimų kiekio pakanka, kad juos būtų galima panaudoti užklausų vykdymui. Pagal vartotojų ir jų draugų įsiregistravimų charakteristikas, tarp turimų duomenų pavyko išskirti 3 vartotojų grupes.

Šiame darbe pristatomi ir įvertinami sekančios vartotojo įsiregistravimo vietovės spėjimo algoritmai. Spėjimus atliekant pagal vartotojo vietovę atsitiktinai parinktiems vartotojams, teisingai nustatyta iki 80% vietovių. Spėjimus atliekant kitais metodais gauti mažesni teisingai atspėtų vietovių procentai, tačiau teisingai atspėtos tos vietovės, kurių nepavyko atspėti tik pagal vartotojo istorinius įsiregistravimų duomenis. Tai rodo, kad pagrindinis sekančios įsiregistravimo vietovės spėjimo parametras yra istorinių įsiregistravimų vietovės duomenys. Įsiregistravimų laikas ir vartotojo draugų įsiregistravimai leidžia atspėti tas vietoves, kurių atspėti vien pagal istorinius įsiregistravimų vietovių duomenis negalima. Taip pat skirtingiems algoritmams spėjimo rezultatai priklauso nuo vartotojo įsiregistravimų tendencijų. Kuo vienas po kito einantys vartotojo įsiregistravimai atliekami didesniais atstumais, tuo sunkiau teisingai nustatyti sekančio įsiregistravimo vietovę pagal istorinius įsiregistravimų duomenis. Taip pat, kuo dažniau vartotojas atlieka įsiregistravimus, tuo tiksliau galima nustatyti sekančią vartotojo vietovę pagal įsiregistravimų laiką.

Vietovės rekomendacijų teikimo rezultatai rodo, kad skirtingoms vartotojų grupėms, modelio parametrų pokyčiai ne vienodai paveikia sėkmingų rekomendacijų procentą. Taip pat šiame darbe parodoma, kad rekomendacijų teikimo rezultatai labiausiai kinta, kai minimalaus klasterio dydis parenkamas $k = 2$.

Ateities tyrimų planas

Šiame darbe sukuriamas vietovės rekomendacijų teikimo modelis gali būti patobulintas daugiau atsižvelgiant į vartotojų įsiregistravimo laiką. Kadangi naudotame duomenų rinkinyje nebuvo tikslaus aprašymo kokių laiko juostų atžvilgiu buvo matuojamas įsiregistravimų laikas, čia nebuvo atsižvelgta į tai, kuriuo paros metu atliekami įsiregistravimai. Turint šią informaciją galima atlikti tikslesnius vietovės spėjimus.

Egzistuoja kitų perėjimo tarp klasterių tikimybių skaičiavimo metodų, kurie galėtų būti išbandyti siekiant gauti tikslesnius rekomendacijų teikimo rezultatus.

Žinant charakteristikas vietovės, kurioje atliekami įsiregistravimai, galima pateikti rekomendacijas remiantis vartotojo pomėgiais. Turint duomenų rinkinį, kuriame atsispindėtų vietovių charakteristikos, galima patobulinti turimą rekomendacijų teikimo modelį.

Literatūros šaltiniai

[1] Gowalla dataset. https://snap.stanford.edu/data/loc-gowalla.html, (žiūrėta 2017-12-17).

[2] Nikos Armenatzoglou, Stavros Papadopoulos, and Dimitris Papadias. A general framework for geo-social query processing. *PVLDB*, 6(10):913–924, 2013.

[3] Jie Bao, Yu Zheng, and Mohamed F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *SIGSPATIAL 2012 International Conference on Advances in Geographic Information Systems (formerly known as GIS), SIGSPATIAL'12, Redondo Beach, CA, USA, November 7-9, 2012*, pages 199–208, 2012.

[4] Mindaugas Bloznelis. *Kominatorikos ir grafų teorijos paskaitos*. VU Leidykla, Vilnius, 2016.

[5] Zaiben Chen, Heng Tao Shen, and Xiaofang Zhou. Discovering popular routes from trajectories. In *Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11-16, 2011, Hannover, Germany*, pages 900–911, 2011.

[6] Hong Cheng, Jihang Ye, and Zhe Zhu. What's your next move: User activity prediction in location-based social networks. In *Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA.*, pages 171–179, 2013.

[7] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1082–1090, 2011.

[8] Chi-Yin Chow, Jie Bao, and Mohamed F. Mokbel. Towards location-based social networking services. In *Proceedings of the 2010 International Workshop on Location Based Social Networks, LBSN 2010, November 2, 2010, San Jose, CA, USA, Proceedings*, pages 31–38, 2010.

[9] Tobias Emrich, Maximilian Franzke, Nikos Mamoulis, Matthias Renz, and Andreas Züfle. Geo-social skyline queries. In *Database Systems for Advanced Applications - 19th International Conference, DASFAA 2014, Bali, Indonesia, April 21-24, 2014. Proceedings, Part II*, pages 77–91, 2014.

[10] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 15-19, 1999, Berkeley, CA, USA*, pages 230–237, 1999.

[11] Laurie J Heyer, Semyon Kruglyak, and Shibu Yooseph. Exploring Expression Data : Identification and Analysis of Coexpressed Genes Exploring Expression Data : Identification and Analysis of Coexpressed Genes. (213):1106–1115, 1999.

[12] Bo Hu, Mohsen Jamali, and Martin Ester. Spatio-temporal topic modeling in mobile social media for location recommendation. In *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, pages 1073–1078, 2013.

[13] Xin Jin and Jiawei Han. *Quality Threshold Clustering*, pages 820–820. Springer US, Boston, MA, 2010.

[14] Jonas Kubilius. *Tikimybių teorija ir matematinė statistika*. Vilniaus universiteto leidykla, 1996.

[15] Po-Ruey Lei, Tsu-Jou Shen, Wen-Chih Peng, and Ing-Jiunn Su. Exploring spatial-temporal trajectory model for location prediction. In *12th IEEE International Conference on Mobile Data Management, MDM 2011, Luleå, Sweden, June 6-9, 2011, Volume 1*, pages 58–67, 2011.

[16] Elizabeth A Leicht, Petter Holme, and Mark EJ Newman. Vertex similarity in networks. *Physical Review E*, 73(2):026120, 2006.

[17] Jeffrey McGee, James Caverlee, and Zhiyuan Cheng. Location prediction in social media based on tie strength. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 459–468, 2013.

[18] Justin J Miller. Graph database applications and concepts with neo4j. In *SAIS 2013 Proceedings*, 2013.

[19] Anastasios Noulas, Salvatore Scellato, Neal Lathia, and Cecilia Mascolo. Mining user mobility features for next place prediction in location-based services. In *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, pages 1038–1043, 2012.

[20] Kevin Wilfong Pamela Vagata. Scaling the facebook data warehouse to 300 pb, 2014 (žiūrėta 2017-12-17).

[21] Moon-Hee Park, Jin-Hyuk Hong, and Sung-Bae Cho. Location-based recommendation system using bayesian user's preference model in mobile devices. In *Ubiquitous Intelligence and Computing, 4th International Conference, UIC 2007, Hong Kong, China, July 11-13, 2007, Proceedings*, pages 1130–1139, 2007.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[24] Notification Services. Microsoft SQL Server. 2017 (žiūrėta 2017-12-17).

[25] Jieming Shi, Nikos Mamoulis, Dingming Wu, and David W. Cheung. Density-based place clustering in geo-social networks. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, pages 99–110, 2014.

[26] Thaddeus Vincenty. Determination of North American Datum 1983 Coordinates of Map Corners. Technical report, National Oceanic and Atmospheric Administration, Maryland, 1976.

[27] Yingzi Wang, Nicholas Jing Yuan, Defu Lian, Linli Xu, Xing Xie, Enhong Chen, and Yong Rui. Regularity and conformity: Location prediction using heterogeneous mobility data. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pages 1275–1284, 2015.

[28] Wenjian Xu, Chi-Yin Chow, Man Lung Yiu, Qing Li, and Chung Keung Poon. Mobifeed: A location-aware news feed framework for moving users. *GeoInformatica*, 19(3):633–669, 2015.

[29] Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik Lun Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011*, pages 325–334, 2011.

[30] Jia-Dong Zhang, Gabriel Ghinita, and Chi-Yin Chow. Di erentially private location recommendations in geosocial networks. In *IEEE 15th International Conference on Mobile Data Management, MDM 2014, Brisbane, Australia, July 14-18, 2014 - Volume 1*, pages 59–68, 2014.