Kvalifikacijski rad

Postupci i tehnike dubinske analize podataka

mr.sc. Filip Ujević
e-mail: filip.ujevic@gmail.com

Sažetak – Postupci dubinske analize podataka omogućuju strojno pronalaženje implicitnih pravilnosti i odnosa koji postoje skriveni u velikim bazama podataka. U ovom radu izložene su osnovne postavke i ideje na kojima se zasnivaju postupci dubinske analize podataka, opisani su temeljni problemi u njihovoj primjeni, kao i načini rješavanja tih problema. Prikazane su najpopularnije tehnike modeliranja pravilnosti u podacima, te su naznačena njihova svojstva i međusobne razlike.

Ključne riječi: dubinska analiza podataka, otkrivanje znanja, strojno učenje, tehnike modeliranja pravilnosti u podacima.

I UVOD

U literaturi postoji više različitih definicija dubinske analize podataka (engl. Data Mining). Iako postoje neke terminološke razlike među njima, smisao svih definicija je isti. Dvije najkorištenije su:

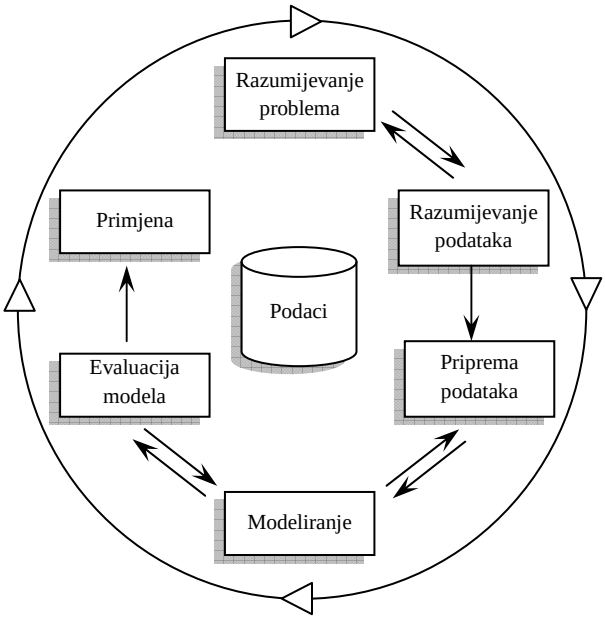
– Dubinska analiza podataka je ekstrakcija implicitnih, dotad nepoznatih i potencijalno korisnih informacija iz glomaznih baza podataka [1].

– Dubinska analiza podataka je potraga za globalnim pravilnostima i odnosima koje postoje u velikim bazama podataka, ali su skriveni u ogromnom mnoštvu podataka. Ti odnosi predstavljaju vrijedno znanje o objektima opisanim bazom podataka, ali i o objektima stvarnog svijeta ukoliko baza podataka vjerno odražava stvarnost [2].

Iako su mnoge tehnike dubinske analize podataka poznate već desetljećima, ova disciplina posaban zamah dobiva tek u nekoliko zadnjih godina, i to pod popularnim (iako nepreciznim i širokim) nazivom *big data* (Gartner: Nexus of Forces [3], IDC: Third Plarform [4]). U znanstvenoj literaturi se osim termina dubinska analiza podataka koristi i termin otkrivanje znanja iz baza podataka (engl. Knowledge Discovery in Databases).

II PROCES DUBINSKE ANALIZE PODATAKA

Srž procesa dubinske analize podataka čine tehnike modeliranja podataka, koje su zasnovane prvenstveno na postupcima strojnog učenja. No, dubinska analiza podataka nije puka primjena sofisticiranih postupaka strojnog učenja na podatke smještene u velikim bazama podataka. Upravo velike baze podataka kao osnovni izvor podataka donose niz problema u primjeni postupaka strojnog učenja, koje je nužno adresirati (obujam podataka, nepotpuni podaci, raspršenost primjera, šum u podacima,...). Stoga proces dubinske analize podataka osim modeliranja obuhvaća i druge faze, jednako važne za uspjeh procesa i kvalitetu rezultata. Slika 1 prikazuje osnovne faze standardnog procesa dubinske analize podataka, njihov redoslijed i međuovisnosti.



Slika 1: Osnovne faze procesa dubinske analize podataka prema CRISP-DM standardu

Ovakva struktura procesa dubinske analize podataka definirana je na osnovu iskustava u praktičnoj primjeni metodologije u vodećim svjetskim tvrtkama, a naziva se CRISP-DM (prema engl. *CRoss Industry Standard Process for Data Mining*) [5].

Središnji i algoritamski najzahtjevniji dio procesa dubinske analize podataka čini modeliranje podataka u širem smislu, a obuhvaća faze pripreme podataka, modeliranja i evaluacije izvedenih modela. Ove tri faze imaju presudan značaj za tijek cijelog procesa, pa se često modeliranje podataka u širem smislu poistovjećuje sa dubinskom analizom podataka.

II. 1 Razumijevanje problema

U fazi razumijevanja problema, problem je potrebno sagledati kako sa stanovišta domene problema, tako i s aspekta primjene tehnika dubinske analize podataka. U obje sfere nastoje se pronaći i razumjeti ograničenja i ostali važni čimbenici koji mogu utjecati na proces i konačni rezultat dubinske

analize podataka. U ovoj fazi definiraju se konačni ciljevi u rješavanju problema, te određuju kriteriji uspješnosti procesa dubinske analize podataka.
Ciljevi i kriteriji uspješnosti specificirani iz perspektive domene promatranog problema razlikuju se od specifikacije prevedene u terminologiju dubinske analize podataka. Primjerice, u domeni poslovnog problema cilj može biti povećanje odziva kupaca u marketinškoj kampanji novog proizvoda, a kriterij kvantitativno izražen kao % očekivanog povećanja. U formulaciji dubinske analize podataka, traži se opis skupine kupaca (u terminima poznatih svojstava) za koje postoji visoka vjerojatnost kupnje spomenutog proizvoda, s definiranim minimalnim nivoom prediktivne točnosti rješenja.

Nakon definicije ciljeva i kriterija uspješnosti, izrađuje se okvirni plan provedbe procesa dubinske analize podataka. Osim pojedinih aktivnosti i njihovih rezultata, plan definira i željeni oblik konačnog rezultata cijelog procesa. Budući da se on ne ovisi samo o tipu problema već i o tehnikama koje se koriste, dobro je najaviti i očekivane tehnike koje se namjeravaju upotrijebiti. Na kraju, plan treba sadržavati procjenu trajanja i koštanja postupka.

II. 2 Razumijevanje podataka

Fokus ove faze je osnovni resurs cijelog procesa – podaci. Faza razumijevanje podataka sadrži tri koraka:

1. Prikupljanje podataka

Nakon utvrđivanja opsega dostupnih podataka i njihove lokacije, odabiru se potrebne za njihovo prikupljanje. Rezultat prikupljanja je lista prikupljenih podataka, te sami podaci na odgovarajući način pohranjeni za daljnje korištenje.

2. Opis osnovnih karakteristika podataka

Svrha ove aktivnosti je istraživanje osnovnih karakteristika i strukture podataka kako bi se stekao dojam o informativnosti podataka u smislu rješavanja postavljenog problema. U tu svrhu često se koriste jednostavne statističke tehnike (srednja vrijednost, standardna devijacija, te ekstremi za numeričke atribute; tablice frekvencija vrijednosti za nominalne atribute, i sl.). Postoje i sofisticiranije statističke metode koje daju kvalitetniju informaciju o važnosti pojedinog atributa za rješavanje konkretnog problema, a uglavnom se zasnivaju na analizi korelacije među različitim atributima.

3. Verifikacija kvalitete podataka

Ovaj korak uključuje provjeru potpunosti i ispravnosti podataka, te uz to vezana poboljšanja i ispravke. Potpunost se uglavnom odnosi na neodređene vrijednosti atributa, a ispravnost na otkrivanje različitih anomalija u podacima. Preciznije, provode se slijedeće aktivnosti:

- provjera konzistentnosti podataka s obzirom na vrijednosti i tip atributa,
- određivanje količine i distribucije primjera s neodređenim vrijednostima pojedinih atributa,
- otkrivanje neočekivanih primjera (*engl. outliers*), koji se svrstavaju u dvije kategorije: šum odnosno pogreška u podacima, te primjeri koji predstavljaju novi fenomen u odnosu na standardnu populaciju u dostupnim podacima.

Neke od tehnika modeliranja mogu biti prilično osjetljive na pojavu neodređenih vrijednosti ili neočekivanih primjera u podacima, pa je u tom slučaju nužno podatke na adekvatan način pripremiti prije faze modeliranja.

II. 3 Priprema podataka

Faza pripreme podataka sastoji se od četiri koraka:

1. Odabir podataka

Na osnovu kriterija kvalitete i tehničkih ograničenja, iz skupa svih prikupljenih podataka odabire se podskup koji će služiti kao osnova za daljnje analize. Odabir se može vršiti na razini primjera i na razini atributa.

Kod odabira podataka na razini primjera, kriterij kvalitete nalaže da se odabiru kompletni i ispravni primjeri. Ukoliko zbog ciljane tehnike modeliranja postoje tehnička ograničenja na količinu podataka, uporabom statističkih tehnika provodi se uzorkovanje primjera uz očuvanje reprezentativnosti podataka.

Odabirom podataka na razini atributa moguće je isključiti atribute s nekvalitetnim vrijednostima (šum u podacima, neodređene vrijednosti i sl.), te reducirati broj atributa kojima su primjeri opisani izbacivanjem atributa sa slabom prediktivnosti klase, visokim stupnjem redundancije s drugom atributom, i sl. Postoje različite tehnike redukcije broja atributa, od spajanja više atributa uz određenu transformaciju, do statistički utemeljenih tehnika (npr. ocjena korelacije atributa, metoda analize osnovnih komponenti).

2. Pročišćavanje podataka

Pročišćavanje podataka je komplementarno odabiru podataka, a također nastoji poboljšati kvalitetu podataka za iduću fazu modeliranja podataka. Ovaj korak može biti vremenski prilično zahtjevan, obzirom da postoji mnoštvo tehnika koje je moguće primijeniti. U najčešće korištene tehnike pročišćavanja podataka spadaju:

- *normaliziranje podataka* (npr. svođenje vrijednosti numeričkih atributa na interval [0,1])
- *zaglađivanje podataka* (npr. diskretizacija numeričkih atributa).
- *tretman neodređenih vrijednosti* (npr. zamjena neodređenih vrijednosti statistički opravdanim vrijednostima)

3. Formiranje novih podataka

Načini formiranja novih podataka uključuju stvaranje novog atributa na osnovu jednog ili više postojećih atributa, stvaranje novih primjera, stvaranje novog atributa agregiranjem informacija iz više primjera i slično, a sve s ciljem povećanja informativnosti podataka.

Iako derivirani podaci u osnovi predstavljaju redundanciju u podacima, sa stanovišta odabrane tehnike modeliranja podataka oni mogu značajno

doprinijeti kvaliteti modela. Primjerice, tehnike modeliranja kod kojih se vrijednosti atributa uspoređuju isključivo s konstantama ne mogu naučiti pojam "uspravan" ili "polegnut" iz atributa *širina* i *visina* geometrijskih likova (ne mogu generirati opis tipa *širina < visina*). Međutim, uvođenjem izvedenog atributa *širina–visina*, jednostavno će generirati opis *širina–visina < 0*.

Zbog toga je odabir načina formiranja novih podataka prvenstveno vezan uz svojstva odabrane tehnike modeliranja podataka.

4. Formatiranje podataka

Formatiranje podataka je završni korak u fazi pripreme podataka. On je tehničke prirode i svodi se na prilagođavanje zapisa pripremljenih podataka uvjetima koje diktira korištena tehnika modeliranja podataka. Međutim, kao što je i vidljivo s dijagrama na slici 1, rezultati faze modeliranja mogu pokazati potrebu za dodatnim zahvatima nad podacima, pa se proces pripreme i modeliranja podataka može iterativno ponavljati.

II. 4 Modeliranje podataka

Modeliranje podataka predstavlja ključnu fazu procesa dubinske analize podataka. Upravo u ovoj fazi se korištenjem postupaka strojnog učenja obavlja istinska analiza podataka i pronalaze skrivene pravilnosti koje u njima postoje. Stoga se prethodne faze mogu smatrati pripremom za ovu najizazovniju fazu procesa. Faza modeliranja podataka se provodi kroz tri međusobno ovisna koraka:

1. Odabir tehnike modeliranja

U ovom koraku se razmatra adekvatnost tehnika modeliranja sugeriranih u fazi razumijevanja problema. Nova saznanja o svojstvima i karakteru podataka koji se analiziraju proizišla iz faza razumijevanja i pripreme podataka mogu dovesti do promjene inicijalnog odabira u korist neke prikladnije tehnike. Reviziji odabira tehnike modeliranja podataka treba pristupiti studiozno, budući da kvaliteta i oblik rezultata cjelokupnog procesa bitno ovise o korištenoj tehnici modeliranja.
Osnovni kriterij odabira treba biti odnos temeljnih karakteristika promatranog problema i raspoloživih podataka prema različitim tehnikama modeliranja i njihovim specifičnim osobinama.

2. Odabir procedure testiranja

Prije konstrukcije modela potrebno je definirati proceduru za testiranje kvalitete generiranih modela. Primjerice, kod klasifikacijskih problema kao mjera kvalitete modela obično se koristi udio pogrešno klasificiranih primjera na testnom uzorku. Da bi ovaj omjer bio realan pokazatelj kvalitete modela, nužno je osigurati da se konstrukcija modela i njegovo testiranje odvijaju na nezavisnim skupovima podataka. Zbog toga je prije konstrukcije modela potrebno definirati testni uzorak, tj. odijeliti podatke za učenje od podataka za testiranje. Upotrebom statističkih tehnika uzorkovanja osigurava se reprezentativnost testnih podataka, kako loša zastupljenost određenog tipa primjera ne bi rezultirala krivom slikom kvalitete modela.

3. Konstrukcija modela

Nakon definiranja testnog uzorka, odabranom tehnikom modeliranja se nad skupom podataka za učenje konstruiraju modeli pravilnosti u podacima. U pravilu se generira veći broj različitih modela, budući da tehnike modeliranja tipično imaju određeni broj parametara koji utječu na postupak stvaranja modela. Tako dobiveni modeli su različitog oblika i kvalitete, pa se izdvajaju oni koji pokazuju bolje rezultate na testnom uzorku podataka. Prema tome, postupak konstrukcije modela je u stvari iterativne prirode, u kojem se variranjem različitih parametara traži njihova optimalna kombinacija koja rezultira kvalitetnim modelima.

II. 5 Evaluacija modela

Konačni model (ili više modela slične kvalitete) potrebno je detaljno interpretirati u smislu njihove kompleksnosti i pouzdanosti rezultata koje proizvode. Ocjena konstruiranih modela provodi se kako s aspekta dubinske analize podataka, tako i s aspekta domene promatranog problema.

Ocjena u domeni dubinske analize podataka provodi se u skladu s prethodno definiranom procedurom testiranja kvalitete modela. Osim ocjene pouzdanosti modela na testnom uzorku, potrebno je ocijeniti njegovu smislenost, te objasniti razloge za konačnu kombinaciju parametara tehnike modeliranja. Ukoliko se ustanove razlozi za dodatnom korekcijom modela, određuje se nova kombinacija parametara tehnike modeliranja s kojom se inicira nova iteracija konstruiranja modela.

Evaluacija modela iz perspektive domene osnovnog problema tipično zahtijeva interpretaciju i ocjenu od strane stručnjaka iz spomenute domene. Procjenjuje se ispravnost i inovativnost modela u odnosu na postojeća saznanja o području, njegova općenitost i primjenjivost, te poboljšanja koja model donosi obzirom na osnovne ciljeve projekta. Nastoje se uočiti bitniji nedostatci modela, te sugerirati način njihovog otklanjanja.

Faza evaluacije modela završava revizijom svih do sada obavljenih faza procesa. Rezultat revizije naglašava eventualne nedostatke u provođenju cijelog procesa analize podataka, moguća poboljšanja, te alternativna rješenja za pojedine faze procesa.
Na osnovu provedene revizije utvrđuje se spremnost prelaska u završnu fazu procesa dubinske analize podataka – primjenu rezultata. Konkretno, utvrđuje se je li potrebno i moguće ponoviti određene faze procesa radi poboljšanja kvalitete modela, odnosno je li potrebno napraviti neke preinake u završnoj fazi primjene modela.

II. 6 Primjena rezultata

Prije same primjene konstruiranih modela u praksi, potrebno je izraditi plan primjene modela u kojem se definira strategija i konkretizira način praktične upotrebe modela. Ovaj korak je posebno važan u

slučaju svakodnevnog korištenja modela u domeni rješavanog problema (npr. detekcija neovlaštenog korištenja kreditnih kartica). U tom slučaju plan primjene modela treba specificirati i način praćenja rezultata i održavanja modela. Povratne informacije o korištenju modela mogu ukazati na njegovu nepravilnu upotrebu ili neplanirane manjkavosti koje tek u upotrebi dolaze do izražaja.

Logičan završetak procesa dubinske analize podataka je završni izvještaj kojim se rezimiraju rezultati projekta. U njemu se objašnjavaju važne pretpostavke koje su prethodile procesu, kao i ograničenja vezana uz problem i dostupne podatke. Izvještaj prikazuje bitna iskustva stečena u procesu analize podataka, te opisuje i objašnjava postignute rezultate.

III TEHNIKE MODELIRANJA PRAVILNOSTI U PODACIMA

Tehnike modeliranja u području dubinske analize podataka porijeklo vuku primarno iz strojnog učenja, ali i iz drugih područja: obrade signala, raspoznavanja uzoraka, statistike, evolucijskog programiranja. Iako međusobno znatno različite, tehnike modeliranja dijele sličnu osnovnu strukturu karakteriziranu trima funkcionalno povezanim komponentama:

1. Reprezentacija modela

Tehnike modeliranja kao rezultat daju funkcionalni (matematički) opis otkrivenih ovisnosti u podacima. Formalno, model sa može prikazati kao funkcija $y = f(x,P)$, gdje x predstavlja ulazne vrijednosti (tj. skup podataka za učenje, podskup *univerzuma* – skupa svih vrijednosti koje primjer za učenje može poprimiti), a P skup parametara koji definiraju specifični oblik modela. Tehnike modeliranja se bitno razlikuju po obliku modela y. Primjerice, kod stabala odlučivanja y predstavlja graf određenih svojstava, a kod indukcije pravila y predstavlja skup pravila određene strukture.

Bitne karakteristike reprezentacije modela su: ograničenja na format ulaznih podataka, razumljivost i ekspresivnost prikaza, sposobnost aproksimacije linearnih odnosno nelinearnih ovisnosti u podacima, konačan oblik rezultata (modela).

2. Funkcija vrednovanja aproksimacije

Uz određeni prikaz modela, interna funkcija vrednovanja predstavlja procjenu kvalitete ponuđenog modela s obzirom na aproksimaciju odnosa među atributima podataka. Ovaj interni kriterij vrednovanja modela ne treba miješati s mjerama i metodama ocjene konačnog modela pravilnosti u podacima. Tipično, funkcija vrednovanja ocjenjuje konstrukciju različitih instanci reprezentacije f, tijekom pretraživanja prostora mogućih rješenja.
Karakteristike funkcije vrednovanja koje bitno određuju konačni model su: osjetljivost i robusnost na dimenzionalnost problema (broj atributa i primjera, te mogući broj instanci f), karakter kriterija (probabilistički, logički). Funkcija vrednovanja aproksimacije bitno se razlikuje od tehnike do tehnike, te je uvjetovana reprezentacijom modela i metodom pretraživanja prostora rješenja.

3. Metoda pretraživanja

Uz zadanu reprezentaciju modela, metoda pretraživanja predstavlja specifičan algoritam koji kontrolira pretraživanje prostora svih mogućih instanci f, koristeći se pritom internom funkcijom vrednovanja aproksimacije. Prema tome, uz specifičnu reprezentaciju modela, tehnike modeliranja funkcioniraju kao optimizacijski algoritmi. Stoga su osnovne karakteristike metoda pretraživanja iste kao i kod optimizacijskih algoritama: temeljni pristup pretraživanju (npr. heuristički, pohlepni), kompleksnost pretraživanja, kontrola procesa pretraživanja (npr. kriteriji zaustavljanja).

U slijedećim poglavljima prikazane su najpoznatije i u praksi najkorištenije tehnike modeliranja za klasifikacijske probleme.

IV STABLA ODLUČIVANJA

IV. 1 Reprezentacija modela

Stablo odlučivanja može se promatrati kao klasifikacijski algoritam izražen u formi povezanog grafa sa strukturom stabla. Unutrašnji čvorovi u stablu odlučivanja označeni su nazivima atributa, a grane koje izlaze iz unutrašnjih čvorova označene su mogućim vrijednostima odgovarajućeg atributa. Listovi stabla odlučivanja označeni su nazivima klasa u koje se primjeri razvrstavaju.
Klasifikacija primjera stablom odlučivanja provodi se slijedeći određeni put od korijena stabla do nekog od listova. Svaki od unutrašnjih čvorova stabla predstavlja test na vrijednost određenog atributa, pa se put gradi dodavanjem one grane koja odgovara vrijednosti atributa u promatranom primjeru. Put završava u nekom od listova, te se primjer klasificira u klasu kojom je list označen.

IV. 2 Postupak pretraživanja

Temeljni algoritam konstrukcije stabala odlučivanja star je nekoliko desetljeća, a razvio ga je J.Ross Quinlan [6]. Osnovna verzija algoritma poznata je pod nazivom *ID3* (od engl. *Induction of Decision Trees*). Kasnije verzije algoritma uklanjale su neka od ograničenja izvornog algoritma, te poboljšavale klasifikacijske performanse. Najpoznatiji i vjerojatno najviše korišten algoritam konstrukcije stabala odlučivanja danas je *C4.5* [7], odnosno njegova komercijalna (ponešto unaprijeđena) verzija *C5.0*.

Osnovni algoritam je rekurzivan i konstruira stablo odlučivanja od korijena prema listovima. U svakom koraku rekurzije generira se podstablo visine 1, uz to sa se koriste samo oni primjeri koji pripadaju tom podstablu. Konstrukcija podstabla se zasniva na odabiru jednog od atributa, a pri tom su iz razmatranja isključeni svi oni atributi koji su prije iskorišteni u istoj grani stabla. Rekurzija se nastavlja sve dok za promatrani čvor stabla nije homogen (sadrži primjere samo jedne klase – čvor se označava tom klasom) ili daljnje grananje nije moguće (svi atributi su već ranije iskorišteni u rekurziji). U ovom slučaju čvor se

označava najfrekventnijom klasom primjera koji su dospjeli do tog čvora. Alternativno, postoji probabilistička interpretacija po kojoj se list označava svim klasama primjera koji su dospjeli do njega, uz pridruživanje pripadajuće vjerojatnosti svakoj od njih. Pridružena vjerojatnost odgovara relativnoj frekvenciji klase unutar lista.

Iz opisa algoritma očigledna je nepovratna strategija pretraživanja: jednom odabrani atribut postaje osnova za grananje stabla, bez mogućnosti da se taj odabir naknadno preispita. Pri tom je upravo odabir atributa za grananje u svakom koraku rekurzije presudan za kvalitetu konačnog rezultata. O odabiru atributa ovisi hoće li rezultirajuće stablo biti glomazna struktura pretjerano prilagođena skupu za učenje, ili kompaktni prikaz općenitih pravilnosti koje postoje u podacima. Stoga kriterij odabira atributa za grananje predstavlja centralni dio algoritma, koji usmjerava pretraživanje u skupu potencijalnih rješenja.

IV. 3 Odabir atributa grananja

Funkcija vrednovanja aproksimacije u algoritmu konstrukcije stabala odlučivanja vezana je uz odabir atributa koji će poslužiti kao kriterij grananja u unutrašnjim čvorovima stabla. Osnovni način zaustavljanja rekurzije su čvorovi kojima pripadaju primjeri samo jedne od klasa. Stoga je poželjno kao kriterij grananja odabirati one atribute koji proizvode što homogenije podskupove primjera za učenje kao rezultat grananja.
Dobra mjera (ne)homogenosti nekog skupa dolazi iz teorije informacija, a naziva se *entropija* ili *informacijska vrijednost*. Neka je sa $|C|$ označen broj klasa prisutnih u skupu podataka za učenje S, a sa p_i relativna frekvencija klase C_i unutar S, $1 \le i \le |C|$. Izraz (1) definira entropiju skupa primjera S.

$$E(S) = \sum_{i=1}^{|C|} - p_i \log_2 p_i \qquad (1)$$

Na osnovu entropije definira se *informacijski dobitak*, koji služi kao mjera efektivnosti atributa u klasifikaciji primjera. Neka je sa A_i označen proizvoljni atribut koji se pojavljuje u skupu podataka za učenje S. Tada se informacijski dobitak atributa A_i u odnosu na S definira izrazom:

$$IGain(S, A_i) = E(S) - \sum_{a_j \in Dom(A_i)} \frac{|S_j|}{|S|} E(S_j), \qquad (2)$$

gdje $S_j \subseteq S$ označava skup $S_j = \{s \in S,\ A_i(s) = a_j\}$.

U algoritmu konstrukcije stabala odlučivanja kao kriterij odabira atributa koristi se upravo informacijski dobitak, te se tako grananjem nastoji što ranije postići homogenost rezultirajućih podskupova. Poznato svojstvo informacijskog dobitka je da favorizira atribute s većim brojem vrijednosti, što može dovesti do anomalija u konstrukciji stabala odlučivanja. Stoga se u praksi obično koristi korigirana mjera dobitka [7], koja penalizira atribute s većim brojem vrijednosti.

IV. 4 Šum u podacima (podrezivanje)

Dok god postoje mogući atributi za grananje i ulazni podskupovi primjera nisu homogeni, algoritam konstrukcije stabala odlučivanja će razgranavati stablo nastojeći akomodirati svaki primjer iz skupa podataka za učenje. U slučaju šuma u podacima za učenje, ovo uzrokuje pretjeranu prilagođenost podacima za učenje (engl. *overfitting*): grananje će se provoditi i na atributima koji samo prividno proizvode informacijski dobitak, dok je stvarni uzrok grananja šum u primjerima za učenje. To dovodi do lošijih klasifikacijskih performansi na dotad neviđenim primjerima, tj. do lošije kvalitete induciranog modela.
Dvije su metode izbjegavanja nepotrebnog grananja stabla [1]:

- *zaustavljanje grananja*, u kojem rekurzija zaustavlja grananje prilikom konstrukcije stabla odlučivanja prije postizanja savršene klasifikacije primjera iz skupa za učenje,
- *podrezivanje*, koje se oslanja na naknadni proces redukcije (podrezivanja) već razgranatog stabla.

Obje metode imaju neke prednosti. Zaustavljanje grananja stabla je efikasniji pristup, jer se izbjegava konstrukcija nepotrebnih podstabala koja opet treba posebnim postupkom podrezivati. S druge strane, postoji rizik od preranog zaustavljanja rasta stabla (npr. odvojeno promatrana dva atributa mogu se činiti gotovo nevažnima, a njihova kombinacija može imati izrazite prediktivne sposobnosti).

IV. 5 Neodređene vrijednosti atributa

Neodređene vrijednosti atributa u primjerima uzrokuju poteškoće kako pri konstrukciji stabla odlučivanja, tako i pri klasifikaciji primjera izgrađenim stablom odlučivanja. Jednostavne prilagodbe koje omogućuju rad s neodređenim vrijednostima uključuju zamjenu neodređene vrijednosti atributa onom vrijednošću koja se najčešće pojavljuje, bilo u cijelom skupu za učenje ili u podskupu koji odgovara promatranom čvoru. Također, moguće je primjere koji sadrže neodređene vrijednosti bitnih atributa u potpunosti ignorirati.

Međutim, postoji učinkovitiji način korištenja primjera s neodređenim vrijednostima, a zasniva se na probabilističkoj interpretaciji i "cijepanju" primjera u dijelove. Sličan mehanizam koristi se i pri klasifikaciji i pri konstrukciji stabla odlučivanja. Ovaj postupak opisan je u [6], gdje je ilustrirana i učinkovitost ovakvog postupka klasifikacije i konstrukcije stabla odlučivanja. Učinkovitost se ogleda u sporom padu točnosti klasifikacije pri povećanju broja neodređenih vrijednosti u primjerima za učenje i klasifikaciju.

IV. 6 Svojstva konstrukcije stabala odlučivanja

Stabla odlučivanja kao tehnika modeliranja pravilnosti u podacima je intenzivno korištena i često izučavana. Istraživane su različite varijacije postupka konstrukcije stabala, od različitih kriterija za odabir atributa grananja, drugih metoda podrezivanja stabla, ili modificiranog oblika testova u čvorovima (npr. korištenjem više od jednog atributa ili vrijednosti [8]). Razmatrane su i korekcije postupka koje smanjuju potrebne računalne i vremenske zahtjeve, što je posebno važno kod primjene na glomaznim

skupovima podataka (npr. metoda *prozora* u skupu primjera za učenje).

Osnovne prednosti stabala odlučivanja kao tehnike modeliranja podataka uključuju:
- razumljivost konstruiranih modela,
- eksplicitno izdvajanje atributa bitnih za konkretni klasifikacijski problem,
- zahtijeva relativno male računalne resurse,
- učinkovito korištenje numeričkih atributa i primjera s neodređenim vrijednostima.

Temeljni nedostatci koje ova tehnika ispoljava su:
- sklonost pretjeranoj prilagodbi podacima za učenje,
- segmentiranje univerzuma razapetog domenama atributa je svedeno na hiperravnine paralelne osima univerzuma, što bitno ograničava mogućnosti omeđivanja klasa.

V KLASIFIKACIJSKA PRAVILA

V. 1 Reprezentacija modela

Postoji više ekvivalentnih načina zapisa klasifikacijskih pravila. Najčešća notacija bilježi klasifikacijsko pravilo kao formulu oblika

$$cond(s) \rightarrow s \in C_i \quad , \qquad (4)$$

gdje je *s* proizvoljni element univerzuma *U*, koji je definiran kao Kartezijev produkt domena svih atributa $U=Dom(A_1)\times ... \times Dom(A_n)$.

U izrazu (4) *cond*(s) se naziva *antecedenta* (uvjet) pravila, a predstavlja uvjet na vrijednosti prognostičkih atributa u primjeru *s*. *Konzekventa* (posljedica) pravila $s \in C_i$ primjeru *s* pridružuje klasu C_i, pod uvjetom da je antecedenta ispunjena, tj. da primjer *s* zadovoljava uvjet *cond*(*s*).
Dozvoljeni oblik antecedente pravila može ovisiti o korištenom algoritmu indukcije pravila. Najjednostavniji oblik antecedente je konjunkcija elementarnih uvjeta na vrijednosti pojedinih atributa, tj. konjunkti su oblika $A_j(s)=a_k$, gdje je $a_k \in Dom(A_j)$. Budući da je u zapisu pravila primjer *s* jedina varijabla, ona se često ispušta, pa konjunkti poprimaju oblik $A_j=a_k$, a pravilo zapis $cond \rightarrow C_i$.

U tehnici indukcije pravila model se reprezentira skupom klasifikacijskih pravila pomoću kojih se vrši klasifikacija primjera. U skupu pravila može postojati više od jednog pravila za svaku od klasa, tj. može postojati više pravila s istom konzekventom (u kojem slučaju njihova semantika odgovara disjunkciji pojedinačnih pravila, odnosno njihovih antecedenti).
Ako primjer *s* zadovoljava antecedentu pravila $r \in R$ kaže se da pravilo *r* prekriva primjer *s*. Skup svih primjera koji su prekriveni pravilom *r* naziva se *prekrivač* pravila *r* i označava σ_r. Važno je primijetiti da se za skup pravila *R* ne zahtijeva disjunktnost prekrivača različitih pravila. Jednako tako se ne zahtijeva da unija prekrivača svih pravila iz *R* prekriva cijeli univerzum *U*.
Stoga ova činjenica može stvarati probleme prilikom klasifikacije primjera, budući je dozvoljena situacija da primjer *s* bude prekriven s više pravila, ili da ne bude prekriven ni jednim pravilom. Ako primjer *s* nije prekriven ni jednim pravilom iz *R*, on se pridjeljuje najfrekventnijoj klasi u skupu za učenje, ili alternativno klasi pravila s najvećim prekrivačem na skupu za učenje. Ako je pak primjer *s* prekriven s više pravila, najčešći način klasifikacije takvog primjera je primjena pravila *r* koje prekriva *s*, a koje istovremeno prekriva najviše primjera iz skupa za učenje.

Postoji više različitih tehnika indukcije klasifikacijskih pravila. Tehnike se mogu razlikovati po dozvoljenom obliku pravila, načinu pretraživanja prostora rješenja, ili korištenoj funkciji vrednovanja.

V. 2 Postupak pretraživanja

Tehnike indukcije pravila se po pristupu pretraživanju prostora rješenja bitno razlikuju od tehnika konstrukcije stabala odlučivanja. Dok se kod stabala odlučivanja odjednom konstruiraju opisi svih klasa, tehnike indukcije pravila odvojeno promatraju svaku od klasa. Postupak se fokusira na jednu od klasa, te pokušava konstruirati pravila koja prekrivaju što više pozitivnih primjera te klase, istovremeno isključujući sve negativne primjere klase (ignorirajući pritom njihove razlike u klasi). Postupak se neovisno ponavlja za svaku od klasa, tako da konačni skup pravila uključuje klasifikacijska pravila za svaku klasu.

Jedna od osnovnih tehnika indukcije pravila nosi naziv *Prism* [9], a razvijena je u osamdesetim godinama dvadesetog stoljeća. Najjednostavniji oblik koristi pravila čija je antecedenta konjunkcija uvjeta na vrijednosti atributa. Spada u tehnike vođene modelom, koje kreću od najopćenitijeg pravila čiju točnost povećavaju uzastopnim operacijama specijalizacije. Operacija specijalizacije koju koristi *Prism* sastoji se od dodavanja konjunkta oblika $A_j=a_k$ u antecedentu pravila. Strategija pretraživanja prostora rješenja je nepovratna. Koristi se metoda uspona na vrh, vođena heuristikom odabira specijalizacije.

Heuristika odabira koristi informacijski dobitak, slično odabiru atributa grananja kod stabala odlučivanja. Neka je sa *q* označena proizvoljna specijalizacija pravila *r*, a sa $\sigma_r^+ \subseteq \sigma_r$ označen skup pozitivnih primjera klase koje prekriva pravilo *r*.. Informacijski dobitak postignut specijalizacijom *q* definiran je izrazom (5).

$$gain(q,r) = \sigma_q^+ \left(\log_2 \frac{|\sigma_q^+|}{|\sigma_q|} - \log_2 \frac{|\sigma_r^+|}{|\sigma_r|} \right) \qquad (5)$$

Heuristika informacijskog dobitka nalaže odabir one specijalizacije pravila koja maksimizira informacijski dobitak. Heuristika informacijskog dobitka stavlja veći naglasak na broj pozitivnih primjera klase koji pravilo prekriva, ne inzistirajući striktno na točnosti pravila. Stoga ova heuristika preferira općenitija pravila tolerirajući ponešto sniženu točnost.

V. 3 Šum u podacima (podrezivanje)

Zbog mogućnosti postojanja šuma u podacima, savršena klasifikacija na skupu za učenje u pravilu je indikator pretjerane prilagodbe modela podacima za

učenje. Kao i kod stabala odlučivanja, postoje dva načina izbjegavanja pretjerano specijaliziranih pravila:

- zaustavljanje dodavanja konjunkata prilikom konstrukcije pravila prije nego što se dosegne potpuna točnost pravila, ili
- naknadno podrezivanje potpuno točnog pravila uklanjanjem pojedinih konjunkata.

U praksi se češće primjenjuje naknadno podrezivanje pravila, zbog izbjegavanja preranog zaustavljanja procesa specijalizacije pravila. Ključan element postupka podrezivanja pravila kriterij na osnovu kojeg se prosuđuje opravdanosti uklanjanja pojedinih konjunkata. Postoji više različitih kriterija usporedbe pravila i njegove generalizacije dobivene podrezivanjem. Jedan od pristupa evaluaciji pravila zasniva se procjeni vjerojatnosti da slučajno odabrano pravilo postigne istu ili veću točnost klasifikacije od promatranog pravila. Proširenje Prism tehnike indukcije pravila koje uključuje podrezivanje pravila upotrebom ovako definiranog kriterija poznato je pod nazivom *Induct* [10], a razvijeno je sredinom devedesetih godina dvadesetog stoljeća.

V. 4 Neodređene vrijednosti atributa

Najbolji način rada s neodređenim vrijednostima atributa prilikom indukcije klasifikacijskih pravila je pretpostavka da one ne zadovoljavaju niti jedan test. Efekt ove pretpostavke je dvojak. S jedne strane, pravilo se gradi izdvajanjem pozitivnih primjera za koje su vrijednosti traženih atributa poznate, pa zasigurno zadovoljavaju antecedentu pravila. S druge strane, odluka za primjere s neodređenom vrijednošću traženog atributa se odgađa za kasnije iteracije procesa, kada je problem jednostavniji jer je većina ostalih primjera već pokrivena pravilima i izdvojena iz skupa za učenje. U tako pojednostavljenoj situaciji s manjim brojem primjera povećava se mogućnost pronalaska testa na neki od atributa s poznatim vrijednostima koji će na zadovoljavajući način prekriti problematične primjere s neodređenim vrijednostima drugih atributa.

Po načinu tretiranja neodređenih vrijednosti tehnike indukcije pravila su u prednosti pred stablima odlučivanja, jer se za primjere sa neodređenim vrijednostima nekih atributa traže pravila koja te atribute ne koriste.

V. 5 Svojstva indukcije klasifikacijskih pravila

Postoji veći broj bitno različitih postupaka indukcije klasifikacijskih pravila. Različitosti sežu od načina prostora pretraživanja rješenja, kriterija odabira specijalizacije/generalizacije pravila, tehnike podrezivanja pravila, pa sve do oblika i interpretacije konstruiranih pravila. Primjeri takvih tehnika su liste odlučivanja [11] koje predstavljaju uređene nizove pravila koji se interpretiraju u zadanom redoslijedu, ili pravila s iznimkama [10] koja dozvoljavaju navođenje uvjeta pod kojima pravilo ne vrijedi. Oni pokazuju bitno različita svojstva, čije razmatranje prelazi okvire ovog rada. Stoga se ovdje analiziraju svojstva samo opisane tehnike indukcije pravila.

Budući da se svako stablo odlučivanja može prikazati kao skup klasifikacijskih pravila, opisana tehnika indukcije pravila dijeli dosta temeljnih svojstava sa konstrukcijom stabala odlučivanja. I jedna i druga tehnika su sklone pretjeranoj prilagodbi podacima za učenje, te ne uspijevaju dobro izraziti nelinearne granice među klasama. S druge strane, konstruiraju razumljive modele, te na sličan način koriste numeričke atribute, dok se u tretmanu neodređenih vrijednosti ponešto razlikuju. I tehnike indukcije pravila efikasno koriste računalne resursa, budući da se broj promatranih primjera smanjuje u svakoj iteraciji postupka.

Za razliku od stabala odlučivanja, klasifikacijska pravila bolje izražavaju disjunkcije u pravilnostima. Razlog tome leži u modularnosti pravila: svako pravilo predstavlja zaseban komadić otkrivenog znanja, neovisan o ostalim pravilima iz skupa pravila. S druge strane, cijelo stablo odlučivanja je jedinstvena struktura, pa izražavanje disjunkcije vodi repliciranju podstabala unutar istog stabla. Iako modularnost klasifikacijskih pravila nosi veću izražajnost i razumljivost, takav prikaz modela plaća cijenu u obliku moguće nejednoznačnosti klasifikacije. Stoga su potrebne dodatne heurističke metode za razrješavanje situacije kada primjer zadovoljava više ili nijedno od konstruiranih pravila [12].

VI KLASIFIKACIJA PAMĆENJEM PRIMJERA

VI. 1 Reprezentacija modela

Klasifikacija pamćenjem primjera spada u najjednostavnije tehnike dubinske analize podataka. Ne vrši eksplicitnu generalizaciju ciljnog pojma na osnovu svojstava izvedivih iz skupa za učenje, već se svodi na memoriranje pojedinačnih primjera iz skupa za učenje. Dakle, osnovni oblik algoritma ne uključuje procesiranje primjera iz skupa za učenje u fazi konstrukcije modela, već samo njihovu pohanu.

Klasifikacija novih primjera se obavlja prema principu *najbližeg susjeda*. Novi primjer se uspoređuje s memoriranim primjerima iz skupa za učenje korištenjem definirane metrike. Metrika definira udaljenost primjera na osnovu vrijednosti njihovih atributa, a korespondira intuitivnom shvaćanju sličnosti primjera: što su primjeri sličniji, udaljenost je manja. Klasifikacija novog primjera tada se svodi na pretraživanja skupa za učenje s ciljem pronalaženje primjera koji mu je (u smislu metrikom definirane udaljenosti) najbliži. Klasa tog primjera se pridjeljuje novom primjeru kojeg je trebalo klasificirati.

Iako eksplicitno ne izražava svojstva koja karakteriziraju pojedine klase, može se govoriti o modelu kojeg inducira ova tehnika modeliranja. Implicitnu generalizaciju koja je u podlozi opisanog načina klasifikacije određuje korištena metrika te skup primjera za učenje. Metrika proširuje utjecaj primjera za učenje *s* na okolni dio univerzuma kojem je *s* najbliži primjer iz skupa za učenje. Klase su omeđene granicom koja se proteže polovicom udaljenosti među najbližim primjerima različitih klasa.

VI. 2 Funkcija udaljenosti primjera

Dakle, oblik klasifikacijskog modela određen je skupom primjer za učenje te metrikom koja proizlazi iz funkcije udaljenosti. U upotrebi je više različitih metrika, a najčešće se koristi *Euklidska*. Neka je sa $x = (x_1, x_{2,\dots,} x_n)$ označen vektor vrijednosti atributa proizvoljnog primjera. Euklidska metrika je tada definirana izrazom (6).

$$\| x \| = \sqrt{\sum_{i=1}^{n} x_i^{\,2}} \qquad (6)$$

Metrika na standardan način definira udaljenost, pa izraz (7) definira Euklidsku udaljenost primjera x i y.

$$d(x, y) = \| x - y \| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (7)$$

Variranjem potencije koordinata vektora u definiciji Euklidske metrike na sličan način mogu se definirati i druge metrike. Primjerice, izostavljanje kvadriranja (odnosno, korištenje potencije 1) uz upotrebu apsolutne vrijednosti daje tzv. pravokutnu metriku. Općenito, višim potencijama se velike razlike u vrijednostima pojedinih koordinata dodatno naglašavaju, nauštrb koordinata kod kojih je razlika u vrijednosti mala.

U praktičnoj primjeni ovih metrika vrijednosti svih atributa se normaliziraju na interval [0,1], čime se izjednačava utjecaj pojedinih atributa na konačan rezultat. Kako su u praksi rijetki problemi kod kojih su svi atributi imaju jednaku klasifikacijsku vrijednost, uz normalizaciju se u pravilu uvode i težinske vrijednosti atributa na način da veća težinska vrijednost daje atributu veći utjecaj na izračun udaljenosti primjera.

Općenito, postoje radikalno drukčiji pristupi definiranju funkcije udaljenosti. Jedan od njih je vjerojatnosni pristup, kod kojeg se definiraju operacije transformacije primjera za učenje. Udaljenost dvaju primjera se utvrđuje promatranjem niza operacija pomoću kojih se jedan od primjera može transformirati u drugog, te izračunom vjerojatnosti da se takva transformacija dogodi uz slučajan odabir operacija i njihovog redoslijeda [13]. Ovakvi pristupi u određenim primjenama mogu rezultirati znatnim prednostima (prirodan tretman specifičnih tipova atributa kao npr. dani u tjednu koji se mjere cirkularnom skalom, i sl.).

VI. 3 Postupak pretraživanja

Osnovni oblik tehnike klasifikacije pamćenjem primjera ne provodi eksplicitnu generalizaciju svojstava ciljnog pojma, tj. ne pretražuje prostor rješenja u potrazi za što boljim modelom. U postupku se pojavljuje samo jedan implicitni klasifikacijski model koji je u potpunosti određen skupom za učenje i funkcijom udaljenosti. Postoje nadogradnje osnovnog algoritma koje u određenom opsegu modificiraju skup primjera za učenje ili funkciju udaljenosti. Modifikacije osnovnog klasifikacijskog modela se provode nizom operacija nad modelom, pa se može se govoriti o postupku pretraživanja prostora rješenja.

Jedna od varijanti klasifikacije pamćenjem primjera nastoji reducirati broj primjera u skupu za učenje, prvenstveno radi smanjenja opsega pretraživanja pri klasifikaciji novih primjera.
Pamćenje samo odabranog podskupa primjera za učenje predstavlja generalizaciju modela koji sadrži cijeli skup za učenje. Sam postupak formiranja takvog podskupa primjera je iterativan, a sastoji se od uvrštavanja ili eliminacije primjera prema unaprijed definiranom kriteriju. Kriterijem se nastoji u skupu zadržati reprezentativne primjere (tipično oni u blizini granice među klasama), koji posredstvom funkcije udaljenosti dobro generaliziraju područje u kojem se nalaze. Iz skupa se izbacuju primjeri koji bitno ne pridonose oblikovanju područja odgovarajuće klase (tipično oni iz unutrašnjosti omeđenog područja klasa).

Iako postoji više različitih kriterija prihvata odnosno eliminacije primjera, uglavnom se radi o nepovratnim strategijama pretraživanja pohlepnog karaktera koje pretraživanju pristupaju po načelu *odozdo na gore* (tj. od pojedinačnih primjera ka reprezentativnim). Jednostavniji postupci/kriteriji eliminacije primjera često pate od izraženih nedostataka, od kojih se najbitniji odnosi na loše ponašanje u uvjetima šuma u podacima. Naime, primjeri sa šumom su po definiciji različiti od okolnih primjera (koji ih dobro ne opisuju), pa postoji tendencija akumuliranja primjera sa šumom u rezultirajući skup primjera. To uvelike smanjuje njegovu reprezentativnost i performanse klasifikacije.
Stoga se u praksi češće upotrebljavaju drugi, ponešto složeniji kriteriji odabira reprezentativnih primjera.

VI. 4 Šum u podacima (podrezivanje)

Osnovni oblik tehnike klasifikacije pamćenjem primjera je prilično podložan problemu šuma u podacima za učenje. Razlog tome leži u činjenici da se klasifikacija novog primjera oslanja na samo jedan (najbliži) primjer iz skupa za učenje. Prirodno proširenje postupka klasifikacije koje bitno smanjuje utjecaj šuma provodi klasifikaciju prema principu *k najbližih susjeda*. Tada se pri klasifikaciji promatranog primjera pronalazi *k* najbližih primjera, od kojih svaki sudjeluje u klasifikaciji novog primjera po principu većinskog glasovanja.
Zbog bitno poboljšane točnosti klasifikacije u uvjetima šuma, varijanta k najbližih susjeda je gotovo u potpunosti istisnula osnovni oblik algoritma.

Drugi pristup tretiranju šuma u podacima sastoji se od detekcije primjera sa šumom i njihovog izdvajanja iz skupa za učenje. Najpoznatija takva varijanta algoritma klasifikacije pamćenjem primjera poznata je pod nazivom *IB3* (od engl. *Instance-Based learner ver.3*), koja na temelju Bernoullijevog procesa određuje pragove odbacivanja i prihvaćanja primjera u skup primjera za pamćenje [14].

VI. 5 Svojstva klasifikacije pamćenjem primjera

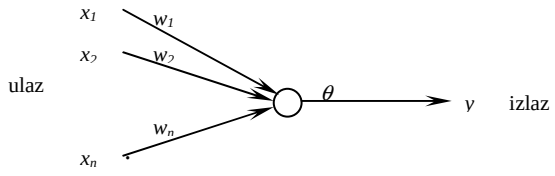
Izvori analize primjera korištenjem najbližih susjeda mogu se tražiti u statistici, gdje se shema k najbližih susjeda koristila još u pedesetim godinama dvadesetog stoljeća [15]. Kao postupak klasifikacije pojavljuje se desetak godina kasnije [16], a intenzivno se koristila na području raspoznavanja uzoraka.

Klasifikacija pamćenjem primjera popularna postaje početkom 1990-ih kroz radove D.Aha [17], u kojima se nadogradnjama osnovnog postupka umanjuju njegovi nedostaci. Pokazano je da uvođenje težinskih vrijednosti atributa i postupak filtriranja primjera sa šumom značajno poboljšavaju klasifikacijske sposobnosti ove tehnike, podižući ih na razinu usporedivu s ostalim popularnim tehnikama.

Najvažnija prednost tehnike klasifikacije pamćenjem primjera u odnosu na stabla odlučivanja i klasifikacijska pravila je mogućnost izražavanja proizvoljnih po dijelovima linearnih granica među klasama, dok su spomenute metode ograničene na linearne granice paralelne s osima univerzuma. Temeljni nedostatak ove tehnike je činjenica da klasifikacijski model nije izražen eksplicitno, u obliku koji bi bio deskriptivan u terminima domene klasifikacijskog problema.

VII NEURONSKE MREŽE

Jedna od primjena umjetnih neuronskih mreža su i klasifikacijski problemi, odnosno postupci strojnog učenja. Neuronske mreže su gusto isprepletene strukture međusobno povezanih računskih elemenata, nalik povezanim usmjerenim grafovima [18]. Osnovni element neuronske mreže naziva se neuron, a nalikuje čvoru grafa s pripadajućim ulaznim i izlaznim granama. Slika 2 predstavlja shematski prikaz neurona.



Slika 2: Neuron, osnovni element neuronske mreže

Ulazni dio neurona čini realni vektor ulaznih vrijednosti (x_1,x_2, ... ,x_n), a izlazni je jedna realna vrijednost y. Neuron karakterizira vektor težinskih vrijednosti (w_1,w_2, ... ,w_n) pridružen ulazima, konstanta θ i nelinearna funkcija f (često se radi o sigmoidnoj funkciji). Računski gledano, neuron transformira ulazni vektor u izlaznu vrijednost na način da računa težinsku sumu ulaza, od nje oduzima prag θ, te rezultat prosljeđuje funkciji f. Dakle, svaki neuron računa vrijednost izlaza y koristeći izraz (9):

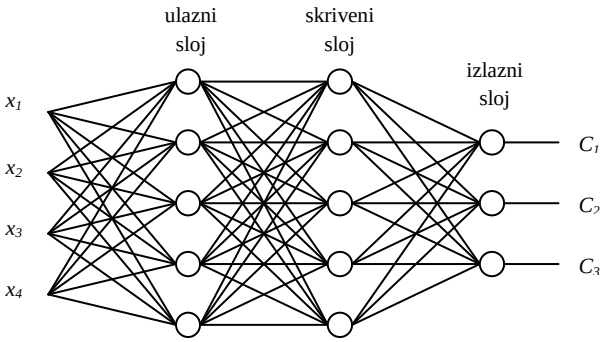
$$y = f\left(\sum_{i=1}^{n} w_i\, x_i - \theta\right) \quad , \qquad (9)$$

gdje f može biti zadana sa

$$f(x) = \frac{1}{1+e^{-x}} \quad . \qquad (10)$$

Neuronske mreže nastaju povezivanjem neurona na način da izlazna vrijednost jednog neurona postaje ulazna vrijednost jednog ili više drugih neurona. Ulaz u mrežu čine neuroni sa slobodnim ulazima, a izlaz mreže neuroni sa slobodnim izlazima.

U praksi se koristi mnogo različitih topologija neuronskih mreža. Kad su klasifikacijski problemi u pitanju, najpoznatija je topologija *višeslojnog perceptrona*. Način povezivanja neurona u višeslojni perceptron prikazan je na slici 3.



Slika 3: Višeslojni perceptron

U topologiji višeslojnog perceptrona neuroni su grupirani u slojeve, na način da izlazne vrijednosti neurona u jednom sloju predstavljaju ulazne vrijednosti slijedećeg sloja. Ulazne vrijednosti prvog sloja su u stvari ulazne vrijednosti u mrežu, dok izlazne vrijednosti zadnjeg sloja predstavljaju i izlaze mreže.

VII. 1 Reprezentacija modela

Višeslojni perceptron je posebno pogodan za aproksimiranje klasifikacijskih funkcija koje primjer određen vektorom vrijednosti atributa (x_1,x_2, ... ,x_n) preslikavaju u jednu ili više klasa C_1, C_2, ... , C_m. Ako primjer x pripada klasi C_i, tada za ulaznu vrijednost x izlaz mreže označen klasom C_i poprima visoku vrijednost (tipično 1). Ukoliko je x negativan primjer klase C_i, izlaz mreže označen klasom C_i će poprimiti nisku vrijednost (tipično 0).
Uz zadanu topologiju neuronske mreže, klasifikacijski model je određen težinskim vektorima w_i i vrijednostima praga θ svakog od neurona. Različiti modeli dobiju se optimiranjem ovih vrijednosti u postupku treniranja mreže na primjerima iz skupa za učenje. Primjer sačinjava ulazni vektor vrijednosti atributa primjera u paru sa željenim izlaznim vektorom, tj. vektorom čija je i-ta komponenta 1 ako primjer pripada klasi C_i, a 0 inače.

VII. 2 Postupak pretraživnja

Algoritmi treniranja neuronskih mreža korištenjem skupa za učenje modificiraju vrijednosti težinskih vektora i pragova neurona. U svjetlu tehnika modeliranja, radi se o postupku pretraživanja prostora rješenja optimiranjem parametara klasifikacijskog modela.

Prilikom treniranja neuronskih mreža, primjeri iz skupa za učenje se koriste jedan po jedan. Za svaki od primjera računa se izlazna vrijednost mreže, te se uspoređuje sa traženim izlazom. Ovisno o razlici stvarnog i traženog izlaza mreže, težinski vektori i vrijednosti praga u neuronima se korigiraju obrnuto proporcionalno veličini greške koju su uzrokovali u izlaznoj vrijednosti. Dakle, ukoliko je rezultat veći od traženog smanjuju se težine ulaza koje generiraju razliku i obrnuto.

Jedna od prvih i najčešće korištenih metoda treniranja neuronskih mreža nosi naziv *propagacija greške unatrag*. Metoda koristi iterativan postupak za propagaciju greške (odnosno razlike stvarnog i traženog izlaza) od neurona izlaznog sloja unatrag prema unutarnjim slojevima mreže. Propagaciju greške prati korekcija odgovarajućih težinskih vrijednosti, pa se na taj način korekcija parametara modela širi od izlaznog prema ulaznom sloju mreže. Algoritam staje kad se sve težinske vrijednosti u neuronskoj mreži stabiliziraju.

VII. 3 Svojstva klasifikacije neuronskim mrežama

Klasifikacija neuronskim mrežama se pokazala vrlo dobrom upravo na težim klasifikacijskim problemima, kod kojih je teško ili nemoguće koristiti klasične tehnike simboličkog učenja. Neuronske mreže mogu izraziti vrlo složene granice među klasama, za razliku od klasičnih tehnika koje presijecaju univerzum linearnim funkcijama, i to obično paralelno sa osima univerzuma. Osim toga, neuronske mreže su dobro prilagođene klasifikaciji u uvjetima šuma u podacima.

Jedan od nedostataka neuronskih mreža je relativno spor i zahtijevan proces indukcije modela u usporedbi s klasičnijim tehnikama, čak do nekoliko redova veličine [19]. Značajan nedostatak je i činjenica da klasifikacijski model reprezentiran neuronskom mrežom nije eksplicitno izražen, u obliku strukturnog opisa važnih odnosa među varijablama. Implicitan model koji skriva odnose varijabli u mrežnoj strukturi i velikom broju težinskih vrijednosti nije razumljiv ni podložan verifikaciji ili interpretaciji u okviru domene izvornog klasifikacijskog problema.

VIII ZAKLJUČAK

Dubinska analiza podataka je relativno mlada disciplina, s velikim potencijalom za daljnje istraživanje i obećavajućim rezultatima primjene u praksi. Svoju popularnost može zahvaliti upravo uspješnoj primjeni u rješavanju nekih tipova problema, prvenstveno u poslovnom svijetu. S druge strane, popularnost sa sobom nosi i teret komercijalnih preuveličavanja same tehnologije i njenih mogućnosti.

Kolike su realne mogućnosti dubinske analize podataka u stvarnim poslovnim primjenama? U kojoj mjeri ova tehnologija rezultatima može konkurirati rezoniranju čovjeka-stručnjaka, potpomognutog današnjim naprednim stručnim i tehničkim alatima u domeni rješavanog problema?
Ovaj rad predstavlja uvod u istraživanje koje će se baviti upravo navedenim pitanjima, na primjerima stvarnih poslovnih problema iz područja osiguranja. Tehnike dubinske analize podataka karakteriziraju svojstva koja se od tehnike do tehnike mogu bitno razlikovati. Odabir adekvatne tehnike modeliranja u svakoj od faza procesa ima presudan utjecaj na oblik i kvalitetu rezultata. Ozbiljan pristup podrazumijeva analizu svojstava promatranog problema, te krojenje procesa dubinske analize podataka prema njegovim specifičnostima. Stoga je za uspješnu primjenu nužno dobro poznavanje procesa i tehnika dubinske analize podataka te njihovih svojstava, čiji je kratak pregled i usporedba sadržaj ovog rada.

REFERENCES:

[1] I.H. Witten, E. Frank: *Data mining: Practical machine learning tool and techniques with Java implementations*, Morgan Kaufmann, San Francisco, 2000.

[2] M. Holsheimer, A. Siebes, "Data mining: The search for knowledge in databases", *Technical Report CS-R9406*, CWI, 1994.

[3] C. Howard, D.C. Plummer, Y. Genovese, J. Mann, D.A. Willis, D.M. Smith, "The Nexus of Forces: Social, Mobile, Cloud and Information", *Gartner Report G00234840*, Gartner, 2012.

[4] F. Gens, "The 3rd Platform: Enabling Digital Transformation", *IDC whitepaper #244515*, IDC, 2013.

[5] P. Chapman, J. Clinton, T. Khabaza, T. Reinartz, R. Wirth, "The CRISP-DM process model", *CRISP-DM Consortium Technical Report*, 1999.

[6] J.R. Quinlan, "Induction of decision trees", *Machine Learning 1*, 1986.

[7] J.R. Quinlan, *C4.5: Programs for machine learning*, Morgan Kaufmann, San Francisco, 1993.

[8] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, *Classification and regression trees*, Wadsworth, Monterey, 1984.

[9] J. Cendrowska, "PRISM: An algorithm for inducing modular rules", *International Journal of Man-Machine Studies 27*, 1987.

[10] B.R. Gaines, P. Compton, "Induction of ripple-down rules applied to modeling large data bases", *Journal of Intelligent Information Systems 5*, 1995.

[11] R.L. Rivest, "Learning decision lists", *Machine Learning 2*, 1987.

[12] V. Cho, B. Wütrich, "Consistent predictions for categorical data", in *Proceedings of the International Conference on Methodologies for Intelligent Systems*, 1996.

[13] J.G. Cleary, L.E. Trigg, "K*: An instance-based learner using an entropic distance measure", in *Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, 1995.

[14] D. Aha, D.Kibler, M.Albert, "Instance-based Learning Algorithms", *Machine Learning, 6*, 1991.

[15] E. Fix, J.L. Hodges, "Discriminatory analysis; non-parametric discrimination: Consistency properties", *Technical Report 21-49-004(4)*, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[16] M.V. Johns, "An empirical Bayes approach to non-parametric two-way classification", *Studies in item analysis and prediction*, Stanford University Press, Palo Alto, 1961.

[17] D. Aha, "Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms", *International Journal of Man-Machine Studies 36*, 1992.

[18] B. Müller, J. Reinhardt, *Neural networks, an introduction*, Physics of Neural Networks, Springer-Verlag, Berlin, 1991.

[19] J.R. Quinlan, "Comparing connectionist and symbolic learning methods", *Computational Learning Theory and Natural Learning Systems, Vol.1*, MIT Press, Cambridge, 1994.

[20] V. Jovanovic, I. Bojicic, C. Knowles, M. Pavlic, "Persistent staging area models for Data Warehouses", *Issues in Information Systems, Vol.13 Issue 1*, 2012.

[21] S. Maržić, P. Krneta, M. Pavlić, "Spend analysis systems", in *37th Internation Convention MIPRO 2014*, 2014