Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Miran Levar

Pregled in primerjava tehnik sočasnega uvrščanja v več razrednih spremenljivk

diplomsko delo

univerzitetni študijski program prve stopnje
računalništvo in informatika

Mentor: prof. dr. Blaž Zupan

Ljubljana 2012

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Univerza *v Ljubljani*
Fakulteta *za računalništvo in informatiko*

Tržaška 25
1001 Ljubljana, Slovenija
telefon: 01 476 84 11
01 476 83 87
faks: 01 426 46 47
01 476 87 11
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00034/2012
Datum: 12.04.2012

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

Kandidat: **MIRAN LEVAR**

Naslov: **PREGLED IN PRIMERJAVA TEHNIK SOČASNEGA UVRŠČANJA V VEČ RAZREDNIH SPREMENLJIVK**

A REVIEW AND COMPARISON OF MULTI-TARGET CLASSIFICATION TECHNIQUES

Vrsta naloge: Diplomsko delo univerzitetnega študija prve stopnje

Tematika naloge:

Tehnike gradnje modelov uvrščanja iz podatkov, ki vsebujejo več razrednih spremenljivk, lahko v teoriji izkoristijo njihovo medsebojno povezavo in so pri tem bolj uspešne od tehnik, ki predpostavijo neodvisnost med razrednimi spremenljivkami in za tovrstne probleme zgradijo ločene napovedne modele. V diplomski nalogi preglejte omenjeno področje, izberite zanj reprezentativne tehnike, te po potrebi reimplementirajte in v eksperimentalni študiji preverite veljavnost zgornje hipoteze.

Mentor:

prof. dr. Blaž Zupan

Fakulteta za računalništvo in informatiko
UNIVERZA V LJUBLJANI
2

Dekan:

prof. dr. Nikolaj Zimic

Izjava o avtorstvu diplomskega dela

Spodaj podpisani Miran Levar, z vpisno številko **63090369**, sem avtor diplomskega dela z naslovom:

Pregled in primerjava tehnik sočasnega uvrščanja v več razrednih spremenljivk

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom prof. dr. Blaža Zupana
- 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 18. septembra 2012 Podpis avtorja:

Zahvaljujem se mentorju prof. dr. Blažu Zupanu za ponujeno priložnost, za vodenje pri izdelavi diplomske naloge in za vso znanje, ki mi ga je posredoval.

Vsem članom Laboratorija za bioinformatiko se zahvaljujem za vse nasvete, pomoč in odlično vzdušje v laboratoriju. Še posebej se zahvaljujem Juretu Žbontarju za pomoč pri razvoju in usmerjanje k znanstveni metodi.

Zahvaljujem se tudi družini, ki me je podpirala vsa leta študija

Kazalo

Povzetek

Da bi odgovorili na vprašanje, ali sočasno uvrščanje v več razrednih spremenljivk uvršča bolje kot uvrščanje v vsako od razrednih spremenljivk neodvisno od preostalih, smo implementirali nekaj tehnik, ki podpirajo sočasno uvrščanje v več spremenljivk. Izbrali smo drevesa za razvrščanje v skupine, nevronske mreže in metodo delnih najmanjših kvadratov. Implementacije teh tehnik smo zbrali v modulu *Orange Multitarget*, dodatku za programski paket *Orange*, sicer odprtokodnem ogrodju za strojno učenje in odkrivanje znanj iz podatkov. Uspešnost implementiranih tehnik smo testirali na mnogih naborih podatkov, tako večznačnih kot tudi naborih z več večrazrednimi spremenljivkami in rezultate teh tehnik primerjali z rezultati standardnih tehnik uvrščanja. Rezultati so pokazali, da standardne tehnike in tehnike za sočasno uvrščanje v več razrednih spremenljivk dosegajo primerljive rezultate in da povečana korelacija med razrednimi spremenljivkami ne izboljša rezultatov tehnik sočasnega uvrščanja v primerjavi s standardnimi tehnikami.

Ključne besede

sočasno uvrščanje v več razrednih spremenljivk, drevesa za razvrščanje v skupine, vrednotenje, strojno učenje

Abstract

Does simultaneous classification of multiple target variables perform better than building a classifier for each of the target variables independently? To answer this question we implemented a set of classification techniques for multi-target classification and integrated them into *Orange Multitarget*, an add-on for *Orange*, an open-source machine learning framework. Performance of both multi-target (clustering trees, neural networks, PLS) and single-target techniques (e.g. random forests) was tested on multiple datasets, which included datasets with binary class variables and datasets with multinomial class variables. The results do not show an advantage for either of the techniques. We have also observed that increased correlation between class variables does not increase the performance of multi-target techniques when compared to single-target techniques.

Keywords

multi-target classification, clustering trees, evaluation, machine learning

Poglavje 1

Uvod

Sočasno uvrščanje v več razrednih spremenljivk je eno izmed področij, ki je zadnje čase vse bolj aktualno tudi zaradi naraščajoče kompleksnosti podatkov z različnih področij. Primera takih področij sta bioinformatika in računska kemija. Programski paket *Orange*, odprtokodno okolje za strojno učenje, odkrivanje znanj iz podatkov in vizualizacijo podatkov, ki ga razvijamo na Univerzi v Ljubljani, smo zato želeli razširiti s tehnikami za sočasno uvrščanje v več razrednih spremenljivk. V pričujoči diplomski nalogi smo želeli tudi odgovoriti na vprašanje, ali lahko s hkratnim napovedovanjem večih razrednih spremenljivk ustvarimo boljše napovedi, kot če za vsako razredno spremenljivko izdelamo ločeni napovedni model in pri tem predpostavimo neodvisnost razrednih spremenljivk. V okviru raziskovanja te domneve je bil razvit dodatek *Orange Multitarget*, ki uporabnikom ponuja implementacije tehnik za sočasno uvrščanje v več razrednih spremenljivk in primerne mere za vrednotenje ustvarjenih napovedi. Izvedena je bila tudi eksperimentalna študija, ki je skušala odgovoriti na prej zastavljeno vprašanje in izmerila uspešnost implementiranih tehnik na različnih naborih podatkov.

Z napovedovanjem večih razrednih spremenljivk so se pričeli ukvarjati raziskovalci, ki so pripisovali oznake tekstovnim dokumentom [33]. Razlika med večznačnimi podatki in podatki z več večrazrednimi spremenljivkami je prikazana v tabeli 1.1. Kasneje se je večznačen pristop k podatkom ši-

z_1	z_2	o_1	o_2	o_3
1	0	0	0	1
2	2	1	1	0
0	2	1	0	1
0	1	0	1	1

z_1	z_2	r_1	r_2	r_3
1	0	2	0	1
2	2	1	2	0
0	2	2	1	1
0	1	0	1	2

Tabela 1.1: Preprosta primera podatkovnih naborov z več razrednimi spremenljivkami. Na levi je večznačen nabor podatkov, pri katerem so razredne spremenljivke binarne (imenovane tudi oznake), na desni je podatkovni nabor z več večrazrednimi spremenljivkami. Večznačnim podatkovnim naborom zaradi redkosti matrike razrednih spremenljivk tipično za vsak primer naštejemo samo pripadajoče oznake (npr. o_1, o_3 za tretji primer).

ril tudi na druga področja, od kategorizacije glasbe in spletnih strani do uvrščanja proteinskih funkcij in napovedovanja reaktivnosti kemijskih spojin [10, 30, 31]. Razvila sta se dva pristopa k obravnavi večznačnih naborov podatkov [33]. Pri prvem se skuša podatke pretvoriti v obliko, ki jo lahko modeliramo s standardnimi tehnikami nadzorovanega učenja, drugi pristop pa je razvoj novih tehnik ali razširitev obstoječih v tehnike, ki omogočajo večznačno uvrščanje.

Razvoj tehnik za večznačno razvrščanje se je torej razcvetel, nismo pa zasledili veliko objav, ki bi preučevale sočasno uvrščanje večih večrazrednih spremenljivk [36]. Zdi se nam, da je uvrščanje v več večrazrednih spremenljivk logičen korak naprej od večznačnega uvrščanja. Ni si namreč težko predstavljati podatkov, pri katerih bi hkrati želeli napovedovati več stvari. Primera takšnih naborov podatkov sta napovedovanje števila sončevih izbruhov različnih kategorij in ugotavljanje intenzivnosti odziva celic na različne učinkovine pri različnih testih. Morda bi kdo pomisli, da bi lahko tudi vsako od teh razrednih spremenljivk napovedovali posebej ali pa problem pretvorili v večznačnega. Oboje je mogoče, a v prvem primeru se odpovemo dodatni informaciji, ki jih lahko prispevajo ostale razredne spremenljivke, v drugem primeru pa povečujemo kompleksnost podatkov in morda onemogočamo teh-

nike, ki bi dobro delovale ravno na večrazrednih podatkih.

V prejšnjem odstavku omenimo dodatne informacije, ki mogoče obstajajo v podatkih z več razrednimi spremenljivkami. Tudi osnovno vprašanje namiguje, da lahko obstaja pri sočasnemu napovedovanju večih razrednih spremenljivk nekaj, kar bi izboljšalo uspešnost napovedi in do česar ne bi imeli dostopa, če bi napovedovali vsako razredno spremenljivko neodvisno od preostalih. Če bi namreč bile vse razredne spremenljivke med sabo neodvisne, bi ne bilo nobene razlike, če bi jih napovedovali vsako posebej ali vse naenkrat. Če pa so razredne spremenljivke med sabo povezane, bi te povezave lahko odkrili le s hkratnim napovedovanjem razrednih spremenljivk. Posledično se vprašanje, ki smo ga postavili na začetku, spremeni v bolj konkretno – ali povezanost med razrednimi spremenljivkami vpliva na uspešnost tehnik za sočasno uvrščanje v več razrednih spremenljivk.

V diplomski nalogi najprej predstavimo standardne tehnike uvrščanja in nato tehnike za sočasno uvrščanje v več razredov. Predstavimo tudi možnosti za razširitve standardnih tehnik. Nadaljujemo z opisom metod za vrednotenje uspešnosti tehnik in mer korelacije med spremenljivkami. Sledi predstavitev dodatka *Orange Multitarget*, kjer opišemo značilnosti in posebnosti implementiranih tehnik. V eksperimentalni primerjavi metod najprej opišemo uporabljene nabore podatkov, nato postopek testiranja in na koncu predstavimo rezultate. V sklepnih ugotovitvah povzamemo nalogo, podamo zaključke ter možnosti nadaljnjega razvoja in izboljšav.

Poglavje 2

Metode

V poglavju so najprej predstavljene vse uporabljene tehnike uvrščanja, najprej standardne tehnike in nato še tehnike za sočasno uvrščanje v več razrednih spremenljivk. Sledijo jim mere za vrednotenje zgrajenih napovednih modelov. Poglavje se zaključi z opisom grafov kritičnih razlik in mer korelacije med spremenljivkami.

2.1 Standardne tehnike uvrščanja

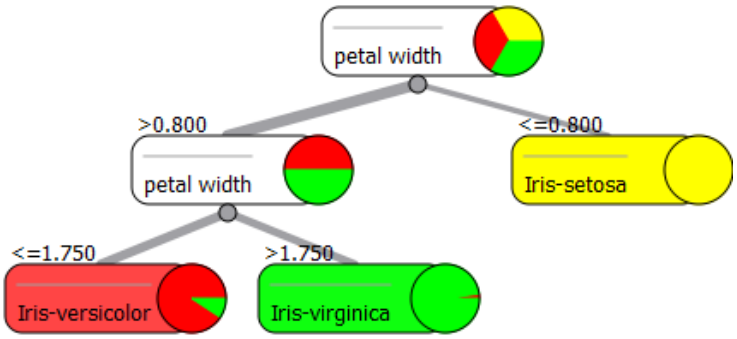
S terminom *standardne tehnike uvrščanja* v tem delu označujemo tehnike, ki so namenjene uvrščanju v eno razredno spremenljivko. Pri nadzorovanem strojnem učenju na podlagi predhodno znanih z značilkami opisanih primerov zgradimo model, ki nov primer glede na vrednosti njegovih značilk uvrsti v pripadajoči razred. Glede na število razredov v razredni spremenljivki ločimo binarne razredne spremenljivke in večrazredne (angl. *multiclass*) spremenljivke. Prve vsebujejo dva možna rezultata, kot na primer ugotavljanje malignosti tumorja ali filtriranje vsiljene elektronske pošte. Večrazredne spremenljivke vsebujejo več možnih rezultatov. Problemi večrazrednega uvrščanja so na primer prepoznavanje števk iz rokopisa ali uvrščanje v redove živali.

2.1.1 Večinski klasifikator

Večinski klasifikator (angl. *majority classifier*) je preprosta metoda, ki primer uvrsti v najbolj pogost razred oziroma najbolj pogosto vrednost razredne spremenljivke. Ker metoda ne upošteva vrednosti neodvisnih spremenljivk, bi jo težko označili za uspešno, lahko pa jo uporabimo kot izhodišče za primerjavo z ostalimi tehnikami. Od preostalih tehnik seveda pričakujemo, da bodo z gradnjo modela uspele razredno spremenljivko napovedati bolje kot večinski klasifikator.

2.1.2 Odločitvena drevesa

Odločitvena drevesa (angl. *top down induction of decision trees*) zgradijo model v obliki drevesa iz podanih podatkov. Vsako nekončno vozlišče drevesa predstavlja eno izmed značilk, končna vozlišča – listi – predstavljajo vrednosti razredov (slika 2.1). V vsakem nekončnem vozlišču je množica vhodnih podatkov ločena glede na vrednosti izbrane značilke v vozlišču in pripadajoče podmnožice so kot podatki podane potomcem vozlišča.



Slika 2.1: Primer odločitvenega drevesa – drevo, ki uvršča irise, je dvakrat ločeno po širini cvetnih listov. V tortnih diagramih je prikazano razmerje razredov, z belo so predstavljena vozlišča, z barvami pa listi.

Osnovni algoritem za grajenje odločitvenih dreves – *ID3*, ki ga je predla-

gal Quinlan leta 1979 [24], najprej preveri, če vsi primeri pripadajo enemu razredu, če, se gradnja ustavi in ustvarjen je list, ki vhodne primere uvrsti v njihov večinski razred. Če množica primerov ni čista, algoritem oceni ustreznost razcepitev te množice in zato izbere značilko z najboljšo oceno. Quinlan je pri svojem algoritmu za mero uspešnosti uporabljal informacijski prispevek značilke, ki je definiran kot prispevek informacije značilke k določitvi vrednosti razredne spremenljivke. Algoritem nato glede na vrednosti izbrane značilke oblikuje podmnožice ter te posreduje na novo ustvarjenim vozliščem. Postopek se rekurzivno ponovi v vsakem od ustvarjenih vozlišč.

Quinlan je leta 1993 objavil posodobljeno verzijo algoritma, ki ga je poimenoval *C4.5* [23]. Med pomembnimi posodobitvami je uporaba razmerja informacijskega prispevka za izbor značilk, ki je določen kot kvocient med informacijskim prispevkom značilke in količino informacije (entropija) značilke. S to razširitvijo mera ni več pristrana do značilk z veliko vrednostmi. Posodobljen algoritem omogoča tudi obravnavo zveznih značilk. Pri njih algoritem preuči vse možne binarne razcepe in uporabi tistega, ki je najbolje ocenjen.

Pri gradnji odločitvenih dreves je pomembno tudi rezanje. Rezanje naprej se izvede, ko gradnjo ustavimo, ker smo zadostili nekemu ustavitvenemu pogoju. Ustavitveni pogoj je pogosto čistost primerov (vsi ali večina primerov pripadajo istemu razredu), lahko pa je to tudi pomanjkanje (dobrih) značilk ali pomanjkanje učnih primerov za zanesljivo nadaljevanje gradnje drevesa. Ko je drevo že zgrajeno, lahko opravimo še naknadno rezanje (angl. *post-pruning*), za kar obstaja mnogo algoritmov, ki pa jih ne bomo obravnavali.

2.1.3 Naključni gozdovi

Naključni gozdovi (angl. *random forests*) so tako imenovana ansambelska (angl. *ensemble*) tehnika. Ansambli klasifikatorjev so tehnike, ki namesto enega modela na učnih podatkih zgradijo več modelov. Za uvrščanje testnega primera vsak model izdela svojo napoved, nato pa z glasovanjem med modeli določimo razred, v katerega bomo primer uvrstili. Enako delujejo na-

ključni gozdovi – klasifikator, ki ga množično gradijo, so odločitvena drevesa; predlagal jih je Breiman [5].

Naključni gozdovi spremenijo postopek gradnje drevesa tako, da vanj vnesejo naključnost. Učni podatki vsakega drevesa so z metodo stremena (angl. *bootstrapping*) zajeta podmnožica celotne učne množice (metoda stremena iz podatkov ustvari podmnožico enake velikosti, primeri so izbrani naključno s ponavljanjem). Med delovanjem algoritma odločitvenih dreves se v vsakem vozlišču posebej izbere še podmnožica značilk, ki jih nato algoritem oceni. Velikost podmnožice značilk je praviloma bistveno manjša od števila vseh značilk, pogosto je koren tega števila. Dreves pri uporabi naključnih gozdov ne režemo.

2.1.4 Metoda delnih najmanjših kvadratov

Metoda delnih najmanjših kvadratov (angl. *partial least squares - PLS*) je pravzaprav množica istovrstnih metod. Prve je v šestdesetih letih prejšnjega stoletja razvil Herman Wold, z razvojem pa je nadaljeval tudi njegov sin Svante Wold. Obširne reference njunih del in tudi del drugih se nahajajo v članku Boulesteix in Strimmerja [3], po katerem tudi mi povzemamo *PLS*. Metoda skuša modelirati povezave med značilkami in razrednimi spremenljivkami s preslikavo značilk v nov prostor, ki najbolje modelira varianco značilk in njihovo korelacijo z razrednimi spremenljivkami [13]. Komponente, ki predstavljajo bazo preslikave, so tiste z največjo kovarianco z razrednimi spremenljivkami.

Ideja *PLS* je poiskati razcepa matrike značilk $\mathbf{X}$ velikosti $n \times p$ in matrike razrednih spremenljivk $\mathbf{Y}$ velikosti $n \times q$. Matriki morata vsebovati centrirane vrednosti, zato najprej od vrednosti primerov posamezne značilke odštejemo povprečno vrednost te značilke. V enačbah razcepa 2.1 je $\mathbf{T}$ matrika latentnih komponent velikosti $n \times c$, $\mathbf{P}$ (velikosti $p \times c$) in $\mathbf{Q}$ (velikosti $q \times c$) sta matriki koeficientov ter $\mathbf{E}$ (velikosti $n \times p$) in $\mathbf{F}$ (velikosti $n \times q$) matriki napak; c je število komponent, v katere algoritem preslika originalni množici. Če so upoštevane vse komponente, sta matriki napak $\mathbf{E}$ in $\mathbf{F}$ enaki nič.

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{P}^T + \mathbf{E} \\ \mathbf{Y} &= \mathbf{T}\mathbf{Q}^T + \mathbf{F} \end{aligned} \tag{2.1}$$

Matrika $\mathbf{T}$ je linearna transformacija matrike $\mathbf{X}$, $\mathbf{T} = \mathbf{X}\mathbf{W}$, kjer je $\mathbf{W} \in \mathbb{R}^{p \times c}$. Vrednosti matrike uteži so pridobljene z enim od algoritmov *PLS*, ki transformacijo izvede. Z uporabo postopka najmanjših kvadratov se izračuna matrika $\mathbf{Q}^T = (\mathbf{T}^T\mathbf{T})^{-1}\mathbf{T}^T\mathbf{Y}$, s katero lahko izračunamo tudi matriko regresijskih koeficientov $\mathbf{B} = \mathbf{W}\mathbf{Q}^T$. Matrika regresijskih koeficientov $\mathbf{B}$ je del modela $\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{F}$. Napoved novega primera $\hat{\mathbf{y}}'_\mathbf{0}$ izračunamo z enačbo 2.2, pri kateri so s črtico označene necentrirane vrednosti. $\mathbf{x}'_\mathbf{0}$ je vhodni vektor primera, ki mu najprej odštejemo povprečja značilk in ga pomnožimo z matriko regresijskih koeficientov $\mathbf{B}$. Izračunanim vrednostim še prištejemo povprečja razrednih spremenljivk.

$$\hat{\mathbf{y}}'_\mathbf{0} = \frac{1}{N}\sum_{i=0}^{N}\mathbf{y}'_\mathbf{i} + \mathbf{B}^T(\mathbf{x}'_\mathbf{0} - \frac{1}{N}\sum_{i=0}^{N}\mathbf{x}'_\mathbf{i}) \tag{2.2}$$

Najpomembnejše pri metodi delnih najmanjših kvadratov je torej iskanje matrike uteži $\mathbf{W}$. Za to je bilo razvitih mnogo algoritmov, ki se v osnovi ločijo v dva sklopa, na tiste, ki napovedujejo samo eno razredno spremenljivko naenkrat (*PLS1*) in na tiste, ki jih napovedujejo več (*PLS2*). Kot najbolj znane iz sklopa slednjih avtorja članka izpostavita algoritem *NIPALS – Nonlinear Iterative Partial Least Squares*, algoritem *Kernel-PLS* in algoritem *SIMPLS – Statistically Inspired Modification of PLS*. Obravnava teh algoritmov presega okvire te diplomske naloge.

2.1.5 Logistična regresija

Logistična regresija (angl. *logistic regression*) je klasifikacijska tehnika, ki se uporablja za napovedovanje verjetnosti binarnih razrednih spremenljivk [17]. Njena osnova je logistična funkcija (sigmoida, 2.3), ki ji podamo linearno kombinacijo uteži in neodvisnih spremenljivk $z = \theta_0 + \theta_1 x_1 + \cdots + \theta_k x_k$.

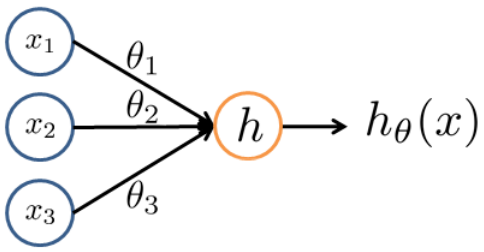
Rezultat izračuna je verjetnost, da je vrednost razredne spremenljivke ena, $P(y = 1|x_1, \ldots, x_k) = P(\mathbf{x}) = f(z)$:

$$f(z) = \frac{1}{1 + e^{-z}} \tag{2.3}$$

Uteži modela so ponavadi izračunane z metodo največjega verjetja, pogosto se uporablja tudi regularizacija. Če želimo z logistično regresijo napovedovati večrazredne spremenljivke, lahko uporabimo tehniko eden proti vsem in za vsako od vrednosti razredne spremenljivke zgradimo svoj model ter nato napovedi združimo.

2.1.6 Nevronske mreže

Nevronskih mreže oz. točneje umetne nevronske mreže (angl. *artificial neural networks*) so tehnika za modeliranje podatkov, ki jo lahko ob uporabi primerne aktivacijske funkcije uporabimo za uvrščanje. Ker je ta aktivacijska funkcija zelo pogosto logistična funkcija, jih lahko pojmujemo tudi kot razširjeno logistično regresijo.

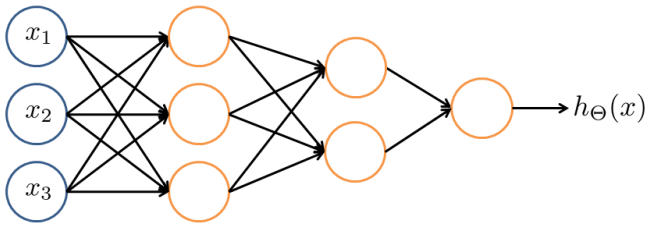


Slika 2.2: Model perceptrona.

Matematični model nevrona, imenovan perceptron, je predlagal Frank Rosenblatt leta 1956 [27] (slika 2.2). Vektor vhodnih vrednosti označenih z x je podan aktivacijski funkciji h. Znotraj funkcije je x skalarno pomnožen z vektorjem uteži θ in s sklarnim produktom funkcija izračuna končno vrednost perceptrona $h_\theta(x)$. Pri učenju se izračunana vrednosti primerja z vrednostjo

učnega primera in uteži se popravijo glede na storjeno napako pri posamezni vhodni vrednosti. To se ponovi za vse učne primere in celoten postopek se ponavlja, dokler perceptron ne ustvari zadosti točnega modela učne množice. Bistvo modeliranja je torej iskanje ustreznih uteži funkcije h_θ.

Obstaja več arhitektur nevronskih mrež, a pri nadzorovanem učenju nas zanimajo predvsem naprej usmerjene več-nivojske nevronske mreže. Te so sestavljene iz treh nivojev: vhodnega, skritega in izhodnega. Vhodni nivo predstavljajo vrednosti posameznega primera (vektor x), izhodni nivo je določen s tipom in številom razrednih spremenljivk. Skriti nivo je sestavljen iz izbranega števila nivojev in vsakega od teh skritih nivojev sestavlja izbrano število perceptronov. Za preprost primer predpostavimo, da je skriti nivo sestavljen iz enega nivoja perceptronov. V vsakega izmed njih so vezani vsi vhodni nevroni, izhodi skritih perceptronov pa predstavljajo vhode perceptronov izhodnega nivoja. Če skritemu nivoju dodajamo nivoje perceptronov, so vhodi le-teh izhodi prejšnjega nivoja in njihovi izhodi vhodi naslednjemu nivoju. Na sliki 2.3 je primer tro-nivojske mreže (vhodnega nivoja ne štejemo) z dvema skritima nivojema.



Slika 2.3: Primer tro-nivojske nevronske mreže s skritima nivojema velikosti tri in dve.

Algoritem več-nivojskih mrež deluje podobno kot osnovni algoritem perceptrona. Najprej se računajo vrednosti funkcij posameznih perceptronov od začetnega nivoja do končnega, vsak perceptron izvrši osnovni algoritem in iz svojih uteži ter vhodnih vrednost izračuna in preda naprej izhodno vrednost. Izračun izhodnega nivoja se preveri z vrednostjo razredne spremenljivke in izračuna se napaka, ki jo je storila funkcija v perceptronu izhodnega nivoja.

Ta napaka se nato utežena z algoritmom vzvratnega razširjanja napake prenese v smeri od izhodnega do prvega skritega nivoja. Napaka posameznega perceptrona v skritem nivoju je utežena vsota napak perceptronov, ki jim sicer pošilja svoje rezultate. S prejeto vrednostjo napake vsak perceptron popravi pripadajoče uteži θ in iteracija algoritma se zaključi, ko so napake razširjene do prvega skritega nivoja. Algoritem izvaja iteracije dokler ne doseže želene točnosti ali pa se uteži ne ustalijo in s tem zaključijo gradnjo modela.

Ker so v vseh perceptronih enake aktivacijske funkcije, bi vsi perceptroni posameznega nivoja z enakimi utežmi vedno vračali enake vrednosti. Da se temu izognemo, so uteži na začetku delovanja algoritma naključno inicializirane z majhnimi vrednostmi. Pomembna sta še dva dodatka osnovnemu algoritmu. Prvi služi zmanjšanju pristranosti modela in vsakemu nivoju doda še dodatno vhodno vrednost x_0, ki je vedno ena in dodatno utež θ_0. Druga izboljšava pa zmanjšuje preveliko prilagajanje učnim podatkom tako, da ob izračunu napake izhodnega nivoja od le-te še odšteje vsoto kvadratov vseh uteži (razen θ_0) pomnoženo s parametrom λ.

Opisane nevronske mreže z enim izhodnim perceptronom služijo napovedovanju binarnih razrednih spremenljivk. Če bi želeli napovedovati večvrednostne spremenljivke, je potrebno za vsako vrednost dodati še en izhodni perceptron. Tako s tehniko eden proti vsem vsak od perceptronov uvršča v eno izmed vrednosti, skupaj pa ustvarijo večrazredni klasifikator.

2.2 Tehnike za sočasno uvrščanje v več razrednih spremenljivk

Tehnike za sočasno uvrščanje v več razrednih spremenljivk se od standardnih tehnik razlikujejo predvsem v tem, da, kakor nakaže že ime, hkrati napovedujejo več razrednih spremenljivk. Kot pri standardnih tehnikah, je tudi pri sočasnem uvrščanju v več razrednih spremenljivk tipičen problem nadzorovanega učenja predstavljen z učno množico, na podlagi katere zgradimo model

za uvrščanje novih primerov. Tudi pri sočasnem uvrščanju v več razrednih spremenljivk obstaja delitev na binarne razredne spremenljivke z dvema razredoma ter na večrazredne spremenljivke. Če so vse razredne spremenljivke binarne, imamo podatke, ki spadajo v posebno podskupino sočasnega uvrščanja v več razrednih spremenljivk – večznačne tehnike uvrščanja (angl. *multi-label classification techniques*). Naše rešitve in pristopi želijo zajeti celotno področje sočasnega uvrščanja, zato so tudi izbrane metode prilagojene temu. Hkrati se zavedamo, da veliko napredka na področju izvira iz večznačnega uvrščanja in da tudi mi nekatere naše metode in pristope prevzemamo iz tega področja.

2.2.1 Drevesa za razvrščanje v skupine

Odločitvena drevesa so tipično razumljena kot tehnike za uvrščanje – listi vsebujejo razrede in poti od korena do listov vsebujejo zadostne informacije za uvrstitev. Hendrik Blockeel in sod. v članku, kjer opisujejo drevesa za razvrščanje v skupine (angl. *clustering trees*) [2], pričnejo s tezo Pata Langleya, da lahko vsa vozlišča drevesa razumemo kot skupine (angl. *clusters*), celotno drevo pa kot hierarhijo ali taksonomijo nabora podatkov. Zaradi tega Langley pojmuje tako tehnike razvrščanja v skupine kot tudi tehnike učenja konceptov (nadzorovano učenje) kot instanci enake splošne tehnike – odkrivanja hierarhije konceptov (angl. *induction of concept hierarchies*). Avtorji članka tako iz odločitvenih dreves razvijejo tehniko dreves za razvrščanje v skupine. Drevo v listih ne vsebuje več razredov, temveč vsako njegovo vozlišče, tudi listi, vsebuje množico oz. skupino primerov. Postopek učenja je enak kot pri klasičnih odločitvenih drevesih, drugačni pa sta izbira in ocenjevanje značilk. Avtorji predpostavijo obstoj mere $d(e_1, e_2)$, ki lahko izračuna razdaljo med dvema primeroma iz učne množice in obstoj funkcije $p(E)$, ki lahko izračuna prototip iz množice primerov. Ta prototip mora biti predstavljen enako kot posamezen primer, da lahko razdaljo med dvema skupinama C_1 in C_2 določimo kot razdaljo med prototipoma teh skupin $d(p(C_1), p(C_2))$. Učenje dreves za razvrščanje v skupine tako med gradnjo išče značilko, ki

najbolje razloči med skupinami, v katere bodo ob izbiri te značilke primeri razvrščeni.

Takšna drevesa so posledično uporabna tako za nadzorovano kot tudi nenadzorovano učenje. Pri prvem se za merjenje razdalj uporabijo samo vrednosti razrednih spremenljivk, pri nenadzorovanem učenju pa se za izračun razdalj uporabijo vrednosti značilk. Način nadzorovanega učenja je uporaben tudi za regresijo.

Ker avtorji za razvoj uporabljajo prolog zaradi specifičnosti svoje implementacije med gradnjo dreves ne govorijo o izbiri značilk temveč o testih, ki preverijo prisotnost neke kombinacije vrednosti in vedno ali uspejo ali ne uspejo. Zaradi tega so njihove razdelitve vedno binarne. Po besedah avtorjev test ni samo kriterij za odločitev ampak tudi opis ustvarjenih podskupin, kar je skladno z Langleyjevo idejo.

Predlagan algoritem naj bi minimiziral razdalje primerov znotraj skupin in maksimiziral razdalje med skupinami. Kot primerno mero avtorji navedejo evklidsko razdaljo, ki za vrednosti prototipov enostavno uporabi centroide skupin, poudarijo pa, da bi lahko za mero uporabili katerokoli mero razdalje skupaj s prototipno funkcijo, ki je z izbrano mero združljiva.

Za ustavitveni kriterij gradnje drevesa avtorji predlagajo F test: $F = \frac{SS/(N-1)}{(SS_L+SS_R)(N-2)}$. Z njim bi zagotavljali, da se varianca značilno zmanjša z razvrstitvijo v skupine (SS je vsota kvadratov razdalj posameznih primerov skupine od centroida skupine, N je število vseh primerov v skupini). Ker je F test teoretično pravilen le za normalno porazdeljene populacije, je ta ustavitveni kriterij bolj hevristika in ne pravi statistični test.

Še ena izboljšava osnovnega algoritma, ki jo predlagajo, je, da za vzvratno rezanje uporabimo nastavitveno množico (angl. *validation set*). Po gradnji uspešnost drevesa preverimo z nastavitveno množico primerov in vozlišča, ki uspešnost zmanjšujejo, porežemo. S testiranjem pokažejo, da je rezanje načeloma izboljša točnost uvrščanja in zmanjša velikost drevesa, a da je resnično uporabno le pri velikih naborih podatkov, pri manjših pa zmanjšanje učne množice škodi natančnosti in rezultati zaradi naključnih vplivov niso več

zanesljivi.

2.2.2 Razširitve standardnih tehnik za uvrščanje v več razredov

Obstajajo tehnike, ki standardnim klasifikatorjem omogočijo, da ustvarijo model tudi za učne podatke z več razrednimi spremenljivkami. V nadaljevanju sta opisani dve taki tehniki, na koncu pa še ustrezni razširitvi nevronskih mrež in metode delnih najmanjših kvadratov.

Ločeni klasifikatorji

Najosnovnejša razširitvena tehnika so ločeni klasifikatorji (angl. *binary relevance classifier*) [26]. Njihovo ime v izvirniku vsebuje besedo binarni, ker so jih razvili za potrebe večznačnega uvrščanja. Bistvo tehnike je, da za uvrščanje nekega števila razrednih spremenljivk ustvarimo enako število modelov izbrane standardne tehnike, po enega za vsako izmed njih. Metoda ločenih klasifikatorjev predpostavi, da med posameznimi razrednimi spremenljivkami ni medsebojne odvisnosti. Ta predpostavka je problematična, saj se z njo že vnaprej odpovemo delu informacij in avtorji članka zato navajajo, da zaradi tega tehnika v stroki ni priljubljena. Vseeno pa opozarjajo, da tehnike zaradi njene preprostosti in robustnosti ne gre odpisati. Kompleksnost ločenih klasifikatorjev narašča linearno z naraščanjem števila razrednih spremenljivk in ohranja nizko računsko zahtevnost, sploh v primerjavi z drugo zelo preprosto in intuitivno razširitveno tehniko – metodo kombiniranja oznak (angl. *label combination method*) (tudi metoda potenčne množice oznak (angl. *label power-set method*)), ki iz vseh možnih kombinacij oznak ustvari eno razredno spremenljivko, katere razredi so te kombinacije. Z dodajanjem novih oznak metoda kombiniranja oznak dosega eksponentno rast, zgornja meja njene računske zahtevnosti je $O(min(N, 2^L))$, pri čemer je N število primerov in L število razrednih spremenljivk oz. v tem primeru oznak.

Iz ločenih klasifikatorjev avtorji izpeljejo tudi naslednjo razširitveno teh-

niko.

Veriga klasifikatorjev

Veriga klasifikatorjev (angl. *classifier chain*) je tehnika, ki nadgrajuje ločene klasifikatorje in zmanjšuje njihovo veliko pomanjkljivost – predpostavko neodvisnosti. Predlagali so jo Read in sod. [26]. Osnovna ideja veriženja klasifikatorjev je uporaba že uvrščenih razrednih spremenljivk kot značilk za učenje preostalih še neuvrščenih. Ko je takšna veriga modelov zgrajena, se tudi pri uvrščanju vrednosti dodajajo med značilke in služijo kot dodatna informacija. Verige klasifikatorjev tako upoštevajo medsebojno informacijo med razrednimi spremenljivkami in ohranijo preprostost ločenih klasifikatorjev.

Zelo pomemben pri učenju verige klasifikatorjev je vrstni red učenja razrednih spremenljivk. Od njega sta odvisni uspešnost in izraba dodatnih informacij. Avtorji so se tega zavedali, zato so predlagali tudi nadgradnjo osnovnih verig klasifikatorjev.

Ansambel verig klasifikatorjev

Ansambel verig klasifikatorjev (angl. *ensemble of classifier chains*) tako kot ostale ansambelske tehnike zgradi več posameznih modelov, v tem primeru to pomeni več verig klasifikatorjev. Za vnos naključnosti ima osnovna veriga klasifikatorjev že vgrajen parameter vrstnega reda razrednih spremenljivk, avtorji pa predlagajo še, da naj bo vsaka posamezna veriga zgrajena na naključni podmnožici celotne učne množice. Tako zgrajene verige naj se med seboj ne bi ponavljale. Za izbiro končnih vrednosti razrednih spremenljivk se prešteje glasove posameznih modelov in izbere najbolj zastopane vrednosti.

Nevronske mreže

Nevronske mreže lahko problemom z večimi razrednimi spremenljivkami prilagodimo tako, kot smo jih prilagodili napovedovanju večrazrednih spremen-

ljivk – z dodajanjem izhodnih perceptronov. Nevronske mreže, ki napovedujejo več razrednih spremenljivk, imajo število izhodnih perceptronov enako številu razrednih spremenljivk. Če pa so spremenljivke večrazredne, se število izhodnih perceptronov poveča še za število razredov vsake takšne spremenljivke.

Metoda delnih najmanjših kvadratov

Metoda delnih najmanjših kvadratov že sama po sebi ponuja algoritme (prej omenjeni sklop *PLS2*), ki znajo obravnavati več razrednih spremenljivk, zato je za razširitev potrebna samo implementacija teh algoritmov.

2.3 Metode vrednotenja

2.3.1 Mere uspešnosti

Logaritem izgube

Read in sod. [26] kot eno izmed možnih mer uspešnosti predlagajo logaritem izgube (angl. *log loss*). Poročajo, da se od ostalih mer, ki jih uporabljajo ($F1_{macro}$ in točnost), razlikuje po tem, da strožje kaznuje hujše napake in da meri zaupanje oz. verjetnost s katero je bila dana določena napoved in ne same napovedi. Tako so napačne napovedi, ki so dane z nizko verjetnostjo, kaznovane manj kot napačne napovedi podane z visoko verjetnostjo. Za omejitev maksimalne napake in glajenje vrednosti pri zelo nizkih verjetnostih predlagajo, da se napaka omeji z logaritmom obratne vrednosti velikosti učne množice N.

$$LogLoss = \frac{1}{N}\sum_{i=0}^{N} -log(max(p_i, \frac{1}{N})) \tag{2.4}$$

Prvotni logaritem izgube so avtorji zapisali za večznačne podatkovne nabore. Za rabo z večrazrednimi spremenljivkami smo ga razširili, enačba 2.4 predstavlja logaritem izgube za eno razredno spremenljivko, pri kateri je p_i

verjetnost, s katero je klasifikator napovedal dejansko vrednost razreda i-tega primera.

(Relativni) koren srednje kvadratne napake

Koren srednje kvadratne napake 2.5 (angl. *root-mean-square error*) je zelo pogosto uporabljana mera za preverjanje uspešnosti regresijskih metod. Osnova definicije je napaka, ki jo regresijska tehnika napravi in je izračunana kot kvadrat razlike med napovedano $\hat{y}_i$ in dejansko vrednostjo razredne spremenljivke y_i i-tega primera. Vsota vseh napak, ki jih tehnika napravi, je povprečena in še korenjena, da so vrednosti vrnjene v prvotne okvire.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(\hat{y}_i - y_i)^2} \tag{2.5}$$

Ker so napake odvisne od merila, v katerem je razredna spremenljivka, je bil v želji po neodvisnosti od tega merila razvit še relativni koren srednje kvadratne napake [36] (angl. *relative root-mean-square error*). Osnovni formuli je dodana še napaka modela, ki vedno napove povprečno vrednost $\bar{y}$ razredne spremenljivke 2.6.

$$RRMSE = \sqrt{\frac{\sum_{i=0}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=0}^{N}(\bar{y} - y_i)^2}} \tag{2.6}$$

Brierjeva mera

Srednja kvadratne napaka in njene izpeljave so primerne samo za ocenjevanje regresijskih tehnik, zato jih pri merjenju uspešnosti uvrščanja ne moremo uporabiti. Lahko pa uporabimo mero, ki jo je predlagal Brier leta 1950 [6] in je po njem tudi poimenovana (angl. *Brier score*). Predlagana mera je podobna srednji kvadratni napaki, a namesto z vrednostmi napovedi računa z verjetnostmi, s katerimi so bile napovedi podane. V enačbi 2.7 je N število primerov in CV število vrednosti razredne spremenljivke. $\hat{p}_{ij}$ je verjetnost, s

katero je klasifikator napovedal j-ti razred i-tega primera, p_{ij} pa idealna verjetnost napovedi, ki je enaka ena za dejansko vrednost razredne spremenljivke in nič za preostale razrede.

$$Brier = \frac{1}{N}\sum_{i=0}^{N}\sum_{j=0}^{CV}(p_{ij} - \hat{p}_{ij})^2 \tag{2.7}$$

Mera nam torej oceni kvadratno napako napovedi verjetnostne distribucije. Če klasifikator z verjetnostjo ena pravilno napove vse vrednosti, je vrednost Brierjeve mere enaka nič, če pa z verjetnostjo ena napačno napove vse razrede, je vrednost mere dva. Zato Kononenko predlaga, da lahko ocenimo kvaliteto klasifikatorja z $1 - Brier/2$ [18].

2.3.2 Povprečna informacijska vsebina odgovora

Mero uspešnosti, ki hkrati upošteva tako apriorne verjetnosti razredov kot tudi verjetnosti, ki jih vrne klasifikator, predlagata Kononenko in Bratko [19]. Mero sta poimenovala povprečna informacijska vsebina odgovora (angl. *information score*). Problem, na katerega opozorita, je neupoštevanje težavnosti klasifikacijskega problema in ga ponazorita s primerjavo prognoze ponovitve raka na dojki, pri katerem nek model dosega 77% točnost, in problem lokalizacije primarnega tumorja, pri katerem drugi model doseže 44% točnost. Prva točnost se zdi bistveno višja od druge in zato na prvi pogled tudi boljša, a ob preučitvi problemov se izkaže, da pri prvem uvrščamo v dva razreda in je verjetnost večinskega razreda 80%, pri drugem pa uvrščamo v 22 razredov in je verjetnost večinskega razreda 25%. Točnost prvega modela je torej slabša od večinskega klasifikatorja, drugi model pa doseže relativno dobre rezultate.

$$Inf = \frac{1}{N}\sum_{i=0}^{N} Inf_i \quad \text{[bit]} \tag{2.8}$$

Predlagana mera je izračunana kot povprečje informacijske vsebine odgovora posameznega primera 2.8. Pri tem je Inf_i, kakor je razvidno iz

enačbe 2.9, odvisen od velikosti apriorne in napovedane verjetnosti. V enačbi je p_i apriorna verjetnost pravilnega razreda i-tega primera, $\hat{p}_i$ pa napovedana verjetnosti za ta pravilni razred. Rezultat povprečne informacijske vsebine odgovora je v bitih.

$$Inf_i = \begin{cases} -log_2\, p_i + log_2\, \hat{p}_i; & \hat{p}_i \geq p_i \\ -(-log_2(1-p_i) + log_2(1-\hat{p}_i)); & \hat{p}_i < p_i \end{cases} \tag{2.9}$$

Vrednost mere za idealen klasifikator, ki pravilni razred vedno napove z ena, je enaka entropiji razredne spremenljivke. Le-ta predstavlja zgornjo mejo, ki jo informacijska vsebina lahko doseže. Za večinski klasifikator je vrednost informacijske vsebine enaka nič in predstavlja spodnjo mejo uspešnosti klasifikatorjev. Avtorja še predlagata, da je včasih dobro predstaviti informacijsko vsebino odgovora normalizirano z entropijo razredne spremenljivke. Enačba 2.10 predstavlja relativno informacijsko vsebino odgovora ($H(C) = -\sum_{j=0}^{CV} f_j\, log_2\, f_j$ je entropija razredne spremenljivke izračunana za vseh CV vrednosti; f_j je verjetnost j-te vrednosti razredne spremenljivke).

$$RInf = \frac{Inf}{H(C)} * 100 \tag{2.10}$$

2.3.3 Mere uspešnosti za več spremenljivk hkrati

Globalna in povprečna točnost

Za merjenje uspešnosti naborov podatkov z več razrednimi spremenljivkami Bielza in sod. [1] predlagajo globalno in povprečno točnost. Globalna točnost 2.11 izračuna delež napovedi, ki se popolnoma ujemajo z vrednostmi razrednih spremenljivk. Povprečna točnost 2.12 izračuna klasifikacijsko točnost (delež pravilno napovedanih vrednosti) za vsako razredno spremenljivko in jih nato povpreči.

$$
\begin{aligned}
Acc_G &= \frac{1}{N}\sum_{i=1}^{N}\delta(\hat{\mathbf{y}}_\mathbf{i}, \mathbf{y}_\mathbf{i}) \\
\delta(\hat{\mathbf{y}}_\mathbf{i}, \mathbf{y}_\mathbf{i}) &= \begin{cases} 1: & \hat{\mathbf{y}}_\mathbf{i} = \mathbf{y}_\mathbf{i} \\ 0: & sicer \end{cases}
\end{aligned}
\tag{2.11}
$$

$$
\begin{aligned}
\overline{Acc} &= \frac{1}{d}\sum_{j=1}^{d}\frac{1}{N}\sum_{i=1}^{N}\delta(\hat{y}_{ij}, y_{ij}) \\
\delta(\hat{y}_{ij}, y_{ij}) &= \begin{cases} 1: & \hat{y}_{ij} = y_{ij} \\ 0: & sicer \end{cases}
\end{aligned}
\tag{2.12}
$$

Ker naj bi bili globalna točnost preveč stroga in povprečna preveč popustljiva, Read in sod. [26] za boljšo mero točnost označijo točnost, ki so jo iz Jaccardove mere podobnosti izpeljali Boutell in sod. [4]. Povprečni Jaccardovi podobnosti 2.13 so dodali še parameter α, ki ga poimenujejo stopnja odpustljivosti (angl. *forgiveness rate*). Z manjšimi α smo bolj tolerantni do napak, z večjimi pa napake strožje kaznujemo.

$$
Acc_{ML} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|\hat{\mathbf{y}}_\mathbf{i} \cap \mathbf{y}_\mathbf{i}|}{|\hat{\mathbf{y}}_\mathbf{i} \cup \mathbf{y}_\mathbf{i}|}\right)^{\alpha} \tag{2.13}
$$

Problem te mere točnosti, zaradi jasnosti jo imenujemo večznačna točnost, je, da je definirana samo za binarne razredne spremenljivke, saj je le na njih definirana Jaccardova podobnost. Zaradi tega je ta mera uporabna samo za merjenje uspešnosti večznačnih naborov podatkov.

Mikro in makro mera $F1$

Mera $F1$ je definirana samo na binarnih razrednih spremenljivkah. Pogosto je uporabljana pri preverjanju uspešnosti večznačnih naborov podatkov [1, 26]. Osnova izračuna te mere sta priklic in natančnost. Priklic je definiran kot kvocient med pravilno napovedanimi pozitivnimi vrednostmi in vsemi pozitivnimi napovedmi (vključno z napovedmi, ki se izkažejo kot napačne). Natančnost je definirana kot kvocient med pravilno napovedanimi pozitivnimi

vrednostmi in vsemi dejansko pozitivnimi primeri (vključno s tistimi, ki smo jih napovedali napačno). $F1$ je harmonična vsota priklica in natančnosti, to je tudi razlog za 1 v njenem imenu, saj obstajajo še druge mere F, ki pa natančnosti in priklica ne upoštevajo v enaki meri.

Obstajata dve variaciji: $F1_{mikro}$ (2.14) za izračun uporabi na vseh razrednih spremenljivkah izračunani povprečni natančnost in priklic, $F1_{makro}$ (2.15) pa izračuna mero $F1$ za vsako razredno spremenljivko posebej ter jih nato povpreči.

$$F1_{micro} = 2 \cdot \frac{\overline{precision} \cdot \overline{recall}}{\overline{precision} + \overline{recall}} \tag{2.14}$$

$$F1_{macro} = \frac{1}{M} \sum_{j=0}^{M} 2 \cdot \frac{precision_j \cdot recall_j}{precision_j + recall_j} \tag{2.15}$$

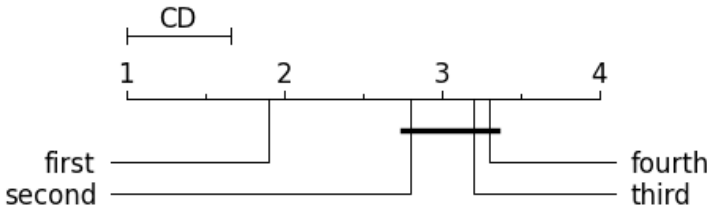
Razširitve standardnih mer

Ker je mer uspešnosti za tehnike uvrščanja v več večrazrednih spremenljivk relativno malo, smo se odločili za prilagoditev rezultatov teh tehnik v obliko, ki jo sprejmejo mere uspešnosti za standardne tehnike uvrščanja. Za to smo uporabili dva pristopa: *povprečno mero* in *sploščeno mero*. Pri prvem izmerimo uspešnost napovedi vsake razredne spremenljivke posebej ter nato te rezultate povprečimo. Druga mera pa matriko rezultatov vseh razrednih spremenljivk splošči v vektor in na tem vektorju s podano mero izmeri uspešnost.

2.4 Prikaz rezultatov – graf kritičnih razlik

Za vizualizacijo primerjave uspešnosti večih klasifikatorjev na več različnih učnih množicah je Demšar [9] razvil graf kritičnih razlik (angl. *critical difference graph*). Na grafu so na oštevilčenem traku prikazani povprečni rangi posameznih tehnik. Z debelejšo črto so povezane vse tehnike med katerimi ni statistično značilnih razlik. Za orientacijo je kritična razlika, ki nam označuje

statistično značilno razliko med posameznimi tehnikami, prikazana v zgornjem levem kotu. Na sliki 2.4 je primer grafa; iz njega lahko sklepamo, da je prva tehnika statistično značilno boljša od preostalih tehnik. Kljub temu, da je druga tehnika nekoliko pred tretjo in četrto, se še vedno od njiju razlikuje za razliko manjšo od kritične in zaradi tega lahko sklepamo le, da ne obstaja dovolj podatkov, ki bi nam omogočili razločitev med temi tremi tehnikami.



Slika 2.4: Primer grafa kritičnih razlik.

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{2.16}$$

Kritična razlika 2.16 je izračunana na podlagi števila klasifikatorjev k in števila učnih množic N. Parameter q_α je vnaprej izračunana kritična vrednost za določeno stopnjo značilnosti testa α (npr. 0.10 ali 0.05) in k. Neka tehnika uvrščanja je od druge boljša s stopnjo značilnosti α, če je razlika med povprečnim rangom tehnik večja od kritične razlike.

2.5 Mere korelacije med spremenljivkami

Če želimo govoriti o odvisnosti dveh spremenljivk, je dobro to odvisnost tudi izmeriti. Izbrana mera je odvisna od tipa spremenljivk, nas zanimajo predvsem povezave med razrednimi spremenljivkami in zato uporabljamo mere, ki izmerijo povezanost dveh spremenljivk z nominalnimi vrednostmi.

Najbolj pogosto je porazdelitev vrednosti dveh nominalnih spremenljivk podana s kontingenčno tabelo, katere celice vsebujejo število pojavitev posamezne kombinacije vrednosti spremenljivk. Te vrednosti se preprosto pre-

računajo v frekvence, le delimo jih s številom vseh primerov. Primer kontingenčne tabele 2.1 velikosti 2×2 lahko vidimo spodaj.

	za	proti	$\sum$
moški	55	5	60
ženske	25	15	40
$\sum$	80	20	100

Tabela 2.1: Preprost primer kontingenčne tabele.

2.5.1 Mere na osnovi χ^2

Pearsonov χ^2 test je eden izmed osnovnih statističnih testov, s katerimi lahko preverimo neodvisnost dveh nominalnih spremenljivk. Ničelna hipoteza testa je, da spremenljivki nista povezani. Za testiranje te hipoteze test primerja teoretično porazdelitev vrednosti, ki bi držala ob ničelni predpostavki, z opazovano porazdelitvijo. Enačba 2.17 predstavlja izračun statistike χ^2; f_{ij} je opazovana frekvenca, f'_{ij} pa teoretična frekvenca kombinacije i-te vrednosti prve in j-te vrednosti druge spremenljivke. Teoretična frekvenca je zmnožek frekvenc i-te in j-te vrednosti spremenljivk: $f'_{ij} = f_i \cdot f_j$. R je število vrednosti prve spremenljivke in tudi število vrstic, C je število vrednosti druge spremenljivke in tudi število stolpcev kontingenčne tabele, saj lahko frekvence oz. verjetnosti najlažje preberemo ravno iz te tabele.

$$\chi^2 = \sum_{i=0}^{R} \sum_{j=0}^{C} \frac{(f_{ij} - f'_{ij})^2}{f'_{ij}} \tag{2.17}$$

Izračunano eksperimentalno vrednost statistike χ^2 je potrebno še ovrednotiti, zato iz porazdelitve χ^2 za izbrano stopnjo značilnosti α in število prostostnih stopenj določeno z $dof = (R-1)(C-1)$ preberemo kritično vrednost statistike. Če je eksperimentalna vrednost večja od kritične, lahko ničelno domnevo zavrnemo in pri stopnji značilnosti α sprejmemo alternativno hipotezo – da sta spremenljivki povezani. Zaloga vrednosti statistike χ^2 je $[0, N(k-1)]$, pri čemer je N število vseh testiranih primerov in $k = min(R, C)$.

$$\phi = \sqrt{\frac{\chi^2}{N}} \tag{2.18}$$

Problem testa χ^2 je, da njegovih rezultatov ne moremo primerjati med sabo in trditi, da sta določeni spremenljivki bolj oz. manj povezani, ker je vrednost njunega testa χ^2 večja oz. manjša od drugih. Zaradi tega so bili na podlagi testa razviti različni koeficienti korelacije, ki naj bi nam to omogočili. Prvi od teh je koeficient ϕ, ki ga je predlagal Pearson, izračunamo ga z enačbo 2.18 in njegova zaloga vrednosti je med 0 in $\sqrt{k-1}$. Za spremenljivke z binarnimi vrednostmi oz. kontingenčne tabele velikosti 2×2 obstaja postopek računanja tega koeficienta, ki temelji na Pearsonovem r. Rezultat tega izračuna je enak koeficientu r dveh binarnih spremenljivk. Izračun je prikazan v enačbi 2.19, f_{ij} so frekvence i-tega stolpca in j-te vrstice. Zaloga vrednosti koeficienta je med minus ena in ena, kar ponazarja korelacijo oz. obratno korelacijo.

$$\phi = \frac{f_{00}f_{11} - f_{01}f_{10}}{\sqrt{(f_{00}+f_{01})(f_{10}+f_{11})(f_{00}+f_{10})(f_{01}+f_{11})}} \tag{2.19}$$

Naslednji je Cramérjev koeficient V, ki je definiran za vse velikosti kontingenčnihi tabel, v primeru velikosti 2×2 pa je enak ϕ. Izračunan je z enačbo 2.20 in zavzame vrednosti na intervalu $[0, 1]$.

$$V = \sqrt{\frac{\chi^2}{N(k-1)}} \tag{2.20}$$

Tretja mera je kontingenčni koeficient C, ki je izračunan z enačbo 2.21. Zaloga njegove vrednosti je $[0, \frac{k-1}{k}]$, zato je predlagano še normiranje te enačbe s $\frac{k-1}{k}$, da se zaloga vrednosti spremeni $[0, 1]$.

$$C = \sqrt{\frac{\chi^2}{N+\chi^2}} \tag{2.21}$$

Reference avtorjev teh mer in testa samega so povzete v članku Goodmana in Kruskala [12]. Na tem mestu omenjamo članek zato, ker avtorja

članka mere, ki temeljijo na χ^2, označita za neuporabne. Med drugim navajata, da nista uspela najti prepričljive objave, ki bi argumentirala rabo teh mer za primerjavo povezanosti večih parov spremenljivk. Za dve vrednosti koeficienta C, avtorja izbereta vrednosti .56 in .24, po njunem ne moremo trditi, da sta spremenljivki prvega rezultata bolj koreliran kot spremenljivki drugega. Zato predlagata svoje mere, ena izmed njih je opisana v naslednjem razdelku.

2.5.2 Mera λ

Mera λ Goodmana in Kruskala meri razmerje napak, ki jih napravimo, če za uvrščanje primera v eno spremenljivko uporabimo znanje o drugi spremenljivki v primerjavi z naključnim uvrščanjem. Napaka, ki jo napravimo, če izbiramo naključne vrednosti, je $1 - \frac{1}{2}(f_{\bullet m} + f_{m\bullet})$, $f_{\bullet m}$ je frekvenca večinske vrednosti prve spremenljivke in $f_{m\bullet}$ frekvenca večinske vrednosti druge spremenljivke. Napaka, ki jo napravimo pri naključnem uvrščanju spremenljivke, če poznamo drugo spremenljivko, pa je $1 - \frac{1}{2}(\sum_{i=0}^{R} f_{im} + \sum_{j=0}^{C} f_{mj})$, pri čemer sta f_{im} in f_{mj} maksimalni frekvenci i-te vrstice in j-tega stolpca v kontingenčni tabeli.

Najpreprosteje je mero izračunati iz vrednosti kontingenčne tabele, v enačbi 2.22 je N število vseh primerov, z v pa so označene vrednosti iz tabele, ki so enake frekvencam f pomnoženim z N.

$$\lambda = \frac{\sum_{i=0}^{R} v_{im} + \sum_{j=0}^{C} v_{mj} - v_{\bullet m} - v_{m\bullet}}{2N - v_{\bullet m} - v_{m\bullet}} \tag{2.22}$$

Mera je simetrična in ni definirana samo, če so vsi primeri v eni celici kontingenčne tabele, sicer je zaloga njene vrednosti $[0, 1]$. Maksimum doseže, če so vse vrednosti vsakega para vrstic in stolpcev v eni celici, kar se zgodi, če so vrednosti razporejene po diagonali. Vrednost 0 pomeni statistično neodvisnost, a vrednost večja od nič statistično še ne zagotavlja korelacije.

2.5.3 Simetrični koeficient nedoločenosti

Koeficient nedoločenosti (angl. *uncertanty coefficient*) je mera korelacije, ki temelji na teoriji informacij. Povzemamo jo po knjigi *Numerical recipes in C* [22]. Zelo podobna je razmerju informacijskega prispevka, le da namesto deleža informacije, ki nam jo k poznavanju razredne spremenljivke prispeva poznavanje značilke, merim količino informacije, ki nam jo poznavanje ene spremenljivke prispeva k poznavanju druge. Ker je medsebojna informacija med spremenljivkama enaka, vrstni red ni pomemben, poznavanje ene spremenljivke prispeva enako količino informacije o drugi. Zaradi tega je simetrični koeficient nedoločenosti definiran z enačbo 2.23. X in Y sta spremenljivki, H je entropija. Izračun entropij posameznih spremenljivk smo že pojasnili, naj pa zapišemo še izračun entropije para spremenljivk, ki je v tem primeru $H(X,Y) = \sum_{i=0}^{R} \sum_{j=0}^{C} f_{ij}\, log_2\, f_{ij}$, pri katerem je f_{ij} frekvenca i-te vrstice j-tega stolpca kontingenčne tabele.

$$U(X,Y) = 2\left(\frac{H(X) + H(Y) - H(X,Y)}{H(X) + H(Y)}\right) \tag{2.23}$$

Zaloga vrednosti simetričnega koeficienta nedoločenosti je $[0,1]$. Nič doseže, če sta spremenljivki neodvisni in ena če poznavanje ene spremenljivke popolnoma napove vrednosti druge.

Poglavje 3

Orange Multitarget - dodatek za sočasno uvrščanje v več razrednih spremenljivk

Orange Multitarget je dodatek za programski paket *Orange* [8], ki združuje obstoječe in dodaja nove tehnike za sočasno uvrščanje v več razrednih spremenljivk. Združuje in dodaja tudi načine za ovrednotenje izdelanih modelov. V tem poglavju opišemo implementacijo dreves za razvrščanje v skupine in pregledamo še implementacije verig uvrščevalnikov, nevronskih mrež in metode delnih najmanjših kvadratov. Izpostavljamo predvsem konkretne rešitve in morebitne spremembe pri implementacijah v primerjavi z opisom metod v poglavju 2.

Na koncu poglavja so na kratko predstavljeni še gradniki, ki so bili v okviru dodatka izdelani za *Orange Canvas*.

3.1 ClusteringTreeLearner

V programskem paketu *Orange* je že pred razvojem dodatka obstajala implementacija dreves za razvrščanje v skupine. Napisana je bila v programskem jeziku *Python* in je posledično ob povečevanju števila primerov v učni množici

za gradnjo drevesa potrebovala preveč časa in spomina. Ta pomanjkljivost je postala zelo očitna, ko je uporabnik želel ta drevesa uporabiti za grajenje naključnih gozdov. Zaradi tega se je pojavila želja po hitrejši implementaciji in v okviru razvoja dodatka so bila drevesa napisana v programskem jeziku *C*.

Kot osnova za implementacijo je služil *SimpleTreeLearner* – poenostavljena implementacija klasičnih odločitvenih dreves v programskem jeziku *C*. *SimpleTreeLearner* je bil napisan iz enakih razlogov kot drevesa za razvrščanje v skupine – želje po hitro delujoči implementaciji odločitvenih dreves, ki se uporablja predvsem za gradnjo naključnih gozdov. Bistveno višjo hitrost kot prvotna odločitvena drevesa dosega ne samo zaradi izbire programskega jezika, temveč tudi zaradi poenostavitev in zmanjšanja prilagodljivosti. Pri gradnji kot meri za izbiro značilk uporablja razmerje informacijskega prispevka pri diskretnih in srednjo kvadratno napako pri zveznih razrednih spremenljivkah, v primeru da ima več značilk enako oceno, vedno izbere prvo in tudi vzvratnega rezanja ne izvede po koncu gradnje. Te poenostavitve omogočajo hitro izgradnjo klasifikatorja, a je zaradi tega uporabniku zelo oteženo spreminjanje in prilagajanje implementacije. Brez spreminjanja izvorne kode lahko uporabniki s parametri določajo samo maksimalno globino drevesa, delež večinskega razreda, pri katerem se gradnja zaključi in število primerov, pri katerem algoritem zaključi z gradnjo zaradi zmanjšane zanesljivosti.

Podobne poenostavitve so bile uporabljene tudi pri implementaciji dreves za razvrščanje v skupine. Algoritem po končanem grajenju ne opravi vzvratnega rezanja in v primeru večih značilk z enako oceno izbere prvo. Uporabnik lahko s parametri določa pogoje končanja grajenja in tudi mero, s katero algoritem izbira značilke.

3.1.1 Ustavitveni kriteriji

Ustavitveni kriteriji, ki jih uporabljamo pri gradnji dreves, so določeni s štirimi parametri. Prvi je *max_ depth*, ki določa največjo dovoljeno globino drevesa. Gradnja se ustavi, ko je ta globina dosežena. Parameter *min_ instances*

določa najmanjše število primerov, pri katerem gradnjo še nadaljujemo. Če je na primer parameter nastavljen na pet, bi značilka, ki razdeli deset primerov na dve skupini s petimi primeri, še lahko bila uporabljena, značilka z razdelitvijo na štiri in šest primerov pa zavrnjena. Naslednja parametra sta *min_ majority* in *min_ MSE*, z njima ustavimo gradnjo, ko čistost vseh razrednih spremenljivk preseže s parametrom nastavljeno vrednost. *min_ majority* je uporabljen pri diskretnih razrednih spremenljivkah in meri deleže večinskih razredov vseh razrednih spremenljivk. Če delež večinskega razreda preseže parameter pri vseh razrednih spremenljivkah, se gradnja ustavi. Podobno deluje parameter *min_ MSE*, ki pri zveznih razrednih spremenljivkah ustavi grajenje, če je srednja kvadratna napaka vseh razrednih spremenljivk manjša od parametra. V enačbi 3.1 je M število razrednih spremenljiv, N število primerov in y_{mi} vrednost i-tega primera m-te razredne spremenljivke.

$$\forall m \in M \; \frac{1}{N}(\sum_{i=0}^{N} y_{mi}^2 - \frac{1}{N}(\sum_{i=0}^{N} y_{mi})^2) < min\_MSE \tag{3.1}$$

3.1.2 Izbira značilk

Implementirane so štiri mere za izbiro značilk: razdalja med skupinami, razdalja znotraj skupine, silhueta razbitja in mera nečistoče Gini. Uporabnik lahko uporabljeno metodo nastavi s parametrom *method*. Če metoda ni eksplicitno izbrana, je uporabljena razdalja med skupinami, ki so jo predlagali tudi avtorji dreves za razvrščanje v skupine. Izbira je ponujena, ker se v literaturi pojavijo različni pristopi: Blockeel in sod. predlagajo razdaljo med skupinami [2], Schietgat in sod. (med njimi tudi Blockeel) pri napovedovanju genskih funkcij uporabljajo razdaljo znotraj skupin [29], načeloma pa se za ugotavljanju uspešnosti razbitij na skupine uporablja silhueta razbitja. Dodana je še mera Gini, ki je namenjena razrednim spremenljivkam z nominalnimi vrednostmi.

Avtorji v svoji implementaciji predvidevajo, da so vsa vozlišča drevesa binarna. Mi s testiranjem nismo odkrili prednosti takšnega pristopa, zato naša

implementacija uporablja standarden pristop k ustvarjanju vozlišč, za vsako možno vrednost značilke je ustvarjeno novo vozlišče. Uporabljena značilka je izvzeta iz množice možnih značilk pri nadaljevanju gradnje. Pri zveznih značilkah preverimo vse mogoče razdelitve učne množice na dva dela in ocena najboljše od teh razdelitev je tudi končna ocena značilke. Zvezne značilke so lahko ponovno uporabljene pri nadaljnji gradnji, s tem omogočimo, da se lahko tudi z zvezno značilko učna množica razdeli na več kot dve podmnožici.

V nadaljevanju so podrobno opisane implementacije vseh štirih mer. Primerjava uspešnosti mer se nahaja v dodatku A.

Razdalja med skupinami

Razdalja med skupinami je zelo preprosta mera, ki so jo predlagali že avtorji dreves za razvrščanje v skupine. Algoritem najprej razdeli primere glede na vrednosti merjene značilke ter za vsako tako ustvarjeno skupino izračuna prototip. Za funkcijo prototip uporabljamo centroid skupine – povprečne vrednosti vseh razrednih spremenljivk. Ocena značilke je povprečje vseh razdalj med skupinami, posamezna razdalja pa je evklidska razdalja med prototipoma. Enačba 3.2 prikazuje izračun razdalje med skupinami; P je matrika prototipov in M je število razrednih spremenljivk, pt_{ik} predstavlja vrednost k-te razredne spremenljivke i-tega prototipa. Razdalje ne korenimo, da prihranimo računanje, saj se vrstni red značilk zaradi korenjenja ne spremeni. S to mero izbrana značilka razvrsti primere v skupine z med seboj najbolj oddaljenimi centroidi.

$$inter\_distance = \binom{|P|}{2}^{-1} \sum_{i=0}^{|P|} \sum_{j=0;i\neq j}^{|P|} \sum_{k=0}^{M} (pt_{ik} - pt_{jk})^2 \tag{3.2}$$

Razdalje znotraj skupin

Razdalja znotraj skupine za funkcijo prototip prav tako uporablja centroide skupin. S to mero izbrana značilka razvrsti primere v skupine z najmanj od centroidov oddaljenimi oz. najmanj različnimi primeri. Enačba 3.3 prikazuje

izračun te razdalje, $|P|$ je število prototipov. Ker za vsako skupino obstaja en prototip, je število prototipov enako številu skupin. C je množica primerov znotraj posamezne skupine, M je število razrednih spremenljiv. Vsota razdalj po vseh razrednih spremenljivkah med primeri posamezne skupine c_{jk} in prototipom skupine pt_{jk} je utežena z velikostjo skupine.

$$intra\_distance = \sum_{i=0}^{|P|} \frac{1}{|C_i|} \sum_{j=0}^{|C_i|} \sum_{k=0}^{M} (c_{jk} - pt_{jk})^2 \tag{3.3}$$

Silhueta razbitja

Silhueta razbitja v skupine (tudi silhuetni koeficient) je mera, ki izračuna uspešnost razvrstitve v skupine, uporabi pa tako razdaljo znotraj kot tudi razdaljo med skupinami. Prvi jo je opisal Rousseeuw [28]. Tipično se izračuna tako, da za vsak primer učne množice $\mathbf{x}_i$, ki je uvrščen v skupino C, izračunamo njegova povprečna oddaljenost od primerov znotraj skupine – $intra(i)$. Za vsako od preostalih skupin $C_j; \mathbf{x}_i \notin C_j$ pa je razdalja primera $\mathbf{x}_i$ do te skupine povprečje razdalj primera $\mathbf{x}_i$ do vseh primerov uvrščenih v skupino C_j. $inter(i)$ je najmanjša od teh razdalj do preostalih skupin. Silhuetni koeficient 3.4 je razlika teh razdalj ulomljena z večjo od obeh in nam ovrednoti, kako primerna je uvrstitev posameznega primera v neko skupino v primerjavi z uvrstitvijo v najbližjo drugo skupino.

Da lahko silhueto uporabimo kot mero za izbiro značilke, je potrebno za vsako izmed njih dvakrat skozi vse učne primere v vsakem vozlišču. To je časovno zahtevna operacija, zato uporabimo mero, ki le temelji na silhuetnem koeficientu. Namesto, da računamo razdalje vsakega primera do vseh ostalih primerov, izračunamo samo razdalje vsakega primera do prototipov. Take je $intra(i)$ razdalja primera do centroida svoje skupine in $inter(i)$ najmanjša od razdalj primera do ostalih prototipov skupin. Še vedno je za izračun koeficienta potrebna iteracija čez vse učne primere $\frac{1}{N}\sum_{i=0}^{N} silhouette(i)$, v vsaki od njih pa računamo samo razdalje do vseh prototipov. Naša implementacija uporablja evklidsko mero razdalje.

$$silhouette(i) = \frac{inter(i) - intra(i)}{max(inter(i), intra(i))} \tag{3.4}$$

Mera nečistoče Gini

Razdalje med primeri niso primerna mera za nominalne razredne spremenljivke, saj razdalje med njihovimi vrednostmi niso definirane. Za merjenje teh razrednih spremenljivk se praviloma uporabljajo mere nečistoče, to sta večinoma ali razmerje informacijskega prispevka ali mera Gini [18]. Mero Gini smo uporabili, ker je implementacija preprostejša od razmerja informacijskega prispevka in ker med merama ni omembe vredne razlike. Primerjavo med merama sta opravila Raileanu in Stoffel [25] in zaključila, da z raziskavami nikomur ni uspelo empirično dokazati, da je posamezna mera boljša, ker se razlikuje samo dva procenta ocen značilk, ki jih podata.

Gini-indeks je definiran z enačbo 3.5, kjer je CV število vrednosti razredne spremenljivke, f_k pa frekvenca oz. verjetnost i-te vrednosti razreda. Gini-indeks lahko interpretiramo kot pričakovano napako uvrščanja – kako pogosto bi naključno izbran primer uvrstili napačno, če bi ga izbirali glede na distribucijo vrednosti.

$$gini\_index = 1 - \sum_{k=0}^{CV} f_k^2 \tag{3.5}$$

Mera nečistoče Gini je razlika med apriornim in pričakovanim aposteriornim Gini-indeksom. Za naše potrebe smo ga še razširili in izračunamo razliko med povprečnim apriornim in povprečnim pričakovanim aposteriornim Gini-indeksom vseh razrednih spremenljivk. V enačbi 3.6 je AV število vrednosti ocenjevane značilke in f_i verjetnost posamezne vrednosti te značilke, s katero je utežen povprečni aposteriorni Gini-indeks razrednih spremenljivk. M je število razrednih spremenljivk in f_{jk} frekvenca k-te vrednosti j-te razredne spremenljivke. CV_j je število vrednosti j-te razredne spremenljivke.

$$Gini = \sum_{i=0}^{AV} f_i \frac{1}{M} \sum_{j=0}^{M} \sum_{k=0}^{CV_j} f_{jk|i}^2 - \frac{1}{M} \sum_{j=0}^{M} \sum_{k=0}^{CV_j} f_{jk}^2 \tag{3.6}$$

3.2 NeuralNetworkLearner

NeuralNetworkLearner je implementacija naprej usmerjenih več-nivojskih nevronskih mrež. Osnovni algoritem je napisal Žbontar za potrebe tekmovanja s področja odkrivanja znanj iz podatkov *JRS 2012 Data Mining Competition* [35]. Za integracijo s programskim paketom *Orange* je bila preko Žbontarjevega programa napisana ovojnica, ki podatke pretvarja v osnovnemu algoritmu primerno obliko in iz njegovih izhodov prebere rezultate.

Osnovni algoritem podpira en skriti nivo perceptronov in za aktivacijsko funkcijo v nevronih uporablja logistično funkcijo ($S(t) = 1/(1 + e^{-t})$), ki je pogosto uporabljena v nevronskih mrežah zaradi preprostega odvoda funkcije, ker je zvezna in omejena, a nikoli ne doseže mej (za razliko od pragovne funkcije) in ker hitreje konvergira pri vzvratnem razširjanju napake.

Za optimizacijo uteži nevronske mreže je uporabljen algoritem *fmin_l_bfgs_b* iz knjižnjice *SciPy* [15]. Ime algoritma je skovanka njegovih sestavnih delov in lastnosti, algoritem namreč temelji na metodi *BFGS*, ki so jo leta 1970 neodvisno drug od drugega razvili Broyden, Fletchet, Goldfarb in Shanno (ime algoritma je sestavljeno iz začetnic njihovih priimkov). Obravnava te metode presega okvir diplomske naloge. Uporabljena različica *L-BFGS-B* ima omejeno porabo prostora in njenim prametrom lahko postavimo preproste meje, zaradi tega je še posebej primerna za probleme z veliko spremenljivkami.

Ker je bil osnovni algoritem napisan za delo z večznačnimi podatkovnimi nabori, ga je bilo potrebno še razširiti za delo z več večrazrednimi spremenljivkami. Algoritem za vsako razredno spremenljivko, ki ima več kot dve možni vrednosti, ustvari izhodni perceptron za vsako od teh vrednosti. Izhodne napovedi nato združi nazaj v sezname verjetnosti in kot napoved uporabi vrednost spremenljivke z največjo verjetnostjo.

Parametri, s katerimi lahko optimiziramo delovanje klasifikatorja, so *reg_fact* – regularizacijski faktor (pri opisu metode imenovan λ), s katerim uravnavamo prilagajanje metode učnim podatkom, *n_mid* – število perceptronov skritega nivoja, ki zelo pomembno vpliva tako na uspešnost učenja in s povečevanjem tudi na računsko zahtevnost metode ter *max_iter* s katerim omejimo zgornjo mejo iteracij optimizacijskega algoritma *fmin_l_bfgs_b*.

3.3 PLSClassificationLearner

V programskem paketu *Orange* je že pred izdelavo dodatka obstajala implementacija metode delnih najmanjših kvadratov za regresijo – *PLSRegressionLearner*. Napisana je bila po zgledu implementacije te metode v programski knjižnici za strojno učenje *scikit-learn* [21]. Za izračun matrike latentnih komponent sta implementirana dva algoritma: *NIPALS* in *SVD*. Za izračun napovedi uporablja dva pristopa, prvi je regresijski, ki izračuna regresijske koeficiente in nove primere napoveduje z njimi (ta algoritem spada glede na število razrednih spremenljivk v sklop *PLS1* oz. *PLS2*) ter drugi, tako imenovan kanoničen oz. pristni pristop, ki je implementacija originalnega Woldovega algoritma imenovanega *PLS-C2A*. Nastavitve lahko spreminjamo s podajanjem parametrov *algorithm* in *deflation_mode*.

Ker je metoda delnih najmanjših kvadrato regresijska, smo v dodatku implementacijo te metode ovili z ovojnico *PLSClassificationLearner* in tako ustvarili klasifikator. Vsaka razredna spremenljivka z več kot dvema možnima vrednostima je nadomeščena z binarnimi razrednimi spremenljivkami za vsako od teh vrednosti. Takšni podatki so nato podani regresijski tehniki, rezultati le-te pa so uporabljeni kot napovedi verjetnosti posameznih razredov vsake od razrednih spremenljivk. Te napovedi so nato združene in razredni spremenljivki napovemo vrednost, ki jo je regresijska tehnika napovedala z največjo verjetnostjo.

3.4 ClassifierChainLearner in EnsembleClassifierChainLearner

ClassifierChainLearner je implementacija verige klasifikatorjev. Kot smo pojasnili že v opisu tehnike poteka gradnja modela zaporedno. Vrstni red razrednih spremenljivk lahko podamo s parametrom *class_ order*, sicer ga klasifikator generira naključno. Po tem vrstnem redu je za vsako razredno spremenljivko zgrajen klasifikator standardne tehnike, ki smo jo razširitveni tehniki podali s parametrom *learner*. Na tej točki se pojavi dilema, ali naj med značilke dodamo vrednosti, ki jih je napovedal zgrajen klasifikator ali pa uporabimo dejanske vrednosti, na katerih se učimo. Avtorji tehnike v članku te dileme eksplicitno ne razrešijo, zato smo v implementaciji dodali parameter *actual_ values*, s katerim lahko izbiramo, ali bomo uporabili dejanske ali napovedane vrednosti.

Pri uvrščanju novih primerov se napovedi posameznih modelov med značilke dodajajo v enakem vrstnem redu, kot smo jih prej gradili in tako sestavijo končno rešitev.

EnsembleClassifierChainLearner je nadgradnja osnovnega verižnega uvrščevalnika, ki smo jo implementirali po algoritmih ansambla verig klasifikatorjev. Enako kot pri osnovnem verižnem klasifikatorju, je tudi pri ansamblu potrebno podati standardno tehniko uvrščanja s parametrom *learner*. Gradnjo verig lahko prilagajamo s še dvema prametroma: z *n_ chains* nastavimo število verig, ki jih želimo zgraditi in s *sample_ size* velikost naključne podmnožice učnih podatkov, na kateri bomo zgradili posamezno verigo.

Ob klasifikaciji vsaka veriga klasifikatorjev poda svoje napovedi, iz katerih so nato z glasovanjem izbrana končne napovedi vrednosti razrednih spremenljivk.

3.5 Ostali elementi implementacije

BinaryRelevanceLearner

Ločeni klasifikatorji so implementirani v *BinaryRelevanceLearner* in podpirajo večino standardnih tehnik za uvrščanje, ki jih vsebuje osnovni programski paket.

Vrednotenje tehnik

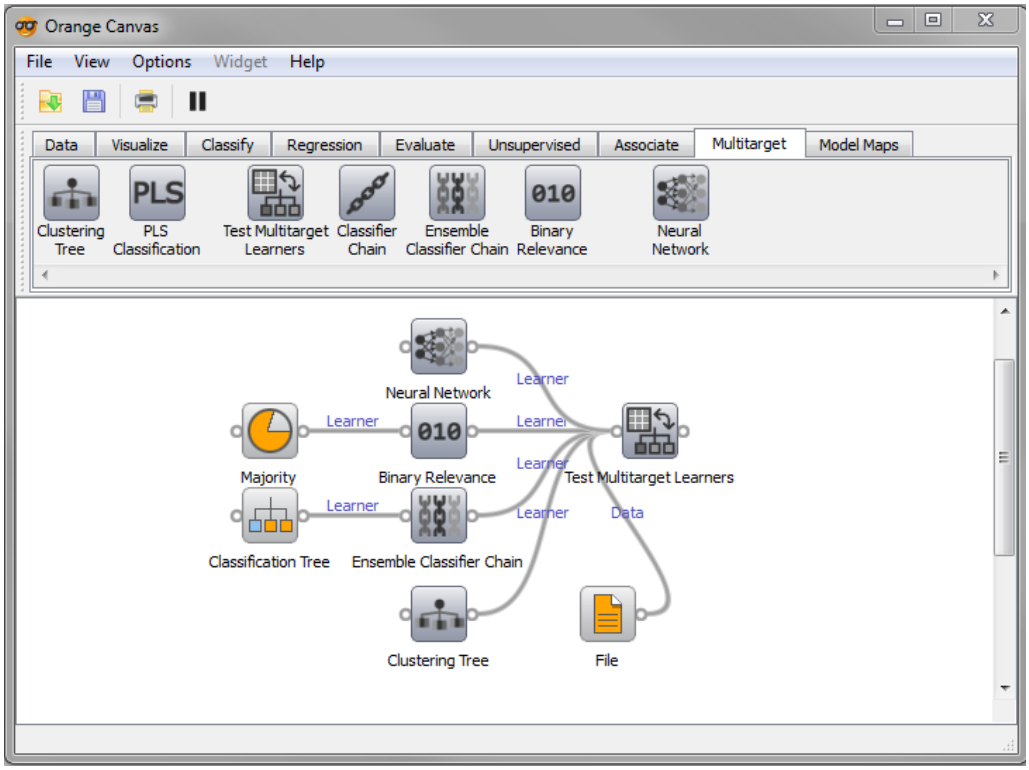
Orange skupaj z dodatkom *Orange Multitarget* podpira vse mere opisane v razdelku 2.3.1. Mere uspešnosti namenjene standardnim tehnikam so implementirane v osnovnem programu, dodatek pa vsebuje funkciji *mt_average_score* in *mt_flattened_score*, s katerima lahko te mere uporabimo na rezultatih tehnik za uvrščanje v več razrednih spremenljivk; prva izmeri povprečno mero, druga pa sploščeno mero.

Mikro in makro meri $F1$ sta kot večznačni meri vključeni v osnovni program, saj vanj vključene tudi druge večznačne mere. V dodatku sta implementirani tudi edini namenski meri za več večrazrednih spremenljivk – – globalna in povprečna točnost. Prva je dosegljiva s klicem funkcije *mt_global_accuracy* in druga s klicem *mt_mean_accuracy*.

Orange Canvas in gradniki dodatka Multitarget

Programski paket *Orange* poleg knjižnice skript sestavlja tudi *Orange Canvas*, namizna aplikacija, ki omogoča hitro, preprosto in uporabnikom prijazno uporabo tehnik strojnega učenja, odkrivanja znanj iz podatkov in vizualizacije podatkov. Osnovni sestavni deli te aplikacije so gradniki (angl. *widgets*), ki jih uporabniki sestavljajo v sheme, ki nato podatke obdelajo v skladu z nastavitvami uporabnikov.

Dodatek *Orange Multitarget* vsebuje gradnike za vse implementirane tehnike in tudi gradnik za njihovo testiranje. S tem skušamo uporabnikom omogočiti hitro in preprosto uporabo tehnik za sočasno uvrščanje v več razrednih spremenljivk. Primer uporabe lahko vidimo na sliki 3.1.



Slika 3.1: Demonstracija uporabe gradnikov dodatka *Orange Multitarget* v okolju *Orange Canvas*.

Poglavje 4

Eksperimentalna primerjava metod

4.1 Podatki

V besedilu, ki sledi, je na kratko opisan vsak od uporabljenih naborov podatkov, saj se nam zdi pomembno, da o uporabljenih podatkih poznamo tudi nekaj ozadja. Nabori so razdeljeni v tri sklope, v prvem so zbrani večznačni nabori, v drugem nabori z več večrazrednimi spremenljivkami in v tretjem umetno ustvarjeni nabori podatkov.

4.1.1 Večznačni nabori podatkov

V tabeli 4.1 so prikazane osnovne značilnosti enajstih naborov podatkov namenjenih testiranju večznačnih tehnik uvrščanja. Podatki obsegajo tako manjše nabore z malo razrednimi spremenljivkami kot tudi relativno velike nabore z veliko značilkami in veliko razrednimi spremenljivkami. Vključeni so tudi trije nabori (*emotions*, *scene* in *yeast*), ki se zelo pogosto pojavljajo med testnimi v objavah, ki testirajo uspešnost večznačnih tehnik uvrščanja [26, 34].

Ime nabora	Št. primerov	Št. atribtov	Št. razr. spremenljivk
bibtex	7395	1836	159
CCRISAZ	657	196	5
CSRISC	7348	55, 165, 306, 1021	5
emotions	593	72	6
enron	1702	1001	53
medical	978	1449	45
scene	2407	294	6
yeast	2417	103	14

Tabela 4.1: Tabela večznačnih naborov podatkov

bibtex

Podatkovni nabor *bibtex* je sestavljen iz metapodatkov, ki se nahajajo v zapisu *.bib*. Te datoteke služijo sistemu BibTeX za zapisovanje bibliografij in posledično vsebujejo podatke o avtorjih in informacije o njihovih publikacijah. Cilj gradnje modela je napovedati, v katere kategorije spada posamezen zapis. Podatki so bili pripravljeni v študiji Katakisa in sod. [16], mi pa uporabljamo prečiščeno in tudi zmanjšano verzijo, ki je dostopna na spletni strani knjižnice *Mulan* [32].

CCRISAZ

Podatkovni nabor *CCRISAZ* je v preučevanje prejel Laboratorij za bioinformatiko Fakultete za računalništvo in informatiko, posredovala ga je sodelujoča skupina iz podjetja AstraZeneca. Podatki so iz področja biokemije, popisujejo pa odziv modelnega organizma na različne učinkovine.

CSRISC

Tudi podatkovni nabor *CSRISC* je v preučevanje prejel Laboratorij za bioinformatiko Fakultete za računalništvo in informatiko od prej omenjenega podjetja. Na začetku je vseboval *smiles* zapise kemičnih spojin in vrednosti razredov. Značilke smo pridobili s pomočjo programa *Open Babel* [20], ki omogoča pridobitev kemijskih odtisov molekul (angl. *molecular fingerprints*).

Kemijski odtis molekule je binarni zapis značilnosti strukture molekule, *Open Babel* ga zna izračunati v štirih oblikah: *fp3* je dolžine 64 bitov, *maccs* dolžine 256 bitov, *fp4* dolžine 512 bitov in *fp2* dolžine 1024 bitov. Iz podatkov smo odstranili značilke, ki nimajo pozitivnih vrednosti in tako pridobili štiri nabore podatkov.

emotions

Trohidis in sod. so postopek ustvarjenja nabora *emotions* obširno opisali v članku [30]. Podatke sestavljajo digitalni zapisi glasbe in pripadajoče oznake občutkov, ki so jih posameznim pesmim pripisali strokovnjaki. Cilj ustvarjenega modela je na podlagi zapisa glasbe napovedati, kako se bo ob poslušanju počutil poslušalec.

enron

Enron je še eden izmed pogosto uporabljenih naborov podatkov za testiranje. Uporabljeno obliko je pripravil Reed [26]. Nabor je podmnožica večjega nabora, ki so ga ustvarili na Univerzi v Kaliforniji, Berkley ter MIT, vsebuje pa elektronsko pošto vodilnih iz znanega propadlega podjetja Enron. Tej elektronski pošti so študentje na univerzah dodali oznake in tako je nastal uporaben nabor podatkov.

medical

Nabor *medical* je nastal za potrebe tekmovanja v pripisovanju oznak besedilom, ki so napisana v zdravstvu. Tekmovalci so morali z obdelavo teh besedil le-tem pripisovati pravilne oznake. Na spletni strani knjižnice *Mulan* [32] so ti podatki dostopni v prečiščeni obliki in jih lahko uporabimo za testiranje klasifikatorjev.

scene

Scene je podatkovni nabor, ki so ga ustvarili Boutell in sod. [4]. Na podlagi digitalnega zapisa fotografij so želeli napovedovati njihovo vsebino oz. opisati sceno, ki jih fotografije prikazujejo (oznake, ki so jih uporabljali so bile plaža, sončni zahod, jesensko listje, urbano okolje in gore ter seveda tudi kombinacije teh oznak). Težave so jim pričakovano povzročale fotografije s kombinacijami scen. Podatkovni nabor je sedaj prosto dostopen, nimamo pa slik, da bi lahko še v praksi preverili, kako dobro uvrščajo zgrajeni modeli. Nekaj slik si je možno ogledati v članku.

yeast

Podatkovni nabor *yeast* sta za testiranje uporabila Elisseeff in Weston [10]. Podatki vsebujejo opise genov ter njihove oznake. Točna drevesna zgradba podatkov je opisana v članku, mi uporabljamo preprostejšo različico, ki je dostopna na spletni strani knjižnice *Mulan* [32].

4.1.2 Podatki z več večrazrednimi spremenljivkami

V tabeli 4.2 so zbrani osnovni podatki o podatkovnih naborih z več večrazrednimi spremenljivkami. Ti nabori so zaradi nerazvitosti področja redki. Lahko bi jih ustvarjali sami s prestavljanjem značilk med razredne spremenljivke, a smo se raje odločili poiskati že uporabljene nabore podatkov.

Ime nabora	Št. primerov	Št. atribtov	Št. razr. spremenljivk
bridges	108	5	5
flare	1066	10	3
issues	1480	9	3
ntp	1935	55, 165, 307, 1021	5
thyroid	9172	29	7

Tabela 4.2: Tabela naborov podatkov z več večrazrednimi spremenljivkami.

bridges

Podatkovni nabor *bridges* so ustvarili gradbeni inženirji z univerze Carnegie Mellon. Nabor vsebuje tehnične podatke o mostovih. V enaki obliki, kot podatke uporabljamo mi, jih je uporabil tudi Ženko v svojem doktoratu [36].

flare

Flare je nabor podatkov pridobljen iz repozitorija strojnega učenja Univerze v Kaliforniji, Irvine [11]. Podaki so sestavljeni iz opazovanj Sončevih peg med 13. februarjem in 27. marcem leta 1969 ter opisujejo lastnosti peg in lastnosti Sonca v času pege. Cilj modela je napovedati število Sončevih izbruhov posameznih kategorij (C - pogoste, M - zmerne, X - silovite). Ker so zmerni in še posebej siloviti izbruhi redki, je distribucija razredov zelo izkrivljena [36]. Ženko je pri svojih testiranjih uporabil manjšega od dveh naborov podatkov, mi uporabljamo večjega, saj zanj avtorji trdijo, da je bolj zanesljiv.

issues

Nabor podatkov *issues* sestavljajo podatki iz študije državnih volitev leta 1992 v Združenih državah Amerike. Volivci so bili anketirani in podali svoje mnenje o nekaterih takrat perečih temah (npr. občutki o prejemnikih socialne podpore ali ocena podpore tradicionalnim moralnim vrednotam). Naloga zgrajenih modelov je ugotoviti spol, raso in iz katere regije prihaja anketiranec. Podatkovni nabor je uporabljen z dovoljenjem prof. Jacobyja [14], ki je podatke zbral in pripravil za testiranje.

ntp

Nabor podatkov *ntp* je v preučevanje prejel Laboratorij za bioinformatiko Fakultete za računalništvo in informatiko od sodelujoče skupine pri podjetju AstraZeneca. Nabor je sestavljen iz *smiles* zapisov molekul in petih večrazrednih spremenljivk. S postopkom, ki smo ga opisali pri naboru *CSRISC* 4.1.1,

smo obdelali tudi tega in pridobili štiri nabore podatkov z večrazrednimi spremenljivkami.

thyroid

Nabor *thyroid* je pridobljen iz repozitorija strojnega učenja Univerze v Kaliforniji, Irvine [11]. Vsebuje zapise o pacientih, ki so jim pregledali ščitnico, razredna spremenljivka pa je doktorjeva diagnoza. Ker se v podatkih pojavijo tudi več diagnoz naenkrat in le-te spadajo v sedem kategorij, Ženko predlaga razdelitev osnovne ene razredne spremenljivke v sedem večrazrednih spremenljivk [36]. Podatke v tej obliki uporabimo tudi mi.

4.1.3 Sintetični nabori podatkov

Za testiranje hipoteze, da je uspešnost tehnik za sočasno uvrščanje v več razrednih spremenljivk odvisna od korelacije med razrednimi spremenljivkami, smo generirali umetne nabore podatkov.

Najprej smo ustvarili sedem naborov podatkov, ki so sestavljeni iz treh značilk in dveh razrednih spremenljivk, tako značilke kot tudi razredne spremenljivke sestavljajo samo binarne vrednosti. Vsi nabori so sestavljeni iz stotih primerov in vrednosti značilk so generirane naključno. Vrednost prve razredne spremenljivke je logična disjunkcija vrednosti prve in tretje značilke, vrednost druge razredne spremenljivke pa je določena z eno izmed sedmih logičnih funkcij med vrednostjo druge značilke in vrednostjo prve razredne spremenljivke. Sedem logičnih funkcij, ki tudi določajo sedem naborov podatkov, sestavljajo naslednje: logična konjunkcija, disjunkcija, ekskluzivna disjunkcija, implikacija, obratna implikacija ter še samo vrednost značilke in samo vrednost razredne spremenljivke. S temi funkcijami je pokrit celoten prostor logičnih funkcij, če izvzamemo negacije naštetih ter tavtologijo in kontradikcijo.

Za vsako od logičnih funkcij je bilo generiranih nekaj tisoč naborov podatkov in izbrani tisti, ki so imeli najbolj korelirane razredne spremenljivke. Ker smo sumili, da so nabori s stotimi primeri premajhni za dobro delovanje

ansambelskih tehnik, smo po enakem postopku izdelali še nabore z 250 in 500 primeri. Ker v teh podatkih ni nobenega šuma razen začetnih vrednosti značilk, smo izdelali še dodatnih sedem naborov z 250 primeri. Pri teh smo dodali še eno razredno spremenljivko, katere vrednosti so določena naključno in nato petini primerov naključno spremenili eno izmed vrednosti razrednih spremenljivk. Da so razredne spremenljivke ostale korelirane je zagotavljala generacija velikega števila naborov in izbira tistih, ki so bili najbolj korelirani.

Povprečno korelacijo razrednih spremenljivk teh osemindvajsetih naborov podatkov smo primerjali s povprečno korelacijo med razrednimi spremenljivkami naborov iz preostalih dveh skupin. Uporabljene mere korelacije smo predstavili v razdelku 2.5, rezultati meritev pa so strnjeni v tabeli 4.3. Iz nje lahko razberemo, da so razredne spremenljivke sintetičnih naborov v povprečju bolje korelirane kot razredne spremenljivke realnih naborov. Podrobnejši rezultati meritev so predstavljeni v dodatku B.

Tip nabora	**Koef. nedoločenosti**	**Koef. λ**	**Koef. C**	**Koef. V**
večznačni	0.095	0.105	0.259	0.238
večrazredni	0.090	0.066	0.317	0.230
sintetični	**0.372**	**0.372**	**0.613**	**0.530**

Tabela 4.3: Tabela povprečij povprečnih ocen koreliranosti razrednih spremenljivk za vsako od skupin podatkov.

4.2 Postopek testiranja

S testiranjem smo primerjali uspešnost posameznih tehnik za uvrščanje v več razrednih spremenljivk, hkrati pa še izvedli primerjavo teh tehnik s standardnimi tehnikami. Kot izhodišče za primerjave uporabljamo večinski klasifikator (v nadaljevanju *Maj*). Naključni gozdovi z odločitvenimi drevesi so standardna tehnika, ki jo uporabimo v ločenih klasifikatorjih (*RF*) in v ansamblu verig klasifikatorjev (*CRF*). Izbrali smo jih, ker veljajo za eno izmed najbolj uspešnih in robustnih tehnik uvrščanja za probleme z eno razredno

spremenljivko [7]. Testirane tehnike za uvrščanje v več razredov pa so nevronske mreže (NN), metoda delnih najmanjih kvadratov (PLS), drevesa za razvrščanje v skupine (CLT) in naključni gozdovi z drevesi za razvrščanje v skupine (CLF). Pri testiranju sintetičnih naborov podatkov smo med testirane tehnike dodali še ločene klasifikatorje z logistično regresijo (LR), saj smo po prvotnih testiranjih sumili, da naključni gozdovi morda niso najboljši pri ocenjevanju verjetnosti svojih napovedi. Logistična regresija nam tako služi kot še ena referenčna standardna tehnika pri interpretaciji rezultatov.

Testiranje smo izvedli v treh delih, najprej smo tehnike preverili na večznačnih podatkovnih naborih, nato na naborih z več večrazrednimi spremenljivkami in na koncu še na sintetičnih naborih podatkov. Delitev je enaka tisti, ki smo jo uporabili pri opisu podatkovnih naborov v prejšnjem razdelku.

Vsaka tehnika je bila na vsakem naboru podatkov preverjena s pet-kratnim prečnim preverjanjem. Med testiranjem smo za vsako tehniko na vsakem naboru izmerili povprečen logaritem izgube in povprečno točnost. Povprečni logaritem izgube smo merili, ker se nam zdi pomembno, s kakšnim zaupanjem tehnike podajajo svoje napovedi. Seveda je zaželeno, da model napoveduje čim bolj točno, a v primeru napačnih napovedi je pomembna verjetnost, s katero so te napovedi ocenjene. Napačne napovedi ocenjene z visoko verjetnostjo so manj zaželene kot napačne napovedi podane z nizko verjetnostjo, slednje namreč kažejo, da je model, ki ga je tehnika zgradila, pri svojih napovedih zadržan in upošteva morebitno pomanjkanje informacij z zmanjšanjem verjetnosti svoje napovedi. Obenem je zaželeno, da ob pravilnih napovedih model le-te napove z visoko verjetnostjo, ima pa zaradi logaritemske narave mere manj verjetna pravilna napoved na oceno bistveno manjši vpliv kot napačne napovedi.

Točnosti je po našem mnenju ena izmed osnovnih mer, ki jo je pri testiranju vedno dobro izmeriti. Izmeri nam delež pravilnih napovedi, pravilne napovedi pa so tudi razlog, da klasifikator sploh gradimo. Uporabili smo povprečno točnost, saj je samo ta definirana tako na večznačnih podatkih kot tudi na naborih podatkov z več večrazrednimi spremenljivkami. Izbrana

je bila namesto globalne točnosti, ker je slednja velikokrat prestroga, sploh če napovedujemo veliko število razrednih spremenljivk.

Ta kombinacija mer po našem mnenju dobro opiše uspešnost tehnik uvrščanja, saj preveri tako napovedi kot tudi verjetnosti, s katerimi so napovedi podane. Ostalim meram, ki so definirane samo na večznačnih podatkovnih naborih (npr. mera $F1$, priklic, natančnost), smo se skušali izogniti, saj v celotnem dodatku stremimo predvsem k čim bolj splošnemu uvrščanju v več razrednih spremenljivk, te mere pa izključujejo večrazredne spremenljivke. Ostale mere na osnovi verjetnosti (npr. Brierjeva mera in povprečna informacija vsebine odgovora) niso bile merjene, ker menimo, da je dovolj ena taka mera.

Imputacija manjkajočih vrednosti

Ker naši implementaciji nevronskih mrež in delnih najmanjših kvadratov ne podpirata manjkajočih vrednosti v razrednih spremenljivkah, smo pri naborih podatkov, ki vsebujejo manjkajoče vrednosti, znotraj prečnega preverjanja pred napovedovanjem izvedli modelsko imputacijo. Model smo zgradili z odločitvenim drevesom, ki smo mu maksimalno globino zaradi želje po hitrem delovanju omejili na dvajset. Ob napovedovanju razredne spremenljivke z manjkajočimi vrednostmi smo med značilke dodali še preostale razredne spremenljivke in modelu omogočili upoštevanje informacijo o preostalih razredih.

Za odločitvena drevesa smo se odločili, ker bi bila imputacija z naključnimi gozdovi preveč časovno zahtevna, osnovnejši pristopi kot na primer napoved večinskega razreda, pa ustvarijo manj točno rekonstrukcijo podatkov. Ker dejanskih vrednosti manjkajočih vrednosti v podatkih ne poznamo, smo za testiranje rekonstrukcije iz podatkov izbrisali nekaj vrednostih, ki jih poznamo in jih poskusili napovedati. Drevesa so se pričakovano manjkajoče vrednosti napovedovala bolje kot večinski klasifikator. Dodatna možnost pri imputaciji je še, da manjkajoče vrednosti dodamo kot novo možno vrednost razredni spremenljivki. To po našem mnenju preveč popači podatke in tudi

pri testiranju se je pokazalo, da zmanjša uspešnost uvrščanja.

Parametri metod

Problem, ki smo ga skušali kar najbolje rešiti, je bil vsaki tehniki uvrščanja omogočiti izdelavo najboljšega možnega modela vsakega od naborov podatkov. Veliko število podatkovnih naborov in parametrov metod nam je onemogočilo optimizacijo parametrov, zato smo metodam, ki optimizacijo potrebujejo, podali nekaj različnih vrednosti parametrov. S tem smo jim skušali omogočiti vsaj delno prilagajanje podatkovnim naborom.

Naključni gozdovi so zelo robustna tehnika, zato smo pri njih samo upoštevali avtorjev nasvet, naj posameznih dreves ne režemo. Tehniki *RF* in *CLF* sta gradili sto dreves, za *CRF* pa smo zaradi naraščajoče kompleksnosti število dreves v posameznem gozdu zmanjšali na petdeset in zgradili petdeset verig.

Pri metodi *PLS* smo uporabili algoritem *nipals* in regresijski način napovedovanja, testirali pa smo nekaj različnih števil latentnih komponent.

Drevesa za razvrščanje v skupine so zelo odvisna od parametrov, zato smo poskusili pet različnih stopenj rezanja. Pri večznačnih podatkovnih naborih smo značilke izbirali z razdaljo med skupinami, pri podatkovnih naborih z večrazrednimi spremenljivkami pa smo uporabili mero Gini.

Nevronska mreže so po našem mnenju med vsemi uporabljenimi tehnikami najbolj občutljive na spremembe parametrov. Že majhne spremembe velikosti skritega nivoja in regularizacijskega faktorja lahko prinesejo velike spremembe, a tudi povečajo računsko zahtevnost in podaljšajo čas kovergence k rešitvi. Zaradi tega smo testirali dve različni števili perceptronov v skritem nivoju (petdeset in sto) ter za vsakega od njiju tri različne regularizacijske faktorje (5, 0.5 in 0.05).

Logistična regresija je uporabljena s privzetimi parametri njene implementacije v *Orange*.

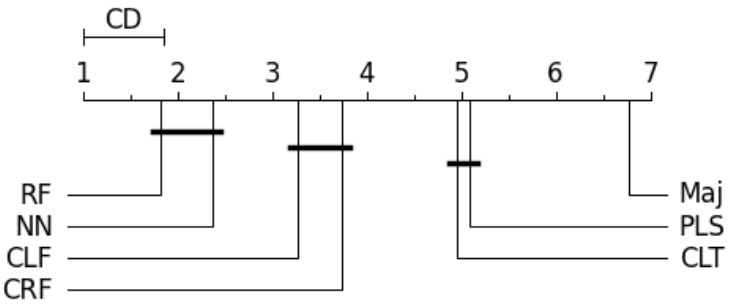
Našteti parametri so delovali dobro, dokler ni bilo potrebno testirati velikih naborov podatkov, problematični so bili *bibtex*, *enron* in *medical*, ki so

zaradi zelo velikega števila razrednih spremenljivk ob gradnji gozdov zasedla preveč spomina. Zaradi tega smo pri teh naborih podatkov zmanjšali število dreves v gozdu in tudi število verig ter povečali parameter *min_ instances* pri drevesih.

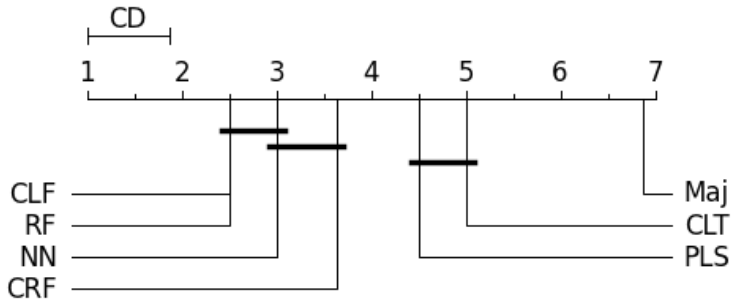
4.3 Rezultati

Rezultati so zaradi želje po jasnosti predstavljenimi z grafi kritičnih razlik. Izrisani so pri stopnji značilnosti $\alpha = 0.05$. Podrobnejši rezultati so na voljo v dodatku C.

Prva skupina rezultatov je izrisana za povprečni logaritem izgube. Kot smo že pojasnili, smo skušali s to mero predvsem preveriti, ali zgrajeni model ovrednoti zanesljivost svojih napovedi in temu prilagodi verjetnost teh napovedi. Na grafu 4.1 so zbrani rezultati vrednotenja večznačnih podatkovnih naborov. Iz njega lahko razberemo, da sta tehniki naključnih gozdov in nevronskih mrež najbolj uspešno prilagajali verjetnosti pravilnim napovedim. Na drugem grafu 4.2, kjer so prikazani rezultati za podatkoven nabore z večrazrednimi spremenljivkami, lahko opazimo, da ni več statistično značilne razlike med vodilnima tehnikama s prvega grafa in preostalima tehnikama, ki tudi uporabljata naključne gozdove. Sploh opazen je napredek naključnih gozdov z drevesi za razvrščanje v skupine.

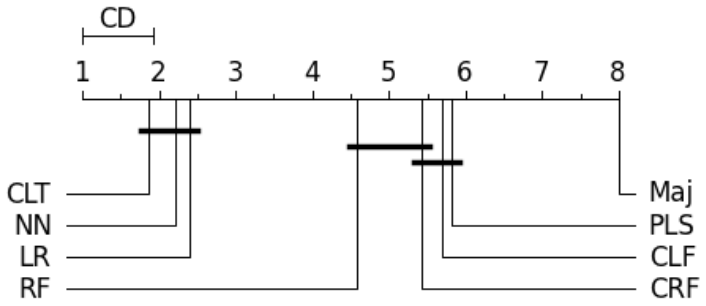


Slika 4.1: Graf kritičnih razlik povprečnega logaritma izgube za večznačne podatkovne nabore.



Slika 4.2: Graf kritičnih razlik povprečnega logaritma izgube za podatke z več večrazrednimi spremenljivkami.

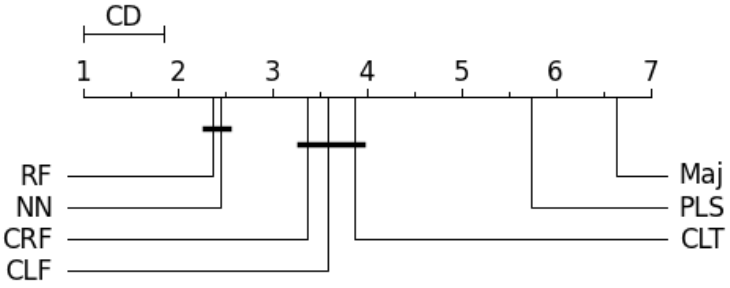
V primerjavi s prejšnjima grafoma lahko na grafu rezultatov sintetičnih naborov podatkov 4.3 opazimo značilen napredek tehnik *NN* in *CLT* pri ocenjevanju verjetnosti svojih napovedi. Ker pa smo pri sintetičnih naborih ocenjevali tudi uspešnost logistične regresije, napredek *NN* in *CLT* ne pomeni uspeha v primerjavi s standardnimi tehnikami, saj je logistična regresija od njiju oddaljena za razliko, ki je manjša od kritične.



Slika 4.3: Graf kritičnih razlik povprečnega logaritma izgube za sintetične nabore podatkov.

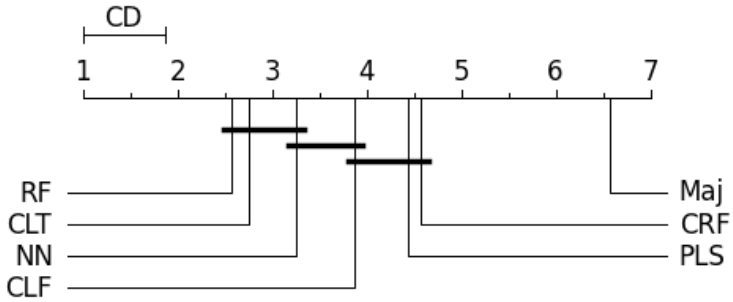
Druga skupina grafov primerja rezultate izmerjene s povprečno točnostjo. Graf 4.4, ki ga sestavljajo rezultati večznačnih naborov podatkov, pokaže, da so tehnike, ki so bile uspešne pri napovedovanju verjetnosti, uspešne tudi pri dejanskem uvrščanju. Opazna sprememba v primerjavi z grafom povprečnega logaritma izgube je napredek tehnike *CLT*, katere povprečna točnost se izkaže

za primerljivo *CRF* in *CLF*. Na račun te spremembe je nazadovala *PLS*, ki je od testiranih tehnik najmanj točna.



Slika 4.4: Graf kritičnih razlik povprečne točnosti za večznače podatkovne nabore.

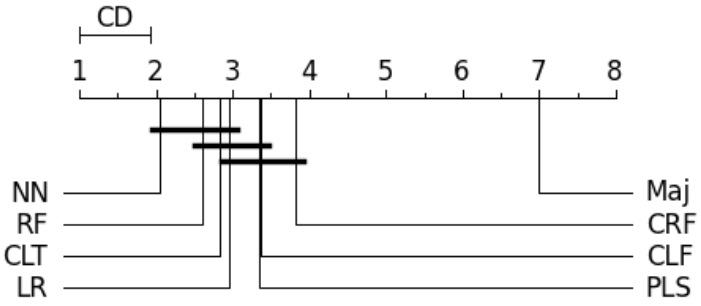
Na grafu naborov večrazrednih naborov podatkov 4.5 so se razlike med tehnikami v primerjavi z merjenjem logaritma izgube še zmanjšale. Opazen je napredek dreves za razvrščanje v skupine, ki so rangirane namesto na peto na približno tretje. Še pomembnejše je, da je razlika med njimi in naključnimi gozdovi manjša od kritične.



Slika 4.5: Graf kritičnih razlik povprečne točnost za podatke z več večrazrednimi spremenljivkami.

Iz grafa 4.6 lahko razberemo, da je merjenje točnosti v primerjavi z merjenjem logaritma izgube bistveno zmanjšalo razlike med tehnikami. Nevronske mreže so še vedno prve, a njihova prednost pred naključnimi gozdovi je manjša od kritične. Uspešnost večine tehnik so torej po izmerjeni točnosti

ne razlikuje bistveno. To spremembo lahko pojasnimo s pregledom podrobnejših rezultatov (tabela C.6 v dodatkih). Iz tabele lahko razberemo, da so ocene točnosti pri nekaterih naborih enake za vse tehnike (razen večinskega klasifikatorja). Očitno so se zaradi preprostosti nekaterih naborov na njih vse tehnike odrezale enako dobro in posledično so tudi razlike med njihovimi povprečnimi rangi majhne. Odstranitev teh meritev malenkost poveča razliko med nevronskimi mrežami in naključnimi gozdovi, a razlika ostane manjša od kritične. Graf nam torej pove, da je uspešnost tehnik pri merjenju točnosti med sabo zelo primerljiva in da nimamo tehnike, ki bi bila izrazito boljša od preostalih.



Slika 4.6: Graf kritičnih razlik povprečne točnosti za sintetične nabore podatkov.

Presenetljiv uspeh v rezultatih dosegajo naključni gozdovi, ki so zgrajeni s tehniko ločenih klasifikatorjev. Ker napovedujejo vsako razredno spremenljivko posebej, bi njihov uspeh lahko pomenil, da korelacija med razrednimi spremenljivkami ne poveča uspešnosti tehnike, ki jo upošteva. To morda lahko pojasnimo z rezultati skupine sintetičnih podatkov, kjer sta razredni spremenljivki popolnoma korelirani (skupina s korenom *syn_clss* v tabeli C.6). Na teh naborih vse tehnike (razen večinskega klasifikatorja) napovedo razredne spremenljivke popolnoma pravilno. Razlog, da tudi naključni gozdovi in logistična regresija uspešno napovedujeta ta nabor je preprost – če sta dve razredni spremenljivki popolnoma korelirani in lahko iz značilk točno napovemo prvo, bomo iz enakih značilk točno napovedali tudi drugo razredno

spremenljivko, saj so ob popolni korelaciji vse njene vrednosti ali enake ali obratne od vrednosti prve. Tudi pri rezultatih merjenj logaritma izgube C.5 ni odstopanj, saj tako drevesa za razvrščanje kot tudi logistična regresija ocenita svoje napovedi popolnoma pravilno. Logaritem izgube naključnih gozdov v ločenih klasifikatorjih pa je enak logaritmu izgube naključnih gozdov z drevesi za razvrščanje v skupine. Rezultati se torej ujemajo z grafi kritičnih razlik sintetičnih naborov podatkov.

Poglavje 5

Sklepne ugotovitve

Pričeli smo z vprašanjem, ali lahko s sočasnim napovedovanjem večih razrednih spremenljivk ustvarimo boljše napovedi, kot če jih napovedujemo vsako posebej. To vprašanje se je razvilo v bolj konkretno: ali povezanost med razrednimi spremenljivkami izboljša uspešnost tehnik za sočasno uvrščanje v več razrednih spremenljivk v primerjavi z gradnjo ločenih modelov za vsako od teh razrednih spremenljivk. Vpliva nam ni uspelo dokazati, saj so rezultati razširjenih standardnih tehnik in rezultati tehnik za sočasno uvrščanje v več razredov med seboj primerljivi tako na relativno nizko koreliranih naborih podatkov iz realnega sveta, kot tudi na umetno ustvarjenih visoko koreliranih podatkovnih naborih.

Uspešnost naključnih gozdov z odločitvenimi drevesi kaže, da je mogoče tudi s tehnikami, ki korelacijo med razredi zanemarijo, uspešno napovedovati probleme z več razrednimi spremenljivkami. Še več, zdi se, da koreliranost med podatki pripomore k uspešnosti teh tehnik, saj visoka korelacija med razrednimi spremenljivkami zagotavlja, da če tehnika uspešno napove eno od njih, bo uspešna tudi pri ostalih. Posledično menimo, da je pri modeliranju podatkov z več razrednimi spremenljivkami dobro testirati oba tipa tehnik, vnaprej namreč ne moremo določiti, katera od njih bo najboljša.

Prispevek diplomskega dela je dodatek *Orange Multitarget*, ki uporabnikom omogoča uporabo tehnik za sočasno uvrščanje v več razredov in tudi

vrednotenje ustvarjenih modelov. Dosegljiv je preko spletne strani programskega paketa *Orange*[1].

Možna pomanjkljivost naših rezultatov je uporaba tehnik za uvrščanje brez optimalnih parametrov. Idealno bi lahko v prihodnosti za programski paket *Orange* razvili neko ovojnico za učne tehnike, ki bi sama iskala optimalne oz. vsaj optimalnejše parametre. Zaradi pomanjkanja te optimizacije so namreč naši rezultati manj natačni, kot bi lahko bili. Še vseeno pa verjamemo, da smo testirali dovolj parametrov, da jih popolna optimizacija ne bi bistveno spremenila.

Na podlagi uspeha dreves za razvrščanje v skupine kot samostojne tehnike, bi bilo dobro implementirati še katero od tehnik vzvratnega rezanja, ki bi njihovo uspešnost lahko še povečala. To je tudi predlog avtorjev algoritma, ki pa je bil v prvotni implementaciji ignoriran zaradi želje po hitrem delovanju dreves znotraj naključnih gozdov.

Omenili bi še dve zanimivosti, ki bi ju po našem mnenju bilo dobro preučiti. V nasprotju z dognanji avtorjev se je naša implementacija ansambla verig klasifikatorjev odrezala slabše kot ločeni klasifikatorji. So za to morda krivi neprimerni parametri ali uporaba napačnih standardnih tehnik? Druga zanimivost, ki bi zaslužila testiranje pa je, ali uporaba določenega modela pri imputaciji vpliva na rezultate uporabe enakega modela pri napovedovanju. Bolj konkretno, če bi namesto odločitvenih dreves pri imputaciji uporabili nevronske mreže, bi to izboljšalo rezultate nevronskih mrež in zmanjšalo uspešnost naključnih gozdov?

[1] `http://orange.biolab.si/addons/`

Literatura

[1] C. Bielza, G. Li in P. Larraòaga, "Multi-dimensional classification with bayesian networks," *Int. J. Approx. Reasoning*, zv. 52, št. 6, str. 705–727, sep 2011.

[2] H. Blockeel, L. De Raedt in J. Ramon, "Top-down induction of clustering trees," in *Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, 1998, str. 55–63.

[3] A.-L. Boulesteix in K. Strimmer, "Partial least squares: a versatile tool for the analysis of high-dimensional genomic data," *Briefings in Bioinformatics*, zv. 8, št. 1, str. 32–44, 2007.

[4] M. R. Boutell, J. Luo, X. Shen in C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, zv. 37, št. 9, str. 1757–1771, 2004.

[5] L. Breiman, "Random forests," *Mach. Learn.*, zv. 45, št. 1, str. 5–32, oct 2001.

[6] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Mon. Wea. Rev.*, zv. 78, št. 1, str. 1–3, jan 1950.

[7] R. Caruana, N. Karampatziakis in A. Yessenalina, "An empirical evaluation of supervised learning in high dimensions," in *Proceedings of the 25th international conference on Machine learning*, ser. ICML '08. USA: ACM, 2008, str. 96–103.

[8] J. Demšar, B. Zupan, G. Leban in T. Curk, "Orange: From experimental machine learning to interactive data mining," in *Knowledge Discovery in Databases: PKDD 2004*, ser. Lecture Notes in Computer Science. Springer, 2004, zv. 3202, str. 537–539.

[9] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, zv. 7, str. 1–30, dec 2006.

[10] A. Elisseeff in J. Weston, "A kernel method for multi-labelled classification," in *Advances in Neural Information Processing Systems 14 (NIPS-01)*, 2002, str. 681–687.

[11] A. Frank in A. Asuncion, "UCI machine learning repository," 2010. Dosegljivo na: http://archive.ics.uci.edu/ml

[12] L. Goodman in W. Kruskal, *Measures of association for cross classifications*, ser. Springer series in statistics. Springer-Verlag, 1979.

[13] T. Hastie, R. Tibshirani in J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[14] W. G. Jacoby, Michigan State University, Department of Political Science, 2008. Dosegljivo na: http://polisci.msu.edu/jacoby/msu/pls802/data/dlist.html

[15] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–. Dosegljivo na: http://www.scipy.org/

[16] I. Katakis, G. Tsoumakas in I. Vlahavas, "Multilabel text classification for automated tag suggestion," in *ECML/PKDD-08 Workshop on Discovery Challenge*, 2008.

[17] D. Kleinbaum, *Logistic regression: a self-learning text.* Springer, 1994.

[18] I. Kononenko, *Strojno učenje.* Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2005.

[19] I. Kononenko in I. Bratko, "Information-based evaluation criterion for classifier's performance," *Machine Learning*, zv. 6, str. 67–80, 1991.

[20] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch in G. R. Hutchison, "Open babel: An open chemical toolbox." *J Cheminform*, zv. 3, št. 1, p. 33, 2011.

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

[22] W. Press, S. Teukolsky, W. Vetterling in B. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge, UK: Cambridge University Press, 1992.

[23] J. R. Quinlan, *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[24] ——, "Induction of decision trees," *Machine Learning*, zv. 1, št. 1, str. 81–106, Mar. 1986.

[25] L. E. Raileanu in K. Stoffel, "Theoretical comparison between the gini index and information gain criteria," *Annals of Mathematics and Artificial Intelligence*, zv. 41, št. 1, str. 77–93, may 2004.

[26] J. Read, B. Pfahringer, G. Holmes in E. Frank, "Classifier chains for multi-label classification," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, ser. ECML PKDD '09. Berlin, Heidelberg: Springer-Verlag, 2009, str. 254–269.

[27] F. Rosenblatt, "The perceptron: A perceiving and recognizing automaton," Ithaca, New York, Tech. Rep. 85-460-1, jan 1957.

[28] P. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, zv. 20, št. 1, str. 53–65, nov 1987.

[29] L. Schietgat, C. Vens, J. Struyf, H. Blockeel, D. Kocev in S. Dzeroski, "Predicting gene function using hierarchical multi-label decision tree ensembles," *BMC Bioinformatics*, zv. 11, št. 2, 2010.

[30] K. Trohidis, G. Tsoumakas, G. Kalliris in I. P. Vlahavas, "Multi-label classification of music into emotions." in *ISMIR*, J. P. Bello, E. Chew in D. Turnbull, Eds., 2008, str. 325–330.

[31] G. Tsoumakas, I. Katakis in I. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," in *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, 2008.

[32] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek in I. Vlahavas, "Mulan: A java library for multi-label learning," *Journal of Machine Learning Research*, zv. 12, str. 2411–2414, 2011. Dosegljivo na: http://http://mulan.sourceforge.net/datasets.html

[33] Y. Yang, "An evaluation of statistical approaches to text categorization," *Journal of Information Retrieval*, zv. 1, str. 67–88, 1999.

[34] J. H. Zaragoza, L. E. Sucar, E. F. Morales, C. Bielza in P. Larrañaga, "Bayesian chain classifiers for multidimensional classification." in *IJCAI*, T. Walsh, Ed. IJCAI/AAAI, 2011, str. 2192–2197.

[35] J. Žbontar, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2012. Dosegljivo na: https://bitbucket.org/jzbontar/ml-class/

[36] B. Ženko, I. Bratko in S. Džeroski, "Learning predictive clustering rules," Doktorska dizertacija, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2007.

Dodatek A

Vpliv mere za izbiro značilke na rezultate dreves za razvrščanje v skupine

Želeli smo ugotoviti, ali in predvsem kako velike razlike obstajajo med posameznimi merami za ocenjevanje značilk pri drevesi za razvrščanje v skupine. Izbrali smo osem naborov podatkov na katerih smo testirali vse štiri metode in rezultati kažejo, da izbira mere ne vpliva bistveno na rezultate. Na grafu A.1 lahko na desni vidimo, da so kar tri od štirih mer znotraj kritične razlike, torej je pri uporabi naključnih gozdov z drevesi za razvrščanje izbira mere ni zelo pomemben faktor. Pri samostojnih drevesih za razvrščanje v skupine, katerih rezultati so prikazani na levem grafu, je za boljše rezultate bolje izbrati mero Gini ali razdaljo med skupinami kot preostali meri.

Če želimo dobre rezultate, je glede na rezultate tabele A.1 pri samostojnih dreves za razvrščanje najbolje uporabiti mero nečistoče Gini, v naključnem gozdu iz dreves za razvrščanje v skupine A.2 pa razdaljo med primeri. Če rezultate tabele pogledamo še podrobneje, je za najboljše rezultate najbolje preveriti vse štiri mere, saj se pri določenih naborih podatkov najbolje izkaže tudi metoda silhuete, ki je sicer na grafih slabša od preostalih.

Zanimivo je še, da mera Gini vrne najboljše rezultati tudi na nekaterih

Slika A.1: Grafa kritičnih razlik povprečne točnosti. Na desni je graf rabe različnih mer pri naključnimi gozdovi z drevesi za razvrščanje v skupine in na levi graf rabe različnih mer za izbiro značilk pri drevesih za razvrščanje v skupine.

Ime nabora	ClT inter	ClT intra	ClT sil	ClT gini	Maj
bridges	0.7204	0.7204	**0.7315**	0.7315	0.6519
CCRISAZ	**0.8359**	0.8289	0.8244	0.8292	0.8359
CSRISC.fp3	0.6990	0.6843	0.6957	**0.7026**	0.6192
CSRISC.maccs	0.8051	0.7157	0.7444	**0.8066**	0.6839
emotions	0.7799	0.7799	0.7799	0.7799	0.6886
issues	0.5782	0.5782	0.5782	0.5782	0.5538
ntp.fp3	0.8582	0.8592	0.8547	**0.8594**	0.8520
ntp.maccs	0.8710	0.8625	0.8535	**0.8726**	0.8535

Tabela A.1: Tabela rezultatov dreves za razvrščanje v skupine merjenih s povprečno točnostjo. Odebeljen je rezultat najbolj uspešne tehnike za vsak nabor podatkov.

Ime nabora	ClF inter	ClF intra	ClF sil	ClF gini	Maj
bridges	**0.7241**	0.7074	0.6741	0.6926	0.6519
CCRISAZ	0.8432	0.8377	**0.8441**	0.8429	0.8359
CSRISC.fp3	**0.7036**	0.6779	0.6775	0.7021	0.6192
CSRISC.maccs	**0.8078**	0.7551	0.7714	0.8049	0.6839
emotions	0.8066	0.8066	0.8066	0.8066	0.6886
issues	0.5885	0.5885	0.5885	0.5885	0.5538
ntp.fp3	**0.8585**	0.8564	0.8520	0.8574	0.8520
ntp.maccs	0.8727	0.8648	0.8535	**0.8732**	0.8535

Tabela A.2: Tabela rezultatov naključnih gozdov z drevesi za razvrščanje v skupine merjenih s povprečno točnostjo. Odebeljen je rezultat najbolj uspešne tehnike za vsak nabor podatkov.

večznačnih naborih in še pomembneje, na na vseh naborih z več večrazrednimi spremenljivkami. Načeloma naj preostalih mer pri slednjih naborih sploh ne uporabljali, saj z merjenjem razdalje na nominalnih podatkih delamo napako. Glede na rezultate bi bilo ob uporabi tehnike morda smiselno tudi na podatkih z nominalnimi razrednimi spremenljivkami poskusiti katero izmed mer na osnovi razdalj, predvsem razdaljo med skupinami.

Naj omenimo še, da je računanje silhuetnega koeficienta najbolj zahtevno, bistveno bolj kot preostale mere, mera Gini in razdalja med skupinami sta po zahtevnosti med sabo primerljivi, nekoliko več časa pa porabi razdalja znotraj skupin, a še vedno bistveno manj kot silhuetni koeficient.

Dodatek B

Podrobni rezultati testov korelacije

V tem dodatku so predstavljeni rezultati merjenj korelacije za vsakega izmed naborov podatkov. Merjeni so bili povprečni koeficient nedoločenosti, povprečni koeficient λ, povprečni koeficient C in povprečni koeficient V. Povprečje je izračunano na koeficientih korelacije vseh parov razrednih spremenljivk posameznega nabora.

V tabeli B.1 so zbrane meritve na večznačnih naborih podatkov. Kot najbolj korelirani se izkažejo nabori družine *CSRISC* in nabor *CCRISAZ*, najmanj pa nabor *medical*. V tabeli B.2 so zbrani rezultati meritev na naborih z več večrazrednimi spremenljivkami, najbolj koreliran izmed njih je nabor *bridges*, najmanj pa nabor *thyroid*.

V tabeli B.3 so zbrane meritve na sintetičnih naborih podatkov. V tabli lahko opazimo trend padanja koreliranosti podatkov posamezne skupine naborov s povečevanjem velikosti nabora. Najvišje ocene imajo nabori velikosti 100, najnižje pa nabori s 500 primeri. Seveda je še manj od vseh treh koreliran šumni nabor (označen s končnico _r). Najbolj korelirani so pričakovano podatki *clss*, ki predstavljajo logično funkcijo, ki je odvisna samo od vrednosti razredne spremenljivke. Posledično je med razrednima spremenljivkama popolna korelacija. Najmanj korelirani so podatki *att*, kjer je

vrednost razredne spremenljivke odvisna samo od vrednosti značilke.

Ime nabora	Keof. nedoločenosti	Koef. λ	Koef. C	Koef. V
bibtex	0.0052	0.0007	0.0155	0.0158
CCRISAZ	0.1901	0.1464	0.3963	0.4391
CSRISC.fp2	0.1530	0.2240	0.4187	0.3388
CSRISC.fp3	0.1530	0.2240	0.4187	0.3388
CSRISC.fp4	0.1530	0.2240	0.4187	0.3388
CSRISC.maccs	0.1530	0.2240	0.4187	0.3388
emotions	0.1161	0.0729	0.2913	0.3124
enron	0.0066	0.0024	0.0371	0.0375
medical	0.0039	0.0006	0.1104	0.1148
scene	0.0591	0.0000	0.1771	0.1804
yeast	0.0528	0.0394	0.1477	0.1596

Tabela B.1: Povprečne vrednosti koeficientov korelacije za vsakega od večznačnih naborov podatkov.

Ime nabora	Keof. nedoločenosti	Koef. λ	Koef. C	Koef. V
bridges	0.2071	0.1448	0.5618	0.4404
flare	0.0581	0.0172	0.3888	0.2697
issues	0.0064	0.0261	0.0871	0.0680
ntp.fp2	0.1098	0.0855	0.3612	0.2546
ntp.fp3	0.1098	0.0855	0.3612	0.2546
ntp.fp4	0.1098	0.0855	0.3612	0.2546
ntp.maccs	0.1098	0.0855	0.3612	0.2546
thyroid	0.0084	0.0009	0.0499	0.0406

Tabela B.2: Povprečne vrednosti koeficientov korelacije za vsakega od naborov podatkov z več večznačnimi razrednimi spremenljivkami.

Ime nabora	Keof. nedoločenosti	Koef. λ	Koef. C	Koef. V
syn_and	0.4431	0.5309	0.7747	0.6547
syn_and250	0.3412	0.2826	0.6973	0.5668
syn_and500	0.2929	0.1503	0.6500	0.5175
syn_and_r	0.0883	0.0879	0.3796	0.2832
syn_att	0.0947	0.1739	0.4253	0.3153
syn_att250	0.0341	0.0743	0.2658	0.1913
syn_att500	0.0179	0.0529	0.2002	0.1430
syn_att_r	0.0210	0.0764	0.2060	0.1479
syn_cimp	0.3584	0.3684	0.7003	0.5700
syn_cimp250	0.3082	0.1591	0.6630	0.5307
syn_cimp500	0.3022	0.1571	0.6606	0.5283
syn_cimp_r	0.1274	0.1152	0.4537	0.3448
syn_clss	1.0000	1.0000	0.9892	0.9788
syn_clss250	1.0000	1.0000	0.9955	0.9910
syn_clss500	1.0000	1.0000	0.9977	0.9955
syn_clss_r	0.2769	0.2643	0.5681	0.4845
syn_mimp	0.7933	0.8333	0.9233	0.8620
syn_mimp250	0.6159	0.6495	0.8767	0.7901
syn_mimp500	0.5549	0.5741	0.8511	0.7535
syn_mimp_r	0.2199	0.1798	0.4887	0.4036
syn_or	0.7461	0.8182	0.9224	0.8605
syn_or250	0.6292	0.6667	0.8829	0.7992
syn_or500	0.5440	0.5431	0.8433	0.7428
syn_or_r	0.1387	0.1373	0.4355	0.3422
syn_xor	0.1334	0.2192	0.5003	0.3782
syn_xor250	0.0310	0.0055	0.2616	0.1882
syn_xor500	0.0146	0.0404	0.1836	0.1309
syn_xor_r	0.0160	0.0478	0.1896	0.1353

Tabela B.3: Povprečne vrednosti koeficientov korelacije za vsakega od sintetičnih naborov podatkov.

Dodatek C

Podrobni rezultati testiranja

V tem dodatku so predstavljeni podrobni rezultati testiranj, na podlagi katerih so izrisani grafi kritičnih razlik. S krepko so izpostavljeni rezultati najboljše tehnike na vsakem od naborov podatkov. Če imajo najboljši rezultat več kot tri tehnike, nismo s krepko poudarili nobenega od njih.

Menimo, da posebnih komentarjev posamezne tabele ne potrebujejo, saj samo podrobno prikazujejo, kar smo že analizirali v razdelku rezultati 4.3.

Ime nabora	**CLT**	**CLF**	**Maj**	**NN**	**PLS**	**RF**	**CRF**
bibtex	0.0546	**0.0503**	0.0749	0.0918	0.0631	0.0581	0.0657
CCRISAZ	0.4419	0.3727	0.4419	0.3972	0.3909	**0.3515**	0.3749
CSRISC.fp2	0.5232	0.4928	0.6259	0.4504	0.5520	**0.4473**	0.5024
CSRISC.fp3	0.5368	0.5351	0.6163	0.5290	0.5686	**0.5269**	0.5991
CSRISC.fp4	0.4566	0.4572	0.5797	**0.4007**	0.5053	0.4214	0.4635
CSRISC.maccs	0.4246	0.3960	0.6133	**0.3638**	0.5039	0.3835	0.4111
emotions	0.5067	0.4190	0.6126	0.4527	0.4815	**0.4117**	0.4224
enron	0.1570	0.1450	0.1676	**0.1312**	0.1562	0.1324	0.1368
medical	0.0775	0.0988	0.0996	**0.0339**	0.0795	0.0444	0.0528
scene	0.4120	0.2611	0.4688	**0.1951**	0.3338	0.2343	0.2515
yeast	0.4984	0.4534	0.4958	**0.4414**	0.4659	0.4510	0.4441

Tabela C.1: Rezultati povprečnega logaritma izgube za večznačne nabore podatkov.

Ime nabora	CLT	CLF	Maj	NN	PLS	RF	CRF
bridges	0.7331	0.7604	0.9234	0.7285	0.7124	**0.7122**	0.7595
flare	0.2806	0.2647	0.2906	0.2663	**0.2614**	0.2676	0.2680
issues	0.8514	0.7937	0.8388	0.7956	0.7969	0.7892	**0.7854**
ntp.fp2	0.4301	**0.3867**	0.4474	0.3897	0.4139	0.3943	0.4052
ntp.fp3	0.4275	**0.4078**	0.5144	0.4497	0.4720	0.4482	0.4635
ntp.fp4	0.4194	0.3892	0.4417	**0.3759**	0.3983	0.3828	0.3920
ntp.maccs	0.4094	**0.3707**	0.4875	0.3711	0.4290	0.3795	0.3774
thyroid	0.0594	0.0563	0.1827	0.0784	0.1399	**0.0416**	0.0512

Tabela C.2: Rezultati povprečnega logaritma izgube za nabore podatkov z več večrazrednimi spremenljivkami.

Ime nabora	CLT	CLF	Maj	NN	PLS	RF	CRF
bibtex	**0.9869**	0.9864	0.9849	0.9782	0.9856	0.9857	0.9850
CCRISAZ	0.8359	0.8432	0.8359	0.8347	0.8341	**0.8533**	0.8411
CSRISC.fp2	0.7414	0.7607	0.6438	**0.8144**	0.7389	0.7906	0.7549
CSRISC.fp3	**0.7162**	0.7125	0.6426	0.7159	0.6946	0.7143	0.7092
CSRISC.fp4	0.7936	0.7772	0.6979	**0.8183**	0.7614	0.8005	0.7793
CSRISC.maccs	0.8130	0.8190	0.6607	**0.8423**	0.7594	0.8281	0.8083
emotions	0.7760	0.8069	0.6886	0.7929	0.7712	0.8061	**0.8094**
enron	0.9482	0.9467	0.9372	**0.9544**	0.9425	0.9541	0.9520
medical	0.9773	0.9723	0.9723	**0.9895**	0.9732	0.9857	0.9820
scene	0.8526	0.8911	0.8210	**0.9252**	0.8450	0.9083	0.9013
yeast	0.7807	0.7923	0.7682	0.7985	0.7881	0.7897	**0.7995**

Tabela C.3: Rezultati povprečne točnosti za večznačne nabore podatkov.

Ime nabora	CLT	CLF	Maj	NN	PLS	RF	CRF
bridges	**0.7333**	0.7259	0.6537	0.7259	0.7315	0.7296	0.7241
flare	0.9303	0.9303	0.9303	0.9290	0.9303	0.9303	0.9303
issues	0.5782	0.5885	0.5538	**0.6014**	0.5838	0.5950	0.5968
ntp.fp2	0.8614	0.8644	0.8545	**0.8728**	0.8626	0.8657	0.8593
ntp.fp3	**0.8639**	0.8628	0.8275	0.8429	0.8387	0.8423	0.8384
ntp.fp4	0.8708	0.8626	0.8562	0.8703	0.8666	**0.8718**	0.8628
ntp.maccs	**0.8770**	0.8748	0.8388	0.8770	0.8541	0.8759	0.8713
thyroid	0.9831	0.9805	0.9600	0.9723	0.9617	**0.9855**	0.9821

Tabela C.4: Rezultati povprečne točnosti za nabore podatkov z več večrazrednimi spremenljivkami.

Ime nabora	CLT	CLF	Maj	NN	PLS	RF	CRF	LR
syn_and	0.1688	0.2489	0.6734	0.0097	0.1681	0.2183	0.3167	**0.0001**
syn_and250	**0.0000**	0.1679	0.6498	0.0037	0.1711	0.1435	0.1582	0.0001
syn_and500	**0.0000**	0.1699	0.6370	0.0018	0.1689	0.1430	0.1347	0.0001
syn_and_r	0.3540	0.3985	0.6547	**0.3411**	0.4018	0.3745	0.3866	0.3488
syn_att	0.1897	0.2100	0.6340	0.0073	0.1026	0.1621	0.2621	**0.0000**
syn_att250	**0.0000**	0.1374	0.6136	0.0027	0.0848	0.1091	0.1449	**0.0000**
syn_att500	**0.0000**	0.1354	0.6213	0.0013	0.0828	0.1111	0.1301	**0.0000**
syn_att_r	0.3760	0.4085	0.6721	**0.3644**	0.3854	0.3861	0.3972	0.3667
syn_cimp	0.1772	0.2447	0.6358	0.0097	0.1761	0.2033	0.3131	**0.0001**
syn_cimp250	**0.0000**	0.1713	0.6470	0.0037	0.1733	0.1447	0.1553	0.0001
syn_cimp500	**0.0000**	0.1687	0.6399	0.0018	0.1737	0.1384	0.1345	0.0001
syn_cimp_r	0.3419	0.3951	0.6618	**0.3343**	0.3948	0.3731	0.3784	0.3420
syn_clss	**0.0000**	0.1920	0.6841	0.0055	0.1521	0.1920	0.2628	**0.0000**
syn_clss250	**0.0000**	0.1283	0.6445	0.0022	0.1552	0.1283	0.1232	**0.0000**
syn_clss500	**0.0000**	0.1263	0.6359	0.0011	0.1546	0.1263	0.1082	**0.0000**
syn_clss_r	**0.3555**	0.3897	0.6279	0.3560	0.4233	0.3866	0.3846	0.3680
syn_mimp	0.0755	0.1555	0.3717	0.0089	0.1465	0.1599	0.2429	**0.0000**
syn_mimp250	**0.0000**	0.1310	0.4933	0.0031	0.1714	0.1186	0.1346	**0.0000**
syn_mimp500	**0.0000**	0.1294	0.5161	0.0015	0.1681	0.1153	0.1134	**0.0000**
syn_mimp_r	0.3197	0.3608	0.5761	**0.3180**	0.3828	0.3569	0.3659	0.3231
syn_or	0.0633	0.1707	0.5388	0.0085	0.1545	0.1738	0.2630	**0.0000**
syn_or250	**0.0000**	0.1208	0.4907	0.0031	0.1640	0.1204	0.1343	**0.0000**
syn_or500	**0.0000**	0.1184	0.4926	0.0015	0.1695	0.1154	0.1150	**0.0000**
syn_or_r	0.3740	0.3974	0.6118	**0.3671**	0.4151	0.3906	0.4025	0.3709
syn_xor	0.1160	0.2879	0.6382	**0.0172**	0.3487	0.2735	0.3782	0.2322
syn_xor250	**0.0000**	0.1936	0.6478	0.0064	0.4747	0.1812	0.2115	0.2852
syn_xor500	**0.0000**	0.1954	0.6402	0.0031	0.3848	0.1772	0.1658	0.2840
syn_xor_r	0.3728	0.4343	0.6606	**0.3667**	0.5189	0.4121	0.4252	0.4996

Tabela C.5: Rezultati povprečnega logaritma izgube za sintetične nabore podatkov.

Ime nabora	CLT	CLF	Maj	NN	PLS	RF	CRF	LR
syn_and	0.9100	0.9200	0.5950	**1.0000**	0.9900	0.9700	0.9000	**1.0000**
syn_and250	1.0000	1.0000	0.6320	1.0000	0.9360	1.0000	1.0000	1.0000
syn_and500	1.0000	1.0000	0.6540	1.0000	1.0000	1.0000	1.0000	1.0000
syn_and_r	**0.8293**	0.8253	0.5960	0.8267	0.7973	0.8293	0.8187	0.8160
syn_att	0.9050	0.9700	0.6550	**1.0000**	**1.0000**	0.9800	0.9450	**1.0000**
syn_att250	1.0000	0.9680	0.6180	1.0000	1.0000	1.0000	0.9680	1.0000
syn_att500	1.0000	1.0000	0.6190	1.0000	1.0000	1.0000	1.0000	1.0000
syn_att_r	**0.8333**	0.8227	0.5507	0.8227	0.8240	0.8133	0.8227	0.8160
syn_cimp	0.8800	0.9600	0.6000	**1.0000**	0.9600	**1.0000**	0.9850	**1.0000**
syn_cimp250	1.0000	1.0000	0.6480	1.0000	0.9820	1.0000	1.0000	1.0000
syn_cimp500	1.0000	1.0000	0.6500	1.0000	0.9810	1.0000	1.0000	1.0000
syn_cimp_r	0.8453	0.8400	0.5827	0.8333	**0.8467**	**0.8467**	0.8400	**0.8467**
syn_clss	1.0000	1.0000	0.6200	1.0000	1.0000	1.0000	0.8800	1.0000
syn_clss250	1.0000	1.0000	0.6640	1.0000	1.0000	1.0000	1.0000	1.0000
syn_clss500	1.0000	1.0000	0.6700	1.0000	1.0000	1.0000	1.0000	1.0000
syn_clss_r	0.8307	0.8280	0.6413	0.8307	0.8200	**0.8333**	0.8253	0.8240
syn_mimp	0.9600	0.8900	0.8800	**1.0000**	0.9600	0.8900	0.8800	**1.0000**
syn_mimp250	1.0000	1.0000	0.8060	1.0000	1.0000	1.0000	0.9660	1.0000
syn_mimp500	1.0000	1.0000	0.7840	1.0000	1.0000	1.0000	1.0000	1.0000
syn_mimp_r	0.8400	0.8400	0.6867	0.8400	0.8293	**0.8413**	0.8187	0.8293
syn_or	0.9800	0.9900	0.7800	**1.0000**	1.0000	0.9950	0.9100	1.0000
syn_or250	1.0000	1.0000	0.8080	1.0000	1.0000	1.0000	0.9720	1.0000
syn_or500	1.0000	1.0000	0.8030	1.0000	1.0000	1.0000	1.0000	1.0000
syn_or_r	0.8160	0.8200	0.6680	0.8293	**0.8333**	0.8173	0.8267	0.8200
syn_xor	0.9500	0.9400	0.6350	**1.0000**	0.9650	0.9750	0.9650	0.9650
syn_xor250	**1.0000**	0.9580	0.6340	**1.0000**	0.9340	0.9580	0.9580	0.9340
syn_xor500	**1.0000**	0.9520	0.5990	**1.0000**	0.8690	0.9590	0.9900	0.8690
syn_xor_r	0.8333	0.8253	0.6013	0.8293	0.7427	0.8227	**0.8360**	0.7373

Tabela C.6: Rezultati povprečne točnosti za sintetične nabore podatkov.