Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Karmen Knavs

Analiza vzorcev obiskovanja turističnih destinacij na podlagi javno dostopnih podatkov

magistrsko delo

magistrski program druge stopnje
računalništvo in informatika

Mentor: izr. prof. dr. Damjan Vavpotič
Somentor: izr. prof. dr. Ljubica Knežević Cvelbar

Ljubljana, 2018

Zahvala

Iskreno se zahvaljujem svojemu mentorju izr. prof. dr. Damjanu Vavpotiču in somentorici izr. prof. dr. Ljubici Knežević Cvelbar za vso pomoč pri izdelavi naloge in čas, ki sta mi ga posvetila. Za podatke se zahvaljujem študentom, ki so raziskovalno delovali v okviru Laboratorija za informatiko na področju luščenja podatkov s spleta.

Posebna zahvala gre mojim staršem, sestrama in fantu, ki mi vedno stojijo ob strani. Hvala tudi vsem, ki so mi nudili moralno in ostalo pomoč pri študiju, zlasti Aniti, Jani in Biljani.

Karmen Knavs, 2018

Moji družini in Luku.

"Živi, kot da je jutri tvoj zadnji dan, uči se, kot da boš živel večno."

— Gandhi

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
MBA	market basket analysis	analiza nakupovalne košarice
TID	transaction identifier	identifikator transakcije
CPM	clique percolation method	razširjanje po klikah
COPRA	community overlap propagation algorithm	algoritem razširjanja prekrivajočih skupnosti
NMI	normalized mutual information	normalizirana medsebojna informacija
VI	variation of information/shared information distance	variacija informacije
ORM	object-relational mapping	objektno-relacijsko preslikovanje
SURS	statistical office of the Republic of Slovenia	statistični urad Republike Slovenije

Povzetek

Naslov: Analiza vzorcev obiskovanja turističnih destinacij na podlagi javno dostopnih podatkov

V okviru magistrskega dela je bil izdelan pristop za analizo obiskovanja turističnih destinacij z algoritmi na grafih na podlagi javno dostopnih podatkov. Namen analize je najti skupine destinacij, ki so pogosto obiskane skupaj in so lahko osnova za skupno trženje destinacij. Javno dostopni podatki so lahko objave turistov povezane z obiskom turistične destinacije z različnih družbenih omrežij ali iz drugih virov. Implementirana rešitev uporablja podatke pridobljene iz javnih objav turistov na spletnem mestu TripAdvisor in iz njih generira graf sočasnih pojavitev destinacij. Utež na povezavi v grafu predstavlja seštevek potovanj, v katerih sta bili obiskani obe destinaciji. Za analizo pogosto obiskanih destinacij smo temeljili na metodah, ki se uporabljajo za analizo nakupovalne košarice (MBA). Preizkušena je bila uporaba algoritma Apriori in dveh algoritmov za odkrivanje skupnosti na grafu: Louvain in Infomap. Rezultati kažejo, da Infomap vrača najbolj razumljive in zanimive pogoste skupine destinacij. Filtriranje omogoča segmentacijo gostov na podlagi njihovega profila. Na testnih podatkih so bile najbolj raznolike skupnosti najdene s segmentacijo uporabnikov po državi.

Ključne besede

teorija grafov, skupine vozlišč, skupnosti, turistične destinacije, destinacijski menedžment

Abstract

Title: Analyzing patterns of visiting tourist destinations based on publicly available data

In thesis an approach for analyzing of visiting tourist destinations with graph algorithms on publicly available data was built. The goal of analysis is to find groups of destinations frequently visited together that can be a base for a joint marketing by destinations. Publicly available data can be tourists' posts related to connected to visit of tourist destination from different social networks or other sources. The implemented solution uses data gathered from public tourists' posts on TripAdvisor as input to generate co-occurrence graph of destinations. Weight of edge represents sum of trips, where both destinations were visited. For analysis of destinations frequently visited together we build on the methods, that are used for market basket analysis (MBA). We used Apriori algorithm and two algorithms for community detection: Louvain and Infomap. The most useful results were obtained by using Infomap. Data filtering enables us to observe different segments of guests based on their profile. The largest difference between communities was detected when using segmentation based on guests' country of origin.

Keywords

graph theory, node groups, communities, tourist destinations, destination management

Poglavje 1

Uvod

V Sloveniji iz leta v leto narašča obisk tako domačih kot tujih turistov. Veča se tako število prihodov[1] kot nočitev turistov, a povprečno število nočitev na turista se zmanjšuje: leta 2017 je bilo zabeleženih skoraj pet milijonov prihodov, kar je 14,6% več kot leta 2016, ter več kot 12 in pol milijonov prenočitev, kar predstavlja 12,6% rast glede na prejšnje leto [2]. Delež prenočitev tujih turistov v skupnem številu prenočitev turistov se stalno povečuje od leta 2009 - takrat so tuji turisti ustvarili 55 % vseh turističnih prenočitev, v letu 2013 62 %, v letu 2017 pa že 68 % [3]. Velika rast se odraža tudi v gospodarskih kazalnikih kot je prispevek turizma k bruto domačem proizvodu in številu zaposlenih prebivalcev v njem.

Razvoj informacijskih tehnologij in mobilnih naprav omogoča obiskovalcem, da lahko enostavno delijo svojo izkušnjo s trenutno obiskano destinacijo. Objavljene izkušnje so pogosto tudi javno dosegljive na spletu in so zato dober vir podatkov za analizo obiskovanj in potovanj. Družbena omrežja imajo pomembno vlogo v turistični dejavnosti, saj se na njihovi podlagi turisti odločajo. Na ta način lahko tudi usmerjamo turiste. Z analizo podatkov z družbenih omrežij oz. turističnih spletnih mest lahko ugotovimo, kakšen je potencial atrakcij glede na zanimanja obiskovalcev in njihove potovalne

[1]Prihod turista je vpis turista v knjigo gostov ob njegovem prihodu v turistični nastanitveni objekt (hotel, kamp itd.). [1]

vzorce [4].

Turistično doživetje je pogosto sestavljeno iz obiska več destinacij. V delu se osredotočamo na turistično izkušnjo in analiziramo, katere destinacije sestavljajo skupno turistično doživetje ter kako se te razlikujejo glede na značilnosti obiskovalcev in časovno obdobje. Na podlagi analize so lahko ponujene združene trženjske kampanje in novi turistični produkti ali doživetja [5]. Urejena je lahko na primer pohodniška pot ali kolesarska povezava med dvema priljubljenima destinacijama ljubiteljev pohodništva ali kolesarstva. Turiste lahko segmentiramo po različnih ključnih motivih - kot so na primer (glede na [6]): druženje, dogajanje in mir ter kombinacije med njimi - potreben bi bil razvoj produktov glede na motive in z vidika, kje so potenciali za povezovanje destinacij oziroma ponudnikov, da turisti ne bi obiskovali le najbolj tipičnih in množično obiskanih turističnih točk [6]. Turistična doživetja prilagojena pričakovanjem specifičnih skupin turistov omogočajo izboljšanje turistične izkušnje. Naš cilj je bil pripraviti pristop za analizo in na njem temelječo programsko rešitev, ki sta nam omogočila identificirati in analizirati vzorce obiskovanja turističnih destinacij na podlagi javno dostopnih podatkov.

Pri pripravi pristopa za analizo obiskovanja turističnih destinacij smo se naslonili na teorijo grafov. Teorija grafov se pogosto uporablja pri analizi nakupovalne košarice, na področju analize podatkov za določanje skupin obiskanih destinacij pa dela, ki bi uporabljalo takšen pristop nismo zasledili. V okviru magisterskega dela smo pripravili inovativen pristop in programsko rešitev, ki omogoča gradnjo grafa iz transakcijskih zapisov in uporabili obstoječe algoritme iz teorije grafov, s katerimi lahko na podlagi mreže izdelkov oziroma destinacij določimo skupnosti. Preučili in primerjali smo rezultate pridobljene z uporabo različnih algoritmov, s pomočjo katerih smo analizirali podatke pridobljene na podlagi javnih objav obiskovalcev spletnega turističnega mesta TripAdvisor. V okviru analize smo se omejili na obiske v Sloveniji in bližnji okolici (tudi na obiske v sosednjih državah, ki so relativno blizu meje).

Z analizo je bilo omogočeno boljše razumevanje značilnosti in motivov obiskovalcev turističnih destinacij. Rezultati analize so tako lahko pomembna podlaga za sprejemanje odločitev, ki vodijo k povečanju privlačnosti in konkurenčnosti destinacij.

Delo obsega 6 poglavij. V Poglavju 2 podamo pregled sorodnih del in rezultatov analiz na področju turizma. Podrobneje predstavimo metode, ki jih uporabljamo za analizo v našem delu. V Poglavju 3 opišemo naš pristop za analizo na grafu, implementacijo, uporabljena orodja in vključene knjižnice. Sledi Poglavje 4 s statistiko in opisom podatkov, nato Poglavje 5 z njihovo analizo. Zaključimo s sklepnimi ugotovitvami v Poglavju 6.

Poglavje 2

Pregled sorodnih del in relevantnih metod

Destinacijski menedžment obsega funkcije managementa: načrtovanje, organizacija, vodenje in nadzorovanje [7]. V zadnjih desetih letih pa vse bolj pogosto v literaturi in praksi menedžmentu dodajamo tudi funkcije trženja destinacije. Velik izziv destinacijskega menedžmenta predstavlja spremljanje vedenja turistov. Destinacijski menedžerji morajo poznati podrobnosti destinacij, ki jih turisti obiskujejo, kaj turiste na vsaki lokaciji privlači, kakšno je njihovo mnenje o turistični izkušnji in nadaljnje vzorce obiskovanja. Za trženje turistične destinacije je poznavanje obiskanih destinacij izjemnega pomena, saj omogoča analizo in identifikacijo povezanih destinacij za pripravo skupnih trženjskih kampanj [5]. Za potrebe analiz so najprej zbirali podatke s pomočjo anketiranja, ki je časovno potratno. Razvoj družbenih omrežij je poenostavil pridobivanje podatkov: prostovoljno deljenje osebnih informacij in nalaganje vsebin na razna družbena omrežja sta ustvarila priložnost za nove oblike analize z upoštevanjem mnenja večjega števila turistov. Zlasti na področju destinacijskega menedžmenta primanjkuje analiz, ki bi temeljile na podatkih dostopnih preko družbenih omrežij, čeprav ti veljajo za uporabne in zanesljive [8]. V nadaljevanju predstavljamo primere analiz na področju turizma, ki so uporabljale podobne pristope in njihove ključne ugotovitve.

Štirje popularni templji v Hong Kongu so bili določeni samodejno z uporabo P-DBSCAN gručenja na fotografijah označenih z lokacijami pridobljenih s spletnega mesta in storitve za deljenje fotografij Flicker [4]. Analiza je pokazala, da imajo turisti iz različnih držav različen izbor. V Avstriji so na podlagi lokacijsko označenih slik s Flickra ugotovili, da turisti pogosto na potovanju obiskujejo kraje, ki so si geografsko blizu [5]. Na podlagi slik s Flickra (iz podatkovne baze, ki jo je objavil Yahoo [9]) so raziskovali tudi tok turistov preko držav [10]. Na podlagi ankete na Siciliji so z uporabo grafičnih modelov in logističnim regresijskim modelom ugotovili, da so glavne determinante potovanja celotno trajanje potovanja, število prejšnjih obiskov in motivacija [11]. V Italiji so s pomočjo objav s Twitterja pridobili podrobne prostorske in demografske podatke o premikanju turistov in s tem izboljšali razumevanje ključnih turističnih tokov [12].

Čeprav je identifikacija destinacij, za katere bi bilo smiselno pripraviti skupne kampanje, pogost problem destinacijskih menedžerjev [13], pa nismo zasledili raziskav, ki bi bili clijno usmerjeni v reševanje tega problema (blizu je delo [5], v katerem analizirajo potovanja iz več destinacij v Avstriji s poudarkom na kategorizacijo potovanj po dolžini trajanja in preslikavah med destinacijami). Da bi lahko pripravili združene trženjske kampanje je potrebno identificirati destinacije, ki jih turisti pogosto obiščejo skupaj, to je v okviru enega potovanja. V našem delu za rešitev tega izziva predlagamo uporabo metod, ki se sicer uporabljajo za analizo nakupovalne košarice. Iskanje skupin obiskanih destinacij lahko prevedemo na iskanje skupin nakupljenih izdelkov v trgovini, pri čemer namesto izdelkov v košarici upoštevamo destinacije med obiski katerih je dovolj majhen časovni razmik. Najpogostejši pristopi pri iskanju skupin izdelkov, ki so največkrat kupljeni (v našem primeru obiskani) skupaj, so analiza nakupovalne košarice z uporabo povezovalnih pravil, odkrivanje pogostih n-teric (angl. frequent item set discovery) in tehnike gručenja (kot so K-means, SOM), vendar z njimi na velikih množicah podatkov pogosto ne dobimo najbolj smiselnih in uporabnih rezultatov [14]. S povezovalnimi pravili lahko dobimo nesmiselne povezave med izdelki [15].

Z uporabo mreže izdelkov oziroma grafa, kjer vozlišča predstavljajo izdelke in vezi povezave med njimi (povezave so odvisne od razmerja med vozlišči in vrste grafa, lahko so usmerjene ali ne, z utežmi ali brez) in načini za odkrivanje skupnosti ali komun (to so skupine tesno povezanih vozlišč, ki so med seboj dobro ločene) tipično dobimo rezultate, ki so analitikom bolj razumljivi. Z uporabo omenjene metode dobimo uporabne rezultate tudi na novih vhodnih podatkih, saj je po prvem generiranju grafa njegovo posodabljanje enostavno in se nastavitev filtriranja lahko spreminja samodejno, postopek pa omogoča tudi razmeroma hitro procesiranje podatkov [14, 15, 16]. Na grafih, ki so generirani iz realnih podatkov, so skupnosti pogosto prekrivajoče. Za analizo prekrivajočih skupnosti se lahko uporabljajo algoritmi kot na primer: COPRA in GANXIS (včasih SLPA), ki sta oba osnovana na algoritmu za izmenjavo oznak (angl. label propagation algorithm) [14], CESNA, ki skupnosti identificira na podlagi strukture povezav in atributov vozlišč [17] in druge metode kot npr. genetski algoritmi [18]. V nekaterih primerih se kot smiselna izkaže tudi kombinacija različnih metod [15].

2.1 Povezovalna pravila in algoritem Apriori

Zelo pogosto uporabljen pristop za iskanje skupnosti so povezovalna pravila, ki jih lahko generiramo na različne načine. Eden izmed najpopularnejših načinov je algoritem Apriori, ki sta ga leta 1994 razvila Agrawal in Srikant in za vhod predvideva množice transakcij, ki predstavljajo dejanske košarice kupcev [19]. Z njim odkrivamo pogoste n-terice oziroma skupine izdelkov, na podlagi njih pa nato še povezovalna pravila. V nadaljevanju podrobneje predstavljamo algoritem.

Naj bo $I = i_1, i_2, ...i_n$ množica izdelkov i in $D = T_i, T_2, ...Tn$ množica transakcij T. Za vsako transakcijo T velja, da ima unikatni identifikator TID in da vsebuje X, ki je množica nekaj izdelkov iz množice vseh izdelkov I [19]. Vsako povezovalno pravilo je iz dveh komponent: prva komponenta ČE je poznana kot predhodnik, druga komponenta POTEM je znana kot posledica

[20]. Pravilo $X \implies Y$ velja v množici transakcij D (pri čemer je njun presek prazna množica: $X \cap Y = \emptyset$) z določenim zaupanjem in podporo:

- Z zaupanjem (angl. confidence) c : c je delež transakcij v D za katere velja, da če vsebujejo X, potem vsebujejo tudi Y. Zaupanje je pogojna verjetnost, da se bo posledica pojavila, če se bo pojavil predhodnik.

- S podporo (angl. support) s: s je delež transakcij v D za katere velja, da vsebujejo X in Y hkrati: $X \cap Y$. Podpora pravila pokaže, kako pogosto se izdelki v pravilu pojavijo skupaj in jo lahko izrazimo kot verjetnost: $X \implies Y = P(X \cap Y)$.

Generiranje povezovalnih pravil je odvisno od nastavitve minimalnega zaupanja in podpore. Razdelimo ga na dva podproblema:

1. Iskanje vseh množic izdelkov, ki imajo minimalno podporo, t.i. pogostih (tudi velikih) množic. Te kombinacije imenujemo tudi frekvenčna množica atributov [20].

2. Uporabo pogostih množic za generiranje pravil.

Da pravilo velja, mora imeti tako zaupanje kot podporo višje od minimalnih določenih vrednosti, ki smo jih nastavili. Povezovalna pravila nam lahko vrnejo zelo veliko število odkritih pravil. Vračajo tudi pravila, ki za človeka niso zanimiva oziroma uporabna.

Algoritem Apriori deluje z večkratnimi iteracijami preko podatkov. Najprej prešteje podporo posameznim izdelkom (angl. items), ter določi, kateri so pogosti. V vsaki nadaljnji iteraciji uporabi ugotovljene pogoste skupine iz prejšnje iteracije za generiranje potencialnih novih - množice kandidatov. Med prehodom čez podatke sešteva dejansko podporo za vsako množico kandidatov. Na koncu koraka določi, katere množice kandidatov so pogoste in te uporabi v naslednji iteraciji za generiranje novih množic kandidatov, dokler ne more najti nobenih novih pogostih množic več.

V primerjavi z ostalimi algoritmi za odkrivanje povezovalnih pravil je razmeroma hiter, saj *apriori* sklepa, katere kombinacije ne morejo imeti nadaljnje minimalne podpore - množico kandidatov s k-izdelki generiramo s korakom združevanja in korakom izločevanja [19]. Najprej združimo pogoste množice, ki imajo $k-1$ izdelkov. Iz teh zgeneriranih množic nato pobrišemo tiste, ki vsebujejo kakšno ne pogosto podmnožico. Apriori podpira tudi več kot en izdelek v posledici oziroma komponenti POTEM. Če imamo pravilo $X \implies A$, ni nujno da velja tudi $X + Y \implies A$, saj lahko ni dosežen minimalen prag za podporo (mora biti višja od vrednosti, ki smo jo nastavili). Podobno, če velja $X \implies Y$ in $Y \implies Z$ ni nujno, da velja tudi $X \implies Z$. Pravilo velja, če je razmerje med podporo predhodnika in podporo posledice $c = s(predhodnik)/s(podpora)$ višje od minimalne določene vrednosti za zaupanje.

2.2 Uporaba grafa za analizo podatkov

S pomočjo grafa lahko odkrivamo zanimive in koristne povezave v podatkih. Podatkovno rudarjenje je proces avtomatskega odkrivanja uporabnih informacij v zbirkah velikih količin podatkov [21]. S tehnikami podatkovnega rudarjenja iščemo nove in uporabne vzorce, ki bi drugače lahko ostali neznani. Pomagajo nam lahko tudi pri napovedih, kot je na primer, koliko bo nova stranka zapravila ali kaj bo še kupila na podlagi že kupljenih izdelkov. Podatkovno rudarjenje je sestavni del odkrivanja znanja v podatkovnih bazah, je proces pretvorbe surovih podatkov v uporabne informacije. Vključuje tako predprocesiranje vhodnih podatkov kot končno obdelavo rezultatov. Vhodni podatki so lahko shranjeni centralizirano ali distribuirano, v več formatih in (lahko) vključujejo šum in atribute, ki so nepomembni za rudarjenje. Uporabnik lahko v naraščajočih družbenih omrežjih deli informacije v več formatih: kot tekstovna objava, digitalna fotografija, videoposnetek; v objavah lahko deli tudi svojo lokacijo, napravo, poveže objavo z drugim družbenim omrežjem (na primer uporaba Twitter oznake teme v objavi na Facebooku),

označi kaj mu je všeč (Facebook), se prijavi na sledenje novicam določene osebe itd. Po obdelavi rezultatov, lahko te vgradimo v odločitvene sisteme ali vizualiziramo za potrebe analize. Ni dovolj, da rezultate le prikažemo, potrebna je tudi ocenitev in interpretacija rezultatov. Pogosto je potrebno tudi filtriranje in razlaga eksperta.

Naloge podatkovnega rudarjenja delimo v dve kategoriji [21]:

- Naloge napovedovanja: cilj teh nalog je napoved vrednosti določenega atributa (odvisne spremenljivke) glede na druge atribute (neodvisne spremenljivke).

- Naloge opisovanja: cilj je izpeljava vzorcev (korelacije, trendi, skupnosti, poti, anomalije), ki povzemajo temeljna razmerja podatkov. Opisne naloge so pogosto raziskovalne po naravi in zahtevajo obdelavo dobljenih rezultatov, npr. z ocenjevanjem in razlago.

Metode za analizo nakupovalne košarice lahko uporabimo tudi za analizo vzorcev obiskovanja turističnih destinacij. Medtem ko v "klasični"analizi nakupovalne košarice iščemo skupine izdelkov, ki so bili pogosto kupljeni skupaj, v analizi vzorcev obiskovanja iščemo skupine destinacij, ki so bile pogosto obiskane skupaj (in najverjetneje bodo tudi vnaprej). Skupine določimo na podlagi transakcij vseh strank oziroma na podlagi vseh znanih zabeleženih obiskov. Graf je sestavljen iz vozlišč, ki predstavljajo izdelke in povezave, ki lahko pomenijo različne relacije in so odvisne od tipa grafa [15]. Različne podatke vezane na vzorce nakupovanja predstavimo z različnimi tipi grafov:

- Graf sočasnih pojavitev (angl. co-occurrence graph): ima neusmerjene povezave z utežmi, ki določajo število nakupov posameznega para izdelkov.

- Verjetnostni graf (angl. probability graph): usmerjene povezave z utežmi določajo verjetnost - povezava iz vozlišča n_1 v vozlišče n_2 z utežjo w ($n_1 \xrightarrow{w} n_2$) pove, kakšna je verjetnost nakupa naslednjega izdelka n_2 na podlagi že kupljenega (n_1).

- Korelacijski graf (angl. correlation graph): ima neusmerjene povezave z utežmi, katere določajo korelacijske koeficiente parov izdelkov in kažejo podobnost nakupovalnih trendov skozi čas.

Grafi ali mreže veljajo za najnaravnejšo predstavitev podatkov različnih sistemov v fiziki, sociologiji, biologiji, inženirstvu, in informacijski tehnologiji [18]. Graf lahko zapišemo tudi v obliki matrike - vrednost v polju matrike i, j predstavlja utež med vozliščema i in j (obstaja več možnih zapisov). Večina digitalnih sledi (kot so dnevniški zapisi brskanja po spletu, zapisi nakupov, seznami predvajanja glasbe s spletnega radia ali objave uporabnikov na družbenih omrežjih) se lahko predstavi kot matrika sledi uporabnika (angl. user–footprint matrix) [22]. Vrstice matrike predstavljajo uporabnike, stolpci digitalne sledi, celice zapisujejo asociacijo med uporabniki in sledmi. V matriko lahko shranjujemo tudi jezikovne podatke (kot so na primer tviti, elektronska pošta, status posodobitve na Facebooku). Takrat stolpci predstavljajo besede ali n-grame, vrednost v celicah pa frekvenco ponovitve določenih besed za vsakega uporabnika. V veliki večini je en uporabnik povezan samo z zelo majhnim deležem vseh možnih sledi, zato je matrika sledi uporabnika redka oziroma razpršena matrika. Ta je zelo velika, večina vrednosti v celicah pa je 0. Po odstranitvi nizkih pojavitev glede na določen prag lahko na podatkih izvedemo analizo vzorcev [22].

Pri reševanju problema katere so skupine nakupljenih izdelkov v trgovini, je ponavadi edini razpoložjivi vir informacij zgodovina prodaj v obliki transakcijskih podatkov [14]. Transakcijske podatke je potrebno transformirati v podatke grafa. Vsaka vrstica v tabeli transakcijskih podatkov predstavlja en tip izdelka v košarici stranke. Pomembni atributi so ID nakupa, ID izdelka, stranka in čas. Pri transformaciji lahko upoštevamo dodatne filtre, kot so lastnosti stranke in časovno periodo (odvisno od poudarka analize). Podatke v transakcijskem formatu lahko transponiramo v matriko sledi uporabnika. Če hočemo dobiti graf sočasnih pojavitev, iz transakcijskih podatkov najprej naredimo tabelo, kjer ena vrstica vsebuje cel nakup z vsemi izdelki, ki so bili v košarici tega nakupa. Nato naredimo podskupine parov izdelkov vsakega

nakupa in jih agregiramo po vseh nakupih - tako določimo uteži na povezavah. Graf z neusmerjenimi povezavami z utežmi med izdelkii ima simetrično matriko sosednosti. Lahko določimo prag, da odstranimo vse povezave z manjšimi utežmi (odstranimo šum ustvarjen z naključnimi nakupi) in z vizualizacijo dodamo informacije, kot so frekvenca pojavitve izdelka, pogostost sočasne pojavitve para (debelina povezave glede na utež), pomembnost vozlišča ali rezultate, ki jih dobimo s pomočjo algoritmov na grafu (na primer metod za detekcijo skupnosti). Iz podatkov lahko zgradimo različne grafe, odvisno od poudarka analize: lahko se osredotočimo na specifično vozlišče ali podgraf [15].

2.3 Metode za detekcijo skupnosti

Grafi (imenovane tudi mreže), ki predstavljajo resnične sisteme, niso enaki grafom iz matematike, saj je v njih red mešan z neredom. Porazdelitev povezav v grafu ni homogena niti globalno niti lokalno. Med vozlišči v posebnih skupinah obstaja visoka koncentracija povezav in med samimi skupinami nizka koncentracija. Metode za detekcijo skupnosti izpostavljajo t.i. skupine oziroma pomembne strukture na grafu in omogočajo razumevanje njene organizacije. Te skupine imenujemo skupnosti (imenovane tudi moduli ali gruče, tudi komune) in so znotraj grafa močno povezana vozlišča, ki pogosto ustrezajo pomembnim funkcijskim enotam in/ali imajo podobno vlogo [23, 24].

Trije glavni pristopi za odkrivanje skupnosti so [23, 24]:

- nični modeli (angl. null models): nični model je naključen graf, ki ustreza orginalnemu samo v nekaterih strukturnih lastnostih. Z njim lahko preverimo, ali originalni graf odraža strukturo skupnosti. Newman in Girvan sta predlagala nični model, ki je naključna verzija originalnega - povezave ima premešane naključno pod omejitvijo, da se ohranja stopnja vozlišča. Metode osnovane na ničnih modelih primerjajo mero povezanosti v skupinah vozlišč s pričakovano vrednostjo v

ničnem modelu (v to skupino spada algoritem Louvain, predstavljen v nadaljevanju).

- Bločni modeli (angl. block models): metode osnovane na bločnih modelih identificirajo bloke vozlišč s skupnimi lastnostmi s pomočjo inferenčne statistike oziroma statističnega zaključevanja. Vozlišča dodeljena istemu bloku so statistično enakovredna glede na povezovanje z vozlišči v istem bloku in z drugimi bloki. Primer: preprost verjetnostni algoritem z maksimiziranjem verjetosti (angl. simple probabilistic algorithm expectation maximization) identificira skupnosti in verjetnost vozlišča, da ji pripada s parametrom maksimalne verjetnosti. Algoritem se ne ustavi, dokler ne konvergira [25].

- Modeli toka (angl. flow models): modeli osnovani na tokovih (podrobneje razloženo v nadaljevanju) raje kot na topološki strukturi delujejo na podlagi dinamičnosti grafa. Primarna funkcija je ujeti tok med komponentami sistema. Skupnosti so iz vozlišč, med katerimi tok po vstopu dolgo časa ne zapusti skupnosti - naključni sprehajalec se nahaja v območju skupnosti dlje, kot če bi bil graf naključen (primer je algoritem Map equation).

Pri določanju skupnosti imajo pomembno vlogo merila na podlagi katerih le te opredelimo. V nadaljevanju so predstavljena 4 ključna merila:

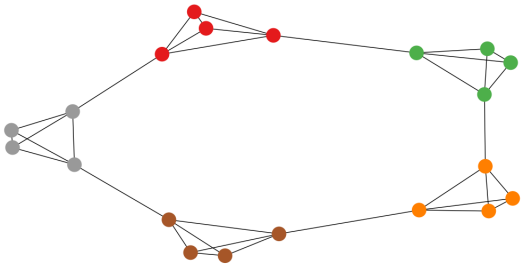
- Povezanost po povezavah (angl. edge connectivity) za par vozlišč je najmanjše število povezav, ki jih je treba odstraniti, da vozlišči postaneta nepovezani - da ni več poti med njima. Graf je povezan, če med vsakim parom vozlišč obstaja pot. V nasprotnem primeru graf sestavlja več medsebojno nepovezanih komponent.

- Čas obhoda med parom vozlišč je pomembna mera podobnosti vozlišč, ki temelji na lastnosti naključnih sprehodov. Definira povprečno število korakov, ki jih potrebuje naključni sprehajalec, da pride od prvega vozlišča prvič do drugega vozlišča in spet nazaj v začetno (prvo) vozlišče [24].

- Bližinska centralnost (angl.betweenness centrality) je mera centralnosti v grafu, ki je osnovana na najkrajših poteh. Za vsak par vozlišč v povezanem grafu obstaja vsaj ena taka najkrajša pot med vozlišči, da je število povezav, ki jih prečka pot (za neutežene grafe) ali vsota uteži (za grafe z utežmi) minimizirana. Bližinska centralnost za vsako točko je število teh najkrajših poti, ki grejo skozi vozlišče. Kriterij za iskanje dobrih skupnosti hoče maksimizirati povezave v skupnostih medtem ko minimizira povezave med skupnostmi. Dobre skupnosti naj bi imele visoko število povezav v skupnostih, kar dosežemo z maksimiziranjem modularnosti.

- Modularnost (angl. modularity) je lahko globalni kriterij za definicijo skupnosti, funkcija kakovosti in ključna sestavina najbolj popularnih metod za odkrivanje skupnosti na grafu. V standardni definiciji je podgraf (skupina vozlišč s povezavami v nekem grafu) skupnost, če število povezav v podgrafu presega pričakovano število notranjih povezav, ki bi jih imel enak podgraf v ničnem modelu [24]. Primeri grafov, kjer visoka modularnost ni pričakovana, so naključni grafi, cikli in drevesa. Mera modularnosti je skalarna vrednost, ki primerja gostoto povezav znotraj skupnosti z gostoto povezav med skupnostmi in je normalizirana tako, da pade med 1 in -1 [26]. V primeru najmočnejše možne modularne strukture, ki je krog klik (klika reda n je podgraf, ki je poln graf na n točkah) se bliža vrednosti 1 - skupnosti najdene na krogu klik so predstavljene na sliki 2.1. Modularnost vrednosti 1 bi predstavljalo popolno odkrivanje skupnosti, brez povezav med skupnostmi in vsemi skupnostmi, ki so tesno povezane, vrednosti pod 0 bi pomenilo zelo slabo odkrivanje skupnosti. Če imata dva grafa enako modularnost, še ne pomeni, da imata enako pomenljive skupnosti. Na redkih grafih lahko modularnost doseže visoko vrednost, ker se bolj osredotoča na ozka grla (manjše število izhajajočih povezav) kot na samo gostoto povezav znotraj skupnosti [27].

Skozi zgodovino so se pojavili algoritmi z uporabo različnih ključnih meril.



Slika 2.1: Pet odkritih skupnosti na krogu klik.

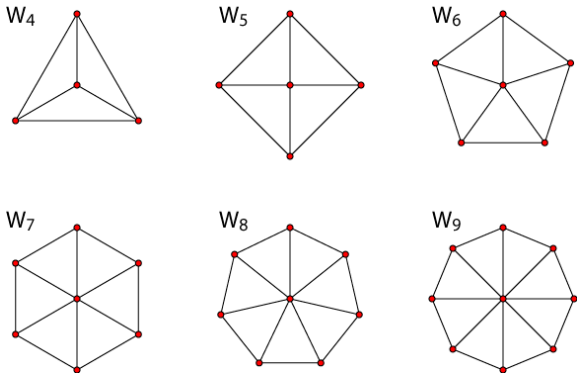
Girvan-Newman algoritem iz leta 2002 progresivno odstranjuje povezave z veliko bližino (za katere je velika možnost, da so med skupnostmi). Iz njega je sledil prvi (računsko) požrešni algoritem osnovan na modularnosti, ki sta ga predstavila leta 2004. Modularnost se uporablja za določanje prenehanja odstranjevanja povezav [27]. Za detekcijo skupnosti se lahko uporablja tudi simuliranje naključnih sprehodov (tokovi) – metode so osnovane na principu, da bo naključni sprehajalec nagnjen h temu, da bo ostal v gosto povezanem delu grafa (Pons, Latapy - algoritem Walktrap leta 2005, Martin Rosvall - algoritem Infomap leta 2008). Prvi algoritmi predvidevajo, da vsako vozlišče pripada le eni skupnosti. Pozneje pa so bili razviti tudi algoritmi, ki upoštevajo prekrivajoče skupnosti (vozlišče lahko pripada več skupnostim). Razširjanje po klikah (angl. clique percolation method - CPM) je postalo popularno z brezplačnim programom CFinder leta 2005. CPM je najbolj znana metoda za detektiranje prekrivajočih skupnosti. Izhaja iz predpostavke, da je graf sestavljen iz veliko klik - vsaki dve vozlišči sta sprva povezani v kliko, nato dve sosednji kliki združimo v verigo, če njuni k-kliki, katerima pripadata, delita k-1 članov [24]. Vozlišče lahko zato pripada več skupnostim. Iskanje klik v grafu je NP-poln problem. Jaewon Yang in Jure Leskovec sta leta 2013 v delu [28] objavila svoj algoritem BigClam, ki deluje tudi na velikih grafih. Posebnost algoritma je skalabilnost zaradi nenegativne matrične faktorizacije, ki matriko faktorizira v dve matriki, pri čemer imajo vse tri

matrike nenegativne elemente. Avtorja trdita v nasprotju z večino hipotez, da so deli skupnosti, ki se prekrivajo z drugo skupnostjo gosteje povezani od delov, ki se ne prekrivajo in da več skupnosti, ki si jih par vozlišč deli, kaže na večjo verjetnost, da sta vozlišči povezani [28].

Evalvacija razdeljevanja je najbolj učinkovita, ko primerjamo dobljene skupnosti s skupnostmi, za katere vemo, da dejansko obstajajo na grafu, ki se imenujejo resnične temeljne skupnosti (angl. ground-truth communities). Primerjavo lahko opravimo z različnimi merami, med najpogostejšimi sta normalizirana medsebojna informacija - NMI (angl. normalized mutual information) in iz nje izpeljana variacija informacije VI (angl. variation of information/shared information distance), ki je prava metrika - to pomeni, da se jo lahko interpretira kot razdaljo med naključnima spremenljivkama X in Y [27]. Obe meri izhajata iz informacijske teorije. NMI meri koliko informacije imamo o eni particiji (podgraf, ki lahko predstavlja skupnost ali njen del), ko poznamo drugo. V primeru popolnega ujemanja resničnih temeljnih skupnosti in skupnosti dobljenih z metodami za odkrivanje bi dobili NMI enak 1 (VI enak 0). Pogosti meri sta tudi F-mera in LFR (Lancichinetti-Fortunato-Radicchi [29]), ki je najbolj realno merilo, ker so zgenerirani grafi na katerih izvaja testiranje najtežja (skupnosti je težko identificirati), a ne upošteva prekrivajočih skupnosti. Kadar ima graf res veliko prekrivajočih skupnosti, algoritmi za odkrivanje skupnosti, ki so temu namenjeni, teh ne znajo detektirati dovolj dobro. Različni tipi in strukture grafov vplivajo na rezultat algoritmov. Izbira najbolj primernega algoritma za detekcijo skupnosti je odvisna od grafa, ki ga preučujemo.

Kateri algoritem bomo uporabili, je odvisno tudi od cilja študije: požrešni algoritmi niso dobri pri detekciji majhnih skupnosti. Z modularnostjo lahko samodejno določimo, koliko skupnosti je na grafu, vendar zaradi težnje k večjim skupnostim - algoritmi osnovnani na njej so nagnjeni k združevanju - prezremo manjše. Ta problem se imenuje omejitev ločljivosti (angl. resolution limit) [27]. Z uporabo t.i. lokalnih metod se mu lahko izognemo,

[1] https://upload.wikimedia.org/wikipedia/commons/3/30/Wheel-graphs.png



Slika 2.2: Graf iz cikla in središčne točke, s katero so povezane vse točke cikla, se imenuje n-kolo.[1]

izgubimo pa samodejno določevanje števila skupnosti. Algoritem Infomap je boljši pri identifikaciji majhnih skupnosti, ki sledijo določenemu voditelju (ponavadi je bolj natančna detekcija skupnosti boljša). Če potrebujemo hiter algoritem, lahko uporabimo algoritem s širjenjem label - LPA (angl. label propagation algorithm) [30]. Pri razdeljevanju v skupnosti LPA vozlišču pripiše oznako, kateri pripada večina njegovih sosedov. Njegova slabost je, da so njegovi rezultati spremenljivi, če ga zaženemo večkrat (ima velik standardni odklon).

Tekom let so razvili variacije in izboljšave za številne začetne algoritme: na primer algoritem SCP je osnovan na CPM-ju, a hitrejši in podpira tudi utežene grafe, COPRA osnovana na LPA-ju podpira prekrivajoče skupine, veliko izboljšav je imel tudi Infomap [24, 23]. V delih [31, 32] primerjajo delovanja različnih algoritmov na podatkovnih bazah, ki se razlikujejo po strukturi in velikosti. Najbolj enostavni graf ima lahko strukturo drevesa ali kolesa (n-kolo na sliki 2.2). CPM deluje bolje na strukturi drevesa, kjer so prekrivanja za algoritem bolj možna in manjša. Algoritem Infomap v splošnem da boljše rezultate, saj je veliko skupnosti odkritih s CPM na drugih strukturah napačnih [24]. Najboljše ujemanje skupin na grafu, ki predstavlja nakupljene izdelke na Amazonu, so v delu [31] dobili z Infomapom in Louvainom. Oba sprva nista podpirala grafa z neusmerjenimi povezavami z utežmi,

vendar je bila podpora zanj dodana kasneje. V delu [32] priporočajo Infomap za relativno majhne grafe (ki imajo število vozlišč manjše od 6000).

Ker je delovanje algoritmov zelo odvisno od strukture grafa, so nam lahko algoritmi za detekcijo skupnosti v pomoč pri odkrivanju novih informacij, ne moremo pa se zanesti na vse detektirane skupnosti. V delu [31] so pokazali, da tradicionalne metode za odkrivanje skupnosti ne uspejo najti resničnih temeljnih skupnosti na veliko grafih. Medtem ko se tradicionalni algoritmi za odkrivanje skupnosti osredotočajo na strukturo grafa, algoritmi za hierarhično gručenje (angl. hierarchical clustering) ponavadi upoštevajo atribute vozlišč. Z interakcijo med strukturo grafa in atributi vozlišč se pojavljajo nove metode za detekcijo skupnosti, ki lahko izboljšajo natančnost in robustnost rezultatov (primer takega algoritma je CESNA)[17].

2.3.1 Louvain

Algoritem Louvain je hitra hevristična metoda za odkrivanje skupnosti, ki so jo razvili Blondel et al. [26] leta 2008 na Univerzi Louvain. Eksperimenti so pokazali, da je njena časovna zahtevnost blizu $O(n \log n)$. Temelji na požrešnem aglomerativnem hierarhičnem gručenju, ki je osnovano na meri modularnosti (aglomerativne metode rekurzivno združujejo podobna vozlišča/skupnosti) [33]. Louvain je optimizacijska metoda, kar pomeni, da maksimizira ciljno funkcijo - modularnost. Formula 2.1 kaže izračun te mere, kjer je n_c število skupnosti, l_c je število povezav znotraj skupnosti, d_c je vsota stopenj vseh vozlišč v c in m je število povezav v grafu [33].

$$Q = \sum_{c=1}^{n_c} \left[\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right] \tag{2.1}$$

V primeru grafov z utežmi je modularnost kvalitete particije definirana kot v formuli 2.2 [26]. A_{ij} predstavlja utež na povezavi med i in j, $k_i = \sum_j A_{ij}$ je vsota uteži povezav v stiku z i, c_i je skupnost, kateri je pripisano vozlišče i. Funkcija delta $\delta(u, v)$ je 1 če $u = v$, drugače 0 in $m = \frac{1}{2} \sum_{i,j} A_{ij}$.

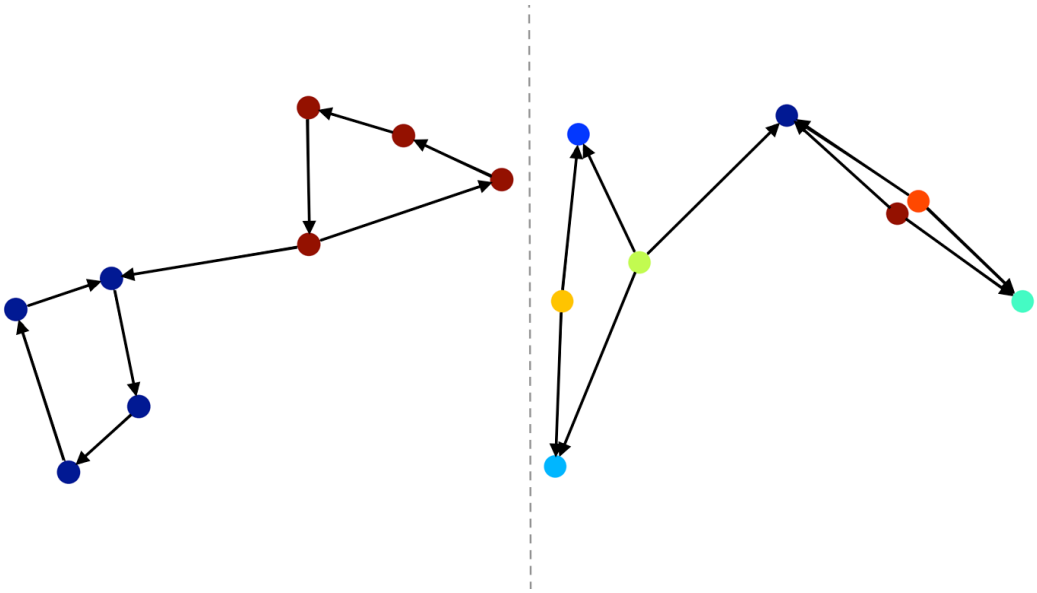
$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta\left(c_i, c_j\right) \tag{2.2}$$

V tej metodi je particija inicializirana tako, da je vsako vozlišče v svoji skupnosti. Potem izračunamo, s katero menjavo vozlišča v sosednjo skupnost pridobimo največ modularnosti in nato izvedemo menjavo. To ponavljamo, dokler ni premaknjeno več nobeno vozlišče. Nato spremenimo graf, tako da združimo vsako skupnost v eno samo vozlišče, povezave med skupnostmi so združene v eno samo z utežmi. Povezave znotraj skupnosti so dodane kot zanke. Ta induciran graf je uporabljen v naslednji iteraciji.

2.3.2 Infomap

Metoda Infomap zahteva močno poznavanje teorije informacij, ki kombinira Huffmanovo kodiranje in Shannonov izrek kodiranja (tudi brezizgubno izvorno kodiranje). Avtorji metode Infomap so pokazali, da je odkrivanje strukture skupnosti na grafih ekvivalentno reševanju problema kodiranja. Njegov namen je zmanjševanje dolžine kode [23]. V nasprotju z maksimiziranjem modularnosti je glavni pristop uporaba poti v grafu. Metoda deluje, če imajo skupnosti povezan tok v sebi, kar pomeni, da če naključno sledimo smerem povezav, ostanemo v isti skupnosti (primer in razlaga na sliki 2.3). Slabše pa deluje v primeru, če je toka malo ali nič, pa tudi če obtiči v mrtvi točki [33]. V nadaljevanju najprej predstavimo algoritem Map equation, ki omogoča identifikacijo skupnosti glede na kodiranje, potem algoritem za iskanje implementiran v Infomapu za minimiziranje Map equation-a preko možnih particij grafa.

Za vsako modularno particijo grafa obstaja informacijski strošek za opis premikov naključnega sprehajalca ali toka, torej sekvenca kod vozlišč v particiji. Particija z najkrajšo dolžino opisa najbolje določa strukturo skupin grafa glede na njeno dinamiko [34]. Vozliščem lahko določimo kode z uporabo Huffmanovega kodiranja. To je koda s prosto predpono, kar pomeni, da nobena koda ni vsebovana na začetku druge, zato lahko združene kode (poti) nedvoumno dekodiramo. Huffmanova koda pripisuje kratke kode pogostim dogodkom ali objektom in dolge kode redkim. S kodo, v kateri bi bile vse kode enako dolge, bi potrebovali več bitov za opis (naključnega)



Slika 2.3: Skupnosti na dveh grafih, ki imata enako strukturo, razen usmerjenosti povezav. Tok v grafu je najlažje razložiti s pomočjo usmerjenih povezav. Na levem grafu Infomap določi dve skupnosti - opazimo lahko dva cikla vozlišč, ki imata usmerjene povezave tako, da je več možnosti, da ostanemo znotraj skupnosti dlje časa. Na desnem grafu so vsa vozlišča ali ponori ali izviri - tu ni toka, iz nobenega vozlišča ne moremo potovati dlje kot za eno povezavo, zato vsako vozlišče predstavlja svojo skupnost.

sprehoda. Ker je veliko grafov strukturiranih v regije, v katerih sprehajalec po vhodu vanjo ostane nekaj časa in so premiki med njimi relativno redki, jim lahko določimo svoj modulski kodirni imenik (ti lahko uporabljajo enake kode). Za preklop med moduli moramo določiti tudi indeksni imenik, ki za vsak modul vsebuje vhodno in izhodno kodo. Z uporabo več imenikov smo transformirali problem minimiziranja dolžine opisa poti v problem najboljše razdelitve grafa z upoštevanjem toka. Bistvo Map equationa je, da lahko ocenimo učinkovitost optimalne kode, brez da bi dejansko našli optimalne kode za dano particijo. Dovolj je izračunati teoretično mejo za različne razdelitve grafa in izbrati tisto z najkrajšim dolžino opisa.

Informacijska vrednost določenega dogodka je logaritmično inverzna njegovi verjetnosti, da se bo zgodil (fomula 2.3). Če je X slučajna spremenjivka in $Pr(X = x) = p(x)$, je informacija dogodka x:

$$I(x) = \log \frac{1}{p(x)} = -\log p(x) \tag{2.3}$$

Največ informacije tako prinese dogodek, ki ima čim manjšo verjetnost (če se dogodek x skoraj nikoli ne zgodi, da veliko informacije, ko se zgodi) [27]. Glede na distribucijo verjetnosti lahko pojasnimo, kakšna je pričakovana informacija povezana z naključno spremenljivko X - mera, ki jo imenujemo entropija. Ta je najmanjša (enaka 0), ko je vrednost spremenljivke natančno določena (verjetnostna porazdelitev je enaka $p(x) = 1, 0, ..., 0$), in največja (enaka log n), ko so verjetnosti za vse možne vrednosti enake. Za modulsko particijo M iz n vozlišč $\alpha = 1, 2, ...n$ v m modulih $i = 1, 2, ...m$ definiramo spodnjo mejo $L(M)$. Za izračun spodnje meje L katerekoli particije, se naslonimo na Shannonov izrek kodiranja: pri uporabi n kod za opis n stanj slučajne spremenljivke X, ki se pojavlja s frekvenco p_i, povprečna dolžina kode ne more biti manjša od entropije slučajne spremenljivke X - formula 2.4.

$$H(x) = \sum_{1}^{n} p_i \log p_i \tag{2.4}$$

Če je osnova logaritma 2, entropijo merimo v bitih. Za izračun povprečne dolžine kode, ki opisuje korak v naključnem sprehodu, moramo utežiti pov-

prečne dolžine kod iz indeksnega imenika in modulskega imenika po njihovi stopnji uporabe. $L(M)$ predstavlja seštevek obeh entropij in je Map equation (več podrobnosti o usmerjenih ali neuteženih grafih v [34]). Za neusmerjene grafe z utežmi je enak (formula 2.5):

$$\begin{aligned} L(M) = w_{out}\log(w_{out}) - 2\sum_{i=1}^{m} w_{i_{out}}\log(w_{i_{out}}) \\ - \sum_{\alpha=1}^{n} w_\alpha \log(w_\alpha) + \sum_{i=1}^{m}(w_{i_{out}} + w_i)\log(w_{i_{out}} + w_i) \end{aligned} \tag{2.5}$$

Frekvenca obiska vozlišča α ustreza relativni uteži w_α vseh povezav vozlišča: je celotna utež vseh povezav vozlišča deljena z dvakratnikom celotne uteži vseh povezav na grafu. Relativna utež modula i je $w_i = \sum_{\alpha\in i} w_\alpha$, relativna utež povezav, ki izhajajo ven iz modula i je $w_{i_{out}}$, celotna relativna utež povezav med moduli je $w_{out} = \sum_m^{i=1} w_{i_{out}}$.

Jedro algoritma Infomap sledi Louvainovi metodi: sosednja vozlišča so združena v module, ki so pozneje združeni v supermodule. Najprej je vsako vozlišče dodeljeno v svoj modul. Potem je v naključnem vrstnem redu, vsako vozlišče premaknjeno v sosednji modul, izbran je premik, ki rezultira v največjem padcu Map equation-a. Če noben premik ne poveča padca, nobeno vozlišče ni premaknjeno in celoten graf je ponovno zgrajen tako, da zadnji moduli predstavljajo vozlišče. Procedura je ponovljena v naključnem vrstnem redu, dokler ni več izboljšav. Natančnost je lahko izboljšana z razbitjem modulov v zadnjem stanju:

- Premiki podmodulov: procedura generira enega ali več podmodulov za vsak modul (kot da bi bil ta cel graf). Te podmodule lahko premikamo med moduli.

- Premiki posameznih vozlišč: vsako vozlišče je svoj podmodul, ki ga lahko premikamo med moduli.

Infomap je hiter stohastičen algoritem. Z večjim številom ponovitev (v [23] priporočajo 100 ali več) je manj možnosti, da je zadnja particija lokalni miniumum. Pri tem je pomembno, da se map equation konceptualno

precej razlikuje od modularnosti, ki jo uporablja algoritem Louvain, saj se prvi osredotoča na medsebojno odvisnost povezav in dinamiko omrežja (kako struktura grafa vpliva na obnašanje), drugi na interakcijo v parih in proces tvorbe (kako je bila struktura ustvarjena). Ker modularnost išče vzorce z visokimi utežmi na povezavah med skupnostmi, da prednost večjemu številu skupnosti kot Infomap - če ni toka, algoritmi osnovani na modularnosti še vedno najdejo skupnosti. Ker je optimiziranje modularnosti NP težek problem, ni mogoče zagotoviti, da je pridobljena rešitev povsem optimalna, vendar pa pristop kljub temu daje dobre rezultate. V praksi se izkaže, da pristopa Infomap in Louvain ponekod dajeta podobne, drugje pa lahko tudi precej različne rezultate, zato je smiselno primerjati rezultate obeh in izbrati najustreznejšega za izbrani problem

Infomap zagotavlja dvo-nivojske, več-nivojske in prekrivajoče rešitve za analizo neusmerjenih, usmerejenih, neuteženih in uteženih grafov [23]. Dvonivojski map equation lahko posplošimo tako, da rekurzivno iščemo večnivojske rešitve. Algoritem rekurzivno poišče optimalno število gnezdenih modulov. Infomap posnema tok z uporabo dinamike prvega reda (kot običajni graf: kam tok teče je odvisno le od kje je trenutno) ali z uporabo dinamike drugega reda/Markove dinamike (kam tok teče je odvisno od pozicije kjer je trenutno in od kje je prišel – vhodne povezave pričakuje v obliki *začetno_vozlišče preko_vozlišča končno_vozlišče opcijska_utež*). Identificira lahko prekrivajoče module, v enoplastnem ali večplastnem grafu (angl. multilayer/multiplex network). V večplastnem grafu lahko vsako vozlišče obstaja v več plasteh z različnimi povezavami na vsakem nivoju. Vidimo, da Infomap podpira res veliko različnih grafov in tako nudi veliko možnosti za preučevanje in analizo.

Poglavje 3

Pristop in rešitev za analizo turističnih potovanj

Z MBA ugotovimo pogoste košarice, to so skupine izdelkov, ki so pogosto kupljeni skupaj in tako odkrijemo povezave med izdelki. Na podlagi rezultatov lahko prodajalci izdelke v trgovini premaknejo, da se nahajajo blizu drug drugega ali pripravijo novo akcijsko ponudbo, ki zajema izdelke v pogostih košarici/košaricah. S pomočjo MBA je lahko v spletni trgovini izbran izpostavljen oglas glede na trenutne izdelke v nakupovalni košarici. Podobno lahko uporabimo tudi rezultate iskanja skupin turističnih destinacij, ki so pogosto obiskane skupaj. Destinacijam seveda ne moremo spremeniti lokacije, lahko le poenostavimo potovanja med njimi: izboljšamo infrastrukturo, organiziramo javni prevoz med njimi in/ali dodamo nove linije. Turistične agencije lahko najdene skupine destinacij ponudijo v enem potovanju. Na posamezni destinaciji so lahko oglaševane ostale članice skupine.

Z raznimi karticami zvestobe prodajalci redno zbirajo informacije o opravljenih nakupih. Promocijsko gradivo lahko prilagodijo tudi na profil nakupovalca. Za zbiranje podatkov o obiskih nimamo na voljo tako zanesljivega vira (statistika podatkov je opisana v Poglavju 4). Število objav turistov povezanih z obiskanimi destinacijami dosegajo dovolj veliko število, da so uporabne za analizo vzorcev obiskovanja. Tudi obiskovalce lahko ločimo po

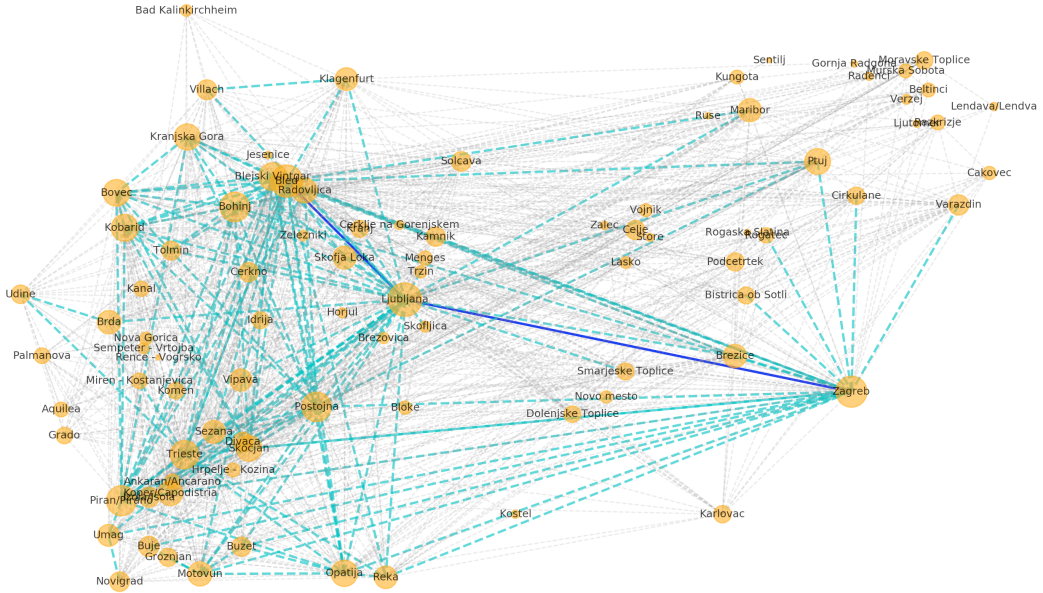
profilih [6].

Destinacije lahko enačimo z izdelki, če gledamo na obisk destinacije kot na poseben izdelek v nakupovalni košarici. Tako destinacija kot izdelek sta lahko oglaševana in ponujena v skupini kot paket izdelkov oziroma potovanje z več destinacijami. Obiskane destinacije v potovanju lahko enačimo s kupljenimi izdelki v nakupovalni košarici: v obeh primerih imamo skupine izdelkov, ki jih lahko združujemo v množice, kjer vrstni red elementov ni pomemben. Poudariti je potrebno še časovni okvir potovanja in košarice: medtem ko je konec nakupa definiran v hipu, ko je košarica plačana (vsi izdelki imajo enak končni čas nakupa), je definicija trajanja in konca potovanja enostavna le za tiste opravljene preko turističnih ponudnikov. Pri objavah obiskovalcev je potrebno definirati razmik med objavami, da obisk destinacije še štejemo v eno potovanje. Določitev razmika je odvisna od samih podatkov, na katerih je opravljena analiza in njenega poudarka. Poleti Slovenijo kot državo v tranzitu prečka veliko turistov, ki imajo za dopust izbrano ciljno destinacijo še bolj na jugu. Za razmik smo določili 7 dni, ker je malo verjetno, da bi turist v tem času dvakrat obiskal isto destinacijo, prav tako je tudi povprečno število dni, kolikor turisti ostanejo v Sloveniji, manjše od 3 dni. Predvidevamo, da je časovno zaporedje večine objav skladna z dejanskim zaporedjem obiskov destinacij, saj mobilne aplikacije (v našem primeru TripAdvisor) turiste zaprosijo, da oddajo mnenje v kratkem času in iz tega lahko sklepamo, da je čas objave v večini primerov približno skladen s časom dejanskega obiska [35].

3.1 Princip delovanja na destinacijah

Za potrebe analize tokov med turističnimi destinacijami na grafu smo izbrali algoritma Louvain in Infomap, ker sta med seboj konceptualno različna in ponavadi dajeta različne rezultate (razlaga v Poglavju 2), sta tudi med pogosteje uporabljenimi in sta relativno hitra. Za gradnjo grafa smo za vozlišča, ki v primeru analize MBA predstavljajo izdelke, vzeli posamezne turistične destinacije.

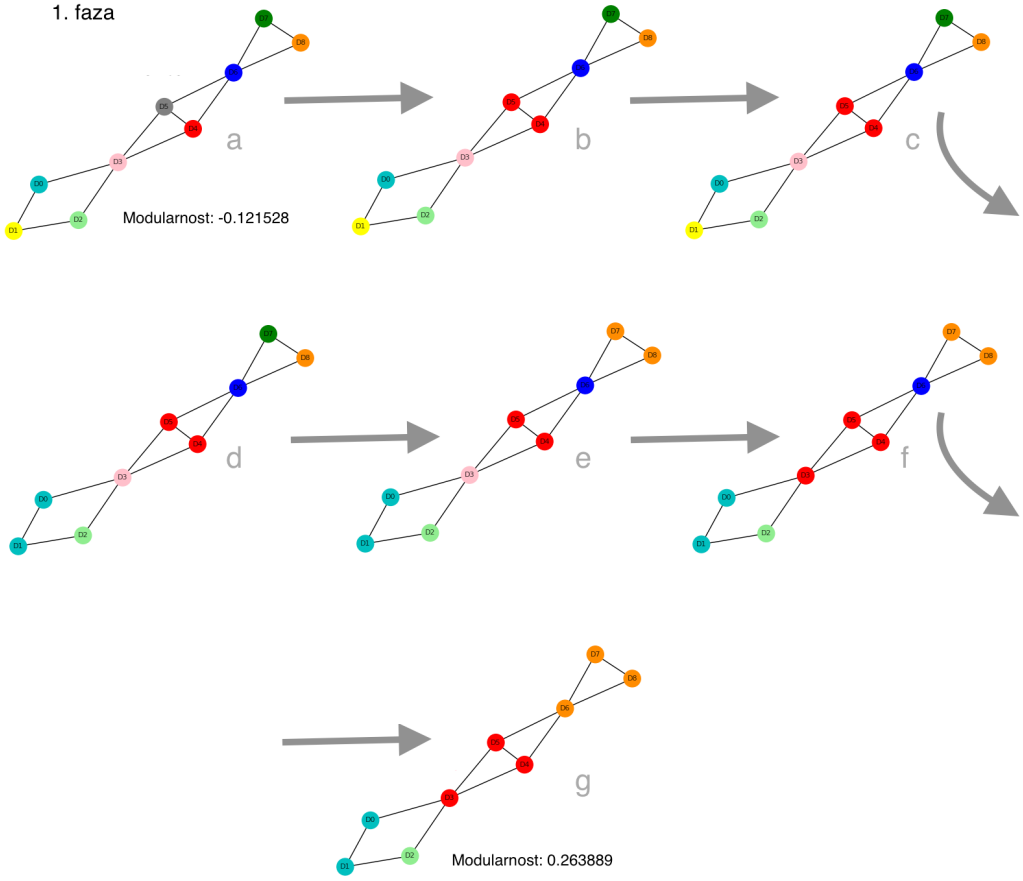
Tako podatke o nakupih kot objave turistov oziroma obiskov imamo na voljo v transakcijski obliki. Iz te izhajamo pri gradnji grafa sočasnih pojavitev, na katerem potem iščemo pogoste skupine izdelkov. Graf sočasnih pojavitev iz vseh transakcijskih zapisov je na sliki 3.1. Sestavljen je iz vozlišč, ki predstavljajo destinacije, in neusmerjenih povezav z utežmi, ki določajo število obiskov posameznega para destinacij. Z grafa lahko predvidevamo, katera vozlišča bodo večkrat prisotna v pogostih košaricah glede na velikost vozlišč, ki so sorazmerna stopnji vozlišča. V neusmerjenem grafu G je stopnja ali valenca vozlišča u enaka številu povezav grafa G, ki imajo vozlišče u za svoje krajišče. Večja vozlišča predstavljajo destinacije, ki se v vseh potovanjih skupno pojavijo z več različnimi destinacijami. Destinacijo sestavlja več atrakcij. Atrakcije so v destinacijo združene na podlagi geografske lokacije (enako kot v delu [35]). Poudariti je potrebno, da lahko analizo opravljamo tudi na samih atrakcijah, odvisno od analize.



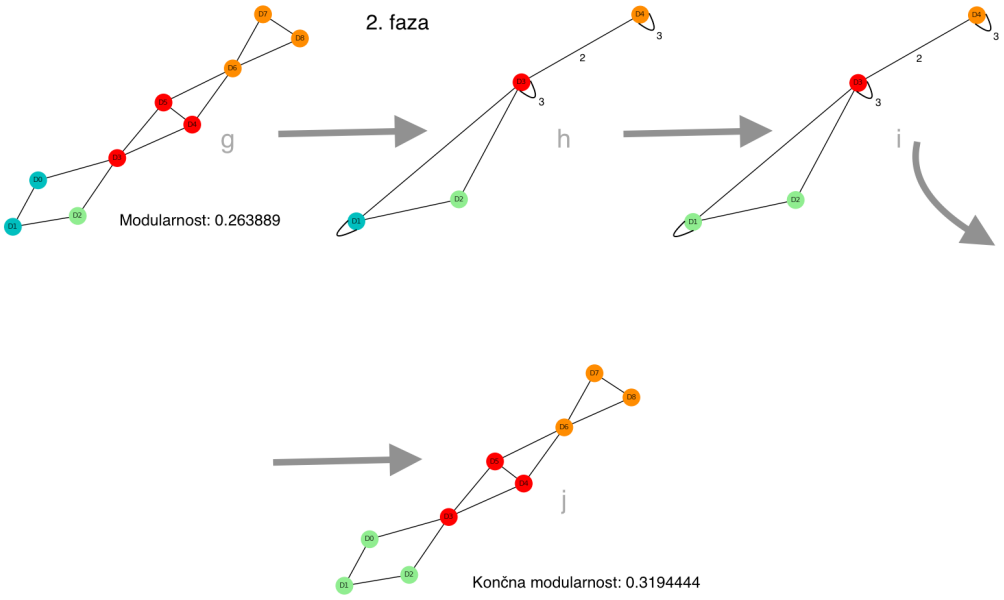
Slika 3.1: Graf sočasnih pojavitev zgeneriran iz vseh transakcijskih zapisov je gost in povezan (iz ene komponente). Vozlišča so pozicionirana glede na geografsko lokacijo. Velikost vozlišča je sorazmerna s stopnjo vozlišča - večkrat obiskane destinacije so zato večje. Graf je sestavljen iz 3 vrst povezav: šibke, srednje močne in močne. Definirane so z velikostjo uteži: siva črtkana črta označuje šibke povezave, to so povezave z utežjo, manjšo od povprečne, temno modra polna črta pomeni močne povezave z utežjo, večjo od polovične največje, svetlo modra črtkana črta srednje močne povezave. Graf ni pregleden, saj se ne vidijo vse povezave in njihove končne točke oziroma vozlišča.

Iz samega grafa ne moremo razbrati, katere so pogosto obiskane skupine destinacij. Na grafu moramo zato pognati še metode s področja MBA. Na slikah 3.2 in 3.3 predstavljamo kako algoritem Louvain določa skupnosti in na slikah 3.4 in 3.5 kako jih določa Infomap. Odkrite skupnosti se razlikujejo. Ker je graf majhen, zadostuje ena ponovitev obeh faz, ki so predstavljene v Poglavju 2 (na večjih grafih je ponovitev obeh faz več). V prvi fazi naključno izberemo vozlišče in ga priključimo k najbolj optimalni sosednji skupnosti (če je izboljšanje modularnosti oziroma map equationa možna). Ko stanje konvergira in ni več izboljšav, sledi druga faza, ki združi vozlišča iste skupnosti v eno samo vozlišče in nato nad njimi izvaja optimizacijo.

Določanje skupnosti Louvain (slika 3.2): na začetku je vsako vozlišče v svoji skupnosti (a). V prvi fazi ponavljamo naslednji postopek: naključno izberemo vozlišče, izračunamo kakšna je sprememba modularnosti, če ga priključimo h katerikoli sosednji skupnosti in izberemo najboljši možen prehod (modularnost bi radi zvišali). V koraku (b) sivo vozlišče priključimo rdeči skupnosti. Vozlišče lahko ostane tudi v isti skupnosti, kot je, če ni izboljšanja modularnosti: v koraku (c) izberemo eno vozlišče v rdeči skupnosti (vseeno katero) - ker z nobeno priključitvijo v sosednjo skupnosti ni pozitivne spremembe modularnosti, ostane vozlišče še naprej v rdeči skupnosti. V koraku (d) rumeno vozlišče priključimo modri skupnosti ... postopek nadaljujemo (e,f,g), dokler po določenih n ponovitvah ni izboljšanja modularnosti (g). V drugi fazi (slika 3.3) iz skupnosti (g) zgradimo nov graf (h), v katerem je vsako vozlišče ena skupnost: povezave znotraj skupnosti postanejo povezava sama vase, njena utež je vsota uteži na povezavah znotraj skupnosti. Nato spet ponavljamo postopek iz faze 1: izboljšanje modularnosti nam prinese le spremememba skupnosti enega vozlišča (i). Na koncu graf pretvorimo v originalna vozlišča (j).

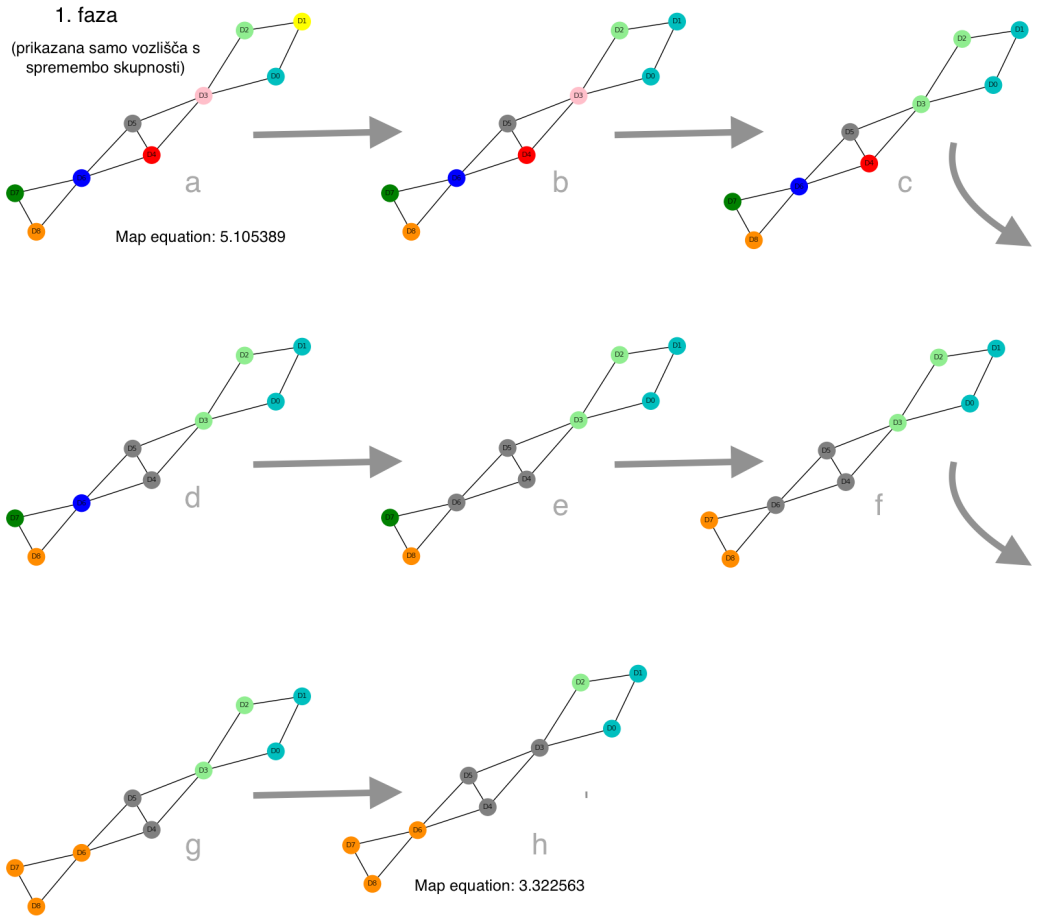


Slika 3.2: Določanje skupnosti Louvain: na začetku je vsako vozlišče v svoji skupnosti (a). V prvi fazi ponavljamo naslednji postopek: naključno izberemo vozlišče, izračunamo kakšna je sprememba modularnosti (izračunana po formuli 2.2), če ga priključimo h katerikoli sosednji skupnosti in izberemo najboljši možen prehod (modularnost bi radi zvišali): koraki (b)-(g), dokler po določenih n ponovitvah ni izboljšanja modularnosti (g).

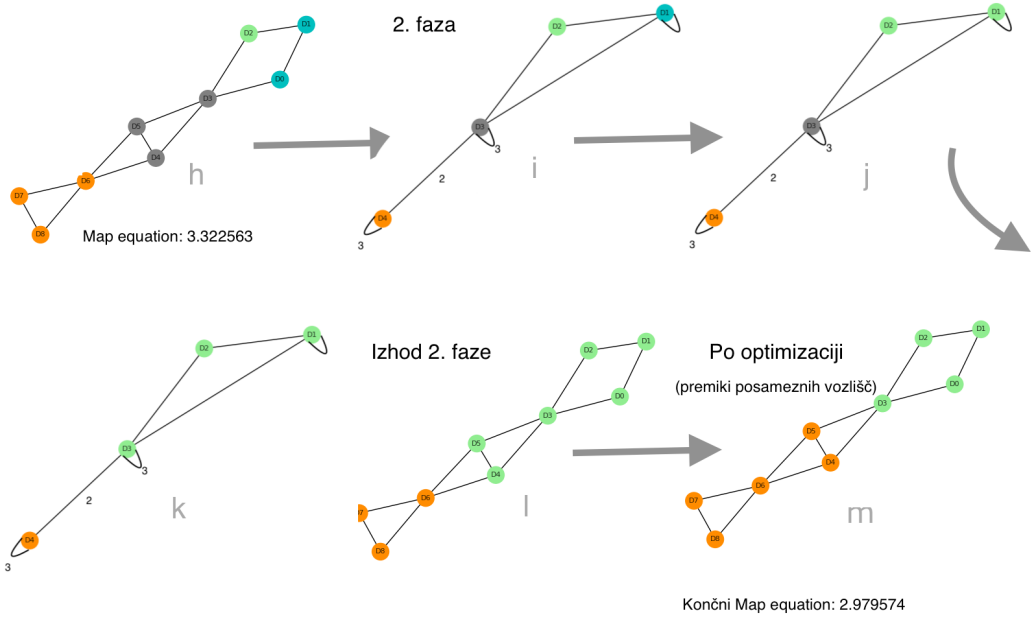


Slika 3.3: Določanje skupnosti Louvain: v drugi fazi iz skupnosti (g) zgradimo nov graf (h), v katerem je vsako vozlišče ena skupnost. Nato spet ponavljamo postopek iz faze 1: izboljšanje modularnosti nam prinese le sprememba skupnosti enega vozlišča (i). Na koncu graf pretvorimo v originalna vozlišča (j).

Določanje skupnosti Infomap (slika 3.4) : na začetku je vsako vozlišče v svoji skupnosti (a). V prvi fazi ponavljamo naslednji postopek: naključno izberemo vozlišče, izračunamo kakšna je sprememba map equationa, če ga priključimo h katerikoli sosednji skupnosti in izberemo najboljši možen prehod (map equation bi radi znižali): koraki (b)-(h). Vozlišče lahko ostane tudi v isti skupnosti, kot je, če ni izboljšanja map equationa. Postopek končamo, ko po določenih n ponovitvah ni izboljšanja (h). V drugi fazi (slika 3.5) iz skupnosti (h) zgradimo nov graf (i), v katerem je vsako vozlišče ena skupnost: povezave znotraj skupnosti postanejo povezava sama vase, njena utež je vsota uteži na povezavah znotraj skupnosti. Nato spet ponavljamo postopek iz faze 1: izboljšanje map equationa nam prinese sprememba skupnosti dveh vozlišč (j, k). Graf pretvorimo nazaj v originalna vozlišča (l). Ob koncu izvedemo še optimizacijo: premike posameznih vozlišč (m).



Slika 3.4: Določanje skupnosti Infomap: na začetku je vsako vozlišče v svoji skupnosti (a). V prvi fazi ponavljamo naslednji postopek: naključno izberemo vozlišče, izračunamo kakšna je sprememba map equationa (izračunan po formuli 2.5), če ga priključimo h katerikoli sosednji skupnosti in izberemo najboljši možen prehod (map equation bi radi znižali): koraki (b)-(h).



Slika 3.5: Določanje skupnosti Infomap: v drugi fazi iz skupnosti (h) zgradimo nov graf (i), v katerem je vsako vozlišče ena skupnost. Izboljšanje map equationa nam prinese sprememba skupnosti dveh vozlišč (j, k). Graf pretvorimo nazaj v originalna vozlišča (l) in izvedemo še optimizacijo (m).

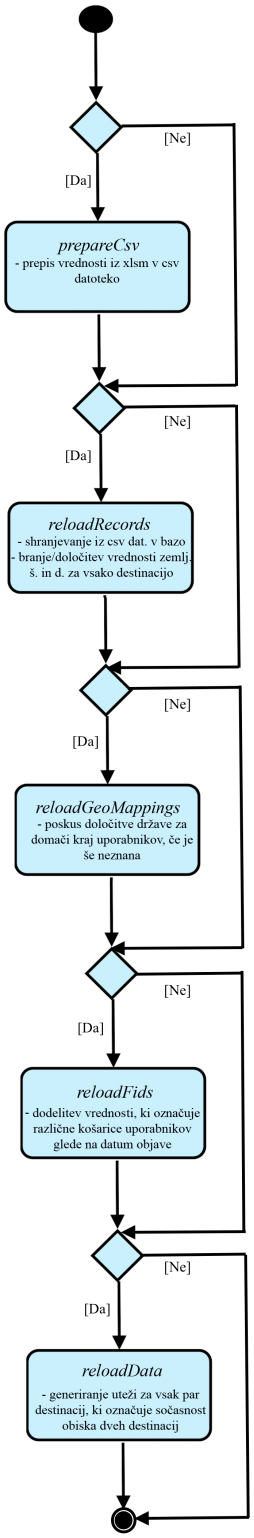
Da bi predstavljeni metodi lahko uporabili nad podatki pridobljenimi s spletnega mesta TripAdvisor smo implementirali ustrezno programsko rešitev, ki je opisana v nadaljevanju.

3.2 Implementacija rešitve za analizo grafov

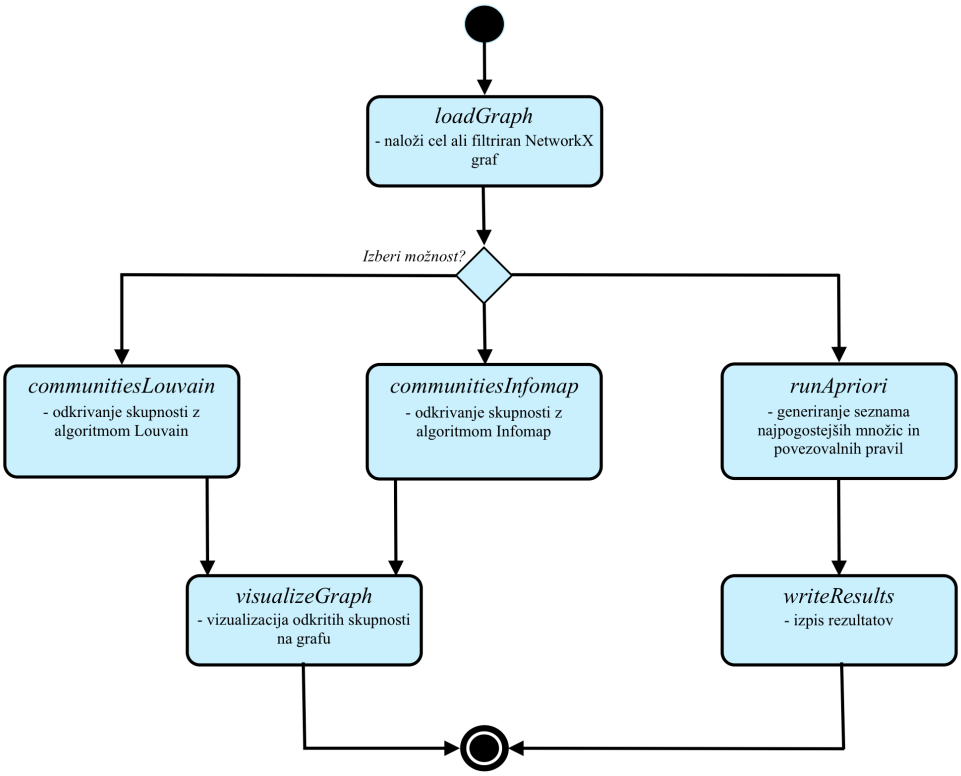
V okviru magistrskega dela smo implementirali programsko rešitev, ki za vhodne podatke sprejme objave v transakcijski obliki. Splošno potrebne naloge za analizo podatkov so [36]:

- Interakcija z zunanjim svetom: branje in pisanje datotek, interakcija s podatkovnimi bazami.
- Priprava: čiščenje, kombiniranje, normaliziranje, transformiranje podatkov za analizo.
- Transformacija: uporaba matematičnih in statističnih operacij na skupinah naborov podatkov za pridobitev novega nabora podatkov kot je na primer agregacija velike tabele po spremenljivkah skupine.
- Modeliranje in računanje: povezovanje podatkov s statističnimi modeli, algoritmi za strojno učenje ali drugimi računskimi orodji.
- Predstavitev: ustvarjanje interaktivnih ali statičnih grafičnih vizualizacij ali tekstovnih povzetkov.

Slika 3.6 opisuje začetne korake izvajanja programa, predprocesiranje in generiranje grafa sočasnih pojavitev (datoteka *LoadData.py*) oziroma interakcijo, pripravo in transformacijo podatkov. Možno je zgraditi graf tudi s filriranjem po turistični sezoni (poletna, zimska, novoletni prazniki) in po atributih, če so ti prisotni v začetnih transakcijskih podatkih (kot na primer starost, spol uporabnika). Da lahko predlagamo skupne kampanje glede na različne lastnosti obiskovalcev je potrebna analiza z uporabo filtriranj na grafu.

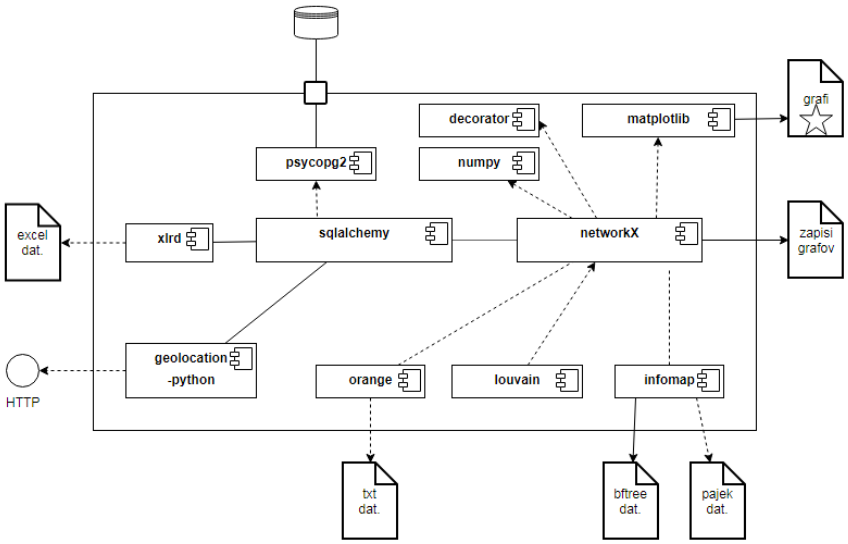


Slika 3.6: Diagram začetnih korakov vključuje predprocesiranje in generiranje grafa sočasnih pojavitev. Ker je vsak od korakov časovno zahteven (zlasti označevanje košaric in generiranja uteži) je možno postopno poganjanje korakov.



Slika 3.7: Diagram korakov, ki sledijo po generiranju grafa sočasnih pojavitev. Za analizo lahko izbiramo med algoritmi Louvain, Infomap in Apriori.

Po končanem generiranju grafa sočasnih pojavitev lahko na njem opravimo analizo - nadaljnje korake na sliki 3.7. Na grafu lahko poženemo metodi za odkrivanje skupnosti (Infomap in Louvain) in te vizualiziramo. Prikazane grafe lahko tudi shranimo. Za pomoč pri razumevanju rezultatov implementirana rešitev omogoča tudi uporabo metode Apriori, ki sama po sebi ni grafovska metoda.



Slika 3.8: Diagram komponent v programski rešitvi kaže njihovo medsebojno interakcijo. Osrednji komponenti sta knjižnici NetworkX in Sqlalchemy.

Analizo delamo s pomočjo vključenih knjižnic. V tabeli 3.1 so predstavljene vse uporabljene knjižnice in orodja.

Na sliki 3.6 je diagram komponent v programski rešitvi, ki prikazuje interakcijo med njimi.

Programska rešitev je izdelana v programskem jeziku Python, interpretnem jeziku, ki je priznan v znanstvenih krogih [36]. Prevzemanje Pyhona v znanstvenih krogih tako v industrijskih aplikacijah kot v akademskih raziskavah znatno raste že od začetka 21. stoletja, deloma tudi zaradi lahke integracije kode spisane v C, C++ in FORTRAN.

Za analizo grafov v Pythonu se najpogosteje uporabljajo naslednje tri prosto dostopne knjižnice [37]: iGraph [38], NetworkX [39] in SNAP [40]. NetworkX omogoča maksimalno fleksibilnost na račun slabše učinkovitosti, medtem ko je iGraph optimiziran za učinkovitost, a manj fleksibilen (dobro podpira samo statične grafe – dinamično dodajanje ali odstranjevanje vozlišč in povezav je drago), SNAP pa je nekje med njima [37]. Ločimo lahko tri

Tabela 3.1: Uporabljene knjižnice in orodja.

Knjižnica ali orodje	**Namen**	**Domača stran**
`xlrd`	Za branje iz Excel zavihkov.	`http://www.python-excel.org/`
`sqlalchemy`	ORM, fleksibilnost SQLa.	`https://www.sqlalchemy.org/`
`psycopg2`	PostgreSQL adapter za Python.	`http://initd.org/psycopg/`
`networkX`	Grafi, funkcije za njihovo preučevanje.	`https://networkx.github.io/`
`decorator`	Potrebuje jo NetworkX.	`http://decorator.readthedocs.io/en/latest/`
`numpy`	Matrične reprezentacije grafov, računanje z matrikami.	`http://www.numpy.org/`
`matplotlib`	Risanje grafov.	`https://matplotlib.org/`
`infomap`	Odkrivanje skupnosti na grafu.	`http://www.mapequation.org/code.html`
`louvain`	Odkrivanje skupnosti na grafu.	`https://github.com/taynaud/python-louvain`
`orange`	Odkrivanje povezovalnih pravil.	`https://orange.biolab.si/`
`geolocation-python`	Google maps API za pridobitev infomacij o lokaciji.	`https://github.com/slawek87/geolocation-python`

vrste metod, ki jih knjižnice podpirajo: metode za generiranje grafa, metode za manipulacijo s strukturo grafa in metode za analizo, ki ne spreminjajo strukture grafa, ampak izračunajo njegove statistike. Odločili smo se za NetworkX, ker omogoča enostavno generiranje in manipulacijo grafa, ima dobro dokumentacijo in podporo za neusmerjene grafe z utežmi (v sekciji 3.2.2 opisujemo, zakaj potrebujemo ravno tak tip grafa in kako ga naredimo). Tudi namestitev ni zahtevna, saj poteka preko ukazne vrstice s pomočjo ukaza *pip*, ki je že samodejno nameščen na novejših verzijah Pythona 2 in 3, medtem ko je pri iGraphu in SNAPu za namestitev lahko potrebna tudi izdelava programa iz izvorne kode.

3.2.1 Interakcija z zunanjim svetom in priprava podatkov

V tem podpoglavju opisujemo naslednje začetne korake z diagrama 3.6: *prepareCsv, reloadRecords, reloadGeoMappings*. Sestavni del vseh treh korakov je shranjevanje v podatkovno bazo. V rešitvi uporabljamo orodje `sqlalchemy`, ki nam omogoča rabo različnih vrst podatkovnih baz [41].

Priporočeno je, da uporabnik za vhod v rešitev poda transakcijske podatke v datoteki csv. Omogočeno je tudi branje iz excel datoteke, vendar je ta prepisana v csv format. V datoteki mora biti za vsako vrstico (obisk destinacije) prisotna veljavna vrednost z unikatnim identifikatorjem destinacije in obiskovalca in zabeleženim časom objave, drugače ta ni vnešena v tabelo transakcijskih podatkov v bazi. Po končanem prepisu sledi vnos zemljepisne širine in dolžine za vsako destinacijo: lokacija se lahko prebere iz posebne csv datoteke ali določi na podlagi lokacij objav, če je ta prisotna v transakcijskih podatkih.

Če je v transakcijskih podatkih prisoten stolpec (po prepisu v bazo atribut), ki vsebuje domači kraj uporabnika, lahko na njem poskusimo določiti državo. Domači kraj najprej primerjamo z vrednostmi s seznama držav (direktno ujemanje), potem pokrajin Združenih držav Amerike, nato sledi poskus iskanja z delnim ujemanjem in primerjavo zadnjih dveh črk z znanimi

kraticami držav. Zatem kličemo Google maps API, ki lahko vrne rezultat z najbolj ustrezno lokacijo in državo ali prazno vrednost (brezplačna uporabe storitve ima dnevno omejitev).

Korak *reloadGeoMappings* je potreben le, če bomo delali analizo s segmentacijo obiskovalcev po državi porekla, drugače lahko že po koraku *reloadRecords* nadaljujemo z generiranjem grafa. Ta postopek opisujemo v naslednjem podpoglavju.

3.2.2 Generiranje grafa sočasnih pojavitev

V tem podpoglavju opisujemo začetna koraka z diagrama na sliki 3.6: *reloadFids* in *reloadData*.

Vsaka vrstica v tabeli transakcijskih podatkov v našem primeru namesto enega izdelka v nakupljeni košarici stranke predstavlja eno obiskano turistično destinacijo v celotnem potovanju. Pomembni atributi so ID potovanja (namesto nakupa), ID destinacije (namesto izdelka), obiskovalec (namesto stranke) in čas objave (namesto nakupa). Iz transakcijskih podatkov bi radi naredili neusmerjen graf z utežmi, ki predstavljajo število sopojavitev obiskov para destinacij. ID nakupa povezuje vse nakupljene izdelke v košarici. V podatkih s spleta imamo lahko prisoten enakovreden atribut ID potovanja. Če ga ni, moramo poskusiti potovanje določiti sami na podlagi časa objave in ID obiska, ki je za vsako obiskano destinacijo unikaten. Glede na poljubno izbran časovni razmik, ki ne sme biti prevelik, lahko sami določimo ID potovanja oziroma nov stolpec, ki bo izražal različna potovanja posameznega uporabnika. Zapise uredimo po času objave in ko naletimo na prvo objavo uporabnika, si zapomnimo čas objave. Da vsako naslednjo objavo štejemo v isto potovanje kot prejšnjo, mora biti čas nove objave znotraj dovoljenega časovnega odmika od prejšnje objave. Na ta način dobimo časovne verige objav, ki predstavljajo objave enega potovanja. Po predprocesiranju transakcijskih podatkov vsak zapis vsebuje unikaten par potovanja in obiska destinacije.

Za generiranje grafa sočasnih pojavitev je potrebno podatke v transak-

cijskem formatu spremeniti v format, v katerem v vsaki vrstici opazujemo potovanje z vsemi obiskanimi destinacijami. V množici obiskanih destinacij posameznega potovanja moramo poiskati vse kombinacije parov destinacij tega potovanja. Podatke parov nato združimo v unikatne pare, ki na grafu predstavljajo povezave, ter določimo njihovo skupno število, ki na grafu predstavljajo uteži. Za analizo lahko določimo mejo za odstranitev manjših uteži. Pri vizualizaciji na sliki 3.1 širina povezav predstavlja število sočasnih pojavitev dveh destinacij v potovanjih. Na tej sliki vidimo tudi, da ima vsako vozlišče res najmanj enega soseda (tako mora biti, ker smo pri generiranju grafa izločili potovanja s samo eno destinacijo).

3.2.3 Povezovanje grafa z algoritmi za analizo in predstavitev

NetworkX ima že podprte nekatere metode za analiziranje grafa, tudi nekaj za odkrivanje skupnosti, a ne naprednih. Za uporabo algoritmov Louvain in Infomap je potrebno dodatno nameščanje. Knjižnica Louvain ima omogočeno enostavno nameščanje preko *pip*a in ima NetworkX grafe zelo dobro podprte, saj jih sprejema kot argument. Za lokalno uporabo ogrodja Infomap moramo zgraditi program iz izvorne kode[1], generiran graf zapisati v datoteko v enem od formatov, ki jih podpira (na primer pajek, datoteke s končnico .net) in podati pot do nje kot argument pri zagonu programa (iz Python kode kličemo program .exe z uporabo modula `subproces`).

V vozlišču NetworkX grafa ne shranjujemo celih zapisov, da nimamo težav s prostorom. V bazi je shranjen samo celotni graf, tisti narejeni s filtriranjem pa se shranijo na disk v več formatih (edgelist za nalaganje z NetworkX, pajek za Infomap). Graf z odkritimi skupnostmi je predstavljen tako, da različne barve vozlišč pomenijo pripadnost različnim skupnostim. Na voljo je izris grafa z vozlišči, ki upošteva tudi geografsko lego destinacij, kjer koordinatne osi predstavljajo zemljepisno širino in višino. Možen je pregled transak-

[1] V času implementacije še ni bila na voljo beta verzija Infomapa, ki omogoča namestitev preko pip-a.

cijskih podatkov in celotnega grafa z vizualizacijo različnih histogramov in izpisom statistik, kar lahko pomaga pri razumevanju rezultatov. Analiziramo lahko tudi s pomočjo algoritma Apriori, ki nam izpiše seznam najpogostejših množic in povezovalna pravila glede na željeno podporo in zaupanje. Za izpis moramo pred uporabo algoritma Apriori iz grafa generirati košarice v tekstovno datoteko (generiranje se zgodi implicitno znotraj koraka *runApriori* in samo, če tekstovna datoteka še ne obstaja).

Poglavje 4

Pregled rasti in razvoja turizma v Sloveniji

Na tem mestu je potrebno za razumevanje analize poudariti, da se javno dostopni podatki na podlagi objav obiskovalcev razlikujejo od znanih statistik in ti javno dostopni podatki na eni strani predstavljajo tako prednosti, na drugi pa tudi slabosti. Najprej moramo torej opredeliti, zakaj v delu uporabljamo primernejši termin obiskovalec in ne turist. Pojem obiskovalca definira osebo, ki potuje v kraj izven stalnega bivališča za manj kot 12 mesecev in čigar glavni razlog potovanja ni opravljanje plačane dejavnosti [42]. Obiskovalce delimo na turiste, ki v obiskani državi ostanejo vsaj eno noč, in na enodnevne obiskovalce oziroma izletnike, ki ostanejo manj kot 24 ur. Statistika obiskovalcev je lahko drugačna od statistike turistov, ker zajema še enodnevne obiskovalce in tranzitne potnike, ki v Sloveniji predstavljajo ogromno tržišče [42].

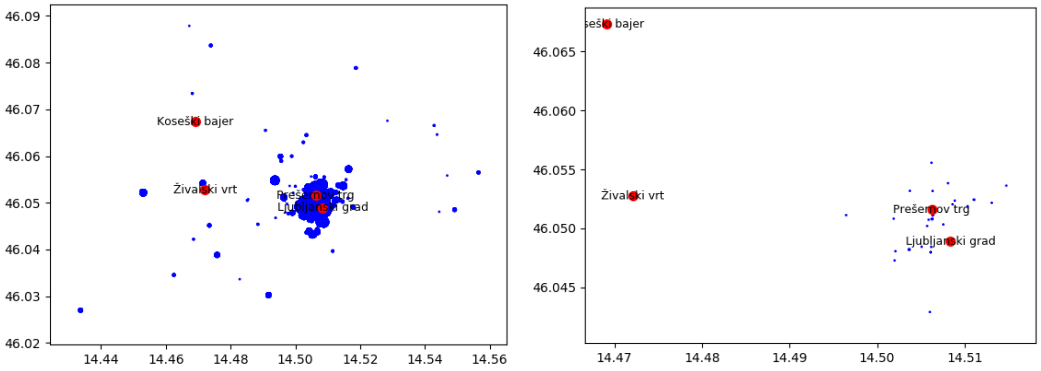
Analiza rasti in razvoja turizma v Sloveniji je narejena na podlagi podatkov Statističnega urada Republike Slovenije (SURS). Ta vodi evidenco o prihodih in nočitvah turistov, ki je javno dostopna na njihovi spletni strani [3]. Uradnih statistik, ki se nanašajo na obiskovalce v Sloveniji, je malo in se naslanjajo na ankete in raziskave. Kot primer navedimo izračun prilivov od tujih potovanj, pri katerem Banka Slovenije upošteva tudi porabo enodnevnih

obiskovalcev, izhajajoč iz mesečnih podatkov mejnih prehodov in raziskave o povprečni potrošnji, ki jo opravlja SURS [43]. Javno dostopni podatki na podlagi objav obiskovalcev zajemajo tako turiste kot enodnevne obiskovalce, kar predstavlja prednost našega pristopa. Slabost javno dostopnih podatkov, ki jih uporabljamo v analizi, je pomanjkanje informacij o glavnem razlogu potovanja, saj glede na opredelitev obiskovalec prvobitno ne opravlja plačane dejavnosti v skladu z definicijo. Verjetnost za izključitev teh popotnikov, ki potujejo zaradi opravljanja plačane dejavnosti, močno povečamo z gradnjo grafa sočasnih pojavitev, saj službeno potovanje najpogosteje vključuje eno destinacijo.

V nadaljevanju bomo najprej predstavili javno dostopne podatke, torej zbrane objave obiskovalcev, na katerih bomo kasneje opravili analizo. Nato sledi primerjava naših zbranih podatkov z znanimi statistikami slovenskega turizma.

4.1 Predstavitev podatkov

V analizi smo uporabili javno objavljene podatke pridobljene s spletnega mesta TripAdvisor [44], ene izmed vodilnih turističnih platform, ki predstavlja raznoliko turistično ponudbo z vsega sveta [45]. Po prepisu iz vhodne datoteke, v kateri je vsaka vrstica ena objava obiskovalca, je skupno število vseh transakcijskih zapisov v podatkovni bazi enako 136724. Vsak zapis je objava obiskovalca ob obisku turistične destinacije, locirane v Sloveniji, ali v sosednji državi blizu meje, ki ima prisotne vse potrebne atribute za grajenje grafa sočasnih pojavitev: ID objave oziroma obiska, ki je za vsak zapis unikaten (*record_id* se samodejno povečuje ob vsakem shranjevanju novega zapisa), ID uporabnika, ki je ustvaril objavo (*user_id*, vseh uporabnikov je 69746), datum objave (*review_date*) in ime destinacije, na katero se objava nanaša (*destination* je ime občine ali kraja, ki je tudi ID, ker ima vsaka destinacija unikatno ime - vse destinacije so vidne v Dodatku A in jih je 92). Če ima vsaka objava le svojo zemljepisno širino (*subject_lat*) in zemljepisno dolžino (*subject_lng*),



Slika 4.1: Diagrama zapisov v transakcijskem zapisu glede na geografsko lego za destinacijo Ljubljana: graf na levi zajema vse transakcije in je geografsko bolj razpršen, graf na desni zajema obdobje novoletnih praznikov, je manjši in ima večje število vozlišč blizu centra mesta.

ne pa tudi destinacije, lahko vsakemu transakcijskemu zapisu glede na geografsko pozicijo objave določimo turistično destinacijo. Lokacije (središča destinacij) in imena destinacij, v katere lahko spada objava, imamo podane v posebni datoteki ali pa jih dobimo s pregledom vseh objav (unikatna imena, povprečenje lokacij). Na grafu objav na sliki 4.1 so z modro prikazane objave (transakcijski zapisi) na območju Ljubljane. Večji radij kroga pomeni, da je bilo s posamezne lokacije več objav. Z rdečo so prikazana vozlišča, ki niso objave, ampak so le primeri znanih turističnih znamenitosti z njihovo geografsko lego za lažjo predstavo o legi na zemljevidu. Vse prikazane objave so transakcijski zapisi, ki imajo za destinacijo določeno Ljubljano, in tako predstavljajo isto vozlišče na grafu sočasnih pojavitev generiranem po opisani metodi v Podpoglavju 3.2.2 (na sliki 3.1).

Za vsako objavo imamo poleg lokacije objave shranjene tudi dodatne atribute, s pomočjo katerih lahko ločimo tip objave (kakšen tip destinacije je bil obiskan) in segmentiramo obiskovalce po starosti, spolu ali stilu potovanja. Dodatne atribute, skupaj z naborom vrednosti in pogostostjo, v podatkih prikazuje Tabela 4.1

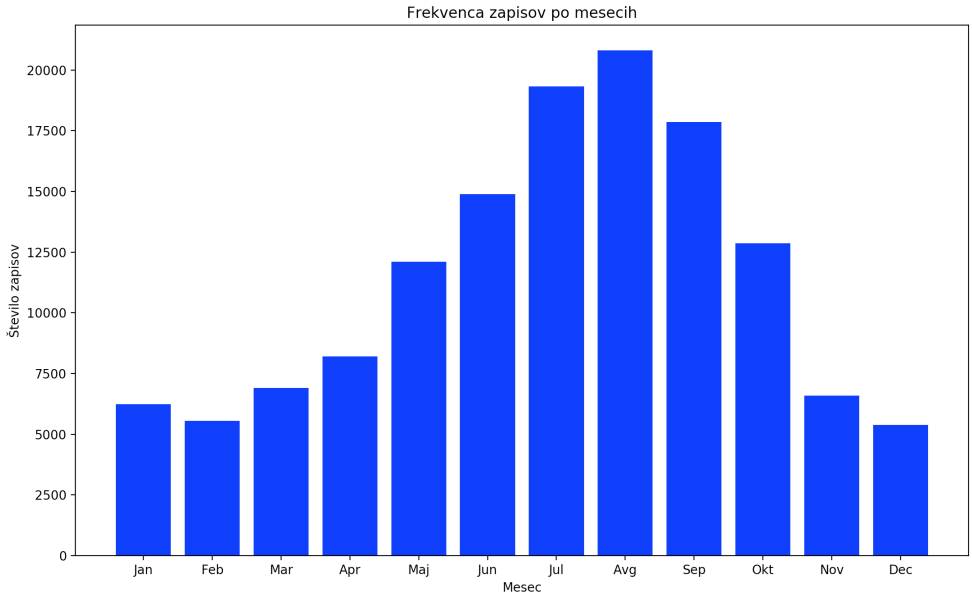
Tabela 4.1: Dodatni atributi, ki jih imamo na voljo v naših podatkih, nabor vrednosti in njihova frekvenca. Uporabnik lahko za user_travel_style označi več vrednosti (v stolpcu *Je enaka* se vidi, koliko uporabnikov izbere le eno).

Atribut (prevod)	**Nabor vrednosti (prevod ali obrazložitev)**	**Je enaka**	**Vsebuje vrednost**
subject_type (tip objave)	attractions (atrakcije)	44674	
	hotels (hoteli)	37628	
	restaurants (restavracije)	54422	
user_hometown (domači kraj uporabnika)	vse vpisane	111533	
	prazno	25191	
gender (spol)	F (female - ženski spol)	27947	
	M (male - moški spol)	35470	
	prazno	73307	
age (starost)	1 (13-17 let)	75	
	2 (18-24 let)	2654	
	3 (25-34 let)	13176	
	4 (35-49 let)	17655	
	5 (50-64 let)	14563	
	6 (65+ let)	4715	
	prazno	83886	
user_travel_style (stil potovanja)	60+ Traveler (Starejši popotnik)	1289	11918
	Art and Architecture Lover (Ljubitelj um. in arh.)	228	23798
	Backpacker (Popotnik z nahrbtnikom)	287	10039
	Beach Goer (Obiskovalec plaž)	174	17735
	Eco-tourist (Ekoturist)	79	9881
	Familiy Vacationer (Družinski počitnikovalec)	1937	18306
	Foodie (Pokuševalec hrane)	936	42179
	History Buff (Zgodovinski zagrizenec)	246	23662
	Like a Local (Kot lokalec)	1048	38168
	Luxury Traveler (Razkošni popotnik)	482	16070
	Nature Lover (Ljubitelj narave)	583	34469
	Nightlife Seeker (Iskalec nočnega življenja)	25	7214
	Peace and Quiet Seeker (Iskalec miru in tišine)	457	23276
	Shopping Fanatic (Nakupovalni fanatik)	30	8648
	Thrifty Traveler (Varčni popotnik)	342	15488
	Thrill Seeker (Iskalec vznemirjenja)	429	16154
	Trendsetter (Določevalec trendov)	172	7477
	Urban Explorer (Raziskovalec mest)	615	32652
	Vegetarian (Vegetarijanec)	74	4936
	prazno	54645	

4.2 Primerjava statistik

Sledi primerjava najdenih uradnih statistik slovenskega turizma s statistikami na naših podatkih. Ker imamo v naših podatkih prisotna tudi enodnevna potovanja, pričakujemo razlike. Preverimo najprej, kakšna je povprečna doba bivanja turistov v Sloveniji, in kakšna je dolžina trajanja potovanj obiskovalcev v naših podatkih. Povprečna doba bivanja v Sloveniji po [2] je 2,5 dneva. Izračun na naših podatkih nam vrne, da potovanje v povprečju traja 1,25 dneva. Pričakovano je, da je trajanje potovanj krajše od bivanja turistov, saj medtem ko statistika potovanj zajema tudi enodnevna potovanja, v statistiki bivanj turistov ta sploh niso vključena. Krajše trajanje je posledica tega, da je večina potovanj enodnevnih (77986), sledijo dvodnevna (2142) in trodnevna (1401). Število potovanj postopoma pada z večanjem dni, potovanj s trajanjem štirih dni je že manj kot 1000. Če za izračun povprečnega trajanja upoštevamo le potovanja, katerih časovni okvir skupno ne presega enega tedna, je ta še malo manjša - 1,23 dneva. Kot smo že omenili, je slabost našega pristopa, da moramo definirati, katere obiske bomo šteli v eno potovanje uporabnika.

Število prihodov turistov ni enakomerno, ampak se razlikuje po mesecih. Vira [3, 46] poročata, da rezidenti EU raje potujejo v poletnih mesecih - leta 2017 je bilo skoraj eno od štirih potovanj opravljenih v juliju ali avgustu. Tudi na sliki 4.2, ki prikazuje razporeditev števila obiskov po mesecih na naših podatkih, je prisoten vrh v juliju in avgustu.

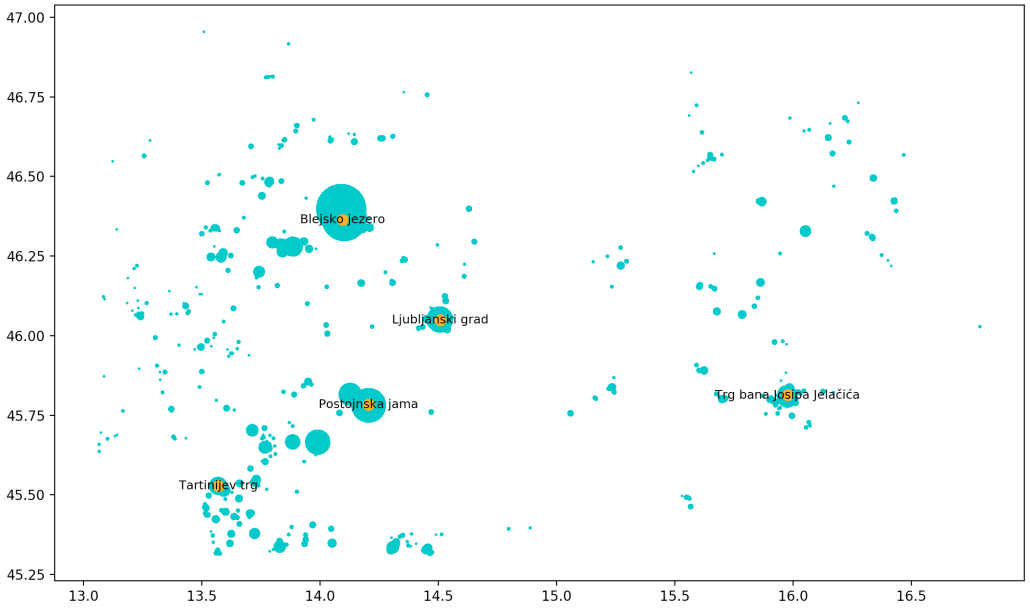


Slika 4.2: Stolpični graf prikazuje število objav glede na mesec v letu. Vsaka objava pomeni en obisk turistične znamenitosti. Vidimo, da je največ objav oziroma obiskov v poletnem letnem času.

Poglejmo statistiko po regijah Slovenije. Največ prihodov turistov je bilo leta 2017 zabeleženih v gorenjski regiji, največ njihovih prenočitev pa v obalno-kraški statistični regiji [47]. Srednje močne regije po številu prenočitev so osrednje-slovenska, savinjska in pomurska (več kot 900000 prenočitev) [1]. Občine z največjim številom prenočitev so Piran, Ljubljana in Bled [2].

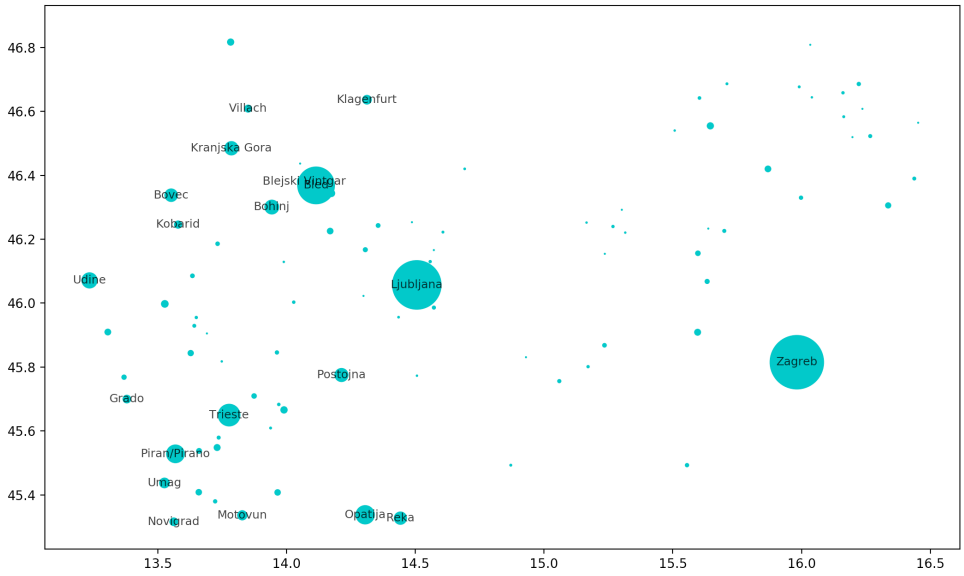
Geografska razporeditev objav oziroma obiskov v naših podatkih, ki niso združene v destinacije (občine ali kraje), je vidna na sliki 4.3. Na sliki 4.4 je prikazana številčnost objav, združenih po destinacijah, kot jih uporabljamo v grafu sočasnih pojavitev, na sliki 4.5 pa številčnost prenočitev po destinacijah iz SURS (občino Gorje smo preimenovali v najbolj znan tam ležeč kraj - Blejski vintgar). Število in velikost krogov, ki je sorazmerna številu objav, tudi na naših podatkih kaže večje število obiskov v gorenjski in obalno-kraški regiji, kar kaže, da je v teh dveh regijah veliko znamenitosti. Tako po številu objav kot številu prenočitev so najmočnejše destinacije Ljubljana, Bled in Piran, ter srednje močne Bohinj in Kranjska Gora. Videti je, da imajo obiskovalci v naših podatkih podobno razporeditev obiskov kot turisti prenočitve. Med močnimi destinacijami izstopa Blejski Vintgar, saj je po številu objav na tretjem mestu, po številčnosti prenočitev pa ga ni med prvimi dvajsetimi - glavna razloga sta, da ima bližnji Bled veliko večjo kapaciteto prenočišč in je Blejski Vintgar pozimi uradno zaprt. SV del Slovenije je relativno slabo pokrit z objavami, vendar je to večinoma zdraviliški turizem, kjer očitno prevladujejo gostje (npr. lokalni gosti, mlade družine, upokojenci), ki manj uporabljajo TripAdvisor. Izstopa primorsko-notranjska regija, ki je med bolje obiskanimi, a nima velikega števila prenočitev (Postojna, Sežana) - očitno gostje tja hodijo samo na ogled nekaj znamenitosti (v tranzitu ali na dnevni izlet), prenočijo pa raje drugje. JV del Slovenije ima tako malo število objav kot prenočitev.

Obe sliki objav (4.3 in 4.4) kažeta tudi dodatne informacije, saj vključujemo tudi obiske blizu meje Slovenije. Sliki nakazujeta povezanost potovanj s sosednjimi državami - po velikem številu objav še posebej izstopa Hrvaško mesto Zagreb. V podatkih s TripAdvisorja imamo sicer destinacije izven Slovenije,

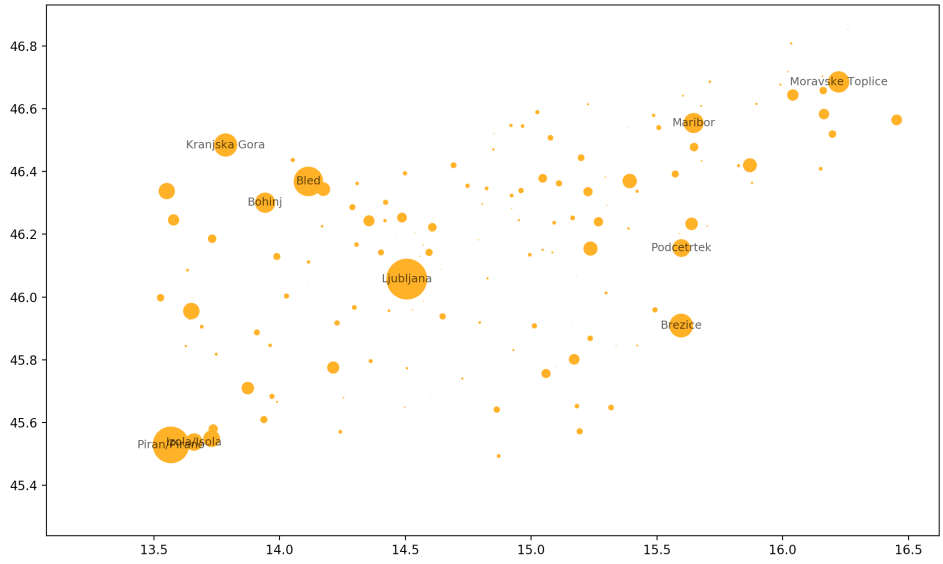


Slika 4.3: Prikaz transakcijskih zapisov oziroma obiskov glede na lokacijo objave. Večji polmer kroga pomeni večje število objav z določene lokacije. Z rumeno je za referenco označenih nekaj najbolj obiskanih znamenitosti. Slika nakazuje povezanost potovanj s sosednjimi državami Slovenije.

a če primerjamo samo destinacije znotraj Slovenije lahko sklepamo, da sta številčno razmerje po destinacijah in razpored objav v primerjavi s tistimi od prenočitev podobni z nekaj razlikami. Razlike so predvsem na destinacijah v vzhodni Sloveniji (kraji s toplicami npr. Brežice, Moravske Toplice).

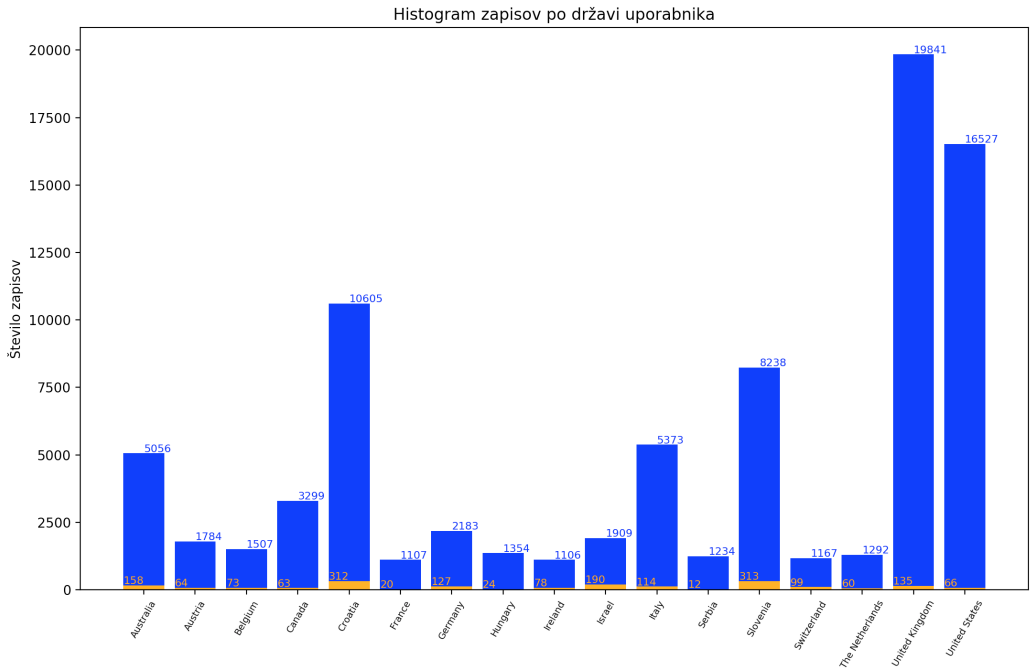


Slika 4.4: Slika prikazuje število objav združenih po destinaciji. Vidnost imena destinacije je omejeno na 20 destinacij z največ objavami.



Slika 4.5: Slika prikazuje število nočitev turistov združenih po destinacijah iz SURS (podatki so le za Slovenijo). Vidnost imena destinacije je omejeno na 10 destinacij z največ prenočitvami.

Preverimo še statistiko po državah, iz katerih prihajajo obiskovalci. Prihodov tujih turistov je približno dvakrat več kot domačih in kaže trend zviševanja [2]. Po prihodih je največ tujih turistov iz Italije, Nemčije in Avstrije (vsaka več kot 300000 od leta 2016 [1]), sledijo še iz Hrvaške, Republike Koreje, Nizozemske, Velike Britanije in Madžarske. Ker lahko uporabniki vnesejo poljubno tekstovno vrednost za državo, je primerjava statistike smiselna le na predprocesiranih podatkih, kjer imamo že dodatno določene države. Uporabniki velikokrat ne navajajo (samo) države, ampak dežele, iz katerih prihajajo, in kratice držav (na naših podatkih najbolj pogosto pri Američanih in Britancih), država ima lahko več različnih nazivov (Združeno kraljestvo - *United Kingdom* in *Great Britain*) ali je vnesena v maternem jeziku (pogosto pri Italijanih). Graf na sliki 4.6 prikazuje število objav glede na domačo državo obiskovalca. Povprečno število objav na turista po državah je podobno - v povprečju vsak turist objavi vsaj dvakrat, manj objavljajo Italijani, Avstrijci in Madžari. Vidimo, da je statistika po državah porekla obiskovalcev na naših podatkih precej drugačna od uradne statistike, saj največ uporabnikov prihaja iz Združenega kraljestva in Združenih držav Amerike. Razlog je najverjetneje v tem, da je večina objav na TripAdvisorju v angleščini in je to njihov materni jezik, pa tudi v priljubljenosti uporabe TripAdvisorja, ki se razlikuje od države do države. Predvidevamo, da ima najvišjo priljubljenost med prebivalci angleško govorečih držav in da ga domači gostje uporabljajo manj kot tuji.



Slika 4.6: Slika prikazuje število objav glede na domačo državo obiskovalca s spodnjo mejo 1000. Največ objav naredijo prebivalci Velike Britanije, Združenih držav Amerike in Hrvaške. Povprečno število objav na turista po državah je podobno. Z rumeno je prikazan stolpični graf brez predprocesiranja podatkov - brez ugotavljanja države iz uporabnikovega poljubnega vnosa. Brez predprocesiranja ne bi mogli izvajati analize s segmentacijo obiskovalcev po državi porekla, saj bi imeli premajhno število podatkov.

Poglavje 5

Analiza

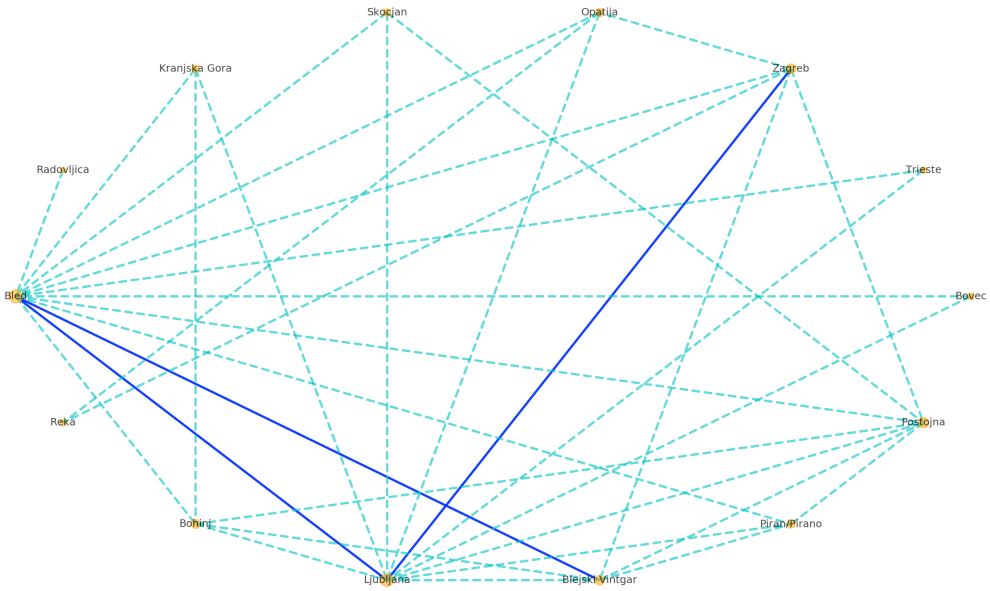
Raziščimo, kakšen graf dobimo metodi opisani v Podpoglavju 3.2.2. Graf, zgeneriran iz vseh transakcijskih zapisov, na sliki 3.1 je gost in nepregleden. Vsebuje 90 vozlišč in 1114 povezav. Na sliki 5.1 smo obdržali le povezave, ki imajo utež vsaj 100. Graf je zdaj lažje berljiv, saj vsebuje 14 vozlišč in 34 povezav, katerim se jasno vidijo končna vozlišča. Manjši graf kaže na to, da bo v analizi pomembno filtriranje povezav po uteži. Na grafu, ki ima vozlišča pozicionirana glede na geografsko lokacijo, ni jasno vidna močna povezava med Bledom in Ljubljano, prav tako je zakrita močna povezava med Bledom in Blejskim vintgarjem. Odkrili smo, da nam geografska lokacija lahko pomaga pri lažji predstavi, vendar lahko pokvari vidnost grafa. Ker za samo iskanje pogostih košaric ni pomembna, je bolje vozlišča (destinacije) risati brez ozira na to, da imajo geografsko lokacijo in s poudarkom na vidnost. Na ta način analizo tudi približamo splošnemu problemu nakupovalne košarice, kjer imamo namesto obiskov destinacij opraviti z izdelki. Šele ob zaključku analize pri razlagi rezultatov je smiselno preveriti tudi vpliv geografske lokacije in si pomagati z grafom, ki ima glede na njo pozicionirana vozišča.

Na sliki 5.2 je predstavljen histogram stopenj vozlišč, s katerim bomo lažje raziskali povezanost grafa iz vseh transakcijskih zapisov. Vidimo, da ima vsako vozlišče res najmanj enega soseda (tako mora biti, ker smo pri

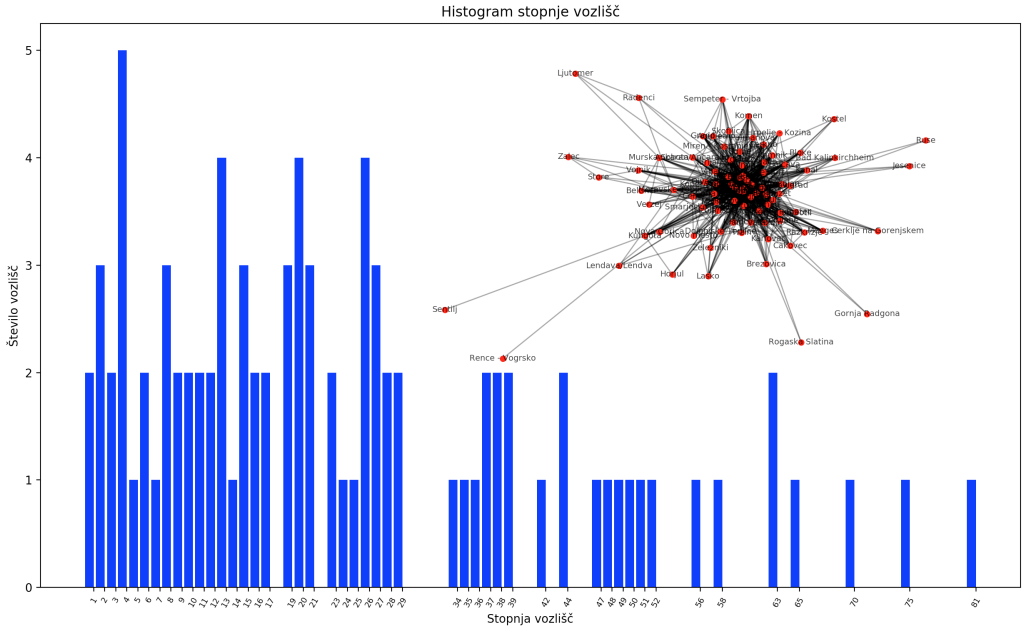
Tabela 5.1: Vozlišča, ki imajo več kot 60 sosednjih vozlišč.

destinacija	Ljubljana	Bled	Zagreb	Piran/Pirano	Bohinj	Postojna
št. sosednjih vozlišč	81	75	70	65	63	63

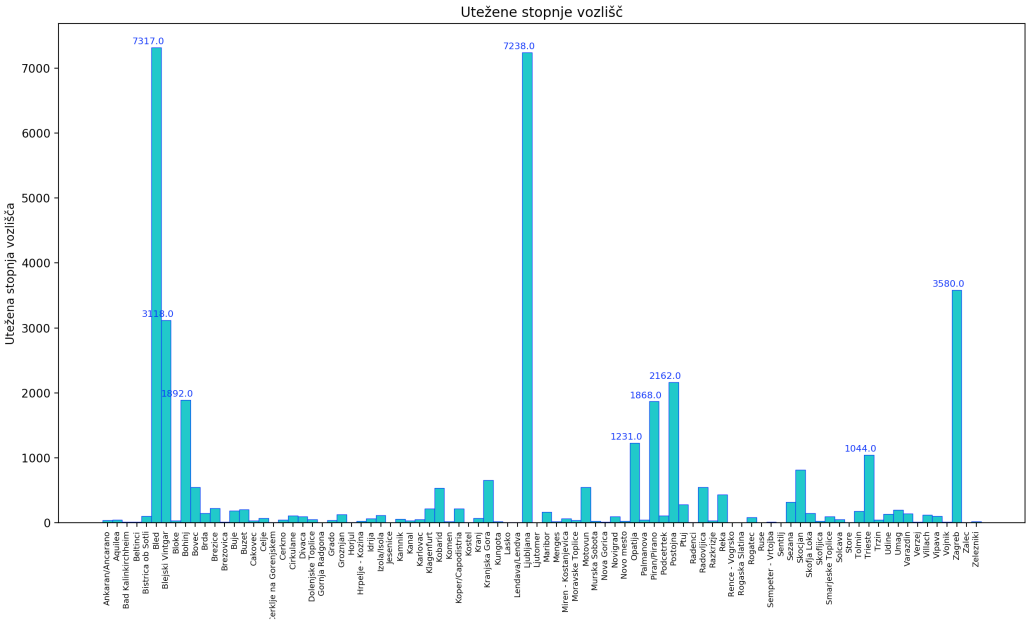
generiranju grafa izločili potovanja s samo eno destinacijo). V Tabeli 5.1 so prikazana vozlišča z najvišjimi stopnjami in sosedi. Tabela nakazuje, katere destinacije se v različnih potovanjih največkrat pojavljajo (skladno z velikostjo vozlišč na predstavitvi grafa). Povprečna stopnja vozlišč je 24,76, mediana je 20 in modus (najpogostejša vrednost) je 4. Najvišjo stopnjo ima Ljubljana, ki ima 81 sosedov (z njo nima nobene povezave le 9 vozlišč), najnižjo oz. le enega soseda imata Šentilj (sosed Kungota) in Rence - Vogrsko (sosed Nova Gorica). Utežena stopnja vozlišča predstavlja seštevek vseh uteži na povezavah vozlišča. Slika 5.3 prikazuje uteženo stopnjo vozlišča za vse destinacije. Najvišjo imata Ljubljana in Bled, kar je glede na statistiko, predstavljeno v prejšnjem poglavju, pričakovano. Najvišjo uteženo stopnjo ima Bled in ne Ljubljana, kar kaže, da Bled nastopa v več potovanjih. Piran je po uteženi stopnji na sedmem mestu in ni uvrščen tako visoko, kot v statistiki prenočitev - videti je, da nastopa še v manj potovanjih kot Zagreb, Blejski Vintgar, Postojna in Bohinj. V Dodatku A lahko za katerikoli dve destinaciji preverimo, ali sta ti povezani (če imata vrednost večjo od nič) in kakšne uteži so na vseh povezavah.



Slika 5.1: Sočasne pojavitve s spodnjo mejo uteži vsaj 100. Vozlišča so pozicionirana v koncentričnem krogu (z uporabo algoritma *shell_layout* iz NetworkX). Velikost vozlišča in vrste povezave so definirane enako kot na polnem grafu sliki 3.1. Zdaj se jasno vidijo tri močne povezave: Bled - Blejski Vintgar, Bled - Ljubljana, Ljubljana - Zagreb.



Slika 5.2: Stopnje vozlišča grafa, generiranega iz vseh transakcij. Večina vozlišč ima stopnjo manjšo od 30. Posamezno vozlišče ima najpogosteje 4 sosede. Vozlišče z najvišjo stopnjo je Ljubljana.



Slika 5.3: Utežene stopnje vozlišča grafa, generiranega iz vseh transakcij. Višja utežena stopnja vozlišča pomeni zastopanost v več potovanjih, saj je ta seštevek uteži na povezavah vozlišča s sosednjimi vozlišči.

V nadaljevanju bomo najprej predstavili analizo grafa, generiranega iz vseh zapisov z relevantnimi metodami iz Poglavja 2, in primerjali rezultate. Nato bomo analizirali še grafe s filtriranjem po atributih oziroma grafe s segmentacijo po različnih atributih za uporabnika, ki jih imamo na voljo.

5.1 Primerjava metod za analizo na grafu

Že z izrisom grafa, generiranega iz vseh zapisov, lahko sklepamo, katere bodo najpomembnejše destinacije v najpogostejših potovanjih, vendar če želimo razbrati, kakšna potovanja oziroma skupina obiskov so najpogostejši, samo izris grafa ni dovolj. Po generiranju grafa moramo zato uporabiti vključene dodatne algoritme za odkrivanje pogostih množic: Apriori, Louvain in Infomap. Sledi analiza grafa z naštetimi metodami v enakem vrstnem redu in na koncu primerjava vseh treh metod.

Algoritem Apriori podpira tako generiranje povezovalnih pravil kot seznama najpogostejših množic, pri čemer se bomo osredotočili predvsem na slednje. Najprej iz generiranega grafa vseh zapisov zapišemo tekstovno datoteko, v kateri vsaka vrstica sestoji iz naštetih obiskanih destinacij v posameznem potovanju, tudi z le enim obiskom. Algoritem (razred *AssociationRulesInducer*) nato poganjamo z vhodnimi podatki (razpršena matrika, ki združuje podatke iz tekstovne datoteke), podporo in zaupanjem. Nastavitev zaupanja ne vpliva na seznam najpogostejših množic, saj ima vsaka množica samo svojo podporo, medtem ko ima povezovalno pravilo podporo enako kot množica, iz katere sestoji, in svoje zaupanje. Ker iščemo potovanja z več obiski, nas zanimajo samo pogoste množice, ki imajo več kot en element. Tabela 5.2 predstavlja rezultate za nekaj različnih vrednosti. Vidimo, da število pogostih množic narašča hitreje kot število povezovalnih pravil. Najhitreje, pri vrednosti 0.025, dobimo najpogostejšo množico *Ljubljana Bled* s podporo 0.026. To pomeni, da število potovanj, v katerih sta prisotna Ljubljana in Bled, delimo s številom vseh potovanj. Najmočnejše pravilo: *Blejski vintgar* $\Longrightarrow$ *Bled* ima podporo 0.017 in zaupanje 0.65 (da Apriori vrne to

Tabela 5.2: Algoritem Apriori za manjše parametre vrača več najpogostejših množic in povezovalnih pravil.

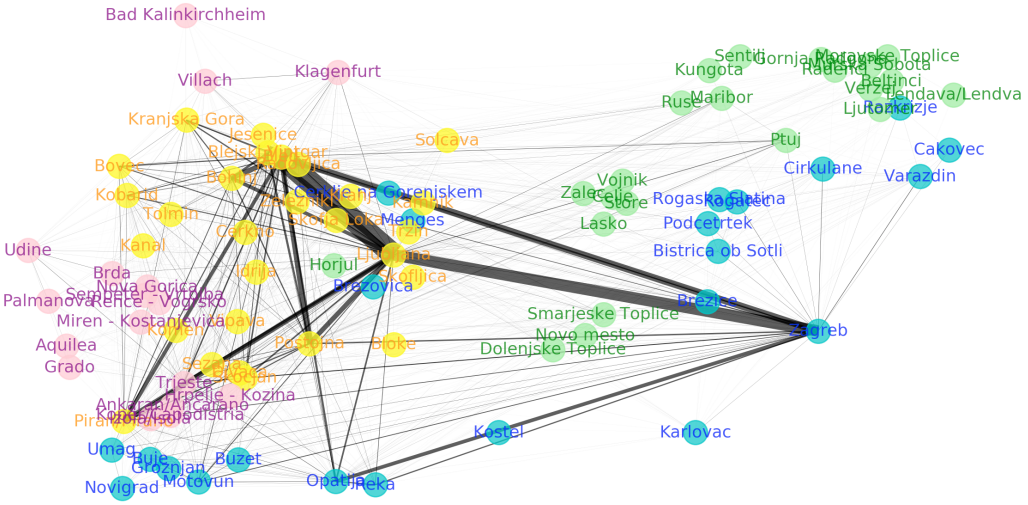
Število množic Podpora Zaupanje	Št. množic z vsaj 2 elementoma (Podpora) Množica z vsaj 2 elem.	(Podpora, Zaupanje) Povezovalno pravilo
9 0.025 0.5	1 (0.026) Ljubljana Bled	/
15 0.015 0.5	2 (0.017) Blejski vintgar Bled (0.026) Ljubljana Bled	(0.017, 0.650) Blejski vintgar $\Longrightarrow$ Bled
18 0.01 0.5	3 (0.017) Blejski vintgar Bled (0.026) Ljubljana Bled (0.014) Zagreb Ljubljana	(0.017, 0.650) Blejski vintgar $\Longrightarrow$ Bled
34 0.005 0.2	9 (0.006) Bohinj Bled (0.017) Blejski vintgar Bled (0.026) Ljubljana Bled (0.006) Ljubljana Blejski vintgar (0.006) Piran/Pirano Ljubljana (0.006) Postojna Bled (0.006) Postojna Ljubljana (0.008) Zagreb Bled (0.014) Zagreb Ljubljana	(0.006, 0.233) Blejski vintgar $\Longrightarrow$ Ljubljana (0.017, 0.650) Blejski vintgar $\Longrightarrow$ Bled (0.006, 0.250) Bohinj $\Longrightarrow$ Bled (0.006, 0.275) Postojna $\Longrightarrow$ Ljubljana (0.006, 0.286) Postojna $\Longrightarrow$ Bled

pravilo, moramo podati argumenta, ki sta manjša od obeh vrednosti). V Tabeli 5.3 so prikazani rezultati z nizko podporo in zaupanjem. Če imamo zelo veliko vhodnih podatkov se nam pri nižanju nastavitve podpore lahko zgodi, da ostanemo brez delovnega spomina. Če v vhodno tekstovno datoteko zapišemo le potovanja, ki imajo vsaj 2 obiska ali več, podpora naraste - za najpogostejšo množico *Ljubljana Bled* na 0.242 - to vrednost podpore lahko izračunamo tudi direktno iz grafa in sicer tako, da utež na povezavi na grafu delimo s številom vseh potovanj.

Tabela 5.3: Rezultat algoritma Apriori za podporo 0.001 in zaupanje 0.1.

Število množic	Podpora Zaupanje	Št. množic dolžine vsaj 2 (Podpora) Množica z več kot 1 elementom	(Podpora, Zaupanje) Povezovalna pravila
111	0.001 0.1	55 (0.006) Bohinj Bled (0.001) Bovec Bled (0.017) Blejski vintgar Bled (0.002) Blejski vintgar Bohinj Bled (0.003) Blejski vintgar Bohinj (0.001) Kobarid Bled (0.002) Kranjska Gora Bled (0.001) Kranjska Gora Bohinj (0.026) Ljubljana Bled (0.002) Ljubljana Bohinj Bled (0.004) Ljubljana Bohinj (0.001) Ljubljana Bovec (0.005) Ljubljana Blejski vintgar Bled (0.001) Ljubljana Blejski vintgar Bohinj Bled (0.001) Ljubljana Blejski vintgar Bohinj (0.006) Ljubljana Blejski vintgar (0.001) Ljubljana Kobarid (0.002) Ljubljana Kranjska Gora (0.001) Motovun Ljubljana (0.002) Opatija Bled (0.002) Opatija Ljubljana (0.004) Piran/Pirano Bled (0.001) Piran/Pirano Bohinj (0.001) Piran/Pirano Blejski vintgar (0.002) Piran/Pirano Ljubljana Bled (0.006) Piran/Pirano Ljubljana (0.006) Postojna Bled (0.001) Postojna Bohinj Bled (0.001) Postojna Bohinj (0.002) Postojna Blejski vintgar Bled (0.002) Postojna Blejski vintgar (0.003) Postojna Ljubljana Bled (0.001) Postojna Ljubljana Blejski vintgar Bled (0.001) Postojna Ljubljana Blejski vintgar (0.006) Postojna Ljubljana (0.001) Postojna Piran/Pirano (0.002) Radovljica Bled (0.001) Radovljica Ljubljana (0.001) Reka Opatija (0.002) Skocjan Bled (0.002) Skocjan Ljubljana (0.001) Skocjan Postojna (0.002) Trieste Bled (0.003) Trieste Ljubljana (0.001) Trieste Piran/Pirano (0.008) Zagreb Bled (0.001) Zagreb Blejski vintgar Bled (0.001) Zagreb Blejski vintgar (0.003) Zagreb Ljubljana Bled (0.014) Zagreb Ljubljana (0.001) Zagreb Motovun (0.004) Zagreb Opatija (0.002) Zagreb Postojna (0.001) Zagreb Reka (0.001) Zagreb Trieste	(0.017, 0.650) Blejski vintgar $\Longrightarrow$ Bled (0.006, 0.233) Blejski vintgar $\Longrightarrow$ Ljubljana (0.006, 0.250) Bohinj $\Longrightarrow$ Bled (0.006, 0.286) Postojna $\Longrightarrow$ Bled (0.006, 0.275) Postojna $\Longrightarrow$ Ljubljana (0.002, 0.293) Radovljica $\Longrightarrow$ Bled (0.002, 0.314) Skocjan $\Longrightarrow$ Bled (0.002, 0.310) Skocjan $\Longrightarrow$ Ljubljana (0.001, 0.212) Skocjan $\Longrightarrow$ Postojna

Nadaljujmo s predstavitvijo rezultatov Louvaina. Na sliki 5.4 imamo predstavljene štiri skupnosti, najdene z algoritmom Louvain na grafu vseh zapisov: rumena skupnost združuje osrednjo Slovenijo, Gorenjsko in del Primorske, zelena destinacije z zdraviliškim turizmom na vzhodu Slovenije, rdeča skupnost morske destinacije z izjemo Pirana in destinacije v Italiji in Avstriji, modra skupnost destinacije v Hrvaški. Nekatere destinacije blizu Ljubljane (Horjul, Mengeš, Brezovica) kljub centralni lokaciji ne pripadajo rumeni skupnosti. Graf prikazuje pomembnost tako geografske lokacije (sam algoritem jo zaznava preko sosednosti vozlišč) kot povezav med vozlišči z višjo stopnjo pri dodeljevanju v skupnosti. V Tabeli 5.4 so zapisane vse destinacije v posameznih skupnostih. V Tabeli 5.5 lahko vidimo, kako vpliva odstranjevanje nižjih uteži in posledično vozlišč z nižjo stopnjo na odkrivanje skupnosti. Vozlišče je tekom procesa odstranjevanja lahko v skupnostih z različnimi člani: Blejski vintgar ima močno povezavo z utežjo 1446 le z Bledom in se v vseh filtriranjih pojavi v isti skupnosti z njim, medtem ko ima Ljubljana več močnih povezav in je lahko v skupnosti z Bledom ali/in hrvaškimi destinacijami. Dodeljevanje Ljubljane v skupnosti z različnimi sosedi nakazuje centriranost obiskovanja obiskovalcev, pa tudi združitev rumene in modre skupnosti.



Slika 5.4: Grafična predstavitev rezultata, ki ga vrne algoritem Louvain na grafu vseh zapisov: vsaka barva vozlišča označuje svojo skupnost, debelina povezave nakazuje veličino uteži. Algoritem najde štiri skupnosti, ki kažejo pomembnost lokacije (sosednjih vozlišč na grafu). Pri določanju v skupnosti so zelo pomembne povezave med vozlišči z višjo stopnjo - meja skupnosti ni striktna (na primer Piran bi lokacijsko oziroma po sosedih bil uvrščen v skupnost rdeče barve, a je zaradi povezave z Ljubljano uvrščen v skupnost rumene barve).

Tabela 5.4: Rezultat algoritma Louvain na grafu vseh zapisov.

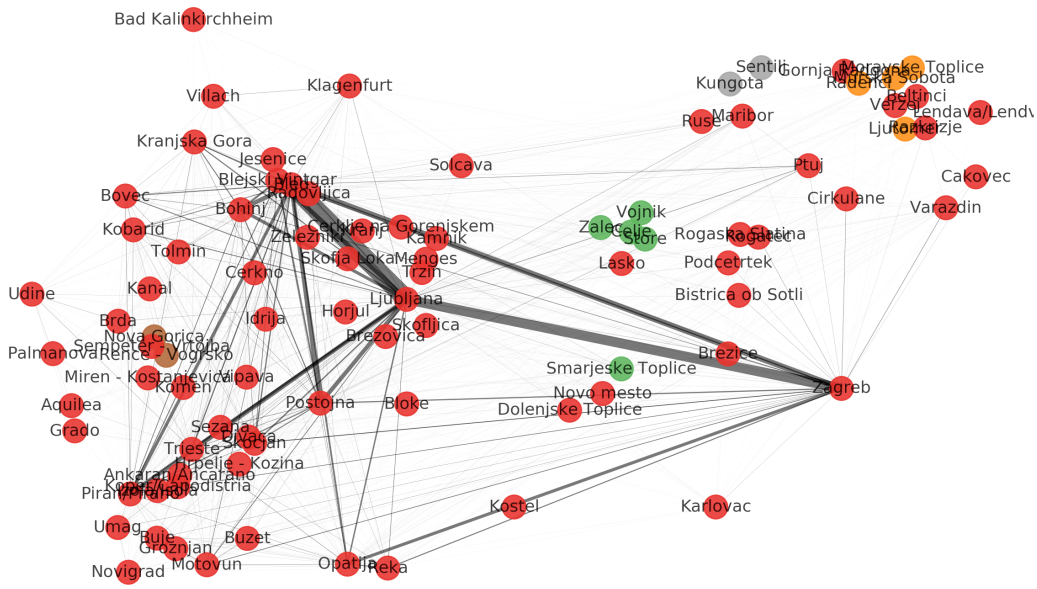
Spodnja meja uteži	Odkrite skupnosti	Razlaga/Komentar
1	Skupnost 1: 'Bistrica ob Sotli', 'Brezice', 'Brezovica', 'Buje', 'Buzet', 'Cakovec', 'Cerklje na Gorenjskem', 'Cirkulane','Groznjan', 'Karlovac', 'Kostel', 'Menges', 'Motovun', 'Novigrad', 'Opatija', 'Podcetrtek', 'Razkrizje', 'Reka','Rogaska Slatina', 'Rogatec', 'Umag', 'Varazdin', 'Zagreb' Skupnost 2: 'Bled', 'Blejski Vintgar', 'Bloke', 'Bohinj', 'Bovec', 'Cerkno', 'Divaca', 'Idrija', 'Jesenice', 'Kamnik', 'Kanal', 'Kobarid', 'Komen', 'Kranj', 'Kranjska Gora', 'Ljubljana', 'Piran/Pirano', 'Postojna', 'Radovljica', 'Sezana', 'Skocjan', 'Skofja Loka', 'Skofljica', 'Solcava', 'Tolmin', 'Trzin', 'Vipava', 'Zelezniki' Skupnost 3: 'Beltinci', 'Celje', 'Dolenjske Toplice', 'Gornja Radgona', 'Horjul', 'Kungota', 'Lasko', 'Lendava/Lendva', 'Ljutomer', 'Maribor', 'Moravske Toplice', 'Murska Sobota', 'Novo mesto', 'Ptuj', 'Radenci', 'Ruse', 'Sentilj', 'Smarjeske Toplice', 'Store', 'Verzej', 'Vojnik', 'Zalec' Skupnost 4: 'Ankaran/Ancarano', 'Aquilea', 'Bad Kalinkirchheim', 'Brda', 'Grado', 'Hrpelje - Kozina', 'Izola/Isola', 'Klagenfurt', 'Koper/Capodistria', 'Miren - Kostanjevica', 'Nova Gorica', 'Palmanova', 'Rence - Vogrsko', 'Sempeter - Vrtojba', 'Trieste', 'Udine', 'Villach'	Na celem grafu so odkrite 4 skupnosti, ki večinoma združujejo destinacije po lokaciji. Italija (skupaj z bližnjimi destinacijami v Sloveniji) in Avstrija sta združeni v eno skupnost - Skupnost 4.

Tabela 5.5: Rezultat algoritma Louvain pri filtriranju z odstranjevanjem nižjih uteži. Vidimo, kako na odkrite skupnosti vpliva odstranjevanje nižjih uteži in posledično vozlišč z nižjo stopnjo. Vozlišče je tekom procesa odstranjevanja lahko v skupnostih z različnimi člani:

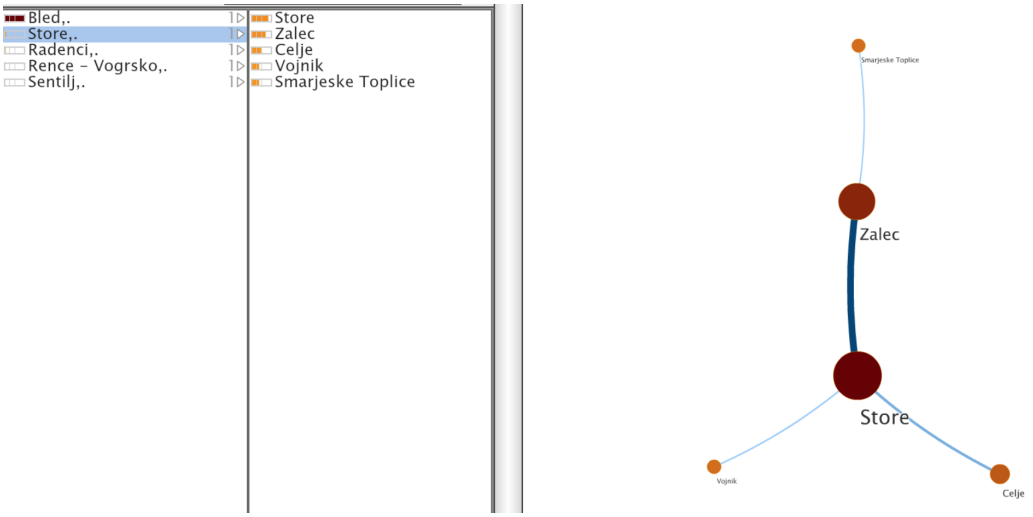
Spodnja meja uteži	Odkrite skupnosti	Razlaga/Komentar
10	Skupnost 1: 'Bled', 'Blejski Vintgar', 'Bohinj', 'Bovec','Celje', 'Divaca', 'Dolenjske Toplice', 'Idrija', 'Izola/Isola', 'Kobarid', 'Koper/Capodistria', 'Kranj', 'Kranjska Gora', 'Ljubljana', 'Maribor', 'Piran/Pirano', 'Postojna', 'Ptuj', 'Radovljica', 'Sezana', 'Skocjan', 'Skofja Loka', 'Skofljica', 'Tolmin', 'Trzin', 'Vipava', 'Zelezniki' Skupnost 2: 'Ankaran/Ancarano', 'Aquilea', 'Brda', 'Miren - Kostanjevica', 'Palmanova', 'Trieste', 'Udine' Skupnost 3: 'Bistrica ob Sotli', 'Brezice', 'Buje', 'Buzet', 'Cakovec', 'Cirkulane', 'Groznjan', 'Karlovac', 'Motovun', 'Novigrad', 'Opatija', 'Podcetrtek', 'Razkrizje', 'Reka', 'Rogatec', 'Smarjeske Toplice', 'Umag', 'Varazdin', 'Zagreb' Skupnost 4: 'Klagenfurt', 'Villach'	Prva skupnost v primerjavi s prej poveča lokacijski obseg, dobimo ločene skupnosti, ki pokrivajo Italijo in Avstrijo.
20	Skupnost 1: 'Bled', 'Blejski Vintgar', 'Bohinj', 'Bovec', 'Izola/Isola', 'Kobarid', 'Koper/Capodistria', 'Kranj', 'Kranjska Gora', 'Ljubljana', 'Maribor', 'Piran/Pirano', 'Postojna', 'Ptuj', 'Radovljica', 'Sezana', 'Skocjan', 'Skofja Loka', 'Smarjeske Toplice', 'Tolmin', 'Trzin', 'Vipava' Skupnost 2: 'Bistrica ob Sotli', 'Brda', 'Brezice', 'Buje', 'Buzet', 'Cirkulane', 'Groznjan', 'Motovun', 'Opatija', 'Reka', 'Rogatec', 'Trieste', 'Udine', 'Umag', 'Varazdin', 'Zagreb' Skupnost 3: 'Klagenfurt', 'Villach'	Primorska je razdeljena na dve regiji, saj se nahaja v dveh skupnosti. Od štajerske regiije sta na grafu prisotna le še Maribor in Ptuj, ki sta priključna k največji skupnosti (Skupnost 1).
50	Skupnost 1: 'Bled', 'Blejski Vintgar', 'Bohinj', 'Bovec', 'Kranjska Gora', 'Postojna', 'Radovljica', 'Sezana', 'Skocjan' Skupnost 2: 'Brezice', 'Kobarid', 'Ljubljana', 'Motovun', 'Opatija', 'Piran/Pirano', 'Ptuj', 'Reka', 'Trieste', 'Varazdin', 'Zagreb'	Ljubljana se poveže z močnejšimi mesti in odcepi od Gorenjske.
90	Skupnost 1: 'Bled', 'Blejski Vintgar', 'Bohinj', 'Bovec', 'Kobarid', Kranjska Gora', 'Ljubljana', 'Piran/Pirano', 'Postojna', 'Radovljica', 'Skocjan' Skupnost 2: 'Opatija', 'Reka', 'Trieste', 'Zagreb'	Nastane močnejša povezava med destinacijami Hrvaške in Trstom. Vse slovenske destinacije so združene v eni skupnosti.
120	Skupnost 1: 'Kranjska Gora', 'Ljubljana', 'Opatija', 'Piran/Pirano', 'Reka', 'Skocjan', 'Trieste', 'Zagreb' Skupnost 2: 'Bled', 'Blejski Vintgar', 'Bohinj', 'Postojna', 'Radovljica'	Po izločitvi povezave med Opatijo in Reko so hrvaške destinacije priključene slovenskim.
280	Skupnost 1: 'Opatija', 'Zagreb' Skupnost 2: 'Bled', 'Blejski Vintgar', 'Bohinj', 'Ljubljana', 'Piran/Pirano', 'Postojna'	Po izločitvi Reke dobimo skupnost Opatija, Zagreb.
370	Skupnost 1: 'Bled', 'Blejski Vintgar', 'Bohinj', 'Ljubljana', 'Piran/Pirano', 'Postojna', 'Zagreb'	Po izključitivi Opatije dobimo eno skupnost.
380	Skupnost 1: 'Ljubljana', 'Piran/Pirano', 'Postojna', 'Zagreb' Skupnost 2: 'Bled', 'Blejski Vintgar', 'Bohinj'	Odstranitev povezave Bohinj-Ljubljana okrepi severni trikotnik: Skupnost 2.
490	Skupnost 1: 'Ljubljana', 'Piran/Pirano', 'Zagreb' Skupnost 2: 'Bled', 'Blejski Vintgar', 'Bohinj', 'Postojna'	Spet sta najdeni dve skupnosti. Postojna je povezana samo še z Bledom, njena povezava z Ljubljano je odpadla, zato je premeščena v Skupnost 2.
500	Skupnost 1: 'Bled', 'Blejski Vintgar', 'Ljubljana', 'Piran/Pirano', 'Zagreb'	Po odstranitvi Bohinja je ne glede na večanje uteži najdena le ena skupnost.

Za analizo z algoritmom Infomap bomo zaradi interaktivnosti uporabljali spletno aplikacijo Network Navigator (skrajšano jo bomo klicali kar Navigator), dosegljivo na [48]. Na sliki 5.5 preverimo rezultat na grafu vseh zapisov in primerjamo izris skupnosti. Algoritem odkrije pet skupnosti, ki se razlikujejo od skupnosti, najdenih z algoritmom Louvain. Navigator poimenuje skupnosti po najmočnejšem vsebovanem vozlišču/destinaciji z največ toka. Skupnost rdeče barve oziroma skupnost Bled je izrazito večja, zato jo bomo imenovali glavna skupnost. Na sliki 5.6 prikazujemo glavno skupnost filtrirano z 10 vozlišči (18 najmočnejšimi povezavami po toku) in z 20 vozlišči (53 povezav). Vozlišča se pri večanju števila povezav samodejno dodajajo in razporejajo. Vsako vozlišče je mogoče premikati in tako vplivati na razpored vozlišč, medtem ko se struktura grafa ohranja (s premikom enega vozlišča vplivamo v zmanjšani meri na videz celega grafa).

Raziščemo lahko tudi, kako se vozlišča dodajajo na graf, če štartamo z vozliščem, ki ima največji pretok (Bled). Najprej se skupnosti priključi Ljubljana, nato Blejski Vintgar, Zagreb, Piran, Bohinj, Postojna, Opatija, Trst, Škocjan, Radovljica, Kranjska Gora, Reka, Bovec, Kobarid itd. Vidimo, da se povezave dodajajo v enakem vrstnem redu kot najdene množice urejene po podpori, ki jih generira algoritem Apriori. Infomap in Louvain najdeta različne skupnosti, ker sta konceptualno različna (njuno iskanje skupnosti je razloženo v 3.1). Louvain nam v primerjavi z Infomapom vrne zelo drugačne skupnosti, ki pa so prevelike, da bi lahko iz njih sklepali zaključke o skupnih kampanjah, in so bolj spremenljive.

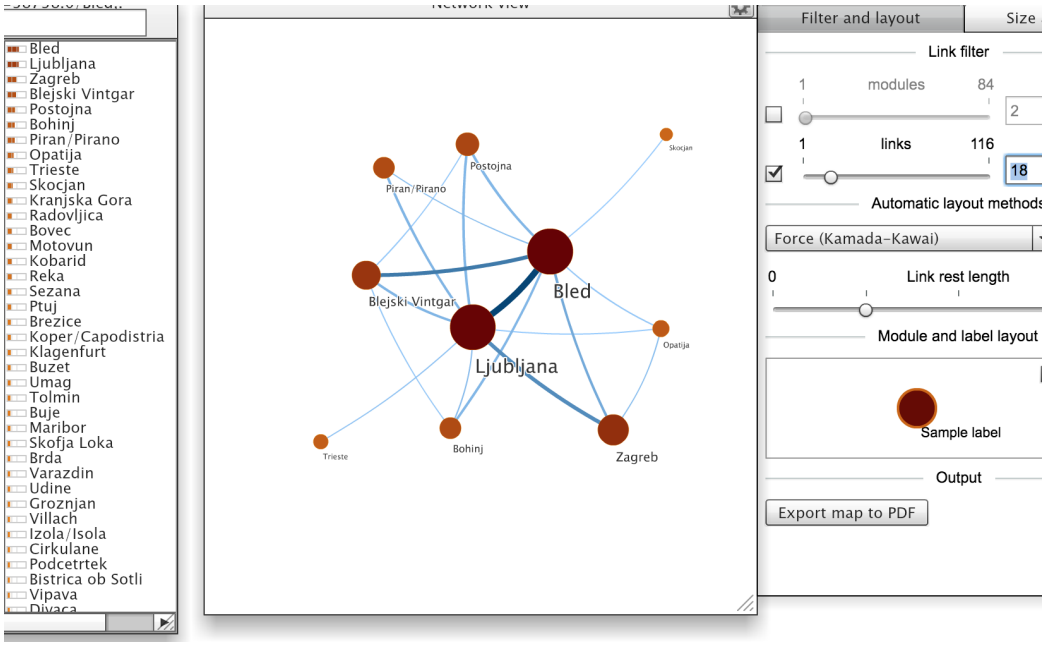


(a) Grafična predstavitev rezultata, ki ga vrne algoritem Infomap v naši programski rešitvi: vsaka barva vozlišča označuje svojo skupnost.

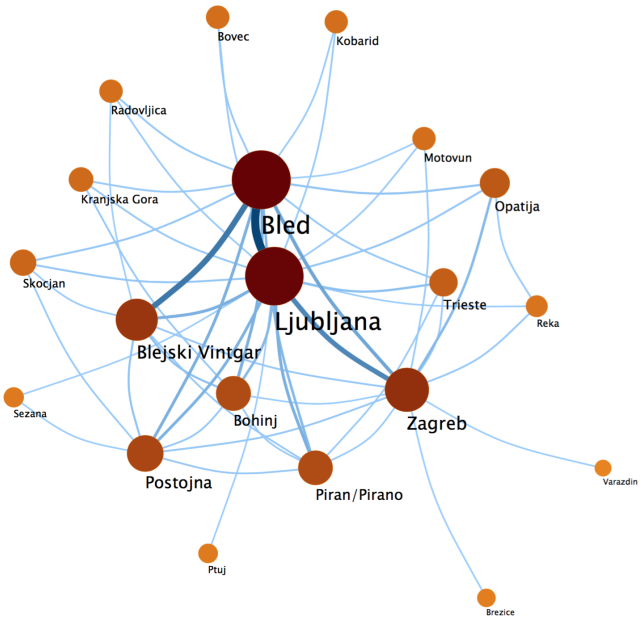


(b) Predstavitev rezultata v Network Navigatorju: levo so naštete najdene skupnosti, po kliku na Store se v sredini izpišejo člani skupnosti Store, desno se prikaže njen graf.

Slika 5.5: Algoritem Infomap na grafu vseh zapisov glede na pretok najde pet skupnosti, od katerih je ena izrazito večja (glavna skupnost). Skupnost zelene barve na (a) je skupnost Store na (b). Debelina povezave je na (a) glede na veličino uteži, na (b) glede na izračunan tok.



(a) Glavna skupnost s prikazanimi 10 vozlišči (levo seznam članov, v sredini graf). Filtriranje je možno glede na pretok povezav in/ali vozlišč (desno).



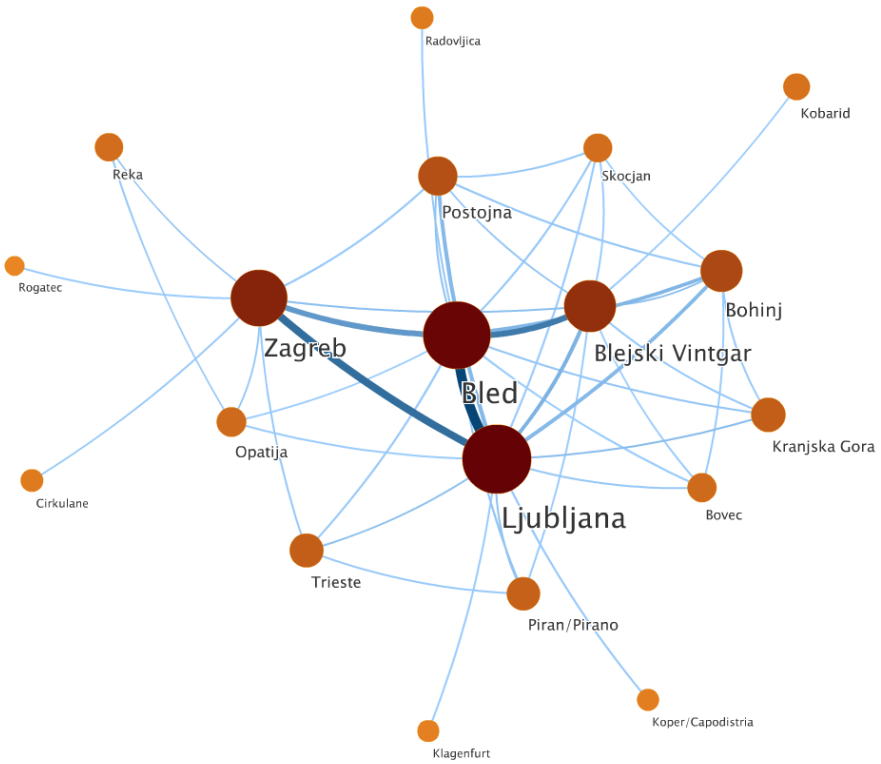
(b) Glavna skupnost s prikazanimi 20 vozlišči.

Slika 5.6: Prikaz filtriranja. Graf na (b) ima dvakrat več vozlišč kot na (a) in trikrat več povezav.

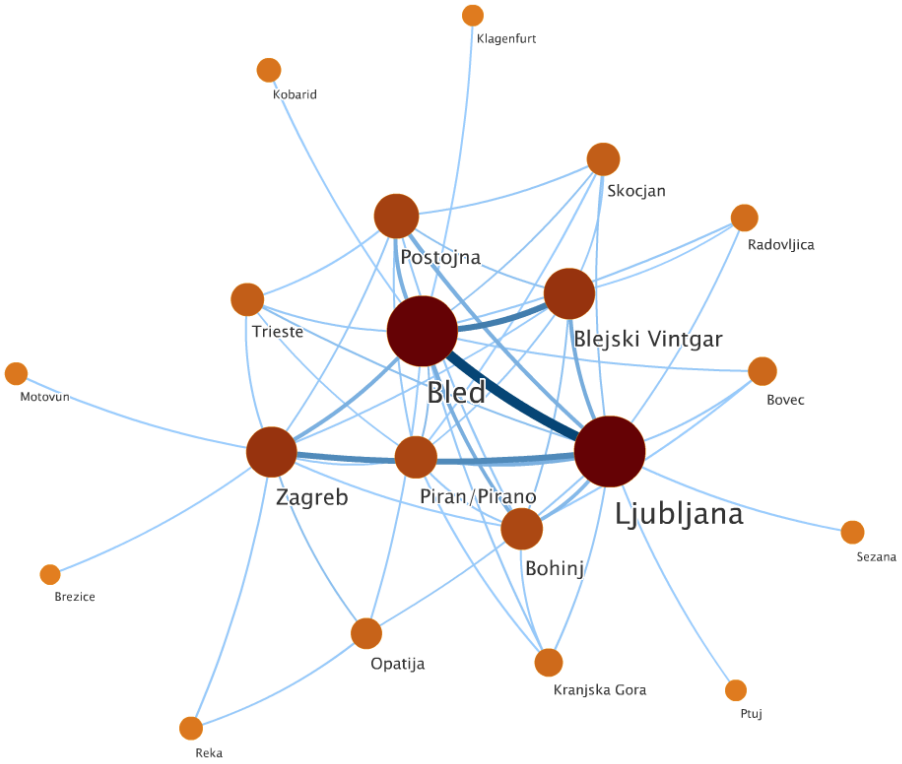
Preverimo, kaj lahko še razberemo s pomočjo algoritma Infomap. Velika glavna skupnost spet kaže problem koncentriranosti slovenskega turizma. Izpostavili bi močno povezanost z Zagrebom, na kar kaže debelina povezave in pojav s filtriranjem po utežeh. Smiselno bi bilo tržiti Bled in Ljubljano v Zagrebu oziroma vsa tri mesta skupaj. Hitri pojav destinacij v sosednjih državah (Zagreb, Trst, Opatija, Reka), kaže da veliko obiskovalcev obišče naše sosednje države, kar je verjetno tudi posledica majhnosti in tranzitnosti Slovenije. Med prvimi petimi destinacijami po pojavitvah glede na utež (v Navigatorjevem grafu Infomapa utež postane izračunan tok med destinacijama) so vse tri destinacije z največ prenočitvami: Bled, Ljubljana, Piran. Skupnosti odkrivamo z omogočenim prekrivanjem kar pomeni, da se lahko destinacija pojavi v več skupnostih.

5.2 Segmentacija po starosti

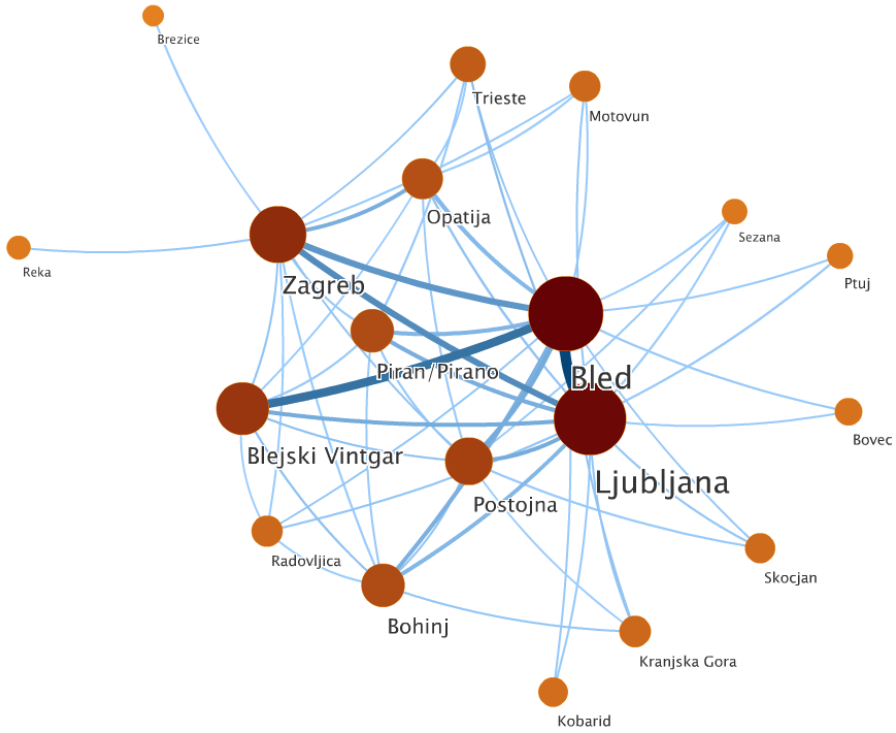
Preverimo, ali se odkrite skupnosti kaj razlikujejo po segmentaciji na tri starostna obdobja: mlada generacija na sliki 5.7, srednja na sliki 5.8 in starejša na sliki 5.9. Če primerjamo slike so starejši najbolj aktivni in obiščejo več krajev - ker verjetno imajo več časa in več denarja. Skupina mladih obišče mesta, kjer so festivali, srednja pa bolj družini primerne destinacije.



Slika 5.7: Glavna skupnost z rezanjem, ki jo vrne Infomap za mlajše turiste (starostna skupina 1 in 2: 13-24 let). V primerjavi z glavno skupino na grafu vseh zapisov se hitreje pojavijo destinacije Kranjska Gora in Bovec kar nakazuje na to, da so mlajše generacije bolj aktivne.



Slika 5.8: Glavna skupnost z rezanjem, ki jo vrne Infomap za turiste srednjih let (starostna skupina 3 in 4: 25-49 let), je podobna glavni skupini na grafu vseh zapisov.



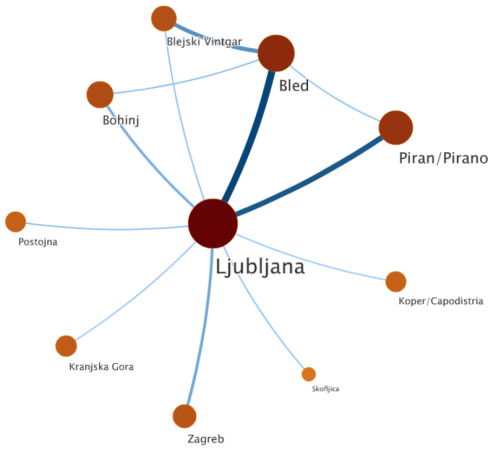
Slika 5.9: Glavna skupnost z rezanjem, ki jo vrne Infomap, za turiste starejše od 50 let (starostna skupina 5 in 6). Zaznamo lahko kasnejšo pojavitev Bovca, ki velja za bolj adrenalinsko destinacijo in hirejši pojavitvi destinacij Ptuj (staro mestno jedro) in Sežana (parki in Lipica).

5.3 Segmentacija po državi porekla

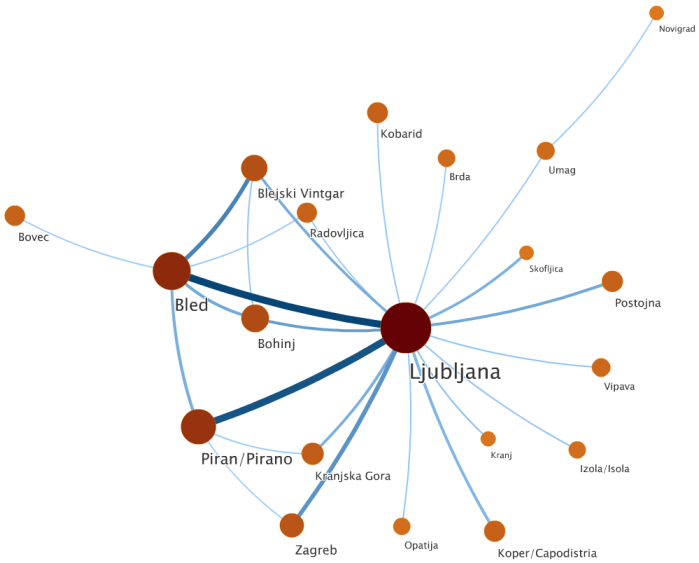
Segmentacijo izvajamo za domače prebivalce (na sliki 5.10), prebivalce držav, ki največ objavljajo (slika 5.11), so po statističnih podatkih med največjim deležem (slika 5.16) in za vse sosednje države (slike 5.12, 5.13, 5.14, 5.15). Poudarek analize je na glavni skupnosti. Za nekatere prebivalce držav (Američane, Britance) je zelo podobna glavni skupnosti na grafu vseh transakcij. Za domače obiskovalce je razvidno, da prevladujejo enodnevna potovanja, ker je graf najbolj razvejan. Ker ima Piran močnejši pretok kot na drugih grafih, je za Slovence smiselna skupna kampanja s sosednjimi vozlišči na sliki 5.10(b). Pri prebivalcih iz sosednjih držav opazimo vpliv izvora tudi na odkrite skupnosti. Za Italijane je zanimiv podgraf, ki ga sestavljata Motovun in Umag (viden na sliki 5.12(b)) in se kar dolgo časa ne poveže z glavnino - to kaže na veliko obiskov teh dveh destinacij hkrati oziroma močno podporo.

SKUPNOSTI
Ljubljana
Smarjeske Toplice
Palmanova
Cirkulane
Nova Gorica
Cerklje na Gorenjskem
Ljutomer

SKUPNOST: LJUBLJANA
Ljubljana
Bled
Piran / Pirano
Bohinj
Blejski Vintgar
Zagreb
Kranjska Gora
Koper / Capodistria
Postojna
Kobarid
Bovec
Radovljica
Ptuj
Vipava
Umag
Maribor
Brda
Izola / Isola
Opatija
Skofja Loka
...

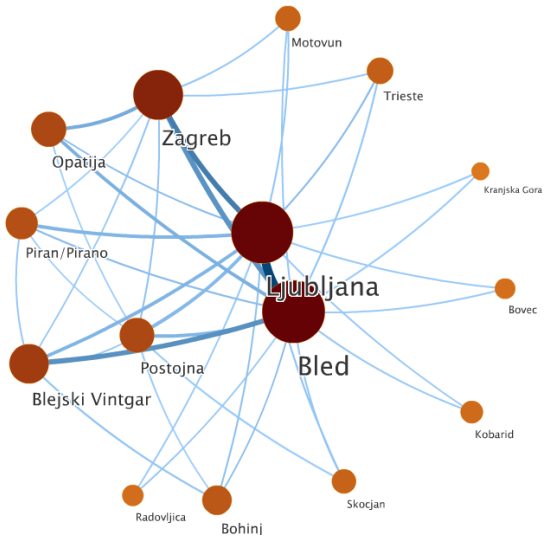


(a) Glavna skupnost domačih obiskovalcev z zajetimi prvimi 10 vozlišči po pretoku. Levo so prikazana tudi imena vseh najdenih skupnosti.

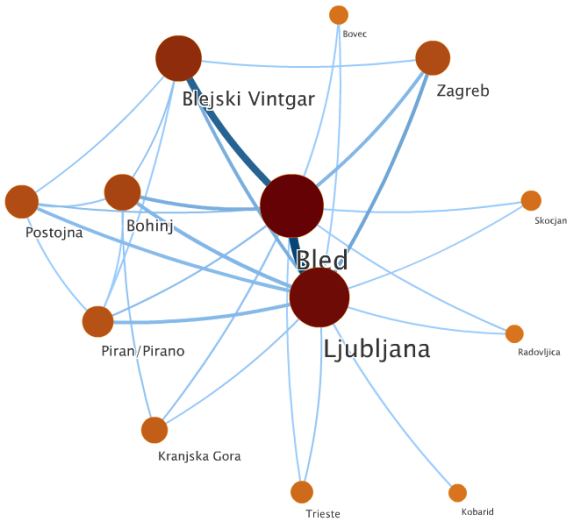


(b) Glavna skupnost domačih obiskovalcev z zajetimi prvimi 20 vozlišči po pretoku.

Slika 5.10: Graf za domače prebivalce je najbolj razvejan. Iz grafa se vidi, da prevladujejo enodnevna potovanja. V primerjavi s tujimi obiskovalci opazimo, da je Piran veliko močnejši.

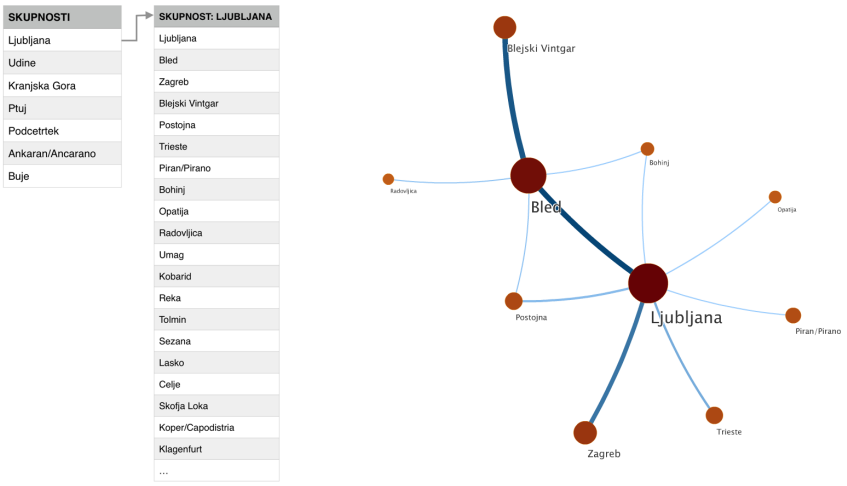


(a) Glavna skupnost obiskovalcev Združenih držav Amerike z rezanjem glede na moč pretoka.

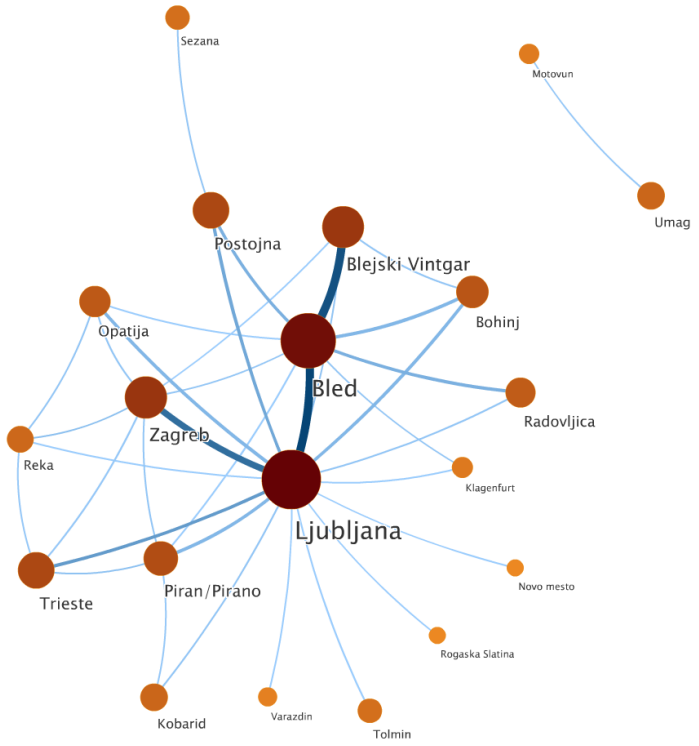


(b) Glavna skupnost obiskovalcev iz Združenega kraljestva z rezanjem glede na moč pretoka.

Slika 5.11: Primerjava glavne skupnosti prebivalcev iz držav, ki največ objavljajo. Glavna skupnost je zelo podobna. Na grafu prebivalcev iz ZDA se opazi malo močnejše povezave s hrvaškimi destinacijami.

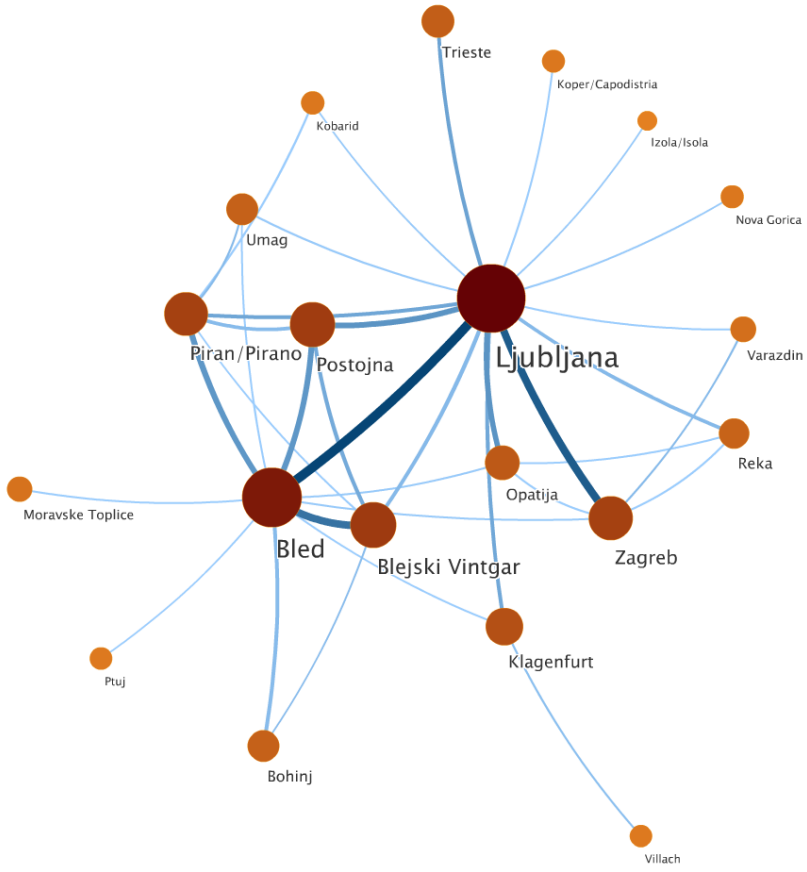


(a) Infomap za obiskovalce iz Italije zazna več manjših skupnosti na Primorskem oziroma blizu Italije. Prikazana je glavna skupnost z več rezanja.

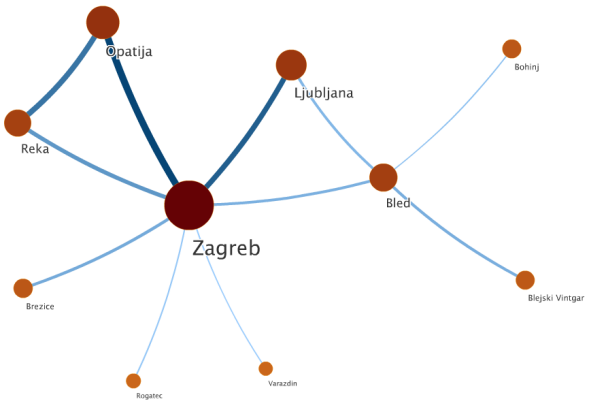


(b) Glavna skupnost obiskovalcev iz Italije z manj rezanja. Zanimiva je dolga nepovezanost krajev Motovun in Umag z glavnino.

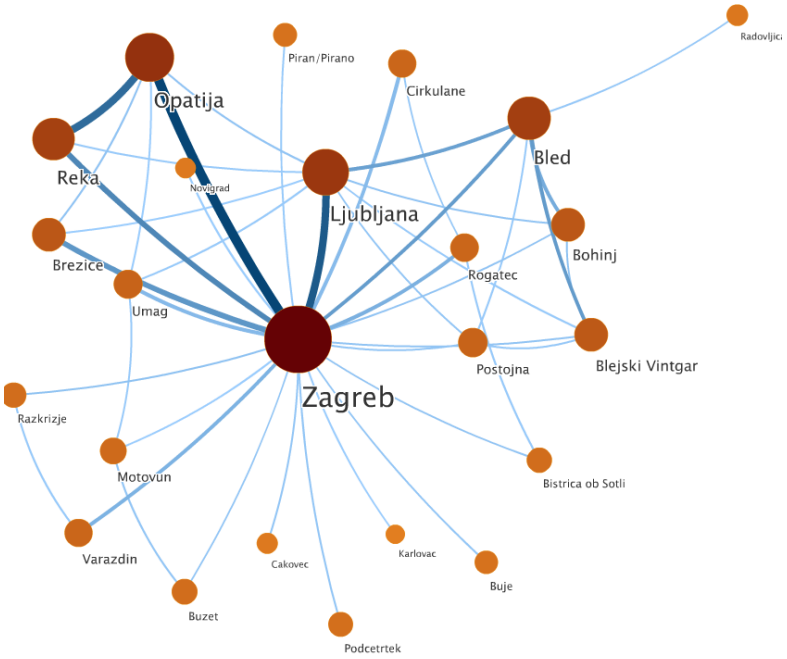
Slika 5.12: Glavna skupnost prebivalcev iz Italije z (a) več in (b) manj rezanja glede na moč pretoka.



Slika 5.13: Glavna skupnost z rezanjem, ki jo vrne Infomap, za prebivalce Avstrije. Vpliv sosednosti se kaže s hitrejšim pojavom destinacij Klagenfurt in Villach.

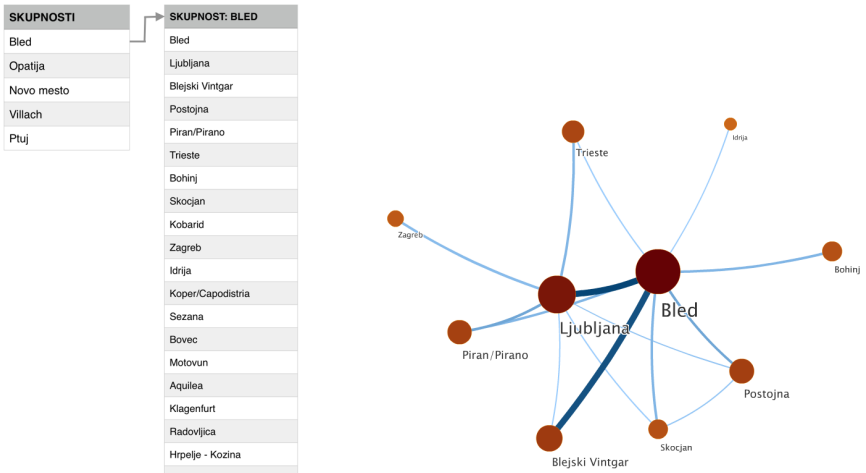


(a) Infomap za obiskovalce iz Hrvaške zazna Zagreb kot najmočnejše vozlišče. Prikazana je glavna skupnost obiskovalcev iz Hrvaške z več rezanja.

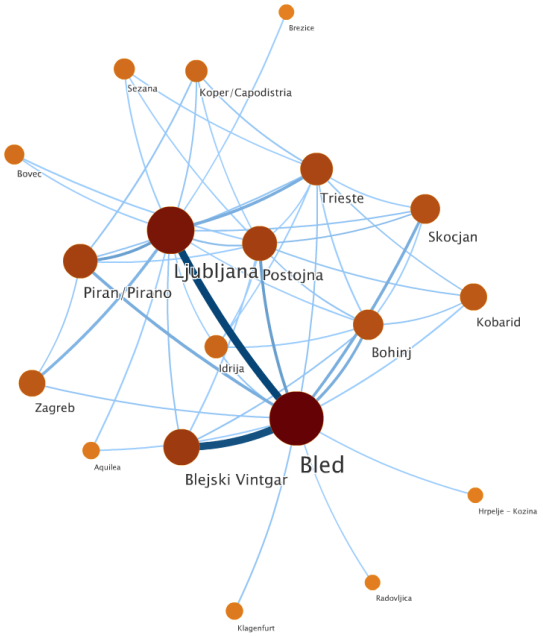


(b) Glavna skupnost obiskovalcev iz Hrvaške z manj rezanja. Zanimanja za morske destinacije ni.

Slika 5.14: Glavna skupnost prebivalcev iz Hrvaške z (a) več in (b) manj rezanja glede na moč pretoka.

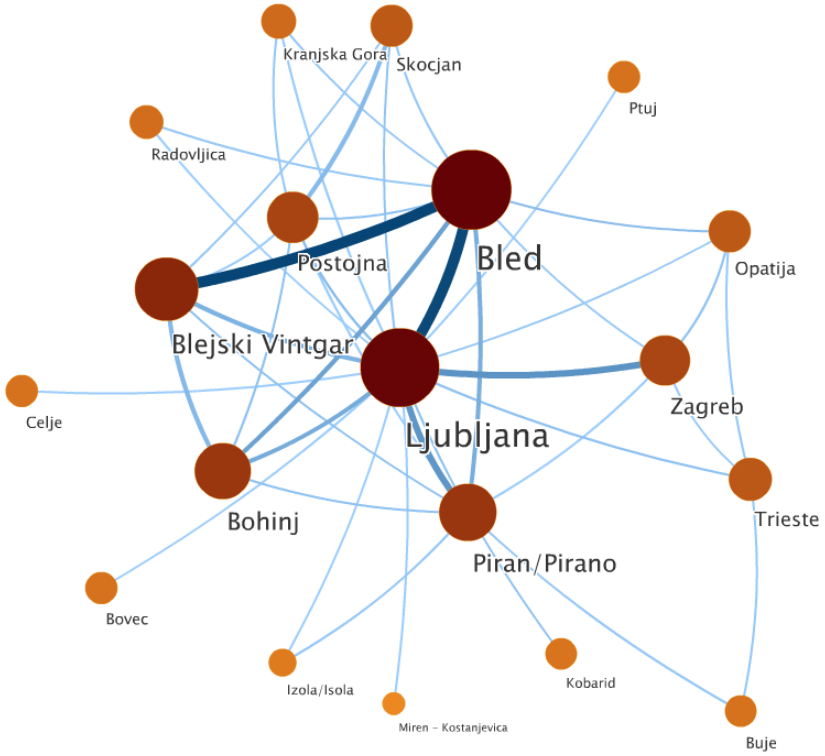


(a) Infomap za obiskovalce z Madžarske zazna več manjših skupnosti na vzhodu države. Prikazana je glavna skupnost z več rezanja.



(b) Glavna skupnost obiskovalcev z Madžarske z manj rezanja. Hitri pojav Idrije in Kobarida nakazujeta, da Madžare zanimajo bolj obiski znamenitosti. Slovenske morske destinacije jim niso tako zanimive. Zagreb je uvrščen nižje.

Slika 5.15: Glavna skupnost prebivalcev z Madžarske z (a) več in (b) manj rezanja glede na moč pretoka.



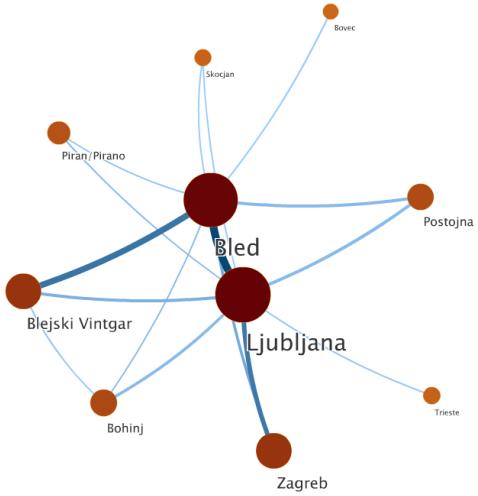
Slika 5.16: Glavna skupnost z rezanjem, ki jo vrne Infomap, za prebivalce Nemčije, je podobna kot na grafu vseh zapisov.

5.4 Segmentacija po lastnostih uporabnika

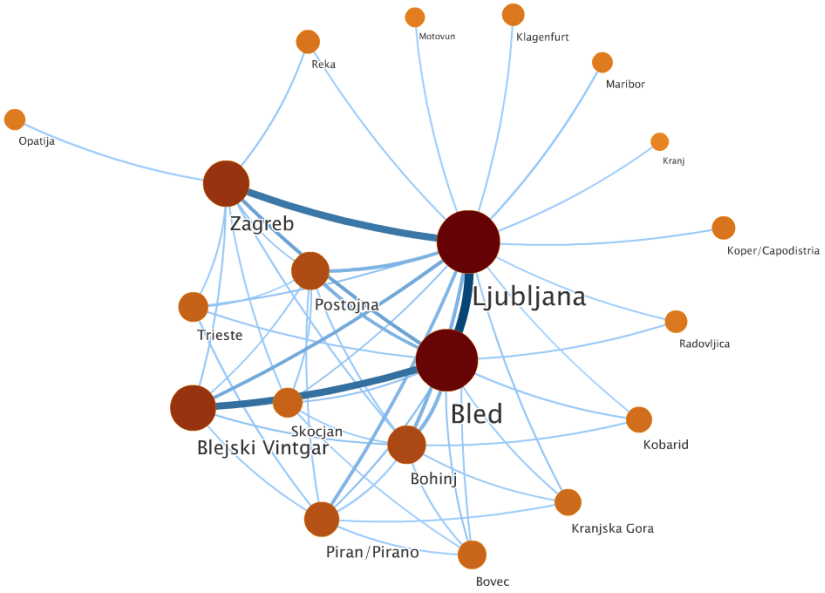
Segmentacijo po lastnostih uporabnika izvajamo po dodatnih atributih naštetih v Tabeli 4.1 (razen domači kraj). Po spolu ne opazimo posebnih razlik. Na voljo imamo veliko atributov za opis stila obiskovalca. Prikazali bomo samo grafe, ki se nam zdijo bolj zanimivi: za popotnika z nahrbtnikom na sliki 5.17 in za iskalca nočnega življenja na sliki 5.18. Kot lahko vidmo, segment turistov popotnikov z nahrtbnikom obišče bolj umaknjene kraje, poleg ključnih atrakcij. Za stile profilov na slovenskih podatkih nismo dobili signifikantnih rezultatov zaradi prevelike centralizacije, zato smo poskusili s segmentacijo te vrste še na podatkih Dunaja. Tam je segmentacija bolj izrazita.

SKUPNOSTI
Ljubljana
Buje
Bistrica ob Sotli
Store
Smarjeske Toplice
Cakovec
Miren - Kostanjevica
Radenci
Rence - Vogrsko
Murska Sobota

SKUPNOST: LJUBLJANA
Ljubljana
Bled
Zagreb
Blejski Vintgar
Bohinj
Postojna
Piran / Pirano
Trieste
Skocjan
Bovec
Kranjska Gora
Kobarid
Reka
Koper / Capodistria
Radovljica
Klagenfurt
Opatija
Ptuj
Maribor
Brda
...



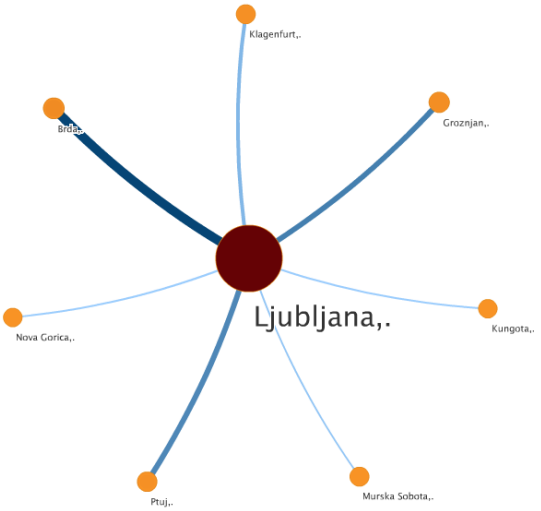
(a) Levo odkrite skupnosti, ki jih je res veliko (10). Desno glavna skupnost z rezanjem, ki jo vrne Infomap za popotnike z nahrbtnikom, ki je podobna glavni skupnosti na grafu vseh zapisov, a opozoriti moramo na hitrejšo pojavitev Bovca.



(b) Glavna skupnost obiskovalcev s stilom popotnika z nahrbtnikom z manj rezanja. Ljubljana je bolj pomembna destinacija kot Bled (iz nje izvira več povezav), najverjetneje zaradi glavne železniške postaje.

Slika 5.17: Glavna skupnost obiskovalcev s stilom popotnika z nahrbtnikom z (a) več in (b) manj rezanja glede na moč pretoka.

SKUPNOSTI
Ljubljana
Brda
Groznjan
Ptuj
Klagenfurt
Murska Sobota
Nova Gorica
Kungota



Slika 5.18: Prikazujemo odkrite skupnosti za iskalca nočnega življenja, ki kažejo na odkrit tok okoli mest. Glavna skupnost Ljubljana je podobna glavni skupnosti na grafu vseh zapisov.

Poglavje 6

Sklepne ugotovitve

V delu smo metode za analizo nakupovalne košarice uporabili za iskanje skupin pogosto obiskanih destinacij. Implementirana rešitev uporablja podatke pridobljene iz javnih objav turistov na spletnem mestu TripAdvisor, omogoča gradnjo grafa sočasnih pojavitev na podlagi obiskov, uporabo treh metod za analizo nakupovalne košarice na domeni obiskov turističnih destinacij in grafično predstavitev tako rezultatov kot vhodnih podatkov. Uporabniku omogoča različno filtriranje in segmentacijo. Na ta način se lahko sestavi nove ponudbe za potovanja, ki povezujejo več destinacij. Različne profile ljudi (lahko) glede na zanimanja privlačijo različne destinacije, zato jih je smotrno ponuditi skupaj v paketu. Nadalje lahko na podlagi odkritih skupin promoviramo manj obiskane turistične destinacije, ki imajo potencial, kar lahko sčasoma pomaga pri razpršenosti turističnega obiskovanja.

Čeprav je zanesljivost podatkov v javnih evidencah visoka pa ti zajamejo le goste, ki so bili vpisani v knjigo gostov, ne pa tudi tranzitnih gostov, ki v državi ne prespijo, pa tudi za tiste, ki v nekem kraju prespijo ne vemo, ali so čez dan potovali tudi v druge kraje. Ena pomembnih prednosti uporabe javnih objav turistov je, da zajemajo tranzitne goste. Čeprav je zaradi načina oddaje ocen zanesljivost podatkov nekoliko nižja (vsak obiskovalec ne odda svoje objave), pa je ob dovolj velikem številu objav z upoštevanjem ponovitev mogoče doseči ustrezno stopnjo zanesljivosti. Zavedati se moramo, da

podatki s TripAdvisorja niso popolnoma zanesljivi, vendar vseeno vsebujejo veliko informacij. Kot nekoliko težavnejša se je izkazala analiza posameznih segmentov trga, saj so bili rezultati zaradi omejenega števila podatkov manj zanesljivi. Podatki nakazujejo, da ima Slovenija veliko enodnevnih obiskovalcev, kar je najverjetneje tudi posledica majhnosti in dobre tranzitnosti. Da bi obiskovalce zadržali za dlje časa in bi bila njihova potovanja sestavljena iz obiska več destinacij v Sloveniji, je potrebno bolje razumeti njihovo obnašanje oziroma želje in potrebe.

Pri segmentaciji vidmo, da se v glavni skupnosti najdeni s pomočjo Infomapa med vozlišči z največ pretoka konstantno nahajajo iste destinacije. Očitno je, da se turisti gibajo okoli podobnih točk, kar kaže na koncentracijo. V teh točkah oziroma destinacijah je sedaj rast previsoka, zato je potrebno razmišljati o modelih razpršitve. Rezultati kažejo, da so najbolj potencialni segmenti - starejši, njihov graf je bil precej razvejan in veliko porabijo, ter popotniki z nahrbtnikom, ki sicer manj porabijo, ampak so pripravljeni videti drugačne dele Slovenijo. Potencial je tudi tranzit. Trženje je potrebno usmeriti tudi na okoliške kraje, predvsem Zagreb, Trst, Opatijo in Reko. Pri segmentaciji po državi porekla vidimo, da se najdene skupnosti razlikujejo za obiskovalce iz sosednjih držav, saj imajo destinacije v bližini meje države, iz katere so obiskovalci, več obiskov. Za prebivalce iz Hrvaške mesto Zagreb postane močnejše od Ljubljane.

Implementacijo bi lahko izboljšali tako, da bi omogočili branje več formatov vhodnih datotek. Potrebno bi bilo povečati hitrost generiranja grafa sočasnih pojavitev. Smiselno bi bilo poskusiti s kombinacijo algoritmov Louvain in Infomap ali z drugimi parametri (primer za razbitje v več skupnosti je v Dodatku B). Vključili bi lahko še več metod za odkrivanje skupnosti na grafu, vendar ne vseh, ampak samo tiste, ki bi se s testiranjem izkazale za uporabne - prekrivajoče skupnosti so pogosto iskane z algoritmoma COPRA in SLPA. Poskusili bi lahko tudi uporabo grafa asociacijskih pravil (angl. association rules networks) in podgrafov centralnih delov (angl. center-piece subgraphs).

Dodatek A

Matrika sosednosti destinacij

Matrika sosednosti destinacij na grafu je zaradi velikosti predstavljena na štirih straneh. Imena destinacij v glavi tabele so skrajšana, na levi strani so vidna polna imena. Vsako polje v tabeli s celoštevilsko vrednostjo ustreza uteži na povezavi med destinacijama. Matrika sosednosti vozlišč grafa z neusmerjenimi povezavami je simetrična.

Tabela A.1: Prva četrtina matrike sosednosti destinacij.

	Ank.	Aqui.	Bad	Kalin.	Beltin.	Bistrica...	Bled	Blejski V.	Bloke	Bohinj	Bovec	Brda	Brezice	Brezov.	Buje	Buzet	Cakovec	Celje	Cerklje...	Cerkno	Cirkul.	Divaca	Dol. Topl.	G. Radg.
Ankaran/Ancarano	0	0	0	0	0	0	1	2	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Aquilea	0	0	0	0	0	0	3	2	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
Bad Kalinkirchheim	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Beltinci	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bistrica ob Sotli	0	0	0	0	0	0	8	0	0	2	0	0	3	0	0	0	0	1	0	0	5	0	0	0
Bled	1	3	1	1	8	0	1446	6	499	117	9	18	0	11	15	1	7	1	5	6	11	5	0	0
Blejski Vintgar	2	2	0	0	0	1446	0	2	228	32	1	8	0	2	3	0	4	0	1	2	11	1	0	0
Bloke	0	0	0	0	0	6	2	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bohinj	1	0	0	1	2	499	228	3	0	45	3	8	1	1	6	0	6	1	1	2	4	2	0	0
Bovec	1	0	0	0	0	117	32	0	45	0	7	1	0	1	2	0	0	0	1	0	2	0	0	0
Brda	1	3	0	0	0	9	1	0	3	7	0	0	0	0	0	0	1	0	1	0	0	0	0	0
Brezice	0	0	0	0	3	18	8	0	8	1	0	0	0	0	2	0	0	1	0	4	1	1	0	0
Brezovica	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Buje	0	0	0	0	0	11	2	0	1	1	0	0	0	0	15	0	0	0	0	0	0	1	0	0
Buzet	0	0	0	0	0	15	3	0	6	2	0	2	1	15	0	0	0	0	0	1	0	0	0	0
Cakovec	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Celje	0	0	0	0	1	7	4	0	6	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
Cerklje na Gorenjskem	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Cerkno	0	0	0	0	0	5	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Cirkulane	0	0	0	0	5	6	2	0	2	0	0	4	0	0	1	0	0	0	0	0	0	0	0	0
Divaca	0	0	0	0	0	11	11	0	4	2	0	1	0	0	0	0	1	0	0	0	0	0	0	0
Dolenjske Toplice	0	0	0	0	0	5	1	0	2	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
Gornja Radgona	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Grado	0	3	0	0	0	2	0	0	1	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0
Groznjan	0	0	0	0	0	9	5	0	3	1	0	0	0	9	9	0	0	0	0	0	0	0	0	0
Horjul	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hrpelje - Kozina	0	0	0	0	0	3	0	0	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Idrija	0	0	0	0	0	10	2	0	3	3	0	0	0	0	0	0	0	0	0	1	0	0	0	0
Izola/Isola	3	0	0	0	0	9	4	0	2	2	1	0	0	2	0	0	0	0	0	0	0	2	0	0
Jesenice	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kamnik	0	0	0	0	0	8	5	1	2	2	1	1	0	0	0	0	0	0	0	3	0	0	0	0
Kanal	0	0	0	0	0	3	1	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Karlovac	0	0	0	0	0	4	2	0	1	0	0	2	0	0	0	1	0	0	0	0	1	0	0	0
Klagenfurt	1	0	4	0	0	37	13	0	14	2	2	0	0	0	0	0	1	0	1	0	1	1	0	0
Kobarid	0	1	0	0	2	87	33	1	46	28	9	1	0	3	2	0	2	0	3	0	2	0	0	0
Komen	1	0	0	0	0	2	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0
Koper/Capodistria	1	0	0	0	0	24	9	0	7	5	1	0	0	0	0	1	0	0	1	0	2	0	0	0
Kostel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kranj	0	0	1	0	0	11	4	0	4	1	1	0	0	1	0	0	0	0	0	0	1	1	0	0
Kranjska Gora	0	0	1	0	2	129	42	2	114	24	4	1	0	0	2	0	2	1	0	0	2	2	0	0
Kungota	0	0	0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Lasko	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Lendava/Lendva	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Ljubljana	4	4	0	3	8	2238	519	3	372	101	25	27	7	13	20	0	10	0	6	5	9	11	1	0
Ljutomer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maribor	0	0	0	1	0	21	8	0	8	1	1	1	0	0	0	1	3	0	1	0	1	0	0	0
Menges	0	0	0	0	0	1	1	0	2	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0
Miren - Kostanjevica	0	4	0	0	0	6	0	0	0	0	6	0	0	1	0	0	0	0	0	0	0	0	0	0
Moravske Toplice	0	0	0	2	0	4	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0
Motovun	0	0	0	0	0	63	12	0	9	3	4	3	1	23	38	1	0	0	1	3	1	0	0	0
Murska Sobota	0	0	0	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Nova Gorica	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Novigrad	0	0	0	0	0	5	1	0	2	1	0	1	0	9	1	0	0	0	0	0	1	0	0	0
Novo mesto	0	0	0	0	0	3	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	3	0	0
Opatija	0	1	1	0	9	209	30	1	13	3	1	10	0	14	22	1	0	0	0	5	2	0	0	0
Palmanova	1	2	0	0	0	5	1	0	2	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
Piran/Pirano	10	0	0	1	2	369	110	0	85	42	6	4	1	15	5	0	5	0	2	0	7	4	0	0
Podcetrtek	0	0	0	0	12	9	2	0	2	4	0	3	0	0	0	0	3	0	1	4	0	0	0	0
Postojna	2	1	0	0	5	498	199	6	124	24	2	11	0	1	7	0	5	0	2	3	9	2	0	0
Ptuj	0	1	0	1	1	49	14	0	16	2	0	3	0	0	0	0	3	0	1	1	1	1	1	1
Radenci	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Radovljica	0	0	0	1	2	141	53	0	38	8	2	7	0	1	3	0	3	0	0	1	3	0	0	0
Razkrizje	0	0	0	0	1	0	1	0	1	1	0	1	0	0	0	3	0	0	0	1	0	0	0	0
Reka	0	0	0	0	1	20	3	0	5	2	0	4	0	3	8	1	1	0	0	3	0	0	0	0
Rence - Vogrsko	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rogaska Slatina	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rogatec	0	0	0	0	10	3	0	0	0	0	1	3	0	1	0	1	0	0	0	13	0	0	0	0
Ruse	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sempeter - Vrtojba	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sentilj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sezana	0	0	0	1	0	46	16	0	8	4	0	1	0	1	1	0	0	0	1	0	1	0	0	0
Skocjan	2	0	0	0	0	154	72	3	51	18	1	4	0	1	4	0	1	0	2	0	14	1	0	0
Skofja Loka	0	0	0	1	0	23	6	1	13	2	1	4	0	0	0	0	1	0	2	0	0	0	0	0
Skofljica	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Smarjeske Toplice	0	0	0	0	0	14	2	0	5	3	0	1	0	0	0	0	0	0	0	0	0	0	6	0
Solcava	0	0	0	0	0	6	3	1	4	1	1	1	0	0	1	0	1	0	1	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
Tolmin	0	0	0	0	0	34	18	0	14	12	1	0	0	0	0	0	0	0	1	0	0	0	0	0
Trieste	10	12	3	0	0	134	36	0	28	5	11	3	1	8	2	0	2	0	0	1	1	0	0	0
Trzin	0	0	0	0	0	10	4	0	0	3	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Udine	0	6	0	0	0	9	1	0	0	3	24	0	0	0	0	0	0	0	0	0	0	0	0	0
Umag	0	0	1	0	0	9	1	0	2	1	1	2	1	13	2	2	0	0	0	0	0	0	0	0
Varazdin	0	0	0	0	4	5	1	0	1	0	0	4	0	2	0	5	0	0	0	4	0	0	0	0
Verzej	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Villach	0	0	3	0	0	15	4	0	2	4	2	0	0	0	0	0	0	0	1	0	0	1	0	0
Vipava	0	0	0	1	0	6	6	0	3	1	1	1	0	2	1	0	0	0	1	0	4	0	0	0
Vojnik	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
Zagreb	0	0	1	1	25	687	127	0	61	17	5	69	1	27	29	14	1	1	1	44	2	5	0	0
Zalec	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zelezniki	0	0	0	0	0	2	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabela A.2: Druga četrtina matrike sosednosti destinacij.

	Grado	Grozn.	Horjul	Hrp.-Koz.	Idrija	Izola	Jesenice	Kamnik	Kanal	Karlovac	Klagen.	Kobarid	Komen	Koper	Kostel	Kranj	Kr. Gora	Kungota	Lasko	Lendava	Lj.	Ljutomer	Marib.
Ankaran/Ancarano	0	0	0	0	0	3	0	0	0	0	1	0	1	1	0	0	0	0	0	0	4	0	0
Aquilea	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	4	0	0
Bad Kalinkirchheim	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	1	1	0	0	0	0	0	0
Beltinci	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	1
Bistrica ob Sotli	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	2	0	0	0	8	0	0
Bled	2	9	1	3	10	9	1	8	3	4	37	87	2	24	0	11	129	1	1	0	2238	0	21
Blejski Vintgar	0	5	0	0	2	4	0	5	1	2	13	33	0	9	0	4	42	1	0	0	519	0	8
Bloke	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	2	0	0	0	3	0	0
Bohinj	1	3	0	2	3	2	0	2	1	1	14	46	1	7	0	4	114	1	0	0	372	0	8
Bovec	1	1	0	0	3	2	0	2	2	0	2	28	1	5	0	1	24	0	0	0	101	0	1
Brda	2	0	0	1	0	1	0	1	0	0	2	9	1	1	0	1	4	0	0	0	25	0	1
Brezice	1	0	0	0	0	0	0	1	0	2	0	1	0	0	0	0	1	0	0	0	27	0	1
Brezovica	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0
Buje	0	9	0	0	0	2	0	0	0	0	0	3	0	0	0	1	0	1	0	0	13	0	0
Buzet	0	9	0	0	0	0	0	0	0	0	0	2	0	0	0	0	2	0	0	0	20	0	0
Cakovec	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1
Celje	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	2	0	1	0	10	0	3
Cerklje na Gorenjskem	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Cerkno	0	0	0	0	1	0	0	3	0	0	1	3	1	1	0	0	0	0	0	0	6	0	1
Cirkulane	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	5	0	0
Divaca	0	0	0	0	0	2	0	0	0	0	1	2	0	2	0	1	2	0	0	0	9	0	1
Dolenjske Toplice	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	2	0	0	0	11	0	0
Gornja Radgona	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Grado	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	2	0	0
Groznjan	0	0	0	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	11	0	0
Horjul	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	3	0	0
Hrpelje - Kozina	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	6	0	0
Idrija	0	0	0	0	0	0	0	0	0	0	0	4	0	1	0	0	1	0	0	0	14	0	0
Izola/Isola	0	1	1	0	0	0	0	0	0	0	0	2	0	9	0	0	0	1	0	0	30	0	1
Jesenice	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Kamnik	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	2	0	0	0	8	0	1
Kanal	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	4	0	0
Karlovac	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	2	0	1
Klagenfurt	0	0	0	0	0	0	0	1	0	0	0	2	1	1	0	0	7	0	0	0	36	0	0
Kobarid	1	2	1	2	4	2	1	1	8	0	2	0	1	2	0	0	12	0	0	0	100	0	3
Komen	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0	3	0	0
Koper/Capodistria	0	0	0	0	1	9	0	1	0	0	1	2	1	0	0	0	4	0	1	0	45	0	3
Kostel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kranj	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	27	0	0
Kranjska Gora	1	0	0	0	1	0	0	2	0	1	7	12	0	4	0	2	0	0	0	0	129	0	2
Kungota	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	4
Lasko	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0
Lendava/Lendva	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Ljubljana	2	11	3	6	14	30	0	8	4	2	36	100	3	45	0	27	129	2	2	1	0	1	35
Ljutomer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Maribor	0	0	0	0	0	1	0	1	0	1	0	3	0	3	0	0	2	4	0	0	35	0	0
Menges	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
Miren - Kostanjevica	2	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	5	0	0
Moravske Toplice	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0	2	5	0	1
Motovun	0	33	0	0	0	1	0	1	1	0	1	8	0	1	0	0	3	0	0	0	88	0	2
Murska Sobota	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	7	1	0
Nova Gorica	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3	0	0
Novigrad	0	3	0	1	0	2	0	0	0	0	2	2	0	1	0	0	1	0	0	0	6	0	0
Novo mesto	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0
Opatija	0	5	0	0	0	4	0	0	0	3	1	3	0	2	3	1	4	0	0	0	166	0	4
Palmanova	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0
Piran/Pirano	2	7	0	3	5	14	0	3	1	1	6	32	1	38	0	2	40	1	0	0	506	0	8
Podcetrtek	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	14	0	1
Postojna	1	5	0	1	5	1	0	6	0	3	14	24	0	9	0	1	41	0	1	1	479	0	9
Ptuj	0	0	0	1	1	4	0	1	1	0	3	2	0	2	0	1	3	0	1	0	56	0	12
Radenci	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
Radovljica	0	1	1	0	2	2	0	2	0	0	4	5	0	2	0	4	10	1	1	0	91	0	5
Razkrizje	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	2	0	0
Reka	0	3	0	0	0	0	0	0	0	2	2	0	0	1	1	1	3	0	0	0	59	0	1
Rence - Vogrsko	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rogaska Slatina	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Rogatec	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	4	0	0
Ruse	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Sempeter - Vrtojba	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
Sentilj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Sezana	0	0	0	0	2	1	0	1	0	0	0	8	2	3	0	0	2	0	0	0	60	0	6
Skocjan	2	3	0	2	3	0	1	2	1	0	2	22	1	4	0	1	6	0	0	0	152	0	3
Skofja Loka	0	0	0	0	0	1	0	1	0	1	1	3	1	1	0	0	3	0	1	0	28	0	1
Skofljica	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	19	0	0
Smarjeske Toplice	0	0	0	0	1	0	0	0	0	0	0	3	0	2	0	0	2	0	0	0	21	0	1
Solcava	0	0	0	0	0	0	0	2	1	0	2	2	1	1	0	0	1	0	0	0	8	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
Tolmin	0	1	0	0	2	0	0	0	2	0	1	20	1	0	0	0	5	0	0	0	20	0	0
Trieste	9	2	0	2	3	4	0	0	1	1	7	13	1	14	0	0	8	0	0	0	274	0	5
Trzin	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	21	0	0
Udine	5	0	0	0	0	0	0	0	2	0	2	5	0	0	0	0	4	0	0	0	12	0	1
Umag	0	2	0	1	0	5	0	0	0	0	0	1	0	2	0	0	3	0	0	0	27	0	0
Varazdin	0	1	0	0	0	0	0	0	0	3	1	0	0	0	0	0	1	0	0	1	10	0	1
Verzej	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	6	0	0
Villach	0	0	0	0	0	0	0	1	0	0	31	1	1	1	0	0	8	0	0	0	18	0	0
Vipava	0	0	0	0	1	1	0	0	0	0	0	2	0	2	1	0	1	1	0	0	20	0	2
Vojnik	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	1
Zagreb	1	10	0	0	3	3	0	1	3	16	8	16	0	6	2	4	15	1	1	0	1166	0	7
Zalec	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
Zelezniki	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0

Tabela A.3: Tretja četrtina matrike sosednosti destinacij.

	Menges	Mir.-Kost.	Mor.Topl.	Motov.	Mur.Sob.	Nova Gor.	Novig.	Novo m.	Opat.	Palman.	Piran	Podcet.	Postoj.	Ptuj	Raden.	Radov.	Razkr.	Reka	Ren.-Vogr.	Rog.Sl.	Rogat.	Ruse
Ankaran/Ancarano	0	0	0	0	0	0	0	0	0	1	10	0	2	0	0	0	0	0	0	0	0	0
Aquilea	0	4	0	0	0	1	0	0	1	2	0	0	1	1	0	0	0	0	0	0	0	0
Bad Kalinkirchheim	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Beltinci	0	0	2	0	1	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0
Bistrica ob Sotli	0	0	0	0	0	0	0	0	9	0	2	12	5	1	0	2	1	1	0	0	10	0
Bled	1	6	4	63	1	1	5	3	209	5	369	9	498	49	0	141	0	20	0	0	3	0
Blejski Vintgar	1	0	1	12	0	0	1	0	30	1	110	2	199	14	0	53	1	3	0	0	0	0
Bloke	0	0	0	0	0	0	0	0	1	0	0	0	6	0	0	0	0	0	0	0	0	0
Bohinj	2	0	0	9	1	0	2	1	13	2	85	2	124	16	0	38	1	5	0	0	0	0
Bovec	0	0	0	3	1	1	1	1	3	0	42	4	24	2	0	8	1	2	0	0	0	0
Brda	0	6	1	4	0	0	0	0	1	3	6	0	2	0	0	2	0	0	0	0	1	0
Brezice	1	0	0	3	0	0	1	0	10	0	4	3	11	3	0	7	1	4	0	0	3	0
Brezovica	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Buje	0	1	0	23	0	0	9	0	14	0	15	0	1	0	0	1	0	3	0	0	1	0
Buzet	0	0	0	38	0	0	1	0	22	0	5	0	7	0	0	3	0	8	0	0	0	0
Cakovec	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	3	1	0	0	1	0
Celje	0	0	0	0	0	0	0	1	0	0	5	3	5	3	0	3	0	1	0	0	0	0
Cerklje na Gorenjskem	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Cerkno	0	0	0	1	0	0	0	0	0	0	2	1	2	1	0	0	0	0	0	0	0	0
Cirkulane	1	0	0	3	0	0	0	0	5	0	0	4	3	1	0	1	1	3	0	0	13	0
Divaca	0	0	1	1	0	0	1	0	2	0	7	0	9	1	0	3	0	0	0	0	0	0
Dolenjske Toplice	1	0	1	0	0	0	0	3	0	0	4	0	2	1	0	0	0	0	0	0	0	0
Gornja Radgona	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Grado	0	2	0	0	0	0	0	0	0	1	2	0	1	0	0	0	0	0	0	0	0	0
Groznjan	0	0	0	33	0	0	3	0	5	1	7	0	5	0	0	1	0	3	0	0	0	0
Horjul	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Hrpelje - Kozina	0	0	0	0	0	0	1	0	0	0	3	0	1	1	0	0	0	0	0	0	0	0
Idrija	0	0	0	0	0	0	0	0	0	0	5	0	5	1	0	2	0	0	0	0	0	0
Izola/Isola	0	0	0	1	0	0	2	0	4	0	14	0	1	4	0	2	0	0	0	0	0	0
Jesenice	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kamnik	0	0	0	1	0	0	0	0	0	0	3	0	6	1	0	2	0	0	0	0	0	0
Kanal	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
Karlovac	0	0	0	0	0	0	0	0	3	0	1	1	3	0	0	0	1	2	0	0	1	0
Klagenfurt	0	1	1	1	0	0	2	0	1	0	6	0	14	3	0	4	1	2	0	0	0	0
Kobarid	0	0	0	8	1	0	2	0	3	0	32	1	24	2	0	5	0	0	0	0	0	0
Komen	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Koper/Capodistria	0	1	1	1	1	1	1	0	2	0	38	0	9	2	0	2	0	1	0	0	0	0
Kostel	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	1	0	0	0	0
Kranj	0	0	0	0	0	0	0	0	1	0	2	0	1	1	0	4	0	1	0	0	0	0
Kranjska Gora	0	1	1	3	0	0	1	0	4	0	40	0	41	3	0	10	0	3	0	0	0	0
Kungota	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
Lasko	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0
Lendava/Lendva	0	0	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Ljubljana	3	5	5	88	7	3	6	9	166	7	506	14	479	56	0	91	2	59	0	1	4	1
Ljutomer	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Maribor	0	0	1	2	0	0	0	0	4	0	8	1	9	12	0	5	0	1	0	0	0	0
Menges	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0	1	0	0	0	0	0	0
Miren - Kostanjevica	0	0	0	1	0	0	1	0	0	0	4	0	2	0	0	0	0	0	0	0	0	0
Moravske Toplice	0	0	0	0	7	0	0	0	0	0	2	0	1	3	1	1	0	0	0	0	0	0
Motovun	0	1	0	0	0	0	11	1	39	0	17	0	27	2	1	5	0	10	0	0	1	0
Murska Sobota	0	0	7	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Nova Gorica	0	0	0	0	0	0	0	0	0	0	2	0	2	0	0	0	0	0	1	0	0	0
Novigrad	0	1	0	11	0	0	0	0	3	0	4	0	2	0	0	0	0	3	0	0	0	0
Novo mesto	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Opatija	1	0	0	39	1	0	3	0	0	1	33	7	50	4	0	17	2	107	0	0	2	0
Palmanova	0	0	0	0	0	0	0	0	1	0	1	0	2	0	0	0	0	2	0	0	1	0
Piran/Pirano	1	4	2	17	0	2	4	1	33	1	0	4	109	21	0	22	0	4	0	0	0	0
Podcetrtek	0	0	0	0	0	0	0	0	7	0	4	0	6	3	0	3	0	0	0	0	4	0
Postojna	1	2	1	27	0	2	2	0	50	2	109	6	0	15	0	32	1	16	0	0	0	0
Ptuj	1	0	3	2	0	0	0	0	4	0	21	3	15	0	0	5	0	1	0	0	0	1
Radenci	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Radovljica	1	0	1	5	0	0	0	0	17	0	22	3	32	5	0	0	0	0	0	0	0	0
Razkrizje	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	2	0	0	0	0
Reka	0	0	0	10	0	0	3	0	107	2	4	0	16	1	0	0	2	0	0	0	1	0
Rence - Vogrsko	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rogaska Slatina	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rogatec	0	0	0	1	0	0	0	0	2	1	0	4	0	0	0	0	0	1	0	0	0	0
Ruse	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Sempeter - Vrtojba	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
Sentilj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sezana	0	0	0	3	0	0	1	0	2	0	25	2	56	2	0	6	0	0	0	0	0	0
Skocjan	0	0	1	8	0	0	0	0	11	0	44	0	104	7	0	6	0	5	0	0	0	0
Skofja Loka	0	0	0	0	0	0	0	0	1	0	9	0	6	2	0	5	0	1	0	0	0	0
Skofljica	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
Smarjeske Toplice	0	0	0	0	0	0	0	4	0	0	8	0	4	2	0	0	0	0	0	0	0	0
Solcava	0	0	0	0	0	0	0	0	0	0	5	0	2	2	0	0	0	0	0	0	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Tolmin	0	0	0	0	0	1	0	0	1	0	9	0	12	0	0	0	0	0	0	0	0	0
Trieste	0	18	1	14	1	2	3	0	44	4	87	1	47	3	0	8	0	23	0	0	0	0
Trzin	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Udine	0	4	0	0	0	0	0	0	2	11	2	0	1	0	0	0	0	0	0	0	0	0
Umag	1	0	0	17	0	0	18	0	11	0	23	0	5	1	0	2	0	7	0	0	0	0
Varazdin	0	0	0	5	0	0	0	0	4	0	2	4	5	2	0	1	6	4	0	0	5	0
Verzej	0	0	0	0	2	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
Villach	0	1	0	0	0	0	0	0	2	0	2	0	2	1	0	1	0	1	0	0	0	0
Vipava	0	0	0	1	0	0	2	0	2	0	12	0	7	2	0	2	0	0	0	0	0	0
Vojnik	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
Zagreb	1	5	3	85	1	1	8	5	366	2	70	17	140	20	2	37	11	124	0	1	34	0
Zalec	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Zelezniki	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

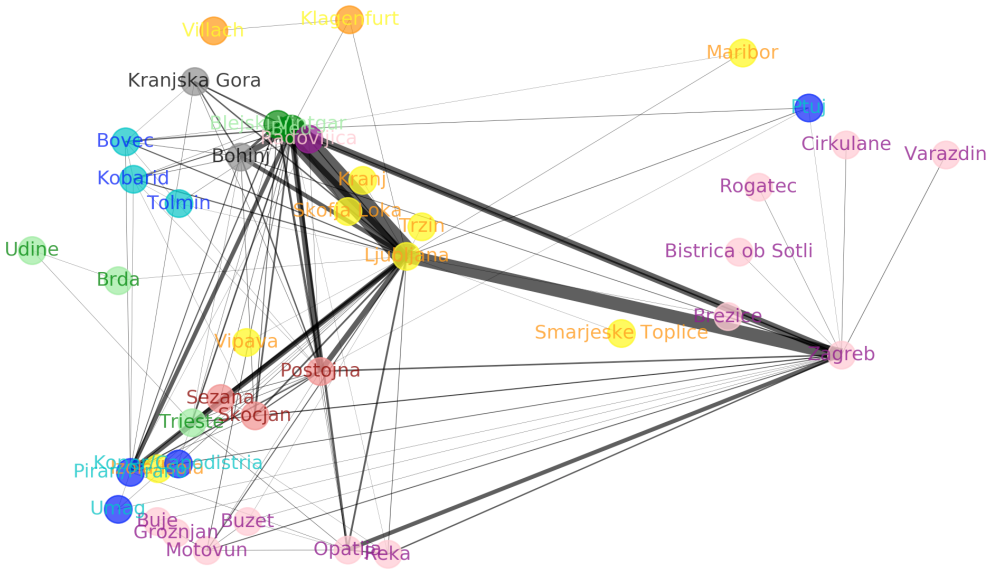
Tabela A.4: Zadnja četrtina matrike sosednosti destinacij.

	Sempet.-Vrtoj.	Sentilj	Sezana	Skocjan	Sk. Loka	Skoflj.	Smarj. Topl.	Solcava	Store	Tolmin	Trieste	Trzin	Udine	Umag	Varazdin	Verzej	Villach	Vipava	Vojnik	Zagreb	Zalec	Zelezniki
Ankaran/Ancarano	0	0	0	2	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0
Aquilea	1	0	0	0	0	0	0	0	0	0	12	0	6	0	0	0	0	0	0	0	0	0
Bad Kalinkirchheim	0	0	0	0	0	0	0	0	0	0	3	0	0	1	0	0	3	0	0	1	0	0
Beltinci	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
Bistrica ob Sotli	0	0	0	0	0	1	0	0	0	0	0	0	0	0	4	0	0	0	0	25	0	0
Bled	0	0	46	154	23	1	14	6	0	34	134	10	9	9	5	2	15	6	1	687	0	2
Blejski Vintgar	0	0	16	72	6	0	2	3	0	18	36	4	1	1	1	0	4	6	0	127	0	0
Bloke	0	0	0	3	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bohinj	0	0	8	51	13	0	5	4	0	14	28	0	0	2	1	0	2	3	1	61	0	2
Bovec	0	0	4	18	2	0	3	1	0	12	5	3	3	1	0	0	4	1	0	17	0	1
Brda	0	0	0	1	1	0	0	1	0	1	11	0	24	1	0	0	2	1	0	5	0	0
Brezice	0	0	1	4	4	0	1	1	0	0	3	0	0	2	4	0	0	1	0	69	0	0
Brezovica	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1	0	0
Buje	0	0	1	1	0	0	0	0	0	0	8	1	0	13	2	0	0	2	0	27	0	0
Buzet	0	0	1	4	0	0	0	1	0	0	2	0	0	2	0	0	0	1	0	29	0	0
Cakovec	0	0	0	0	0	0	0	0	0	0	0	0	0	2	5	0	0	0	0	14	0	0
Celje	0	0	0	1	1	0	0	1	2	0	2	0	0	0	0	0	0	0	4	1	0	0
Cerklje na Gorenjskem	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Cerkno	0	0	1	2	2	0	0	1	0	1	0	0	0	0	0	0	1	1	0	1	0	0
Cirkulane	0	0	0	0	0	0	0	0	0	0	1	0	0	0	4	0	0	0	0	44	0	0
Divaca	0	0	1	14	0	0	0	0	0	0	1	0	0	0	0	0	0	4	0	2	0	0
Dolenjske Toplice	0	0	0	1	0	0	6	0	0	0	0	0	0	0	0	0	1	0	0	5	0	0
Gornja Radgona	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Grado	1	0	0	2	0	0	0	0	0	0	9	0	5	0	0	0	0	0	0	1	0	0
Groznjan	0	0	0	3	0	0	0	0	0	1	2	0	0	2	1	0	0	0	0	10	0	0
Horjul	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hrpelje - Kozina	0	0	0	2	0	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0
Idrija	0	0	2	3	0	0	1	0	0	2	3	0	0	0	0	0	0	1	0	3	0	0
Izola/Isola	0	0	1	0	1	0	0	0	0	0	4	0	0	5	0	0	0	1	0	3	0	0
Jesenice	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Kamnik	0	0	1	2	1	0	0	2	0	0	0	0	0	0	0	0	1	0	0	1	0	0
Kanal	0	0	0	1	0	0	0	1	0	2	1	1	2	0	0	0	0	0	0	3	0	0
Karlovac	0	0	0	0	1	0	0	0	0	0	1	0	0	0	3	0	0	0	0	16	0	0
Klagenfurt	0	0	0	2	1	0	0	2	0	1	7	0	2	0	1	0	31	0	0	8	0	0
Kobarid	0	0	8	22	3	0	3	2	0	20	13	0	5	1	0	1	1	2	0	16	1	0
Komen	0	0	2	1	1	0	0	1	0	1	1	0	0	0	0	0	1	0	0	0	0	0
Koper/Capodistria	1	0	3	4	1	1	2	1	0	0	14	0	0	2	0	0	1	2	1	6	0	0
Kostel	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0
Kranj	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
Kranjska Gora	0	0	2	6	3	0	2	1	0	5	8	1	4	3	1	0	8	1	1	15	0	0
Kungota	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
Lasko	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Lendava/Lendva	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
Ljubljana	1	0	60	152	28	19	21	8	4	20	274	21	12	27	10	6	18	20	1	1166	1	10
Ljutomer	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Maribor	0	0	6	3	1	0	1	0	0	0	5	0	1	0	1	0	0	2	1	7	0	0
Menges	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
Miren - Kostanjevica	0	0	0	0	0	0	0	0	0	0	18	0	4	0	0	0	1	0	0	5	0	0
Moravske Toplice	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	3	0	0
Motovun	0	0	3	8	0	0	0	0	0	0	14	0	0	17	5	0	0	1	1	85	0	0
Murska Sobota	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0	0	0	1	0	0
Nova Gorica	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	1	0	0
Novigrad	0	0	1	0	0	0	0	0	0	0	3	0	0	18	0	0	0	2	0	8	0	0
Novo mesto	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
Opatija	0	0	2	11	1	0	0	0	0	1	44	0	2	11	4	0	2	2	0	366	0	0
Palmanova	0	0	0	0	0	0	0	0	0	0	4	0	11	0	0	0	0	0	0	2	0	0
Piran/Pirano	1	0	25	44	9	1	8	5	0	9	87	1	2	23	2	3	2	12	1	70	0	0
Podcetrtek	0	0	2	0	0	0	0	0	0	0	1	0	0	0	4	0	0	0	0	17	0	0
Postojna	1	0	56	104	6	1	4	2	0	12	47	0	1	5	5	0	2	7	1	140	0	0
Ptuj	0	0	2	7	2	0	2	2	0	0	3	0	0	1	2	0	1	2	0	20	0	0
Radenci	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
Radovljica	0	0	6	6	5	0	0	0	0	0	8	0	0	2	1	0	1	2	0	37	0	0
Razkrizje	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	11	0	0
Reka	0	0	0	5	1	0	0	0	0	0	23	0	0	7	4	0	1	0	0	124	0	0
Rence - Vogrsko	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Rogaska Slatina	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Rogatec	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	34	0	0
Ruse	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sempeter - Vrtojba	0	0	0	0	0	0	0	0	0	0	6	0	2	0	0	0	0	0	0	0	0	0
Sentilj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sezana	0	0	0	26	3	0	1	0	0	4	9	1	1	1	0	0	0	1	0	9	0	0
Skocjan	0	0	26	0	2	0	0	2	0	10	16	0	0	1	0	0	1	1	0	36	0	1
Skofja Loka	0	0	3	2	0	0	0	1	0	1	5	0	1	1	0	0	1	4	0	4	0	3
Skofljica	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0
Smarjeske Toplice	0	0	1	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	10	1	0
Solcava	0	0	0	2	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0	2	0	0
Store	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3	0
Tolmin	0	0	4	10	1	0	0	0	0	0	2	0	1	0	0	0	0	1	0	5	0	0
Trieste	6	0	9	16	5	1	3	1	0	2	0	0	29	7	2	1	6	2	0	97	0	0
Trzin	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0
Udine	2	0	1	0	1	0	0	0	0	1	29	0	0	0	0	0	2	0	0	5	0	0
Umag	0	0	1	1	1	0	0	0	0	0	7	0	0	0	1	0	0	2	0	27	0	0
Varazdin	0	0	0	0	0	0	0	1	0	0	2	0	0	1	0	0	0	0	0	54	0	0
Verzej	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Villach	0	0	0	1	1	0	0	1	0	0	6	0	2	0	0	0	0	0	0	7	0	0
Vipava	0	0	1	1	4	1	0	0	0	1	2	0	0	2	0	0	0	0	0	3	0	0
Vojnik	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Zagreb	0	0	9	36	4	0	10	2	0	5	97	5	5	27	54	0	7	3	0	0	0	2
Zalec	0	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
Zelezniki	0	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0

Dodatek B

Uporaba Louvaina in Infomapa z dodatnim parametrom za razbitje skupnosti

Oba algoritma za detekcijo skupnosti imata opcijski parameter, ki omogoča iskanje manjših skupnosti od privzetih. Louvain ima privzeto nastavljen parameter *resolution* na 1. Z manjšo vrednostjo Louvain daje prednost manjšim skupnostim. Na sliki B.1 predstavljamo najdene skupnosti z vrednostjo 0.5 na grafu vseh zapisov. Pri Infomapu vplivamo na velikost skupnosti z uporabo parametra *markov-time*, ki ima prav tako privzeto vrednost 1. Parameter definira, koliko korakov naredi naključni sprehajalec, preden je njegova pozicija zašifrirana. S krajšim Markovim časom je vozlišče kodirano večkrat zapored, kar rezultira v več in manjših skupnosti. Na slikah B.2, B.3, B.4 predstavljamo najdene skupnosti z vrednostjo 0.75. Z manjšanjem Markovega časa lahko dosežemo nestanovitnost rezultatov.



SKUPNOST 0
Bovec
Kobarid
Tolmin

SKUPNOST 1
Izola/Isola
Kranj
Ljubljana
Maribor
Skofja Loka
Smarjeske Toplice
Trzin
Vipava

SKUPNOST 2
Brda
Trieste
Udine

SKUPNOST 3
Bistrica ob Sotli
Brezice
Buje
Buzet
Cirkulane
Grožnjan
Motovun
Opatija
Reka
Rogatec
Varazdin
Zagreb

SKUPNOST 4
Postojna
Sezana
Skocjan

SKUPNOST 5
Bohinj
Kranjska Gora

SKUPNOST 6
Koper / Capodistria
Piran / Pirano
Ptuj
Umag

SKUPNOST 7
Klagenfurt
Villach

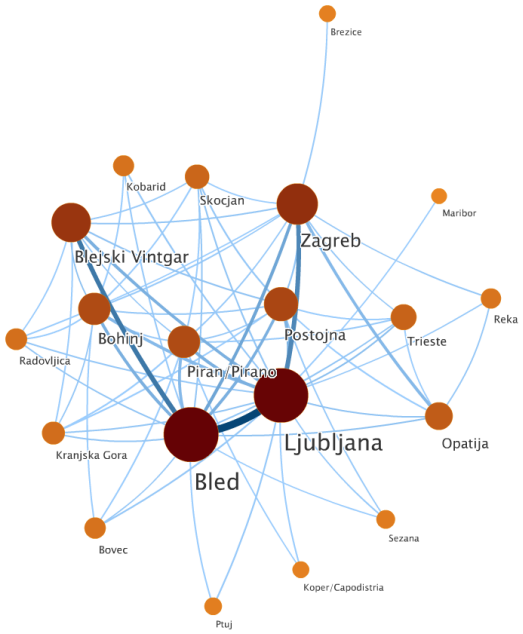
SKUPNOST 8
Bled
Blejski Vintgar

SKUPNOST 9
Radovljica

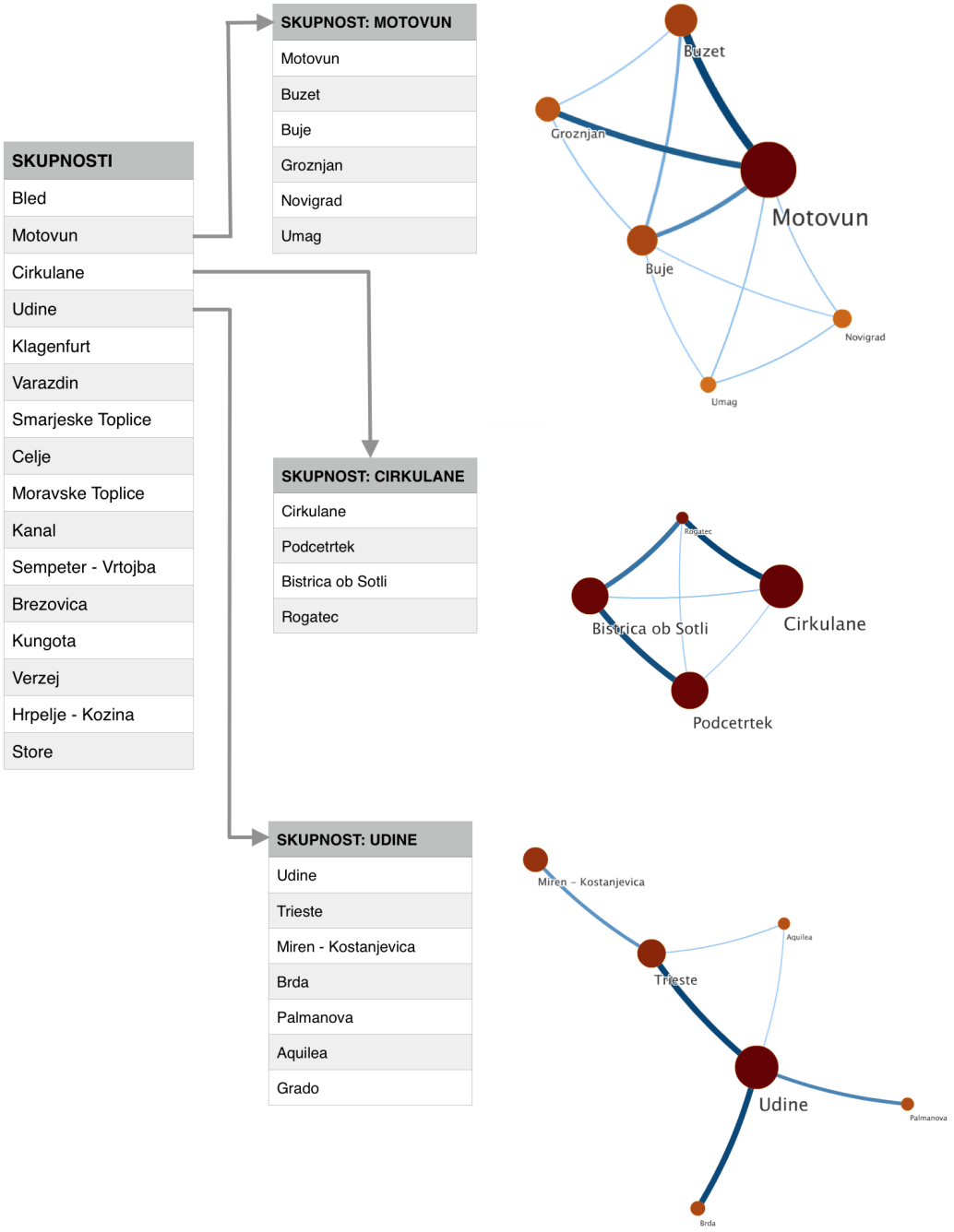
Slika B.1: Skupnosti najdene na grafu vseh zapisov z uporabo Louvaina in parametrom *resolution* enakim 0.5.

SKUPNOSTI
Bled
Motovun
Cirkulane
Udine
Klagenfurt
Varazdin
Smarjeske Toplice
Celje
Moravske Toplice
Kanal
Sempeter - Vrtojba
Brezovica
Kungota
Verzej
Hrpelje - Kozina
Store

SKUPNOST: BLED
Bled
Ljubljana
Zagreb
Blejski Vintgar
Postojna
Bohinj
Piran / Pirano
Opatija
Trieste
Skocjan
Kranjska Gora
Radovljica
Bovec
Kobarid
Reka
Sezana
Ptuj
Brezice
Koper / Capodistria
Tolmin
...



Slika B.2: Glavna skupnost najdena na grafu vseh zapisov z uporabo Infomapa in parametrom *markov-time* enakim 0.75.



Slika B.3: Močnejše skupnosti najdene na grafu vseh zapisov z uporabo Infomapa in parametrom *markov-time* enakim 0.75.

SKUPNOST: KLAGENFURT
Klagenfurt
Villach
Bad Kalinkirchheim

SKUPNOST: VARAZDIN
Varazdin
Cakovec
Razkrizje
Brezice

SKUPNOST: SMARJESKE TOPLICE
Smarjeske Toplice
Dolenjske Toplice
Novo mesto

SKUPNOST: CELJE
Celje
Vojnik

SKUPNOST: MORAVSKE TOPLICE
Moravske Toplice
Murska Sobota

SKUPNOST: KANAL
Kanal
Kobarid

SKUPNOST: SEMPETER - VRTOJBA
Sempeter - Vrtojba
Trieste

SKUPNOST: BREZOVICA
Brezovica

SKUPNOST: KUNGOTA
Kungota
Maribor

SKUPNOST: VERZEJ
Verzej

SKUPNOST: HRPELJE - KOZINA
Hrpelje - Kozina

SKUPNOST: STORE
Store

Slika B.4: Šibkejše skupnosti najdene na grafu vseh zapisov z uporabo Infomapa in parametrom *markov-time* enakim 0.75.

Dodatek C

Izvorna koda

Izvorna koda z navodili je dosegljiva na javnem GitHub repozitoriju: `https://github.com/kknavs/PatternAnalysisTD`.

Literatura

[1] S. urad RS, Povsod je lepo ... Turisti in turizem v številkah, http://www.stat.si/StatWeb/File/DocSysFile/9602/povsod_je_lepo.pdf, [Zadnji dostop 17.7.2018] (2018).

[2] S. turistična organizacija, I feel Slovenia. Slovenski turizem v številkah 2017, https://www.slovenia.info/uploads/dokumenti/raziskave/turizem_v_stevilkah_2017_1.pdf, [Zadnji dostop 17.7.2018] (2018).

[3] S. urad RS, Statistični urad RS. Državna statistika v letu 2017, http://www.stat.si/StatWeb/News/Index/7446/, [Zadnji dostop 17.7.2018] (2018).

[4] R. Leung, H. Q. Vu, J. Rong, Understanding tourists' photo sharing and visit pattern at non-first tier attractions via geotagged photos, INFORMATION TECHNOLOGY & TOURISM 17 (1, SI) (2017) 55–74.

[5] I. Oender, Classifying multi-destination trips in Austria with big data, TOURISM MANAGEMENT PERSPECTIVES 21 (2017) 54–58.

[6] S. turistična organizacija, I feel Slovenia. Tržni profili, https://www.slovenia.info/sl/poslovne-strani/raziskave-in-analize/trzni-profili, [Zadnji dostop 17.7.2018] (2018).

[7] I. Aleksič, Diplomsko delo: Destinacijski management - destinacija Dolenjska, http://www.cek.ef.uni-lj.si/u_diplome/aleksic3354.pdf, [Zadnji dostop 20.9.2018] (2008).

[8] S. J. Miah, H. Q. Vu, J. Gammack, M. McGrath, A big data analytics method for tourist behaviour analysis, Information & Management 54 (6) (2017) 771 – 785, smart Tourism: Traveler, Business, and Organizational Perspectives.

[9] Yahoo Labs. One Hundred Million Creative Commons Flickr Images for Research 2014, https://research.yahoo.com/news/one-hundred-million-creative-commons-flickr-images-research/, [Zadnji dostop 20.11.2017] (2014).

[10] Y. Yuan, M. Medel, Characterizing International Travel Behavior from Geotagged Photos: A Case Study of Flickr, PLOS ONE 11 (5).

[11] M. Ferrante, A. Abbruzzo, S. De Cantis, Graphical models for estimating network determinants of multi-destination trips in Sicily, TOURISM MANAGEMENT PERSPECTIVES 22 (2017) 109–119.

[12] A. Chua, L. Servillo, E. Marcheggiani, A. V. Moere, Mapping Cilento: Using geotagged social media data to characterize tourist flows in southern Italy, TOURISM MANAGEMENT 57 (2016) 295–310.

[13] C.-C. Lue, J. L. Crompton, D. R. Fesenmaier, Conceptualization of multi-destination pleasure trips, Annals of Tourism Research 20 (2) (1993) 289 – 301.
URL http://www.sciencedirect.com/science/article/pii/0160738393900569

[14] I. F. Videla-Cavieres, S. A. Rios, Extending market basket analysis with graph mining techniques: A real case, EXPERT SYSTEMS WITH APPLICATIONS 41 (4, 2) (2014) 1928–1936.

[15] T. Raeder, N. V. Chawla, Market basket analysis with networks, Social Network Analysis and Mining 1 (2) (2011) 97–113.
URL https://doi.org/10.1007/s13278-010-0003-7

[16] S. A. Rios, I. F. Videla-Cavieres, Generating groups of products using graph mining techniques, in: Jedrzejowicz, P and Czarnowski, I and Howlett, RJ and Jain, LC (Ed.), KNOWLEDGE-BASED AND INTELLIGENT INFORMATION & ENGINEERING SYSTEMS 18TH ANNUAL CONFERENCE, KES-2014, Vol. 35 of Procedia Computer Science, Gdynia Maritime Univ; KES Int, 2014, pp. 730–738, 18th Annual International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES), Pomeranian Sci & Technol, Gdynia, POLAND, SEP 15-17, 2014.

[17] J. Yang, J. McAuley, J. Leskovec, Community detection in networks with node attributes, 2013 IEEE 13th International Conference on Data Mining (2013) 1151–1156.

[18] M. Guerrero, F. G. Montoya, R. Banos, A. Alcayde, C. Gil, Adaptive community detection in complex networks using genetic algorithms, NEUROCOMPUTING 266 (2017) 101–113.

[19] R. Agrawal, R. Srikant, Fast algorithms for mining association rules, in: Proc. of 20th Intl. Conf. on VLDB, 1994, pp. 487–499.

[20] G. Boštjančič, Diplomsko delo: Podatkovno rudarjenje s sistemom Oracle Data Mining (11g), `http://eprints.fri.uni-lj.si/1001/1/Bo%C5%A1tjan%C4%8Di%C4%8D_G-_VS.pdf`, [Zadnji dostop 20.11.2017] (2010).

[21] P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Pearson Education, 2006.

[22] M. Kosinski, Y. Wang, H. Lakkaraju, J. Leskovec, Mining Big Data to Extract Patterns and Predict Real-Life Outcomes, PSYCHOLOGICAL METHODS 21 (4, SI) (2016) 493–506.

[23] L. Bohlin, D. Edler, A. Lancichinetti, M. Rosvall, Community Detection and Visualization of Networks with the Map Equation Framework,

Springer International Publishing, Cham, 2014, pp. 3–34.
URL https://doi.org/10.1007/978-3-319-10377-8_1

[24] S. Fortunato, Community detection in graphs, Physics Reports 486 (3) (2010) 75 – 174.
URL http://www.sciencedirect.com/science/article/pii/S0370157309002841

[25] L. Shuo, B. Chai, Discussion of the community detection algorithm based on statistical inference, Perspectives in Science 7 (2016) 122 – 125, 1st Czech-China Scientific Conference 2015.
URL http://www.sciencedirect.com/science/article/pii/S2213020915000658

[26] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment 2008 (10) (2008) P10008.
URL http://stacks.iop.org/1742-5468/2008/i=10/a=P10008

[27] V. Traag, Algorithms and Dynamical Models for Communities and Reputation in Social Networks, Springer, 2014.
URL https://doc.lagout.org/science/0_Computer%20Science/2_Algorithms/Algorithms%20and%20Dynamical%20Models%20for%20Communities%20and%20Reputation%20in%20Social%20Networks%20%5BTraag%202014-05-28%5D.pdf

[28] J. Yang, J. Leskovec, Overlapping community detection at scale: A nonnegative matrix factorization approach, in: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, WSDM '13, ACM, New York, NY, USA, 2013, pp. 587–596.
URL http://doi.acm.org/10.1145/2433396.2433471

[29] A. Lancichinetti, S. Fortunato, Community detection algorithms: A comparative analysis, PHYSICAL REVIEW E 80 (5, 2).

[30] N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks 76 (2007) 036106.

[31] D. Hric, R. K. Darst, S. Fortunato, Community detection in networks: Structural communities versus ground truth 90.

[32] Z. Yang, R. Algesheimer, C. J. Tessone, A Comparative Analysis of Community Detection Algorithms on Artificial Networks, SCIENTIFIC REPORTS 6.

[33] P. Held, B. Krause, R. Kruse, Dynamic Clustering in Social Networks using Louvain and Infomap Method, in: 2016 THIRD EUROPEAN NETWORK INTELLIGENCE CONFERENCE (ENIC 2016), ENGINE Ctr; Wroclaw Univ Sci & Technol; Univ Calgary; Blekinge Inst Technol; Univ Basque Country; AGH Univ Sci & Technol, 2016, pp. 61–68, 3rd European Network Intelligence Conference (ENIC), Wroclaw, POLAND, SEP 05-07, 2016.

[34] M. Rosvall, D. Axelsson, C. T. Bergstrom, The map equation, The European Physical Journal Special Topics 178 (1) (2009) 13–23.
URL `https://doi.org/10.1140/epjst/e2010-01179-1`

[35] L. K. Cvelbar, M. Mayr, D. Vavpotič, Geographical mapping of visitor flow in tourism: A user-generated content approach, Tourism Economics 24 (6) (2018) 701–719.
URL `https://doi.org/10.1177/1354816618776749`

[36] M. Wes, Python for Data Analysis, 1st Edition, O'Reilly Media, Inc., 2012.

[37] J. Leskovec, R. Sosič, Snap: A general-purpose network analysis and graph-mining library, ACM Transactions on Intelligent Systems and Technology (TIST) 8 (1) (2016) 1.

[38] T. igraph core team, igraph - network analysis software, `http://igraph.org/`, [Zadnji dostop 10.7.2018] (2018).

[39] N. developers, Networkx, https://networkx.github.io, [Zadnji dostop 10.7.2018] (2018).

[40] snap.py - snap for python, https://snap.stanford.edu/snappy/index.html, [Zadnji dostop 10.7.2018] (2018).

[41] Engine configuration - sqlalchemy documentation, http://docs.sqlalchemy.org/en/latest/core/engines.html, [Zadnji dostop 25.7.2018] (2018).

[42] I. Černič, Diplomsko delo: Ocena in prikaz statistike turizma v sloveniji za potrebe satelitskih računov v turizmu, http://www.cek.ef.uni-lj.si/u_diplome/cernic1898.pdf, [Zadnji dostop 10.7.2018] (2018).

[43] D. Tatjana Pihlar, Slovenski turizem: še vedno se "drogiramo"le s številom turistov, https://www.dnevnik.si/1042763985, [Zadnji dostop 10.7.2018] (februar 2017).

[44] Tripadvisor: Read reviews, compare prices and book, https://www.tripadvisor.com/, [Zadnji dostop 26.7.2018] (2018).

[45] Tripadvisor.com analytics, https://www.similarweb.com/website/tripadvisor.com, [Zadnji dostop 26.7.2018] (2018).

[46] Eurostat, Statistika turizma - statistics explained, http://ec.europa.eu/eurostat/statistics-explained/index.php?title=Tourism_statistics/sl, [Zadnji dostop 10.7.2018] (2018).

[47] S. urad RS, SURS. Regije v številkah, http://www.stat.si/StatWeb/File/DocSysFile/9959, [Zadnji dostop 17.7.2018] (2018).

[48] Mapequation - network navigator, http://www.mapequation.org/apps/NetworkNavigator.html, [Zadnji dostop 20.9.2018] (2018).