UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Domen Jesenovec

ANALIZA PODATKOV O PROMETNIH NESREČAH V SLOVENIJI

Diplomsko delo
na univerzitetnem študiju

Mentorica: prof. dr. Neža Mramor Kosta
Somentorica: prof. dr. Nada Lavrač

Ljubljana, 2007

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Namesto te strani **vstavite** original izdane teme diplomskega dela s podpisom mentorja in dekana ter žigom fakultete, ki ga diplomant dvigne v študentskem referatu, preden odda izdelek v vezavo!

Zahvala

Zahvaljujem se somentorici prof. dr. Nadi Lavrač, za idejo vsebine diplomske naloge in njeno usmerjanje skozi celoten čas nastajanja diplomske naloge.
Zahvaljujem se mentorici prof. dr. Neži Mramor Kosta za njeno strokovno pomoč in nasvete.
Zahvaljujem se mag. Aleksandru Puru za predstavitev njegovega dela, pomoč pri pridobivanju podatkov o geografskih lokacijah nesreč in dodatna pojasnila o podatkih. Poleg tega je k sodelovanju povabil tudi prometnega strokovnjaka mag. Zvonka Zavasnika, ki se mu zahvaljujem za njegove praktične ocene in komentarje.
Zahvaljujem se Generalni policijski upravi Operativno komunikacijskemu centru (GPU OKC) in predvsem vodji Tomažu Pečjaku, za dostop do podatkov o geografskih lokacijah nesreč in pojasnila glede podatkov.
Zahvala gre tudi dr. Annalisi Appice, ki mi je med svojim začasnim delom na Institutu Jožef Stefan predstavila algoritem SPADA.
Pri spoznavanju zame novega področja geografskih informacijskih sistemov mi je bil s svojimi nasveti v veliko pomoč mag. Andrej Kobler, za kar sem mu lepo zahvaljujem.
Na koncu pa se zahvaljujem tudi svoji družini, katere člani so me podpirali in vzpodbujali skozi celoten čas študija.

Kazalo

Seznam uporabljenih kratic in simbolov

Seznam kratic in simbolov uporabljenih v diplomskem delu:

- SQL - Poizvedbeni jezik (angl. structured query language).
- MySQL - Ime odprtokodne podatkovne baze (*www.mysql.com*).
- GPU - Generalna policijska uprava.
- OKC - Operativno komunikacijski center.
- GURS - Geodetska uprava republike Slovenije (*http://www.gu.gov.si/*).
- SHP - Datotečni format za vektorske geografske podatke (angl. shapefile). Ta standard je razvilo podjetje ESRI (*http://www.esri.com/*), vodilno podjetje na področju geografskih informacijskih sistemov. Običajno ne gre le za eno datoteko, ampak za skupino datotek z različnimi končnicami (shp, dbf ,shx,...).
- GIS - Geografski informacijski sistem (angl. geographic information system).
- D48 - Oznaka slovenskega geografskega koordinatnega sistema (Slovenija je ravno sredi prehoda na nov sistem ESRS).
- WGS84 - Oznaka mednarodnega koordinatnega sistema (angl. world geodetic system) iz leta 1984, ki se med drugim uporablja tudi v GPS navigaciji.
- IBM DB2 - IMB je računalniško podjetje (*http://www.ibm.com*), v katerem je bila razvita podatkovna baza DB2 (*http://www-306.ibm.com/software/data/db2/*).

- KML - Jezik za opis geografskih podatkov (angl. keyhole markup language) (*http://code.google.com/apis/kml/documentation/*).
- XML - Označevalni jezik (angl. extensible markup language), katerega prvotni namen je izmenjava podatkov med različni operacijskimi sistemi in preko interneta.
- ZIP - Je najbolj razširjen format stiskanja računalniških datotek.

Povzetek

Namen diplomskega dela je analiza podatkov o prometnih nesrečah v Sloveniji s tehnikami rudarjenja podatkov. S pomočjo uporabljenih orodij želimo izboljšati razumevanje problemske domene in posledično nuditi podporo pri odločanju. Uporabljene tehnike rudarjenja podatkov so: razvrščanje kratkih časovnih vrst v skupine, razvrščanje točk v skupine in povezovalna pravila.

Podrobno so predstavljeni razpoložljivi podatkovni viri in priprava podatkov za uporabljena orodja. Pregledno je predstavljeno področje razvrščanja kratkih časovnih vrst v skupine. Rezultat uporabe te tehnike so skupine upravnih enot s podobnimi trendi. Sledi opis implementacije razvrščanja točk (nesreč) v skupine v programskem jeziku Java in opis vizualizacije najdenih skupin nesreč v programu Google Earth. Vizualizacija skupin omogoča hitro odkrivanje krajev, kjer je potrebno ukrepanje za zmanjšanje števila prometnih nesreč. Vizualizacija na satelitskih posnetkih v Google Earth pogosto omogoča tudi odkrivanje vzrokov za skupine prometnih nesreč, saj so dobro vidni tudi objekti ob cestah. Povezovalna pravila omogočajo iskanje novih povezav med prometnimi nesrečami. Predstavljena so uporabljena orodja za iskanje povezovalnih pravil in primeri najdenih zanimivih povezovalnih pravil.

Uporabljene tehnike rudarjenja podatkov so omogočile nove poglede na podatke, ki jih je ovrednotil in komentiral ekspert za področje prometne varnosti.

Ključne besede:

GIS, SQL, prometne nesreče, časovne vrste, razvrščanje v skupine, povezovalna pravila

Poglavje 1

Uvod

Prometna varnost je področje, ki je javnosti izjemno zanimivo, saj je danes vsakdo udeleženec v prometu. Dnevno smo deležni poročil o stanju na cestah, opozoril o nesrečah in zastojih. Pri poročilih o prometni varnosti gre običajno le za statistične podatke. Ob množici zbranih in razpoložljivih podatkov se je pojavila možnost uporabe tudi bolj naprednih orodij za rudarjenje podatkov [9] (angl. data mining). Ta orodja ponujajo drugačno pot do novih spoznanj in bi lahko močno pripomogla k boljšim ukrepom policije in boljšemu delu inštitucij zadolženih za načrtovanje prometne infrastrukture. Materialna škoda in predvsem poškodbe in smrti udeležencev nesreč imajo velik vpliv na družbo. Vsi pa si želimo čim manjšega števila prometnih nesreč. Da je rudarjenje podatkov na tem področju lahko koristno dokazujejo tudi številne raziskave s tega področja [5, 6, 10, 14]. V nadaljevanju so podrobneje predstavljeni nekateri pojmi z obravnavanega področja.

1.1 Prostorsko rudarjenje podatkov

Prostorsko rudarjenje podatkov [8] (angl. spatial data mining) je eno od novejših in v zadnjem času hitro se razvijajočih področij računalniške znanosti. Globalni navigacijski sistem (GPS) omogoča enostavno zbiranje podatkov o geografskih lokacijah. Vedno večje zmogljivosti računalniških pomnilnikov omogočajo zbiranje in obdelovanje ogromnih količin podatkov. Zaradi obsežnosti in kompleksnosti podatkovnih baz je analiza in iskanje znanja v obsežnih geografskih podatkovnih bazah zahtevna naloga, ki se jo lotevamo s posebnimi metodami in algoritmi.

1.2 Geografski informacijski sistem

Geografski informacijski sistem [2] (angl. geographic information system) je sistem za zajemanje, shranjevanje, urejanje in analiziranje podatkov, ki so prostorsko povezani z zemeljsko površino. Omogoča torej upravljanje z geografskimi podatki. Področje uporabe geografskih informacijskih sistemov je zelo široko in med drugim obsega: upravljanje virov, okoljske raziskave, prostorsko načrtovanje, kartografijo, kriminaliteto, oglaševanje, promet, itd.

V geografskih informacijskih sistemih so podatki lahko predstavljeni v dveh oblikah [2]: vektorsko in rastersko. Pri vektorski predstavitvi so podatki sestavljeni iz opisov posameznih objektov v prostoru (točke, črte, liki). Pri rasterskih podatkih pa je zemeljska površina razdeljena na mrežo pravokotnih celic (število celic določa ločljivost podatkov). Podatki se v tem primeru nanašajo na posamezno celico (celotno površino celice). Vsaka predstavitev ima svoje prednosti in slabosti. Vektorski podatki so prostorsko varčnejši, vendar pa je njihova obdelava računsko potratnejša. Ravno obratno je z rasterskimi podatki. Uporabljata se obe predstavitvi. Velja pravilo, da se uporabi predstavitev, ki je za rešitev naloge najprimernejša.

1.3 Časovne vrste

Časovna vrsta [8] (angl. time series) je časovno zaporedje meritev istih spremenljivk. So pogosta oblika podatkov in za njihovo obdelavo obstajajo številni algoritmi in metode. V tej diplomski nalogi je uporabljeno razvrščanje časovnih vrst v skupine (angl. time series clustering), torej združevanje podobnih časovnih vrst v skupine. Razvoj algoritmov za razvrščanje (predvsem kratkih) časovnih vrst v skupine je trenutno izjemno zanimivo področje. Razlog so številni praktični problemi s področja biologije in ekonomije, ki se rešujejo prav z iskanjem skupin kratkih časovnih vrst. Primer so analize DNA mikropolj v biologiji.

1.4 Programski jezik R

R[1] je brezplačna različica statističnega programskega jezika S. Vključuje tudi veliko paketov zanimivih za področje te diplomske naloge. Med drugim vsebuje pakete za rudarjenje podatkov, pakete za urejanje geografskih podatkov in pakete za analiziranje geografskih podatkov. Največja moč programskega

[1] Domača stran projekta R: *http://cran.r-project.org/*

jezika R je množica razpoložljivih funkcionalnosti v obliki paketov in enostavnost izdelave vizualizacij.

1.5 Google Earth

Google Earth[2] je program, ki je uporabnikom brezplačno na voljo na internetu. Omogoča pregledovanje satelitskih posnetkov za celotno zemeljsko površino. Poleg tega omogoča prikazovanje zunanjih podatkov na razpoložljivih satelitskih slikah. Podatki morajo biti v datoteki formata KML (angl. keyhole markup language). Format KML omogoča prikaz točk, poti in celo tri-dimenzionalnih objektov (npr. stavb).

1.6 SPADA

SPADA[3] [1] (angl. spatial pattern discovery algorithm) je algoritm za iskanje prostorskih povezovalnih pravil [8] (angl. spatial association rules) v podatkih. Algoritem išče povezovalna pravila v podatkih, ki so lahko tudi iz večih tabel - relacij. Algoritem je bil razvit v okviru projekta SPIN[4] (angl. spatial mining for data of public interest).

1.7 Motivacija in struktura diplomskega dela

Glavna motivacija za analiziranje podatkov o prometnih nesrečah je spoznavanje novega področja računalniške znanosti (prostorsko rudarjenje podatkov, geografski informacijski sistemi). Opravljene analize ponujajo nov pogled na podatke in tudi možnost primerjave rezultatov s podobno raziskavo [10] opravljeno na podatkih za Veliko Britanijo v okviru projekta SolEUNet[5].

Struktura diplomskega dela je sledeča. Poglavje 2 vsebuje opis uporabljenih podatkov, opis priprave podatkov in nekaj osnovnih predstavitev podatkov. Poglavje 3 predstavi razvrščanje časovnih vrst v skupine, poglavje 4 je namenjeno razvrščanju prometnih nesreč v skupine, poglavje 5 pa iskanju povezovalnih pravil. Sledi poglavje 6, ki zaključuje diplomsko delo s sklepnimi ugotovitvami.

[2]Domača stran programa Google Earth: *http://earth.google.com/*

[3]Domača stran algoritma SPADA: *http://www.di.uniba.it/ malerba/software/ ARES/index.htm*

[4]Domača stran projekta SPIN: *http://www.ais.fraunhofer.de/KD/SPIN/index.html*

[5]Domača stran projekta SolEUNet: *http://soleunet.ijs.si/website/html/euproject.html*

Poglavje 2

Podatki

V tem poglavju so podrobno predstavljene vse podatkovne zbirke uporabljene v okviru tega diplomskega dela. Opisan je celoten postopek priprave podatkov, ki je vključeval tudi spremembo koordinatnega sistema pri geografskih atributih. Kot rezultat je nastala enotna podatkovna baza, ki se je uporabljala pri vseh nadaljnih analizah. Za boljše razumevanje podatkov so bile opravljene osnovne statistične predstavitve podatkov.

2.1 Podatkovne zbirke

Glavnino uporabljenih podatkov tvorijo podatki (statistične datoteke) pridobljeni iz spletnih strani slovenske policije[1].

Podatki so v času, ko se je delo na diplomski nalogi začelo, obsegali obdobje od leta 1995 do leta 2005. Ravno ob koncu dela pa so bili dodani še podatki za leto 2006, ki so v tej diplomski nalogi uporabljeni le pri nekaterih analizah.

Poleg podatkov o nesrečah, vsebovanih v statističnih datotekah, so za prostorsko rudarjenje podatkov potrebni tudi podatki o geografskih lokacijah nesreč. Te podatke (datoteke formata SHP) smo pridobili na GPU OKC (generalna policijska uprava, operativno komunikacijski center).
Za risanje mej upravnih enot smo potrebovali še geogrfske podatke o mejnih točkah upravnih enot. Tovrstni podatki so prosto dostopni na spletnih straneh GURS[2].

[1]Statistične datoteke: *http://www.policija.si/portal/statistika/promet/promet.php*

[2]Brezplačni podatki o upravnih enotah: *http://www.gu.gov.si/si/delovnapodrocja_gu/podatki_gu/brezplani_podatki/brezplani_podatki_upravne_enote/*

2.2 Opis podatkov

2.2.1 Statistične datoteke

Podatki za posamezno leto so vsebovani v relacijski podatkovni bazi, razdeljeni v dve datoteki. Ena datoteka vsebuje podatke o nesrečah, druga pa podatke o osebah udeleženih v nesrečah. Ena vrstica vsebuje podatke o eni nesreči ali osebi. V statističnih datotekah je zabeleženo skupno 453451 nesreč v letih 1995-2005 (približno 41000 nesreč letno). V teh prometnih nesrečah je bilo udeleženih 866296 oseb (približno 79000 oseb letno).

Struktura datoteke s podatki o nesrečah:

1. številka zadeve - enoznačna številka zadeve, pod katero policija vodi posamezno prometno nesrečo
2. klasifikacija nesreče glede na posledice (šifrant PRPO)
3. upravna enota, na območju katere se je zgodila prometna nesreča (šifrant LOOB)
4. datum nesreče - v obliki: DD.MM.YYYY
5. ura nesreče - v obliki: HH.MM (novi format: HH)
6. indikator ali se je nesreča zgodila v naselju ali izven
7. kategorija ceste na kateri je prišlo do nesreče (šifrant: LOVC)
8. oznaka ceste ali šifra naselja, kjer je prišlo do nesreče
9. tekst ceste ali naselja, kjer je prišlo do nesreče
10. oznaka odseka ceste ali šifra ulice, kjer je prišlo do nesreče
11. tekst odseka ali ulice, kjer je prišlo do nesreče
12. točna stacionaža ali hišna številka, kjer je prišlo do nesreče
13. opis prizorišča nesreče (šifrant: PRKD)
14. glavni vzrok nesreče (šifrant: PRVZ)
15. tip nesreče (šifrant: PRTN)

16. vremenske okoliščine v času nesreče (šifrant: PRVR)
17. stanje prometa v času nesreče (šifrant: PRSP)
18. stanje vozišča v času nesreče (šifrant: PRPV)
19. stanje površine vozišča v času nesreče (šifrant: PRSV)

Struktura datoteke s podatki o osebah:

1. številka zadeve, povezovalni parameter na bazo prometnih nesreč
2. kot kaj nastopa oseba v prometni nesreči (povzročitelj ali udeleženec)
3. starost osebe - starost v obliki: LLMM (novi format: LL)
4. spol - (1 - moški, 2 - ženski)
5. občina stalnega prebivališča - (šifrant: LOOB)
6. državljanstvo osebe - (šifrant: LODZ)
7. poškodba osebe (šifrant: PRPO)
8. vrsta udeleženca v prometu (šifrant: PRVU)
9. ali je oseba uporabljala varnostni pas ali čelado (polje se interpretira v odvisnosti od vrste udeleženca)
10. vozniški staž osebe za kategorijo, ki jo potrebuje glede na vrsto udeleženca v prometu, v obliki: LLMM
11. vrednost alkotesta za osebo, če je bil opravljen, v obliki: n.nn
12. vrednost strokovnega pregleda za osebo, če je bil odrejen in so rezultati že znani, v obliki: n.nn

Uporabljeni šifranti so priloženi v poglavju priloge.

Med datotekami za leta 1995-2000 (starejši format) in kasnejšimi (novejši format) obstajajo manjše razlike.

Razlike pri podatkih o nesrečah:

- ločilo - Pri starejšem formatu ločila med posameznimi atributi ni bilo, predpisana je bila dolžina (število znakov) posameznega atributa. V novejšem formatu pa se za ločilo uporablja znak "$".
- šifra nesreče - Pri starejšem formatu je bila ta številka 9-mestna, kasneje le še 6-mestna.
- ura nesreče - Pri starejšem formatu je bil čas nesreče podan v urah in minutah (v obliki HH.MM), v novejšem pa le še v urah (v obliki HH).
- indikator ali se je nesreča zgodila v naselju ali izven - V starejšem formatu je ta idikator imel vrednosti D (v naselju) in N (izven naselja). V novem formatu pa sta vrednosti 1 (v naselju) in 0 (izven naselja).
- stacionažna ali hišna številka - V starejšem formatu je bila ta številka 4-mestna, v novem formatu pa je 9-mestna.
- stanje vozišča in stanje površine vozišča - Ta dva atributa sta v različnem vrstnem redu v novem in starem formatu.

Razlike pri podatkih o osebah:

- ločilo - Enako kot pri datotekah o nesrečah novi format uporablja znak "$", stari pa predpisane dolžine atributov.
- šifra nesreče - Tako kot pri nesrečah je tudi tukaj v novem formatu šifra krajša (namesto 9 samo 6 znakov)
- kot kaj nastopa oseba v prometni nesreči - Stare vrednosti tega indikatorja so D (povzročitelj) in N (udeleženec). Nove vrednosti pa so 1 (povzročitelj) in 0 (udeleženec).
- starost osebe - V starejših podatkih je bila starost podana v letih in mesecih (v obliki LLMM) v novejšem formatu pa je starost podana le še v letih (v obliki LL).
- ali je oseba uporabljala varnostni pas/čelado - Vrednosti tega atributa v starem formatu: 0 (manjkajoč podatek), D (uporabljen varnostni pas/čelada), N (ni uporabljen varnostni pas/čelada). Vrednosti atributa v novem formatu: 0 (ni uporabljen varnostni pas/čelada), 1 (uporabljen varnostni pas/čelada), 2 ali * (manjkajoč podatek).

- alkotest - Pred kratkim je policija spremenila mersko enoto, ki se uporablja za merjenje alkoholiziranosti. Stara enota je bila gram alkohola na kilogram krvi, nova pa je miligram alkohola v litru izdihanega zraka. V razpoložljivih podatkih so še stare enote za merjenje alkoholiziranosti, v novejših podatkih pa bodo uporabljene nove. Pred analizo novejših podatkov bi bilo potrebno izvedeti, kdaj točno se je zgodila ta sprememba.

2.2.2 SHP datoteke prometnih nesreč

Za izrisovanje nesreč z uporabo geografskih informacijskih sistemov oziroma analiziranje prostorskih povezav med nesrečami so nujno potrebni podatki o geografski lokaciji nesreč. Te podatke smo pridobili z GPU OKC (Generalna policijska uprava Operativno komunikacijski center) v obliki SHP datotek.

SHP je datotečni standard najbolj znanega izdelovalca GIS orodij - ESRI, namenjen vektorskemu opisu geografskih podatkov. Podatki za eno leto so sestavljeni iz treh obveznih datotek (SHP - opisuje objekte, DBF - tabela podatkov, vrstica ustreza enemu objektu iz SHP, SHX - indeksna datoteka objektov) in dveh pomožnih indeksnih datotek (SBN, SBX). Datoteka SHP vsebuje točke, DBF pa podatke vezane na točko - nesrečo. Podatki iz datoteke DBF so isti kot podatki, predhodno pridobljeni iz statističnih datotek.

Podatki o prometnih nesrečah se v osrednjo podatkovno bazo vnašajo na podlagi policijskih zapisnikov. Od dogodka pa do vnosa v bazo sme preteči največ 15 dni. Vnos v bazo lahko opravi policist sam ali pa administrativno osebje na policijski postaji. Na kraju nesreče policist lokacije ne določi z GPS napravo temveč jo opiše z oznako ceste/naselja, imenom ulice in stacionažno/hišno številko. Na podlagi teh podatkov se nesreči avtomatično pripišejo geografske koordinate (z uporabo šifrantov naselij, ulic, stacionažnih/hišnih številk). Proces določanja koordinat je popolnoma avtomatiziran in ni stoodstotno uspešen. Nesrečam, ki jim sistem ne uspe določiti koordinat, se pripiše koordinate zunaj Slovenskih meja (torej bodo upoštevane pri analizah vezanih na ostale - negeografske atribute). Takih nesreč je nekje od 5 do 10% letno. Ročno se njihovih podatkov ne obdeluje, ker se predvideva, da so porazdeljene tako kot ostale nesreče, ki še vedno predstavljajo zadovoljivo velik vzorec za analize. Ažurne šifrante naselij, cest in ulic pridobijo z ministrstva za okolje in prostor.

Podatki v policijski podatkovni bazi niso sami sebi namen in se na njih opravljajo številne analize. Uporabljajo se GIS orodja (policija ima lasten GIS sistem zasnovan na osnovi ArcView (program podjetja ESRI) in poenostavljen GIS sistem, dosegljiv vsem uslužbencem preko intraneta). Poleg teh pa uporabljena podatkovna baza IBM DB2 ponuja tudi množico orodij za rudar-

jenje podatkov.

Podatki niso popolnoma točni, saj je npr. možno, da je oseba udeležena v nesreči umrla v bolnici, v bazi pa je zabeležena težka poškodba. Ker se statistične datoteke izdelajo ob koncu koledarskega leta, spremembe pa se v bazo lahko vnašajo za nazaj, tudi tu prihaja do odstopanj. Potrebno je omeniti odstopanja v številu nesreč med statističnimi datotekami in SHP datotekami.

2.2.3 SHP datoteka mej upravnih enot

Datoteke z mejami upravnih enot za razliko od datotek prometnih nesreč ne vsebujejo opisov točk, temveč vsebujejo opise poligonov. Datoteke DBF pa vsebujejo le en podatek za vsak objekt (upravno enoto) in sicer njeno ime. Podatki o mejah upravih enot so brezplačno na voljo na spletnih straneh GURS.

2.3 Priprava podatkov

Podatke je pred uporabo potrebno ustrezno pripraviti, kar pomeni predvsem ukrepanje v primeru manjkajočih vrednosti in odpravljanje nekonsistentnosti v podatkih. Poleg tega se je pri geografskih atributih pokazala potreba po spremembi koordinatnega sistema.

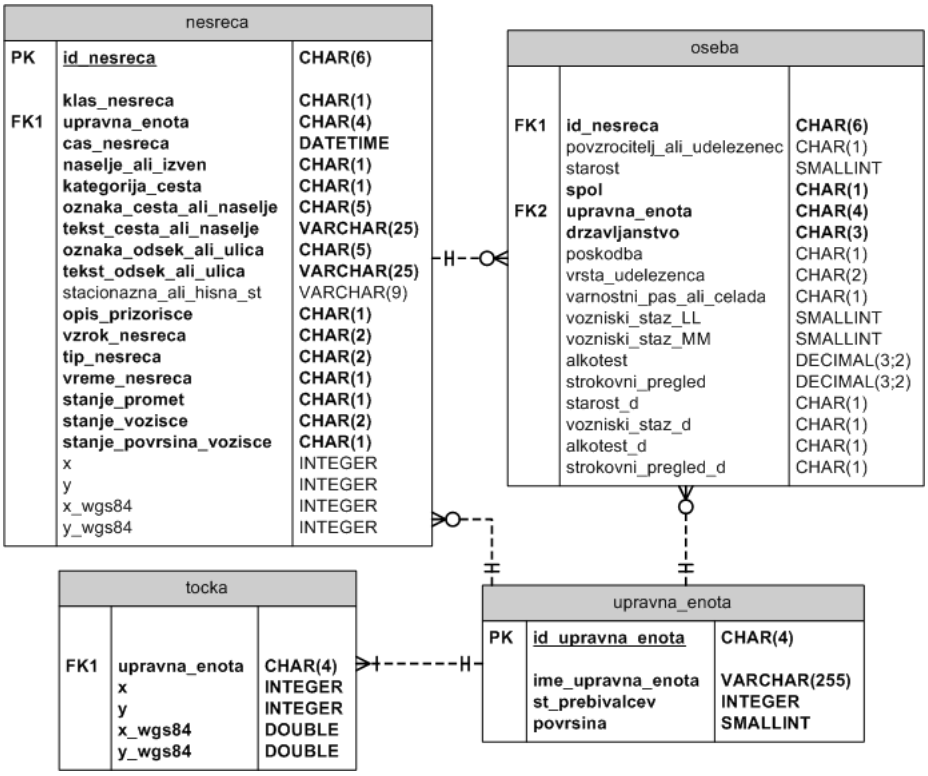
2.3.1 ER diagram podatkovne baze

Podatke sem iz statističnih datotek prenesel v podatkovno bazo (izbral sem priljubljeno relacijsko bazo MySQL), zato ker je nad podatki v podatkovni bazi s pomočjo poizvedovlnega jezika SQL zelo enostavno izvajati kompleksne poizvedbe in s tem izbore podatkov. Ker je za vsako obdelavo potreben drugačen izbor zapisov in nabor atributov, je ta možnost izjemnega pomena. Poleg tega pa je zelo enostavno poiskati in ustrezno popraviti manjkajoče vrednosti. Za prenos podatkov iz tekstovnih datotek v datoteke primerne za polnjenje podatkovne baze sem napisal nekaj krajših programov v jeziku C.

Pri izbiri podatkovnih tipov sem se trudil kar najbolj varčevati s prostorom, saj so pri tako obsežni podatkovni bazi prihranki opazni. Varčevanje s prostorom pa pomeni tudi večjo hitrost. Kjer je velikost znakovnih atributov stalna, sem uporabljal podatkovni tip CHAR(x), kjer je x dolžina niza (tako se za vsak znak porabi en bajt). Kjer je dolžina znakovnega niza spremenljiva, pa sem uporabil tip VARCHAR(x), kjer je x največja dolžina niza. Tukaj se porabi bajt za vsak znak in dodaten bajt, ki podaja dolžino niza. Pri numeričnih atributih sem uporabil TINYINT UNSIGNED (porabi le en bajt

in dopušča vrednosti med 0 in 255). Na entitetno-relacijskem (ER) diagramu [11] je namesto podatkovnega tipa TINYINT uporabljen SMALLINT, ker je TINYINT tip značilen le za MySQL podatkovno bazo. Uporabljeni model podatkovne baze je predstavljen na entitetno-relacijskem diagramu na sliki 2.1.

Poleg tabel *nesreča* in *oseba*, ki izhajata neposredno iz statističnih datotek, sta v bazi še tabeli *upravna_enota* in *točka*. V tabeli *upravna_enota* se hranijo osnovni podatki o upravnih enotah, ki omogočajo primerjavo upravnih enot glede na število nesreč na prebivalca, ali pa število nesreč na km^2. Tabela *točka* pa hrani mejne točke za vse upravne enote v D48 koordinatnem sistemu in WGS84 sistemu. Podatki o upravnih enotah so pridobljeni s portala upravnih enot[3].



Slika 2.1: Entitetno-relacijski (ER) diagram.

[3]Portal upravnih enot: *http://upravneenote.gov.si*

2.3.2 Obdelava statističnih datotek

Vnašanje podatkov v podatkovno bazo je možno na dva načina:

- ukaz INSERT - Običajen način vnašanja, za vnos vsakega zapisa je potreben en *insert* ukaz.
- ukaz LOAD DATA - Ukaz omogoča prenos vsebine datoteke neposredno v bazo z enim ukazom. Omogoča tudi definiranje ločila med zapisi, ločila med vrsticami, zato je možno polnenje iz skoraj vsake tekstovne datoteke (npr. csv).

Prednost ukaza *load data* je v hitrosti, saj je tak vnos podatkov običajno 20-krat hitrejši od *insert* ukazov. V primeru baze prometnih nesreč gre za veliko količino podatkov, zato sem za polnenje baze uporabil ukaz *load data*.

Podatke v statističnih datotekah sem pretvoril v datoteke ločene s tabulatorjem. Ta postopek je obsegal:

- Uskladitev vseh razlik med novim in starim formatom statističnih datotek. Datoteke v novem formatu sem prevedel na starejši format, ki je pri nekaterih atributih natančnejši (npr. čas nesreče, kjer so v starem formatu podane tudi minute, v novem formatu sem dodal ničle za minute). Izjema je atribut šifra nesreče, kjer sem uporabil 6 znakov zaradi varčevanja s prostorom.
- Kjer je pri nesreči ali osebi manjkal podatek o upravni enoti, sem prazno polje nadomestil z vrednostjo 5599, ki je rezervirana za neznano upravno enoto v šifrantu upravnih enot.
- Pri podatkih v letih 2001, 2002, 2003, 2004 je verjetno prišlo do napake pri kreiranju statističnih datotek, saj je atribut stanje površine vozišča skrajšan na en znak (namesto običajnih vrednosti AH, AN, AZ, MA, OS, so uporabljene vrednosti A, M, O). Zaradi uskladitve podatkov sem ta atribut na en znak skrajšal tudi pri statističnih datotekah za ostala leta.
- Manjkajoč podatek o državljanstvu oseb je bil povsod zamenjan z vrednostjo 999, ki v šifrantu državljanstev predstavlja neznano državljanstvo.
- Manjkajoče vrednosti pri atributih, kjer ni definirane posebne vrednosti za manjkajoč podatek, so bile zamenjane z posebnim nizom "\N", ki ga MySQL prepozna kot vrednost *null* (torej manjkajoč podatek).

Zanimivosti, opažene med obdelavo statističnih datotek:

- Šifrant PRPO (Poškodba osebe ali klasifikacija nesreče) - Poleg vrednosti navedenih v šifrantih se leta 2004 začne pojavljati vrednost U, ki predstavlja uradni zaznamek (zaradi zavarovalnic).
- Šifrant LOVC (Kategorija ceste, naselja) - Poleg vrednosti navedenih v šifrantu se pojavljajo tudi vrednosti: M - magistralna cesta, R - regionalna cesta, A - lokalna cesta.
- Šifrant PRKD (Opis kraja dogodka) - V šifrantu so vrednosti, ki niso nikoli uporabljene: E - železniško postajališče, O - naravovarstveno območje, V - vlak. Od leta 2001 naprej pa se uporabljata le še vrednosti: N - naravno okolje, C - cesta.
- Šifrant PRSV (Vrsta vozišča v času prometne nesreče) - V letih 2001 - 2004 je prišlo do napake in je ta atribut skrajšan na en znak. To pomeni tri različne vrednosti namesto petih.

Opravljena je bila tudi diskretizacija numeričnih atributov pri podatkih o osebah (starost, vozniški staž, alkotest, strokovni pregled). Diskretizacija je potrebna, ker večina programov za iskanje povezovalnih pravil ne deluje nad numeričnimi (zveznimi) atributi.

Podatki v tabeli *nesreča* so zelo kvalitetni, saj je zelo malo manjkajočih vrednosti atributov. Manjkajoče vrednosti se pojavljajo le pri atributih: upravna enota (0.01% nesreč), stacionazna_ali_hisna_st (40%). Drugače je pri tabeli oseb, kjer je manjkajočih vrednosti precej več, npr. atributi: starost (0.14%), spol (0.38%), alkotest (43%). Med razlogi za več neznanih vrednosti pri osebah so udeleženci, ki so pobegli s kraja nesreče in pa udeleženci, ki jih ni bilo možno identificirati zaradi prehudih poškodb. Podatek o opravljenem alkotestu pa pogosto manjka zato, ker alkotest ni opravljen pri vseh udeležencih prometnih nesreč.

2.3.3 Obdelava SHP datotek

Iz datotek SHP je bilo potrebno izluščiti geografske koordinate prometnih nesreč in jih dodati v podatkovno bazo. Obstajajo brezplačna orodja za pregledovanje datotek SHP, vendar pa so le redka sposobna tudi zahtevnejših operacij. Za to nalogo sem izbral programski jezik R, ki ima knjižnico *shapefiles* z ukazi za urejanje in ustvarjanje datotek SHP. Predstavljena je celotna koda kratkega programa, ki iz datoteke SHP naredi tekstovno datoteko s koordinatami, s tabulatorjem kot ločilom.

```
# uvoz knjižnice za SHP datoteke
library(shapefiles)

# Branje datoteke SHP
shapefile = read.shapefile("pn00")

# Dodajanje X in Y na seznam atributov
shapefile = add.xy(shapefile)

# Shranjevanje podatkov v tekstovno datoteko
write.table(shapefile$dbf$dbf[,c("ID_PN","X","Y")], file="pn00xy.csv",
append=F, quote=F, sep=";", na="\N", dec=".", row.names=F,
col.names=F)
```

Ustvarjeno datoteko sem s pomočjo enostavnega programa v jeziku Java spremenil v *update* ukaze SQL, ki so obstoječim zapisom dodali atributa z geografsko lokacijo.

Med obdelavo SHP datotek sem opazil odstopanja v številu nesreč v statističnih datotekah in SHP datotekah. Primerjavo števila nesreč v obeh podatkovnih zbirkah prikazuja tabela 2.1.

	statistične datoteke	SHP datoteke
1995	38923	38915
1996	39109	39108
1997	40639	40639
1998	36704	32412
1999	40687	40663
2000	39297	39533
2001	39722	39722
2002	39733	39601
2003	41319	41173
2004	43156	42177
2005	54162	31663
2006	55543	31966

Tabela 2.1: Primerjava števila nesreč v različnih podatkovnih zbirkah.

Podobno sem iz datoteke SHP v tekstovno obliko izpisal tudi mejne točke upravnih enot. Za vsako upravno enoto sem ustvaril datoteko njenih mejnih točk (tabulator kot ločilo).

2.3.4 Pretvorba koordinat

Koordinate prometnih nesreč pridobljene iz SHP datotek so bile v slovenskem državnem koordinatnem sistemu D48. To pomeni, da so ti podatki neposredno uporabni samo v kombinaciji z podatki v istem koordinatnem sistemu (npr. zemljevidi s katerimi razpolaga GURS). Da bi lahko za vizualizacijo uporabil program Google Earth, je bila potrebna pretvorba koordinat nesreč v mednarodni koordinatni sistem WGS84. Algoritem za pretvorbo je kompleksen in ni nikjer na internetu javno dokumentiran. Obstajajo le različne aplikacije, ki opravljajo pretvorbe med koordinatnimi sistemi. Komercialni aplikaciji za pretvorbe sta TRANSDAT[4] in CoordTrans[5]. Obe imata prijazen grafični vmesnik, vendar sta v preizkusni različici preveč omejeni. Za pretvorbo sem zato uporabil brezplačni program PROJ.4 [6]. Program nima grafičnega vmesnika, uporablja se preko konzole. Za pretvorbe v in iz D48 je potrebna datoteka s parametri koordinatnega sistema D48. Spodaj je predstavljena vsebina te datoteke (ime datoteke je *user*).

⟨slovenia⟩ +proj=tmerc +lat_0=0 +lon_0=15 +k=0.999900 +x_0=500000 +y_0=-5000000 +ellps=bessel +units=m +towgs84=668,-205,472 +no_defs no_defs ⟨⟩

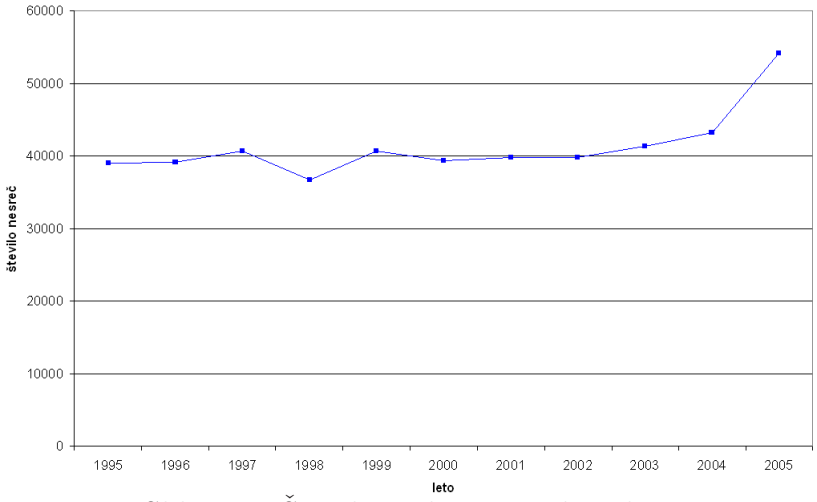
Ukaz, ki s pomočjo datoteke user pretvori koordinate iz sistema D48 (datoteka d48data.txt) v kooridinatni sistem WGS84 (datoteka wgs84data.txt) je:
cs2cs -I +datum=WGS84 +proj=longlat +to +init=user:slovenia d48data.txt -f %0.6f ⟩ wgs84data.txt

Poleg podatkov o geografskih lokacijah prometnih nesreč, je bilo potrebno v sistem WGS84 pretvoriti tudi mejne točke upravnih enot. Tako sta za vsako upravno enoto nastali dve datoteki (ena s točkami v D48 sistemu in druga s točkami v WGS84 sistemu). Ti dve datoteki sem nato z enostavnim programom v jeziku Java združil v datoteko primerno za polnenje baze z ukazom *load data*.

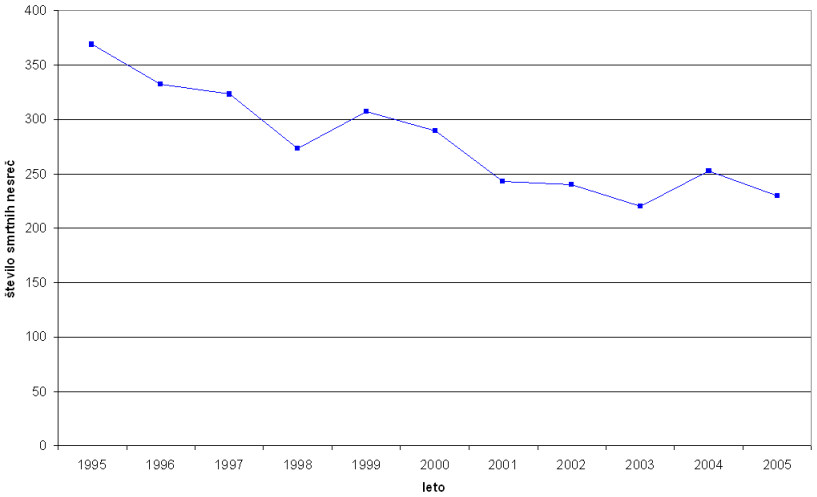
[4]Domača stran programa TRANSDAT: *http://www.killetsoft.de/p_trds_e.htm*
[5]Domača stran programa CoordTrans: *http://franson.com/coordtrans/index.asp*
[6]Domača stran programa PROJ.4: *http://proj.maptools.org/*

2.4 Statistične predstavitve podatkov

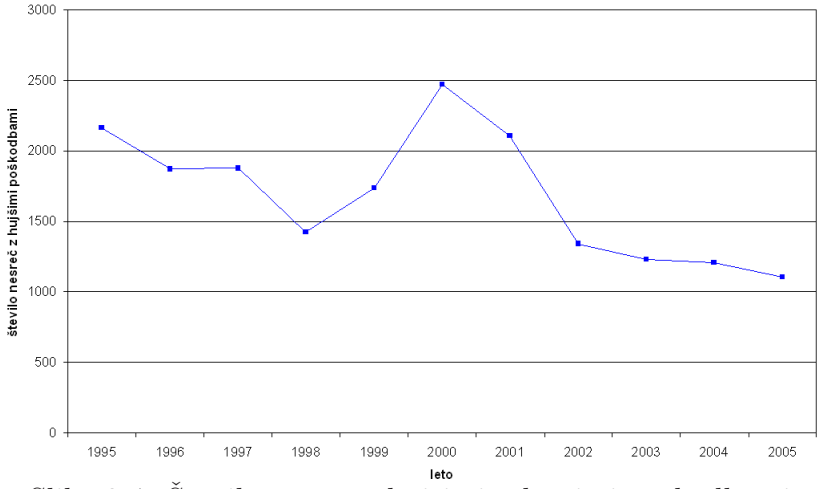
Podatki v podatkovni bazi omogočajo enostaven izbor podatkov s poizvedbenim jezkom SQL. Poleg tega pa SQL omogoča tudi izpis rezultata poizvedbe v tekstovno datoteko in formatiranje izpisa. Tako lahko z eno poizvedbo pripravimo podatke za obdelavo v ostalih programih (statistične obdelave, rudarjenje podatkov, vizualizacije, ...). V tem razdelku je predstavljenih nekaj osnovnih vizualizacij podatkov, pridobljenih na ta način. Namen teh vizualizacij je spoznavanje podatkov pred začetkom ostalih obdelav.



Slika 2.2: Število vseh nesreč skozi leta.

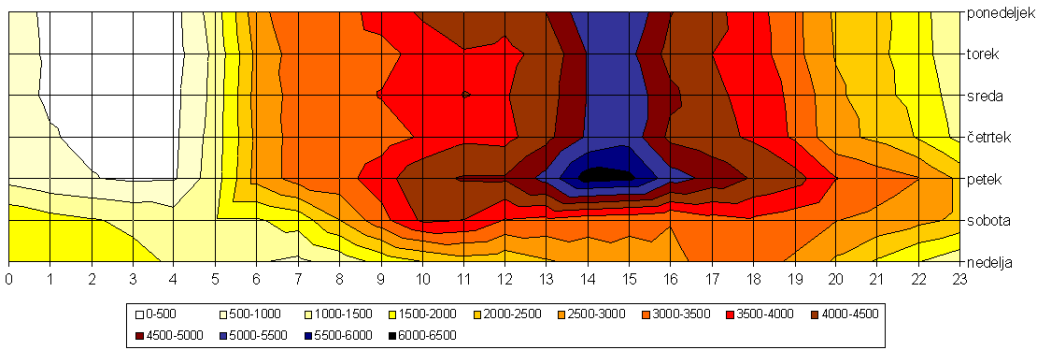


Slika 2.3: Število nesreč s smrtnim izidom.

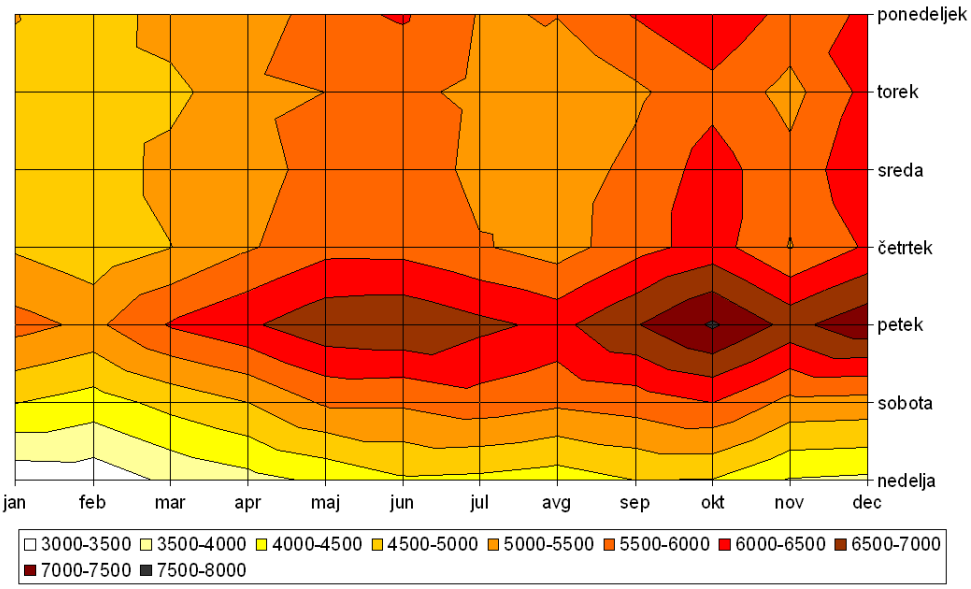


Slika 2.4: Število nesreč s hujšimi telesnimi poškodbami.

Iz grafov števila nesreč skozi leta (slike: 2.2, 2.3, 2.4) je razvidno, da se skupno število nesreč skozi leta ni bistveno spreminjalo, v zadnjih letih je opazen celo porast števila nesreč. Nasprotno pa je opazen trend zmanjševanja števila nesreč s smrtnim izidom in hujšimi telesnimi poškodbami. Večje število nesreč lahko razumemo predvsem kot posledico povečanja prometa. Med razlogi za zmanjševanje števila hudih prometnih nesreč pa so: boljša prometna infrastruktura, boljši avtomobili, uspešno delo policije (in ostalih organov zadolženih za prometno varnost), itd.



Slika 2.5: Porazdelitev nesreč skozi ure dneva in dni v tednu.



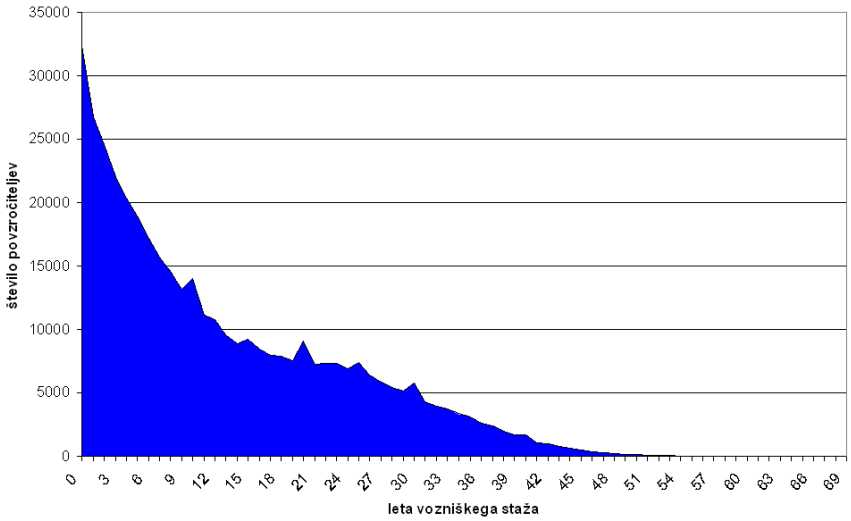
Slika 2.6: Porazdelitev nesreč skozi mesece in dni v tednu.

Iz grafov porazdelitve števila prometnih nesreč skozi različna časovna obdobja (slike: 2.6, 2.5) so jasno opazne časovne zgostitve prometnih nesreč.

Gledano po mesecih se število nesreč poveča zgodaj poleti (maj, junij), se pozno poleti zmanjša in proti koncu leta spet povečuje (vrhunec oktobra in decembra). Prometni strokovnjak je to gibnje povezal z vremenom. Opažajo namreč, da se število nesreč poveča v primeru, ko lepo vreme sledi daljšemu obdobju slabega vremena. Takrat naj bi bili ljudje bolj sproščeni in vozili hitreje in manj previdno.

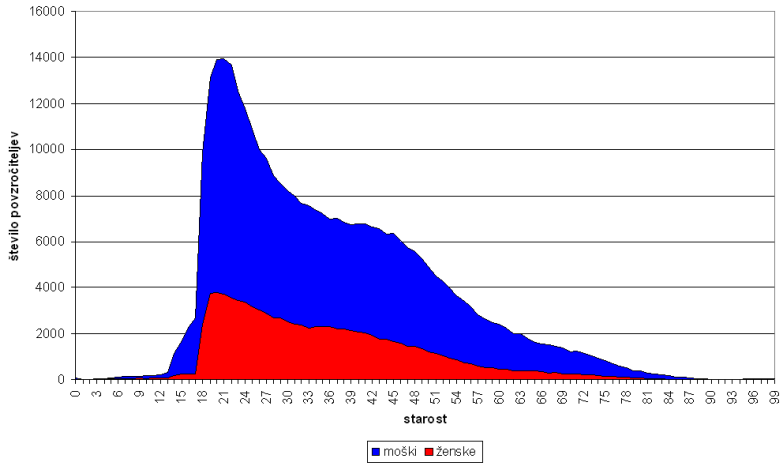
Število nesreč skozi dneve tedna narašča proti koncu tedna, z vrhuncem v petek. To gibanje bi lahko povezali z utrujenostjo ljudi, ki se preko delovnih dni povečuje. Vrhunec v petek pa je očiten odraz dejstva, da je v Sloveniji petek dan za zabavo in izlete, kar vse prepogosto pomeni alkoholizirane in objestne udeležence v prometu. Poleg tega je v petek večja gostota prometa zaradi tedenskih migracij.

Po urah dneva število nesreč narašča proti 14. in 15. uri, kjer je vrhunec, nato pa spet pada. Najočitnejši razlog take porazdelitve je dejstvo, da je takrat konec večine dopoldanskih izmen in začetek popoldanskih. Zanimivo je, da ni opaznega povečanja zjutraj, ko ljudje odahajajo na delo. To je bistveno odstopanje od podatkov za Veliko Britanijo [10], kjer je opazen vrhunec števila nesreč tudi ob 8. uri zjutraj.

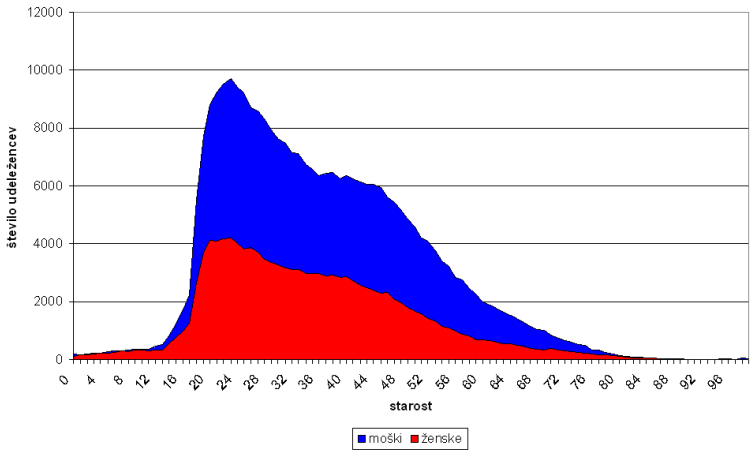


Slika 2.7: Število nesreč glede na vozniški staž.

Graf števila nesreč glede na vozniški staž (slika 2.7) je zelo zgovoren in jasno kaže, da so vozniki s krajšim vozniškim stažem nevarnejši. Seveda se je potrebno zavedati tudi porazdelitve voznikov po starostnih skupinah.



Slika 2.8: Število povzročiteljev glede na starost in spol.



Slika 2.9: Število udeležencev glede na starost in spol.

Zgovorna sta tudi grafa števila udeležencev v nesrečah glede na spol in starost (sliki: 2.8, 2.9). Oba grafa kažeta vrhunec pri starosti med 20 in 25 letom. Poleg tega pa še veliko prevlado moških, predvsem pri povzročiteljih. Eden od razlogov za večje število moških povzročiteljev in udeležencev je zagotovo večje število moških voznikov. Ženske pa so v splošnem tudi bolj previdne in manj agresivne za volanom, kar je potrdil tudi prometni strokovnjak.

Poglavje 3

Analiza časovnih vrst

Po zgledu analize opravljene v [10] sem opravil razvrščanje kratkih časovnih vrst v skupine. Pri tem sem preizkusil metodo uporabljeno v [10] in podrobneje predstavljeno v [15].

3.1 Algoritem za razvrščanje v skupine

Poznani sta dve vrsti algoritmov za razvrščanje v skupine: hierarhični in particijski. Hierarhični algoritmi iščejo skupine v iteracijah (na podlagi prejšnjih skupin), particijski pa določijo vse skupine naenkrat. Hierarhični algoritmi so lahko združevalni (angl. agglomerative, bottom-up), ali pa razdruževalni (angl. divisive, top-down). Združevalni začnejo z vsakim elementom kot svojo skupino in skupine postopoma združujejo v večje skupine. Razdruževalni algoritmi začnejo s skupino vseh elementov in jo postopoma delijo na manjše skupine [13, 16].

Odločil sem se za uporabo hierarhičnega združevalnega algoritma za razvrščanje v skupine (angl. hierarhical agglomerative clustering). Pri tovrstnem razvrščanju v skupine je običajno potrebno izgraditi matriko razdalj, kjer je število v i-ti vrstici in j-tem stolpcu razdalja med i-tim in j-tim elementom. Med potekom združevanja se vrstice in stolpci združujejo, tako kot se združujejo skupine se spreminjajo razdalje med elementi. To je najobičajnejša implementacija te vrste združevanja, njena prednost pa je, da ima predpripravljene razdalje med elementi [13, 16].

3.2 Razdalja med skupinami

Pri razvrščanju v skupine je pomembno, kako določimo razdaljo med dvema skupinama. Vzemimo na primer skupini $\mathcal{A} = \{\mathrm{a}, \mathrm{b}\}$ in $\mathcal{B} = \{\mathrm{e}, \mathrm{f}, \mathrm{g}\}$. Razdalja med skupinama je najpogosteje definirana kot [13]:

- Največja razdalja med elementi obeh skupin (angl. complete linkage clustering):
$$\max\{\mathrm{d}(\mathrm{x}, \mathrm{y}) : \mathrm{x} \in \mathcal{A}, \mathrm{y} \in \mathcal{B}\} \tag{3.1}$$
- Najmanjša razdalja med elementi obeh skupin (anlg. single linkage clustering):
$$\min\{\mathrm{d}(\mathrm{x}, \mathrm{y}) : \mathrm{x} \in \mathcal{A}, \mathrm{y} \in \mathcal{B}\} \tag{3.2}$$
- Povprečje razdalj med elementi obeh skupin (angl. average linkage clustering):
$$\frac{1}{|\mathcal{A}| \cdot |\mathcal{B}|} \sum_{x \in \mathcal{A}, y \in \mathcal{B}} \mathrm{d}(\mathrm{x}, \mathrm{y}) \tag{3.3}$$
- Povečanje variance (Wardova metoda):
$$D(\mathcal{A} \cup \mathcal{B}) - (D(\mathcal{A}) + D(\mathcal{B})), \tag{3.4}$$
kjer $\mathrm{D}(\mathcal{X})$ označuje varianco, definirano kot:
$$D(\mathcal{X}) = \sum_{x \in \mathcal{X}} (c_{\mathcal{X}} - x)^2 \tag{3.5}$$
če je $c_{\mathcal{X}}$ centroid skupine $\mathcal{X}$.

Obstajajo tudi drugi načini, vendar so zgoraj našteti najpogostejši. Preizkusil sem vse štiri, najbolje pa se je izkazala največja razdalja med elementi skupin (enačba 3.1), ki je tudi uporabljena pri nadaljnih poskusih.

3.3 Razdalja med časovnimi vrstami

Poleg algoritma za razvrščanje v skupine in načina za določanje razdalje med skupinama, je potrebno določiti še merilo, ki bo uporabljeno za določanje razdalj med elementi. V tem primeru so elementi, ki se bodo združevali v skupine, kratke časovne vrste. To so zaporedja meritev iste spremenljivke v

času. Kratke časovne vrste vsebujejo malo meritev (5-15), zato algoritmi in merila uporabljani pri običajnih časovnih vrstah odpovejo. Običajno se vektor števil (n - terica) med seboj primerja z evklidsko ali manhattansko razdaljo. Ti dve merili nista primerni za časovne vrste, saj je pri časovnih vrstah bolj zanimivo obnašanje vrste skozi čas (oblika krivulje). Merila najdena v literaturi so:

- Razdalja, ki temelji na razlikah koeficientov [12]. Če so časi meritev T=$[t_0, t_1, ...t_n]$ in sta časovni vrsti X=$[x_0, x_1, ..., x_n]$ in Y=$[y_0, y_1, ..., y_n]$, je razdalja med njima definirana kot:

$$d^2(X,Y) = \sum_{k=0}^{n-1} \left(\frac{y_{k+1} - y_k}{t_{k+1} - t_k} - \frac{x_{k+1} - x_k}{t_{k+1} - t_k} \right)^2 \tag{3.6}$$

- Razdalja, ki temelji na korelacijskem koeficientu [15]. Korelacijski koeficient med časovnima vrstama X, Y je definiran kot:

$$r(X,Y) = \frac{E[(X - E[X]) \cdot (Y - E[Y])]}{E[(X - E[X])^2] \cdot E[(Y - E[Y])^2]}, \tag{3.7}$$

  kjer oznaka E[X] označuje matematično upanje. Merilo za razdaljo med časovnima vrstama X in Y je potem:

$$d(X,Y) = \sqrt{0.5 \cdot (1 - r(X,Y))} \tag{3.8}$$

- Kvalitativno merilo na temelju vseh razlik koeficientov [15]. Naj bosta X=$[x_0, x_1, x_2, ..., x_n]$ in Y=$[y_0, y_1, y_2, ..., y_n]$ časovni vrsti. Za definicijo tega merila moramo najprej definirati funkcijo $diff((x_i, x_j), (y_i, y_j))$:

$diff$	$x_i > x_j$	$x_i = x_j$	$x_i < x_j$
$y_i > y_j$	0	0.5	1
$y_i = y_j$	0.5	0	0.5
$y_i < y_j$	1	0.5	0

Tabela 3.1: Funkcija $diff((x_i, x_j), (y_i, y_j))$.

  Razdalja med časovnima vrstama X in Y je definirana kot:

$$d(X,Y) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2 \cdot diff((x_i, x_j), (y_i, y_j))}{n \cdot (n-1)} \tag{3.9}$$

Faktor $\frac{2}{n \cdot (n-1)}$ je uporabljen za normalizacijo vrednosti na interval [0,1]. Razdaljo bi lahko opisali kot ocenjevanje razlik med vsemi pari točk obeh časovnih vrst. Ocenjujemo naraščanje, padanje in stagnacijo.
Pri tem merilu je potrebno omeniti, da se v praksi redko pojavi primer $x_i = x_j$, $y_i = y_j$, kadar imata X in Y veliko ali celo zvezno zalogo vrednosti. Funkcijo razdalje bi se v takem primeru dalo izboljšati, če bi namesto stroge relacije "je enako"(=) uporabili "je skoraj enako"($\approx$). Možna izboljšava bi bila tudi, da bi primera $x_i > x_j$, $y_i > y_j$ in $x_i < x_j$, $y_i < y_j$ podrobneje obdelali in dodali ocene glede na to, kako močno vrsti naraščata/padata na intervalu (i,j).

Preizkusil sem vsa tri zgoraj opisana merila in se na podlagi primerjave izpisov grafov najdenih skupin odločil za zadnje merilo (kvalitativno merilo, enačba 3.8).

3.4 Eksperimenti

Za prvi eksperiment sem s pomočjo SQL poizvedb pripravil časovne vrste, kjer je vsak podatek v vrsti predstavljal število nesreč v posameznem mesecu (mesečna vsota za obdobje 1995-2004). Pripravil sem časovno vrsto za vsako upravno enoto v Sloveniji (58 upravnih enot). Dobljene časovne vrste so seveda dolžine 12 (leto ima 12 mesecev). Primer mesečne časovne vrste prikazuje tabela 3.2.

mesec	jan	feb	mar	apr	maj	jun	jul	avg	sep	okt	nov	dec
št. nesreč	360	314	446	472	563	651	796	760	464	411	359	377

Tabela 3.2: Mesečna časovna vrsta za upravno enoto Piran.

Druga skupina časovnih vrst je vsebovala podatke o številu nesreč po letih (obdobje 1995-2004). Spet sem pripravil časovne vrste za vsako upravno enoto. Časovne vrste so bile tokrat dolžine 10 (obdobje 10 let). Primer letne časovne vrste prikazuje tabela 3.3.

leto	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004
št. nesreč	538	519	541	497	566	556	561	592	418	393

Tabela 3.3: Letna časovna vrsta za upravno enoto Piran.

Tretja skupina časovnih vrst so bile mesečne časovne vrste, ki pa so vsebovale le podatke o številu nesreč, v katerih so bili udeleženi tujci.

Nad mesečnimi časovnimi vrstami sem uporabil hierarhično razvrščanje v skupine, s kvalitativnim merilom razdalje med vrstami. Program sem implementiral v programskem jeziku R. R že vsebuje implementacijo hierarhičnega razvrščanja v skupine (paket *stats*). Potrebno je bilo razpoložljivo implementacijo uporabiti in dodati kodo za pripravo matrike razdalj, torej tudi funkcijo za izračun razdalje med dvema vrstama. Seveda je potrebno na začetku branje podatkov in na koncu izris najdenih skupin časovnih vrst (centroidov skupin - povprečnih časovih vrst za vsako skupino).

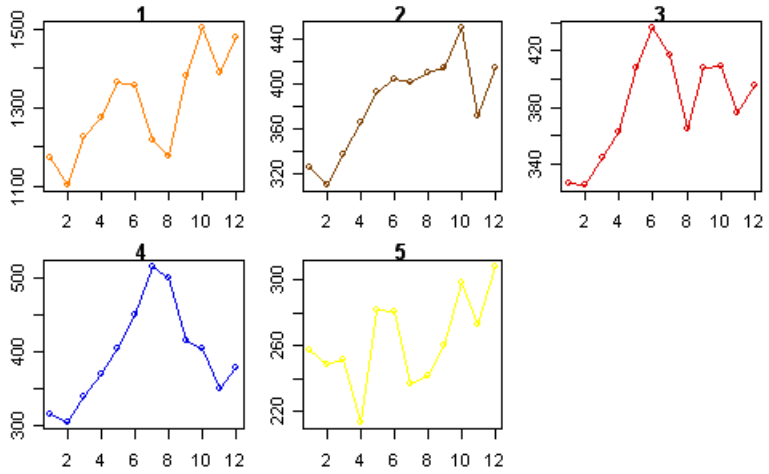
Isti program sem preizkusil tudi na letnih časovnih vrstah, vendar so letne časovne vrste med seboj zelo različne, v vrstah pa so zelo pogosta večja nihanja, zato so najdene skupine zelo nehomogene. Odločil sem se, da upravne enote raje razdelim na 3 skupine:

- Upravne enote, kjer je opazen trend upadanja števila nesreč.
- Upravne enote, kjer je opazen trend naraščanja števila nesreč
- Upravne enote, kjer je opazna stagnacija števila nesreč

Nad mesečnimi časovnimi vrstami za tujce sem uporabil isti program za hierarhični iskanje skupin kot za običajne mesečne časovne vrste.

3.5 Rezultati

Program za iskanje skupin časovnih vrst je mesečne časovne vrste združil v 5 različnih skupin. Centroide najdenih skupin predstavlja slika 3.1

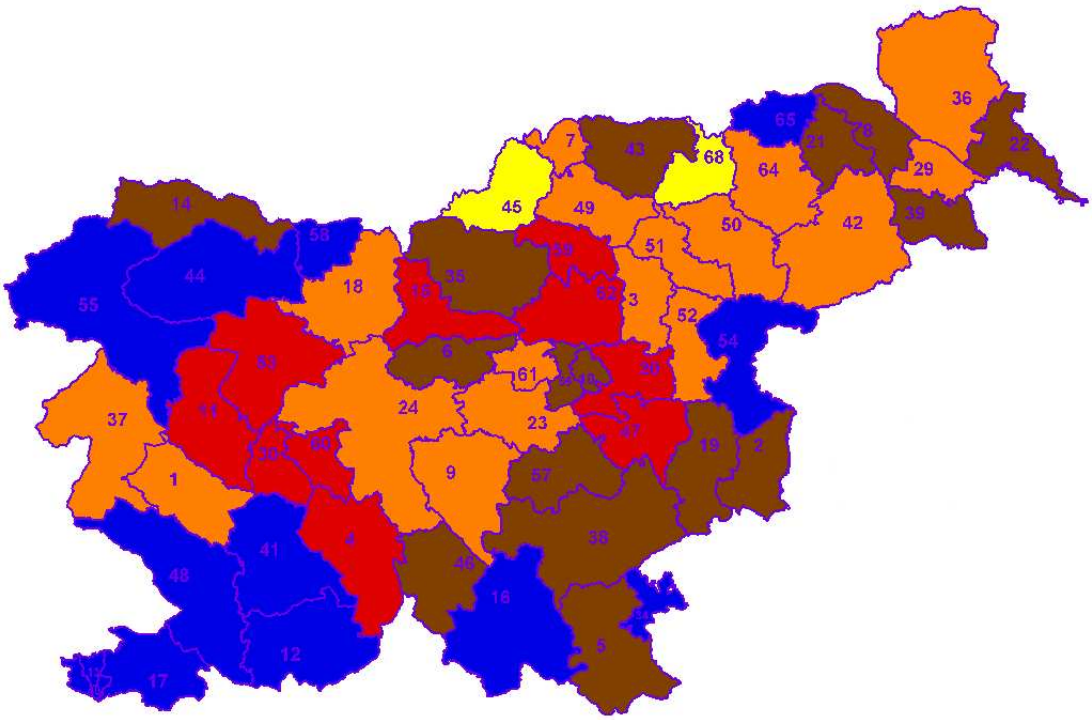


Slika 3.1: Centroidi skupin upravnih enot, za mesečne časovne vrste

Opis najdenih skupin:

- Skupina 1: Dva izrazita vrhunca, prvi je nižji in se pojavi ob koncu pomladi, v začetku poletja (maj, junij), drugi pa je višji in se pojavi pozno jeseni, in začetek zime (oktober, november, december).
- Skupina 2: Naraščanje števila nesreč skozi leto, z vrhuncem ob koncu leta (oktober, november, december).
- Skupina 3: Dva izrazita vrhunca, prvi je višji in se pojavi ob koncu pomladi, in začetku poletja (maj, junij), drugi je nižji, pojavi pa se jeseni (september, oktober, november).
- Skupina 4: En sam, zelo izrazit vrhunec, ki se pojavi poleti (julij, avgust).
- Skupina 5: Dva vrhunca, prvi je nižji in se pojavi konec pomladi, v začetku poletja (maj, junij), drugi je višji in se pojavi jeseni (oktober). Gre za skupino podobno skupini 1, vendar z ostrejšimi vrhunci in drugače razporejenimi minimumi.

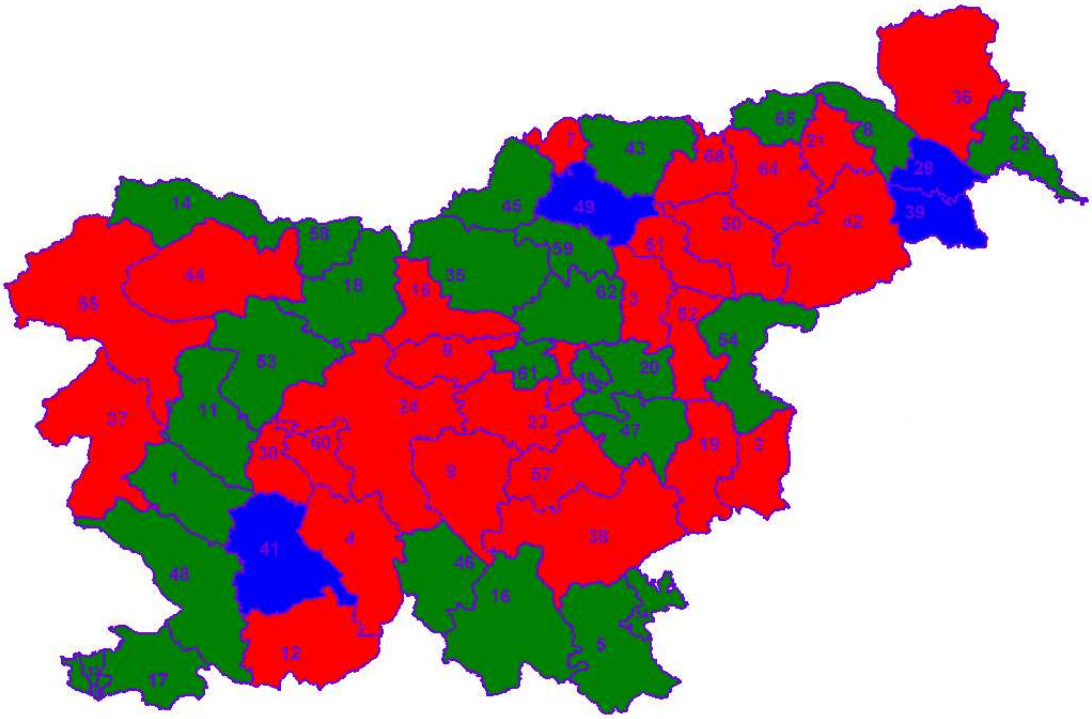
Vsaka skupina je označna s svojo barvo. Te barve so uporabljene za barvanje upravnih enot na sliki 3.2. Z vizualizacijo na zemljevidu je precej lažje poiskati razloge za pripadnost upravne enote posamezni skupini. Zanimivo pa je tudi, da so enako obarvane upravne enote povezane, ali pa zelo blizu.



Slika 3.2: Razvrstitev upravnih enot v skupine glede na trend mesečnih časovnih vrst.

Naslednja vizualizacija (slika 3.3) prikazuje delitev upravnih enot v tri skupine glede na trend letnih časovnih vrst. Barve prestavljajo:

- zelena - Trend upadanja števila nesreč.
- rdeča - Trend naraščanja števila prometnih nesreč.
- modra - Trend stagnacije števila nesreč.



Slika 3.3: Razvrstitev upravnih enot v skupine glede na trend letnih časovnih vrst.

Tudi pri tej vizualizaciji je zanimiva povezanost upravnih enot z enakim trendom.

Če združimo rezulate iskanja skupin pri mesečnih in letnih časovnih vrstah, pridemo do tabel 3.4, 3.5, 3.6, 3.7, 3.8. Vsaka tabela predstavlja upravne enote z enakim mesečnim režimom, tabele pa so razdeljene na dva stolpca. Levi (rdeča barva) predstavlja upravne enote z trendom naraščanja števila prometnih nesreč, desni (zelena barva) pa upravne enote z trendom upadanja števila prometnih nesreč. S pomočjo teh tabel lahko upravnim enotam določimo upravne enote, po katerih bi se morale zgledovati, oziroma s katerimi naj se primerjajo pri ukrepih za izboljšanje prometne varnosti.

Prometni strokovnjak je rezultate komentiral kot zanimive, predlagal pa je še analizo mesečnih časovnih vrst po posameznih letih (kako se spreminja "prometni profil"upravnih enot po letih). Predlagal je tudi, da se trend upadanja/naraščanja števila prometnih nesreč opazuje v krajšem obdobju (npr. zadnjih 5 let), saj se prometna infrastriktura v Sloveniji hitro spreminja.

Skupina 1	
03 - Celje	01 - Ajdovščina
07 - Dravograd	18 - Kranj
09 - Grosuple	29 - Ljutomer
23 - Litija	49 - Slovenj Gradec
24 - Ljubljana	61 - Zagorje ob Savi
36 - Murska Sobota	
37 - Nova Gorica	
42 - Ptuj	
50 - Slovenska Bistrica	
51 - Slovenske Konjice	
52 - Šentjur pri Celju	
64 - Maribor	

Tabela 3.4: Upravne enote v skupini 1.

Skupina 2	
02 - Brežice	05 - Črnomelj
06 - Domžale	08 - Gornja Radgona
19 - Krško	10 - Hrastnik
21 - Lenart	14 - Jesenice
38 - Novo Mesto	22 - Lendava
56 - Trbovlje	35 - Mozirje
57 - Trebnje	39 - Ormož
	43 - Radlje ob Dravi
	46 - Ribnica

Tabela 3.5: Upravne enote v skupini 2.

Skupina 3	
04 - Cerknica	11 - Idrija
15 - Kamnik	20 - Laško
30 - Logatec	47 - Sevnica
60 - Vrhnika	53 - Škofja Loka
	59 - Velenje
	62 - Žalec

Tabela 3.6: Upravne enote v skupini 3.

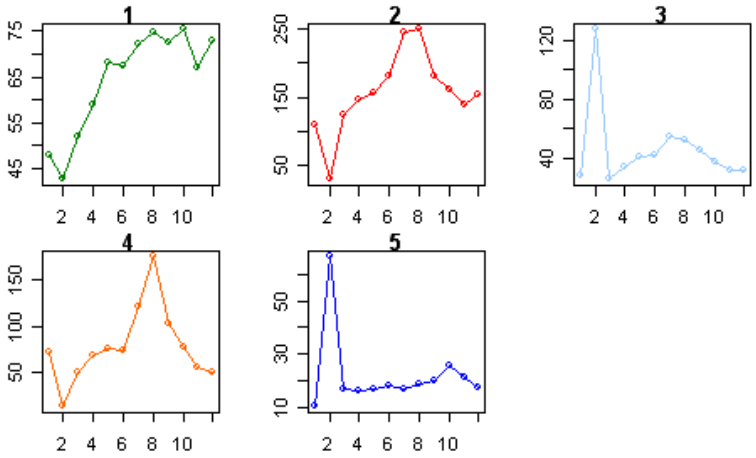
Skupina 4	
12 - Ilirska Bistrica	13 - Izola
44 - Radovljica	16 - Kočevje
55 - Tolmin	17 - Koper
	34 - Metlika
	40 - Piran
	41 - Postojna
	48 - Sežana
	54 - Šmarje pri Jelšah
	58 - Tržič
	65 - Pesnica

Tabela 3.7: Upravne enote v skupini 4.

Skupina 5	
68 - Ruše	45 - Ravne na Koroškem

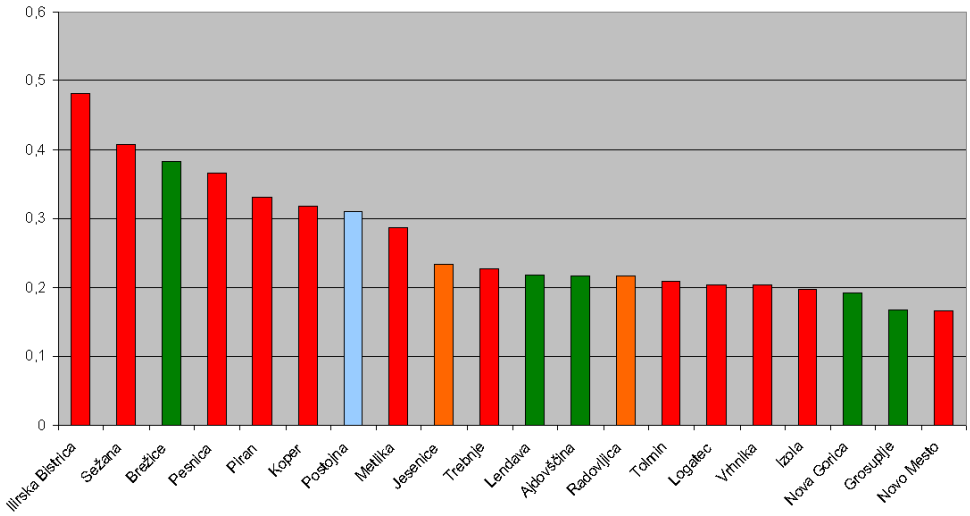
Tabela 3.8: Upravne enote v skupini 5.

Mesečne časovne vrste za tujce je program za razvrščanje v skupine razporedil v pet skupin. Centroide najdenih skupin prikazuje slika 3.4.



Slika 3.4: Centroidi skupin upravnih enot za tujce.

V barvah centroidov so obarvani tudi stolpci na grafu (slika 3.5), ki prikazuje deleže nesreč, v katerih so bili udeleženi tujci. Prikazuje le podatke za 20 upravnih enot, ki imajo največji delež nesreč z vpletenimi tujci.



Slika 3.5: Upravne enote z največjim deležem nesreč z vpletenimi tujci.

Poglavje 4

Razvrščanje v skupine in vizualizacija z Google Earth

Hierarhično razvrščanje točk (nesreč) v skupine omogoča iskanje krajev, kjer ne zgodi več prometnih nesreč. Takim krajem bi morali policija in odgovorni za cestno infrastrukturo posvečati posebno pozornost. Vizualizacija s programom Google Earth je v nekaterih primerih lahko v pomoč tudi pri odkrivanju vzroka za skupino nesreč. Na satelitskih slikah, uporabljenih v Google Earth, je namreč dobro vidna pokrajina in prometna infrastruktura.

4.1 KML

KML (angl. keyhole markup language) je jezik, ki temelji na XML jeziku. Namenjen je za upravljanje in prikaz tridimenzionalnih geografskih podatkov v programih Google Earth, Google Maps, Google Mobile in WorldWind [17].

Datoteka KML določa množico elementov (točke, slike, poligoni, 3D modeli, tekstovni opisi,...) za prikazovanje v naštetih aplikacijah. Vsak element ima pripadajočo zemljepisno širino in dolžino. Pogled lahko dodatno obogatimo z ostalimi podatki kot so: nagib(tilt), smer (heading), višina (altitude) [17].

KML datoteke so pogosto na voljo kot KMZ datoteke, ki so KML datoteke stisnjene z ZIP algoritmom. Ko KMZ datoteko razpakiramo, dobimo KML datoteko s pripadajočimi slikami in ikonami [17].

Primer KML datoteke [17]:

```
⟨?xml version="1.0"encoding="UTF-8"?⟩
⟨kml xmlns="http://earth.google.com/kml/2.0"⟩
    ⟨Placemark⟩
        ⟨description⟩New York City⟨/description⟩
        ⟨name⟩New York City⟨/name⟩
        ⟨Point⟩
            ⟨coordinates⟩ -74.006393, 40.714172,0⟨/coordinates⟩
        ⟨/Point⟩
    ⟨/Placemark⟩
⟨/kml⟩
```

4.1.1 KML stili

Običajno je začetek KML dokumenta namenjen definiciji potrebnih stilov, ki jih bodo kasnejši elementi uporabljali. Stili določajo grafično podobo elementov, ki se bodo izrisali v uporabljenem programu. Stil definiramo z ukazom:

```
⟨Style id="krog_stil"⟩
    ...
⟨/Style⟩
```

Kasneje v dokumentu stil uporabimo na naslednji način:

```
⟨Placemark⟩
    ⟨styleUrl⟩#krog_stil⟨/styleUrl⟩
    ...
⟨/Placemark⟩
```

Možno je tudi definirati dva stila za en element. Prvi stil je običajen, drugi pa se uporabi, kadar je element označen z miško (angl. highlight). V takem primeru se uporabi element *style map* in definirata se dva stila. Koda elementa *style map*:

```
⟨StyleMap id="msn_icon59_copy1"⟩
    ⟨Pair⟩
        ⟨key⟩normal⟨/key⟩
        ⟨styleUrl⟩#sn_icon59_copy1⟨/styleUrl⟩
    ⟨/Pair⟩
    ⟨Pair⟩
        ⟨key⟩highlight⟨/key⟩
        ⟨styleUrl⟩#sh_icon51_copy1⟨/styleUrl⟩
    ⟨/Pair⟩
⟨/StyleMap⟩
```

4.1.2 KML elementi

Jezik KML omogoča uporabo veliko različnih elementov, kot na primer: točk, poti, poligonov, slik, 3d objektov, html opisov... Poleg tega se KML tudi zelo aktivno razvija in neprestano dobiva nove zmožnosti.

Za predstavitev najdenih skupin prometnih nesreč so bili uporabljeni naslednji KML elementi:

- *Point* (točka) - Uporabljena je za prikaz centroidov skupin prometnih nesreč.
- *LineString* (pot) - Uporabljena je za izris kroga, ki predstavlja radij skupine prometnih nesreč.
- *LinearRing* (obroč iz daljic) - Uporabljen je za izris meja upravnih enot.

4.2 Opis programa za razvrščanje v skupine

Za razvrščanje v skupine (angl. clustering) je uporabljeno hierarhično združevalno razvrščanje v skupine, tako kot pri razvrščanju časovnih vrst v skupine. Vendar pa je program implementiran v programskem jeziku Java, zato je bilo potrebno poskrbeti tudi za implementacijo algoritma za razvrščanje v skupine. Program je sestavljen iz treh delov:

- Inicializacija - Iz podatkovne baze se preberejo podatki o nesrečah za izbrano leto in upravno enoto. Na podlagi teh podatkov se zgradi seznam skupin prometnih nesreč, kjer na začetku vsaka skupina vsebuje le eno nesrečeo. Poleg tega se inicializira matrika razdalj med skupinami.

Uporabljena je evklidska razdalja. Ker matrika razdalj zasede precej pomnilnika je možno, da program prekorači dovoljeno mejo in se posledično preneha izvajati. Zato ima program vhodni parameter, ki določa koliko nesreč naj se največ obravnava. Če je v danem letu in upravni enoti več nesreč od tega števila, se izbere naključni vzorec v velikosti tega števila. V praksi sem uporabljal omejitev 1500 nesreč.

- Združevanje - V zanki se združujejo skupine prometnih nesreč. V vsakem koraku se združita najbližji skupini. Za razdaljo med skupinama je tukaj uporabljena razdalja med centroidoma skupin. Združevanje poteka tako dolgo, dokler ne pride do združevanja skupin na razdalji, ki je večja od programu podane meje. V praksi je bila ta omejitev 100 metrov. Pri vsakem združevanju skupin je potrebno osvežiti seznam skupin in matriko razdalj med skupinami.

- Izpis - Na koncu programa naredi izpis najdenih skupin v KML datoteko. Najprej je potrebna ustrezna glava datoteke z definicijo potrebnih stilov. Sledi še izpis centroidov skupin, mej skupin in mej upravne enote. Kot vhodni parameter programu podamo tudi število nesreč, ki se obravnava kor skupina in se izriše. Običajno je bila ta omejitev 3 (tri nesreče ali več so obravnavane kot skupina).
  V izpisu rumen trikotnik z klicajem označuje skupino nesreč, rdeč trikotnik z klicajem označuje posebno veliko skupino nesreč (7 ali več nesreč), rdeči križ pa označuje skupine nesreč, kjer je v nesrečah prišlo do hujših poškodb ali celo smrti.

4.3 Eksperimenti in rezultati

Prednost uporabe Google Earth za prikazovanje geografskih podatkov je, da je za izris podatkov že poskrbljeno, uporabnik mora zagotoviti le podatke, ki bi jih rad izrisal. Izris na satelitskih slikah je zelo informativen, saj je na satelitskih slikah dobro vidna prometna infrastruktura in ostali objekti ob cesti, ki lahko vplivajo na število nesreč (zožanje ceste, nepregleden ali oster ovinek, neustrezno križišče, parkirišče,...). Google Earth pa ponuja tudi veliko različnih pogledov na neko območje (obračanje slike, povečevanje, 3D pogled s strani).

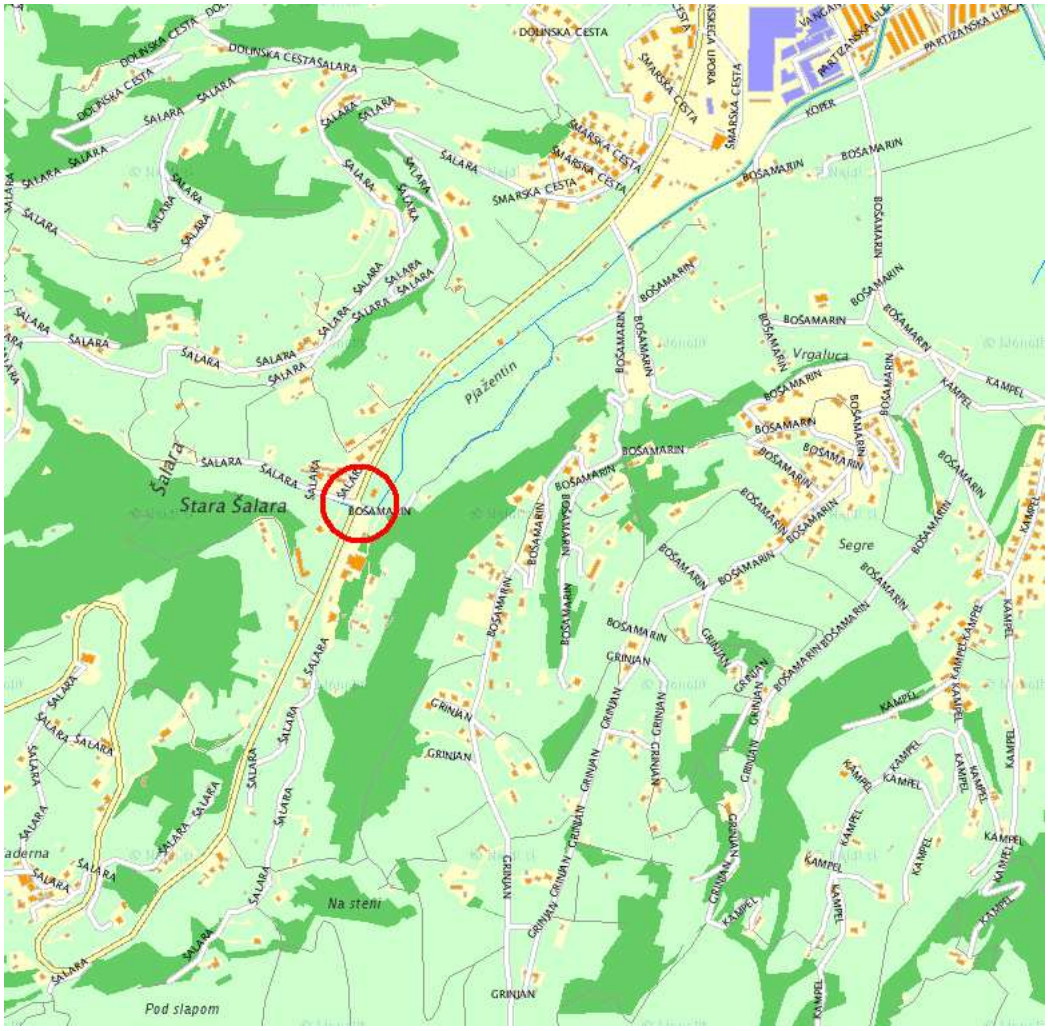
Slabost Google Erarth je, da je le manjši del slovenskega ozemlja pokrit z dovolj podrobnimi satelitskimi slikami. Od 58 upravnih enot je zadovoljivo podrobno pokritih le 8 upravnih enot (Cerknica, Ljubljana, Murska Sobota,

Izola, Piran, Koper, Sežana, Nova Gorica). Od teh so zelo zanimive za iskanje skupin obmorske upravne enote (Koper, Piran, Izola), ki imajo tudi zelo visoko razmerje nesreč na površino. Google Earth za uporabo zahteva povezavo z internetom, saj satelitskih slik ne hrani lokalno ampak jih prenaša preko interneta. To je za marsikoga lahko ovira pri uporabi.

Slika 4.1: Zgostitve prometnih nesreč v Kopru leta 2006.

Kot primer uporabe sem pregledal podatke o najdenih skupinah za upravno enoto Koper. Primer izrisa najdenih skupin prikazuje slika 4.1. Najdenih je bilo precej skupin, katerih lokacija se skozi leta ne spreminja bistveno, kar opozarja na lokacijo, kjer je verjetnost nesreč bistveno povečana. V takem primeru bi bilo potrebno obiskati lokacijo in se podrobneje pozanimati o vzroku zgostitve prometnih nesreč. Glede na odkrit vzrok pa seveda tudi ustrezno ukrepati. Zanimivo je tudi opažanje, ko nekatere zgostitve naenkrat izginejo,

oziroma se pojavijo, kar kaže na spremembe v prometni infrastrukturi in s tem v prometnem režimu in pa tudi pravilno ukrepanje policije. Kot primer podajam opaženo zgostitev južno od Koprskega mestnega jedra, v okolici križišča pri Stari Šalari. Točnejšo lokacijo prikazuje zemljevid na sliki 4.2 (zemljevid je pridobljen s spletne strani *http://zemljevid.najdi.si/*).



Slika 4.2: Lokacija najdene zgostitve prometnih nesreč.

Opažena zgostitev prometnih nesreč pa ni stalna, v zadnjih letih je celo skoraj izginila. Tabela 4.1 prikazuje gibanje števila nesreč v okolici križišča po letih za katera obstajajo podatki (podatek v oklepaju pomeni število nesreč s

hujšimi telesnimi poškodbami ali smrtjo). Očiten je pozitiven trend, saj se je število nesreč močno zmanjšalo, hujših nesreč pa ni več.

leto	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005
nesrece	6	6(1)	4	5	2	4(1)	1	2	0	0	0

Tabela 4.1: Število nesreč v okolici križišča pri Šalari.

Prometni strokovnjak je vizualizacije pozdravil z zanimanjem, hkrati pa omenil da se število nesreč na mestu kjer je policija ukrepala res zmanjša, vendar se zato pogosto nesreče začnejo pojavljati drugje (število nesreč ostane isto, spremeni se le njihova lokacija).

Poglavje 5

Povezovalna pravila

Povezovalna pravila so zelo pogosto uporabljana tehnika rudarjenja podatkov, saj omogočajo odkrivanje zanimivih vzorcev, skritih v množici podatkov. Uporabljenih je bilo več programov, ki omogočajo iskanje povezovalnih pravil. Predstavljeni so tudi primeri najdenih rezultatov.

5.1 Prostorska povezovalna pravila in klasična povezovalna pravila

Podatki se danes najpogosteje hranijo v relacijskih podatkovnih bazah. To pomeni, da so strukturirani v tabele (relacije), te pa so sestavljene iz posameznih zapisov (transakcij). Eden od pogosto uporabljanih načinov za odkrivanje znanja v tabelaričnih podatkih so povezovalna pravila [8, 13] (angl. association rules). Običajno se iščejo pravila le znotraj ene tabele, saj je iskanje pravil, ki zajemajo atribute več tabel veliko kompleksnejša naloga. Posebej znane so praktične uporabe povezovalnih pravil na področju oglaševanja [8, 13].

Klasično asociacijsko pravilo lahko opišemo kot pravilo oblike: $\mathcal{X} \Rightarrow \mathcal{Y}$, kjer sta $\mathcal{X}$ in $\mathcal{Y}$ množici predikatov (oziroma vrednosti atributov). Primer (izmišljenega) povezovalnega pravila iz prometne domene:
(dan_v_tednu=petek,ura=23,vrsta_udeleženca=pešec)⇒(tip_nesreče=smrt).

Levemu delu povezovalnega pravila ($\mathcal{X}$) pravimo tudi telo (angl. body, antecedent), desnemu delu ($\mathcal{Y}$) pa glava (angl. head, consequent). Za pravila se običajno izračunajo tudi cenilke, ki nam povedo, kako dobro je neko pravilo. Najpogostejše cenilke povezovalnih pravil so:

- podpora (angl. support) - verjetnost, da za zapis (t) v tabeli veljajo vrednosti atributov v $\mathcal{X}$. sup=p(X(t))

- zaupanje (angl. confidence) - verjetnost, da ima zapis (t) z vrednostmi stributov iz $\mathcal{X}$ tudi vrednosti atributov iz $\mathcal{Y}$. con=p(Y(t)|X(t))
- dvig (angl. lift) - faktor, ki pove, kolikokrat je desna stran (glava) pogostejša kot v celotni tabeli, če je zadoščeno levi strani (telo). Za ilustracijo opišimo primer dviga: imejmo 1000 zapisov, telo pokriva 200 zapisov, glava 100 zapisov, glava pa pokriva 50 od zapisov, ki jih pokriva telo. Običajna verjetnost vrednosti atributov v glavi je 100/1000=0.1, če pa je zadoščeno telesu pravila je verjetnost glave 50/200=0.25, torej je dvig 0.25/0.1=2.5.

Prostorska povezovalna pravila [8] se od običajnih pravil razlikujejo v temu, da je vsaj eden od atributov v glavi ali telesu pravila prostorski (se nanaša na lokacijo zapisa). Poleg povezovalnih pravil, ki se nanašajo na eno tabelo, pa nekateri algoritmi omogočajo tudi iskanje pravil, v katerih nastopajo atributi večih tabel [3, 4, 7] (med seboj povezanih preko ključev). Poleg tega pa lahko na podlagi geografskih atributov dodajamo predikate, ki opisujejo prostorska razmerja med zapisi (blizu, prečka, je vzporeden,...). To je za prostorska povezovalna pravila še posebno zanimivo, saj nas zanimajo prav prostorske povezave med objekti v prostoru in njihov medsebojni vpliv.

5.2 Uporaba algoritma SPADA

5.2.1 Predstavitev programa

Algoritem SPADA [1] (Spatial Pattern Discovery Algorithm) je le del projekta SPIN. V okviru tega projekta so bila razvita tudi grafična orodja, ki omogočajo iz podatkovne baze izluščiti željene podatke in jih shraniti v tekstovno obliko, primerno za uporabo z algoritmom SPADA. Uporabljena je bila podatkovna baza podjetja Oracle, zaradi dobre podpore shranjevanju prostorskih podakov (Oracle Spatial[1]). Sam algoritem SPADA pa je implementiran v programskem jeziku Prolog. Uporabi Prologa je prolagojen tudi format podatkov, ki jih SPADA sprejme (predikatni zapis podatkov). Algoritem SPADA išče povezovalna pravila v večih tabelah, poleg tega pa upošteva že dodatne predikate, ki opisujejo povezave med zapisi in so običajno prostorski predikati.

V uporabo sem dobil samo algoritem SPADA[2] brez celotnega okolja SPIN

[1]Domača stran Oracle Spatial: *http://www.oracle.com/technology/products/spatial/index.html*

[2]Domača stran algoritma SPADA: *http://www.di.uniba.it/ malerba/software/ARES/index.htm*

s pripadajočimi aplikacijami, zato sem moral sam poskrbeti za pretvorbo podatkov v ustrezen format in generiranje prostorskih predikatov. V ta namen sem spisal program v programskem jeziku Java, ki glede na vhodne parametre (upravna enota, leto) pripravi potrebne datoteke.

5.2.2 Priprava podatkov

Datoteke, ki so potrebne za algoritem SPADA:

- DB (angl. data base) - Datoteka s končnico DB vsebuje podatke o objektih. Zapis o prometni nesreči in vanjo udeleženi osebi prenešen v predikatno obliko:

  % Podatki o nesreči s ključem 394016
  accident_id(394016).
  class(394016,'B').
  day_of_week(394016,'2').
  day_of_month(394016,'[1..5]').
  month(394016,'1').
  hour(394016,'13').
  road_category(394016,'N').
  scene_description(394016,'N').
  cause_of_accident(394016,'VR').
  accident_type(394016,'TV').
  accident_weather(394016,'J').
  traffic_condition(394016,'G').
  road_condition(394016,'SU').

  % Podatki o osebi udeleženi v nesreči s ključem 394016
  people_in_accident(394016,p1).
  caused_the_accident(p1,'D').
  age(p1,'E').
  sex(p1,'2').
  injury(p1,'B').
  person_type(p1,'OA').
  safety_belt_or_helmet(p1,'1').
  years_license(p1,'D').
  alcohol(p1,'N').

Poleg podatkov o objektih se v datoteki DB nahajajo tudi predikati, ki opredeljujejo povezave med objekti. V primeru nesreč sem definiral predikat closeto(A,B), ki opisuje dejstvo, da sta nesreči A in B blizu (manj kot 100 metrov evklidske razdalje).

- LB (angl. language bias) - Vsebina datoteke LB:

  1. Definicija glavnih objektov in ostalih objektov, ter njihovo članstvo v hierarhijah (več o hierarhijah pri opisu datoteke BK). Objekte ločimo na dve vrsti: RO (angl. reference objects) - so glavni objekti zanimanja, TRO (angl. task relavant objects) pa so objekti, ki vplivajo na glavne objekte. V primeru prometnih nesreč so glavni objekti nesreče, ostali objekti pa so osebe. V datoteki LB je to definirano kot:
     key(accident_id(new ro)).
     trh(people).
     trh(accident).

  2. Definicija zalog vrednosti za vse predikate.
  3. Parametri, ki določajo delovanje algoritma. Pomembnejši parametri:
     % Najmanjša podpora (support)
     min_sup(1,0.5).
     % Najmanjše zaupanje (confidence)
     min_conf(1,0.8).
     % Najmanjša dolžina povezovalnega pravila (število predikatov)
     param(minimum_pattern_length, 5).
     % Ali naj se najdeni vzorci pretvorijo v pravila
     param(patterns_to_rules , yes).
     % Največja dolžina povezovalnih pravil in s tem št. korakov algoritma
     param(max_ref_steps, 9).
     ...

- BK (angl. background knowledge) - Datoteka BK opisuje hierarhije objektov. Na primer, če bi v analizo vključil podatke o cestah, bi lahko ceste razvrstili v hierahijo, ali pa bi celo ustvaril hierarhijo transportnih poti, ki bi vključevala poleg cest tudi železnice. V primeru obravnavanih podatkov so objekti le nesreče in osebe. Obe vrsti objektov pa sta del hierarhije višine ena (koren hierarhije in nivo z dejanskimi objekti). Koda datoteke BK:

% Hierarchy of people
hierarchy(people, 1, null, [people]).
hierarchy(people, 2, people, [p1,p2,p3,p4,p5,p6,p7,...])

% Hierarchy of accidents
hierarchy(accident, 1, null, [accident]).
hierarchy(accident, 2, accident, [394016,394234,394235,394236,...])

5.2.3 Eksperimenti in rezultati

Ko so podatki pripravljeni lahko algoritem poženemo v obliki enostavnega konzolnega programa.

Pri uporabi na pripravljenih podatkih sta se takoj pokazali dve veliki slabosti algoritma. Prva slabost je velika časovna zahtevnost, ki je posledica implementacije v Prologu in pa same zahtevnosti naloge. Čas potreben za obdelavo podatkov se tipično meri v urah in nič nenavadnega ni, če pri večjih upravnih enotah obdelava enoletnih podatkov traja več kot 24 ur. Posledica velike časovne zahtevnosti algoritma je, da je opravljenih manj eksperimentov, s tem pa je daljša pot do uporabnih rezultatov (potrebnih je več eksperimentov na istih podatkih, da se poiščejo prave vrednosti parametrov). Druga težava pa je pregledovanje rezultatov. Običajno vsebujejo datoteke z rezultati algoritma veliko število povezovalnih pravil, ki jih je potrebno ročno pregledati in med najdeno množico poiskati res zanimiva pravila.

Da bi bilo med najdenimi pravili več zanimivih pravil, lahko podrobneje definiramo, kakšna pravila želimo. Običajno nas pri prometnih nesrečah zanima smrtnost in parametri, ki vplivajo na smrtnost in poškodbe udeležencev. To zapišemo kot omejitve v datoteki LB. Omejitve, ki sem jih dodal so:

% Glava pravila mora vsebovati enega od predikatov class() ali injury()
head_constraint([class(_), injury(_,_)], 1, 1)
% Dolžina glave pravila je ena ali dva predikata
rule_head_length(1, 2)
% V telesu pravila se mora točno enkrat pojaviti predikat closeto()
body_constraint([closeto(_,_)], 1, 1)

Uporaba omenjenih omejitev zagotavlja, da bo med najdenimi pravili več zanimivih pravil, in tudi najdeno bo manjše število pravil. Da bodo sploh najdena kakršnakoli povezovalna pravila, ki ustrezajo tako strogim omejitvem

je potrebno spustiti vrednost parametra *support* (podpora). Žal pa sem tukaj naletel na nov problem. S tem ko se pomanjša parameter podpore, se preišče veliko več vzorcev (angl. patterns), torej je večji iskalni prostor. Omejitve pa števila generiranih in ocenjehih vzorcev ne zmanjšajo. Če nek vzorec P omejitvi ne ustreza, ji bo morda ustrezal bolj podroben vzorec z dodanimi predikati. Namen omejitev je torej samo olajšati preiskovanje rezultatov. Ker se število preiskanih vzorcev z zmanjšanim parametrom podpore zelo poveča z omejitvami pa nič ne zmanjša, pridemo do izjemne prostorske in časovne zahtevnosti. V praksi to pomeni, da zaradi pomnilniških omejitev program nikoli ne dokonča svojega dela, torej se zaustavi, ko mu zmanjka pomnilnika.

Še zadnja pomankljivost pa je, da algoritem za vsako pravilo izračuna le podporo in zaupanje, ne pa tudi dviga, ki sem ga želel uporabiti za oceno zanimivosti najdenih pravil. Edini pridobljeni rezultati z algoritmom SPADA so velike datoteke najdenih povezovalnih pravil (brez uporabe omejitev).

Med pregledovanjem teh datotek sem naletel na naslednja bolj ali manj zanimiva (prostorska) povezovalna pravila:

% upravna enota: Izola, leto: 2004
% 89% moških udeležencev v prometnih nesrečah je uporabljalo varnostni pas/čelado
accident_id(A), people_in_accident(A,B), is_a(B,people), sex(B,1) ⇒ safety_belt_or_helmet(B,1), (supp=78 conf=89)

% 82% moških, ki so udeleženi v nesrečah brez poškodb, je povzročiteljev
accident_id(A), people_in_accident(A,B), is_a(B,people), sex(B,1), class(A,B) ⇒ caused_the_accident(B,D), (supp=50 conf=82)

% V 81% nesreč, kjer sta vsaj dva udeleženca, sta dva udeležena uporabljala varnostni pas/čelado
accident_id(A), people_in_accident(A,B), is_a(B,people), people_in_accident(A,C), $C\bar{B}$, is_a(C,people), safety_belt_or_helmet(C,1) ⇒ safety_belt_or_helmet(B,1), (supp=60 conf=81)

% upravna enota: Vrhnika, leto: 2004
% Kadar sta vsaj dva udeležena uporabljala varnostni pas/čelado in je vsaj eden od udeležencev moškega spola, se nesreča v 92% konča brez poškodb
accident_id(A), people_in_accident(A,B), is_a(B,people), people_in_accident(A,C), $C\bar{B}$, is_a(C,people), sex(B,1), safety_belt_or_helmet(C,1), safety_belt_or_helmet(B,1) ⇒ injury(C,B), (supp=52 conf=92)

% V dveh nesrečah, ki sta blizu, sta oba udeleženca v 99,6% primerov tudi povzročitelja
accident_id(A), closeto(B,A), is_a(B,accident), people_in_accident(B,C), $C\bar{B}$, is_a(C,people), people_in_accident(A,D), $D\bar{B}$, $D\bar{C}$, is_a(D,people), caused_the_accident(D,D) ⇒ caused_the_accident(C,D), (supp=51 conf=99.6)

5.3 Uporaba drugih programov za iskanje povezovalnih pravil

Zaradi težav pri uporabi algoritma SPADA sem se odločil preizkusiti tudi bolj znane prosto dostopne programe, ki omogočajo iskanje klasičnih povezovalnih pravil. Preizkusil sem jih na tabeli nesreč. Zanimalo me je predvsem, ali omogočajo tudi izračun dviga kot merila zanimivosti povezovalnega pravila in pa, kakšne so možnosti pri definiranju omejitev. Želel bi si namreč dodati omejitv, naj telo povezovalnega pravila vsebuje podatke o cesti, naselju ali ulici.
Preizkušeni programi:

- Orange[3] - Gre za program, ki omogoča uporabo množice tehnik rudarjenja podatkov, med njimi tudi povezovalnih pravil. Orange razvijajo na Fakulteti za računalništvo in informatiko v Ljubljani v laboratoriju za umetno inteligenco. Omogoča tudi iskanje klasifikacijskih povezovalnih pravil. To so pravila, kjer vrednosti atributov v telesu napovedujejo vrednost atributa v glavi. V glavi je vedno le en sam vnaprej predpisan atribut. To lahko uporabimo kot omejitv za glavo pravil. Žal pa ni možno definirati omejitev glede atributov, ki naj se pojavijo v telesu povezovalnih pravil. Orange omogoča tudi izračun dviga, poleg tega pa še nekaj ocen zanimivosti pravil: pokritost (angl. coverage), moč (angl. stregth), vplivnost (angl. leverage).

- Weka[4] - Je najbolj znan program na področju rudarjenja podatkov, ker je bil eden prvih, in pa zaradi velikega števila algoritmov, metod, ki jih vključuje. Weka tako kot Orange ponuja klasifikacijska povezovalna pravila in nobenih omejitev glede telesa pravil. Ponuja tudi dvig kot merilo zanimivosti. Glede na funkcionalnosti sta si Weka in Orange skoraj povsem enaka, izkaže pa se, da je Orange sposoben dela s precej večjimi količinami podatkov.

[3]Domača stran programa Orange: *http://www.ailab.si/orange*
[4]Domača stran programa Weka: *http://www.cs.waikato.ac.nz/ml/weka/*

- R[5] - Programski jezik R omogoča uporabo povezovalnih pravil (paket *arules*) in sicer preko uporabe implementacije algoritma Apriori, ki je na voljo tudi na strani avtorja: *http://www.borgelt.net/fpm.html*. Ni možna uporaba nikakršnih omejitev glede glave, ali telesea pravil, razen izbiranja med že najdenimi pravili. Kot mero zanimivosti imamo na voljo le podporo in zaupanje. Edina prednost povezovalnih pravil v jeziku R je učinkovitost implementacije, kar pomeni hitrost in pa možnost obdelave velikih količin podatkov.

- Yale[6] (Yet Another Learning Enviroment) - Kot Orange in Weka je tudi Yale skupek algoritmov za rudarjenje podatkov. Pred kratkim se je program preimenoval v RapidMiner. Ponuja tudi iskanje povezovalnih pravil. Žal pa nudi program daleč najmanj funkcionalnosti. Ne ponuja nikakršnih omejitev, kot merilo zanimivosti pa le podporo in zaupanje.

Nekaj povezovalnih pravil najdenih s programom Orange, za upravno enoto Koper in vsa leta (napovedovanje poškodb):
SUPP CONF LIFT
0.010 0.848 7.343 prizorisce=N vreme=N oznaka_cesta_naselje=17042 ⇒ klas_n=U
0.012 0.803 6.952 prizorisce=N vreme=N ⇒ klas_n=U
0.010 0.759 6.574 vreme=N oznaka_cesta_naselje=17042 ⇒ klas_n=U
0.013 0.975 1.349 prizorisce=P oznaka_cesta_naselje=17042 ⇒ klas_n=B
0.016 0.914 1.265 prizorisce=C oznaka_cesta_naselje=93701 ⇒ klas_n=B
0.013 0.827 1.145 vreme=J oznaka_cesta_naselje=00311 ⇒ klas_n=B
0.012 0.826 1.143 prizorisce=C oznaka_cesta_naselje=00311 ⇒ klas_n=B
0.035 0.821 1.137 oznaka_cesta_naselje=00002 ⇒ klas_n=B
0.028 0.805 1.114 prizorisce=R oznaka_cesta_naselje=17042 ⇒ klas_n=B
0.020 0.803 1.111 prizorisce=C vreme=J oznaka_cesta_naselje=00002 ⇒ klas_n=B
0.012 0.803 1.111 prizorisce=C vreme=O oznaka_cesta_naselje=00011 ⇒ klas_n=B
0.011 0.798 1.104 vreme=O oznaka_cesta_naselje=00010 ⇒ klas_n=B

Povezovalna pravila najdena s programom Orange, za upravno enoto Koper in vsa leta (napovedovanje vrste nesreče):
SUPP CONF LIFT
0.001 0.840 5.696 mesec=4 prizorisce=C vreme=J oznaka_cesta_naselje=00111 ⇒ vrsta_n=NT
0.001 0.800 5.425 ura=9 prizorisce=C vreme=J oznaka_cesta_naselje=00111 ⇒ vrsta_n=NT

[5]Domača stran programa R: *http://cran.r-project.org/*
[6]Domača stran programa Yale: *http://rapid-i.com/*

0.001 0.800 5.425 ura=9 vreme=J oznaka_cesta_naselje=00111 ⇒ vrsta_n=NT
0.001 0.762 5.166 ura=18 prizorisce=C vreme=J oznaka_cesta_naselje=00111
⇒ vrsta_n=NT
0.001 0.750 5.086 mesec=4 vreme=J oznaka_cesta_naselje=00111 ⇒ vrsta_n=NT
0.002 0.735 4.986 ura=11 prizorisce=C vreme=J oznaka_cesta_naselje=00111
⇒ vrsta_n=NT
0.001 0.727 4.931 ura=15 prizorisce=C vreme=J oznaka_cesta_naselje=00111
⇒ vrsta_n=NT
0.001 0.720 4.882 ura=18 prizorisce=C oznaka_cesta_naselje=00111 ⇒ vrsta_n=NT
0.002 0.714 4.843 ura=11 vreme=J oznaka_cesta_naselje=00111 ⇒ vrsta_n=NT
0.001 0.750 2.944 mesec=11 ura=15 prizorisce=N oznaka_cesta_naselje=17042
⇒ vrsta_n=BT
0.001 0.727 2.855 mesec=3 d_v_t=2 prizorisce=R ⇒ vrsta_n=BT
0.001 0.720 2.826 mesec=5 prizorisce=R vreme=J oznaka_cesta_naselje=17042
⇒ vrsta_n=BT
0.001 0.720 2.826 d_v_t=2 prizorisce=R oznaka_cesta_naselje=93701 ⇒ vrsta_n=BT

Poglavje 6

Sklepne ugotovitve

V diplomskem delu so predstavljene tri skupine tehnik rudarjenja podatkov (razvrščanje kratkih časovnih vrst v skupine, razvrščanje točk v skupine, povezovalna pravila).

Z razvrščanjem kratkih časovnih vrst v skupine, sem uspel identificirati upravne enote, ki so si podobne po prometnem režimu. Najdene skupine so lahko v pomoč pri drugih analizah, ali pa pri načrtovanju strategij za izboljšanje prometne varnosti.

Razvrščanje točk v skupine, oziroma iskanje zgostitev nesreč, ponuja hitro pot do odkrivanja krajev (odsekov, križišč,...), ki si zaslužijo posebno pozornost prometnih strokovnjakov.

Pri povezovalnih pravilih pa žal nisem naletel na orodje, ki bi bilo dovolj zmogljivo in ponujalo dovolj nastavitev (omejitve glede željenih pravil), da bi nudilo enostavno pot do novega znanja.

V nadaljne raziskave s področja prometnih nesreč bi bilo predvsem potrebno vključiti dodatne podatke. Primer takih podatkov so podatki o štetju prometa. Ti podatki bi bistveno izboljšali razumevanje najdenih zgostitev prometnih nesreč (nevarne so zgostitve, kjer je veliko razmerje števila nesreč na število vozil). Potrebno pa bi bilo podrobneje pregledati področje iskanja povezovalnih pravil in poiskati programe, ki so bolj primerni za obravnavano domeno prometnih nesreč. Možna smer nadaljnega dela je uporaba algoritmov za odkrivanje podskupin (SD, CN2-SD, Apriori-SD, RSD). Ena od možnih smeri nadaljnega dela pa je tudi vključitev podrobnejših vremenskih podatkov med podatke uporabljene pri iskanju povezovalnih pravil.

Dodatek A

Šifranti

Priloženi so šifranti, uporabljeni v statističnih datotekah prometnih nesreč.

PRPO - POŠKODBA OSEBE IN KLASIFIKACIJA NESREČE
B - BREZ POŠKODBE
H - HUDA TELESNA POŠKODBA
L - LAŽJA TELESNA POŠKODBA
P - SLED POŠKODBE
S - SMRT

LOVC - KATEGORIJA CESTE, NASELJA
H - HITRA CESTA
L - LOKALNA CESTA
N - NASELJE Z ULIČNIM SISTEMOM
T - TURISTIČNA CESTA
V - NASELJE BREZ ULIČNEGA SISTEMA
0 - AVTOCESTA
1 - GLAVNA CESTA I. REDA
2 - GLAVNA CESTA II. REDA
3 - REGIONALNA CESTA I. REDA
4 - REGIONALNA CESTA II. REDA
5 - REGIONALNA CESTA III. REDA

PRKD - OPIS KRAJA DOGODKA
Ž - ŽELEZNIŠKI PREHOD
A - AVTOBUSNA POSTAJA
C - CESTA

E - ŽELEZNIŠKO POSTAJALIŠČE
K - KOLESARSKA STEZA ALI PLOČNIK
M - KROŽNO KRIŽIŠČE
N - NARAVNO OKOLJE
O - NARAVOVARSTVENO OBMOČJE
P - PARKIRNI PROSTOR
R - KRIŽIŠČE
V - VLAK
Z - PREHOD ZA PEŠCE

PRVZ - VZROK PROMETNE NESREČE
CE - NEPRAVILNOSTI NA CESTI
HI - NEPRILAGOJENA HITROST
NP - NEPRAVILNOSTI PEŠCA
OS - OSTALO
PD - NEUPOŠTEVANJE PRAVIL O PREDNOSTI
PR - NEPRAVILNO PREHITEVANJE
PV - PREMIKI Z VOZILOM
SV - NEPRAVILNA STRAN / SMER VOŽNJE
TO - NEPRAVILNOSTI NA TOVORU
VO - NEPRAVILNOSTI NA VOZILU
VR - NEUSTREZNA VARNOSTNA RAZDALJA

PRTN - TIP PROMETNE NESREČE
ČT - ČELNO TRČENJE
BT - BOČNO TRČENJE
NT - NALETNO TRČENJE
OP - OPLAŽENJE
OS - OSTALO
PP - POVOŽENJE PEŠCA
PR - PREVRNITEV VOZILA
PZ - POVOŽENJE ŽIVALI
TO - TRČENJE V OBJEKT
TV - TRČENJE V STOJEČE / PARKIRANO VOZILO

PRVR - VREMENSKE OKOLIŠČINE
D - DEŽEVNO
J - JASNO
M - MEGLA

N - NEZNANO
O - OBLAČNO
S - SNEG
T - TOČA
V - VETER

PRSP - STANJE PROMETA V ČASU PROMETNE NESREČE
E - NEZNANO
G - GOST
N - NORMALEN
R - REDEK
Z - ZASTOJI

PRPV - STANJE VOZIŠČA V ČASU PROMETNE NESREČE
BL - BLATNO
MO - MOKRO
OS - OSTALO
PN - POLEDENELO - NEPOSIPANO
PP - POLEDENELO - POSIPANO
SL - SNEŽENO - PLUŽENO
SN - SNEŽENO - NEPLUŽENO
SP - SPOLZKO
SU - SUHO

PRSV - VRSTA VOZIŠČA V ČASU PROMETNE NESREČE
AH - HRAPAV ASFALT / BETON
AN - NERAVEN ASFALT / BETON
AZ - ZGLAJEN ASFALT / BETON
MA - MAKADAM
OS - OSTALO

LODZ - ŠIFRANT DRŽAV

PRVU - ŠIFRANT VRSTE UDELEŽENCA V PROMETU
AV - VOZNIK AVTOBUSA
DS - VOZNIK DELOVNEGA STROJA
KM - VOZNIK KOLESA Z MOTORJEM
KO - KOLESAR
KR - X-KRŠITELJ - JRM

KV - VOZNIK KOMBINIRANEGA VOZILA
LK - VOZNIK LAHKEGA ŠTIRIKOLESA
MK - VOZNIK MOTORNEGA KOLESA
MO - VOZNIK MOPEDA
OA - VOZNIK OSEBNEGA AVTOMOBILA
OD - ODGOVORNA OSEBA
OS - OSTALO
PE - PEŠEC
PO - PRAVNA OSEBA
PT - POTNIK
SK - VOZNIK ŠTIRIKOLESA
SM - SKRBNIK MLADOLETNIKA
SP - SAMOSTOJNI PODJETNIK
SV - VOZNIK SPECIALNEGA VOZILA
TK - VOZNIK TRIKOLESA
TR - VOZNIK TRAKTORJA
TV - VOZNIK TOVORNEGA VOZILA

LOOB - ŠIFRANT UPRAVNIH ENOT IN STARIH OBČIN
5501 - AJDOVŠČINA
5502 - BREŽICE
5503 - CELJE
5504 - CERKNICA
5505 - ČRNOMELJ
5506 - DOMŽALE
5507 - DRAVOGRAD
5508 - GORNJA RADGONA
5509 - GROSUPLJE
5510 - HRASTNIK
5511 - IDRIJA
5512 - ILIRSKA BISTRICA
5513 - IZOLA
5514 - JESENICE
5515 - KAMNIK
5516 - KOČEVJE
5517 - KOPER
5518 - KRANJ
5519 - KRŠKO
5520 - LAŠKO

5521 - LENART
5522 - LENDAVA
5523 - LITIJA
5524 - LJUBLJANA BEŽIGRAD
5525 - LJUBLJANA CENTER
5526 - LJUBLJANA MOSTE POLJE
5527 - LJUBLJANA ŠIŠKA
5528 - LJUBLJANA VIČ RUDNIK
5529 - LJUTOMER
5530 - LOGATEC
5534 - METLIKA
5535 - MOZIRJE
5536 - MURSKA SOBOTA
5537 - NOVA GORICA
5538 - NOVO MESTO
5539 - ORMOŽ
5540 - PIRAN
5541 - POSTOJNA
5542 - PTUJ
5543 - RADLJE OB DRAVI
5544 - RADOVLJICA
5545 - RAVNE NA KOROŠKEM
5546 - RIBNICA
5547 - SEVNICA
5548 - SEŽANA
5549 - SLOVENJ GRADEC
5550 - SLOVENSKA BISTRICA
5551 - SLOVENSKE KONJICE
5552 - ŠENTJUR PRI CELJU
5553 - ŠKOFJA LOKA
5554 - ŠMARJE PRI JELŠAH
5555 - TOLMIN
5556 - TRBOVLJE
5557 - TREBNJE
5558 - TRŽIČ
5559 - VELENJE
5560 - VRHNIKA
5561 - ZAGORJE OB SAVI
5562 - ŽALEC

5564 - MARIBOR
5565 - PESNICA
5568 - RUŠE
5598 - MNZ
5599 - NEZNANA OBČINA

Dodatek B

SQL ukazi

SQL ukazi za kreiranje tabel (insert) in polnenje baze (load data).

```
CREATE TABLE nesreca (
id_nesreca CHAR(6) NOT NULL,
klas_nesreca CHAR(1) NOT NULL,
upravna_enota CHAR(4) NOT NULL,
cas_nesreca DATETIME NOT NULL,
naselje_ali_izven CHAR(1) NOT NULL,
kategorija_cesta CHAR(1) NULL,
oznaka_cesta_ali_naselje CHAR(5) NOT NULL,
tekst_cesta_ali_naselje VARCHAR(25) NOT NULL,
oznaka_odsek_ali_ulica CHAR(5) NOT NULL,
tekst_odsek_ali_ulica VARCHAR(25) NOT NULL,
stacionazna_ali_hisna_st VARCHAR(9) NULL,
opis_prizorisce CHAR(1) NOT NULL,
vzrok_nesreca CHAR(2) NOT NULL,
tip_nesreca CHAR(2) NOT NULL,
vreme_nesreca CHAR(1) NOT NULL,
stanje_promet CHAR(1) NOT NULL,
stanje_vozisce CHAR(2) NOT NULL,
stanje_povrsina_vozisce CHAR(2) NOT NULL,
x INT(11) NULL,
y INT(11) NULL,
x_wgs84 DOUBLE NULL,
y_wgs84 DOUBLE NULL,
PRIMARY KEY (id_nesreca));
```

```
CREATE TABLE oseba (
id_nesreca CHAR(6) NOT NULL,
povzrocitelj_ali_udelezenec CHAR(1) NULL,
starost TINYINT(3) unsigned NULL,
spol CHAR(1) NULL,
upravna_enota CHAR(4) NOT NULL,
drzavljanstvo CHAR(3) NOT NULL,
poskodba CHAR(1) NULL,
vrsta_udelezenca CHAR(2) NULL,
varnostni_pas_ali_celada CHAR(1) NULL,
vozniski_staz_LL TINYINT(3) unsigned NULL,
vozniski_staz_MM TINYINT(3) unsigned NULL,
alkotest DECIMAL(3, 2) NULL,
strokovni_pregled DECIMAL(3, 2) NULL,
starost_d CHAR(1) NULL,
vozniski_staz_d CHAR(1) NULL,
alkotest_d CHAR(1) NULL,
strokovni_pregled_d CHAR(1) NULL);

LOAD DATA INFILE 'D:/users/domen/promet_baza/nesreca.tab' INTO
TABLE
nesreca FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\r\n';

LOAD DATA INFILE 'D:/users/domen/promet_baza/oseba.tab' INTO
TABLE oseba
FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\r\n';
```

Slike

Tabele

Literatura

[1] A. Appice, M. Berardi, M. Ceci, M. Lapi, D. Malerba, A. Turi, "Mining Interesting Spatial Association Rules: Two Case Studies," v zborniku *Dodicesimo Convegno Nazionale su Sistemi Evoluti per Basi di Dati*, Cagliari, Italija, junij 2004, str. 86-97.

[2] T. Bernhardsen, *Geographic Information Systems: an Introduction*, New York: Wiley, 2002.

[3] H. Blockeel, L. De Raedt, "Top-down induction of first order logical decision trees," *Artificial Intelligence*, št. 1-2, zv. 101, str. 285-297, 1998.

[4] H. Blockeel, L. De Raedt, J. Ramon, "Top-down induction of clustering trees," v zborniku *15th International Conference on Machine Learning*, Wisconsin, ZDA, julij 1998, str. 55-63.

[5] M. Chong, A. Abraham, M. Paprzycki, "Traffic Accident Data Analysis Using Machine Learning Paradigms," *Informatica: An International Journal of Computing and Informatics*, št. 1, zv. 29, str. 89-98, 2005.

[6] M. Chong, A. Abraham, M. Paprzycki, "Traffic Accident Analysis Using Decision Trees and Neural Networks," v zborniku *IADIS International Conference on Applied Computing*, Portugalska, 2004, zv. 2, str. 39-42.

[7] L. Dehaspe, H. Toivonen, "Discovery of frequent datalog patterns," *Data mining and Knowledge Discovery*, št. 1, zv. 3, str. 7-36.

[8] J. Han, M. Kamber, *Data Mining: Concepts and Techniques*, San Francisco [itd.]: Morgan Kaufmann, 2001.

[9] D. Hand, H. Mannila, P. Smith, *Principles of Data Mining*, Cambridge, London: MIT Press, 2001, pogl. 1.

[10] D. Mladenić, N. Lavrač, M. Bohanec, S. Moyle *Data mining and decision support: integration and collaboration*, Boston/Dordrecht/London: Kluwer Academic Publishers, 2003, pogl. 12.

[11] T. Mohorič, *Načrtovanje Relacijskih Podatkovnih Baz*, Ljubljana: BI-TIM, 1997, pogl. 3.

[12] C. Möller-Levet, F. Klawonn, K.-H. Cho, O. Wolkenhauer, "Fuzzy Clustering of Short Time-Series and Unevenly Distributed Sampling Points," v zborniku *IDA'2003*, Berlin, Nemčija, Avgust 2003, str. 330-340.

[13] P.-N. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Boston [itd.]: Pearson Addison Wesley, 2006.

[14] T. B. Tesema, A. Abraham, C. Grosan, "Rule Mining And Classification Of Road Traffic Accidents Using Adaptive Regression Trees," *International Journal of Simulation: Systems, Science & Technology*, št. 10-11, zv. 6, str. 80-94, 2005.

[15] L. Todorovski, P. Ljubič, N. Lavrač, S. Džeroski, R. Bellazzi, *Qualitative clustering of short time series*, 2003.

[16] (2007) Data Clustering - Wikipedia, the free encyclopedia. Dostopno na: http://en.wikipedia.org/wiki/Data_clustering

[17] (2007) Keyhole Markup Language - Wikipedia, the free encyclopedia. Dostopno na: http://en.wikipedia.org/wiki/Keyhole_Markup_Language

Izjava

Izjavljam, da sem diplomsko nalogo izdelal samostojno pod vodstvom mentorice prof. dr. Neže Mramor Kosta in somentorice prof. dr. Nade Lavrač. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Ljubljana, 15.6.2007 Domen Jesenovec