UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Minja Zorc

UČENJE OPTIMALNE ODLOČITVE S KLASIFIKACIJSKIMI DREVESI

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

Mentor: doc. dr. Janez Demšar

Ljubljana, 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: 01553/2009
Datum: 15.03.2009

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

Kandidat: **MINJA ZORC**

Naslov: **UČENJE OPTIMALNE ODLOČITVE S KLASIFIKACIJSKIMI DREVESI**
LEARNING OPTIMAL DECISIONS WITH CLASSIFICATION TREES

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Eno najpomembnejših področij strojnega učenja je učenje klasifikacijskih in regresijskih modelov. Te sestavimo iz učnih primerov, njihov namen pa je napovedovanje razreda oziroma zvezne vrednosti za nove primere. V praksi pa bi pogosto potrebovali model, s katerim bi za dani novi primer določili optimalno odločitev, kot npr. najboljšo terapijo pri določeni vrsti bolezni za določenega pacienta.

Spremenite algoritem za gradnjo klasifikacijskih dreves tako, da bo sestavljal drevesa, ki bodo namesto razreda oziroma zvezne vrednosti napovedovali odločitev, ki pripelje do čim boljšega razreda oziroma vrednosti. Določite primeren način za določanje kvalitete tako dobljenih modelov in preskusite delovanje algoritma na umetno konstruiranih in resničnih podatkih.

Mentor:

doc. dr. Janez Demšar

Fakulteta za računalništvo in informatiko
2
UNIVERZA V LJUBLJANI

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!

Izjava o avtorstvu diplomskega dela

Spodaj podpisana **Minja Zorc**, z vpisno številko **63020183**, sem avtorica diplomskega dela z naslovom: **Učenje optimalne odločitve s klasifikacijskimi drevesi**.

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom doc. dr. Janeza Demšarja,
- 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 3.6.2009 Podpis avtorice: _____

Zahvala

Na tem mestu bi se najprej zahvalila družini in prijateljem, ki so mi pomagali in stali ob strani v času celotnega študija. Zahvaljujem mentorju, doc. dr. Janezu Demšarju, ki mi je predstavil zanimiv problem ter mi skozi proces izdelovanja diplomskega dela pomagal z nasveti in pristopi k reševanju.

Družini in prijateljem

Kazalo

Seznam uporabljenih kratic, simbolov in izrazov

kratica	pomen
KL	Kullback-Leibler
CA	Klasifikacijska točnost (classification accuracy)
WA	Najmanjša sprejemljiva ocena (worst acceptable)
f_p	Pričakovana frekvenca dogodkov
f_o	Opazovana frekvenca dogodkov
GUI	Grafični uporabniški vmesnik (graphical user interface)

Tabela 0.1: *Tabela uporabljenih kratic*

Povzetek

Eno najpomembnejših področij strojnega učenja je učenje klasifikacijskih in regresijskih modelov. Klasifikacijske in regresijske modele sestavimo iz učnih primerov, njihov namen pa je napovedovanje razreda oziroma zvezne vrednosti za nove primere. V praksi bi pogosto potrebovali model, s katerim bi za dani novi primer določili optimalno odločitev, kot na primer najboljšo terapijo pri določeni vrsti bolezni za določenega pacienta.

Glavni cilj diplomske naloge je spremeniti algoritem za gradnjo klasifikacijskih dreves tako, da bo sestavljal drevesa, ki bodo namesto razreda oz. zvezne vrednosti napovedovali odločitev, ki pripelje do čimboljšega razreda oz. vrednosti, in določiti primeren način za določanje kvalitete tako dobljenih modelov ter preskusiti delovanje algoritma na umetno konstruiranih in resničnih podatkih.

Ključne besede:

strojno učenje, klasifikacijsko drevo, optimalna odločitev, mera za izbiro atributa

Abstract

Learning classification and regression models is one of the most important subfields of machine learning. Classification and regression models are constructed from learning set and used to classify new examples. In practice, we would often need model, which proposes optimal decision, for instance the best therapy for a certain type of illness for a particular patient.

The main aim of the diploma thesis is to adapt the algorithm for the construction of classification trees to construct trees that would not predict the outcome but rather the decision leading to the desired outcome. Besides that, we had to develop the methods for measuring the quality of such models, and use it to test them on synthetic and real-world data sets.

Key words:

machine learning, classification tree, attribute selection measure, optimal therapy

Poglavje 1

Uvod

Strojno učenje je eno najpomembnejših in v zadnjih letih morda eno najdejavnejših področij v okviru umetne inteligence. Ukvarja se z razvojem tehnik za odkrivanje modelov iz podatkov. S pomočjo orodij strojnega učenja lahko iz podatkov izluščimo skrito znanje, ki ga lahko nato uporabimo. S pomočjo znanja in izkušenj iz primerov, ki so opisani z množico atributov, novim primerom napovedujemo nepoznane atribute. Eno najpomembnejših področij strojnega učenja je učenje klasifikacijskih in regresijskih modelov. Te sestavimo iz učnih primerov, njihov namen pa je napovedovanje razreda oziroma zvezne vrednosti za nove primere. Če bi na podlagi podatkov o že zdravljenih pacientih zgradili klasifikacijski model, bi lahko za novega pacienta napovedali izid zdravljenja. V praksi pa bi razen napovedovanja razreda pogosto potrebovali model, s katerim bi za dani novi primer določili optimalno odločitev, kot na primer najboljšo terapijo pri določeni vrsti bolezni za določenega pacienta.

Med najbolj uveljavljene pristope klasifikacije sodijo odločitvena (klasifikacijska oziroma regresijska) drevesa, odločitvena pravila, metoda *k* najbližjih sosedov, nevronske mreže, naivni Bayesov klasifikator in funkcijska dekompozicija. Namen tega diplomskega dela je razviti in predstaviti tako spremenjen algoritem za gradnjo klasifikacijskih dreves, da sestavlja drevesa, ki namesto razreda napovedujejo optimalno odločitev, ki pripelje do ciljnega razreda. V listih tako zgrajenih dreves niso napovedani razredi, temveč priporočene optimalne odločitve, ki naj bi pripomogle k temu, da bodo primeri v ciljnem razredu. V listih modela za klasifikacijo pacientov bi bile priporočene terapije, ki bi pripomogle k temu, da pacient ozdravi. Pri gradnji takšnega modela nas zanima, ali obstajajo podskupine primerov, v katerih odločitev (terapija) povzroča heterogene učinke (razred, izid), in če je tako, za katere podskupine primerov gre in katere odločitve so zanje optimalne.

Temeljne raziskave klasifikacijskih dreves sta postavila Quinlan in Breiman s sodelavci, v zadnjih desetletjih pa je bilo razvitih veliko različic osnovnega algoritma. Med najbolj znanimi so ID3, C4, Cart in Assistant. Različice dopolnjujejo osnovni algoritem tako, da dovoljujejo tudi delo z zveznimi atributi, manjkajočimi vrednostmi, šumom itd. Pri ocenjevanju kvalitete atributa uporabljajo različne mere, kot so informacijski prispevek, gini-indeks, gini′-index, RelieF, gain-ratio, J-mera, R-mera. V diplomskem delu smo dodali osnovnemu algoritmu za gradnjo klasifikacijskih dreves dve spremembi. Na drevesu smo izvedli transformacijo, za oceno kvalitete atributa pa smo uporabili statistični test, ki meri razliko v porazdelitvi vzorcev. V nalogi smo

preizkusili dva statistična testa, Hi-kvadrat in Kullback-Leibler, ter ju primerjali med seboj. Metodo smo preizkusili na umetno konstruiranih in resničnih podatkih.

Najprej smo na umetno konstruiranih podatkih brez šuma preizkusili delovanje metode. Preverili smo, ali spremenjeni algoritem za gradnjo klasifikacijskih dreves ustvari model, ki predlaga pravilne odločitve (takšne, ki klasifikacijski primer uvrstijo v ciljni razred). Na različno konstruiranih umetnih podatkih smo preverili tudi napovedano točnost klasifikatorja, robustnost metode, ki označuje odpornost metode na različne nepopolnosti v učnih primerih, kot je na primer šum, narisali smo krivuljo učenja, ki kaže povezavo med številom učnih primerov in napovedno točnostjo modela ter z risanjem klasifikacijskih dreves preizkusili razumljivost modela, ki je še posebej pomembna pri modeliranju v medicini. Klasifikacijsko točnost modela smo skušali povečati z različnimi načini rezanja drevesa.

Algoritem smo preizkusili na dveh naborih resničnih medicinskih podatkov. Za določanje natančnosti klasifikacije smo uporabili metodo desetkratnega prečnega preverjanja. Uspeh iskanja optimalne terapije klasifikacijskega drevesa smo merili tako, da smo merili delež pravilnih odločitev (ozdravljenih pacientov) modela samo takrat, kadar je model predlagal enako terapijo, kot jo je pacientu izbral zdravnik, saj ne moremo vedeti, kakšen bi bil izid zdravljenja, če bi se zdravnik odločil drugače.

Diplomsko delo je sestavljeno iz petih poglavij. V prvem, uvodnem poglavju, smo predstavili tematiko, ki smo jo v nadaljevanju obravnavali, ter opisali problem in pristop k rešitvi. Drugo poglavje smo namenili kratkemu pregledu že obstoječih pristopov k obravnavani problematiki, in sicer klasifikacijskim drevesom, statističnim testom, algoritmu, opisanem v članku »Subgroup Analysis via Recursive Partitioning«, in odločitvenim nomogramom. V tretjem poglavju smo opisali metode in orodja, s pomočjo katerih smo razvili in preizkusili algoritem za iskanje optimalne odločitve. V četrtem poglavju smo opisali preizkuse delovanja razvitega algoritma na umetno konstruiranih in resničnih podatkih. V zadnjem poglavju so sklepne ugotovitve ter primerjava razvitih in preizkušenih različic algoritma.

Poglavje 2

Pregled obstoječih pristopov

Ideja o iskanju optimalne odločitve na podlagi množice že znanih podatkov ni nova. Obstajajo uveljavljeni pristopi k reševanju tega problema. V tem poglavju smo opisali že znane pristope s področja statistike in strojnega učenja ter izpostavili razliko med njimi in pristopom, razvitem v pričujočem delu.

2.1 Klasifikacijska drevesa

Klasifikacijsko drevo je ena najbolj razširjenih metod strojnega učenja. Raziskala sta ga Quinlan (1986) in Breiman (1984). V zadnjih desetletjih je bilo razvitih še veliko različic osnovnega algoritma (ID3, C4, Cart, Assistant), ki dopolnjujejo osnovni algoritem tako, da dovoljujejo tudi delo z zveznimi atributi, manjkajočimi vrednostmi, šumom itd.

Klasifikacijsko drevo [4] predstavlja klasifikacijsko funkcijo, ki je hkrati simbolični opis in povzetek zakonitosti v dani problemski domeni. Sestavljeno je iz vozlišč, vej in listov. Pot v drevesu od korena do lista ustreza enemu odločitvenemu pravilu. Pri gradnji klasifikacijskih dreves se uporablja atributna predstavitev učnih primerov. Vsak učni primer je opisan z vektorjem vrednosti atributov. Notranja vozlišča drevesa predstavljajo atribute, veje predstavljajo podmnožice vrednosti atributov, listi pa ustrezajo razredom. Osnovni algoritem učenja odločitvenih dreves razdeli podano množico primerov na podmnožice glede na vrednosti izbranega najboljšega atributa. Postopek se rekurzivno ponavlja na podmnožicah primerov. Algoritem se ustavi, ko je izpolnjen ustavitveni pogoj, na primer, če je učna množica dovolj čista (če so vsi primeri člani istega razreda), če je premalo učnih primerov za zanesljivo nadaljevanje gradnje drevesa ali pa, če zmanjka atributov. Takrat označimo vozlišče kot list in mu določimo razred. Ključnega pomena pri gradnji klasifikacijskega drevesa je izbira najboljšega atributa. Za izbiro atributa se najpogosteje uporabljajo mere: informacijski prispevek, gini-indeks, gini'-index, RelieF, gain-ratio, J-mera, R- mera. Zgrajeno drevo lahko uporabimo za klasifikacijo novih primerov. Od korena potujemo po ustreznih vejah do lista. List vsebuje informacijo o številu učnih primerov iz posameznih razredov. Na podlagi frekvenc učnih primerov ocenimo verjetnostno porazdelitev razredov.

Klasifikacijsko drevo je zanimivo predvsem za strokovnjake iz dane problemske domene, saj je iz hierarhične strukture drevesa možno razbrati zakonitosti domene. Iz strukture drevesa, ki ga algoritem zgradi, je včasih možno razbrati, katera odločitev je najbolj primerna za določeno podmnožico primerov, vendar osnovni algoritem gradnje klasifikacijskega drevesa tega eksplicitno ne poda.

2.2 Statistični testi

S statističnimi testi, kot so na primer Hi-kvadrat, Kullback-Leibler, Kruskal Wallis, t-test [8] in podobni, lahko preizkušamo domneve. Denimo, da želimo raziskati, ali je učinek zdravljenja skladen med tremi starostnimi skupinami: mladi, srednjih let, starejši. Do ocene pridemo s pomočjo testa interakcije med terapijo in starostjo. Če je interakcija statistično signifikantna, s testom ugotovimo, da različnim starostnim skupinam bolnikov ustrezajo različne terapije. Podskupine, kakor tudi število podskupin, ki jih je treba preučiti, določi raziskovalec pred analizo, zaradi česar je lahko analiza subjektiven proces. Celo za strokovnjake na področju raziskovanja je določanje, katere posebne podskupine je treba uporabiti v analizi podskupin, zahtevno, vprašljivo in subjektivno (lahko pripelje do dvomljivih rezultatov in manipulacij). Pri statističnih testih je treba vnaprej predvideti, za kateri atribut (v opisanem primeru je to atribut, ki opisuje starost), je treba preizkusiti domnevo. To je njihova slabost v primerjavi z metodo, razvito v tem diplomskem delu. Poleg tega je lahko učinek zdravljenja odvisen od kombinacije več atributov, to pa je lažje ugotoviti z gradnjo drevesa, kot s statističnimi testi.

2.3 Odkrivanje skupin z rekurzivno delitvijo

V času, ko je bila ideja pričujočega diplomskega dela o učenju optimalne odločitve že implementirana in v preizkusu, je v reviji *Journal of Machine Learning Research* izšel članek z naslovom *Subgroup Analysis via Recursive Partitioning* [6], ki opisuje postopek, podoben temu, ki smo ga predstavili v tem diplomskem delu.

Avtorji članka so razvili algoritem, ki na podlagi učnih podatkov zgradi model, ki išče optimalno odločitev za domene z zveznim razredom. Model za iskanje optimalne odločitve zgradijo v petih korakih. Najprej zgradijo klasifikacijsko drevo, ga porežejo, določijo najboljšo velikost drevesa, združijo nekatera notranja vozlišča in nazadnje uporabijo ocenjevanje pomembnosti atributa, da izluščijo pomembne odvisnosti med atributi in izbiro odločitve.

Algoritem zgradi klasifikacijsko drevo tako, da za izbiro najboljšega atributa uporabi kriterij, ki meri odvisnost med vzorci (t-test). Potem drevo režejo s podobnim postopkom, kot sta ga predstavila LeBlanc in Crowley leta 1993 [5]. Vsako vozlišče v drevesu je koren poddrevesa. Vsakemu vozlišču ocenijo kompleksnost interakcij v poddrevesu (LeBlanc in Crowley sta merila kompleksnost razdelitve, *split complexity*). Najslabše ocenjenemu vozlišču odrežejo veje. Tako

nastalemu drevesu zopet ocenijo vsa vozlišča in najslabšemu vozlišču odrežejo veje. Postopek ponavljajo, dokler ne odrežejo drevesu vseh vej. Eno izmed vseh dreves, ki so nastajala v opisanem postopku, je rezultat tega rezanja. Katero drevo je pravo, določijo tako, da poiščejo najboljšo velikost drevesa, ki jo določijo z oceno kompleksnosti interakcij drevesa. Zatem določijo število podskupin, pri katerih ima terapija različen učinek (to pomeni, da določijo število listov drevesa). Za vsak par listov izračunajo vrednost t-testa in združijo pare listov v en list, če t-test pokaže majhno heterogenost vzorca. Temu postopku sledi ocenjevanje pomembnosti atributov z metodo naključnih gozdov.

Metodo so preizkusili na umetnih podatkih tako, da so sestavili šest modelov, po katerih so konstruirali umetne podatke. Vsak nabor podatkov je vseboval zvezen razred Y, diskreten razred, ki opisuje terapijo, in štiri atribute X1–X4, ki so zavzemali diskretne vrednosti. Šest modelov je bilo sestavljenih tako, da so na njihovi podlagi konstruirali umetne nabore podatkov z različnimi odvisnostmi med vrednostmi atributov, izbiro terapije (odločitve) in vrednostmi razredov.

Algoritem so preizkusili tudi na resničnih podatkih, ki so jih dobili v CPS bazi, ki vsebuje podatke o približno 60.000 ameriških gospodinjstvih. Iz teh podatkov so sestavili nabor podatkov o 16.602 posameznikih, ki jih opisuje 16 atributov (starost, izobrazba, rasa, poklic,…). Atribut, ki predstavlja odločitev (terapijo), je spol, razred pa je zaslužek. Z algoritmom učenja optimalne odločitve so torej ugotovili, v kakšnih podskupinah družbe so ženske slabše plačane od moških.

2.4 Odločitveni nomogrami

Demšar in sodelavci [2] so predstavili tehniko gradnje modelov, ki lahko pomagajo pri izbiri optimalne terapije zdravljenja bolnikov. Modeli temeljijo na naivnem Bayesovem klasifikatorju in so vizualizirani z nomogrami.

Podobno kot model, ki smo ga raziskali v tem diplomskem delu, ima odločitveni nomogram preprosto matematično ozadje, je enostaven za razumevanje in uporabo. Iz odločitvenega nomograma je razvidno, kateri atributi so pomembni pri izbiri terapije bolnika. Uporaben je pri izbiri optimalne terapije določenega pacienta. Tudi pri tem modelu je glavna pomanjkljivost ta, da njegova gradnja temelji na nezanesljivih ocenah verjetnosti, saj zdravniki običajno izberejo boljša zdravljenja, torej predpisano zdravljenje ni neodvisno od atributov.

Metoda z odločitvenimi nomogrami ima prednost pred iskanjem optimalne odločitve z gradnjo klasifikacijskega drevesa, saj je iz takšnega modela lažje razbrati argumente za in proti izbiri posamezne terapije za določenega pacienta. Poleg tega je omenjeni model kot celota veliko bolj pregleden od klasifikacijskega drevesa, ki postane, ob prikazu vseh atributov, razvejano in nepregledno. Rezanje drevesa poveča preglednost, lahko pa se zgubijo atributi in s tem pregled nad celotno domeno. Struktura drevesa je nestabilna, bistveno jo spreminjajo že manjše spremembe parametrov.

Poglavje 3

Metode in orodja

Metode strojnega učenja so namenjene odkrivanju modelov iz podatkov. Iz množice znanih primerov, ki so opisani z atributi, lahko napovedujemo novim primerom nepoznane atribute. Takšnemu matematičnemu modelu pravimo klasifikator.

V pričujočem delu smo raziskali delovanje novega klasifikacijskega algoritma. Okvirno priporočilo za postopek preizkušanja novega klasifikacijskega algoritma [7] predlaga primerjavo novega algoritma z drugimi, že obstoječimi algoritmi, vključno s tistim, ki je novemu najbolj podoben. Za ponazoritev delovanja algoritma predlaga izbor podatkov, ki pokažejo odlike novega algoritma. Točnost klasifikacije je treba izmeriti s prečnim preverjanjem, za primerjavo klasifikacijske točnosti pa je treba uporabiti binomski ali McNemarov test.

Algoritem iskanja optimalne odločitve zgradi klasifikacijski model, ki pa se od že obstoječih modelov razlikuje po tem, da ne klasificira nove primere v razred. Algoritem za nove primere predlaga optimalno odločitev (terapijo), ki verjetneje privede do uvrstitve primera v ciljni razred. Predlog odločitve lahko obravnavamo kot klasifikacijo, ne moremo pa ga primerjati s klasifikacijo ostalih algoritmov. V drugem poglavju sta opisana algoritma, ki podobno, kot algoritem, ki smo ga razvili v tem delu, iščeta novim primerom optimalno odločitev. Čeprav se v delu posvečamo preizkušanju novega klasifikacijskega algoritma, delo ne temelji na primerjavi z že obstoječima podobnima algoritmoma, ampak na raziskavi osnovnih lastnosti algoritma.

V tem poglavju smo predstavili osnovne metode in orodje strojnega učenja, s pomočjo katerih smo razvili in preizkusili algoritem učenja optimalne odločitve.

3.1 Implementacija algoritma

Python [9] je visokonivojski objektno usmerjen skriptni jezik, ki ga je sestavil Guido van Rossum leta 1991. Ponaša se z majhno, čisto in berljivo sintakso, jedrnato semantiko in obsežno standardno knjižnico. V programe, napisane v Pythonu, lahko vključimo knjižnice iz

programskega paketa Orange [3], ki vključujejo različne algoritme za strojno učenje, ter podatkovno rudarjenje in rutine za urejanje in predpripravo vhodnih podatkov. Orange je dostopen prek skript, napisanih v Pythonu, ali prek grafičnih gradnikov (angl. *widgets*). Algoritem za učenje optimalne odločitve je napisan v Pythonu za Orange.

3.2 Mera kvalitete atributa

Ključnega pomena pri gradnji klasifikacijskih dreves s standardnim algoritmom je izbira najboljšega atributa za postavitev vsakega vozlišča. Pred gradnjo vsakega vozlišča je treba izbrati najboljši atribut, ki najbolje razdeli podatke na čimbolj čista vozlišča. Kriterij za izbiro atributa je običajno količina informacije, ki jo atribut vsebuje (v uporabi je enačba za entropijo iz informacijske teorije). Pri učenju optimalne odločitve v listih drevesa ne želimo čistih skupin. Iščemo podskupine, pri katerih odločitev o izbiri terapije vidno vpliva na učinek zdravljenja. Za mero kakovosti atributa uporabimo statistični test, ki testira različnost porazdelitev znotraj skupine. Za domene z diskretnim razredom lahko uporabimo teste, kot sta Hi-kvadrat in test Kullback-Leibler.

Statistika Hi kvadrat (χ2)

Statistiko Hi-kvadrat uporabimo v primerih, ko imamo kvalitativne podatke ali, če porazdelitev teh podatkov odstopa od normalne. Hi-kvadrat računa samo s frekvencami. Hi-kvadrat je zelo praktičen test, kadar želimo ugotoviti, ali opažene frekvence odstopajo od pričakovanih frekvenc. S testom lahko ugotovimo tudi, ali obstaja povezanost med dvema spremenljivkama in verjetnost njune povezanosti. Število stopenj prostosti je definirano kot število neodvisnih spremenljivk, vključenih v izračun χ^2. Statistiko χ^2 računamo po formuli (1).

$$\chi^2 = \frac{(f_o - f_p)^2}{f_p} \qquad (1)$$

Kullback-Leiblerjeva divergenca

Kullback-Leiblerjeva divergenca meri podobnost med dvema diskretnima distribucijama P in Q. V našem primeru ga uporabljamo za primerjavo pričakovanih distribucij razredov pri različnih odločitvah. Njegova omejitev v primerjavi s hi-kvadrat je, da je omejen le na binarne odločitve in ne deluje za domene, pri katerih predstavlja terapijo atribut, ki zavzema več kot dve vrednosti. Kullback-Leiblerjevo divergenco računamo po formuli (2).

$$KL = \sum_i P(i) \log \frac{P(i)}{Q(i)} + Q(i) \log \frac{Q(i)}{P(i)} \qquad (2)$$

3.3 Transformacija drevesa

Standardni algoritem za gradnjo klasifikacijskih dreves je tako spremenjen, da za gradnjo vozlišča ne izbere atributa, ki opisuje odločitev. Zaradi lažje razlage ta atribut poimenujmo *z*. Algoritem atributa *z* med gradnjo drevesa pravzaprav ne ocenjuje. Ko za gradnjo vozlišč zmanjka vseh ostalih atributov, algoritem zgradi vozlišča z atributom *z*. Ta vozlišča imajo toliko vej, kolikor različnih vrednosti zavzema atribut *z*. Vsaka veja predstavlja eno od teh vrednosti in vodi v list. V listu je vrednost razreda, kateremu pripada večina primerov v listu. Sedaj je drevo zgrajeno in temu sledi transformacija. Najprej odstranimo liste. Novi listi so sedaj vozlišča, zgrajena iz atributa *z*. V vsakemu novemu listu priredimo vrednost priporočene odločitve, ki je pravzaprav vrednost atributa *z*, ki je pripeljala v list, v katerem ima ciljni razred največjo verjetnost.

Kadar klasificiramo nov primer s pomočjo naučenega drevesa, se pomikamo po vejah drevesa glede na vrednosti atributov novega primera. Lahko se zgodi, da pridemo do vozlišča, kjer ima nov primer vrednost atributa, ki je v naučenem drevesu ni. V tem primeru nam manjka veja. Do tega pride, kadar med učnimi podatki ni bilo primera s takšnimi vrednostmi atributa, kot jih ima novi primer, ki ga klasificiramo. Takšnemu primeru določimo optimalno odločitev tako, da izberemo tisto odločitev (terapijo), ki je najbolj pogosta v poddrevesu, v katerem se primer nahaja.

3.4 Klasifikacijsko drevo

3.4.1 Kontingenčna matrika

Algoritem učenja optimalne odločitve pred gradnjo novega vozlišča oceni, kateri atribut je najboljši. Za vsak atribut zgradi kontingenčno matriko.

$$\begin{array}{c} \textit{vrednost} \quad \textit{atributa} \\ 0 \qquad 1 \qquad 2 \end{array}$$

$$\left[\begin{array}{c} \left[\begin{array}{ccccc} [3 & 1] & [1 & 0] & [1 & 1] \end{array}\right] \\ \left[\begin{array}{ccccc} [0 & 1] & [0 & 0] & [1 & 1] \end{array}\right] \end{array}\right] \begin{array}{l} 0 \\ \\ 1 \end{array} \quad \textit{terapija}$$

Slika 3.1: *Primer kontingenčne matrike*

Na sliki 3.1 je primer kontingenčne matrike, ki jo zgradi algoritem učenja optimalne odločitve pred ocenjevanjem atributa za gradnjo vozlišča drevesa. Atribut, ki ga algoritem ocenjuje s

pomočjo matrike na sliki 3.1, je diskreten in lahko zavzame vrednosti 0, 1 in 2. Domena ima dve možni terapiji, 0 in 1, ter dva razreda, z vrednostma 0 in 1. V matriko je razvrščenih deset primerov. Primer: trije primeri imajo vrednost atributa 0, pri njih je bila opravljena terapija 0, in spadajo v razred 0.

Atributom, ki so predstavljeni s kontingenčno matriko, kot je prikazana na sliki 3.1., ocenimo kvaliteto. Atribut lahko ocenjujemo tako, da ga razdelimo na podskupine primerov, ki imajo enako vrednost atributa, in ocenjujemo vsako skupino posebej, potem pa upoštevamo uteženo povprečje ocen. Utež določimo glede na delež primerov v posamezni skupini primerov z enako vrednostjo atributa. Takšno ocenjevanje v nadaljevanju poimenujemo *ocenjevanje atributa s povprečenjem*. *Brez povprečenja* ocenimo atribut tako, da ga ne razdelimo na podskupine in ga ocenimo.

3.4.2 Binarizacija zveznih atributov

Vozlišča klasifikacijskega drevesa predstavljajo atributi. Vozlišča se razdelijo na veje glede na vrednosti atributa. Vrednosti atributa morajo biti diskretne, zato zvezne atribute diskretiziramo. Zvezni atributi zavzemajo vrednosti na določenem intervalu. Ta interval razdelimo na dva dela. Treba je poiskati mejo, ki optimalno razdeli interval. Vozlišča, ki jih predstavljajo zvezni atributi, razdelijo primere v dve podmnožici: na tiste z večjo, in tiste z manjšo vrednostjo atributa od meje.

Najboljšo mejo poiščemo tako, da primere najprej uredimo po vrednostih danega zveznega atributa. Najprej določimo mejo med prvo vrednostjo atributa z urejenega seznama in ostalimi vrednostmi. Zatem ocenimo kvaliteto atributa, ki je tako razdeljen. Mejo premikamo po seznamu in ocenjujemo atribut s tako določenimi mejami. Izberemo mejo, pri kateri je imel atribut najboljšo oceno.

3.4.3 Rezanje drevesa

Zaradi nezanesljivosti nižjih nivojev drevesa, kjer vozliščem ustreza majhno število učnih primerov, je treba posvetiti tem nivojem še posebno pozornost. Ustavitveni pogoji poskušajo ustaviti gradnjo, ko le-ta postane nezanesljiva ali nepotrebna. Tako ublažimo ali skoraj v celoti odpravimo posledice prekomernega prilagajanja učnim podatkom. Z rezanjem torej onemogočimo tvorbo, ali pa odstranimo nepomembne veje, in s tem omejimo velikost odločitvenega drevesa, s tem pa povečamo klasifikacijsko točnost modela.

Rezanje glede na oceno atributov

Ustavitveni pogoj za gradnjo vozlišča je vnaprej določena najnižja sprejemljiva ocena atributa. Če je ocena najboljšega atributa nižja od določene najmanjše sprejemljive ocene, algoritem ne ustvari vozlišča s tem atributom. Algoritem ocenjuje atribute s statističnimi testi. Pri ocenah testov z vrednostjo p zavzame večina ocen atributov vrednost, ki je zelo blizu 1 ali 0. Za ocene atributov smo uporabili vrednosti statistik testov, ki niso omejene. Zaradi tega smo imeli težave pri določanju optimalne vrednosti najmanjše sprejemljive ocene, ki je lahko za vsako domeno in vsak nabor učnih podatkov drugačna.

Rezanje glede na število primerov v listih

Ustavitveni pogoj pri gradnji drevesa je lahko vnaprej določeno število primerov v listu. Če ima algoritem za gradnjo novega vozlišča na voljo manj primerov, kot je določeno z najmanjšim številom primerov v listu, algoritem ne zgradi tega vozlišča.

Omejevanje globine drevesa

Ustavitveni pogoj pri gradnji drevesa je lahko vnaprej določena globina drevesa. Če je maksimalna globina drevesa nastavljena na vrednost 0, algoritem zgradi samo koren drevesa.

3.4.4 m-ocena

Pri strojnem učenju pogosto ocenjujemo verjetnost. Lahko je podana vnaprej, bodisi jo moramo oceniti iz vhodnih podatkov. V slednjem primeru je pogosto potrebno oceniti verjetnost iz majhne množice vhodnih podatkov. Taka ocena verjetnosti je nezanesljiva.

m-ocena je ocena verjetnosti, ki so jo neodvisno razvili Smyth in Goodman (1990) ter Cestnik (1990). m-ocena je sestavljena iz absolutne frekvence *r*, ki je utežena s številom podatkov n, in iz apriorne ocene p_0, ki je utežena s parametrom m. m-ocena omogoča izbiro apriorne verjetnosti in njeno utežitev s parametrom m. Namesto ocenjene verjetnosti iz majhne množice vhodnih podatkov uporabimo m-oceno, ki jo računamo po formuli (3).

$$p = \frac{r + mp_0}{n + m} \qquad (3)$$

V strojnem učenju je m-ocena pogosto uporabljena pri rezanju dreves. V tem delu smo uporabili m-oceno v kontingenčnih matrikah, ki jih algoritem ustvari pred ocenjevanjem atributov. Namesto dejanskih frekvenc so v kontingenčnih matrikah frekvence, popravljene z m-oceno. Kot apriorno frekvenco smo uporabili frekvenco pojavitev primerov celotnega drevesa, popravili pa smo kontingenčne matrike poddreves.

3.5 Ocenjevanje uspešnosti učenja

3.5.1 Umetni podatki

Osnovne značilnosti učenja algoritma smo raziskali na umetnih domenah, saj smo brez težav konstruirali poljubno velike neodvisne testne množice podatkov. Poleg klasifikacijske točnosti in razumljivosti modela smo raziskali tudi krivuljo učenja ter vpliv šuma na klasifikacijsko točnost.

Klasifikacijska točnost

Napovedna točnost je prav gotovo najpomembnejša lastnost klasifikatorja. Najbolj enostavno jo izrazimo kot delež pravilno klasificiranih primerov.

Po modelu, sestavljenem za konstruiranje umetnih podatkov, smo sestavili množico učnih podatkov. Na podlagi učnih podatkov algoritem zgradi klasifikacijsko drevo. Sestavili smo množico testnih podatkov, ki imajo naključne vrednosti atributov (brez vrednosti terapije in razreda). Učnim podatkom v drevesu smo poiskali napovedano optimalno terapijo. Izbira optimalne terapije pomeni, da bo primer v ciljnem razredu. Testnim podatkom manjka samo še razred, ki smo ga sedaj, ko imamo napovedano terapijo, konstruirali po istem modelu, kot so bili sestavljeni učni podatki. Uspešnost učenja je izražena v odstotku ujemanja vrednosti razredov naučenih testnih podatkov s ciljnim razredom.

Razumljivost modela

Razumljivost modela je lastnost, ki označuje, v kolikšni meri je model razumljiv človeku in, ali je iz njega razvidno, kako je prišel do svojih odločitev (klasifikacije). Razumljivost je še posebej pomembna pri modeliranju v medicini.

3.5.2 Resnični podatki

Zaradi majhnega števila primerov v nekaterih domenah si pogosto ne moremo privoščiti testiranja klasifikacijske točnosti modela na neodvisni testni množici. V takšnih primerih nabor podatkov razdelimo na učno in testno množico. Na učni množici izvršimo učenje, rezultate preverimo na testni množici, postopek večkrat ponovimo, rezultat testiranja predstavlja povprečje opravljenih poskusov.

Uspešnost učenja smo ocenjevali z desetkratnim prečnim preverjanjem. Desetkratno prečno preverjanje razdeli množico razpoložljivih učnih primerov na deset približno enako močnih podmnožic in učenje ter testiranje izvede desetkrat. V i–tem izvajanju testira klasifikacijsko točnost modela na i–ti podmnožici, potem ko se je učil na primerih iz preostalih devetih podmnožic. Klasifikacijska točnost je povprečje rezultatov teh desetih testiranj. Testne množice so med seboj neodvisne, vsak primer pa se za testiranje uporabi natanko enkrat.

Poglavje 4

Poskusi

V tem poglavju so opisani poskusi delovanja algoritma učenja optimalne odločitve, ki smo jih opravili. Raziskali smo obnašanje algoritma pri različnih vrednostih parametrov, kot so najnižja sprejemljiva ocena atributa, najmanjše število primerov v listu, največja globina drevesa in m-ocena. Prav tako smo raziskali obnašanje algoritma pri različnih domenah in pri različno veliki množici učnih primerov. Algoritem učenja optimalne odločitve smo preizkusili z umetnimi in resničnimi podatki.

4.1 Resnični medicinski podatki

Algoritem učenja optimalne odločitve smo preizkusili na resničnih medicinskih podatkih, ki zajemajo podatke o nekaterih lastnostih pacientov, ter o vrsti in izidu njihovega zdravljenja. Optimalna odločitev, ki jo iščemo, je izbira primerne terapije za posameznega pacienta. Ciljni razred je uspešno zdravljenje pacienta.

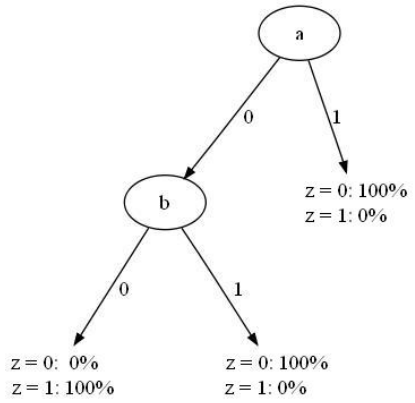
4.2 Modeli za konstruiranje umetnih podatkov

V tem poglavju smo opisali modele, po katerih smo konstruirali umetne podatke. Modeli so odločitvena drevesa. V listih teh dreves so možne odločitve (terapije) in verjetnost ciljnega razreda, v primeru določene odločitve. Na podlagi modela, ki opisuje domeno, smo konstruirali umetne podatke z generatorjem naključnih števil.

4.2.1 Trivialen model

Trivialen model (slika 4.1) služi predstavitvi delovanja algoritma. Umetna domena ima dva diskretna atributa (*a* in *b*), diskretni atribut *z* predstavlja odločitev (terapijo, zdravljenje), razred je diskreten in lahko zavzame vrednosti 0 in 1. Podatki so brez šuma.

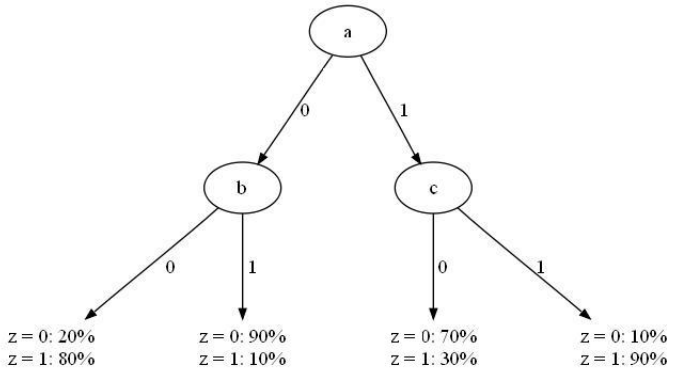
Na voljo imamo dve odločitvi (npr. dve terapiji) in možna sta dva razreda (dva izida). Eden od razredov je naš ciljni razred. Denimo, da je vrednost atributa *a* enaka 0, vrednost atributa *b* je enaka 1, naš ciljni razred je 1. Če se pri takem primeru odločimo za terapijo 0, bo v 100% primerov izid 1. V nasprotnem primeru bo v 100% primerov izid 0. To pomeni, da je terapija 1 optimalna za primere z vrednostmi atributov a = 0 in b = 1. Terapija 1 pa je zelo slaba za primere z vrednostmi atributov a = 1 in b = 0, saj je v primeru, da se odločimo zanjo, v 100% primerov izid enak 0, kar pa ni naš ciljni razred.



Slika 4.1: *Trivialen model za konstruiranje umetnih podatkov. Odstotki v listih predstavljajo deleže ciljnega primera pri posamezni odločitvi (z).*

4.2.2 Enostaven model

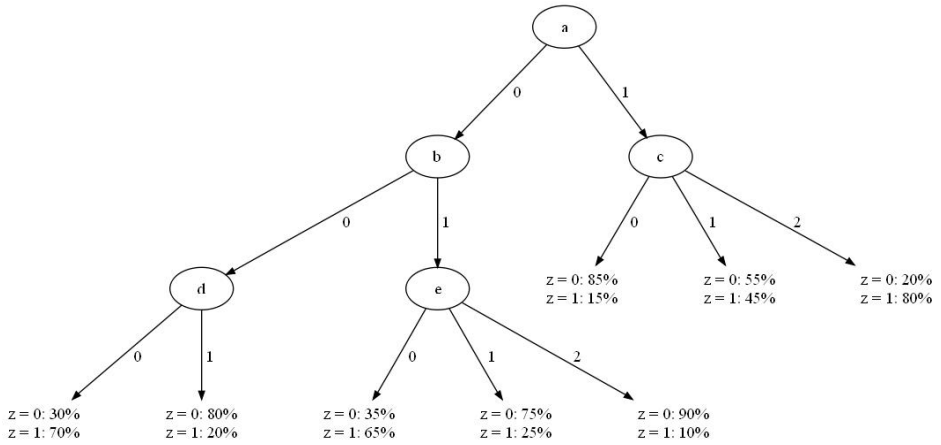
Enostaven model (slika 4.2) ima tri diskretne atribute, dve možni odločitvi in diskreten razred, ki lahko zavzame dve različni vrednosti. Trivialen model služi testiranju delovanja algoritma s šumom v podatkih.



Slika 4.2: *Enostaven model za konstruiranje umetnih podatkov. Odstotki v listih predstavljajo deleže ciljnega primera pri posamezni odločitvi (z).*

4.2.3 Kompleksen model

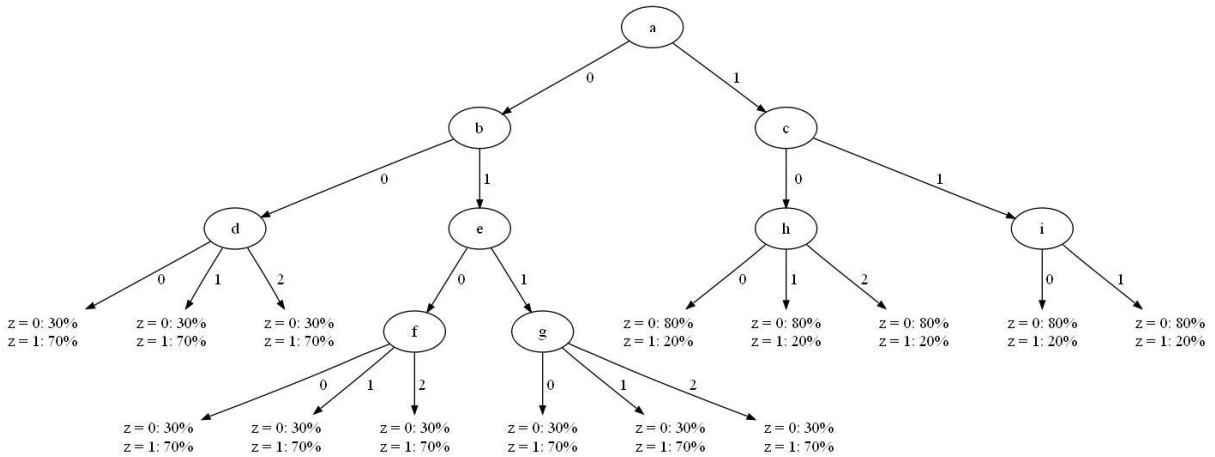
Kompleksen model (slika 4.3) ima pet atributov. Služi testiranju delovanja algoritma pri bolj kompleksni domeni, ki ima več atributov, od tega nekateri zavzemajo več kot le dve vrednosti, podatki pa imajo šum.



Slika 4.3: *Kompleksen model za konstruiranje umetnih podatkov. Odstotki v listih predstavljajo deleže ciljnega primera pri posamezni odločitvi (z).*

4.2.4 Model za rezanje

Model (slika 4.4) ima devet atributov. Služi testiranju vpliva rezanja na klasifikacijsko točnost drevesa. Ključnega pomena za klasifikacijo je le atribut *a*, vsi ostali atributi v modelu pa so pomembni samo za preizkus, ali bo algoritem odkril ključni atribut.



Slika 4.4: *Model za rezanje za konstruiranje umetnih podatkov*

4.3 Klasifikacijska točnost

Klasifikacijsko točnost izmerimo tako, da po istem modelu konstruiramo učne in testne primere. Algoritem učenja optimalne odločitve zgradi klasifikacijsko drevo na podlagi učnih podatkov. Testne podatke klasificiramo s pomočjo zgrajenega drevesa. Klasifikacijsko točnost izrazimo kot odstotek pravilno klasificiranih primerov oziroma, v našem primeru, delež primerov, kjer je drevo izbralo optimalno terapijo.

Najprej izmerimo klasifikacijsko točnost za umetne domene, brez rezanja drevesa in ostalih metod za povečanje kvalifikacijske točnosti. Meritev smo opravili na 500, 1000, 1500 in 2000 podatkih po 10–krat (vsakič na novem naboru podatkov) in upoštevali povprečno kvalifikacijsko točnost.

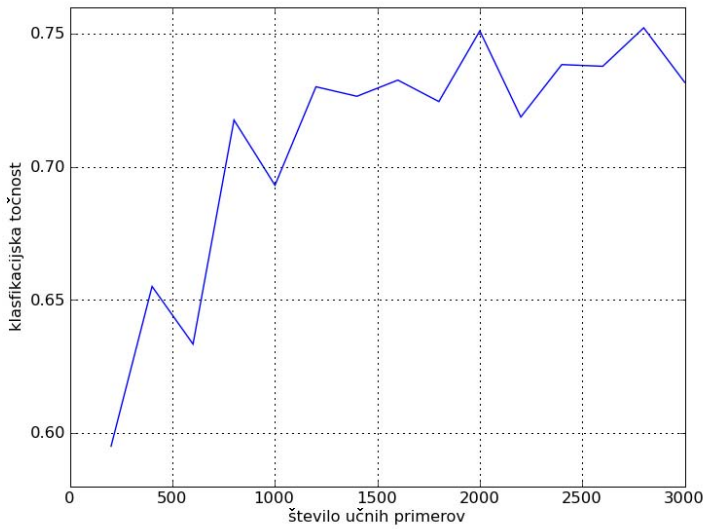
Število učnih primerov	HI-kvadrat s povprečenjem	HI-kvadrat brez povprečenja	KL s povprečenjem	KL brez povprečenja
Trivialen model (odločitev z=1 je boljša v 75 % primerov)				
500	100 %	100 %	75,2 %	75,3 %
1000	100 %	100 %	75,0 %	74,7 %
1500	100 %	100 %	75,0 %	75,0 %
2000	100 %	100 %	75,0 %	75,0 %
Enostaven model (vsaka odločitev je boljša v 50 % primerov)				
500	82,9 %	82,5 %	82,2 %	82,8 %
1000	82,5 %	82,6 %	82,5 %	82,4 %
1500	82,8 %	82,4 %	82,6 %	82,3 %
2000	82,5 %	82,5 %	82,6 %	82,5 %
Kompleksen model (odločitev z=0 je boljša v 63 % primerov)				
500	68,4 %	67,5 %	68,9 %	68,8 %
1000	71,1 %	71,1 %	70,6 %	70,7 %
1500	72,2 %	71,9 %	72,3 %	72,6 %
2000	73,0 %	73,2 %	73,2 %	72,9 %
Model za rezanje (vsaka odločitev je boljša v 50 % primerov)				
500	59,7 %	59,1 %	63,0 %	64,0 %
1000	60,5 %	59,5 %	62,9 %	64,9 %
1500	60,1 %	60,0 %	63,1 %	64,6 %
2000	61,2 %	60,5 %	63,4 %	65,0 %

Tabela 4.1: *Klasifikacijska točnost dreves, ki jih gradi algoritem učenja optimalne odločitve na umetno konstruiranih učnih podatkih*

V tabeli 4.1 so rezultati meritev klasifikacijske točnosti drevesa, ki ga zgradi algoritem učenja optimalne odločitve. Drevo ni rezano in nima nobenih drugih izboljšav (npr. m-ocena). Iz rezultatov je razvidno, da algoritem deluje, saj delež primerov, v katerih je izbral pravilno odločitev, presega delež primerov, v katerih je v splošnem boljša določena odločitev. Izjema je Kulback-Leiblerjeva divergenca na trivialnem modelu, kjer bi dosegli enak uspeh, če bi vedno izbrali terapijo 0. Po pričakovanjih klasifikacijska točnost v splošnem pada z večanjem kompleksnosti domene in narašča z večanjem števila učnih podatkov.

4.4 Krivulja učenja

V poskusu smo raziskali, kako na učenje algoritma vpliva število učnih primerov. Spreminjali smo velikost učne množice in merili klasifikacijsko točnost, ki jo je dosegal klasifikator. Poskus smo izvajali vsakič na 100 različnih učnih množicah in kot rezultat upoštevali povprečno klasifikacijsko točnost. Na sliki 4.5 je graf odvisnosti klasifikacijske točnosti od števila učnih primerov za test KL brez povprečenja. Tudi pri ostali treh načinih ocenjevanja atributa, ki smo jih preizkusili, ima krivulja grafa enako obliko.

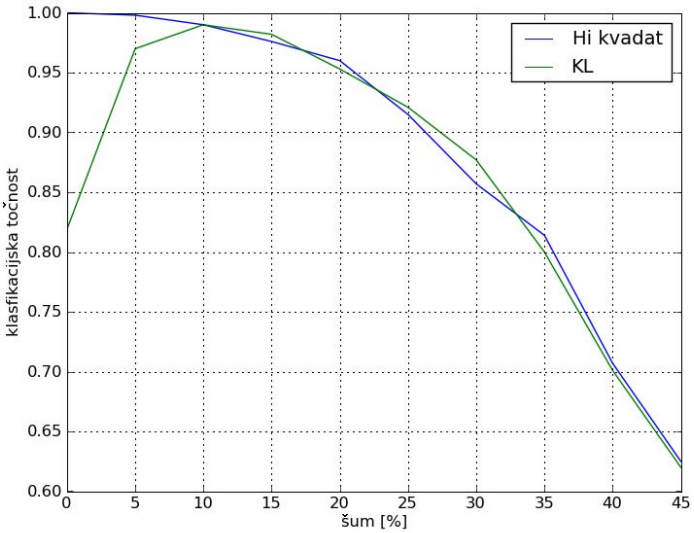


Slika 4.5: *Krivulja učenja (drevo je zgrajeno po enostavnem modelu, mera kvalitete atributa je KL brez povprečenja)*

Klasifikacijska točnost je odvisna od števila učnih primerov. Z večanjem števila učnih primerov narašča klasifikacijska točnost modela. Do tega pride zato, ker je zanesljivost ocene kvalitete atributa odvisna od števila učnih primerov. Atribute merimo s statističnimi testi, ki dajejo bolj zanesljive ocene, kadar se testirajo večji vzorci.

4.5 Občutljivost metode na šum

S poskusom smo raziskali, kako na učenje algoritma vpliva šum v podatkih. Spreminjali smo delež šumnih podatkov učne množice in merili klasifikacijsko točnost, ki jo je dosegal klasifikator. Šum je dodan vsem atributom. Pri generiranju primerov je vnaprej določeno, kakšen delež generiranih učnih primerov pripada razredu, kot ga določa model, in, kakšen delež generiranih učnih podatkov od modela odstopa (ti primeri predstavljajo šum). Osnova za model, po katerem smo konstruirali učne podatke, je enostaven model. Poskus smo izvajali vsakič na 50 različnih učnih množicah in kot rezultat upoštevali povprečno klasifikacijsko točnost. Na sliki 4.6 je graf odvisnosti klasifikacijske točnosti od števila učnih primerov za testa KL in Hi-kvadrat brez povprečenja. Tudi pri drugih dveh načinih ocene atributa (s povprečenjem) ima krivulja grafa enako obliko.



Slika 4.6: *Odvisnost klasifikacijske točnosti od šuma v učnih podatkih (drevesa so zgrajena na umetno konstruiranih podatkih, ki jim spreminjamo šum)*

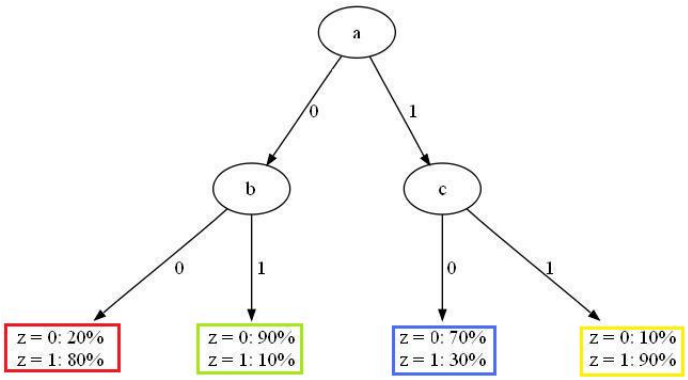
Uspešnost učenja klasifikatorja je odvisna od šuma v podatkih. Več kot je šuma, manjša je natančnost klasifikacije.

Test KL je pri podatkih z malo šuma (do 10%) slabši od testa Hi-kvadrat. To je razvidno tudi iz rezultatov meritve klasifikacijske točnosti v tabeli 4.1, kjer dosega klasifikacijsko točnost približno 75% na podatkih, zgrajenih po trivialnem modelu, ki nimajo šuma (Hi-kvadrat dosega točnost 100%). Pri šumu, večjem od 10%, sta testa približno enako dobra. Da bi lahko izpostavili enega od njiju kot boljšega, bi bilo treba izvesti več poskusov na različno velikih množicah in na večjem številu domen.

4.6 Pravilnost dobljenih modelov

V naslednjem poskusu smo primerjali model, po katerem smo konstruirali podatke s klasifikacijskim drevesom, ki ga zgradi algoritem učenja optimalne odločitve na podlagi konstruiranih podatkov.

Enostaven model



Slika 4.7: *Enostaven model*

Klasifikacijsko drevo, ki ga zgradi algoritem učenja optimalne odločitve na podatkih, konstruiranih po enostavnem modelu

```
b=0
|     c=0
|     |     a=0: z = 1
|     |     a=1: z = 0
|     c=1
|     |     a=0: z = 1
|     |     a=1: z = 1
|     |
b=1
|     a=0
|     |     z = 0
|     |
|     a=1
|     |     c=0: z = 0
|     |     c=1: z = 1
```

Slika 4.8: *Klasifikacijsko drevo, ki ga zgradi algoritem učenja optimalne odločitve na podatkih, konstruiranih po enostavnem modelu*

Drevo (slika 4.8) ni enake oblike kot model (slika 4.7), po katerem so bili podatki konstruirani, vendar natančnejši pregled pokaže, da daje enake odločitve, kot bi jih dajalo preprostejše optimalno drevo. Drevo ne dosega klasifikacijske točnosti 100% zaradi šuma v konstruiranih podatkih (dosega natančnost med 80% in 85%).

Iz drevesa na sliki 4.8 lahko razberemo priporočene vrednosti atributa *z* (zdravljenje oz. terapija) v listih. Če ima primer vrednosti atributov *b*, *c* in *a* enake 0, je za takšne primere priporočena terapija 1. Iz modela na sliki 4.8 razberemo, da je za primere z vrednostmi atributov *a* in *b* enake 0 pri izbiri terapije 0 v 20% izid zdravljenja uspešen, pri terapiji 1 pa pri 80%. Pri takšnih primerih je bolje izbrati terapijo 1. Klasifikacijsko drevo je torej uspešno predlagalo terapijo.

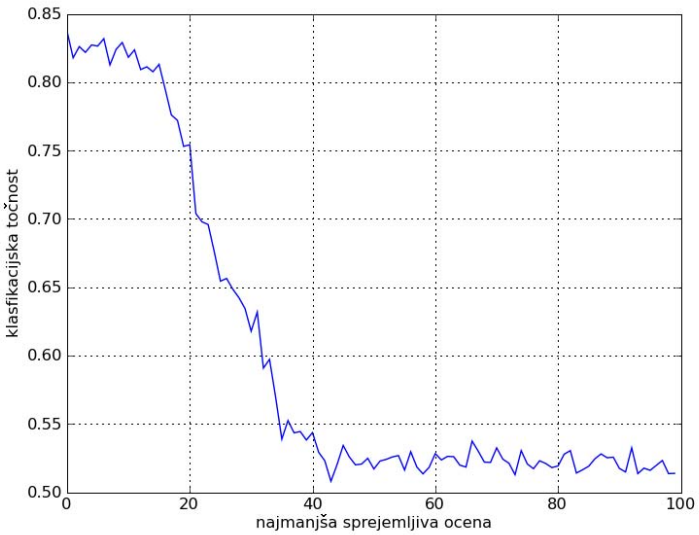
4.7 Rezanje drevesa

Klasifikacijsko točnost lahko povečamo z rezanjem drevesa. Z rezanjem ublažimo posledice prekomernega prilagajanja učnim podatkom. Z rezanjem torej onemogočimo tvorbo, ali pa odstranimo nepomembne veje. V naslednjih poskusih smo preverili, kako vpliva rezanje na klasifikacijsko točnost dreves, zgrajenih po dveh modelih za konstruiranje umetnih podatkov. Modela sta konstruirana tako, da ima na njiju rezanje ravno nasproten vpliv. Poskuse izvedemo z mero kvalitete atributa hi-kvadrat s povprečenjem na 500 učnih podatkih.

4.7.1 Rezanje glede na oceno atributa

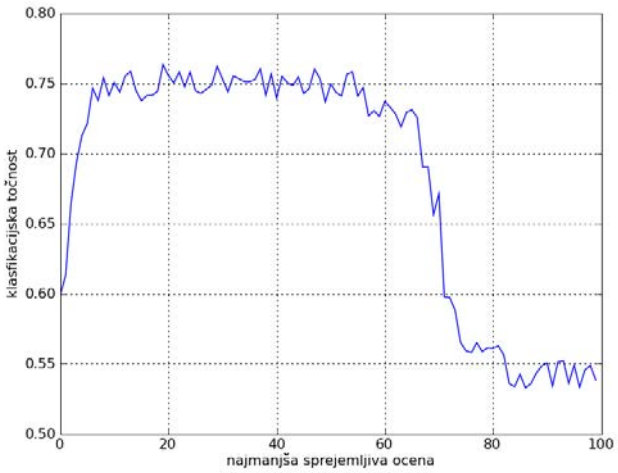
Z večanjem najmanjše sprejemljive ocene režemo drevo tako, da preprečimo nastajanje vozlišč iz atributov, ki so ocenjeni manj, kot je vnaprej določena najmanjša sprejemljiva ocena (WA).

Enostaven model



Slika 4.9: *Odvisnost klasifikacijske točnosti od najmanjše sprejemljive ocene (drevo je zgrajeno po enostavnem modelu, mera kvalitete atributa je hi-kvadrat s povprečenjem)*

Model za rezanje



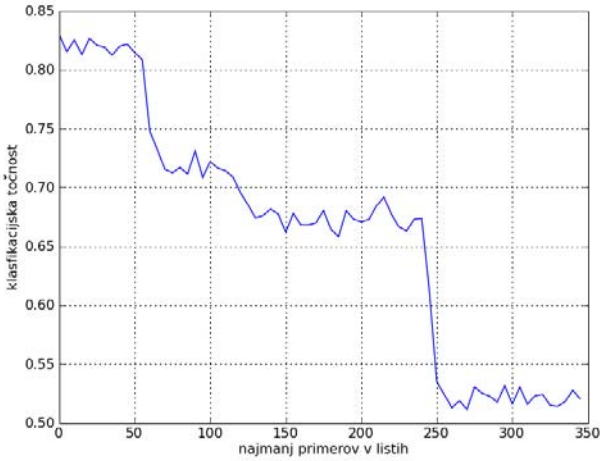
Slika 4.10: *Odvisnost klasifikacijske točnosti od najmanjše sprejemljive ocene (drevo je zgrajeno po modelu za rezanje, mera kvalitete atributa je hi-kvadrat s povprečenjem)*

Vpliv rezanja z določanjem najmanjše sprejemljive ocene je odvisen od domene. Pri enostavnem modelu se klasifikacijska točnost poslabša z določanjem WA (slika 4.9), pri modelu za rezanje pa se izboljša (slika 4.10). Pri enostavnem modelu ima vsak atribut vpliv na optimalno odločitev, medtem ko je pri modelu za rezanje pomemben samo atribut *a*. Pri modelu za rezanje se CA poveča z večanjem WA, saj z omejevanjem WA onemogočimo nastajanje nepotrebnih vozlišč.

4.7.2 Rezanje glede na število primerov v listih

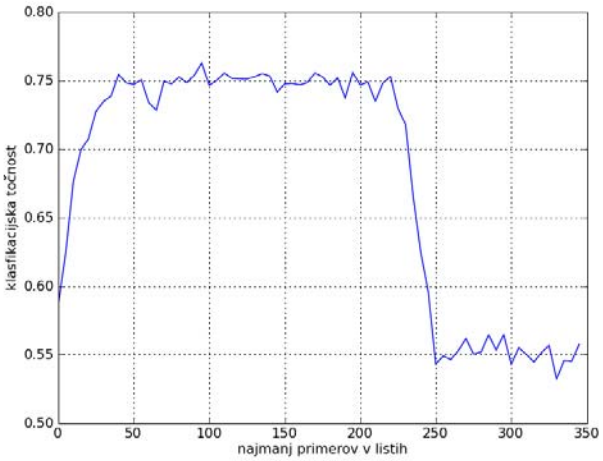
Z določanjem najmanjšega števila primerov v listih onemogočamo tvorbo vozlišč z manj primeri, kot jih vnaprej določimo. Na ta način režemo drevo, saj onemogočamo tvorbo vozlišč, ko ni več dovolj primerov.

Enostaven model



Slika 4.11: *Odvisnost klasifikacijske točnosti od števila najmanj primerov v listih (drevo je zgrajeno po enostavnem modelu, mera kvalitete atributa je hi-kvadrat s povprečenjem)*

Model za rezanje



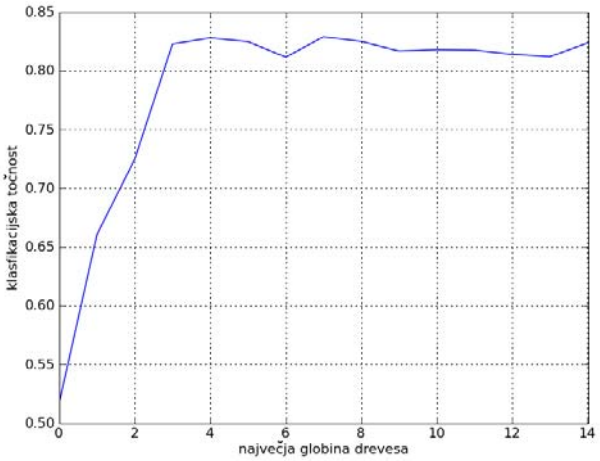
Slika 4.12: *Odvisnost klasifikacijske točnosti od števila najmanj primerov v listih (drevo je zgrajeno po modelu za rezanje, mera kvalitete atributa je hi-kvadrat s povprečenjem)*

Vpliv rezanja z določanjem najmanjšega števila primerov v listih je odvisen od domene. Pri enostavnem modelu se klasifikacijska točnost poslabša z določanjem najmanjšega števila primerov v listih (4.11), pri modelu za rezanje pa se izboljša (slika 4.12). Takšen izid poskusa je pričakovan. Pri enostavnem modelu ima vsak atribut vpliv na optimalno odločitev, medtem ko je pri modelu za rezanje pomemben samo atribut *a*. Pri modelu za rezanje se klasifikacijska točnost poveča, ker z rezanjem onemogočimo nastajanje nepotrebnih vozlišč.

4.7.3 Omejevanje globine drevesa

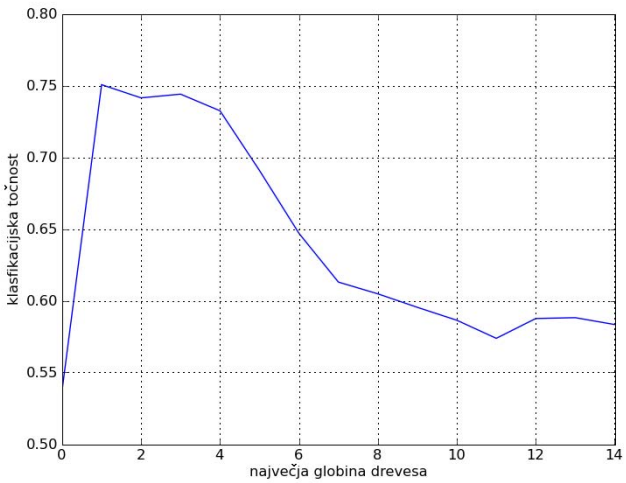
Z določanjem največje globine drevesa režemo drevo. S tem preprečimo nastajanje nepomembnih vozlišč.

Enostaven model



Slika 4.13*: Odvisnost klasifikacijske točnosti od globine drevesa (drevo je zgrajeno po enostavnem modelu, mera kvalitete atributa je hi-kvadrat s povprečenjem)*

Model za rezanje



Slika 4.14: *Odvisnost klasifikacijske točnosti od globine drevesa (drevo je zgrajeno po modelu za rezanje, mera kvalitete atributa je hi-kvadrat s povprečenjem)*

Klasifikacijska točnost drevesa je odvisna od oblike drevesa. CA doseže maksimum pri enostavnem drevesu pri globini drevesa 3 (slika 4.13). Enostaven model je sestavljen tako, da so pomembni vsi atributi in ima globino 3. Dokler je drevo preveč porezano, torej ima globino manjšo od 3, ima nižjo klasifikacijsko točnost. Pri modelu za rezanje doseže CA maksimum pri globini 1 (slika 4.14). V drevesu je pomemben samo atribut *a*, zato drevo doseže največjo klasifikacijsko točnost, kadar ima samo eno vozlišče.

4.8 Resnični podatki

V tem poglavju so opisani opravljeni poskusi na resničnih medicinskih podatkih.

4.8.1 Opis podatkov

Ponovitev raka na dojki - 1988

Podatke o bolnicah z rakom na dojki je zbral Onkološki inštitut v Ljubljani (leto 1988) [1]. Podatki obsegajo 286 primerov, opisanih z devetimi atributi. Podatki o pacientkah so razdeljeni na dva razreda glede na to, ali se je pri njih rak na dojki ponovil. Ponovil se je pri 85 pacientkah. Odločitev oz. terapijo predstavlja atribut irradiat, ki lahko zavzema eno od dveh vrednosti. Cilj klasifikacije z modelom, ki ga zgradi algoritem iskanja optimalne odločitve, je določiti za vsak primer takšno vrednost atributa irradiat, da bo primer uvrščen v ciljni razred.

Atributi:

razred: *no-recurrence-events, recurrence-events*
terapija (irradiat): *yes, no*
age*: 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99*
menopause: *lt40, ge40, premeno*
tumor-size: *0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59*
inv-nodes: *0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39*
node-caps: *yes, no*
deg-malig: *1, 2, 3*
breast: *left, right*
breast-quad: *left-up, left-low, right-up, right-low, central*

Tabela 4.2: *Atributi resnične domene Ponovitev raka na dojki - 1988*

Ponovitev raka na dojki - 2001

Podatki obsegajo 696 primerov, opisanih s 14 atributi. Podatki so zbrani na Onkološkem inštitutu v Ljubljani med letoma 1997 in 2001 in zajemajo podatke o zdravljenih pacientkah z zgodnjim rakom dojk. Zdravljenje je označeno za uspešno, če se bolezen ni ponovila v treh letih po zdravljenju.

Atributi:

razred: *bolezen se ponovi, bolezen se ne ponovi*
terapija: *Anthracyclines, CMF*
vrsta_kt: *2, 4*
progres: *0, 1*
DFS: *continuous*
tumor grade: *I, II, III*
tumor type: *invas. duct., other*
menopausal status: *peri/pre, post*
hormonal receptors: *neg, pos*
tumor size: *<=20, >20*
arm: *2, 3*
cardiovascular invasion: *yes, no*
uPA & PAI levels: *both low, otherwise*
affected lymph nodes: *1-3, >3, neg*

Tabela 4.3: *Atributi resnične domene Ponovitev raka na dojki - 2001*

4.8.2 Uspešnost odločanja klasifikacijskega modela

Iz resničnih učnih podatkov razberemo, kakšen delež zdravljenj je uspel, torej kakšen je zdravnikov uspeh zdravljenja. Algoritem učenja optimalne odločitve zgradi model na podlagi učnih podatkov in za vsak primer predlaga terapijo. Primerov, za katere se je zdravnik drugače odločil od klasifikacijskega modela, ne obravnavamo, saj ne moremo vedeti, kakšen bi bil izid zdravljenja, če bi zdravnik izbral drugo terapijo, zato ne moremo vedeti, ali se je klasifikacijski model pravilno odločil. Opazujemo samo primere, kjer klasifikacijski model predlaga enako terapijo, kot jo je pacientki predpisal zdravnik. Med temi primeri izračunamo delež uspelih zdravljenj, ki nam oceni uspešnost izbire terapije klasifikacijskega modela. Preverjamo torej, ali

je model zmožen odkriti, pri katerih pacientkah je zdravnik izbral ustrezno terapijo. Način merjenja uspešnosti ni čisto pravilen, saj ne preverjamo odločitev, ko klasifikacijsko drevo predlaga drugačno terapijo, kot jo je izbral zdravnik. Drevo lahko "goljufa" tako, da se pri težavnih pacientih odloči drugače, kot zdravnik.

Klasifikacijsko drevo zgradimo na podlagi podatkov o pacientkah. Zdravnik je izbiral med terapijami, zdravljenje pa je uspelo pri določenem deležu pacientk. Pri testiranju uspešnosti klasifikacijskega modela želimo doseči uspešnost klasifikacije, ki je večja od deleža pacientk z uspešnim izidom zdravljenja v naboru podatkov. Če primerom naključno izbiramo terapijo in med temi primeri opazujemo samo tiste, ki imajo enako terapijo, kot jo je izbral zdravnik, dobimo podmnožico primerov, pri kateri je verjetnost izida zdravljena enaka, kot na celi množici primerov.

V tabeli so meritve uspešnosti določanja optimalne terapije za klasifikacijska drevesa, zgrajena na dveh različnih resničnih domenah.

	HI-kvadrat s povprečenjem	**HI-kvadrat** brez povprečenja	**KL** s povprečenjem	**KL** brez povprečenja
Ponovitev raka dojke - 1988 (delež uspešno pozdravljenih 70,3%)				
Uspešnost odločanja klasifikacijskega drevesa	77,8 %	74,4 %	75,2 %	74,8 %
Delež enakih odločitev zdravnika in klasifikacijskega modela	75,2 %	73,1 %	74,5 %	78,7 %
Ponovitev raka dojke - 2001 (delež uspešno pozdravljenih 74,3 %)				
Uspešnost odločanja klasifikacijskega drevesa	79,6 %	73,9 %	78,0 %	79,0 %
Delež enakih odločitev zdravnika in klasifikacijskega modela	71,8 %	59,4 %	71,0 %	74,1 %

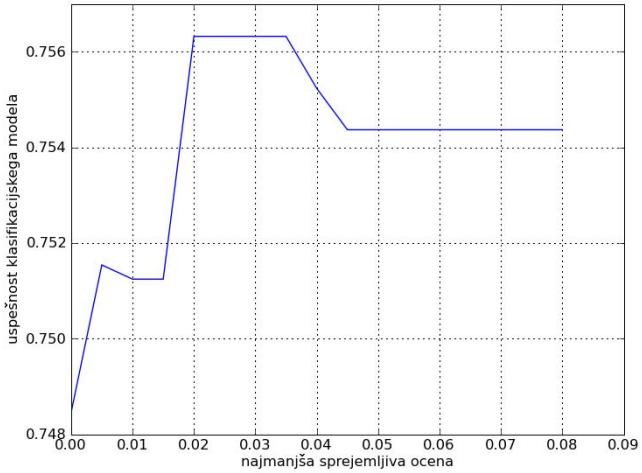
Tabela 4.4: *Uspešnost odločanja klasifikacijskega drevesa, zgrajenega na resničnih podatkih*

4.8.3 Rezanje drevesa

Z rezanjem klasifikacijskega drevesa smo poskusili odpraviti posledice prekomernega prilagajanja učnim podatkom. V naslednjih poskusih smo preizkusili več načinov rezanja drevesa.

Rezanje glede na oceno atributa

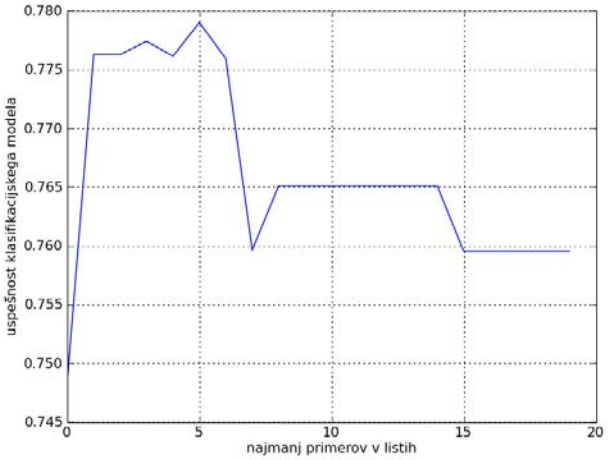
Vpliv določanja najmanjše sprejemljive ocene na kvaliteto modela je zanemarljiv in verjetno povsem naključen (slika 4.15).



Slika 4.15: *Odvisnost klasifikacijske točnosti od najmanjše sprejemljive ocene (drevo je zgrajeno na resnični domeni Ponovitev raka dojke - 1988, mera kvalitete atributa je KL brez povprečenja)*

Rezanje glede na število primerov v listih

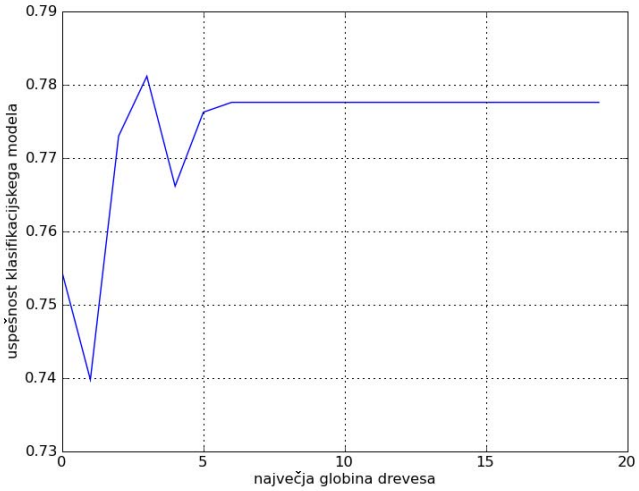
Rezanje z najmanj primeri v listih poveča klasifikacijsko točnost modela. Kadar je najmanj primerov v listih enako 5, dosega drevo klasifikacijsko točnost 78% (slika 4.16).



Slika 4.16: *Odvisnost klasifikacijske točnosti od najmanj primerov v listih (drevo je zgrajeno na resnični domeni Ponovitev raka dojke - 1988, mera kvalitete atributa je KL brez povprečenja)*

Omejevanje globine drevesa

Pri največji globini drevesa, enaki 3, dosega model točnost približno 0.78% (slika 4.17).



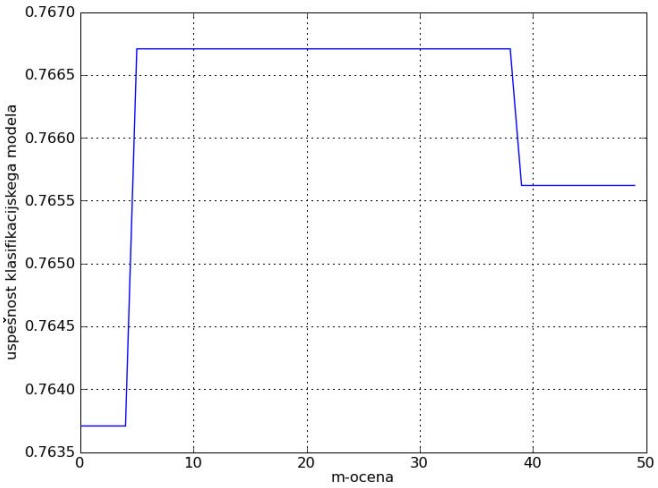
***Slika 4.17**: Odvisnost klasifikacijske točnosti od globine drevesa (drevo je zgrajeno na resnični domeni Ponovitev raka dojke 1988, mera kvalitete atributa je hi-kvadrat s povprečenjem)*

Poskusi rezanja drevesa kažejo, da je najbolj primeren način rezanja z najmanj primerov v listih, saj s tem dosegamo največjo klasifikacijsko točnost. Z rezanjem z najmanj primeri v listih pozitivno vplivamo na zanesljivost statističnih testov, s katerimi ocenjujemo kvaliteto atributov za gradnjo vozlišč drevesa.

4.8.4 Vpliv m-ocene

Pri iskanju optimalne odločitve s klasifikacijskim drevesom je treba pred vsako gradnjo vozlišča oceniti vse atribute. Ocena atributov temelji na oceni verjetnosti porazdelitve primerov. Da bi omilili posledice majhne množice podatkov, iz katerih je treba oceniti verjetnost, predvsem pri gradnji nižjih nivojev drevesa, uporabimo pri ocenjevanju verjetnosti m-oceno.

Pri tem poskusu smo opazovali, kako na natančnost klasifikacije vpliva m-ocena. Spreminjali smo vrednost m-ocene in merili klasifikacijsko točnost modela. Vsako meritev smo izvedli na 50 različnih učnih množicah.



Slika 4.18: *Odvisnost klasifikacijske točnosti od m-ocene (drevo je zgrajeno na resnični domeni Ponovitev raka dojke 1988, mera kvalitete atributa je hi-kvadrat s povprečenjem)*

m-ocena vpliva na uspešnost klasifikacijskega modela. Vpliv m-ocene je zanemarljiv. Na sliki 4.18 je prikazana odvisnost klasifikacijske točnosti modela od m-ocene za algoritem, ki ocenjuje atribute s testom hi-kvadrat in ne uporablja povprečenja.

4.8.5 Klasifikacijsko drevo

Za oba nabora resničnih podatkov smo narisali klasifikacijski drevesi (sliki 4.19 in 4.20), ki ju zgradi algoritem iskanja optimalne odločitve. Obe drevesi sta porezani in dosegata visoko klasifikacijsko točnost.

Ponovitev raka dojke - 1988

```
menopause=lt40: no
menopause=ge40 no
menopause=premeno
|    deg-malig=1: no
|    deg-malig=2
|    |    node-caps=no: no
|    |    node-caps=yes: yes
|    deg-malig=3
|    |    node-caps=no: no
|    |    node-caps=yes: yes
```

Slika 4.19: *Klasifikacijski model za domeno Ponovitev raka dojke - 1988*

Ponovitev raka dojke - 2001

```
Hormonal receptors=neg: CMF
Hormonal receptors=pos
|    Tumor size<=20
|    |    Affected lymph nodes=1-3
|    |    |    vrsta_kt=2: CMF
|    |    |    vrsta_kt=4: Anthracyclines
|    |    Affected lymph nodes>3: Anthracyclines
|    |    Affected lymph nodes=neg
|    |    |    Tumor grade=I: Anthracyclines
|    |    |    Tumor grade=II: Anthracyclines
|    |    |    Tumor grade=III
|    |    |    |    Cardiovascular invasion=no: CMF
|    |    |    |    Cardiovascular invasion=yes: Anthracyclines
|    Tumor size>20
|    |    progres=0: Anthracyclines
|    |    progres=1
|    |    |    Cardiovascular invasion=no: Anthracyclines
|    |    |    Cardiovascular invasion=yes: CMF
```

Slika 4.20: *Klasifikacijski model za domeno Ponovitev raka dojke- 2001*

V drugem poglavju smo povzeli članek, v katerem so avtorji predstavili odločitvene nomograme, s pomočjo katerih so poiskali optimalne odločitve za domeno Ponovitev raka dojke - 2001.

Na sliki 4.20 lahko razberemo, da klasifikacijski model predlaga primeru z vrednostjo atributa *hormonal receptors = neg* izbiro terapije *CMF*. Tudi odločitveni nomogram se nagiba k izbiri terapije *CMF* za paciente z vrednostjo atributa *hormonal receptors = neg*. Vendar pri nomogramu to še ni dokončna predlagana odločitev, saj nanjo vplivajo tudi vrednosti ostalih atributov.

Učenje na resničnih podatkih smo zaradi desetkratnega prečnega preverjanja izvedli desetkrat. Drevo na sliki 4.20 je le eno izmed dreves, ki so nastala. V dodatku je primer še enega izmed nastalih dreves. Algoritem klasifikacijskega modela je zelo občutljiv. Že pri majhnih odstopanjih v različnih vzorcih učnih podatkov zgradi različna drevesa. Tudi nastavitve parametrov, kot so različni testi (Hi-kvadrat, KL, s povprečenjem, brez povprečenja) in nastavitve za rezanje drevesa (WA, najmanj primerov v listih), ki dajejo približno enake klasifikacijske točnosti, znatno vplivajo na zgradbo drevesa.

Poglavje 5

Zaključek

V pričujočem delu smo razvili spremenjeni algoritem za gradnjo klasifikacijskega modela, ki temelji na učenju iz učnih podatkov. Algoritem za gradnjo klasifikacijskih dreves smo spremenili tako, da sestavlja drevesa, ki novih primerov ne uvršča v razrede, temveč jim določa optimalno odločitev (terapijo). Spremembi algoritma sta transformacija drevesa in kriterij za oceno kvalitete atributa. Drevo smo transformirali tako, da njegovi listi določajo optimalno odločitev (terapijo). Kriterij za oceno kvalitete atributa ne meri informacijskega prispevka atributa, kakor v standardnih algoritmih gradnje klasifikacijski dreves, temveč različnost porazdelitev znotraj skupine. Za kriterij smo uporabili statistični test, ki meri različnost porazdelitve primerov. V delu smo preizkusili delovanje algoritma z dvema različnima statistikama (hi-kvadrat in Kullback-Leiblerjeva divergenca) in primerjali njuni uspešnosti. Pri vsakem od testov smo uporabili dva načina ocenjevanja atributa. Pri enem od načinov smo uporabili uteženo povprečje ocen testa za skupine primerov z različnimi vrednostmi atributa, pri drugem načinu pa oceno testa za vse primere, ne glede na njihovo vrednost atributa.

Delovanje algoritma smo preizkusili na umetno konstruiranih in resničnih podatkih. Pri obeh testih, Hi-kvadrat in Kullback Leibler, je iskanje optimalne odločitve približno enako uspešno. Uspešnost iskanja optimalne odločitve je zelo odvisna od oblike klasifikacijskega drevesa, ki ga zgradi algoritem. Najbolj uspešna so drevesa, ki čimbolj povzemajo zakonitost domene. Natančnost klasifikacije modela se povečuje z večanjem nabora učnih podatkov.

Določanje kvalitete dobljenih modelov, zgrajenih na umetno konstruiranih podatkih, ni bilo težavno. Učenju drevesa na učnih podatkih je sledilo testiranje dobljenega modela na testnih podatkih, ki so bili konstruirani po enakem postopku, kot učni podatki. Klasifikacijska točnost modela je bila izražena kot delež pravilno določenih terapij. Z večanjem deleža šuma v podatkih se je klasifikacijska točnost nižala. Klasifikacijski model je dosegal visoke klasifikacijske točnosti, ki so se s pravilnim rezanjem še povečale. Klasifikacijski model prav tako uspešno povzema zakonitost umetnih domen, saj je zgradil zelo podobno, pogosto celo ekvivalentno drevo tistemu, po katerem so bili umetni podatki konstruirani.

Težavnejše je bilo določanje kvalitete modelov, zgrajenih na resničnih podatkih. Uspešnost določanja terapije s pomočjo klasifikacijskega modela lahko preverimo samo v primerih, ko se klasifikacijski model odloči enako, kot se je odločil zdravnik, saj le v teh primerih vemo, kakšen

je bil izid zdravljenja. Klasifikacija realnih podatkov je uspešna, saj je model dosegal višje deleže uspešno določenih terapij, kot so bile v učnih podatkih.

Izkazalo se je, da algoritem iskanja optimalnih odločitev deluje, saj uspešno išče optimalne odločitve. Slabosti modela so nestabilnost strukture drevesa, nepreglednost modela in nezanesljiva preverljivost kvalitete modela (pri resničnih domenah). Kljub naštetim slabostim je model lahko uporaben, predvsem, če bo izpopolnjen z nadaljnjimi raziskavami in izboljšavami. V medicini se metode strojnega učenja uporabljajo v diagnostiki in prognostiki, pojavljati pa so se začeli tudi algoritmi, ki bodo lahko neposredno uporabni pri določanju ustrezne terapije.

Dodatek

Transformacija drevesa

Spodnja funkcija v Pythonu spremeni klasifikacijsko drevo tako, da namesto napovedovanja razreda izbira optimalno odločitev.

```
def transformTree(node, treatment, goalClass):
    treatments = [0.] * len(treatment.values)
    if node.branches:
        hasNull = False
        for br in node.branches:
            if br:
                for treat, n  in enumerate(transformTree(br, treatment,
goalClass)):
                    treatments[treat] += n
            else:
                hasNull = True
        if hasNull:
            bestBranch = treatments.index(max(treatments))
            defaultNode = orange.TreeNode()
            defaultNode.nodeClassifier = orange.DefaultClassifier(treatment)
            defaultNode.nodeClassifier.defaultVal = orange.Value(treatment,
bestBranch)
            defaultNode.nodeClassifier.defaultDistribution = node.distribution =
[i==bestBranch for i in range(len(treatment.values))]
            for i, br in enumerate(node.branches):
                if not br:
                    node.branches[i] = defaultNode
    else:
        bestProb, bestBranch = -1, None
        for bi, br in enumerate(node.contingency["z"]):
            if goalClass == -1:
                if (not bestBranch or br.average() > bestProb) and br.cases:
                    bestProb, bestBranch = br.average(), bi
            else:
                if br[goalClass] > bestProb:
                    bestProb, bestBranch = br[goalClass], bi
        node.nodeClassifier = orange.DefaultClassifier(treatment)
        node.nodeClassifier.defaultVal = orange.Value(treatment, bestBranch)
        node.nodeClassifier.defaultDistribution = node.distribution =
[i==bestBranch for i in range(len(treatment.values))]
        treatments[bestBranch] = node.contingency["z"].outerDistribution.abs
    return treatments
```

Klasifikacijski model

Domena: Ponovitev raka dojke - 2001

```
Affected lymph nodes=1-3
|    Tumor size<=20: CMF
|    Tumor size>20
|    |    Tumor type=other: Anthracyclines
|    |    Tumor type=invas. duct.
|    |    |    Cardiovascular invasion=no
|    |    |    |    Arm=3: Anthracyclines
|    |    |    |    Arm=2
|    |    |    |    |    Hormonal receptors=neg: CMF
|    |    |    |    |    Hormonal receptors=pos: Anthracyclines
|    |    |    Cardiovascular invasion=yes
|    |    |    |    Hormonal receptors=neg: CMF
|    |    |    |    Hormonal receptors=pos
|    |    |    |    |    Arm=2: Anthracyclines
|    |    |    |    |    Arm=3: CMF
Affected lymph nodes>3
|    Tumor size<=20
|    |    Cardiovascular invasion=no: Anthracyclines
|    |    Cardiovascular invasion=yes: CMF
|    Tumor size>20
|    |    Tumor type=invas. duct.
|    |    |    Arm=2: CMF
|    |    |    Arm=3
|    |    |    |    Cardiovascular invasion=no: Anthracyclines
|    |    |    |    Cardiovascular invasion=yes
|    |    |    |    |    Hormonal receptors=neg: CMF
|    |    |    |    |    Hormonal receptors=pos
|    |    |    |    |    |    progres=0: Anthracyclines
|    |    |    |    |    |    progres=1: CMF
|    |    Tumor type=other
|    |    |    progres=0: Anthracyclines
|    |    |    progres=1: CMF
Affected lymph nodes=neg
|    Cardiovascular invasion=no
|    |    progres=0: Anthracyclines
|    |    progres=1
|    |    |    Hormonal receptors=neg: CMF
|    |    |    Hormonal receptors=pos: Anthracyclines
|    Cardiovascular invasion=yes
|    |    Tumor size<=20: Anthracyclines
|    |    Tumor size>20: CMF
```

Seznam slik

Seznam tabel

Literatura

[1] A. Asuncion, D. J. Newman, *UCI Machine Learning Repository*, University of California, Irvine, School of Information and Computer Sciences, 2009. Dostopno na: `http://www.ics.uci.edu/~mlearn/MLRepository.html.`

[2] J. Demšar, A. Sadikov, T. Čufer, "Decision nomograms", delo še ni objavljeno.

[3] J. Demšar, B. Zupan in G. Leban, "Orange: From Experimental Machine Learning to Interactive Data Mining". White Paper (`www.ailab.si/orange`), Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, Ljubljana 2004.

[4] I. Kononenko, *Strojno učenje*, Ljubljana: Fakulteta za računalništvo in informatiko, 2005.

[5] M. Leblanc and J. Crowley, "Survival trees by goodness of split", *Journal of the American Statistical Association*, junij 1993, str. 457–467.

[6] X. Su, C. Tsai, H. Wang, D. M. Nickerson, B. Li, "Subgroup Analysis via Recursive Partitioning", *Journal of Machine Learning Research*, februar 2009, str 141-158.

[7] G. Vidmar, J. Demšar, "O nekaterih statističnih vprašanjih v zvezi s primerjavo postopkov in modelov strojnega učenja", *Informacijska družba IS'2000*, IJS, Ljubljana, 2000, str. 177-181.

[8] J. H. Zar, *Biostatistical Analysis*, Upper Saddle River (New Jersey): Prentice Hall International, 1999.

[9] Python. Dostopno na: http://www.python.org/.