SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

POSTUPCI ANALIZE PODATAKA U IZGRADNJI PROFILA KORISNIKA USLUGA

MAGISTARSKI RAD

Zagreb, 2004.

Magistarski rad je izrađen u
Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave

Mentor: Prof.dr.sc. Xxxx Xxxxxx

Magistarski rad ima: 137 stranica

Rad br.:

Povjerenstvo za ocjenu magistarskog rada:

1. Prof.dr.sc. Xxxxx Xxxx – predsjednik
2. Prof.dr.sc. Xxxx Xxxx – mentor
3. Dr.sc. Xxxx Xxxx – Institut "Ruđer Bošković" Zagreb

Povjerenstvo za obranu magistarskog rada:

4. Prof.dr.sc. Xxxx xxxxxx – predsjednik
5. Prof.dr.sc. Xxxx Xxxxx– mentor
6. Dr.sc. Xxxxx Xxxxxxxx – Institut "Ruđer Bošković" Zagreb

Rad obranjen 8. travnja 2004.g.

Sadržaj

1. Uvod

Informacijsko doba stvorilo je moderne računalne i telekomunikacijske tehnologije koje omogućuju prikupljanje i pohranu ogromnih količina podataka. Sveprisutna elektronska i računalna pomagala koriste se u praktički svim aspektima poslovnog i društvenog života. Osim što pojednostavljuju život, elektronska pomagala postaju generatori podataka, bilo da im je to osnovna namjena ili tek popratni efekt.
Podaci nastaju i bilježe se u gotovo svakoj svakodnevnoj situaciji. Možda nismo ni svjesni koliki dio naših odluka, izbora i navika je zabilježen u različitim bazama podataka. Primjerice, blagajna u trgovini bilježi svaki kupljeni proizvod; knjižnice i videoteke svaku posuđenu knjigu ili filmski naslov; telekomunikacijski operater ima podatke o vremenu, trajanju i odredištu svakog upućenog poziva; banke i davatelji kreditnih kartica raspolažu informacijama o financijskim navikama i mogućnostima, te iznosima i lokacijama različitih kupnji; web serveri bilježe posjećenost različitih stranica, a samim time navike i preferencije posjetitelja… Primjera je zaista mnogo.
Svi smo svjedoci pretrpanosti podacima. Količina i dostupnost podataka u današnjem svijetu je ogromna, a svakim danom se još više povećava. Nažalost, s porastom količine dostupnih podataka, mogućnost njihovog praćenja i razumijevanja drastično opada. Stoga se sve više povećava jaz između rastućeg trenda generiranja podataka i mogućnosti njihovog iskorištavanja.

1.1. EVOLUCIJA PROCESA ANALIZE PODATAKA

Današnje tehnologije čine prikupljanje podataka jednostavnim, a njihovu pohranu jeftinom. Stoga prikupljanje i pohrana podataka prestaju biti problemi, a u fokus dolazi njihova analiza i razumijevanje.
Široka rasprostranjenost baza podataka stvorila je potrebu za snažnim analitičkim alatima koji pohranjene podatke mogu pretvoriti u korisne informacije. Tradicionalne tehnike za analizu podataka su prvenstveno usmjerene na izračunavanje kvantitativnih i statističkih obilježja iz podataka. Međutim, ovakve tehnike ne mogu proizvesti kvalitativan opis pravilnosti i uzoraka (klasa) koji nisu eksplicitno izraženi u podacima, niti mogu izvoditi pretpostavke o razlozima grupiranja sličnih primjera u kategorije. Da bi se nosio s ovakvim zadacima, sustav za analizu podataka treba uključivati analitičke postupke s polja strojnog učenja (područje inteligentne analize podataka i otkrivanja znanja).

Evolucija mogućnosti i potreba u načinu korištenja i analizi prikupljenih podataka prikazana je na slici 1. Usporedo s napretkom tehnologije rasla je i mogućnost pohrane i procesiranja sve većih količina podataka.

U šezdesetim i sedamdesetim godinama dvadesetog stoljeća moguće je bilo pohraniti i efektivno koristiti vrlo ograničenu količinu podataka, koja se s današnjeg stanovišta čini smiješno malom. U takvim uvjetima pomno su se birali najnužniji podaci koji će se pohranjivati i obrađivati, te se opseg pohranjenih podataka nastojao svesti što manju mjeru (jedna od nuspojava ovog pristupa je i poznati Y2K problem). Tadašnji stupanj razvoja tehnologije, kao i mala količina i opseg pohranjenih podataka ograničavali su manipulaciju podacima na najosnovnije operacije.

	Evolucijski korak	Tipična pitanja	Tehnologija	Oblik rezultata (odgovora)	
1960-70	Sakupljanje podataka	Koliko jedinica proizvoda X ima na skladištu?	Računalne trake, diskovi	Brojke	Kb
1980-90	Pristup podacima	Koliki je iznos ukupne mjesečne prodaje proizvoda za određenu poslovnicu?	Relacijske baze podataka, SQL	Fiksni izvještaji, grafikoni	
	Skladištenje podataka	Kakav je ostvareni profit za proizvod X po prodajnim kanalima u odnosu na prošlu godinu?	Skladište podataka	Dinamičko izvještavanje, grafovi, tablice	Mb
2000	Inteligentna analiza podataka	Kakvi klijenti odlaze konkurenciji i zašto? Koja je vjerojatnost da klijent X prijeđe konkurenciji?	Algoritmi iz područja umjetne inteligencije, strojnog učenja	Modeli pravilnosti u podacima (stabla odlučivanja, klasifikacijska pravila, …)	Gb, Tb

Slika 1: Evolucija procesa analize podataka

Već osamdesete, a posebno devedesete godine dvadesetog stoljeća karakterizira sveobuhvatna primjena relacijskih baza podataka. Računalni resursi prestaju biti problem, što omogućuje pohranu većih količina podataka, kao i složenije analize. Relacijske baze podataka svoju primarnu primjenu našle su u uspostavi transakcijskih baza podataka. Nešto kasnije se kao nadgradnja transakcijskih sustava javljaju skladišta podataka, čija je osnovna svrha stvaranje okruženja za analitičko procesiranje podataka. Naime, transakcijske baze podataka su organizirane i izgrađene za podršku svakodnevnim operativnim poslovima. U tom smislu je i analiza transakcijskih podataka u transakcijskim sustavima uglavnom ograničena na ugrađene izvještaje fiksne strukture, kojima se transakcijski podaci agregiraju po unaprijed zadanim kriterijima. S druge strane, skladišta podataka integriraju podatke s više izvora, te ih organiziraju u strukture pogodne analiziranju u smislu postavljanja dinamičkih, ad-hoc upita. Tipičan element strukture skladišta podataka su tzv. *zvjezdaste sheme* i/ili *pahuljaste sheme*, u kojima se podaci od interesa pohranjuju s vezama na veći broj kategorija kojima pripadaju (*dimenzije*). Na taj način je omogućeno dinamičko razvrstavanje, agregiranje, sortiranje i usporedba podataka po različitim dimenzijama, što nudi daleko veću fleksibilnost u odnosu na fiksne, unaprijed definirane izvještaje. Osim toga, skladišta podataka često uključuju određeni skup analitičkih alata za utvrđivanje trendova i drugih kvantitativnih obilježja podataka.

Međutim, pravi potencijal izgrađenih transakcijskih baza podataka nije u samim sirovim podacima ili njihovim agregacijama, već u implicitnim pravilnostima koje su u njima skrivene.

Čest je slučaj da pojave u svijetu ovise o velikom broju čimbenika i njihovih interakcija. Zbog toga je nezgodno utvrditi kako koji čimbenik utječe na krajnji ishod kojeg pojava manifestira. Za očekivati je da ipak postoje određene zakonitosti po kojima se promatrana pojava ravna, te da se analiziranjem dovoljnog broja primjera za poznatim ishodom takve pravilnosti mogu izdvojiti. Upravo inteligentna analiza podataka predstavlja proces pronalaženja pravilnosti u podacima kojemu je cilj analiziranjem mnoštva sirovih podataka generirati općenitiji oblik znanja o problematici opisanoj podacima. Osim za stjecanje uvida u analiziranu problematiku, izvedene pravilnosti mogu služiti za prognoziranje ishoda u slučajevima za koje stvarni ishod nije poznat.
Primjenu inteligentne analize podataka u praksi omogućio je eksponencijalni rast procesorske snage dostupnih računala. Naime, iako su osnovni oblici mnogih tehnika inteligentne analize podataka poznati već desetljećima, u pravilu se radi se o algoritmima za značajnim zahtjevima na računalne resurse. Jednako tako, složenost većine algoritama bitno ovisi o količini podataka koji se analiziraju, što ih je činilo praktično neprimjenjivim na realnim problemima s glomaznim skupovima podataka. Sve snažnija i dostupnija računala bitno su promijenila ovakvu situaciju, posebno u posljednjih nekoliko godina. Široka dostupnost potrebne tehnološke osnovice, uz prepoznavanje potencijala i mogućnosti koje nudi doveli su inteligentnu analizu podataka u fokus novih poslovnih tehnologija. Tako se već danas postupci inteligentne analize podataka široko upotrebljavaju u poslovnoj praksi, gdje mogu uvelike doprinijeti kvalitetnijem poslovnom odlučivanju. Omogućuju donošenje informiranijih odluka uvažavanjem pravilnosti izvučenih iz već postojećih baza podataka, koje bi inače predstavljale pasivnu masu mnoštva prikupljenih podataka.

Inteligentna analiza podataka je interesantna gdje god postoje opsežnije baze podataka i potencijalne pravilnosti skrivene u njima. Takvih je situacija svakim danom sve više. S današnjim tempom rasta, procjenjuje se da se količina pohranjenih podataka u svijetu udvostruči svakih dvadeset mjeseci. Značaj inteligentne analize podataka se povećava kako raste količina podataka pohranjenih u transakcijskim bazama, jer omogućuje bolje razumijevanje i iskorištavanje ogromnog mnoštva prikupljenih podataka. Stoga se može očekivati daljnje povećanje interesa za ovo područje, kako u primjeni rezultata tako i u daljnjem razvoju i istraživanju.

1.2. INTELIGENTNA ANALIZA PODATAKA I INDUKTIVNO UČENJE

U literaturi postoji više različitih definicija inteligentne analize podataka. Iako postoje neke terminološke razlike među njima, smisao svih definicija je isti. Evo nekih od njih.

- Inteligentna analiza podataka je ekstrakcija implicitnih, dotad nepoznatih i potencijalno korisnih informacija iz glomaznih baza podataka [58].
- Inteligentna 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 [27].

– Otkrivanje znanja iz baza podataka je proces identificiranja ispravnih, novih, potencijalno korisnih i razumljivih modela i struktura implicitno sadržanih u postojećim podacima. Inteligentna analiza podataka predstavlja korak u tom procesu čija je srž primjena specifičnih algoritama koji uz prihvatljiva ograničenja na računalne resurse mogu generirati, odnosno pronaći implicitne pravilnosti, strukture ili modele u velikim količinama podataka [18].

Veći dio relevantne literature ne razlikuje termine otkrivanje znanja iz baza podataka (engl. *Knowledge Discovery in Databases*) i inteligentna analiza podataka (engl. *Data Mining*), pa će se i ovdje oni tretirati kao sinonimi.
Zajedničko svim definicijama inteligentne analize podataka je izdvajanje implicitnih pravilnosti koje je moguće izolirati analizom velikih količina podataka. Dakle, temeljni zadatak postupka je pronalaženje općenitih pravila o svojstvima promatranih objekata, uz uvjet da tako formulirana pravila imaju statističku potvrdu u analiziranim podacima. Pri tome osnovni problem leži u činjenici da je broj potencijalnih pravilnosti ogroman, što onemogućava validaciju svake od njih. Stoga algoritmi strojnog učenja u pravilu koriste različite pretpostavke i heuristike koje nauštrb ispravnosti rezultirajućih pravila poboljšavaju brzinu pronalaženja ili razumljivost i jednostavnost samih pravila.

Općenito govoreći, inteligentna analiza podataka generalizacijom svojstava pojedinačnih primjera izvodi općenitije zakonitosti i odnose koje primjeri ilustriraju. Stoga se može reći da u osnovi postupaka inteligentne analize podataka leže principi induktivnog učenja. Kognitivna bića pokušavaju razumjeti svoje okruženje stvaranjem i upotrebom pojednostavljenih modela samog okruženja i pojava u njemu. Stvaranje takvih pojednostavljenih modela okruženja postupkom generalizacije naziva se *induktivno učenje*. U fazi učenja, kognitivna bića promatraju okruženje i opažaju sličnosti među objektima i pojavama u njemu. Slične objekte grupiraju u klase i stvaraju pravila koja opisuju svaku klasu, tj. svojstva i pojave karakteristične za članove svake klase. Tako stvorena općenita pravila o objektima i pojavama predstavljaju višu razinu informacije o samom okruženju, odnosno razinu koja se može nazvati znanjem. Bitno je naglasiti da postupak induktivnog učenja stvara pravila koja nisu nužno ispravna, jer su zasnovana na ograničenom iskustvu o okruženju.
Istraživanjem automatizacije procesa induktivnog učenja bavi se strojno učenje, disciplina koja spada u šire područje računalne inteligencije. Inteligentna analiza podataka je čvrsto vezana uz strojno učenje. U užem smislu, ona predstavlja primjenu postupaka strojnog učenja na glomaznim skupovima podataka, prvenstveno velikim bazama podataka. To sa sobom nosi dodatne probleme, vezane prvenstveno uz obujam i kvalitetu korištenih podataka.

Osnovna razlika induktivnog i deduktivnog izvođenja informacija je činjenica da se dedukcijom izvode informacije koje su nužno ispravne, jer predstavljaju logičku posljedicu već postojećih informacija. Za razliku od toga, induktivno izvedene informacije su ispravne samo u kontekstu primjera iz kojih su generalizirane, što ne garantira općenitu ispravnost. U svjetlu baza podataka, postupak dedukcije kao rezultat daje dokazive tvrdnje o stvarnom svijetu, pod pretpostavkom da baza podataka korektno opisuje stvarnost. S druge strane, rezultat postupka indukcije su

tvrdnje koje imaju potvrdu u bazi podataka, ali nisu nužno istinite u stvarnosti. Naravno, uzrok tome je činjenica da je samo manji dio relevantne stvarnosti prikazan bazom podataka. Stoga je jedan od najvažnijih aspekata strojnog učenja izdvajanje onih tvrdnji za koje je osim podrške u bazi podataka izgledno da će se pokazati točnima i u stvarnosti.
Zbog velikog broja potencijalnih pravilnosti i potrebe njihovog validiranja u odnosu na bazu podataka, indukcija je sa stanovišta računalne složenosti bitno složenija od dedukcije.

Standardne tehnike strojnog učenja pokušavaju pronaći i opisati strukturne pravilnosti u podacima. Prevladavaju dva osnovna tipa takvih postupaka [25]. U *klasifikacijskom učenju*, koristeći skup već klasificiranih primjera (učenje s učiteljem ili nadzirano učenje) algoritam pokušava naučiti način klasifikacije novih primjera, tj. konstruirati opis svake od klasa. U *asocijacijskom učenju* primjeri nisu klasificirani (učenje bez učitelja ili nenadzirano učenje), a cilj postupka je uočiti implicitne odnose koji postoje među podacima. Oba pristupa su iterativna: u cilju ostvarivanja vjerodostojnih rezultata, uzastopno se konstruiraju različiti modeli i ocjenjuju njihove performanse. Ovisno o korištenoj tehnici modeliranja, rezultirajući model može biti prediktivnog ili deskriptivnog karaktera. Prediktivni modeli prvenstveno omogućuju prognoziranje ishoda u novim situacijama, dok deskriptivni modeli osim toga pružaju i uvid u strukturu otkrivenog znanja.

1.3. ODNOS S DRUGIM ZNANSTVENIM PODRUČJIMA

Inteligentna analiza podataka je relativno mlada disciplina, koja svoje korijene vuče iz više različitih područja znanosti. Tako i danas njen razvoj profitira iz istraživanja u nekoliko dobro utemeljenih područja [27]:

- *Strojno učenje*
  Osim značajnih teoretskih rezultata, područje strojnog učenja ima dugu tradiciju osmišljavanja i eksperimentiranja s formalizmima za prikaz znanja i različitim strategijama pretraživanja. Iz ovog područja potječu mnoge tehnike modeliranja pravilnosti u podacima. Njihov osnovni oblik često je usmjeren na pronalaženje determinističkih odnosa u manjim skupovima podataka. Međutim, većina ovih metoda može se prilagoditi većim skupovima podataka i nadograditi postupcima tretiranja šuma u podacima. Ove i druge prilagodbe omogućavaju njihovu primjenu u inteligentnoj analizi podataka, gdje je većina odnosa probabilističkog, a ne determinističkog karaktera.
- *Statistika*
  Značajan utjecaj na razvoj inteligentne analize podataka ima i statistika. Osim statistički orijentiranih tehnika modeliranja, rezultati i metode s područja statistike se primjenjuju u gotovo svim fazama standardnog procesa inteligentne analize podataka.
  Statistički pristup modeliranju pravilnosti je prirodan u uvjetima velikih količina podataka koji u pravilu sadrže određenu dozu šuma. Probabilistički oblik rezultata također odgovara karakteru traženih pravilnosti. Međutim, statistika i inteligentna analiza podataka imaju ponešto drugačiji pristup. Tradicionalno, statistika se više bavi testiranjem već postavljenih hipoteza, dok je inteligentna analiza podataka usmjerena na samu konstrukciju hipoteza. Za to su osim tehnika

validacije hipoteza potrebni i algoritmi pretraživanja prostora potencijalnih hipoteza.

Proces inteligentne analiza podataka osim modeliranja sadrži i druge faze u kojima se klasične statističke metode obilno koriste. Uklanjanje irelevantnih i redundantnih atributa, detekcija šuma u podacima, modeliranje numeričkih atributa, testiranje pretjerane prilagodbe modela podacima za učenje, evaluacija konstruiranih modela – sve su to kandidati za direktnu primjenu standardnih statističkih postupaka.

- *Napredna sučelja baza podataka*
  Inteligentna analiza podataka podrazumijeva korištenje glomaznih skupova podataka. Dakako, velike količine analiziranih podataka predstavljaju poteškoću u primjeni tehnika modeliranja pravilnosti u podacima. Naime, pri pretraživanju prostora hipoteza formuliraju se različite hipoteze koje je potrebno validirati na podacima za učenje. Na velikim skupovima podataka to zna biti vremenski vrlo zahtjevno, pa su bilo kakva poboljšanja na tom polju dobrodošla.
  Zbog toga se prostor za unapređenje performansi tehnika inteligentne analize podataka može tražiti u sučelju prema bazi podataka, koja predstavlja osnovni izvor podataka. Primjerice, način korištenja priručne memorije (engl. *cache*) unutar baze podataka može bitno ubrzati validaciju hipoteza. Opravdanim se pokazao razvoj posebnih postupaka "keširanja" unutar baze podataka koji tijesno surađuju s algoritmom pretraživanja prostora hipoteza. Jedna od poželjnih karakteristika baze podataka je i mogućnost paralelizacije izvršavanja upita, o čemu treba voditi računa prilikom konstrukcije samih upita.
- *Raspoznavanje uzoraka*, *procesiranje signala*, *ekspertni sustavi*, kao i neka druga područja također su doprinijela razvoju inteligentne analize podataka.

1.4. PRIMJENE INTELIGENTNE ANALIZE PODATAKA

U posljednje vrijeme evidentno je rastuće zanimanja za područje inteligentne analize podataka, kako u istraživanjima tako i u praktičnoj primjeni. Pokretačka snaga koja je uzrokovala takav zamah je područje ekonomije, odnosno primjena tehnika inteligentne analiza podataka u poslovnim sustavima. Poslovne primjene i dalje snažno podržavaju razvoj inteligentne analize podataka upravo zbog značajnog mjesta kojeg su takvi alati zauzeli unutar poslovnih organizacija. Alati inteligentne analize podataka postaju neophodan dio današnjih sustava za potporu poslovnom odlučivanju, gdje iz postojećih sirovih podataka pribavljaju kvalitetno, strukturalno znanje o raznim aspektima poslovnih procesa [14]. Kvalitetno informiranje upravljačke strukture omogućuje donošenje racionalnijih odluka, pa time i uspješnije vođenje poduzeća.
Zbog svoje često strateške važnosti, rezultati praktične primjene metoda inteligentne analize podataka u pravilu su vrlo povjerljivog karaktera. Iako konkretni rezultati primjene nisu široko dostupne informacije, ne krije se činjenica da se upotrebljavaju i da donose rezultate.

1.4.1. Područja primjene

Trend rastuće popularnosti metoda inteligentne analize podataka zahvatio je gotovo sva područja ljudske djelatnosti. Evo kraćeg pregleda različitih djelatnosti sa

najčešćim problemima koji se rješavaju primjenom metoda inteligentne analize podataka [37], [40].

Bankarstvo i osiguranje
- razvoj prediktivnih modela i modela ocjene rizika za financijske institucije, npr. analiza kreditnih sposobnosti klijenata, analiza rizičnosti osiguranika

Investicijsko poslovanje
- predviđanje kretanja tečajeva valuta
- predviđanje promjena cijena dionica i drugih vrijednosnih papira
- analiza kretanja tržišnih segmenata

Detekcija prijevara
- detekcija prijevara i predikcija tipičnog korištenja kreditnih kartica u trgovinama
- otkrivanje prijevara u korištenju kartica kod transakcija u Internet trgovinama
- otkrivanje neovlaštenih upada u telekomunikacijske i računalne mreže

Znanost
- medicina: analize simptoma bolesti radi podrške u dijagnostici
- kemija: analiza kemijskog sastava i strukture složenih molekula
- biologija: istraživanja ljudskog genoma
- astronomija: analiza podataka prikupljenih teleskopima

Biotehnologija i farmaceutska industrija
- razvoj specifičnih rješenja (algoritama, modela, vizualizacijskih paketa) za farmaceutska i biotehnološka poduzeća, npr. metode za analizu sekvenci DNA, otkrivanje novih lijekova

Donošenje odluka u realnom vremenu
- integracija analitičkih postupaka u transakcijske sustave koji bilježe događaje u realnom vremenu
- detekcija problema i kvarova u industrijskim postrojenjima
- praćenje i optimizacija tehnoloških procesa; pokretanje predviđenih akcija
- automatizirani sistemi za interakciju s klijentima

Telekomunikacije
- analiza postojećih i osmišljavanje novih produkata i usluga
- analiza prelazaka korisnika konkurenciji

Ljudski resursi
- uparivanje potreba poslodavaca i kandidata, odabir kandidata s najboljim referencama za potrebe poslodavca

Upravljanje odnosima s klijentima
- integriranje i analiza podataka o klijentima s Interneta i drugih, tradicionalnih izvora podataka radi njihovog privlačenja, zadržavanja i povećanja profitabilnosti

Marketing
- kreiranje, optimiranje i realizacija marketinških kampanja
- usmjeravanje izravnih marketinških akcija na osnovu analize dosadašnje prodaje
- analiza svih interakcija s kupcem, te primjena modela u realnom vremenu radi stvaranja ciljanih ponuda kupcu

Prodaja u velikim trgovačkim centrima
- podrška za planiranje buduće potražnje, sezonskih rasprodaja, nabave proizvoda
- profiliranje segmenata kupaca: analiza navika, razumijevanje i predikcija ponašanja kupaca
- analiza potrošačkih košarica, optimiranje cijena i prodajnog prostora

- analiza prodajnih kanala, modeliranje marketinških promocija

Elektroničko trgovanje

- razvoj alata za razumijevanje, otkrivanje i interakciju sa kupcima
- analiza posjetitelja WEB-stranica, upravljanje marketinškim akcijama na WEB-u
- analiza ponašanja kupaca u elektroničkom kupovanju te automatsko generiranje personaliziranih, ciljanih marketinških poruka za različite segmente kupaca

Analiza WEB-a

- analiza navigacije posjetitelja WEB stranicama
- segmentiranje posjetitelja stranica, automatsko kreiranje personaliziranog sadržaja prema profilu posjetitelja stranice
- pronalaženje i prikupljanje relevantnih informacija na WEB-u prema zadanim kriterijima

1.4.2. Izgradnja profila korisnika usluga

U današnjem visoko kompetitivnom poslovnom okruženju globalne konkurencije, ekonomija u fokus stavlja korisnika i razinu usluge koja mu se pruža. Na razini proizvoda teško je graditi konkurentsku prednost, jer konkurencija lako može kopirati svojstva proizvoda, a vrijedi i obrat. Dugoročna strategija konkurentske prednosti može se graditi samo na vrijednostima koje je teško kopirati, a najvažnija takva vrijednost je upravo znanje. U korisnički orijentiranoj ekonomiji, znanje o korisnicima usluga predstavlja jedan od osnovnih preduvjeta za stjecanje i održavanje značajnog tržišnog udjela. Inteligentna analiza podataka je alat koji omogućava pronalaženje pravilnosti u ponašanju korisnika usluga, bolje razumijevanje njihovih motiva i predviđanje budućeg ponašanja, te u konačnici i iskorištavanje otkrivenih spoznaja o njihovim navikama i obilježjima [5].
Iz ovih razloga, analiza profila korisnika usluga je najaktivnije područje primjene inteligentne analize podataka u poslovnom svijetu. Analiza ponašanja klijenata je interesantna u svim područjima djelatnosti koja uključuju odnose s klijentima [6], što je vidljivo i iz prethodnog pregleda primjene metoda inteligentne analize podataka.
Značajan dio transakcijskih baza poslovnih subjekata nastaje bilježenjem interakcija s korisnicima usluga. Koliko god je nemoguće zamisliti normalno funkcioniranje poslovnog subjekta bez transakcijske baze podataka, tek primjena inteligentne analize podataka otkriva njen pravi potencijal. Dok transakcijska baza podržava operativne poslove, informacije dobivene inteligentnom analizom transakcijskih podataka su iznimno vrijedan resurs u donošenju strateških poslovnih odluka. Kvalitetne informacije su podloga za donošenje racionalnijih odluka i proaktivno vođenje poduzeća. Omogućuju anticipaciju budućnosti predviđanjem različitih scenarija, te adekvatnu pripremu za različite situacije.

Danas su poduzeća pretrpana podacima, dok im s druge strane nedostaje korisnih informacija. Tipično poduzeće analizira samo 10% prikupljenih podataka. Inteligentna analiza podataka nudi način kako iskoristiti preostalih 90%. Od relativno pasivne gomile podataka, godinama stvarane transakcijske baze poslovnih subjekata postaju dragocjen resurs, koji s razvojem tehnika inteligentne analize podataka sve više dobiva na vrijednosti.
Zbog toga su osim rezultata primjene inteligentne analize podataka i sami primjeri konkretnih transakcijskih podataka iz stvarnog svijeta rijetko javno dostupni. To

donekle otežava neovisan razvoj i eksperimentiranje sa stvarnim podacima, pa specifično razvijana rješenja nisu rijetkost.
Tijekom vremena, u okvirima analize profila korisnika usluga iskristaliziralo se nekoliko tipičnih problema, zajedničkih za većinu korisnički orijentiranih djelatnosti.

- *Zadržavanje klijenata*
  Nalaženje novih klijenata je višestruko skuplje od zadržavanja postojećih. Ako poduzeće uspije smanjiti odlazak klijenata konkurenciji za samo 10%, može značajno povećati svoju zaradu. Zbog toga je identifikacija nezadovoljnih klijenata koji bi mogli prijeći konkurenciji važan problem. U prosjeku se manje od 5% klijenata direktno žali na lošu kvalitetu proizvoda ili usluga. Uobičajeno ponašanje je izbjegavanje konkretnog proizvoda, odnosno prelazak na slične konkurentske proizvode.
  Dakle, potrebno je pronaći posredne načine identifikacije klijenata s većom vjerojatnošću prelaska konkurenciji. Pod pretpostavkom postojanja podataka o bivšim klijentima koji su prešli konkurenciji, tehnikama inteligentne analize podataka moguće je konstruirati općeniti opis takvih klijenata, kao i onih koji će vjerojatno ostati lojalni.
  Primjenom takvog opisa na podatke o postojećim korisnicima izdvaja se skupina rizičnih u smislu prelaska konkurenciji, te se oni mogu posebno tretirati kako bi ih se zadržalo uz tvrtku.
- *Identifikacija ključnih klijenata*
  Nije rijedak slučaj da je samo 20% ukupnog broja klijenata odgovorno za više od polovice profita poduzeća. Takve klijente je bitno zadržati, ali i privući nove klijente sličnih karakteristika. Izdvajanje konkretnih ključnih klijenata nije složen posao. Posao inteligentne analize podataka je izgraditi općeniti profil ključnog klijenta, odnosno odrediti koja svojstva ključnih klijenata ih razlikuju od ostalih klijenata. Na osnovu takvog općenitog opisa ključnog klijenta mogu se realizirati novi proizvodi ili ciljane marketinške kampanje usmjerene na privlačenje sličnih klijenata.
- *Procjena rizičnosti klijenta*
  Procjena rizika povezanog s klijentom je posebno značajna u financijskoj industriji. Procjena i upravljanje rizicima je temeljna zadaća osiguratelјnih društava, a o uspješnosti na tom polju direktno ovisi i uspjeh poduzeća. Osiguravajuće kuće imaju najizraženija potrebu za procjenom rizičnosti klijenta, jer nastoje ostvariti što više polica osiguranja za koje se neće dogoditi osigurani slučaj.
  Banke također nastoje smanjiti rizičnost svojih ulaganja. Kad su u pitanju odnosi s klijentima, tu se prvenstveno radi o procjeni njihove kreditne sposobnosti prilikom odobravanja kredita. Kartičarskim kućama je također cilj osigurati urednu naplatu potraživanja od klijenata, pa je jedan od osnovnih kriterija prilikom izdavanja kreditne kartice procjena boniteta klijenta.
  Profil rizičnih klijenata može se konstruirati tehnikama inteligentne analize podataka. Analizirani podaci trebaju uključivati klijente za koje se ispostavilo da nisu poželjni u smislu pridruženog rizika za poslovanje, kao i poželjne klijente. Rezultati se mogu koristiti za odbijanje nepoželjnih klijenata ili za korekciju cijene usluge koja im se pruža.

- *Ciljane marketinške promocije*

  Isti proizvod neće pobuditi jednako zanimanje svih klijenata. Dio proizvoda je namijenjen širokoj potrošnji, a dio se osmišljava obzirom na unaprijed zadanu ciljnu skupina korisnika. Međutim, u stvarnosti struktura zainteresiranih klijenata može biti znatno drugačija od planirane. Stoga je i u slučaju definirane ciljne skupine korisnika interesantno analizirati svojstva stvarnog odziva.

  Na osnovu ograničenog skupa podataka o inicijalnom odzivu klijenata, tehnike inteligentne analize podataka mogu formulirati općeniti opis tipičnih klijenata s interesom za promatrani proizvod. On može poslužiti za određivanje ili korekciju fokusa marketinške promocije na potencijalno zainteresirane klijente. Time se u pravilu mogu znatno smanjiti troškovi promocije, uz istodobno povećanje odziva klijenata. Upotreba inteligentne analize podataka u usmjeravanju marketinških promocija posebno se često koristi u području izravnog marketinga.

- *Osmišljavanje novih proizvoda*

  U nekim industrijskim granama proizvodi su vrlo fleksibilnog karaktera i lako promjenjivih svojstava. Ilustrativni primjeri takvih proizvoda i usluga mogu se naći u financijskom i telekomunikacijskom sektoru.

  Fleksibilnost svojstava je osnovni uzrok postojanja raznovrsne palete u osnovi sličnih proizvoda. Cilj svakog od konkurentskih poduzeća je stvoriti proizvode po mjeri klijenta, kojima će zadržati postojeće i privući nove klijente.

  Inteligentna analiza podataka nudi alate kojima se mogu detektirati karakteristična ponašanja klijenata koja ukazuju na prostor za plasiranje novih ili preoblikovanje postojećih proizvoda i usluga. Rezultati analize mogu sugerirati i grupiranje proizvoda i usluga u klijentima atraktivne pakete. Analizom se mogu identificirati ustaljene navike korisnika, te se korekcijom cijena u skladu s navikama korisnika i njihovom kupovnom moći može povećati profitabilnost poslovanja.

- *Analiza potrošačkih košarica*

  Cilj analize potrošačkih košarica je utvrđivanje grupa proizvoda koji se često kupuju istodobno. Tendencija pojavljivanja istih elemenata u većem broju transakcija može se otkriti primjenom asocijacijskih tehnika inteligentne analize podataka.

  Ovakva saznanja mogu se iskoristiti na više načina. Jedan od njih je poticanje neplanirane kupnje odobravanjem popusta na skupnu kupnju odabranih proizvoda. Osim toga, politika prigodnih popusta može se planirati na način da se popust ograniči na samo jedan od proizvoda koji se često zajedno kupuju. Adekvatnim razmještajem proizvoda moguće je dulje zadržati kupce u prodajnom prostoru, ili im pak omogućiti bržu kupnju.

  Podaci o kupovinama dobivaju dodatnu vrijednost ako je moguće identificirati kupce koji su ih obavili. Povezivanje svih kupnji pojedinačnih kupaca omogućuje analizu ponašanja kupaca kroz vrijeme. Tako uočene pravilnosti mogu se iskoristiti u osmišljavanju precizno usmjerenih posebnih ponuda ili marketinških akcija, a sve u cilju povećanja prodaje. Rastući broj potrošačkih kartica trgovačkih lanaca zorno govori o svijesti da vrijednost

identifikacije pojedinačnih kupaca višestruko nadmašuje vrijednost tako odobrenih popusta.

1.4.3. Inteligentna analiza podataka i etika

Kada se radi o primjeni inteligentne analize podataka u izgradnji profila korisnika usluga, rezultati analize često se koriste za određeni vid diskriminacije. Primjerice, modeli pravilnosti konstruirani tehnikama inteligentne analize podataka pomažu u odlučivanju kome će se odobriti zajam a kome neće, kome će se ponuditi posebne pogodnosti a kome neće i slično. Ova pojava je najizraženija u osiguranjeljnoj djelatnosti, gdje je procjena rizičnosti klijenta direktno utječe na cijenu proizvoda. Moglo bi se reći da osiguravajuća društva rade diskriminaciju među klijentima na osnovu određenih stereotipa, koji su doduše statistički opravdani. Dakako, to nipošto ne znači da će konkretni klijent odgovarati stereotipu u koji je svrstan.
Podaci o korisnicima koje prikupljaju poslovni subjekti često sadrže podatke osobnog karaktera. Korištenje osobnih podataka u postupku inteligentne analize može povlačiti određena etička pitanja pri primjeni dobivenih rezultata. Naime, diskriminacija po određenim osnovama (rasnim, spolnim, vjerskim i slično) ne samo da je neetična, već gotovo svugdje i protuzakonita. Međutim, etičnost i legalnost primjene takvih modela može ovisiti i o domeni problema koji se analizira. Primjerice, nitko neće osporiti upotrebu informacije o spolu u svrhu medicinske dijagnostike.
Dakle, problem upotrebe osobnih podataka nije jednostavan. Stoga korisnici alata inteligentne analize podataka moraju biti svjesni etičke dimenzije problema koji istražuju i odgovorno se odnositi prema podacima osobnog karaktera koji su im dostupni. Ustaljena praksa koja je ponegdje pretočena i u zakonske norme traži da se pri prikupljanju osobnih podataka ljudi obavijeste na koji način i u koje svrhe će se podaci koristiti. Osim toga, ljudima treba objasniti koje su eventualne posljedice uskraćivanja traženih podataka, komu će sve podaci biti dostupni i kako će se štititi njihova tajnost, koliko dugo će se podaci čuvati, te imaju li pravo naknadno tražiti izuzimanje svojih podataka.
Pri eventualnom naknadnom korištenju podataka koji su inicijalno prikupljeni s drugom svrhom svakako treba voditi računa o usklađenosti primjene s namjenama za koje su ljudi dali svoj pristanak. To je u pravilu stvar prosudbe, jer je često nemoguće ili nepraktično tražiti eksplicitnu potvrdu od samih ljudi na koje se podaci odnose.

U konačnici, puno toga ovisi o odgovornosti ljudi koji koriste osjetljive osobne podatke. Inteligentna analiza podataka je ipak samo alat koji pomaže u donošenju teških odluka. Ljudi su ti koji interpretiraju rezultate analize, i u skladu s ostalim svojim saznanjima odlučuju kakve akcije treba poduzeti. One mogu implicirati određeni oblik diskriminacije, pa uz ostale elemente koji su utjecali na odluku treba s etičkog stanovišta sagledati i modele proizvedene inteligentnom analizom podataka.

1.4.4. Važnost razumijevanja postupaka i rezultata

Inteligentna analiza podataka nije tehnologija koju se može slijepo primijeniti i očekivati dobre rezultate. Produktivna primjena tehnika inteligentne analize podataka zahtijeva razumijevanje osnovnih principa njihovog rada.

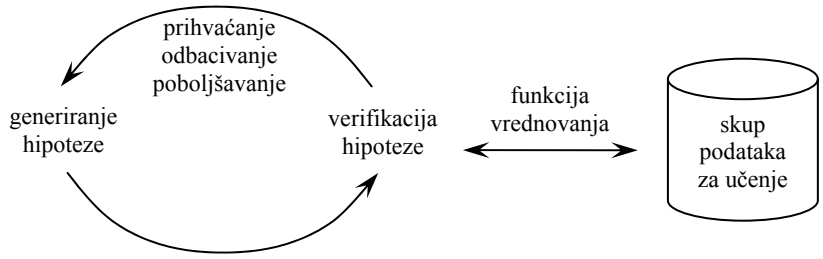
Područje inteligentne analize podataka se još uvijek intenzivno razvija, osmišljavaju se novi i usavršavaju postojeći postupci i tehnike. Kod primjene na konkretni problem, korisnost krajnjih rezultata u velikoj mjeri ovisi o odabiru tehnike modeliranja podataka i načinu njene primjene. Usprkos tome, vrlo rijetko je odmah jasno koje tehnike najbolje odgovaraju danoj situaciji. Budući da različiti problemi traže različite pristupe i tehnike, potrebno je poznavati svojstva pojedinih tehnika i sagledati raspon mogućih rješenja.
Rezultat primjene inteligentne analize podataka je model kojim su izražene pravilnosti uočene u promatranim podacima. U praksi se nebrojeno puta potvrdilo nepisano pravilo da su modeli onoliko dobri koliko i osoba koja ih interpretira. Zbog toga čak i sami korisnici modela trebaju imati osnovnu predodžbu o načinu njihova nastajanja, kako bi mogli cijeniti snagu, ali i ograničenja inherentna tehnologiji inteligentne analize podataka. Odluke ne donose modeli, već ljudi. Da bi mogli kvalitetno doprinijeti donošenju odluka, izvedene modele treba razumjeti.
Dakle, poželjno je da svi sudionici procesa inteligentne analize podataka imaju određenu razinu znanja o korištenoj tehnologiji, primjerenu svojoj ulozi u procesu. Voditelji analize trebaju detaljno poznavati same tehnike i njihova svojstva, a korisnici oblik i svojstva dobivenih rezultata.

1.5. STRUKTURA RADA

Struktura ovog rada također je vođena idejom o potrebi razumijevanja postupaka inteligentne analize podataka radi njihove uspješne primjene. Stoga su u prvom dijelu izložene osnovne postavke na kojima se zasnivaju postupci inteligentne analize podataka, te potom opisane različite tehnike modeliranja i njihova svojstva. Zaključno, u završnom dijelu prikazan je odabir i primjena opisanih postupaka u analizi stvarnog problema s područja izgradnje profila korisnika usluga.
Konkretno, nakon prvog uvodnog poglavlja, drugo poglavlje prikazuje strojno učenje u svjetlu pretraživanja prostora hipoteza. U trećem poglavlju ukratko je opisan proces inteligentne analize podataka i faze od kojih se sastoji. Slijedećih nekoliko poglavlja posvećeno je detaljnijem prikazu tehnika koje se koriste u pojedinim fazama procesa inteligentne analize podataka. Četvrto poglavlje se bavi fazom prikupljanja i pripreme podataka, s naglaskom na različite tehnike pročišćavanja i transformacije podataka. U fokusu petog poglavlja su tehnike modeliranja pravilnosti u podacima i njihova svojstva. Ovo je najopširnije poglavlje, sukladno središnjoj ulozi faze modeliranja u cjelokupnom procesu inteligentne analize podataka. Tema šestog poglavlja je faza evaluacija otkrivenog znanja. Preciznije, u tom poglavlju su opisane različite mjere i metode za evaluaciju klasifikacijskih modela. Konačno, sedmo poglavlje posvećeno je praktičnoj primjeni postupaka inteligentne analize podataka. Konkretan problem iz domene osiguravajućih društava analizira se primjenom kombinacije tehnika probabilističkog i deskriptivnog modeliranja. Na tim osnovama je izgrađen i programski produkt namijenjen problemima s izraženim nedeterminističkim obilježjima. Osmo poglavlje je zaključak u kojem su sažeti osnovni naglasci rada.

2. Strojno učenje

Strojno učenje predstavlja automatizaciju procesa induktivnog učenja. Za razliku od kognitivnih bića koja uče iz opažanja svoje okoline, kod strojnog učenja nema direktne interakcije između sustava i njegovog okruženja. Ulazne informacije za proces strojnog učenja predstavlja unaprijed pripremljen i adekvatno kodiran skup opažanja koji se naziva *skup podataka za učenje*. Cilj procesa strojnog učenja je pronalaženje skrivenih pravilnosti i uzoraka koji postoje u skupu podataka za učenje. U stvari, strojno učenje je iterativan postupak u kojem se konstruiraju opisi tih pravilnosti. Samo konstruiranje traženog opisa može se promatrati kao postupak pretraživanja u kojem se između svih opisa koje je moguće konstruirati traži onaj koji najbolje odgovara pravilnostima u skupu podataka za učenje. Slika 2 prikazuje strojno učenje kao iterativno pretraživanje skupa hipoteza.



Slika 2: Strojno učenje kao proces pretraživanja

Proces počinje odabirom inicijalne hipoteze (tj. inicijalnog opisa). Hipoteza se verificira upotrebom *funkcije vrednovanja*. Funkcija vrednovanja uporabom statističkih tehnika daje ocjenu ispravnosti hipoteze u odnosu na skup podataka za učenje. Na osnovu toga hipoteza se može prihvatiti, odbaciti ili iterativno poboljšavati dok se ne pronađe hipoteza s adekvatnom potvrdom u skupu podataka za učenje. Hipoteza se modificira primjenom unaprijed definiranih operacija, npr. sužavanjem uvjeta hipoteze (specijalizacija hipoteze) ili njihovim proširivanjem (generalizacija hipoteze). Budući da način modificiranja hipoteze usmjerava proces pretraživanja, modifikacija hipoteza se kontrolira određenim uvjetima ili primjerenom heuristikom.

2.1. OSNOVNI POJMOVI

Neke od gore spomenutih pojmova potrebno je preciznije definirati, te uvesti standardne oznake u svrhu koncicznijeg izražavanja.
Kao što je već rečeno, ulaz u proces strojnog učenja predstavljaju opažanja zapisana u nekom strojno čitljivom obliku. U tu svrhu se odabire fiksan, unaprijed definiran skup obilježja, tzv. *atributa*, koja će karakterizirati svako opažanje. Svaki atribut može poprimiti jednu od unaprijed određenih vrijednosti. Prema tome, svako opažanje se bilježi kao skup vrijednosti koje poprimaju pojedini atributi.

Definicija

Neka je $A=\{A_1, \ldots, A_n\}$ skup atributa sa domenama $Dom(A_1), \ldots, Dom(A_n)$.
Kartezijev produkt domena $U=Dom(A_1)\times \ldots \times Dom(A_n)$ naziva se *univerzum*.

Konačan podskup S univerzuma U naziva se *skup podataka za učenje*, $S \subseteq U$. Svaki element o skupa podataka za učenje naziva se *primjer*, $o \in S$.

Postoji nekoliko osnovnih tipova atributa koji imaju različit tretman u algoritmima strojnog učenja. Tip atributa određuje struktura domene kojoj atribut pripada.

Nominalni atributi pripadaju domeni koja se sastoji od predefiniranog, konačnog skupa nezavisnih simbola. Pretpostavlja se da među različitim vrijednostima domene nema nikakvih relacija, tj. ne postoji uređaj nad elementima domene.

Linearni atributi odgovaraju totalno uređenim domenama. Budući da postoji totalni uređaj među elementima domene, moguće je govoriti o intervalima vrijednosti. Postoje tri podkategorije linearnih atributa: *ordinalni*, *intervalni* i *razmjerni*. Domene ordinalnih atributa ne podrazumijevaju postojanje bilo kakve operacije nad vrijednostima domene. Za razliku od njih, kod intervalnih atributa je definirana operacija zbrajanja vrijednosti (ali ne nužno i množenja). Takav atribut je npr. temperatura: ima smisla govoriti o razlici temperatura, dok pojmovi tipa "dvostruko više" nemaju značenje. Konačno, kod razmjernih atributa i operacija zbrajanja i operacija množenja imaju razumne interpretacije (npr. realni brojevi).

Parcijalno uređenim atributima odgovaraju domene nad kojima je definiran parcijalni uređaj. Dakle, svi elementi domene ne moraju biti usporedivi, ali postoji pojam maksimuma. Parcijalno uređenim atributima se najčešće opisuju hijerarhije, gdje roditelj označava općenitiji pojam od djece.

U praksi, velika većina algoritama strojnog učenja koristi samo nominalne i ordinalne atribute. Ordinalni atributi se gotovo uvijek izražavaju brojevima, pa se uvriježio naziv *numerički atributi*. Ostali tipovi atributa rjeđe se koriste i uglavnom su ograničeni na specijalizirana područja primjene.

Kad je riječ o učenju s učiteljem, potrebno je u skupu podataka za učenje definirati *klase* ili *ciljne pojmove*. Bez smanjenja općenitosti može se pretpostaviti da skup podataka za učenje sadrži jedan ili više atributa koji jednoznačno određuju klasu primjera. Takvi atributi nazivaju se *prognozirani atributi*, a ostali atributi skupa podataka za učenje nazivaju se *prognostički atributi*. Klasa primjera se definira uvjetima na prognozirane atribute.

Definicija

Neka je $cond_i$ uvjet izražen pomoću ograničenja na prognozirane atribute skupa podataka za učenje S. Klasa C_i je podskup skupa podataka za učenje S sastavljen od svih primjera koji zadovoljavaju uvjet $cond_i$:

$$C_i = \{o \in S \mid cond_i(o)\} .$$

Primjer o koji zadovoljava uvjet $cond_i$ (tj. $o \in C_i$) naziva se *pozitivni primjer* ili *instanca* klase C_i. Primjer koji ne zadovoljava uvjet $cond_i$ (tj. $o \notin C_i$) naziva se *negativni primjer* klase C_i.

Osnovni zadatak strojnog učenja je pronalaženje opisa za svaku od definiranih klasa. Naravno, opisi se trebaju sastojati samo od uvjeta na prognostičke atribute skupa

podataka za učenje. U idealnom slučaju, svi pozitivni primjeri klase trebali bi zadovoljavati opis klase, dok nijedan negativni primjer ne bi trebao zadovoljavati opis klase.
Sintaksa i semantika opisa klasa razlikuje se od algoritma do algoritma strojnog učenja, a ovisi o odabranom načinu prikaza znanja. Najčešće korišten način opisa klasa su klasifikacijska pravila, koja predstavljaju podskup selekcijskih uvjeta relacijske algebre.

Definicija

Neka je $A=\{A_1, \ldots, A_n\}$ skup prognostičkih atributa skupa podataka za učenje S.

Elementarni opis je formula oblika $A_{i_1}=c_{i_1} \wedge \ldots \wedge A_{i_m}=c_{i_m}$ takva da vrijedi

- $A_{i_j} \in A \;\; i \;\; k \neq l \Rightarrow A_{i_k} \neq A_{i_l}$,
- $c_{i_j} \in Dom(A_{i_j})$.

Opis je neprazna disjunkcija elementarnih opisa.

Opis klase u obliku klasifikacijskog pravila povezuje konstruirane opise s odgovarajućim klasama.

Definicija

Neka je sa D označen opis, a sa C klasa iz skupa podataka za učenje S nad univerzumom U. Za svaki element o univerzuma U koji zadovoljava opis D kaže se da je *prekriven* opisom D. Skup svih elemenata skupa $T \subseteq U$ koje prekriva opis D označava se sa $\sigma_D(T)$.
Klasifikacijsko pravilo je formula oblika

$$\forall o \in U : o \in \sigma_D(U) \rightarrow o \in C \;\; . \qquad (1)$$

Neformalno rečeno, pravilo je ispravno u odnosu na skup podataka za učenje ako opis prekriva sve pozitivne primjere i niti jedan negativni primjer klase, tj. kad je $\sigma_D(S)=C$. Treba primijetiti da opis klase nije jednoznačno definiran uvjetom ispravnosti pravila u odnosu na skup podataka za učenje. Međutim, iako je moguće konstruirati različita pravila koja su ispravna u odnosu na skup podataka za učenje, svako od tih pravila neće ispravno klasificirati dotad neviđene primjere. Upravo funkcija vrednovanja predstavlja mehanizam odabira kvalitetnijih pravila, tj. onih pravila za koja se drži da će pokazati bolje rezultate na dotad neviđenim primjerima.

Kada se radi o primjeni algoritama strojnog učenja u inteligentnoj analizi podataka, skup podataka za učenje je baza podataka (ili neki njen dio). Upravo je ova činjenica izvor nekih teškoća koje se susreću u procesu inteligentne analize podataka. Naime, primarna funkcija baza podataka nije osiguravanje potrebnih podataka za proces inteligentne analize podataka. Svojstva objekata iz stvarnog svijeta koja se bilježe u bazi podataka su odabrana prema potrebama temeljnih aplikacija kojima baza služi. To znači da atributi ili vrijednosti koje bi pojednostavnile proces inteligentne analize podataka nisu nužno prisutni u bazi podataka. Posebice, često se javljaju *neodređene vrijednosti atributa* – zapis u bazi koji bi trebao služiti kao primjer za učenje nema zabilježene vrijednosti svih atributa.

Osim toga, podaci u bazama podataka su u pravilu mjestimično prošarani greškama. Stoga, algoritmi koji se koriste u inteligentnoj analizi podataka ne trebaju očekivati pažljivo odabrane laboratorijske podatke, već moraju imati robusne mehanizme tretiranja šuma u kako u skupu podataka za učenje, tako i u podacima za klasifikaciju.
U konačnici, potrebno je voditi računa o obimnosti podataka pohranjenih u bazama podataka. U inteligentnoj analizi podataka skup podataka za učenje nerijetko može sadržavati i više stotina tisuća primjera. Budući da je pristup podacima u bazi vremenski relativno skupa operacija, česti upiti na bazu podataka mogu značajno degradirati performanse algoritma.

Dvije su osnovne pretpostavke vezane uz promatranje baze podataka kao skupa podataka za učenje.
Prva je takozvana *pretpostavka univerzalne relacije*: pretpostavlja se da su svi potrebni podaci sadržani u samo jednoj relaciji, tj. onaj dio baze podataka koji se koristi kao skup podataka za učenje predstavljen je jednom relacijom iz baze podataka. Zapisi te relacije predstavljaju primjere za učenje. Međutim, ova pretpostavka u stvari nije ograničavajuća – dobro je poznato da se bilo koja shema baze podataka ili bilo kakav upit mogu prikazati u obliku jedne univerzalne relacije. Pravi razlog uvođenja ove pretpostavke je pojednostavljena notacija i baratanje podacima.
Druga pretpostavka nije kozmetičke prirode, već zaista uvodi neka ograničenja. Ona implicira nezavisnost primjera za učenje, tj. nezavisnost zapisa u bazi podataka. Konkretno, pretpostavlja se da zapisi univerzalne relacije bilježe samo svojstva promatranih objekata, ali ne i veze među njima. Dakle, svaki promatrani objekt opisan je jednim zapisom u bazi podataka, a svakom zapisu odgovara jedan objekt. Ova pretpostavka ograničava primjenu inteligentne analize podataka na istraživanje odnosa među pojedinim svojstvima objekata, a isključuje istraživanje odnosa među različitim objektima.
U nekim razmatranjima se koriste još neke manje bitne pretpostavke. Jedna od takvih je i pretpostavka o konačnosti domena svih atributa. Općenito, ta pretpostavka nije točna za numeričke atribute. Međutim, konačnost skupa podataka za učenje implicira i konačan broj korištenih vrijednosti domene svakog atributa, pa se bez smanjenja općenitosti može promatrati konačan skup intervala nad beskonačnom domenom.

2.2. PROSTOR RJEŠENJA

Uz zadani skup podataka za učenje S i definirane klase C_i koje ga particioniraju, sustav za inteligentnu analizu podataka može konstruirati veliki broj opisa pojedine klase. Neki od tih opisa će biti bolji od drugih, u smislu da će bolje klasificirati dotad neviđene primjere. Bolje performanse na neviđenim primjerima znače da je opis kvalitetnije zabilježio (dotad nepoznate) pravilnosti koje postoje u podacima. Da bi se kvaliteta različitih opisa mogla uspoređivati, nužno je definirati neku mjeru za kvalitetu opisa. Uz definiranu mjeru kvalitete opisa, strojno učenje se svodi na problem pretraživanja: potrebno je pronaći najkvalitetniji opis klase u skupu svih opisa koje je moguće konstruirati.

Definicija

Neka je sa $\mathcal{D}$ označen skup svih opisa koje je moguće konstruirati u odabranoj metodi prikaza znanja ($\mathcal{D}$ se ponekad naziva *prostor hipoteza*).

Neka je sa O označen skup odabranih operacija nad opisima, a sa $f: \mathcal{D} \to \mathbb{R}$ funkcija vrednovanja opisa.

Tada uređenu trojku $(\mathcal{D}, f, O)$ nazivamo *prostor rješenja*.

Naravno, odabrana metoda prikaza znanja definira skup svih opisa $\mathcal{D}$. Njegova kardinalnost je u pravilu manja od kardinalnosti partitivnog skupa univerzuma U, pa u tom slučaju nije moguće opisati svaki pojedinačni podskup univerzuma. Svakom opisu $D \in \mathcal{D}$ odgovara podskup univerzuma $\sigma_D(U)$ kojeg opis prekriva. Izražajnijim metodama prikaza znanja odgovaraju brojniji skupovi svih mogućih opisa $\mathcal{D}$ sa finijom granulacijom podskupova univerzuma koje opisi prekrivaju.
Prema tome, izbor metode prikaza znanja direktno utječe na razinu detalja koju je moguće izraziti opisima. Iako se u prvi mah čini da su izražajnije metode prikaza znanja bolje, uz izražajnost dolazi i problem pretjerane prilagodbe podacima za učenje. Osim toga, one generiraju glomaznije skupove opisa $\mathcal{D}$, što može imati negativne posljedice na performanse pretraživanja i vrednovanja različitih opisa. Karakteristike problema nad kojim se vrši analiza podataka trebaju biti osnovni kriterij pri odabiru pravog omjera izražajnosti i performansi.

2.2.1. Operacije

Operacije kojima se u procesu strojnog učenja transformiraju opisi su uglavnom unarne, tj. operiraju nad jednim opisom. Mogu se podijeliti na operacije *generalizacije* i operacije *specijalizacije*. Operacijom generalizacije se "oslabljuje" opis, tj. novonastali opis prekriva više primjera od polaznog, dok operacije specijalizacije "jača" opis, tj. smanjuje broj primjera koje opis prekriva.

Definicija

Neka je sa $op: \mathcal{D} \to \mathcal{D}$ označena operacija nad opisima odabrane metode prikaza znanja. Operacija *op* je *operacija generalizacije* ako i samo ako vrijedi

$$\forall D \in \mathcal{D}: \sigma_D(U) \subseteq \sigma_{op(D)}(U) \ . \qquad (2)$$

Dakle, ako je primjer pokriven nekim opisom, bit će pokriven i svakom generalizacijom tog opisa. Prema tome, ako opis D pogrešno klasificira primjer o (tj. primjer je pokriven opisom D, ali ne pripada klasi C koja odgovara opisu), tada će svaka generalizacija tog opisa $op(D)$ također pogrešno klasificirati primjer o. Kaže se da generalizacija čuva neispravnost opisa, tj. ispravnost opisa na primjerima koje opis pogrešno klasificira ne može se popraviti generalizacijom opisa.
Cilj upotrebe operacija generalizacije u sustavima strojnog učenja je dobivanje općenitijeg opisa koji prekriva više pozitivnih primjera ciljne klase od polaznog opisa.

Definicija

Neka je sa $op : \mathcal{D} \to \mathcal{D}$ označena operacija nad opisima odabrane metode prikaza znanja. Operacija *op* je *operacija specijalizacije* ako i samo ako vrijedi

$$\forall D \in \mathcal{D} : \sigma_D(U) \supseteq \sigma_{op(D)}(U) \ . \qquad (3)$$

Definicija govori da je svaki primjer koji je pokriven specijalizacijom *op*(*D*) opisa *D* nužno pokriven i samim opisom *D*, ali obrat ne vrijedi. Stoga sa operacije specijalizacije u sustavima strojnog učenja koriste kako bi se iz primjera koje opis prekriva isključili negativni primjeri ciljne klase, te tako povećala ispravnost pravila.

U terminima opisa i operacija nad opisima, prostor rješenja se može promatrati kao usmjereni graf, gdje opisi čine čvorove grafa, a operacije lukove među čvorovima.

2.2.2. Funkcija vrednovanja

Funkcija vrednovanja $f : \mathcal{D} \to \mathbb{R}$ svakom opisu pridružuje realni broj koji odražava kvalitetu opisa. Dva su aspekta kvalitete opisa: *ispravnost* i *valjanost*. Ispravnost se definira u terminima primjera iz skupa podataka za učenje *S*: opis bi trebao ispravno klasificirati sve primjere iz skupa *S*. S druge strane, valjanost se odnosi na klasifikaciju dotad neviđenih primjera: od opisa se očekuje da dotad neviđene primjere također ispravno klasificira. Funkcija vrednovanja kombinira ova dva kriterija u svrhu izračunavanja ukupne kvalitete opisa. Radi boljeg razumijevanja mehanizama funkcije vrednovanja, potrebno je detaljnije pojasniti ove kriterije.

Valjanost. Osnovni problem strojnog učenja (i induktivnog učenja uopće) je činjenica da uz zadani skup podataka za učenje sustav može konstruirati više različitih opisa koji su ispravni u odnosu na primjere iz tog skupa. Međutim, to ne znači da će svi ti opisi biti jednako dobri i na dotad neviđenim podacima. Upravo je ispravnost klasifikacije dotad neviđenih opisa indikator valjanosti otkrivenog znanja – inače bi se strojno učenje svelo na puko memoriranje klasifikacije poznatih primjera.
Drugim riječima, osnovni problem je razlikovati stvarne pravilnosti koje su u osnovi promatranog problema od prividnih pravilnosti koje općenito ne vrijede, već se pojavljuju samo zbog ograničenog broja promatranih primjera. Bit problema leži u tome da se valjanost opisa općenito ne može direktno provjeriti – ne postoji način da se ispravnost opisa provjeri za sve moguće dotad neviđene situacije. Prema tome, potrebna je neka ocjena valjanosti opisa koja bi kvantificirala izglednost da će opis pravilno klasificirati neviđene primjere. Većina sustava strojnog učenja koristi neku formu *Ockhamove oštrice* [54]: što je opis jednostavniji, to su veći izgledi da će ispravno klasificirati nove primjere. Načelo koje stoji iza ovog pravila je prilično razumno: ukoliko postoji više objašnjenja nekog fenomena, ima smisla očekivati da upravo ono najjednostavnije odražava pravu prirodu fenomena.
Stoga bi mjera valjanosti trebala imati veće vrijednosti za jednostavnije opise, a jednostavnost opisa je pak moguće mjeriti (primjerice, dužinom opisa).

Ispravnost. Kao što je već spomenuto, ispravnost opisa se ogleda u klasifikaciji primjera iz skupa podataka za učenje: opis D klase C je ispravan ako prekriva sve pozitivne primjere klase i niti jedan negativni primjer, tj. ako je $\sigma_D(S)=C$. Međutim, tijekom iterativnog procesa konstrukcije opisa, sustav će nailaziti na mnoge opise koji nisu ispravni, ali su svejedno vrlo korisni. Oni predstavljaju trenutni doseg procesa konstrukcije opisa, te se njihovim daljnjim oblikovanjem formiraju novi, poboljšani opisi. Da bi se iz skupa opisa koji nisu ispravni mogao odabrati onaj koji najviše obećava u smislu daljnjeg poboljšavanja, potrebno je proširiti pojam ispravnosti. Umjesto kategoričkog suda o (ne)ispravnosti opisa, koriste se brojčane mjere stupanja ispravnosti nekog opisa.
Postoji više pokazatelja ispravnosti opisa u odnosu na neku klasu.

Definicija

Preciznost klasifikacije opisa D za klasu C se definira kao udio pozitivnih primjera u primjerima koje opis prekriva:

$$\textit{preciznost klasifikacije} = \frac{|\sigma_D(S) \cap C|}{|\sigma_D(S)|} \quad . \tag{4}$$

Prema tome, preciznost klasifikacije odgovara vjerojatnosti da opis ispravno klasificira primjere iz skupa podataka za učenje, tj. vjerojatnosti da primjer pokriven opisom zaista pripada odgovarajućoj klasi.

Definicija

Stupanj prekrivanja opisa D za klasu C definira se kao omjer pozitivnih primjera koje opis prekriva i svih pozitivnih primjera:

$$\textit{stupanj prekrivanja} = \frac{|\sigma_D(S) \cap C|}{|C|} \quad . \tag{5}$$

Dakle, stupanj prekrivanja se može promatrati kao vjerojatnost da je pozitivan primjer iz skupa podataka za učenje pokriven opisom.
Na osnovi ovih pokazatelja, razlikujemo nekoliko vrsti opisa:

- *Deterministički opisi*. Opis je deterministički ako mu preciznost klasifikacije iznosi 1. Tada svaki primjer pokriven opisom pripada klasi, tj. $C \supseteq \sigma_D(S)$. Prema tome, opis je dovoljan uvjet za klasu.
- *Potpuni opisi*. Opis je potpun ako mu stupanj prekrivanja iznosi 1. To znači da su svi pozitivni primjeri klase pokriveni opisom klase, tj, $C \subseteq \sigma_D(S)$. Drugim riječima, opis je nužan uvjet za klasu.
- *Ispravni opisi*. Potpuni deterministički opisi su ispravni opisi, tj. kod ispravnih opisa i preciznost klasifikacije i stupanj prekrivanja iznose 1. Tada je $C = \sigma_D(S)$, pa je opis nužan i dovoljan uvjet za klasu.

Analogno navedenim pokazateljima, mjera ispravnosti opisa D trebala bi pridružiti vrijednost 1 ispravnim opisima, dok bi opisima koji nisu ispravni pridruživala vrijednosti manje od 1. Piatetsky-Shapiro [45] predlaže principe za konstrukciju funkcije koja svakom opisu pridružuje broj koji odražava njegovu ispravnost.

Principi se formuliraju u terminima skupa $\sigma_D(S)$, klase C, te njihovog presjeka $\sigma_D(S) \cap C$. Ako sa f_c označimo traženu funkciju, ona treba zadovoljavati tri zahtjeva:

1. $f_c(D) = 0$ ako su $\sigma_D(S)$ i C statistički nezavisni ($\sigma_D(S)$ i C su statistički nezavisni ako je vjerojatnost da odabrani primjer $e \in \sigma_D(S)$ ujedno pripada klasi C jednaka vjerojatnosti da bilo koji primjer $e \in S$ pripada klasi C),
2. f_c monotono raste sa $|\sigma_D(S) \cap C|$, uz konstantan $|\sigma_D(S)|$,
3. f_c monotono pada sa $|\sigma_D(S)|$, uz konstantan $|\sigma_D(S) \cap C|$.

Najjednostavnija funkcija koja zadovoljava ove uvjete je

$$f(D) = \left|\sigma_D(S) \cap C\right| - \frac{\left|\sigma_D(S)\right| \left|C\right|}{\left|S\right|} \quad . \tag{6}$$

Intuitivno, ova funkcija se može promatrati kao razlika broja primjera koje opis ispravno klasificira i očekivanog broja kada bi C i $\sigma_D(S)$ bili nezavisni.

Kao što je na početku rečeno, funkcija vrednovanja odražava ukupnu kvalitetu opisa razmatrajući i valjanost i ispravnost opisa. Funkcija vrednovanja može uzeti u obzir i neke druge čimbenike, kao što su cijena evaluacije opisa, ili cijena mjerenja vrijednosti atributa uključenih u opis.
Postoji više načina na koji funkcija vrednovanja može kombinirati različite kriterije u svrhu izračunavanja ukupne kvaliteta opisa. Često se koristi težinska suma različitih kriterija: svakom kriteriju se pridružuje realan broj (težina) koji indicira njegovu relativnu važnost u usporedbi s drugim kriterijima. Odabir optimalnih težina predstavlja oblik finog ugađanja sustava.
Osim težinske sume, susreće se i funkcional leksikografske evaluacije [43], kod kojeg je potrebno prema važnosti poredati sve kriterije. Evaluacija kreće od najvažnijeg kriterija, zadržavajući samo najbolje opise u okviru određene tolerancije. Slijedno evaluiranje ostalih kriterija u konačnici izdvaja najbolji opis.

Primjena funkcije vrednovanja na skup svih opisa koje je moguće konstruirati u odabranoj metodi prikaza znanja rezultira krajolikom u kojem vrhovi označavaju opise visoke kvalitete, a doline opise niske kvalitete.

2.3. ALGORITMI PRETRAŽIVANJA

Uz definiranu funkciju vrednovanja, strojno učenje se svodi na pretraživanje prostora rješenja u potrazi za opisom koji maksimizira funkciju vrednovanja. Za istraživanje prostora rješenja mogu se koristiti različiti algoritmi pretraživanja. Pretraživanje polazi od odabranog inicijalnog opisa D_1, te ga iterativno modificira sve dok kvaliteta dobivenog opisa ne dosegne unaprijed odabrani kriterij. Vezano uz odabir inicijalnog opisa, razlikuju se dva osnovna pristupa.
Pristup vođen podacima ili pristup *odozdo na gore* kao polazni opis uzima skup sastavljen od svih primjera ciljne klase. Ovaj opis je očito ispravan, ali je pretjerano prilagođen podacima za učenje i prekompleksan za razumijevanje. Radi poboljšanja kvalitete i smanjenja složenosti, polazni opis se modificira opetovanim generalizacijama. Rezultat je općenitije pravilo koje ispravno (ili ispravno do na određenu mjeru) klasificira podatke za učenje.

Pristup vođen modelom ili pristup *odozgo na dolje* polazi od najopćenitijeg opisa koji prekriva cijeli univerzum. U ovom slučaju je polazni opis vrlo jednostavan, ali nema nikakvu klasifikacijsku vrijednost. Broj primjera koje opis prekriva se smanjuje uzastopnim operacijama specijalizacije, nastojeći isključiti što više primjera koji ne odgovaraju ciljnoj klasi. Postupak se zaustavlja kad kvaliteta opis postigne zadovoljavajuću razinu.
U praksi su vrlo česte modifikacije ovih osnovnih pristupa, koje se u cilju postizanja što kvalitetnijih opisa upotrebljavaju i generalizaciju i specijalizaciju pri modifikaciji inicijalnog opisa. Primjerice, standardni postupak izgradnje stabla odlučivanja u prvoj fazi koristi specijalizaciju pri konstrukciji stabla, dok se u fazi podrezivanja stabla služi operacijom generalizacije.
Dakle, proces učenja se može poimati kao pretraživanje prostora rješenja. Prostor rješenja se sastoji od svih opisa podataka koje je moguće konstruirati koristeći odabranu metodu prikaza znanja. Veličina prostora rješenja je u pravilu ogromna u svim praktičnim primjenama, a ovisi o ekspresivnosti metode prikaza znanja. Gornju granicu predstavljaju metode prikaza znanja koje mogu opisati svaki podskup univerzuma U. Kardinalnost prostora rješenja je tada $2^{|U|}$, tj. eksponencijalno raste s veličinom univerzuma. Kardinalnost univerzuma ovisi o broju i domenama atributa koji se koriste za opis primjera, tako da je i sam $|U|$ često velik broj. Zbog ovako glomaznog prostora rješenja, iscrpno pretraživanje praktički nikad nije primjereno.

2.3.1. Standardni načini pretraživanja

Cilj pretraživanja prostora rješenja su oni opisi koji zadovoljavaju prethodno odabrani kriterij kvalitete opisa. Pretraživanje nastoji što prije stići do nekog od tih opisa, počevši od odabranog inicijalnog opisa. Kroz prostor rješenja se napreduje tako da se iz postojećih opisa konstruiraju novi primjenom operacija modifikacije opisa. Prema tome, konstrukcija opisa ciljnog pojma se može promatrati kao proces pretraživanja u kojem se inicijalni opis modificira primjenom operacija iz O, sve dok se ne pronađe niz operacija koji proizvodi opis zadovoljavajuće kvalitete. Slijed kojim se provjeravaju nizovi operacija naziva se *strategija pretraživanja*.
Postoji više podjela strategija pretraživanja. Jedna od osnovnih dijeli strategije na *nepovratno pretraživanje* i *pretraživanje s povratkom*.
Kod strategije nepovratnog pretraživanja, odabire se neka od operacija i nepovratno primjenjuje, bez mogućnosti naknadne revizije ove odluke. Dakle, provjerava se samo jedan niz operacija, odnosno jedan put u prostoru rješenja. Ova karakteristika je značajan ograničavajući faktor, pa je upotreba nepovratnog pretraživanja najčešće ograničena samo na situacije kad postoji samo jedna primjenjiva operacija ili kada redoslijed primjene operacija nije bitan. Nepovratno pretraživanje je podložno problemu zamke lokalnog maksimuma: pretraživanju mogu promaknuti najbolji opisi jer se proces može zaustaviti na opisu koji je samo lokalno najbolji, tj. svaka operacija primijenjena na ovaj opis daje opis lošije kvalitete. Problem lokalnih maksimuma može se ublažiti modifikacijom postupka u kojoj pretraživanje počinje s više inicijalnih opisa ili se dopušta grananje pretraživanog puta. Na ovaj način se više alternativnih nizova operacija ispituje paralelno, a kao rezultat se vraća najbolji od njih. Ovakav postupak se naziva *zrakasto pretraživanje*, a broj aktivnih alternativa koje se istodobno ispituju naziva se širina zrake.

Pretraživanje s povratkom karakterizira svojstvo da odabir operacije koja modificira opis nije nepromjenjiv; proces pretraživanja se može naknadno vratiti na mjesto odabira i primijeniti neku drugu operaciju. Stoga se pri odabiru svake operacije ustanovljava mjesto povratka, pa proces u svakom momentu osim trenutnog opisa mora pamtiti i prijeđeni put od inicijalnog opisa, te za svaki čvor na tom putu i operacije koje su do tada na njemu isprobane. Ukoliko se u procesu pretraživanja naiđe na teškoće, pretraživanje se vraća na prethodno mjesto povratka gdje se umjesto primijenjene odabire neka druga operacija, pa proces nastavlja u drugom smjeru. Osim ove varijante, postoje i modifikacije pretraživanja s povratkom gdje se ne pamti samo put od inicijalnog opisa, već i cijeli graf do tada istraženih čvorova. To omogućava veću fleksibilnost budući da se novi opisi mogu generirati iz bilo kojeg čvora grafa, ali rezultira i nešto složenijim postupkom i većim prostornim zahtjevima.
Prema dostupnim informacijama koje se koriste prilikom pretraživanja strategije pretraživanja se dijele na *slijepo pretraživanje* i *usmjereno pretraživanje*.

2.3.1.1. SLIJEPO PRETRAŽIVANJE

Kod slijepog pretraživanja jedine dostupne informacije su početni opis, dozvoljene operacije nad opisima i test na rješenje problema. Proces pretraživanja napreduje sistematski kroz prostor rješenja pretražujući čvorove u nekom prethodno definiranom redoslijedu ili birajući ih slučajno. Dvije osnovne metode slijepog pretraživanja su pretraživanje u širinu i pretraživanje u dubinu. Pretraživanje u širinu ispituje sve čvorove na istoj udaljenosti od početnog opisa prije nego generira slijedeću razinu čvorova. Nova razina nastaje ekspanzijom svih čvorova tekuće razine. Budući da se pri ekspanziji koriste sve primjenjive operacije, nova razina sadrži sve čvorove na slijedećoj udaljenosti od početnog opisa. Nasuprot tome, pretraživanje u dubinu napreduje po prostoru rješenja nastojeći se što prije udaljiti od početnog opisa. Proces pretraživanja se nastavlja ekspanzijom zadnjeg generiranog čvora sve dok se ne dođe do rješenja, ili se ne prijeđe zadani limit dubine.
Sve metode slijepog pretraživanja dijele karakteristiku velike vremenske složenosti. Slijepo pretraživanje uvijek generira veliki broj opisa, a za svaki od njih provjerava se kriterij kvalitete opisa. To je u pravilu vremenski zahtjevna operacija koja uključuje pregled skupa podataka za učenje, koji je u pravilu prilično obiman. Zbog toga se u algoritmima inteligentne analize podataka rijetko koristi slijepo pretraživanje, budući da se nastoji reducirati broj generiranih opisa odabirom onih operacija za koje se vjeruje da skraćuju put do rješenja.

2.3.1.2. USMJERENO PRETRAŽIVANJE

Postupci usmjerenog pretraživanja nastoje iskoristiti dostupne informacije o rješavanom problemu s ciljem povećanja efikasnosti pretraživanja. Ideja usmjerenog pretraživanja je smanjiti opseg pretraživanja pažljivim odabirom primjenjivanih operacija na način da se opis zadovoljavajuće kvalitete pronađe što ranije. Drugim riječima, postupak koristi dodatne informacije kako bi pronašao što kraći put od inicijalnog opisa do jednog od ciljnih opisa. Informacije o naravi opisa, o cijeni transformacije u novi opis, o osobinama cilja i slično mogu se oblikovati u heurističku funkciju vrednovanja. Heuristika označava iskustveno pravilo ili tehniku prosuđivanja koja može pomoći u traženju cilja, ali ne osigurava njegovo

pronalaženje. Neovisno o funkciji vrednovanja, postoji više metoda usmjerenog pretraživanja.

Metoda uspona na vrh uvijek bira onu operaciju koja rezultira najvećim poboljšanjem kvalitete opisa prema funkciji vrednovanja. Postupak pretraživanja napreduje kroz prostor rješenja slično kao kod pretraživanja u dubinu, s tim da se za ekspanziju odabire čvor koji daje najveću vrijednost funkcije vrednovanja. Prema tome, potrebno je izračunati ili procijeniti kvalitetu generiranih opisa korištenjem funkcije vrednovanja. U tu svrhu se definira *funkcija očekivanja* koja vrednuje kvalitetu opisa koji bi nastali primjenom određene operacije. Ako je sa f označena heuristička funkcija vrednovanja, a sa $o(D)$ opis koji nastaje primjenom operacije o na opis D, tada se funkcija očekivanja $F : D \times O \rightarrow [0,1]$ definira kao

$$F(D,o) = f(o(D)) \quad . \tag{7}$$

Funkcija očekivanja može se generalizirati promatranjem uzastopne primjene više operacija:

$$F^n(D,o) = \max_{o_i \in O} F^{(n-1)}\left(o(D), o_i\right) \quad , \tag{8}$$

gdje je $F^{(1)}$ jednostavna funkcija očekivanja definirana kao u (7). Prema generaliziranoj funkciji očekivanja, očekivanje operacije o nad opisom D je maksimalna kvaliteta opisa koji se mogu konstruirati iz opisa D u n koraka, počevši sa primjenom operacije o.

Umjesto računanja funkcije vrednovanja nad svim opisima koje je moguće konstruirati iz trenutnog opisa, moguće je i procjenjivati vjerojatnost da će primjena određene operacije proizvesti pravilo zadovoljavajuće kvalitete. Primjerice, standardni algoritam konstrukcije stabala odlučivanja koristi mjeru iz teorije informacija kako bi procijenio izglede da će primjena odabrane operacije rezultirati ispravnim i jednostavnim stablom odlučivanja.

Metoda najboljeg prvog također spada u usmjereno pretraživanje. Za razliku od metode uspona na vrh, ova metoda pamti procjene vrijednosti heurističke funkcije vrednovanja za sve prethodno generirane opise, te izabire najbolji opis. Drugim riječima, opis za ekspanziju se bira između svih do tada generiranih opisa, bez obzira gdje se oni nalaze u grafu pretraživanja. U slučaju da ekspanzija tog opisa ne dovede do povećavanja funkcije vrednovanja, proces se vraća na najbolji do tada konstruirani opis i nastavlja pretraživanje od njega. Jednako kao i u metodi uspona na vrh, može se koristiti generalizirana funkcija očekivanja pri procjeni kvalitete opisa nastalih primjenom niza operacija.

Efikasnost svih metoda usmjerenog pretraživanja snažno ovisi o svojstvima heurističke funkcije vrednovanja, budući da ona kontrolira i upravlja napredovanjem pretraživanja kroz prostor rješenja. Funkcija vrednovanja je utoliko bolja koliko dobro aproksimira kvalitetu nekog opisa i što je manja cijena računanja vrijednosti funkcije nad odabranim opisom. Zbog toga je sastavljanje dobre funkcije vrednovanja ključan element uspjeha pri primjeni usmjerenog pretraživanja. Budući da se heurističke funkcije vrednovanja oslanjaju na specijalistička znanja vezana uz određenu problematiku, ne postoji općeniti recept za sastavljanje funkcije vrednovanja. Postoje samo određene smjernice koje upućuju na neke oblike iskoristivih informacija. Primjerice, u slučaju da u skupu podataka za učenje postoje atributi koji nisu relevantni za rješavani problem, funkcija vrednovanja bi trebala

sankcionirati dodavanje uvjeta nad takvim atributima. Nadalje, ukoliko postoje neki poznati odnosi među određenim atributima, od funkcije vrednovanja se očekuje da ignorira modifikacije opisa koje ukazuju na takve odnose. Proces pretraživanja može iskoristiti prije otkrivene ili nepotpune opise pojedinih klasa. Ovo je posebno značajno kod inkrementalnog učenja, kad je potrebno dotjerati opise klasa kako bi bili u skladu s nanovo dodanim primjerima za učenje. Jedan od načina uključivanja heurističkih informacija u proces pretraživanja je i interakcija s korisnikom, kada od ponuđenih opisa informirani korisnik bira one izglednije, nad kojima želi nastaviti proces daljnjeg istraživanja.

2.3.2. Alternativni načini pretraživanja

Heuristikom vođeno sistematsko pretraživanje prostora rješenja spada u standardne tehnike simboličkog računanja. Uz odabir pogodne heurističke funkcije vrednovanja, ovaj pristup postiže prilično dobre rezultate, u pravilu smanjujući složenost pretraživanja sa eksponencijalne na polinomijalnu razinu. Međutim, u odsustvu primjerene heuristike efikasnost postupka drastično opada, neprihvatljivo produžujući vrijeme pretraživanja. Osim toga, ponekad proces pretraživanja ne može izbjeći zamkama lokalnih maksimuma. Postoje alternativne metode pretraživanja koje nastoje nadvladati ove nedostatke.

2.3.2.1. EVOLUCIJSKI ALGORITMI

Za razliku od standardnih postupaka pretraživanja koji se zasnivaju na simuliranju sistematskog rezoniranja, evolucijski postupci modeliraju postupak pretraživanja po mehanizmima prirodne selekcije [22]. Temeljna struktura koju koriste evolucijski postupci naziva se *populacija*. Populacija je podskup prostora rješenja sastavljen od pojedinačnih opisa koji se nazivaju *organizmi*. Osnovna ideja evolucijskih algoritama je iterativno poboljšavanje kvalitete organizama u populaciji na način da se stvaraju novi naraštaji organizama koristeći najbolje jedinke iz trenutne populacije. Na početku svake iteracije biraju se najbolji opisi iz kojih će nastati novi naraštaj. Dvije su osnovne operacije nad opisima pomoću kojih se formiraju novi opisi: *križanje* i *mutacija*. Križanje je binarna operacija u kojoj se kombiniranjem dijelova dvaju postojećih opisa (roditelji) stvaraju novi opisi (djeca). Nasuprot tome, mutacija je unarna operacija u kojoj novi opis nastaje slučajnom izmjenom nekog dijela postojećeg opisa.

Za potrebe ovih operacija, opisi se prikazuju kao nizovi simbola iz odabranog skupa simbola. Najčešće je to skup binarnih znamenki $\{0,1\}$. Niz simbola koji predstavlja opis podijeljen je u podnizove koje predstavljaju svaki od korištenih atributa A_i. U podnizu koji predstavlja određeni atribut A_i svaka pozicija predstavlja jednu vrijednost c_{ij} iz odgovarajuće domene $Dom(A_i)$. Dakle, svaki opis predstavljen je nizom simbola dužine $\Sigma_i |Dom(A_i)|$.

$$\underbrace{1\,0\,1\,0\,\ldots\,0}_{A_1} \quad \underbrace{0\,1\,1\,0\,\ldots\,0}_{A_2} \quad \ldots \quad \underbrace{0\,1\,0\,1\,\ldots\,0}_{A_n} \tag{9}$$

Svaka pozicija u nizu simbola kojima je predstavljen opis označava neku vrijednost $c_{ij} \in Dom(A_i)$ određenog atributa A_i. Simbol 1 na toj poziciji predstavlja uvjet $A_i=c_{ij}$ na atribut A_i. Semantika podniza koji odgovara atributu A_i odgovara disjunkciji ovakvih

uvjeta. Semantika cijelog niza, tj. opisa kojeg on predstavlja odgovara konjunkciji ovako konstruiranih uvjeta na pojedine atribute. Dakle, niz (9) odgovara opisu

$$A_1=c_{11} \vee A_1=c_{13} \vee ... \wedge A_2=c_{22} \vee A_2=c_{23} \vee ... \wedge ... \wedge A_n=c_{n2} \vee A_n=c_{n4} \vee ... \quad (10)$$

ili jednostavnije zapisano

$$A_1 \in \{c_{11}, c_{13}, ...\} \wedge A_2 \in \{c_{22}, c_{23}, ...\} \wedge ... \wedge A_n \in \{c_{n2}, c_{n4}, ...\} \ . \quad (11)$$

Postoje i drugi načini kodiranja opisa u nizove simbola kojima je moguće izraziti i kompleksnije opise. Međutim, kada se radi o primjenama evolucijskih algoritama na probleme inteligentne analize podataka, u pravilu se koristi gore opisani način kodiranja. On omogućuje vrlo jednostavnu implementaciju operacija križanja i mutacije. Križanje se svodi na cijepanje polaznih nizova na istom mjestu i formiranje novih nizova iz nastalih dijelova, a mutacija na slučajnu izmjenu pojedinih simbola niza.

U usporedbi s tradicionalnim metodama inteligentne analize podataka, evolucijski algoritmi su se pokazali nadmoćnima u slučajevima kada ne postoji adekvatna heuristika koja bi usmjeravala pretraživanje. Također, evolucijski algoritmi pokazuju dobre rezultate kada su opisi ciljnog pojma vrlo složeni (tj. sastoje se od većeg broja konjunkcija i disjunkcija). Međutim, evolucijski algoritmi imaju i dva važna nedostatka. Prvi se odnosi na ograničenu mogućnost uporabe heuristike: iako je do neke mjere moguće ugraditi znanje o problemu modificiranjem evolucijskih operatora, funkcije vrednovanja ili odabirom inicijalne populacije, to nema presudan utjecaj na performanse algoritma. U slučajevima kada se dostupno znanje o problemu može uobličiti u heurističku funkciju vrednovanja, tradicionalne metode pretraživanja pokazuju znatno bolje rezultate. Drugi nedostatak vezan je za broj izračunavanja funkcije vrednovanja. U svakoj iteraciji potrebno je vrednovati organizme tekuće populacije kako bi se odabrali najbolji. Budući da se populacije često sastoje od više stotina organizama, a i potreban broj iteracija (ovisno o problemu) može biti prilično velik, jasno je da se funkcija vrednovanja vrlo često poziva. U inteligentnoj analizi podataka to rezultira čestim upitima na bazu podataka, što značajno degradira performanse evolucijskih algoritama na većim bazama podataka.

2.3.2.2. SIMULIRANO KALJENJE

Jedan od osnovnih nedostataka tradicionalnih heuristikom vođenih metoda pretraživanja je zamka lokalnih maksimuma: proces pretraživanja može zapeti na lokalno najboljem opisu bez načina daljnjeg napredovanja po prostoru rješenja. Tehnika simuliranog kaljenja nastoji prevladati ovaj problem istražujući prostor rješenja s određenom dozom proizvoljnosti. Naime, algoritam simuliranog kaljenja ne odabire nužno onu operaciju koja rezultira maksimalnim povećanjem funkcije vrednovanja. Sam postupak odabira operacije je stohastičkog karaktera: za svaku od primjenjivih operacija postoji vjerojatnost da će biti odabrana. Vrijednost vjerojatnosti odabira pojedine operacije ovisi o dva parametra:

- povećanje funkcije vrednovanja
  Operacijama čija primjena rezultira većim povećanjem funkcije vrednovanja pridružuje se veća vjerojatnost odabira. Međutim, razmjer utjecaja ovog parametra na iznos vjerojatnosti kontrolira drugi faktor:

– temperatura
  Temperatura je globalni parametar koji regulira stupanj proizvoljnosti u postupku odabira operacije. Što je temperatura viša, to su razlike među vjerojatnostima pojedinih operacija manje. Posebno, kad je temperatura maksimalna, svim operacijama se pridružuje jednaka vjerojatnost. Snižavanjem temperature raste utjecaj povećanja funkcije vrednovanja na vjerojatnost odabira operacije. Kad je temperatura minimalna, simulirano kaljenje se svodi na metodu uspona na vrh – uvijek se bira najperspektivnija operacija.

Algoritam simuliranog kaljenja počinje pretraživanje s visokom vrijednošću temperature, dozvoljavajući širenje po prostoru rješenja kako bi se izbjegli lokalni maksimumi. Kako proces napreduje, temperatura se polako snižava usmjeravajući pretraživanje prema opisima s visokom vrijednosti funkcije vrednovanja.
Valja napomenuti kako simulirano kaljenje ne garantira pronalaženja globalnog maksimuma, ali proširujući opseg pretraživanja povećava mogućnost pronalaženja kvalitetnijih rješenja.

2.4. TEORIJA RAČUNALNOG UČENJA

Osnovni problem s kojim se opisane metode pretraživanja prostora rješenja pokušavaju nositi je dosezanje dovoljno dobrog rješenja uz razumnu složenost u uvjetima ogromnog prostora rješenja. Postojanje i složenost algoritama strojnog učenja je područje kojim se bavi teorija računalnog učenja [3], čije razmatranje izlazi iz dosega ovog rada. Ovdje će se navesti samo jedan od rezultata teorije računalnog učenja i ukratko analizirati složenost iscrpnog pretraživanja, radi ilustracije okvira složenosti u kojima operiraju algoritmi strojnog učenja.

2.4.1. Učenje prebrojavanjem

Odabrana metoda prikaza znanja određuje kardinalnost skupa svih opisa koje je moguće konstruirati. Učenje prebrojavanjem je u stvari iscrpno pretraživanje koje zahtijeva provjeru cijelog prostora rješenja.
U slučaju učenja s unaprijed definiranim klasama (učenje s učiteljem), provjerava se svaki opis u potrazi za opisom koji najbolje odgovara promatranoj klasi. Prema tome, složenost postupka odgovara kardinalnosti skupa svih opisa. U slučaju prikaza znanja prije spomenutim klasifikacijskim pravilima, opis predstavlja disjunkciju elementarnih opisa. Promatrajmo proizvoljni elementarni opis ϕ. Atribut A_i se može pojaviti u ϕ u obliku uvjeta $A_i=c$ za neki $c \in Dom(A_i)$, ili ϕ uopće ne sadrži A_i. Uz pretpostavku konačnih domena, broj različitih elementarnih opisa iznosi

$$\prod_A \left(|Dom(A_i)| + 1 \right) \quad . \tag{12}$$

Broj različitih disjunkcija koje je moguće sastaviti od elementarnih opisa tada iznosi

$$2^{\prod_A \left(|Dom(A_i)|+1 \right)} - 1 \quad . \tag{13}$$

Međutim, različitih opisa ima nešto manje, budući da među ovim disjunkcijama ima i nekih koje su logički ekvivalentne. Naime, slučaj da se A_i ne pojavljuje u elementarnom opisu ϕ je ekvivalentan slučaju u kojem se dozvoljava da A_i poprimi bilo koji vrijednost iz $Dom(A_i)$. Stoga je ukupan broj različitih opisa u slučaju klasifikacijskih pravila dan sa

$$2^{\Pi_A|Dom(A_i)|}-1 \quad . \tag{14}$$

Vrijedi primijetiti da ukupan broj opisa ne ovisi o kardinalnosti skupa podataka za učenje S, već samo o kardinalnostima domena atributa, tj. o veličini univerzuma U. Dakle, složenost učenja prebrojavanjem kod učenja s učiteljem eksponencijalno ovisi o kardinalitetu univerzuma.

Situacija je još nepovoljnija kod učenja bez učitelja, jer je osim pronalaženja opisa potrebno i particionirati skup podataka za učenje. Dakle, za svaku klasu u svakom od mogućih particioniranja potrebno je provjeriti svaki od opisa. Neka je sa N označen kardinalitet skupa podataka za učenje (N=|S|), sa $P(N,m)$ broj različitih načina na koje se skup od N elemenata može particionirati na m nepraznih disjunktnih podskupova, a neka $D(A)$ označava ukupan broj opisa koje je moguće konstruirati u odabranoj metodi prikaza znanja. Tada je ukupan broj particioniranja u kombinaciji s opisima odgovarajućih klasa dan sa

$$\sum_{m=1}^{N} P(N,m)\times m \times D(A) \quad . \tag{15}$$

Broj opisa $D(A)$ ne ovisi o broju klasa m ni kardinalnosti skupa podataka za učenje N i za slučaj klasifikacijskih pravila je dan s (14). Općenito, broj različitih particioniranja skupa od N elemenata u m podskupova iznosi

$$P(N,m)=\frac{1}{m!}\sum_{j=0}^{m}\binom{m}{j}(-1)^j(m-j)^N \quad . \tag{16}$$

Funkcija $P(N,m)$ raste eksponencijalno sa N. Može se pokazati da za velike N vrijedi

$$P(N,m)\approx\frac{m^N}{m!}\approx m^{N-m}e^m\sqrt{2\pi m} \quad . \tag{17}$$

Prema tome, koristeći (14), (15) i (17), ukupan broj particioniranja u kombinaciji s opisima odgovarajućih klasa može se ocijeniti sa

$$\sum_{m=1}^{N}\frac{m^{N+1}}{m!}2^{\Pi_A|Dom(A_i)|} \quad \approx \quad \sum_{m=1}^{N} m^{N-m+1}e^m\sqrt{2\pi m}\,2^{\Pi_A|Dom(A_i)|} \quad . \tag{18}$$

U stvarnosti je broj podskupova particije (tj. broj klasa) mali u usporedbi sa veličinom skupa podataka za učenje, odnosno vrijedi $m << N$. Stoga sumacija u formuli (18) u praktičnim primjenama ne seže do N, već obuhvaća samo manji broj početnih sumanada. No i uz tu pretpostavku, rezultirajući broj odnosno složenost postupka je zapanjujuća.

2.4.2. Vjerojatno približno točno učenje

U opisu učenja prebrojavanjem, implicitno se pretpostavljalo da su svi pozitivni primjeri klase *C* sadržani u skupu podataka za učenje *S*. Međutim, budući da skup podataka za učenje općenito predstavlja samo manji dio univerzalnog skupa *U*, nije vjerojatno da će ova pretpostavka biti ispunjena. U pravilu se može očekivati samo da će klasa *C* biti zastupljena u skupu podataka za učenje s nekim brojem pozitivnih primjera.
Činjenica da svi mogući pozitivni primjeri klase *C* nisu sadržani u skupu podataka za učenje *S* implicira da se sa sigurnošću ne može tvrditi da će pronađeni opis skupa $S \cap C$ dobro opisivati klasu *C*, tj. odgovarati svim mogućim instancama klase *C*. Ipak, pri primjeni algoritama poželjna je određena doza sigurnosti u ispravnost izvedenih opisa. Preciznije, potrebno je minimizirati vjerojatnost da izvedeni opis krivo klasificira dotad neviđeni primjer.
Zamisao o sigurnosti u ispravnost izvedenih opisa formalizirana je kroz pojam vjerojatno približno točnog učenja ili PAC-učenja (engl. *Probably Approximately Correct learning*), [57]. Prije formuliranja ovog pojma, potrebno se je podsjetiti nekih osnovnih definicija iz teorije vjerojatnosti.

Distribucija vjerojatnosti nad *U* je funkcija $\mu : \wp(U) \to [0,1]$ takva da vrijedi

1. $\mu(\varnothing) = 0$;
2. $\mu(U) = 1$;
3. za u parovima disjunktne skupove $S_1, \ldots, S_n$ vrijedi $\mu(\bigcup_{i=1}^{n} S_i) = \sum_{i=1}^{n} \mu(S_i)$.

Dakako, pretpostavlja se da je *U* konačan. Za skup $E \subseteq U$, $\mu(E)$ označava vjerojatnost da slučajno odabrani $x \in U$ pripada skupu *E*. Ako *C* koristimo kao oznaku predikata sa značenjem $C(x)$ je istinito ako i samo ako $x \in C$, tada je potrebno minimizirati

$$Er_{\mu}(D,C) = \mu\{x \in U \mid D(x) \neq C(x)\} \quad . \tag{19}$$

Dakle, $Er_{\mu}(D,C)$ predstavlja vjerojatnost da izvedeni opis *D* krivo klasificira neki primjer. Jasno je da izvedeni opis *D* ovisi o korištenom skupu podataka za učenje. Ako je sa $L(C,S)$ označen opis kojeg algoritam strojnog učenja *L* pronađe za klasu *C* koristeći *S* kao skup podataka za učenje, tada treba minimizirati izraz

$$er_{\mu}(L,C,S) = \mu\{x \in U \mid L(C,S)(x) \neq C(x)\} \quad . \tag{20}$$

Budući da se skup podataka za učenje ne može birati, već je zadan po principu slučajnog uzorka, potrebno je minimizirati $er_{\mu}(L,C,S)$ neovisno o skupu podataka za učenje *S*. Radi formalizacije, potrebna je distribucija $\mu^n : \wp(U^n) \to [0,1]$, gdje $\mu^n(Y)$ označava vjerojatnost da slučajni uzorak od *n* elemenata pripada skupu *Y*. Vjerojatno približno točno učenje definira se korištenjem distribucije μ^n.

Definicija

Algoritam *L* može naučiti *C* vjerojatno približno točno ako

$$\begin{aligned} &\forall\, 0<\varepsilon<1, \forall\, 0<\delta<1, \exists m_0, \forall c \in C, \forall \mu, \forall m \geq m_0 : \\ &\mu^m\{S \mid m=|S| \wedge er_{\mu}(L,C,S) < \varepsilon\} > 1-\delta \quad . \end{aligned} \tag{21}$$

Drugim riječima, za odabrane male vrijednosti ε i δ uvijek postoji m_0 takav da će za gotovo sve skupove podataka za učenje čija kardinalnost je veća od m_0 rezultat algoritma biti opis koji je skoro uvijek ispravan. Algoritam učenja je konzistentan ako je ispravan na skupu podataka za učenje. Važan rezultat teorije računalnog učenja govori da su svi konzistentni algoritmi strojnog učenja (inteligentne analize podataka) vjerojatno približno točni, pod pretpostavkom da je U konačan.
Utjecaj ovog rezultata na praktične primjene inteligentne analize podataka nije presudan iz dva razloga. Kao prvo, podaci u bazama podataka su u pravilu nepotpuni i podložni nepravilnostima. Zbog toga se od algoritma strojnog učenja ne može tražiti konzistentnost na danim podacima. Drugi (možda čak i važniji) razlog je to što dokaz teorema ovisi o pretpostavci da su primjeri koje skup podataka za učenje sadrži neovisni. Međutim, podaci u stvarnim bazama podataka podaci su neizbježno međusobno povezani, poznatim ili nepoznatim vezama, što narušava pretpostavku neovisnosti primjera za učenje. Drugim riječima, u praksi teorem nije primjenjiv budući da nisu ispunjeni zahtjevi koje postavlja.

3. Proces inteligentne analize podataka

Primjena algoritama strojnog učenja u inteligentnoj analizi podataka podrazumijeva upotrebu baze podataka kao izvora podataka. U pravilu se radi o transakcijskim bazama podataka koje bilježe svojstva različitih događaja i objekata vezanih uz područje primjene za koje je baza projektirana. Obim i detaljnost svojstava koja se bilježe u bazi podataka uvjetovani su zahtjevima transakcijskih sustava koji operiraju nad bazom podataka, a ne potrebama inteligentne analize podataka. Osim toga, ljudski faktor i specifičnosti aplikacija koje koriste bazu podataka uzrok su neminovnih neispravnosti u podacima koje mogu biti slučajnog ili sistematskog karaktera. Budući da su upravo relevantni i kvalitetni podaci osnovni resurs inteligentne analize podataka, nužno se pojavljuju određene poteškoće uvjetovane specifičnostima koje sa sobom nosi izvor podataka.

3.1. PROBLEMI U PRIMJENI TEHNIKA STROJNOG UČENJA

Deskriptivnost i ispravnost podataka u bazi, dostupnost i brojnost primjera bitnih za analizu, voluminoznost i dinamički karakter baza podataka – sva ova svojstva su potencijalni uzroci problema s kojima se susreću algoritmi strojnog učenja u inteligentnoj analizi podataka. Poklanjanje adekvatne pažnje ovim problemima bitno povećava izglede za uspjeh procesa analize podataka.
U nastavku je dan pregled karakterističnih problema uvjetovanih bazom kao izvorom podataka, kao i najčešćih načina njihovog prevladavanja.

3.1.1. Nepotpuni podaci

U prikazu načela na kojima počivaju algoritmi strojnog učenja, implicitno se pretpostavljalo da se korištenjem skupa prognostičkih atributa može jednoznačno odrediti klasa primjera, tj. da postoji deterministički opis klase. Međutim, većina stvarnih problema nije takvog karaktera: često je iz prirode problema jasno da ne postoji skup prognostičkih atributa koji bi jednoznačno determinirao klasu svakog primjera. Čak i u slučajevima kada takav skup atributa postoji, malo je vjerojatno da će svi oni biti zabilježeni u bazi podataka.
U takvim slučajevima mora se odustati od zahtjeva ispravnosti opisa nad skupom podataka za učenje. Umjesto toga, teži se opisima kod kojih će broj pogrešno klasificiranih primjera biti što manji. Alternativno, mogu se koristiti *probabilistički opisi*, koji ne rezultiraju kategoričkim predviđanjem klase primjera, već kao rezultat daju ocjenu vjerojatnosti da primjer pripada ciljnoj klasi.

3.1.2. Raspršenost primjera

Pri konstrukciji opisa klase, algoritmi strojnog učenja u stvari određuju granice klase u okviru univerzuma svih mogućih primjera. Granice klase će biti moguće preciznije odrediti ukoliko u skupu podataka za učenje postoje primjeri koji su na samom rubu klase ili neposredno izvan njega. Drugim riječima, primjeri iz skupa podataka za učenje trebali bi biti raspršeni po cijelom univerzumu, odnosno odražavati što različitije ponašanje.

Nažalost, baze podataka u pravilu sadrže primjere koji su grupirani oko karakterističnih područja i pokrivaju razmjerno mali dio univerzuma. U takvoj situaciji su granice klasa relativno nejasne i ne mogu se precizno odrediti. Stoga je čest slučaj da se dotad neviđeni primjeri koji su locirani podalje od skupa podataka za učenje pogrešno klasificiraju.
Neki sustavi ovaj problem nastoje prevladati oblikom interakcije s okolinom: sustav nastoji generirati zanimljive primjere kojih nema u skupu podataka za učenje, indicirajući tako poželjnu strukturu skupa podataka za učenje [53]. Budući da skup podataka za učenje ne obuhvaća cijelu bazu podataka, postoji mogućnost da se prema sugestijama sustava sastavi kvalitetniji izbor podataka za učenje, s bolje pokrivenim graničnim područjima.

3.1.3. Šum u podacima

U dosadašnjim razmatranjima, implicitno se pretpostavljalo da su podaci nad kojima se provode postupci strojnog učenja potpuno točni, tj. da vrijednosti zabilježenih atributa odgovaraju stvarnim svojstvima promatranih objekata. Kada su izvor podataka stvarne baze podataka, ova pretpostavka gotovo nikad nije točna. Osim neminovnih slučajnih pogrešaka pri unosu podataka, vrijednosti nekih atributa mogu biti rezultat subjektivne procjene ili razmjerno nepreciznih mjerenja. Neki primjeri čak mogu biti i pogrešno klasificirani. Neispravnosti ovog tipa (netočne vrijednosti prognostičkih ili prognoziranih atributa za pojedine primjere) koje nisu sistematskog karaktera nazivaju se *šum u podacima*.
Šum u podacima predstavlja poteškoću u dva različita procesa: prilikom konstrukcije opisa, te prilikom klasifikacije dotad neviđenih primjera.
Šum u skupu podataka za učenje može uzrokovati deformiranje opisa kako bi se pokrili primjeri s netočnim podacima, rezultirajući tako opisom lošijih klasifikacijskih sposobnosti. Sustavi inteligentne analize podataka nastoje statističkim tehnikama izolirati šum u podacima, te ukloniti one primjere za koje se pretpostavlja sadrže netočne podatke. Sve takve tehnike zasnivaju se na tezi da se mala količina neuobičajenih podataka smatra šumom, pa se takvi primjeri ignoriraju.
Šum stvara poteškoće i prilikom klasifikacije novih primjera kada podaci za klasifikaciju sadrže šum. Netočne vrijednosti prognostičkih atributa mogu uzrokovati pogrešnu klasifikaciju. I u ovom slučaju statističke metode uklanjanja šuma iz podataka mogu povećati preciznost klasifikacije. Međutim kod nekih algoritama strojnog učenja zamijećen je zanimljiv fenomen vezan uz klasifikaciju primjera koji sadrže šum. Naime, pokazalo se da opisi izvedeni iz skupa podataka za učenje koji također sadrži šum bolje klasificiraju takve primjere od opisa izvedenih iz pročišćenog skupa podataka za učenje [48]. Stoga, ukoliko je primarna zadaća takvih sustava klasifikacija dotad neviđenih primjera, trud uložen u eliminaciju šuma možda neće dati očekivane rezultate.

3.1.4. Neodređene vrijednosti atributa

Kao i šum u podacima, tako i neodređene vrijednosti atributa mogu uzrokovati probleme i u procesu konstrukcije opisa i pri klasifikaciji novih primjera.
Postoji više pristupa problemu neodređenih vrijednosti atributa u skupu podataka za učenje. Najjednostavniji, ali i najmanje korišteni način je odbacivanje primjera s

neodređenom vrijednošću nekog atributa. Ovaj pristup se rijetko koristi zbog činjenice da u stvarnim bazama podataka s velikim brojem atributa većina primjera nema zabilježenu vrijednost svakog atributa. Drugi pristup nalaže da se sve neodređene vrijednosti zamijene posebnom vrijednošću "nepoznato", kako u skupu podataka za učenje, tako i u primjerima za klasifikaciju. Ovaj pristup često se koristi u slučajevima kad činjenica da atribut nema zabilježenu vrijednost nosi nekakvo značenje, koje se na ovaj način uvažava. Sofisticiraniji pristupi pokušavaju na različite načine nadomjestiti neodređene vrijednosti, bilo statističkim metodama ili čak korištenjem tehnika strojnog učenja (predviđanjem neodređenih vrijednosti na osnovu poznatih atributa i informacije o klasi), [48]. Postoje i sustavi koji vjerojatnosnim pristupom prevladavaju problem neodređenih vrijednosti.
Kad je riječ o klasifikaciji dotad neviđenih primjera u uvjetima neodređenih vrijednosti, opisi koji ispituju vrijednost određenog atributa ne mogu se direktno primijeniti ako primjeru nedostaje vrijednost upravo tog atributa. Jedna od tehnika rješavanja ovog problema je izračunavanje vjerojatnosti da opis prekriva promatrani primjer. Vjerojatnost se računa kao umnožak vjerojatnosti da svaka od neodređenih vrijednosti u primjeru odgovara onoj koju opis zahtijeva. Sama vjerojatnost da će atribut poprimiti određenu vrijednost ocjenjuje se na osnovu vrijednosti tog atributa u skupu podataka za učenje. U konačnici, primjer se klasificira koristeći onaj opis za kojeg je vjerojatnost prekrivanja primjera najveća.
Iako je ova tehnika prilično jednostavna, pokazano je da prilično dobro održava preciznost klasifikacije s povećanjem broja neodređenih vrijednosti atributa u primjerima za klasifikaciju.

3.1.5. Obujam podataka

Za razliku od drugih primjena postupaka strojnog učenja, inteligentna analiza podataka u pravilu operira nad vrlo velikim skupovima podataka za učenje. Općenito, baze podataka su vrlo glomazne, kako u terminima broja zapisa u bazi tako i u količini informacija po zapisu. I jedna i druga činjenica mogu predstavljati problem u smislu performansi algoritma strojnog učenja.
Brojnost atributa u skupu podataka za učenje donosi potencijalne prednosti, ali i nedostatke. Budući da je temeljni cilj inteligentne analize podataka pronalaženje opisa zadane klase podataka, izgledi da će pravilnost biti moguće adekvatno izraziti u terminima prognostičkih atributa rastu s povećanjem broja atributa. Međutim, s povećanjem broja atributa raste i broj opisa koje je moguće konstruirati, pa je time pronalaženje adekvatnog opisa teže. Konkretan broj opisa koje je moguće konstruirati ovisi o odabranoj metodi prikaza znanja, ali u pravilu ne pada ispod eksponencijalne ovisnosti o broju atributa i kardinalnostima odgovarajućih domena (gornju granicu predstavlja broj svih podskupova univerzuma $2^{|U|}$). Stoga je u slučaju stvarnih baza podataka ukupan broj mogućih opisa ogroman, pa je nemoguće pretražiti cijeli skup mogućih opisa u potrazi za najboljim opisom klase. Stoga sustavi inteligentne analize podataka koriste različita ograničenja i heuristike koje sužavaju opseg pretraživanja. Dakako, suženi opseg mijenja i konačni cilj pretraživanja: ne traži se najbolji opis, već jedan iz skupa "dovoljno dobrih".
S druge strane, brojnost primjera u bazi podataka nameće drugačiju vrstu zahtjeva na performanse algoritama strojnog učenja. U procesu konstrukcije opisa klase, potrebno je provjeriti kvalitetu svakog generiranog opisa. Provjera ispravnosti opisa

zahtijeva traženje potvrde u skupu podataka za učenje. Drugim riječima, određivanje kvalitete opisa zahtijeva pretraživanje baze podataka. Budući da broj generiranih opisa u postupku pretraživanja može biti značajan, a pretraživanje glomaznih baza podataka je vremenski zahtjevna operacija, ukupni učinak može poražavajuće djelovati na performanse sustava.
Jedan od načina prevladavanja ovog problema je uzorkovanje podataka iz baze podataka. Kvaliteta opis se računa korištenjem manjeg, unaprijed odabranog reprezentativnog uzorka primjera iz baze podataka (ponekad se takav uzorak naziva *prozor*). Najbolji od ovako odabranih opisa se naknadno podvrgavaju provjeri kvalitete nad cijelom bazom podataka. Postupak može biti iterativan, tako da se nakon provjere nad cijelom bazom u uzorak uključe neki od nekorektno klasificiranih primjera. Ovako formiran novi uzorak koristi se u slijedećoj iteraciji konstrukcije opisa.

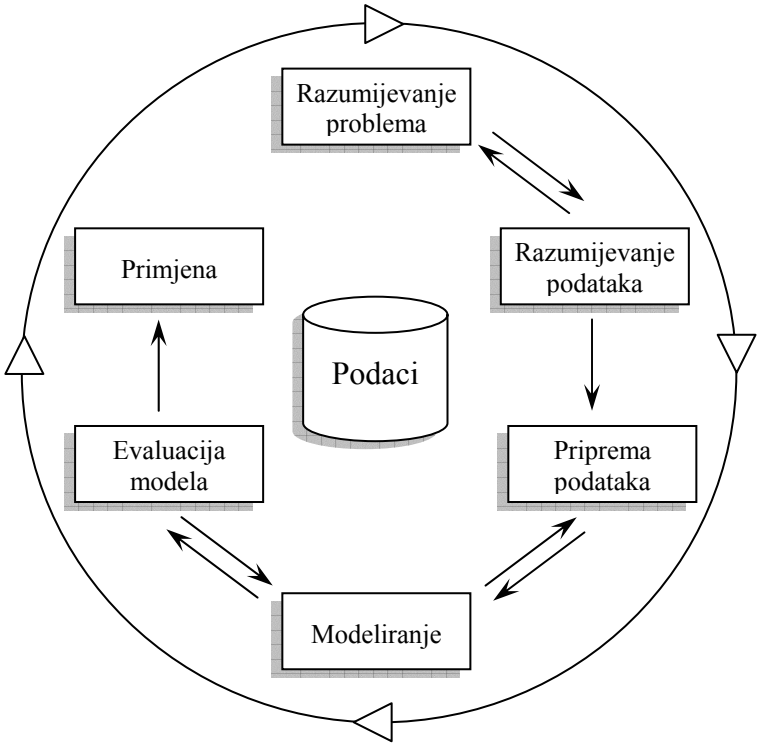
3.1.6. Ažuriranje podataka

U skladu s primarnom namjenom baza podataka, podaci u bazi često se mijenjaju: dodaju se novi podaci, neki podaci se mijenjaju ili brišu. Opisi koji su u nekom trenutku izvedeni korištenjem podataka iz baze stoga mogu postati nekonzistentni s novim stanjem baze. Poželjno svojstvo sustava inteligentne analize podataka je prilagođavanje ovakvim promjenama. Prilagođavanje podrazumijeva usklađivanje opisa s najnovijim podacima u bazi, ponekad i nauštrb starijih podataka. Smisao toga je naglašavanje aktualnih pravilnosti, budući da se svojstva primjera u bazi podataka mogu mijenjati kroz vrijeme u skladu s procesima i trendovima iz okruženja. Stoga je potrebno podatke u bazi vrednovati u skladu s njihovom ažurnošću, tj. dodjelom većih težinskih vrijednosti novijim podacima usmjeravati proces pretraživanja ka novijim spoznajama.
Potreba prilagođavanja novim podacima može se detektirati praćenjem ispravnosti klasifikacije: veći broj pogrešno klasificiranih primjera ukazuje na zastarjelost opisa. Preciznije, opis je potrebno korigirati kada stvarna uspješnost klasifikacije u nekom periodu statistički značajno odstupi od predviđene uspješnosti procijenjene u trenutku izrade opisa.

3.2. PROCES INTELIGENTNE ANALIZE PODATAKA

Da bi postupak inteligentne analize podataka donio primjerene rezultate, nužno je apsolvirati opisane probleme koji se javljaju pri uporabi baze podataka kao osnovnog izvora podataka. Stoga je jasno da inteligentna analiza podataka nije puka primjena sofisticiranih postupaka strojnog učenja na podatke smještene u bazama podataka. Inteligentna analiza podataka predstavlja složeni proces pronalaženja novih, potencijalno korisnih i razumljivih pravilnosti u velikim količinama podataka. Iako srž samog procesa čine tehnike modeliranja podataka zasnovane prvenstveno na postupcima strojnog učenja, i ostale faze procesa su jednako važni čimbenici uspjeha procesa i kvalitete rezultata.
Slika 3 prikazuje osnovne faze standardnog procesa inteligentne analize podataka, njihov redoslijed i međuovisnosti.

Ovakva struktura procesa inteligentne analize podataka definirana je na osnovu iskustava u praktičnoj primjeni metodologije u vodećim svjetskim kompanijama, a naziva se CRISP-DM (prema engl. *CRoss Industry Standard Process for Data Mining*) [9]. Prema ovom prikazu, inteligentna analiza podataka uključuje sve bitne procese prilikom rješavanja određenog problema za koji postoje podaci potencijalno dovoljni za njegovo kvalitetno rješavanje.



Slika 3: Osnovne faze procesa inteligentne analize podataka prema CRISP-DM standardu

Faze razumijevanja problema i podataka na početku procesa i primjene rezultata na kraju procesa su tijesno vezane uz domenu konkretnog problema koji se nastoji riješiti primjenom inteligentne analize podataka. Središnji i algoritamski najzahtjevniji dio procesa inteligentne 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 inteligentnom analizom podataka.

3.2.1. Razumijevanje problema

Korak kojim započinje proces inteligentne analize podataka je razumijevanje problema koji je potrebno riješiti. Problem je potrebno sagledati kako sa stanovišta domene problema, tako i s aspekta primjene tehnika inteligentne analize podataka. U obje sfere nastoje se pronaći i razumjeti ograničenja i ostali važni čimbenici koji mogu utjecati na proces i konačni rezultat inteligentne analize podataka. U ocjeni situacije obično se procjenjuje nivo znanja o problemu, te podaci dostupni za njegovo rješavanje. Budući da upravo podaci predstavljaju centralni dio procesa

inteligentne analize podataka, potrebno je provjeriti njihov potencijal u smislu obuhvata dovoljne količine informacija za rješavanje problema.
U ovoj fazi se definiraju konačni ciljevi u rješavanju problema. Ciljevi specificirani iz perspektive domene promatranog problema razlikuju se od specifikacije ciljeva prevedene u terminologiju inteligentne analize podataka. Primjerice, cilj u domeni poslovnog problema može biti povećanje odziva kupaca u planiranju marketinške kampanje novog proizvoda. U formulaciji inteligentne analize podataka, traži se opis skupine kupaca (u terminima poznatih svojstava) za koje postoji visoka vjerojatnost kupnje spomenutog proizvoda.
Nakon što su specificirani ciljevi, bitno je odrediti i kriterije uspješnosti procesa inteligentne analize podataka. Kriteriji mogu biti kvantitativno izraženi, kao npr. povećan broj kupaca nekog proizvoda (ili, u terminologiji inteligentne analize podataka, određivanje minimalnog nivoa prediktivne točnosti rješenja). Ukoliko se od rješenja traži novi uvid u odnose varijabli iz skupa podataka kojima je problem predstavljen, kriteriji uspješnosti se u pravilu iskazuju kvalitativno. U tom slučaju uobičajeno je da stručnjak iz čije domene problem dolazi ocijeni korisnost rezultata procesa inteligentne analize podataka s obzirom na postojeća saznanja o području.
Svi spomenuti koraci mogu poslužiti za stvaranje okvirnog plana za izvedbu procesa inteligentne analize podataka. U tom se planu predviđaju pojedine aktivnosti u samom procesu, kao i rezultati tih aktivnosti. Poželjno je definirati 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, da bi se mogla ocijeniti opravdanost projekta inteligentne analize podataka, u planu se treba naći i okvirna procjena njegovog trajanja odnosno koštanja, te očekivanog dobitka od uspješno provedenog projekta.

3.2.2. Razumijevanje podataka

Nakon postavljanja cilja projekta i okvirnog plana njegovog ostvarenja, u fokus dolazi osnovni resurs cijelog procesa – podaci. Prvi korak ove faze je inicijalno prikupljanje podataka. Ovdje bi se trebali otkriti eventualni problemi u prikupljanju podataka i pronaći načini njihovog otklanjanja. Nakon utvrđivanja lokacije podataka i metoda potrebnih za njihovo prikupljanje, pokreće se postupak njihovog prikupljanja. Stvara se lista prikupljenih podataka, te se podaci na odgovarajući način pohranjuju za daljnje korištenje.
Prije same primjene specifičnih tehnika modeliranja, potrebno je proučiti i opisati osnovne karakteristike podataka koji su na raspolaganju. Svrha ove aktivnosti je istraživanje osnovne strukture podataka kako bi se stekao dojam o informativnosti podataka u smislu rješavanja postavljenog problema. U osnovne karakteristike podataka spadaju:

- količina podataka, tj. broj atributa i broj primjera
- identitet i značenje pojedinih atributa
- inicijalni format zapisa vrijednosti pojedinih atributa
- osnovna svojstva distribucije vrijednosti pojedinih atributa.

Uvid u osnovna svojstva distribucije vrijednosti atributa stječe se uporabom jednostavnih statističkih tehnika. Kod numeričkih atributa to uključuje izračun srednje vrijednosti atributa, standardne devijacije, te minimuma i maksimuma

vrijednosti. Za nominalne atribute izrađuju se tablice frekvencija pojavljivanja određenih vrijednosti.
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.
Završni korak faze razumijevanja podataka je verifikacija kvalitete podataka. To 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 neispravnosti 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*).

U neočekivane primjere spadaju oni kod kojih je vrijednost jednog ili više atributa značajno različita od vrijednosti ostalih sličnih primjera, odnosno leži značajno izvan intervala očekivanih vrijednosti za takav primjer. Neočekivani primjeri se obično svrstavaju u dvije kategorije: one koji predstavljaju šum odnosno pogrešku, te one koji predstavljaju novi fenomen u odnosu na populaciju primjera 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.

3.2.3. Priprema podataka

Prije same faze modeliranja podataka provodi se konačna priprema podataka za proces modeliranja. Pažljivo provedena priprema podataka može u velikoj mjeri doprinijeti kvaliteti konačnog rezultata procesa inteligentne analize podataka. Ugrubo, priprema podataka se može podijeliti u četiri koraka: odabir podataka, pročišćavanje podataka, formiranje novih podataka i formatiranje podataka.

Odabir podataka podrazumijeva izdvajanje određenog podskupa iz skupa svih prikupljenih podataka koji će služiti kao osnova za daljnje analize. Odabir se u pravilu vrši na osnovu kriterija kvalitete i tehničkih ograničenja. Odabir se može vršiti na razini primjera i na razini atributa.
Kada je riječ o odabiru podataka na razini primjera, kriterij kvalitete nalaže da se odabiru kompletni i ispravni primjeri. Analogno tome, na razini atributa moguće je isključiti atribute s nekvalitetnim vrijednostima (npr. velik broj primjera s neodređenim vrijednostima ili šumom u podacima).
Ukoliko postoje tehnička ograničenja na količinu podataka koja se može na efikasan način modelirati odabranom tehnikom, uporabom statističkih tehnika može se provesti uzorkovanje primjera kako bi se smanjio ukupan broj primjera, a modeliranje ipak provodilo na što reprezentativnijem skupu podataka. Osim toga, moguće je reducirati i broj atributa kojima su primjeri opisani. Najčešće se na osnovu prethodno ispitane informativne vrijednosti atributa iz skupa podataka izbacuju atributi koji imaju slabu prediktivnost klase, ili pak imaju visok stupanj redundancije s nekim izražajnijim atributom. Postoje različite tehnike redukcije broja atributa, od

spajanja više atributa uz određenu (linearnu ili nelinearnu) transformaciju, do statistički utemeljenih tehnika (npr. ocjena korelacije atributa, metoda analize osnovnih komponenti).

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

  Ovdje spada npr. normaliziranje vrijednosti numeričkih atributa na interval [0,1], ili na vrijednost standardne devijacije vrijednosti promatranog atributa.

– zaglađivanje podataka

  Jedan od primjera zaglađivanja podataka je diskretizacija numeričkih atributa (ovo je nužan preduvjet za neke tehnike modeliranja podataka).

– tretman neodređenih vrijednosti

  Postoji nekoliko pristupa problemu neodređenih vrijednosti, ali ni jedan nije u potpunosti korektan. Zbog toga je osnovna preporuka eksperimentirati s različitim pristupima u fazi modeliranja, te odabrati onaj koji se pokaže najprimjerenijim. Jednostavni pristupi uključuju odbacivanje podataka s neodređenim vrijednostima i zamjenu neodređenih vrijednosti konstantama ili statistički opravdanim vrijednostima. Ispravniji, ali znatno kompleksniji pristup je nadomještanje neodređenih vrijednosti nekom od tehnika modeliranja podataka (čime to postaje zaseban problem inteligentne analize podataka).

Formiranje novih podataka je zahvat kojim se iz postojećih odabranih i pročišćenih podataka konstruiraju novi podaci, 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. Banalan primjer je učenje pojma "uspravan" ili "polegnut" na primjerima geometrijskih likova. Intuitivno, u terminima širine i visine pojmu "uspravan" odgovarao bi opis $širina < visina$. Međutim, tehnike modeliranja kod kojih se vrijednosti atributa uspoređuju isključivo s konstantama (nije moguće testiranje vrijednosti usporedbom dvaju atributa) ne mogu izvesti ovakav opis. Problem se može riješiti uvođenjem novog, izvedenog atributa *širina–visina*, koji vodi do opisa $širina–visina < 0$. Zbog toga je odabir načina konstrukcije novih podataka prvenstveno vezan uz svojstva odabrane tehnike modeliranja podataka.
Načini formiranja novih podataka uključuju stvaranje novog atributa na osnovu dva ili više postojećih atributa, stvaranje novih primjera, stvaranje novog atributa transformacijom postojećeg atributa (npr. diskretizacija), stvaranje novog atributa agregacijom informacija iz više primjera i slično.

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. Promjene koje se u provode su sintaktičkog karaktera; primjerice promjena u redoslijedu atributa ili primjera, promjena načina odvajanja vrijednosti u primjerima, reduciranje duljine nominalnih atributa i slično.

Rezultat faze pripreme podataka se na odgovarajući način pripremljeni i zapisani podaci koji predstavljaju ulaz u fazu modeliranja. Međutim, kao što je i vidljivo s dijagrama na slici 3, rezultati faze modeliranja mogu pokazati potrebu za dodatnim zahvatima nad podacima, pa se proces pripreme i modeliranja podataka može iterativno ponavljati.

3.2.4. Modeliranje podataka

Modeliranje podataka predstavlja ključnu fazu procesa inteligentne 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: odabir tehnike modeliranja, definiranje testnog uzorka, te proces konstrukcije modela.

Problem odabira adekvatne tehnike modeliranja podataka razmatran je već u fazi razumijevanja problema. Međutim, prethodne faze razumijevanja i pripreme podataka mogu donijeti nova saznanja o svojstvima i karakteru podataka koji se analiziraju. Zbog toga se u ovom koraku ponovno razmatra adekvatnost tehnike modeliranja koja je sugerirana na početku procesa, budući da nove okolnosti 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 prema različitim tehnikama modeliranja i njihovim specifičnim osobinama. Svojstva problema treba promotriti s različitih aspekata (klasifikacijski ili asocijacijski problem, deskriptivni ili prediktivni tip problema, razina neizvjesnosti koju problem pokazuje i slično) kako bi se što bolje modelirale njegove pravilnosti.

Prije same konstrukcije modela potrebno je definirati proceduru za testiranje kvalitete generiranih modela. Primjerice, kod klasifikacijskih problema se kao mjera kvalitete modela obično 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.

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. Konačni model (ili više

modela slične kvalitete) potrebno je detaljno interpretirati u smislu njegove kompleksnosti i pouzdanosti rezultata koje proizvodi.

3.2.5. Evaluacija modela

Ocjena konstruiranih modela provodi se kako s aspekta inteligentne analize podataka, tako i s aspekta domene promatranog problema.
Ocjena performansi tipičnih za područje inteligentne analize podataka provodi se u skladu s prethodno definiranom procedurom testiranja kvalitete modela. To je u stvari tehnička provjera rezultata modela. Osim ocjene pouzdanosti modela na testnom uzorku, potrebno je ocijeniti njegovu smislenost, te objasniti razloge za konačnu kombinaciju parametara tehnike modeliranja. Ukoliko postoji razlog 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. Osim obilježja direktno povezanih s ciljevima obrade podataka, dobro je razmotriti i ostala svojstva modela nevezana uz promatrani problem. Šira perspektiva može otkriti dodatne informacije i sugestije, koje je moguće upotrijebiti u svrhu poboljšanja procesa inteligentne analize podataka.
Faza evaluacije modela završava revizijom do sada obavljenih faza procesa i određivanjem narednih aktivnosti. U ovom koraku kontrolira se kvaliteta izvođenja cijelog procesa analize podataka i identificiraju eventualni propusti. Provjerava se razina zadovoljenosti osnovnih kriterija uspješnosti procesa, te se u tom svjetlu razmatraju odluke donesene u pojedinim fazama procesa. Rezultat revizije naglašava eventualne nedostatke u provođenju procesa, moguća poboljšanja, te alternativna rješenja za pojedine faze procesa analize podataka.
Na osnovu provedene revizije utvrđuje se spremnost prelaska u završnu fazu procesa inteligentne 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.

3.2.6. 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 inteligentne analize podataka je završni izvještaj kojim se rezimiraju rezultati projekta. U njemu se objašnjavaju sve 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. Pri tome se naglasak stavlja na perspektivu domene promatranog problema, budući da je izvještaj namijenjen prije svega inicijatorima samog procesa, tj. ljudima koji su problem i postavili.

4. Prikupljanje i priprema podataka

4.1. PRIKUPLJANJE PODATAKA

U novije vrijeme inteligentna analiza podataka se najčešće vrši nad podacima iz informacijskih sustava poslovnih subjekata. Međutim, u stvarnosti je rijedak slučaj da poslovni subjekt ima izgrađen jedinstven integralni informacijski sustav u kojem se bilježe i obrađuju svi poslovni podaci. U pravilu se radi o više međusobno nepovezanih aplikacija koje pokrivaju samo pojedine funkcionalnosti poslovnog sustava. Međutim, kompleksnost stvarnih poslovnih primjena inteligentne analize podataka zahtijeva združivanje podataka iz takvih heterogenih izvora unutar organizacije, a nerijetko i podataka iz vanjskog svijeta.

Upravo zbog potrebe integriranja podataka iz različitih izvora, prikupljanje i priprema podataka mogu biti vremenski najzahtjevniji dio procesa inteligentne analize podataka. Iako aktivnosti koje se provode u ovoj fazi nisu posebno složene, praksa pokazuje da treba uložiti značajan napor kako bi se došlo do jedinstvenog skupa podataka zadovoljavajuće kvalitete. Problemi na koje se pri tom nailazi različiti su od slučaja do slučaja: razočaravajuća kvaliteta podataka, različiti načina praćenja podataka, raznolike konvencije zapisa, neusklađeni vremenski periodi, različite razine agregacije podataka, neusklađenost identifikatora, šarolike pogreške u podacima i slično. Iako nisu direktno vezani za inteligentnu analizu podataka, sve ove probleme je potrebno prevladati kako bi došli do osnovnog resursa – relevantnih i kvalitetnih podataka.

Na sreću, mnogi poslovni subjekti su već prepoznali značaj integracije podataka na razini organizacije. Fragmentirani podaci čija je primarna namjena obrada i bilježenje svakodnevnih poslova u izoliranom podsustavu imaju ograničenu informativnu vrijednost. Globalan pogled na pokazatelje uspješnosti cijele organizacije zahtjeva integraciju podataka iz svih izvora unutar organizacije. Zbog značaja integriranih podataka za upravljačke strukture poslovnih subjekata, u praksi se sve više afirmira koncept *skladišta podataka* [33]. Skladište podataka predstavlja repozitorij poslovnih podataka integriranih na razini cijele organizacije. Ono podrazumijeva prikupljanje podataka iz svih podsustava organizacije, njihovo združivanje i pročišćavanje. Osnovna namjena skladišta podataka je osiguravanje infrastrukturne podloge sustavima za podršku odlučivanju.

Sa stanovišta inteligentne analize podataka, postojanje skladišta podataka značajno olakšava prikupljanje i pripremu podataka, jer skladište preuzima zahtjevnu zadaću integracije heterogenih podataka. Ukoliko skladište podataka ne postoji, mnoge aktivnosti vezane uz uspostavu skladišta podataka biti će nužno odraditi u okviru procesa inteligentne analize, iako najčešće u reduciranom opsegu. Međutim, sama dostupnost skladišta podataka ne znači da se u njemu nalaze svi potrebni podaci. Nerijetko će biti potrebno posegnuti i za izvorima podataka izvan organizacije kako bi se došlo do podataka relevantnih za analizu, a koje organizacija sistematski ne bilježi. Budući da se takvi podaci u pravilu dobivaju od institucija specijaliziranih za prikupljanje podataka (statistički zavodi i slične institucije), oni su uglavnom dobro opisani i unaprijed pročišćeni, pa njihovo združivanje s poslovnim podacima organizacije ne iziskuje veći napor. Stoga su skladišta podataka od iznimne koristi za prikupljanje podataka u poslovnim primjenama inteligentne analize podataka.

4.2. PRIPREMA PODATAKA

Nakon što su svi potrebni podaci prikupljeni i integrirani u jedinstvenu relaciju, predstoji inicijalno upoznavanje s podacima. U ovoj aktivnosti nužna je suradnja sa stručnjacima iz domene rješavanog problema. Oni se trebaju intenzivno uključiti u objašnjavanje anomalija u podacima, tumačenje značenja neodređenih vrijednosti, razjašnjavanje prirode, preciznosti i ažurnosti promatranih podataka i slično.
Cilj inicijalnog upoznavanja s podacima je stjecanje uvida u strukturu podataka i njihovu informativnu vrijednost. U tu svrhu se koriste jednostavne statističke tehnike i vizualizacijski alati. Izrađuju se histogrami distribucije vrijednosti nominalnih atributa, ispituje distribucija numeričkih atributa, grafički prikazuju vrijednosti pojedinih atributa u odnosu na klasu ili neki drugi atribut. Pomoću ovakvih tehnika vizualizacije podataka relativno jednostavno se uočavaju neočekivani podaci (engl. *outliers*), jer pokazuju značajan odmak od obrasca koji se provlači kroz ostatak vrijednosti. Oni mogu predstavljati pogreške u prikupljenim podacima, sistematske nedosljednosti pri unosu podataka ili konvencije u zapisu nestandardnih situacija koje nigdje nisu eksplicitno spomenute (npr. nepoznata godina zapisuje se kao 9999 i slično). Osim toga, grafički prikaz podataka može pojednostaviti komunikaciju s stručnjacima iz domene problema.
Kvaliteta podataka može biti narušena i drugim nedostacima koji se također mogu jednostavno uočiti pri inicijalnoj provjeri kvalitete podataka. Jedan od uzroka loše kvalitete podataka može biti i dupliciranje zapisa o nekim primjerima iz skupa podataka za učenje. Do toga najčešće dolazi prilikom združivanja podataka iz više izvora, budući da se podaci o nekim stvarnim objektima mogu pratiti u više različitih podsustava. Većina tehnika modeliranja podataka proizvest će različite modele ukoliko se u skupu podataka za učenje neki primjeri multipliciraju. Uzrok tome je činjenica da ponavljanjem podaci dobivaju na važnosti, pa im se povećava i utjecaj na konačni rezultat modeliranja.
Tipografske greške pri unosu podataka također imaju loš utjecaj na kvalitetu podataka. Kod numeričkih atributa grube tipografske pogreške će rezultirati odmakom od standardnog raspona vrijednosti, dok će manje greške biti relativno teško detektirati. Kad su nominalni atributi u pitanju, tipografske greške rezultiraju novim vrijednostima koje proširuju domenu atributa. Histogram učestalosti vrijednosti atributa izdvojit će rijetko korištene vrijednosti kao potencijalne tipografske greške. Pažljivim pregledom domene nominalnog atributa mogu se otkriti i greške koje nisu tipografskog karaktera, već predstavljaju različite načine zapisa istog podatka (npr. razlike u zapisivanju naziva ulica).
Kvalitetu podataka narušava i korištenje neažurnih podataka. Zbog toga je potrebno voditi računa o razini ažurnosti podataka u sustavu iz kojeg podaci potječu. Mnoga svojstva stvarnih objekata se s vremenom mijenjaju, pa podaci o njima zastarijevaju. Baze podataka u kojima se ti podaci prate dizajnirane su sukladno potrebama temeljnih aplikacija koje bazu koriste. Od podataka koje aplikacija inicijalno prikupi, redovito se ažuriraju samo oni podaci koji su ključni za funkcionalnost aplikacije. Svi podaci čija točnost nije nužna u svakodnevnom radu aplikacije će se u pravilu rijetko (ili nikako) ažurirati. Stoga je, u slučaju da postoji mogućnost izbora, podatke dobro uzeti iz sustava u kojem je njihova važnost veća. Moguća je i diskriminacija podataka u čiju se ažurnost sumnja dodjeljivanjem manje težinske vrijednosti, pa se tako umanjuje njihov utjecaj na konačni rezultat modeliranja.

4.2.1. Formiranje novih atributa

Često se iz interakcije sa stručnjacima iz domene problema može razaznati novo značenje pojedinih podataka i iz toga formirati novi atributi. Primjerice, prve dvije znamenke matičnog broja klijenta mogu označavati godinu prvog kontakta s klijentom, početni dio broja računa može sadržavati informaciju o podružnici koja je račun izdala i slično. Ovakvo implicitno kodiranje informacija je široko prihvaćeno, pa za neke standardne podatke neće trebati suradnja stručnjaka u razjašnjavanju značenja (npr. pozivni telefonski broj otkriva geografsku lokaciju, registarska oznaka automobila sugerira prebivalište i slično). Budući da tehnike modeliranja podataka ne mogu baratati implicitno zapisanim podacima, uputno je izdvojiti što više implicitno zapisanih podataka u zasebne atribute kako bi se takve informacije mogle iskoristiti u fazi modeliranja. Time se dobiva na deskriptivnosti korištenih podataka, te u konačnici proširuju mogućnosti njihovog modeliranja. Radi izbjegavanja eventualne redundancije formiranih atributa, tehnike odabira atributa se obično primjenjuju nakon dodavanja novih atributa u skup podataka za učenje.
Bilježenje implicitnih podataka nije jedina svrha formiranja novih atributa. Novi atributi mogu se sintetizirati da bi se postojeći podaci transformirali u oblik koji je prilagođeniji odabranoj tehnici modeliranja podataka. Primjerice, mnoge tehnike modeliranja podataka mogu dijeliti univerzum primjera samo paralelno s dimenzijskim osima (tj. vrijednost atributa moguće je uspoređivati isključivo sa konstantama). Ako postoji naznaka da su atributi A i B u nekoj linearnoj ovisnosti, definiranje novog atributa kao njihovog omjera A/B omogućit će takvim tehnikama razmatranje njihovog odnosa. Ovakve intervencije mogu značajno poboljšati iskoristivost prikupljenih podataka i dramatično povećati kvalitetu konačnog rezultata. Sam način transformacije podataka sugeriraju svojstva odabrane tehnike modeliranja i semantika prikupljenih podataka. Stoga je ključno dobro poznavanje korištenih tehnika modeliranja, te razumijevanje rješavanog problema i prikupljenih podataka.

4.2.2. Normalizacija numeričkih atributa

Potreba za normaliziranjem numeričkih atributa ovisi o načinu na koji odabrana tehnika modeliranja tretira numeričke atribute. Većina tehnika se prema numeričkim atributima odnosi kao prema ordinalnim vrijednostima i koristi samo usporedbe vrijednosti tipa *veći od* ili *manji od*. Budući da se svaki atribut razmatra zasebno, kod takvih tehnika nisu bitni razmjeri veličine vrijednosti različitih atributa – vrijednosti svakog atributa se promatraju na vlastitoj skali vrijednosti koja je vezana za taj atribut. Međutim, postoje tehnike modeliranja koje numeričke atribute tretiraju kao razmjerne vrijednosti, te ih koriste za izračunavanje "udaljenosti" među različitim primjerima. U takve tehnike spadaju razne varijante regresijske analize i metode učenja temeljene na promatranju najbližih susjeda. Budući da se metrika u univerzumu primjera definira tako da uključuje vrijednosti svih atributa, međusobni razmjeri vrijednosti različitih atributa postaju značajni. Ukoliko metrika takva da jednako tretira sve atribute, presudan utjecaj na izračun udaljenosti će imati atributi s velikim rasponom vrijednosti, dok će oni malog raspona vrijednosti biti gotovo

nebitni. Zbog toga je nužno uskladiti skale na kojima se mjere vrijednosti različitih atributa. Takav postupak naziva se *normalizacija numeričkih atributa*.
Dva su najčešća postupka normalizacije numeričkih atributa: svođenje na fiksni interval i standardizacija statističke varijable.
Svođenje na fiksni interval je jednostavan postupak pomoću kojeg se sve vrijednosti nominalnog atributa svode na unaprijed odabrani fiksni interval, najčešće [0,1]. U tom slučaju se transformacija vrijednosti atributa obavlja funkcijom

$$f(x) = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad , \tag{22}$$

gdje su $x_{\min}$ i $x_{\max}$ najmanja odnosno najveća vrijednost promatranog atributa.
Ništa kompliciraniji postupak nije ni standardizacija statističke varijable. Ona uključuje izračun aritmetičke sredine i standardne devijacije uzorka, a transformacija se provodi izrazom

$$s(x) = \frac{x - \bar{x}}{\sigma_x} \quad , \tag{23}$$

gdje $\bar{x}$ označava aritmetičku sredinu svih vrijednosti atributa, a σ_x njihovu standardnu devijaciju. Za razliku od svođenja na fiksni interval, standardizacija statističke varijable ne rezultira fiksnim intervalom vrijednosti, već skupom vrijednosti čija je srednja vrijednost 0, a standardna devijacija 1.

4.2.3. Neodređene vrijednost atributa

Već ranije je ukazano na probleme koje donose neodređene vrijednosti atributa kako u konstrukciji tako i u primjeni konstruiranog modela, te su spomenuti načini prevladavanja tih problema. Ovdje će ukratko biti riječi o drugom aspektu te problematike, o značenju same činjenice da vrijednosti nekih atributa u primjeru nedostaju.
Naime, može postojati više razloga zbog kojih se pojavljuju neodređene vrijednosti atributa. Neki od mogućih razloga su združivanje više sličnih (ali ne identičnih) skupova podataka, odluka da se neko mjerenje ne provede, odbijanje otkrivanja nekih osobnih podataka, nepostojanje određenog obilježja na promatranom primjeru i slično. Ovakve pozadinske informacije otkrivaju nova saznanja o promatranim primjerima koja nisu zabilježena u skupu podataka za učenje, a mogu biti značajna prilikom modeliranja pravilnosti u podacima. Primjerice, nepoznata vrijednost podatka o visini biljke u prvoj godini života može značiti da podatak nije izmjeren, ili da je biljka uginula prije tog roka pa podatak nema smisla. Nebilježenje ove razlike u skupu podataka za učenje može biti značajan propust za studiju utjecaja podneblja na rast biljaka.
Zbog toga tretiranje svih neodređenih vrijednosti na isti način ignorira potencijalno korisne nove informacije o promatranom primjeru. Ilustrativan primjer o značaju neodređenih vrijednosti za inteligentnu analizu podataka dolazi iz medicinske domene. Radi određivanja dijagnoze pacijenta, liječnici pacijente šalju na različite preglede. Na osnovu rezultata obavljenih pregleda određuje se dijagnoza i savjetuje terapija. Analizom podataka o obavljenim pregledima na pacijentu ustanovilo se da se u određenim okolnostima ispravna dijagnoza može postaviti samo na osnovu

odluke liječnika na koje preglede pacijenta poslati, bez obzira na rezultate tih pregleda. Dakle, u tom slučaju je za postavljanje dijagnoze dovoljna informacija o tome na kojim pregledima pacijent nije bio – dovoljna je samo informacija o neodređenim vrijednostima, dok se zabilježene vrijednosti atributa mogu ignorirati.
Analiza skrivenog značenja u neodređenim vrijednostima atributa zahtjeva suradnju s stručnjacima iz domene problema. Samo osoba upoznata sa značenjem podataka i stvarnim pojavama na koje se podaci odnose može na adekvatan način prosuditi postoji li implicitno značenje u činjenici da je vrijednost nekog atributa u konkretnom primjeru neodređena. Ukoliko se ispostavi da u podacima postoji više različitih vrsta neodređenih vrijednosti, odmah je jasno da implicitno značenje postoji, te ga je potrebno na odgovarajući način artikulirati.
Tehnike modeliranja podataka uglavnom operiraju pod pretpostavkom da neodređene vrijednosti atributa ne nose nikakvo posebno značenje, te sve neodređene vrijednosti tretiraju na jednak način. Stoga se informacije o značenju neodređenih vrijednosti atributa ne mogu iskoristiti ukoliko nisu eksplicitno zapisane u skupu podataka za učenje. Najčešći način kodiranja takvih podataka je uvođenje nove vrijednosti promatranog atributa kojom se obilježavaju svi primjeri s implicitnim značenjem neodređene vrijednosti. Moguće je i uvođenje više takvih vrijednosti tipa "nepoznato", "nije mjereno", "nemoguće" i slično, kojima se obilježavaju različita značenja neodređenih vrijednosti. Ponekad se u skup podataka uvode i zasebni atributi koji bilježe značenje neodređenih vrijednosti.

4.2.4. Automatska detekcija šuma

Značaj kvalitete podataka za proces inteligentne analize podataka naglašava se već pri inicijalnom upoznavanju s podacima. Još u toj ranoj fazi nastoje se uočiti i korigirati neki nedostaci koji narušavaju kvalitetu podataka, kao što su tipografske greške, sistematske nedosljednosti ili dupliciranje podataka. Međutim, manje evidentne nepravilnosti u podacima nije jednostavno uočiti, a one su u velikim bazama podataka više pravilo nego izuzetak. Nepouzdane znaju biti ne samo vrijednosti pojedinog atributa, već i sam podatak o klasi primjera. Alternativa mukotrpnom pregledavanju velikih količina podataka je automatizacija traženja neočekivanih vrijednosti. Tehnike automatske detekcije šuma se često oslanjaju upravo na postupke strojnog učenja kako bi se u prikupljenih podacima detektirale anomalije i izdvojili primjeri s neočekivanim vrijednostima pojedinih atributa.
Najčešće korištena tehnika automatske detekcije šuma zasniva se na odbacivanju krivo klasificiranih primjera. U postupku se koristi neka od tehnika strojnog učenja koja ne mora biti ista kao ona koja će se upotrijebiti u fazi modeliranja podataka. Primjenom odabrane tehnike na skup podataka za učenje dobiva se opis klasa u terminima prognostičkih atributa. U stvarnosti, ispravnost induciranih opisa na skupu podataka za učenje neće biti potpuna, tj. opisi će krivo klasificirati neke od primjera za učenje. To je očekivana situacija budući da funkcija vrednovanja osim ispravnosti uzima u obzir i valjanost opisa, pa prihvaća manje žrtve u ispravnosti klasifikacije ako one znatnije poboljšavaju valjanost opisa. Motivacija iza ovakvog mehanizma je izbjegavanje pretjerane prilagodbe specifičnostima konkretnog skupa podataka za učenje, u nadi da će to poboljšati klasifikacijske performanse na testnom skupu, odnosno u stvarnoj primjeni. Pojednostavljeno gledano, funkcija vrednovanja će na osnovu kriterija valjanosti opisa odlučiti da neke primjere iz skupa podataka za

učenje treba ignorirati, držeći da su oni artefakti skupa podataka za učenje koji ne odražavaju globalne pravilnosti u podacima. Postupak automatske detekcije šuma sastoji se u tome da se takvi primjeri proglase šumom u podacima.
Opisani postupak može se iterativno ponavljati. Nakon što su iz skupa podataka za učenje izbačeni primjeri koje inducirani opisi krivo klasificiraju, nova iteracija započinje primjenom tehnike strojnog učenja na reduciranom skupu podataka za učenje. Postupak staje kada broj krivo klasificiranih primjera padne ispod unaprijed određene granice. Ukoliko je udio ukupnog broja odbačenih primjera u skupu podataka za učenje značajan, to upućuje na loš odabir korištene tehnike strojnog učenja. Nakon odabira tehnike kojom je moguće bolje izraziti zanimljive pravilnosti u podacima, postupak se može ponoviti.
U ovu svrhu najčešće se koristi tehnika indukcije stabala odlučivanja [28]. Eksperimenti na standardnim skupovima podataka za učenje su pokazali da odbacivanje krivo klasificiranih primjera u većini slučajeva nema statistički značajan utjecaj na ispravnost klasifikacije. Međutim, ova tehnika neosporno utječe na veličinu rezultirajućih stabala odlučivanja. Stabla dobivena iz reduciranog skupa podataka za učenje su u pravilu bitno manja od originalnih, iako pokazuju gotovo istu ispravnost klasifikacije. Razlog tomu leži u postupku konstrukcije stabala odlučivanja. Pretjerana prilagodba modela skupu podataka za učenje izbjegava se podrezivanjem konstruiranog stabla. Podrezivanje se zasniva na provjeri statističke opravdanosti postojanja određenog podstabla, te se nepoželjna podstabla odbacuju. Neke primjere koje je nepodrezano stablo ispravno klasificiralo podrezano stablo će krivo klasificirati. Stoga je ispravnost klasifikacije žrtvovana već u inicijalnom stablu odlučivanja, ali se utjecaj takve odluke na strukturi stabla reflektira samo lokalno, u odbačenom podstablu. Ponovna konstrukcija stabla iz reduciranog skupa za učenje takvu odluku samo propagira na globalnu razinu, omogućujući drugačiji odabir grananja na višim razinama u stablu. Konačni rezultat je jednostavnije stablo odlučivanja, a uz dobro odabranu strategiju podrezivanja ispravnost klasifikacije će biti očuvana.
Upravo činjenica da su rezultirajuća stabla jednostavnija, a sličnih klasifikacijskih performansi odgovara principu Ockhamove oštrice. Stoga ovaj princip opravdava odbacivanje krivo klasificiranih primjera kao šuma u podacima.

Međutim, svaki oblik automatske detekcije šuma uvijek nosi opasnost da se pri odbacivanju naizgled loših primjera odbace i sasvim ispravni podaci. Bez konzultacija s stručnjakom iz domene problema, nema pravog kriterija prema kojem se za konkretan primjer može odlučiti spada li u neispravne podatke ili jednostavno ne odgovara tipu korištenog modela. Zbog toga je dobra praksa izdvojene primjere prezentirati stručnjaku iz domene problema, koji će verificirati odbacivanje podataka ili eventualno korigirati uočene neispravnosti. Ponekad ova opcija nije realna zbog ogromnih količina podataka ili nepostojanja načina provjere određenih podataka.
U nedostatku komunikacije sa stručnjacima iz domene problema, sumnja u adekvatnost modela korištenog za detekciju šuma u podacima može se umanjiti uporabom više različitih tehnika modeliranja. Konzervativniji pristup će primjer proglasiti neispravnim tek ako ga krivo klasificiraju sve upotrijebljene tehnike modeliranja. Manje suzdržan pristup traži da većina konstruiranih modela krivo klasificira primjer. Kod prvog pristupa postoji opasnost zaostajanja veće količine

neispravnih podataka u skupu podataka za učenje. Nedostatak drugog pristupa je mogućnost da najpogodnija tehnika modeliranja bude jednostavno preglasana.
Konačno, pri primjeni bilo koje tehnike automatske detekcije šuma u podacima dobro je voditi računa o udjelu pojedinih klasa kako u odbačenim tako i u zadržanim primjerima. Može se dogoditi da se filtriranjem neopazice odbacuju uglavnom primjeri određene klase (ili grupe klasa) kako bi se poboljšala ispravnost klasifikacije na preostalim klasama. Ovakva diskriminacija klasa vodi lošem skupu podataka za učenje, i u konačnici rezultira modelom koji neadekvatno opisuje pravilnosti u podacima.

4.2.5. Odabir atributa

Tipične baze podataka sadrže vrlo velike količine podataka, bilo da se govori o broju zapisa koje sadrže ili o broju atributa od kojih se zapisi sastoje. Budući da skup podataka za učenje nastaje prikupljanjem podataka iz različitih baza, obično se i on sastoji od razmjerno velikog broja atributa koji opisuju primjere. Realno je očekivati da će od tako velikog broja atributa neki od njih biti nevažni za problem koji se istražuje. Osim toga, neki atributi će zacijelo biti redundantni, ne samo zbog velikog broja atributa već i zbog pretpostavke o univerzalnoj relaciji. Naime, skup podataka za učenje je jedna relacija koja u pravilu nastaje selekcijom podataka iz većeg broja originalnih relacija u bazi podataka, pa su određene denormalizacije podataka nužne.
Iako bi u načelu veći broj atributa trebao biti poželjan jer bolje opisani primjeri za učenje nude veću mogućnost razlučivanja, praktično je situacija drugačija. Naime, prisutnost nevažnih ili redundantnih atributa u skupu podataka za učenje može značajno otežati postupak modeliranja pravilnosti u podacima. Posljedice mogu biti smanjena preciznost klasifikacije, manja razumljivost izvedenih modela ili povećanje trajanja i prostornih zahtjeva postupka. Stoga je opravdano nastojanje da se iz skupa podataka za učenje uklone nevažni i redundantni atributi – oni nisu značajni u kontekstu rješavanog problema, a uklanjanjem se mogu postići bolji rezultati.
Tehnike modeliranja podataka se razlikuju po načinu na koji se nose s problemom nebitnih i redundantnih atributa. Sa jedne strane leže tehnike koje ovom problemu uopće ne pridaju važnost. Jedna od takvih je tehnika najbližih susjeda: novi primjeri se klasificiraju pronalaženjem najbližih primjera iz skupa podataka za učenje, pa se na osnovu njih određuje klasa novog primjera. Pri tom se u računanju udaljenosti primjera na jednak način koriste podaci svih atributa. Velika razlika u vrijednostima nebitnih atributa može značajno narušiti ocjenu udaljenosti pojedinih primjera, te tako prouzročiti pogrešnu klasifikaciju.
S druge strane se nalaze tehnike koje se eksplicitno pokušavaju usredotočiti na najbitnije atribute i ignorirati preostale. Primjer takve tehnike je konstruiranje stabala odlučivanja. U svakom koraku postupka bira se atribut koji najviše obećava, odnosno atribut pomoću kojeg se skup podataka za učenje dijeli u što homogenije podskupove. Teoretski, postupak nikad neće odabrati nevažan atribut, pa bi stablo odlučivanja trebalo sadržavati samo najprediktivnije atribute.
Međutim, eksperimenti na standardnim skupovima podataka za učenje su pokazali da dodavanje nevažnog atributa (npr. generiranog bacanjem pravilnog novčića) može degradirati klasifikacijske performanse stabla odlučivanja, u nekim primjerima i do 10% [29]. Utvrdilo se da je u nekom dijelu stabla odlučivanja nevažni atribut ipak bio izabran kao kriterij podjele podataka za učenje. To se na prvi pogled kosi s

načinom odabira atributa, ali se ovakva pojava ipak može objasniti. Naime, na nižim razinama stabla sve je manje i manje primjera na osnovu kojih se donosi odluka o odabiru najpogodnijeg atributa. Osim toga, broj čvorova u stablu se eksponencijalno povećava s dubinom stabla. Zbog velikog broja čvorova u kojima se odluke donose na osnovu male količine podataka, vjerojatnost da će nevažni atribut u nekom trenutku biti izabran nije zanemariva, a multiplikativno se povećava s dubinom stabla. Odabir nevažnog atributa kao kriterija klasifikacije uzrokuje slučajne greške prilikom primjene modela na novim primjerima, pa ispravnost klasifikacije opada.
Ovom problemu su manje podložne tehnike koje ne fragmentiraju skup podataka za učenje. Takav je naivni Bayesov postupak, koji se zasniva na pretpostavci da su svi prognostički atributi međusobno neovisni. Budući da umjetno dodani nebitni atributi ispunjavaju ovu pretpostavku, postupak ih uspješno ignorira. Međutim, zbog iste ove pretpostavke klasifikacijske performanse postupka mogu biti narušene dodavanjem redundantnih atributa u skup podataka za učenje.
Zaključno, vidi se da provođenje postupka odabira atributa prije faze modeliranja može biti korisno, bez obzira pokušava li tehnika modeliranja podataka sama odabrati pogodne atribute. Smanjivanje broja atributa reducira dimenzionalnost prostora hipoteza, pa algoritmi modeliranja rade brže i traže manje prostora. Odabir atributa može povećati klasifikacijske performanse konačnog modela, posebno kod tehnika koje su osjetljive na postojanje nevažnih ili redundantnih atributa. Osim toga, smanjeni broj atributa u pravilu rezultira jednostavnijim i razumljivijim modelom koji bolje opisuje ciljne pojmove.

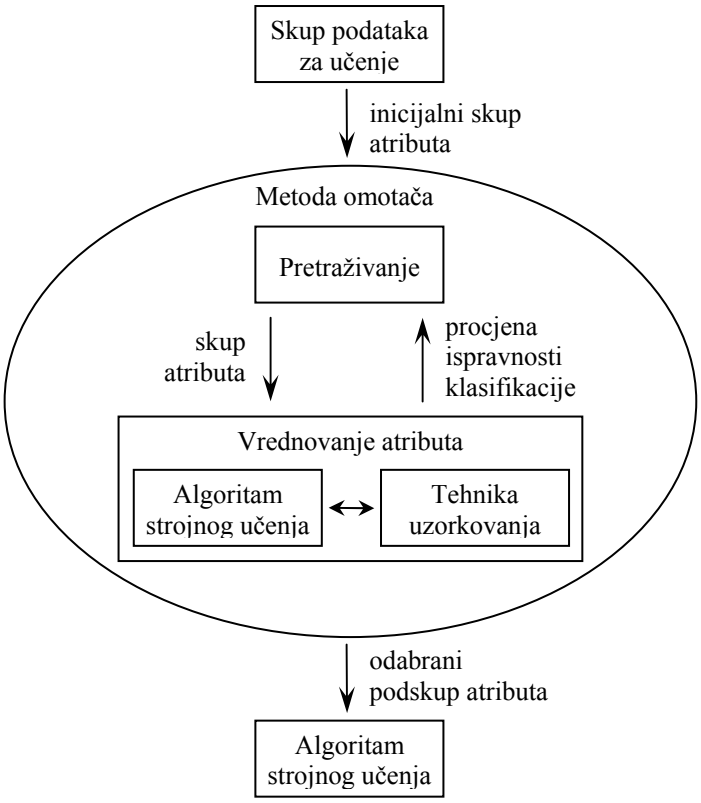
Cilj postupka odabira atributa je izdvajanje nevažnih i redundantnih atributa iz skupa podataka za učenje. Kako bi se u skupu podataka za učenje zadržali samo poželjni atributi, potrebna je neka mjera vrijednosti atributa u odnosu na klasifikacijski problem. U literaturi postoji više različitih kriterija pomoću kojih se ocjenjuje vrijednost atributa (ili skupa atributa), ali niti jedan nije općeprihvaćen. Stoga se u postoji više različitih tehnika odabira atributa u inteligentnoj analizi podataka.
Osnovna podjela grupira postupke odabira atributa prema njihovoj povezanosti s odabranom tehnikom strojnog učenja koja će se primijeniti u fazi modeliranja podataka [36]:

- *metode omotača*
  Tehnike koje spadaju u ovu skupinu prosuđuju vrijednost podskupa atributa koristeći upravo ciljni algoritam strojnog učenja u kombinaciji sa statističkim tehnikama uzorkovanja.
- *metode filtera*
  Tehnike ove skupine se pri ocjenjivanju vrijednosti atributa oslanjaju na heuristike zasnovane na općim svojstvima podataka.

4.2.5.1. METODE OMOTAČA

Metode omotača su tijesno vezane za odabrani algoritam strojnog učenja. Vrijednost određenog skupa atributa izražava se pomoću stupnja ispravnosti klasifikacije koju postiže model konstruiran uz korištenje samo tih atributa. Uz zadani podskup atributa, ispravnost klasifikacije se ocjenjuje korištenjem tehnika uzorkovanja, primjerice unakrsnom validacijom (engl. *cross-validation*). Drugim riječima, za svaki promatrani podskup atributa izgrađuje se model i ocjenjuju njegove

performanse. Bolje performanse nekog modela ukazuju na bolji izbor atributa iz kojih je model nastao.



Slika 4: Odabir atributa metodama omotača

Cijeli postupak odabira atributa je kod metoda omotača računski vrlo zahtjevan zbog učestalog izvođenja samog algoritma strojnog učenja. Za svaki promatrani podskup atributa potrebno je dobiti ocjenu performansi odgovarajućeg modela, a metode ocjene ispravnosti modela uglavnom zahtijevaju usrednjavanje rezultata po većem broju izgrađenih modela. Dakle, za svaki promatrani podskup atributa izgrađuje se više modela, a ukupan broj podskupova eksponencijalno raste s povećanjem broja atributa. Stoga je iscrpno pretraživanje svih mogućnosti neprikladno, pa se pribjegava drugim načinima pretraživanja. Najčešće se koriste pohlepne tehnike koje prostor rješenja prelaze tako da u svakom koraku pregledavaju lokalno dostupne alternative, pa proces pretraživanja nastavljaju od najbolje od njih (tehnika uspona na vrh).

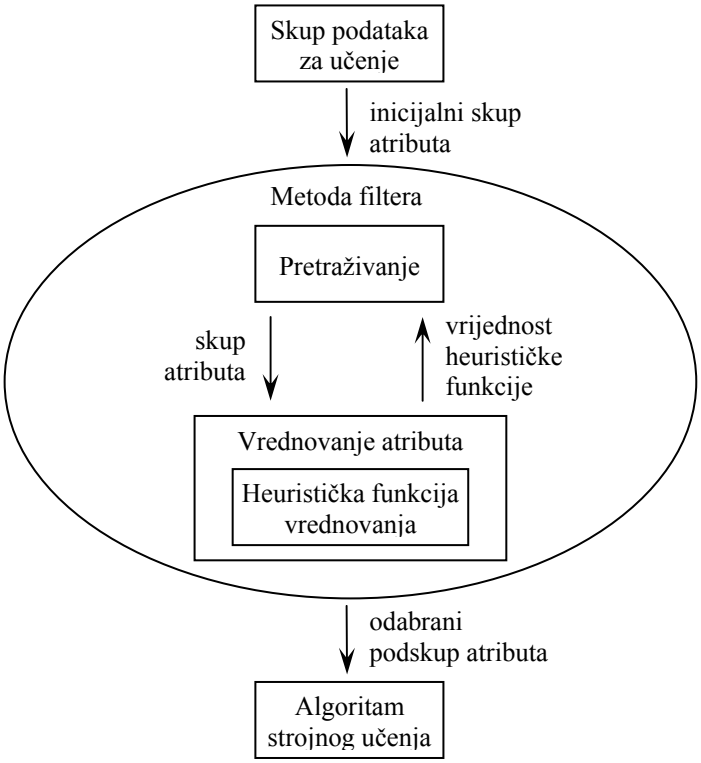
```
1.  s := početno stanje
2.  ponavljaj
    2.1. state := s
    2.2. pronađi sve lokalno dostupne alternative za state
    2.3. izračunaj ocjenu e(t) svaku od pronađenih alternativa
    2.4. s := alternativa s najvećom ocjenom e(s)
    dok je e(s) > e(state)
3.  vrati state
```

Slika 5: Postupak pohlepnog pretraživanja tehnikom uspona na vrh

Do lokalnih alternativa se dolazi na način da se promatranom podskupu atributa dodaje ili oduzima jedan atribut. Postupak koji počinje s praznim skupom atributa, a svaki korak postupka dodaje po jedan novi atribut naziva se *pozitivna selekcija*. Obrnuti postupak koji počinje s punim skupom atributa i u svakom koraku oduzima po jedan atribut naziva se *negativna eliminacija*. Iako su ovi postupci vrlo jednostavni, pokazalo se da daju rezultate usporedive sa složenijim tehnikama pohlepnog pretraživanja kao što su zrakasto pretraživanje ili metoda najboljeg prvog. Ukoliko se sa n označi ukupan broj atributa, pozitivna selekcija i negativna eliminacija imaju složenost $O(n^2)$. Budući da proizvode prihvatljive rezultate u razumnom vremenu, upravo ove dvije tehnike pretraživanja se najčešće koriste u odabiru atributa metodama omotača.
Općenito gledano, negativna eliminacija preferira veće podskupove atributa i može rezultirati nešto boljim klasifikacijskim performansama od pozitivne selekcije. Razlog tome leži u činjenici da se vrijednost skupa atributa mjeri procjenom ispravnosti klasifikacije, pa zbog samo jedne optimistične procjene oba postupka mogu preuranjeno završiti. U tom slučaju negativna eliminacija će odabrati previše atributa, a pozitivna selekcija premalo. Nedostatak prognostičkih atributa može ograničiti sposobnost razlučivanja, što se odražava nešto slabijim klasifikacijskim performansama. S druge strane, manji broj odabranih atributa je poželjan u slučajevima kada je primarni cilj razumijevanje međuovisnosti i pravilnosti u podacima, jer su konstruirani modeli jednostavniji i naglašavaju najprediktivnije atribute.
Najvažniji nedostatak metoda omotača u odnosu na filtere je sporost pri izvođenju uzrokovana višekratnim pozivanjem ciljnog algoritma strojnog učenja. Iz tog razloga metodama omotača ne odgovaraju obimni skupovi podataka za učenje s većim brojem atributa.
Općenito se drži da zbog tijesne povezanosti s ciljnim algoritmom strojnog učenja, metode omotača omogućuju postizanje nešto boljih performansi klasifikacije. S druge strane to predstavlja opasnost jer pretjerana prilagodba skupa za učenje ciljnom algoritmu može apostrofirati njegove nedostatke.

4.2.5.2. METODE FILTERA

Za razliku od omotača, metode filtera funkcioniraju neovisno o odabranom algoritmu strojnog učenja. Vrijednost atributa se heuristički procjenjuje analizom općenitih karakteristika podataka iz skupa za učenje. Budući da postoji više načina heurističkog vrednovanja atributa, metode filtera broje više različitih tehnika odabira atributa.
Metode filtera se dijele u dvije osnovne skupine, ovisno o tome vrednuje li korištena heuristika podskupove atributa ili pojedinačne atributa.
Kod metoda čija heuristika vrednuje pojedinačne atribute, odabir atributa se vrši rangiranjem atributa prema ocjeni vrijednosti koju proizvodi heuristika. U podskup izabranih atributa ulaze oni atributi čija ocjena vrijednosti prelazi neki unaprijed odabrani prag. Podskup izabranih atributa može se formirati i tako da se unaprijed odredi broj (ili relativni omjer) atributa koje podskup treba sadržavati, pa se odgovarajući atributi pročitaju s vrha rangirane liste.



Slika 6: Odabir atributa metodama filtera

Najpoznatija metoda filtriranja kod koje se vrednuju pojedinačni atributi nazvana je *Relief* [34], a originalno je zamišljena za klasifikacijske probleme sa samo dvije klase. Relief je iterativna metoda koja u svakoj iteraciji korigira podatak o važnosti pojedinog atributa. Inicijalno, svi atributi imaju jednake težinske vrijednosti. U svakoj iteraciji postupka se na slučajan način odabire jedan primjer iz skupa podataka za učenje. Zatim se u skupu podataka za učenje pronalaze njegovi najbliži susjedi iz iste i suprotne klase. Na osnovu uspoređivanja vrijednosti svakog atributa u odabranom primjeru i pronađenim susjedima ažuriraju se težinske vrijednosti atributa. Načelno, važni atributi bi trebali imati bliske vrijednosti za primjere iste klase, a različite za primjere suprotnih klasa. Stoga se različite vrijednosti atributa za primjere suprotnih klasa boduju pozitivno, a za primjere iste klase negativno. Opisani postupak se ponavlja unaprijed definirani broj puta. Na kraju, težinske vrijednosti atributa predstavljaju ocjenu njihove vrijednosti. Općenito, pouzdanost ocjene vrijednosti raste s brojem iteracija, ali se produžava i vrijeme izvođenja.
Naknadno je ovaj postupak proširen (*ReliefF*) na probleme s više klasa i dodan je mehanizam tretiranja šuma u podacima [39]. ReliefF umanjuje utjecaj šuma u podacima usrednjavanjem doprinosa k najbližih susjeda iz iste i različitih klasa za svaki slučajno odabrani primjer. Problemi s više klasa tretiraju se promatranjem najbližih susjeda iz svih preostalih klasa, te se njihov utjecaj uzima u ovisnosti o apriornoj vjerojatnosti svake od klasa.
Negativna strana ReliefF metode je proizvoljnost u odabiru primjera. Svako pokretanje postupka vrednovanja atributa proizvest će različite težinske vrijednosti zbog drugačijeg odabira primjera.

Općeniti nedostatak filtera koji vrednuju pojedinačne atribute je nemogućnost detekcije redundantnih atributa. Koreliranost nekog atributa s drugim rezultira sličnom ocjenom vrijednosti za oba atributa, pa će u pravilu oba atributa biti prihvaćena ili odbačena. Drugi nedostatak leži u činjenici da je uvrštavanje atributa u konačni podskup prepušteno vanjskim kriterijima praga vrijednosti ili broja atributa.

Metode filtera čija heuristika vrednuje podskupove atributa ne pate od problema uvrštavanja atributa u konačni podskup, jer rezultat kojeg vraćaju nije rangirana lista pojedinačnih atributa već najbolje rangirani podskup atributa. Osim toga, budući da se razmatraju podskupovi atributa, moguće je provjeravati i redundantnost atributa u podskupu. Da bi se takvi atributi eliminirali, potrebno je konstruirati heurističku funkciju vrednovanja na način da penalizira postojanje redundantnih atributa u promatranom podskupu.
Budući da heuristika vrednuje podskupove atributa, potrebno je pronaći podskup koji maksimizira heurističku funkciju vrednovanja. Kao i kod metoda omotača, iscrpno pretraživanje svih podskupova skupa atributa je nepraktično, pa se koriste i slični načini pretraživanja. Pozitivna selekcija i negativna eliminacija i ovdje daju prihvatljive rezultate, a često se koristi i metoda najboljeg prvog. Zbog potrebe pretraživanja podskupova atributa, metode filtera koje vrednuju podskupove atributa su vremenski zahtjevnije od filtera koji vrednuju pojedinačne atribute. Ipak, ti zahtjevi su neusporedivo manji u usporedbi s metodama omotača jer se u svakom koraku pretraživanja izračunava samo vrijednost heuristike vrednovanja, a nije potrebno višekratno pozivanje algoritma strojnog učenja.

Zbog činjenice da se oslanjaju na opća svojstva podataka, heuristike koje vrednuju podskupove atributa često imaju uporište u statističkim postupcima analize podataka. Jedna od takvih heuristika zasniva se na procjeni korelacije među različitim atributima. Za svaki atribut promatranog podskupa ocjenjuje se korelacija s atributom klase, kao i međusobna korelacija atributa u podskupu. Vrijednost heurističke funkcije raste s povećanjem korelacije atributa i klase, a opada ako se povećava međusobna koreliranost atributa. Stoga će nevažni atributi biti odbačeni jer nisu korelirani s klasom, a redundantni zbog visoke korelacije s preostalim atributima podskupa.
Osim na statističkim postupcima, heuristika se može zasnivati i na postupcima strojnog učenja. Već prije je spomenuto da neki algoritmi strojnog učenja imaju ugrađene mehanizme odabira atributa. Mehanizmi koji su inherentni jednom postupku mogu se iskoristiti i u drugim postupcima kod kojih takvi mehanizmi ne postoje. Tipičan primjer je upotreba stabala odlučivanja za odabir atributa. Na potpunom skupu podataka za učenje konstruira se stablo odlučivanja, te se odabiru samo oni atributi koji se zaista koriste u konstruiranom stablu. Naravno, ovakav postupak ima smisla samo ako se u fazi modeliranja koristi drugi algoritam strojnog učenja. Postupak je opravdan ako taj algoritam na reduciranom skupu podataka za učenje pokaže bolje klasifikacijske performanse od algoritma korištenog za odabir podataka.

Općenito govoreći, odabir atributa metodama filtera traje znatno kraće u usporedbi s metodama omotača, posebno kad su u pitanju skupovi podataka s većim brojem atributa [24]. Stoga su za inteligentnu analizu podataka filteri često praktičnije

rješenje. Osim toga, zbog neovisnosti o algoritmu strojnog učenja mogu se koristiti u kombinaciji s bilo kojom tehnikom modeliranja podataka, za razliku od omotača koji se moraju ponovo izvoditi pri svakoj promjeni ciljne tehnike modeliranja.

4.2.6. Diskretizacija numeričkih atributa

Neki algoritmi strojnog učenja nisu prilagođeni radu s numeričkim atributima, tako da operiraju samo nad nominalnim atributima. Da bi se takvi algoritmi koristili za analizu općenitih skupova podataka, numeričke atribute je potrebno diskretizirati, odnosno predstaviti ih u obliku manjeg broja disjunktnih intervala. Čak i algoritmi prilagođeni numeričkim atributima ponekad rade bolje ukoliko se numerički atributi na zadovoljavajući način diskretiziraju. Dobrobiti se mogu ogledati u brzini rada (neki algoritmi opetovano sortiraju vrijednosti numeričkih atributa) ili u klasifikacijskim performansama (implicitne pretpostavke o distribuciji numeričkih atributa uvijek ne odgovaraju stvarnom stanju).
Neovisno o korištenoj tehnici diskretizacije, postoje dva različita načina prikaza diskretiziranih podataka. Jednostavniji način svaki interval diskretizacije tretira kao jednu vrijednost nominalnog atributa: rezultat je jedan nominalni atribut s domenom čija kardinalnost odgovara broju intervala diskretizacije. Međutim, ovaj način prikaza odbacuje potencijalno korisne informacije o uređaju među intervalima diskretizacije – među nastalim intervalima diskretizacije postoji implicitan uređaj induciran uređajem u numeričkoj domeni.
Drugi način prikaza čuva informacije o uređaju među intervalima. Iz svakog diskretiziranog numeričkog atributa nastaje k–1 binarnih atributa, gdje je k broj intervala diskretizacije. Ako vrijednost numeričkog atributa nekog primjera spada u i-ti interval diskretizacije, vrijednosti prvih i–1 binarnih atributa se postavljaju na $\bot$, a vrijednosti preostalih atributa na $\top$. Drugim riječima, i-ti binarni atribut odgovara podatku je li i-ti interval veći od intervala kojem pripada vrijednost numeričkog atributa u primjeru. Ukoliko algoritam strojnog učenja iskoristi neki od ovako formiranih binarnih atributa, implicitno je iskoristio informaciju o uređaju koju atribut sadrži.

Dva su osnovna tipa diskretizacije numeričkih atributa, a razlikuju se po korištenju podatka o klasi primjera iz skupa podataka za učenje [15]:

- *neinformirana diskretizacija* ne koristi podatak o klasi primjera, dok
- *informirana diskretizacija* uzima u obzir klasifikaciju primjera.

Neinformirana diskretizacija je jedina mogućnost kada se radi o problemima segmentiranja, pa podatak o klasi ne postoji. Za klasifikacijske probleme neinformirana diskretizacija nije dobar izbor, jer ignoriranje podatka o klasi primjera nosi ozbiljne nedostatke.
Najjednostavniji oblik neinformirane diskretizacije je *metoda jednakih intervala*: raspon vrijednosti numeričkog atributa dijeli se na unaprijed definirani broj intervala jednake širine. Ovisno o distribuciji vrijednosti numeričkog atributa može se dogoditi da u neke intervale ne upada niti jedan primjer, dok drugi sadrže velik broj primjera.
Nasuprot tome, *metoda jednakih frekvencija* inzistira na ravnomjernoj raspodjeli primjera u intervale diskretizacije. Ova metoda određuje granice intervala ovisno o distribuciji vrijednosti numeričkog atributa, i to na način da svaki interval sadrži (uvjetno rečeno) jednak broj primjera iz skupa za učenje. Stoga rezultirajući intervali

neće biti jednake širine (osim u umjetnom slučaju uniformne distribucije). Iz očitih razloga, ova metoda se još naziva i *izjednačavanje histograma*.
Klasifikacijski algoritmi strojnog učenja od atributa očekuju visoku prediktivnu sposobnost, pa bi temeljni kriterij određivanja intervala trebala biti homogenost primjera koje oni sadrže. Metode neinformirane diskretizacije ne mogu uvažiti ovaj kriterij jer ne koriste podatak o klasifikaciji primjera. Stoga se za klasifikacijske probleme u praksi koriste isključivo metode informirane diskretizacije.

4.2.6.1. ENTROPIJSKA DISKRETIZACIJA

Entropijska diskretizacija [17] spada u skupinu informiranih metoda diskretizacije numeričkih atributa. To je u osnovi rekurzivan postupak u kojem se inicijalni raspon vrijednosti numeričkog atributa dijeli na dva podintervala, na koje se opet primjenjuje isti postupak. Točka podjele bira se tako da rezultirajući podintervali imaju što veću homogenost u terminima klasifikacije primjera koje sadrže. Kao mjera nehomogenosti intervala koristi se entropija, po čemu je postupak i dobio ime. Slika 7 detaljnije prikazuje postupak entropijske diskretizacije.

```
1.  set := skup podataka za učenje
2.  sortiraj primjere iz set prema vrijednosti numeričkog atributa A
3.  odredi raspon vrijednosti atributa A, [a_min, a_max]
4.  izvrši EntDiscr(a_min, a_max)
5.  vrati listu točaka podjele

postupak EntDiscr(min, max)
a.  za svaku potencijalnu točku podjele s izračunaj entropiju
    rezultirajućih intervala [min, s], [s, max]
b.  t := točka podjele s najmanjom entropijom rezultirajućih intervala
c.  ako nije dosegnut kriterij zaustavljanja onda
          c.1.  dodaj t u skup točaka podjele
          c.2.  izvrši EntDiscr(min, t)
          c.3.  izvrši EntDiscr(t, max)
```

Slika 7: Postupak entropijske diskretizacije

Entropija kao mjera nehomogenosti skupa preuzeta je iz teorije informacija. Smisao entropije je da odražava količinu informacije koja je potrebna da se specificira klasa bilo kojeg primjera iz promatranog skupa primjera. Entropija skupa primjera *S* dana je izrazom:

$$Entropy(S) = -\sum p_i \log_2 p_i \quad , \tag{24}$$

gdje sumacija ide po svim klasama zastupljenim u *S*, a p_i predstavlja apriornu vjerojatnost klase *i* u skupu S.
U postupku se računa ukupna entropija para intervala. Ona se dobije kao težinska suma entropija pojedinačnih intervala, s težinama koje odgovaraju udjelu primjera koji spadaju u pojedini interval. Budući da se traže što homogeniji intervali, odabire se ona točka podjele za koju je entropiju rezultirajućih intervala najmanja.
Kad se promatra sortirani skup primjera za učenje, teoretski se može pokazati da se točka podjele koja minimizira entropiju intervala nikad neće smjestiti između dva primjera iste klase. Iz toga se mogu iščitati potencijalne točke podjele: potrebno je razmatrati samo one vrijednosti koje razdvajaju primjere različitih klasa. U pravilu se kao točka podjele uzima aritmetička sredina vrijednosti atributa u tim primjerima.
Nakon što je pronađena najbolja točka podjele, interval se dijeli na dva podintervala na kojima se rekurzivno primjenjuje isti postupak. Potrebno je još razjasniti kriterij

zaustavljanja. On se temelji na *principu najkraćeg opisa* [51] koji se može shvatiti kao posebna formulacija Ockhamove oštrice. Primijenjen na entropijsku diskretizaciju, ovaj princip traži da ukupna količina informacije potrebne za specifikaciju intervala diskretizacije i nehomogenosti u njima bude što manja. Mjera nehomogenosti intervala je entropija, a intervali se specificiraju navođenjem točaka podjele. Da bi podjela intervala u nekom koraku rekurzije bila opravdana, nužno je da ukupna entropija dobivenih podintervala bude manja od entropije polaznog intervala. Princip najkraćeg opisa zahtjeva još i da razlika bude veća od količine informacije koja je potrebna za specifikaciju točke podjele. Konkretno, podjela intervala je opravdana ukoliko je razlika entropija veća od

$$\frac{\log_2(N-1)}{N} + \frac{\log_2(3^k-2) - kE + k_1E_1 + k_2E_2}{N} \quad , \qquad (25)$$

gdje N označava broj primjera, k broj klasa, a E entropiju primjera u polaznom intervalu. Sa k_1 i k_2 je označen broj klasa u podintervalima, a sa E_1 i E_2 entropija primjera u podintervalima. Prvi pribrojnik u izrazu odgovara količini informacije koja je potrebna za specifikaciju točke podjele, a drugi pribrojnik dodaje količinu informacije kojom se specificiraju klase koje odgovaraju pojedinim podintervalima.

4.2.6.2. OSTALE METODE INFORMIRANE DISKRETIZACIJE

Entropijska diskretizacija je jedna od najboljih metoda informirane diskretizacije. Međutim, postoje i druge metode koje diskretizaciji prilaze na nešto drugačiji način.

Primjerice, umjesto rekurzivne podjele intervala sve dok nije zadovoljen kriterij zaustavljanja, diskretizacija može teći obrnutim smjerom. Inicijalno se svakom primjeru može dodijeliti vlastiti podinterval, pa se u postupak diskretizacije sastoji od stapanja susjednih intervala kad god je to opravdano. U svakom koraku se statističkim tehnikama pronalaze dva intervala najpogodnija za stapanje, a odluka o stapanju se donosi ako statistički kriterij prijeđe unaprijed zadani prag pouzdanosti. Jedan od pogodnih kriterija je χ^2 test, koji se najčešće i koristi [32]. Postupak se ponavlja dok god je moguće pronaći potencijalno stapanje koje zadovoljava odabrani kriterij. Postoje i složenije varijante postupka u kojima se adekvatan prag pouzdanosti određuje automatski [42].

Druga skupina diskretizacijskih metoda homogenost intervala mjeri brojem nepoželjnih primjera. Naime, svakom intervalu diskretizacije se pridruži oznaka klase koja u njemu prevladava, pa je nehomogenost uzrokovana primjerima koji ne pripadaju toj klasi. U tom smislu su primjeri čija klasa ne odgovara klasi intervala nepoželjni, pa je osnovna mjera nehomogenosti diskretizacije ukupan broj takvih primjera. Budući da intervalima diskretizacije pridružuje oznaka klase, ovakav oblik diskretizacije naziva se *klasifikacijska diskretizacija*.

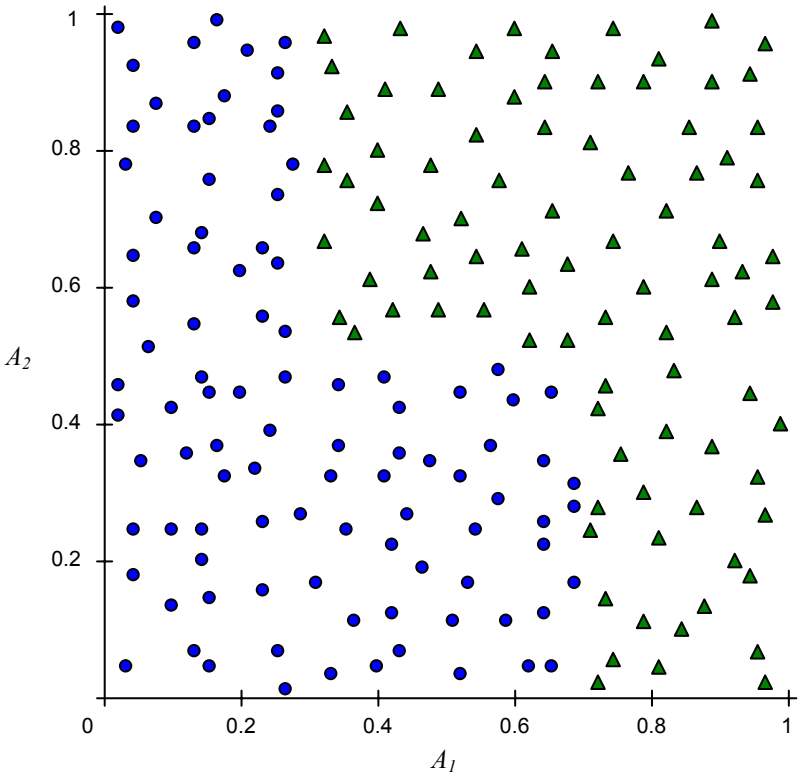
Najjednostavniji postupak klasifikacijske diskretizacije promatra skup podataka za učenje sortiran prema vrijednosti numeričkog atributa. Točke podjele se postavljaju između primjera različite klasifikacije, s iznimkom zahtjeva da svaki interval mora imati najmanje n primjera, gdje je n unaprijed zadan. Stoga su mogući intervali koji sadrže primjere različitih klasa. Ovakav postupak preferira diskretizacije s velikim brojem intervala, budući da se tako može postići manji broj nepoželjnih primjera.

U praksi se nominalni atributi čija domena ima prevelik broj vrijednosti izbjegavaju, pa se obično ograničava broj podintervala koje diskretizacija može proizvesti. Uz

unaprijed zadani broj intervala k, klasifikacijska diskretizacija se može promatrati kao optimizacijski problem: potrebno je pronaći particiju domene numeričkog atributa u k intervala koja minimizira broj nepoželjnih primjera. Najbanalniji postupak koji rješava ovaj problem je eksponencijalan u k, pa nije primjeren. Metode dinamičkog programiranja daju puno bolje rezultate, pa se zadani problem može svesti na postupak linearan u N, gdje je N ukupan broj primjera.
Inače, metodama dinamičkog programiranja se bilo koja mjera nehomogenosti intervala može minimizirati na k-članoj particiji u vremenu proporcionalnom s kN^2 [20].

4.2.6.3. PREDNOSTI ENTROPIJSKE U ODNOSU NA KLASIFIKACIJSKU DISKRETIZACIJU

Klasifikacijska diskretizacija se izvodi znatno brže od entropijske, ali u odnosu na nju ima jedan bitan nedostatak. Naime, klasifikacijska diskretizacija ne može proizvesti susjedne intervale koji bi bili označeni istom klasom. Razlog je jednostavan: spajanje takvih intervala ne utječe na broj nepoželjnih primjera, a oslobađa jedan interval koji drugdje može umanjiti njihov broj.
Iako se ovo na prvi pogled ne doima kao ozbiljan nedostatak, slijedeći primjer pokazuje suprotno [38]. Na slici 8 prikazan je univerzum kojeg razapinju dva numerička atributa A_1 i A_2 s domenama [0,1]. Skup podataka za učenje sadrži primjere dviju klasa C_1 i C_2, koji su na slici označeni krugovima odnosno trokutima.



Slika 8: Primjer skupa podataka za učenje

Odmah se vidi da primjeri pripadaju klasi C_1 ako im je vrijednost atributa A_1 manja od 0.3, ili je istodobno A_1 manji od 0.7 i A_2 manji od 0.5. Ostali primjeri pripadaju klasi C_2.

Stoga je domenu atributa A_1 najbolje diskretizirati intervalima [0, 0.3), [0.3, 0.7) i [0.7, 1], a domenu atributa A_2 intervalima [0, 0.5) i [0.5, 1]. Diskretizacija atributa A_2 ne predstavlja problem, ali vrijedi promotriti intervale diskretizacije atributa A_1. Naime, kad bi se intervalima pridruživale oznake prevladavajućih klasa, prvi interval bi dobio oznaku C_1, a zadnji oznaku C_2. Koju god oznaku dobije srednji interval, bit će susjedan s intervalom iste oznake (u primjeru na slici, to bi bila oznaka C_1). Stoga se ovakva diskretizacija atributa A_1 ne može dobiti metodama klasifikacijske diskretizacije, jer one ne mogu proizvesti susjedne intervale s istom oznakom klase.
Ono što sugerira točku podjele na vrijednosti 0.3 nije promjena prevladavajuće klase, već promjena njene distribucije. Iako je i lijevo i desno od točke podjele prevladavajuća klasa C_1, njena distribucija se dramatično mijenja sa 100% na nešto više od 50%. Za razliku od klasifikacijskih metoda, metoda entropijske diskretizacije je osjetljiva na promjene u distribuciji klase, pa može proizvesti traženu diskretizaciju atributa A_1.

4.2.7. Transformiranje nominalnih atributa u numeričke

Komplementaran problem diskretizaciji numeričkih atributa je prikaz nominalnih atributa numeričkim veličinama. Potreba za ovakvom transformacijom je rjeđa, ali ipak postoji. Naime, neke tehnike modeliranja podataka prirodno operiraju samo s numeričkim veličinama. U takve spada tehnika najbližih susjeda: primjeri se klasificiraju na osnovu udaljenosti u univerzumu primjera, a izračun udaljenosti uključuje računanje razlike vrijednosti atributa. Međutim, nominalne vrijednosti ne poznaju pojam razlike vrijednosti, pa ih je potrebno prikazati u numeričkom obliku. Doduše, u konkretnom primjeru tehnike najbližih susjeda problem se može riješiti i dodefiniranjem funkcije udaljenosti primjera.

Transformacija nominalnog atributa u numerički oblik se najčešće provodi na način da se nominalni atribut zamijeni s više sintetičkih binarnih atributa. Atribut čija domena broji k elemenata zamijenit će se s k binarnih atributa. Svaki atribut odgovara jednoj nominalnoj vrijednosti, te ima vrijednost 1 ako je odgovarajuća vrijednost prisutna u primjeru, a 0 inače. Ukoliko postoje neke suptilnije razlike među vrijednostima nominalne domene, one se mogu modelirati različitim težinskim vrijednostima binarnih atributa.

Ukoliko su vrijednosti nominalne domene linearno uređene, mogu se koristiti drugi načini transformacije koji čuvaju informaciju o uređaju. Jednostavnija varijanta mijenja nominalni atribut cjelobrojnim numeričkim atributom, uz preslikavanje vrijednosti koje čuva uređaj. Međutim, ovaj oblik transformacije implicira i metriku među vrijednostima atributa.
Ukoliko implicirana metrika nije poželjna, može se primijeniti druga varijanta transformacije prema kojoj se nominalni atribut zamjenjuje sa k–1 sintetičkih binarnih atributa, gdje je k kardinalnost domene nominalnog atributa. Vrijednost i-tog atributa se postavlja na 0 ako nominalna vrijednost u primjeru spada u prvih i vrijednosti, a na 1 inače. Drugim riječima, vrijednost i-tog atributa je 1 ako je promatrana vrijednost veća od i-te prema uređaju u nominalnoj domeni. Dakle, ova varijanta transformacije čuva informaciju o uređaju nominalnih vrijednosti, a ne implicira nikakvu metriku nad njima.

5. Tehnike modeliranja pravilnosti u podacima

Tehnike modeliranja u području inteligentne analize podataka porijeklo vuku iz različitih područja istraživanja. Osim strojnog učenja, na razvoj tehnika modeliranja utjecala su dostignuća iz obrade signala, raspoznavanja uzoraka, statistike, evolucijskog programiranja. Neovisno o porijeklu, sve tehnike modeliranja iskazuju jednu zajedničku karakteristiku: automatizirano otkrivanje novih veza i ovisnosti atributa u promatranim podacima. Autonomno otkrivanje međuovisnosti predstavlja osnovnu prednost u odnosu na tradicionalne statističke analize, koje su uglavnom namijenjene samo konfirmacijskoj ulozi pri testiranju unaprijed postavljenih hipoteza.

Spomenuto obilježje automatizma pri otkrivanju pravilnosti u podacima čini važnu sponu tehnika modeliranja sa postupcima strojnog učenja. Upravo ova činjenica uvjetuje da im je struktura međusobno vrlo slična, karakterizirana 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), 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).

Iako dijele istu osnovnu strukturu, tehnike modeliranja se značajno razlikuju po načinu na koji su pojedine od komponenti uobličene. Posljedica toga su različitosti u svojstvima koja karakteriziraju pojedine tehnike modeliranja. Opisi najpoznatijih tehnika modeliranja ilustriraju različitosti u osnovnim strukturnim komponentama, te ukazuju na razlike u svojstvima koje iz toga proizlaze. Iako opisi ne ulaze u sve detalje implementacije pojedine tehnike modeliranja, naznačeni su osnovni problemi na koje se nailazi u praksi, te načini njihova zaobilaženja.

5.1. STABLA ODLUČIVANJA

5.1.1. 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.
Algoritam konstrukcije stabla odlučivanja operira nad prostorom pretraživanja koji se sastoji od svih stabala koje je moguće konstruirati koristeći atribute i vrijednosti iz skupa podataka za učenje. Operacija transformacije kojom se prelazi prostor pretraživanja modificira stablo na način da jedan od listova zamijeni podstablom visine 1.
Funkcija vrednovanja koja usmjerava postupak pretraživanja ovisi o točnosti klasifikacije, ali i veličini rezultirajućeg stabla. Prema principu Ockhamove oštrice, uz sličnu točnost klasifikacije, izglednije je da će jednostavnija (tj. manja) stabla odlučivanja bolje klasificirati dotad neviđene primjere. Način djelovanja funkcije vrednovanja zasnovan je na konceptima iz teorije informacija, a ogleda se u odabiru grananja pri konstrukciji stabla odlučivanja.

5.1.2. Postupak pretraživanja

Temeljni algoritam konstrukcije stabala odlučivanja star je nekoliko desetljeća, a razvio ga je J.Ross Quinlan [48]. 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* [49], odnosno njegova komercijalna (ponešto unaprijeđena) verzija *C5.0*.

Algoritam konstrukcije stabala odlučivanja pretraživanju pristupa po načelu *odozgo na dolje*, tj. od općenitijeg ka specifičnom. Koristi se strategija nepovratnog pretraživanja, i to pohlepna metoda uspona na vrh. Stoga je postupak podložan zamci lokalnih maksimuma, ali je pretraživanje relativno brzo jer se pregledava samo manji dio prostora pretraživanja.
Osnovni algoritam je u svojoj osnovi rekurzivan i sastoji se od četiri osnovna slučaja. Neka je sa S označen skup podataka za učenje, sa $C=\{C_i, 1 \le i \le |C|\}$ skup odgovarajućih klasa, a sa $A=\{A_i, 1 \le i \le |A|\}$ skup odgovarajućih atributa. Osim toga, neka su $S' \subseteq S$ i $A' \subseteq A$ parametri koji se prosljeđuju algoritmu, s tim da se inicijalno u rekurziju ulazi sa $S' = S$ i $A' = A$. Osnovni koraci algoritma su:

1. Ako je S' prazan, stablo odlučivanja je list označen globalno najfrekventnijom klasom C_i unutar S.
2. Ako se S' sastoji od primjera samo jedne klase C_j, stablo odlučivanja je list označen klasom C_j.
3. Ako je A' prazan, stablo odlučivanja je list označen najfrekventnijom klasom C_k unutar S'.
4. Inače, odaberi atribut $A_i \in A'$. Neka je $\{a_j, 1 \le j \le n\}$ skup svih vrijednosti atributa A_i. Particioniraj S' na n podskupova S'_j tako da S'_j sadrži sve primjere iz S' kod kojih atribut A_i ima vrijednost a_j, tj. $S'_j = \{s \in S', A_i(s) = a_j\}$. Stvori unutarnji čvor označen atributom A_i, te grane označene njegovim vrijednostima a_j. Za granu označenu vrijednošću a_j konstruiraj podstablo rekurzivnim pozivom postupka s parametrima S'_j i $A' - \{A_i\}$.

Dakle, algoritam 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. Stoga se svaki atribut može pojaviti najviše jednom na bilo kojem putu od korijena do lista. Vrijedi primijetiti da se u ovom osnovnom obliku algoritma implicitno pretpostavlja da su svi atributi nominalnog tipa.
Rekurzija se nastavlja sve dok za promatrani čvor stabla nije zadovoljen jedan od dva kriterija:

- skup podataka za učenje koji pripada čvoru je prazan ili se sastoji od primjera samo jedne klase (pa je daljnje grananje nepotrebno), ili
- svi atributi su već iskorišteni na putu od korijena do promatranog čvora (pa daljnje grananje nije moguće).

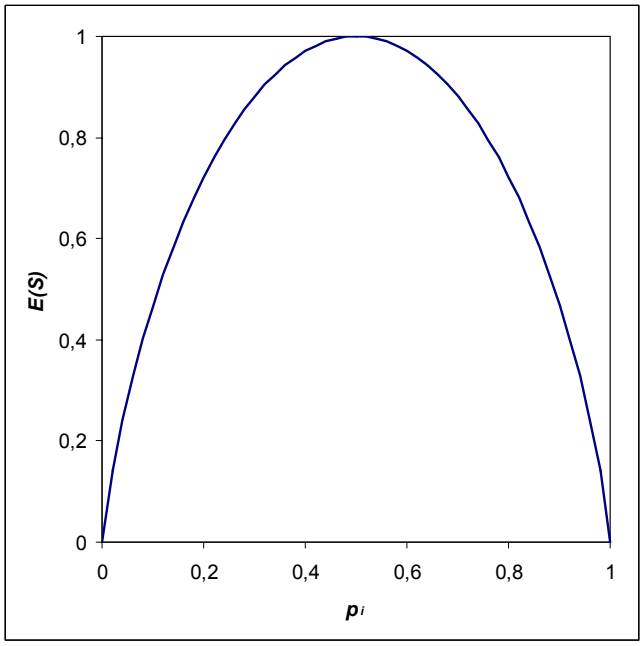
Iz opisa algoritma očigledna je nepovratna strategija pretraživanja: jednom odabrani atribut postaje osnova za grananje stabla, bez mogućnosti da se taj odabir naknadno preispita. Međutim, uopće nije bilo riječi o načinu odabira atributa. Iako bi bilo koji odabir atributa u konačnici rezultirao konačnim stablom odlučivanja koje dobro klasificira primjere iz skupa za učenje, način odabira atributa je presudan za kvalitetu konačnog rezultata. Naime, struktura stabla odlučivanja ovisi isključivo o odabiru atributa za grananje u svakom koraku rekurzije. Stoga kriterij odabira atributa za grananje predstavlja centralni dio algoritma, koji usmjerava pretraživanje u skupu potencijalnih rješenja. O načinu odabira atributa ovisi hoće li rezultirajuće stablo biti glomazna struktura pretjerano prilagođena skupu za učenje, ili kompaktni prikaz općenitih pravilnosti koje postoje u podacima.

5.1.3. 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. Budući da je cilj konstrukcije što manje stablo odlučivanja, treba nastojati što ranije zaustaviti rekurzivni proces grananja. 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 za klasifikacijski problem sa dvije klase sa p_1 označena relativna frekvencija klase C_1, a sa p_2 relativna frekvencija klase C_2 u skupu primjera za učenje S. Entropija skupa za učenje tada je dana sa:

$$E(S) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \quad , \qquad (26)$$

uz pretpostavku da vrijedi $0 \log_2 0 = 0$. Budući da je $p_1, p_2 \leq 1$ (jer $p_1 + p_2 = 1$), logaritmi u izrazu (26) su negativni, pa je entropija uvijek veća ili jednaka 0. Važno je primijetiti da minimalnu vrijednost 0 entropija poprima kada svi primjeri iz skupa za učenje S pripadaju istoj klasi, tj. kada je $p_1 = 1$ ili $p_2 = 1$. Entropija dostiže maksimalnu vrijednost 1 kada skup za učenje S sadrži jednak broj primjera obiju klasa, tj. kada je $p_1 = p_2$.
Za klasifikacijski problem sa dvije klase, slika 9 prikazuje funkciju entropije u ovisnosti o relativnoj frekvenciji jedne od klasa u skupu podataka za učenje.



Slika 9: Entropija binarne klasifikacije

Jedinice u kojima se mjeri entropija nazivaju se *bitovi*. Jedna od interpretacija entropije iz teorije informacija govori da ona specificira minimalnu količinu informacije (izraženu u bitovima) potrebne da se kodira klasifikacija slučajno odabranog primjera iz skupa S, uz činjenicu da primjer pripada promatranom čvoru.

Za klasifikacijske probleme s više klasa potrebno je uopćiti definiciju entropije, uz zadržavanje poželjnih svojstava vezanih uz dosezanje minimuma i maksimuma. 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 (27) definira entropiju skupa primjera S.

$$E(S) = \sum_{i=1}^{|C|} - p_i \log_2 p_i \qquad (27)$$

Treba primijetiti da maksimalna vrijednost entropije u ovom slučaju iznosi $log_2 |C|$, a postiže se pri jednakoj zastupljenosti svih klasa unutar S. Minimum entropije i dalje iznosi 0, za slučaj kada svi primjeri iz S pripadaju istoj klasi.

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:

$$IGain(S, A_i) = E(S) - \sum_{a_j \in Dom(A_i)} \frac{|S_j|}{|S|} E(S_j) \quad , \qquad (28)$$

gdje $S_j \subseteq S$ označava skup $S_j = \{s \in S,\ A_i(s) = a_j\}$.
Prvi član izraza (28) predstavlja entropiju originalnog skupa S, a težinska suma u drugom članu iskazuje očekivanu vrijednost entropije podskupova nastalih grananjem na osnovu atributa A_i. Dakle, informacijski dobitak $IGain(S, A_i)$ predstavlja očekivanu redukciju entropije (tj. dobitak na homogenosti) uzrokovanu poznavanjem vrijednosti atributa A_i.
U algoritmu konstrukcije stabala odlučivanja kao kriterij odabira atributa koristi se upravo informacijski dobitak, budući da se grananjem nastoji što ranije postići homogenost rezultirajućih podskupova. Stoga se u svakom čvoru od dostupnih atributa za grananje odabire onaj koji proizvodi najveći informacijski dobitak. Treba primijetiti da ova heuristika zasnovana na teoriji informacija ne garantira konstrukciju najmanjeg stabla odlučivanja, ali dobro ispunjava svoju ulogu redukcije opsega pretraživanja u skupu potencijalnih rješenja.

Poznato svojstvo informacijskog dobitka je da favorizira atribute s većim brojem vrijednosti, što pri konstrukciji stabala odlučivanja može predstavljati problem. Ekstremni primjer atributa koji ima različitu vrijednost za svaki primjer iz skupa za učenje (npr. matični broj osobe) najbolje ilustrira takvu situaciju. Grananje na osnovu ovog atributa particionira skup za učenje na jednočlane podskupove, pa je svaki od podskupova homogen u smislu klase primjera. Stoga je entropija takvog grananja 0, pa je informacijski dobitak maksimalan. Rezultat je stablo koje se sastoji samo od

korijena i listova za svaki od primjera iz skupa za učenje. Nepotrebno je naglašavati da je takvo stablo beskorisno u smislu prediktivnih sposobnosti.
Korekcija kriterija za odabir atributa koja smanjuje utjecaj ovog problema u obzir uzima broj i kardinalnost podskupova koji nastaju kao rezultat grananja. Tako se pri konstrukciji stabala odlučivanja kao kriterij odabira atributa u pravilu koristi korigirana mjera dobitka (29):

$$Gain(S, A_i) = \frac{IGain(S, A_i)}{-\sum_{a_j \in Dom(A_i)} \frac{|S_j|}{|S|} \log_2 \frac{|S_j|}{|S|}} \quad . \tag{29}$$

Korektivni faktor u izrazu (29) je nazivnik, koji raste s povećanjem broja rezultirajućih podskupova odnosno smanjenjem njihove kardinalnosti. Na taj način se penaliziraju atributi s većim brojem vrijednosti.
Postoji mogućnost da primjena ove heurističke korekcije informacijskog dobitka u nekim slučajevima pretjerano preferira atribute s manjim brojem vrijednosti, na štetu informacijski vrijednijih atributa. Standardna provjera kojom se ova mogućnost eliminira traži da atribut koji maksimizira korigirani dobitak mora imati informacijski dobitak jednak ili veći od prosjeka svih promatranih atributa.

5.1.4. Šum u podacima (podrezivanje)

Opisani algoritam konstrukcije stabala odlučivanja nastoji konstruirati stablo koje savršeno klasificira primjere iz skupa podataka za učenje. U tome ne uspijeva samo u granama za koje se rekurzija zaustavlja zbog nedostatka atributa za grananje (korak 3 u opisu algoritma). U tom slučaju skup primjera S' koji odgovara promatranom listu sadrži više od jedne klase, pa se list označava najfrekventnijom klasom među njima. Alternativa je probabilistička interpretacija, po kojoj se list označava svim klasama iz S', uz pridruživanje pripadajuće vjerojatnosti svakoj od njih. Pridružena vjerojatnost odgovara relativnoj frekvenciji klase unutar S'.
Dakle, dok god postoje mogući atributi za grananje, algoritam će razgranavati stablo nastojeći akomodirati svaki primjer iz skupa podataka za učenje. Međutim, savršena klasifikacijska primjera iz skupa podataka za učenje ne garantira dobre klasifikacijske performanse na dotad neviđenim primjerima, što je cilj izgradnje modela. Uzrok tome može biti šum u podacima za učenje (bilo u prognostičkim atributima ili u klasi), ili činjenica da skup podataka za učenje nije reprezentativan uzorak cijele populacije primjera. Posljedica će biti pretjerano razgranato stablo odlučivanja zbog grananja na atributima koji samo prividno proizvode informacijski dobitak, dok je stvarni uzrok grananja šum u primjerima za učenje. Ovakva pojava naziva se pretjerana prilagođenost podacima za učenje (engl. *overfitting*).
Ilustrativan primjer grananja koje degradira klasifikacijske performanse na novim primjerima uključuje promatranje slučajno generiranih klasa. Neka su primjeri iz skupa podataka za učenje S slučajno raspoređeni u klase C_1 i C_2, tj. tako da ne postoji korelacija između prognostičkih atributa i klase primjera. Neka je relativna frekvencija klase C_1 unutar S označena sa p, a klase C_2 sa $1–p$. Bez smanjenja općenitosti može se pretpostaviti $p \geq 0.5$. Očekivana frekvencija grešaka najjednostavnijeg stabla odlučivanja koje se sastoji samo od korijena označenog klasom C_1 iznosi $1–p$, budući da takvo stablo svaki primjer klasificira u klasu C_1.

Nasuprot tome, promotrimo stablo nastalo grananjem koje se sastoji od korijena i dva lista označena klasama C_1 i C_2. Neka test u korijenu stabla funkcionira na načina da primjeru pridjeljuje klasu C_1 s vjerojatnošću q, a klasu C_2 s vjerojatnošću $1-q$. Tada je očekivana frekvencija grešaka takvog stabla

$$q(1-p)+(1-q)p = p+q-2pq \quad . \tag{30}$$

Međutim, vrijedi

$$1-p \le p+q-2pq \tag{31}$$

za svaki q, uz $p \ge 0.5$.
Stoga će u ovom slučaju klasifikacijske performanse jednostavnog nerazgranatog stabla s jednim čvorom nadmašiti bilo koje binarno stablo s 3 čvora.

Dva su načina izbjegavanja nepotrebnog grananja stabla:
- zaustavljanje grananja prilikom konstrukcije stabla odlučivanja prije postizanja savršene klasifikacije primjera iz skupa za učenje,
- naknadno podrezivanje razgranatog stabla koje se provodi nakon procesa konstrukcije stabla odlučivanja.

Oba pristupa 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. Primjerice, odvojeno promatrana dva atributa mogu se činiti gotovo nevažnima, a da njihova kombinacija ima izrazite prediktivne sposobnosti. Ovakva situacija pogoduje pristupu podrezivanja, koje kao podlogu uzima potpuno razgranato stablo.

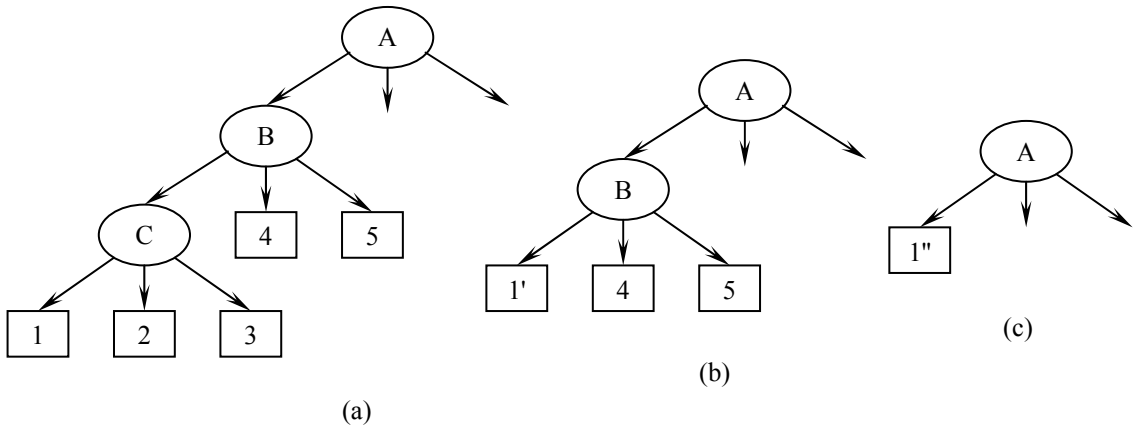
Kriteriji za zaustavljanje grananja prilikom konstrukcije stabala odlučivanja uglavnom se oslanjaju na statističke tehnike ocjene relevantnosti atributa odabranog za grananje. Često se koristi χ^2 test, kojim se nastoji utvrditi statistička neovisnost vrijednosti atributa A_i i klase primjera u skupu za učenje S. Neka je sa $p_k(S)$ označena relativna frekvencija klase C_k unutar S, a sa $p'_k(S_j)$ očekivana relativna frekvencija klase C_k unutar S_j, gdje je $S_j = \{s \in S,\ A_i(s) = a_j\}$. Ako su vrijednosti atributa A_i i klase neovisne, tada za sve skupove S_j koji nastaju kao rezultat grananja na atributu A_i vrijedi $p'_k(S_j) = p_k(S)$. Tada izraz (32) ima χ^2 distribuciju sa $|Dom(A_i)|-1$ stupnjeva slobode.

$$\sum_{a_j \in Dom(A_i)} \sum_{k=1}^{|C|} \frac{\left(p_k(S_j) - p'_k(S_j)\right)^2}{p'_k(S_j)} \tag{32}$$

Koristeći (32), u algoritmu konstrukcije stabala odlučivanja zaustavljanje grananja se provodi na način da se u obzir uzimaju samo oni atributi čija neovisnost u odnosu na klasu može biti odbačena s vrlo visokom pouzdanošću.

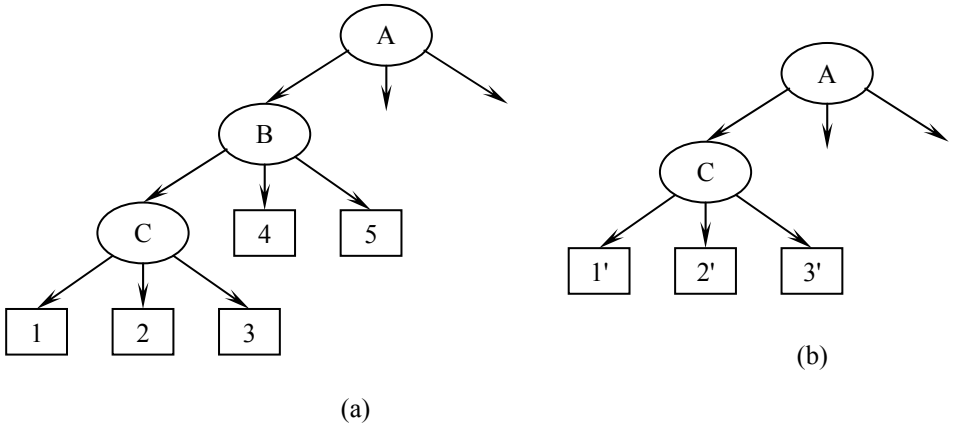
Dvije su najpoznatije tehnike podrezivanja stabala odlučivanja: *zamjena podstabala* i *izdizanje podstabala*, a primjena jedne od njih ne isključuje drugu.

Zamjena podstabala je jednostavnija operacija, koja može cijelo podstablo reducirati na jedan list, pri čemu se mijenjaju vjerojatnosti klasa u listu. Postupak se provodi nad unutrašnjim čvorovima stabla koji kao djecu imaju samo listove, tj. promatraju se samo podstabla visine 1. Za svako takvo podstablo razmatra se opravdanost zamjene samo jednim listom, uz pripadajuću korekciju razdiobe vjerojatnosti među klasama. Postupak napreduje od listova prema korijenu stabla, tako da se uzastopnom primjenom listom mogu zamijeniti podstabla proizvoljne visine.
Slika 10 na primjeru ilustrira uzastopnu primjenu zamjene podstabala na čvorovima *C* i *B*, tako da je cijelo podstablo s korijenom *B* na slici (a) u konačnici zamijenjeno jednim listom *1″* na slici (c).



Slika 10: Primjer zamjene podstabala

Izdizanje stabala je složenija operacija podrezivanja, a provodi se nad unutrašnjim čvorovima koji kao djecu imaju barem jedan čvor koji nije list. Odabire se jedno od djece takvoga čvora, te se njime zamjenjuje polazni čvor. Na taj način je cijelo podstablo pod odabranim djetetom izdignuto na jednu razinu više u odnosu na polazno stablo. Budući da ovim postupkom efektivno nestaju ostala djeca polaznog čvora, primjere koji su pripadali njima potrebno je reklasificirati u novonastalo podstablo.



Slika 11: Primjer izdizanja podstabala

Primjer na slici 11 prikazuje stablo odlučivanja prije i poslije primjene izdizanja podstabla na čvoru *B*. Čvor *B* se zamjenjuje djetetom *C*, a primjeri koji su pripadali ostaloj djeci čvora *B* reklasificiraju se u novo podstablo s korijenom *C*. Naravno, reklasifikacija utječe na vjerojatnosti pridružene klasama u listovima novog podstabla.
Upravo zbog potrebe za reklasifikacijom, izdizanje stabala je vremenski potencijalno zahtjevna operacija. Zbog toga se u stvarnim implementacijama izdizanja podstabla kao kandidat za izdizanje razmatra samo najpopularnije dijete polaznog čvora, tj. ono kojemu pripada najviše primjera za učenje.

U svakoj tehnici podrezivanja stabla odlučivanja ključan element je kriterij na osnovu kojeg se odlučuje treba li potencijalnu operaciju podrezivanja zaista i provesti. Budući da je cilj podrezivanja smanjenje frekvencije grešaka na dotad neviđenim primjerima, nužno je procijeniti stvarnu frekvenciju grešaka u svakom od čvorova stabla. Odluka o provođenju operacije podrezivanja tada se svodi na usporedbu procijenjene frekvencije grešaka za originalno podstablo i njegovu predloženu alternativu dobivenu podrezivanjem.
Postoji više načina procjene stvarne frekvencije grešaka u čvorovima stabla odlučivanja. Najčešće se koristi standardna verifikacijska tehnika na odvojenom skupu primjera. Izvorni skup primjera za učenje se dijeli na dva dijela:

- skup koji će služiti za generiranje stabla odlučivanja
- validacijski skup koji će služiti za provjeru opravdanosti operacije podrezivanja stabla.

Mana ovog pristupa leži u činjenici da se stablo odlučivanja konstruira iz manjeg broja primjera, zbog izdvajanja validacijskog skupa. Postoje i druge metode koje se oslanjaju na skup podataka za učenje, a imaju heuristički pristup ocjeni stvarne frekvencije grešaka. Iako se koriste nekim statističkim izračunima, statistička utemeljenost postupka je upitna, zbog korištenja istog skupa podataka za učenje na osnovu kojeg je izgrađeno stablo. Međutim, u praksi pokazuju dobre rezultate pri određivanju opsežnosti operacija podrezivanja, što opravdava njihovu primjenu.

5.1.5. Numerički atributi

Osnovni oblik ID3 algoritma za konstrukciju stabla odlučivanja ograničen je na nominalne atribute. Iako je moguće korištenje numeričkih atributa uz prethodnu diskretizaciju, kasnije verzije algoritma na jednostavan način uvode efikasnu podršku radu s numeričkim atributima.
Grananje stabla korištenjem nominalnog atributa rezultira granom za svaku od mogućih vrijednosti promatranog atributa. Budući da se takav oblik testa ne može primijeniti na numeričke atribute, kod njih se u pravilu testiranje vrijednosti ograničava na binarni test oblika $A_i(s) < x$, gdje je x konstanta odabrana za taj test. Unutrašnji čvor stabla koji odgovara ovakvom testu ima dvije izlazne grane: jednu za pozitivni, a drugu za negativni ishod testa.
Pripadajući informacijski dobitak se za ovakvo grananje računa na standardni način, uz napomenu da notacija sumacije po različitim vrijednostima atributa nije prikladna, već se sumira po (binarnoj) particiji skupa za učenje. Na ovaj način opisani testovi na numeričkim atributima sudjeluju u procesu odabira atributa za grananje, zajedno s ostalim atributima koji su na raspolaganju pri konstrukciji stabla odlučivanja.

Ostaje samo pitanje odabira granice interesantnih intervala za formiranje testa na numeričkom atributu, tj. vrijednosti konstante x. Iako je kardinalnost domene numeričkog atributa beskonačna, u skupu podataka za učenje pojavljuje se samo konačan skup njegovih vrijednosti. Uobičajeno je da se kao kandidati za granicu x promatraju aritmetičke sredine susjednih vrijednosti. Stoga je potrebno sortirati primjere za učenje prema vrijednosti promatranog numeričkog atributa. Neka numerički atribut A_i u skupu za učenje S' poprima m različitih vrijednosti, te neka je $(v_j)_{j=1}^{m}$ niz uzlazno sortiranih vrijednosti atributa A_i, tj. $v_j \leq v_{j+1}$ za svaki $j \in \{1,...,m-1\}$. Tada (33) daje $m-1$ kandidata za vrijednost granice testa.

$$x_j = \frac{v_j + v_{j+1}}{2}, \quad j \in \{1,...,m-1\} \tag{33}$$

Naravno, kao granica testa se odabire ona vrijednost x_j za koju je informacijski dobitak maksimalan. Može se pokazati da vrijednosti x_j koje maksimiziraju informacijski dobitak uvijek leže između primjera koji pripadaju različitim klasama (u nizu primjera sortiranom prema vrijednostima atributa A_i). Stoga je dovoljno promatrati samo one vrijednosti x_j za koje skup $\{s \in S', A_i(s)=v_j \vee A_i(s)=v_{j+1}\}$ sadrži primjere barem dvije klase.

Dakle, jedna od izmjena algoritma konstrukcije stabala odlučivanja odnosi se na sortiranje primjera prema vrijednosti svakog od numeričkih atributa. Na prvi pogled se čini da je sortiranje potrebno provoditi u svakom koraku rekurzije, tj. za svaki unutrašnji čvor stabla u kojem se razmatraju numerički atributi. Međutim, redoslijed primjera u roditelju inducira redoslijed u djeci, pa je sortiranje potrebno izvršiti samo jednom, u korijenu stabla.

Druga izmjena algoritma vezana uz numeričke atribute odnosi se na prosljeđivanje skupa atributa u slijedeći korak rekurzije. Kada se stablo grana na nominalnom atributu A_i, svaka grana odgovara jednoj vrijednosti tog atributa. Stoga svako daljnje ispitivanje vrijednosti atributa A_i u bilo kojoj od nastalih grana nema smisla, pa se slijedećem koraku rekurzije prosljeđuje skup atributa $A' - \{A_i\}$. Kod numeričkih atributa to nije slučaj, jer binarni test ne iskorištava u potpunosti informacije koje atribut nosi. Ponovno grananje korištenjem istog numeričkog atributa (ali uz drugu granicu testa) može rezultirati povećanjem informacijskog dobitka, pa višestruko testiranje istog atributa na putu od korijena do lista ima smisla. Da bi se to omogućilo, kod grananja na numeričkom atributu u slijedeći korak rekurzije prosljeđuje se cijeli skup atributa A'.

5.1.6. Neodređene vrijednosti atributa

Neodređene vrijednosti atributa u primjerima uzrokuju poteškoće kako pri konstrukciji stabla odlučivanja, tako i pri klasifikaciji primjera izgrađenim stablom odlučivanja.

Ukoliko činjenica da vrijednost atributa u primjerima nije zabilježena nosi neko značenje, prikladno je uvođenje nove vrijednosti tipa "nepoznato", kojom se mijenjaju sve neodređene vrijednosti atributa u primjerima za učenje i u primjerima za klasifikaciju. Novouvedena vrijednost se i pri konstrukciji i pri klasifikaciji tretira na standardan način, pa nikakve izmjene algoritama nisu potrebne.

Ukoliko neodređene vrijednosti atributa nemaju posebno značenje, ovakvo rješenje nije prikladno. Primjerice, u tom slučaju postojanje neodređenih vrijednosti za neki atribut povećava njegov očekivani informacijski dobitak, što nije poželjno svojstvo. Stoga su potrebni drugi načini tretiranja neodređenih vrijednosti.
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 "cijepanju" primjera u dijelove. Sličan mehanizam koristi se i pri klasifikaciji i pri konstrukciji stabla odlučivanja.

- Klasifikacija primjera
  Pri klasifikaciji primjera problem nastaje u čvoru koji ispituje atribut čija je vrijednost u primjeru neodređena. U tom slučaju istražuju se sve grane koje vode iz čvora, i to uz korištenje relativne frekvencije primjera za učenje koja odgovara pojedinoj grani. Spomenute relativne frekvencije se koriste u smislu vjerojatnosti da primjer pripada odabranoj grani. Stoga se u slučaju više neodređenih vrijednosti atributa na putu od korijena do lista pripadajuće vjerojatnosti odabranih grana množe. U konačnici, spuštanje po stablu će završiti u više listova s potencijalno različitim klasama. Pripadajuće vjerojatnosti za svaku od klasa se zbrajaju, te se tako dolazi do konačne klasifikacije primjera.
- Konstrukcija stabla odlučivanja
  U uvjetima postojanja neodređenih vrijednosti atributa u skupu za učenje, uvode se težinske vrijednosti primjera. Svim primjerima iz skupa za učenje se inicijalno dodjeljuje težinska vrijednost 1.
  Kada postupak konstrukcije stabla odlučivanja odabere atribut grananja A_i čija je vrijednost u nekim primjerima za učenje neodređena, postoji problem particioniranja skupa za učenje S' prema vrijednosti atributa A_i. Primjeri s neodređenom vrijednost atributa A_i se tada po sličnom principu "cijepaju" na dijelove koji odgovaraju relativnim frekvencijama ostalih primjera u granama. Konkretno, ako je $S'_j = \{s \in S', A_i(s) = a_j\}$, a $s \in S'$ primjer za kojeg je vrijednost atributa A_i neodređena, tada on sudjeluje u skupu S'_j (za svaki j) uz korekciju težinske vrijednosti faktorom $|S'_j| / |S'|$. Treba primijetiti da u ovom slučaju $|S'_j|$ i $|S'|$ ne označavaju kardinalnosti skupova, već sumu težinskih vrijednosti primjera. Na ovaj način primjer s sudjeluje u svim skupovima S'_j (ali uz različite težinske vrijednosti), pa na taj način doprinosi odlukama u nižim razinama stabla.
  Generalizacija definicije informacijskog dobitka zbog uvođenja težinskih vrijednosti primjera je očita, a svodi se na uvažavanje težinskih vrijednosti pri računanju relativne frekvencije klasa u skupu primjera za učenje.

Učinkovitost ovakvog postupka klasifikacije i konstrukcije stabla odlučivanja ogleda se u sporom padu točnosti klasifikacije pri povećanju broja neodređenih vrijednosti u primjerima za učenje i klasifikaciju.

5.1.7. 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 [7]). Razmatrane su i korekcije postupka koje smanjuju potrebne računalne i vremenske zahtjeve, što je posebno važno kod primjene na glomaznim skupovima podataka. Jedna od takvih nalaže da se pri konstrukciji stabla koristi samo manji, slučajno odabrani dio skupa primjera za učenje, tzv. *prozor*. Tako konstruiranim stablom se klasificira preostali dio skupa za učenje, te se izdvajaju krivo klasificirani primjeri. Oni se pridodaju prozoru te se postupak iterativno ponavlja na ovako izmijenjenom skupu primjera, sve dok se ne postigne zadovoljavajuća točnost klasifikacije na preostalim primjerima. Eksperimentalni rezultati pokazuju da se postupak u pravilu zaustavlja nakon samo nekoliko iteracija, te da je ovaj pristup zamjetno brži od konstrukcije stabla na cijelom skupu 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 svedeno na hiperravnine paralelne osima univerzuma, što bitno ograničava mogućnosti omeđivanja klasa.

Spomenute prednosti i nedostatke dobro je imati u vidu pri odabiru tehnike modeliranja za konkretni klasifikacijski problem.

5.2. TEHNIKA INDUKCIJE PRAVILA

5.2.1. Reprezentacija modela

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

$$cond(s) \rightarrow s \in C_i \quad , \tag{34}$$

gdje je s proizvoljni element univerzuma U.
U izrazu (34) *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 $cond \rightarrow C_i$.
Dakako, postoje i bitno složeniji oblici antecedente klasifikacijskih pravila.

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. Podskup R_{Ci} skupa pravila R koji sadrži sva pravila sa konzekventom $s \in C_i$ predstavlja opis klase C_i. Semantika podskupa klasifikacijskih pravila sa istom konzekventom 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.
Prema tome, dozvoljena je situacija da primjer s bude prekriven s više pravila, ili da ne bude prekriven ni s jednim pravilom iz R. Ova činjenica može stvarati probleme prilikom klasifikacije primjera pomoću skupa pravila R. Neka je sa $R(s) \subseteq R$ označen skup svih pravila koja prekrivaju primjer s. Razlikuju se tri slučaja:

- $|R(s)| = 0$, tj. $R(s) = \emptyset$.
  Primjer s nije prekriven ni jednim pravilom. Ovisno o pristupu, načini klasifikacije primjera s uključuju pridjeljivanje najfrekventnijoj klasi iz skupa za učenje, ili klasi pravila s najvećim prekrivačem na skupu za učenje, ili se signalizira nemogućnost klasifikacije takvog primjera.
- $|R(s)| = 1$ ili $R(s) \subseteq R_{Ci}$ za neki i.
  Ovo nije konfliktna situacija, pa se primjeru s pridjeljuje klasa C_i.
- $|R(s)| > 1$ i ne postoji i takav da $R(s) \subseteq R_{Ci}$.
  Primjer s je prekriven s više pravila različite konzekvente. Najčešći način klasifikacije takvog primjera je primjena onog pravila $r \in R(s)$ koje prekriva najviše primjera iz skupa za učenje ili signalizacija nemogućnosti klasifikacije.

Postoji više različitih tehnika indukcije klasifikacijskih pravila. Tehnike se mogu razlikovati po dozvoljenom obliku pravila, načinu pretraživanja prostora rješenja, ili korištenoj funkciji vrednovanja. U svakom slučaju, prostor pretraživanja se sastoji od svih pravila dozvoljenog oblika koje je moguće konstruirati korištenjem atributa i vrijednosti iz skupa podataka za učenje. Operacije transformacije pravila se razlikuju od tehnike do tehnike, a mogu se podijeliti u dvije osnovne skupine:

- operacije *specijalizacije* pravila, koje smanjuju prekrivač pravila
- operacije *generalizacije* pravila, kojima se prekrivač pravila povećava.

Operacije specijalizacije koriste tehnike vođene modelom, koje polaze od najopćenitijih pravila (pristup odozgo na dolje), dok operacijama generalizacije se koriste tehnike vođene podacima, koje polaze od pojedinačnih primjera (pristup

odozdo na gore). Funkcije vrednovanja koje usmjeravaju postupak pretraživanja u pravilu ovise o točnosti i kardinalnosti prekrivača klasifikacijskog pravila.

5.2.2. Postupak pretraživanja

Tehnike indukcije pravila se po pristupu pretraživanju prostora rješenja bitno razlikuju od tehnika konstrukcije stabala odlučivanja. Pri konstrukciji stabala odlučivanja se u svakom koraku bira atribut čije vrijednosti najbolje razdvajaju primjere različitih klasa. Na taj način se particionira skup primjera za učenje, te se postupak rekurzivno primjenjuje na svakoj od particija.
Alternativni pristup kojeg koriste tehnike indukcije pravila je odvojeno promatranje svake od klasa. Postupak se fokusira na jednu od klasa, te pokušava konstruirati pravila koja prekrivaju što više pozitivnih primjera te klase, istovremeno isključujući negativne primjere klase. Pri tome se svi negativni primjeri klase tretiraju jednako, tj. među njima se ignorira razlika 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* [8], 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 sastoji se od dodavanja konjunkta oblika $A_j=a_k$ u antecedentu pravila, tj. od dodavanja elementarnih uvjeta na vrijednosti atributa. Strategija pretraživanja prostora rješenja je nepovratna. Koristi se metoda uspona na vrh, vođena heuristikom odabira specijalizacije.
Koraci postupka indukcije pravila za klasu C_i tehnikom Prism prikazani su na slici 12. Kompletan postupak dobije se ponavljanjem opisanih koraka za svaku od klasa iz skupa primjera za učenje.

```
1.  S' := skup primjera za učenje S; A' := skup svih atributa iz S
2.  dok S' sadrži primjere klase Ci ponavljaj
    2.1. napravi pravilo r sa praznom antecedentom i konzekventom Ci
    2.2. dok r ne dosegne zadanu točnost i A' nije prazan ponavljaj
        2.2.1.  odaberi uvjet oblika Aj=ak, gdje je Aj∈A' i ak∈Dom(Aj)
        2.2.2.  specijaliziraj r dodavanjem uvjeta Aj=ak u antecedentu
        2.2.3.  iz A' izbaci Aj, tj. A' := A'-{Aj}
    2.3. dodaj r u skup pravila R
    2.4. iz S' izbaci primjere koje prekriva r, tj. S' := S'-σr
3.  vrati skup pravila R
```

Slika 12: Prism tehnika indukcije pravila za klasu C_i

Pojedinačna klasifikacijska pravila generiraju se u unutrašnjoj petlji prikazanog postupka. Svako pravilo se gradi od inicijalno praznog pravila koje prekriva cijeli skup dostupnih primjera. Prekrivač pravila se u svakoj iteraciji petlje smanjuje operacijom specijalizacije, odnosno dodavanjem novog uvjeta na vrijednost nekog atributa. Cilj operacije specijalizacije je izbacivanje negativnih primjera klase iz prekrivača pravila. Specijalizacija pravila se nastavlja dok se ne postigne tražena točnost pravila. U osnovnom obliku algoritma se traži potpuna točnost pravila, iako postoje i bitno složeniji kriteriji zaustavljanja.

Svaka iteracija vanjske petlje postupka generira po jedno klasifikacijsko pravilo. Generirano pravilo se dodaje u skup pravila, a iz promatranog skupa primjera za učenje se izbacuju svi primjeri koje pravilo prekriva. Na taj način se slijedeća iteracija postupka usmjerava na preostale pozitivne primjere koji nisu prekriveni nijednim pravilom. Postupak generiranja novih pravila se zaustavlja kad su pravilima prekriveni svi pozitivni primjeri klase.
Jasno je da način odabira specijalizacije pravila ima presudan utjecaj na konačni oblik i kvalitetu pravila. Kvaliteta pravila se ogleda u njegovoj točnosti i općenitosti, pa se preferiraju točnija pravila s većim prekrivačem. Postoji više različitih kriterija za odabir specijalizacije kojima se nastoji postići ovaj cilj.
Neka je sa $\sigma_r^+ \subseteq \sigma_r$ označen skup pozitivnih primjera klase koje prekriva pravilo r. Najjednostavniji kriterij odabira specijalizacije nastoji maksimizirati točnost pravila, tj. omjer $|\sigma_r^+|/|\sigma_r|$. U slučaju da se maksimalna točnost postiže na više specijalizacija, odabire se ona čiji je prekrivač najveći, tj. gdje je $|\sigma_r|$ maksimalan.
Drugi kriterij odabira specijalizacije zasniva se na teoriji informacija. Heuristika odabira koristi informacijski dobitak, slično odabiru atributa grananja kod stabala odlučivanja. Neka je sa q označena proizvoljna specijalizacija pravila r. Informacijski dobitak postignut specijalizacijom q definiran je izrazom (35).

$$gain(q,r) = \sigma_q^+ \left(\log_2 \frac{|\sigma_q^+|}{|\sigma_q|} - \log_2 \frac{|\sigma_r^+|}{|\sigma_r|} \right) \tag{35}$$

Heuristika informacijskog dobitka nalaže odabir one specijalizacije pravila koja maksimizira informacijski dobitak.
Osnovna razlika ovih kriterija za odabir specijalizacije je u naglasku koji stavljaju na kardinalnost prekrivača pravila. Heuristika zasnovana na točnosti preferira točnija pravila, bez obzira na broj primjera koje pravilo prekriva. Tek u slučaju iste točnosti više pravila, promatra se kardinalnost njihovih prekrivača. S druge strane, 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.

5.2.3. Šum u podacima (podrezivanje)

Osnovni oblik Prism algoritma indukcije pravila generira pravila koja su potpuno točna na skupu podataka za učenje. Međutim, 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.
Cilj podrezivanja pravila je poboljšanje kvalitete previla u smislu klasifikacijskih performansi na novim primjerima. Stoga je ključan element postupka podrezivanja

pravila kriterij na osnovu kojeg se prosuđuje opravdanosti uklanjanja pojedinih konjunkata. Postoji više različitih pristupa evaluaciji kvalitete klasifikacijskih pravila, pa tako i više kriterija usporedbe pravila i njegove generalizacije dobivene podrezivanjem. Općenito, da bi bilo koja procjena kvalitete klasifikacijskog pravila bila validna u statističkom smislu, nužno je da se sama procjena odvija na odvojenom, validacijskom skupu primjera, koji nije bio korišten pri indukciji pravila.

Jedan od pristupa evaluaciji pravila zasniva se procjeni vjerojatnosti da slučajno odabrano pravilo postigne istu ili veću točnost klasifikacije od promatranog pravila. Neka je sa r označeno promatrano klasifikacijsko pravilo za klasu C_i, a sa $T^+ \subseteq T$ skup svih pozitivnih primjera klase C_i u skupu primjera T nad kojim se vrši evaluacija. Pod slučajnim pravilom se podrazumijeva pravilo q čiji prekrivač nastaje slučajnim odabirom primjera iz T, a po kardinalnosti odgovara prekrivaču pravila r, tj. $|\sigma_q| = |\sigma_r|$.
Vjerojatnost da slučajno pravilo prekriva točno j pozitivnih primjera klase C_i dana je izrazom (36).

$$P(|\sigma_q^+| = j) \;=\; \frac{\binom{|T^+|}{j}\binom{|T|-|T^+|}{|\sigma_r|-j}}{\binom{|T|}{|\sigma_r|}} \tag{36}$$

Brojnik izraza (36) odgovara odabiru j primjera od $|T^+|$ pozitivnih primjera klase u T, uz odabir preostalih $|\sigma_r| - j$ primjera među $|T| - |T^+|$ negativnih primjera klase u T.

Korištenjem (36), vjerojatnost da slučajno pravilo q postigne istu ili veću točnost klasifikacije od r dobije se sumiranjem po svim $j \geq |\sigma_r^+|$. Stoga se prema ovom pristupu mjera kvalitete $m(r)$ pravila r može definirati s (37):

$$m(r) = \sum_{j=|\sigma_r^+|}^{\min(|\sigma_r|,|T^+|)} P\left(|\sigma_q^+| = j\right) \quad . \tag{37}$$

Budući da ova mjera predstavlja vjerojatnost da je slučajno pravilo točnije od r, pravilo je bolje što je vrijednost mjere $m(r)$ manja. Stoga se radi konzistencije u izražavanju mjera kvalitete obično zapisuje u obliku $1–m(r)$.
Nedostatak ovako definirane mjere je razmjerno zahtjevan postupak izračuna. Ukoliko je ukupan broj primjera $|T|$ velik, dobra aproksimacija vjerojatnosti $P(|\sigma_q^+| = j)$ može se dobiti upotrebom odabira s ponavljanjem (38).

$$P(|\sigma_q^+| = j) \;\approx\; \binom{|\sigma_r|}{j}\left(\frac{|T^+|}{|T|}\right)^j\left(1-\frac{|T^+|}{|T|}\right)^{|\sigma_r|-j} \text{, za veliki } |T|. \tag{38}$$

Izraz (38) je bitno jednostavniji za računanje od (36). Sumiranjem po j analogno izrazu (37) dobije aproksimacija kvalitete pravila koja se zbog jednostavnijeg izračuna u praksi često koristi. Proširenje Prism tehnike indukcije pravila koje

uključuje podrezivanje pravila upotrebom ovako definiranog kriterija poznato je pod nazivom *Induct* [21], a razvijeno je sredinom devedesetih godina dvadesetog stoljeća.

Neka je sa $G(r)$ označen skup svih pravila koja se mogu dobiti uklanjanjem konjunkata iz antecedente pravila r. Uz definiranu mjeru kvalitete pravila, podrezivanje pravila r se svodi na pretraživanje skupa $G(r)$ u potrazi za pravilom koje maksimizira mjeru kvalitete. Budući da $|G(r)|$ eksponencijalno raste s povećanjem broja konjunkata u r, iscrpno pretraživanje je obično neprimjereno. Osnovni oblik Induct algoritma koristi vrlo jednostavnu heuristiku za pretraživanje skupa $G(r)$. Pretraživanje se vrši uzastopnim uklanjanjem konjunkata iz antecedente pravila obrnutim redoslijedom od onoga kojim su dodavani prilikom specijalizacije pravila. Postupak prestaje kad uklanjanje konjunkta ne rezultira boljim pravilom u smislu definirane mjere kvalitete pravila.
Iako je redoslijed uklanjanja konjunkata koji nalaže heuristika ponešto neobičan, dobri rezultati koje pokazuje u praksi opravdavaju njenu primjenu.

5.2.4. Numerički atributi i neodređene vrijednosti atributa

U tehnikama indukcije klasifikacijskih pravila numerički atributi se tretiraju na sličan način kao pri konstrukciji stabala odlučivanja. U klasifikacijskim pravilima za numerički atribut A_i također su dozvoljeni samo binarni testovi oblika $A_i < x$, gdje je x odabrana konstanta. Primjeri iz skupa za učenje se sortiraju prema vrijednostima atributa A_i radi odabira pogodnih konstanti za granicu testa. Granice se biraju na razmeđama klasa u sortiranom nizu primjera. Za svaku granicu x_j se formira test oblika $A_i < x_j$, te on služi kao definicija novog binarnog atributa. U postupku indukcije pravila se takvi binarni atributi tretiraju na uobičajen način.
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.

5.2.5. 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.

Primjerice, liste odlučivanja [52] predstavljaju uređene nizove pravila koji se interpretiraju u zadanom redoslijedu, te se pojedinačna pravila ne mogu promatrati van konteksta niza. Na taj način su izbjegnuti problemi višeznačne klasifikacije, jer se primjer klasificira na način da se pravila isprobavaju po utvrđenom redoslijedu. Primjeru se pridjeljuje klasa prvog pravila čiju antecedentu primjer zadovolji.
Drugi oblik klasifikacijskih pravila čija interpretacija ovisi o kontekstu su pravila s iznimkama [21]. Takva pravila predstavljaju nadogradnju standardnih pravila jer dozvoljavaju navođenje uvjeta pod kojima pravilo ne vrijedi. Dozvoljena su ugnježđivanja pravila i iznimki proizvoljne dubine, pa se svako pravilo/iznimka može promatrati samo u kontekstu pravila/iznimki koje su dovele do njega.

Budući da se svako stablo odlučivanja može prikazati kao skup klasifikacijskih pravila, osnovna 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 [11].

5.3. KLASIFIKACIJA PAMĆENJEM PRIMJERA

5.3.1. Reprezentacija modela

Klasifikacija pamćenjem primjera spada u najjednostavnije tehnike inteligentne analize podataka. Ne vrši eksplicitnu generalizaciju ciljnog pojma na osnovu svojstava izvedivih iz skupa za učenje, već se svodi na puko memoriranje skupa za učenje, odnosno pojedinačnih primjera koje sadržava. 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.
Dakle, element tehnike klasifikacije pamćenjem primjera koji utječe na oblik modela je metrika. 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 (39).

$$\| x \| = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{39}$$

Metrika na standardan način definira udaljenost, pa izraz (40) definira Euklidsku udaljenost primjera x i y.

$$d(x, y) = \| x - y \| = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{40}$$

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 izrazima (39) i (40) se implicitno pretpostavlja da su svi atributi numeričkog tipa, tj. da su vrijednosti koordinata brojevi. Da bi se definicija Euklidske udaljenosti mogla primijeniti na nominalne atribute, potrebno je definirati operaciju razlike nad nominalnim vrijednostima.
Neka su sa $a_i, a_j \in Dom(A_i)$ označene dvije proizvoljne vrijednosti nominalnog atributa A_i. Razlika vrijednosti a_i i a_j definirana je *0–1 funkcijom razlikovanja* (41).

$$a_i - a_j = \begin{cases} 0, & \text{za } a_i = a_j \\ 1, & \text{inače} \end{cases} \tag{41}$$

Problem Euklidske i sličnih metrika vezan uz numeričke atribute su različite skale mjerenja za različite numeričke atribute. Primjerice, atribut čiji se raspon vrijednosti kreće unutar desetih dijelova mjerne jedinice imat će zanemariv utjecaj na konačni rezultat u odnosu na atribut sa rasponom vrijednosti od nekoliko desetaka mjernih jedinica. Zbog toga je za tehniku klasifikacije pamćenjem primjera uobičajen postupak normalizacije svih numeričkih atributa na interval [0,1]. Normalizacije se provodi na standardan način, korištenjem funkcije (42):

$$f(x) = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad , \tag{42}$$

gdje $x_{\min}$ i $x_{\max}$ označavaju najmanju, odnosno najveću vrijednost promatranog atributa.
U tehnici klasifikacije pamćenjem primjera neodređene vrijednosti atributa se tretiraju slično nominalnim atributima, tj. proširuje se definicija operacije razlike. Razlika se definira na način da je neodređena vrijednost maksimalno udaljena od bilo koje promatrane vrijednosti atributa. Neka su sa x i a označene proizvoljne (ali poznate) vrijednosti numeričkog odnosno nominalnog atributa, te neka simbol *null* označava neodređenu vrijednost bilo kojeg atributa. Izrazima (43) operacija razlike se poopćuje na neodređene vrijednosti atributa, uz pretpostavku normalizirane vrijednosti numeričkih atributa.

$$\begin{aligned} null-null &= 1 \\ a-null = null-a &= 1 \\ x-null = null-x &= \max(x,1-x) \end{aligned} \tag{43}$$

Uz ovako definiranu funkciju razlike, može se koristiti standardna definicija Euklidske udaljenosti (40) i za primjere s neodređenim vrijednostima atributa.

5.3.2. 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. Skup primjera za učenje promatran u svjetlu funkcije udaljenosti u pravilu sadrži veliki broj redundantnih primjera. Za klasifikaciju su najvažniji primjeri koji se nalaze u blizini granica među klasama, a primjeri iz unutrašnjosti omeđenog područja klasa mogu se izostaviti bez posljedica na točnost klasifikacije.
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. U skupu se nastoje zadržati reprezentativni primjeri, koji posredstvom funkcije udaljenosti dobro generaliziraju područje u kojem se nalaze. Iz skupa se izbacuju primjeri koji bitno ne pridonose oblikovanju područja odgovarajuće klase.
Za razliku od stabala odlučivanja i klasifikacijskih pravila, formiranje reprezentativnog skupa primjera pretraživanju pristupa po načelu *odozdo na gore*, tj. od pojedinačnih primjera ka reprezentativnim primjerima sa proširenom sferom utjecaja. Iako postoji više različitih kriterija prihvata odnosno eliminacije primjera, uglavnom se radi o nepovratnim strategijama pretraživanja pohlepnog karaktera. Najjednostavniji kriterij prosuđuje primjer prema rezultatu klasifikacije korištenjem skupa do tada izdvojenih reprezentativnih primjera. U slučaju netočne klasifikacije

primjer se pridodaje u skup reprezentativnih primjera, zbog činjenice da evidentno mijenja granice klasa. Točno klasificirani primjer se proglašava suvišnim, pod pretpostavkom da je njegova informativnost već sadržana u skupu pomoću kojeg je klasificiran.
Opisani kriterij odabira primjera ima više nedostataka. U ranoj fazi procesa postoji nezanemariva vjerojatnost odbacivanja primjera koji se mogu pokazati bitnima za točnost klasifikacije rezultirajućeg modela. Osim toga, odabrani podskup reprezentativnih primjera ne ovisi samo o polaznom skupu, već i o redoslijedu evaluacije primjera. Možda najbitniji nedostatak se odnosi na loše ponašanje u uvjetima šuma u podacima. Budući da se u skup uvrštavaju netočno klasificirani primjeri, ovaj kriterij ima tendenciju akumuliranja primjera sa šumom u rezultirajući skup primjera, što uvelike smanjuje njegovu reprezentativnost. Stoga se u praksi češće upotrebljavaju drugi, ponešto složeniji kriteriji odabira reprezentativnih primjera.

5.3.3. Funkcija udaljenosti primjera

Na oblik klasifikacijskog modela se osim odabirom primjera za pamćenje može utjecati i modifikacijom funkcije udaljenosti. Jedno od svojstava Euklidske udaljenosti je jednak utjecaj svih atributa u primjeru na konačan rezultat. Međutim, u praksi su rijetki problemi kod kojih su svi atributi imaju jednaku klasifikacijsku vrijednost. Ova činjenica stvara prostor za moguće poboljšanje tehnike klasifikacije pamćenjem primjera.
Jedno od mogućih rješenja je modifikacija funkcije udaljenosti na način da valorizira klasifikacijski potencijal različitih atributa. Standardno proširenje Euklidske udaljenosti koje podrazumijeva uvođenje težinskih vrijednosti atributa je korak u tom smjeru. Neka je sa w_i označena težinska vrijednost pridružena atributu A_i. Izraz (44) definira modificiranu Euklidsku udaljenost primjera x i y.

$$d_w(x,y) = \sqrt{\sum_{i=1}^{n} w_i^2 (x_i - y_i)^2} \qquad (44)$$

Veća težinska vrijednost pridružena atributu daje mu veći utjecaj na izračun udaljenosti primjera. Prema tome, variranje težinskih vrijednosti atributa je jedan od načina korekcije klasifikacijskog modela u tehnici klasifikacije pamćenjem primjera.

Potrebno je još razjasniti način određivanja težinskih vrijednosti za pojedine atribute. Iako postoje neovisni postupci ocjene klasifikacijske vrijednosti atributa, u kontekstu ove tehnike modeliranja težine atributa se najčešće određuju interno, prilikom formiranja relevantnog podskupa primjera za učenje.
Inicijalno je svim atributima pridružena težinska vrijednost 1, koja se iterativno modificira pri razmatranju svakog od primjera iz skupa za učenje. Kao i pri klasifikaciji primjera, u podskupu relevantnih primjera se pronalazi primjer *y* najbliži promatranom primjeru *x*. Ukoliko primjeri *x* i *y* pripadaju istoj klasi, smanjuje se težinska vrijednost atributa čije se vrijednosti u primjerima *x* i *y* najviše razlikuju, jer se razlika u vrijednostima tih atributa pripisuje slaboj korelaciji s klasom. Analogno tome, u slučaju da primjeri *x* i *y* pripadaju različitim klasama, težinska vrijednost atributa s najvećom razlikom vrijednosti se povećava. Iznos smanjenja odnosno

povećanja težinske vrijednosti proporcionalan je razlici vrijednosti atributa u primjerima *x* i *y*.

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]. Robusnost se poboljšava promatranjem svih nizova operacija koji dovode do tražene transformacije primjera, zajedno sa vjerojatnostima svakog od njih.
Prednost ovakve definicije udaljenosti je mogućnost uniformnog tretiranja numeričkih i nominalnih atributa, definiranjem odgovarajućih operacija transformacije za svaki od njih. Čak je moguće na prirodan način tretirati i specifične tipove atributa, kao npr. stupnjeve kuta ili dane u tjednu, koji se mjere cirkularnom skalom. Osim toga, vjerojatnosna interpretacija udaljenosti osim kategoričke klasifikacije može kao rezultat ponuditi i distribuciju vjerojatnosti pripadanja svakoj od klasa, što je u nekim primjenama značajna prednost.

5.3.4. 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*. Umjesto izdvajanja samo jednog najbližeg primjera iz skupa za učenje, postupak klasifikacije pronalazi *k* najbližih primjera, za neki unaprijed određeni mali broj *k*. Svih *k* pronađenih primjera sudjeluje u klasifikaciji novog primjera po principu većinskog glasovanja, tj. primjeru se pridružuje najfrekventnija klasa unutar izdvojenih *k* primjera.
Osnovni algoritam klasifikacije je specijalni slučaj ovog poopćenja za k=1. Općenito, prikladna vrijednost konstante *k* ovisi o količini šuma u podacima za učenje: što je više šuma, povoljnije su veće vrijednosti konstante *k*. Zbog bitno poboljšane točnosti klasifikacije u uvjetima šuma, varijanta k najbližih susjeda je gotovo u potpunosti istisnula osnovni oblik algoritma. Može se pokazati da za $|S|\to\infty|$ i $k\to\infty$ na način da $k/|S|\to 0$, vjerojatnost pogrešne klasifikacije teži teoretskom minimumu za promatrani skup primjera.

Drugi pristup tretiranju šuma u podacima sastoji se od detekcije primjera sa šumom i njihovog izdvajanja iz skupa za učenje. Uobičajen način detekcije nepoželjnih primjera je praćenje klasifikacijskih performansi svakog od primjera iz skupa za učenje. Unaprijed se odrede dva praga točnosti klasifikacije, u svrhu odbacivanja i prihvaćanja primjera u skup primjera za pamćenje. Primjeri čija točnost klasifikacije padne ispod praga odbacivanja eliminiraju se iz skupa za učenje. Za klasifikaciju se koriste oni primjeri za koje točnost klasifikacije prijeđe prag prihvaćanja. Primjeri čija se točnost nalazi između dva praga se ne koriste pri klasifikaciji, ali se prate i korigiraju njihove klasifikacijske performanse svaki put kada su izdvojeni kao najbliži primjeri prilikom klasifikacije drugih primjera.

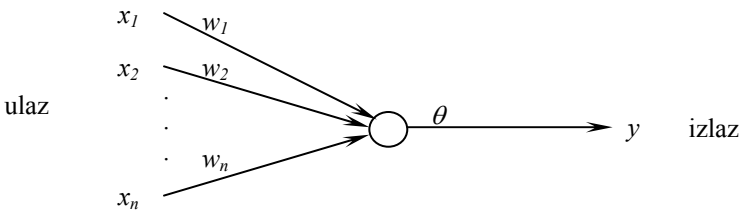
Pri određivanju pragova odbacivanja i prihvaćanja primjera koristi se pretpostavka Bernoullijevog procesa, pri čemu točna klasifikacija odgovara uspješnom ishodu eksperimenta. Pragovi se određuju usporedbom očekivane frekvencije uspjeha pojedinog primjera i očekivane frekvencije uspjeha slijepe klasifikacije za odgovarajuću klasu (tj. klasifikacije koja uvijek predviđa spomenutu klasu). Očekivana frekvencija uspjeha izražava se terminima intervala pouzdanosti. Kriterij odbacivanja zahtijeva da za promatrani primjer gornja granica intervala pouzdanosti ocjene frekvencije uspjeha bude niža od donje granice intervala pouzdanosti slijepe klasifikacije. Analogno, kriterij prihvaćanja je određen uvjetom da donja granica intervala pouzdanosti promatranog primjera bude viša od gornje granice intervala pouzdanosti slijepe klasifikacije.
Nije nužno koristiti istu razinu pouzdanosti za određivanje praga odbacivanja i prihvaćanja primjera. Primjerice, varijanta algoritma klasifikacije pamćenjem primjera poznata pod nazivom *IB3* (od engl. *Instance-Based learner ver.3*) za prag prihvaćanja koristi pouzdanost od 95%, a za praga odbacivanja pouzdanost od 87.5%. Dakle, kriterij odbacivanja je nešto blaži, jer se ne gubi puno odbacivanjem primjera umjereno slabih klasifikacijskih performansi – veliki su izgledi da će u kasnijoj fazi procesa biti nadomješteni primjerima s boljim rezultatima klasifikacije.

5.3.5. 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 [19]. Kao postupak klasifikacije pojavljuje se desetak godina kasnije [31], a najintenzivnije se koristila na području raspoznavanja uzoraka.
Međutim, osnovni oblik klasifikacije pamćenjem primjera ima više nedostataka. Sam postupak klasifikacije novih primjera može biti prilično spor u slučaju glomaznih skupova za učenje, budući da klasifikacija svakog primjera zahtijeva pretraživanje cijelog skupa za učenje. Osim toga, bez korištenja više najbližih primjera pokazuje prilično osjetljivost na šum u podacima za učenje. Konačno, nije prilagođen problemima kod kojih atributi imaju različit klasifikacijski potencijal, a klasifikacijske performanse posebno narušavaju irelevantni atributi.
Klasifikacija pamćenjem primjera popularnost vraća početkom 1990-ih kroz radove D.Aha [2], u kojima se nadogradnjama osnovnog postupka umanjuju spomenuti 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. Osim toga, odbacivanje nepotrebnih primjera dramatično reducira broj primjera koji se pamte, značajno smanjujući potrebne resurse i vrijeme klasifikacije.
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.

5.4. NEURALNE MREŽE

Jedna od primjena umjetnih neuralnih mreža su i klasifikacijski problemi, odnosno postupci strojnog učenja. Neuralne mreže su gusto isprepletene strukture međusobno povezanih računskih elemenata, nalik povezanim usmjerenim grafovima [44]. Osnovni element neuralne mreže naziva se *neuron*, a nalikuje čvoru grafa s pripadajućim ulaznim i izlaznima granama. Slika 13 predstavlja shematski prikaz neurona.



Slika 13: Neuron, osnovni element neuralne mreže

Ulazni dio neurona čini realni vektor ulaznih vrijednosti (x_1,x_2, ... ,x_n), a izlazni je jedna realna vrijednost y. Komponente ulaznog vektora i izlaz neurona u pravilu poprimaju vrijednosti iz intervala [0,1]. 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 (45):

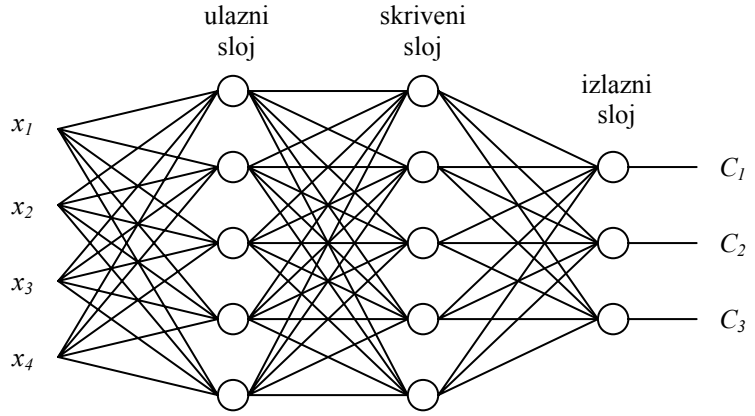
$$y = f\left(\sum_{i=1}^{n} w_i x_i - \theta\right) \quad , \tag{45}$$

gdje f može biti zadana sa

$$f(x) = \frac{1}{1+e^{-x}} \quad . \tag{46}$$

Neuralne mreže nastaju povezivanjem neurona na način da izlazna vrijednost jednog neurona postaje ulazna vrijednost jednog ili više drugih neurona. Neuralna mreža može imati više ulaza i izlaza. Ulaz u mrežu čine neuroni sa slobodnim ulazima, a izlaz mreže neuroni sa slobodnim izlazima.

U praksi se koristi mnogo različitih topologija neuralnih mreža. Kad su klasifikacijski problemi u pitanju, najpoznatija je topologija *višeslojnog perceptrona*. Način povezivanja neurona u višeslojni perceptron prikazan je na slici 14.
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.



Slika 14: Višeslojni perceptron

5.4.1. 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 unaprijed zadanu topologiju mreže, konačni oblik mreže (odnosno klasifikacijske funkcije koju predstavlja) određuje se optimiranjem težinskih faktora w_i i praga θ svakog od neurona. Odabirom odgovarajućih težina i pragova, ista topologija mreže može predstavljati raznolike klasifikacijske funkcije. Dakle, uz zadanu topologiju neuralne mreže, klasifikacijski model je određen težinskim vektorima i vrijednostima praga svakog od neurona. Adekvatni težinski faktori se mogu odrediti treniranjem 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.

5.4.2. Postupak pretraživanja

Algoritmi treniranja neuralnih mreža koji korištenjem skupa za učenje modificiraju težinske vektore i pragove neurona. U svjetlu tehnika modeliranja, radi se o postupku pretraživanja prostora rješenja optimiranjem parametara klasifikacijskog modela.
Prilikom treniranja neuralnih 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 neuralnih 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 neuralnoj mreži stabiliziraju.

5.4.3. Svojstva klasifikacije neuralnim mrežama

Klasifikacija neuralnim mrežama se pokazala vrlo dobrom upravo na težim klasifikacijskim problemima, kod kojih je teško ili nemoguće koristiti klasične tehnike simboličkog učenja. Neuralne 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, neuralne mreže su dobro prilagođene klasifikaciji u uvjetima šuma u podacima.
Jedan od nedostataka neuralnih mreža je relativno spor i zahtijevan proces indukcije modela u usporedbi s klasičnijim tehnikama, čak do nekoliko redova veličine [50]. Značajan nedostatak je i činjenica da klasifikacijski model reprezentiran neuralnom 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.

5.5. ASOCIJATIVNA PRAVILA

Sve do sada razmatrane tehnike modeliranja odnosile su se na klasifikacijske probleme, kojima je osnovni cilj konstrukcija opisa klasa u terminima prognostičkih atributa. Za razliku od njih, tehnika asocijativnih pravila nije namijenjena klasifikacijskim problemima, već analizi međuovisnosti među svim atributima skupa za učenje. Kod analize međuovisnosti ne postoji distinkcija između prognostičkih i prognoziranih atributa; bilo koji atribut (ili skup atributa) se može koristiti za predviđanje vrijednosti bilo kojeg drugog atributa (ili skupa atributa).

5.5.1. Reprezentacija modela

Asocijativna pravila su po svojoj formi analogna klasifikacijskim pravilima, osim što konzekventa pravila nije ograničena na predikciju klase. Tako za slučaj predviđanja vrijednosti jednog atributa asocijativno pravilo ima oblik (47):

$$cond(s) \rightarrow A_i(s) = a_j \quad , \tag{47}$$

gdje je $a_j \in Dom(A_i)$. Najčešći oblik antecedente pravila je konjunkcija elementarnih uvjeta na vrijednosti pojedinih atributa. Ukoliko pravilo predviđa vrijednost više od jednog atributa, isti oblik ima i konzekventa. Drugi oblici antecedente i konzekvente rijetko se koriste.
Semantika pojedinačnog pravila također odgovara semantici klasifikacijskog pravila: ako primjer s zadovoljava uvjet $cond(s)$ za njega se predviđaju vrijednosti atributa prema konzekventi pravila. Međutim, semantika skupa asocijativnih pravila nema posebno značenje. Dok se skup klasifikacijskih pravila promatra kao cjelina pri

klasifikaciji primjera, svako asocijativno pravilo se primjenjuje neovisno o ostalima. Razlog tomu je činjenica da se različita asocijativna pravila odnose na različite pravilnosti u skupu za učenje, odnosno općenito predviđaju vrijednosti različitih atributa.

Dvije su osnovne mjere kojima se izražava kvaliteta asocijativnih pravila: *značaj* i *pouzdanost*. Neka je sa $\omega_r \subseteq \sigma_r$ označen skup svih primjera koji zadovoljavaju i antecedentu i konzekventu pravila r. Skup ω_r se često naziva skup pozitivnih primjera pravila r. Značaj i pouzdanost pravila r definirani su izrazima (48):

$$\begin{aligned} značaj(r) &= \frac{|\omega_r|}{|S|}, \\ pouzdanost(r) &= \frac{|\omega_r|}{|\sigma_r|}. \end{aligned} \tag{48}$$

Značaj asocijativnog pravila je definiran kao udio pozitivnih primjera pravila u skupu za učenje, tj. ilustrira generalnost pravila. S druge strane, pouzdanost asocijativnog pravila kao udio pozitivnih primjera u prekrivaču pravila odgovara točnosti predikcije.

Dakako, kvalitetna asocijativna pravila su značajna i pouzdana, odnosno traže se što generalnija pravila uz adekvatnu točnost predikcije.

5.5.2. Postupak pretraživanja

U terminologiji asocijativnih pravila, elementarni uvjet na vrijednost atributa oblika $A_i(s)=a_j$ za $a_j \in Dom(A_i)$ naziva se *element*. Općenito govoreći, antecedenta i konzekventa asocijativnog pravila može biti konjunkcija bilo kojeg podskupa elemenata, pa je kardinalnost skupa potencijalnih pravila ogromna. Stoga je strategija iscrpnog pretraživanja prostora rješenja neprimjerena.

Cilj indukcije asocijativnih pravila je pronalaženje pravila čiji značaj i pouzdanost prelaze unaprijed definirani prag značaja odnosno pouzdanosti pravila. Standardni postupak indukcije asocijativnih pravila nosi naziv *Apriori* [1], a kapitalizira činjenicu da su interesantna samo pravila visokog značaja. Postupak je svojoj biti vrlo jednostavan, a sastoji se od dvije faze:

1. konstrukcija skupa pravila R čiji je značaj veći od praga značaja m_z,
2. izdvajanje pravila iz R prema kriteriju minimalne pouzdanosti m_p.

Budući da $s \in \omega_r$ mora zadovoljiti i antecedentu i konzekventu pravila, za izračun značaja pravila nije bitna razlika između antecedente i konzekvente. Iz tog razloga se u prvoj fazi postupka konstruiraju samo općenite konjunkcije elemenata čija značaj prelazi m_z. Tek u drugoj fazi procesa se razdvajanjem konjunkcija na antecedentu i konzekventu formiraju asocijativna pravila.

Prva faza indukcije asocijativnih pravila je iterativan postupak čiji su koraci prikazani na slici 15. Postupak iterira po broju elemenata konjunkcije, tj. u i-toj iteraciji postupka se promatraju konjunkcije od i različitih elemenata. Skup kandidatnih konjunkcija R' od i elemenata se konstruira koristeći već izdvojene konjunkcije od i–1 elemenata. Naime, da bi konjunkcija od i elemenata imala značaj veći od m_z, svaka njena generalizacija od i–1 elemenata također mora imati značaj

veći od m_z. Konzultiranjem konjunkcija od i–1 elemenata već prisutnih u hash tablici H značajno se smanjuje broj razmatranih kandidatnih konjunkcija od i elemenata.

```
1.  S := skup za učenje; A := skup svih atributa iz S
2.  za i := 1...|A| ponavljaj
    2.1. konstruiraj skup R' kandidatnih konjunkcija od i različitih elemenata
    2.2. izračunaj značaj svake konjunkcije iz R' korištenjem skupa za učenje S
    2.3. iz R' izbaci sve konjunkcije čiji značaj ne prelazi prag mz
    2.4. ako je R' prazan prekini ponavljanje
    2.5. sve konjunkcije iz R' s pripadajućim značajem dodaj u hash tablicu H
3.  hash tablica H sadrži sve konjunkcije čiji značaj prelazi mz
```

Slika 15: Prva faza postupka indukcije asocijativnih pravila

Međutim, nužan uvjet na značaj generalizacija konjunkcije nije i dovoljan. Stoga se zadovoljenje kriterija značaja za svaku kandidatnu konjunkciju mora eksplicitno provjeriti na skupu za učenje. Svaka iteracija postupka uključuje jedan prolazak kroz skup za učenje radi izračuna značaja kandidatnih konjunkcija. Kandidati čiji značaj prelazi prag m_z isključuju se iz razmatranja, a ostali se pridodaju hash tablici H.
Postupak može imati najviše $|A|$ iteracija, jer konjunkcija ne može sadržavati više elemenata s istim atributom. Broj iteracija može biti i manji, ukoliko nije moguće konstruirati konjunkcije s traženim značajem.

U drugoj fazi indukcije asocijativnih pravila formiraju se dovoljno pouzdana asocijativna pravila na osnovu izdvojenih konjunkcija zadovoljavajućeg značaja. Postupak je također iterativnog karaktera, a prikazan je na slici 16.

```
1.  H := hash tablica značajnih konjunkcija
2.  za svaku konjunkciju c iz tablice H ponavljaj
    2.1. iz c konstruiraj skup kandidatnih pravila R' razdvajanjem elemenata
         konjunkcije c na antecedentu i konzekventu pravila
    2.2. izračunaj pouzdanost svakog pravila iz R' korištenjem hash tablice H
    2.3. iz R' izbaci sva pravila čija pouzdanost ne prelazi prag mp
    2.4. sva pravila iz R' dodaj u skup asocijativnih pravila R
3.  vrati R kao rezultirajući skup asocijativnih pravila
```

Slika 16: Druga faza postupka indukcije asocijativnih pravila

Svaka iteracija postupka razmatra jednu konjunkciju iz hash tablice značajnih konjunkcija. Različitim kombinacijama razdvajanja elemenata konjunkcije na antecedentu i konzekventu pravila nastaje skup kandidatnih pravila R'. Jednostavniji oblik algoritma konstruira asocijativna pravila koja predviđaju vrijednost samo jednog atributa, tj. čija konzekventa ima samo jedan element. U tom slučaju skup kandidatnih pravila R' tvori se naizmjeničnim izdvajanjem svakog od elemenata konjunkcije u konzekventu pravila.
Vrijedi naglasiti da se u drugoj fazi indukcije asocijativnih pravila uopće ne koristi skup podataka za učenje S. Naime, svi potrebni parametri za računanje pouzdanosti pravila iz R' mogu se iščitati iz hash tablice značajnih konjunkcija. Antecedenta pravila predstavlja generalizaciju početne konjunkcije, pa se i sama mora naći u hash tablici značajnih konjunkcija. Svrha korištenja hash tablice je upravo lakše pronalaženje potrebnih konjunkcija. Nakon računanja pouzdanosti, iz skupa

kandidatnih pravila R' izbacuju se pravila koja ne prelaze prag pouzdanosti m_p, a preostala ulaze u završni skup asocijativnih pravila R.

Verzija algoritma koja konstruira pravila s konzekventama od više elemenata mijenja način konstrukcije kandidatnog skupa pravila R'. Iscrpni pristup bi razmatrao svaki podskup elemenata polazne konjunkcije kao konzekventu pravila. Ovakav način konstrukcije skupa R' nije praktičan za brojnije konjunkcije, zbog eksponencijalnog porasta broja podskupova u odnosu na broj elemenata.
Međutim, postoji i bolji način koji kandidatna pravila konstruira iterativno, prema broju elemenata u konzekventi pravila. Slično prvoj fazi indukcije pravila, u iteraciji koja konstruira pravila s konzekventom od i elemenata koriste se rezultati prethodne iteracije. Naime, da bi pravilo s konzekventom od i elemenata imalo pouzdanost veću od m_p, svako od pravila s generalnijom konzekventom nastalo iz iste polazne konjunkcije također mora imati pouzdanost veću od m_p. Pravila sa generalnijim konzekventama od i–1 članova su generirana u prethodnoj iteraciji, pa se koriste pri odabiru kandidata za i-člane konzekvente u smislu gornjeg nužnog uvjeta. Spomenuti uvjet nije i dovoljan, što znači da su konstruirana pravila samo kandidati čiju pouzdanost treba utvrditi korištenjem podataka iz hash tablice.
Ovakav način konstrukcije pravila s višečlanim konzekventama značajno smanjuje broj razmatranih kandidatnih pravila i reducira složenost samog postupka.

5.5.3. Svojstva indukcije asocijativnih pravila

Asocijativna pravila nisu namijenjena problemima klasifikacijskog tipa, već analizi općenitih ovisnosti među atributima skupa za učenje. Međutim, ponekad se koriste u pripremi podataka za klasifikacijske probleme, gdje mogu poslužiti pri nadomještanju neodređenih vrijednosti atributa. Kao samostalna tehnika asocijativna pravila imaju primjenu u mnogim domenama, gdje se zbog svoje jednostavnosti, jasnoće i međusobne neovisnosti lako interpretiraju u domeni izvornog problema.

Algoritmi kojima se generiraju asocijativna pravila su u osnovi vrlo jednostavni, ali ne treba podcjenjivati njihove zahtjeve za računalnim resursima i vremensku složenost postupka. Budući da se u pravilu primjenjuju na glomaznim skupovima podataka sa velikim brojem atributa, cijeni se svako poboljšanje performansi postupka indukcije asocijativnih pravila.
Opisani postupak prolazi jednom kroz cijeli skup za učenje u svakoj iteraciji prve faze postupka. Ukoliko je dohvat podataka iz skupa za učenje usko grlo, moguće je u jednom prolazu kroz skup za učenje odraditi više uzastopnih iteracija postupka. To će rezultirati manjim brojem prolaza kroz skup za učenje, ali će se zbog odlaganja izračuna značaja konjunkcije povećati broj kandidatnih pravila u razmatranju.
Vrijeme indukcije asocijativnih pravila bitno ovisi o odabranom pragu značaja m_z, dok prag pouzdanosti m_p nema veći utjecaj. Niži prag značaja može enormno povećati broj razmatranih značajnih konjunkcija i kandidatnih asocijativnih pravila. Osim toga, utječe na kriterij zaustavljanja iteracije u prvoj fazi postupka, pa može povećati i broj prolazaka kroz skup za učenje. Stoga se preporuča problemu inicijalno pristupiti s visokim pragom značaja, te ga postupno smanjivati ukoliko je potrebno.

Najčešća primjena asocijativnih pravila je analiza potrošačkih košarica, prvenstveno zbog jasnoće i iskoristivosti dobivenih pravila. Cilj analize je istražiti u kojoj mjeri su važni proizvodi korelirani u smislu istodobne kupnje više proizvoda. Svaka obavljena kupnja, odnosno skup istodobno kupljenih proizvoda čini jedan primjer skupa za učenje. Atributi primjera odgovaraju proizvodima koji se prodaju, binarnog su tipa i označavaju prisutnost proizvoda u obavljenoj kupnji.
Dakle, u analizi potrošačkih košarica broj atributa u skupu za učenje odgovara broju proizvoda za prodaju, što je u pravilu ogroman broj. Budući da svaka kupnja sadržava relativno mali broj proizvoda, primjeri za učenje se iz razloga praktičnosti često zapisuju navođenjem samo prisutnih proizvoda. Osim memorijskih zahtjeva, veliki broj atributa povećava i vremensku zahtjevnost postupka, budući da broj mogućih konjunkcija raste eksponencijalno s povećanjem broja atributa. Jedan od načina borbe protiv prevelikog broja atributa je taksonomija proizvoda, odnosno svrstavanje srodnih proizvoda u kategorije. Ponekad se stotine proizvoda može svesti na svega nekoliko kategorija, uz zadržavanje generalnih svojstava bitnih za analizu potrošačkih košarica. Analiza se tada ne vrši nad pojedinačnim proizvodima, već na kategorijama, što bitno smanjuje broj korištenih atributa.
Negativni efekti prevelikog broja atributa mogu se ograničiti promjenom kriterija zaustavljanja iteracije u prvoj fazi indukcije pravila. Ograničavanjem broja iteracija iz razmatranja se isključuju konjunkcije s većim brojem elemenata. To za posljedicu ima smanjenje maksimalne složenosti rezultirajućih pravila u smislu broja elemenata koje pravilo sadrži. Kod primjena u čijem su fokusu jednostavnija pravila, ovo ograničenje ne predstavlja značajniju kvalitativnu redukciju rezultirajućeg skupa asocijativnih pravila.

5.6. TEHNIKE SEGMENTIRANJA PODATAKA

Za razliku od klasifikacijskih tehnika, tehnike segmentiranja podataka spadaju u skupinu postupaka nenadziranog učenja (učenje bez učitelja). Skup primjera za učenje kod tehnika nenadziranog učenja nema definirane prognozirane atribute, odnosno klasu primjera. Dok je cilj postupaka nadziranog učenja konstrukcija opisa klasa, postupci nenadziranog učenja pokušavaju otkriti globalne strukture u podacima, ne praveći razliku među pojedinim atributima.
Konkretno, cilj tehnika segmentiranja podataka je grupiranje primjera iz skupa za učenje po principu sličnosti. Skup za učenje nastoji se podijeliti u više podskupova ili grupa primjera tako da su zadovoljena dva osnovna kriterija:

- svaki grupa predstavlja homogenu cjelinu, tj. primjeri koji pripadaju istoj grupi međusobno su slični,
- podjela na grupe odražava razlike među primjerima, tj. primjeri koji pripadaju različitim grupama međusobno se značajno razlikuju.

Dakle, tehnike segmentiranja podataka pokušavaju otkriti prirodno grupiranje sličnih primjera. Ovisno o samoj tehnici segmentiranja, grupe mogu biti definirane na različite načine:

- identificirane grupe mogu biti disjunktne, tako da svaki primjer pripada samo jednoj od grupa;
- grupe se mogu preklapati, tj. primjer može istovremeno pripadati nekolicini grupa;

- grupe mogu biti definirane probabilistički, u tom slučaju primjer pripada svakoj od grupa s određenom vjerojatnošću;
- grupe mogu biti hijerarhijski strukturirane, sa najgrubljom podjelom primjera u vrhu hijerarhije, uz postupno razlaganje svake od grupa niže u hijerarhiji.

Osnovni principi tehnika segmentiranja podataka se razlikuju od klasifikacijskih tehnika, jer se ne baziraju na promatranju ciljnog atributa. Međutim, elementi postupaka pojedinih tehnika segmentiranja pokazuju zamjetne sličnosti s postupcima poznatim iz klasičnih klasifikacijskih tehnika.
Postoji veći broj različitih tehnika segmentiranja podataka. Najjednostavnija od njih je algoritam *k srednjih vrijednosti* [26], koji je u upotrebi već više desetljeća. Kasnije su razvijene bitno složenije tehnike segmentiranja, primjerice probabilističke metode od kojih je najpoznatiji *AutoClass* algoritam [10]. Za ilustraciju temeljnih principa segmentiranja podataka dovoljan je prikaz algoritma *k* srednjih vrijednosti.

5.6.1. Algoritam k *srednjih vrijednosti*

Segmentiranje podataka algoritmom *k* srednjih vrijednosti primjere dijeli u *k* disjunktnih grupa, gdje je *k* unaprijed definirani parametar. To je u osnovi jednostavan postupak koji u mnogim elementima nalikuje tehnici klasifikacije pamćenjem primjera prema principu najbližeg susjeda. Postupak se zasniva na računanju udaljenosti primjera korištenjem definirane metrike. Kao i kod tehnike najbližih susjeda, najčešće se koristi Euklidska metrika.
Iz istih razloga, izračun udaljenosti primjera podrazumijeva normalizaciju numeričkih atributa. Proširenje definicije udaljenosti na nominalne atribute i neodređene vrijednosti atributa također se oslanja na standardna poopćenja operacije razlike na vrijednostima atributa.
Algoritam *k* srednjih vrijednosti je iterativna procedura u kojoj centralnu ulogu igra pojam *centroida*. Centroid je točka u univerzumu koja predstavlja srednju (prosječnu) lokaciju određene grupe primjera. Koordinate centroida računaju se kao prosječne vrijednosti koordinata svih primjera grupe koju centroid predstavlja. Koraci algoritma *k* srednjih vrijednosti prikazani su na slici 17.

```
1.  k := definirani broj grupa; Δc := prag pomaka centroida
2.  na slučajan način odaberi k točaka u univerzumu kao početne vrijednosti
    centroida svih k grupa
3.  iteriraj
    3.1. pridijeli svaki primjer centroidu kojem je primjer najbliži,
         formirajući na taj način k disjunktnih grupa primjera
    3.2. izračunaj pozicije novih centroida grupa usrednjavanjem koordinata
         primjera koji pripadaju grupi
    3.3. izračunaj pomak lokacije svakog centroida
4.  ponavljaj dok postoji centroid s pomakom > Δc
5.  segmentacija primjera u k grupa je definirana funkcijom udaljenosti i
    lokacijama centroida
```

Slika 17: Algoritam *k* srednjih vrijednosti

Inicijalno se lokacije *k* početnih centroida biraju na slučajan način. To mogu biti proizvoljne točke univerzuma, ali je čest slučaj da se početni centroidi biraju među konkretnim primjerima iz skupa podataka za učenje.

Svaka iteracija postupka vrši korekciju pozicije svakog od *k* centroida, postavljajući centroid u "središte" grupe primjera koja mu je pridijeljena. Promjena lokacije centroida općenito uzrokuje preraspodjelu primjera po grupama, a time i novi pomak lokacije centroida u slijedećoj iteraciji. Postupak staje kada se pozicije svih *k* centroida stabiliziraju. U praksi se zadovoljavajuća konvergencija obično postiže već nakon razmjerno malog broja iteracija.

5.6.2. Svojstva tehnika segmentiranja podataka

Algoritam *k* srednjih vrijednosti je jednostavna i razmjerno djelotvorna tehnika segmentiranja podataka koja postoji već više desetljeća. Osnovni oblik algoritma zbog sličnog pristupa dijeli mnoga svojstva s klasifikacijom po principu najbližeg susjeda. Tako kod glomaznih skupova za učenje postupak može biti vremenski vrlo zahtjevan. Također, pretpostavlja se da su svi atributi jednako vrijedni i međusobno neovisni. Stoga u osnovnom obliku algoritma ne postoji način reguliranja utjecaja pojedinih atributa, pa irelevantni ili međusobno korelirani atributi mogu značajno degradirati kvalitetu segmentacije. Međutim, tijekom godina je razvijen veći broj varijacija i unapređenja osnovnog postupka, što je donijelo zamjetna poboljšanja u kvaliteti segmentacije.

Jedan od temeljnih nedostataka algoritma *k* srednjih vrijednosti je činjenica da se broj rezultirajućih grupa primjera unaprijed zadaje. Ukoliko odabrani parametar *k* nije primjeren promatranom problemu, ni rezultati ne mogu biti kvalitetni. Ispravan pristup odabiru broja grupa je eksperimentiranje s različitim brojem grupa. U konačnici se odabire ono grupiranje koje ima najpovoljniji omjer intra-grupnih i inter-grupnih udaljenosti primjera. Sofisticiranije tehnike segmentiranja (npr. AutoClass) same mjere kvalitetu grupiranja i optimiraju broj grupa u dodatnoj petlji postupka.

Slučajnost inicijalnog odabira lokacija centroida unosi nedeterminizam u algoritam *k* srednjih vrijednosti. Naime, rezultat segmentacije općenito ovisi o početnim lokacijama centroida koje se biraju na slučajan način, što predstavlja drugi važan nedostatak ove tehnike. Gledano u svjetlu segmentacije kao pretraživanja, različite početne točke dovode do različitih lokalnih minimuma. Efekt se donekle može ublažiti ponavljanjem cijelog postupka više puta uz različite početne lokacije centroida, te odabirom najboljeg od dobivenih grupiranja.

U prednosti algoritma *k* srednjih vrijednosti može se ubrojiti činjenica da je rezultirajući model operativan, odnosno može se koristiti i na dotad neviđenim primjerima u smislu dodjele jednoj od otkrivenih grupa. Iako ne proizvodi eksplicitne opise otkrivenih grupa, vizualizacija grupa određenih centroidima i funkcijom udaljenosti je jednostavna, a u nižim dimenzijama i vrlo ilustrativna.

Uz male modifikacije, moguće je proizvesti i drugačiji oblik modela. Na primjer, hijerarhijska struktura grupa može se dobiti primjenom algoritma za k=2 na izvornom skupu za učenje, te rekurzivnom primjenom istog postupka na svakoj od dobivenih grupa. Konačan rezultat je hijerarhija grupa u obliku binarnog stabla.

Općenito, tehnike segmentiranja koriste se u slučajevima kada se očekuje postojanje prirodnih grupa u podacima. Otkriveni segmenti ili grupe podataka trebali bi predstavljati grupe primjera koji imaju mnogo toga zajedničkog. Korisnost dobivenih

rezultata u velikoj mjeri ovisi o smislenosti i mogućnosti primjene u okviru domene promatranog problema.
Većina tehnika segmentiranja ne daje deskriptivan opis otkrivenih grupa. Kako bi se rezultat mogao verificirati u terminima domene promatranog problema, takve grupe je potrebno adekvatno interpretirati. Za to postoji nekoliko načina:

- Pripadnost određenoj grupi može se iskoristiti kao poseban ciljni atribut za novi, klasifikacijski problem. Neka od deskriptivnih tehnika modeliranja podataka (npr. stabla odlučivanja) može se iskoristiti za opis pojedinih grupa primjera.
- Grupe se mogu vizualno prikazati korištenjem 2D ili 3D grafova, ili nekom od novijih višedimenzionalnih vizualizacijskih tehnika.
- Razlike među vrijednostima atributa pojedinih grupa mogu se razmatrati pojedinačno, od atributa do atributa.

Tehnike segmentiranja podataka svoju primjenu imaju i kod klasičnih klasifikacijskih problema, gdje se mogu koristiti u fazi pripreme podataka za neku od klasifikacijskih tehnika. Naime, primjerena podjela skupa za učenje može znatno reducirati kompleksnost određenog problema. Skup primjera za učenje se dijeli na više podskupova tehnikom segmentiranja podataka, a dobiveni podskupovi se potom odvojeno modeliraju odabranom klasifikacijskom tehnikom. Ovakva dvostepena procedura na kraju može rezultirati boljim konačnim rezultatima (bilo u prediktivnom ili deskriptivnom smislu) nego bez prethodne primjene tehnika segmentiranja podataka.

5.7. OSTALE TEHNIKE MODELIRANJA

Ovaj pregled tehnika modeliranja u inteligentnoj analizi podataka ni u kojem smislu nije potpun. Opisane su samo neke od poznatijih tehnika modeliranja, i to ponegdje samo do razine osnovnih elemenata postupka. U stvarnosti često postoji značajna razlika između osnovnih principa i stvarnih implementacija opisanih postupaka koje se primjenjuju u praksi. Osnovni mehanizam, kao i oblik ulaza i rezultirajućeg modela su isti. Međutim, stvarne implementacije karakterizira daleko veća posvećenost detaljima, a time i složenost samog postupka. Uzrok tome je potreba da se na adekvatan i robustan način nose sa uvjetima i specifičnostima koje sa sobom donose stvarni problemi iz različitih domena.
Zbog toga u pravilu postoji veći broj različitih varijacija i nadogradnji istog temeljnog postupka, ili čak kombinacija više postupaka. Mjesta za poboljšanje često se pronalaze u naprednijim tehnikama pretraživanja, ublažavanju pretpostavki pod kojima tehnika operira, različitim optimizacijama koje povećavaju efikasnost postupka, te brojnim prilagodbama stvarnim problemima i situacijama. Dobar primjer varijabilnosti tehnika modeliranja su klasifikacijska pravila. Ovdje je opisana samo jedna od mnogih tehnika indukcije klasifikacijskih pravila. Drugi oblici tehnike indukcije pravila mogu biti varijacije na opisani postupak, ali se mogu i bitno razlikovati u pristupu. Primjerice, neke tehnike indukcije pravila upotrebljavaju drugi oblik pravila, a samim time i postupak indukcije. Neke u pretraživanju koriste pristup odozdo na gore, odnosno polaze od pojedinačnih primjera ciljne klase. Postoje i

kombinacije s drugim tehnikama, kod kojih se primjerice postupak indukcije pravila oslanja na konstrukciju parcijalnih stabala odlučivanja.
Osim ovdje opisanih tehnika modeliranja postoje i druge koje se zasnivaju na drugim oblicima reprezentacije modela i različitim mehanizmima indukcije modela. Primjerice, postoji cijela skupina strukturnih načina reprezentacije modela u koju spadaju formalizmi poput *semantičkih mreža* ili *okvira i shema* [12]. Također, nije spomenuta cijela klasa problema koju karakterizira predviđanje vrijednosti numeričkih atributa. Problemi numeričke predikcije su u osnovi nalik na klasifikacijske probleme, osim što ciljni atribut nije nominalnog već numeričkog tipa. Dakle, modeli koje konstruiraju tehnike numeričke predikcije služe predviđanju vrijednosti odabranog numeričkog atributa. U takve tehnike spadaju npr. *regresijska stabla* [7] ili *lokalno naglašena linearna regresija* [4].

Uzrok raznolikosti tehnika modeliranja leži u činjenici da ne postoji jedna superiorna tehnika modeliranja koja bi proizvodila najbolje rezultate za svaki tip problema i u svim oblicima primjene. Različite tehnike karakteriziraju različita svojstva, i neke su bolje od drugih prilagođene jednoj vrsti problema ili obliku pravilnosti koji se u podacima može pojaviti. Postoje čak specijalizirane tehnike modeliranja izgrađene za točno određenu namjenu i prilagođene točno određenoj domeni i problemu. U takve spada npr. *Meta-Dendral*, koji ima primjenu u analizi strukture molekula korištenjem spektroskopskih metoda.

6. Evaluacija otkrivenog znanja

Postoji veći broj metoda modeliranja pravilnosti u podacima, a i pojedina metoda na istom skupu primjera za učenje rezultira različitim modelima uz variranje parametara metode. Činjenica da isti problem i isti skup podataka za učenje može proizvesti veliki broj različitih modela naglašava potrebu vrednovanja kvalitete modela u odnosu na promatrani problem. Zbog toga je evaluacija otkrivenog znanja jedna od bitnih komponenti procesa inteligentne analize podataka. Ovisno o karakteristikama promatranog problema, mogu se primijeniti različite metode evaluacije. Budući da su u inteligentnoj analizi podataka najčešći klasifikacijski problemi, prvenstveno će biti riječi o evaluaciji klasifikacijskih modela.

6.1. MJERE ZA EVALUACIJU KLASIFIKACIJSKIH MODELA

Temeljni zadatak evaluacije klasifikacijskih modela je izmjeriti u kojem stupnju klasifikacija sugerirana izgrađenim modelom odgovara stvarnoj klasifikaciji primjera. Ovisno o načinu promatranja performansi modela, postoji više različitih mjera za evaluaciju modela. Odabir najpogodnije mjere treba biti uvjetovan karakteristikama promatranog problema i načinom njegove primjene.

6.1.1. Standardne mjere evaluacije modela

Osnovni pojam pri evaluaciji klasifikacijskih modela je pojam greške. Greškom se smatra pogrešna klasifikacija primjera: primjena klasifikacijskog modela na odabrani primjer rezultira prognozom klase koja je različita od stvarne klase primjera. Ukoliko se svakoj grešci pridaje jednak značaj, tada je ukupan broj grešaka na promatranom skupu primjera dobar indikator rada klasifikacijskog modela.
Točnost kao mjera za evaluaciju kvalitete klasifikacijskih modela oslanja se upravo na ovaj pristup. Točnost se definira kao omjer broja ispravno klasificiranih primjera prema ukupnom broju klasificiranih primjera.

$$Točnost = \frac{broj\ ispravno\ klasificiranih\ primjera}{ukupan\ broj\ primjera} \qquad (49)$$

Dva su osnovna nedostatka točnosti kao mjere za evaluaciju:
- zanemaruju se razlike između tipova grešaka;
- zavisna je o distribuciji klasa u skupu podataka, a ne o karakteristikama primjera.

Naime, u većini praktičnih situacija je vrlo važno razlikovati određene tipove grešaka. Primjerice, sustav koji detektira potencijalno neovlaštenu uporabu kreditnih kartica može pogrešno označiti regularnu uporabu kao neovlaštenu, ili neovlaštenu uporabu označiti kao regularnu. Pritom lažno alarmiranje o neovlaštenoj uporabi ima bitno manju težinu od propusta uočavanja stvarne zlouporabe kartice.
Stoga se u slučajevima kada je potrebno razlikovati više tipova grešaka rezultat klasifikacije prikazuje u obliku dvodimenzionalne *matrice grešaka*. Svaki redak matrice odgovara jednoj klasi, te bilježi broj primjera kojima je to prognozirana klasa. Svaki stupac matrice također je obilježen po jednom klasom, a bilježi broj primjera kojima je to stvarna klasa. Slika 18 je ilustracija matrice grešaka za neki klasifikacijski problem s tri klase.

		Stvarna klasa		
		A	**B**	**C**
Prognozirana klasa	**A**	78	0	1
	B	3	47	2
	C	0	1	52

Slika 18: Primjer matrice grešaka za klasifikacijski problem s tri klase

Broj točno klasificiranih primjera nalazi se uzduž dijagonale matrice. Svi ostali elementi matrice označavaju broj primjera koji su neispravno klasificirani kao neka od preostalih klasa. Na primjer, iz matrice se vidi da su od ukupnog broja primjera klase *C* pogrešno klasificirana 3 primjera, i to na način da je jedan svrstan u klasu *A*, a dva u klasu *B*. Na taj način matrica grešaka omogućava kvalitetniju analizu različitih tipova grešaka.
Najveći broj mjera za evaluaciju klasifikacijskih modela odnosi se na klasifikacijske probleme s dvije klase. To ne predstavlja posebno ograničenje za upotrebljivost tih mjera, budući da se problemi s većim brojem klasa mogu prikazati u obliku niza problema s dvije klase. Svaki od njih posebno izdvaja jednu od klasa kao ciljnu klasu, a skup podataka se dijeli na pozitivne i negativne primjere ciljne klase, s tim da u negativne spadaju primjeri svih ostalih klasa. Slika 19 je prikaz matrice grešaka za klasifikacijski problem s dvije klase.

	Pozitivni primjeri klase *C*	Negativni primjeri klase *C*
Pozitivna prognoza klase C	Stvarno pozitivni (*TP*)	Lažno pozitivni (*FP*)
Negativna prognoza klase *C*	Lažno negativni (*FN*)	Stvarno negativni (*TN*)

Slika 19: Matrica grešaka za klasifikacijski problem s dvije klase

Iz tablice se vidi da su moguća četiri različita rezultata prognoze. *Stvarno pozitivni* i *stvarno negativni* ishodi predstavljaju ispravnu klasifikaciju. *Lažno pozitivni* i *lažno negativni* ishodi predstavljaju dva moguća tipa greške. Lažno pozitivan primjer je negativan primjer klase *C* koji je pogrešno klasificiran kao pozitivan, dok je lažno negativan u stvari pozitivan primjer klase *C* koji je pogrešno klasificiran kao negativan.

Senzitivnost i *specifičnost* spadaju u mjere za evaluaciju klasifikacijskih modela koje razlikuju dva spomenuta tipa greške. Zasnivaju se na istom principu kao i točnost: senzitivnost mjeri točnost u pozitivnim primjerima, a specifičnost predstavlja točnost u negativnim primjerima.

$$\begin{aligned} Senzitivnost &= \frac{TP}{TP + FN} \\ Specifičnost &= \frac{TN}{TN + FP} \end{aligned} \tag{50}$$

Ovaj par mjera se vrlo često koristi za evaluaciju modela u medicinskoj dijagnostici. Nedostaci tipični za točnost kod njih se ne pojavljuju: mjere uzimaju u obzir različite vrste grešaka, a ne ovise ni o distribuciji klasa budući da je svaka fokusirana na samo jednu od klasa.

Odziv i *preciznost* su drugi par mjera za evaluaciju klasifikacijskih modela koji je osjetljiv na različite tipove grešaka. Najčešće se koristi u problemima kod kojih je broj stvarno pozitivnih primjera malen u odnosu na broj stvarno negativnih primjera (tj. $TP << TN$). Dobar primjer takvog problema je pronalaženje informacija: broj dokumenata relevantnih za neki upit je bitno manji od ukupnog broja dostupnih dokumenata.
Odziv i preciznost se definiraju izrazima:

$$Odziv = \frac{TP}{TP + FN} \quad ,$$
$$Preciznost = \frac{TP}{TP + FP} \quad . \tag{51}$$

Jednako kao i senzitivnost, odziv je definiran kao točnost u pozitivnim primjerima. S druge strane, preciznost je točnost u pozitivnoj prognozi ciljne klase. U kontekstu problema pronalaženja informacija, odziv predstavlja omjer pronađenih relevantnih dokumenata u odnosu na ukupan broj relevantnih dokumenata, dok je preciznost udio relevantnih dokumenata u ukupnom broju pronađenih dokumenata.

Pozitivna i *negativna prediktivna vrijednost* su još jedan par mjera koji uvažava različite tipove klasifikacijskih grešaka. Zasnivaju se na principima preciznosti: pozitivna prediktivna vrijednost predstavlja preciznost u pozitivnim primjerima, dok je negativna prediktivna vrijednost preciznost u negativnim primjerima.

$$Pozitivna\ prediktivna\ vrijednost = \frac{TP}{TP + FP}$$
$$Negativna\ prediktivna\ vrijednost = \frac{TN}{TN + FN} \tag{52}$$

Jedno od bitnih obilježja evaluacije klasifikacijskih modela uporabom parova mjera je komplementarnost. Naime, navedeni parovi mjera ilustriraju specifičnu točnost klasifikacijskog modela sa donekle suprotstavljenih pozicija. U pravilu se variranjem parametara odabrane tehnike modeliranja može nauštrb jedne od mjera povećati specifična točnost modela prikazana drugom mjerom. Stoga odabir prikladnih parametara odabrane tehnike modeliranja predstavlja optimizacijski problem u kojem se uz određene uvjete na vrijednost jedne mjere želi maksimizirati druga mjera.

Ponekad je kvalitetu klasifikacijskog modela potrebno izraziti jednim brojem, a ne parom ovisnih mjera. To se može postići i uporabom parova mjera, na način da se vrijednost jedne mjere fiksira, te se uz taj uvjet se promatra samo druga mjera.

Primjerice, može se promatrati mjera preciznosti uz fiksiranu vrijednost odziva od 10%. Ovako izvedena mjera naziva se *preciznost na 10%*. Međutim, češće se koristi usrednjavanje jedne od mjera po više fiksiranih vrijednosti druge mjere. Tako se u pronalaženju informacija često koristi *srednja preciznost u tri točke* (u pravilu se radi o vrijednostima odziva od 20%, 50% i 80%).
Osim ovako izvedenih mjera, postoje i mjere koje se ne zasnivaju na fiksiranju jedne komponente para mjera. U takve spada *F-mjera*, koja se definira sa

$$F\text{–}mjera = \frac{2 \times odziv \times preciznost}{odziv + preciznost} = \frac{2TP}{2TP + FP + FN} \quad . \tag{53}$$

Jedna od pojedinačnih mjera koja se zasniva na točnosti definirana je izrazom

$$\frac{TP \times TN}{(TP + FN) \times (FP + TN)} \quad , \tag{54}$$

što je u stvari produkt mjera senzitivnosti i specifičnosti.

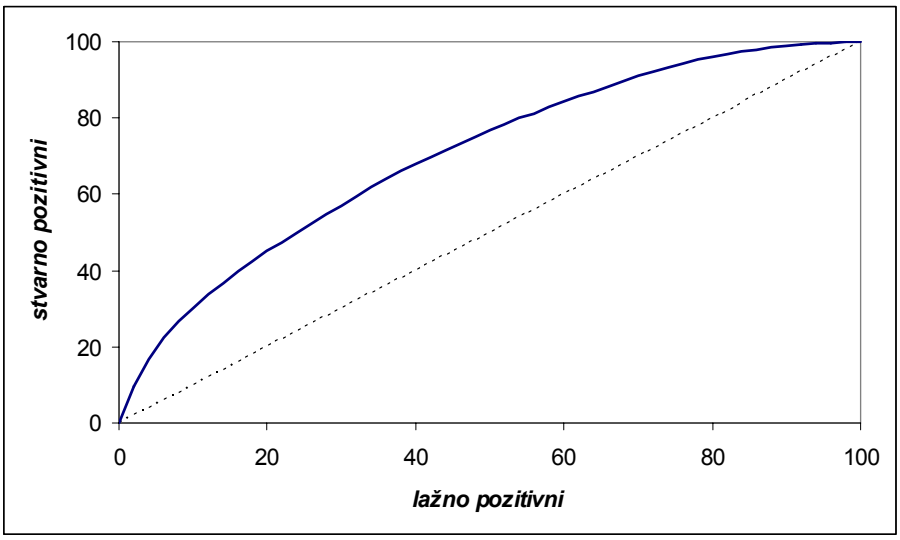
6.1.2. ROC analiza

Kvaliteta klasifikacijskog modela izražena jednim brojem ili parom brojeva ne može u potpunosti ilustrirati varijabilnost rješenja koja se mogu izvesti upotrebom odabrane tehnike modeliranja. Međusobne ovisnosti različitih mjera i/ili parametara bolje se mogu opisati upotrebom grafova kojima je moguće prikazati velik raspon različitih mogućnosti. Jedna od takvih obuhvatnijih, grafički orijentiranih mjera kvalitete klasifikacijskog modela je ROC analiza [46]. Preuzeta je iz područja detekcije signala, odakle joj potječe i ime (ROC je kratica engleskog termina *Receiver Operating Characteristic*).
Za razumijevanje ROC analize najvažniji je pojam graničnih vrijednosti pouzdanosti određenog klasifikatora. Primjerice, senzitivnost i specifičnost su dvije mjere koje karakteriziraju specifičnu točnost klasifikatora u pozitivnim i negativnim primjerima. Kada je granična vrijednost pouzdanosti postavljena visoko, tj. naglašen je kriterij pouzdanosti klasifikatora, senzitivnost klasifikatora će biti niska, a specifičnost visoka. Ako se taj kriterij pouzdanosti snizi, senzitivnost će rasti, a specifičnost padati. Na taj način se dva klasifikacijska modela mogu komparirati preko širokog spektra pouzdanosti, tipično generirajući jednu krivulju koja opisuje ovisnost broja stvarno pozitivnih primjera naspram broja lažno pozitivnih primjera detektiranih modelom. Ukoliko je sa $T = (x, y)$ označena točka na ROC krivulji promatranog klasifikatora, vrijednosti koordinata x i y dobiju se pomoću izraza (55).

$$\begin{aligned} x &= \frac{FP}{FP + TN} \\ y &= \frac{TP}{TP + FN} \end{aligned} \tag{55}$$

Dakle, na vertikalnoj osi ROC krivulja označava broj stvarno pozitivnih primjera izražen kao udio u ukupnom broju pozitivnih primjera. S druge strane, na

horizontalnoj osi ROC krivulja bilježi odgovarajući broj lažno pozitivnih primjera izražen kao udio u ukupnom broju negativnih primjera. Na slici 20 prikazan je primjer ROC krivulje koja ilustrira rad klasifikatora uz različitim uvjetima pouzdanosti.



Slika 20: Primjer ROC krivulje

Dijagonala naznačena na slici koja spaja donji lijevi i gornji desni kut predstavlja potpuno slučajnu klasifikaciju. Nasuprot tome, ROC krivulja savršenog klasifikatora bi slijedila lijevu i gornju os grafa. Krivulje realnih klasifikatora leže između ova dva slučaja, dakle u gornjem trokutu grafa, iznad naznačene dijagonale.
Vrijedi napomenuti da detalji oblika ROC krivulje nekog klasifikatora ovise o skupu primjera koji je korišten za izradu ROC krivulje. Ovisnost o specifičnostima korištenog skupa primjera može se umanjiti upotrebom više različitih skupova primjera, te usrednjavanjem rezultata.

6.1.3. Vjerojatnosne mjere

Sve do sada razmatrane mjere za evaluaciju klasifikacijskih modela zasnivaju se na usporedbi broja ispravno i pogrešno klasificiranih primjera, s tim da neke od mjera razlikuju i tipove ispravne ili pogrešne klasifikacije.
Ovakve mjere su prvenstveno namijenjene tehnikama modeliranja koje rezultiraju kategoričkom predikcijom klase. Za razliku od toga, tehnike modeliranja koje pridružuju vjerojatnost svakoj predikciji omogućuju rafiniraniju ocjenu točnosti modela. Prirodan način nijansiranja različitih ishoda predikcije je upravo vjerojatnost pridružena predikciji, koju mjera evaluacije može uzeti u obzir. Primjerice, ispravna klasifikacija s pridruženom vjerojatnošću od 99% trebala bi imati više utjecaja na ocjenu točnosti nego ispravna klasifikacija s vjerojatnošću od 51%.
Naravno, upotreba mjera za evaluaciju koje uzimaju u obzir vjerojatnost pridruženu predikciji ovisi o promatranom problemu i načinu primjene klasifikacijskog modela. Ukoliko je u konačnici bitna samo ispravna klasifikacija, a realistična procjena vjerojatnosti klasifikacije nije od značaja, prikladnije su standardne mjere evaluacije.

Međutim, bilo kakve analize koje su usmjerene na strukturu rezultata predikcije zahtijevaju uporabu vjerojatnosnih mjera evaluacije.

6.1.3.1. KVADRATNA FUNKCIJA GUBITKA

U klasifikacijskom problemu s k klasa, rezultat predikcije probabilističkog klasifikacijskog modela je k-dimenzionalni vektor vjerojatnosti $p = (p_1, \ldots, p_k)$ čija suma komponenti iznosi 1. Stvarna klasa primjera također se može izraziti vektorom vjerojatnosti $a = (a_1, \ldots, a_k)$, čija je i-ta komponenta 1 ako je i stvarna klasa primjera, a ostale komponente su 0.
Kvadratna funkcija gubitka penalizira lošu procjenu vjerojatnosti klasa izrazom

$$\sum_{j=1}^{k}(p_j - a_j)^2 \quad . \tag{56}$$

Budući da su sve komponente vektora a jednake 0 osim komponente a_i, suma (56) se može napisati kao

$$1 - 2p_i + \sum_{j=1}^{k} p_j^{\,2} \quad , \tag{57}$$

gdje je i stvarna klasa primjera. Ukupna vrijednost kvadratne funkcije gubitka dobije se sumiranjem vrijednosti za svaki primjer iz skupa primjera S:

$$\sum_{S}\left(1 - 2p_i + \sum_{j=1}^{k} p_j^{\,2}\right) \quad . \tag{58}$$

Ilustrativno je promotriti djelovanje kvadratne funkcije gubitka na skupu podataka kod kojeg je stvarna klasa primjera generirana probabilistički, po nekoj pretpostavljenoj distribuciji klasa $p^* = (p_1^*, \ldots, p_k^*)$. Tada se očekivana vrijednost kvadratne funkcije gubitka na primjeru klasifikacije $a = (a_1, \ldots, a_k)$ može napisati kao

$$E\left[\sum_{j=1}^{k}(p_j - a_j)^2\right] = \sum_{j=1}^{k}\left(E\left[p_j^{\,2}\right] - 2E\left[p_j a_j\right] + E\left[a_j^{\,2}\right]\right) \quad . \tag{59}$$

Budući da a_j uvijek ima vrijednost 1 ili 0, a $E[a_j] = p_j^*$, izraz (59) se može pisati kao

$$\sum_{j=1}^{k}\left(p_j^{\,2} - 2p_j p_j^{\,*} + p_j^{\,*}\right) = \sum_{j=1}^{k}\left[\left(p_j - p_j^{\,*}\right)^2 + p_j^{\,*}\left(1 - p_j^{\,*}\right)\right] \quad . \tag{60}$$

Sada je jasno da je za minimizaciju kvadratne funkcije gubitka najbolje odabrati vektor vjerojatnosti p koji odgovara vektoru stvarnih vjerojatnosti klasa p^*. Tada kvadratni član sumacije iščezava, a ostatak u stvari predstavlja varijancu stvarne distribucije klasa. Prema tome, kvadratna funkcija gubitka će preferirati one modele koji najbolje procjenjuju stvarnu distribuciju vjerojatnosti klasa.

6.1.3.2. INFORMACIJSKA FUNKCIJA GUBITKA

Informacijska funkcija gubitka je vjerojatnosna mjera evaluacije klasifikacijskih modela koja korijene vuče iz teorije informacija. Vrijednost informacijske funkcije gubitka predstavlja količinu informacije u bitovima koja je potrebna da se izrazi stvarna klasa primjera uz poznatu procjenu distribucije vjerojatnosti $(p_1, \ldots, p_k)$ koju

proizvodi klasifikacijski model. Za primjer čija je stvarna klasifikacija $a=(a_1, \ldots, a_k)$, informacijska funkcija gubitka je definirana sa

$$\sum_{j=1}^{k} - a_j \log_2 p_j \quad . \tag{61}$$

Ako je sa i označena stvarna klasa primjera, $a_j=1$ samo za $i=j$, a 0 inače. Stoga se suma iz (61) svodi na

$$- \log_2 p_i \quad . \tag{62}$$

Ukupna vrijednost informacijske funkcije gubitka dobije se sumiranjem po svim primjerima iz skupa primjera S.
Očekivana vrijednost informacijske funkcije gubitka na primjeru iz probabilistički generiranog skupa čija je stvarna distribucija klasa $p^*=(p_1^*, \ldots, p_k^*)$ dana je izrazom

$$\sum_{j=1}^{k} - p_j^* \log_2 p_j \quad . \tag{63}$$

Kao i kod kvadratne funkcije gubitka, ovaj izraz je minimalan za $p_j = p_j^*$, i u tom slučaju izraz (63) postaje entropija stvarne distribucije vjerojatnosti klasa:

$$\sum_{j=1}^{k} - p_j^* \log_2 p_j^* \quad . \tag{64}$$

Pri upotrebi informacijske funkcije gubitka pojavljuje se problem vezan uz ishode kojima klasifikator dodjeli vjerojatnost 0. Naime, ukoliko je stvarna klasa primjera upravo ona kojoj je klasifikator dodijelio vrijednost 0, vrijednost informacijske funkcije gubitka je $+\infty$. Ovaj problem je moguće zaobići manjim korekcijama u definiciji informacijske funkcije gubitka. Međutim, najčešće se tehnika modeliranja modificira na način da rezultirajući modeli ni jednom ishodu ne pridjeljuju vjerojatnost 0 (kao npr. uporaba Laplaceovog estimatora u naivnom Bayesovom klasifikatoru).

Iako obje vjerojatnosne mjere za evaluaciju klasifikacijskih modela preferiraju modele koji točno predviđaju stvarnu distribuciju vjerojatnosti klasa, među njima postoje razlike koje je potrebno uzeti u obzir pri odabiru jedne od njih. Dok informacijska funkcija gubitka ovisi samo o vjerojatnosti koja je pridružena stvarnoj klasi primjera, vrijednost kvadratne funkcije gubitka ovisi i o vjerojatnostima koje su pridružene ostalim klasama. Svaka od preostalih klasa pridonosi vrijednosti funkcije kvadratom svoje vjerojatnosti, pa će kvadratna funkcija gubitka biti najmanja ako se ostatak vjerojatnosti ravnomjerno raspodjeli na preostale klase.
Osim toga, informacijska funkcija gubitka strogo penalizira slučaj kad je stvarnoj klasi primjera dodijeljena mala vjerojatnost. Kada vjerojatnost koja je dodijeljena stvarnoj klasi primjera teži prema 0, vrijednost informacijske funkcije gubitka raste prema beskonačnosti. S druge strane, kvadratna funkcija gubitka je u ovom slučaju blaža, budući da je njena vrijednost na pojedinom primjeru ograničena sa

$$1 + \sum_{j=1}^{k} p_j^2 \quad , \tag{65}$$

što ne može prijeći 2.

6.1.4. Modifikacija pojma greške: troškovi pogrešne klasifikacije

Primarna mjera za evaluaciju rada klasifikacijskog modela je količina grešaka. Međutim, postoje i brojne varijacije na temu pogrešne klasifikacije.

Prirodna alternativa brojanju pogrešno klasificiranih primjera je pridjeljivanje različitih troškova pojedinim tipovima grešaka. Naime, u praksi je vrlo česta situacija da različite vrste grešaka nemaju jednaku važnost. Na primjer, u klasifikacijskim problemima s dvije klase ciljna klasa može biti razmjerno malo zastupljena u odnosu na primjere druge klase. Neuočavanje pozitivnog primjera tada ima bitno veću težinu od pogrešnog izdvajanja negativnog primjera. Stoga je u ovakvim slučajevima trošak lažno negativne klasifikacije u pravilu veći od troška lažno pozitivne klasifikacije.

Trošak pogrešne klasifikacije pridružen nekom tipu greške je broj koji predstavlja "kaznu" za situaciju kada klasifikacija primjera rezultira greškom tog tipa. Uz ovakve postavke, cilj izgradnje klasifikacijskog modela postaje minimizacija troškova pogrešne klasifikacije. Ako je sa n označen ukupan broj klasa, tada matrica grešaka $E = [e_{i,j}]$ ima n^2 elemenata. Troškovi pogrešne klasifikacije se definiraju matricom troškova $C = [c_{i,j}]$ koja na dijagonali ima nule, tj. $c_{i,i} = 0$. Matrica troškova C također ima n^2 elemenata, a element $c_{i,j}$ predstavlja trošak vezan uz tip greške čiju učestalost bilježi element $e_{i,j}$ matrice grešaka. Ukupni trošak se tada računa kao suma troškova svih tipova grešaka:

$$Cost = \sum_{i=1}^{n} \sum_{j=1}^{n} e_{i,j}\, c_{i,j} \quad . \tag{66}$$

Prosječan trošak pogrešne klasifikacije primjera može se dobiti usrednjavanjem ukupnog troška po broju pogrešno klasificiranih primjera.
Brojenje pogrešno klasificiranih primjera predstavlja specijalan slučaj evaluacije promatranjem troškova pogrešne klasifikacije kada je svim tipovima greške pridružen trošak 1.

Podizanjem ili spuštanjem troškova pogrešne klasifikacije za određene vrste grešaka moguće je usmjeravati evaluaciju prema klasifikacijskim modelima određenih karakteristika. Međutim, iako se uvođenjem troškova korigiraju preferencije pri evaluaciji modela, to ne znači da troškovi utječu na sam mehanizam konstrukcije modela. Štoviše, većina metoda za konstrukciju klasifikacijskih modela nije osjetljiva na troškove, tj. rezultirat će istim modelom bez obzira na troškove koji su pridijeljeni pojedinim vrstama grešaka. Na primjer, algoritam konstrukcije stabala ne odlučivanja ne koristi nikakve informacije o eventualnim troškovima vezanim za pojedine vrste grešaka.
Prirodno je očekivati da bi korištenje informacija o troškovima u fazi konstrukcije modela moglo bitno poboljšati klasifikacijske performanse rezultirajućeg modela. Za klasifikacijske probleme sa dvije klase postoji jednostavna intervencija kojom se u fazi konstrukcije modela postiže osjetljivost na troškove pogrešne klasifikacije. Postupak se zasniva na modifikaciji načina formiranja skupa podataka za učenje i ne zahtijeva izmjene u samom algoritmu konstrukcije modela, pa se može koristiti u kombinaciji sa različitim metodama konstrukcije modela.

Osjetljivost na trošak u fazi konstrukcije modela postiže se multipliciranjem primjera određene klase u skupu podataka za učenje. Tako se formira novi skup podataka za učenje koji ima drugačiju zastupljenost klasa od polaznog skupa. Ukoliko metoda konstrukcije modela nastoji smanjiti broj pogrešno klasificiranih primjera, rezultirajući model će izbjegavati greške na klasi čiji su primjeri multiplicirani, budući da se efektivno svaka takva greška višestruko penalizira. Najjednostavniji način multipliciranja primjera u skupu podataka za učenje je generiranje novih primjera koji su identični već postojećim. Puno elegantniji način je pridjeljivanje težinskih vrijednosti primjerima u skupu podataka za učenje, što je moguće samo kod onih metoda konstrukcije modela koje znaju baratati težinama primjera. Težine primjera u skupu podataka za učenje se postavljaju u skladu s relativnim odnosom troškova pridruženih lažno pozitivnim odnosno lažno negativnim ishodima.
Osnova opisanog postupka uvažavanja troškova u fazi izgradnje modela je multipliciranje primjera u skupu podataka za učenje. Stoga je jasno da ovakva adaptacija neće imati efekta u kombinaciji s metodama koje ignoriraju višestruke kopije primjera za učenje. Na primjer, metoda najbližeg susjeda pri klasifikaciji uzima u obzir samo najbliži primjer poznate klasifikacije, pa eventualne višestruke kopije primjera nemaju nikakav utjecaj. S druge strane, već metoda k najbližih susjeda odluku donosi na osnovu više primjera iz skupa za učenje, pa multipliciranje primjera i primjena opisanog postupka imaju smisla.

Osim pojma troška vezanog uz pogrešnu klasifikaciju, proces odlučivanja može uzimati u obzir i očekivani dobitak od ispravne klasifikacije. U takvim primjenama se i evaluacija klasifikacijskog modela može prilagoditi pojmu dobitka vezanog uz ispravnu klasifikaciju. Iznos dobitka u slučaju ispravne klasifikacije može ovisiti o promatranoj klasi. Podaci o očekivanom dobitku ovisno o klasi također se bilježe u modificiranoj matrici troškova $C' = [c'_{i,j}]$, i to korištenjem elemenata dijagonale: element $c_{i,i}$ predstavlja dobitak u slučaju ispravne klasifikacije primjera koji pripada i-toj klasi. Cilj izgradnje klasifikacijskog modela tada postaje maksimizacija dobitka. Ukupni očekivani dobitak (koji može biti i negativan, tj. gubitak) je razlika između dobitaka ostvarenih ispravnom klasifikacijom i gubitaka zbog pogrešne klasifikacije.

Zaključno, uvođenjem pojmova troškova i dobitaka vezanih uz pogrešnu odnosno ispravnu klasifikaciju modificira se pojam greške klasifikacijskog modela. Na taj način se omogućuje evaluacija performansi klasifikacijskog modela u jedinicama tipičnim za domenu problema i stječe podloga za ocjenu isplativosti njegove primjene.

6.2. METODE ZA EVALUACIJU KLASIFIKACIJSKIH MODELA

U osnovi svih mjera za evaluaciju klasifikacijskih modela je pojam greške, odnosno frekvencije grešaka na nekom skupu primjera. Neovisno o tipu promatranih grešaka, empirička frekvencija grešaka može se definirati kao omjer broja pogrešno klasificiranih primjera naspram ukupnog broja promatranih primjera. Iz definicije je jasno je da empirička frekvencija grešaka nekog klasifikatora bitno ovisi o skupu promatranih primjera – mjerenja na različitim skupovima primjera će rezultirati različitim vrijednostima empiričke frekvencija grešaka.

S druge strane, stvarna frekvencija grešaka klasifikacijskog modela je statistički definirana kao frekvencija grešaka na asimptotski velikom broju primjera koji konvergiraju stvarnoj populaciji primjera.
Dakle, u slučaju neograničenog broja primjera, empirička frekvencija grešaka teži ka stvarnoj kako se broj promatranih primjera približava beskonačnosti. Međutim, u realnim situacijama broj primjera je uvijek konačan i relativno malen. Osnovni zadatak metoda za evaluaciju klasifikacijskih modela je ekstrapolacija empiričke frekvencije grešaka izmjerene na konačnom broju primjera na stvarnu, asimptotsku vrijednost.
Postoji više metoda za ocjenu stvarne frekvencije grešaka klasifikacijskog modela, a razlikuju se po pristupu problemu i svojstvima koje pokazuju.

6.2.1. Procjena na osnovu testnog skupa primjera

Svaka metoda za ocjenu stvarne frekvencije grešaka klasifikacijskog modela operira nad nekim konačnim skupom primjera poznate klasifikacije. Na osnovu empirijske frekvencije grešaka mjerene na tom skupu primjera procjenjuje se stvarna frekvencija grešaka. Međutim, svaki skup primjera poznate klasifikacije nije jednako dobar.

Prvi skup primjera koji je na raspolaganju je skup primjera za učenje, na osnovu kojeg je i izgrađen klasifikacijski model. Frekvencija grešaka mjerena na skupu primjera za učenje naziva se *reklasifikacijska frekvencija grešaka*. U slučaju neograničenog broja primjera za učenje koji konvergiraju stvarnoj populaciji primjera, reklasifikacijska frekvencija grešaka izmjerena na dovoljno velikom skupu primjera za učenje bi bila vrlo blizu stvarne frekvencije grešaka. Međutim, u realnim situacijama to nikad nije slučaj. Zbog ograničenog broja primjera u izgradnji klasifikacijskog modela dolazi do pretjerane prilagodbe podacima za učenje (engl. *overfitting*). Praktično, to znači da će rezultirajući model dobro klasificirati primjere iz skupa podataka za učenje, dok će ispravnost klasifikacije novih primjera biti bitno manja. Stoga je reklasifikacijska frekvencija grešaka u pravilu znatno manja od stvarne, tj. daje preoptimističnu ocjenu stvarne frekvencije grašaka. Iako ne predstavlja dobru osnovu za procjenu stvarne frekvencije grešaka, reklasifikacijsku frekvenciju grešaka je ponekad dobro znati. Primjerice, razlika stvarne i reklasifikacijske frekvencije grešaka je dobra mjera stupnja pretjerane prilagodbe modela podacima za učenje.

Stvarni kvalitet klasifikatora određen je sposobnošću ispravne klasifikacije primjera koji nisu bili uključeni u proces stvaranja modela. Stoga je uobičajeno da se u postupku generiranja modela ne koriste svi dostupni primjeri poznate klasifikacije. Umjesto toga, inicijalni skup primjera se dijeli na dva dijela:

- *skup primjera za učenje* koji se koristi za generiranje modela, te
- *testni skup primjera* koji služi za evaluaciju rezultirajućeg modela.

Pri tome je nužno da ova dva skupa budu slučajno odabrana i nezavisna. Preciznije, slučajan odabir podrazumijeva da rezultirajući skupovi primjera moraju biti slučajni uzorci promatrane populacije primjera, a nezavisnost zabranjuje postojanje bilo kakve korelacije između ova dva skupa, osim činjenice da potječu iz iste populacije primjera. Navedeni zahtjevi su vrlo važni pri ocjeni stvarne frekvencije grešaka

klasifikacijskog modela. Ukoliko se ne poštuju, postoji vjerojatnost da će procjena greške biti sistematski kriva, jer je izvedena na osnovu uzorka koji nije reprezentativan, tj. loše predstavlja stvarne karakteristike populacije.
Podjela inicijalnog skupa primjera na skup za učenje i testni skup osigurava da se evaluacija modela odvija nad podacima koji nisu korišteni pri njegovoj izgradnji. Ova je situacija proceduralno identična stvarnom korištenju modela, budući da se radi s novim, dotad neviđenim primjerima. Stoga frekvencija grešaka mjerena na testnom skupu primjera predstavlja procjenu stvarne greške klasifikacijskog modela. Međutim, ovaj pristup ne garantira dobru procjenu na svim distribucijama primjera. Iz tog razloga je potrebno razmotriti i pitanje pouzdanosti procjene frekvencije grešaka korištenjem testnog skupa.

6.2.2. Procjena stvarne greške klasifikacijskog modela

Iz definicije stvarne frekvencije grešaka jasno je da pouzdanost procjene značajno ovisi o broju primjera u testnom skupu, tj. da je procjena pouzdanija što je testni skup brojniji. Promatranje evaluacije klasifikatora kao modela Bernoullijevog eksperimenta daje formalnu vezu broja primjera u testnom skupu i pouzdanosti procjene frekvencije grešaka.
Bernoullijev eksperiment se zasniva na pretpostavci da postoji konstantna vjerojatnost p koja je pridružena uspješnom ishodu u svakom pokušaju. Vjerojatnost da će n pokušaja rezultirati sa k uspješnih ishoda dana je binomnom distribucijom (67):

$$P(\#S{=}k \mid n, p) = \binom{n}{k} p^k (1-p)^{n-k} \quad , \tag{67}$$

gdje je sa $\#S$ označen broj uspješnih ishoda.
Ako n pokušaja rezultira sa k uspješnih ishoda, najbolja procjena vjerojatnosti uspjeha u svakom pokušaju p je dana sa k/n. Dakako, za očekivati je da se stvarna vrijednost vjerojatnosti p razlikuje od procjene k/n. Uobičajen način izražavanja pouzdanosti procjene je određivanje *intervala pouzdanosti*.
Uz zadane n i k, $(1{-}\delta)$ interval pouzdanosti se definira kao najmanji interval $CI(n,k,\delta) \subseteq [0,1]$ takav da $P(p \notin CI(n,k,\delta)) \leq \delta$, tj. vjerojatnost da $p \notin CI(n,k,\delta)$ je manja od δ.
Interval pouzdanosti $CI(n,k,\delta)$ može se izračunati korištenjem binomne distribucije (68):

$$P\left(\left|k/n - p\right| > \varepsilon\right) = \sum_{k:\left|k/n-p\right|>\varepsilon} \binom{n}{k} p^k (1-p)^{n-k} \quad . \tag{68}$$

Korištenjem Chernoffove ograde u obliku (69) može se dobiti konzervativna aproksimacija intervala $CI(n,k,\delta)$ koju je lako računati.

$$P(X > \mu{+}\varepsilon) \;\leq\; e^{-\frac{\varepsilon^2}{4\sigma^2}} \tag{69}$$

Naime, primjena Chernoffove ograde (69) na izraz (68) uz uvrštavanje varijance $\frac{p(1-p)}{n}$ daje nejednakost (70):

$$P\left(\left|k/n - p\right| > \varepsilon\right) \;\leq\; 2e^{-\frac{n\varepsilon^2}{4p(1-p)}} \quad . \tag{70}$$

Konzervativna aproksimacija intervala $CI(n,k,\delta)$ dobije se zahtijevajući da desna strana izraza (70) bude manja od δ. Iz ove nejednakosti jednostavno je vidjeti da interval $\left[k/n-c,\ k/n+c\right]$ predstavlja traženu konzervativnu aproksimaciju $CI(n,k,\delta)$, uz

$$c = \sqrt{\frac{1}{n}\ln\frac{2}{\delta}} \quad . \tag{71}$$

Dakle, uz pouzdanost $(1-\delta)$ se može tvrditi da se frekvencija grešaka mjerena na testnom skupu ne razlikuje od stvarne frekvencije grešaka za više od c. Također se vidi da se s porastom broja primjera n interval pouzdanosti sužava, tj. procjena je pouzdanija.

Osim na ovaj način, interval pouzdanosti se za veliki broj primjera n može aproksimirati korištenjem normalne distribucije. Neka je sa $f = k/n$ označena slučajna varijabla koja bilježi frekvenciju uspješnih ishoda u n pokušaja Bernoullijevog eksperimenta. Za velike n, distribucija ove slučajne varijable se približava normalnoj distribuciji.
Vjerojatnost $P(X \geq z)$ da će normalna slučajna varijabla X srednje vrijednosti 0 i varijance 1 poprimiti vrijednost veću od z može se iščitati iz statističkih tablica za normalnu distribuciju. Zbog simetričnosti normalne distribucije, iz ove vjerojatnosti se lako dobije vjerojatnost $P(-z \leq X \leq z)$ da slučajna varijabla X leži u intervalu oko 0 širine $2z$ (72).

$$P(-z \leq X \leq z) \ = \ 1 - P(X \geq z) - P(X \leq -z) \ = \ 1 - 2\,P(X \geq z) \tag{72}$$

Još je potrebno slučajnu varijablu f srednje vrijednosti p i varijance $\frac{p(1-p)}{n}$ svesti na srednju vrijednost 0 i varijancu 1. To se postiže oduzimanjem srednje vrijednosti i dijeljenjem sa standardnom devijacijom, pa se promatra vjerojatnost (73) čija je vrijednost označena sa v:

$$P\left(-z \leq \frac{f-p}{\sqrt{\frac{p(1-p)}{n}}} \leq z\right) = v \quad . \tag{73}$$

Sada je postupak određivanja intervala pouzdanosti jasan. Za odabranu razinu pouzdanosti v uz uvažavanje (72) u tablicama normalne distribucije pronalazi se odgovarajuća vrijednost parametra z. Uz poznati z, nejednakost iz (73) se može riješiti po p. Dobije se kvadratna jednadžba čija su rješenja dana izrazom (74):

$$p_{1,2} = \frac{f + \frac{z^2}{2n} \pm z\sqrt{\frac{f}{n} - \frac{f^2}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}} \quad . \tag{74}$$

Dva rješenja kvadratne jednadžbe predstavljaju gornju i donju granicu intervala pouzdanosti. Iako je izraz (74) naoko složen, njegova primjena je krajnje jednostavna. Vrijedi još jednom naglasiti da se ovaj način ocjene intervala pouzdanosti zasniva na pretpostavci o normalnoj distribuciji, koja je valjana samo za velike vrijednosti n, tj. za testne skupove s velikim brojem primjera.

Dakle, dovoljan broj primjera u testnom skupu je ključan za pouzdanost procjene frekvencije grešaka. Međutim, za fazu konstrukcije klasifikatora je bitno da i u skupu primjera za učenje bude dovoljan broj primjera. Jasno je da s premalim brojem primjera u skupu za učenje dizajn klasifikacijskog modela ne može biti kvalitetan. Stoga je uobičajeno da se iz inicijalnog skupa klasificiranih primjera veći dio odvoji u skup za učenje. U praksi je čest slučaj da se 2/3 ukupnog broja primjera izdvoji u skup primjera za učenje, a 1/3 u testni skup primjera. Ako je inicijalni skup primjera dovoljno brojan, i faza konstrukcije i faza evaluacije klasifikacijskog modela raspolagat će s dovoljnim brojem primjera.
Međutim, u slučaju ograničenog broja primjera bolji rezultati se postižu korištenjem metoda evaluacije koje su primjerenije takvoj situaciji. Takve metode se uglavnom baziraju na višestrukom ponavljanju postupka procjene greške na testnom skupu, uz adekvatno particioniranje inicijalnog skupa primjera.

6.2.3. Metoda unakrsne validacije

Nije rijedak slučaj da je za cijeli proces inteligentne analize podataka dostupan samo razmjerno mali broj unaprijed klasificiranih primjera. Na primjer, medicinske studije se vrlo često preliminarno provode na vrlo malom broju pacijenata. U takvim situacijama su proces konstrukcije klasifikacijskog modela, kao i njegova evaluacija iznimno otežani. Zbog toga je ključno da se inicijalni skup klasificiranih primjera što bolje iskoristi i pri konstrukciji i pri evaluaciji modela.
Jedan od uzroka moguće nepreciznosti metode evaluacije korištenjem testnog skupa je problem oslanjanja na jednu, moguće nekarakterističnu particiju skupa primjera za učenje, odnosno testiranje modela. Iako je slučajan odabir jedan od osnovnih zahtjeva pri formiranju skupova podataka za učenje i testiranje, mogućnost da odabrani skupovi podataka ne predstavljaju reprezentativan uzorak populacije raste kako se ukupan broj raspoloživih primjera smanjuje. Zbog specifičnosti u skupovima podataka za učenje odnosno testiranje koji nisu svojstvo populacije nego defekt uzrokovan odabirom skupova, evaluacija korištenjem testnog skupa može rezultirati nepreciznom procjenom frekvencije grešaka.
Osnovni način izbjegavanja ovakvih anomalija je višestruko ponavljanje procesa evaluacije na testnom skupu koristeći različite slučajno odabrane skupove za učenje i testiranje, te usrednjavanje dobivenih procjena frekvencije grešaka. Metoda unakrsne validacije se zasniva na ovom principu, uz svojevrsnu zamjenu skupa podataka za učenje i testnog skupa u svakoj iteraciji [35].
U postupku k-struke unakrsne validacije najprije se inicijalni skup primjera po načelu slučajnog odabira razdijeli u k međusobno različitih particija približno iste veličine. Sam postupak je iterativan s tim da se u jednoj iteraciji k–1 particija koristi kao skup za učenje, a konstruirani model se testira na preostaloj particiji koja predstavlja testni skup. Postupak iterira k puta, tako da je svaka od particija po jednom u ulozi testnog skupa primjera. Prosječna frekvencija grešaka preko svih k iteracija postupka predstavlja ocjenu stvarne frekvencije grešaka klasifikacijskog modela.
Prilikom podjele inicijalnog skupa primjera u k particija često se postupak slučajnog odabira modificira na način da osigura približno jednaku zastupljenost klasa u svakoj od particija. Ovaj postupak naziva se *stratifikacija*, a osnovna mu je svrha poboljšanje reprezentativnosti svake od particija. Iako se na ovaj način postiže da je

u svakoj iteraciji zastupljenost klasa u skupu za učenje i testiranje približno jednaka zastupljenosti u inicijalnom skupu primjera, stratifikacija ni u kom smislu ne osigurava reprezentativnost skupa za učenje ili testiranje. Međutim, empirički rezultati govore da stratifikacija blago poboljšava rezultate evaluacije, posebice na manjim skupovima primjera.
O broju particija, odnosno iteracija unakrsne validacije direktno ovisi računalna složenost evaluacije klasifikacijskog modela ovom metodom, budući da svaka iteracija uključuje zasebno konstruiranje i testiranje modela. Iako to nije pravilo, u praksi se najčešće koristi stratificirana 10-struka unakrsna validacija. Pokazala se kao dovoljno točna, a nije računalno prezahtjevna.
Prednosti unakrsne validacije kao metode za evaluaciju klasifikacijskih modela ogledaju se u činjenici da su svi dostupni primjeri iskorišteni za testiranje, a i konstrukcija modela se u svakoj iteraciji koristi velikom većinom dostupnih primjera. Osim toga, ima umjerene računske zahtjeve u odnosu na neke druge iterativne metode evaluacije, ali ipak primjetno veće od klasične evaluacije korištenjem testnog skupa.
Međutim, iako usrednjavanje greške po više testnih skupova u velikoj mjeri ublažava ovisnost o odabiru skupa primjera za učenje i testiranje, određena varijabilnost ipak postoji. Njen uzrok leži u načelu slučajnosti pri formiranju particija na početku procesa evaluacije. Zbog toga je za očekivati da će metoda unakrsne validacije uz istu tehniku modeliranja i skup dostupnih podataka, ali uz različito particioniranje tog skupa dati ponešto drugačiju procjenu frekvencije grešaka. Ublažavanje ovog efekta može se postići ponavljanjem cijelog postupka unakrsne validacije više puta, te usrednjavanjem rezultata.

6.2.4. Metoda izostavljanja jednog primjera

Metoda evaluacije klasifikacijskih modela izostavljanjem jednog primjera je specijalan slučaj metode unakrsne validacije. Ako je sa n označen ukupan broj inicijalno dostupnih primjera, izostavljanje jednog primjera je jednostavno n-struka unakrsna validacija. Dakle, u svakoj od n iteracija metode izostavljanja jednog primjera klasifikacijski model se konstruira na osnovu $n-1$ primjera, a testira na jednom preostalom primjeru. Dakako, konačna procjena frekvencije grešaka je usrednjenje grešaka po svim iteracijama postupka.
Dva su razloga privlačnosti ove metode:

- maksimalna iskoristivost inicijalnog skupa primjera
  Svaki pojedini primjer koristi se kao testni primjer, a u svakoj iteraciji klasifikacijski model se gradi na maksimalnom mogućem skupu podataka za učenje.
- postupak je deterministčkog karaktera
  Budući da se svaka particija sastoji od samo jednog primjera, izbjegnut je proces slučajnog uzorkovanja. To znači da će evaluacija metodom izostavljanja jednog primjera uz istu tehniku modeliranja i isti skup dostupnih primjera uvijek rezultirati istom procjenom frekvencije grešaka.

Zbog ovih svojstava, metoda evaluacije izostavljanjem jednog primjera postiže iznimno dobre rezultate u procjeni stvarne frekvencije grešaka i rijetko kada sistematski odstupa od nje. Nedostatak ove metode je velika računalna složenost postupka, budući da se klasifikacijski model konstruira i testira n puta. Stoga je

izostavljanje jednog primjera preferirana metoda za manje skupove primjera, dok za veće skupove zna biti preskupa.

6.2.5. Bootstrap metoda

Bootstrap metoda evaluacije klasifikacijskih modela je, kao i metoda izostavljanja jednog primjera, namijenjena uglavnom problemima s malim brojem dostupnih primjera [16]. Iako metoda izostavljanja jednog primjera daje vrlo dobru procjenu frekvencije grešaka klasifikatora, postoje neki nedostaci koji loše utječu na konačni rezultat. To se prije svega odnosi na veliku varijancu procjene greške na malom broju primjera, koja dominira u ukupnoj nepreciznosti ove metode. Primjenom bootstrap metoda evaluacije smanjuje se efekt velike varijance.
Bootstrap metoda se pri formiranju skupa podataka za učenje bazira na uzorkovanju s ponavljanjem. To znači da se u skupu podataka za učenje isti primjer iz inicijalnog skupa primjera može pojaviti više puta, tj. dozvoljeno je multipliciranje primjera iz inicijalnog skupa.
Neka je sa n opet označen ukupan broj inicijalno dostupnih primjera. Kod bootstrap metode skup primjera za učenje također sadrži n elemenata, a nastaje slučajnim odabirom s ponavljanjem. Na taj način će u skupu za učenje (gotovo sigurno) doći do ponavljanja nekih primjera iz inicijalnog skupa. Jednako tako će postojati primjeri iz inicijalnog skupa koji uopće nisu zastupljeni u skupu za učenje – oni formiraju testni skup primjera. Svaka iteracija bootstrap metode uključuje ovakvo formiranje skupa primjera za učenje i testiranje, kao i konstrukciju i testiranje klasifikacijskog modela. Broj iteracija postupka nije čvrsto određen, a u praksi se obično radi o nekoliko stotina ponavljanja. Srednja vrijednost greške po svim iteracijama naziva se *e0* procjenom greške.
Za umjereno velike skupove podataka *e0* procjena daje pesimističnu procjenu stvarne greške. Uzrok ovog sistematskog odstupanja leži u činjenici da se klasifikacijski model izgrađuje na osnovu relativno malog broja različitih primjera. Naime, prosječan broj međusobno različitih primjera u skupu za učenje iznosi 0.632 ukupnog broja primjera. Vjerojatnosno opravdanje spomenutog udjela je jednostavno: prilikom odabira svakog pojedinog primjera u skup za učenje, vjerojatnost da određeni primjer iz inicijalnog skupa neće biti odabran je $1-\frac{1}{n}$. Budući da se u skup za učenje bira n primjera, vjerojatnost da određeni primjer neće biti odabran u n pokušaja iznosi

$$\left(1-\frac{1}{n}\right)^n \approx e^{-1} \approx 0.368 \quad . \tag{75}$$

Naravno, aproksimacija sa e^{-1} u (75) vrijedi za dovoljno velike n. Stoga će testni skup sadržavati 0.368 ukupnog broja primjera za dovoljno velike inicijalne skupove primjera. Ostatak od 0.632 ukupnog broja primjera daje broj različitih primjera u skupu za učenje.
Pesimističnost *e0* procjene nastoji umanjiti *0.632B*, druga procjena greške koja se također koristi u bootstrap metodi evaluacije. Izraz (76) definira *0.632B* procjenu greške

$$0.632B = 0.632\ e0 + 0.368\ e_r \quad , \tag{76}$$

gdje e_r predstavlja reklasifikacijsku frekvenciju grešaka. Dakle, *0.632B* procjena je linearna kombinacija pesimistične *e0* procjene i optimistične reklasifikacijske frekvencije grešaka.
I *e0* i *0.632B* su procjene s malom varijancom. Iako je generalno gledajući *e0* pesimistična procjena greške, ona daje vrlo dobru ocjenu kada je stvarna greška klasifikacijskog modela relativno visoka. Sa povećanjem broja primjera *0.632B* procjena postaje preoptimistična, no za male skupove primjera s relativno niskom greškom modela daje vrlo dobre rezultate.
Zbog većeg broja iteracija bootstrap metoda je računski prilično zahtjevna, pa se uglavnom primjenjuje na probleme s manjim brojem dostupnih primjera. Iako ne pati od problema velike varijance, bootstrap metoda nije uvijek superiorna metodi izostavljanja jednog primjera na manjim skupovima primjera. No, mala *e0* procjena greške je bolji indikator dobrog modela od procjene metodom izostavljanja jednog primjera.

Konačno, tablica na slici 21 uspoređuje neke osnovne karakteristike spomenutih metoda za ocjenu greške klasifikacijskog modela.

	procjena na osnovu testnog skupa	*k*-struka unakrsna validacija	metoda izostavljanja jednog primjera	bootstrap metoda
broj primjera u skupu za učenje	*j* (najčešće 2/3 *n*)	(*k*–1) *n*/*k*	*n*–1	*n*, *j* različitih (*j*≈0.632 *n*)
broj primjera u testnom skupu	*n*–*j* (najčešće 1/3 *n*)	*n*/*k*	1	*n*–*j* (≈0.368 *n*)
broj iteracija	1	*k*	*n*	nekoliko stotina

Slika 21: Neka svojstva metoda za ocjenu greške klasifikacijskog modela

7. Inteligentna analiza podataka u primjeni

U prethodnim poglavljima opisane su osnovne faze postupka inteligentne analize podataka, te je napravljen detaljniji pregled tehnika koje se koriste u pojedinim fazama. Ovo poglavlje ilustrira oblikovanje procesa analize podataka odabirom adekvatnih tehnika u svakoj njegovoj fazi, sukladno svojstvima ciljne skupine problema. Prikazana je i primjena tako osmišljenog sustava na konkretnom problemu iz poslovne prakse. Analizirani problem dolazi iz osiguratelјne branše, a odnosi se na profiliranje korisnika usluga.

Na osnovu provedenih analiza i eksperimenata izrađeno je aplikativno rješenje koje implementira razrađeni postupak inteligentne analize podataka za ciljnu skupinu problema. Aplikacija je izvedena u obliku integriranog okruženja za klasifikacijsku analizu podataka koje podržava sve faze klasičnog procesa inteligentne analize podataka, uključujući i ocjenu performansi izgrađenih modela [55]. Sustav je primarno namijenjen stvarnim poslovnim problemima koje karakterizira visok stupanj neizvjesnosti i šuma u podacima. Izražena nedeterministička obilježja poslovnih problema rezultirala su odgovarajućim prilagodbama i specifičnim rješenjima. Arhitektura sustava na nov način nadovezuje niz dorađenih i modificiranih algoritama inteligentne analize podataka, u cilju povećanja efikasnosti klasifikacijskih postupaka i izvedenih opisa modela. Naredni odlomci opisuju građu i bitne komponente novorazvijenog sustava za inteligentnu analizu podataka. Sustav je implementiran u Java programskom jeziku, a koristi SQL upite za direktan pristup bazama podataka. Analitički podsustav upotrebljava varijaciju naivnog Bayesovog klasifikatora potpomognutu algoritmom odabira atributa za uklanjanje redundantnih atributa. Ciljni pojam opisuje se stablom odlučivanja koje konstruira standardni C4.5 algoritam. Integrirani vizualni alat za ocjenu klasifikacijskih performansi omogućava znatno brže dosezanje konačnog modela, te omogućuje interaktivno upravljanje procesom indukcije znanja. Prediktivne sposobnosti opisanog sustava ilustrirane su na tipičnom primjeru ciljne skupine problema.

7.1. OPIS CILJNE SKUPINE PROBLEMA

Evidentno je da postoji veći broj bitno različitih tehnika inteligentne analize podataka. Razlike sežu od pretpostavki na ulazne podatke i svojstva rješavanih problema, preko primijenjenih algoritama strojnog učenja, pa do načina prikaza otkrivenog znanja. S obzirom na značajne razlike, niti jedna od njih ne može postići najbolje rezultate u svim mogućim primjenama. Budući da različite tehnike inteligentne analize podataka pogoduju različitim vrstama problema, temeljne karakteristike ciljne skupine problema bitno utječu na arhitekturu sustava za inteligentnu analizu podataka. U ovom primjeru, poslovna praksa osiguravajućih društava bila je izvor motivacije za odabir ciljne skupine problema. Fokus je stavljen na specifičnu skupina problema iz šire problematike izgradnje profila korisnika usluga. Prvenstveno su promatrani problemi kod kojih je potrebno izolirati razmjerno malu skupinu zanimljivih korisnika.

Problemi koji podrazumijevaju modeliranje ponašanja ljudi neizbježno uključuju element neizvjesnosti. Stoga se prirodno nameće korištenje probabilističkih umjesto

kategoričkih modela za takve probleme. Primjerice, nerealno je očekivati postojanje modela koji bi točno predvidio hoće li potencijalni klijent reagirati na promotivne letke primljene poštom. Prikladnije je ovaj problem poimati kao Bernoullijev eksperiment, gdje je neka vjerojatnost uspjeha pridružena svakom pokušaju. Ovakva perspektiva sugerira model koji rezultira procjenom vjerojatnosti da će određena osoba reagirati na promotivnu ponudu. Osim profiliranja korisnika, modeli mnogih drugih poslovnih problema nisu deterministčkog, već probabilističkog karaktera. Razlog može biti sama priroda problema, ali i nedostatne ili ograničene mogućnosti razumijevanja problema. Stoga je u nekim poslovnim područjima probabilistički pristup jedina realna opcija u modeliranju istraživanih pojava.
U skladu s dva moguća ishoda Bernoullijevog eksperimenta, razvijeno okruženje za analizu podataka je prvenstveno projektirano za probleme klasifikacije i opisa dviju međusobno isključivih klasa. Osim toga, ciljnu skupinu problema karakteriziraju izražena nedeterministička obilježja. Primjerice, čak i kod podataka za učenje često nije moguće razlučiti klasifikaciju nekih primjera. Razlog tomu je postojanje primjera s istim vrijednostima svih atributa, ali s različitom klasifikacijom. Međutim, ovo nije svojstvo odabranog skupa atributa kojim su primjeri opisani, već inherentna karakteristika samog problema: iz prirode problema jasno je da ne postoji skup atributa koji bi mogao jednoznačno razdvojiti različite klase. Na primjer, bez obzira koliko detaljno se pratila svojstva klijenata, lako je zamisliti situaciju u kojoj dva klijenta sa istim zabilježenim svojstvima različito reagiraju na određenu promotivnu ponudu. Naravno, nedeterministička obilježja mogu imati i drugi poslovni problemi nevezani s profiliranjem korisnika.

Nadalje, specifičnost izgrađenog sustava je usmjerenost na probleme koji u pravilu uključuju izdvajanje razmjerno malog broja zanimljivih entiteta (obično ne više od 5 do 10 posto ukupnog broja primjera). U svjetlu već spominjanog primjera profiliranja korisnika, realno će samo mali dio potencijalnih klijenata koji prime promotivni letak zaista i kupiti oglašeni proizvod ili uslugu. Općenito, bitna značajka entiteta koje je cilj izdvojiti je grupiranje na probabilističkoj osnovi. To znači da za svaki entitet postoji određena vjerojatnost pripadanja ciljnoj skupini. Entiteti sa višim vjerojatnostima su zanimljivi, te ih pokušavamo odijeliti od entiteta sa nižim vjerojatnostima. Drugim riječima, kriterij za odvajanje zanimljivih entiteta je procijenjena vrijednost vjerojatnosti pripadanja ciljnoj skupini.
Međutim, kod većine stvarnih poslovnih problema unaprijed je jasno da su i najviše vjerojatnosti s kojima se barata iznosom relativno male (primjerice, vrijednosti preko 0.3 su vrlo rijetke). Ovo implicira da treba promatrati samo relativne odnose među vjerojatnostima. Bilo koji proces odlučivanja koji bi se zasnivao na apsolutnim vrijednostima procijenjenih vjerojatnosti (kao npr. pravilo 50 postotnog praga) ne bi bio primjeren tipu rješavanih problema.

Ilustrativan primjer problema ovog tipa dolazi iz osiguratelјne djelatnosti: procjena koji osiguranici će postaviti odštetne zahtjeve je važan čimbenik uspjeha poslovanja. Da bi što bolje poslovalo, osiguravajuće društvo mora zadovoljiti dva naoko kontradiktorna zahtjeva. Prvo, ukupni iznos premije prikupljen u nekoj godini treba biti visok barem kao ukupni iznos isplaćenih odšteta. Ovo sugerira da su više premije povoljnije za osiguravajuće društvo. S druge strane, premije koje zaračunava društvo ne bi trebale biti veće od onih kod konkurentskih društava. U ovom slučaju,

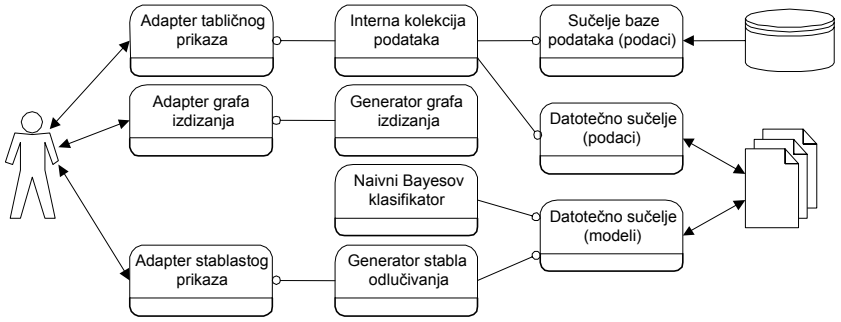
povoljnije su niže premije. Stoga, osiguravajuće društvo treba identificirati skupine osiguranika s visokom vjerojatnošću postavljanja odštetnog zahtjeva u nekoj godini. Ovakva saznanja su osnova profiliranja visine premije za neke vrste osiguranja. Što su precizniji opisi rizičnih osiguranika, osiguravajuće društvo može uspješnije konkurirati na tržištu. Za osiguravajuća društva su profili osiguranika prema stupnju pridružene rizičnosti očito vrijedno oružje u borbi s konkurencijom.
Ovaj problem odgovara prije opisanom tipu poslovnih problema. Broj osiguranika koji zaista postave odštetni zahtjev je relativno malen (oko 6% ukupnog broja osiguranika, ovisno o vrsti osiguranja). Samo postavljanje odštetnog zahtjeva je događaj s vrlo izraženim nedeterminističkim obilježjima. Uobičajeno je da postoje osiguranici jednakih karakteristika, od kojih neki postave jedan ili više odštetnih zahtjeva, dok drugi uopće ne postave zahtjev. Ova pojava je uzrokovana prirodom problema: osoba se može osigurati samo od rizika vezanih uz neizvjesne buduće događaje koji su neovisni o volji osiguranika ili njegovim namjerama. Nadalje, svaki osiguranik potencijalno može postaviti odštetni zahtjev, ali je a priori vjerojatnost da će točno određeni osiguranik to učiniti relativno mala (to je osnova osigurateljnog poslovanja). Razvijeno okruženje za klasifikacijsku analizu podataka prilagođeno je poslovnim problemima ovakvih karakteristika, a značaj samog problema i raspoloživost podataka koje osiguravajuća društva održavaju čine ga primjerenim izazovom za tehnike inteligentne analize podataka.

7.2. ARHITEKTURA SUSTAVA

Standardni proces inteligentne analize podataka odvija se u iteracijama koje se sastoje od nekoliko uzastopnih faza: dohvat i priprema podataka, primjena algoritma inteligentne analize podataka, prezentacija induciranog znanja, ocjena performansi sustava, te eksploatacija dobivenih rezultata. Izgrađeno integrirano okruženje za inteligentnu analizu podataka podržava sve faze procesa na konzistentan način, omogućujući nesmetane prijelaze među fazama. Sustav uključuje i više automatiziranih postupaka (npr. automatski odabir atributa, procjena distribucije vjerojatnosti), izbjegavajući tako zamaranje korisnika nepotrebnim detaljima. Interaktivnost grafičkog korisničkog sučelja za upravljanje procesom analize podataka daje korisniku više kontrole nad tijekom cijelog procesa. Razmatranjem više različitih scenarija korisnik preuzima aktivnu ulogu u ocjeni performansi sustava, te tako pridonosi efikasnosti konačnog modela.

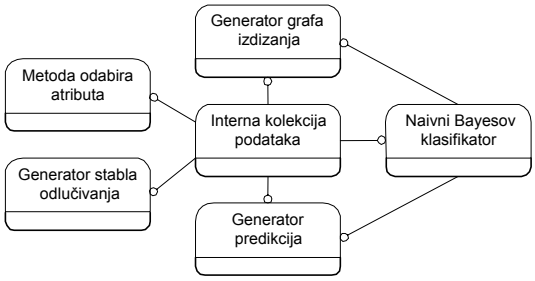
Cijelo aplikacijsko okruženje je implementirano korištenjem Java programskog jezika, koji osigurava platformsku neovisnost. Interna arhitektura sustava sastoji se od dva različita sloja: *jezgre sustava* koja implementira sve postupke analize podataka, te *prezentacijskog sloja* koji je zadužen za interakciju s korisnikom i ulazno-izlazne operacije. Osnovne klase prezentacijskog sloja i njihovi odnosi s odgovarajućim klasama jezgre prikazane su na slici 22.
Skupove podataka za učenje, testiranje i predikciju sustav pohranjuje i rukuje njima koristeći internu strukturu kolekcije podataka. Transformirani u ovaj oblik, podaci se mogu prikazati korisniku u obliku tablice, gdje ih može pregledavati i mijenjati, ili analizirati rezultate predikcije. Izvor podataka može biti baza podataka ili datotečni sustav, a modificirani podaci se mogu pohraniti u datotečni sustav. Performanse induciranog Bayesovog klasifikatora se procjenjuju metodom grafa izdizanja (engl.

lift chart). Odgovarajući adapter vizualizira rezultate ocjene performansi u obliku interaktivnog dvodimenzionalnog grafa pomoću kojeg korisnik izdvaja zanimljive entitete istražujući i uspoređujući više različitih scenarija. Generator stabla odlučivanja koristi se za izradu opisa ciljnog pojma. Rezultat se prikazuje u obliku stablaste strukture s mogućnošću otvaranja i zatvaranja pojedinih grana. Da bi se izbjeglo ponovno generiranje klasifikatora svaki put kad se koriste, i naivni Bayesov klasifikator i stablo odlučivanja mogu se pospremiti u datotečni sustava radi kasnije uporabe.



Slika 22: Odnos klasa jezgre s klasama prezentacijskog sloja

Klase jezgre sustava čine analitičku osnovu klasifikacijskog okruženja. One su zadužene za filtriranje ulaznih podataka, konstruiranje probabilističkih i deskriptivnih modela podataka, ocjenu performansi izgrađenih modela i klasificiranje novih podataka primjenom induciranih modela. Slika 23 prikazuje osnovne klase jezgre i njihove odnose.

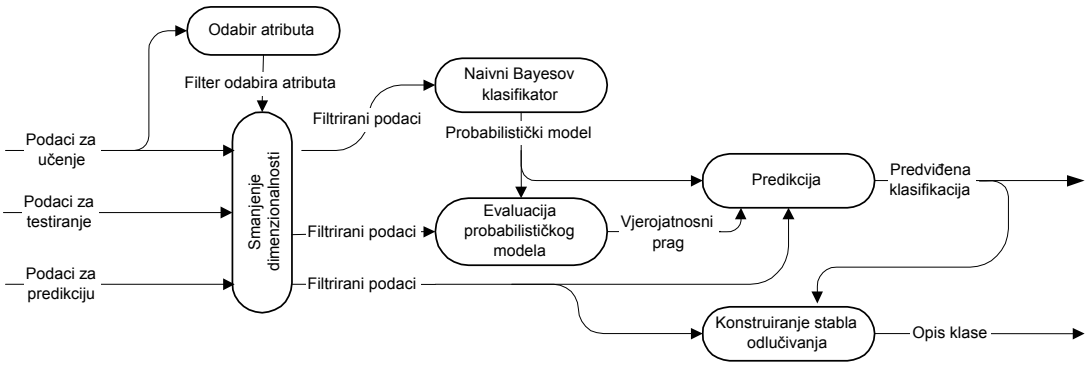


Slika 23: Odnosi među klasama jezgre sustava

Sve klase za analizu podataka kao izvor podataka koriste internu kolekciju podataka. Ta je klasa nezaobilazna u dohvatu, obradi i pohrani podataka u svim analitičkim procesima. Klasa odabira atributa proizvodi filter za odabir atributa analizirajući različite podskupove atributa iz kolekcije podataka za učenje. Konstruirani filter se primjenjuje na kolekciju podataka, i proizvodi novu, filtriranu kolekciju podataka. Naivni Bayesov klasifikator generira probabilistički klasifikacijski model koristeći kolekciju podataka za učenje. Kod primjene izvedenog modela, klasifikator kao izlaz daje procjenu raspodjele vjerojatnosti za svaki primjer iz kolekcije podataka. U procesu ocjene performansi Bayesovog klasifikatora metodom grafa izdizanja koristi se kolekcija podataka poznate klasifikacije, tako što se uspoređuju procijenjene

vjerojatnosti koje proizvodi klasifikator sa stvarnom klasifikacijom podataka. Generator predikcija klasificira kolekciju podataka nepoznate klasifikacije primjenjujući inducirani klasifikator, proizvodeći tako kolekciju podataka sa procijenjenom klasifikacijom. Generator stabla odlučivanja upotrebljava ovu kolekciju za izradu opisa ciljne klase. Stablo odlučivanja se ne koristi u predikcijske svrhe.

Dijagram toka podataka zorno ilustrira način komunikacije klasa jezgre sustava. Dijagram na slici 24 prikazuje tijek transformacije sirovih podataka u modele za opis i predviđanje analiziranih pojmova.



Slika 24. Skica dekompozicije procesa i tijeka podataka

Kopija podataka za učenje najprije se analizira algoritmom odabira atributa. Algoritam procjenjuje valjanost skupa atributa uzimajući u obzir stupanj redundancije među njima, ali i njihovu prediktivnu vrijednost. Rezultat ove analize je filter za selekciju atributa koji se kasnije primjenjuje na podatke za učenje, podatke za testiranje i podatke za izradu predikcija. Naivni Bayesov algoritam klasifikacije konstruira probabilistički klasifikacijski model na osnovu filtriranih podataka za učenje. Osjetljivost klasifikacijskog algoritma na postojanje redundantnih atributa kompenzira se eliminiranjem redundantnih atributa u fazi pripreme podataka, što poboljšava točnost klasifikacije. Performanse induciranog probabilističkog modela se ocjenjuju uporabom vizualnog alata za ocjenu performansi. Iterativnim ponavljanjem ovog postupka uz variranje ulaznih parametara poboljšavaju se performanse inicijalnog modela. Nakon postizanja zadovoljavajućih rezultata bira se vjerojatnosni prag pomoću kojeg se odjeljuju zanimljivi entiteti. Koristeći ovaj prag i inducirani probabilistički model predviđa se klasifikacija novih, dotad neviđenih primjera. U konačnici se izdvojeni skup zanimljivih entiteta može opisati u obliku stabla odlučivanja, pružajući vrijedan uvid u strukturu induciranog znanja.

Odjeljci koji slijede daju detaljan opis sastavnih dijelova sustava za inteligentnu analizu podataka.

7.2.1. Pristup bazama podataka i dohvat podataka

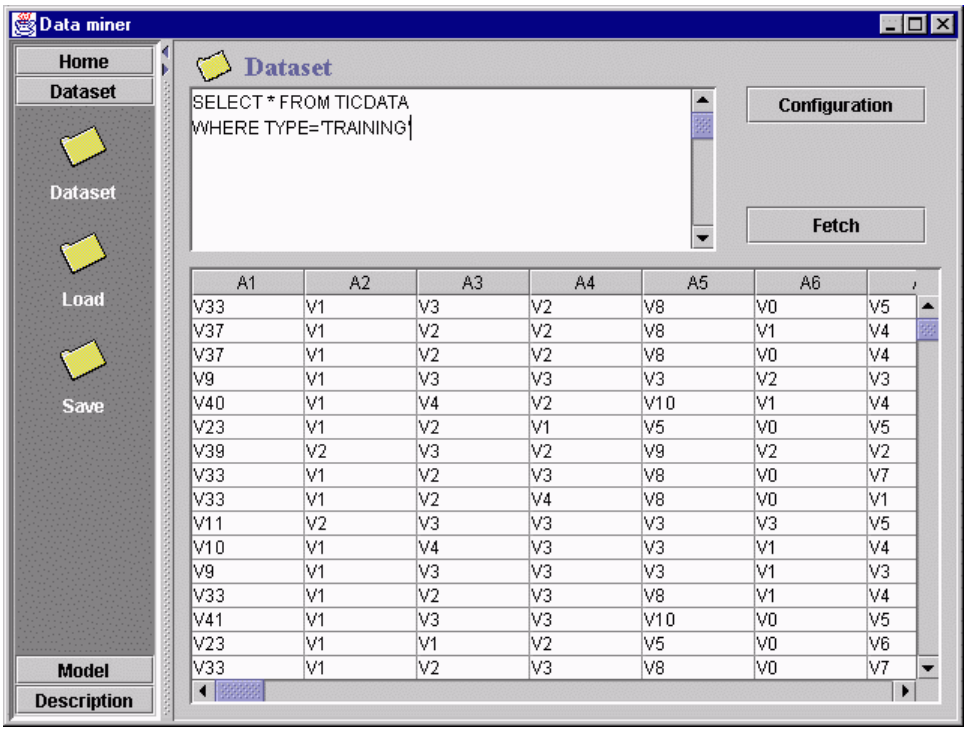
Prvi korak u procesu inteligentne analize podataka je određivanje cilja same analize. Nakon utvrđivanja cilja, prikupljaju se dostupni podaci relevantni za promatrani

problem radi podvrgavanja procesu analize. Prikupljanje relevantnih podataka kojima se mogu izraziti pravilnosti koje se istražuju može imati ključan utjecaj na uspjeh cijelog procesa analize podataka. Adekvatan skup podataka za učenje je glavni preduvjet za sve tehnike inteligentne analize podataka.

U poslovnim primjenama, veliki dio analiza koje se provode odnosi se na transakcijske podatke. U pravilu su transakcijski podaci poslovnih sustava pohranjeni u bazama podataka. Mnoge veće tvrtke uspostavljaju i održavaju skladišta podataka koja ujedinjuju podatke iz različitih izvora unutar organizacije. Kao jedinstvena točka pristupa integriranim poslovnim podacima, skladišta podataka predstavljaju iznimno koristan resurs u prikupljanju relevantnih podataka za proces inteligentne analize.
Smatrajući baze podataka i skladišta podataka primarnim izvorom podataka, prirodno je omogućiti sustavu za inteligentnu analizu direktan pristup ovim izvorima. Današnji standard za pristup relacijskim bazama podataka je SQL (engl. *Structured Query Language*). Iako se XML (od engl. *eXtended Markup Language*) pojavljuje kao nadolazeći standard za razmjenu podataka među aplikacijama, mnogi postojeći poslovni sustavi su zasnovani na bazama podataka koje ne podržavaju XML. Postoji još jedna pogodnost pri korištenju SQL-a za dohvat podataka u sustavima inteligentne analize podataka. Gotovo sve tehnike inteligentne analize podataka operiraju pod *pretpostavkom univerzalne relacije* – očekuju da su svi potrebni podaci sadržani u samo jednoj relaciji. Iako su podaci u bazama podataka raspoređeni u mnogo različitih relacija, rezultat jednog SQL upita je uvijek jedna relacija, bez obzira koliko baznih relacija upit uključuje. Dakle, prikladan način dohvata podataka u sustavu inteligentne analize podataka je prosljeđivanje SQL upita bazi podataka, te prihvat rezultata upita kojeg baza pripremi.
U razvijenom klasifikacijskom okruženju, pristup bazama podataka je ostvaren uporabom JDBC (engl. *Java Database Connectivity*). Uz platformsku neovisnost koju omogućava Java programski jezik, JDBC osigurava i neovisnost o bazama podataka. Da bi pristupio različitim bazama podataka, sustav za analizu podataka upotrebljava odgovarajući JDBC upravljački program kojeg dobavlja proizvođač baze podataka. Kroz upravljački program sustav uspostavlja vezu s bazom podataka, omogućujući korisniku dohvat podataka iz baze putem SQL upita. Rezultat upita se pretvara u internu kolekciju podataka koja je pogodna za transformacije kakve koriste algoritmi inteligentne analize podataka. U klasifikacijskom okruženju također postoji mogućnost učitavanja podatkovnih datoteka s datotečnog sustava i pretvaranja u internu kolekciju podataka (npr. datoteka u *csv* formatu, gdje su vrijednosti podataka odvojene zarezom). Jednom kad su podaci pretvoreni u internu kolekciju, prikazuju se na ekranu u tabelarnom prikazu, gdje se mogu pregledavati, mijenjati ili pospremiti u datotečni sustav za kasnije korištenje. Grafičko korisničko sučelje za dohvat i pregled podataka prikazano je na slici 25.

Postoje dva osnovna tipa atributa koji se tretiraju različito u većini postupaka inteligentne analize podataka: *nominalni* i *numerički* atributi. Nominalni atributi poprimaju vrijednosti iz predefiniranog, konačnog skupa mogućnosti, bez predmnijevanja bilo kakvih relacija među različitim vrijednostima. Numerički atributi se izražavaju brojevima (cjelobrojnim ili realnim), uz postojanje svih standardnih relacija među brojevima. Pri dohvatu podataka iz baze podataka i

pretvaranju u internu kolekciju, klasifikacijsko okruženje utvrđuje tipove atributa provjerom tipa podataka koji se u bazi koristi za njihovu pohranu. Zakovni podaci se pretvaraju u nominalne atribute, a brojčani u numeričke atribute. Usprkos tome, postoje situacije kad ovakav pristup ne uspijeva odrediti pravu narav podataka. Primjerice, identifikacijske brojeve treba tretirati kao nominalni atribut, iako se u bazama podataka oni često bilježe kao brojčani podatak. Ovaj problem se lako može zaobići uporabom SQL funkcija za promjenu tipa podatka pri konstrukciji upita za dohvat podataka.



Slika 25. Korisničko sučelje za dohvat i pregled podataka

7.2.2. Priprema podataka

Cilj postupaka pripreme podataka je poboljšanje kvalitete podataka za iduću fazu procesa inteligentne analize: modeliranja pravilnosti u podacima. Stoga se pri odabiru postupaka za pripremu podataka treba rukovoditi poznatim svojstvima algoritma strojnog učenja koji se koristi u fazi modeliranja. Razvijeni sustav analize podataka upotrebljava varijaciju naivnog Bayesovog algoritma za indukciju probabilističkog modela. Bayesov klasifikator pojednostavljuje problem klasifikacije korištenjem jedne bitne pretpostavke: pretpostavlja da su svi prognostički atributi međusobno nezavisni uz uvjet poznate klasifikacije. Postojanje visoko koreliranih atributa u skupu podataka za učenje može znatno deformirati proces analize podataka. Kao krajnost, istaknut je primjer dodavanja novog atributa s vrijednostima identičnim nekom postojećem atributu u skupu podataka za učenje. Učinak tog atributa na rezultat klasifikacije bio bi znatno uvećan: sve njegove vjerojatnosti bile bi kvadrirane, dajući mu nerealno visok utjecaj na konačnu odluku.

U stvarnim poslovnim problemima ovisnosti među različitim atributima su uobičajene. U svrhu poboljšanja performansi na takvim skupovima podataka, klasifikacijsko okruženje uključuje automatizirani postupak odabira atributa koji uklanja redundantne atribute.

Odabir jednog od dva osnovna pristupa odabiru atributa također je uvjetovan svojstvima ciljnih problema. Iako se općenito drži da algoritmi omotača rezultiraju boljim podskupovima atributa, oni su prilično spori pri izvođenju jer se algoritam strojnog učenja opetovano poziva. Iz tih razloga, postupci omotača ne prilagođuju se dobro glomaznim skupovima podataka s većim brojem atributa. S druge strane, algoritmi filtera funkcioniraju neovisno o algoritmima strojnog učenja – nepoželjni atributi se izdvajaju prije nego proces učenja započne. Oni se općenito izvode znatno brže od algoritama omotača i stoga su praktičnije rješenje za primjenu na većim skupovima podataka.
Stvarni poslovni problemi kakvima je namijenjeno razvijeno klasifikacijsko okruženje u pravilu podrazumijevaju operiranje nad obimnim korporativnim bazama podataka. Stoga je u klasifikacijskom sustavu primijenjen pristup filtera u odabiru poželjnih atributa. Implementirani filter oslanja se na heuristiku koja se temelji na procjeni korelacije među atributima u skupu podataka za učenje [23]. Ova heuristika ocjenjuje podskupove atributa uzimajući u obzir korisnost pojedinih atributa pri predviđanju klase primjera, zajedno sa stupnjem korelacije među atributima iz podskupa. Heuristika se zasniva na hipotezi koja se može formulirati ovako:

> *Dobri podskupovi atributa sadrže međusobno nekorelirane atribute koji su istovremeno visoko korelirani s atributom klase.*

Izraz (77) formalizira spomenutu heuristiku:

$$G_s = \frac{k\,\overline{r_{cf}}}{\sqrt{k + k(k-1)\,\overline{r_{ff}}}} \quad , \tag{77}$$

gdje je G_S heuristička mjera podskupa atributa S koji sadržava k atributa, $\overline{r_{cf}}$ je prosječna korelacija atributa i klase ($f \in S$), a $\overline{r_{ff}}$ je srednja korelacija među atributima. Brojnik ukazuje na prediktivnost grupe atributa u odnosu na klasu, dok nazivnik izražava stupanj redundancije među atributima. Time heuristika diskriminira nevažne atribute jer imaju lošu prediktivnost klase, a redundantni atributi se eliminiraju jer su visoko korelirani s jednim ili više preostalih atributa. Dakle, ovako definirana funkcija vrednovanja će tražiti podskupove atributa s niskom međusobnom korelacijom, uz visoku koreliranost s atributom klase.
Da bi se pomoću (77) vrednovali podskupovi atributa, potrebno je odrediti način izračuna vrijednosti korelacije među atributima. Radi pojednostavljenja problema, prije izračunavanja iznosa potrebnih korelacija svi numerički atributi se pretvaraju u nominalne primjenom metode entropijske diskretizacije. Budući da nakon toga skup podataka sadrži samo nominalne atribute, dovoljno je utvrditi metodu izračuna korelacije dvaju nominalnih atributa. Dakle, izbjegnut je izračun korelacije dvaju numeričkih atributa i korelacije numeričkog i nominalnog atributa, kao i potreba usklađivanja različitih metodi izračuna. Valja napomenuti da se diskretizacija numeričkih atributa odnosi samo na postupak odabira atributa; postupci probabilističkog i deskriptivnog modeliranja koji slijede opet koriste početne

numeričke atribute. Osnova za procjenu razine koreliranosti dvaju nominalnih atributa je mjera posuđena iz teorije informacija: ako su X i Y dvije diskretne slučajne varijable, sa (78) i (79) je izražena entropija od Y prije i nakon opažanja X.

$$E(Y) = -\sum_{y \in Y} p(y) \log_2 p(y) \tag{78}$$

$$E(Y|X) = -\sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log_2 p(y|x) \tag{79}$$

Vrijednost za koju se entropija od Y smanji nakon opažanja X (i obrnuto) izražava razinu dodatnih informacija o varijabli Y koju nosi poznavanje vrijednosti varijable X. Ova mjera naziva se *dobitak informacije* (80).

$$\begin{aligned} IG(X,Y) &= E(Y) - E(Y|X) \\ &= E(X) - E(X|Y) \\ &= E(Y) + E(X) - E(X,Y) \end{aligned} \tag{80}$$

Dobitak informacije je očito simetrična mjera, pa se stoga može koristiti za ocjenu korelacije dvaju prognostičkih atributa, gdje nijedan atribut nije posebno izdvojen kao oznaka klase. Međutim, mjera dobitka informacije neopravdano favorizira atribute s više vrijednosti. Osim toga, da bi iznosi korelacija u (77) bili usporedivi i imali jednak učinak, potrebno ih je normalizirati. *Simetrična neizvjesnost* (81) kompenzira naklonjenost dobitka informacije brojnijim atributima i normalizira vrijednost na interval [0,1].

$$SU(X,Y) = 2 \times \left[\frac{IG(X,Y)}{E(Y) + E(X)} \right] \tag{81}$$

Vrijednost 0 naznačuje da vrijednosti varijabli X i Y uopće nisu povezane, dok vrijednost 1 govori da se poznavanjem vrijednosti varijable X mogu potpuno predvidjeti vrijednosti varijable Y.

Uz funkciju vrednovanja podskupova atributa, postupak odabira atributa mora uključivati i algoritam pretraživanja da bi istražio prostor potencijalnih rješenja. Potencijalno rješenje je bilo koji podskup inicijalnog skupa atributa. Budući da je veličina prostora rješenja eksponencijalna u odnosu na broj atributa, iscrpno pretraživanje je obično neprimjereno. Od mnoštva heurističkih metoda pretraživanja, implementirani postupak odabira atributa koristi metodu najboljeg prvog, budući da je ova metoda pokazala nešto bolje rezultate u pojedinim slučajevima. Metoda najboljeg prvog polazi od praznog podskupa atributa i generira sva moguća proširenja s jednim atributom. Odabire se podskup s najvećom vrijednosti prema funkciji vrednovanja, te se on proširuje na isti način, dodajući jedan atribut. Ako proširenje nekog podskupa ne rezultira povećanjem funkcije vrednovanja, postupak pretraživanja se nastavlja počevši od najboljeg dotad neproširenog podskupa. Bez dodatnih ograničenja, ova metoda bi pretražila cijeli prostor potencijalnih rješenja, pa se obično ograniči broj proširenja koja ne rezultiraju povećanjem funkcije vrednovanja. Kad postupak pretraživanja završi, kao rezultat se vraća najbolji pronađeni podskup atributa prema funkciji vrednovanja.

7.2.3. Klasifikacijski postupci

Zbog presudnog utjecaja kojeg tehnike modeliranja podataka imaju na cijeli proces inteligentne analize podataka, modeliranje podataka predstavlja najizazovniju fazu cijelog procesa. Prvi korak ove faze je odabir odgovarajuće tehnike modeliranja između brojnih postojećih tehnika inteligentnog modeliranja podataka. Pri donošenju odluke treba se prvenstveno rukovoditi prirodom promatranih problema: potrebno je pažljivo razmotriti osnovne karakteristike ciljne skupine problema u odnosu s različitim tehnikama inteligentne analize podataka i njihovim specifičnostima. Rukovodeći se ovim načelima, izgrađeno klasifikacijsko okruženje kombinira i nadovezuje različite tehnike inteligentne analize podataka pri izradi probabilističkog i deskriptivnog modela [56].

Probabilistički karakter tipičnih poslovnih problema traži tehniku analize podataka koja pridružuje vjerojatnost svakoj predikciji. Nadalje, relativno mali udio primjera koji predstavljaju ciljni pojam sugerira da tehnika modeliranja treba uzimati u obzir cijenu pogrešne klasifikacije. Ako se ova dva bitna zahtjeva ignoriraju, jako dobrim se čini jednostavan (ali beskoristan) algoritam koji bi uvijek predviđao odsutnost ciljnog pojma. Zbog malog udjela ciljnih entiteta, može se očekivati da bi ovakav algoritam ispravno klasificirao više od 90% primjera, što je impresivan nivo točnosti u bilo kojoj poslovnoj domeni. Nije potrebno naglašavati da je stvarna prediktivna vrijednost ovakvog algoritma zapravo nikakva. Razvijeni sustav inteligentne analize podataka implementira naivni Bayesov klasifikator s procjenom distribucije vjerojatnosti pri modeliranju numeričkih atributa. Bayesovi klasifikatori udovoljavaju obama postavljenim zahtjevima i predstavljaju obećavajući pristup probabilističkom otkrivanju znanja: inherentno su osjetljivi na cijenu pogrešne klasifikacije, te eksplicitno tretiraju pitanja neizvjesnosti i šuma u podacima, što su središnji problemi svakog zadatka otkrivanja znanja u stvarnim poslovnim primjenama.
Iako mnoge druge klasifikacijske tehnike mogu generirati probabilističke predikcije i mnoge od njih se mogu prilagoditi razmatranju cijene pogrešne klasifikacije (primjerice, dodjeljivanjem težinskih vrijednosti primjerima za učenje), eksperimenti na stvarnim podacima su više puta pokazali da je Bayesov pristup konkurentan mnogim sofisticiranijim tehnikama indukcije znanja [41]. U vlastitom eksperimentu evaluirani su različiti klasifikatori korištenjem kvadratne funkcije gubitka na podacima nekoliko tipičnih poslovnih problema s nedeterminističkim obilježjima. Statističko ispitivanje korištenjem uparenog t-testa u više slučajeva je pokazalo statistički značajnu razliku u performansama naivnog Bayesovog klasifikatora u usporedbi s drugima.

Kao što je već rečeno, naivni Bayesov klasifikator predstavlja semantički jasan i jednostavan pristup prikazu, korištenju i indukciji probabilističkog znanja. Nazvan je naivnim jer pojednostavljuje problematiku oslanjajući se na dvije važne pretpostavke: pretpostavlja da su prognostički atributi uvjetno nezavisni uz poznatu klasifikaciju, te drži da ne postoje prikriveni atributi koji bi mogli utjecati na proces predikcije. Ove pretpostavke omogućuju vrlo efikasne algoritme kako za klasifikaciju tako i za strojno učenje.

Radi ilustracije, neka je C slučajna varijabla koja označava klasu primjera, a X vektor slučajnih varijabli koji označava vrijednosti prognostičkih atributa primjera iz skupa podataka. Nadalje, neka je sa c i x označena određena vrijednost klase, odnosno određena vrijednost vektora prognostičkih atributa. Pri klasifikaciji testnih primjera korištenjem Bayesove formule (82) izračunava se vjerojatnost svake od klasa uz uvjet zadane vrijednosti vektora x koja označava vrijednosti prognostičkih atributa testnog primjera. Klasa s najvećim iznosom vjerojatnosti predstavlja rezultat predikcije.

$$p(C=c|X=x) = \frac{p(C=c)\,p(X=x|C=c)}{p(X=x)} \tag{82}$$

U (82) $X=x$ označava događaj $X_1=x_1 \wedge X_2=x_2 \wedge ... \wedge X_k=x_k$. Budući da je ovaj događaj konjunkcija uvjetno nezavisnih događaja (atributi su međusobno nezavisni prema pretpostavci), izraz je moguće napisati kao umnožak pojedinačnih vjerojatnosti (83). Takve pojedinačne vjerojatnosti jednostavno je procijeniti iz skupa podataka za učenje.

$$\begin{aligned} p(X=x|C=c) &= p\left(\bigwedge_i X_i=x_i|C=c\right) \\ &= \prod_i p(X_i=x_i|C=c) \end{aligned} \tag{83}$$

Pri izračunima se vrijednost u nazivniku izraza (82) u pravilu ignorira, budući da se pojavljuje u nazivniku za svaku klasu c, pa se može smatrati normalizacijskim faktorom. Utjecaj nazivnika se eliminira u završnoj normalizaciji, kada se osigurava da je suma vrijednosti $p(C=c \mid X=x)$ po svim klasama jednaka 1.
Vrijedi napomenuti da ovakav način klasifikacije osigurava prirodan tretman neodređenih vrijednosti atributa i u fazi učenja i u fazi predikcije. Potrebne procjene vjerojatnosti na osnovu skupa za učenje, kao i izračuni prilikom predikcije u obzir uzimaju samo poznate vrijednosti atributa. Završna normalizacija osigurava usporedivost vrijednosti s primjerima u kojima su sve vrijednosti atributa poznate.

Naivni Bayesov algoritam tretira nominalne i numeričke atribute ponešto drugačije. Za svaki nominalni atribut X_i, $p(X_i=x_i \mid C=c)$ se modelira kao jedan realni broj između 0 i 1, čija vrijednost se procjenjuje iz podataka za učenje. Prema tome, za svaki par klase i nominalnog atributa potrebno je procijeniti vjerojatnost da će atribut poprimiti svaku od mogućih vrijednosti odgovarajuće domene, uz uvjet zadane klase. Najbolja ocjena vjerojatnosti da će diskretna slučajna varijabla poprimiti određenu vrijednost je frekvencija uzorka: kvocijent broja primjera kod kojih je odabrana vrijednost prisutna i broja svih primjera. Međutim, striktna primjena kvocijenta kao ocjene vjerojatnosti može dovesti do neželjenih efekata pri upotrebi izraza (83). Problemi se javljaju ako u skupu za učenje postoji vrijednost nekog atributa koja se ne pojavljuje u kombinaciji sa svakom od klasa. Ilustracije radi, neka je sa c označena proizvoljna klasa, a sa x_i vrijednost atributa X_i takva da u skupu za učenje ne postoji primjer klase c za koji je $X_i=x_i$. U tom slučaju procjena vjerojatnosti $p(X_i=x_i \mid C=c)$ na osnovu frekvencije uzorka iznosi 0. Uvrštavanje u (83) nulificira cijeli produkt, bez obzira na vrijednosti ostalih faktora. Ovakva vrsta veta na krajnji rezultat nije poželjno svojstvo, pa se radi izbjegavanja situacija ovog tipa način ocjene vjerojatnosti iz

frekvencije uzorka blago korigira. Za izbjegavanje vrijednosti 0 kao ocjene vjerojatnosti često se koristi Laplaceov estimator, budući da se svodi na inicijalizaciju brojača konstantom 1 umjesto 0.
S druge strane, svaki numerički atribut se modelira nekom neprekidnom funkcijom distribucije vjerojatnosti nad domenom promatranog atributa. Radi jednostavnosti, u praksi se vrlo često koristi normalna distribucija vjerojatnosti. Normalna distribucija se zadaje u terminima srednje vrijednosti i standardne devijacije, čije se vrijednosti mogu efikasno procijeniti iz podataka za učenje. Dakle, pri izračunu $p(X_i=x_i \mid C=c)$ za numerički atribut X_i koristi se funkcija gustoće vjerojatnosti za normalnu distribuciju:

$$p\left(X_i=x_i|C=c\right) = g(x_i; \mu_c, \sigma_c) \quad , \tag{84}$$

gdje je

$$g(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad . \tag{85}$$

Međutim, (84) nije potpuno točno: vjerojatnost da kontinuirana slučajna varijabla poprimi točno određenu vrijednost je uvijek 0. Kod kontinuiranih slučajnih varijabli promatra se vjerojatnost da vrijednost varijable leži u nekom intervalu.

$$P(x \le X \le x+\Delta) = \int_{x}^{x+\Delta} g(x; \mu, \sigma)\, dx \tag{86}$$

Tada je po definiciji derivacije

$$\lim_{\Delta \to 0} \frac{P(x \le X \le x+\Delta)}{\Delta} = g(x; \mu, \sigma) \quad . \tag{87}$$

Prema tome, za neku vrlo malu konstantu Δ je $p(X=x) \approx g(x; \mu, \sigma) \times \Delta$. Faktor Δ bi se pojavio u brojniku izraza (82) za svaku od klasa. Pri završnoj normalizaciji ovi faktori se poništavaju, pa se može koristiti (84). Prije primjene (84), potrebno je procijeniti srednju vrijednost i standardnu devijaciju svakog numeričkog atributa uz uvjet poznate klase, što se jednostavan zadatak uz korištenje skupa podataka za učenje.

Ohrabrujući rezultati ocjene performansi ovih algoritama stvorili su motivaciju za istraživanje dorada i modifikacija Bayesovog postupka koje bi umanjile ovisnost o spomenutim pretpostavkama, ali zadržale prvotnu jednostavnost i jasnu probabilističku semantiku. Postupak odabira atributa u fazi pripreme podataka koji uklanja redundantne atribute u velikoj mjeri ublažava pretpostavku nezavisnosti atributa. Iako nije prirođena Bayesovom postupku, pretpostavka da vrijednosti numeričkih atributa unutar svake klase imaju normalnu ili Gaussovu distribuciju vjerojatnosti se skoro podrazumijeva. Iako korištenje normalne distribucije pri modeliranju numeričkih atributa može predstavljati razumnu aproksimaciju u mnogim stvarnim problemima, to zasigurno nije uvijek najbolja aproksimacija. Razvijeno klasifikacijsko okruženje napušta pretpostavku normalne distribucije i koristi proširenje naivnog Bayesovog algoritma koje primjenjuje statističke metode za neparametarsku ocjenu gustoće vjerojatnosti numeričkih atributa. Od velikog broja takvih metoda odabrana je ocjena gustoće vjerojatnosti korištenjem Gaussovih

jezgri [30] (kao jezgra se mogu koristiti i druge funkcije). Prema ovoj metodi, za procjenu vjerojatnosti $p(X_i=x_i \mid C=c)$ numeričkog atributa X_i koristi se izraz (88):

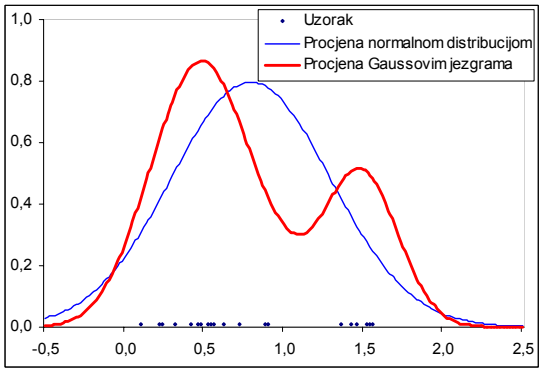
$$p(X_i=x_i|C=c) = \frac{1}{n}\sum_k g(x_i;\mu_k,\sigma_c) \quad , \tag{88}$$

gdje k prolazi svim vrijednostima atributa X_i u podacima za učenje koji imaju klasu c, a $\mu_k = x_k$. Ovo je vrlo slično izrazu (84), samo što se procijenjena gustoća vjerojatnosti usrednjava po velikom skupu Gaussovih jezgri. Vrijedi primijetiti da je (88) ekvivalentno standardnom izrazu za ocjenu gustoće vjerojatnosti korištenjem jezgri

$$p(X=x|C=c) = \frac{1}{nh}\sum_k K\left(\frac{x-\mu_k}{h}\right) \quad , \tag{89}$$

gdje je $h=\sigma$ i $K(x)=g(x;0,1)$.

Važan element metode ocjene gustoće vjerojatnosti korištenjem jezgri je odabir parametra širine σ. Može se pokazati da ova metoda ocjene gustoće vjerojatnosti pokazuje neka dobra teoretska svojstva ako se parametar σ približava nuli kako brojnost uzorka raste ka beskonačnosti [30]. U statističkoj literaturi postoji mnoštvo različitih preporuka za odabir širine jezgre, ali svaka heuristika ima neke pretpostavke na funkciju gustoće vjerojatnosti. U korištenom algoritmu primjenjuje se $\sigma_c = 1/\sqrt{n_c}$, gdje je n_c broj primjera za učenje sa klasom c. Stoga procjena gustoće vjerojatnosti postaje sve lokalnija što je više vrijednosti u uzorku. Tanka plava crta na slici 26 prikazuje gustoću vjerojatnosti procijenjenu normalnom distribucijom, dok deblja crvena crta označava procjenu gustoće korištenjem Gaussovih jezgri. Obje procjene se temelje na istom uzorku koji sadrži 20 vrijednosti.



Slika 26. Usporedba metoda procjene gustoće vjerojatnosti numeričkih atributa

U svrhu opisivanja ciljnog pojma, izgrađeni sustav inteligentne analize podataka uključuje varijantu poznatog C4.5 algoritma za konstrukciju stabala odlučivanja [49]. Generator stabla odlučivanja se primjenjuje nakon što se interesantni primjeri izdvoje u procesu evaluacije probabilističkog modela. Uočena unutrašnja obilježja izdvojenog podskupa interesantnih primjera se predočuju stablom odlučivanja koje povezuje određena svojstva primjera s rezultatom klasifikacije. Ovakav opis otkriva vrijedne informacije o istraživanom poslovnom problemu koje mogu bitno usmjeriti i

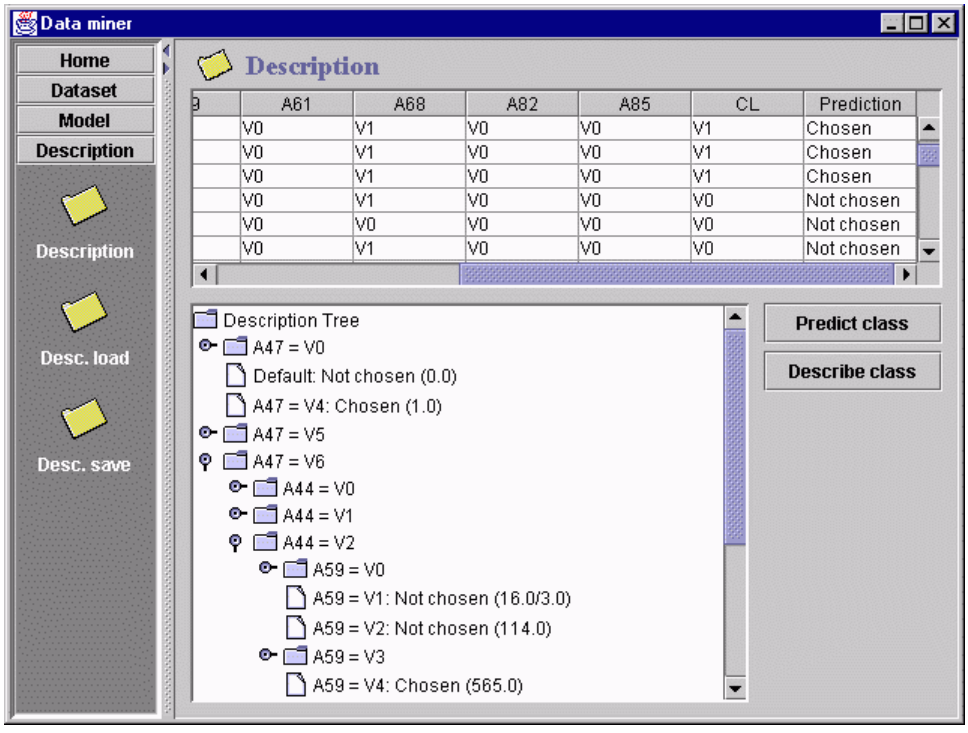
olakšati proces donošenja odluka. Stjecanje uvida u narav poslovnog problema otkrivanjem novih informacija koje imaju potporu u stvarnim podacima nekad može biti važnije od mogućnosti predviđanja ishoda za nove primjere nepoznate klasifikacije.
Kod podataka s izraženim šumom, postupci indukcije stabala odlučivanja su podložni problemu pretjerane prilagodbe podacima za učenje. Standardno rješenje ovog problema su tehnike podrezivanja stabla odlučivanja. U klasifikacijskom okruženju koriste se dva različita postupka podrezivanja kojima se nastoji izbjeći problem pretjerane prilagodbe: *zamjenu podstabala* i *izdizanje podstabala*. Oba postupka nastoje preurediti strukturu pogodnih grana uklanjajući nepotrebne čvorove, te tako smanjuju i pojednostavljuju inicijalno stablo odlučivanja. Opseg obiju operacija podrezivanja može se kontrolirati odabirom odgovarajućih parametara kroz samu aplikaciju.

7.2.4. Vizualna evaluacija klasifikacijskih postupaka

Proces inteligentne analize podataka odvija se u ciklusima koji uključuju odabir skupa podataka za učenje, uređivanje i filtriranje podataka, primjenu algoritma inteligentne analize podataka, odgovarajući prikaz otkrivenog znanja, te ocjenu performansi sustava na novom skupu podataka. Samo jedan prolaz kroz ove faze obično ne može dati očekivani rezultat. Zbog toga proces inteligentne analize podataka uzastopno konstruira i ocjenjuje više izvedenih modela, slično kao što bi nekoliko stručnjaka analiziralo problem s drugačijih stajališta i s različitim kriterijima.
Ugradnjom pogodne metode ocjene performansi u sustav analize podataka mogu se ostvariti značajna unapređenja procesa inteligentne analize podataka. Mogućnost ocjene performansi induciranih modela pomaže u namještanju parametara sustava u svakoj iteraciji modeliranja. Razvijeno klasifikacijsko okruženje podržava sve faze procesa inteligentne analize podataka, pa tako i ocjenu performansi izvedenih modela. Ugrađeni vizualni alat omogućava djelotvornu i vjerodostojnu evaluaciju otkrivenog znanja, omogućujući tako znatno brže dosezanje konačnog modela.

Kao rezultat klasifikacije, naivni Bayesov klasifikator daje procjenu vjerojatnosti pripadanja svakoj od klasa. Međutim, nije uputno koristiti ove vjerojatnosti u svrhu dobivanja pouzdane konačne prognoze klase. Takav pristup podcjenjuje narav i složenost ciljne skupine problema, ali i precjenjuje mogućnosti svakog postupka inteligentne analize podataka. Kod stvarnih poslovnih problema s izraženim nedeterminističkim obilježjima, kategorična klasifikacija nije primjerena formulacija procesa predikcije. Realističnija formulacija je pronalaženje podskupa primjera s visokom zastupljenošću ciljnog pojma, što je ekvivalentno izdvajanju podskupa primjera kod kojih je vjerojatnost pripadanja ciljnoj klasi veća od neke granične vjerojatnosti. Kardinalnost traženog podskupa (ili odgovarajuća granična vjerojatnost) obično ovisi o iznosu troškova i koristi koje proizlaze iz domene poslovnog problema. U praksi je često teško procijeniti troškove i korist s potrebnom razinom pouzdanosti. Obično se razmatra veći broj podskupova različitih veličina prije nego što se ustanovi optimalna kardinalnost.
Sposobnost alata za inteligentnu analizu podataka da izdvoji podskupove primjera s visokim udjelom ciljnog pojma ovisi o kardinalnosti podskupa. Jasno je da su

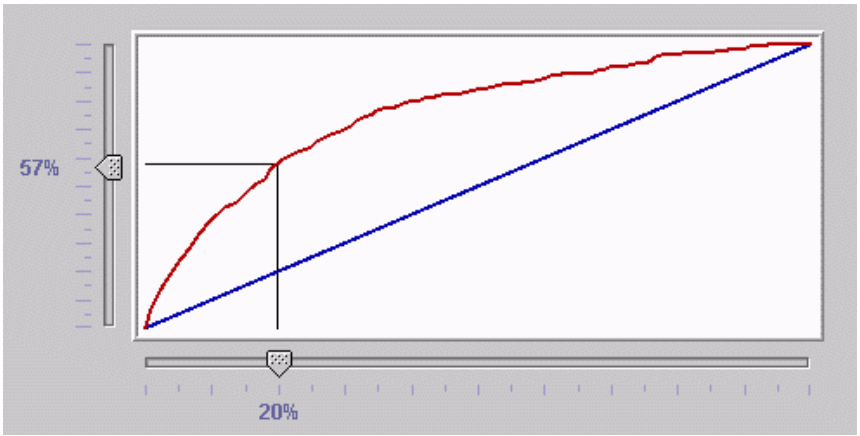
performanse sustava bolje ako uz zadanu kardinalnost pronađe podskup s većim udjelom ciljnog pojma. Ovisnost udjela ciljnog pojma o kardinalnosti podskupa može se grafički prikazati u obliku dvodimenzionalnog grafa. Horizontalna os grafa označava kardinalnost podskupa izraženu kao postotak ukupnog broja primjera. Na vertikalnu os se nanosi broj primjera ciljne klase sadržanih u pronađenom podskupu odgovarajuće kardinalnosti (često se izražava kao postotak ukupnog broja primjera ciljne klase). Ovako dobiven graf naziva se *graf izdizanja*, te predstavlja vrijedan alat za vizualnu ocjenu performansi modela [5]. Često se koristi u raznim poslovnim područjima, a posebno u marketingu. Metoda grafičkog predočavanja performansi izvedenih modela pomoću grafa izdizanja je dobro prihvaćena od strane stručnjaka iz poslovnog područja, pa omogućava bržu evaluaciju modela i podešavanje parametra sustava, što u konačnici rezultira poboljšanom preciznošću klasifikacije.
Klasifikacijsko okruženje koristi metodu grafa izdizanja za ocjenu performansi izvedenog Bayesovog klasifikatora. Uzimajući u obzir koristi i troškove više različitih scenarija, korisnik može postaviti odabranu kardinalnost traženog podskupa. Na ovaj način bira se podskup koji najviše obećava, a ugrađeni generator stabala odlučivanja može dodatno opisati njegova svojstva. Slika 27 prikazuje primjer generiranog stabla odlučivanja.



Slika 27. Otkriveno znanje u obliku stabla odlučivanja

Implementacija metode grafa izdizanja je prilično jednostavna. Nakon primjene izvedenog klasifikatora na odabranom skupu podataka s poznatom klasifikacijom, primjeri se sortiraju silazno po procijenjenoj vjerojatnosti pripadanja ciljnoj klasi. Nakon ovog koraka, iznos vjerojatnosti više nije bitan – nužan je samo utvrđeni redoslijed primjera. Kad se traži udio pozitivnih primjera klase u podskupu zadane kardinalnosti *n* kojeg klasifikator može izdvojiti, promatra se prvih *n* primjera s liste. Budući da je stvarna klasa svakog primjera poznata, traženi udio se određuje

prebrojavanjem pozitivnih primjera koje uzorak sadrži, a postotak se dobiva dijeljenjem s ukupnim brojem pozitivnih primjera. Iteriranjem preko svih mogućih kardinalnosti podskupova nastaje graf izdizanja. Na slici 28 prikazan je primjer grafa izdizanja kakvog generira klasifikacijsko okruženje.



Slika 28. Primjer grafa izdizanja

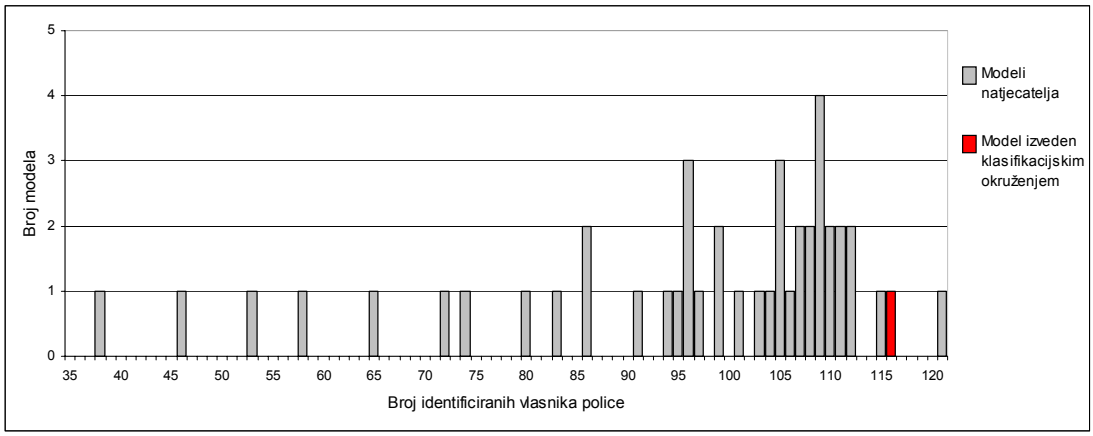
Dijagonalna crta predstavlja očekivani rezultat za slučajne uzorke različitih veličina. Gornja crta je izvedena korištenjem uzoraka s visokim udjelom ciljnog pojma koje uspijeva izdvojiti alat za inteligentnu analizu podataka, ukazujući tako na performanse izvedenog klasifikatora. Očito, operativno područje grafa izdizanja je gornji trokut; cilj je biti što bliže gornjem lijevom kutu. Jedan od mogućih scenarija je istaknut na grafu: klasifikator je sposoban izdvojiti podskup od 20% ukupnog broja primjera koji sadrži 57% svih pozitivnih primjera.

7.3. EKSPERIMENTALNA OCJENA PREDIKTIVNIH SPOSOBNOSTI SUSTAVA

Prediktivne sposobnosti opisanog sustava za inteligentnu analizu podataka mogu se ilustrirati na primjeru poslovnog problema iz osigurateljne djelatnosti. Problem je izvorno formuliran kao natjecanje u inteligentnoj analizi podataka koje je organizirao *Computational Intelligence and Learning Cluster (CoIL)*, mreža više organizacija sponzorirana od strane EU [47]. Natjecanje je nazvano *CoIL Challenge 2000: The Insurance Company Case* i održano je u 2000. godini, a ovdje se koristi samo kao mjera klasifikacijskih sposobnosti opisanog sustava.
Zadatak natjecanja je zasnovan na stvarnom problemu izravnog oglašavanja osigurateljnog proizvoda putem pošte. Izravno slanje promotivnih materijala potencijalnim korisnicima je djelotvoran način oglašavanja proizvoda ili usluge. Međutim, veliki dio ovakvih pisama završi u košu za smeće jer oglašeni proizvod nije zanimljiv većini ljudi koji ga prime. Stoga bolje profiliranje budućih klijenata igra važnu ulogu u organiziranju efikasnijih promidžbenih akcija putem pisama: ograničavanje slanja promotivnih pisama na korisnike koji će vjerojatno kupiti određeni proizvod može povećati odaziv klijenata, istovremeno smanjujući troškove i opseg same promidžbene akcije.

Cilj natjecateljskog zadatka bio je predvidjeti tko će biti zainteresiran za kupnju police osiguranja kamp-kućice i objasniti zašto. Podaci o klijentima sastojali su se od 85 atributa koji su bilježili sociodemografska svojstva (43 atributa, npr. tip korisnika, broj kuća/stanova, razina kupovne moći i sl.) i oblik poslovanja s klijentom (42 atributa, npr. broj osiguranja od požara, broj osiguranja čamaca, premija automobilskih osiguranja i sl.). Svi sociodemografski podaci izvedeni su iz mjesta prebivanja; svi klijenti s istim prebivalištem imali su iste vrijednosti sociodemografskih atributa. Podatke je priskrbila nizozemska tvrtka za inteligentnu analizu podataka *Sentient Machine Research*. Skup podataka za učenje sastojao se od 5822 primjera klijenata s poznatom klasifikacijom, a testni skup podataka od 4000 primjera za koje su samo organizatori znali klasifikaciju.

Sa stanovišta analize podataka, zadatak je izdvojiti podskup klijenata s vjerojatnošću kupnje police osiguranja kamp-kućice iznad neke granične vjerojatnosti. Granica ovisi o koristi i troškovima akcije, kao što su trošak slanja pisma i korist od kupnje police osiguranja. Radi jednostavnosti, zadatak je bio iz testnog skupa podataka izolirati podskup od 800 klijenata koji sadrži što više vlasnika osiguranja kamp-kućice. Stvarni broj osiguranika u izdvojenom podskupu klijenata rangira rješenja natjecatelja. Od 147 prijavljenih natjecatelja, 43 su ponudili rezultate. Dijagram na slici 29 prikazuje rangirane rezultate.



Slika 29. Prikaz rezultata natjecanja

U testnom skupu podataka bilo je ukupno 238 vlasnika police osiguranja. Pobjednički model je izdvojio 121 vlasnika, a drugoplasirani i trećeplasirani 115 odnosno 112 vlasnika police. Razvijeno klasifikacijsko okruženje uspjelo je iz testnog skupa izdvojiti 116 vlasnika police u prvoj iteraciji modeliranja. Ovaj rezultat bi na natjecanju bio smješten između prvog i drugog mjesta, što je vrijedan rezultat uzimajući u obzir akademski i industrijski ugled natjecatelja.

Zanimljivo svojstvo izvedenog modela je činjenica da je samo 8 od 85 raspoloživih atributa bilo korišteno u procesu predviđanja. Algoritam odabira atributa izdvojio je samo jedan sociodemografski atribut i sedam atributa koji opisuju poslovanje s klijentom. Koristeći ovaj prilično sužen skup atributa, klasifikacijsko okruženje proizvelo je efikasan probabilistički model za predviđanje potencijalnih korisnika.

Osim toga, operacija odabira atributa poboljšala je i kvalitetu izvedenog opisa – smanjeni broj atributa rezultirao je relativno malim i razumljivim stablom odlučivanja koje opisuje potencijalne korisnike.

7.4. OSVRT NA SVOJSTVA INTEGRIRANOG KLASIFIKACIJSKOG OKRUŽENJA

Usprkos pojednostavljujućim pretpostavkama koje leže u osnovi naivnog Bayesovog klasifikatora, oni predstavljaju glavnu alternativu sofisticiranijim pristupima u inteligentnoj analizi podataka kad se radi o stvarnim poslovnim problemima. Ključnim faktorom uspjeha nad stvarnim podacima smatra se teoretski zasnovan pristup neizvjesnosti i šumu u podacima. Ovdje su istražena proširenja i dorade Bayesovog postupka koja umanjuju utjecaj pretpostavki na koje se algoritam oslanja. Pretpostavka nezavisnosti atributa se kompenzira algoritmom odabira atributa, a pretpostavka normalne distribucije numeričkih atributa se napušta u korist procjene gustoće vjerojatnosti korištenjem Gaussovih jezgri.
Eksperimenti sa standardnim algoritmima strojnog učenja su pokazali da je korištena metoda odabira atributa konkurentna metodama omotača, a zahtjeva znatno manje izračuna [24]. Nadalje, zbog globalne procjene korelacija među atributima, ova metoda je sklona konstruiranju prilično agresivnih filtera, koji u pravilu odbacuju više od polovice inicijalnih atributa. Ovako smanjena dimenzionalnost podataka omogućava postupcima učenja brži i efikasniji rad. Utjecaj postupka odabira atributa u klasifikacijskom okruženju bio je povoljan za oba korištena algoritma strojnog učenja. Prema očekivanjima, preciznost klasifikacije kod naivnog Bayesovog postupka je povećana budući da su redundantni atributi uklonjeni iz podataka za učenje. S druge strane, C4.5 algoritam konstrukcije stabala odlučivanja eksplicitno nastoji koristiti samo relevantne atribute. Zbog toga je u ovom slučaju odabir atributa pokazao samo manji utjecaj na preciznost klasifikacije. Međutim, rezultirajuća stabla odlučivanja su često bila manja i kompaktnija, te stoga i razumljivija.
Umjesto pretpostavke normalne distribucije, korištena je procjene gustoće vjerojatnosti uporabom Gaussovih jezgri pri modeliranju numeričkih atributa. Eksperimentalno je pokazano da je ova modifikacija Bayesovog postupka vrlo korisna, iako vodi manjem povećanju složenosti i prostornih zahtijeva algoritma [30]. U većini slučajeva gdje je narušena pretpostavka normalnosti performanse su znatno bolje, dok je utjecaj neznatan u slučajevima kad pretpostavka stoji.
Osim toga, integracija svih faza procesa inteligentne analize podataka unutar istog aplikacijskog okruženja omogućava bolju kontrolu nad samim procesom, čineći ga interaktivnijim i stoga djelotvornijim. Ovo se posebno odnosi na integrirani alat za ocjenu performansi, koji osigurava neposredne kontrolu preciznosti klasifikacije i ubrzava dosezanje konačnog modela.
Konačno, prikazani rezultati eksperimentalne evaluacije pokazuju da opisano klasifikacijsko okruženje koristi efikasnu i robusnu kombinaciju metoda inteligentne analize podataka pri rješavanju stvarnih poslovnih problema, gdje su zanimljive pravilnosti često skrivene među ogromnim količinama nebitnih podataka.

8. Zaključak

Inteligentna 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.
Realno gledano, inteligentna analiza podataka počiva na skupu praktičnih i lako shvatljivih tehnika kojima je svrha izdvajanje korisnih informacija iz sirovih podataka. Njihova primjena se ne svodi na puko iniciranje procesa analize podataka i čekanje rezultata obrade. Proces inteligentne analize podataka se sastoji od više faza, a završni uspjeh ovisi o pažljivoj pripremi i izvođenju svake od njih.
Tehnike inteligentne analize podataka karakteriziraju svojstva koja se od tehnike do tehnike mogu bitno razlikovati. Odabir adekvatne tehnike u svakoj od faza modeliranja pravilnosti u podacima ima presudan utjecaj na oblik i kvalitetu rezultata. Općenito, ne postoji dobitna kombinacija tehnika koja bi davala dobre rezultate u svim primjenama. Stoga su gotova programska rješenja za inteligentnu analizu podataka rijetka. Ozbiljan pristup podrazumijeva analizu svojstava promatranog problema, te krojenje procesa inteligentne analize podataka prema njegovim specifičnostima. Stoga je za uspješnu primjenu nužno dobro poznavanje tehnika inteligentne analize podataka i njihovih svojstava.

Jedna od najraširenijih primjena inteligentne analize podataka u poslovnom svijetu (a i općenito) je izgradnja profila korisnika usluga. Otkrivanje pravilnosti u ponašanju korisnika omogućuje bolje razumijevanje njihovih navika, potreba i motiva. U današnjoj korisnički orijentiranoj ekonomiji ovakve spoznaje mogu biti značajan izvor potencijalnih prednosti nad konkurencijom. Na osnovu njih se mogu provoditi ciljane akcije usmjerene na zadržavanje postojećih i privlačenje novih korisnika. Jednako tako, iskorištavanje otkrivenih informacija može ići u smjeru uobličavanja proizvoda i njihovih cijena u skladu s različitim profilima korisnika i njihovim ustaljenim navikama.
Ovo su razlozi zbog kojih je primjena inteligentne analize podataka u izgradnji profila korisnika usluga interesantna u svim korisnički orijentiranim djelatnostima, a posebice onim koje dolaze u dodir s većim brojem klijenata i bilježe interakcije s njima.
Problemi koji se svrstavaju pod zajednički termin profiliranja korisnika usluga dolaze u više oblika: od usmjeravanja marketinških promocija, procjene rizičnosti klijenata, analize sastava tipične potrošačke košarice, pa do analize uzroka prelaska konkurenciji ili prelazaka iz konkurencije. Međutim, bez obzira na oblik, postoje tipična svojstva koja su zajednička za sve probleme ovog tipa.

– *Neizvjesnost*.
  Kad god su u pitanju ljudi i njihovo ponašanje, uvijek je prisutan element neizvjesnosti. Ljudi sličnih osobina neće se jednako ponašati u istim situacijama. Čak i isti čovjek neće uvijek jednako reagirati na iste ili slične situacije. Ponašanje ljudi uvjetovano je mnogim čimbenicima od kojih većinu nije moguće objektivno mjeriti. Stoga se u i samim podacima za učenje može očekivati određeni stupanj neizvjesnosti, dok je kod modela koji opisuju pripadnost korisnika određenom profilu neizvjesnost neizbježan element.

- *Nebitni i redundantni podaci*.
  Poslovni subjekti podatke o svojim klijentima bilježe na više mjesta. Tipično, za različite oblike poslovanja s klijentom postoje različite aplikacije koje prikupljaju njima bitne podatke o klijentima. Ukupnost podataka o klijentu može se dobiti integracijom podataka iz svih raspoloživih izvora. Iako je brojnost i raznovrsnost dostupnih obilježja klijenata načelno poželjna, postojanje atributa koji su irelevantni za promatrani problem može imati negativan utjecaj na proces inteligentne analize podataka. Ovisno o primijenjenoj tehnici modeliranja, uvođenje nebitnih ili redundantnih atributa može se negativno odraziti na točnost izvedenih modela, kao i na vremensku i prostornu zahtjevnost postupka.
- *Neodređene vrijednosti i šum u podacima*.
  Budući da skup prikupljenih podataka o klijentu u pravilu ovisi o obliku poslovanja s njim, za svakog klijenta neće biti dostupan jednak opseg podataka. Stoga se može očekivati nezanemariv udio neodređenih vrijednosti atributa u podacima i prilikom konstrukcije i pri upotrebi izvedenih modela.
  Osim toga, uz neizbježno postojanje šuma u transakcijskim bazama podataka, integracija podataka iz više izvora često je uzrok dodatnih nekorektnosti u podacima.

Naravno, osim ovih karakterističnih svojstava, konkretni problem može nametnuti i neke nove momente. Međutim, ukoliko se nabrojanim važnim obilježjima tipičnih problema u profiliranju korisnika usluga posveti adekvatna pažnja, mogu se očekivati kvalitetni rezultati primjene tehnika inteligentne analize podataka.
Razvoj klasifikacijskog okruženja prikazanog u prethodnom poglavlju vođen je upravo ovim idejama. Odabir i provedene modifikacije tehnika pripreme i modeliranja podataka, kao i način njihovog nadovezivanja ilustriraju koncept prilagodbe procesa analize podataka karakteristikama ciljnih problema. Konačna arhitektura sustava nije osmišljena odjednom, već je rezultat eksperimentiranja na stvarnim podacima, te analize mogućih poboljšanja i oblika njihove implementacije. Konačan rezultat su respektabilne klasifikacijske performanse na skupini ciljnih problema, čiji je tipičan predstavnik izgradnja profila korisnika usluga.

Razvoj inteligentne analiza podataka kao discipline ni u kojem smislu nije dovršen. To je istraživački još uvijek vrlo aktivno područje, sa širokim i raznovrsnim opsegom istraživačkih tema. Kao i ostale primjene inteligentne analize podataka, i izgradnja profila korisnika usluga može profitirati od rezultata tih istraživanja.
Primjerice, jedan smjer istraživanja usmjeren je na smanjenje vremenskih i prostornih zahtjeva tehnika inteligentne analize podataka u uvjetima glomaznih skupova podataka. Profiliranje korisnika kreće se upravo tim pravcem, jer se brojnost i opseg podataka o korisnicima usluga rapidno povećava s globalizacijom tržišta koju omogućuju moderne komunikacije. Vremenski zahtjevi nastoje se umanjiti npr. paralelizacijom korištenih algoritama, specifičnim tehnikama filtriranja podataka ili upotrebom sofisticiranih heuristika radi smanjenja složenosti pretraživanja. Zahtjeve za memorijskim prostorom moguće je gotovo anulirati razvojem inkrementalnih verzija algoritama modeliranja podataka, kod kojih se slijedno obrađuje jedan po jedan primjer. Gdje takvi algoritmi ne postoje, upotreba memorijskog prostora nastoji se racionalizirati posebno prilagođenim postupcima indeksiranja podataka i

načinom korištenja priručne memorije, sukladno potrebama korištene tehnike modeliranja.
Još jedna od istraživačkih tema čiji bi rezultati mogli biti posebno zanimljivi u izgradnji profila korisnika usluga odnosi se na inkorporiranje metapodataka u proces inteligentne analize podataka. Termin *metapodaci* se odnosi na već postojeće znanje u domeni promatranog problema koje se može na svrsishodan način iskoristiti u postupku indukcije pravilnosti. To mogu biti npr. informacije o specifičnim svojstvima pojedinih atributa (npr. implicitni uređaj, cirkularna skala mjerenja i slično), informacije o semantičkim, kauzalnim i funkcionalnim odnosima među različitim atributima, kao i već poznate pravilnosti relevantne za promatrani problem. Kod poslovanja s korisnicima s vremenom se nedvojbeno dođe do određenih spoznaja o njihovim osobinama, međusobnim sličnostima i razlikama. Iskorištene na adekvatan način, takve informacije značajno ubrzavaju proces profiliranja korisnika, a ponekad mogu imati i presudan utjecaj na uspjeh i kvalitetu rezultata procesa inteligentne analize podataka.
Dakako, i drugi smjerovi istraživanja na ovom području mogu biti korisni za konstrukciju korisničkih profila. Općeniti rezultati su jednako vrijedni za sve primjene inteligentne analize podataka.

Zaključno, izgradnja profila korisnika usluga je iznimno plodno tlo za primjenu tehnika inteligentne analize podataka. Uspjesi u praksi motiv su sve široj primjeni, a razvojem inteligentne analize podataka kao discipline u budućnosti se mogu se očekivati i još bolji rezultati. Unatoč tome, nisu izgledna gotova aplikacijska rješenja koja bi na najbolji način pokrila sve probleme iz domene profiliranja korisnika. Zbog raznolikosti tipova problema i posebnosti koje može donijeti pojedinačni problem, najbolji rezultati se i dalje mogu očekivati od specifičnih, problemu prilagođenih rješenja.

Literatura

[1] R. Agrawal, R.Srikant, "Fast algorithms for mining association rules in large databases", in *Proceedings of the International Conference on Very Large Databases*, Morgan Kaufmann, San Francisco, 1994.

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

[3] M. Anthony, N. Biggs, "Computational learning theory: an introduction", *Cambridge Tracts in Theoretical Computer Science Vol.30*, Cambridge University Press, 1992.

[4] C.G. Atkeson, S.A. Schaal, A.W. More, "Locally weighted learning", *AI Review 11*, 1997.

[5] M.J.A. Berry, G. Linoff, *Data mining techniques for marketing, sales, and customer support*, John Wiley, New York, 1997.

[6] M.J.A. Berry, G. Linoff, *Mastering data mining: The art and science of customer relationship management*, Wiley & Sons, 2000.

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

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

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

[10] P. Cheesman, J. Stutz, "Bayesian classification (AutoClass): Theory and results", *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, 1995.

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

[12] P. Clark, "Knowledge representation in machine learning", *Machine and Hunan Learning, Advances in European Research*, Michael Horwood, London, 1989.

[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] V. Dhar, R. Stein, *Seven methods for transforming corporate data into business intelligence*, Prentice Hall, Upper Saddle River, 1997.

[15] J. Dougherty, R. Kohavi, M. Sahami, "Supervised and unsupervised discretization of continuous features", in *Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, 1995.

[16] B. Efron, R. Tibshirani, *An introduction to the bootstrap*, Chapman and Hall, London, 1993.

[17] U.M. Fayyad, K.B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning", in *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, San Francisco, 1993.

[18] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, *Advances in knowledge discovery and data mining*, AAAI Press/MIT Press, Menlo Park, 1996.

[19] 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.

[20] T. Fulton, S. Kasif, S. Salzberg, "Efficient algorithms for finding multi-way split for decision trees", in *Proceedings of the 12th International Conference on Machine Learning*, Morgan Kaufmann, San Francisco, 1995.

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

[22] D.E. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Reading, 1989.

[23] M.A. Hall, L.A. Smith, "Practical feature subset selection for machine learning", in *Proceedings of the 21st Australasian Computer Science Conference*, Springer, 1998.

[24] M.A. Hall, L.A. Smith, "Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper", in *Proceedings of the 22nd Australasian Computer Science Conference*, 1999.

[25] J. Han, M. Kamber, *Data mining: Concepts and techniques*, Morgan Kaufmann, San Francisco, 2001.

[26] J.A. Hartigan, *Clustering algorithms*, John Wiley, New York, 1975.

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

[28] G.H. John, "Robust decision trees: Removing outliers from databases", in *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, 1995.

[29] G.H. John, *Enhancements to the data mining process*, PhD Dissertation, Computer Science Department, Stanford University, 1997.

[30] G.H. John, P. Langley, "Estimating continuous distributions in Bayesian classifiers", in *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, San Mateo, 1995.

[31] 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.

[32] R. Kerber, "Chimerge: Discretization of numeric attributes", in *Proceedings of the 10th National Conference on Artificial Intelligence*, AAAI Press, Menlo Park, 1992.

[33] R. Kimball, *The data warehouse toolkit*, John Wiley, New York, 1996.

[34] K. Kira, L. Rendell, "A practical approach to feature selection", in *Proceedings of the 9th International Conference on Machine Learning*, Morgan Kaufmann, 1992.

[35] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection", in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, San Francisco, 1995.

[36] R. Kohavi, G.H. John, "Wrappers for feature subset selection", *Artificial intelligence 97*, 1997.

[37] R. Kohavi, F. Provost, "Machine learning: Special issue on applications of machine learning and the knowledge discovery process", *Machine Learning 30*, 1998.

[38] R. Kohavi, M. Sahami, "Error-based and entropy-based discretization of continuous features", in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, 1996.

[39] I. Kononenko, "Estimating attributes: Analysis and extensions of Relief", in *Proceedings of the 7th European Conference on Machine Learning*, Springer-Verlag, 1994.

[40] P. Langley, H.A. Simon, "Applications of machine learning and rule induction", *Communications of the ACM 38*, 1995.

[41] P. Langley, W. Iba, K. Thompson, "An analysis of Bayesian classifiers", in *Proceedings of the 10th National Conference on Artificial Intelligence*, AAAI Press, Menlo Park, 1992.

[42] H. Liu, R. Setiono, "Feature selection via discretization", *IEEE Transactions on Knowledge and Data Engineering 9*, 1997.

[43] R.S. Michalski, "A theory and methodology of inductive learning", in *Machine Learning, an Artificial Intelligence Approach, Vol.1*, Morgan Kaufmann, San Mateo, 1983.

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

[45] G. Piatetsky-Shapiro, "Discovery, analysis, and presentation of strong rules", in *Knowledge Discovery in Databases*, AAAI Press, Menlo Park, 1991.

[46] F. Provost, T. Fawcett, "Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions", in *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, 1997.

[47] P. van der Putten, M. van Someren, "CoIL challenge 2000: The insurance company case", *Leiden Institute of Advanced Computer Science Technical Report 2000-09*, Netherlands, 2000.

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

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

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

[51] J. Rissanen, "The minimum description length principle", in *Encyclopedia of Statistical Sciences, Vol.5*, John Wiley, New York, 1985.

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

[53] C. Sammut, R.B. Banerji, "Learning concepts by asking questions", in *Machine Learning, an Artificial Intelligence Approach, Vol.2*, Morgan Kaufmann, San Mateo, 1986.

[54] D.C. Sills, "William of Ockham", in *International Encyclopedia of the Social Sciences*, Macmillan Company & The Free Press, New York, 1968.

[55] F. Ujevic, N. Bogunovic, "An integrated intelligent system for data mining and decision making", in *Proceedings of the 6th International Conference on Intelligent Engineering Systems*, 2002.

[56] F. Ujevic, N. Bogunovic, "Enhancing the efficiency of data mining process by circular coupling of multiple schemes", in *Proceedings of the MIPRO 2002 International Convention*, 2002.

[57] L.G. Valiant, "A theory of the learnable", *Communications of the ACM 27*, 1984.

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

Životopis

Filip Ujević

Rođen sam 07. prosinca 1974. godine u Splitu. Osnovnu školu pohađao sam u Krivodolu, a matematičku gimnaziju sam završio u srednjoj školi dr. Mate Ujevića u Imotskom.

Na Prirodoslovno-matematički fakultet Sveučilišta u Zagrebu upisao sam se 1993. godine. Zvanje diplomiranog inženjera matematike stekao sam 1997. godine, kada sam diplomirao na Matematičkom odjelu, kao prvi u klasi studenata smjera Računarstvo. Dodjela stipendije Sveučilišta u Zagrebu potvrda je izvrsnog uspjeha tijekom studija. Znanstveno i stručno usavršavanje nastavio sam na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu, gdje sam upisao poslijediplomski studij na smjeru Primijenjeno računarstvo. Objavio sam više radova s područja strojnog učenja i inteligentne analize podataka.

Od 1998. godine zaposlen sam u Sektoru za informatiku Croatia osiguranja, na poslovima projektiranja i izgradnje informacijskih sustava. Bogato praktično iskustvo stekao sam sudjelovanjem i vođenjem više projekata. U područje mog užeg interesa spadaju sustavi za poslovno izvještavanje i podršku odlučivanju, gdje sam vodio i uspješno dovršio projekte koji su danas u produkcijskom radu. Radovima i izlaganjima aktivno sudjelujem na raznim stručnim skupovima i seminarima.

Sažetak

Suvremene računalne i telekomunikacijske tehnologije omogućile su jednostavno prikupljanje i jeftinu pohranu ogromnih količina podataka. U poslovnom svijetu prikupljeni podaci su uglavnom transakcijskog oblika, a najveći dio nastaje bilježenjem interakcija s korisnicima usluga. Pravi potencijal tako izgrađenih baza podataka nije u sirovim transakcijskim podacima, već u implicitnim pravilnostima koje su u njima skrivene. U današnjem svijetu globalne konkurencije veliku ulogu u stjecanju i održavanju tržišnog udjela ima upravo pronalaženje, razumijevanje i iskorištavanje pravilnosti u obilježjima i ponašanju korisnika usluga.

Postupci inteligentne 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 inteligentne analize podataka, opisani su temeljni problemi u njihovoj primjeni, kao i načini rješavanja tih problema. Prikazane su najpopularnije tehnike svake od faza procesa inteligentne analize podataka, te su naznačena njihova svojstva i međusobne razlike. Posebice, rad istražuje adekvatnost, efikasnost i uspješnost primjene opisanih postupaka inteligentne analize podataka na stvarne probleme iz poslovnog svijeta. Analiziraju se konkretni podaci na tipičnom problemu profiliranja korisnika iz domene osiguravajućih društava. Provedena razmatranja rezultirala su specifičnom kombinacijom tehnika probabilističkog i deskriptivnog modeliranja, koja je prvenstveno namijenjena problemima s izraženim nedeterminističkim obilježjima. Na tim osnovama je izgrađen i programski produkt koji implementira postupke analize podataka u kontekstu promatranih problema.

Ključne riječi

inteligentna analiza podataka,
otkrivanje znanja,
strojno učenje,
profili korisnika usluga,
stvarni poslovni problemi,
tehnike modeliranja,
neizvjesnost

Data mining techniques in constructing customer profiles

Summary

Contemporary information and communication technology facilitates data acquisition and provides affordable storage for vast quantities of data. Business world resides on transactional data, and the largest part of transactional data comes from recording interactions with customers. The true potential of transactional databases is not in raw transactional data, but in implicit regularities that underlie the data. In the modern world of global competition, discovering, understanding and exploiting patterns in customer characteristics and behavior has an important role in obtaining and maintaining substantial market share.

Data mining techniques enable automatic extraction of implicit regularities and hidden relationships that exist in large databases. This thesis explicates fundamental principles and ideas that form the basis of data mining techniques, explains essential problems in their practical application, and offers standard solutions to these problems. The most popular techniques for each phase of data mining process are also described, along with their properties and important differences. Particularly, the thesis explores adequacy, efficiency and usefulness of practical application of the described data mining techniques on real-world business problems. As a typical example, a problem of constructing customer profiles is taken from the insurance business. Conducted analysis and experiments resulted in a specific combination of probabilistic and descriptive modeling techniques, primarily suited for problems that exhibit a high degree of undeterministic behavior. Derived from these results, a computer application was developed, implementing data mining techniques in the context of target problem type.

Keywords

data mining,
knowledge discovery,
machine learning,
customer profiles,
real-world business problems,
modeling techniques,
uncertainty

BESPLATNI GOTOVI SEMINARSKI, DIPLOMSKI I MATURSKI RAD.

RADOVI IZ SVIH OBLASTI, POWERPOINT PREZENTACIJE I DRUGI EDUKATIVNI MATERIJALI.

WWW.SEMINARSKIRAD.ORG

WWW.MATURSKIRADOVI.NET

WWW.MATURSKI.NET

WWW.SEMINARSKIRAD.INFO

WWW.MATURSKI.ORG

WWW.ESSAYSX.COM

WWW.FACEBOOK.COM/DIPLOMSKIRADOVI

NA NAŠIM SAJTOVIMA MOŽETE PRONAĆI SVE, BILO DA JE TO SEMINARSKI, DIPLOMSKI ILI MATURSKI RAD, POWERPOINT PREZENTACIJA I DRUGI EDUKATIVNI MATERIJAL. ZA RAZLIKU OD OSTALIH MI VAM PRUŽAMO DA POGLEDATE SVAKI RAD, NjEGOV SADRŽAJ I PRVE TRI STRANE TAKO DA MOŽETE TAČNO DA ODABERETE ONO ŠTO VAM U POTPUNOSTI ODGOVARA. U BAZI SE NALAZE GOTOVI SEMINARSKI, DIPLOMSKI I MATURSKI RADOVI KOJE MOŽETE SKINUTI I UZ NJIHOVU POMOĆ NAPRAVITI JEDINSTVEN I UNIKATAN RAD. AKO U BAZI NE NAĐETE RAD KOJI VAM JE POTREBAN, U SVAKOM MOMENTU MOŽETE NARUČITI DA VAM SE IZRADI NOVI, UNIKATAN SEMINARSKI ILI NEKI DRUGI RAD RAD NA LINKU IZRADA RADOVA. PITANjA I ODGOVORE MOŽETE DOBITI NA NAŠEM FORUMU ILI NA MATURSKIRADOVI.NET@GMAIL.COM