



MATEMATIČKI FAKULTET
UNIVERZITET U BEOGRADU

Master rad

**Primena Gausovih uslovnih
slučajnih polja na predviđanje
vremena izvršavanja SAT
rešavača**

Autor:

Jelena Marković

Mentor:

doc. dr Mladen Nikolić

2018.

Beograd

Članovi komisije:

prof. dr Predrag Janičić

Matematički fakultet
Univerzitet u Beogradu

prof. dr Filip Marić

Matematički fakultet
Univerzitet u Beogradu

Datum odbrane: _____

Sadržaj

1	Uvod	1
2	SAT problem i SAT rešavači	4
2.1	Iskazna logika	4
2.1.1	Sintaksa iskazne logike	4
2.1.2	Semantika iskazne logike	5
2.2	SAT problem	6
2.2.1	NP kompletnost	6
2.2.2	Normalne forme	7
2.3	SAT rešavači	9
2.3.1	DPLL algoritam	9
2.3.2	CDCL sistem	10
2.3.3	Stohastički rešavači	11
3	Relevantni pojmovi mašinskog učenja	12
3.1	Definicija, kratak istorijat i primene	12
3.2	Osnovni pojmovi	13
3.2.1	Podela metoda mašinskog učenja	13
3.2.2	Vrste modela	14
3.2.3	Regresija	15
3.2.4	Regularizacija	15
3.2.5	Smanjenje dimenzionalnosti prostora atributa	17
3.2.6	Evaluacija i izbor modela	21
3.3	Metode mašinskog učenja za predviđanje vremena izvršavanja SAT rešavača	23
3.3.1	Linearna i grebena regresija	24

3.3.2	Regresiona stabla odlučivanja	26
3.3.3	Slučajne šume	28
3.3.4	Gausova uslovna slučajna polja	29
4	Ekperimentalna evaluacija	34
4.1	Detalji implementacije	34
4.2	Skupovi podataka	40
4.2.1	Podaci za eksperimentalnu evaluaciju modela grebene regresije i slučajnih šuma	40
4.2.2	Podaci za eksperimentalnu evaluaciju modela Gauso- vih uslovnih slučajnih polja	42
4.2.3	Atributi SAT instance	43
4.3	Opšte eksperimentalne postavke	45
4.4	Grebena regresija	46
4.5	Slučajne šume	50
4.6	Gausova uslovna slučajna polja	54
5	Zaključak	58
	Dodatak	60
A	Rezultati eksperimentalne evaluacije	60
A.1	Grebena regresija	60
A.2	Slučajne šume	63
A.3	Gausova uslovna slučajna polja	70
	Literatura	81

1 Uvod

Ispitivanje zadovoljivosti iskaznih formula je problem odlučivanja sa širokom praktičnom primenom (koristi se za formalnu verifikaciju hardvera i softvera, automatizaciju dizajna digitalnih kola, u granama veštačke inteligencije poput automatskog planiranja [14] itd). Ovo je takođe prvi problem za koji je 70-ih godina prošlog veka pokazano da je NP kompletan, istraživanjima koja su nezavisno sprovedi Stiven Kuk i Leonid Levin [5, 6]. Njima u čast, teorema koja dokazuje NP kompletnost SAT problema nazvana je Kuk-Levinova teorema.

Problem odlučivanja možemo posmatrati kao pitanje na koje su DA i NE jedini mogući odgovori. Ispitivanje zadovoljivosti iskaznih formula je problem odlučivanja koji je NP kompletan, što znači da pripada klasi složenosti NP (nedeterministički polinomijalna složenost) i da je NP težak, tj. da se svaki problem iz klase NP u polinomijalnom vremenu može svesti na ispitivanje zadovoljivosti iskaznih formula. Algoritmi za ispitivanje zadovoljivosti iskazne formule u određenom obliku nazivaju se SAT rešavači. Rešavanje problema iz klase NP zahteva u najgorem slučaju eksponencijalno vreme u odnosu na veličinu ulaza. Moderni SAT rešavači na instancama iz prakse uglavnom rade mnogo brže, ali često ispoljavaju velike varijacije u vremenu izvršavanja čak i na ulazima iste veličine. Sa druge strane, ista instanca na ulazu može da ispolji drastične razlike u trajanju rešavanja u zavisnosti od primenjenog algoritma. Za ovakvo ponašanje i dalje ne postoji adekvatno teorijsko objašnjenje. Imajući u vidu činjenicu da ne postoji univerzalno najbolji SAT rešavač, kao bitno pitanje nameće se odluka o izboru pogodnog SAT rešavača na osnovu osobina instance problema.

Metode nadgledanog mašinskog učenja, zasnovane na regresionim modelima obučenim na unapred poznatom skupu atributa instanci i performansama raznih rešavača, koriste se da predvide vreme izvršavanja datog rešavača za datu iskaznu formulu. Ovi modeli onda omogućavaju izbor pogodnog rešavača za određenu instancu SAT problema. Ideja modelovanja vremena izvršavanja algoritama nije nova; u ovom radu naveden je pregled metoda koje su ispitane u sličnim radovima. Takođe su (prema referentnom radu [1]) navedeni detalji modelovanja uz pomoć dveju od ovih metoda: grebene regresije (eng. ridge regression), koja je najčešće korišćena metoda, i slučajne šume (eng. random forest), što je trenutno najuspešnija metoda za modelovanje vremena izvršavanja algoritama. Evaluirano je nekoliko modela grebene regresije i slučajnih šuma, i upoređeni su rezultati evaluacije sa ocenama modela iz pomenutog referentnog rada.

U ovom radu takođe je ispitana mogućnost primene Gausovih uslovnih slučajnih polja (eng. GCRF - Gaussian Conditional Random Fields) u cilju boljeg predviđanja vremena izvršavanja skupa SAT rešavača na pojedinačnim iskaznim formulama. Ideja je da novi model, osim zavisnosti vremena izvršavanja od atributa instance SAT problema, uzima u obzir i sličnosti između SAT rešavača. GCRF je diskriminativni model, što znači da modeluje uslovnu raspodelu opažanja (što su u našem slučaju vremena izvršavanja SAT rešavača) za dati vektor atributa. Dodatna pretpostavka je da pomenuta uslovna raspodela ima Gausovu formu, odnosno formu normalne raspodele. Za razliku od ostalih primenjivanih metoda koje prave uobičajenu pretpostavku o nezavisnosti među predviđanjima, GCRF modeluje uslovnu raspodelu tako što pravi nagodbu između predviđanja jednog ili više drugih regresionih modela i predviđenih vremena izvršavanja za druge, slične rešavače. Drugim rečima, što su sličnija dva rešavača, očekujemo da su im sličnije performanse na novim instancama. Ovo utiče na uslovnu verovatnoću vrednosti opažane promenljive. Regresioni GCRF modeli su upoređeni sa standardnom tehnikom korišćenom za modelovanje relevantnog problema: slučajnom šumom.

Poglavlje 1 predstavlja uvod, kratak opis i značaj teme ovog rada. U poglavlju 2 govori se o definiciji i osnovnim pojmovima iskazne logike, problemu zadovoljivosti iskaznih formula i SAT rešavačima. Poglavlje 3 sadrži kratak pregled osnovnih pojmova mašinskog učenja, sa posebnim osvrtom na metode i tehnike korišćene za modelovanje vremena izvršavanja SAT rešavača u zavisnosti od atributa instance SAT problema. U ovom poglavlju opisuje se i metoda Gausovih uslovnih slučajnih polja i njena primena na predviđanje vremena izvršavanja SAT rešavača. Eksperimentalni deo bavi se evaluacijom nekoliko modela grebene regresije i slučajnih šuma, kao i evaluacijom modela dobijenog primenom Gausovih uslovnih slučajnih polja sa idejom unapređenja predviđanja slučajnih šuma. Opis eksperimenata i eksperimentalnih rezultata naveden je u poglavlju 4.

Eksperimentalnom evaluacijom utvrđeno je da među nekim atributima instanci SAT problema postoji korelacija koja onemogućava dobar kvalitet predviđanja linearnih modela grebene regresije. U referentnom radu [1] ovaj nedostatak se umanjuje primenom metode pohlepne selekcije atributa unapred, a u ovom radu su isprobane još neke tehnike za smanjenje dimenzionalnosti prostora atributa. Jedna od tih tehnika je rekursivna eliminacija atributa, koja je unapredila linearne modele evaluirane u ovom radu tako da imaju nešto bolji kvalitet predviđanja u odnosu na linearne modele iz referentnog rada [1]. Takođe je utvrđeno da metoda slučajne šume bolje modeluje vreme izvršavanja SAT rešavača od linearnih metoda. Štaviše, ek-

sperimentalna evaluacija pokazala je da nelinearni modeli slučajne šume bolje predviđaju vreme izvršavanja SAT rešavača od ostalih evaluiranih modela i da predloženi GCRF modeli ipak nisu ispunili očekivanja u smislu poboljšanja kvaliteta predviđanja slučajnih šuma.

2 SAT problem i SAT rešavači

Ovo poglavlje sadrži definicije osnovnih pojmova iskazne logike i problema zadovoljivosti iskaznih formula (prema [4]). Navedene su i neke od metoda za rešavanje problema zadovoljivosti iskaznih formula (kraće SAT problema, od eng. reči SATisfiability) na kojima su zasnovani moderni SAT rešavači [31].

2.1 Iskazna logika

Iskazna logika je moderna verzija Bulove algebre, koja je precizno predstavljena 1847. godine u knjizi Džordža Bula “Matematička analiza logike” (eng. The Mathematical Analysis of Logic). Pre Bula, formalistički pristup je bio najprisutniji u algebri.

Formalni sistem je precizno definisan sistem koji se sastoji od:

- konačnog skupa simbola (poznatog i kao alfabet; niske ovih simbola zovu se formule)
- gramatike koja predstavlja sintaksna pravila za građenje formula
- skupa aksioma (aksiom ili postulat je tvrđenje za koje znamo da je tačno i koje se ne dokazuje)
- skupa pravila izvođenja

Bul je primetio da je moguće uspostaviti formalni sistem izračunavanja nad logičkim iskazima, nalik na aritmetička izračunavanja.

2.1.1 Sintaksa iskazne logike

Alfabet iskazne logike čini unija sledećih skupova:

- prebrojivog skupa iskaznih promenljivih
- skupa logičkih operatora $\{\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow\}$
 - \neg je unarni operator koji predstavlja negaciju
 - \wedge je binarni operator konjunkcije (logičko *i*)
 - \vee je binarni operator disjunkcije (logičko *ili*)

- \Rightarrow je binarni operator implikacije
- \Leftrightarrow je binarni operator ekvivalencije
- skupa logičkih konstanti $\{\top, \perp\}$
- skupa pomoćnih simbola $\{(\ , \)\}$

Izraz iskazne logike je tvrdjenje koje može biti tačno ili netačno, a koje zovemo *iskazna formula* (u daljem tekstu su kao kraći sinonimi korišćeni izrazi *iskaz* ili *formula*). Grade se rekurzivno na osnovu sledećih pravila:

- iskazne promenljive i logičke konstante su iskazi (zovemo ih još i *atomički iskazi*)
- ako je F iskaz, onda je i $\neg F$ iskaz
- ako su F i G iskazi, onda su i $F \wedge G$, $F \vee G$, $F \Rightarrow G$, $F \Leftrightarrow G$ iskazi

Atomički iskaz ili negacija atomičkog iskaza kraće se zove *literal*, a disjunkcija literala *klauza*.

2.1.2 Semantika iskazne logike

Semantika iskazne logike govori o značenju iskaznih formula. Nalik algebarskim izrazima, iskazne formule mogu imati vrednost ukoliko im se dodele vrednosti promenljivih. Međutim, za razliku od algebarskih promenljivih čije su vrednosti predstavljene brojevima, iskazne promenljive imaju istinitosnu vrednost (tačno ili netačno).

Valuacije su funkcije koje određuju istinitosne vrednosti iskaznih promenljivih. Domen valuacije je bilo koji dvočlani skup u kome jedan element predstavlja tačno, a drugi netačno. Najčešće se koristi skup $\{0, 1\}$, gde 1 predstavlja tačno. Skup $\{\top, \perp\}$ se izbegava kako bi se napravila jasna razlika između sintaksnih simbola logičkih konstanti i semantičke vrednosti iskazne formule.

Interpretacija za valuaciju v je funkcija I_v koja skup iskaznih formula slika u skup $\{0, 1\}$ i definiše se na sledeći način:

- $I_v(p) = v(p)$ za svaku iskaznu promenljivu p
- $I_v(\top) = 1, I_v(\perp) = 0$

- $I_v(\neg F) = 1$ ako je $I_v(F) = 0$ i $I_v(\neg F) = 0$ ako je $I_v(F) = 1$
- $I_v(F \wedge G) = \begin{cases} 1, & \text{ako je } I_v(F) = 1 \text{ i } I_v(G) = 1 \\ 0, & \text{inače} \end{cases}$
- $I_v(F \vee G) = \begin{cases} 0, & \text{ako je } I_v(F) = 0 \text{ i } I_v(G) = 0 \\ 1, & \text{inače} \end{cases}$
- $I_v(F \Rightarrow G) = \begin{cases} 0, & \text{ako je } I_v(F) = 1 \text{ i } I_v(G) = 0 \\ 1, & \text{inače} \end{cases}$
- $I_v(F \Leftrightarrow G) = \begin{cases} 1, & \text{ako je } I_v(F) = I_v(G) \\ 0, & \text{inače} \end{cases}$

Kažemo da je valuacija v *zadovoljavajuća* za iskaznu formulu F (ili da predstavlja *model* iskazne formule F) ukoliko je $I_v(F) = 1$. Iskazna formula F je *zadovoljiva* ukoliko ima model.

Kao što u algebri postoje izrazi koji su uvek tačni (bez obzira na vrednosti promenljivih; npr. $(x + y)^2 = x^2 + 2xy + y^2$), tako i u iskaznoj logici postoje iskazi za koje je svaka valuacija model. Takvi iskazi zovu se *tautologije* ili *valjane formule*, dok su iskazi za koje ne postoji model *kontradikcije*. Ukoliko postoji valuacija koja nije model iskazne formule F , kažemo da je formula F *poreciva*.

2.2 SAT problem

Problem zadovoljivosti iskaznih formula, odnosno SAT problem, je problem odlučivanja da li je neka iskazna formula u KNF obliku zadovoljiva. Drugim rečima, za neki iskaz F pitamo se da li postoji valuacija v takva da je istinitosna vrednost iskaza F u valuaciji v $I_v(F) = 1$. Ovo je jedan od centralnih problema teorijskog računarstva.

2.2.1 NP kompletnost

Podsetimo se da je problem efikasno rešiv ako postoji algoritam koji sve njegove instance rešava u broju koraka koji je polinomijalno ograničen veličinom ulazne instance (vreme izvršavanja je $O(P(n))$, gde je n veličina ulazne instance). Za ovakav problem takođe kažemo da pripada klasi složenosti P .

Za SAT problem još uvek ne znamo da li pripada klasi P , ali znamo da pripada klasi NP - klasi nedeterministički polinomijalne složenosti. Iskaz sa n promenljivih ima 2^n interpretacija, a za svaku interpretaciju možemo u polinomijalnom vremenu da proverimo istinitost. Dakle, algoritam koji rešava SAT problem u najgorem slučaju se završava nakon $O(2^n)$ koraka, gde je n veličina ulaza. Nije poznato da li postoji polinomijalni algoritam. SAT problem je takođe NP težak jer je svaki problem iz klase NP polinomijalno svodljiv na problem ispitivanja zadovoljivosti iskaza. Ovo tvrđenje, poznato kao Kuk-Levinova teorema, nezavisno su dokazali Stiven Kuk 1971. [5] i Leonard Levin 1973. godine [6]. NP teški problemi koji pripadaju klasi NP kraće se zovu NP kompletni problemi. SAT problem je prvi problem za koji je dokazano svojstvo NP kompletnosti.

2.2.2 Normalne forme

SAT problem se tehnički jednostavnije rešava ukoliko je u nekoj od *normalnih formi*:

- *KNF - konjuktivna normalna forma* (iskaz oblika $F_1 \wedge F_2 \wedge \dots \wedge F_n$, pri čemu je F_i ($1 \leq i \leq n$) disjunkcija literala, odnosno klauza)
- *DNF - disjunktivna normalna forma* (iskaz oblika $F_1 \vee F_2 \vee \dots \vee F_n$, pri čemu je F_i ($1 \leq i \leq n$) konjunkcija literala)

Ukoliko je iskaz F logički ekvivalentan iskazu G (u svakoj valuaciji F i G imaju iste vrednosti, tj. svaki model za F je ujedno i model za G i obratno), i iskaz G je u konjuktivnoj normalnoj formi, onda kažemo da je G KNF za F . Analogno važi za DNF. Kako je $\{\wedge, \vee, \neg\}$ potpun skup logičkih operatora [4], svaki iskaz ima KNF, odnosno DNF. Štaviše, svaki iskaz ima više logički ekvivalentnih iskaza u KNF (DNF). Takođe, formula koja je u KNF (DNF) može biti logički ekvivalentna sa više iskaznih formula. Zadovoljivost se trivijalno ispituje za formulu u DNF, ali postupak prevođenja u DNF nije trivijalan. Sa druge strane, postoji efikasan postupak za prevođenje formule u KNF, uz napomenu da ne čuva ekvivalentnost, ali čuva zadovoljivost (što je dovoljno za rešavanje SAT problema). Naime, osnovni algoritam za prevođenje iskaza u ekvivalentnu konjuktivnu (disjunktivnu) normalnu formu podrazumeva sledeće korake:

1. eliminacija ekvivalencije
 - iterativno menjamo svako pojavljivanje $F \Leftrightarrow G$ sa $(F \Rightarrow G) \wedge (G \Rightarrow F)$

2. eliminacija implikacije
 - iterativno menjamo svako pojavljivanje $F \Rightarrow G$ sa $\neg F \vee G$
3. primena De Morganovih pravila
 - iterativno menjamo svako $\neg(F \vee G)$ sa $\neg F \wedge \neg G$ i svako $\neg(F \wedge G)$ sa $\neg F \vee \neg G$
4. eliminacija višestruke negacije
 - svako pojavljivanje $\neg\neg F$ zamenimo sa F
5. primena distributivnosti
 - u slučaju prevođenja u KNF, iterativno zamenimo svako pojavljivanje $F \vee (G \wedge H)$ sa $(F \vee G) \wedge (F \vee H)$ i $(F \wedge G) \vee H$ sa $(F \vee H) \wedge (G \vee H)$;
 - u slučaju prevođenja u DNF, iterativno zamenimo svako pojavljivanje $F \wedge (G \vee H)$ sa $(F \wedge G) \vee (F \wedge H)$ i $(F \vee G) \wedge H$ sa $(F \wedge H) \vee (G \wedge H)$

Navedene transformacije čuvaju ekvivalenciju, ali ovaj postupak može dovesti do eksponencijalnog uvećanja formule. To je neizbežno ako želimo da dobijena formula bude ekvivalentna polaznoj formuli, ali za ispitivanje zadovoljivosti dovoljno nam je da bude ispunjen nešto slabiji uslov: da dobijena formula bude ekvizadovoljiva sa polaznom. Ispostavlja se da postoji algoritam koji polaznu formulu prevodi u *definicionu (Cajtinovu) KNF* [7] koja je samo za konstantni faktor veća od polazne, a pritom je ekvizadovoljiva. Ovo se postiže uvođenjem novih promenljivih za podformule i uvođenjem definicionih ekvivalencija koje se posle klasično prevode u KNF.

Primer Cajtinove transformacije:

1. $p \vee (q \wedge r)$ (polazna formula)
2. $(p \vee s_1) \wedge (s_1 \Leftrightarrow (q \wedge r))$
3. $s_2 \wedge (s_1 \Leftrightarrow (q \wedge r)) \wedge (s_2 \Leftrightarrow (p \vee s_1))$
4. $s_2 \wedge (\neg s_1 \vee q) \wedge (\neg s_1 \vee r) \wedge (\neg s_2 \vee p) \wedge (\neg s_2 \vee s_1)$ (Cajtinova KNF)

2.3 SAT rešavači

Postoji više pristupa rešavanju SAT problema:

- metod istinitosnih tablica - naivan metod, ne koristi se u praksi jer su poznate značajno efikasnije metode
- klauzalni algoritmi za ispitivanje zadovoljivosti formula u KNF
 - iskazna rezolucija
 - DPLL (Dejvis-Patnam-Logman-Loveland) procedura i njena unapređenja
 - stohastičke metode

Moderni SAT rešavači uglavnom koriste algoritme zasnovane na sistemu pravila sa analizom konflikata (eng. CDCL - Conflict Driven Clause Learning; unapređenje DPLL procedure). Međutim, algoritmi zasnovani na randomizaciji, odnosno stohastičkoj lokalnoj pretrazi, pokazali su bolje rezultate u nekim primenama poput automatskog planiranja.

Bitna osobina procedure za ispitivanje zadovoljivosti iskaza je saglasnost: ako procedura prijavi nezadovoljivost, polazni iskaz je zaista nezadovoljiv, odnosno ako procedura prijavi zadovoljivost, polazni iskaz je zaista zadovoljiv. Ostale poželjne karakteristike su zaustavljanje (za svaki ulaz procedura se zaustavlja u konačnom broju koraka) i potpunost (ako je polazni iskaz nezadovoljiv, procedura će prijaviti nezadovoljivost; slično u slučaju da je polazni iskaz zadovoljiv). Algoritmi zasnovani na DPLL proceduri imaju sva tri svojstva, dok stohastički SAT rešavači ne garantuju zaustavljanje, pa samim tim nisu ni potpuni.

2.3.1 DPLL algoritam

Ovaj algoritam objavili su 1962. godine Martin Dejvis, Džordž Logman i Donald Loveland [8] kao unapređenje Dejvis-Patnam algoritma (1961) [9] zasnovanog na metodu rezolucije.

DPLL algoritam primenjuje se na formulu u KNF, koju možemo predstaviti i kao skup klauza. Klauzu možemo predstaviti kao skup literala. Možemo pretpostaviti da KNF ne sadrži dve iste klauze i da nijedna klauza ne sadrži dva suprotna literala (l i \bar{l}). DPLL procedura zasniva se na pohlepnoj pretrazi za zadovoljavajućom valuacijom. Naivna pretraga podrazumevala bi

da izaberemo neki literal l i damo mu vrednost tačno, dok njemu suprotnom literalu \bar{l} damo vrednost netačno. Nakon toga formula se uprosti (izbace se klauze koje sadrže l , a \bar{l} se izbací iz svih klauza koje ga sadrže) i ponavlja se postupak sa sledećim literalom. Ako iscrpljivanje svih literala ne prijavi zadovoljivost, vraćamo se na početak i isprobavamo suprotnu kombinaciju: l postavimo na netačno, a \bar{l} na tačno. Pretraga se zaustavlja kada skup klauza ostane prazan (u tom slučaju formula je zadovoljiva) ili kada se pojavi prazna klauza (to znači da je formula nezadovoljiva).

Naivna DPLL pretraga može da se poboljša smanjenjem prostora pretrage. Sa tim ciljem uvode se dva nova koraka:

- propagacija jediničnih klauza
 - ako formula sadrži klauzu sa tačno jednim literalom $\{l\}$, taj literal ne može biti netačan u zadovoljavajućoj valuaciji
- eliminacija čistih literala
 - ako se u formuli pojavljuje literal l , a ne pojavljuje njemu suprotan literal \bar{l} , taj literal može da se postavi na tačno

Nakon toga bira se literal koji postavljamo na tačno (korišćenjem pametnih heuristika) i dalje se postupa kao kod naive pretrage.

2.3.2 CDCL sistem (eng. Conflict Driven Clause Learning)

Ispostavlja se da je DPLL procedura podložna nepotrebnoj redundantnosti kada dođe do konflikata (tj. kada pretpostavka o vrednosti nekog literala vodi do nekompatibilnosti sa prethodnim pretpostavkama). Ovaj nedostatak može da se reši analizom konflikata i povratnim skokovima [10, 11, 12], pri čemu se "uče" klauze povratnih skokova (formula se njima proširuje). Kada dodje do konflikta, analiziraju se koraci koji su doveli do tog konflikta. Kao rezultat analize dobijamo klauzu koja se pamti (uči) tako što se dodaje u polaznu formulu. Naučene klauze predstavljaju semantičku posledicu polaznog skupa klauza, tj. mogu biti izvedene iz polaznog skupa klauza primenom rezolucije. Korak povratnog skoka ne mora biti hronološki poslednji korak nepotpunog grananja stabla pretrage, odnosno poslednji korak u kome smo odlučili da nekom literalu dodelimo neku vrednost a nismo isprobali pretragu sa suprotnom vrednošću, jer se može desiti da poslednja pretpostavka nije relevantna za konflikt koji je nastupio, pa samim tim suprotna pretpostavka vodi ka istom konfliktu. Umesto toga vraćamo se na korak odluke u kome je dodeljena vrednost nekom od literala koji pripada naučenoj klauzi. Naučena

klauza je značajna u situacijama kada se ponovo nađemo u kontekstu u kom smo bili kad je klauza naučena, jer možemo da ustanovimo njenu netačnost u tekućoj valuaciji i odsećemo granu pretrage.

Očigledan nedostatak ovakvog pristupa je to što broj naučenih klauza može da postane prevelik, pa se neke klauze posle izvesnog vremena uklanjaju. Nekad je u toku pretrage korisno prekinuti i započeti ponovo, jer postoji šansa da će nas naučene klauze odvesti u neku lakšu granu pretrage. Obe tehnike kontrolišu se heuristikama.

CDCL algoritam je implementiran u mnogim popularnim SAT rešavačima (MiniSAT, Zchaff SAT, Z3, ManySAT itd). Doveo je do značajnog poboljšanja performansi SAT rešavača na praktičnim instancama SAT problema.

2.3.3 Stohastički rešavači

Osim rešavača zasnovanih na CDCL okviru, postoje i rešavači zasnovani na drugačijem pristupu ispitivanju zadovoljivosti iskaza u KNF obliku: stohastičkoj lokalnoj pretrazi [13]. Osnovna ideja je jednostavna: najpre se svim promenljivim dodele slučajne vrednosti. Ukoliko je slučajno izabrana valuacija zadovoljavajuća, završili smo. Inače se jednoj promenljivoj zameni vrednost suprotnom, a ostalima se dodele nove slučajne vrednosti i postupak se ponavlja dok se ne maksimizuje broj zadovoljenih klauza (ili minimizuje broj nezadovoljenih klauza). Dva najpoznatija stohastička rešavača, WalkSAT i GSAT, razlikuju se po izboru promenljive kojoj se obrće vrednost:

- WalkSAT algoritam najpre izabere klauzu koju trenutna valuacija ne zadovoljava, a zatim obrne vrednost neke promenljive iz te klauze. Promenljiva se bira tako da se minimizuje broj klauza koje postaju nezadovoljive nakon obrtanja, sa određenom verovatnoćom slučajnog izbora.
- GSAT algoritam obrne vrednost neke promenljive tako da minimizuje broj nezadovoljenih klauza u tako dobijenoj valuaciji, ili sa određenom verovatnoćom vrši slučajan izbor promenljive za obrtanje.

Dešava se da se ovi algoritmi zaglave u lokalnim minimumima za broj nezadovoljenih klauza. Ovaj problem može da se reši tako što se uzme nova slučajna valuacija i počne ispočetka.

WalkSAT se pokazao naročito uspešno u nalaženju optimalne valuacije za iskazni model problema automatskog planiranja.

3 Relevantni pojmovi mašinskog učenja

Poglavlje 3 posvećeno je mašinskom učenju. Definicije i pregled osnovnih pojmova mašinskog učenja oslanjaju se na [3]. Analizirane su metode mašinskog učenja koje se koriste za modelovanje vremena izvršavanja SAT rešavača: grebena regresija i slučajna šuma (eng. random forest).

3.1 Definicija, kratak istorijat i primene

Mašinsko učenje je disciplina koja predstavlja podoblast veštačke inteligencije. Prva poznata definicija veštačke inteligencije potiče iz 50ih godina prošlog veka: “Veštačka inteligencija predstavlja mašine koje mogu samostalno da izvršavaju zadatke karakteristične ljudskoj inteligenciji” (Džon Merkarti, 1956). Nedugo zatim (1959) Artur Semjuel je definisao mašinsko učenje kao “sposobnost mašina da uče iako nisu eksplicitno isprogramirane”. Dakle, umesto programiranja rutina sa specifičnim instrukcijama za izvršavanje određenog zadatka, mašinsko učenje podrazumeva trening algoritma koji se sam prilagođava i unapređuje tako da donosi bolje zaključke ili tačnije vrši predviđanja na osnovu ulaznih podataka. Smatra se da osnove prirodne inteligencije čine induktivno i deduktivno zaključivanje, a njihovi pandani u veštačkoj inteligenciji su mašinsko učenje i automatsko rezonovanje. Za razliku od računarske logike koja se bavi automatizacijom deduktivnog rezonovanja (od opšteg ka specifičnom), mašinsko učenje se bavi generalizacijom (od specifičnih obzervacija nad ograničenim uzorkom dolazimo do uopštenih, univerzalnih zaključaka, što čini induktivni pristup) [3].

Iako je ogromnu popularnost počelo da stiče tek krajem prošlog veka, mašinsko učenje je prisutno znatno duže. 50-ih godina prošlog veka razvijen je prvi perceptron - algoritam koji prepoznaje jednostavne zakonitosti i predstavlja osnovu za razvoj neuronskih mreža (tvorac je bio američki psiholog Frenk Rozenblat 1957. godine). Moderne neuronske mreže su dobrim delom zaslužne za trenutnu popularnost mašinskog učenja, iako su njihov razvoj 90-ih godina usporile metode zasnovane na kernelima, koje su tada pokazivale bolje rezultate na relevantnim problemima.

S obzirom na to da je u ljudskoj prirodi želja da se bolje razume inteligencija i sposobnost učenja, razvoj discipline poput mašinskog učenja je logična posledica. Ako na to dodamo širok spektar praktičnih primena, jasno je zašto se ova disciplina istražuje i eksploatiše kako u akademskim, tako i u industrijskim krugovima. Neke od uspešnih primena mašinskog učenja već su postale deo naše svakodnevice i rešavaju vrlo značajne praktične probleme. Jedna od

zanimljivih primena je i predviđanje vremena izvršavanja raznih algoritama na instancama NP kompletnih problema, koja je u kontekstu SAT problema detaljnije analizirana u nastavku.

3.2 Osnovni pojmovi

Mašinsko učenje je bogata disciplina koju čini mnoštvo metoda i tehnika za rešavanje raznovrsnih problema. U nastavku se razmatraju osnovni pojmovi mašinskog učenja. Govori se o osnovnoj podeli metoda mašinskog učenja, vrstama modela prema modelovanim zavisnostima i prema načinu reprezentacije, opisane su neke korisne tehnike koje se koriste prilikom treniranja modela poput regularizacije i smanjenja dimenzionalnosti prostora atributa i objašnjen je proces ocene kvaliteta različitih modela u cilju izbora najboljeg.

3.2.1 Podela metoda mašinskog učenja

Osnovna podela metoda mašinskog učenja se vrši na osnovu prirode problema učenja na navedene tri kategorije.

- Nadgledano učenje
 - Ukoliko imamo problem koji podrazumeva predviđanje neke vrednosti na osnovu nekih opažanja, rešavanju se može pristupiti tehnikom nadgledanog učenja. Potrebno je ustanoviti vezu između vrednosti ulaza (*atributa*) i ciljne promenljive koju želimo da predvidimo. Cilj je naći funkciju koja opisuje ovu vezu. Ova funkcija naziva se *model*. Od modela se očekuje da prilikom predviđanja ciljne promenljive na osnovu vrednosti atributa retko pravi velike greške. Za učenje je potrebno imati podatke o nekom uzorku instanci polaznog problema gde su poznate i vrednosti atributa i vrednosti ciljne promenljive. Metode nadgledanog učenja dalje delimo na osnovu tipa ciljne promenljive na:
 - regresione metode - podrazumeva predviđanje vrednosti ciljne promenljive definisane na neprekidnom domenu
 - klasifikacione metode - ciljna promenljiva pripada konačnom diskretnom domenu
- Nenadgledano učenje
 - Nenadgledano učenje se obično bavi pronalaženjem neke strukture u podacima, pošto nije poznata informacija o tome šta treba naučiti.

- Učenje potkrepljivanjem (eng. reinforcement learning)
Učenje potkrepljivanjem inspirisano je bihevioralnom psihologijom. Koristi se u problemima koji podrazumevaju preduzimanje niza akcija u određenom okruženju tako da se maksimizuje kumulativna nagrada, pri čemu se akcije nagrađuju proporcionalno tome koliko su doprinele približavanju nekom zadatom cilju. Važno svojstvo je da se prilikom obučavanja ne zna koja je akcija bila prava u kom stanju, već samo do kakvog je ishoda doveo niz preduzetih akcija.

3.2.2 Vrste modela

Prema tome koje zavisnosti modeluju, modeli mašinskog učenja dele se na:

- probabilističke generativne
 - modeluju zajedničku raspodelu podataka, koja opisuje i zavisnosti između atributa i ciljne promenljive, i međusobne zavisnosti atributa
- probabilističke diskriminativne
 - modeluju uslovnu raspodelu koja opisuje zavisnost ciljne promenljive od atributa
- neprobabilističke diskriminativne
 - ovi modeli ne daju nikakvu informaciju o raspodeli, modeluju samo funkciju $y = f(x)$

Prema načinu reprezentacije, modele mašinskog učenja delimo na dve kategorije.

- Parametarski modeli
Parametarska reprezentacija modela unapred pretpostavlja formu modela i predstavlja funkciju $f_{\omega}(x)$ gde je x vektor vrednosti atributa, a ω vektor parametara modela fiksirane dužine. Izbor modela se svodi na izbor parametara ω tako da se minimizuje greška predviđanja
- Neparametarski modeli
U slučaju neparametarske reprezentacije, forma modela nije zadata unapred i zavisi od podataka. Možda je neintuitivan naziv “neparametarski modeli” s obzirom na to da ovi modeli imaju parametre, ali se ne mogu izraziti pomoću fiksiranog broja parametara koji ne zavisi od broja instanci u skupu za obučavanje. Drugim rečima kompleksnost modela se povećava sa povećanjem broja instanci, za razliku od parametarskih modela koje definišu vrednosti fiksiranog broja parametara.

Glavni nedostatak parametarskih reprezentacija je to što modelu unapred zadajemo određenu formu. Neparametarske reprezentacije su fleksibilnije u tom pogledu jer prave manje pretpostavki o formi modela. Ipak, i neparametarski modeli imaju nedostatke, a glavni nedostaci su računaska zahtevnost predviđanja (jer su modeli izraženi u terminima podataka za obučavanje) i opasnost od preprilagođavanja (što je model fleksibilniji, to se lakše prilagodi podacima, što može negativno uticati na moć generalizacije; preprilagođavanje i tehnike za kontrolisanje preprilagođavanja detaljnije se razmatraju u poglavlju 3.2.4).

3.2.3 Regresija

Regresija je vrsta nadgledanog učenja i predstavlja problem predviđanja neprekidne ciljne promenljive. Kao što je pomenuto, u slučaju parametarske reprezentacije forma modela se definiše kao $f_\omega(x)$, a izbor modela zapravo je izbor parametara ω takvih da je greška predviđanja minimalna. Neka je N veličina uzorka, y_i vrednosti ciljne promenljive, a x_i vektori vrednosti atributa za $1 \leq i \leq N$. Ukoliko sa $L(y_i, f_\omega(x_i))$ označimo funkciju greške predviđanja (eng. loss function) na jednoj instanci uzorka, izbor parametara modela (poznat i kao *obučavanje* ili *treniranje* modela) može se definisati kao minimizacioni problem srednje greške predviđanja:

$$\min_{\omega} \frac{1}{N} \sum_{i=1}^N L(y_i, f_\omega(x_i))$$

U regresionim problemima za funkciju greške obično se uzima kvadratna greška: $L(y_i, f_\omega(x_i)) = (y_i - f_\omega(x_i))^2$, pa se problem svodi na minimizaciju srednjekvadratne greške:

$$\min_{\omega} \frac{1}{N} \sum_{i=1}^N (y_i - f_\omega(x_i))^2$$

Linearna regresija (poglavlje 3.3.1) je predstavnik parametarskih regresionih modela mašinskog učenja.

Postoje i neparametarski regresioni modeli mašinskog učenja. U poglavlju 3.3.2 razmatra se jedan takav model i odgovarajući optimizacioni problem.

3.2.4 Regularizacija

Obučavanje modela možemo da posmatramo i kao prilagođavanje modela podacima. Kako je vrlo važno svojstvo modela da dobro generalizuje, od-

nosno da greška predviđanja na novim, neviđenim podacima ne bude velika, ispostavlja se da nije dobro da model bude previše prilagođen podacima na kojima je obučavan. Fleksibilni modeli su vrlo skloni preprilagođavanju, često mogu odlično da se prilagode podacima za učenje, ali na novim podacima da daju slaba predviđanja, zato što su se previše prilagodili šumu u podacima ili zato što u skupu za obučavanje svi podaci pripadaju nekom specifičnom prostoru atributa. Za uspešnu generalizaciju potrebno je da kontrolišemo fleksibilnost reprezentacije modela u toku obučavanja, a tehnika kojom se to postiže naziva se *regularizacija*.

Razmotrimo najpre slučaj parametarskih modela, gde je nalaženje optimalnih parametara zapravo prilagođavanje modela podacima. Regularizacija predstavlja izmenu minimizacionog problema dodavanjem regularizacionog izraza $R(\omega)$ koji otežava prilagođavanje modela podacima. Novi minimizacioni problem izgleda ovako:

$$\min_{\omega} \frac{1}{N} \sum_{i=1}^N L(y_i, f_{\omega}(x_i)) + \lambda R(\omega)$$

λ u prethodnom izrazu predstavlja *regularizacioni parametar*¹. Za regularizacioni izraz se obično uzima kvadrat l_2 norme vektora parametara modela, pa minimizacioni problem postaje sledeći izraz:

$$\min_{\omega} \frac{1}{N} \sum_{i=1}^N L(y_i, f_{\omega}(x_i)) + \|\omega\|^2$$

Kada je $\lambda = 0$, regularizacije nema i model lako može da se preprilagodi. Sa druge strane, za jako veliko λ srednja vrednost greške predviđanja u minimizacionom problemu je zanemarljiva i teži se ka minimizaciji regularizacionog izraza, odnosno ka minimizaciji izraza $\|\omega\|^2$, koji minimum dostiže kada je $\omega = 0$, čime dobijamo potpuno neprilagodljiv model. Izborom regularizacionog parametra kontroliše se koliki značaj ima greška prilagođavanja, a koliki značaj pripada regularizacionom izrazu.

Kod neparametarskih modela koji su zasnovani na podacima vrlo lako može doći do preprilagođavanje usled njihove fleksibilnosti. Neke od tehnika regularizacije za stabla odlučivanja pomenute su u poglavlju 3.3.2.

¹Da bi se izbegla zabuna, svi parametri minimizacionog problema koji nisu parametri modela i ne dobijaju se obučavanjem zovu se *metaparametri modela*; regularizacioni parametar je metaparametar modela.

3.2.5 Smanjenje dimenzionalnosti prostora atributa

Ponekad se prilikom konstrukcije modela koriste tehnike smanjenja dimenzionalnosti prostora atributa. Postoje tehnike selekcije podskupa najinformativnijih atributa i metode numeričke linearne algebre za smanjenje dimenzionalnosti prostora atributa: singularna dekompozicija (eng. SVD - Singular Value Decomposition) i analiza glavnih komponenti (eng. PCA - Principal Component Analysis). Smisao ovih tehnika objašnjava se sledećim razlozima:

- model sa manjim brojem atributa je jednostavniji i lakši za interpretaciju
- merenje vrednosti pojedinih atributa može biti vremenski zahtevno, komplikovano ili skupo, pa ako ti atributi nisu mnogo informativni, korisno je izbaciti ih
- kraće vreme obučavanja
- izbegava se *prokletstvo dimenzionalnosti* (eng. curse of dimensionality)²
- smanjenje dimenzionalnosti prostora atributa je takođe vid regularizacije pošto vodi ograničenju prilagođavanja modela

Selekcija atributa Metode za selekciju atributa biraju attribute tako da zadrže ili unaprede kvalitet predviđanja. Postoje tri tipa ovih metoda.

- Metode filtriranja skupa atributa (eng. filter methods)
Ove metode koriste statističke mere da dodele skor svakom atributu. Nakon što su atributi rangirani po skor, po nekom kriterijumu se izbacuju ili zadržavaju u skupu.
- Metode omotača (eng. wrapper methods)
Metode omotača su zapravo metode pretrage koje traže najbolji podskup skupa atributa. Za svaki podskup izvrši se evaluacija nekog prediktivnog modela, pri čemu ocena modela predstavlja skor datog podskupa. Bira se podskup sa najboljim skorom. Proces pretrage ima razne varijante, podskupovi mogu da se biraju pohlepnim algoritmom, korišćenjem heuristika ili korišćenjem stohastičkih metoda.

²Kod probablističkih generativnih modela, u slučaju visoke dimenzionalnosti prostora atributa, potrebna nam je velika količina podataka za obučavanje da bismo bili sigurni da imamo uzorak sa svim mogućim kombinacijama vrednosti atributa; štaviše, broj potrebnih instanci raste eksponencijalno sa povećanjem dimenzionalnosti prostora atributa.

- Ugrađene metode (eng. embedded methods)
Ove metode su “ugrađene” u neke reprezentacije modela mašinskog učenja, poput laso regresije. Laso (eng. lasso - least absolute shrinkage and selection operator) je zapravo l_1 regularizacija - u minimizacionom problemu regresije kao regularizacioni izraz upotrebljava se l_1 norma vektora parametara modela:

$$\|\omega\|_1 = \sum_{i=1}^n |\omega_i|$$

Za razliku od l_2 regularizacije koja teži smanjenju apsolutnih vrednosti parametara modela tako da veliki broj parametara postane mali, ali različit od 0, laso regularizacija vodi ka tome da vrednosti nekih parametara postanu baš 0. To znači da su odgovarajući atributi neinformativni i da ih možemo izbaciti iz skupa atributa.

Tehnike selekcije atributa smanjuju dimenzionalnost biranjem podskupa atributa koji nosi najveću količinu informacije. Sa druge strane, metode numeričke linearne algebre koje takođe vode smanjenju dimenzionalnosti prostora atributa koriste aproksimaciju matrice podataka matricom nižeg ranga (singularna dekompozicija) i projektovanje prostora atributa na potprostor čiju bazu čine vektori duž kojih je varijabilnost podataka najveća (analiza glavnih komponenti). Osnovna razlika je u tome što ove metode vrše i nekakvu transformaciju podataka, dok kod selekcije vrednosti atributa ostaju nepromenjene. Navedene metode numeričke linearne algebre (prema [29]) su detaljnije objašnjene u nastavku.

Singularna dekompozicija matrice U numeričkom rešavanju problema linearne algebre dekompozicije matrica pružaju veću numeričku stabilnost i brzinu metoda. Singularna dekompozicija matrice A dimenzija $m \times n$ (pretpostavimo bez umanjenja opštosti da je $m \geq n$) je razlaganje na proizvod ortogonalne matrice U ($m \times m$), dijagonalne matrice Σ ($m \times n$) i transponata ortogonalne matrice V ($n \times n$):

$$A = U\Sigma V^T$$

Matrica Σ ima $m - n$ vrsta sačinjenih isključivo od nula, pa je matricu A moguće razložiti i na matricu ortonormiranih kolona U' ($m \times n$), dijagonalnu matricu Σ' ($n \times n$) i transponat ortogonalne matrice V ($n \times n$):

$$A = U'\Sigma'V^T$$

Dijagonalni elementi matrice Σ predstavljaju *singularne vrednosti* matrice A , sortirane su u nerastućem poretku i označavaju se sa σ_i , $i = 1, 2, \dots, n$. U implementacijama singularne dekompozicije obično se ne čuva cela matrica Σ , već samo niz singularnih vrednosti.

Neka od bitnih svojstava singularne dekompozicije su sledeća:

- singularne vrednosti σ_i matrice A su nenegativne i predstavljaju kvadratne korene sopstvenih vrednosti matrica AA^T i $A^T A$
- rang matrice A predstavlja broj singularnih vrednosti različitih od nule
- kolone matrice U su sopstveni vektori matrice AA^T , odnosno *levi singularni vektori* matrice A
- kolone matrice V su sopstveni vektori matrice $A^T A$, odnosno *desni singularni vektori* matrice A
- kolone matrice U koje odgovaraju singularnim vrednostima različitim od nule predstavljaju ortonormiranu bazu prostora kolona matrice A
- kolone matrice V koje odgovaraju singularnim vrednostima različitim od nule predstavljaju ortonormiranu bazu prostora vrsta matrice A
- $\|A\| = \sigma_1$
- uslovljenost matrice A je $Cond(A) = \frac{\sigma_1}{\sigma_n}$

Postoje razne primene singularne dekompozicije, ali one koje su zanimljive u kontekstu mašinskog učenja tiču se konstrukcije vektorskih reprezentacija raznih entiteta - npr. vektorska reprezentacija reči na osnovu matrice frekvencija reči u dokumentima, zatim vektorska reprezentacija korisnika u sistemu preporučivanja filmova itd. U ovim problemima se vrši aproksimacija matrice podataka nekom matricom nižeg ranga, jer se očekuje da najmanjim singularnim vrednostima odgovara šum u podacima; ovaj korak možemo posmatrati kao korak učenja [29].

Analiza glavnih komponenti U praksi se podaci često predstavljaju visokodimenzionalnim vektorima, iako većina vektora podataka često leži blizu nekog niskodimenzionog potprostora. Projektovanjem na taj potprostor gubi se deo informacije, ali se dimenzionalnost podataka može (ponekad drastično) smanjiti, što je poželjno zbog vizualizacije i skladištenja podataka, kao i zbog toga što algoritmi mašinskog učenja često daju bolje rezultate ukoliko podaci

imaju nižu dimenzionalnost. Takođe je poželjno eliminisati korelacije među atributima, jer atributi koji su (skoro) linearno zavisni nemaju informativni značaj.

Analiza glavnih komponenti je metoda analize podataka kojom se traži skup vektora duž kojih je varijacija podataka najveća, tako da projekcije podataka na te vektore nisu korelirane [29]. Ti vektori predstavljaju novu bazu prostora podataka, i zovu se *glavne komponente*. Kako bi projekcije bile nekorelirane, glavne komponente moraju biti ortogonalne. Proces konstrukcije glavnih komponenti svodi se na pronalaženje vektora duž kojeg podaci najviše variraju (projekcije podataka na pravac tog vektora su najraštrkanije), varijabilnost duž tog vektora se ukloni i proces se nastavlja.

Označimo sa X matricu centriranih podataka (vrste su instance, kolone atributi, proseci svih kolona su jednaki 0³). Prva glavna komponenta se dobija kao rešenje sledećeg problema:

$$\max_{\|v\|=1} \sum_{i=1}^n (x_i v)^2 = \max_{\|v\|=1} \frac{\|Xv\|^2}{\|v\|^2} = \max_{\|v\|=1} \frac{\|Xv\|}{\|v\|}$$

Maksimalna vrednost izraza koji se maksimizuje je $\|X\|$, što je takođe najveća singularna vrednost σ_1 matrice X , odnosno koren po modulu najveće sopstvene vrednosti λ_1 matrice $X^T X$, pa se za prvu glavnu komponentu uzima odgovarajući sopstveni vektor v_1 matrice $X^T X$. Sledeća glavna komponenta dobija se kada se prethodni postupak ponovi na matricu $X_1 = X - Xv_1v_1^T$. Ispostavlja se da je druga glavna komponenta zapravo sopstveni vektor matrice $X^T X$ koji odgovara sopstvenoj vrednosti λ_2 . Nastavljanjem postupka utvrđuje se da su glavne komponente $[v_1, v_2, \dots, v_n]$ zapravo sopstveni vektori (sortirani opadajuće po odgovarajućim modulima sopstvenih vrednosti) matrice $X^T X$. Drugi način nalazjenja glavnih komponenti je singularnom dekompozicijom matrice X : $X = U\Sigma V^T$, gde su kolone matrice V sopstveni vektori matrice $X^T X$. Glavne komponente predstavljaju novu (ortonormiranu) bazu prostora podataka u čijim koordinatama matrica X postaje XV . Smanjenje dimenzionalnosti postiže se izostavljanjem manje važnih glavnih komponenti, odnosno izostavljanjem kolona matrice V koje odgovaraju manje važnim komponentama.

Glavnoj komponenti v_i može se pridružiti udeo varijanse koju ona predstavlja:

$$\frac{\sigma_i^2}{\sum_{j=1}^n \sigma_j^2}$$

³Podaci se mogu centrirati tako što se od elemenata svake kolone oduzme prosek te kolone.

Udeo varijanse objašnjen pomoću prvih k komponenti predstavljen je sledećom sumom:

$$\sum_{i=1}^k \frac{\sigma_i^2}{\sum_{j=1}^n \sigma_j^2}$$

3.2.6 Evaluacija i izbor modela

Bitan segment mašinskog učenja čine evaluacija i izbor modela. Za kvantifikovanje sposobnosti predviđanja modela koriste se mere kvaliteta. Mere kvaliteta koje se najčešće koriste za regresiju su:

- srednjekvadratna greška i njen koren (eng. mean square error - MSE, root mean square error - RMSE)

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2$$

MSE je greška koja se direktno minimizuje u mnogim regresionim problemima. RMSE je greška izražena na istoj skali kao i ciljna promenljiva, što je korisno za interpretaciju dobijenih vrednosti.

- srednja relativna greška izražena u procentima (eng. mean absolute percentage error - MAPE)

$$MAPE = \frac{100}{N} \sum_{i=1}^N \left| \frac{(y_i - f(x_i))}{y_i} \right|$$

- koeficijent determinacije R^2

$$R^2 = 1 - \frac{MSE}{var(y)}$$

R^2 predstavlja udeo objašnjene varijanse ciljne promenljive za neki model. Drugim rečima, R^2 poredi srednjekvadratnu grešku u odnosu na model (MSE) sa srednjekvadratnom greškom u odnosu na prosek ciljne promenljive ($var(y)$)

Reprezentacije modela mašinskog učenja mogu biti nekonfigurabilne (bez metaparametara) i konfigurabilne. U slučaju konfigurabilnih reprezentacija možemo da pričamo o izboru modela, u smislu izbora njegovih metaparametara. U oba slučaja bitna je evaluacija modela. Ključna stvar kod evaluacije

modela je da se prilikom evaluacije ne koriste podaci sa kojima se model obučava. Jedna mogućnost je da skup podataka podelimo na podatke za *treniranje* ili *obučavanje* i *testiranje*, istreniramo aproksimaciju modela na podacima za trening, a na podacima za testiranje odredimo kvalitet aproksimacije koji koristimo kao ocenu modela. Na kraju za praktičnu primenu model istreniramo na svim podacima. Želimo da model obučavamo na svim podacima jer je u suprotnom smanjena pouzdanost modela.

Ipak, ovakav pristup ima nedostatke. Ocena modela varira u zavisnosti od izbora skupa za testiranje, koji pritom može biti izabran pristrasno (npr. ukoliko sadrži samo instance sa najvišom vrednošću ciljne promenljive, ili u slučaju klasifikacije ako sve instance za testiranje pripadaju istoj klasi). Drugi pristup evaluaciji modela zove se *unakrsna K-slojna validacija* (eng. K-fold cross validation). Podatke delimo na K (približno) jednakih, disjunktih podskupova ili *slojeva*. Za svaki od slojeva S_k ($1 \leq k \leq K$) obučavamo model na podacima izvan S_k , a na S_k izvršimo predviđanja. Nakon toga imamo predviđanja na celom skupu podataka⁴, i na osnovu njih ocenjujemo model (koji dobijamo obučavanjem na celom skupu podataka). Ovaj pristup, iako vremenski zahtevniji, vodi pouzdanijoj oceni modela nego podela na skupove za trening i testiranje. Za broj slojeva K se obično uzimaju vrednosti 5 ili 10.

Za konfigurabilne reprezentacije modela različite konfiguracije vode različitim modelima, i postavlja se pitanje kako izabrati najbolju konfiguraciju, odnosno najbolje metaparametre. Postupak izbora modela i evaluacije je sledeći:

- skup podataka podelimo na disjunktne skupove za treniranje ($X_{training}$) i testiranje (X_{test})
- za svaku konfiguraciju c na podacima iz $X_{training}$ odredi se evaluacija modela sa konfiguracijom c (podelom skupa $X_{training}$ na podskupove za treniranje i validaciju) i zapamti se ocena
- neka je c^* konfiguracija sa najboljom ocenom iz prethodnog koraka; model sa konfiguracijom c^* obučimo na podacima iz skupa $X_{training}$
- na podacima iz skupa X_{test} vršimo predviđanja korišćenjem modela iz prethodnog koraka i ocenjujemo kvalitet predviđanja

Ili korišćenjem *ugneždene unakrsne K-slojne validacije*:

⁴Predviđanja dobijena aproksimacijama modela koji ocenjujemo

- skup podataka D delimo na K (približno) jednakih, disjunktih podskupova (slojeva): $D = S_1 \cup S_2 \cup \dots \cup S_K$
- za svaki sloj S_k ($1 \leq k \leq K$):
 - izaberemo najbolju konfiguraciju pomoću unakrsne validacije na podacima iz $D \setminus S_k$
 - pomoću konfiguracije iz prethodnog koraka obučimo model na podacima iz $D \setminus S_k$ i vršimo predviđanje na podacima iz S_k
- računamo ocenu kvaliteta na osnovu svih predviđanja na celom skupu podataka

3.3 Metode mašinskog učenja za predviđanje vremena izvršavanja SAT rešavača

Predviđanje vremena izvršavanja raznih algoritama je problem kome je posvećeno dosta pažnje protekle decenije. Evaluirani su mnogi modeli koji na osnovu atributa instance problema predviđaju vreme izvršavanja rešavačkih algoritama na datoj instanci. Ovakvi modeli primenjuju se da bi se analizirali rešavački algoritmi, zatim za selekciju najboljeg algoritma za pojedinačne instance ili automatsku parametrizaciju konfigurabilnih algoritama.

Metode mašinskog učenja za koje je u referentnim radovima analizirana primena na predviđanje vremena izvršavanja SAT rešavača su grebena regresija [17, 19, 22, 23, 24, 25, 26, 27, 28], neuronske mreže [20], Gausovi procesi [19], regresiona stabla odlučivanja [21] i slučajne šume [1]. Pretpostavka ovih metoda je da su vremena izvršavanja nezavisna, odnosno da je model $f(x)$ funkcija vektora atributa x , i da y_i zavisi samo od x , ali ne i od drugih y_j , $j \neq i$. U ovom radu isproban je nov pristup koji uzima u obzir moguću zavisnost između y_i i y_j , $i \neq j$: metoda Gausovih uslovnih slučajnih polja. Opravdanje leži u činjenici da bi dva slična rešavača, koji na većini instanci imaju slične performanse (odnosno približna vremena izvršavanja), na novoj instanci takođe trebalo da imaju slično trajanje izvršavanja. U cilju poređenja, u ovom radu je evaluirano nekoliko linearnih modela primenom metode grebene regresije i nekoliko nelinearnih modela primenom metoda slučajne šume i Gausovih uslovnih slučajnih polja. Slučajna šuma je metoda koja se najbolje pokazala u dosadašnjim radovima koji se bave problematikom predviđanja trajanja izvršavanja raznih nedeterminističkih algoritama [1]. Detalji o ostalim metodama korišćenim u referentnim radovima za predviđanje vremena izvršavanja SAT rešavača (neuronske mreže,

Gausovi procesi) nisu obuhvaćeni ovim radom, ali se mogu pronaći u literaturi [2].

3.3.1 Linearna i grebena regresija

Linearna⁵ regresija je najjednostavnija parametarska reprezentacija modela. Kada imamo podatke koje želimo da iskoristimo za nekakva predviđanja uglavnom najpre krenemo od evaluacije linearnih modela koji se često ispostave dovoljno dobrim za veliki broj praktičnih primena, što linearnu regresiju čini jednom od najšesće korišćenih metoda mašinskog učenja.

Iz probabilističke perspektive, osnovne pretpostavke linearne regresije su normalna raspodela ciljne promenljive u zavisnosti od vrednosti atributa i linearna forma modela:

$$p_{\omega}(y|x) = \mathcal{N}(f_{\omega}(x), \sigma^2)$$

$$f_{\omega}(x) = \omega x$$

Ocena parametara ω metodom maksimalne verodostojnosti svodi se na minimizaciju srednjekvadratne greške (N je veličina uzorka, M broj atributa):

$$\min_{\omega} \frac{1}{N} \sum_{i=1}^N (y_i - \omega x_i)^2$$

$$\omega = [\omega_1 \quad \omega_2 \quad \dots \quad \omega_M], x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iM} \end{bmatrix}$$

Minimizacioni problem možemo da zapišemo i matrično:

$$\min_{\omega} \|y - X\omega^T\|^2$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & x_{2M} \\ \vdots & \vdots & \dots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{NM} \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

⁵Model $f_{\omega}(x)$ je funkcija linearna po parametrima ω , a ne nužno po promenljivoj x , stoga linearni model može da bude i polinom.

Rešenje može da se dobije primenom numeričkih gradijentnih metoda za optimizaciju, ili metodom najmanjih kvadrata⁶ koja daje algebarsku formu rešenja:

$$\omega^* = (X^T X)^{-1} X^T y \quad (3.1)$$

Problem grebene regresije dobijamo l_2 regularizacijom problema linearne regresije:

$$\begin{aligned} \min_{\omega} \|y - X\omega^T\| + \lambda \|\omega\|^2 \\ \omega^* = (X^T X + 2\lambda I)^{-1} X^T y \end{aligned}$$

Smisao regularizacije u mašinskom učenju objašnjen je u poglavlju 3.2.4. U jednačini (3.1) primećujemo da se može desiti da matrica $X^T X$ bude loše uslovljena, ili čak neinvertibilna, što znači da u algebarskoj interpretaciji povećanje parametra regularizacije λ vodi ka invertibilnosti i boljoj uslovljenosti invertovane matrice.

U kontekstu predviđanja vremena izvršavanja SAT rešavača, grebena regresija je najčešće korišćena metoda [15, 16, 17, 18, 19]. Predviđa se logaritam vremena izvršavanja SAT rešavača u zavisnosti od atributa instance problema, koji može biti bolje aproksimiran polinomijalnom nego linearnom funkcijom, stoga se prostor atributa proširuje interakcijama (označimo sa M broj atributa; ulazni vektor atributa $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,M}]$ proširuje se vrednostima $x_{i,j} * x_{i,k}$ za $j = 1, \dots, M$ i $k = j, \dots, M$).

Čak i pored regularizacije, moć generalizacije modela grebene regresije opada ako su neki atributi neinformativni ili su visoko korelirani sa drugim atributima. Zato se prostor atributa smanjuje primenom selekcije atributa. U [27] i [28] predložen je metod pohlepne selekcije unapred (eng. forward feature selection) u dve faze (prva je pre, a druga nakon računanja interakcija; detaljniji opis može se pogledati u poglavlju 4.4). Razlog za prvu fazu selekcije je skalabilnost algoritma za obučavanje, jer je matrica podataka za ovučavanje X veličine $N \times M$ (N je br. instanci, M br. atributa), dok je veličina matrice dobijene nakon računanja interakcija $N \times \frac{M(M+3)}{2}$, što znači da matrica podataka može kritično da naraste za veliko N .

U ovom radu isprobane su još dve metode za smanjenje dimenzionalnosti prostora atributa prilikom modelovanja problema od interesa grebenom regresijom: rekurzivna eliminacija atributa (eng. RFE - Recursive Feature Elimination) i analiza glavnih komponenti. Rekurzivna eliminacija atributa

⁶Metoda najmanjih kvadrata rede se koristi jer zahteva računanje inverza matrice $X^T X$, koja je u problemima mašinskog učenja uglavnom velikih dimenzija.

je, kao i selekcija unapred, jedna od metoda omotača (pogledati poglavlje 3.2.5). Slično kao u slučaju selekcije unapred, atributi su rekurzivno eliminisani u 2 faze (pre i nakon računanja interakcija) i evaluirano je nekoliko modela. Detaljniji opis selekcije atributa rekurzivnom eliminacijom, kao i opis konfiguracija i rezultati eksperimentalne evaluacije mogu se naći u poglavlju 4.4. Ispostavilo se da je ovaj metod selekcije dao nešto bolje rezultate u odnosu na pohlepnu selekciju unapred.

Analiza glavnih komponenti predstavlja projektovanje podataka na ortonormiranu bazu potprostora atributa u kome je sadržana glavna varijabilnost podataka [29] (poglavljje 3.2.5). Smanjenje dimenzionalnosti podataka korišćenjem analize glavnih komponenti proizvelo je najslabije ocenjene modele. Detalji eksperimentalne evaluacije nalaze se u poglavlju 4.4.

3.3.2 Regresiona stabla odlučivanja

Stablo odlučivanja u kontekstu mašinskog učenja predstavlja nadgledanu neparаметarsku metodu koja se koristi za klasifikaciju i regresiju. Predviđanje se vrši na osnovu jednostavnih pravila odlučivanja izvedenih iz atributa. Proces obučavanja se svodi na konstrukciju optimalnog stabla. Nalaženje globalno optimalnog stabla odlučivanja je NP kompletan problem, pa se u praksi koriste pohlepni algoritmi koji u svakom čvoru prave lokalno optimalnu poddelu. Oni ne garantuju globalnu optimalnost dobijenog stabla, ali ovaj problem može biti ublažen korišćenjem metode slučajne šume, kojoj je posvećena pažnja u poglavlju 3.3.3.

Najpopularniji praktičan algoritam za konstrukciju optimalnog stabla odlučivanja koje može da se koristi i za regresione i za klasifikacione probleme je CART (eng. Classification And Regression Trees) [30]. Struktura CART modela je binarno stablo čiji listovi sadrže instance za koje je varijansa ciljne promenljive (lokalno) optimizovana, pa se regresiono predviđanje svodi na pronalazak odgovarajućeg lista i računanje proseka vrednosti ciljne promenljive nad instancama tog lista.

Skica CART algoritma za regresiju

Neka su x_i vektori atributa a y_i vrednosti ciljne promenljive ($1 \leq i \leq N$). Rekurzivno delimo skup instanci (x_i, y_i) na osnovu vrednosti nekog atributa tako da su instance sa istom vrednošću ciljne promenljive grupisane zajedno.

Korak podele

Koren stabla sadrži skup svih instanci, označimo taj skup sa D . Tražimo parametre $\theta = (j, t)$ (gde je j indeks jednog od atributa, a t prag za vrednost tog atributa) optimalne podele skupa D na 2 nova disjunktna podskupa: D_{levo} i D_{desno} .

$$D_{levo}(\theta) = \{(x, y) | x[j] \leq t\}$$

$$D_{desno}(\theta) = D \setminus D_{levo}(\theta) = \{(x, y) | x[j] > t\}$$

Optimalna vrednost parametara θ^* dobija se kao rešenje sledećeg minimizacionog problema:

$$\min_{\theta} \left[\frac{|D_{levo}(\theta)|}{|D|} H(D_{levo}(\theta)) + \frac{|D_{desno}(\theta)|}{|D|} H(D_{desno}(\theta)) \right]$$

U slučaju regresije, funkcija H uglavnom se definiše kao varijansa ciljne promenljive na instancama odgovarajućeg podskupa podele:

$$H(D_{levo}(\theta)) = \frac{1}{|D_{levo}(\theta)|} \sum_{(x_i, y_i) \in D_{levo}(\theta)} (y_i - \bar{y})^2,$$

$$\bar{y} = \frac{1}{|D_{levo}(\theta)|} \sum_{(x_i, y_i) \in D_{levo}(\theta)} y_i$$

Analogno za $H(D_{desno}(\theta))$. Na kraju, optimalna podela zadata je sa $D_{levo} = D_{levo}(\theta^*)$ i $D_{desno} = D_{desno}(\theta^*)$.

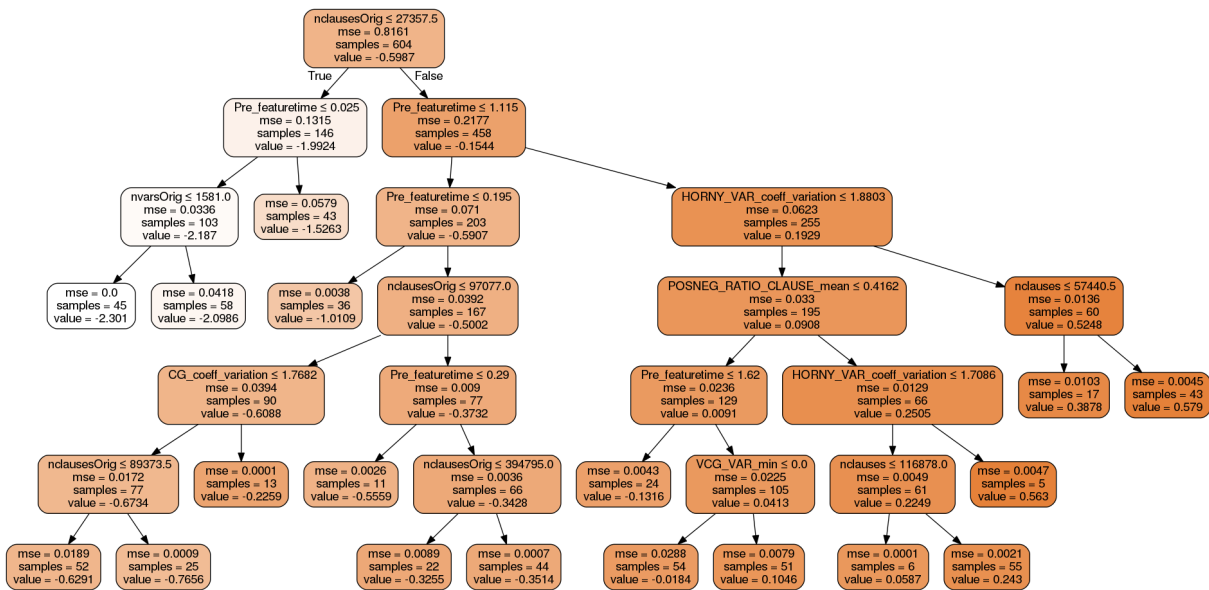
Rekurzija i zaustavljanje

Rekurzivno ponavljamo korak podele na skupovima D_{levo} i D_{desno} dok ne bude zadovoljen neki od kriterijuma zaustavljanja:

1. broj instanci u čvoru (odnosno broj elemenata skupa D) postane 1
2. varijansa ciljne promenljive postane manja od neke unapred zadate vrednosti
3. dostigne se maksimalna unapred zadata dubina stabla
4. broj instanci u čvoru postane manji od unapred zadatog minimuma

Očigledno je da stablo dobijeno opisanim algoritmom može da bude prekompleksno i prilagođeno podacima za obučavanje. Kriterijumi 2, 3 i 4 predstavljaju vid regularizacije, jer doprinose smanjenju kompleksnosti stabla i vode boljoj generalizaciji modela. Još jedna tehnika regularizacije je odsecanje pojedinih podstabala (eng. pruning).

Prednosti stabla odlučivanja, osim što je primenljivo i na regresione i na klasifikacione probleme, su visoka interpretabilnost (stablo vrlo lako možemo vizualizovati da bismo analizirali proces predviđanja) i to što nije potrebna priprema podataka za obučavanje u vidu standardizacije i normalizacije. Primer vizuelizacije na podacima za predviđanje vremena izvršavanja SAT rešavača dat je na slici 1. Uzeti su podaci iz skupa SWV (detalji o skupovima podataka koji su korišćeni u ekperimentima mogu se naći u poglavlju 4.2), a u cilju ilustrativnosti stablo je pojednostavljeno podešavanjem minimalnog broja instanci na kojima se može izvršiti dalja podela na 60.



Slika 1: Primer stabla odlučivanja

3.3.3 Slučajne šume

Slučajna šuma zapravo predstavlja ansambl stabala odlučivanja. Metode ansambla kombinuju rezultate predviđanja nekoliko baznih modela sa ciljem da se unapredi moć generalizacije rezultujućeg modela. Razlikujemo dva tipa

ovih metoda:

- metode uprosečavanja (eng. averaging methods)
 - nekoliko modela se nezavisno obučiti, pa se kao rezultujuće predviđanje uzima prosek predviđanja tih modela; ove metode predstavljaju vid regularizacije jer im je cilj da smanje varijansu⁷ rezultujućeg modela, pa su pogodne za fleksibilne bazne modele sklone preprilagođavanju
- metode dopunjavanja (eng. boosting methods)
 - bazni modeli se obučavaju sekvencijalno, tako da svaki model popravlja nedostatke svog prethodnika u cilju smanjenja sistematskog odstupanja rezultujućeg modela

Slučajna šuma je primer metode uprosečavanja. Svako stablo u ansamblu (šumi) se konstruiše nad uzorkom sa ponavljanjem⁸ (eng. bootstrap sample). Dodatno, prilikom koraka podele bira se optimalna podela nad slučajnim podskupom skupa atributa umesto nad skupom svih atributa.

Već je spomenuto da algoritam za konstrukciju stabla odlučivanja zarad računске efikasnosti žrtvuje globalnu optimizaciju, što je problem koji je ublažen korišćenjem slučajnih šuma. Sada je takođe zaključeno da se ansambl ponaša kao regularizaciona tehnika za stabla odlučivanja, što između ostalog znači da ublažava nestabilnost rešenja (mala promena vrednosti podataka može rezultovati velikom razlikom u dobijenim stablima, što je zapravo posledica visoke varijanse).

Modeli slučajne šume daju najbolji kvalitet predviđanja trajanja izvršavanja SAT rešavača na instanci SAT problema [1]. Opis isprobanih konfiguracija i rezultati eksperimentalne evaluacije mogu se videti u poglavlju 4.5.

3.3.4 Gausova uslovna slučajna polja

Uobičajena pretpostavka metoda mašinskog učenja je da su opažanja uslovno nezavisna i pripadaju istoj raspodeli. To znači da je model $y_i = f(x_i)$ funkcija vektora atributa x , odnosno da za konkretnu instancu x_i odgovarajuće opažanje y_i zavisi samo od tog x_i , a ne i od y_j , $j \neq i$. Drugim rečima, uslovnu

⁷Za preprilagođen model kažemo da ima *visoku varijansu*, dok model koji je nedovoljno fleksibilan da se prilagodi podacima ima *visoko sistematsko odstupanje*.

⁸Na osnovu raspoloživih instanci nekog uzorka kreira se novi uzorak, istog obima kao izvorni uzorak, slučajnim biranjem i vraćanjem iz skupa raspoloživih instanci; ovo znači da svaka instanca ima jednaku verovatnoću da uđe u uzorak i da neka instanca može da se pojavi više puta, a neka nijednom.

verovatnoću $P(y_1, \dots, y_n | x_1, \dots, x_n)$ možemo faktorisati na pojedinačne verovatnoće i predstaviti kao proizvod $\prod_{i=1}^n P(y_i | x_i)$. Postavlja se pitanje koliko je ova pretpostavka realistična. Posmatrajmo dve osobe koje se intenzivno družu, označimo ih sa A i B. Recimo da znamo da A i B imaju sličan muzički ukus i predviđamo da li će se osobi A svideti neka pesma. Dok dato opažanje svakako zavisi od atributa same pesme, verovatnoća da će se osobi A svideti pesma je utoliko veća ukoliko se pesma sviđa osobi B. Ovo poglavlje opisuje jednu metodu mašinskog učenja koja uzima u obzir zavisnosti između y_i i y_j , $j \neq i$: metodu Gausovih uslovnih slučajnih polja za regresiju (prema [37]), kao i njenu primenu na problem predviđanja vremena izvršavanja SAT rešavača.

Neprekidna uslovna slučajna polja Metoda neprekidnih uslovnih slučajnih polja modeluje uslovnu raspodelu opažane promenljive y za dati vektor atributa \mathbf{x} . To znači da su modeli neprekidnih uslovnih slučajnih polja diskriminativni modeli. Međutim, za razliku od uobičajenih metoda mašinskog učenja, ova metoda ne pretpostavlja faktorizaciju na pojedinačne verovatnoće, već modeluje uslovnu raspodelu u skladu sa nekim zadatim grafom zavisnosti. Na primer, ukoliko govorimo o nekakvom predviđanju u društvenoj mreži, korisnike možemo posmatrati kao čvorove grafa, grane su povezanosti između korisnika a kompatibilnost između korisnika može se predstaviti težinama grana u grafu. Forma uslovne raspodele je sledeća:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x}, \alpha, \beta)} \exp \left(\sum_{i=1}^N A(\alpha, y_i, \mathbf{x}) + \sum_{i \sim j} I(\beta, y_i, y_j, \mathbf{x}) \right) \quad (3.2)$$

($i \sim j$ označava povezanost y_i i y_j u grafu zavisnosti)

Z je normalizujuća funkcija koja služi da bi navedena forma predstavljala gustinu raspodele i označava integral navedene eksponencijalne funkcije. Funkcija A predstavlja potencijal pridruživanja (eng. association potential) između \mathbf{x} i y_i . To je funkcija koja ima utoliko veću vrednost ako y_i bolje odgovara vektoru atributa \mathbf{x} . Funkcija I je potencijal interakcije (eng. interaction potential), odnosno mera kompatibilnosti za vrednosti y_i i y_j , posmatrano samo po i i j koji su povezani u grafu zavisnosti. Kompatibilnost između y_i i y_j zavisi i od \mathbf{x} , jer su možda y_i i y_j u jednom delu prostora atributa kompatibilni, a u drugom ne. Na primer, ukoliko smo na nekoj društvenoj mreži povezani sa nekim ko ima odličan muzički ukus, verovatnoća da ćemo kopirati od njega neki muzički sadržaj je veća, što možda ne važi za sadržaj iz nekih drugih oblasti. U ovom primeru, vektor \mathbf{x} nam govori da li se radi o muzici ili nečem drugom. α i β predstavljaju parametre modela.

Korak predviđanja je jednostavan: ako imamo uslovnu raspodelu za dato \mathbf{x} zanima nas \mathbf{y} koje je najviše verovatno; dakle, maksimizacija po \mathbf{y} :

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} P(\mathbf{y}|\mathbf{x})$$

Učenje se svodi na maksimizaciju logaritma uslovne verovatnoće (eng. log-likelihood). Tražimo one parametre α i β za koje je verovatnoća na skupu za obučavanje maksimalna:

$$\begin{aligned} L(\alpha, \beta) &= \log P(\mathbf{y}|\mathbf{x}) \\ (\hat{\alpha}, \hat{\beta}) &= \arg \max_{\alpha, \beta} L(\alpha, \beta) \end{aligned}$$

Gausova uslovna slučajna polja Neka su u jednačini (3.2) funkcije A i I predstavljene kao linearne kombinacije nekih atributskih funkcija (eng. feature functions) f_k i g_l :

$$\begin{aligned} A(\alpha, y_i, \mathbf{x}) &= \sum_{k=1}^K \alpha_k f_k(y_i, \mathbf{x}) \\ I(\beta, y_i, y_j, \mathbf{x}) &= \sum_{l=1}^L \beta_l g_l(y_i, y_j, \mathbf{x}) \end{aligned}$$

Ostaje nam pitanje kako da odaberemo funkcije f_k i g_l tako da dobijemo efikasan trening i predviđanje. Osnovno zapažanje je da ukoliko asocijativni i interakcioni potencijal definišemo kao linearne kombinacije funkcija kvadratnih u odnosu na y_i , $P(\mathbf{y}|\mathbf{x})$ postaje višedimenziona Gausova raspodela. Gausova raspodela je poželjna jer se lako ocenjuje. Predviđanje je takođe lako jer se svodi na računanje proseka te raspodele. Dalje razmatramo kako izabrati atributske funkcije na koristan način. Pretpostavimo da za svaki čvor grafa zavisnosti imamo jedan regresioni model (označavamo ga sa R_i) i da imamo jedan graf zavisnosti. U opštem slučaju, za svaki čvor možemo imati nekoliko regresionih modela R_{ik} , $k = 1, \dots, K$ (koji se nazivaju i *nestrukturirani modeli*, dok je GCRF model *strukturiran*), oni vrše predviđanja uzimajući u obzir možda različite aspekte problema po različitim principima, jedan regresioni model može da bude neuronska mreža, drugi slučajna suma ili nešto treće itd. Takođe možemo imati više grafova zavisnosti G_l , $l = 1, \dots, L$. Atributske funkcije definišemo na sledeći način:

$$\begin{aligned} f_k(y_i, \mathbf{x}) &= -(y_i - R_{ik}(\mathbf{x}))^2, & k &= 1, \dots, K \\ g_l(y_i, y_j, \mathbf{x}) &= -e_{ij}^l S_{ij}^l(\mathbf{x})(y_i - y_j)^2, & l &= 1, \dots, L \end{aligned}$$

Pritom važi

$$e_{ij}^l = \begin{cases} 1, & \text{ako } (i, j) \in G_l \\ 0, & \text{inače} \end{cases}$$

Puna forma raspodele $P(\mathbf{y}|\mathbf{x})$ je sledeća:

$$\frac{1}{Z(\mathbf{x}, \alpha, \beta)} \exp \left(- \sum_{i=1}^N \sum_{k=1}^K \alpha_k (y_i - R_{ik}(\mathbf{x}))^2 - \sum_{i \sim j} \sum_{l=1}^L \beta_l e_{ij}^l S_{ij}(\mathbf{x}) (y_i - y_j)^2 \right)$$

Pojednostavljena forma koja sadrži sve što je suštinski važno, a lakša je za razumevanje, je sledeća:

$$\frac{1}{Z(\mathbf{x}, \alpha, \beta)} \exp \left(- \sum_{i=1}^N \alpha (y_i - R_i(\mathbf{x}))^2 - \sum_{i \sim j} \beta S_{ij}(\mathbf{x}) (y_i - y_j)^2 \right)$$

S_{ij} predstavlja meru sličnosti čvorova i i j u grafu, i može da zavisi od \mathbf{x} . Recimo da imamo vrednosti predviđanja regresionih modela za svaki od čvorova. Ako y_i odstupa od predviđanja regresionog modela R_i , kvadrirana razlika postaje velika, a negativna je i to nam smanjuje verovatnoću za to y_i . U nekoj meri verujemo regresionom modelu R_i i ne želimo drastično da promenimo njegovo predviđanje. Parametar α je parametar poverenja u taj regresioni model. Osvrnimo se sada na vrednost $S_{ij}(\mathbf{x})$ (sličnost između čvorova i i j). Ukoliko je pomenuta sličnost jednaka nuli, odnosno nema nikakve kompatibilnosti između čvorova i i j , nema nikakvog razloga da približavamo vrednosti koje regresioni modeli na tim čvorovima daju. Međutim, ako je sličnost velika, a dođe do velikog razdvajanja predviđanja y_i i y_j , ovo utiče osetno negativno i smanjuje verovatnoću, pa će model prihvatiti da odstupa od predviđanja regresionog modela R_i kako bi približio vrednosti za slične čvorove. Dakle, ako imamo nekakva predviđanja nad čvorovima grafa zavisnosti, i ta predviđanja možda značajno odskakuju jedna od drugih, ono što GCRF zapravo radi je približavanje predviđanja ako su odgovarajući čvorovi slični, u onoj meri u kojoj mu sličnost to nalaže. Parametri α i β služe za nagodbu između toga koliko verujemo nestrukturiranom regresionom modelu, a koliko meri sličnosti između čvorova.

Razmotrimo uslove koji moraju biti zadovoljeni da bi navedena forma bila forma Gausove raspodele:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{y} - \mu)^T \Sigma^{-1} (\mathbf{y} - \mu) \right)$$

Potrebno je upariti izraze u eksponentu:

$$-\frac{1}{2} (\mathbf{y} - \mu)^T \Sigma^{-1} (\mathbf{y} - \mu) = -\frac{1}{2} \mathbf{y}^T \Sigma^{-1} \mathbf{y} + \frac{1}{2} \mathbf{y}^T \Sigma^{-1} \mu + const$$

i

$$- \sum_{i=1}^N \sum_{k=1}^K \alpha_k (y_i - R_{ik}(\mathbf{x}))^2 - \sum_{i \sim j} \sum_{l=1}^L \beta_l e_{ij}^l S_{ij}(\mathbf{x}) (y_i - y_j)^2$$

Nakon uparivanja dobija se sledeće:

$$\Sigma_{ij}^{-1} = \begin{cases} 2 \sum_{k=1}^K \alpha_k + 2 \sum_{k=1}^K \sum_{l=1}^L \beta_l e_{ik}^l S_{ik}(\mathbf{x}), & i = j \\ -2 \sum_{l=1}^L \beta_l e_{ij}^l S_{ij}(\mathbf{x}), & i \neq j \end{cases}$$

$$\mu = \Sigma \mathbf{b}$$

$$b_i = 2 \left(\sum_{k=1}^K \alpha_k R_{ik}(\mathbf{x}) \right)$$

Dalje, da bi data raspodela bila Gausova, matrica preciznosti Σ^{-1} mora biti pozitivno semi-definitna, što znači da je skalar $z^T \Sigma^{-1} z$ nenegativan za svaki vektor z odgovarajuće dimenzije. Dovoljan uslov je da matrica bude simetrična i dijagonalno dominantna (apsolutna vrednost na dijagonali je veća ili jednaka sumi apsolutnih vrednosti koje se javljaju van dijagonale u istom redu i istoj koloni); to je zadovoljeno ako su α i β veći od 0.

Primena na predviđanje vremena izvršavanja SAT rešavača

Uobičajene metode mašinskog učenja modeluju problem predviđanja vremena izvršavanja SAT rešavača kao funkciju koja zavisi samo od atributa ulazne formule. Kako se predviđaju vremena izvršavanja skupa rešavača, moguće je da postoji zavisnost među njima, jer ako su dva rešavača slična u smislu da imaju slične performanse na većini instanci, verujemo da će ta sličnost da utiče na uslovnu raspodelu $P(\mathbf{y}|\mathbf{x})$.

U ovom radu kao nestruktu modeli R_i upotrebljeni su modeli slučajne šume. Graf zavisnosti G ima $N \times M$ čvorova, gde je N broj SAT rešavača za koje imamo vremena izvršavanja na M instanci skupa za obučavanje. Pritom se G sastoji od M komponenti koje nisu među sobom povezane, jer modelujemo zavisnosti između vremena izvršavanja različitih rešavača, pa čvorovi koji predstavljaju vremena izvršavanja istog rešavača na različitim instancama nisu povezani. Sličnost dva rešavača zavisi od euklidskog rastojanja između vektora kojim su ti rešavači predstavljeni i skalirana je na $[0, 1]$ (naj-sličniji rešavači su na najmanjem rastojanju i za njih je mera sličnosti 1). Rezultati eksperimentalne evaluacije nalaze se u poglavlju 4.6.

4 Eksperimentalna evaluacija

Problem na koji se fokusira ovaj rad je predviđanje vremena izvršavanja skupa SAT rešavača na zadatim instancama iskaznih formula koje su predstavljene vektorima svojih atributa. U tom kontekstu, evaluirano je nekoliko modela raznih metoda mašinskog učenja. Ovo poglavlje bavi se detaljima implementacije eksperimentalne evaluacije, opisivanjem skupova podataka na kojima su rađeni eksperimenti, kao i opisivanjem samih eksperimenata, koji podrazumevaju evaluaciju nekoliko modela iz referentnih radova korišćenjem metoda grebene regresije i slučajnih šuma i nekoliko modela predloženih u ovom radu. Predložena je mala modifikacija linearnog modela dobijenog grebenom regresijom, i predložene su dve varijante modela Gausovih uslovnih slučajnih polja.

4.1 Detalji implementacije

Eksperimentalni deo obavljen je u programskom jeziku Python (verzija 3.5.2), uz korišćenje sledećih biblioteka:

- `scikit-learn` za implementacije metode grebene regresije, metode slučajne šume, analize glavnih komponenti i mera kvaliteta predviđanja (srednjekvadratne greške i R^2 skora)
- `numpy` za strukture podataka i numerička izračunavanja (optimizaciju i singularnu dekompoziciju)
- `scipy` za neke numeričke metode optimizacije
- `pandas` za čitanje podataka⁹
- `matplotlib` za iscrtavanje grafika

Za implementaciju regresionih Gausovih uslovnih slučajnih polja zahvaljujem se Andriji Petroviću. Programski kod za pokretanje eksperimenata i podaci mogu se pronaći na https://github.com/shisharka/algorithm_runtime_prediction. Struktura koda opisana je u nastavku.

- Datoteka `requirements.txt` sadrži spisak Python biblioteka neophodnih za pokretanje eksperimenata. Biblioteke se mogu instalirati komandom `pip install -r requirements.txt`.

⁹Atributi instanci i vremena izvršavanja rešavača čuvaju se u `.csv` datotekama.

- Direktorijum *data/* sadrži sledeće vrste datoteka:
 - datoteke čiji su nazivi u formatu *<oznaka skupa instanci>- <naziv rešavača>.csv* u kojima se nalaze vremena izvršavanja rešavavača
 - datoteke čiji su nazivi u formatu *<oznaka skupa instanci>-feat.csv* u kojima se nalaze vrednosti atributa instanci

U ovom direktorijumu nalaze se podaci za eksperimentalnu evaluaciju modela grebene regresije i slučajnih šuma. Detaljniji opis ovih podataka nalazi se u poglavlju 4.2.1.

- Direktorijum *SATzilla2012_data/* sadrži sledeće datoteke:
 - *SATALL12S.csv*
 - *SATHAND12S.csv*
 - *SATINDU12S.csv*
 - *SATRAN12S.csv*

U ovim datotekama čuvaju se vremena izvršavanja različitih rešavača i vrednosti atributa instanci. Ovi podaci koriste se za eksperimentalnu evaluaciju strukturiranih modela Gausovih uslovnih slučajnih polja i nestrukturiranih modela slučajne šume. O ovim podacima detaljnije se govori u poglavlju 4.2.2.

- Datoteka *read_data.py* sadrži funkciju za učitavanje podataka za eksperimentalnu evaluaciju.
- U datoteci *data_preprocessing.py* definisane su pomoćne konstante i funkcije za transformacije podataka pre obučavanja:
 - `BROKEN_FEATURE_VALUES` je predefinisani niz vrednosti kojima su u podacima obeležene nedostajuće vrednosti atributa
 - `LOG10_TRESHOLD` predstavlja prag za logaritamsku transformaciju ciljnih promenljivih
 - `STD_TRESHOLD` predstavlja donji prag za standardnu devijaciju kolona matrice podataka (vrste ove matrice predstavljaju instance, dok kolone predstavljaju attribute); kolone, odnosno atributi, sa standardnom devijacijom manjom od `STD_TRESHOLD` se odbacuju jer je varijabilnost ovih atributa beznačajna (drugim rečima, vrednost ovih atributa je bliska konstanti)

- funkcija `mean(X)` računa vektor srednjih vrednosti kolona matrice podataka X , ignorišući vrednosti iz niza `BROKEN_FEATURE_VALUES`
- funkcija `std(X)` računa vektor standardnih devijacija kolona matrice podataka X , ignorišući vrednosti iz niza `BROKEN_FEATURE_VALUES`
- funkcija `remove_const_cols(X)` transformiše matricu podataka X uklanjanjem konstantnih atributa, odnosno uklanjanjem kolona čija je standardna devijacija manja od praga zadatog sa `STD_TRESHOLD`
- funkcija `center(X, mean_vec)` transformiše matricu podataka X tako što od vrednosti x_{ij} (za sve moguće vrednosti i) oduzme odgovarajuću j -tu koordinatu vektora `mean_vec`, ignorišući pritom vrednosti iz niza `BROKEN_FEATURE_VALUES`; ukoliko je `mean_vec = mean(X)`, ova transformacija predstavlja centriranje matrice X tako da je srednja vrednost svake kolone 0
- funkcija `standardize(X, mean_vec, std_vec)` transformiše matricu podataka X tako što od vrednosti x_{ij} (za sve moguće vrednosti i) oduzme odgovarajuću j -tu koordinatu vektora `mean_vec` i dobijeni rezultat podeli odgovarajućom j -tom koordinatom vektora `std_vec`, ignorišući pritom vrednosti iz niza `BROKEN_FEATURE_VALUES`; ukoliko je `mean_vec = mean(X)` i `std_vec = std(X)`, ova transformacija predstavlja standardizovanje matrice X tako da je srednja vrednost svake kolone 0, a standardna devijacija svake kolone 1
- funkcija `calculate_interactions(X)` računa interakcije nad matricom podataka X , takođe ignorišući nedostajuće vrednosti iz niza `BROKEN_FEATURE_VALUES`
- funkcija `log10_transform(Y)` transformiše vektor vrednosti ciljne promenljive Y tako što vrednosti koje su manje od `LOG10_TRESHOLD` najpre postavi na `LOG10_TRESHOLD`, a zatim nad svim koordinatama primeni logaritam sa osnovom 10
- funkcija `handle_broken_features(X, mean_vec)` transformiše matricu podataka X tako što vrednosti iz niza `BROKEN_FEATURE_VALUES` menja odgovarajućom koordinatom vektora `mean_vec` (ukolik oje `mean_vec = mean(X)`, ovom transformacijom se zapravo nedostajuće vrednosti u matrici podataka zamene srednjim vrednostima odgovarajućih kolona)

O preprocesiranju podataka za obučavanje detaljnije se govori u poglavljima 4.3 i 4.4.

- U datoteci *regularization_metaparam.py* definisana je funkcija `determine_regularization_metaparam(X_train, Y_train)` koja 10-strukom unakrsnom validacijom pronalazi najbolji regularizacioni parametar za model grebene regresije u predefinisanoj nizu vrednosti $[10^{-6}, 10^{-5}, \dots, 10^2]$. Argumenti *X_train* i *Y_train* predstavljaju matricu podataka i vektor vrednosti ciljne promenljive (respektivno) sa instancama za obučavanje, a najbolji regularizacioni parametar je onaj za koji je minimalna srednjekvadratna greška predviđanja.
- Datoteka *run_experiment.py* je skript koji pokreće evaluaciju 10-strukom unakrsnom validacijom za nekoliko varijanti modela grebene regresije i slučajnih šuma. Za svaku varijantu modela definisana je funkcija `validate(X, Y, dataset, attempt = 0)` koja kao parametre prima matricu podataka *X* dimenzija $N \times M$ (*N* je broj instanci, *M* je broj atributa), vektor vrednosti ciljne promenljive *Y* dužine *N*, parametar *dataset* koji predstavlja naziv skupa instanci korišćenog za evaluaciju i opcioni parametar *attempt* koji uzima neku od predefinisanih celobrojnih vrednosti i predstavlja različite konfiguracije modela. Povratna vrednost funkcije `validate` je uređeni par $(rmse, r_2)$, gde *rmse* predstavlja koren srednjekvadratne greške, a *r₂* je R^2 skor evaluiranog modela. U svakoj iteraciji unakrsne validacije čuvaju se predviđanja na skupu za testiranje, što znači da po završetku izvršavanja funkcije `validate` imamo predviđanja modela na svim instancama iz skupa *dataset*. U nastavku je lista datoteka u kojima je definisana `validate` funkcija za svaku od varijanti evaluiranih modela, dok su sami modeli i njihove konfiguracije detaljnije opisani u poglavljima 4.4 i 4.5.
 - *plain_ridge.py* je datoteka u kojoj je definisana `validate` funkcija za jednostavne modele grebene regresije
 - *plain_ridge_with_interactions.py* je datoteka u kojoj je definisana `validate` funkcija za jednostavne modele grebene regresije nad atributima sa interakcijama
 - *ridge_with_ffs.py* je datoteka u kojoj je definisana `validate` funkcija za modele grebene regresije sa dvostrukom selekcijom atributa unapred
 - *ridge_with_pca.py* je datoteka u kojoj je definisana `validate` funkcija za modele grebene regresije sa smanjenjem dimenzionalnosti prostora atributa korišćenjem analize glavnih komponenti

- *ridge_with_rfe.py* je datoteka u kojoj je definisana `validate` funkcija za modele grebene regresije sa dvostrukom rekurzivnom eliminacijom atributa
 - *ridge_with_rfe_cv.py* je datoteka u kojoj je definisana `validate` funkcija za modele grebene regresije sa dvostrukom rekurzivnom eliminacijom atributa, gde broj atributa odabranih u bar jednoj od faza selekcije predstavlja metaparametar modela
 - *random_forest.py* je datoteka u kojoj je definisana `validate` funkcija za modele slučajne šume
 - *random_forest_tuning_metaparams.py* je datoteka u kojoj je definisana `validate` funkcija za modele slučajne šume sa metaparametrima
- U datoteci *forward_feature_selection.py* definisana je funkcija `get_important_features(X_train, Y_train, num_iterations)` kojom se u matrici podataka za obučavanje *X_train* bira najinformativniji podskup atributa (kolona) sa najviše *num_iterations* elemenata metodom pohlepne selekcije atributa unapred (o ovoj metodi detaljnije se govori u poglavlju 4.4). *Y_train* predstavlja vektor vrednosti ciljne promenljive za instance iz skupa za obučavanje.
 - U datoteci *feature_selection_pca.py* definisana je funkcija `pca_feature_selection(X_train, X_test, num_components)`. Argumenti *X_train* i *X_test* respektivno predstavljaju matrice podataka sa instancama za obučavanje, odnosno testiranje. Povratna vrednost funkcije je uređeni par (*X_train_transformed*, *X_test_transformed*) matrica podataka transformisanih metodom analize glavnih komponenti. Argument *num_components* može da ima pozitivnu celobrojnu vrednost i u tom slučaju predstavlja broj glavnih komponenti koje zadržavamo prilikom transformacije. Ukoliko ovaj argument ima vrednost između 0 i 1, onda on predstavlja udeo objašnjene varijanse koji želimo da postignemo transformacijom.
 - Datoteka *GCRF.py* sadrži implementaciju metode Gausovih uslovnih slučajnih polja, čiji je autor Andrija Petrović. Za instancu klase `GCRF` definisane su metode `predict(self, R, Se)` za predviđanje i `fit(self, R, Se, y, learn, x0 = None, maxiter = 1000, learnrate = 0.01)` za obučavanje. U nastavku je opis argumenata:
 - *R* predstavlja matricu predviđanja nestrukturiranih modela čije kolone predstavljaju broj različitih modela, a redovi su instance

- *Se* je tenzor koji predstavlja veze u grafovima zavisnosti, oblika [*broj instanci*, *broj grafova*, *broj čvorova(y)*, *broj čvorova(y)*]
- *y* je matrica vrednosti ciljne promenljive, redovi ove matrice predstavljaju instance, dok kolone predstavljaju čvorove grafa zavisnosti
- argument *learn* određuje koji optimizacioni algoritam se koristi prilikom obučavanja; za eksperimentalnu evaluaciju GCRF modela korišćena je SLSQP metoda (eng. Sequential Least Squares Programming) iz paketa `scipy`
- argument *x₀* predstavlja početnu tačku numeričke optimizacije
- *maxiter* predstavlja maksimalan broj iteracija numeričke optimizacije
- *learnrate* predstavlja korak numeričke optimizacije
- U datoteci `gcrf_similarity_matrix.py` definisana je funkcija `similarity_matrix(Y, delta, svd = False, reduce_dim_by = 1)`. Ova funkcija vraća simetričnu matricu dimenzija *broj rešavača* × *broj rešavača* čije su vrednosti mere sličnosti između rešavača (na dijagonali matrice su nule). Sličnost između rešavača definisana je u poglavlju 4.6. Opis argumenata funkcije je u nastavku.
 - *Y* je matrica čije vrste predstavljaju instance, a kolone različite rešavače
 - *delta* je parametar sličnosti
 - *svd* uzima vrednosti `True` ili `False` u zavisnosti od toga da li vektorske reprezentacije rešavača računamo korišćenjem singularne dekompozicije matrice *Y* (`True`) ili koristimo kolone matrice *Y* kao vektorske reprezentacije rešavača (`False`)
 - *reduce_dim_by* argument se koristi ukoliko *svd* uzima vrednost `True` i predstavlja broj najmanjih singularnih vrednosti koje odbacujemo nakon singularne dekompozicije matrice *Y*
- Datoteka `gcrf_with_rf.py` predstavlja skript za računanje predviđanja varijante GCRF modela u kojoj su vektorske reprezentacije rešavača vektori vrednosti ciljne promenljive. Predviđanja se vrše na instancama za testiranje korišćenjem modela obučenog na instancama za treniranje u okviru 5-struke unakrsne validacije, što znači da su na kraju dostupna predviđanja na celom skupu instanci. Čuvaju se predviđanja i za strukturirani GCRF model i za nestrukturirane modele slučajne

šume. Koraci eksperimentalne evaluacije detaljno su opisani u poglavlju 4.6.

- Datoteka *gcrf_with_rf_svd.py* predstavlja skript za računanje predviđanja varijante GCRF modela u kojoj se vektorske reprezentacije rešavača računaju uz pomoć singularne dekompozicije. Slično kao kod *gcrf_with_rf.py*, čuvaju se predviđanja i za strukturirane i za nestrukturirane modele.
- U direktorijumima *ridge_predictions/*, *rf_predictions/* i *gcrf_predictions/* nalaze se datoteke sa predviđanjima modela grebene regresije, slučajnih šuma i strukturiranih GCRF modela zajedno sa odgovarajućim predviđanjima za nestrukturirane modele slučajne šume (respektivno). Ove datoteke dobijaju se kao rezultat pokretanja skriptova *run_experiment.py*, *gcrf_with_rf.py* i *gcrf_with_rf_svd.py*.
- Datoteke *ridge_evaluation.py*, *rf_evaluation.py* i *gcrf_evaluation.py* predstavljaju skripte za računanje mera kvaliteta redom modela grebene regresije, modela slučajne šume i strukturiranih GCRF modela zajedno sa nestrukturiranim modelima slučajne šume.
- *plots.ipynb* koristi se za generisanje grafika koji ilustruju raspodelu greške predviđanja za razne modele.

4.2 Skupovi podataka

Ovo poglavlje posvećeno je podacima koji su korišćeni za eksperimentalnu evaluaciju ispitanih modela. Važno je napomenuti da su jedni skupovi podataka korišćeni za evaluaciju modela grebene regresije i slučajnih šuma, a drugi za evaluaciju modela Gausovih uslovnih slučajnih polja. Razlog je što su među podacima korišćenim za evaluaciju modela grebene regresije i slučajnih šuma dostupna vremena izvršavanja za svega 5 rešavača. Kako forma Gausovih uslovnih slučajnih polja modeluje moguću zavisnost među rešavačima, za evaluaciju modela Gausovih uslovnih slučajnih polja korišćeni su podaci sa vremenima izvršavanja većeg broja rešavača.

4.2.1 Podaci za eksperimentalnu evaluaciju modela grebene regresije i slučajnih šuma

Radi mogućnosti poredjenja rezultata, za eksperimentalnu evaluaciju modela grebene regresije i slučajnih šuma korišćeni su podaci iz referentnog rada [1].

Među tim podacima nalaze se instance SAT problema podeljene u 5 klasa na osnovu porekla instanci:

- INDU je skup industrijskih instanci
- HAND je skup ručno pravljenih instanci
- RAND je skup nasumično generisanih instanci
- SWV je skup instanci za verifikaciju softvera
- IBM je skup instanci za verifikaciju hardvera

Operacijama nad navedenim skupovima dobijena su još 3 skupa instanci:

- COMPETITION predstavlja uniju skupova INDU, HAND i RAND
- SWV-IBM predstavlja uniju skupova SWV i IBM
- RANDSAT je podskup skupa RAND i sadrži samo zadovoljive instance

Nad instancama svih navedenih skupova dostupno je vreme izvršavanja Minisat 2.0 rešavača [32]. Nad instancama skupova INDU, SWV i IBM dostupna su vremena izvršavanja dodatna dva rešavača: CryptoMinisat [33] i SPEAR [34]. Za evaluaciju performansi algoritama lokalne pretrage, u referentnom radu [1] su korišćene instance skupa RANDSAT. Na njima su primenjena dva rešavača: tnm [35] i algoritam dinamičke lokalne pretrage SAPS [36]. Pregled skupova instanci sa kardinalnostima i listom rešavača za koje su dostupne performanse na instancama odgovarajućeg skupa nalazi se u tabeli 1.

Naziv skupa	Broj instanci	Rešavači
COMPETITION	7012	Minisat
HAND	1955	Minisat
RAND	3381	Minisat
INDU	1676	Minisat, CryptoMinisat, SPEAR
SWV-IBM	1369	Minisat, CryptoMinisat, SPEAR
IBM	765	Minisat, CryptoMinisat, SPEAR
SWV	604	Minisat, CryptoMinisat, SPEAR
RANDSAT	2076	tnm, SAPS

Tabela 1: Skupovi podataka za evaluaciju modela grebene regresije i modela slučajne šume; treća kolona sadrži spisak rešavača za koje su dostupne performanse u vidu vremena izvršavanja na instancama datih skupova

Vreme izvršavanja rešavača izraženo u sekundama, sa maksimalnom vrednošću 3600. Izvršavanje SAT rešavača prilikom prikupljanja podataka prekinuto je nakon sat vremena ukoliko nije završeno u međuvremenu. Za instance na kojima je prekinuto izvršavanje upisana je maksimalna vrednost.

4.2.2 Podaci za eksperimentalnu evaluaciju modela Gausovih uslovnih slučajnih polja

Za evaluaciju modela Gausovih uslovnih slučajnih polja korišćeni su skupovi podataka za obučavanje modela koji koristi SATzilla 2012 metoda za odabir SAT rešavača po instanci SAT problema na osnovu portfolija¹⁰ [28], koja je odnela pobedu na SAT izazovu (eng. SAT challenge) iz 2012. godine (<http://www.cs.ubc.ca/labs/beta/Projects/SATzilla/>). Starije verzije SATzilla selektora koristile su model grebene regresije sa dvostrukom selekcijom atributa unapred¹¹ (detaljnije o ovoj metodi govori se u poglavlju 4.4), dok je SATzilla 2012 unapređena tako što koristi slučajnu šumu za kreiranje portfolija [38]. Među pomenutim podacima mogu se pronaći vremena izvršavanja 31 različitog rešavača nad instancama koje su grupisane u sledeće skupove:

- SATHAND12S (ručno pravljene instance)
- SATINDU12S (industrijske instance)
- SATRAND12S (nasumično generisane instance)
- SATALL12S (podskup unije prethodna tri skupa)

Kardinalnosti skupova instanci predstavljene su u tabeli 2, a spisak rešavača je sledeći: *ebglucose*, *ebminisat*, *glucose2*, *glueminisat*, *lingeling*, *lrglshr*, *minisatpsm*, *mphaseSAT64*, *precosat*, *qutersat*, *rcl*, *restartsat*, *cryptominisat2011*, *spear-sw*, *spear-hw*, *eagleup*, *sparrow*, *marchrw*, *mphaseSATm*, *satime11*, *tnm*, *mx09*, *gnoveltyp2*, *sattime*, *sattimep*, *clasp2*, *clasp1*, *picosat*, *mphaseSAT*, *sapperlot*, *sol*. Na instancama ovih skupova evaluirani su i nestrukturirani

¹⁰U ovom slučaju portfolio je zasnovan na predviđanju performansi različitih rešavača na jednoj instanci, i na osnovu njega može se automatizovati izbor najboljeg rešavača za tu instancu.

¹¹Uprkos tome što linearni model grebene regresije sa dvostrukom selekcijom atributa unapred nema dobar kvalitet predviđanja, za većinu rešavača koren srednjekvadratne greške ne prelazi red veličine, što je bilo dovoljno da omogući da SATzilla osvoji po pet medalja na SAT takmičenjima iz 2007. i 2009. godine. Prelazak na nelinearne modele slučajne šume posle 2009. godine dodatno je unapredio performanse SATzilla selektora.

modeli slučajne šume, radi poređenja kvaliteta predviđanja sa strukturiranim modelima Gausovih uslovnih slučajnih polja.

Naziv skupa	Broj instanci
SATALL12S	1614
SATHAND12S	767
SATINDU12S	1167
SATRAN12S	1362

Tabela 2: Skupovi podataka za modele Gausovih uslovnih slučajnih polja

Slično kao kod podataka za evaluaciju modela grebene regresije i slučajnih šuma, vreme izvršavanja rešavača je izraženo u sekundama, ali je maksimalna vrednost 12000 jer je izvršavanje SAT rešavača prekidano nakon 200 minuta prilikom sakupljanja podataka ako nije završeno u međuvremenu. Za instance na kojima je prekinuto izvršavanje upisana je vrednost 12000.

4.2.3 Atributi SAT instance

Atributi iskaznih formula u KNF obliku na osnovu kojih se vrše predviđanja navedeni su u tabeli 3. Ukupan broj atributa je 138. Označeni su po pripadnosti jednoj od četiri kategorije: trivijalno, jeftino, umereno i skupo, prema efikasnosti izračunavanja vrednosti odgovarajućeg atributa. Iako su vrednosti svakog od atributa izračunljive u polinomijalnom vremenu, za velike instance SAT problema neki od ovih atributa su vremenski veoma skupi za izračunavanje. Pomenuta kategorizacija je od koristi ukoliko za model koji predviđa vreme izvršavanja SAT rešavača imamo primenu koja je osetljiva na brzinu predviđanja. Kod ovakvih primena trebalo bi obučiti model na podskupu atributa koji sadrži što manje vremenski skupih atributa, odnosno napraviti kompromis između kvaliteta modela i brzine predviđanja na neviđenoj instanci SAT problema.

Atributi veličine ulaza:	
1 - 2.	broj promenljivih i klauza u originalnoj formuli (trivijalno): označeni sa b i c , respektivno
3 - 4.	broj promenljivih i klauza nakon simplifikacije korišćenjem SATE-lite¹²preprocesora (jeftino): označeni sa v' i c' , respektivno
5 - 6.	redukcija broja promenljivih i klauza nakon simplifikacije (jeftino): $(v - v')/v'$ i $(c - c')/c'$
7.	odnos broja promenljivih u odnosu na broj klauza (jeftino): v'/c'
VCG¹³atributi:	
8 - 12.	statistike stepena čvora promenljive (skupo): prosek, koeficijent varijacije, minimum i maksimum
13 - 17.	statistike stepena čvora klauze (jeftino): prosek, koeficijent varijacije, minimum, maksimum i entropija
VG¹⁴atributi (skupo):	
18 - 21.	statistike stepena čvora: prosek, koeficijent varijacije, minimum i maksimum
22 - 26.	dijametar: prosek, koeficijent varijacije, minimum, maksimum i entropija
CG¹⁵atributi (skupo):	
27 - 31.	statistike stepena čvora: prosek, koeficijent varijacije, minimum i maksimum
32 - 36.	koeficijent klasterovanja: prosek, koeficijent varijacije, minimum, maksimum i entropija
Atributi ravnoteže:	
37 - 41.	odnos pozitivnih i negativnih literala u svakoj klauzi (jeftino): prosek, koeficijent varijacije, minimum, maksimum i entropija
42 - 46.	odnos pozitivnih i negativnih pojavljivanja svake promenljive (skupo): prosek, koeficijent varijacije, minimum, maksimum i entropija
47 - 49.	udeo unarnih, binarnih i ternarnih klauza (jeftino)
Približnost Hornovoj formuli (skupo):	
50.	udeo Hornovih klauza
51 - 55.	broj pojavljivanja svake promenljive u Hornovoj klauzi: prosek, koeficijent varijacije, minimum, maksimum i entropija
Atributi DPLL pretrage:	
56 - 60.	broj propagacija jediničnih klauza (skupo): izračunati na dubinama 1, 4, 16, 64 i 256
61 - 62.	procena veličine prostora pretrage (jeftino): prosek dubine do kontradikcije, procena logaritma broja čvorova
LP¹⁶atributi (umereno):	
63 - 66.	vektor celobrojnih ograničenja kojima se nejednakost transformiše u jednakost (eng. integer slack vector): prosek, koeficijent varijacije, minimum i maksimum
67.	udeo celobrojnih promenljivih u LP rešenju
68.	vrednost ciljne funkcije
Atributi lokalne pretrage¹⁷(jeftino):	
69 - 78.	broj koraka do najboljeg lokalnog minimuma: prosek, medijan, koeficijent varijacije, 10. i 90. procenat
79 - 82.	prosek odstupanja od najboljeg lokalnog minimuma: prosek i koeficijent varijacije odstupanja po koraku
83 - 86.	udeo odstupanja od prvog lokalnog minimuma: prosek i koeficijent varijacije
87 - 90.	najbolje rešenje: prosek i koeficijent varijacije
Atributi učenja klauza¹⁸(jeftino):	
91 - 99.	broj naučenih klauza: prosek, koeficijent varijacije, minimum, maksimum, 10%, 25%, 50%, 75% i 90%
100 - 108.	dužina naučene klauze: prosek, koeficijent varijacije, minimum, maksimum, 10%, 25%, 50%, 75% i 90%
Atributi propagacije istraživanja (umereno):	
109 - 117.	poverenje propagacije istraživanja: za svaku promenljivu, računa se maksimum vrednosti $P(0)/P(1)$ i $P(1)/P(0)$; zatim se izračunaju prosek, koeficijent varijacije, minimum, maksimum, 10%, 25%, 50%, 75% i 90%
118 - 126.	slobodne promenljive (eng. unconstrained variables): za svaku promenljivu, računa se verovatnoća da je slobodna; zatim se izračunaju prosek, koeficijent varijacije, minimum, maksimum, 10%, 25%, 50%, 75% i 90%
Vremenski atributi:	
127 - 138.	procesorsko vreme potrebno za izračunavanje vrednosti atributa: meri se vreme 12 različitih blokova koda za izračunavanje vrednosti prethodno navedenih atributa

Tabela 3: Atributi instanci

4.3 Opšte eksperimentalne postavke

U nastavku se diskutuju detalji eksperimentalne evaluacije nekih metoda mašinskog učenja korišćenih u sličnim radovima za predviđanje vremena izvršavanja SAT rešavača: grebene regresije [15, 16, 17, 18, 19] i slučajnih šuma [1], i metode za modelovanje pomenutog problema predložene u ovom radu: Gausovih uslovnih slučajnih polja. Ovde ćemo navesti detalje koji su zajednički svim eksperimentima: preliminarne transformacije pripreme podataka koje olakšavaju modelovanje i korišćene mere kvaliteta i tehnike ocenjivanja. Dodatna preprocesiranja nad podacima specifična za određeni model opisana su u odgovarajućim poglavljima.

Podaci su predstavljeni matricom dimenzija $N \times M$ i vektorom vrednosti ciljne promenljive dužine N . N predstavlja broj instanci u skupu podataka, a M broj atributa.

Ciljna promenljiva, odnosno vreme izvršavanja SAT rešavača, posmatra se kao mera performanse tog rešavača. Ona se najpre transformiše primenom logaritamske funkcije (sa osnovom 10), što znači da modeli efektivno predviđaju logaritam performanse rešavača. Ova transformacija pokazala se bitnom zbog visoke varijacije među netransformisanim vrednostima ciljne promenljive. Zbog rezolucije CPU časovnika, vremena kraća od 0.01s nisu detektovana, pa da bi logaritam bio dobro definisan u ovim slučajevima, pre logaritamske transformacije postavlja se donja granica ciljne promenljive na 0.005 [1].

Nedostajuće vrednosti atributa prisutne su u podacima, ali se retko pojavljuju, stoga je iskorišćena jednostavna tehnika za baratanje ovim vrednostima: zamenjene su prosečnom vrednošću odgovarajućeg atributa na skupu za obučavanje, tako da budu minimalno informativne. U modelima grebene regresije ova transformacija se malo drugačije sprovodi: podaci za obučavanje se za neke modele standardizuju (transformacija nakon koje je prosek 0, a

¹²SATElite je metoda za minimizovanje broja promenljivih i klauza u iskaznoj formuli u KNF, obično se koristi za pripremu SAT instance pre nego što se pokrene SAT rešavač

¹³Od eng. Variable-Clause Graph: bipartitivni graf sa čvorom za svaku promenljivu i svaku klauzu, između kojih postoji grana ukoliko se odgovarajuća promenljiva pojavljuje u odgovarajućoj klauzi

¹⁴Od eng. Variable Graph: graf sa čvorom za svaku promenljivu, između kojih postoji grana ukoliko postoji klauza u kojoj se odgovarajuće promenljive zajedno pojavljuju

¹⁵Od eng. Clause Graph: graf sa čvorom za svaku klauzu, između kojih postoji grana ukoliko odgovarajuće klauze dele bar jedan negativan literal

¹⁶Atributi zasnovani na linearnom programiranju

¹⁷Zasnovani na pokretanju SAPS i GSAT rešavača u trajanju od 2 sekunde

¹⁸Zasnovani na pokretanju Zchaff-rand rešavača u trajanju od 2 sekunde

standardna devijacija 1 za svaku kolonu matrice podataka za obučavanje) a za neke samo centriraju (tako da je prosek po svakoj koloni 0). Pritom se prilikom ovih transformacija zanemaruju nedostajuće vrednosti, a nakon transformacija se te vrednosti postavljaju na 0. Ovo vodi potpunom ignorisanju nedostajućih vrednosti u modelima grebene regresije jer se odgovarajući koeficijent modela množi nulom.

Za ocenu kvaliteta korišćeni su koren srednjekvadratne greške (RMSE) i R^2 skor. Modeli su evaluirani korišćenjem ugneždene unakrsne validacije (10-struke za modele grebene regresije i slučajne šume, odnosno 5-struke za modele Gausovih uslovnih slučajnih polja¹⁹).

4.4 Grebena regresija

U ovom radu ocenjeno je nekoliko modela grebene regresije. Korišćena je implementacija `scikit-learn` paketa²⁰ koja podržava automatski odabir najbolje metode za rešavanje minimizacionog problema tokom obučavanja na osnovu osobina matrice podataka. Prilikom evaluacije ugneždenom unakrsnom 10-strukom validacijom, metaparametar regularizacije λ uzima vrednosti iz $[10^{-6}, 10^{-5}, \dots, 10^2]$ i čuvaju se predviđanja na instancama za testiranje, a na kraju se nad tim predviđanjima računaju koren srednjekvadratne greške (RMSE) i R^2 skor. Modeli koji su korišćeni su opisani u nastavku.

1. Jednostavan model grebene regresije nad originalnim podacima

Koraci evaluacije ovog modela su sledeći:

- eliminišu se konstantne kolone: iz podataka za obučavanje najpre se odstrane konstantni atributi, odnosno kolone matrice podataka za koje je vrednost standardne devijacije manja od 10^{-6} ; iste attribute eliminišemo iz podataka za testiranje
- podaci se standardizuju: standardizacija transformiše matricu podataka za obučavanje tako da prosek po kolonama bude 0, a standardna devijacija 1; ovo se postiže tako što se od svakog elementa matrice podataka oduzme srednja vrednost, a zatim se to подели standardnom devijacijom odgovarajuće kolone; podaci za testira-

¹⁹Za evaluaciju modela Gausovih uslovnih slučajnih polja korišćena je 5-struka umesto 10-struke unakrsne validacije zbog zahtevnosti obučavanja i predviđanja kod ovih modela

²⁰http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

nje se transformišu korišćenjem srednjih vrednosti i standardnih devijacija kolona matrice podataka za obučavanje

- regularizacioni parametar se odredi 10-strukom unakrsnom validacijom na podacima za obučavanje
- model se obučava i zapamte se predviđanja obučenog modela na podacima za testiranje

Ocene kvaliteta predviđanja ovog modela mogu se videti na slici 2. Greške predviđanja ovog modela su oko reda veličine.

2. Jednostavan model grebene regresije nad podacima sa interakcijama

Ovaj model razlikuje se od prethodnog jedino po tome što se pre evaluacije skup podataka proširi interakcijama atributa:

- računaju se interakcije nad matricom podataka (svaka vrsta $x_i = [x_{i,1}, x_{i,2}, \dots, x_{i,M}]$ proširuje se vrednostima $x_{i,j} * x_{i,k}$ za $j = 1, \dots, M$ i $k = j, \dots, M$)
- dalje se postupa kao u slučaju 1.

Evaluacija ovog modela pokazala je slabiji kvalitet predviđanja u odnosu na model grebene regresije nad originalnim skupom atributa (bez interakcija). Ovo nas navodi na zaključak da su interakcije verovatno visoko korelirane, zbog čega je problem grebene regresije lošije uslovljen. Međutim, vreme izvršavanja SAT rešavača u zavisnosti od atributa instance SAT problema bolje se aproksimira polinomom nego linearnom funkcijom. U nastavku su opisani linearni polinomijalni modeli koji koriste metode za eliminaciju visoko koreliranih atributa. Ocene kvaliteta predviđanja ovog modela navedene su u dodatku A.1.

3. Modeli grebene regresije sa dvostrukom selekcijom atributa unapred

Familija modela grebene regresije sa dvostrukom selekcijom atributa unapred evaluirana je u referentnom radu [1]. Koraci evaluacije su sledeći:

- eliminišu se konstante kolone
- napravi se kopija podataka za obučavanje koja se standardizuje za prvu fazu selekcije atributa (jer ne želimo da standardizujemo podatke za obučavanje pre računanja interakcija, ali želimo da koristimo standardizovane podatke za selekciju)

- prva faza selekcije atributa (nad standardizovanim kopijom podataka za obučavanje iz prethodnog koraka): krene se od praznog skupa atributa, u svakom koraku zapamti se podskup sačinjen od atributa iz prethodnog koraka kojima je dodat jedan od preostalih atributa tako da se minimizuje srednjekvadratna greška dobijena 5-strukom unakrsnom validacijom nad modelom grebene regresije sa fiksiranim regularizacionim parametrom $\lambda = 10^{-2}$; zaustavljamo se nakon l koraka (vrednost l , koja predstavlja maksimalan broj selektovanih atributa, fiksirana je na 30) i biramo podskup sa najmanjom greškom; iz skupova za testiranje i obučavanje izbacujemo sve attribute koji ne pripadaju odabranom podskupu
- računaju se interakcije nad preostalim atributima i ponovo se eliminišu konstantne kolone
- podaci se standardizuju
- imamo dve varijante u zavisnosti od toga šta posmatramo kao metaparametre modela:
 - prva varijanta: broj koraka do zaustavljanja u drugoj fazi selekcije atributa q se fiksira na 20, regularizacioni parametar λ se odredi 10-strukom unakrsnom validacijom na podacima za obučavanje
 - druga varijanta: 5-strukom unakrsnom validacijom na podacima za obučavanje odrede se metaparametri modela: regularizacioni parametar λ i broj koraka do zaustavljanja u drugoj fazi selekcije atributa q , pri čemu q uzima vrednosti iz [1, 2, 4, 8, 16, 32, 64]
- druga faza selekcije atributa: slično prvoj fazi, samo ovog puta nad atributima proširenim interakcijama; zaustavljanje nakon q koraka
- model se obučava i zapamte se predviđanja obučenog modela na podacima za testiranje

U referentnom radu [1] objavljene su vrednosti korena srednjekvadratne greške (RMSE) opisanih modela. Rezultati evaluacije najboljeg linearnog modela iz pomenutog referentnog rada predstavljeni su radi poređenja na slici 2.

4. Modeli grebene regresije sa dvostrukom selekcijom atributa tehnikom rekurzivne eliminacije

U ovom radu dat je predlog modifikacije prethodno navedene familije modela. Modifikacija podrazumeva da se umesto selekcije atributa

unapred koristi metoda rekurzivne eliminacije atributa, takođe u dve faze. Cilj ove metode je da rekurzivno eliminiše attribute, kao što je navedeno nazivom. Postupak je sledeći: najpre se nad celim skupom atributa istrenira model grebene regresije (sa fiksiranim regularizacionim parametrom $\lambda = 10^{-2}$), zatim se izbaci atribut kome odgovara najmanji koeficijent u dobijenom modelu i postupak se ponavlja sa preostalim atributima. Do zaustavljanja dolazi kada preostane p atributa (gde je p unapred zadat broj), ili ukoliko p nije zadato unapred, atributi se rekurzivno izbacuju dok ne ostane samo jedan, a usput se nad svakim podskupom pamti ocena modela dobijena 5-strukom unakrsnom validacijom, pa se na kraju uzima podskup sa najboljom ocenom. Koraci eksperimentalne evaluacije identični su koracima u slučaju 3, osim što se za selekciju atributa koristi rekurzivna eliminacija. Ocenjeno je nekoliko varijanti (l predstavlja broj atributa za prvu fazu selekcije, q za drugu; λ je regularizacioni parametar):

- fiksirano $l = q = 30$, λ je jedini metaparametar modela
- fiksirano $l = 30$, $q = 45$, λ je jedini metaparametar modela
- fiksirano $l = 30$, $q = 60$, λ je jedini metaparametar modela
- fiksirano $l = 30$, $q = 75$, λ je jedini metaparametar modela
- fiksirano $l = 30$, λ i q su metaparametri modela, q uzima vrednosti iz $[5, 10, 15, 20, \dots, 60]$
- fiksirano $l = 30$, λ i q su metaparametri modela, q uzima bilo koju vrednost između 1 i ukupnog broja atributa

Najbolje se pokazao model konfigurisan tako da se u prvoj fazi izabere $l = 30$ atributa, a u drugoj $q = 60$, dok je regularizacioni koeficijent λ metaparametar modela. Za ovaj model predviđanja se razlikuju od stvarnih vremena izvršavanja rešavača za pola do celog reda veličine, što je bolje od modela sa dvostrukom selekcijom atributa unapred čije greške predviđanja prelaze red veličine za neke skupove podataka. Ocene kvaliteta predviđanja ovog modela prikazane su na slici 2, dok se ocene ostalih modela iz ove familije mogu pronaći u dodatku A.1.

5. Modeli grebene regresije nad interakcijama sa smanjenjem dimenzionalnosti prostora atributa korišćenjem analize glavnih komponenti

Ova familija modela je rezultat još jedne modifikacije linearnog modela

grebene regresije iz referentnog rada [1]: umesto selekcije atributa unapred, za smanjenje dimenzionalnosti prostora atributa koristi se analiza glavnih komponenti. Koraci evaluacije su sledeći:

- računaju se interakcije nad atributima
- eliminišu se konstantne kolone
- podaci se centriraju oduzimanjem srednje vrednosti nad kolonama matrice podataka za obučavanje
- smanji se dimenzionalnost prostora atributa korišćenjem analize glavnih komponenti (na skupu za obučavanje) uz unapred zadat udeo objašnjene varijanse; za udeo objašnjene varijanse isprobane su vrednosti $1 - 10^{-3}$, $1 - 10^{-6}$ i $1 - 10^{-10}$
- regularizacioni parametar se odredi 10-strukom unakrsnom validacijom na podacima za obučavanje
- model se obučava i zapamte se predviđanja obučenog modela na podacima za testiranje

Ovaj pristup nije skalabilan jer se analiza glavnih komponenti primenjuje na matricu interakcija, što je memorijski skupo za skupove podataka sa velikim brojem instanci. U ovom slučaju bio je primenljiv s obzirom na to da skupovi korišćeni u eksperimentima nemaju više od oko 7000 instanci. Međutim, modeli iz ove familije pokazali su jako loš kvalitet predviđanja, pa ocene modela nisu prikazane u ovom poglavlju, ali se mogu pronaći u dodatku A.1.

4.5 Slučajne šume

Metoda slučajne šume pokazala se najboljom za modelovanje vremena izvršavanja SAT rešavača u zavisnosti od atributa SAT instance. Slučajne šume predstavljaju značajno unapređenje u odnosu na linearne modele grebene regresije, što je pokazano u referentnom radu [1]. U ovom radu, za eksperimentalnu evaluaciju modela slučajne šume korišćena je implementacija `scikit-learn` biblioteke²¹. Pritom su za većinu konfiguracionih parametara korišćene uobičajene (eng. `default`) vrednosti koje definiše `scikit-learn`, osim za udeo slučajno izabranih atributa koji se razmatraju u koraku podele algoritma za konstrukciju stabla (*max.features*) i najmanji broj instanci u

²¹<http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>

čvoru stabla nad kojima može da se vrši podela (*min_samples_split*). Za udeo slučajno izabranih atributa koji se razmatraju u koraku podele uobičajena vrednost `scikit-learn` biblioteke je 1, dok je u ovom radu za neke modele ta vrednost fiksirana na 0.5, a za neke predstavlja slobodan metaparametar koji uzima vrednosti iz [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]. Što se tiče najmanjeg broja instanci nad kojima može da se vrši podela, u nekim modelima je taj broj fiksiran na 5, dok u drugim predstavlja metaparametar koji vrednosti uzima iz [2, 3, 4, 5, 6, 7, 8, 9, 10]. Broj stabala u slučajnoj šumi je takođe promenljiv u familiji analiziranih modela: ocenjeni su modeli sa 10, 20, 30, 50, 100, 200, 1000 i 2000 stabala.

Kao i u slučaju linearnih modela, ocena se dobija 10-strukom unakrsnom validacijom, a kao mere kvaliteta koriste se koren srednjekvadratne greške (RMSE) i R^2 skor. Za model čija konfiguracija nije fiksirana, metaparametri se određuju 10-strukom unakrsnom validacijom tako da srednjekvadratna greška na skupu za obučavanje bude minimalna. Potpuni rezultati eksperimentalne evaluacije dostupni su u dodatku A.2, a na slici 2 mogu se videti ocene za najuspešniji model slučajne šume, gde je broj stabala 200, a udeo slučajno izabranih atributa koji se razmatraju u koraku podele algoritma za konstrukciju stabla (*max_features*) i najmanji broj instanci u čvoru stabla nad kojima može da se vrši podela (*min_samples_split*) predstavljaju metaparametre. Greška predviđanja slučajnih šuma uglavnom ne prelazi pola reda veličine, dok je udeo objašnjene varijanse ciljne promenljive oko 90% (čak i na skupovima sa svega nekoliko stotina instanci). Na slici 3 može se videti grafički prikaz raspodele greške predviđanja na instancama skupa SWV-IBM za slučajnu šumu sa 200 stabala, kao i za neke linearne modele (radi poređenja).

Povećanje broja stabala i uvođenje metaparametara čini proces obučavanja slučajnih šuma vremenski zahtevnijim. Eksperimentalna evaluacija pokazala je da model sa fiksiranom konfiguracijom i 200 stabala predstavlja dobar kompromis između kvaliteta modela i efikasnosti treniranja (pogledati datak A.2), stoga je ova konfiguracija iskorišćena za nestrukturirane modele prilikom evaluacije strukturiranog modela Gausovih uslovnih slučajnih polja.

rešavač	skup instanci	grebena regresija sa 2 x selekcijom atributa unapred	jednostavan model grebene regresije		grebena regresija sa 2 x rekurzivnom eliminacijom atributa		slučajna šuma		varijanta slučajne šume iz ref. rada
		RMSE	RMSE	R2	RMSE	R2	RMSE	R2	RMSE
Minisat	COMPETITION	1.01	1.042923	0.666227	0.911784*	0.744888	0.471026	0.931917	0.47
	HAND	1.05	1.097528	0.655961	0.976941*	0.727408	0.526037	0.920967	0.51
	RAND	0.64	0.615926	0.874635	0.531354*	0.906699	0.356164*	0.95808	0.37
	INDU	0.94	0.913403	0.716439	0.888544*	0.731664	0.554433	0.895523	0.52
	SWV-IBM	0.53	0.615198	0.856445	0.37302*	0.947222	0.160061*	0.990282	0.17
	IBM	0.51	0.758848	0.811347	0.53167	0.907394	0.203823	0.98639	0.19
	SWV	0.35	0.334839	0.86262	0.249233*	0.923886	0.083649	0.991426	0.08
Cryptominisat	INDU	0.94	0.908654	0.732454	0.89435*	0.740811	0.701254*	0.84065	0.72
	SWV-IBM	0.77	0.797553	0.806472	0.669898*	0.863466	0.46594*	0.933948	0.48
	IBM	0.65	0.79909	0.808176	0.702439	0.851772	0.426735	0.945295	0.41
	SWV	0.76	0.759614	0.780512	0.703566*	0.811707	0.514381	0.899354	0.51
SPEAR	INDU	0.95	0.9099	0.743719	0.881088*	0.759692	0.593445	0.890984	0.58
	SWV-IBM	0.67	0.699241	0.841205	0.535328*	0.906928	0.348854*	0.960475	0.38
	IBM	0.6	0.698201	0.857336	0.600746	0.894382	0.374706*	0.95891	0.38
	SWV	0.49	0.583156	0.774472	0.530672	0.813241	0.371665	0.908392	0.34
tnm	RANDSAT	1.01	1.026545	0.752234	0.959337*	0.783615	0.838402*	0.834732	0.88
SAPS	RANDSAT	0.94	0.974197	0.814134	0.843332*	0.860715	0.645279*	0.918454	0.66

$l = 30, q = 20,$
 $\lambda = 1e-2$

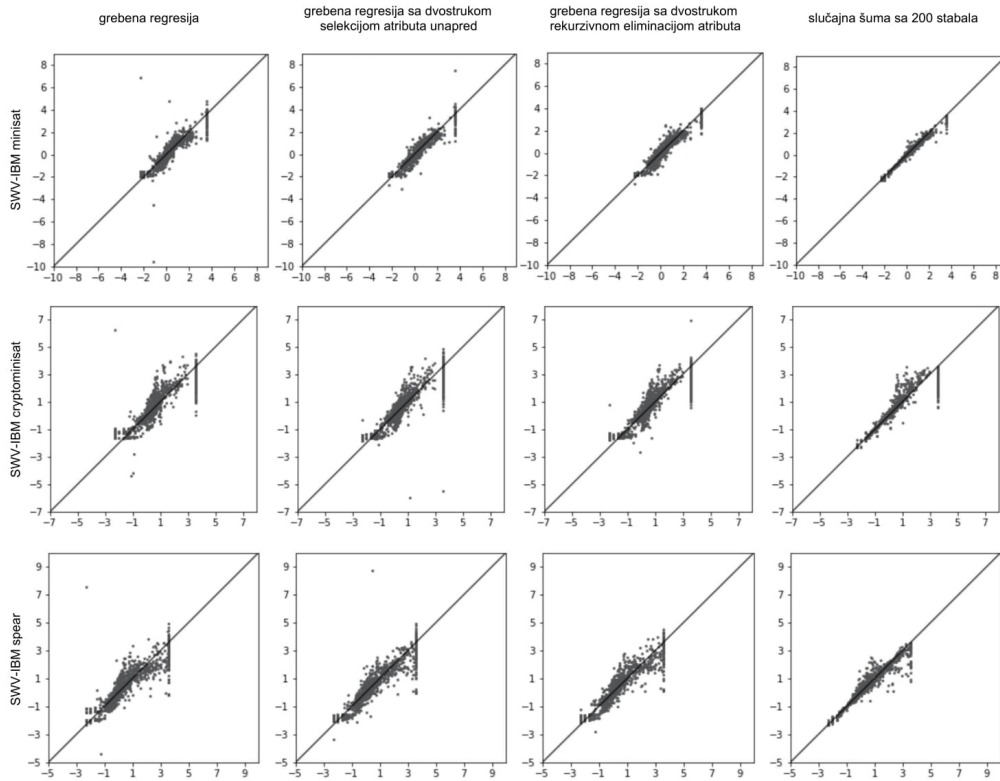
bez interakcija

$l = 30, q = 60,$
 λ je metaparametar

200 stabala,
 $(max_features, min_samples_split) \in \{0.1, 0.2, \dots, 1\} \times \{2, 3, \dots, 10\}$

10 stabala,
 $max_features = 0.5,$
 $min_samples_split = 5$

Slika 2: Ocene kvaliteta predviđanja evaluiranih modela. U prvoj koloni nalaze se ocene modela grebene regresije sa dvostrukom selekcijom atributa unapred iz referentnog rada [1] (u pomenutom referentnom radu prijavljen je samo koren srednjekvadratne greške), poslednja kolona prikazuje ocene modela slučajne šume iz referentnog rada [1]; 2. i 3. kolona prikazuju koren srednjekvadratne greške (RMSE) i R^2 skor za jednostavan model grebene regresije, u 4. i 5. koloni prikazane su iste ocene za model grebene regresije sa dvostrukom rekurzivnom eliminacijom atributa (debljim fontom označene su vrednosti korena srednjekvadratne greške koje su iste ili bolje od odgovarajućih ocena modela iz referentnog rada [1], bolje ocene su dodatno označene zvezdicom); 6. i 7. kolona prikazuju koren srednjekvadratne greške i R^2 skor za model slučajne šume evaluiran u ovom radu (takođe su označene ocene koje su iste ili bolje od odgovarajućih ocena modela iz referentnog rada [1]); u dnu tabele nalaze se konfiguracije modela (za modele grebene regresije λ predstavlja regularizacioni parametar, l i q predstavljaju broj atributa za selektovanje u prvoj, odnosno drugoj fazi; za slučajne šume, $max_features$ predstavlja udeo slučajno izabranih atributa koji se razmatraju u koraku podele algoritma za konstrukciju stabla, a $min_samples_split$ predstavlja najmanji broj instanci u čvoru stabla nad kojima može da se vrši podela).



Slika 3: Grafičko poređenje raspodela grešaka predviđanja modela jednostavne grebene regresije na originalnom skupu atributa (bez interakcija), modela grebene regresije sa dvostrukom selekcijom atributa unapred ($l = 30$, q i λ su metaparametri), modela grebene regresije sa dvostrukom rekurzivnom eliminacijom atributa ($l = 30$, $q = 60$, λ je metaparametar) i modela slučajne šume sa 200 stabala. Iznad grafika je naznačen odgovarajući model, a na levoj strani je naznačen rešavač. Na graficima su prikazane sve instance iz skupa SWV-IBM (svaka tačka odgovara jednoj instanci). Na x-osi predstavljeni su logaritmi stvarnih vremena izvršavanja, a na y-osi su predviđeni logaritmi vremena izvršavanja dobijeni 10-strukom unakrsnom validacijom. Primećuje se da je grafik za model slučajne šume manje raštrkan oko prave $y = x$, što znači da pravi manje greške prilikom predviđanja. Slične grafičke reprezentacije za ostale skupove instanci mogu se naći u dodatku A.2.

4.6 Gausova uslovna slučajna polja

U slučaju metode Gausovih uslovnih slučajnih polja postavlja se pitanje kako izabrati nestrukturirane modele i kako definisati sličnost između čvorova grafa zavisnosti. U ovom radu kao nestrukturirani modeli korišćeni su modeli slučajne šume, dok je sličnost definisana kao funkcija euklidskog rastojanja između vektorskih reprezentacija rešavača i skalirana na $[0, 1]$. Matrica sličnosti S konstruiše se na sledeći način:

$$S_{ij} = S_{ji} = \begin{cases} \frac{e^{-\delta d^2(r_i, r_j)}}{M}, & i \neq j \\ 0, & i = j \end{cases}, i, j = 1..N$$

U navedenoj formuli M predstavlja maksimalnu vrednost u matrici sličnosti, r_i i r_j su vektorske reprezentacije rešavača, dok δ predstavlja metaparametar sličnosti.

Isprobane su dve varijante vektorskih reprezentacija rešavača:

1. rešavač je predstavljen kao vektor dužine M (M je broj instanci u skupu za obučavanje) čije su koordinate vrednosti ciljne promenljive, odnosno vremena izvršavanja tog rešavača na instancama iz skupa za obučavanje, transformisana logaritmom
2. vektorska reprezentacija rešavača dobija se singularnom dekompozicijom matrice Y dimenzije $M \times N$ čije kolone predstavljaju vrednosti ciljne promenljive po rešavaču (M je broj instanci u skupu za obučavanje, a N broj rešavača): $Y = U\Sigma V^T$; matrica Y najpre se aproksimira matricom nižeg ranga tako što se odbaci k najmanjih singularnih vrednosti (k će prilikom obučavanja biti metaparametar), kao i odgovarajuće kolone matrice U i V , nakon čega dobijamo aproksimiranu matricu Y' i matrice U' i V' , pri čemu kolone matrice U' koje odgovaraju singularnim vrednostima različitim od 0 predstavljaju ortonormiranu bazu prostora kolona matrice Y' (poglavlje 3.2.5), pa se za vektorske reprezentacije rešavača uzimaju kolone matrice V'

U prvom slučaju, jedini metaparametar prilikom obučavanja je koeficijent sličnosti δ , dok u drugom slučaju uvodimo još jedan metaparametar: broj odbačenih singularnih vrednosti prilikom aproksimacije matrice Y , koji označavamo sa k .

Koraci eksperimentalne evaluacije detaljno su opisani pseudokodom na slici 4. Modeli su evaluirani ugnežđenom 5-strukom unakrsnom validaci-

jom. Metaparametri se određuju 5-strukom unakrsnom validacijom i uzimaju sledeće vrednosti:

- $\delta \in [0.01, 0.02, \dots, 0.1]$ - u slučaju da su vektorske reprezentacije rešavača vektori čije su koordinate vrednosti odgovarajućih ciljnih promenljivih
- $(\delta, k) \in \{1, 2, \dots, 10\} \times \{1, 2, \dots, N\}$, $N = 31$ je ukupan broj rešavača - u slučaju da se vektorske reprezentacije rešavača dobijaju singularnom dekompozicijom

za $i = 1..5$:

```
# X_trening_i predstavlja trening instance i-te iteracije unakrsne validacije
# X_test_i predstavlja test instance i-te iteracije unakrsne validacije
# X_trening_i deli se na 5 slojeva:
za j = 1..5:
  za k = 1..N: # N je ukupan broj rešavača
    # X_trening_trening_j predstavlja trening skup j-te iteracije
    # X_trening_test_j predstavlja test skup j-te iteracije
    na skupu X_trening_trening_j obučava se nestruktuirani model;
    na X_trening_test_j izračunaju se i zapamte predviđanja modela istreniranog u
    prethodnom koraku;
    # ovako se dobijaju predviđanja nestruktuiranih modela na celom X_trening_i skupu
    za l = k+1..N:
      # r_k, r_l - vektorske reprezentacije rešavača k i l
      sličnost između rešavača k i l računa se kao  $S_{kl} = S_k = \exp(-\delta d^2(r_k, r_l))$ ,
      skalirano na [0, 1];
    # sada su poznate sve vrednosti potrebne za obučavanje GCRF modela
    obučava se GCRF model na skupu X_trening_i;

    obučava se nestruktuirani model na celom X_trening_i;
    računaju se mere sličnosti na celom X_trening_i;

    izračunaju se i sačuvaju predviđanja GCRF modela na X_test_i (koristeći nestruktuirani
    model istreniran na celom X_trening_i, i mere sličnosti dobijene na celom X_trening_i);
    izračunaju se i sačuvaju predviđanja nestruktuiranog modela na X_test_i;
    # ovako dobijamo predviđanja GCRF modela i nestruktuiranih modela na celom skupu
    podataka

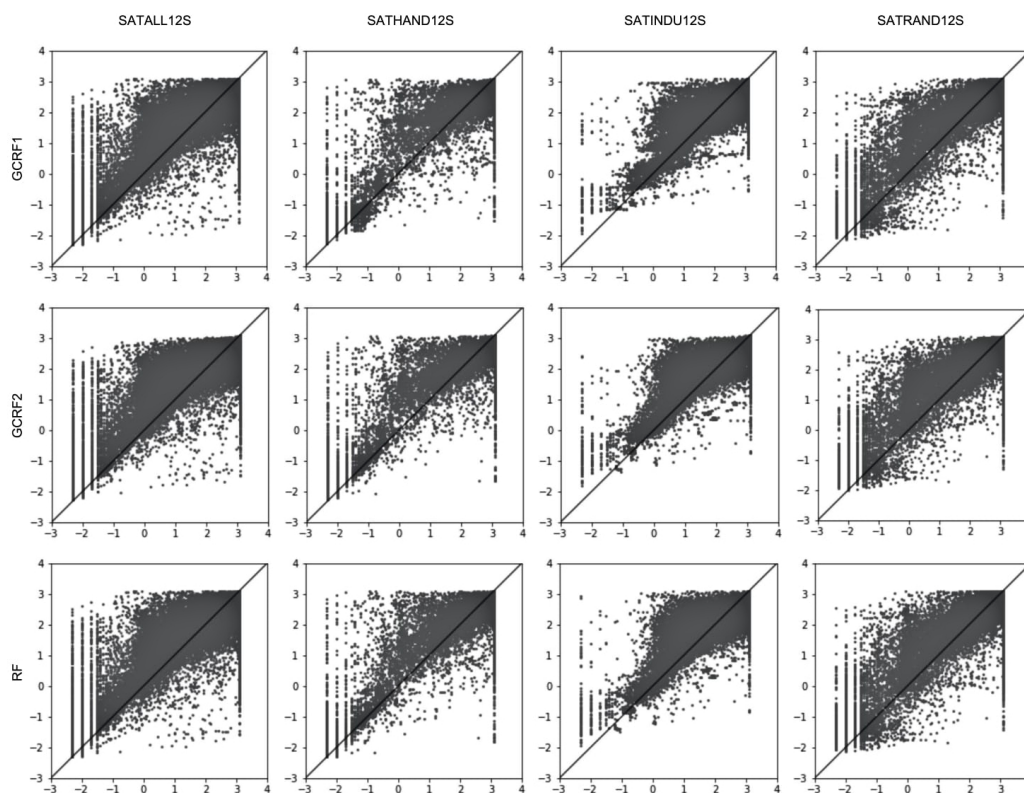
na dobijenim predviđanjima računamo mere kvaliteta: R2 i RMSE # možemo da odredimo mere
kvaliteta i za GCRF model i za nestruktuirani model radi poređenja
```

Slika 4: Koraci eksperimentalne evaluacije modela Gausovih uslovnih slučajnih polja

Na slici 5 prikazane su ocene strukturiranih modela Gausovih uslovnih slučajnih polja izračunate na svim rešavačima odjednom, zajedno sa ocenama nestrukturiranih modela slučajne šume. Vidi se da Gausova uslovna slučajna polja ipak ne predstavljaju unapređenje metode slučajnih šuma u kontekstu predviđanja vremena izvršavanja SAT rešavača. Obe metodologije rezultuju modelima za koje je koren srednjekvadratne greške predviđanja logaritma vremena izvršavanja između 0.5 i 0.7, što znači da se predviđena i stvarna vremena izvršavanja razlikuju za nešto više od pola reda veličine. R^2 skor nam govori da modeli objašnjavaju između 70 i 80 posto varijanse ciljne promenljive. Razlike u kvalitetu predviđanja su male, ali u korist modela slučajne šume. U dodatku A.3 mogu se videti ocene kvaliteta predviđanja izračunate posebno za svaki rešavač. Na slici 6 je grafički prikaz raspodela greške predviđanja za strukturirane GCRF modele i nestrukturirane modele slučajne šume.

skup instanci	model	RMSE	R2
SATALL12S	GCRF1	0.686863	0.734391
	GCRF2	0.701571	0.722894
	RF	0.686841	0.734408
SATHAND12S	GCRF1	0.635939	0.790754
	GCRF2	0.626386	0.796993
	RF	0.610744	0.807005
SATINDU12S	GCRF1	0.54889	0.731491
	GCRF2	0.527889	0.751644
	RF	0.514527	0.764058
SATRAND12S	GCRF1	0.609328	0.774146
	GCRF2	0.608054	0.77509
	RF	0.603225	0.778647

Slika 5: Rezultati evaluacije ugneždenom 5-strukom unakrsnom validacijom strukturiranih GCRF modela i nestrukturiranih modela slučajne šume; GCRF1 označava model sa direktnom vektorskom reprezentacijom rešavača, GCRF2 je oznaka za model koji koristi singularnu dekompoziciju za vektorsku reprezentaciju rešavača, dok je RF oznaka za slučajnu šumu.



Slika 6: Grafički prikaz raspodela grešaka predviđanja GCRF modela kod koga se rešavači predstavljaju vektorima vrednosti ciljne promenljive (GCRF1), GCRF modela kod koga se vektorske reprezentacije rešavača računaju korišćenjem singularne dekompozicije (GCRF2) i nestrukturiranog modela slučajne šume (RF). Iznad grafika je naznačeni su odgovarajući skupovi podataka. Na graficima su prikazane sve instance iz odgovarajućih skupova i svi rešavači (svaka tačka odgovara jednom paru instance i rešavača). Na x-osi predstavljeni su logaritmi stvarnih vremena izvršavanja, a na y-osi su predviđanja logaritma vremena izvršavanja dobijena 5-strukom unakrsnom validacijom.

5 Zaključak

U ovom radu razmatran je problem predviđanja vremena izvršavanja SAT rešavača. U referentnim radovima [15, 16, 17, 18, 19, 27, 28] pokazano je da linearne metode dovoljno dobro modeluju vreme izvršavanja rešavača u zavisnosti od atributa SAT instance, ali su nelinearni modeli, konkretno slučajne šume, mnogo pouzdaniji u ovom kontekstu, što pokazuje referentni rad [1]. Pomenute metode evaluirane su u ovom radu. Ocenjeni linearni modeli prave greške u predviđanju koje idu i do reda veličine na nekim skupovima podataka. Sa druge strane, greška predviđanja modela slučajne šume na tim skupovima podataka ne prelazi pola reda veličine, i ovaj model pritom objašnjava preko 90% varijanse ciljne promenljive. Ove dve metodologije upoređene su na slikama 2 i 3.

Osnovni motiv ovog rada je ispitivanje novog pristupa za predviđanje performansi SAT rešavača: metode Gausovih uslovnih slučajnih polja. Potencijalni nedostatak aktuelnih metoda je to što ne modeluju moguće zavisnosti između različitih SAT rešavača, jer ako su dva rešavača dovoljno slična u smislu da na većini instanci za obučavanje imaju približna vremena izvršavanja, logično bi bilo očekivati da će i na novoj, nevidenoj instanci takođe imati slične performanse. Predloženi regresioni modeli Gausovih uslovnih slučajnih polja (strukturirani modeli) računaju predviđanja na osnovu kompromisa: sa jedne strane oslanjaju se na predviđanja nestrukturiranih modela slučajne šume, a sa druge strane na predviđanja vremena izvršavanja za druge, slične rešavače. Tokom obučavanja, ukoliko predviđanja za slične rešavače mnogo odudaraju, smanjiće se poverenje u nestrukturirane modele.

Objašnjeni su razlozi zbog kojih se čini da metoda Gausovih uslovnih slučajnih polja možda može da unapredi kvalitet predviđanja slučajnih šuma, ali eksperimentalni rezultati predstavljeni na slici 5 ipak pokazuju suprotno. Kvalitet predviđanja ocenjenih modela Gausovih uslovnih slučajnih polja je zanemarljivo bolji na pojedinim klasama instanci za neke rešavače (ocene modela za pojedinačne rešavače mogu se videti u dodatku A.3), ali modeli slučajne šume u većini slučajeva ipak daju tačnija predviđanja. Pored toga, nedostatak modela Gausovih uslovnih slučajnih polja ogleda se i u činjenici da su i obučavanje i predviđanje vremenski zahtevnije operacije od obučavanja, odnosno predviđanja modela slučajne šume. Ima prostora za eventualna unapređenja predloženog pristupa: može se krenuti od obogaćivanja skupa nestrukturiranih modela (npr. mogu se kombinovati linearni i nelinearni nestrukturirani modeli, recimo modeli grebene regresije i slučajne šume). Novi nestrukturirani modeli možda bolje modeluju različite

aspekte problema i na taj način mogu da povećaju tačnost predviđanja modela Gausovih uslovnih slučajnih polja.

Dodatak

A Rezultati eksperimentalne evaluacije

A.1 Grebena regresija

Slike 7, 8, 9, 10 i 11 ilustruju mere kvaliteta evaluiranih linearnih modela. Zelenom bojom su označeni najbolji, a crvenom najslabiji rezultati u svakoj vrsti.

rešavač	skup instanci	grebena regresija			
		RMSE	R2	RMSE	R2
Minisat	COMPETITION	1.042923	0.666227	1.318389	0.466623
	HAND	1.097528	0.655961	3.264692	-2.044114
	RAND	0.615926	0.874635	0.43605	0.937167
	INDU	0.913403	0.716439	1.252247	0.467031
	SWV-IBM	0.615198	0.856445	0.732653	0.796397
	IBM	0.758848	0.811347	0.962877	0.696265
	SWV	0.334839	0.86262	0.542711	0.639098
Cryptominisat	INDU	0.908654	0.732454	3.902019	-3.933775
	SWV-IBM	0.797553	0.806472	0.929245	0.737285
	IBM	0.79909	0.808176	1.020609	0.687081
	SWV	0.759614	0.780512	0.935199	0.667315
SPEAR	INDU	0.9099	0.743719	5.227621	-7.45938
	SWV-IBM	0.699241	0.841205	0.846703	0.767167
	IBM	0.698201	0.857336	1.031075	0.688875
	SWV	0.583156	0.774472	0.774515	0.602177
tnm	RANDSAT	1.026545	0.752234	0.930911	0.796248
SAPS	RANDSAT	0.974197	0.814134	0.802805	0.87378

bez interakcija

sa interakcijama

Slika 7: Rezultati evaluacije 10-strukom unakrsnom validacijom jednostavnih modela grebene regresije sa i bez interakcija nad atributima

Evaluacijom jednostavnih modela grebene regresije, kod kojih se prilikom obučavanja ne koriste tehnike smanjivanja dimenzionalnosti prostora atributa, može se primetiti da za većinu SAT rešavača bolju moć predviđanja ima model koji radi sa originalnim skupom atributa, bez interakcija. To je najverovatnije zbog visoke koreliranosti interakcija, što dovodi do lošije

uslovljenosti problema linearne regresije. Sa druge strane, očekujemo da se logaritam vremena izvršavanja SAT rešavača u zavisnosti od atributa instance problema može bolje aproksimirati polinomom nego linearnom funkcijom, stoga očekujemo i da bi linearni polinomijalni modeli koji koriste metode za eliminaciju visoko koreliranih atributa trebalo da imaju bolji kvalitet predviđanja.

rešavač	skup instanci	grebena regresija sa dvostrukom selekcijom atributa unapred					
		RMSE	R2	RMSE	R2	RMSE	R2
Minisat	COMPETITION	1.124772	0.611781	1.13593	0.604041	0.973038	0.70946
	HAND	1.540324	0.322357	1.524413	0.336285	1.118204	0.642876
	RAND	0.619134	0.873326	0.683425	0.845652	0.56578	0.894217
	INDU	0.997601	0.661752	1.007015	0.655338	0.982206	0.672111
	SWV-IBM	0.536617	0.890776	0.549257	0.88557	0.401689	0.938798
	IBM	0.530881	0.907669	0.556982	0.898367	0.449792	0.933721
	SWV	0.244915	0.926501	0.26809	0.911933	0.226126	0.937345
Cryptominisat	INDU	1.008422	0.670478	1.01331	0.667275	0.937313	0.715311
	SWV-IBM	0.757961	0.825209	0.777743	0.815966	0.720942	0.841866
	IBM	0.646099	0.874596	0.738289	0.836256	1.09201	0.641767
	SWV	0.694549	0.816502	0.70758	0.809552	0.686917	0.820512
SPEAR	INDU	1.075032	0.642255	1.057428	0.653876	1.043507	0.66293
	SWV-IBM	0.65614	0.860178	0.663544	0.857005	0.601523	0.882487
	IBM	0.599386	0.89486	0.664674	0.870708	0.863013	0.782034
	SWV	0.471679	0.852455	0.504242	0.83138	0.474268	0.850831
tnm	RANDSAT	0.98253	0.773025	1.060057	0.735794	1.016419	0.757098
SAPS	RANDSAT	0.872647	0.850864	0.993569	0.806669	0.882907	0.847336

$l = 30, q = 20, \lambda = 1e-2$

$l = 30, q = 20, \lambda$ je metaparametar

$l = 30, q$ i λ su metaparametri

Slika 8: Rezultati evaluacije 10-strukom unakrsnom validacijom modela grebene regresije sa dvostrukom selekcijom atributa unapred

Od ispitanih modela grebene regresije sa dvostrukom selekcijom atributa unapred najveći kvalitet predviđanja na većini skupova instanci ima model sa konfiguracijom u kojoj je broj atributa u prvoj fazi selekcije l fiksan na 30, dok su broj atributa u drugoj fazi q i regularizacioni parametar λ metaparametri modela. Prilikom evaluacije ovog modela, u svakom koraku 10-struke ugneždene unakrsne validacije određuje se $(q, \lambda) \in \{1, 2, 4, 8, 16, 32, 64\} \times \{10^{-6}, 10^{-5}, \dots, 10^2\}$ 5-strukom unakrsnom validacijom tako da se minimizuje srednjekvadratna greška predviđanja. Kvalitet predviđanja je nešto bolji nego kod jednostavnog modela grebene regresije bez interakcija.

rešavač	skup instanci	grebena regresija sa dvostrukom rekurzivnom eliminacijom atributa							
		RMSE	R2	RMSE	R2	RMSE	R2	RMSE	R2
Minisat	COMPETITION	0.969538	0.711546	0.919639	0.740473	0.911784	0.744888	0.989449	0.699576
	HAND	0.943896	0.745537	1.342512	0.485231	0.976941	0.727408	1.027216	0.69863
	RAND	0.568556	0.893177	0.537725	0.904448	0.531354	0.906699	0.528661	0.907642
	INDU	0.925867	0.708648	0.903171	0.722756	0.888544	0.731664	0.87353	0.740655
	SWV-IBM	0.44553	0.924709	0.401643	0.938812	0.37302	0.947222	0.368607	0.948464
	IBM	0.498828	0.918482	6.518235	-12.919176	0.53167	0.907394	0.564004	0.895788
	SWV	0.248444	0.924368	0.255933	0.919739	0.249233	0.923886	0.247425	0.924987
Cryptominisat	INDU	0.933649	0.717533	0.914408	0.729055	0.89435	0.740811	0.911968	0.7305
	SWV-IBM	0.6794	0.859565	0.671618	0.862764	0.669898	0.863466	0.670238	0.863327
	IBM	0.726789	0.841318	0.687891	0.857848	0.702439	0.851772	0.687445	0.858033
	SWV	0.707644	0.809517	0.709396	0.808573	0.703566	0.811707	0.705769	0.810526
SPEAR	INDU	0.918791	0.738686	0.906563	0.745594	0.881088	0.759692	0.887748	0.756045
	SWV-IBM	0.57293	0.893393	0.547879	0.902512	0.535328	0.906928	0.525082	0.910456
	IBM	0.659715	0.87263	0.634617	0.882137	0.600746	0.894382	0.597151	0.895643
	SWV	0.47987	0.847286	0.498436	0.835241	0.530672	0.813241	0.4987	0.835066
tnm	RANDSAT	0.983351	0.772646	0.969467	0.779021	0.959337	0.783615	0.964966	0.781068
SAPS	RANDSAT	0.89081	0.844591	0.862582	0.854284	0.843332	0.860715	0.845906	0.859864

$l = 30, q = 30,$ $l = 30, q = 45,$ $l = 30, q = 60,$ $l = 30, q = 75,$
 λ je metaparametar λ je metaparametar λ je metaparametar λ je metaparametar

Slika 9: Rezultati evaluacije 10-strukom unakrsnom validacijom modela grebene regresije sa dvostrukom rekurzivnom eliminacijom atributa, 1. deo

rešavač	skup instanci	grebena regresija sa dvostrukom rekurzivnom eliminacijom atributa			
		RMSE	R2	RMSE	R2
Minisat	COMPETITION	0.919771	0.740399	0.909812	0.74599
	HAND	0.976941	0.727408	1.024787	0.700054
	RAND	0.531354	0.906699	0.547415	0.900973
	INDU	0.890369	0.73056	0.981337	0.672691
	SWV-IBM	0.374982	0.946666	0.895942	0.695528
	IBM	0.53547	0.906066	0.584732	0.887987
	SWV	0.28527	0.900284	0.252586	0.921824
Cryptominisat	INDU	0.901874	0.736432	1.14539	0.574884
	SWV-IBM	0.670482	0.863228	0.649382	0.8717
	IBM	0.704704	0.850815	1.116946	0.62522
	SWV	0.706613	0.810072	0.702353	0.812355
SPEAR	INDU	0.881088	0.759692	1.042577	0.66353
	SWV-IBM	0.535226	0.906963	0.501957	0.918169
	IBM	0.654108	0.874786	0.62873	0.884313
	SWV	0.485609	0.843612	0.495776	0.836995
tnm	RANDSAT	0.959337	0.783615	1.044858	0.743315
SAPS	RANDSAT	0.843332	0.860715	0.863704	0.853905

$l = 30, q \in \{5, 10, 15, \dots, 60\}$ $l = 30, q \in \{5, 10, 15, \dots, 60\}$

Slika 10: Rezultati evaluacije 10-strukom unakrsnom validacijom modela grebene regresije sa dvostrukom rekurzivnom eliminacijom atributa, 2. deo

Modeli koji umesto dvostruke selekcije atributa unapred koriste dvostruku rekurzivnu eliminaciju atributa pokazali su se kao najbolji linearni modeli za predviđanje vremena izvršavanja SAT rešavača. Pritom su najbolji kvalitet predviđanja prijavili modeli sa fiksiranim brojem atributa za obe faze selekcije: $l = 30$ i $q = 60$, odnosno $l = 30$ i $q = 75$.

rešavač	skup instanci	grebena regresija sa analizom glavnih komponenti					
		RMSE	R2	RMSE	R2	RMSE	R2
Minisat	COMPETITION	1.804223	0.001087	2.802033	-1.409316	9.595517	-27.254264
	HAND	1.865301	0.006256	1.861342	0.01047	2.307257	-0.520437
	RAND	1.612302	0.140962	1.184098	0.536665	0.895317	0.735105
	INDU	1.704006	0.013121	6.18215	-11.989748	8.083906	-21.210798
	SWV-IBM	1.568085	0.067332	1.450858	0.201568	1.15906	0.490434
	IBM	1.700613	0.052532	1.624926	0.134991	1.252926	0.485714
	SWV	0.676576	0.4391	0.562512	0.612282	0.499739	0.693988
Cryptominisat	INDU	1.75771	-0.00114	4.03415	-4.273569	2.392224	-0.854404
	SWV-IBM	1.784527	0.031118	1.61317	0.208256	1.268043	0.510793
	IBM	1.79697	0.02995	1.682453	0.149649	1.274191	0.512268
	SWV	1.07025	0.564292	0.934735	0.667644	1.090121	0.547962
SPEAR	INDU	1.785378	0.013288	5.162412	-7.249651	6.637776	-12.638779
	SWV-IBM	1.705495	0.055325	1.565212	0.20434	1.239404	0.501107
	IBM	1.806289	0.045163	1.700139	0.154091	1.286835	0.515381
	SWV	0.846236	0.525088	0.718553	0.657588	0.695355	0.679341
tnm	RANDSAT	1.827458	0.214801	1.534336	0.446489	1.15471	0.686504
SAPS	RANDSAT	1.945946	0.258403	1.620202	0.485904	1.192139	0.72167

udeo objašnjene varijanse 1 - 1e-3 udeo objašnjene varijanse 1 - 1e-6 udeo objašnjene varijanse 1 - 1e-10

Slika 11: Rezultati evaluacije 10-strukom unakrsnom validacijom modela grebene regresije sa smanjenjem dimenzionalnosti prostora atributa korišćenjem analize glavnih komponenti

Možemo primetiti da su se linearni modeli koji koriste analizu glavnih komponenti za smanjivanje dimenzionalnosti prostora atributa najlošije pokazali. Osim neskalabilnosti algoritma za obučavanje, problem je i veoma slab kvalitet predviđanja obučenog modela. Štaviše, najbolje predviđanje performanse Cryptominisat rešavača na skupu instanci INDU je lošije od predviđanja modela koji vraća srednju vrednost ciljne promenljive, što nam govori negativna vrednost R^2 skora.

A.2 Slučajne šume

Rezultati eksperimentalne evaluacije za slučajne šume ilustrovani su na slikama 12, 13 i 14. Zelenom bojom označene su najbolje vrednosti mera kvaliteta, a crvenom najlošije.

rešavač	skup instanci	slučajna šuma					
		RMSE	R2	RMSE	R2	RMSE	R2
Minisat	COMPETITION	0.509608	0.920307	0.489251	0.926547	0.483223	0.928346
	HAND	0.584495	0.902425	0.550053	0.913586	0.542222	0.916029
	RAND	0.391464	0.949359	0.36896	0.955014	0.365484	0.955858
	INDU	0.577682	0.886577	0.561708	0.892764	0.552289	0.89633
	SWV-IBM	0.168373	0.989247	0.165732	0.989582	0.167991	0.989296
	IBM	0.236221	0.981719	0.221108	0.983984	0.214806	0.984884
	SWV	0.08322	0.991514	0.082118	0.991737	0.08163	0.991835
Cryptominisat	INDU	0.740978	0.822085	0.718824	0.832565	0.715599	0.834064
	SWV-IBM	0.510665	0.920659	0.484657	0.928535	0.479924	0.929924
	IBM	0.440417	0.941731	0.434641	0.943249	0.426859	0.945263
	SWV	0.536674	0.890441	0.522161	0.896287	0.524709	0.895272
SPEAR	INDU	0.642961	0.872032	0.606782	0.886029	0.59743	0.889515
	SWV-IBM	0.369313	0.955703	0.362645	0.957288	0.362978	0.95721
	IBM	0.387694	0.956012	0.371799	0.959545	0.375742	0.958683
	SWV	0.395115	0.896467	0.382791	0.902825	0.38379	0.902317
tnm	RANDSAT	0.889222	0.814089	0.853	0.828926	0.845845	0.831784
SAPS	RANDSAT	0.653805	0.916285	0.640205	0.919732	0.635439	0.920922

10 stabala, max_features = 0.5, min_samples_split = 5
 20 stabala, max_features = 0.5, min_samples_split = 5
 30 stabala, max_features = 0.5, min_samples_split = 5

Slika 12: Rezultati evaluacije 10-strukom unakrsnom validacijom modela slučajne šume, 1. deo

rešavač	skup instanci	slučajna šuma					
		RMSE	R2	RMSE	R2	RMSE	R2
Minisat	COMPETITION	0.478795	0.929653	0.476312	0.930381	0.474654	0.930864
	HAND	0.535023	0.918243	0.531658	0.919269	0.528862	0.920116
	RAND	0.358716	0.957477	0.356165	0.95808	0.35596	0.958128
	INDU	0.55232	0.896318	0.553659	0.895815	0.552909	0.896097
	SWV-IBM	0.167628	0.989342	0.161393	0.99012	0.159955	0.990295
	IBM	0.211079	0.985404	0.207894	0.985841	0.206502	0.98603
	SWV	0.080302	0.992098	0.080699	0.99202	0.081128	0.991935
Cryptominisat	INDU	0.707194	0.837939	0.701138	0.840703	0.695378	0.84331
	SWV-IBM	0.477943	0.930501	0.470195	0.932736	0.468559	0.933204
	IBM	0.430684	0.944278	0.428053	0.944956	0.423702	0.94607
	SWV	0.517534	0.898116	0.514025	0.899494	0.512269	0.900179
SPEAR	INDU	0.593524	0.890955	0.595369	0.890276	0.595711	0.890149
	SWV-IBM	0.359193	0.958098	0.357359	0.958524	0.355752	0.958897
	IBM	0.375662	0.9587	0.372602	0.95937	0.369221	0.960104
	SWV	0.381374	0.903543	0.376869	0.905809	0.378411	0.905036
tnm	RANDSAT	0.845441	0.831945	0.840291	0.833986	0.833798	0.836542
SAPS	RANDSAT	0.633312	0.921451	0.630497	0.922147	0.630223	0.922215

50 stabala, max_features = 0.5, min_samples_split = 5
 100 stabala, max_features = 0.5, min_samples_split = 5
 200 stabala, max_features = 0.5, min_samples_split = 5

Slika 13: Rezultati evaluacije 10-strukom unakrsnom validacijom modela slučajne šume, 2. deo

rešavač	skup instanci	slučajna šuma					
		RMSE	R2	RMSE	R2	RMSE	R2
Minisat	COMPETITION	0.471026	0.931917	0.472388	0.931523	0.472479	0.931496
	HAND	0.526037	0.920967	0.526781	0.920743	0.52709	0.92065
	RAND	0.356164	0.95808	0.355592	0.958215	0.355891	0.958144
	INDU	0.554433	0.895523	0.55339	0.895916	0.553772	0.895772
	SWV-IBM	0.160061	0.990282	0.156445	0.990716	0.155765	0.990797
	IBM	0.203823	0.98639	0.207527	0.985891	0.207111	0.985947
	SWV	0.083649	0.991426	0.080412	0.992077	0.080424	0.992075
Cryptominisat	INDU	0.701254	0.84065	0.694714	0.843609	0.695515	0.843248
	SWV-IBM	0.46594	0.933948	0.466963	0.933658	0.466078	0.933909
	IBM	0.426735	0.945295	0.423449	0.946134	0.422413	0.946397
	SWV	0.514381	0.899354	0.511923	0.900314	0.51089	0.900716
SPEAR	INDU	0.593445	0.890984	0.59164	0.891646	0.59243	0.891356
	SWV-IBM	0.348854	0.960475	0.354005	0.9593	0.354362	0.959217
	IBM	0.374706	0.95891	0.370861	0.959749	0.369935	0.95995
	SWV	0.371665	0.908392	0.377229	0.905629	0.377498	0.905494
tnm	RANDSAT	0.838402	0.834732	0.833464	0.836673	0.833878	0.83651
SAPS	RANDSAT	0.645279	0.918454	0.630056	0.922256	0.629617	0.922365

200 stabala, $(\max_features, \min_samples_split) \in \{0.1, 0.2, \dots, 1\} \times \{2, 3, \dots, 10\}$

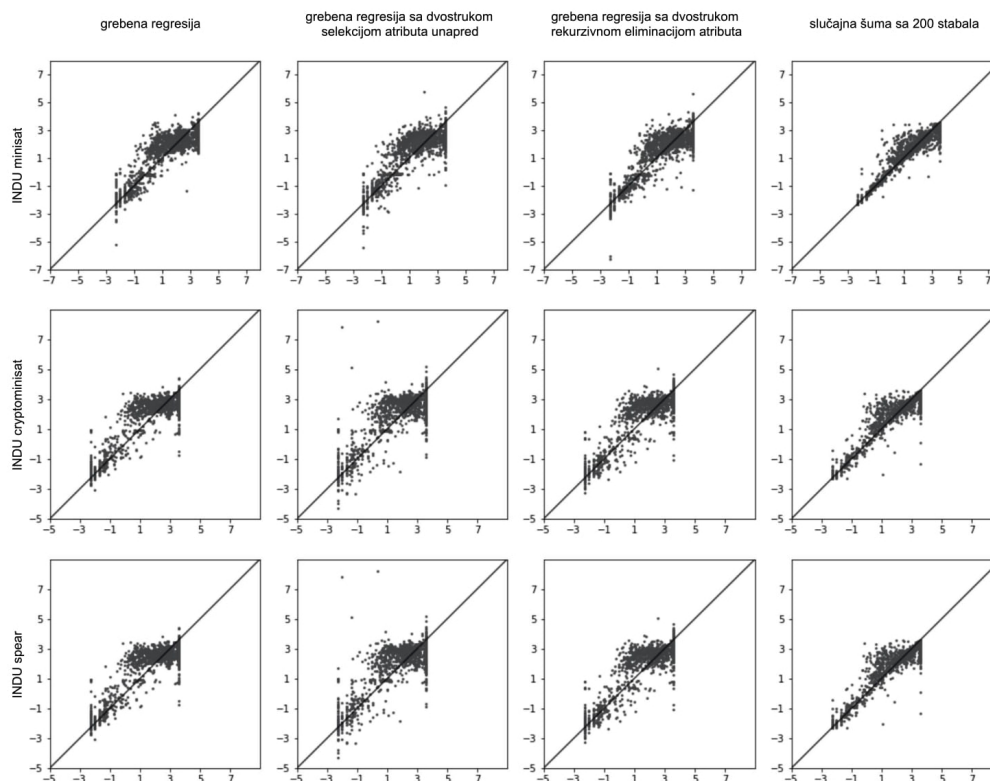
 1000 stabala, $\max_features = 0.5, \min_samples_split = 5$

 2000 stabala, $\max_features = 0.5, \min_samples_split = 5$

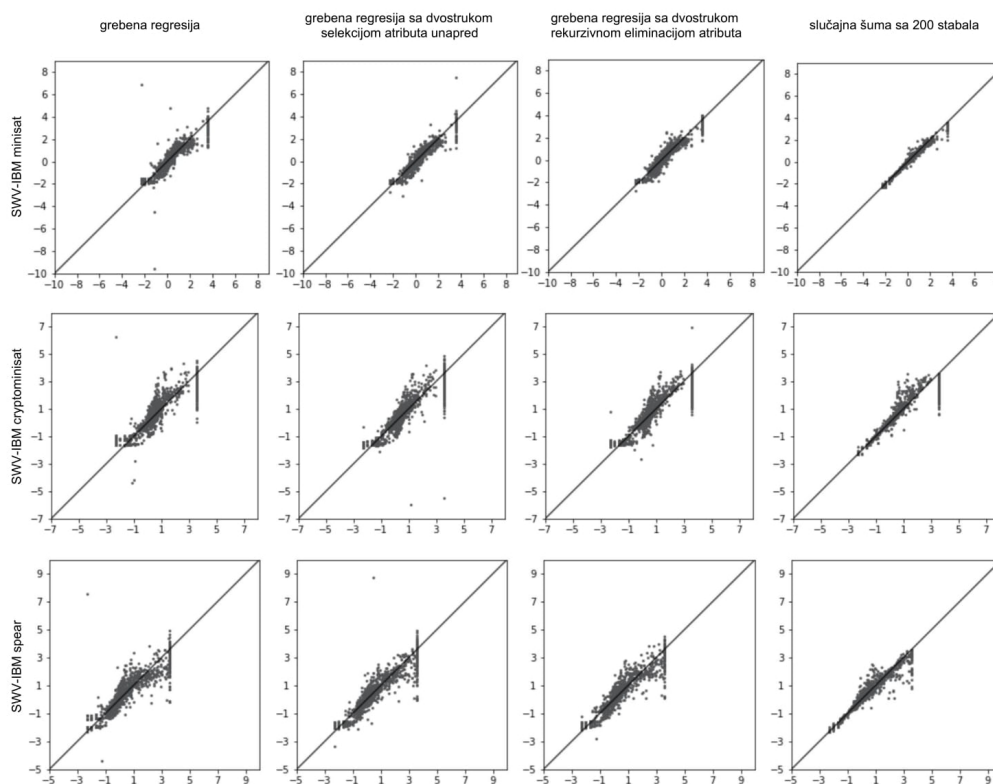
Slika 14: Rezultati evaluacije 10-strukom unakrsnom validacijom modela slučajne šume, 3. deo

Kao što vidimo, rezultati su značajno bolji nego kod linearnih modela. Čak i najslabiji model slučajne šume (sa 10 stabala i fiksiranom konfiguracijom) pravi greške u predviđanju u granicama od oko $10^{0.89} \approx 7.76$ sekundi. Može se primetiti da modeli imaju veći kvalitet predviđanja sa povećanjem broja stabala u slučajnoj šumi. Uvođenje metaparametara takođe je malo unapredilo kvalitet predviđanja.

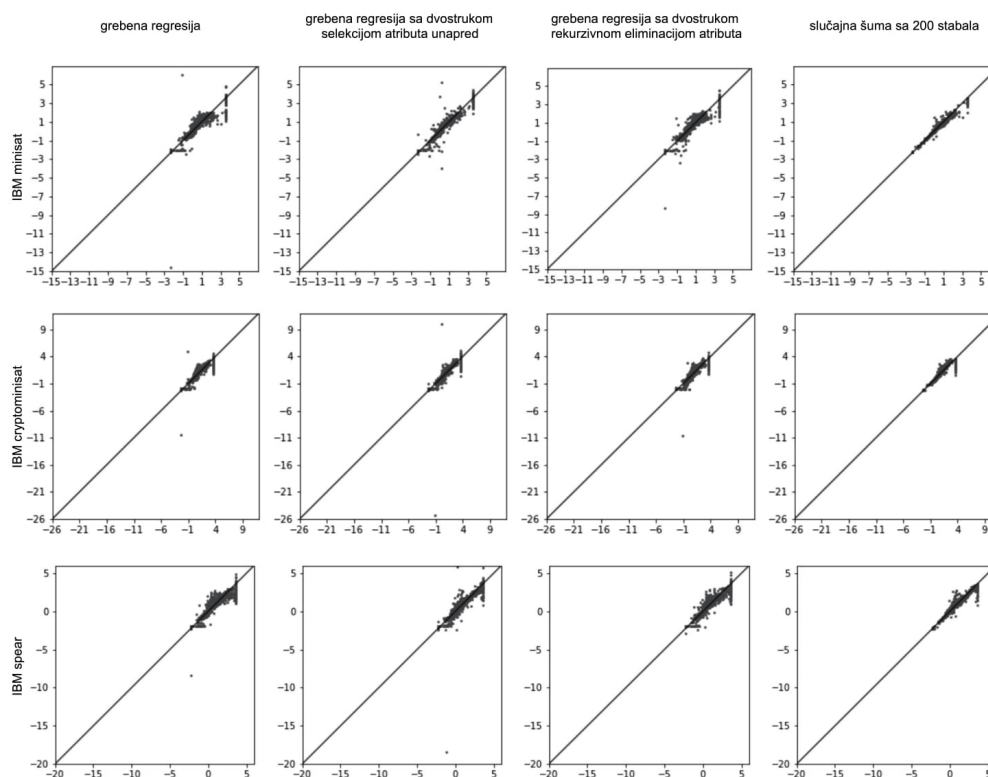
Na slikama 15, 16, 17, 18 i 19 nalazi se grafički prikaz raspodela greške predviđanja za neke od evaluiranih linearnih modela i za model slučajne šume sa 200 stabala. Svaka tačka na grafiku odgovara jednoj instanci, na x-osi su logaritmi stvarnih vremena izvršavanja SAT rešavača a na y-osi predviđanja modela na instancama za testiranje, dobijena 10-strukom unakrsnom validacijom. Na ovim slikama primećuje se da su grafici za modele slučajne šume manje raštrkani oko prave $y = x$, što znači da manje greše prilikom predviđanja.



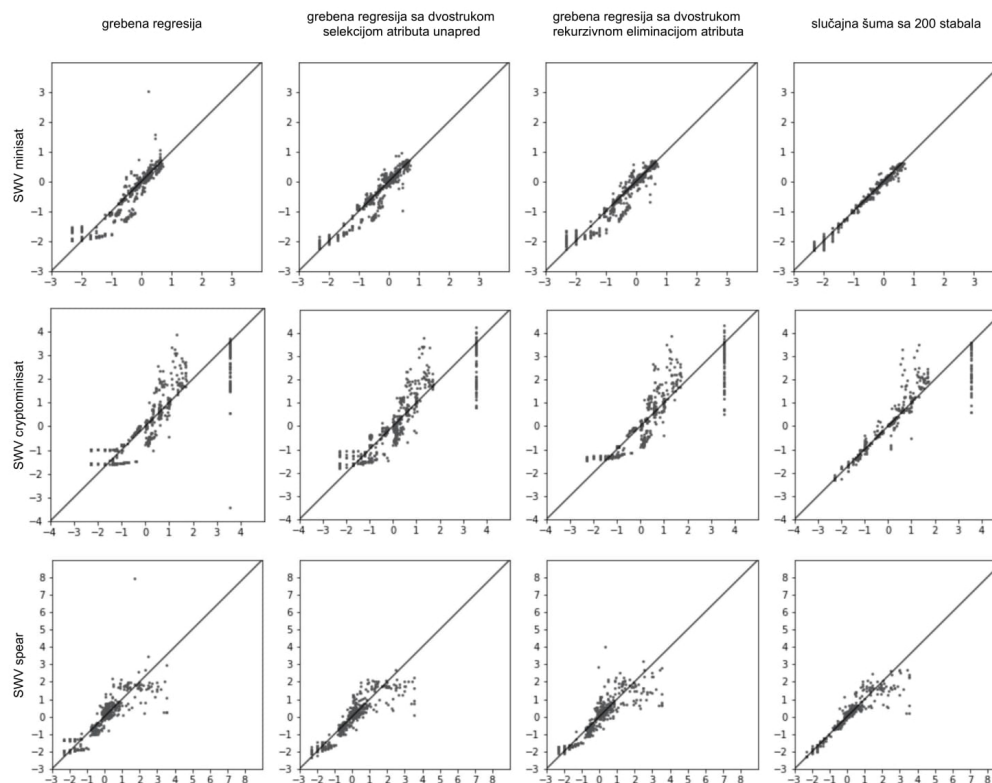
Slika 15: Grafičko poređenje raspodela grešaka predviđanja modela jednostavne grebene regresije na originalnom skupu atributa (bez interakcija), modela grebene regresije sa dvostrukom selekcijom atributa unapred ($l = 30$, q i λ su metaparametri), modela grebene regresije sa dvostrukom rekurzivnom eliminacijom atributa ($l = 30$, $q = 60$, λ je metaparametar) i modela slučajne šume sa 200 stabala. Iznad grafika je naznačen odgovarajući model, a na levoj strani je naznačen rešavač. Na graficima su prikazane sve instance iz skupa INDU (svaka tačka odgovara jednoj instanci). Na x-osi predstavljeni su logaritmi stvarnih vremena izvršavanja, a na y-osi su predviđeni logaritmi vremena izvršavanja dobijeni 10-strukom unakrsnom validacijom.



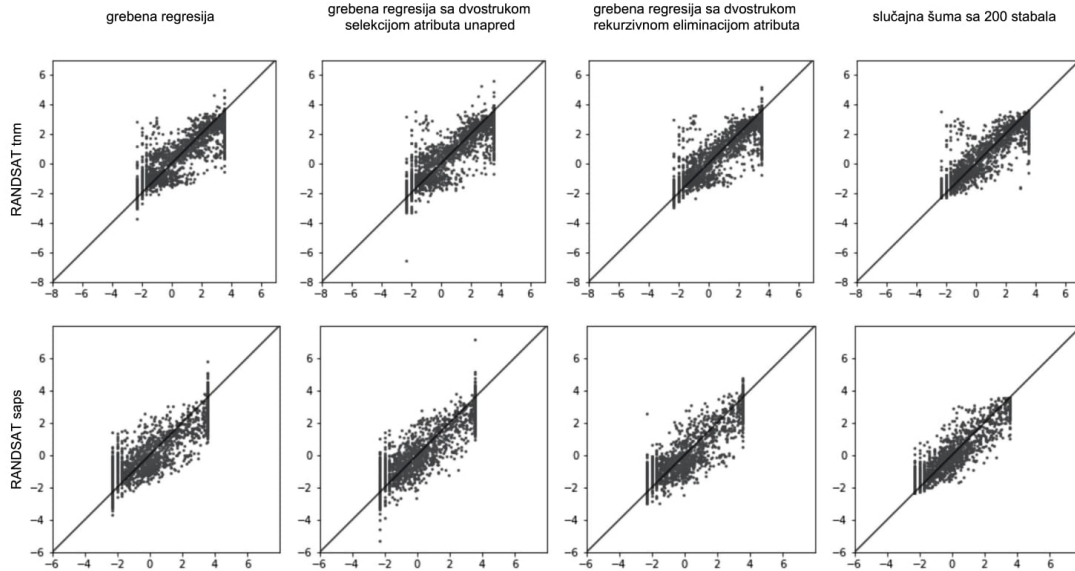
Slika 16: Grafičko poređenje raspodela grešaka predviđanja modela jednostavne grebene regresije na originalnom skupu atributa (bez interakcija), modela grebene regresije sa dvostrukom selekcijom atributa unapred ($l = 30$, q i λ su metaparametri), modela grebene regresije sa dvostrukom rekurzivnom eliminacijom atributa ($l = 30$, $q = 60$, λ je metaparametar) i modela slučajne šume sa 200 stabala. Iznad grafika je naznačen odgovarajući model, a na levoj strani je naznačen rešavač. Na graficima su prikazane sve instance iz skupa SWV-IBM (svaka tačka odgovara jednoj instanci). Na x-osi predstavljeni su logaritmi stvarnih vremena izvršavanja, a na y-osi su predviđeni logaritmi vremena izvršavanja dobijeni 10-strukom unakrsnom validacijom.



Slika 17: Grafičko poređenje raspodela grešaka predviđanja modela jednostavne grebene regresije na originalnom skupu atributa (bez interakcija), modela grebene regresije sa dvostrukom selekcijom atributa unapred ($l = 30$, q i λ su metaparametri), modela grebene regresije sa dvostrukom rekurzivnom eliminacijom atributa ($l = 30$, $q = 60$, λ je metaparametar) i modela slučajne šume sa 200 stabala. Iznad grafika je naznačen odgovarajući model, a na levoj strani je naznačen rešavač. Na graficima su prikazane sve instance iz skupa IBM (svaka tačka odgovara jednoj instanci). Na x-osi predstavljeni su logaritmi stvarnih vremena izvršavanja, a na y-osi su predviđeni logaritmi vremena izvršavanja dobijeni 10-strukom unakrsnom validacijom.



Slika 18: Grafičko poređenje raspodela grešaka predviđanja modela jednostavne grebene regresije na originalnom skupu atributa (bez interakcija), modela grebene regresije sa dvostrukom selekcijom atributa unapred ($l = 30$, q i λ su metaparametri), modela grebene regresije sa dvostrukom rekurzivnom eliminacijom atributa ($l = 30$, $q = 60$, λ je metaparametar) i modela slučajne šume sa 200 stabala. Iznad grafika je naznačen odgovarajući model, a na levoj strani je naznačen rešavač. Na graficima su prikazane sve instance iz skupa SWV (svaka tačka odgovara jednoj instanci). Na x-osi predstavljene su logaritmi stvarnih vremena izvršavanja, a na y-osi su predviđeni logaritmi vremena izvršavanja dobijeni 10-strukom unakrsnom validacijom.



Slika 19: Grafičko poređenje raspodela grešaka predviđanja modela jednostavne grebene regresije na originalnom skupu atributa (bez interakcija), modela grebene regresije sa dvostrukom selekcijom atributa unapred ($l = 30$, q i λ su metaparametri), modela grebene regresije sa dvostrukom rekurzivnom eliminacijom atributa ($l = 30$, $q = 60$, λ je metaparametar) i modela slučajne šume sa 200 stabala. Iznad grafika je naznačen odgovarajući model, a na levoj strani je naznačen rešavač. Na graficima su prikazane sve instance iz skupa RANDSAT (svaka tačka odgovara jednoj instanci). Na x-osi predstavljeni su logaritmi stvarnih vremena izvršavanja, a na y-osi su predviđeni logaritmi vremena izvršavanja dobijeni 10-strukom unakrsnom validacijom.

A.3 Gausova uslovna slučajna polja

Na slici 20 prikazane su ocene strukturiranih modela Gausovih uslovnih slučajnih polja i nestrukturiranih modela slučajne šume, izračunate na svim rešavačima odjednom. Na slikama 21, 22, 23, 24, 25, 26, 27 i 28 prikazane su uporedne vrednosti mera kvaliteta za ocenjene modele Gausovih uslovnih slučajnih polja i modele slučajne šume, izračunate na pojedinačnim rešavačima. Modeli Gausovih uslovnih slučajnih polja sa direktnom vektorskom reprezentacijom rešavača označeni su sa **GCRF1**, modeli koji koriste singularnu dekompoziciju za vektorsku reprezentaciju rešavača označeni su

sa **GCRF2**, dok su modeli slučajne šume označeni sa **RF**. Debljim fontom označene su ocene modela Gausovih uslovnih slučajnih polja koje su bolje od odgovarajućih ocena modela slučajne šume. Zanimljivo je primetiti da nam slika 20 pokazuje da je GCRF1 u opštem slučaju (na svim rešavačima) uglavnom bolje ocenjen od GCRF2, a sa druge strane, ako ocenjujemo po rešavaču i poredimo sa modelima slučajne šume, GCRF2 modeli uvek daju slabija predviđanja od slučajnih šuma, dok su GCRF1 modeli u nekoliko slučajeva zanemarljivo bolje ocenjeni. Ipak, u većini slučajeva modeli slučajne šume bolje predviđaju i od GCRF1 modela. Slika 29 predstavlja grafički prikaz raspodela greške predviđanja za strukturirane GCRF modele i nestrukturirane modele slučajne šume.

skup instanci	model	RMSE	R2
SATALL12S	GCRF1	0.686863	0.734391
	GCRF2	0.701571	0.722894
	RF	0.686841	0.734408
SATHAND12S	GCRF1	0.635939	0.790754
	GCRF2	0.626386	0.796993
	RF	0.610744	0.807005
SATINDU12S	GCRF1	0.54889	0.731491
	GCRF2	0.527889	0.751644
	RF	0.514527	0.764058
SATRAN12S	GCRF1	0.609328	0.774146
	GCRF2	0.608054	0.77509
	RF	0.603225	0.778647

Slika 20: Rezultati evaluacije ugnježdom 5-strukom unakrsnom validacijom strukturiranih GCRF modela i nestrukturiranih modela slučajne šume na svim rešavačima

rešavač	model	skup instanci			
		SATALL12S		SATHAND12S	
		RMSE	R2	RMSE	R2
ebglucose	GCRF1	0.567945	0.745472	0.621123	0.717317
	GCRF2	0.576809	0.737464	0.599767	0.736422
	RF	0.567946	0.745471	0.598724	0.737338
ebminisat	GCRF1	0.649128	0.749333	0.662235	0.796827
	GCRF2	0.655203	0.744619	0.632093	0.814901
	RF	0.649129	0.749332	0.627391	0.817645
glucose2	GCRF1	0.569675	0.758571	0.648389	0.719363
	GCRF2	0.580801	0.749048	0.625394	0.738915
	RF	0.569675	0.758571	0.622642	0.741208
glueminisat	GCRF1	0.602125	0.782263	0.63845	0.806536
	GCRF2	0.612909	0.774394	0.606196	0.82559
	RF	0.602125	0.782263	0.599437	0.829457
lingeling	GCRF1	0.54132	0.775915	0.614561	0.780283
	GCRF2	0.550304	0.768415	0.57769	0.805856
	RF	0.54132	0.775915	0.573137	0.808905
lrglshr	GCRF1	0.617034	0.767224	0.665349	0.787358
	GCRF2	0.625898	0.760488	0.607655	0.822637
	RF	0.617034	0.767224	0.598786	0.827777
minisatpsm	GCRF1	0.575625	0.763355	0.576554	0.794513
	GCRF2	0.583434	0.756891	0.551912	0.811703
	RF	0.575626	0.763354	0.547965	0.814387
mphaseSAT64	GCRF1	0.77473	0.665049	0.672284	0.803606
	GCRF2	0.78809	0.653397	0.689625	0.793344
	RF	0.77473	0.665049	0.660228	0.810587

Slika 21: Usporedni rezultati evaluacije ugneždenom 5-strukom unakrsnom validacijom modela GCRF i slučajnih šuma za rešavače *ebglucose*, *ebminisat*, *glucosat2*, *glueminisat*, *lingeling*, *lrglshr*, *minisatpsm* i *mphaseSAT64* na skupovima instanci *SATALL12S* i *SATHAND12S*

rešavač	model	skup instanci			
		SATALL12S		SATHAND12S	
		RMSE	R2	RMSE	R2
precosat	GCRF1	0.580584	0.787833	0.6772	0.787998
	GCRF2	0.590796	0.780304	0.625754	0.818985
	RF	0.580584	0.787833	0.613879	0.82579
qutersat	GCRF1	0.538163	0.752834	0.587021	0.768784
	GCRF2	0.545406	0.746135	0.552627	0.795084
	RF	0.538163	0.752834	0.546809	0.799376
rcl	GCRF1	0.578517	0.750199	0.573908	0.788203
	GCRF2	0.545406	0.746135	0.552627	0.795084
	RF	0.578519	0.750197	0.542461	0.810777
restartsat	GCRF1	0.596603	0.777431	0.592063	0.826024
	GCRF2	0.604516	0.771488	0.56295	0.842713
	RF	0.596605	0.77743	0.556444	0.846328
cryptominisat2011	GCRF1	0.566424	0.764112	0.632555	0.786608
	GCRF2	0.574239	0.757558	0.603705	0.805629
	RF	0.566424	0.764112	0.597786	0.809422
spear-sw	GCRF1	0.571205	0.758122	0.560746	0.828247
	GCRF2	0.577444	0.752809	0.525905	0.848927
	RF	0.571205	0.758122	0.522779	0.850718
spear-hw	GCRF1	0.593004	0.740148	0.571722	0.823583
	GCRF2	0.596363	0.737197	0.51212	0.858449
	RF	0.593004	0.740148	0.506277	0.86166
eagleup	GCRF1	0.852927	0.708021	0.532542	0.773117
	GCRF2	0.875073	0.692662	0.544075	0.763183
	RF	0.852927	0.708021	0.53505	0.770974

Slika 22: Uporedni rezultati evaluacije ugneždenom 5-strukom unakrsnom validacijom modela GCRF i slučajnih šuma za rešavače *precosat*, *qutersat*, *rcl*, *restartsat*, *cryptominisat2011*, *spear-sw*, *spear-hw* i *eagleup* na skupovima instanci *SATALL12S* i *SATHAND12S*

rešavač	model	skup instanci			
		SATALL12S		SATHAND12S	
		RMSE	R2	RMSE	R2
sparrow	GCRF1	0.930107	0.693238	0.710702	0.754979
	GCRF2	0.954245	0.67711	0.725461	0.744696
	RF	0.930107	0.693238	0.708757	0.756318
marchrw	GCRF1	0.538133	0.825835	0.512006	0.877787
	GCRF2	0.554274	0.81523	0.542239	0.862927
	RF	0.538133	0.825835	0.511584	0.877988
mphaseSATm	GCRF1	0.735459	0.70891	0.680853	0.819044
	GCRF2	0.78188	0.671003	0.758675	0.775312
	RF	0.735099	0.709194	0.680253	0.819362
satime11	GCRF1	0.919693	0.673286	0.713815	0.736254
	GCRF2	0.939577	0.659006	0.735265	0.720164
	RF	0.919695	0.673284	0.719053	0.732369
tnm	GCRF1	0.894947	0.694754	0.709228	0.726771
	GCRF2	0.919784	0.677577	0.731356	0.709455
	RF	0.894947	0.694754	0.714951	0.722343
mxc09	GCRF1	0.622048	0.746871	0.627638	0.794607
	GCRF2	0.627228	0.742637	0.596379	0.814557
	RF	0.622048	0.746871	0.591219	0.817751
gnoveltyp2	GCRF1	0.929814	0.682498	0.684398	0.760343
	GCRF2	0.950762	0.668031	0.687446	0.758204
	RF	0.929814	0.682498	0.671029	0.769614
sattime	GCRF1	0.910138	0.676804	0.738969	0.727428
	GCRF2	0.930106	0.662467	0.76269	0.709648
	RF	0.910139	0.676803	0.748582	0.72029

Slika 23: Usporedni rezultati evaluacije ugneždenom 5-strukom unakrsnom validacijom modela GCRF i slučajnih šuma za rešavače *sparrow*, *marchrw*, *mphaseSATm*, *satime11*, *tnm*, *mxc09*, *gnoveltyp2* i *sattime* na skupovima instanci *SATALL12S* i *SATHAND12S*

rešavač	model	skup instanci			
		SATALL12S		SATHAND12S	
		RMSE	R2	RMSE	R2
sattimep	GCRF1	0.866748	0.643153	0.692195	0.743362
	GCRF2	0.881106	0.631233	0.709315	0.73051
	RF	0.866748	0.643153	0.695277	0.741072
clasp2	GCRF1	0.609682	0.754052	0.609522	0.806699
	GCRF2	0.617051	0.748071	0.570562	0.83062
	RF	0.609682	0.754052	0.560728	0.836408
clasp1	GCRF1	0.580289	0.769768	0.610519	0.798042
	GCRF2	0.589739	0.762208	0.565517	0.826718
	RF	0.580289	0.769768	0.554933	0.833144
picosat	GCRF1	0.652926	0.751407	0.646057	0.813673
	GCRF2	0.65811	0.747444	0.596679	0.841066
	RF	0.652926	0.751407	0.588194	0.845554
mphaseSAT	GCRF1	0.766967	0.689248	0.679362	0.819481
	GCRF2	0.813541	0.650362	0.757165	0.775766
	RF	0.766693	0.689471	0.677934	0.820239
sapperlot	GCRF1	0.579938	0.771467	0.610251	0.813124
	GCRF2	0.584886	0.767551	0.568302	0.837933
	RF	0.579938	0.771467	0.56454	0.840072
sol	GCRF1	0.559063	0.710114	0.587579	0.762398
	GCRF2	0.570891	0.697719	0.607244	0.746228
	RF	0.559063	0.710114	0.585454	0.764113

Slika 24: Uporedni rezultati evaluacije ugnežđenom 5-strukom unakrsnom validacijom modela GCRF i slučajnih šuma za rešavače *sattimep*, *clasp2*, *clasp1*, *picosat*, *mphaseSAT*, *sapperlot* i *sol* na skupovima instanci *SATALL12S* i *SATHAND12S*

rešavač	model	skup instanci			
		SATINDU12S		SATRAND12S	
		RMSE	R2	RMSE	R2
ebglucose	GCRF1	0.598432	0.70614	0.250269	0.40603
	GCRF2	0.565017	0.738041	0.264281	0.337655
	RF	0.554376	0.747815	0.253099	0.39252
ebminisat	GCRF1	0.633531	0.704571	0.322749	0.550667
	GCRF2	0.611493	0.724767	0.305492	0.597433
	RF	0.600787	0.73432	0.303914	0.601581
glucose2	GCRF1	0.608367	0.700809	0.270001	0.496398
	GCRF2	0.558013	0.748287	0.274064	0.481126
	RF	0.544931	0.759951	0.27243	0.487293
glueminisat	GCRF1	0.628852	0.710521	0.251807	-0.034844
	GCRF2	0.593254	0.742367	0.235408	0.095557
	RF	0.580234	0.753551	0.200108	0.346468
lingeling	GCRF1	0.534434	0.703743	0.269977	0.455541
	GCRF2	0.511022	0.72913	0.280984	0.410243
	RF	0.502608	0.737977	0.279758	0.415377
lrglshr	GCRF1	0.638876	0.707668	0.285805	0.482417
	GCRF2	0.594979	0.74646	0.28888	0.47122
	RF	0.5829	0.75665	0.2874	0.476622
minisatpsm	GCRF1	0.610223	0.693707	0.286279	0.175101
	GCRF2	0.57505	0.727999	0.261667	0.310844
	RF	0.563447	0.738865	0.254686	0.347125
mphaseSAT64	GCRF1	0.560646	0.677027	0.846189	0.629321
	GCRF2	0.532447	0.7087	0.860513	0.616666
	RF	0.524428	0.717408	0.849344	0.626552

Slika 25: Uporedni rezultati evaluacije ugneždenom 5-strukom unakrsnom validacijom modela GCRF i slučajnih šuma za rešavače *ebglucose*, *ebminisat*, *glucosat2*, *glueminisat*, *lingeling*, *lrglshr*, *minisatpsm* i *mphaseSAT64* na skupovima instanci *SATINDU12S* i *SATRAND12S*

rešavač	model	skup instanci			
		SATINDU12S		SATRAND12S	
		RMSE	R2	RMSE	R2
precosat	GCRF1	0.574373	0.714202	0.285763	0.487783
	GCRF2	0.554949	0.733206	0.299196	0.438495
	RF	0.544899	0.742781	0.298785	0.440037
qutersat	GCRF1	0.562681	0.660303	0.254111	0.423403
	GCRF2	0.525003	0.704273	0.258856	0.401665
	RF	0.515785	0.714567	0.257438	0.408206
rcl	GCRF1	0.598179	0.704613	0.288931	0.20053
	GCRF2	0.577539	0.724645	0.259753	0.353849
	RF	0.568248	0.733433	0.254834	0.378091
restartsat	GCRF1	0.614406	0.720034	0.256041	0.598501
	GCRF2	0.590903	0.741043	0.254647	0.602861
	RF	0.579716	0.750755	0.25205	0.61092
cryptominisat2011	GCRF1	0.542903	0.704569	0.265136	0.361352
	GCRF2	0.513371	0.735836	0.263769	0.367922
	RF	0.505061	0.744319	0.262602	0.373501
spear-sw	GCRF1	0.588821	0.693215	0.260349	-1.319688
	GCRF2	0.561204	0.721319	0.188555	-0.216731
	RF	0.553115	0.729294	0.169853	0.012659
spear-hw	GCRF1	0.615458	0.642321	0.281346	0.572177
	GCRF2	0.594327	0.66646	0.275222	0.5906
	RF	0.588918	0.672504	0.272833	0.597676
eagleup	GCRF1	0.318863	0.183623	1.050783	0.689608
	GCRF2	0.283356	0.355317	1.056455	0.686248
	RF	0.264521	0.438173	1.050784	0.689607

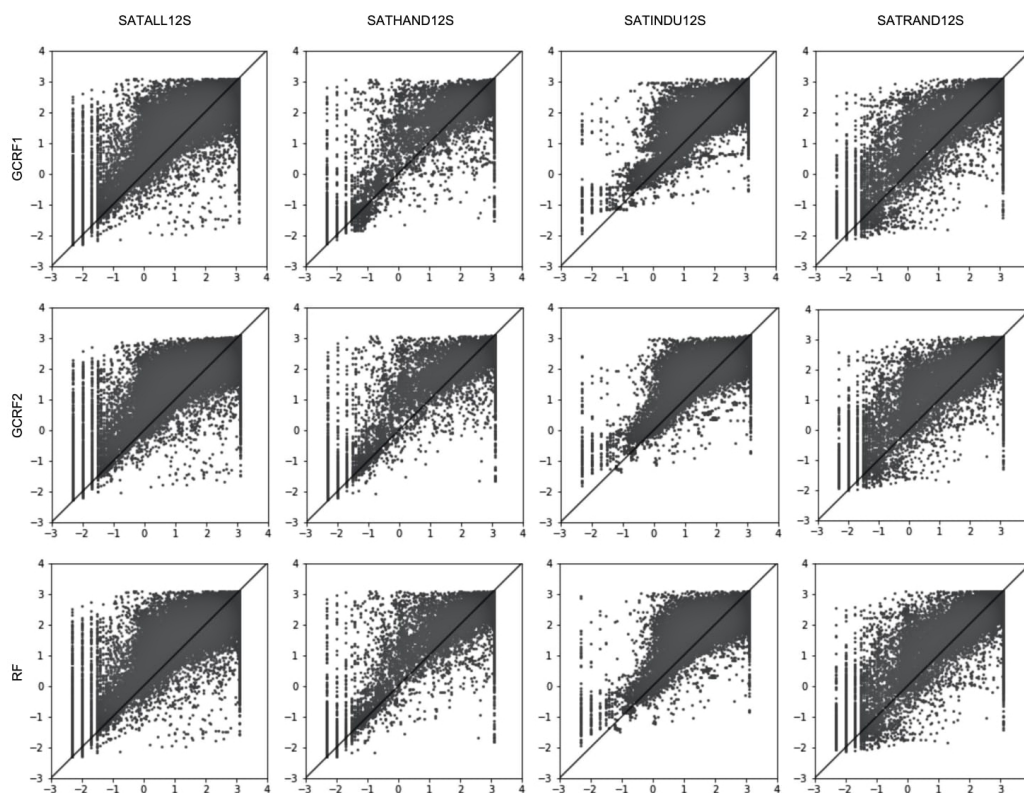
Slika 26: Uporedni rezultati evaluacije ugnežđenom 5-strukom unakrsnom validacijom modela GCRF i slučajnih šuma za rešavače *precosat*, *qutersat*, *rcl*, *restartsat*, *cryptominisat2011*, *spear-sw*, *spear-hw* i *eagleup* na skupovima instanci *SATINDU12S* i *SATRAND12S*

rešavač	model	skup instanci			
		SATINDU12S		SATRAND12S	
		RMSE	R2	RMSE	R2
sparrow	GCRF1	0.363943	0.178581	1.097717	0.641222
	GCRF2	0.377277	0.117291	1.103931	0.637149
	RF	0.348589	0.246428	1.098211	0.640899
marchrw	GCRF1	0.438914	0.505922	0.482983	0.803948
	GCRF2	0.414414	0.559542	0.484247	0.802921
	RF	0.411169	0.566413	0.482716	0.804165
mphaseSATm	GCRF1	0.587499	0.678981	0.808026	0.6549
	GCRF2	0.575206	0.692274	0.822655	0.642291
	RF	0.555842	0.712644	0.808083	0.654851
satime11	GCRF1	0.409843	0.628441	1.02839	0.687998
	GCRF2	0.420241	0.609349	1.035615	0.683599
	RF	0.368025	0.700396	1.031101	0.686351
tnm	GCRF1	0.379712	0.335186	1.059927	0.676951
	GCRF2	0.359792	0.403108	1.066381	0.673005
	RF	0.352363	0.427504	1.061094	0.676239
mxc09	GCRF1	0.629957	0.70166	0.317734	0.581128
	GCRF2	0.606705	0.723277	0.288759	0.654039
	RF	0.596828	0.732214	0.287934	0.656014
gnoveltyp2	GCRF1	0.366151	-0.013909	1.06037	0.68999
	GCRF2	0.376062	-0.06954	1.066132	0.686612
	RF	0.358008	0.03069	1.06144	0.689364
sattime	GCRF1	0.409464	0.627267	0.994585	0.714271
	GCRF2	0.430618	0.587759	1.000588	0.710812
	RF	0.373611	0.689682	0.996662	0.713077

Slika 27: Uporedni rezultati evaluacije ugneždenom 5-strukom unakrsnom validacijom modela GCRF i slučajnih šuma za rešavače *sparrow*, *marchrw*, *mphaseSATm*, *satime11*, *tnm*, *mxc09*, *gnoveltyp2* i *sattime* na skupovima instanci *SATINDU12S* i *SATRAND12S*

rešavač	model	skup instanci			
		SATINDU12S		SATRAND12S	
		RMSE	R2	RMSE	R2
sattimep	GCRF1	0.38551	0.642442	0.938802	0.690827
	GCRF2	0.38429	0.644703	0.941492	0.689053
	RF	0.354092	0.698347	0.93888	0.690776
clasp2	GCRF1	0.639838	0.680068	0.395754	0.53044
	GCRF2	0.616549	0.702934	0.324865	0.683592
	RF	0.608592	0.710552	0.323578	0.686096
clasp1	GCRF1	0.601071	0.71142	0.382739	0.544761
	GCRF2	0.575819	0.735159	0.314715	0.6922
	RF	0.567335	0.742906	0.313411	0.694744
picosat	GCRF1	0.664866	0.677753	0.370204	0.549487
	GCRF2	0.634831	0.70621	0.32635	0.6499
	RF1	0.626323	0.714031	0.325911	0.65084
mphaseSAT	GCRF1	0.577753	0.69026	0.807293	0.654471
	GCRF2	0.571821	0.696588	0.822203	0.64159
	RF	0.548711	0.720617	0.807144	0.654599
sapperlot	GCRF1	0.568806	0.717414	0.283359	0.599479
	GCRF2	0.556903	0.729117	0.275255	0.62206
	RF	0.548758	0.736983	0.274576	0.623921
sol	GCRF1	0.480385	0.697566	0.311643	0.278457
	GCRF2	0.478629	0.699773	0.317298	0.252035
	RF	0.474484	0.70495	0.317035	0.253273

Slika 28: Uporedni rezultati evaluacije ugnežđenom 5-strukom unakrsnom validacijom modela GCRF i slučajnih šuma za rešavače *sattimep*, *clasp2*, *clasp1*, *picosat*, *mphaseSAT*, *sapperlot* i *sol* na skupovima instanci *SATINDU12S* i *SATRAND12S*



Slika 29: Grafički prikaz raspodela grešaka predviđanja GCRF modela kod koga se rešavači predstavljaju vektorima vrednosti ciljne promenljive (GCRF1), GCRF modela kod koga se vektorske reprezentacije rešavača računaju korišćenjem singularne dekompozicije (GCRF2) i nestrukturiranog modela slučajne šume (RF). Iznad grafika je naznačeni su odgovarajući skupovi podataka. Na graficima su prikazane sve instance iz odgovarajućih skupova i svi rešavači (svaka tačka odgovara jednom paru instance i rešavača). Na x-osi predstavljeni su logaritmi stvarnih vremena izvršavanja, a na y-osi su predviđanja logaritma vremena izvršavanja dobijena 5-strukom unakrsnom validacijom.

Literatura

- [1] Frank Hutter, Lin Xu, Holger H. Hoos, Kevin Leyton-Brown (2013): *Algorithm Runtime Prediction: Methods & Evaluation*, Department of Computer Science, University of British Columbia
- [2] C. M. Bishop (2006): *Pattern recognition and machine learning*, Springer
- [3] Mladen Nikolić, Anđelka Zečević: *Mašinsko učenje*, Matematički fakultet, Univerzitet u Beogradu
- [4] Predrag Janičić (2008): *Matematička logika u računarstvu*, Matematički fakultet, Univerzitet u Beogradu
- [5] S. A. Cook (1971): *The complexity of theorem-proving procedures*, Proceedings of the third annual ACM symposium on theory of computing
- [6] L. Levin (1973): *Universal search problems*
- [7] G. S. Tseitin (1968): *On the complexity of derivation in propositional calculus*, Studies in Constructive Mathematics and Mathematical Logic
- [8] G. Logemann, M. Davis, D. Loveland (1962): *A Machine Program for Theoremproving*, Communications of the ACM
- [9] M. Davis, H. Putnam (1960): *A Computing Procedure for Quantification Theory*, Journal of the ACM
- [10] J.P. Marques-Silva, Karem A. Sakallah (1996): *GRASP-A New Search Algorithm for Satisfiability*, Digest of IEEE International Conference on Computer-Aided Design (ICCAD)
- [11] J.P. Marques-Silva, Karem A. Sakallah (1999): *GRASP: A Search Algorithm for Propositional Satisfiability*, IEEE Transactions on Computers
- [12] Roberto J. Bayardo Jr, Robert C. Schrag (1997): *Using CSP look-back techniques to solve real world SAT instances*, Proc. 14th Nat. Conf. on Artificial Intelligence (AAAI)
- [13] H. Kautz, B. Selman (1996): *Pushing the envelope: planning, propositional logic, and stochastic search*, Proceedings of the Thirteenth National Conference on Artificial Intelligence
- [14] Malik Ghallab, Dana S. Nau, Paolo Traverso (2004): *Automated Planning: Theory and Practice*, Morgan Kaufmann Publishers

- [15] E. Fink (1998): *How to solve it automatically: Selection among problem-solving methods*, Proceedings of the Fourth International Conference on AI Planning Systems
- [16] A. E. Howe, E. Dahlman, C. Hansen, M. Scheetz, A. Mayrhauser (2000): *Exploiting competitive planner performance*, Recent Advances in AI Planning, Lecture Notes in Computer Science, Springer Berlin Heidelberg
- [17] E. Nudelman, K. Leyton-Brown, H. H. Hoos, A. Devkar, Y. Shoham (2004): *Understanding random SAT: beyond the clauses-to-variables ratio*, Proceedings of the 10th International Conference on Principles and Practice of Constraint Programming
- [18] L. Xu, H. H. Hoos, K. Leyton-Brown (2007): *Hierarchical hardness models for SAT*, Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming
- [19] F. Hutter, Y. Hamadi, H. H. Hoos, K. Leyton-Brown (2006): *Performance prediction and automated tuning of randomized and parametric algorithms*, Proceedings of the 12th International Conference on Principles and Practice of Constraint Programming
- [20] K. Smith-Miles, J. van Hemert (2011): *Discovering the suitability of optimisation algorithms by learning from evolved instances*, Annals of Mathematics and Artificial Intelligence
- [21] T. Bartz-Beielstein, S. Markon (2004): *Tuning search algorithms for real-world applications: a regression tree based approach*, Proceedings of the 2004 Congress on Evolutionary Computation
- [22] E. A. Brewer (1994): *Portable high-performance supercomputing: high-level platform-dependent optimization*, PhD thesis, Massachusetts Institute of Technology
- [23] E. A. Brewer (1995): *High-level optimization via automated statistical modeling*, Proceedings of the 5th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming
- [24] K. Leyton-Brown, E. Nudelman, Y. Shoham (2002): *Learning the empirical hardness of optimization problems: the case of combinatorial auctions*, Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming

- [25] K. Leyton-Brown, E. Nudelman, Y. Shoham (2009): *Empirical hardness models: methodology and a case study on combinatorial auctions*, Journal of the ACM
- [26] E. Nudelman, K. Leyton-Brown, G. Andrew, C. Gomes, J. McFadden, B. Selman, Y. Shoham (2003): *Satzilla 0.9. Solver description*, 2003 SAT Competition
- [27] L. Xu, F. Hutter, H. H. Hoos, K. Leyton-Brown (2007): *SATzilla-07: the design and analysis of an algorithm portfolio for SAT*, Proceedings of the 13th International Conference on Principles and Practice of Constraint Programming
- [28] L. Xu, F. Hutter, H. H. Hoos, K. Leyton-Brown (2008): *SATzilla: portfolio-based algorithm selection for SAT*, Journal of Artificial Intelligence Research
- [29] Mladen Nikolić, Anđelka Zečević: *Naučno izračunavanje*, Matematički fakultet, Univerzitet u Beogradu
- [30] L. Breiman, J. Friedman, R. Olshen, C. Stone (1984): *Classification and Regression Trees*, Wadsworth, Belmont, CA
- [31] Filip Marić, Milan Banković: *Beleške sa predavanja*, kurs Automatsko rezonovanje, Matematički fakultet, Univerzitet u Beogradu
- [32] N. Een, N. Sorensson (2004): *An extensible SAT-solver*, Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT'03)
- [33] M. Soos, (2010): *CryptoMiniSat 2.5.0. Solver description*, SAT Race 2010.
- [34] D. Babić, F. Hutter (2007): *Spear theorem prover*, SAT competition 2007.
- [35] W. Wei, C. M. Li (2009): *Switching between two adaptive noise mechanisms in local search for SAT*, SAT competition 2009.
- [36] F. Hutter, D. A. D. Tompkins, H. H. Hoos (2002): *Scaling and probabilistic smoothing: efficient dynamic local search for SAT*, Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming (CP'02)

- [37] Mladen Nikolić (2018): *Regresiona Gausova uslovna slučajna polja*, predavanje na seminaru
- [38] L. Xu, F. Hutter, H. H. Hoos, K. Leyton-Brown (2012): *Evaluating component solver contributions in portfolio-based algorithm selectors*, Proceedings of the 15th International Conference on Theory and Applications of Satisfiability Testing (SAT'12)