UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Marko Žerjal

Napovedovanje padavin z atributnim strojnim učenjem

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: prof. dr. Ivan Bratko

Ljubljana, december 2009

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 01575/2009
Datum: 01.09.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **MARKO ŽERJAL**

Naslov: **NAPOVEDOVANJE PADAVIN Z ATRIBUTNIM STROJNIM UČENJEM**

PREDICTING THE AMOUNT OF RAINFALL WITH PROPOSITIONAL MACHINE LEARNING

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Tema te diplomske naloge je napovedovanje količine padavin z uporabo strojnega učenja iz podatkov o padavinah na radarskih slikah ter sinoptičnih podatkov iz avtomatskih merilnih postaj v Sloveniji. Naučeni model naj napoveduje količino padavin za 60 x 80 osnovnih prostorskih enot za 10, 20, 30, ... minut vnaprej. Primerjajte dve metodi napovedovanja: (a) z učenjem napovedi za 10 min vnaprej ter uporabo tega modela za simulacijo za daljše obdobje z 10 minutnim simulacijskim korakom, (b) z naposrednim učenjem napovedi za 10, 20 30, ... minut vnaprej. V obeh primerih uporabite znane metode atributnega učenja, kot so drevesa, pravila in naivni Bayesov klasifikator. Preučite tudi možnost avtomatskega grupiranja manjših prostorskih enot v regije ter učenje napovednih modelov za posamezne regije.

Mentor:

akad. prof. dr. Ivan Bratko

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Franc Solina

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!

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani Marko Žerjal,

z vpisno številko 63040186,

sem avtor diplomskega dela z naslovom:

Napovedovanje padavin z atributnim strojnim učenjem

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdela samostojno pod mentorstvom

  prof. dr. Ivana Bratka

- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela

- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 14. 12. 2009 Podpis avtorja:

Zahvala

Najprej bi se zahvalil prof. dr. Ivanu Bratku za mentorstvo pri diplomi. Prav tako bi se zahvalil vsem zaposlenim na XMEDIA projektu za vse nasvete, ki so mi pomagali pri delu, posebej pa Andreju Oblaku, ki mi je svetoval glede samega oblikovanja diplome in je bil soavtor algoritma za določanje smeri in hitrosti vetra, ki je bil uporabljen v tej diplomski nalogi. Poleg tega bi se zad zahvalil tudi Nini Barbič, ki je lektorirala celotno diplomsko nalogo in pa Tomažu Potočniku, ki mi je pomagal prevesti povzetek.

Kazalo

Seznam uporabljenih kratic in simbolov

KNN - k najbližjih sosedov (k nearest neighbors)

CA - Klasifikacijska točnost (classification accuracy)

RMSE - Koren povprečne kvadratne napake (root mean squared error)

Povzetek

V diplomski nalogi smo se ukvarjali z napovedovanjem padavin na podlagi podatkov pridobljenih z združevanjem numeričnih meteoroloških meritev in radarskih slik padavin. Ti dve skupini podatkov smo združili tako, da smo radarske slike enakomerno razdelili na celice, katerim smo lahko določil numerične vrednosti na podlagi prevladujoče stopnje padavin znotraj njih, kar je primerna oblika za zapis atributa pri metodah, ki smo jih uporabili za napovedi.

Združene podatke smo nato razdelili na učno in testno množico in z izbranimi metodami atributnega strojnega učenja skušali priti do čim boljšega modela za napovedovanje padavin. Uporabili smo naslednje metode: klasifikacijsko drevo, KNN, naivni Bayesov klasifikator in regresijsko drevo. Najboljše rezultate sta dosegla klasifikacijsko drevo ter KNN. Regresijsko drevo smo z ostalimi metodami primerjali tako, da smo izhode drevesa diskretizirali in izkazalo se je, da so na tak način dobljeni rezultati slabši, kot rezultati klasifikacijskega drevesa ter KNN algoritma. Tudi naivni Bayes se je izkazal kot manj primeren pristop za obdelavo tovrstnih podatkov.

Poleg že obstoječih metod atributnega strojnega učenja smo tekom naloge razvili še skupino metod za delitev ozemlja, ki ga pokrivajo radarske slike padavin, na regije, kar omogoča upoštevanje lokalnih značilnosti vremenskih pojavov v Sloveniji. Z dobljenimi rezultati delitve na regije smo zadovoljni, a je na tem področju še veliko možnosti za izboljšavo, ki jih moramo v bodoče še raziskati.

Ključne besede

atributno strojno učenje, združevanje podatkov, napovedovanje padavin, delitev na regije

Abstract

In this diploma thesis we tried to create rain forecasts with the use of data obtained by combining numerical meteorological information with information gathered from radar images of rain activity. We have combined the two groups of data by partitioning the radar images into equally sized cells, which were then assigned a numerical value based on the prevailing degree of rainfall inside the cells. This is in an appropriate format of attributes in the models used to make the forecasts.

The combined set of data was then split into a learning and a testing set and we have tried to make the most accurate forecasts possible with the use of selected attribute-based machine learning methods. The following methods were used: decision trees, KNN, naive Bayesan classifier and regression trees. The best results were obtained by the decision trees and by the KNN algorithm. Regression trees were compared with other methods by discretizing their results which revealed that they were not as accurate as decision trees and the KNN algorithm. The naive Bayesan classifier also proved to be less adequate for use on this type of data.

Beside the already existing methods of machine learning we also developed a set of methods which divide the area covered by the radar images into regions, thus enabling us to take into account the local weather characteristics of various parts of Slovenia. We assess the results of regional division as good, but there are still many possibilities for improvement to be researched in the future.

Key words

attribute-based machine learning, combining data from different sources, rainfall forecasts, regional division

1 - Uvod

Napovedovanje vremena je notorično težaven problem, saj zaradi mnogih spremenljivk, ki jih po večini niti ne poznamo, zelo težko predvidevamo, kaj se bo zgodilo naslednji teden, dan, uro, ali, kot se izkaže, celo čez deset minut. Večina pristopov k napovedovanju vremena, česar del je tudi napovedovanje padavin, temelji na ekspertnem znanju in na fizikalnih modelih, ki skušajo s pomočjo uveljavljenih in dokazanih fizičnih pravil simulirati vremenske pojave, ki bi lahko sledili nekemu danemu stanju.

V tej diplomski nalogi smo se odločili za radikalno drugačen pristop k napovedovanju vremena oziroma natančneje padavin. Namesto upoštevanja ekspertnega znanja za vremensko domeno smo se odločili preizkusiti, kako se na reševanju težave izkažejo metode strojnega učenja. Torej bi radi z iskanjem vzorcev v vremenskih spremembah skušali priti do dovolj zanesljive napovedi, da bi bil tak način tudi praktično uporaben. Pri izdelavi napovedovalnega modela smo se odločili za uporabo metod atributnega strojnega učenja.

Atributno strojno učenje je področje umetne inteligence, ki se ukvarja z odkrivanjem znanja iz podatkov. Iz učne množice podatkov se metode atributnega strojnega učenja naučijo odvisnosti med vrednostmi atributov in vrednostjo razreda, na testni množici pa se preverja dejansko uspešnost metod, saj so rezultati pri testiranju modela na učni množici pogosto zavajajoči, ker se metode z lahkoto preveč prilagodijo na šum v učni množici. Če imamo na primer v učni množici eno samo jabolko in je to zeleno, se bi model naučil, da so jabolka vedno zelena brez izjeme, kar pa seveda ni čisto res in vsaka testna množica z nekaj več jabolk bi tako zmotno hipotezo zavrnila. Ker se vse več podatkov hrani v digitalni obliki, so te metode vedno bolj uporabne in tudi komercialno zanimive ter se že precej uporabljajo na področjih industrije, medicine, ekonomije, spletne trgovine ter oglaševanja itd., tako da bi bilo smiselno predvidevati, da lahko s temi metodami kaj dosežemo tudi na področju napovedovanja vremena.

Za našo nalogo imamo na voljo več vrst podatkov, in sicer numerične meritve iz meteoroloških postaj in radarske slike nad območjem Slovenije. To je prva težava, ki smo jo srečali pri delu na nalogi, saj podatki v takih oblikah niso združljivi. Numerične meritve bi se verjetno dalo kako spremeniti v slikovno obliko, toda še boljša ideja je sliko razbiti v logične numerične vrednosti. Tu imamo srečo, saj so slike iz računalniškega vidika zelo lahko berljive. Vsaka slikovna točka ima lahko namreč eno od petih vrednosti:

brez padavin – 0.0 mm/h

malo padavin – 0.0 do 0.8 mm/h

srednje veliko padavin – 0.8 do 5.0 mm/h

veliko padavin – 5.0 do 31.6 mm/h

intenzivne padavine – 31.6 do 100 mm/h

Cilj, ki smo si ga zadali v tej diplomski nalogi, je izdelava modela, ki bi lahko napovedoval padavine nad območjem Slovenije z ožjo okolico natančno in brez človeške pomoči. Toda na tem mestu se je treba vprašati, na kakšno natančnost lahko sploh upamo pri takem projektu? Vremenske spremembe so lahko zelo kaotične in ni enostavno najti načina za primerjavo s popolno natančnostjo. Zato je začetni cilj, ki smo si ga zadali, narediti model, ki lahko izdela točnejše napovedi kot jih dobimo, če za napoved uporabimo kar radarsko sliko prejšnjega stanja.

Problema smo se lotili tako, da smo združili podatke merilnih postaj in podatke, ki smo jih razbrali iz radarskih slik padavin, v dve ločeni podatkovni bazi. Podatki so bili pridobljeni iz slik dovolj časovno narazen, da ne bi zaradi podobnosti vremenskih pojavov v krajšem časovnem razmaku prišlo do lažno visoke klasifikacijske natančnosti pri testiranju. Vse izbrane metode za napovedovanje smo nato na podatkih testirali in ocenili, da se za nadaljnjo uporabo najbolj izplača izbrati klasifikacijsko drevo, ker je imelo najvišjo klasifikacijsko točnost in sprejemljivo nizko časovno zahtevnost. To točnost smo nato uspeli še izboljšati z uporabo algoritmov za deljenje radarskih slik na regije.

Dobljeni rezultati so zadovoljivi, saj je klasifikacijska točnost že če uporabimo za napoved zgolj regresijsko drevo, naivni Bayes, KNN ali klasifikacijsko drevo nad klasifikacijsko točnostjo napovedi predhodnega stanja. Za rezultate deljenja radarskih slik na regije pa verjamemo, da bi se dalo narediti še veliko izboljšav, a je iz dobljenih rezultatov že razvidno, da je deljenje na regije smiselno početje in to spoznanje je za namene te diplomske naloge tudi zadovoljiv rezultat.

2 - Pridobivanje podatkov

2.1 Delitev slik na celice

Za napovedovanje smo se odločili uporabiti pristop, pri katerem ne napovemo kar celotne slike padavin naenkrat, ampak namesto tega radarski posnetek raje razdelimo na več enot in nato delamo napovedi za vsako posamično tako dobljeno celico. Tak pristop smo izbrali zato, ker ima nekaj prednosti pred napovedovanjem cele slike naenkrat, saj v primeru, ko moramo delati napoved za sliko, ki je zelo različna čemerkoli, kar imamo v učni množici, še vedno najde podobnosti na celičnem nivoju in je zato v takih primerih napoved veliko boljša. Posledica tega je tudi, da lahko napovedujemo razmeroma natančno tudi z dokaj majhno učno množico.

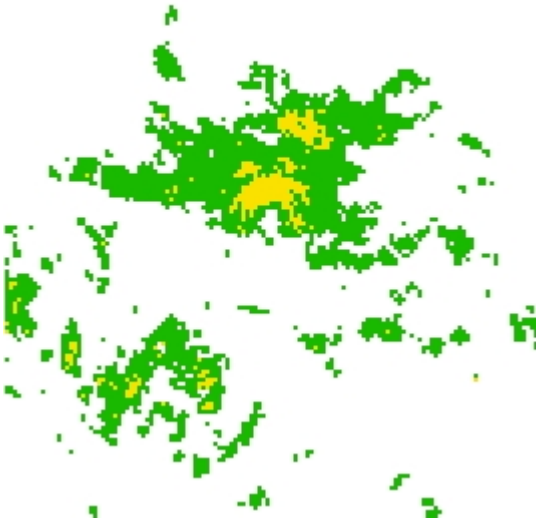
Pri delitvi slike na celice smo naleteli na dve ključni vprašanji, in sicer kakšne celice izbrati in kako diskretizirati vrednosti za celice.

2.1.1 Izbira oblike in velikosti celic

Za obliko celic smo izbrali kar kvadrate, saj tako z največjo lahkoto razdelimo celo sliko in so tako celice med seboj vedno primerljive. Če bi sliko razdelili denimo na geografske regije, bi izgubili na prenosljivosti podatkov, saj ne bi mogli iz podatkov za Prekmurje nič napovedovati za nobeno drugo regijo, ker bi že same celice imele drugačne atribute in bi potemtakem za vsako rabili narediti ločen model, medtem ko pri homogenih celicah ni takih omejitev in se da deljenje na regije implementirati naknadno.

Velikost celic pa po drugi strani ni bila tako trivialna izbira. Če izberemo prevelike celice, izgubimo pri diskretizaciji veliko več informaciji kot z malimi celicami, medtem ko izbira malih celic dovoljuje boljšo resolucijo naših napovedi, vendar strmo poveča časovno zahtevnost izvajanja, saj imamo več učnih primerov in poleg tega moramo narediti več atributov, če želimo vključiti dovolj široko okolico vsake celice, da ne izgubimo na klasifikacijski točnosti.

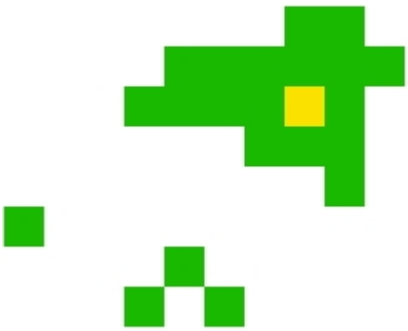
Kako radarska slika izgleda pri celicah velikosti 2X2 je prikazano na sliki 2-1, pri celicah 5X5 na sliki 2-2, pri celicah 10X10 na sliki 2-3, in pri celicah 20X20 na sliki 2-4.



Slika 2-1 Radarska slika padavin pri celicah velikosti 2X2

Slika 2-2 Radarska slika padavin pri celicah velikosti 5X5

Slika 2-3 Radarska slika padavin pri celicah velikosti 10X10



Slika 2-4 Radarska slika padavin pri celicah velikosti 20X20

Po nekaj preizkusih smo se odločili za celice velikosti 5X5 točk, ker so se take mere izkazale za najboljši kompromis med obema možnostma.

2.1.2 Diskretiziranje vrednosti

Pri napovedovanju nas zanima, kakšna stopnja padavin bo v celici, podatek koliko točk bo deževnih in koliko ne pa je stranskega pomena, zato je pomembna in potrebna diskretizacija vrednosti padavin. Najbolj enostavna rešitev bi bila seveda izbira večinske stopnje padavin za vsako celico in verjetno taka izbira niti ne bi bila preveč napačna, a bi potemtakem za celico s 34% točk brez padavin, 33% točk z lahkimi padavinami in 33% točk s srednjimi padavinami določili, da je brez padavin, kar ni prekrivno z dejanskim stanjem. Zato smo v izogib tovrstnim diskretizacijam raje uporabili postopek, pri katerem vsaka točka brez padavin šteje kot 0 točk, točka z malo padavinami šteje kot 1 točka, točka s srednje veliko padavinami kot 2 točki, točka z veliko padavinami kot 3 točke in točka z intenzivnimi padavinami kot 4 točke. Na podlagi teh podatkov seštejemo vse točke v celici in rezultat delimo s številom točk. Če je dobljen rezultat pod 0.5, je celica brez padavin, če je rezultat med 0.5 in 1.5, ima celica malo padavin, od 1.5 do 2.5 so srednje velike padavine, od 2.5 do 3.5 so velike padavine in od 3.5 naprej so intenzivne padavine. V prej navedenem primeru bi tako dobili:

$$\frac{(34*0+33*1+33*2)}{100}=0.99$$

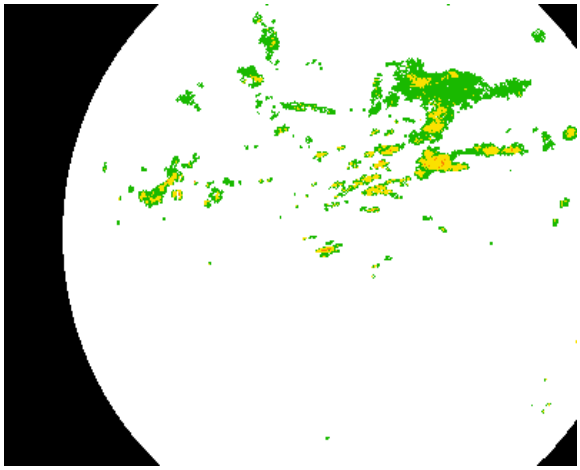
0.99 sodi v lahke padavine, kar je v našem primeru veliko bolj natančno.

2.2 Ugotavljanje smeri in hitrosti gibanja padavin

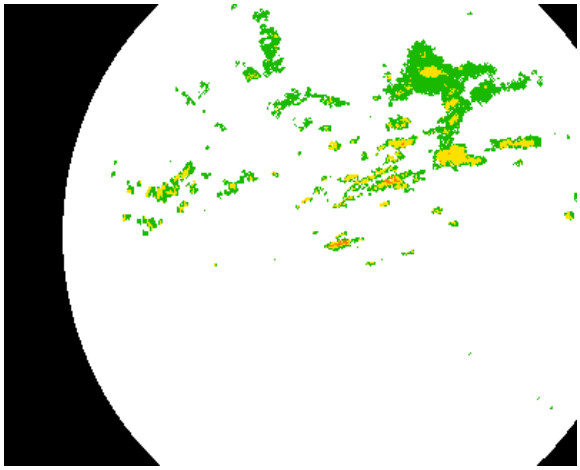
Kot pomembna dejavnika pri napovedovanju padavin se pojavita tudi smer in hitrost vetra. A pri tem naletimo na težavo, saj so sinoptične meritve opravljene pri tleh, mi pa moramo ugotoviti hitrost in smer vetra na višinah, kjer potujejo deževni oblaki. V ta namen sva v sodelovanju z Andrejem Oblakom razvila metodo, ki ugotovi premik padavin, in z njeno pomočjo je mogoče predpostaviti tudi smer in hitrost vetra na nadmorski višini, ki nas zanima. Ker pa veter iz radarskih slik ni jasno razvden, ga naša metoda ne računa direktno, ampak dobi zanj neko posredno oceno iz gibanja padavin.

Metoda deluje po naslednjih korakih:

1. Poleg slike, iz katere napovedujemo, potrebujemo še sliko predhodnega stanja, saj bomo ugotavljali premik padavin iz razlike teh dveh slik, v tem primeru slike 2-5, ki je slika predhodnega stanja, in slike 2-6, ki prikazuje trenutno stanje.

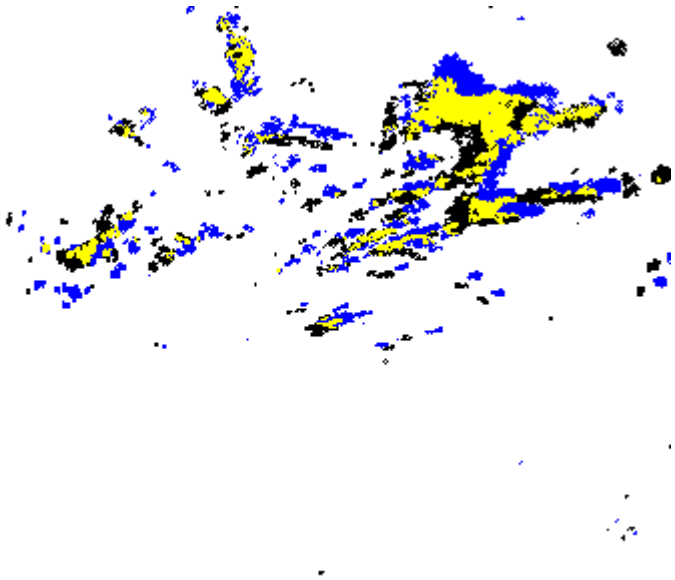


Slika 2-5 Radarska slika padavin ob času *t*



Slika 2-6 Radarska slika padavin ob času *t+1*

2. V drugem koraku sliki "odštejemo", pri čemer področje, kjer so bile padavine na starejši sliki, obarvamo črno, področje, kjer so padavine na novejši sliki, obarvamo modro, območje, kjer so padavine na obeh slikah, pa obarvamo rumeno. Rezultat je v našem primeru viden na sliki 2-7

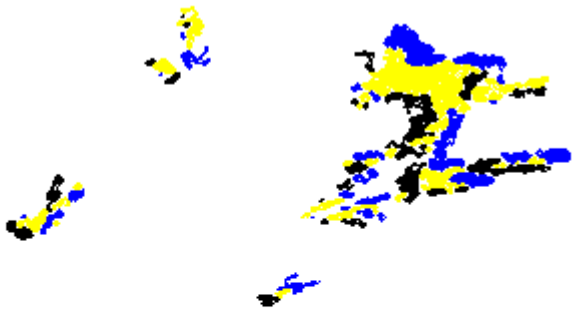


Slika 2-7 Razlika med sliko 2-6 in 2-5

3. V naslednjem koraku se slika filtrira, da se odstranijo moteče majhne gmote, kot je prikazano na sliki 2-8

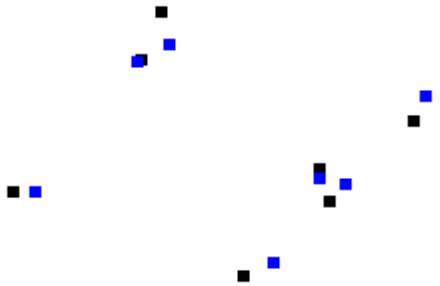
Slika 2-8 Rezultat filtriranja slike 2-7

4. Sliko se nato segmentira, kar pomeni, da se poveže pobarvana območja skupaj glede na sosednost. Ohrani se le tiste segmente, ki vsebujejo področja vseh treh barv, kot je to prikazano na sliki 2-9



Slika 2-9 Prikaz rezultata segmentacije

5. Zadnji korak metode je še ugotovitev premika padavin. To se naredi tako, da se za vsak segment izračuna središče črne ter središče modre barve, pri čemer se rumena barva v obeh računih upošteva za določanje središča. S tem postopkom dobimo sliko



2-10:

Slika 2-10 Začetne in končne točke vektorjev premika

6. Črne in modre točke nam sedaj predstavljajo začetek in konec vektorjev premika za posamezni segment. Potrebno je le še sešteti vektorje ter jih deliti z njihovim številom in dobimo povprečen vektor premika padavin za celotno sliko.

Razmišljali smo tudi o opciji, da bi se uporabljalo premik vsake deževne gmote posebej, tako da bi različne celice na isti sliki lahko imele tudi različne podatke v tem atributu, a smo se raje odločili za zgoraj opisani pristop, ker se s tem, ko vzamemo povprečje več ocen znebimo nekaj šuma in torej dobimo boljšo oceno, drugi razlog pa je ta, da je na tako majhnem območju kot ga radarske slike pokrivajo veter (in posledično tudi gibanje padavin) večinoma enakomeren.

2.3 Vključevanje sinoptičnih podatkov

Sinoptični podatki so pridobljeni iz avtomatskih merilnih postaj, ki so po Sloveniji razporejene kot kaže slika 2-11:

Slika 2-11 Prikaz meteoroloških postaj v Sloveniji

Na voljo smo imeli podatke o datumu in času, postaji, zračni temperaturi, temperaturi rosišča, zračnem tlaku, hitrosti vetra, smeri vetra, oblačnosti, višini oblačnosti, vodoravni vidljivosti, količini padavin, količini padavin namerjeni v zadnjih urah, dežju ter napovedi dežja za prihodnji dan. Od tega smo pa pri napovedovanju uporabili le podatke o postaji, datumu in času, zračni temperaturi, temperaturi rosišča ter zračnem tlaku. Drugi podatki, kot na primer smer in hitrost vetra, so se izkazali za nekoristne, saj smo veliko boljše meritve lahko razbrali iz radarskih slik z metodo opisano v nadaljevanju.

Sinoptične meritve so večinoma opravljene vsako uro in ne vsakih deset minut, kakor radarske slike, zato smo za vsako sliko uporabili podatke iz zadnje polne ure za najbližjo vremensko postajo. Če podatki za zadnjo polno uro niso na voljo, uporabimo podatke izpred dveh ur in če teh ni, pa podatke izpred treh ur. V primeru da nobenih od teh podatkov ni na voljo, se za podatke zapiše neznano vrednost.

2.4 Sestavljanje učne in testne množice

2.4.1 Izbira atributov

Prvo vprašanje s katerim smo se srečali pri sestavljanju učne in testne množice je, kateri atributi naj bodo vključeni v podatke. Tu gre za več kot le za izbiro atributov, saj smo nekatere morali smiselno sestaviti. Končen seznam atributov izgleda tako:

Slika

Meta atribut, kjer je shranjeno ime slike s katere je bil primer vzet. Ker je meta atribut, se pri klasifikaciji ne uporablja in je bil namenjen zgolj kot povratna referenca.

X

Meta atribut, ki pove x koordinato celice na sliki. Uporabljan je bil pri delitvi slik na regije, kar je natančneje razloženo v naslednjem poglavju.

Y

Meta atribut, ki pove y koordinato celice na sliki.

Smer vetra

Bolj natančno ime bi bilo smer gibanja padavin. Atribut je bil izračunan na podlagi algoritma iz prejšnjega odseka. Je diskretiziran in izražen v smereh neba.

Hitrost vetra

Bolj natančno ime bi bilo hitrost gibanja padavin. Atribut je bil izračunan na podlagi algoritma iz prejšnjega odseka. Izražen je v dolžini vektorja premika v točkah.

Napoved

Atribut predstavlja klasifikacijski razred. Zaseda lahko vrednosti 0, 1, 2, 3 ali 4, ki pomenijo jakost padavin.

Grafični prikaz določanja sosednjih celic je razviden na sliki 2-12:

Slika 2-12 Barvni prikaz delitve sosednjih celic glede na smeri neba(celice z dvema barvama pripadajo obema smerem neba)

Prejšnje stanje

Jakost padavin v celici v stanju iz katerega delamo napoved. Zaseda lahko vrednosti 0, 1, 2, 3 ali 4.

Severozahod

Seštevek jakosti padavin v celicah severozahodno od tiste za katero klasificiramo. Zaseda lahko vrednosti od 0 do 16.

Severovzhod

Seštevek jakosti padavin v celicah severovzhodno od tiste za katero klasificiramo. Zaseda lahko vrednosti od 0 do 16.

Jugozahod

Seštevek jakosti padavin v celicah jugozahodno od tiste za katero klasificiramo. Zaseda lahko vrednosti od 0 do 16.

Jugovzhod

Seštevek jakosti padavin v celicah jugovzhodno od tiste za katero klasificiramo. Zaseda lahko vrednosti od 0 do 16.

Sever

Seštevek jakosti padavin v celicah severno od tiste za katero klasificiramo. Zaseda lahko vrednosti od 0 do 16.

Jug

Seštevek jakosti padavin v celicah južno od tiste za katero klasificiramo. Zaseda lahko vrednosti on 0 do 16.

Vzhod

Seštevek jakosti padavin v celicah vzhodno od tiste za katero klasificiramo. Zaseda lahko vrednosti od 0 do 16.

Zahod

Seštevek jakosti padavin v celicah zahodno od tiste za katero klasificiramo. Zaseda lahko vrednosti od 0 do 16.

Zračna temperatura

Izražena v stopinjah Celzija, izraža temperaturo zraka izmerjeno v najbližji merilni postaji ob zadnji polni uri.

Temperatura rosišča

Izražena v stopinjah Celzija, izraža temperaturo zraka izmerjeno v najbližji merilni postaji ob zadnji polni uri.

Zračni tlak

Izražen v milibarih, izraža zračni tlak izmerjen v najbližji merilni postaji ob zadnji polni uri.

Vsi atributi razen meta atributov x in y so diskretni, ker sem tako dobil boljše rezultate pri klasifikaciji.

2.4.2 Izbira slik ter izločanje podatkov

Ker je podatkovna baza slik, ki smo jo dobili za to nalogo ogromna, saj šteje kar preko dvanajst tisoč slik, smo se morali omejiti na manjše število slik. Po nekaj poskusih se je izkazalo, da že zbirka okoli 50 slik zagotavlja dovolj veliko podatkovno bazo, da se lahko iz nje naučimo zadovoljivega klasifikacijskega modela. Pri tem smo morali zagotoviti, da so slike dovolj raznovrstne, tako da so bile zastopane vse smeri gibanja padavin ter da je za vsako smer bila na slikah predstavljena vsaka stopnja jakosti padavin.

Kljub temu, da v podatkovni bazi ni bilo slik, ki bi bile čisto brez padavin, se je vseeno zgodilo, da je velika večina celic v podatkovni bazi bila brez padavin. Posledica tega je, da je klasifikacijska točnost za take podatke zelo visoka in torej težje primerljiva za modele, v podatkovni bazi pa imamo veliko učnih in testnih podatkov, ki nam pravzaprav ne dajo nobenega koristnega znanja, ampak vseeno povzročajo veliko časovno zahtevnost pri napovedovanju, zato smo naključno izločili 95% teh “praznih“ podatkov, da smo zmanjšali razlike v zastopanosti vseh jakosti padavin v podatkih.

Za učno in testno množico smo seveda izbrali dve ločeni skupini slik in se v obeh primerih držali načel izpuščanja praznih slik ter dela praznih celic. Pri tem smo tudi bili pozorni, da je med vsako sliko v učni množici in vsako sliko v testni množici vsaj en dan presledka, tako da lokalna časovna podobnost ne bi povzročila previsoke ocene klasifikacijske točnosti.

3 - Preizkušanje in uporaba različnih metod klasifikacije

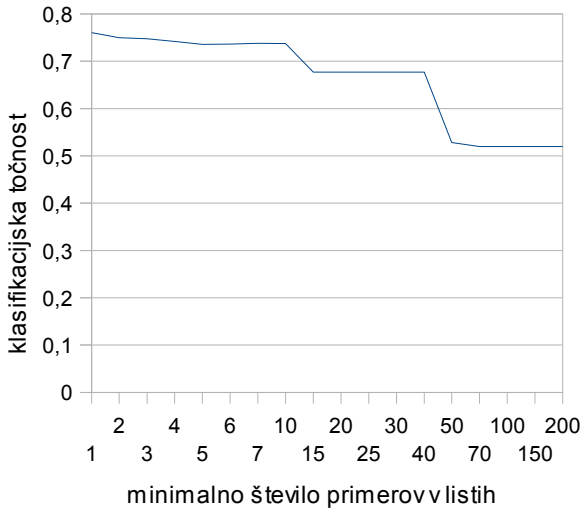
Za klasifikacijo smo uporabljali metode iz programskega paketa Orange [2]. Programski paket je odprtokoden in prosto dosegljiv na spletnem naslovu www.ailab.si/orange.

3.1 Klasifikacijsko drevo

Prva metoda, ki smo jo uporabili pri delu, je klasifikacijsko drevo. Klasifikacijsko drevo je sestavljeno iz notranjih vozlišč, ki predstavljajo atribute, povezav, ki predstavljajo podmnožice vrednosti atributov in listov, ki predstavljajo diskretne oznake razredov [1].

Metoda je že od samega začetka izgledala obetavna, saj lahko prenese odvisnosti med atributi in ta lastnost je bistvenega pomena pri napovedovanju domen kot na primer vreme. Metoda ima tudi sprejemljivo časovno zahtevnost, kar je pomembno za deljenje na regije, ki bo kasneje opisano, a je zahtevala nekoliko več dela iz naše strani, saj je od parametrov rezanja drevesa v veliki meri odvisna njegova klasifikacijska točnost.

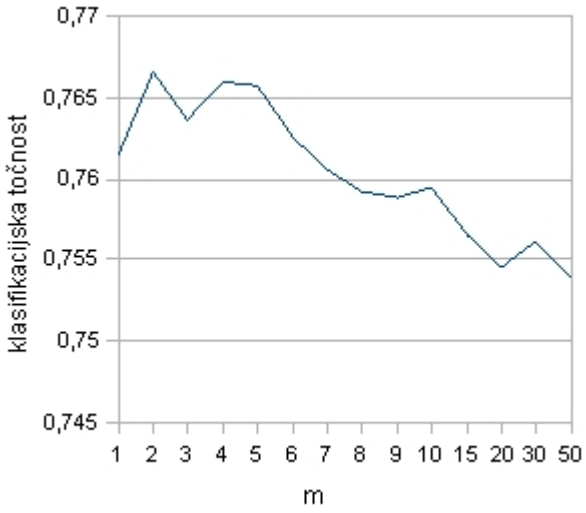
Rezanje drevesa je lahko izvedeno med gradnjo ali naknadno, na že zgrajenem drevesu [1]. Najprej smo poskusili drevo rezati že med samo gradnjo, tako da smo omejili liste na minimalno število primerov, ki jih morajo vsebovati. Glede na izbrano število primerov smo dobili graf prikazan na sliki 3-1.



Slika 3-1 Klasifikacijska točnost drevesa pri rezanju med gradnjo

Kot je razvidno iz grafa, se na podatkih najbolj splača uporabiti drevo z najmanj enim primerom na list, kar je zanimivo, saj pri 12.000 učnih primerih to pomeni zelo veliko in razvejano drevo. To je znak, da v domeni veljajo zelo kompleksna pravila in da je med atributi veliko odvisnosti.

Seveda smo klasifikacijsko drevo testirali tudi pri rezanju po gradnji, s spreminjanjem vrednosti *m*, ki se uporablja pri *m*-oceni. Rezultati so prikazani na sliki 3-2.

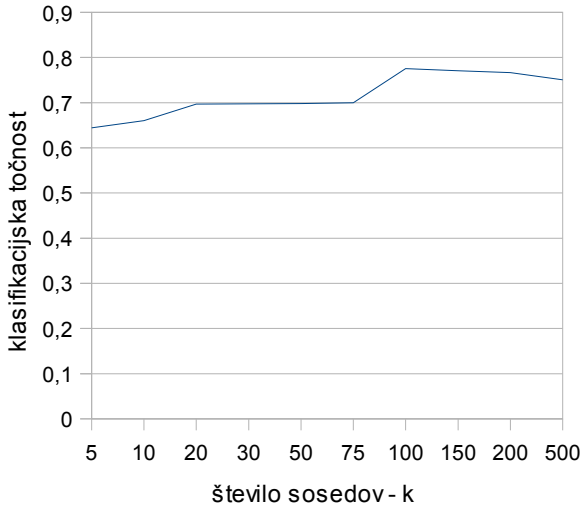


Slika 3-2 Klasifikacijska točnost drevesa pri rezanju po gradnji

Rezanje po gradnji se obnese kar v redu, celo bolje kakor rezanje med gradnjo. Po rezultatih je razvidno, da je boljše če uporabimo za parameter m nižjo vrednost, saj s povečevanjem parametra izgubljamo na točnosti.

3.2 K najbližjih sosedov

Poznan tudi kot KNN algoritem deluje tako, da poišče v učni množici *k* primerov, ki so obravnavanemu najbolj podobni in primer nato klasificira v večinski razred najdenih sosedov. Pristop je zelo enostaven, a se izkaže tudi za učinkovitega. Rezultati v odvisnosti od parametra *k* so prikazani na sliki 3-3.



Slika 3-3 Rezultati KNN algoritma

Z algoritmom KNN smo dobili boljše podatke pri velikih *k*-jih, kar pomeni, da so podatki, ki jih uporabljamo zelo šumni, saj je bolje izbrati več primerov, čeprav niso več toliko podobni testnemu, kot pa da izberemo peščico primerov, ki so testnemu skoraj povsem enaki.

3.3 Naivni Bayes

Naivni Bayesov klasifikator predpostavlja pogojno neodvisnost atributov glede na razred [1]. Preizkušali smo ga tako z uporabo Laplacovega ocenjevanja verjetnosti, kot z ocenjevanjem verjetnosti po relativni frekvenci in rezultati so sledeči:

Laplace: CA=0.7349

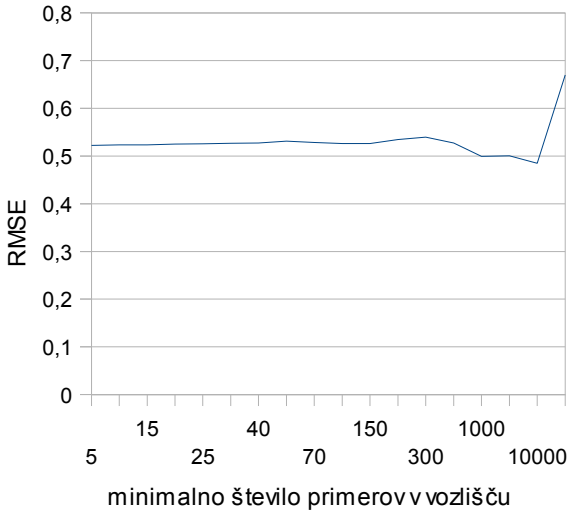
Relativna frekvenca: CA=0.7361

Izkazalo se je, da je uporaba relativne frekvence v našem primeru relevantnejša. Podatke smo poskusili razdeliti šeglede na smer vetra in nato nad vsako podmnožico uporabiti Bayesov klasifikator, kjer smo dobili rezultat CA=0.5726

Pričakovali smo, da bo rezultat po zardelitvi podatkov boljši kot prej, a očitno pri taki delitvi podatke preveč zazdrobimo in se klasifikator zato nauči preveč specifičnega modela, ki se ne obnese dobro niti v primerih z enako smerjo vetra.

3.4 Regresijsko drevo

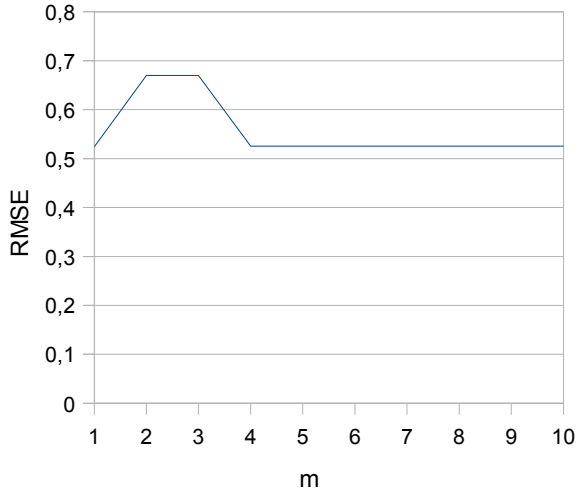
Regresijska drevesa delujejo na podobnem principu kot klasifikacijska drevesa, le da je razred zvezna in ne diskretna spremenljivka. Tako kot pri grajenju klasifikacijskih dreves imamo možnost rezanja drevesa med gradnjo in po njej. Najprej smo preizkusili rezanje drevesa med gradnjo z omejitvijo števila primerov, ki jih lahko vozlišče vsebuje, da ga še nadaljnje cepimo. Rezultati so prikazani na sliki 3-4:



Slika 3-4 RMSE pri rezanju med gradnjo

Namesto klasifikacijske točnosti je v tem primeru uporabljen za ocenjevanje uspešnosti modela koren povprečne kvadratne napake, kar pomeni, da manjša vrednost predstavlja točnejši rezultat. Zanimivo je, da so tu rezultati znatno različni od tistih, ki smo jih dobili pri klasifikacijskem drevesu, saj je napaka manjša pri zelo visokih omejitvah, kar pomeni, da je bolje, če imamo majhno regresijsko drevo, medtem ko smo pri klasifikacijskih drevesih dobili naravnost nasprotne rezultate.

Nato smo preizkusili še rezanje regresijskega drevesa po gradnji z *m*-oceno in dobili rezultate, kot so prikazani na sliki 3-5:



Slika 3-5 RMSE pri rezanju po gradnji

Iz slike je razvidno, da je rezultat enak za katerikoli izbrani *m* nad 3 in napaka je večja kot pri pravi izbiri parametra za rezanje med gradnjo, v nasprotju s klasifikacijskim drevesom.

3.5 Diskusija rezultatov

Če primerjamo klasifikacijsko drevo, KNN in naivni Bayes po klasifikacijski točnosti je opazno, da prva dva dosti točneje klasificirata primere v prave razrede. Razlog za slabše rezultate pri naivnemu Bayesu je prav njegova predpostavka, da so atributi med seboj pogojno neodvisni, kar se je pri vremenski domeni izkazalo za očitno napačno predpostavko.

Regresijsko drevo je težje primerjati z ostalimi metodami, saj uporablja podatke z zveznim razredom. Zato smo spisali algoritem, ki dobljene rezultate diskretizira in primerja s ciljnim razredom in smo na tak način dobili preračun klasifikacijske vrednosti 0.7412. Torej se tudi regresijsko drevo odreže bolje kot naivni Bayes, ga pa vseeno KNN in klasifikacijsko drevo znatno premagata.

Toda klasifikacijske točnosti metod nam ne povejd prav veliko, če jih ne primerjamo z ostalimi podatki, zato smo pripravili lestvico različnih metod za napovedovanje padavin:

KNN	0.7752
Klasifikacijsko drevo	0.7666
Regresijsko drevo	0.7412
Naivni Bayes	0.7361
Napoved prejšnjega stanja	0.6773
Napoved večinskega razreda	0.5198

Kot je iz lestvice razvidno, smo z vsemi izbranimi metodami presegli mejo, ki smo si jo na začetku diplomske naloge začrtali kot prvotni cilj.

Pojavi se vprašanje o tem, koliko so rezultati primerni za praktično uporabo? Kot je bilo razvidno iz testiranja algoritma KNN, so podatki zelo šumni, tako da je res dober model težko narediti. Je pa na drugi strani tudi dejstvo, da boljših rezultatov kot so dobljeni zaenkrat še ni, tako da bi bilo smiselno v primeru razvijanja aplikacije za napovedovanje vremena za kratka obdobja uporabiti ta model, vsaj dokler se ne razvije boljšega pristopa.

4 - Delitev slik na regije

4.1 Motivacija za delitev na regije

Do sedaj smo podatke obravnavali, kot da ni pomembno, kje iz slike podatke vzamemo in da so vsi med seboj primerljivi, kar nam je dovoljevalo, da že iz majhnih zbirk podatkov zgradimo zanesljive modele. Pri tem pa se pojavi vprašanje, če je obnašanje padavin res neodvisno od geografske lege opazovanega področja? Logika narekuje, da geografske značilnosti nekega območja gotovo vplivajo na razvoj in nastanek padavin na območju, tako da bi bilo lahko preizkušanje te teorije razmeroma obetavno.

4.2 Različne metode za delitev

Takoj na začetku dela smo se srečali z vprašanjem, kako pravzaprav sploh deliti slike na regije? Odgovor na vprašanje ni enostaven, saj je možnosti za delitev naravnost ogromno. V procesu razvijanja delitvenega algoritma smo tako prišli do dveh različnih osnovnih pristopov za delitev, iskanje regij neodvisno od ostalih regij in iskanje unikatnih regij.

Pri iskanju regij neodvisno od ostalih regij ni moteče, če se dve ali več regij prekrivajo in če obstajajo celice, ki ne pripadajo nobeni regiji. Ta pristop je dosti bolj enostavno izvedljiv, saj nam ni treba paziti na to, da ne bi prišlo do prekrivanj in lahko vsako regijo razvijamo posebej.

Pri iskanju unikatnih regij pridemo do mnogo več zapletov. Paziti moramo, da vsaka celica pripada natanko eni regiji in najti moramo način za določanje števila in položaja začetnih regij. Druga težava je še odločilnejša od prve, saj ni lahko najti načina za izbiro začetnih regiji, ko o njih ne vemo še nič. Prednost tega pristopa je v tem, da dobimo na koncu sliko razdeljeno v smiselne celote, ki jih znamo razložiti.

Tako za iskanje regiji neodvisno od ostalih, kot za iskanje unikatnih regij smo v toku dela razvili več možnih pristopov in primerjali njihove rezultate:

4.2.1 Skupne točke delitvenih metod

Za vse metode razen zadnje, delitev z vključitvijo podatka o legi celice, velja naslednje:

- metoda je bila izdelana v programskem jeziku Python [3],

- učno množico razdelimo na dve množici, učni del učne množice in testni del učne množice,

- za določanje klasifikacijske točnosti za določeno regijo nad učno množico se:

-v množico X shrani podatke iz učnega dela učne množice, katerih osrednje celice ležijo znotraj obravnavane regije,

-v množico Y shrani podatke iz testnega dela učne množice, katerih osrednje celice ležijo znotraj obravnavane regije,

-zgradi klasifikacijsko drevo na podatkih X in ga testira na podatkih Y,

- za določanje klasifikacijske točnosti za neko regijo nad testno množico se:

-v množico X shrani podatke iz učne množice, katerih osrednje celice ležijo znotraj obravnavane regije,

-v množico Y shrani podatke iz testne množice, katerih osrednje celice ležijo znotraj obravnavane regije,

-zgradi klasifikacijsko drevo na podatkih X in ga testira na podatkih Y,

- za gradnjo in testiranje klasifikacijskega drevesa se uporablja metode iz Orangeve knjižnice.

- drevesa so rezana po gradnji, pri parametru m z vrednostjo 2

4.2.2 Iskanje regij z dodajanjem celic

```
regije=[]
FOR (i=0; i<n; i++){
        dodaj [[randomInt(0,79),randomInt(0,59)]] na konec seznama regije
}
FOR i IN regije{
        WHILE TRUE{
                sosednje=najdiSosednje(i)
                maxPovecanje=0
                maxCelica=NULL
                FOR s IN sosednje{
                        IF povecanje(i,s)>maxPovecanje{
                                maxCelica=s
                                maxPovecanje=povecanje(i,s)
                        }
                }
                IF maxPovecanje>0{
                        dodaj  maxCelica na konec seznama i
                }
```

```
            ELSE{
                BREAK
            }
        }
}
```

n- število regij

randomInt(x,y) – funkcija vrne naključno celo število, ki je manjše ali enako y ter večje ali enako x.

najdiSosednje(x) – funkcija vrne vse celice, ki mejijo na celice regije x in v njej niso vsebovane.

povecanje(x,y) – funkcija vrne vrednost za katero se spremeni klasifikacijska točnost za regijo x, če ji dodamo celico y.

Pri tem pristopu smo iskali regije neodvisno od ostalih regij. Kljub preprostosti pristopa je njegova računska zahtevnost precej visoka, saj mora ob vsakem klicu funkcije *povecanje()* zgraditi in preizkusiti model na celicah, ki mejijo na regijo in že pri nizkem številu regij *n*, to traja nekaj časa. Rezultati modela tudi niso bili dobri, saj so bili rezultati na učni množici zelo dobri, a se je na testni množici izkazalo, da je metoda z dodajanjem ene same celice naenkrat regijo zgolj prilagodila šumu znotraj učne množice na podoben način, kot se klasifikacijsko drevo prekomerno prilagodi šumu, če ga ne režemo ali omejimo.

4.2.3 Delitev cele slike na regije

```
regije=[]
FOR (i=0; i<n; i++){
        dodaj [[randomInt(0,79),randomInt(0,59)]] na konec seznama regije
}
FOR (i=0; i<80; i++){
        FOR (j=0; j<60; j++){
            dodajCelicoNajblizji( i, j )
        }
}
WHILE TRUE{
        mejne=najdiMejneCelice()
        maxCelica=NULL
        maxPovecanje=0
        FOR m IN mejne{
```

```
            IF povecanje(m)>maxPovecanje{
                maxCelica=m
                maxPovecanje=povecanje(m)
            }
        }
        IF (maxPovecanje>0){
            premakni(maxCelica)
        }
        ELSE{
            BREAK
        }
}
```

n – začetno število regij

randomInt(x,y) – funkcija vrne naključno celo število, ki je manjše ali enako y ter večje ali enako x.

najdiMejneCelice() – funkcija poišče vse celice, ki mejijo na drugo regijo kot tisto kateri pripadajo in vrne kot rezultat seznam položaja celic in število regije, kateri pripadajo ter regije, na katero mejijo; če celica meji na več regij se v izhodu funkcije pojavi večkrat, za vsako regijo na katero meji enkrat.

povecanje(x) – funkcija vrne za vhod x, ki vsebuje položaj celice, regijo, kateri pripada in regijo, na katero meji spremembo klasifikacijske točnosti za celoten model, v primeru, ko celico premaknemo iz prvotne regije v mejno regijo.

premakni(x) – funkcija za vhod x, ki vsebuje položaj celice, regijo, kateri pripada in regijo, na katero meji, izvede premik celice iz originalne regije v mejno regijo

Ta postopek je primer iskanja unikatnih regij. Njegova največja težava je, da je izredno časovno zahteven. Tudi, če je bil algoritem omejen na tri obhode glavne zanke, traja poganjanje algoritma več ur in rezultati niso dovolj dobri, da bi lahko sklepali, da se bo nadaljnje poganjanje izplačalo. Kot v primeru z iskanjem regij z dodajanjem celic, se algoritem ponovno pretirano prilagaja šumu v učnih podatkih.

4.2.4 Delitev cele slike na regije z obteževanjem

```
regije=[]
obtezitve=[]
FOR (i=0; i<n; i++){
    dodaj [[randomInt(0,79),randomInt(0,59)]] na konec seznama regije
```

```
        dodaj 1 na konec seznama obtezitve
}
dolociRegijePoObtezitvi()
WHILE TRUE{
        maxObtezitev=0
        maxPovecanje=0
        FOR (i=0; i< velikost seznama obtezitve; i++){
                ca1=klasifikacijskaTocnost()
                obtezitve[i] += 0.1
                dolociRegijePoObtezitvi()
                ca2=klasifikacijskaTocnost()
                IF ca2-ca1 > maxPovecanje{
                        maxObtezitev=i
                        maxPovecanje= ca2-ca1
                }
                obtezitve[i] -= 0.1
        }
        IF (maxPovecanje>0){
                obtezitve[maxObtezitev]+=0.1
        }
        ELSE{
                BREAK
        }
}
```

n – število regij

dolociRegijePoObtezitvi() – funkcija za vsako celico izračuna obteženo oddaljenost od vsake regije po formuli $\sqrt{(x1-x2)^2+(y1-y2)^2}*obtezitev$ in nato doda celico regiji z najmanjšo obteženo oddaljenostjo.

klasifikacijskaTocnost() – funkcija vrne povprečno klasifikacijsko točnost vseh regij, normalizirano po številu primerov v regijah

Tudi ta metoda je primer iskanja unikatnih regij. Je veliko hitrejša od prejšnje metode in tudi odpornejša na šum, a z njo žal nismo uspeli priti do dobrih rezultatov verjetno ravno zaradi tega, ker je premalo prilagodljiva in ker ohranja težavo, ki jo je imela že prejšnja metoda, da je težko zanjo izbrati dobre začetne točke.

4.2.5 Iskanje regij s povečevanjem radija

```
regije=[]
FOR (i=0; i<n; i++){
        dodaj [ randomInt(0,79), randomInt(0,59), zacetniRadij ] na konec seznama
regije
}
FOR i IN regije{
        WHILE TRUE{
                ca1=klasifikacijskaTocnost(i)
                i[2]++
                ca2=klasifikacijskaTocnost(i)
                IF ca2 < ca1{
                        i[2]--
                        BREAK
                }
        }
}
```

n – število regij

zacetniRadij – začetna velikost regij

klasifikacijskaTocnost(x) – funkcija vrne za vhodni parameter x, ki je sestavljen iz središča ter radia regije, klasifikacijsko točnost za drevo zgrajeno in testirano na podatkih iz regije.

Ta metoda išče regije neodvisno od ostalih regij. Je zelo hitra in dovolj prilagodljiva, da smo z njo prišli do sprejemljivih rezultatov. Omejitev je pri njej le to, da zaradi hitrejšega računanja lahko iščemo le regije okrogle oblike, a se izkaže, da je taka poenostavitev vseeno sprejemljiva.

4.2.6 Iskanje regij z vključevanjem podatka o legi celice

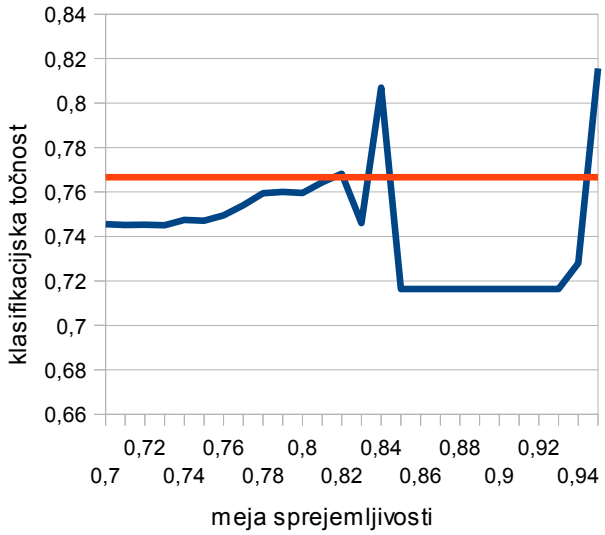
Metoda išče unikatne regije. Deluje tako, da meta atributa x ter y, ki sta do sedaj služila le za združevanje po regijah, spremenimo v prava atributa, ki bosta vključena pri napovedovanju padavin za celico. Na tak način klasifikacijsko drevo ugotovi morebitne povezave med lego celice in padavinami v njenem območju. Metoda je daleč najhitrejša od vseh predstavljenih, saj komaj poveča časovno zahtevnost klasifikacije, ravno zato ker gre le

za dodajanje dveh atributov skupini že obstoječih. Edini problem z metodo je, da tak pristop ni spremenil klasifikacijske točnosti modela, kar pomeni, da nova atributa nista nudila dovolj novih informacij, da bi jih drevo izbralo. Zaradi tega je metoda popolnoma neuporabna, saj ne moremo dobiti od nje nobenih podatkov o regijah.

4.3 Rezultati delitve slik na regije

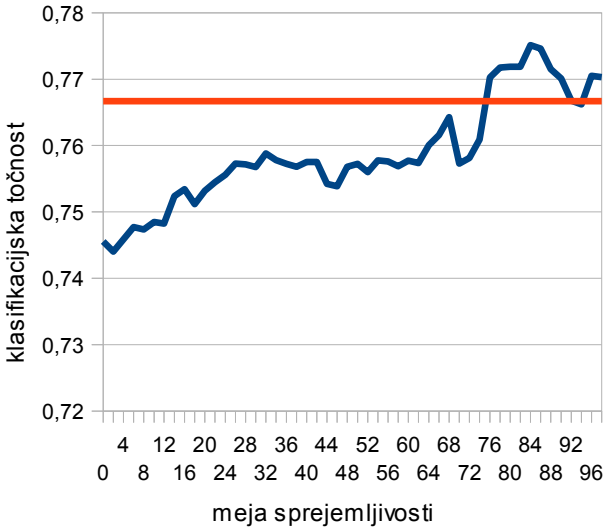
Med vsemi metodami, ki smo jih imeli na voljo, smo izbrali metodo s povečevanjem radija, ker je hitra in nam je zagotovila dobre rezultate. Algoritem smo za ekstenzivno testiranje tudi malo dopolnili, da so rezultati natančnejši. Namesto izbiranja naključnih začetnih celic za regije, smo v seznam regij dodali vsako celico na sliki. Poleg tega smo algoritmu dodali dvostopenjski pogled vnaprej, kar pomeni, da smo v glavni zanki pogledali še, kaj se zgodi s klasifikacijsko točnostjo, če radij povečamo za 2 ter za 3 in če je bila katera koli od povečav boljša od trenutnega stanja, smo radij povečali in nadaljevali.

Za ugotavljanje klasifikacijske točnosti dveh regij smo regije naprej sortirali v padajočem vrstnem redu po dveh kriterijih, in sicer po klasifikacijski točnosti ter razliki klasifikacijske točnosti regije in klasifikacijske točnosti modela pomnoženi s številom učnih primerov v regiji. Nato smo vsako celico iz testne množice klasificirali glede na napoved prvega modela v urejenem seznamu, ki jo vsebuje. To metodo smo preizkusili z nastavljanjem meje za sprejemljivost regije po klasifikacijski točnosti in obteženi razliki točnosti. Rezultati po klasifikacijski točnosti so prikazani v obliki grafa na sliki 4-1.



Slika 4-1 Klasifikacijska točnost regij na testni množici po klasifikacijski točnosti na regijah

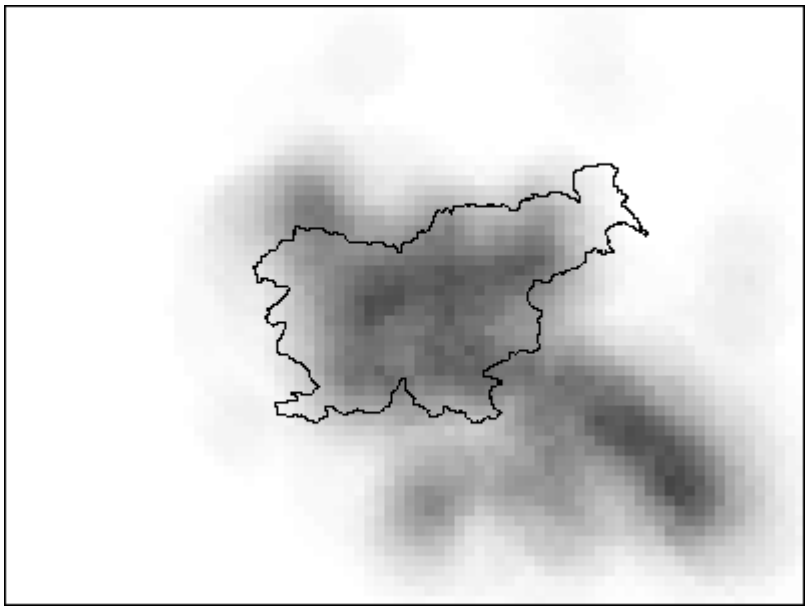
Po obteženi razliki točnosti dobimo pa graf na sliki 4-2:



Slika 4-2 Klasifikacijska točnost regij na testni množici po obteženi razliki točnosti

Na obeh slikah je modra črta klasifikacijska točnost regij nad testno množico, rdeča črta pa klasifikacijska točnost modela zgrajenega na učni množici nad testno množico. Iz grafov je razvidno, da je ocena sprejemljivosti regije na podlagi obtežene razlike točnosti boljša kot ocena na podlagi klasifikacijske točnosti regije nad učno množico. Sicer ima graf ocene na podlagi klasifikacijske točnosti dva presenetljivo visoka vrha, ki sta boljša kot pri obteženi razliki točnosti, a nastopita pri tako visoki klasifikacijski točnosti nad učno množico, da je komaj katera regija še sprejemljiva, zato je varno sklepati, da so vrhovi prej rezultat naključnega ujemanja, kot pa dejanske skladnosti med regijami. Da bi se takim napakam izognili, bi morali uporabiti veliko večjo zbirko podatkov kot je pričujoča, a bi to povečalo že tako visoko časovno zahtevnost izvajanja.

Pomembno je, da smo z uporabo omejitve z obteženo razliko točnosti uspeli zagotoviti klasifikacijsko točnost, ki je višja kot točnost osnovnega modela, kar pomeni, da je metoda delitve na regije s povečevanjem radija res učinkovita. Njen največji problem je, da nam ne vrne slike razdeljene na regije, ampak množico krogov podanih z njihovimi radiji in centri ter za vsako regijo, ki jo krog pokriva, tudi klasifikacijsko točnost na testnem delu učnih podatkov. Če omejimo regije na tiste s klasifikacijsko točnostjo nad 0.72 ter za vsako od njih na sliki potemnimo območje, ki ga pokriva, dobimo sliko 4-3:



Slika 4-3 Prikaz pogostosti odobrenih regij pri omejitvi klasifikacijske točnosti na regijah na 0.72

To žal ni tako jasen pregled, kot bi ga želeli, a je kljub temu opazno, da je napovedovanje za nekatere dele očitno bolj točno, če ga opravljamo ločeno, kot pa za ostale. To bi lahko razumeli kot znak, da geografske značilnosti tega območja vplivajo na padavine do te mere, da pridemo do boljše napovedi, če za regijo uporabimo le podatke iz njenega območja, medtem ko drugje takega močnega vpliva ni in zato pri napovedovanju več pridobimo, če uporabimo večjo množico podatkov, kot pa če se omejimo le na podatke iz območja.

Zaključne ugotovitve

Tekom izdelave diplomske naloge smo prikazali praktičen primer združevanja dveh različnih vrst podatkov, pokazali smo namreč, da je mogoče z metodami atributnega strojnega učenja priti do sprejemljivih padavinskih napovedi in da je mogoče upoštevati lokalne vremenske značilnosti pokrajine tudi pri enakomerni delitvi radarskih slik na celice.

Za združevanje podatkov bi lahko ubrali tudi kakšen drugačen način, kot denimo delitev slik na vnaprej določene regije, kot so na primer Primorje, Julijske Alpe, Posočje in ostale geografske regije, a bi to spremenilo naš pristop k nalogi in tudi rezultati bi v takem primeru bili veliko manj specifični, saj lahko naše slike napovedujemo za vsako 5X5 točk veliko celico.

Cilj, ki smo si ga v uvodu zastavili, je bil preseči klasifikacijsko točnost, ki jo dobimo, če uporabljamo za napovedovanje kar prejšnje stanje - kar pomeni, da napovemo, da ne bo razlik med sliko ob času t in sliko ob času t+1. To napoved smo izboljšali z uporabo klasifikacijskih dreves in metode KNN, a smo nato skušali dobljene rezultate še dodatno izboljšati s širjenjem okolice, upoštevanjem sinoptičnih meritev in uporabo algoritmov za delitev slik na regije. Izdelali smo več metod, a smo se na koncu odločili za uporabo metode s povečevanjem radijev, saj je med vsemi metodami imela najboljše rezultate in sprejemljivo časovno zahtevnost. Na tem področju bi bilo mogoče doseči veliko izboljšav, saj smo pri metodi, ki je bila v tej nalogi uporabljena, žrtvovali natančnost za hitrost, saj so bile vse ostale metode časovno tako zahtevne, da je bila njihova uporaba praktično neizvedljiva. Že metoda s povečevanjem radijev se je poganjala dve uri nad zbirko okoli 15 tisoč podatkov, kar glede na domeno ni veliko podatkov.

Glede na to, da smo z metodo povečevanja radijev dosegli boljše rezultate kot z osnovnim modelom, lahko zaključimo, da lokalne vremenske značilnosti v Sloveniji res vplivajo na nastanek in razvoj padavin na njenem ozemlju. To je bil tudi en izmed ciljev zastavljenih tekom dela na tej diplomski nalogi in potrdimo lahko, da je bil uspešno dosežen.

Metode, izdelane v diplomski nalogi, so uporabne na vsaki bazi vremenskih podatkov, če jih prej spremenimo v sprejemljivo obliko, gre pa tudi omeniti, da sam model, ki smo ga naredili, ni vsesplošno uporaben, saj so vsi podatki, ki smo jih uporabljali, iz poletnega časa in za podatke iz drugih letnih časov napovedi ne bi bile tako natančne.

Z isto težavo, kot smo se ukvarjali v tej diplomski nalogi, se je pred nami ukvarjal že Matic Standeker v svoji diplomski nalogi z naslovom *Napovedovanje diplomskih stanj z metodo najbližjih sosedov* [4]. Njegov pristop je bil znatno drugačen od našega, saj se je ukvarjal z napovedovanjem celotnih slik naenkrat in ne posameznih celic. Poleg tega je napovedoval vreme kar za dve uri vnaprej in ne samo za 10 minut, tako kot smo se v tej nalogi odločili. Zato tudi naših in njegovih rezultatov ne moremo neposredno primerjati in se je treba primerjave lotiti na drugačen način.

Standeker je v svoji diplomski nalogi navedel, da je z metodo, ki jo je razvil, dosegel povprečno napako okoli 3%. To pomeni, da je njegova metoda imela klasifikacijsko točnost 0.97. A zgolj ta podatek je lahko zavajajoč, saj ga nimamo s čim primerjati. Kot smo napisali v drugem poglavju, smo izbrali za podatkovno bazo le skupino slik, kjer so razvidne

padavine. V celotni podatkovni bazi slik, ki smo jo uporabili pri delu in je enaka tisti, ki jo je uporabil gospod Standeker, je kar nekaj slik, kjer ni nobenih padavin. Izkaže se celo, da je takih slik velika večina. Zato smo se sprehodili po vseh slikah in prešteli vse točke, v katerih se pojavijo padavine in tiste, kjer padavin ni in izkaže se, da je 97.3% točk brez padavin. Torej napoved večinskega razreda izboljša Standekerjevo napoved za 0.3%, oz 0.75%, če je štel kot pravilno klasificirane točke, ki so na vseh slikah črne. Iz tega lahko zaključimo, da je naš model uspešnejši in čeprav ga nismo preizkusili na napovedovanju za dve uri vnaprej, lahko z gotovostjo sklepamo, da bi dosegel boljše rezultate kakor Standekerjev.

Literatura

[1] I. Kononenko, M. Kukar, *Machine learning and data mining*, Horwood publishing limited, 2007

[2] Orange, dostopno na
http://www.ailab.si/orange

[3] Python programski jezik, dostopno na
http://www.python.org

[4] M. Standeker, *Napovedovanje diplomskih stanj z metodo najbližjih sosedov*, Diplomska naloga na univerzitetnem študiju računalništva in informatike, Ljubljana, 2007