

Mladen Nikolić

Andželka Zečević

MAŠINSKO UČENJE

Beograd
2018.

Sadržaj

Sadržaj	2
1 Uvod	5
I Nadgledano učenje	9
2 Teorijske osnove nadgledanog učenja	11
2.1 Postavka problema nadgledanog učenja	12
2.2 Princip minimizacije empirijskog rizika	13
2.3 Preprilagođavanje	18
2.4 Regularizacija	21
2.5 Nagodba između sistematskog odstupanja i varijanse	24
2.6 Teorijske garancije kvaliteta generalizacije	28
2.7 Veza statističke teorije učenja sa filozofijom nauke	35
2.8 Vrste modela	37
2.9 Dimenzije dizajna algoritama nadgledanog učenja	39
3 Probabilistički modeli	41
3.1 Linearna regresija	41
3.2 Logistička regresija	48
3.3 Multinomijalna logistička regresija	51
3.4 Uopšteni linearni modeli	53
3.5 Naivni Bajesov algoritam	54
4 Modeli zasnovani na širokom pojasu	57
4.1 Metod potpornih vektora za klasifikaciju	57
4.2 Metod potpornih vektora za regresiju	64
4.3 Algoritam k najbližih suseda zasnovan na širokom pojasu	67
5 Modeli zasnovani na instancama	71
5.1 Osnove neparametarske ocene gustine raspodele	71
5.2 Metodi zasnovani na kernelima	75
5.3 Metodi zasnovani na najbližim susedima	83

5.4	Algoritam k najbližih suseda	83
6	Evaluacija i izbor modela	89
6.1	Mere kvaliteta modela	89
6.2	Tehnike evaluacije i izbora modela	95
6.3	Napomene vezane za pretprocesiranje	99
7	Regularizacija	103
7.1	Proredeni modeli	103
7.2	Modeli složenije strukture i uključivanje domenskog znanja . . .	107
7.3	Učenje više poslova odjednom	109
8	Optimizacija	113
8.1	Gradijentni spust	113
8.2	Metod inercije	117
8.3	Nestorovljev ubrzani gradijentni spust	117
8.4	Adam	117
8.5	Stohastički gradijentni spust	119
9	Neuronske mreže i duboko učenje	123
9.1	Neuronske mreže sa propagacijom unapred	124
9.2	Konvolutivne neuronske mreže	132
II	Dodatak	143
10	Matematičko predznanje	145
10.1	Sopstvene vrednosti i sopstveni vektori	145
10.2	Definitnost	146
10.3	Norma i skalarni proizvod	147
10.4	Izvod, parcijalni izvod i gradijent	148
10.5	Konveksnost	149
10.6	Lokalni optimumi	150
10.7	Integral	151
10.8	Verovatnoća	152
10.9	Sredina i rasipanje slučajne promenljive	154
10.10	Statističke ocene i njihova svojstva	156
10.11	Statistički modeli	157
10.12	Metod maksimalne verodostojnosti	157

Na korisnim sugestijama zahvaljujemo se kolegi Milošu Jovanoviću i studentima Milošu Stankoviću, Blagoju Ivanoviću, Nemanji Mićoviću...

Glava 1

Uvod

Od početka dvehiljaditih, razvoj veštačke inteligencije je dobio nov zamah. Niz izrazito važnih problema, od kojih se za neke pretpostavljalo da će još dugo ostati van domašaja, biva rešen. U nekim domenima, u kojima računari do tada po uspešnosti nisu mogli da se porede sa ljudima, postižu se rezultati superiorni u odnosu na rezultate ljudskih eksperata. U srcu ovog novog zamaha, nalazi se *mašinsko učenje*. Iako u prvi plan izbjija upravo dvehiljaditih, ova oblast ima dugu istoriju razvoja. Zamišljena u radovima Alena Tjuringa, četrdesetih godina prošlog veka, aktivno se razvija od pedesetih kada se razvija *perceptron* prvi sistem koji uči jednostavne zakonitosti i predstavlja dalekog preteču modernih *neuronskih mreža* koje se uz uspone i padove razvojavaju do devedesetih, kada primat uzimaju *metod potpornih vektora* i drugi metodi zasnovani na *kernelima*. Ipak, za skorašnji uspon mašinskog učenja, zasluzna je baš renesansa neuronskih mreža koja je dovela do toga da se danas veštačka inteligencija i mašinsko učenje u opštoj percepciji neretko poistovjećuju sa njima. Ipak, ove oblasti su neuporedivo šire.

Dugi razvoj mašinskog učenja motivisan je s jedne strane željom da se bolje razume ljudski i životinjski potencijal za učenje, koji se nalazi u srcu onoga što nazivamo inteligencijom, a s druge, željom da se takav proces oponaša u praktične svrhe. Ove dve motivacije verovatno u značajnoj meri korespondiraju i sa dva žarišta razvoja mašinskog učenja – akademskim svetom u kojem je poniklo i u kojem je dovedeno od određenog nivoa upotrebljivosti i privredom koja je u njemu prepoznala potencijal za praktične primene i daje ogroman doprinos njegovom razvoju u toku dvehiljaditih godina. Precizno definisanje naučnih disciplina, nezahvalan je i neizgledan, a možda i nepotreban poduhvat. Zato mu i nećemo posvetiti mnogo pažnje. U opštoj percepciji mašinsko učenje predstavlja disciplinu koja se bavi konstrukcijom sistema koji se prilagođavaju i popravljaju svoje performanse sa povećanjem iskustva, oličenog u količini relevantnih podataka. Ovaj pogled naglašava njegovu praktičnu stranu. Ipak, mašinsko učenje ima i svoju fundamentalnu dimenziju. Kao što se logika bavi proučavanjem *dedukcije* objašnjavajući šta čini neki zaključak potpuno opravdanim i time formalizuje jedan važan vid ljudskog zaključivanja, mašinsko

učenje se bavi proučavanjem *indukcije*, odnosno *generalizacije* i time formalizuje drugi vid ljudskog zaključivanja – uopštavanje od ograničenog broja uzoraka ka univerzalnim zaključcima. Ovaj drugi problem se može smatrati i težim. Osnove dedukcije razumeo je (uprkos nekim propustima) Aristotel pre više od dve hiljade godina. Indukcija se ozbiljnije izučava tek od strane Frencisa Bejkona na prelazu sa šesnaestim na sedamnaestim vekom. Deduktivno zaključivanje se, kroz formalnu logiku proučava od devetnaestog veka i početkom dvadesetog veka, već je na čvrstim nogama. Induktivno zaključivanje se, kroz statističku teoriju učenja, u nekoj meri formalizuje tek krajem dvadesetog veka.

Kako i deduktivno i induktivno zaključivanje imaju važnu ulogu u prirodnoj inteligenciji, odgovarajuće discipline – automatsko rezonovanje i mašinsko učenje imaju važne uloge u veštačkoj inteligenciji. Postavlja se pitanje, kada su metode koje od ovih oblasti pogodniji izbor za rešavanje konkretnog problema. Metode zasnovane na logici, koje se razvijaju u okviru automatskog rezonovanja, pogodne su u slučajevima u kojima je problem moguće precizno matematički definisati. Obično se radi o problemima koje čovek može relativno lako da formuliše, ali ih vrlo teško rešava (najčešće zbog kombinatorne eksplozije pri pretrazi prostora mogućih rešenja) i u kojima nisu prihvativljiva pogrešna rešenja. S druge strane, mašinsko učenje je posebno pogodno upravo za suprotnu vrstu problema – probleme koje čovek ne može lako ni da definiše, iako neke od njih čak vrlo lako rešava (neke, s druge strane, ne) i u kojima je prihvativljiva povremena greška. Jedan primer takvog problema je prepoznavanje lica. Osim u slučajevima specifičnih neuroloških poremećaja, svi ljudi su vrlo dobri u rešavanju ovog problema. Čak je neobično nazvati ga problemom i govoriti o njegovom rešavanju. Ipak, ukoliko pokušamo da taj problem precizno definišemo, naletimo na mnoštvo problema. Prvi naivni pokušaj definisanja bi se verovatno sastojao u nekom opisu poput toga da je lice nešto što se sastoji od nosa, očiju, usta čela, jagodica i obrva. Ovo ne samo što vodi daljem pitanju definisanja tih pojmoveva, već postavlja i pitanje definisanja njihovih relativnih pozicija i slično i uprkos trudu, bićemo prinuđeni da odustanemo. Stoga se ovakvim problemima ne pristupa metodama automatskog rezonovanja. S druge strane, kao i ljudi, metode mašinskog učenja mogu vrlo uspešno da se nose sa ovim problemom.

Neki od problema na koje je mašinsko učenje uspešno primenjeno su prepoznavanje lica na slikama, prepoznavanje različitih objekata na slikama i videu, prepoznavanje tumora na medicinskim snimcima, autonomna vožnja automobila, autonomno letenje, igranje igara na tabli poput šaha i igre go, ali i računarskih igara kao što je Super Mario ili Doom, klasifikacija teksta, mašinsko prevođenje, automatsko opisivanje sadržaja slika, analiza osećanja izraženih u tekstu, predviđanje razvoja bolesti kod pacijenata i preporučivanje terapije, analiza društvenih mreža, prepoznavanje i sinteza govora i tako dalje. U mnogim od ovih primena, mašinsko učenje je već prevazišlo nivo efikasnosti ljudskih eksperata. Sve nabrojane primene predstavljaju očigledno primere važnih praktičnih problema, ali iza svih stoji i ozbiljna teorija. Možda je upravo ovaj spoj ključ uspeha mašinskog učenja.

Nabrojani problemi su vrlo raznorodni kako po svojoj prirodi, tako i po metodama mašinskog učenja koje se koriste za njihovo rešavanje. Te metode se mogu razvrstati na mnogo načina, ali osnovna podela bi bila prema prirodi problema učenja. Obično se identificuju tri grupe problema mašinskog učenja, a to su problemi *nadgledanog učenja* (eng. *supervised learning*), problemi *nенадгледаног учења* (eng. *unsupervised learning*) i problemi *учења поткрепљивањем* (eng. *reinforcement learning*).¹

Nadgledano učenje je verovatno najznačajniji vid mašinskog učenja. Osnovna karakteristika mu je da se podaci sastoje iz parova opisa onoga na osnovu čega se uči i onoga što je iz toga potrebno naučiti. Naziv je motivisan analogijom sa procesom učenja pri kojem učitelj učeniku zadaje zadatke, ali mu nakon njegovih odgovora, radi poređenja, daje i tačna rešenja. Problem prepoznavanja lica na slici bi zajedno sa slikom uključivao i informaciju da li se na slici nalazi lice ili ne. Problem prepoznavanja objekata bi uz sliku uključivao i spisak objekata koji se na njoj nalaze, eventualno i sa njihovim pozicijama. Prilikom mašinskog prevođenja, zajedno sa rečenicom u polaznom jeziku, bila bi data i rečenica na ciljnem jeziku.

Nenadgledano učenje se karakteriše upravo nedostatkom informacije o tome šta je potrebno naučiti. Na prvi pogled, ovakve metode mogu zvučati ili nemoguće ili neuporedivo moćnije od metoda nadgledanog učenja. Kakva je to metoda koja je u stanju da nauči šta treba, a da joj nije rečeno šta treba? Navezno, to je metoda kojoj nedostaje opštost. Metoda pomoću koje se ne može učiti bilo šta, već isključivo nešto za što je metoda eksplicitno dizajnirana. Zapravo, ovaj vid učenja se obično bavi pronalaženjem neke vrste strukture u podacima, a metode koje na taj način uče su obično napravljene polazeći od konkretnе vrste strukture koja se traži. Problem *klasterovanja* je jedan problem nalaženja strukture u podacima – strukture grupa. U mnogim primenama potrebno je identifikovati grupe podataka. Primera radi, grupisati se mogu slični tekstovi, slične slike, akcije čije se cene slično kreću na berzi i tako dalje. U poslednjem primeru, razlog za grupisanje bi recimo mogao biti taj što je za jednu akciju dostupna relativno kratka istorija cena, koja ne omogućava obučavanje modela nadgledanog učenja koji će predviđati buduće cene. Međutim, ako se veliki broj akcija čije se cene slično ponašaju grupiše, podataka može biti dovoljno. Drugi vid pronalaženja strukture u podacima bi bilo učenje reprezentacije podataka. Na primer, podaci koji su visoko dimenzionalni, često zapravo leže u linearном potprostoru ili na nekoj površi značajno manje dimenzije. Izražavanjem polaznih podataka u terminima koordinata u takvim prostorima, smanjuje se njihova dimenzionalnost, što često dovodi do dosta boljih performansi algoritama nadgledanog učenja koji se primenjuju na tim podacima. U oba slučaja, pomenuto je da se nakon primene metode nenadgledanog učenja, na te podatke primenjuju metode nadgledanog učenja. To nije uvek slučaj, ali neretko metode nenadgledanog učenja mogu biti vid

¹Inspiracija za poslednji prevod uzeta je iz psihologije u kojoj se tako zove odgovarajući način učenja kod životinja i ljudi.

preprocesiranja podataka kako bi se na njima primenile metode nadgledanog učenja.

Učenje potkrepljivanjem je, neformalno rečeno, između prethodna dva posmenuta pristupa. Koristi se u situacijama u kojima je potrebno rešiti neki problem preduzimajući niz akcija, čijim se zajedničkim dejstvom dolazi do rešenja problema. Pretpostavlja se da postoji *agent* (odnosno, neko ko dela) koji opaža tekuće *stanje okruženja*, u mogućnosti je da preduzima *akcije* usled kojih dobija *nagrade* predstavljene numeričkom vrednošću. Ishod učenja je *optimalna politika*, odnosno preslikavanje stanja u akcije koje vodi maksimalnoj (ili, u praksi, dovoljno visokoj) ukupnoj nagradi. Pritom, ključna je prepostavka da nije poznato koja od preduzetih akcija je bila prava u datom kontekstu, a koja nije. U suprotnom, radilo bi se o problemu nadgledanog učenja. Primera radi, razmotrimo problem autonomne vožnje. Agent je sistem koji vozi automobil i koji je u stanju da opaža pozicije drugih automobila, pešaka, saobraćajne znake, svetla semafora i slično (a što čini okruženje), a koji je u stanju da menja smer kretanja i da povećava i smanjuje brzinu kretanja automobila (što su akcije). Agent pritom dobija nagrade koje su recimo 1 za svaki kilometar pređenog puta, 100 za stizanje na cilj, -100 u slučaju sudara i -1000 u slučaju smrtnog ishoda u saobraćaju. Optimalnu politku nije lako opisati na osnovu ličnog znanja (što je tipičan razlog za upotrebu mašinskog učenja!), ali ona bi verovatno uključivala kočenje na žutom i crvenom svetlu ili pred pešacima, skretanje tamo gde je znakom naznačeno da je obavezno skrenuti itd.

U daljem tekstu, najviše pažnje biće posvećeno nadgledanom učenju, kako zbog njegove najšire primene, tako i zbog razvijenosti fundamentalne teorije.

Deo I

Nadgledano učenje

Glava 2

Teorijske osnove nadgledanog učenja

Kao što je rečeno, nadgledano učenje se karakteriše time da su uz vrednosti ulaza, date i vrednosti izlaza koje im odgovaraju. Potrebno je ustanoviti odnos koji važi između ulaza i izlaza. Na osnovu ovog odnosa se najčešće za neke buduće ulaze vrši predviđanje izlaza. Ulaz i izlaz se najčešće predstavljaju u vektorskom obliku i označavaju sa x i y , pri čemu je x tipično vektor vrednosti nekih promenljivih koje se nazivaju *atributima* (eng. *features*), dok je y tipično jedna promenljiva koja se naziva *ciljnom promenljivom* (eng. *target variable*). Mogući su i mnogo opštiji scenariji. Na primer oni u kojima je y takođe višedimenzionalno, ali i oni u kojima ni x ni y nisu predstavljeni numeričkim vrednostima, već mogu predstavljati sekvene, grafove i slično.

Problemu otkrivanja veze između nekih promenljivih na osnovu opažanja može se pristupiti na različite načine. Na primer, već hiljadama godina naučnici na osnovu opažanja postavljaju hipoteze o odnosima nekih veličina, a onda predviđanja dobijena na osnovu tih hipoteza testiraju u praksi i na osnovu toga odlučuju o verodostojnosti tih hipoteza. Jedan primer takvog odnosa je formula $F = ma$ koja uspostavlja vezu između sile, mase i ubrzanja. Do ove formule se došlo ljudskim uvidom, a na osnovu opažanja iz iskustva. Ovakav pristup je moguć i uobičajen pod prepostavkom da fenomen i interakcije među razmatranim veličinama nisu previše komplikovani za ljudsko poimanje. Ipak, u vreme kada je uobičajeno da raspolaćemo gigabajtima, pa i terabajtima podataka koji predstavljaju milione ili milijarde opažanja sa desetinama hiljada promenljivih, potrebne su metode koje su u stanju da uočavaju takve veze automatski.

Ovakve metode obično nalaze funkcije koje na neki način izražavaju vezu između vrednosti atributa i vrednosti ciljne promenljive. Ove funkcije i njihova svojstva su od centralnog značaja u mašinskom učenju i nazivaju se *modelima*. Modela može biti (beskonačno) mnogo, ali ne možemo od svih, pa čak ni od jednog, očekivati da savršeno opisuje zavisnosti koje važe među promenljivim. Ono što se od modela očekuje je da dobro *generalizuje*, odnosno da prilikom

predviđanja vrednosti ciljne promenljive na osnovu vrednosti atributa, retko pravi velike greške. Idealan slučaj u kojem grešaka nema, nije realističan i ne dešava se u praksi. Pojam generalizacije je centralni pojam mašinskog učenja i biće mu posvećena posebna pažnja.

Dve osnovne vrste problema nadgledanog učenja su *regresija* i *klasifikacija*. Regresija je problem predviđanja neprekidne ciljne promenljive. Na primer, moguće je predviđati cenu deonica na berzi na osnovu njihovih cena u pret-hodnih nekoliko dana i globalnih kvantitativnih pokazatelja tržišta ili količinu teških metala u zemljištu na osnovu udaljenosti od zagadivača, udaljenosti od vodenih tokova, vrste zemljišnog pokrivača i slično. Klasifikacija je problem predviđanja kategoričke ciljne promenljive. Kategoričkim se smatraju promenljive koje uzimaju konačan broj vrednosti među kojima nema uređenja. Na primer, prepoznavanje jedne iz skupa poznatih osoba čije se licne nalazi na slici je problem klasifikacije. Prepoznavanje da li se novinski članak tiče ekonomije, sporta ili politike je takođe problem klasifikacije.

2.1 Postavka problema nadgledanog učenja

O vrednostima atributa se može razmišljati kao o okolnostima u kojima nastaje neki ishod koji je predstavljen vrednošću ciljne promenljive. Na primer, ukoliko je dan letnji u ukoliko nema oblačnosti, očekuje se da je temperatura visoka. Ipak, dva merenja temperature po sunčanom letnjem danu se mogu značajno razlikovati. Na primer, zbog razlike u geografskoj širini, udaljenosti od vodenih površina, ili kretanja hladnijih vazdušnih masa. Može se očekivati da uključivanjem većeg broja promenljivih u atribute može biti uočena jača veza sa ciljnom promenljivom. Ipak, u praksi nije realistično da se mogu identifikovati i adekvatno kvantifikovati svi faktori koji igraju ulogu u nekom fenomenu, pa čak i ako fenomen deluje jednostavno. Zbog toga, možemo očekivati da u podacima za iste vrednosti atributa vidimo različite vrednosti ciljne promenljive. Očito, opis veze između atributa i ciljne promenljive se može utemeljiti na pojmovima verovatnoće.

U najopštijem slučaju, pretpostavlja se da je odnos između atributa i ciljne promenljive zadat zajedničkom raspodelom verovatnoće $p(x, y)$. Najčešće, pa i sada, ćemo pod ovim podrazumevati gustinu raspodele. Poznavanje ovog verovatnosnog zakona među promenljivim od interesa predstavlja potpuno znanje o njihovim odnosima. Kako takva raspodela nije dostupna, pristupa se određivanju modela $f(x)$ koji vrednostima atributa pridružuje vrednost ciljne promenljive. Takvih modela može biti puno, ali od značaja je u nekom smislu najbolji takav model. Ipak, pojam kvaliteta je potrebno definisati. Poželjno je da važi $y \approx f(x)$, odnosno da je razlika između pravih vrednosti ciljne funkcije i njihove aproksimacije modelom mala. Otud je prvo potrebno definisati *funkciju greške* (eng. *loss*) koja meri odstupanje predviđenih i stvarnih vrednosti ciljne promenljive. Ovu funkciju ćemo označavati sa L . Ipak $L(y, f(x))$, predstavlja razliku jedne prave vrednosti i jednog predviđanja. Bilo kog, ali pojedinačno.

Ni jedno konkretno x i y nisu dovoljni da opišu kvalitet modela. Umesto toga, potrebno je izraziti kvalitet modela po svim parovima x i y . Greška $L(y, f(x))$ nastaje kada je za neke vrednosti atributa x prava vrednost ciljne promenljive y . Ipak, neke kombinacije x i y su vrlo verovatne, a neke nisu, a ne moraju biti ni moguće. Stoga, postavlja se pitanje da li su baš sve kombinacije vrednosti atributa i ciljne promenljive podjednako važne. Naravno, nisu. Utoliko je važnija greška koja se dešava na verovatnjim kombinacijama. Otud se gustina raspodele ovih kombinacija $p(x, y)$ prirodno koristi za uprosečavanje grešaka, čime se definiše *funkcional rizika*¹ ili *stvarni rizik* ili samo *rizik*:

$$R(f) = \mathbb{E}[L(y, f(x))] = \int L(y, f(x))p(x, y)dxdy$$

Smisao rizika je da predstavlja grešku uprosečenu po svim mogućim podacima.

Kada je pojmom rizika definisan (ne)kvalitet modela, prirodno je tražiti funkciju koja minimizuje rizik, odnosno rešavati problem

$$\min_f R(f)$$

Ovo je *problem nadgledanog učenja* u svom teorijskom obliku. Ipak, ovaj problem nije moguće direktno rešiti u datom obliku iz dva razloga. Prvi je što je gustina raspodele $p(x, y)$ nedostupna. Taj problem je fundamentalan i ne može se ignorisati ni u teorijskim razmatranjima. Drugi problem se tiče skupa funkcija kojem pripada funkcija f . Teorijski, može se razmatrati skup svih mogućih funkcija iz \mathbb{R}^n u \mathbb{R} , gde je n broj atributa (sada i zauvek u ovoj knjizi!), ali takav scenario nije od praktičnog značaja, pošto je pitanje kako bi se rešenje tako formulisanog problema uopšte moglo izraziti na upotrebljiv način.

2.2 Princip minimizacije empirijskog rizika

Svaki metod mašinskog učenja počiva na nekim prepostavkama o skupu potencijalnih modela. Nekad su ove prepostavke implicitne u konstrukciji metoda, a nekad su eksplicitne. Nekad i korisnik ima mogućnost ograničenog izbora tih prepostavki. U nastavku prepostavljamo da postoji određena *repräsentacija modela* kojom je definisan skup svih modela. Ona će najčešće biti parametarska, odnosno model je funkcija $f_w(x)$ koja zavisi od parametara w . Izborom ovih parametara dobijaju se modeli različitih kvaliteta. Otud se nadalje minimizacija po skupu svih funkcija svodi na minimizaciju po skupu svih vrednosti parametara i problem postaje:

$$\min_w R(f_w(x))$$

Predstavljajući rizik kao funkciju parametara, to možemo pisati i kao

$$\min_w R(w)$$

¹Funkcional je preslikavanje vektorskog prostora u skup njegovih skalara. Primera radi, integral integrabilnim funkcijama pridružuje vrednosti iz njihovog kodomena.

Primer reprezentacije modela je recimo *linearna reprezentacija*, koja prepostavlja linearnost modela *po parametrima*. Tipičan linearni model se može zapisati na sledeći način:

$$f_w(x) = w_0 + \sum_{i=1}^n w_i x_i$$

Predstavljanje slobodnog člana w_0 se često izbegava tako što se prepostavi da je vrednost prvog atributa uvek 1.²

Drugi pomenuti problem, problem nedostupnosti gustine raspodele $p(x, y)$ se rešava aproksimacijom na osnovu uzorka

$$\mathcal{D} = \{(x_i, y_i) | i = 1, \dots, N\}$$

Time se rizik zamenjuje *empirijskim rizikom*:

$$E(w, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^N L(y_i, f_w(x_i))$$

koji ćemo ubuduće nazivati *prosečnom greškom* ili samo *greškom*, osim u situacijama u kojima je potrebno naglasiti razliku u odnosu na stvarni rizik. Kad god nije naveden, skup podataka \mathcal{D} se podrazumeva.

Sada je moguće formulisati *princip minimizacije empirijskog rizika* (eng. *empirical risk minimization principle*) na kojem se tipično zasnivaju algoritmi nadgledanog mašinskog učenja – funkcija koja minimizuje prosečnu grešku $E(w, \mathcal{D})$ se uzima za aproksimaciju funkcije koja minimizuje rizik $R(w)$.

Treba imati u vidu da ovaj princip ne sledi logičkom nužnošću iz osnovne postavke problema nadgledanog učenja. Stoga je potrebno zapitati se kakva su svojstva ovog principa, odnosno da li vodi dobroj aproksimaciji funkcije koja minimizuje stvarni rizik. Pre svega, da li sa povećanjem slučajnog uzorka dolazi do konvergencije rešenja koje ovaj princip nudi ka optimalnom teorijskom rešenju. Ako to ne važi u opštem slučaju, bitno je znati u kojim slučajevima važi. Kako je rizik definisan kao očekivanje, a empirijski rizik kao prospekt i kako znamo da prospekti teže očekivanjima, kad veličina slučajno izabranog uzorka raste, u iskušenju smo da pomislimo da je odgovor lak i potvrdan. Ipak, to što pomenuto svojstvo važi, ne znači da minimum aproksimacije funkcionala mora uvek da teži minimumu funkcionala. Odnosno činjenica da važi

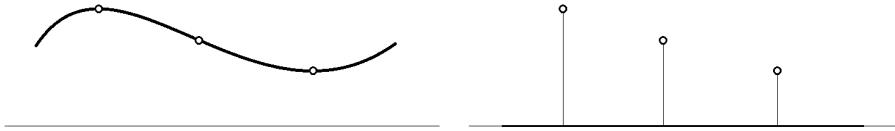
$$E(w, \mathcal{D}) \rightarrow R(w) \text{ kad } (|\mathcal{D}| \rightarrow \infty)$$

ne znači da važi

$$\operatorname{argmin}_w E(w, \mathcal{D}) \rightarrow \operatorname{argmin}_w R(w) \text{ kad } (|\mathcal{D}| \rightarrow \infty)$$

jer se prvo svojstvo odnosi na bilo koju, ali fiksiranu vrednost parametara w , istu i za E i za R , dok se drugo odnosi na potencijalno različite vrednosti

²U praksi se vektori podataka često eksplisitno proširuju jedinicom.



Slika 2.1: Primer modela koji pokazuje da minimum empirijskog rizika (desno) ne mora da teži minimumu stvarnog rizika (levo). Za svaki uzorak, može se konstruisati funkcija poput funkcije prikazane desno i koja se razlikuje od optimalne skoro svuda.

parametara w jer E i R ne moraju dostizati minimum u istoj tački. Ipak, poželjno je demonstrirati da drugo svojstvo u nekom slučaju zaista ne važi. Takav primer je dat na slici 2.1. Prepostavlja se da je skup svih modela skup svih mogućih funkcija f takvih da važi $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Leva slika prikazuje optimalnu funkciju f^* u odnosu na stvarni rizik, pri čemu se prepostavlja da je $R(f^*) = 0$ (prema grafiku važi $y_i = f(x_i)$ za svaku vrednost $i = 1, \dots, N$), dok desna prikazuje funkciju \hat{f} koja minimizuje empirijski rizik. Ona svakoj tački x_i pridružuje baš vrednost y_i , ali je u svim ostalim tačkama jednaka 0. Očito važi $E(\hat{f}) = 0$. Za svaki skup podataka \mathcal{D} funkcije f^* i \hat{f} se poklapaju na skupu podataka, ali svaki skup podataka je mera nula, što znači da se razlikuju skoro svuda, odnosno da konvergencija ne mora da važi.

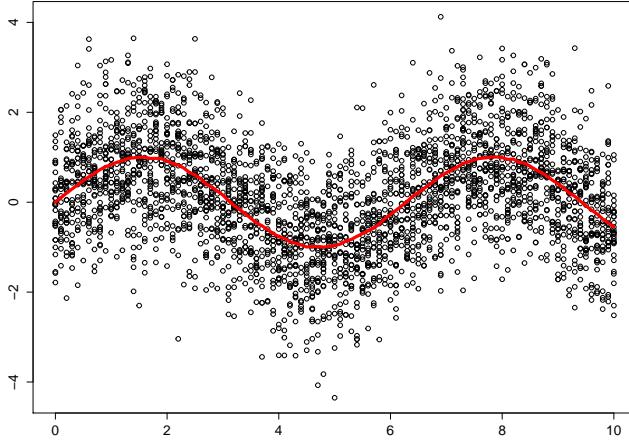
To zavisi od svojstava skupa funkcija po kojem radimo minimizaciju. Odgovor na ovo pitanje je netrivijalan. Proučavanjem ovakvih problema se bavi *statistička teorija učenja*. Na kraju ovog dela će biti skiciran odgovor na ovo pitanje, ali je detaljna razrada van okvira ove knjige.

Pored prethodnih teorijskih pitanja, potrebno je dati odgovor i na jedno krajnje praktično – kako se rešava problem minimizacije prosečne greške modela? Odgovor u najvećem broju slučajeva nude metode matematičke optimizacije, o kojima će biti reči kasnije. Za sad je dovoljno prepostaviti da postoji metod za rešavanje problema minimizacije. Rešavanje takvog problema se naziva *obučavanjem modela* na datom skupu primera za obučavanje.

2.2.1 Minimizacija empirijskog rizika u slučaju regresije

Kao što je već rečeno, jednoj vrednosti atributa, ne mora odgovarati tačno jedna vrednost ciljne promenljive, već ima smisla govoriti o raspodeli vrednosti ciljne promenljive pri datim vrednostima atributa. Stoga se postavlja pitanje koju vrednost ciljne promenljive izabrati prilikom predviđanja. Jedan intuitivan odgovor je za date vrednosti atributa izabrati srednju vrednost ciljne promenljive od svih koje im odgovaraju. Sredina se često formalizuje pojmom očekivanja. Željena funkcija se onda može definisati kao

$$r(x) = \mathbb{E}[y|x] = \int y p(y|x) dy$$



Slika 2.2: Ilustracija regresione funkcije.

i naziva se *regresionom funkcijom*. Regresiona funkcija je ilustrovana na slici 2.2.

Definisanju regresione funkcije može se pristupiti i sa druge strane. Potrebno je da vrednosti koje daje model što bolje odgovaraju pravim vrednostima, odnosno da je razlika $y - f_w(x)$ što manja. U tom slučaju i kvadrat razlike je mali. Primetimo da se kvadrat razlike može uzeti za funkciju greške i da se rizik onda definiše kao

$$\mathbb{E}[(y - f_w(x))^2]$$

Važi

$$\min_w \mathbb{E}[(y - f_w(x))^2] =$$

$$\min_w \left\{ \int (y - f_w(x))^2 p(x, y) dx dy \right\} =$$

$$\min_w \left\{ \int (y - r(x) + r(x) - f_w(x))^2 p(x, y) dx dy \right\} =$$

$$\min_w \left\{ \int [(y - r(x))^2 + 2(y - r(x))(r(x) - f_w(x)) + (r(x) - f_w(x))^2] p(x, y) dx dy \right\} =$$

$$\min_w \left\{ \int (y - r(x))^2 p(x, y) dx dy + \int [2(y - r(x))(r(x) - f_w(x)) + (r(x) - f_w(x))^2] p(x, y) dx dy \right\} =$$

$$\min_w \left\{ \int [2(y - r(x))(r(x) - f_w(x)) + (r(x) - f_w(x))^2] p(x, y) dx dy \right\} =$$

$$\begin{aligned}
\min_w \left\{ 2 \int (y - r(x))(r(x) - f_w(x)) p(x, y) dx dy + \int (r(x) - f_w(x))^2 p(x, y) dx dy \right\} &= \\
\min_w \left\{ 2 \int (y - r(x))(r(x) - f_w(x)) p(y|x) p(x) dx dy + \int (r(x) - f_w(x))^2 p(x, y) dx dy \right\} &= \\
\min_w \left\{ 2 \int (r(x) - f_w(x)) \left(\int (y - r(x)) p(y|x) dy \right) p(x) dx + \int (r(x) - f_w(x))^2 p(x, y) dx dy \right\} &= \\
\min_w \left\{ 2 \int (r(x) - f_w(x)) \left(\int y p(y|x) dy - r(x) \int p(y|x) dy \right) p(x) dx + \int (r(x) - f_w(x))^2 p(x, y) dx dy \right\} &= \\
\min_w \left\{ 2 \int (r(x) - f_w(x))(r(x) - r(x)) p(x) dx + \int (r(x) - f_w(x))^2 p(x, y) dx dy \right\} &= \\
\min_w \left\{ \int (r(x) - f_w(x))^2 p(x, y) dx dy \right\} &= \\
\min_w \left\{ \int (r(x) - f_w(x))^2 p(y|x) p(x) dx dy \right\} &= \\
\min_w \left\{ \int (r(x) - f_w(x))^2 \left(\int p(y|x) dy \right) p(x) dx \right\} &= \\
\min_w \left\{ \int (r(x) - f_w(x))^2 p(x) dx \right\} &= \\
\min_w \mathbb{E}[(r(x) - f_w(x))^2]
\end{aligned}$$

Očigledno, ako regresiona funkcija pripada skupu modela, u njoj se postiže minimum datog rizika. Ukoliko ne pripada, kako poslednji integral predstavlja metriku, jasno je da je najbolja funkcija ona koja joj je u smislu te metrike najbliža.

Kako je rizik dat izrazom $\mathbb{E}[(y - f_w(x))^2]$, prirodna formulacija principa minimizacije empirijskog rizika za regresiju je

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

Ovaj pristup je sveprisutan u problemima regresije. Naglasimo da je funkcija greške data izrazom

$$L(u, v) = (u - v)^2$$

i da se često naziva *kvadratnom greškom* (eng. squared loss), a odgovarajuća srednja greška *srednjekvadratnom greškom*. Zamislite su i drugačije funkcije greške. Na primer $L(u, v) = |u - v|$. Tada optimalno rešenje problema minimizacije rizika nije više uslovno očekivanje $\mathbb{E}[y|x]$, već uslovna medijana $m[y|x]$. Pored toga, funkcija greške ne mora biti ni simetrična. Na primer, prilikom predviđanja cene akcija na berzi, važnije je predvideti da li će cena akcija porasti ili opasti, nego koliko. Stoga, ukoliko cena raste, greška naniže, koja može rezultovati negativnim predviđanjem (padom) je opasnija od greške naviše usled koje će zarada od kupovine takvih akcija biti manja nego što je očekivano, ali će kupac i dalje biti na dobitku.

2.2.2 Minimizacija empirijskog rizika u slučaju klasifikacije

U slučaju klasifikacije, formulacija principa empirijskog rizika je još jedostavnija. Model je utoliko bolji ukoliko pravi manje grešaka pri klasifikaciji. Neka je F tvrđenje. *Indikatorska funkcija* se definiše tako da važi

$$I(F) = \begin{cases} 1 & \text{ako } F \\ 0 & \text{ako } \neg F \end{cases}$$

Onda se princip minimizacije empirijskog rizika za klasifikaciju može definisati kao rešavanje problema

$$\min_w \frac{1}{N} \sum_{i=1}^N I(y \neq f_w(x))$$

Funkcija greške je

$$L(u, v) = I(u \neq v)$$

i naziva se prosto *greška klasifikacije*. Moguć je, a često i poželjan drugačiji izbor funkcije greške. Na primer, ne moraju sve greške biti jednakovražne. Neke klase se mogu smatrati srodnijim od drugih, pa je pogrešnu klasifikaciju između tih klasa lakše tolerisati nego pogrešnu klasifikaciju između klasa koje nisu srodne. Ovo je primer *klasifikacije osetljive na cenu greške* (eng. *cost sensitive classification*). Takođe, funkcija greške ne mora biti ni simetrična. Nekada je opasnije instancu jedne klase klasifikovati kao instancu druge nego obrnuto. Na primer, prilikom klasifikacije medicinskih snimaka, pogrešna detekcija kancerogenog oboljenja može biti potresna za pacijenta, ali će se daljim testovima ustanoviti da je dijagnoza bila pogrešna. S druge strane, ukoliko se ustanovi da pacijent nije bolstan u slučaju kada jeste, greška može biti fatalna. Stoga u ovom kontekstu i funkcija greške treba da pridruži različite vrednosti različitim vrstama grešaka.

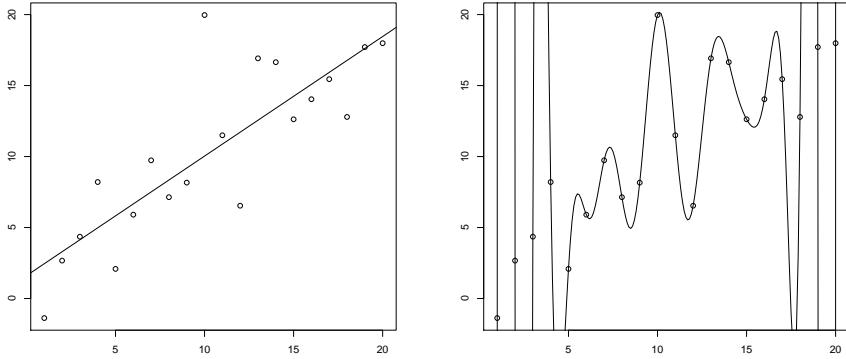
2.3 Preprilagođavanje

Minimizacija srednje greške izborom parametara modela predstavlja prilagođavanje modela podacima. Pojam prilagođavanja je jednostavno i vizuelno približiti. Neka je dat skup tačaka u dve dimenzije. Dimenzija x predstavlja vrednost atributa, a dimenzija y vrednost ciljne promenljive. Razmotrimo dve vrste modela. Prvi predstavlja skup svih modela oblika

$$f_w(x) = w_0 + w_1 x$$

a drugi, skup svih modela oblika

$$g_w(x) = \sum_{i=0}^n w_i x^i$$



Slika 2.3: Najbolja aproksimacija tačaka pravom i polinomom visokog stepena.

za proizvoljno $n \geq 0$. Prvi skup očito sadrži prave, a drugi polinome proizvoljnog stepena. Drugi skup je bogatiji u smislu da su sve prave istovremeno i polinomijalne krive. Otud se može očekivati da će se među funkcijama drugog skupa pre naći model sa manjom srednjom greškom bez obzira kakvi su podaci dati. I zaista, dok god za istu vrednost koordinate x u skupu tačaka ne postoje dve tačke sa različitim vrednostima koordinate y , postoji neki interpolacioni polinom koji prolazi kroz sve tačke skupa. Na slici 2.3 prikazani su prava i polinom sa najmanjim srednjekvadratnim greškama na datom skupu tačaka. Postavlja se pitanje koji je model bolji. U smislu srednjekvadratne greške, očito to je polinomijalni model. Ipak, dok se u slučaju linearne modela jasno uočava da, uprkos odstupanjima, prati opšti trend rasta koji je vidljiv u tačkama, u slučaju polinomijalnog modela, to se ne može jasno reći zbog drastičnih oscilacija, posebno u krajevima. Dok su za linearni model vrednosti predviđene u tačkama između datih bliske vrednostima u datim tačkama, polinomijalni model u takvim tačkama daje vrednosti koje su nekoliko redova veličine veće od okolnih. Otud bi svako ko bi u nekom relevantnom problemu za predviđanje koristio ovakav model bio na šteti.

Ovo zapažanje govori da minimizacija srednje greške nije nužno najbolji kriterijum, odnosno da postoje situacije u kojima funkcija koja minimizuje srednju grešku na dostupnim podacima pravi drastične greške na ostalim podacima iz iste raspodele, odnosno da vrlo slabo aproksimira funkciju koja minimizuje stvarni rizik, a koja bi davala relativno malu grešku i na podacima van datog uzorka. Drugim rečima model koji minimizuje srednju grešku na podacima ne mora nužno dobro generalizovati. Ovaj problem, da se minimizacijom srednje greške model prilagođava podacima u toj meri da izgubi moć generalizacije, naziva se *problemom preprilagođavanja modela podacima* ili samo *preprilagođavanjem*. Ovaj problem predstavlja jedan od teorijski i praktično

najznačajnijih problema mašinskog učenja. Iako bi početnik lako pomislio da je glavni izazov naći model koji se što bolje prilagođava podacima, to zapravo nije teško – poznat je veliki broj vrlo prilagodljivih modela. Suštinski problem je ne prilagoditi model podacima previše. Iz proučavanja ovog problema potekli su najvredniji teorijski uvidi mašinskog učenja, o čemu će biti reči kasnije.

Često se navodi da gubitak generalizacije nastaje iz preteranog prilagođavanja modela šumu. Iako to može biti deo problema, dato zapažanje ne opisuje kompletan problem. Naime, moguće je da u konačnom (nekad i malom) skupu za obučavanje svi podaci pripadaju nekom specifičnom delu prostora atributa. Preprilagođavanjem, model uči da su i neke nesuštinske specifičnosti datih podataka od presudnog značaja za odnos sa cilnjom promenljivom koji se u podacima vidi i kad ta svojstva nisu prisutna, greši.

Kao još jedna ilustracija problema preprilagođavanja, u cilju njegovog približavanja intuiciji, biće dat jedan pomalo veštački, ali ilustrativan primer. Pretpostavimo da profesor na ispitu daje zadatke iz obimne zbirke koja je dostupna studentima. Jedan pristup polaganju ispita, koji podržavamo, bi mogao biti da student razume suštinske principe materije, koji se često mogu izložiti u drastično manjem obimu od obima pomenute zbirke. Primenom tih principa student je u stanju da u velikom broju slučajeva reši zadatak. Možda ne u svakom slučaju, ali dovoljno da položi ispit. Drugi pristup polaganju ispita, koji ne podržavamo, bi mogao biti da student napamet nauči bas sve zadatke i njihova rešenja iz pomenute zbirke, uprkos njenom obimu. U tom slučaju student sa savršenim uspehom rešava sve zadatke na ispitu. Ipak, ukoliko bi pomenuti profesor u nezgodnom trenutku otišao na odmor i prepustio ispit drugom profesoru koji bi dao zadatke van pomenute zbirke, student koji bi učio po prvom principu bi kod drugog profesora prošao podjednako dobro, dok bi student koji je učio po drugom principu kod drugog zasigurno pao ispit. Sličnost sa primerom linearnih i polinomijalnih modela je upečatljiva. Linearni model sa manjim brojem parametara odgovara uočavanju zavisnosti koje se mogu predstaviti u obimu mnogo manjem od zbirke. Polinomijalni model koji se savršeno prilagođava podacima ima onoliko parametara koliko ima tačaka kojima se prilagođava, baš kao što je student u drugom slučaju morao imati onoliko memorije koliko u zbirci ima zadataka.

Deluje da je za uspešnu generalizaciju potrebno birati modele iz relativno siromašnog skupa, a koji relativno dobro odgovaraju podacima, pre nego modele iz vrlo bogatog skupa koji savršeno odgovaraju podacima. Ipak, vrlo je teško unapred odrediti koliko bogat skup treba da bude. Stoga se problemu pristupa drugačije – moguće je dozvoliti vrlo fleksibilnu reprezentaciju modela (koja odgovara bogatom skupu modela), ali kontrolisati fleksibilnost te reprezentacije u vreme obučavanja *regularizacijom*.

2.4 Regularizacija

Pristup učenju izborom modela iz siromašnog skupa funkcija pati od očiglednog nedostatka. Ukoliko se u datom skupu ne nalazi nijedan model sa relativno malom srednjom greškom, ne može se očekivati dobra generalizacija, jer model nije ništa naučio. S druge strane u bogatom skupu se može očekivati da se nalazi model koji dobro generalizuje, ali ga je teško naći. Jedna tehnika koja često omogućava izbor modela koji dobro generalizuje iz bogatog skupa funkcija, smanjujući fleksibilnost reprezentacije u vreme obučavanja, naziva se *regularizacijom*. Ona predstavlja modifikaciju minimizacionog problema, koja se sastoji u dodavanju takozvanog *regularizacionog izraza* $\Omega(w)$ koji otežava prilagođavanje modela podacima i tako predstavlja kontratež preprilagođavanju. Naravno, ako se prilagođavanje potpuno onemogući obučavanje je nemoguće. Stoga je potrebno kontrolisati dve pomene suprotstavljene težnje. Tome služi *regularizacioni parmetar* λ kojim se kontroliše koliko se težine pridaje minimizaciji srednje greške, a koliko regularizacionom izrazu. *Regularizovani problem* je sledeći

$$\min_w \frac{1}{N} \sum_{i=1}^N L(y_i, f_w(x_i)) + \lambda \Omega(w)$$

Za regularizacioni izraz se često uzima kvadrat ℓ_2 norme vektora koeficijenata, pa minimizacioni problem postaje sledeći

$$\min_w \frac{1}{N} \sum_{i=1}^N L(y_i, f_w(x_i)) + \lambda \|w\|_2^2$$

pri čemu se podrazumeva da se minimizuje ceo zbir, a ne da se regularizacioni izraz dodaje na minimalnu vrednost srednje greške.

Smisao regularizacije se može lakše razumeti posmatrajući ekstreme. Prvi, u slučaju kada važi $\lambda = 0$ znači da regularizacije nema i ukoliko je reprezentacija modela dovoljno fleksibilna, lako dolazi do preprilagođavanja. Drugi, u slučaju kada je vrednost parametra λ velika (neformalno, razmišljajmo kao da je beskonačna), minimizovanjem srednje greške se ništa značajno ne dobija. Jedino je važno minimizovati regularizacioni izraz, koji svoj minimum dostiže kada važi $w = 0$, što je jedan jedini model, od kojeg se ne može očekivati nikakva prilagodjivost. Stoga, prvi sabirak zahteva prilagođavanje modela podacima, dok drugi to otežava, a regularizacionim parametrom se vrši vaganje između tih suprotstavljenih tendencija.

Još jedana zanimljiva interpretacija³ je sledeća. Nauka uopšte, bavi se uspostavljanjem veza između različitih pojava. Pretpostavimo da ih možemo predstaviti numerički atributima i cilnjom promenljivom. Uobičajena pret-

³Na kojoj se zahvaljujemo kolegi Milošu Jovanoviću.

postavka u nauci, koja sledi iz principa Okamove oštice⁴ je da veza između posmatranih pojava nema. Ona se u statistici naziva nultom hipotezom. Onaj ko želi da pokaže da neka veza postoji mora prezentovati argumente u korist te hipoteze koja je alternativa nultoj. Regularizacioni izraz predstavlja pretpostavku da veze nema jer, zaista, ukoliko mu se u potpunosti udovolji, ishod minimizacije je model $w = 0$, a ukoliko su parametri modela 0, to znači da promene u vrednostima atributa nikako ne utiču na promenu u vrednosti ciljne promenljive. Srednja greška, koja uključuje informaciju o podacima, predstavlja težinu argumentacije da veze ima. Potrebna je jaka argumentacija, odnosno visoka vrednost srednje greške na opaženim podacima da bi nas uverila da je naša nulta hipoteza pogrešna i da treba prihvati činjenicu da neka veza postoji, odnosno da koficijenti modela ne treba da budu nula.

Još jedan, više tehnički uvid u smisao regularizacije se može dobiti razmatranjem slučaja linearog modela. U slučaju linearog modela važi $\nabla_x f_w(x) = w$. Odnosno, problem učenja se može zapisati na sledeći način:

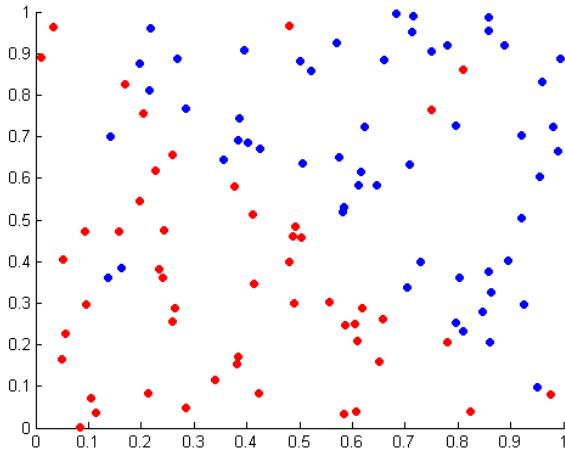
$$\min_w \frac{1}{N} \sum_{i=1}^N L(y_i, f_w(x_i)) + \lambda \|\nabla_x f_w(x)\|_2^2$$

U slučaju linearne funkcije, što je veći regularizacioni parametar, manje je izgledno da će rešenje imati visoku vrednost gradijenta, pošto se u minimizacionom problemu insistira na njegovoj maloj vrednosti. Funkcija koja ima relativno malu vrednost gradijenta ne može brzo rasti i opadati i stoga se ne može lako prilagoditi bilo kakvim podacima. Na primer, kada usled jakog šuma u podacima ciljna promenljiva u bliskim tačkama uzima čas visoke čas niske vrednosti, bilo bi potrebno da funkcija brzo raste ili opada kako bi savršeno odgovarala podacima. Zamislivo je ovakvu regularizaciju primeniti i na druge vrste modela, osim linearnih, ali se u praksi ovakva ipak regularizacija ne koristi.

Treba imati u vidu da regularizaciju nije nužno uvek od koristiti. Posebno je značajna u slučaju da je količina podataka za obučavanje mala u odnosu na prilagodljivost modela (što nije lako kvantitativno izraziti). Kada je količina podataka velika, regularizacija nije neophodna. Nekada se kaže da je povećanje količine podataka najbolja regularizacija, ali velika količina podataka nije uvek dostupna.

U opštijem smislu, regularizacijom se često naziva bilo kakva modifikacija problema koja ograničava prilagodljivost modela i čini ga manje podložnim preprilagođavanju. Čak i određene pretpostavke o reprezentaciji modela koje je ograničavaju se nekad u literaturi tako nazivaju. U najopštijem smislu, van konteksta mašinskog učenja, regularizacijom se naziva modifikacija bilo kog matematičkog problema koja ga čini bolje uslovjenim, odnosno čini da za

⁴Okamova oštica je jedan od osnovnih principa naučnog zaključivanja i kaže da entitete kojima se nešto objašnjava ne treba umnožavati preko potrebe ili, drugim rečima, najjednostavnije objašnjenje saglasno sa podacima je najbolje. Pitanje jednostavnosti objašnjenja je netrivijalno i potpada u domen filozofije nauke. O tome će biti reči kasnije.



Slika 2.4: Tačke u ravni koje pripadaju dvema klasama.

male promene ulaznih parametara problema i promene rešenja problema budu male.

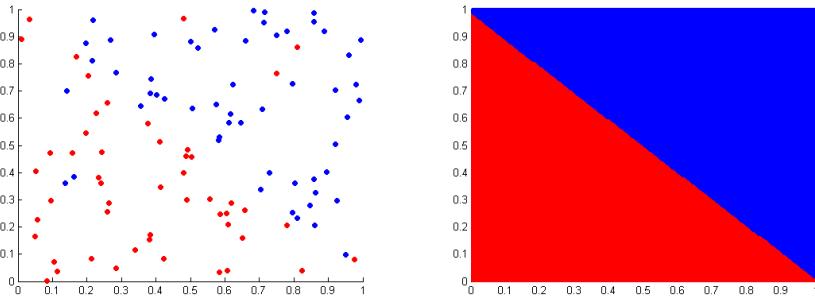
Za kraj, razmotrimo primer dejstva regularizacije na klasifikacione modele. Neka su u ravni date tačke koje pripadaju dvema klasama, kao na slici 2.4. Kao jedan prirodan izbor modela klasifikacije, nameće se prava, odnosno linearni model oblika:

$$f_w(x) = w_0 + w_1x_1 + w_2x_2$$

Alternativno, moguće je koristiti i linearni model koji predstavlja polinom dve promenljive (linearnost je po parametrima, ne po promenljivim!):

$$g_w(x) = \sum_{i=0}^n \sum_{j=0}^i w_{ij} x_1^j x_2^{i-j}$$

U oba slučaja indikatorom klase se smatra znak funkcije. Na slici 2.5 prikazano je koje klase svim tačkama u datom segmentu ravni pridružuje najbolja prava. Globalna zakonitost deluje ispravno ustanovljena, iako sa obe strane ima pogrešno klasifikovanih tačaka. Na slici 2.6 prikazana je klasifikacija polinomijalnim modelom za različite, rastuće, vrednosti regularizacionog parametra. Može se uočiti da granica između dve klase postaje sve jednostavnija sa povećanjem regularizacionog parametra, dok u poslednjem slučaju ekstremno visoke vrednosti regularizacionog parametra, model ne postane konstantan. Jedno pitanje koje vredi razmotriti je zašto model konstantno predviđa baš plavu klasu, a ne crvenu. Razlog je taj što na slici postoji jedna plava tačka više. Ipak, ako je vrednost regularizacionog parametra velika (a jeste) i ako u tom slučaju model ne bi trebalo da ima moć prilagođavanja, kako se model prilagodio odnosu



Slika 2.5: Klasifikacija tačaka linearnim modelom.

broja tačaka? Odgovor se krije u čestoj praksi, koja je primenjena i ovde, da se slobodni član modela, koji ne množi nijedan atribut izuzeće iz regularizacije.

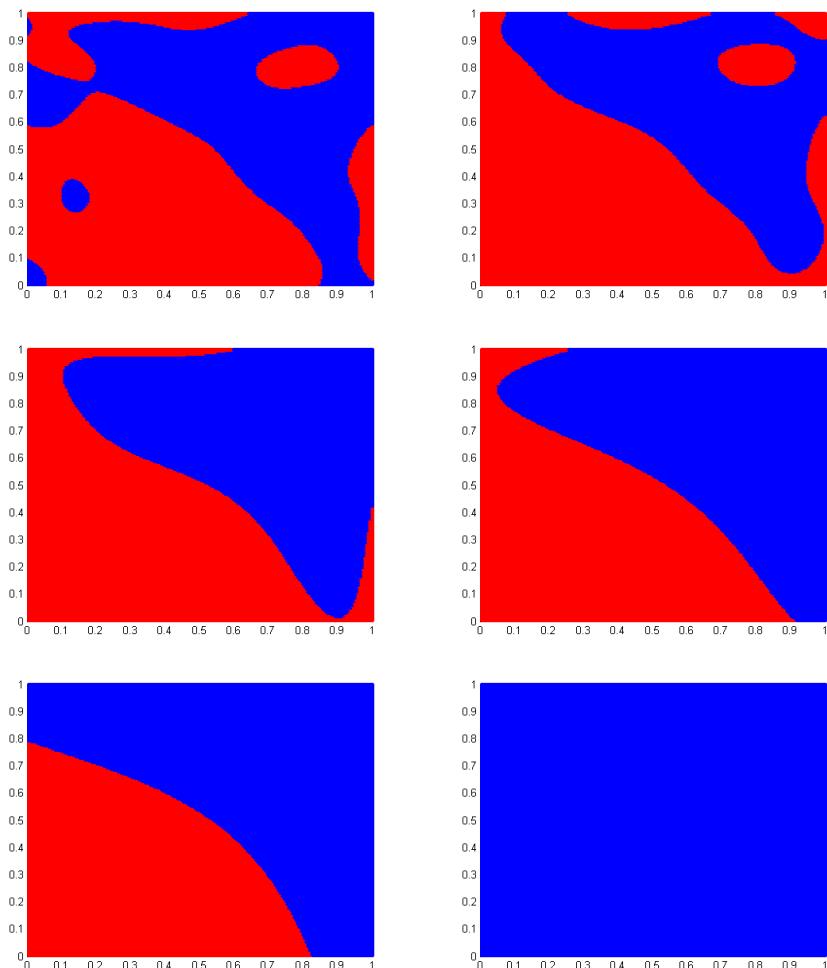
2.5 Nagodbu između sistematskog odstupanja i varijanse

Kako se prilikom regularizacije menja optimizacioni problem, menja se i njegovo rešenje, to jest model koji se dobija optimizacijom. Na osnovu prethodne diskusije, ovo je u slučaju nedovoljne količine podataka upravo željeni efekat. U slučaju velike količine podataka, regularizacija bi tipično onemogućila model da im se se adekvatno prilagodi. Ovo ponašanje se da osvetliti sa još jedne strane nagodbom između sistematskog odstupanja i varijanse. Razmotrimo slučaj regresije. Kada je analiziran princip minimizacije rizika za regresiju, pokazano je da važi

$$\min_w \mathbb{E}[(y - f_w(x))^2] = \min_w \mathbb{E}[(r(x) - f_w(x))^2]$$

Odnosno, optimalna funkcija koju je potrebno naučiti je regresiona funkcija $r(x)$. Razmotrimo sada sledeći misaoni eksperiment. Neka se iz raspodele podataka puno puta generiše skup nezavisnih opažanja \mathcal{D} i neka se na njemu obučava model $f_{\mathcal{D}}(x)$. U ovom konkretnom slučaju umesto parametara nagašavamo skup instanci na osnovu kojih je model dobijen. U svakoj tački x , različiti modeli, dobijeni na osnovu različitih skupova podataka, imaće različita odstupanja od idealne regresione funkcije. Performanse konkretnog algoritma učenja se u svakoj tački mogu proceniti kao prosek velikog broja takvih odstupanja. Ova procedura zapravo u svakoj tački ocenjuje očekivanje funkcije greške po svim skupovima podataka $\mathbb{E}_{\mathcal{D}}[(r(x) - f_{\mathcal{D}}(x))^2]$. Razmotrimo prvo vizuelno šta se dešava za različite skupove podataka na sledećem jednostavnom primeru.

Neka se više puta generiše skup podataka tako što se nasumično biraju tačke x u intervalu $[0, 2]$, na svaku tačku se primeni eksponencijalna funkcija i doda mali šum, čime se dobija vrednost y . Svi generisani skupovi predstavljaju isti



Slika 2.6: Klasifikacija tačaka polinomijalnim modelom za rastuće vrednosti regularizacionog parametra.

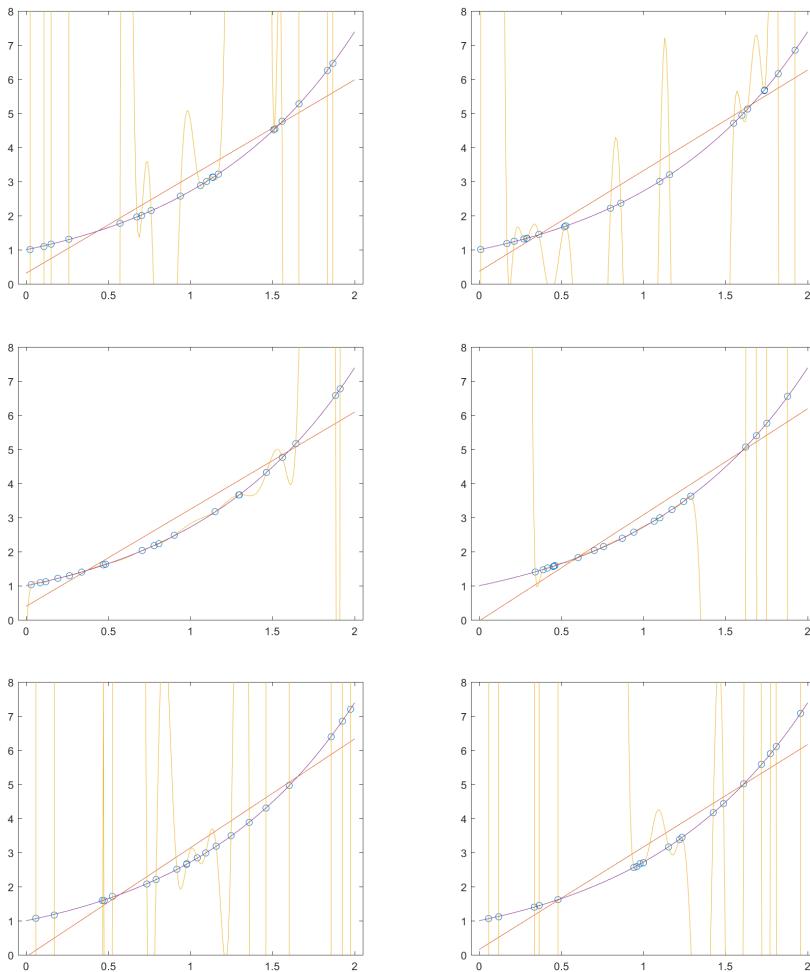
eksponencijalni zakon koji povezuje promenljive x i y , odnosno, može se reći da su promene u podacima male. Na slici 2.7 je prikazano pet takvih skupova podataka i za svaki od njih prava i polinom sa najmanjom srednjekvadratnom greškom. Oba modela bi trebalo da ocenjuju eksponencijalnu funkciju. Ukoliko čitalac obrati pažnju na bilo koju vrednost na x osi i pogleda kako se od slike do slike menjaju vrednosti oba modela u toj tački, primetiće da vrednosti koje predviđa prava malo variraju. Obično su sve manje ili sve veće od vrednosti eksponencijalne funkcije i zbog toga se pri uprosećavanju ne poništavaju. Zaključujemo da u različitim tačkama, prava tipično sistematski odstupa od regresione funkcije, ali malo varira. S druge strane, vrednosti polinoma u istoj tački izuzetno variraju. Nekada su veće od vrednosti eksponencijalne funkcije, a nekad manje i stoga se pri uprosećavanju poništavaju.

Prethodna zapažanja se mogu formalizovati kroz sledeću analizu očekivanja $\mathbb{E}_{\mathcal{D}}[(r(x) - f_{\mathcal{D}}(x))^2]$. Oznaka skupa podataka će biti izostavljena zbog čitljivosti, ali se podrazumeva. Važi:

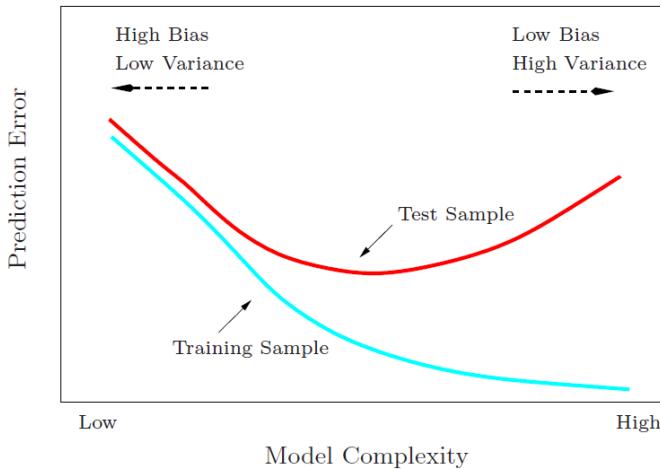
$$\begin{aligned}\mathbb{E}[(r(x) - f(x))^2] &= \mathbb{E}[(r(x) - \mathbb{E}[f(x)] + \mathbb{E}[f(x)] - f(x))^2] = \\ \mathbb{E}[(r(x) - \mathbb{E}[f(x)])^2] &+ 2\mathbb{E}[(r(x) - \mathbb{E}[f(x)])(\mathbb{E}[f(x)] - f(x))] + \mathbb{E}[(\mathbb{E}[f(x)] - f(x))^2] \\ \text{Kako regresiona funkcija ne zavisi od skupa podataka, ovaj izraz je jednak izrazu} \\ (r(x) - \mathbb{E}[f(x)])^2 &+ 2(r(x) - \mathbb{E}[f(x)])\mathbb{E}[\mathbb{E}[f(x)] - f(x)] + \mathbb{E}[(\mathbb{E}[f(x)] - f(x))^2] = \\ (r(x) - \mathbb{E}[f(x)])^2 &+ 2(r(x) - \mathbb{E}[f(x)])(\mathbb{E}[f(x)] - \mathbb{E}[f(x)]) + \mathbb{E}[(\mathbb{E}[f(x)] - f(x))^2] = \\ (r(x) - \mathbb{E}[f(x)])^2 &+ \mathbb{E}[(\mathbb{E}[f(x)] - f(x))^2]\end{aligned}$$

Prvi izraz je kvadrat sistematskog odstupanja modela od regresione funkcije u tački x – vrednost regresione funkcije u tački x je ono što model ocenjuje, ali kako za različite podatke dobijamo različite modele, možemo razmatrati očekivanje vrednosti koje ti modeli daju u tački x . A razlika između onoga što treba oceniti i očekivanja onoga čime se ocena vrši je po definiciji sistematsko odstupanje. Drugi izraz je očigledno varijansa modela. Pojasnimo o kakvoj varijansi modela je reč. Ponovo, kako se za različite podatke dobijaju različiti modeli koji u nekoj tački nude različita predviđanja, u svakoj tački postoji određena varijansa tih predviđanja. Ta varijansa se naziva varijansom modela u toj tački.

Izvedena jednakost se naziva *dekompozicijom greške na sistematsko odstupanje i varijansu*. Vrlo jednostavnii modeli poput pravih, zbog svoje rigidnosti obično ne mogu dobro da se prilagode podacima, zbog čega njihova predviđanja sistematski dosta odstupaju od regresione funkcije bez obzira na konkretni izbor skupa za obučavanje. S druge strane, vrlo fleksibilni modeli se tipično preprilagođavaju specifičnim podacima i već za male promene u podacima daju značajno različita predviđanja. Time prave veliku grešku koja je odraz njihove visoke varijanse. Ovo je shematski prikazano na slici 2.8. Regularizacija



Slika 2.7: Prava i polinom najmanje srednjekvadratne greške za različite podatke iz istog eksponencijalnog zakona.



Slika 2.8: Ponašanje greške na podacima za obučavanje i na odvojenim podacima za evaluaciju iz iste raspodele u slučaju модела разлиčite fleksibilnosti.

омогућава контролу fleksibilnosti модела. Избором прве вредности регуларizacionог метапараметра могуће је значајно смањити варијансу, а по цену умереног раста систематског одступања. Ово се назива *nagodbom između sistematskog odstupanja i varijanse*. Тиме се објашњава успењност регуларизације у обуčавању модела машињског учења. Када се бира прва вредност регуларizacionог метапараметра биће објашњено касније.

2.6 Teorijske гаранције квалитета генерализације

Као што је већ више пута наглашено, моћ генерализације неког алгоритма машињског учења који минимизује empiriјски ризик (у овом делу користимо овај израз уместо израза средња грешка), зависи од богатства скупа из којег се модел бира, односно од fleksibilnosti reprezentације модела. Такозвана *статистичка теорија учења*, бави се прoučавањем моћи генерализације на основу различитих скупова модела. Нjeni резултати се обично приказују у виду граница за вредност ризика. Основна пitanja која постављају су:

- Уколико је познат empiriјски ризик модела f , колики може бити njegov stvarni ризик? Одговор се своди на проналачење горње границе стварног ризика која је израžена у terminima empiriјског ризика $E(f)$, величине скупа података N и скупа свих могуćih модела \mathcal{F} из којих се бира најбољи. Резултат треба да буде неједнакост облика $R(f) \leq E(f) + c(N, \mathcal{F})$.
- Колико је стварни ризик изабраног модела f виши од стварног ризика најбољег модела $f^* \in \mathcal{F}$? Резултат треба да буде облика $R(f) \leq R(f^*) + c(N, \mathcal{F})$

- Koliko je stvarni rizik izabranog modela f viši od stvarnog rizika najboljeg mogućeg modela (koji ne mora stvarno biti u \mathcal{F})? Taj rizik označavamo sa R^* i ne bavimo se time kom modelu odgovara. Rezultat treba da bude oblika $R(f) \leq R^* + c(N, \mathcal{F})$.

U nastavku se bavimo prvim pitanjem kao verovatno najvažnijim. Njegov značaj je u tome što daje odgovor na to koliko prosečna greška na svim zamslivim podacima (stvarni rizik) može odstupiti od greške dobijene na skupu za obučavanje (empirijski rizik). Ukoliko je greška na skupu za obučavanje mala, zanima nas možemo li i pod kojim uslovima očekivati da je i greška na ostalim podacima mala. Kako bismo to znali, potrebno je da izvedemo veličinu c koja se pominje u prvom pitanju, a koju možemo smatrati širinom intervala poverenja za stvarni rizik, kada nam je dat empirijski rizik. Fundamentalni rezultat je da se pod određenim uslovima sa povećanjem veličine uzorka, uniformno za sve funkcije, širina intervala poverenja smanjuje, odnosno da se empirijskoj oceni može verovati.

U nastavku će biti razmatran samo slučaj binarne klasifikacije sa funkcijom greške $L(u, v) = I(u \neq v)$, a rezultati se mogu uopštiti i na višeklasnu klasifikaciju, na druge funkcije greške i na regresiju.

Za fiksiranu funkciju f , potrebno je okarakterisati odstupanje veličine $R(f)$ od $E(f)$, odnosno razliku:

$$R(f) - E(f) = \mathbb{E}[L(y, f(x))] - \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i))$$

Prema zakonu velikih brojeva važi

$$P\left(\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) - \mathbb{E}[y, f(x)] = 0\right) = 1$$

Brzina ove konvergencije se čak može preciznije kvantifikovati pomoću Hefdingove (eng. Hoeffding) nejednakosti:

Teorema 1 (Hefdingova nejednakost) *Neka su X_1, \dots, X_N nezavisne i jednakos raspodeljene slučajne promenljive takve da važi $X_i \in [a, b]$ za $1 \leq i \leq N$. Tada za svako $\varepsilon > 0$ važi*

$$P\left(\frac{1}{N} \sum_{i=1}^N (\mathbb{E}[X_i] - X_i) > \varepsilon\right) \leq \exp\left(-\frac{2N\varepsilon^2}{(b-a)^2}\right)$$

Kako je funkcija greške $L(u, v) = I(u \neq v)$ ograničena i uzima vrednosti u intervalu $[0, 1]$ iz prethodne teoreme sledi

$$P\left(\frac{1}{N} \sum_{i=1}^N (\mathbb{E}[L(y, f(x))] - L(y_i, f(x_i))) > \varepsilon\right) \leq \exp(-2N\varepsilon^2)$$

$$P \left(\mathbb{E}[L(y, f(x))] - \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) > \varepsilon \right) \leq \exp(-2N\varepsilon^2)$$

$$P(R(f) - E(f) > \varepsilon) \leq \exp(-2N\varepsilon^2)$$

Poželjno je što desna strana opada eksponencijalno sa povećavanjem broja instanci, ali prisustvo ε^2 usporava opadanje ove granice. Označimo desnu stranu sa δ i izrazimo ε kako bismo dobili željenu širinu c intervala poverenja. Iz $\delta = \exp(-2N\varepsilon^2)$ sledi

$$\varepsilon = \sqrt{\frac{\log \delta}{2N}}$$

Zamenjujući u prethodnu nejednakost dobijamo

$$P \left(R(f) - E(f) > \sqrt{\frac{\log \frac{1}{\delta}}{2N}} \right) \leq \delta$$

odnosno

$$P \left(R(f) - E(f) \leq \sqrt{\frac{\log \frac{1}{\delta}}{2N}} \right) \geq 1 - \delta$$

Drugačije, za svako $\delta > 0$ sa verovatnoćom od bar $1 - \delta$ važi

$$R(f) \leq E(f) + \sqrt{\frac{\log \frac{1}{\delta}}{2N}}$$

Drugim rečima za svaku funkciju f postoji skup C_f mere δ uzoraka koji krše gornju nejednakost, odnosno kod kojih je stvarni rizik značajno veći od empirijskog. Algoritam učenja uvek dobija neki uzorak \mathcal{D} na kojem uči i obično minimizuje regularizovani empirijski rizik birajući neku od funkcija iz skupa \mathcal{F} . Za izabranu funkciju f može se desiti da je bas \mathcal{D} nepovoljan uzorak za koji je gornja nejednakost prekršena, odnosno $\mathcal{D} \in C_f$. Razlog za ovo je što je pri izvođenju gornje nejednakosti analizirana proizvoljna funkcija bez osvrta na to da algoritam učenja može da bira bilo koju iz šireg skupa funkcija \mathcal{F} . Kako prethodna granica važi za fiksiranu funkciju f , nazivamo je *pojedinačnom granicom*. Kako za svaku funkciju f može postojati različit skup C_f , skup svih uzoraka u kojima granica ne važi bar za neku funkciju

$$C = \bigcup_{f \in \mathcal{F}} C_f$$

može biti mere veće od δ , pa se postavlja pitanje kolika treba da bude granica da bi se moglo garantovati da je skup C koji sadrži sve uzorce na kojima bilo koja funkcija pravi grešku veću od ε mere ne veće od δ . Ovakva granica se naziva *uniformnom granicom* jer važi za sve funkcije odjednom i, naravno, šira je od pojedinačne granice.

Kako bismo izveli uniformnu granicu, pretpostavimo prvo da je skup \mathcal{F} konačan. Potom, umesto ograničavanja razlike $R(f) - E(f)$, potrebno je ograničiti

$$\sup_{f \in \mathcal{F}} (R(f) - E(f))$$

Takva granica mora važiti za sve funkcije iz \mathcal{F} . Važi

$$\begin{aligned} P\left(\sup_{f \in \mathcal{F}} (R(f) - E(f)) > \varepsilon\right) &= \\ P\left(\{\mathcal{D} \mid \sup_{f \in \mathcal{F}} (R(f) - E(f, \mathcal{D})) > \varepsilon\}\right) &= \\ P\left(\bigcup_{f \in \mathcal{F}} \{\mathcal{D} \mid R(f) - E(f, \mathcal{D}) > \varepsilon\}\right) &\leq \\ \sum_{f \in |\mathcal{F}|} P(\{\mathcal{D} \mid R(f) - E(f, \mathcal{D}) > \varepsilon\}) &= \\ \sum_{f \in |\mathcal{F}|} P(R(f) - E(f, \mathcal{D}) > \varepsilon) &\leq \\ \sum_{f \in |\mathcal{F}|} \exp(-2N\varepsilon^2) &= |\mathcal{F}| \exp(-2N\varepsilon^2) \end{aligned}$$

Uvođenjem oznake δ za desnu stranu, dobija se

$$\varepsilon = \sqrt{\frac{\log |\mathcal{F}| + \log \frac{1}{\delta}}{2N}}$$

odnosno, za svako $\delta > 0$ sa verovatnoćom od bar $1 - \delta$, za svaku funkciju $f \in \mathcal{F}$ važi

$$R(f) \leq E(f) + \sqrt{\frac{\log |\mathcal{F}| + \log \frac{1}{\delta}}{2N}}$$

Neformalno, za većinu skupova podataka, za sve funkcije $f \in \mathcal{F}$ važi

$$R(f) - E(f) \in O\left(\sqrt{\frac{\log |\mathcal{F}|}{N}}\right)$$

Osnovni problem sa poslednjim rezultatom je što se oslanja na pretpostavku konačnosti skupa modela. Uopštenje ovog rezultata na prebrojivo beskonačan skup modela nije teško i preporučuje se čitaocu za vežbu. Pravi izazov dolazi sa neprebrojivošću skupa modela. Osnovna ideja je da u mašinskom učenju uvek radimo sa konačnim skupovima podataka i da stoga nije zaista bitno koliko ima različitih funkcija u skupu \mathcal{F} , već koliko ima funkcija koje se ponašaju različito na datom skupu podataka. Uvedimo oznake $z_1 = (x_1, y_1), \dots, z_N = (x_N, y_N)$ i

$$\mathcal{L}_{z_1, \dots, z_N} = \{(L(y_1, f(x_1)), \dots, L(y_N, f(x_N))) \mid f \in \mathcal{F}\}$$

Veličina ovog skupa je broj načina na koje podaci mogu biti klasifikovani (tačno ili netačno) i očito je konačna čak i kad skup \mathcal{F} to nije.

Definicija 1 (Funkcija rasta) *Funkcija rasta je maksimalan broj načina na koji N tačaka mogu biti klasifikovane pomoću funkcija iz skupa \mathcal{F} :*

$$S_{\mathcal{F}}(N) = \sup_{z_1, \dots, z_N} |\mathcal{L}_{z_1, \dots, z_N}|$$

Ispostavlja se da se funkcija rasta može iskoristiti kao mera „veličine“ skupa \mathcal{F} .

Teorema 2 (Vapnik-Červonenkis) *Za svako $\delta > 0$, sa verovatnoćom bar $1 - \delta$ za svako $f \in \mathcal{F}$,*

$$R(f) \leq E(f) + 2\sqrt{2\frac{\log S_{\mathcal{F}}(2N) + \log \frac{2}{\delta}}{N}}$$

U konačnom slučaju uvek važi $S_{\mathcal{F}}(N) \leq 2^N$, pa je ova teorema korisna i tada. Ipak, bitno je razumeti zašto važi u opštem slučaju. Glavni oslonac u dokazu ove teoreme daje takozvana lema o simetrizaciji. Njena osnovna ideja je da se izražavanje u terminima stvarnog rizika zameni izražavanjem u terminima empirijskog rizika na nekom odvojenom skupu za evaluaciju. Ovo zvuči razumno, tim pre što se u praksi tako i procenjuju performanse modela na podacima koji nisu viđeni u vreme obučavanja.

Teorema 3 (Lema o simetrizaciji) *Neka su data dva uzorka z_1, \dots, z_N i z'_1, \dots, z'_N i neka je E' empirijski rizik na drugom uzorku. Za svako $\varepsilon > 0$, tako da važi $N\varepsilon^2 \geq 2$, važi*

$$P(\sup_{f \in \mathcal{F}} (R(f) - E(f)) > \varepsilon) \leq 2P(\sup_{f \in \mathcal{F}} (E'_n(f) - E_n(f)) \geq \varepsilon/2)$$

Dokaz. Neka je funkcija f^* funkcija za koju se dostiže supremum na levoj strani nejednakosti. Tada važi:

$$\begin{aligned} I(R(f^*) - E(f^*) > \varepsilon)I(R(f^*) - E'(f^*) < \varepsilon/2) &= \\ I(R(f^*) - E(f^*) > \varepsilon \wedge E'(f^*) - R(f^*) \geq -\varepsilon/2) &\leq \\ I(E'(f^*) - E(f^*) > \varepsilon/2) \end{aligned}$$

Uzimajući očekivanje po drugom uzorku dobija se:

$$I(R(f^*) - E(f^*) > \varepsilon)P(R(f^*) - E'(f^*) < \varepsilon/2) \leq P(E'(f^*) - E(f^*) > \varepsilon/2)$$

Prema Čebišovljevoj nejednakosti, važi

$$P(R(f^*) - E'(f^*) > \varepsilon/2) \leq \frac{4 \text{var}[f^*]}{N\varepsilon^2} \leq \frac{1}{N\varepsilon^2}$$

Poslednja nejednakost važi jer nijedna promenljiva sa vrednostima na intervalu ne može imati veću varijansu od $1/4$. Odavde sledi

$$I(R(f^*) - E(f^*) > \varepsilon) \left(1 - \frac{1}{N\varepsilon^2}\right) \leq P(E'(f^*) - E(f^*) > \varepsilon/2)$$

Prelaskom na očekivanje po prvom uzorku, dobija se:

$$P(R(f^*) - E(f^*) > \varepsilon) \left(1 - \frac{1}{N\varepsilon^2}\right) \leq P(E'(f^*) - E(f^*) > \varepsilon/2)$$

Na osnovu uslova da važi $N\varepsilon^2 \geq 2$ i polaznog izbora funkcije f^* kao funkcije za koju se postiže supremum, dobija se traženi rezultat. \square

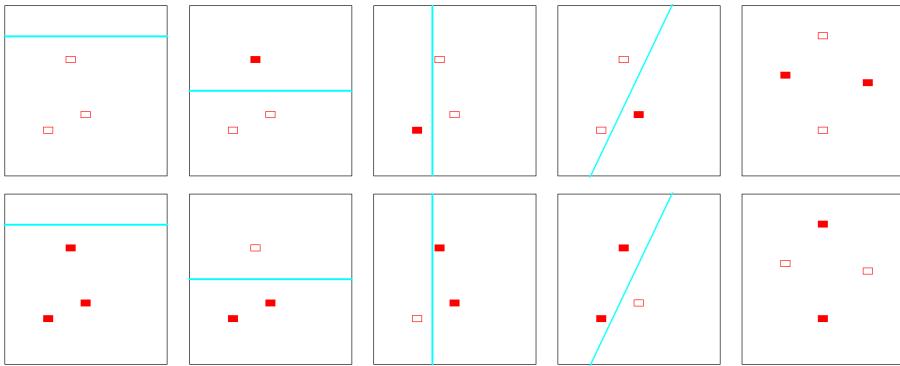
Sada je lako dokazati prethodnu teoremu:

$$\begin{aligned} P(\sup_{f \in \mathcal{F}} (R(f) - E(f)) > \varepsilon) &\leq \\ 2P(\sup_{f \in \mathcal{F}} (E'(f) - E(f)) > \varepsilon/2) &= \\ 2P\left(\sup_{l \in \mathcal{L}_{z_1, \dots, z_N, z'_1, \dots, z'_N}} \left(\frac{1}{N} \sum_{i=N+1}^{2N} l_i - \frac{1}{N} \sum_{i=1}^N l_i\right) > \varepsilon/2\right) &\leq \\ 2 \sum_{l \in \mathcal{L}_{z_1, \dots, z_N, z'_1, \dots, z'_N}} P\left(\frac{1}{N} \sum_{i=N+1}^{2N} l_i - \frac{1}{N} \sum_{i=1}^N l_i > \varepsilon/2\right) &= \\ 2 \sum_{l \in \mathcal{L}_{z_1, \dots, z_N, z'_1, \dots, z'_N}} P\left(\frac{1}{N} \sum_{i=N+1}^{2N} l_i - R(f) + R(f) - \frac{1}{N} \sum_{i=1}^N l_i > \varepsilon/2\right) &\leq \\ 2 \sum_{l \in \mathcal{L}_{z_1, \dots, z_N, z'_1, \dots, z'_N}} \left(P\left(\frac{1}{N} \sum_{i=N+1}^{2N} l_i - R(f) > \varepsilon/2\right) + P\left(R(f) - \frac{1}{N} \sum_{i=1}^N l_i > \varepsilon/2\right)\right) &\leq \\ 2 \sum_{l \in \mathcal{L}_{z_1, \dots, z_N, z'_1, \dots, z'_N}} (\exp(-N\varepsilon^2/2) + \exp(-N\varepsilon^2/2)) &= \\ 4 \sum_{l \in \mathcal{L}_{z_1, \dots, z_N, z'_1, \dots, z'_N}} \exp(-N\varepsilon^2/2) &= \\ 4S_{\mathcal{F}}(2N) \exp(-N\varepsilon^2/2) & \end{aligned}$$

Odnosno

$$P(\sup_{f \in \mathcal{F}} (R(f) - E(f)) > \varepsilon) \leq 4S_{\mathcal{F}}(2N) \exp(-N\varepsilon^2/2)$$

Zamenjujući desnu stranu sa δ , teorema je dokazana. Ipak, postavlja se pitanje kako se funkcija rasta izračunava.



Slika 2.9: Sve klasifikacije skupa od tri tačke pomoću pravih u ravni i problematičan primer četiri tačke u opštem rasporedu.

Definicija 2 (VC dimenzija) Vapnik-Červonenkisova (VC) dimenzija skupa funkcija \mathcal{F} je najveći broj N , takav da važi

$$S_{\mathcal{F}}(N) = 2^N$$

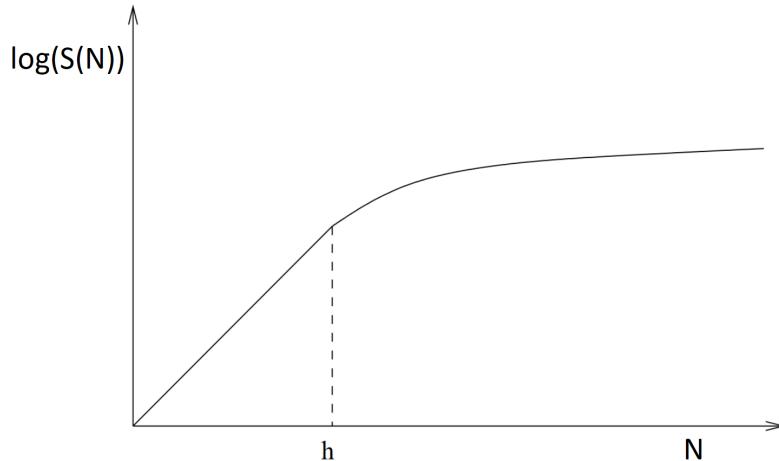
Ukoliko takav broj ne postoji, VC dimenzija je beskonačna.

Drugim rečima, to je najveći broj N za koji postoji skup tačaka koji se funkcijama iz skupa \mathcal{F} mogu klasifikovati na sve moguće načine. Pritom, dovoljno je da postoji jedan takav skup. Nije potrebno da se svaki skup tačaka može klasifikovati na sve moguće načine. Primer za slučaj dve dimenzije dat je na slici 2.9. Neka skup pravih predstavlja skup modela. Svakoj pravoj odgovaraju dva modela, koji suprotno označavaju poluravni koje prava indukuje. U prostoru \mathbb{R}^2 postoje tri nekolinearne tačke koje se svim pravima mogu razdvojiti na sve moguće načine, ali ne postoje 4 takve tačke. Otud je skup svih pravih u ravni VC dimenzije 3. Analogno, skup svih hiperravnih u prostoru \mathbb{R}^N je VC dimenzije $N + 1$. To je upravo broj parametara koji ima hiperravan u ovom prostoru. Ispostavlja se da broj parametara i VC dimenzija, iako se poklapaju za linearne modele, ne moraju uvek biti isti. Štaviše, ne moraju biti ni u kakvoj relaciji. Na primer, skup funkcija sa jednim parametrom

$$\{\operatorname{sgn}(\sin(wx)) \mid w \in \mathbb{R}\}$$

ima beskonačnu VC dimenziju.

Prema definiciji VC dimenzije, ako je h VC dimenzija skupa \mathcal{F} , za $N \leq h$ važi $S_{\mathcal{F}}(N) = 2^N$, za $N > h$ važi $S_{\mathcal{F}}(N) < 2^N$. Zapravo, ispostavlja se i jače – do vrednosti h , funkcija rasta je eksponencijalna, a nakon nje polinomijalna. Slika 2.10 ilustruje ovo svojstvo.



Slika 2.10: Logaritam funkcije rasta.

Lema 1 (Sauerova lema) *Neka je \mathcal{F} skup funkcija konačne VC dimenzije h . Tada za sve $N \in \mathbb{N}$ važi*

$$S_{\mathcal{F}}(h) \leq \sum_{i=0}^h \binom{N}{i}$$

a za $N \geq h$

$$S_{\mathcal{F}}(h) \leq \left(\frac{en}{h}\right)^h$$

Ovu lemu ne dokazujemo. Iz Vapnik-Červonenkisove teoreme i Sauerove leme direktno se dobija da za skup funkcija \mathcal{F} VC dimenzije h , za svako $\delta > 0$ sa verovatnoćom bar $1 - \delta$ za svaku funkciju $f \in \mathcal{F}$ važi

$$R(f) \leq E(f) + 2\sqrt{2 \frac{h \log \frac{2eN}{h} + \log \frac{2}{\delta}}{N}}$$

ili ugrubo

$$R(f) - E(f) \in O\left(\sqrt{\frac{h \log N}{N}}\right)$$

Imajući sve dosadašnje rezultate u vidu, zaključuje se da u slučaju skupa modela konačne VC dimenzije, dovoljno velik broj podataka garantuje povereće u empirijsku ocenu stvarnog rizika.

2.7 Veza statističke teorije učenja sa filozofijom nauke

Već je pomenuto jedno od osnovnih načela naučnog zaključivanja – Okamotova oštrica i rečeno je da kaže da entitetu kojima se nešto objašnjava ne treba

umnožavati preko potrebe, odnosno da je najjednostavnije objašnjenje najbolje. Postavlja se pitanje šta znači da je objašnjenje najjednostavnije. Da li to znači da se može izraziti pomoću najmanje reči? Ili da se uklapa u ustaljena viđenja? Ili nešto drugo? Prethodno pomenute interpretacije nekome mogu biti jednostavne, ali je pitanje da li su relevantne, a svakako su subjektivne i kulturno zavisne. Jedan način razumevanja složenosti objašnjenja je vezan za to koliko toga objašnjenje može da objasni. Stoga je najjednostavnije objašnjenje ono koje najmanje toga može da objasni, a najbolje objašnjenje neke pojave u smislu Okamove oštice je ono koje objašnjava tu pojavu i što manje toga pored nje.

Primetimo da modele mašinskog učenja ima smisla posmatrati kao objašnjenja veza između promenljivih koje podaci predstavljaju, a koje predstavljaju neke fenomene iz stvarnog sveta. Atributi i ciljna promenljiva variraju – menjaju vrednosti. Model objašnjava promenu ciljne promenljive tako što je povezuje sa promenom vrednosti atributa. Slično rade i fizički zakoni. Na primer zakon $F = ma$ objašnjava promenu u veličini F promenom u veličinama m i a . U terminima statističke teorije učenja Okamova oštice ima spremnu interpretaciju, pošto je pojam jednostavnosti već formalizovan VC dimenzijom. Skup modela je utoliko jednostavniji što mu je VC dimenzija niža. Otud, najbolji model je model koji odgovara podacima, a koji je izabran iz skupa niske VC dimenzije.

Treba primetiti da statistička teorija učenja formalizuje pojam indukcije – govori pod kojim uslovima je verovatno da će greška induktivnog zaključivanja (predviđanje modela) biti mala.

Pored Okamove oštice, dubok epistemološki uvid predstavlja Popperova filozofija nauke, koja se pre svega fokusira na razlučivanje između empirijske (naučne) teorije i metafizičke teorije. Prema Popisu, teorija je empirijska ukoliko je *poreciva* (eng. *falsifiable*). Pod porecivošću se podrazumeva zamislivost eksperimenta koji bi tu teoriju oborio. Na prvi pogled, mogućnost obaranja teorije je njena slabost, a nemogućnost obaranja zvuči preferirano. Međutim to je pogrešno. Moguće je konstruisati najrazličitije neporecive teorije bez ikakvog uporišta u empirijskim opažanjima. Čuveni primer Raselovog čajnika postulira da postoji čajnik koji orbitira oko Sunca negde između orbita Marsa i Jupitera, ali je napravljen od takvog materijala i takve je veličine da se nikako ne može detektovati. Očigledno, nemogućnost detektovanja čajnika čini da uprkos mnoštvu eksperimenata kojima pokušavamo da ustanovimo njegovo postojanje, ne možemo da razlikujemo njegovo posedovanje pomenutih svojstava od njegovog nepostojanja. Nešto bliže terminima mašinskog učenja, kakvi god da su podaci, teza o Raselovom čajniku je sa njima saglasna. To je upravo ono što važi u slučaju beskonačne VC dimenzije – kakvi god da su podaci, postoji model koji ih objašnjava.

Primer Raselovog čajnika nekome može delovati nezanimljiv, pošto u njega niko ni u jednom trenutku nije poverovao. To sa naučne tačke gledišta nije ni najmanje relevantan kriterijum, a puno puta se ispostavilo da važe stvari koje niko nije očekivao (npr. opšta relativnost, Gedelove teoreme, teorija evolucije, i slično). Ipak, postoje i teorije koje su bile naučno prihvaćene, ali su nepore-

cive. Primer je Ptolemejeva geocentrična teorija, koja pretpostavlja da se sva nebeska tela kreću oko Zemlje ili u krugovima ili u krugovima čiji su centri na prethodno uvedenim krugovima i tako dalje. Ova teorija je bila u stanju da uvođenjem dovoljnog broja kurugova na krugovima objasni proizvoljno kretanje na nebu. Štaviše, pokazano je da se ovakvim funkcijama može aproksimirati bilo koje zamislio kretanje (ovo znači beskonačnu VC dimenziju sistema funkcija). To znači da ne postoji baš nikakvo opaženo kretanje koje može pobiti ovu teoriju, odnosno da je ona neporeciva. S druge strane, heliocentrična teorija pretpostavlja da se nebeska tela kreću oko drugih po elipsama, pri čemu je telo oko kojeg se drugo kreće u žiži elipse i pri čemu radijus vektor između dva tela u istim vremenskim intervalima prelazi iste površine. Ukoliko se desi da Merkur malo odstupi od predviđene putanje oko Sunca, heliocentročna teorija u obliku u kojem je znamo će biti oborenja.⁵ Upravo je održavanje teorije, uprkos testovima u kojima je mogla biti oborenja osnov poverenja u tu teoriju.

2.8 Vrste modela

Modeli mašinskog učenja se mogu podeliti na tri vrste, prema tome koliku količinu informacije pokušavaju da modeluju – na *probabilističke generativne modele*, *probabilističke diskriminativne modele* i *neprobabilističke diskriminativne modele*.

2.8.1 Probabilistički generativni modeli

Probabilistički generativni modeli, ili kraće generativni modeli, modeluju zajedničku raspodelu najčešće datu svojom gustinom $p(x, y)$, koja opisuje ne samo zavisnosti između atributa i ciljne promenljive, već i zavisnosti među samim atributima. Pomoću takve raspodele bilo bi moguće vršiti predviđanja za bilo koji podskup promenljivih ukoliko su poznate vrednosti nekih od promenljivih. Na primer, neka je $x = (x_1, x_2, x_3, x_4)$. Tada, ukoliko su poznate vrednosti promenljivih x_2 i y , moguće je predvideti ostale na sledeći način

$$(x_1^*, x_3^*, x_4^*) = \underset{x_1, x_3, x_4}{\operatorname{argmax}} p(x_1, x_3, x_4 | x_2, y)$$

Pri čemu se gustina raspodele $p(x_1, x_3, x_4 | x_2, y)$ dobija na sledeći način

$$p(x_1, x_3, x_4 | x_2, y) = \frac{p(x_1, x_2, x_3, x_4, y)}{p(x_2, y)} = \frac{p(x_1, x_2, x_3, x_4, y)}{\int \int \int p(x_1, x_2, x_3, x_4, y) dx_1 dx_3 dx_4}$$

Pored samog predviđanja, ako je poznata raspodela verovantoće, moguće je i izračunati pouzdanost predviđanja pomoću intervala poverenja. Ukoliko je poznata zajednička raspodela podataka, moguće je i generisati nove podatke iz

⁵Zapravo, relativistički efekti dovode do malih razlika, pa heliocentrična teorija i nije precizna u svojoj izvornoj formulaciji.

te raspodele, koji po svojim statističkim svojstvima liče na dostupne podatke, što je nekada takođe korisno.

S druge strane, modelovanje zajedničke raspodele zahteva značajnu količinu podataka, a samim tim i vreme za obučavanje. Naime, ukoliko je potrebno uočiti zavisnosti između velikog broja promenljivih, potrebno je u skupu za obučavanje imati veliki broj kombinacija vrednosti tih promenljivih. Zapravo, kako se dimenzionalnost podataka povećava, da bi se održala gustina podataka u nekom jediničnom intervalu, potrebno je da broj podataka eksponencijalno raste! A gustinu raspodele nije moguće dobro oceniti ako gustina podataka nije zadovoljavajuća. Ovo je primer ozloglašenog problema mašinskog učenja, poznatog pod nazivom *prokletstvo dimenzionalnosti* (eng. *curse of dimensionality*). Ovo nije sveprisutan problem u mašinskom učenju, ali modelovanje zajedničke raspodele je tipičan kontekst u kojem se javlja.

U praktičnim kontekstima, predviđanje najčešće funkcioniše od atributa ka ciljnoj promenljivoj i kako su vrednosti atributa date, modelovanje zajedničke raspodele obično nije neophodno, pa nema osnova ni zahtevati tako velike količine podataka za modelovanje odnosa koji nisu važni. Stoga, ukoliko dati problem to ne zahteva, generativni model nije prvi model koji bi trebalo primeniti.

2.8.2 Probabilistički diskriminativni modeli

Probabilistički diskriminativni modeli, modeluju uslovnu raspodelu $p(y|x)$, koja opisuje samo zavisnost ciljne promenljive od atributa. Ne i zavisnosti među atributima. Kako su vrednosti atributa tipično date i samo vrednost ciljne promenljive nedostaje, to je dovoljno u većini praktičnih konteksta. Povrh predviđanja, koje se vrši na sledeći način

$$y^* = \operatorname{argmax}_y p(y|x)$$

i ovi modeli zahvaljujući poznavanju raspodele pružaju i procenu pouzdanosti predviđanja. Pomoću njih nije moguće generisati podatke, ali to najčešće nije ni potrebno. Pošto ne modeluju odnose među atributima, ukoliko se modeluje raspodela samo jedne ciljne promenljive, ne pate od prokletstva dimenzionalnosti.

U ovom kontekstu je važna jedna napomena. Ako bi se za date vrednosti atributa x modelovao veliki broj ciljnih promenljivih odjednom, uzimajući u obzir njihove međuzavisnosti, uprkos uslovnoj formi raspodele, taj model bi ponovo trebalo smatrati generativnim u odnosu na ciljne promenljive. Problemi poput prokletstva dimenzionalnosti su utoliko blaži što nije potrebno modelovati odnose među atributima, ali broj potrebnih podataka ponovo raste eksponencijalno sa brojem ciljnih promenljivih koje se modeluju.

2.8.3 Nепропабилистички дискриминативни модели

Nепропабилистички дискриминативни модели моделишу само функцију $y = f(x)$ и не дaju никакву информацију о расподели било које од променљивих. Отуд не дaju ни информацију о pouzданости предвиђања. Није реткост да се неки од непропабилистичких модела надогради тако да постане probabilistički, али то није neophodno.

2.9 Димензије дизајна алгоритама nadgledanog učenja

Алгоритми nadgledanog mašinskog učenja se могу зnačajno razlikovati по mnogim svojstvima, konstrukцији и намени. Ипак, код многих од њих је могуће приметити zajedničку структуру, односно уочити да представљају instance jedne општије sheme dizajna. Poznavanje ове sheme је корисно како прilikom dizajna алгоритама, тако и прilikom razumevanja постојећих алгоритама зato што се испоставља да različiti aspekti ponašanja алгоритама učenja zavise od конкретних odluka donetih prilikom dizajna i da ih је могуће povezati sa različitim dimenzijama pomenute sheme. Под dimenzijama neformalno подразумевамо različite elemente koji se могу birati kako bi se konstruisao алgoritam. Димензије дизајна алгоритма nadgledanog učenja, sa неким примерима могуćih избора, су угрубо sledeće:

- Vrsta modela – generativni или неки од дискриминативних.
- Forma modela – linearni, zasnovан наinstancama, neuronska mreža itd.
- Funkcija greške – srednjekvadratna, prosečna apsolutna, unakrsna entropija itd.
- Regularizacija – ℓ_1 , ℓ_2 , grupна itd.
- Optimizacioni algoritam – gradijentni spust, Nestorovljev algoritam, Adam, itd.

Iзбори по različitim dimenzijama se nekad mogu doneti nezavisno, ali nije retkost da неки од избора по jednoј dimenziji nije kompatibilan sa неким izborima по другим dimenzijama. На primer, gradijentni spust ne функционише добро са ℓ_1 regularizacijom, која користи $\|\cdot\|_1$ норму (sumu apsolutnih vrednosti razlika koordinata dva vektora), zbog njene nediferencijabilности, односно nemogućnosti računanja gradijenата u svim tačкама. Такви проблеми се prevazilaze ili primenom drugih постојећих алгоритама ili razvojem novih koji imaju odgovarajuća svojstva.

Glava 3

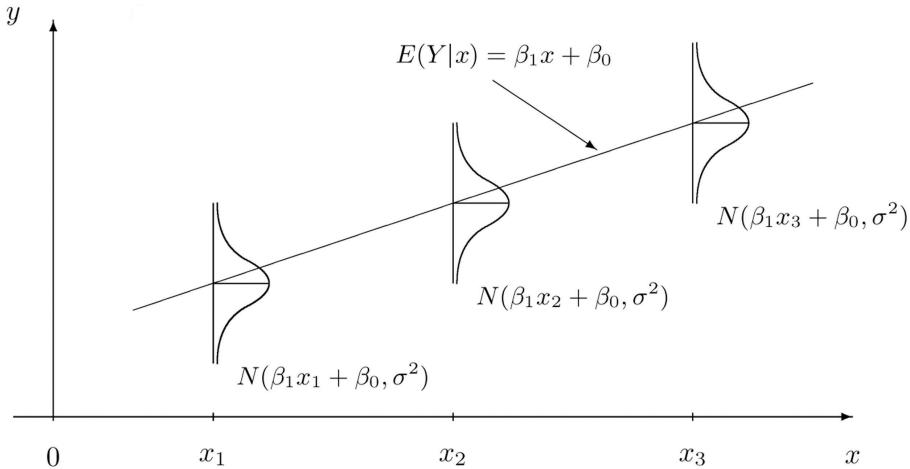
Probabilistički modeli

Verovatnoća predstavlja prirodan okvir za modelovanje neizvesnosti koja je sveprisutna u induktivnom zaključivanju. Otud se veliki broj algoritama mašinskog učenja u nekoj meri oslanja na probabilističke koncepte. Neki u potpunosti. Dizajn algoritama probabilističkih modela mašinskog učenja se zasniva na definisanju raspodele verovatnoće koju je potrebno oceniti iz podataka. Kako bi ocena, a kasnije i zaključivanje bili računski izvodljivi, obično se uvode neke pretpostavke vezane za to kako promenljive zavise jedne od drugih i kakva je forma raspodele. Forma raspodele, na primer, može biti normalna, dok struktura zavisnosti može biti izabrana tako da su vrednosti ciljnih promenljivih na različitiminstancama nezavisne ili da je zavisnost definsana nekim grafom ili nekako drugačije. U oba slučaja, izbor pogrešnih pretpostavki može značajno uticati na performanse algoritma. Ako je pretpostavljena neka forma raspodele koja ne odgovara raspodeli podataka, vrlo je verovatno da će predviđanja biti lošija, a još verovatnije je da će informacija o pouzdanosti koju model pruža biti nekorektna. Slično važi i ukoliko se pretpostavi nezavisnosti promenljivih koje su zapravo zavisne. Jednostavnosti radi, u nastavku ćemo se baviti samo modelima kod kojih se pretpostavlja međusobna nezavisnost vrednosti ciljne promenljive, ali naravno modeluje se zavisnost ciljne promenljive od atributa.

3.1 Linearna regresija

Linearna regresija predstavlja jedan od najjednostavnijih i najčešće korišćenih modela mašinskog učenja. Postoje različiti načini njenog uvođenja. Jedan je probabilistički i verovatno otkriva više o ovom metodu od ostalih. Sa probabilističke tačke gledišta, osnovna pretpostavka je pretpostavka normalne raspodele ciljne promenljive y , pri datim vrednostima atributa x . Odnosno, važi

$$p(y|x) = \mathcal{N}(f(x), \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - f(x))^2}{2\sigma^2}\right)$$



Slika 3.1: Prikaz normalne raspodele konstantne standardne devijacije sa linearnim modelom proseka.

gde je f funkcija koja uspostavlja vezu između atributa i očekivanja ciljne promenljive. U zavisnosti od forme ove funkcije, moguće je dobiti vrlo različite modele. Tehnički najjednostavnija forma modela je linearna:

$$f_w(x) = w \cdot x$$

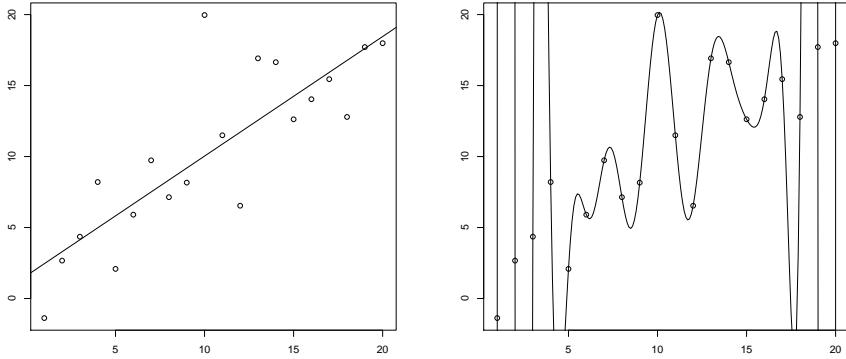
i ta pretpostavka je sastavni deo modela linearne regresije. Slika 3.1 prikazuje ovakvu raspodelu sa prosekom koji zavisi od neke promenljive. Najčešće se podrazumeva da je skup atributa proširen atributom koji je konstantne vrednosti 1, kako bi model sadržao slobodni koeficijent. Ukupni model je opisan na sledeći način:

$$p_w(y|x) = \mathcal{N}(w \cdot x, \sigma^2)$$

Kako modeluje uslovnu raspodelu, ovaj model je očigledno probabilistički diskriminativni.

Na slici 3.2, prikazan je primer dva poznata modela linearne regresije na istom skupu podataka. Kako linearost označava linearost po parametrima, ne treba da bude iznenađenje da linearni model može da predstavlja polinom. To je takođe linearna funkcija, ali nad bazom $(1, x, \dots, x^n)$.

Treba primetiti i da model pretpostavlja konstantnost varijanse normalne raspodele. Odnosno, očekuje se da je odstupanje predviđanja od ciljne vrednosti podjednako veliko i u slučaju velikih vrednosti ciljne promenljive i u slučaju malih. Ovo nije uvek realistično, a često nije ni poželjno. Recimo, ukoliko se model koristi za predviđanje dobiti u nekom poslu, greška od 10 hiljada dinara je zanemarljiva ukoliko je očekivana dobit oko 10 miliona dinara. Ipak, ako je očekivana dobit oko 10 hiljada dinara, onda je predviđanje sa takvom greškom potpuno beskorisno.



Slika 3.2: Dva modela dobijena linearnom regresijom. Jedan nad bazom $(1, x)$ i drugi nad bazom $(1, x, \dots, x^n)$.

Prilikom ocene parametara probabilističkih modela, tipično se koristi metod maksimalne verodostojnosti – potrebno je odrediti parametre za koje su dostupni podaci najverovatniji. Funkcija verodostojnosti je sledeća

$$\mathcal{L}(w) = p_w(y_1, \dots, y_N | x_1, \dots, x_N)$$

Uz prepostavku nezavisnosti instanci, dolazi se do jednostavnije forme ove funkcije:

$$\mathcal{L}(w) = \prod_{i=1}^N p_w(y_i | x_i)$$

Potrebno je rešiti problem

$$\max_w \mathcal{L}(w)$$

Funkcija verodostojnosti predstavlja proizvod gustina normalne raspodele izračunatim u različitim tačkama. Reprezentacija u vidu proizvoda se smatra nepoželjnom iz dva razloga. Prvi se tiče mogućnosti prekoračenja ili potkoračenja kada se množe veliki ili mali brojevi. Drugi se tiče praktične komplikovanosti izračunavanja parcijalnih izvoda proizvoda, koji su najčešće potrebni optimizacionim metodama. Stoga se umesto funkcije verodostojnosti češće koristi njen logaritam, kojim se proizvod prevodi u sumu. Kako je logaritam monotono rastuća funkcija, viša, pa i maksimalna, vrednost logaritma verodostojnosti nužno znači i višu vrednost verodostojnosti. Takođe, kako se u praksi češće govori o minimizaciji, nego o maksimizaciji, umesto logaritma verodostojnosti, češće se koristi njegova negativna vrednost (eng. *negative log likelihood*) ili skraćeno *NNL*, pa je problem koji se rešava sledeći:

$$\min_w -\log \mathcal{L}(w)$$

pri čemu važi

$$-\log \mathcal{L}(w) = -\sum_{i=1}^N \log p_w(y_i|x_i)$$

Primetimo da se funkcija $-\log p(y|x)$ može uzeti za funkciju greške. Ukoliko je verovatnoća vrednosti y za dato x velika, na primer, bliska 1, $-\log p(y|x)$ je blizu 0, a ako je mala, bliska nuli, ovaj izraz postaje ogroman. Nastavimo sa izvođenjem:

$$-\log \mathcal{L}(w) = -\sum_{i=1}^N \log \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - f_w(x_i))^2}{2\sigma^2}\right)$$

odnosno

$$-\log \mathcal{L}(w) = \frac{1}{2} \sum_{i=1}^N \log(2\pi\sigma^2) + \sum_{i=1}^N \frac{(y_i - f_w(x_i))^2}{2\sigma^2}$$

Kako prva suma i standardna devijacija ne zavise od parametara w , minimizacioni problem se svodi na

$$\min_w \sum_{i=1}^N (y_i - w \cdot x_i)^2$$

ili matrično

$$\min_w \|y - Xw\|^2$$

gde je X matrica redova x_1, \dots, x_N , a y kolona vektor (y_1, \dots, y_N) :

Lako je uveriti se da je kvadratna greška konveksna po parametrima w – svi sabirci su konveksne funkcije ovih parametara pošto se kvadrat koji je konveksna funkcija primenjuje na afinu transformaciju ovih parametara (pogledati dodatak 10). Ukoliko je moguće rešiti sistem $\nabla E(w) = 0$, problem je rešen.

$$\frac{\partial E(w)}{\partial w_j} = -2 \sum_{i=1}^N x_{ij} (y_i - w \cdot x_i) = 0$$

ili matrično:

$$X^T(y - Xw) = 0$$

$$X^T X w = X^T y$$

$$w = (X^T X)^{-1} X^T y$$

Izraz $(X^T X)^{-1} X^T$ naziva se *Mur-Penrouzovim pseudoinverzom* (eng. *Moore-Penrose pseudoinverse*) matrice X . Evo razloga za takav naziv. Norma $\|y - Xw\|$ bi mogla biti najmanje 0 ukoliko bi matrica X bila kvadratna i invertibilna. Tada bi rešenje bilo lako $w = X^{-1}y$. Kako ovaj uslov uopšte nije

realističan, rešenje je bilo potrebno izvesti na prikazani način (mada ima i drugih). Ali Mur-Penrouzov pseudoinverz se ponaša prilično slično inverzu:

$$(X^T X)^{-1} X^T X = I$$

Množenjem matrice X njenim pseudoinverzom sleva, dobija se jedinična matrica, baš kao da je kvadratna i inveritibilna matrica množena svojim inverzom.

Kako bi ovaj model dao interval poverenja, potrebno je oceniti i parametar σ^2 . I ovo se može raditi ocenom maksimalne verodostojnosti. Ispostavlja se da je preporučena ocena

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

Ipak, ova ocena nije korisna u praksi. U slučaju preprilagođavanja, ova ocena nije od koristi. Na primer, ova ocena može biti i 0, što sigurno ne znači da će naša predviđanja biti savršeno pouzdana. Naprotiv, samo je ocena varijanse loša zbog preprilagođavanja. Otud bi bolja strategija bila oceniti varijansu na odvojenom skupu podataka koji nije korišćen za obučavanje.

Algebarska forma rešenja problema linearne regresije jasno ukazuje na potencijalni problem. Matrica $X^T X$ ne mora biti invertibilna! Ovo se može desiti u slučaju linearnih zavisnosti među njenim kolonama. Da li je to realistično? Ukoliko se među podacima nalaze nabavna cena proizvoda i porez plaćen na taj proizvod, već te dve kolone su međusobno linearno zavisne, pa time i kolone matrice ukupno. Čak i ako matrica jeste invertibilna, može biti loše uslovljena zato što su kolone iako ne linearne zavisne, ipak visoko korelisane. Loša uslovlenost znači da za male promene elemenata matrice, moguće je dobiti drastično različite inverze ili rešenja odgovarajućeg sistema jednačina. Ovo je vrlo nepoželjno svojstvo u praksi, jer podatke nikada ne znamo sa savršenom tačnošću, odnosno male promene u odnosu na realne podatke su opšte mesto praktične primene. Postoji više načina rešavanja ovog problema, ali jedan, koji nam je već poznat, je regularizacija, odnosno rešavanje problema

$$\min_w \|y - Xw\|^2 + \frac{\lambda}{2} \|w\|_2^2$$

gde je polovina tu zbog lepšeg izgleda rešenja:

$$w = (X^T X + \lambda I)^{-1} X^T y$$

Povećavanjem vrednosti parametra λ , invertovana matrica postaje sve bolje uslovljena, ali se naravno regularizovani problem udaljava od polaznog problema, kao što je sa regularizacijom uvek slučaj.

Još jedan praktični izazov i u polaznom i u regularizovanom problemu je potencijalno velika dimenzionalnost matrice $X^T X$. Ukoliko matrica X ima značajno više kolona nego vrsta, regularizacija i dalje omogućava invertibilnost matrice, ali njene dimenzije mogu postati prevelične za praktičnu upotrebu.

Stoga se od izведенog rešenja često odustaje u korist gradijentnih tehnika optimizacije koje će biti objasnjene kasnije.

Pored već pomenute nerealističnosti i potencijalne neadekvatnosti pretpostavke o konstantnoj varijansi, potencijalne neadekvatnosti linearne forme modela i mogućih problema u minimizaciji mogući su i drugi problemi. U podacima za obučavanje, neretko se dešava da se nađe mali broj podataka koji značajno odstupaju od zakonitosti koja važi za ostale podatke. Takvi podaci nazivaju se *odudarajućim* (eng. *outliers*). Kako takvi podaci utiču na model linearne regresije? Jedan način razmišljanja je probabilistički. Prstup obučavanju polazi od principa maksimalne verodostojnosti – potrebno nači parametre modela za koje su podaci najverovatniji. Kako verovatnoća nekog podatka eksponencijalno opada sa udaljavanjem od proseka normalne raspodele, da bi taj podatak, a time i proizvod verovatnoća svih podataka, bio iole verovatan, neophodno je približiti mu prosek. Na taj način prisustvo odudarajućih podataka može značajno uticati na dobijeni model i nekada učiniti da on loše modeluje trend u podacima koji bi u suprotnom mogao modelovati. Zbog toga se ovakvi podaci nekada odstranjuju iz skupa za obučavanje. Ipak, to ne treba raditi po automatizmu, već treba proučiti prirodu konkretnih podataka i uveriti se da je isključivanje takvih podataka opravданo. Alternativa izbacivanju je modelovanje podataka nekom drugom raspodelom umesto normalne, a koja sporije opada, pa time ne daje drastično male vrednosti odudarajućim podacima i time im daje manji uticaj na model, ali je onda potrebno izvesti i nov algoritam koji odgovara toj raspodeli.

Iz čisto algebarske perspektive, prethodna osetljivost linearne regresije na odudarajuće podatke se mogla uočiti iz korišćenja kvadratne greške. Naime, kvadrat će velike razlike koje odgovaraju odudarajućim podacima učiniti još većim i time će se proces optimizacije neproporcionalno fokusirati na smanjivanje grešaka modela na tim podacima.

Jedna važna praktična prednost linearnih modela je njihova *interpretabilnost*, odnosno mogućnost analize i interpretacije, kojom se saznaće nešto o vezama koje važe između ciljne promenljive i atributa. Naime, linearni model

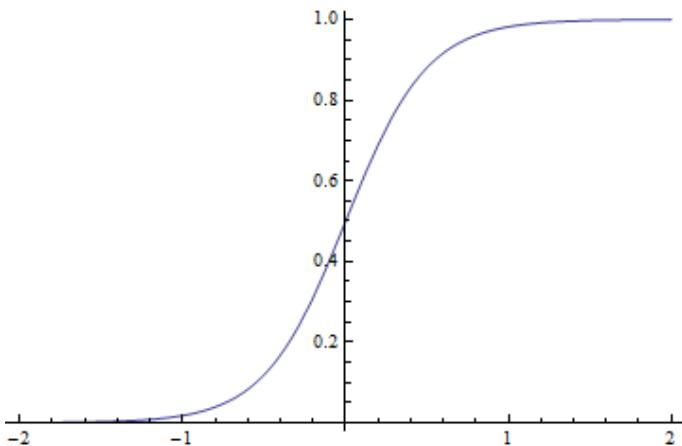
$$y = 5x_1 + 0x_2 - 0.5x_3$$

pokazuje da promena vrednosti promenljive x_1 proizvodi proporcionalnu promenu vrednosti ciljne promenljive, gde je koeficijent proporcionalnosti 5. Kako je ovaj koeficijent po apsolutnoj vrednosti veći od ostalih koeficijenata, zaključuje se da je promenljiva pojedinačno x_1 najvažnija za predviđanje vrednosti ciljne promenljive. Kako uz promenljivu x_2 стоји 0, jasno je da ta promenljiva uopšte nije korisna za predviđanje ciljne promenljive. Kada se njena vrednost menja, to se nikako ne odražava na ciljnu promenljivu. Promenljiva x_3 ima osetno manji uticaj od promenljive x_1 , ali dodatno primećujemo da je taj uticaj negativan. Često se kaže da je promenljiva x_1 pozitivno, a promenljiva x_3 negativno korelirana sa ciljnom promenljivom. Ovakvom analizom moguće je uočiti šta i koliko nam vrednosti promenljivih govore o vrednostima ciljne promenljive.

Prethodna diskusija važi samo ukoliko su zadovoljeni neki uslovi. Prvo, potrebno je da model ima malu grešku. Ukoliko model vrlo neuspješno predviđa vrednost ciljne promenljive, ovakva analiza nije od značaja. Drugo, promenljive se moraju meriti na istoj skali. Ukoliko se promenljiva x_1 meri u metrima, a promenljiva x_3 u milimetrima, ispostavlja se da mnogo drastičniju promenu ciljne promenljive uzrokuje promena promenljive x_3 za jedan metar, nego promenljive x_1 , iako koeficijenti sugerisu drugačije. Stoga je tipično da se pre primene linearne regresije izvrši *preprocesiranje* podataka kojim se sve promenljive svode na istu skalu. Često korišćen vid takve transformacije je *standardizacija* koja se sastoji u tome da se od svake vrednosti nekog atributa oduzme prosek svih vrednosti tog atributa, pa da se potom svaka vrednost tog atributa podeli standardnom devijacijom svih vrednosti tog atributa. Time se obezbeđuje da svaki atribut ima prosek 0 i standardnu devijaciju 1.

Transformacije poput standardizacije se ne koriste samo zbog interpretabilnosti, već i zbog boljih računskih svojstava, poput brže konvergencije metoda. Ipak, postoji još jedan kontekst u kojem je suštinski važno voditi računa o redovima veličine u kojima se promenljive izražavaju. Ukoliko promenljive nisu standardizovane, neki koeficijenti mogu biti veliki samo zbog skale na kojoj se vrednost promenljive meri. Ukoliko model uključuje regularizaciju, poput ℓ_2 regularizacije, taj koeficijent će biti više umanjen nego drugi koeficijenti, iako razlog za njegovu veličinu nije suštinski vezan za odnose među promenljivim. Otud je pravilo da se standardizacija ili neka slična transformacija vrši uvek.

Još jedno praktično razmatranje odnosi se na upotrebu kategoričkih promenljivih. Njima se mogu pridružiti numeričke oznake kako bi se predstavile u računaru, ali te numeričke oznake se ne mogu koristiti kao numerički atributi. Naime, ako su klase novinskih članaka *ekonomija*, *sport* i *politika* označene brojevima 0, 1 i 2 i u modelu im odgovara koeficijent w , pojava vrednosti *sport* utiče na ciljnu vrednost kao sabirak w , a pojava vrednosti *politika*, kao sabirak $2w$, dok pojava vrednosti *ekonomija* ne utiče uopšte. Ovakva aritmetika sa kategoričkim atributima nema smisla, a dodatno se postavlja pitanje šta bi bilo kada bismo drugačije označili različite kategorije. Otud se ovakav pristup *nikad* ne koristi. Umesto njega, koristi se *binarno kodiranje* (eng. *dummy coding*) tako što se uvode nove promenljive kojima se predstavljaju kategoričke promenljive. Ukoliko kategorička promenljiva x ima C vrednosti, uvodi se $C - 1$ novih binarnih promenljivih x_1, \dots, x_{C-1} takve da se i -ta kategorija za $1 \leq i \leq C - 1$ predstavlja vrednostima $(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_{C-1}) = (0, \dots, 0, 1, 0, \dots, 0)$, dok se kategorija C predstavlja vrednosima $(x_1, \dots, x_C) = (0, \dots, 0)$. Zapravo, pridruživanje kategorija datim kombinacijama bitova je proizvoljno. Bilo koje je prihvatljivo dok god se sva razlikuju. Šta bi bilo ako bismo umesto datog kodiranja koristili kodiranje koje ne definiše kategoriju C na specijalan način, već primenjuje dato kodiranje za $0 \leq i \leq C$? U tom slučaju kolone koje odgovaraju novim promenljivim bi se uvek sumirale na 1. Kako se slobodni član može videti baš kao parametar koji uvek množi jedinicu, i njemu u matrici podataka odgovara jedinica, što znači da su te kolone i kolona jedinica međusobno linearно zavisne, a kao što je već naglašeno, to vodi neinvertibilnosti matrice



Slika 3.3: Sigmoidna funkcija.

$$X^T X.$$

3.2 Logistička regresija

Dok linearna regresija predstavlja regresioni model, logistička regresija, uprkos svom nazivu, predstavlja model binarne klasifikacije. Osnovna pretpostavka sa probabilističke tačke gledišta je prepostavka Bernulijeve raspodele ciljne promenljive y , pri datim vrednostima atributa x . Drugim rečima za date vrednosti atributa x , postoji parametar $\mu \in [0, 1]$ tako da važi

$$p(y|x) = \begin{cases} \mu, & y = 1 \\ 1 - \mu, & y = 0 \end{cases}$$

pri čemu je $p(y|x)$ diskretna funkcija raspodele. Umesto ovog izraza, češće se piše samo $p(y = 1|x) = \mu$, dok se vrednost za $p(y = 0|x)$ odatile jednoznačno izračunava. Ovaj model nije kompletan, jer nije ustanovljena zavisnost parametra μ od vrednosti atributa x . Kako taj parametar mora biti u intervalu $[0, 1]$ da bi verovatnoća bila ispravno definisana, do sada korišćeni linearni model nije prihvatljiv. Ipak, ukoliko bi se vrednost linearne modela, koja može biti u intervalu $[-\infty, \infty]$, transformisala nekom nengativnom, monotonom, neprekidnom i diferencijabilnom funkcijom u interval $[0, 1]$, takav model bi bio prihvatljiv. Jedna takva funkcija, koja se često koristi u mašinskom učenju, je *sigmoidna funkcija*:

$$\sigma(t) = \frac{1}{1 + \exp(-t)}$$

Njen grafik je prikazan na slici 3.3. Postoje i druge funkcije koje zadovoljavaju navedene kriterijume. Jedan od razloga za korišćenje sigmoidne funkcije je

jednostavno izračunavanje njenog izvoda, što je u mašinskom učenju vrlo česta operacija. Naime, važi

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

tako da ukoliko je poznata vrednost funkcije, trivijalno se dobija i vrednost njenog izvoda.

Ukoliko se vrednost linearnog modela transformiše sigmoidnom funkcijom, model logističke regresije će biti određen sledećom relacijom:

$$p_w(y = 1|x) = \sigma(w \cdot x)$$

Time je definisana zavisnost parametra Bernulijeve raspodele od vrednosti atributa x . Verovatnoća $p_w(y = 1|x)$ pripadanja klasi 1 je utoliko veća, što je tačka x dalje od hiperravnih definisanih relacija $w \cdot x = 0$ u njenom pozitivnom poluprostoru. Verovatnoća pripadanja drugoj klasi je utoliko veća što je tačka dublje u negativnom poluprostoru.

Primetimo da se izrazom $\sigma(w \cdot x)$ ne definiše raspodela, već pre verovatnoće da je $y = 1$. Puna specifikacija se može zapisati u obliku

$$p_w(y|x) = \sigma(w \cdot x)^y (1 - \sigma(w \cdot x))^{1-y}$$

Očito, kada se umesto y uvrsti 1 ili 0, dobijaju se očekivane vrednosti verovatnoće. Kako modeluje uslovnu raspodelu, ovaj model je očigledno probabilistički diskriminativni.

Obučavanje ovog modela, odnosno ocena njegovih parametara se i u ovom slučaju zasniva na principu maksimalne verodostojnosti. Funkcija verodostojnosti je data izrazom

$$\mathcal{L}(w) = p_w(y_1, \dots, y_N | x_1, \dots, x_N)$$

Kao i u slučaju linearne regresije, z pretpostavku nezavisnosti instanci, dolazi se do jednostavnije forme ove funkcije:

$$\mathcal{L}(w) = \prod_{i=1}^N p_w(y_i|x_i)$$

i potrebno je rešiti problem

$$\max_w \mathcal{L}(w)$$

Prelaskom na negativnu vrednost logaritma funkcije verodostojnosti, dobija se

$$\begin{aligned} -\log \mathcal{L}(w) &= -\sum_{i=1}^N \log(\sigma(w \cdot x)^y (1 - \sigma(w \cdot x))^{1-y}) = \\ &= -\sum_{i=1}^N y \log \sigma(w \cdot x) + (1 - y) \log(1 - \sigma(w \cdot x)) \end{aligned}$$

Za ovu funkciju se može pokazati da je konveksna po w . Obično ima (globalni) minimum, koji se obično pronađe pomoću Njutnove metode ili bilo kojom gradijentnom metodom.

Primetimo da $L(u, v) = -u \log v - (1-u) \log(1-v)$ predstavlja funkciju greške koja se naziva *unakrsnom entropijom* i koristi se često u kontekstu probabilističke klasifikacije. Lako je videti da ova funkcija ima smisla u ulozi funkcije greške. Ukoliko je $u = 1$ i $v = 0$, zbog $\log v$, dobija se beskonačna greška. Analogno u slučaju kad je $u = 0$ i $v = 1$. S druge strane, kada je $u = 1$ i $v = 1$ ili $u = 0$ i $v = 0$ uz dogovor $0 \log 0 = 0$, važi $L(u, v) = 0$. Odnosno, kada nema greške vrednost funkcije je 0, a kad je imena vrednost je pozitivna, što je očekivano ponašanje funkcije greške.

Napomenimo i da se logistička regresija tipično ne koristi tačno u prikazanom obliku, već se najčešće koristi regularizovana varijanta. Sledeći primer demonstrira zanimljivo neregularizovane logističke regresije.

Primer 1 Neka je dat skup za obučavanje $\mathcal{D} = \{(-1, 0), (1, 1)\}$ takav da postoji jedan atribut i ciljna promenljiva. Minimizacioni problem se može svesti na sledeći:

$$\min_{w_0, w_1} -\log(1 - \sigma(w_0 - w_1)) - \log \sigma(w_0 + w_1)$$

odnosno

$$\min_{w_0, w_1} \log(1 + \exp(w_0 - w_1)) + \log(1 + \exp(-w_0 - w_1))$$

Nije teško uveriti se da ova funkcija nema minimum uprkos konveksnosti, već monotono opada sa povećanjem parametra w_1 . Ovo znači da gradijent nikad neće biti nula i da gradijentne metode optimizacije neće konvergirati osim usled zadovoljenja unapred zadate preciznosti. Očito, to što model za tekuće vrednosti parametara ispravno klasificuje podatke u skupu za obučavanje, ne znači da greška ne postoji. Ali onda se postavlja pitanje u čemu se sastoji ta greška? Vratimo se na polaznu formulaciju problema. Potrebno je maksimizovati verodostojnost bernulijeve promenljive. Za dati skup podataka, verodostojnost se maksimizuje kada važi $p(y=1|x=-1) = 0$ i $p(y=1|x=1) = 1$. Međutim, model prepostavlja da važi $p(y=1|x) = \sigma(w_0 + w_1 x)$. Sigmoidna funkcija ne može uzeti vrednosti 0 i 1, ali im se može približiti proizvoljno blizu. Otud je jasno da data formulacija kažnjava nesigurnost modela u situaciji u kojoj bi posmatrajući skup za obučavanje mogao biti potpuno siguran u svoja predviđanja. Takođe, što se procesom optimizacije više uveća koeficijent w_1 , to je model sigurniji u predviđanja na skupu za obučavanje. Da li je ovakvo ponašanje poželjno? Čitalac bi trebalo da je primetio da ovo zapravo predstavlja preprilagođavanje. Nema smisla očekivati od modela koji je obučen na dve instance da bude potpuno siguran u predviđanje. Tako nešto je vrlo nepoželjno. To je još jedna ilustracija potrebe za korišćenjem regularizacije.

Osvrnamo se na dato ponašanje na još jedan način. Optimalna vrednost za w_0 je 0, tako da vredi posmatrati samo funkciju $\sigma(w_1 x)$. Podešavanjem parametra w_1 kontroliše se strmost uspona sigmoidne funkcije u okolini nule.

Kako w_1 teži beskonačnosti, tako data funkcija teži indikatorskoj funkciji $I(x \geq 0)$.¹ To bi značilo da je model siguran da su sve instance za koje je $x < 0$ instance klase 0, a sve instance za koje je $x > 0$ instance klase 1, uprkos tome što u skupu za obučavanje nema ni jedne instance čija je vrednost atributa x unutar intervala $(-1, 1)$ i moglo bi se desiti da je prava granica između dve klase bilo gde unutar tog intervala, kao i da uopšte ne postoji jasna granica, pa osnova za potpunu sigurnost modela nema.

Jedan čest pogled na logističku regresiju je taj da modeluje logaritam količnika verovatnoća (eng. *log odds ratio*) dve različite klase linearnim modelom. Naime, važi

$$\log \frac{p(y=1|x)}{p(y=0|x)} = \log \frac{\frac{1}{1+\exp(-w \cdot x)}}{\frac{\exp(-w \cdot x)}{1+\exp(-w \cdot x)}} = w \cdot x$$

3.3 Multinomijalna logistička regresija

Multinomijalna logistička regresija predstavlja metod višeklasne klasifikacije. Uprkos nazivu, ipak ne počiva na multinomijalnoj, već na *kategoričkoj raspodeli*. Kao što binomna raspodela predstavlja raspodelu broja uspeha u N realizacija Bernulijeve promenljive, tako multinomijalna raspodela predstavlja raspodelu broja različitih ishoda pri reailzacijama kategoričke raspodele. Kategorička raspodela je raspodela koja svakom ishodu i , iz konačnog skupa ishoda, dodeljuje verovatnoću p_i . Stoga, multinomijalna logistička regresija ocenjuje kategoričku raspodelu, odnosno:

$$p(y|x) = \text{Cat}(p_1, \dots, p_C) = \begin{cases} p_1, & y = 1 \\ p_2, & y = 2 \\ \vdots & \vdots \\ p_C, & y = C \end{cases}$$

gde je C broj klasa, pri čemu mora važiti $p_1 + \dots + p_C = 1$ i $p_i \geq 0$ za svako $i = 1, \dots, C$. Očito, ova metoda je u stanju da vrši klasifikaciju u više klasa.

Ova raspodela se može modelovati analogno modelovanju koje vrši model logističke regresije. Krenimo od poslednje relacije u prethodnom odeljku. Modelujmo linearnim modelom logaritam odnosa verovatnoća svih klasa u odnosu na jednu privilegovanu. Na primer, poslednju.

$$\log \frac{p(y=i|x)}{p(y=C|x)} = w_i \cdot x \quad i = 1, \dots, C \quad (3.1)$$

Specifično za $i = C$ mora važiti

$$0 = \log \frac{p(y=C|x)}{p(y=C|x)} = w_i \cdot x$$

¹Potpuno precizno, teži funkciji koja je jednaka datoj indikatorskoj funkciji svuda osim u nuli, gde ima vrednost 0.5.

za svako x , odnosno $w = 0$, odnosno $\exp(w_C \cdot x) = 1$. Poslednja činjenica će biti korišćena više puta u nastavku. Primenom eksponencijalne funkcije na obe strane relacija datih jednakostima 3.1 dobija se:

$$p(y = i|x) = p(y = C|x) \exp(w_i \cdot x)$$

Sumirajući ovakve jednakosti po $i = 1, \dots, C - 1$, dobija se

$$1 - p(y = C|x) = p(y = C|x) \sum_{i=1}^{C-1} \exp(w_i \cdot x)$$

$$p(y = C|x) = \frac{1}{1 + \sum_{i=1}^{C-1} \exp(w_i \cdot x)} = \frac{\exp(w_C \cdot x)}{\exp(w_C \cdot x) + \sum_{i=1}^{C-1} \exp(w_i \cdot x)} = \frac{\exp(w_C \cdot x)}{\sum_{i=1}^C \exp(w_i \cdot x)}$$

Uvrštavanjem u formulu za $p(y_i|x)$ dobija se

$$p(y = i|x) = \frac{\exp(w_i \cdot x)}{\sum_{i=1}^C \exp(w_i \cdot x)} \quad i = 1, \dots, C$$

Za $C = 2$, dobija se baš logistički model.

Izvođenje optimizacionog problema je analogno izvođenju za logističku regresiju uz jedan dogovor. Neka je vrednost ciljne promenljive $y = i$ predstavljena vektorom dužine C , koji ima sve elemente 0, osim na mestu i na kojim ima vrednost 1. Funkcija verodostojnosti je onda data izrazom:

$$\mathcal{L}(w) = \prod_{i=1}^N \prod_{j=1}^C \left(\frac{\exp(w_j \cdot x_i)}{\sum_{k=1}^C \exp(w_k \cdot x_i)} \right)^{y_{ij}}$$

Optimizacioni problem je onda

$$\max_w \mathcal{L}(w)$$

Prelaskom na negativnu vrednost logaritma funkcije verodostojnosti, dobija se

$$\begin{aligned} -\log \mathcal{L}(w) &= -\sum_{i=1}^N \sum_{j=1}^C y_{ij} \log \left(\frac{\exp(w_j \cdot x_i)}{\sum_{k=1}^C \exp(w_k \cdot x_i)} \right) = \\ &= -\sum_{i=1}^N \left[\sum_{j=1}^C y_{ij} w_j \cdot x_i + \sum_{j=1}^C y_{ij} \log \left(\sum_{k=1}^C \exp(w_k \cdot x_i) \right) \right] \end{aligned}$$

Kako $\log \sum_{k=1}^C \exp(w_k \cdot x_i)$ ne zavisi od j i kako je $\sum_{j=1}^C y_{ij} = 1$, dobija se minimizacioni problem

$$\min_{w_1, \dots, w_{C-1}} -\sum_{i=1}^N \left[\sum_{j=1}^C y_{ij} w_j \cdot x_i + \log \left(\sum_{k=1}^C \exp(w_k \cdot x_i) \right) \right]$$

Podsećamo da je vektor koeficijenata w_C jednak 0, pa se po njemu ne minimizuje. Logaritam sume eksponencijalnih funkcija je konveksna funkcija, pa je i ceo problem konveksan i stoga pogodan za gradijentne metode optimizacije.

Primetimo da je model multinomijalne logističke regresije manje interpretabilan nego model standardne logističke regresije. Naime, potrebno je interpretirati koeficijente više modela, a u različitim modelima mogu dominirati parametri različitih atributa. Tada je teže proceniti koji atributi su važniji od drugih.

3.4 Uopšteni linearni modeli

Ispostavlja se da se svi pomenuti probabilistički modeli mogu predstaviti kao instance jedne opštije vrste modela – *uopštenih linearnih modela*. Uopšteni linearni modeli se sastoje iz tri komponente. Prva je forma raspodele $p(y|x)$, druga je linearni model i treća je *funkcija veze* (eng. *link function*) ili samo *vezu* koja povezuje taj linearni model sa očekivanjem raspodele. Odnosno, pretpostavlja se da važi

$$g(\mu) = w \cdot x$$

U slučaju linearne regresije, upravo je prosek bio modelovan linearnim modelom, pa se linearna regresija može videti kao instanca uopštenog linearog modela kod koje je uslovna raspodela normalna, a veza je identitet $g(\mu) = \mu$. U slučaju logističke regresije, veza je bila

$$g(\mu) = \log\left(\frac{\mu}{1 - \mu}\right)$$

a raspodela je bila Bernulijeva. Slična relacija se može izvesti i za multinomijalnu logističku regresiju, samo što će sve veličine biti vektorske.

Forma raspodele se obično bira iz *eksponencijalne familije*, što je familija kojoj pripadaju sve pomenute raspodele, ali i mnoge druge poput Puasonove, eksponencijalne, χ^2 , Dirlleove, beta i gama raspodele. Eksponencijalna familija ima mnoga poželjna svojstva, ali u njih ovde nećemo ulaziti. Forma gustine raspodele za ovu familiju je sledeća²

$$p(x) = \exp\left(\frac{\theta \cdot x - A(\theta)}{B(\sigma)} + c(x, \sigma)\right)$$

Nenegativan parametar σ naziva se parametrom disperzije, a vektor θ vektorom prirodnih parametrara.

Može se pokazati da važi

$$\nabla_{\theta} A(\theta) = \mathbb{E}[x] \quad \nabla_{\theta}^2 A(\theta) = \text{cov}[x]B(\sigma)$$

²Postoje i nešto drugačije formulacije.

Raspodela	Domen	Upotreba	Veza
Normalna	\mathbb{R}	neprekidne promenljive	$g(\mu) = \mu$
Eksponencijalna	\mathbb{R}^+	neprekidne pozitivne promenljive	$g(\mu) = \mu^{-1}$
Puasonova	\mathbb{N}	broja događaja u jedinici vremena	$g(\mu) = \log(\mu)$
Bernulijeva	$\{0, 1\}$	binarne promenljive	$g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$

Tabela 3.1: Raspodele eksponencijalne familije, njihovi domeni, vrste podataka na koje se primenjuju i kanonske funkcije veze.

Za raspodele koje smo do sada koristili se lako može pokazati da pripadaju eksponencijalnoj familiji. Za normalnu raspodelu važi

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \exp\left(\frac{x\mu - \mu^2/2}{\sigma^2} - \frac{1}{2}\left\{\frac{x^2}{\sigma^2} + \log(2\pi\sigma^2)\right\}\right)$$

$$\text{pa je } \theta = \mu, A(\theta) = \frac{\mu^2}{2} = \frac{\theta^2}{2}, B(\sigma) = \sigma^2 \text{ i } c(x, \sigma) = -\frac{1}{2}\left(\frac{x^2}{\sigma^2} + \log(2\pi\sigma^2)\right)$$

Za Bernulijevu raspodelu važi

$$p(x) = \mu^x(1-\mu)^{(1-x)} = \exp(x \log \mu + (1-x) \log(1-\mu))$$

$$\text{pa je } \theta = \log\left(\frac{\mu}{1-\mu}\right), A(\theta) = \log\left(\frac{1}{1-p}\right) \log(1 + \exp(\theta)), B(\sigma) = 1 \text{ i } c(x, \sigma) = 0.$$

Otud je $\mu = \sigma(\theta)$.

Slično se može pokazati i za kategoričku raspodelu.

Veza za koju važi $g(\mu) = \theta$ naziva se *kanonskom vezom*. U tabeli 3.1 prikazane su neke funkcije eksponencijalne familije i njihove kanonske veze. Ipak, veza koja se koristi ne mora nužno biti kanonska.

Uopšteni linearni modeli obično se ocenjuju metodom maksimalne verodostojnosti.

3.5 Naivni Bajesov algoritam

Naivni Bajesov algoritam se zasniva na modelovanju raspodele ciljne promenljive y pri datim vrednostima promenljive x , korišćenjem Bajesove formule:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

Najčešće se primenjuje na problem klasifikacije, pa se najčešće govori o naivnom Bajesovom klasifikatoru. Iako to nije neophodno, u nastavku ćemo pretpostaviti da su sve promenljive kategoričke. Pitanje koje bi odmah trebalo da se nametne je zašto je lakše modelovati desnu stranu jednakosti od leve i odgovor nije očigledan. Načelno je lako modelovati raspodele jedne promenljive. Ukoliko je promenljiva diskretna, kao što smo pretpostavili, moguće je oceniti njenu raspodelu pomoću frekvencija njenih različitih vrednosti u skupu za

obučavanje. Ipak, raspodela sa leve strane je uslovna i to nije lako jer bismo morali brojati pojavljivanja vrednosti promenljive y u slučajevima u kojima su odgovarajuće vrednosti atributa baš one date vektorom x . Ukoliko se baš te vrednosti atributa nisu našle u skupu za obučavanje ili su vrlo retke, onda nije lako adekvatno oceniti raspodelu promenljive y za dato x . Ako je vektor x visokodimenzionalan, to je i vrlo verovatno. Sa desne strane, lako je oceniti raspodelu promenljive y , pošto je ona jednodimenzionalna, ali x je visokodimenzionalno i prokletstvo dimenzionalnosti predstavlja prepreku za ocenu $p(x|y)$ i $p(x)$. Primetimo da $p(x)$ uopšte ne zavisi od y . Ta vrednost je ista za sve vrednosti promenljive y . Kako je u predviđanju potrebno samo naći najverovatniju vrednost ciljne promenljive, $p(x)$ se uopšte ne mora izračunavati, već je dovoljno izračunati vrednost u brojocu, odnosno nije važna tačna vrednost, već proporcionalnost:

$$p(y|x) \sim p(x|y)p(y)$$

Ipak, ovo ne olakšava problem zato što je i dalje potrebno modelovati raspodelu $p(x|y)$. Kako bi se prevazišlo prokletstvo dimenzionalnosti, prepostavlja se uslovna nezavisnost atributa kada je data vrednost ciljne promenljive, odnosno da važi:

$$p(x|y) = \prod_{i=1}^n p(x_i|y)$$

Obratimo pažnju da x_i ne predstavlja i -tu instancu, već vrednost i -og atributa. Uslovne raspodele $p(x_i|y)$ su jednodimenzionalne i lako se mogu modelovati – za dato y , na osnovu instanci iz skupa za obučavanje koje imaju vrednost ciljne promenljive baš y , modeluje se jednodimenzionalna raspodela promenljive x_i . Ukoliko je promenljiva x_i kategorička, to je jednostavno – svodi se na računanje frekvencija njenih različitih vrednosti. Kao što je rečeno u delu 10, uslovna nezavisnost je slabija pretpostavka od nezavisnosti, pa se uz takvu pretpostavku ne gubi sva informacija koja bi bila izgubljena uz punu pretpostavku o nezavisnosti. Ipak, ne možemo očekivati da će ova pretpostavka u praksi biti ispunjena, pa je rešenje aproksimativno, a epitet *naivni* potiče upravo od ove pretpostavke.

Pun model je dat relacijom

$$p(y|x) \sim \prod_{i=1}^n p(x_i|y)p(y)$$

Kako modeluje uslovnu raspodelu, deluje da se može reći da je model diskriminativni. Ipak, ova uslovna raspodela se modeluje tako što se modeluje $p(x|y)p(y)$, što je jednako $p(x, y)$. Odnosno, da bi se rešio lakši problem, rešava se teži – modelovanje zajedničke raspodele, pa se zapravo radi o generativnom modelu! Pretpostavka uslovne nezavisnosti ne znači da model nije generativni, već samo da možda nije dobar generativni model.

Primer 2 Kod lekara dolazi pacijent koji se žali na osećaj hladnoće i blagu glavobolju. Lekar nije bas najstručniji i pokušava da prepostavi dijagnozu pre-

Hladnoća	Curenje iz nosa	Glavobolja	Groznica	Grip
Da	Ne	Blaga	Da	Ne
Da	Da	Ne	Ne	Da
Da	Ne	Jaka	Da	Da
Ne	Da	Blaga	Da	Da
Ne	Ne	Ne	Ne	Ne
Ne	Da	Jaka	Da	Da
Ne	Da	Jaka	Ne	Ne
Da	Da	Blaga	Da	Da

Tabela 3.2: Tabela podataka za problem dijagnostifikovanja gripa.

turajući po kartonima drugih pacijenata, pokušavajući da ustanovi kakve su simptome imali pacijenti koji su imali grip, a kakve oni koji nisu. Kako nema mnogo vremena, mora da odlučuje na osnovu malog uzorka. U tabeli 3.2 dati su podaci o nekoliko pacijenata čije je kartone našao. Da je dobar matematičar (što je, imajući u vidu kakav je lekar, malo verovatno), lekar bi na osnovu ovog uzorka izračunao odgovarajuće proizvode koji su proporcionalni uslovnim verovatnoćama da pacijent ima grip i da nema grip:

$$p(Da|Da, Ne, Blaga, Ne) \sim p(H = Da|Da)p(C = Ne|Da)p(Gl = B|Da)p(Gr = Ne|Da)p(Grip = Da) \\ = \frac{3}{5} \cdot \frac{1}{5} \cdot \frac{2}{5} \cdot \frac{1}{5} \cdot \frac{5}{8} = \frac{3}{500}$$

$$p(Ne|Da, Ne, Blaga, Ne) \sim p(H = Da|Ne)p(C = Ne|Ne)p(Gl = B|Ne)p(Gr = Ne|Ne)p(Grip = Ne) \\ = \frac{1}{3} \cdot \frac{2}{3} \cdot \frac{1}{3} \cdot \frac{2}{3} \cdot \frac{3}{8} = \frac{1}{54}$$

Iako nije izračunao tačne verovatnoće da pacijent ima ili nema grip, lekar može da zaključi da je verovatnije da ga nema.

Prednost naivnog Bajesovog algoritma je da se ocene verovatnoća lako ažuriraju kako pristižu novi podaci – prebrojavanjem različitih vrednosti promenljivih. Takođe, iako je ovde diskutovan u kontekstu kategoričkih atributa, može se primeniti i nad kontinualnim ukoliko se diskretizuju ili ukoliko se pretpostavi neka forma raspodele atributa i uradi ocenu parametara te raspodele. Pored pretpostavke uslovne nezavisnosti atributa, postoji još jedna važna mana ovog algoritma. Ukoliko je neka vrednost nekog atributa malo verovatna, može se desiti da se ona ne pojavi u skupu za obučavanje. U tom slučaju verovatnoća te vrednosti je 0. Kada se u predviđanju pojavi ta vrednost, njen prisustvo u proizvodu verovatnoća $\prod_{i=1}^n p(x_i|y)$ čini da ceo proizvod bude nula. Ovakvo ponašanje očito nije poželjno, pa se umesto nule, ovakvim vrednostima dodeljuje neka vrlo mala pozitivna vrednost. Postoje i neke složenije tehnike za rešavanje ovog problema, ali se njima nećemo baviti.

Glava 4

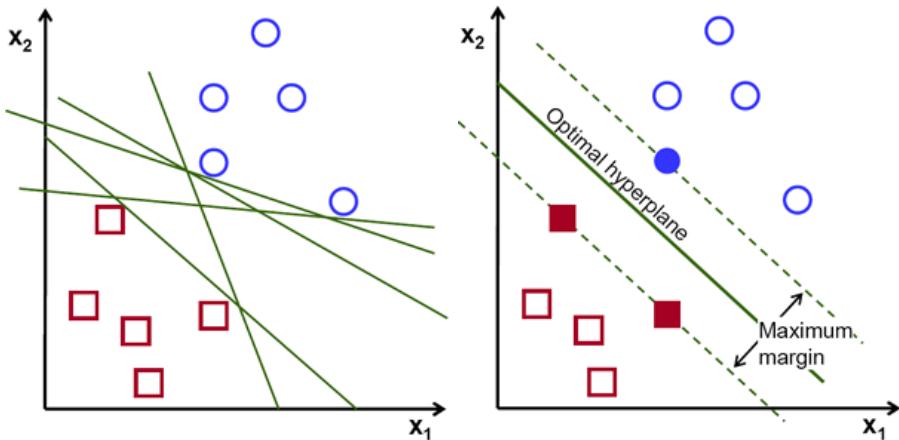
Modeli zasnovani na širokom pojasu

Pouzdanost predviđanja je od kritičnog značaja u mašinskom učenju. Pojam pouzdanosti se verovatno može definisati na više različitih načina. U ovoj glavi će biti razmotren jedan način dizajna algoritama mašinskog učenja koji se zasniva baš na razmatranju kada se predviđanje može smatrati pouzdanim. Radi se o *modelima zasnovanim na širokom pojasu* (eng. *large margin*).

Razmotrimo sledeći neformalan primer. Neka se održava trka na 100 metara. Pobednik se, pouzdanosti radi, proglašava na osnovu foto finiša. Ipak, kamera je omanula baš u toku trke. Pod kojim uslovima sudija može biti pouzdan da je ispravno proglašio pobednika? Ukoliko su prva dva trkača zajedno prošla kroz cilj, teško je reći ko je prvi. Odluka je pouzdana tek ako je prvi trkač ostavio drugog na bezbednom odstojanju! Razmotrimo još jedan primer. Dve populacije iste vrste mrava sakupljaju hranu relativno blizu jedni od drugih. Ukoliko entomolog uoči mrava na nekoj lokaciji, pod kojim uslovima može samo na osnovu lokacije (vrsta je ista, pa mravi isto izgledaju) biti siguran kojoj populaciji mrav pripada? Možda onož čiji je mravinjak najbliži? Mrav bi verovatno trebalo da bude blizu svog mravinjaka, ali to nije sigurno, pošto jedna populacija može pokrivati nešto veću površinu, a mogu i zalisti jedni drugima u teritoriju. Ukoliko bi utvrdio da ove dve populacije izbegavaju jedna drugu, odnosno da se drže jedna od druge na bezbednom odstojanju, mogao bi lako utvrditi koja teritorija pripada kojoj populaciji. U oba slučaja, od značaja je koncept bezbednog odstojanja, odnosno praznog prostora između objekata koje treba razlikovati. Način na koji se to može matematički definisati zavisi od konteksta, ali ključno je voditi se tom intuicijom.

4.1 Metod potpornih vektora za klasifikaciju

Metod potpornih vektora (eng. *support vector machine*) je jedan od važnijih metoda mašinskog učenja. Zasnovan je na jasnoj geometrijskoj intuiciji. Pret-



Slika 4.1: Prave koje razdvajaju dve klase

postavimo da imamo dve klase tačaka u ravni i neka su klase takve da se između elemenata te dve klase može povući prava, tako da su svi elementi jedne klase sa jedne strane, a elementi druge klase sa druge strane. Ovaj uslov linearne razdvojivosti nije realističan uslov, ali ćemo za sad pretpostaviti da važi. Ako nacrtamo različite rasporede takvih tačaka, primetićemo da prava koja ih razdvaja praktično nikad nije jedna, već da je moguće povući više njih. Ovo je prikazano na slici 4.1. Ipak, neke prave nam deluju bolje od ostalih. Na istoj slici je prikazana i optimalna prava, što je prava sa najvećim rastojanjem do najbliže joj tačke podataka, odnosno sa najširim pojasom praznog prostora oko nje. Intuitivno, posmatrajući sliku, prava koja bi bila pod drugačijim uglom i prolazila bliže nekoj od tačaka podataka bi nosila veći rizik da neka tačka koja nije u datim podacima završi sa pogrešne strane prave. Sada je prikazani princip potrebno formalizovati.

Jednačina hiperravnji je

$$w \cdot x + w_0 = 0$$

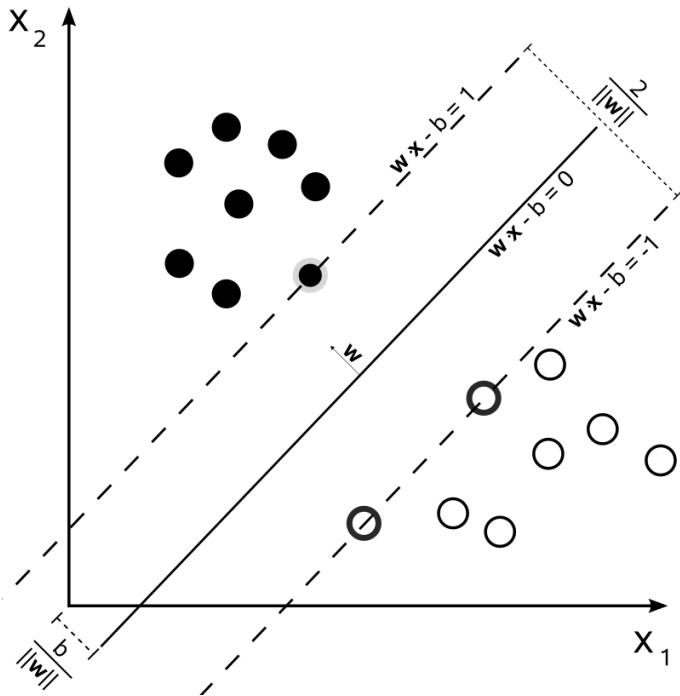
gde je w_0 slobodni član. *Optimalna hiperravan*, odnosno *hiperravan najšireg pojasa* je podjednako udaljena od najbližih predstavnika obe klase. Ako bi bila bliže jednoj klasi, mogla bi se udaljiti ka drugoj, kako bi se povećalo minimalno rastojanje. Stoga, hiperravnji paralelne optimalnoj imaju jednačine

$$w \cdot x + w_0 = c$$

$$w \cdot x + w_0 = -c$$

Deljenjem svih jednačina sa c , za neke nove koeficijente w i w_0 za koje ćemo zadržati iste oznake, dobijaju se jednačine sve tri hiperravnji:

$$w \cdot x + w_0 = 0$$



Slika 4.2: Optimalna hiperravan i paralelne joj hiperravnini koje leže na potpornim vektorima.

$$w \cdot x + w_0 = 1$$

$$w \cdot x + w_0 = -1$$

kao što je prikazano na slici 4.2. Tačke podataka koje se nalaze na pomenutim hiperravnima paralelnim optimalnoj nazivaju se potpornim vektorima, pošto deluju kao da pružaju potporu datom sistemu od tri hiperravnini – tako da ne može da mrdne ni levo ni desno! Po njima je ova metoda i dobila ime. Rastojanje između optimalne hiperravni i jedne od pomenutih hiperravnini koje su joj paralelne je upravo pojas koji treba da bude što veći. Na osnovu jednačine rastojanja tačke od hiperravnini

$$\frac{|w \cdot x + w_0|}{\|w\|_2}$$

i činjenice da za svaku od tačaka sa ovih hiperravnini važi $|w \cdot x + w_0| = 1$, dobija se da je ukupno rastojanje između klasa, u pravcu normalnom u odnosu na optimalnu hiperravan $2/\|w\|$. Otud se optimalna hiperravan dobija pronalaženjem koeficijenata koji maksimizuju ovaj izraz pod uslovima da su sve tačke sa pravih strana te hiperravnini. Ako se izrazimo u terminima minimizacije, umesto

maksimizacije dobijamo sledeći optimizacioni problem:

$$\min_{w, w_0} \frac{\|w\|_2}{2}$$

$$y_i(w \cdot x_i + w_0) \geq 1 \quad i = 1, \dots, N$$

pri čemu se podrazumeva da važi $y_i \in \{-1, 1\}$. Dodatni uslovi izražavaju potrebu da sve tačke budu na većem rastojanju od optimalne hiperravnih nego što su potporni vektori koji su na rastojanju 1. Za rešavanje ovog optimizacionog problema i problema izvedenih iz njega, koriste se posebno konstruisani algoritmi.¹

Suštinski problem sa ovom formulacijom predstavlja činjenica da u praksi retko možemo očekivati linearnu razdvojivost klasa. Prosto, stvarni problemi su komplikovani. Otud je neophodno prihvatići neke greške, uz zahtev da budu što manje. Do nove formulacije se dolazi uvođenjem novih promenljivih ξ_i za svaku instancu u skupu za obučavanje, koje mere koliko je svaka instanca daleko od hiperravnih odredene potpornim vektorima njene klase, ali samo pod pretpostavkom da je sa pogrešne strane. Ovakav metod naziva se metodom potpornih vektora sa *mekim pojasom* (eng. *soft margin*), a optimizacioni problem izgleda ovako:

$$\min_{w, w_0} \frac{\|w\|_2}{2} + C \sum_{i=1}^N \xi_i$$

$$y_i(w \cdot x_i + w_0) \geq 1 - \xi_i \quad i = 1, \dots, N$$

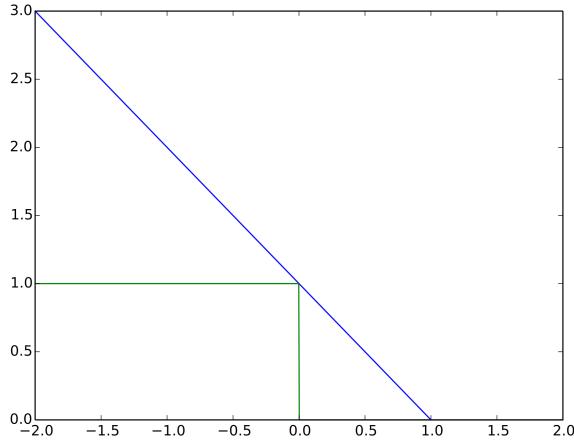
$$\xi_i \geq 0 \quad i = 1, \dots, N$$

Uloga metaparametra C koji mora biti nenegativan je da kontroliše koliko težine se pridaje greškama. Razmotrimo ponašanje ovog problema za različite vrednosti metaparametra C . Ukoliko važi $C = 0$, greške uopšte nisu važne i mogu biti proizvoljno velike. Otud je optimalno rešenje $w = 0$. Ukoliko je C ogromno, onda su greške izuzetno važne, a pravac hiperravnih i širina pojasa koji joj odgovara nisu mnogo važni.

U svetu ranije pomenute sheme dizajna algoritama nadgledanog učenja, zanimljivo je razmisiliti gde se u strukturi ovog problema krije koji od elemenata, konkretno funkcija greške i regularizacija. Primetimo da promenljiva ξ_i (bilo koja) ima vrednost veću od nule ako važi $y_i(w \cdot x_i + w_0) < 1$ i da je u tom slučaju njena vrednost najmanje $1 - y_i(w \cdot x_i + w_0)$. Kako se u minimizaciji insistira na njenim što manjim vrednostima, onda možemo smatrati da važi baš jednakost $\xi_i = 1 - y_i(w \cdot x_i + w_0)$. Dodatno, kako ova promenljiva ne može biti negativna, već je u slučaju da važi $y_i(w \cdot x_i + w_0) > 1$ jednaka 0, zaključujemo da važi

$$\xi_i = \max(0, 1 - y_i(w \cdot x_i + w_0))$$

¹Na primer SMO (eng. *sequential minimal optimization*).



Slika 4.3: Greška u vidu šarke kao aproksimacija indikatorske funkcije. Vrednost greške $L(u, v)$ data je u odnosu na proizvod uv .

odnosno da se prethodni problem može predstaviti kao:

$$\min_{w, w_0} \frac{\|w\|_2}{2} + C \sum_{i=1}^N \max(0, 1 - y_i(w \cdot x_i + w_0))$$

što je ekvivalentno sa

$$\min_{w, w_0} \sum_{i=1}^N \max(0, 1 - y_i(w \cdot x_i + w_0)) + \lambda \|w\|_2$$

što nam je već dobro poznata forma. Primetimo da greška nije srednja nego ukupna, ali kako su te dve greške direktno proporcionalne, to ne menja problem, osim utoliko što će druga vrednost parametra λ biti optimalna. Takođe, za regularizaciju smo ranije umesto norme koristili njen kvadrat. Zaista, i u metodu potpornih vektora se češće koristi kvadrat norme, ali to nije od suštinskog značaja.

Razmotrimo funkciju greške $L(u, v) = \max(0, 1 - uv)$. Radi se o takozvanoj *funkciji greške u vidu šarke* (eng. *hinge loss*) koja, kao što prikazuje slika 4.3, predstavlja konveksnu aproksimaciju grešku klasifikacije $L(u, v) = I(u \neq v)$.

Može se pokazati da je rešenje optimizacionog problema oblika:

$$w = \sum_{i=1}^N \alpha_i y_i x_i$$

gde su α_i takozvani *Lagranžovi množioci* za koje važi $0 \leq \alpha_i \leq C$. Potporni vektori su instance x_i za koje važi $\alpha_i > 0$. Za vrednost w_0 važi

$$w_0 = -\frac{\min_{(x,1) \in \mathcal{D}} w \cdot x + \max_{(x,-1) \in \mathcal{D}} w \cdot x}{2}$$

Intuicija ovakvog izbora koeficijenta w_0 je da je optimalna hiperravan tačno na sredini između potpornih vektora dve klase. Model je onda dat funkcijom

$$f_{w,w_0}(x) = \sum_{i=1}^N \alpha_i y_i (x_i \cdot x) + w_0$$

a predviđena klasa je data znakom modela

$$\operatorname{sgn}(f_{w,w_0}(x))$$

Svi podaci osim potpornih vektora su nebitni, pošto rešenje ne zavisi od njih. Ovo omogućava i filtriranje podataka i detekciju onih koji su dovoljni da okarakterišu zavisnosti između atributa i ciljne promenljive. Ovo se u nekim primenama može iskoristiti za smanjenje količine skladištenih podataka. Iako odbacivanje podataka može zvučati loše, ukažimo jednom analogijom da to može biti i intuitivno. Recimo, u diskusiji osoba koja pruža veći broj argumentata može delovati ubedljivije, ako su ti argumenti iole kvalitetni. Ipak, da li se dobra argumentacija zasniva baš na količini argumentata? Ukoliko se uz pomoć nekoliko argumentata uspostavi verodostojnost neke tvrdnje, dalji argumenti su nezanimljivi i ne doprinose diskusiji, osim eventualno u nesuštinskom smislu retoričkog utiska.

Za Vapnik-Červonenkisovu dimenziju metoda potpornih vektora važi

$$h \leq \min(R^2 \|w\|^2, n) + 1$$

gde je R radijus sfere koja obuhvata sve podatke u skupu za obučavanje. Kako metod potpornih vektora minimizuje $\|w\|^2$, posredno minimizuje i Vapnik-Červonenkisovu dimenziju skupa funkcija iz kojih bira najbolju, čime se objašnjavaju njegove dobre performanse. Još jedno bitno zapažanje je da ukoliko je $R^2 \|w\|^2$ meanje od n , što tipično jeste, data gornja granica ne zavisi direktno od dimenzionalnosti prostora (iako neka zavisnost postoji kroz normu vektora w), zbog čega je ovaj metod često korišćen u primenama vezanim za visokodimenzionalne prostore, poput obrade prirodnog jezika i bioinformatike. U praksi se ponaša bolje od većine drugih metoda, zahvaljujući čemu je (uz izmene o kojima govorimo kasnije) bio najpopularniji algoritam mašinskog učenja devedesetih, dok primat nisu preuzele duboke neuronske mreže.

Do sada smo razmotrili dva klasifikaciona algoritma koji počivaju na razdvajanju klase hiperravnima – logističku regresiju i metod potpornih vektora. Bitno je razumeti u čemu je razlika. Poučno je razmisliti i o tome kako doći do odgovora na to pitanje. Već je rečeno da opšta shema dizajna algoritama

mašinskog učenja koja je skicirana u delu 2.9 može pomoći u razumevanju specifičnosti ponašanja algoritama. Oslonimo se na nju. Po pitanju vrste modela, razlika postoji – jedan je probabilistički, a drugi nije. Forma modela je ista – u oba slučaja radi se o hiperravnji. Funkcija greške je različita. Postoji razlika i u regularizaciji – ona je u metod potpornih vektora ugrađena po konstrukciji, a u logističku regresiju nije. Ipak, to nije suštinska razlika jer se regularizacija uvek može dodati u model logističke regresije. Optimizacioni metod je bitan za brzinu obučavanja, ali ukoliko je optimizacija uspešno završena približnim nalaženjem minimuma, nema posledica po ponašanje modela u predviđanju. Stoga to nije ni važno. Zaključujemo da različitost ova dva metoda može biti ili do probabilističke prirode logističke regresije ili do funkcije greške ili do oba. Zapitajmo se kako ta probabilistička priroda utiče na izbor modela? Konkretna probabilistička pretpostavka je vezana za Bernulijevu raspodelu i metod maksimalne verodostojnosti. Pažljivom analizom izvođenja optimizacionog problema, može se videti da ona direktno uslovjava izbor funkcije greške. Zaključujemo da se obe razlike svode na istu – funkciju greške! Stoga će se dalja analiza fokusirati na razlike u funkciji greške. Kako bi takva analiza bila moguća, potrebno je eliminisati određene razlike u formulacijama. Metod potpornih vektora podrazumeva da su oznake klase 1 i -1, dok logistička regresija podrazumeva oznake 0 i 1. Nije teško pokazati (eto korisne vežbe!) da u slučaju oznaka 1 i -1 problem logističke regresije ima formu

$$\min_w \sum_{i=1}^N \log(1 + e^{-y_i w \cdot x_i})$$

odnosno da je funkcija greške

$$L_{LR}(u, v) = \log(1 + e^{-uv})$$

a u slučaju metoda potpornih vektora, to je

$$L_{SV}(u, v) = \max(0, 1 - uv)$$

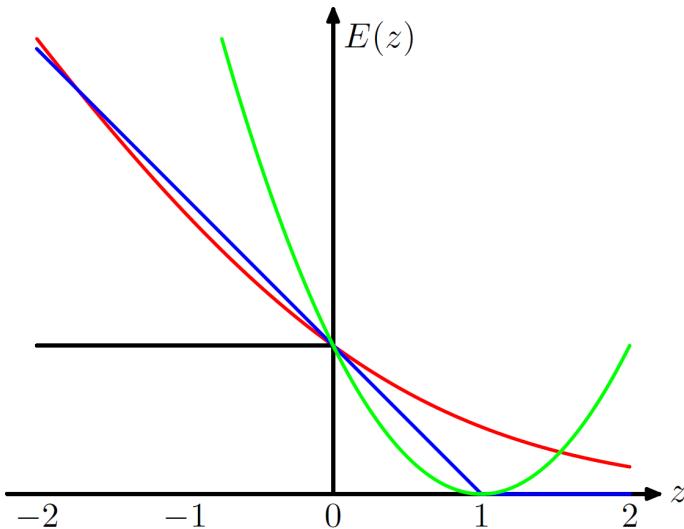
Kako bi se lakše uporedile i kako je množenje konstantom pri minimizaciji nebitno (zato je recimo nebitno i da li se minimizuje prosek ili suma grešaka), podelimo funkciju greške logističke regresije sa $\log 2$, tako da obe prolaze kroz tačku $(0, 1)$. Poređenja radi, dodajmo u poređenje i metod koji bi radio na osnovu kvadratne funkcije greške

$$L_{SE}(u, v) = (u - v)^2$$

pošto ju je zaista moguće primeniti i u slučaju binarne klasifikacije kada su oznake klase numeričke. Uzmimo u obzir i standardnu grešku klasifikacije

$$L_{CE}(u, v) = I(u \neq v)$$

Grafički ovih funkcija prikazani su na slici 4.4. Na horizontalnoj osi je prikazana vrednost poizvoda uv , a na vertikalnoj odgovarajuća vrednost greške. Positivne vrednosti uv odgovaraju tačno klasifikovanim instancama, a negativne



Slika 4.4: Grafici četiri funkcije greške – logističke (crveno), metoda potpornih vektora (plavo), kvadratne (zeleno) i greške klasifikacije (crno).

pogrešno klasifikovanim. Očigledno, logistička funkcija greške kažnjava ne samo pogrešno, već i ispravno klasifikovane instance, čak i ako su daleko od razdvajajuće hiperravnji. To vodi tome da se ne bira hiperravan najšireg pojasa, kao i da sve instance učestvuju u definisanju modela. Upravo to što funkcija greške metoda potpornih vektora dostiže nulu, vodi kako maksimalnom odstojanju, tako i zanemarivanju većine podataka. Još jedna razlika je da logistička funkcija greške raste brže što je \$uv\$ manje, pa je stoga i osetljivija na odudarajuće podatke. Obe ove funkcije predstavljaju nekakvu aproksimaciju (indikatorske) funkcije greške klasifikacije. Kvadratna funkcija greške očito predstavlja vrlo lošu aproksimaciju, pošto jako kažnjava i ispravno klasifikovane tačke, čim je proizvod \$uv\$ veći od 1.

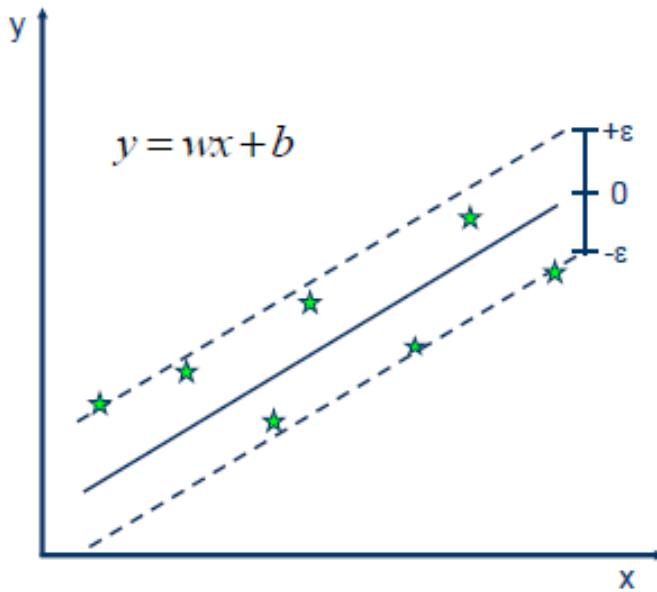
4.2 Metod potpornih vektora za regresiju

Metod potpornih vektora se možda još prirodnije formuliše za regresiju. Funkcija koja se minimizuje je i dalje $\|w\|_2^2$. Uslove tačnog predviđanja bi trebalo prilagoditi kontekstu regresije i model bi mogao da izgleda ovako:

$$\min_w \|w\|_2^2$$

$$|w \cdot x + w_0 - y| = 0$$

Ipak, jedna važna tehnička razlika u odnosu na klasifikaciju je u tome što čak ni u osnovnoj varijanti metoda nema smisla tražiti tačna predviđanja, što je u



Slika 4.5: Osnovni pristup regresiji pomoću metoda potpornih vektora.

linearno razdvojivom slučaju kod klasifikacije bio zahtev. Naime, kod binarne klasifikacije postoje dva moguća ishoda – 1 i –1 i sve vrednosti koje linearni model daje zaokružuju se na njih. Nije potrebno da model da baš vrednost 1 ili –1. U slučaju regresije postoji kontinuum ishoda i zahtev za tačnom jednakostu je prejak. Dodatno, često bi mogao biti i štetan. Naime, podaci retko predstavljaju merenja promenljivih veličina sa savršenom tačnošću. Ako podaci sadrže grešku, nema smisla insistirati da se ta greška nauči. Stoga, uvodi se parametar tolerancije ε koji izražava razliku između predviđanja i stvarne vrednosti koja se smatra potpuno prihvatljivom. Osnovni model izgleda ovako:

$$\min_w \|w\|_2^2$$

$$|w \cdot x_i + w_0 - y_i| \leq \varepsilon \quad i = 1, \dots, N$$

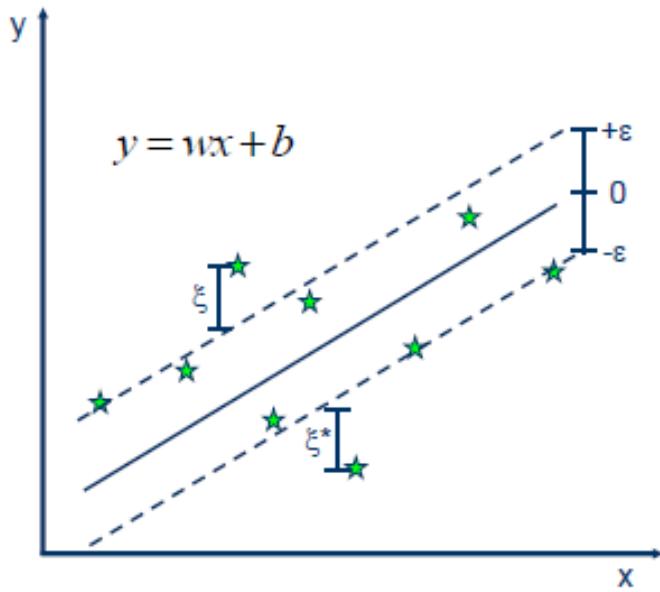
odnosno

$$\min_w \|w\|_2^2$$

$$w \cdot x_i + w_0 - y_i < \varepsilon \quad i = 1, \dots, N$$

$$y_i - w \cdot x_i - w_0 < \varepsilon \quad i = 1, \dots, N$$

Ilustracija je data na slici 4.5.



Slika 4.6: Regresija pomoću metoda potpornih vektora sa mogućnošću grešaka.

Primetimo da se ova formulacija može jednostavno interpretirati. Ograničenja nalažu da predviđanja ne mogu biti daleko od pravih vrednosti, dok minimizacija norme sprečava izbor modela koji brzo menja vrednosti, odnosno umanjuje prilagodljivost.

Ova formulacija, kao i u slučaju klasifikacionog problema, ima nedostatak da ne dozvoljava greške u predviđanjima (osim za fiksiranu vrednost ε). Slično slučaju mekog pojasa, tolerancija na greške se omogućava uvođenjem novih promenljivih:

$$\min_w \|w\|_2^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*)$$

$$w \cdot x_i + w_0 - y_i < \varepsilon + \xi_i \quad i = 1, \dots, N$$

$$y_i - w \cdot x_i - w_0 < \varepsilon + \xi_i^* \quad i = 1, \dots, N$$

$$\xi_i \geq 0 \quad i = 1, \dots, N$$

$$\xi_i^* \geq 0 \quad i = 1, \dots, N$$

Ilustracija je data na slici 4.6.

Rešenje ovog problema ima vrlo sličnu formu rešenju klasifikacionog problema.

Jedno pitanje koje zaslužuje razmatranje je – gde je u slučaju ovog metoda široki pojaz? U ε okolini modela ne bi mogao biti, pošto se u idealnom

slučaju baš tu nalaze svi podaci, a u idealnom slučaju, baš u njemu ne bi trebalo da budu. Razmislimo² kako funkcija greške kažnjava instance u slučaju klasifikacije. Potporni vektori i instance koje su dalje od njih u odgovarajućem poluprostoru ne doprinose grešci. Instance koje zađu pojas dorphismose grešci u skladu proporcionalno udaljenosti od hiperravnog na kojoj leže potporni vektori. U regresionom slučaju, tačke čija se vrednost razlikuje od vrednosti modela za manje od ε , ne doprinose grešci. Čim se razlikuju za više od ε , doprinose proporcionalno toj dodatnoj razlici. To nas navodi na ideju da je u regresionom slučaju široki pojas zapravo prostor tačaka koje se po y osi razlikuju od regresione krive za više od ε .

4.3 Algoritam k najbližih suseda zasnovan na širokom pojasu

Algoritam najbližih suseda verovatno je najjednostavniji algoritam mašinskog učenja. Može služiti za klasifikaciju sa proizvoljnim brojem klasa, kao i za regresiju. Osnovna prepostavka ovog algoritma je postojanje rastojanja nad *prostором атрибута* (eng. *feature space*). Najčešće se prepostavlja vektorska reprezentacija instanci i euklidsko rastojanje, ali moguće su i opštije prepostavke.

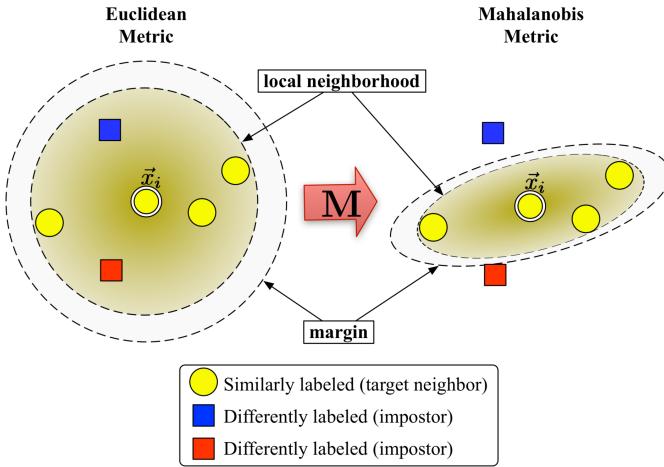
Algoritam k najbližih suseda klasificuje nepoznatu instancu tako što pronalazi k instanci iz skupa za obučavanje koje su joj najbliže u smislu neke izabrane metrike i pridružuje joj klasu koja se najčešće javlja među tih k instanci. U slučaju regresije, za predviđanje se uzima prosečna vrednost k najbližih suseda iz skupa za obučavanje. Ovaj algoritam retko predstavlja najbolji izbor za rešavanje nekog problema, ali neretko daje relativno dobre rezultate, a izuzetno lako se implementira i primenjuje. Više reči o detaljima ovog algoritma biće kasnije. Sada ćemo se fokusirati na jednu njegovu vrlo specifičnu klasifikacionu varijantu.

Jedan od problema vezanih za algoritam k najbližih suseda je činjenica da se funkcija rastojanja bira nezavisno od podataka. U slučaju da domenski ekspert zna nešto više o svojstvima podataka, moguće je napraviti meru rastojanja koja će biti prilagođena datom problemu, ali još bolji način bi bio da se ta mera rastojanja uči. Ukoliko je M pozitivno semidefinitna matrica, funkcija

$$d_M(x, x') = (x - x')^T M (x - x')$$

je funkcija rastojanja. Ovako definisano rastojanje, naziva se *Mahalanobisovim rastojanjem*. Ako važi $M = I$, mesto tačaka jednakog rastojanja od neke fiksirane tačke C je sfera sa centrom u tački C . Ukoliko je matrica M dijagonalna, mesto tačaka jednakog rastojanja od C je elipsoid sa centrom u tački C , čije su ose paralelne koordinatnim osama. U opštem slučaju, radi se o proizvoljnom

²I zahvalimo za ovo razmišljanje kolegi Milošu Jovanoviću.



Slika 4.7: Ilustracija algoritma k najbližih suseda zasnovanog na širokom pojasu.

elipsoidu sa centrom u tački C .³ Razmotrimo još jedan način razumevanja matrice M . Kako je matrica M pozitivno semidefinitna, može se predstaviti kao $M = Q^T Q$. Tada se metrika predstavlja kao

$$d_M(x, x') = (x - x')^T Q^T Q (x - x') = (Qx - Qx')^T (Qx - Qx')$$

Zamenom koordinata $t = Qx$ dobija se

$$d_M(x, x') = (t - t')^T (t - t') = d_I(t, t')$$

Drugim rečima, matrica M transformiše prostor atributa tako da se metrika d_M u novim koordinatama može računati kao standardna euklidska metrika.

Umesto da se matrica M zada unapred, poželjno je učiti je iz podataka. Ipak, postavlja se pitanje kriterijuma u odnosu na koji se uči. Dobra matrica rastojanja bi bila ona za koju su tačke iz iste klase blizu, dok su sve tačke iz različitih klasa međusobno daleko. Ilustracija je data na slici 4.7. Leva slika prikazuje okolinu tačke u odnosu na euklidsku metriku, dok desna prikazuje okolinu u odnosu na Mahalanobisovu metriku naučenu tako da instance iz iste klase budu u toj okolini, a da instance drugih klasa budu van, razdvojene od te okoline širokim pojasom.

Neka je

$$\mathcal{Z} = \{(x, x', x'') | (x, y), (x', y), (x'', y'') \in \mathcal{D} \wedge y \neq y''\}$$

³Strogo gledano, ako je matrica M pozitivno semidefinitna, ali nije pozitivno definitna, pomenuti elipsoid može biti i degenerisan.

skup svih trojki vektora atributa takvih da prva dva pripadaju istoj klasi, kojoj treći ne pripada. Elemente skupa \mathcal{Z} , označavaćemo z_i za $i = 1, \dots, |\mathcal{Z}|$. Jedna formulacija metoda za određivanje matrice M bi mogla biti:

$$\min_M \sum_{(x,y),(x',y) \in \mathcal{D}} d_M(x, x')$$

$$d_M(x, x') < d_M(x, x'') \text{ za sve } (x, x', x'') \in \mathcal{Z}$$

$$M \succeq 0$$

gde poslednji uslov označava pozitivnu semidefinitnost matrice M . Bolji pristup, koji insistira na postojanju širokog pojasa između elemenata klase je sledeći

$$\min_M \sum_{(x,y),(x',y) \in \mathcal{D}} d_M(x, x')$$

$$d_M(x, x') + 1 \leq d_M(x, x'') \text{ za sve } (x, x', x'') \in \mathcal{Z}$$

$$M \succeq 0$$

Konstanta 1 izgleda kao proizvoljan izbor i to i jeste. Kao i u slučaju metoda potpornih vektora za klasifikaciju, to bi mogla biti bilo koja stroga pozitivna vrednost, ali deljenjem svih izraza tom vrednošću dobija se data formulacija.

Kao i u slučaju metoda potpornih vektora, krutost ograničenja se prevazi-lazi mekim pojasom, odnosno uvođenjem novih promenljivih koje čine model tolerantnijim na greške. Finalna formulacija glasi:

$$\min_M \sum_{(x,y),(x',y) \in \mathcal{D}} d_M(x, x') + C \sum_{i=1}^{|\mathcal{Z}|} \xi_i$$

$$d_M(x_i, x'_i) + 1 \leq d_M(x_i, x''_i) + \xi_i \text{ za sve } (x_i, x'_i, x''_i) \in \mathcal{Z}$$

$$\xi_i \geq 0 \quad i = 1, \dots, |\mathcal{Z}|$$

$$M \succeq 0$$

Specifičnost ovog problema je upravo u zahtevu pozitivne semidefinitnosti i iako takav uslov može voditi zahtevnim računskim problemima, u slučaju ovog problema postoje efikasni algoritmi za njegovo rešavanje.

Glava 5

Modeli zasnovani na instancama

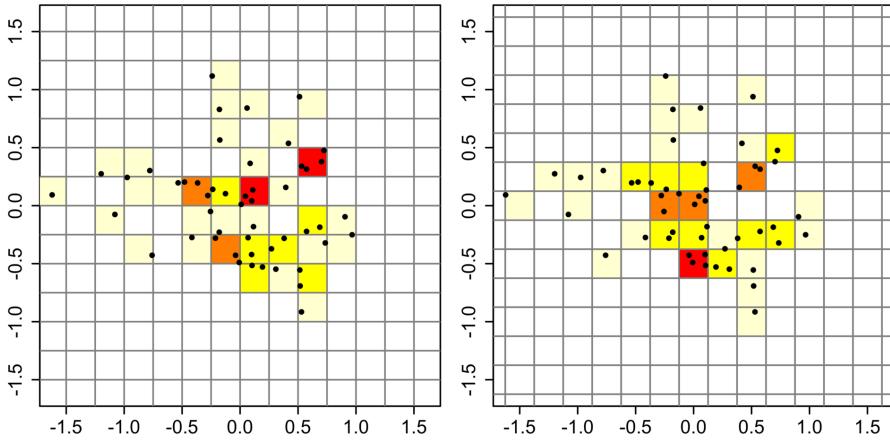
Modeli zasnovani na instancama predstavljaju *neparametarski* pristup mašinskom učenju. Pritom, izraz neparametarski zahteva dodatno objašnjenje. *Parametarski statistički modeli* prepostavljaju postojanje konačnog skupa parametara čije vrednosti definišu model. *Neparametarski modeli* su modeli koji se ne mogu opisati konačnim skupom parametara i čiji broj parametara može zavisiti od veličine skupa za obučavanje i stoga je neograničen. Ovakvi metodi često moraju da čuvaju skup podataka za obučavanje kako bi davali predviđanja na noviminstancama, jer su modeli često i izraženi u terminima tih podataka. Zbog toga i predviđanje na osnovu ovakvih metoda može biti računskih zahtevno. To je očito mana ove vrste metoda, ali njihova prednost je da ne prepostavljaju formu modela tako striktno kao parametarski modeli (npr. prepostavka normalne raspodele), već ta forma može slobodnije da zavisi od podataka.

5.1 Osnove neparametarske ocene gustine raspodele

Ocena gusitne raspodele predstavlja najteži problem mašinskog učenja. Osnovna intuicija iza metoda ocene gustine raspodele je da regioni prostora atributa u kojima se nalazi više tačaka podataka imaju više vrednosti gustine raspodele, dok oni u kojima se nalazi manji broj tačaka imaju manju vrednost gustine. Ipak, svaka tačka u skupu podataka svedoči samo o svom pojavljanju i bilo bi moguće da se samo tim tačkama pridruži nenula vrednost gustine raspodele, a da sve ostale tačke dobiju vrednost nula. Ovakva ocena predstavlja ekstreman slučaj preprilagođavanja. Umesto toga, moguće je smatrati da svaka tačka podataka povećava ne samo vrednost gustine raspodele u toj tački, već i tačaka u nekoj njenoj okolini. Naravno, postavlja se pitanje definisanja takve okoline, a i definisanja načina na koji se takav model konstruiše.

Jedan pristup oceni gustine raspodele koji odgovara prethodnom opisu je histogram. Ipak, taj pristup nije dovoljno dobar iz više razloga:

1. tako ocenjena gustina raspodele zavisi od pozicija korpica, koje bi se pri konstrukciji histograma, mogle proizvoljno translirati (ovo je ilustrovano



Slika 5.1: Dva histograma čije su korpice translirane. Boje izražavaju vrednost funkcije.

slikom 5.1),

2. tačke prekida histograma nisu posledica gustine podataka, već izbora lokacija korpica,
3. oblik histograma drastično zavisi od širine, odnosno broja korpica (ovo je ilustrovano slikom 5.2),
4. zbog prokletstva dimenzionalnosti, većina korpica će biti prazna u slučaju prostora veće dimenzionalnosti.

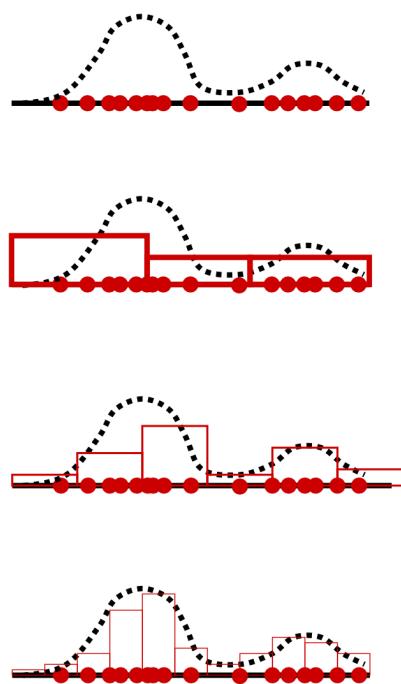
Ovi razlozi čine upotrebu histograma vrlo pipavom i u svrhe preliminarne analize podataka, a za ocenu gustine raspodele ga ne treba koristiti. Ipak, ne znači ni da bolje alternative mogu dati odgovor na baš sve ove probleme.

Držeći se i dalje principa iz prvog paragrafa, razmislimo o alternativama. Jedan razlog za pomenute mane histograma je što se pri njegovoј konstrukciji polazi od particonisanja prostora koje se nakon toga smatra fiksiranim. Alternativa bi bila krenuti od pozicija tačaka na osnovu kojih se gustina određuje. Ilustracija jednog (ali ne jedinog) načina na koji je to moguće uraditi je prikazana na slici 5.3.

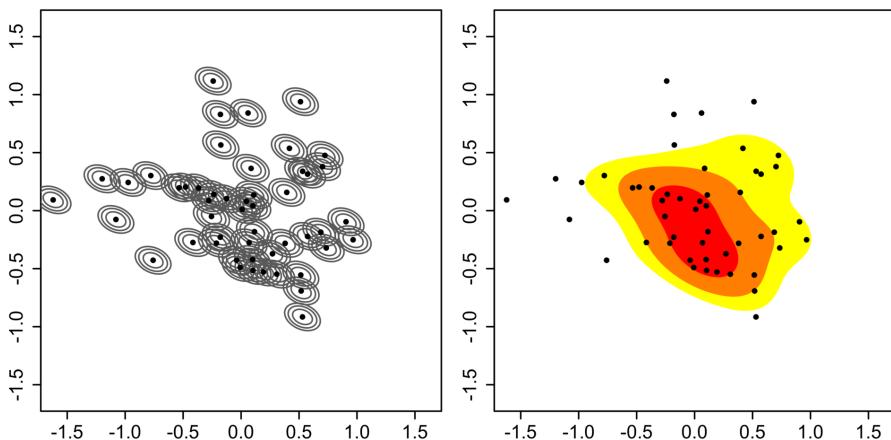
Razmotrimo oblast \mathcal{R} koja sadži tačku x u čijoј okolini treba oceniti gustinu raspodele $p(x)$. Verovatnoća pridružena ovoј oblasti je

$$P = \int_{\mathcal{R}} p(x)dx$$

Ako N opažanja dolazi iz raspodele $p(x)$, svako ima verovatnoću P pripadanja oblasti \mathcal{R} . Broj tačaka k koje pripadaju ovoј oblasti je raspodeljen u skladu sa



Slika 5.2: Histogrami sa različitim brojem korpica konstruisani nad istim podacima.



Slika 5.3: Alternativa histogramu – ocena gustine zasnovana na doprinosima tačaka svojim okolinama. Doprinos tačke opada sa udaljavanjem od nje.

binomnom raspodelom:

$$p(k) = \binom{N}{k} P^k (1-P)^{N-k}$$

Važi $\mathbb{E}(k) = NP$ i $\text{var}[k] = NP(1-P)$. Na osnovu toga, očekivanje udela tačaka koje upadaju u region \mathcal{R} je $\mathbb{E}[k/N] = P$, a varijansa je $\text{var}[k/N] = P(1-P)/N$. Sa povećanjem broja N , varijansa veličine k/N se smanjuje, pa je sve bliža proseku, odnosno važi

$$k \approx NP$$

Slično, ukoliko je zapremina oblasti \mathcal{R} dovoljno mala, tako da se funkcija $p(x)$ ne menja mnogo u njoj, važi

$$P \approx p(x)V$$

gde je V zapremina oblasti \mathcal{R} . Prethodne relacije daju ocenu ¹

$$p(x) \approx \frac{k}{NV}$$

Među veličinama k i V postoji očigledna zavisnost. U većoj zapremini V , očekuje se više tačaka k . Slično, kako bi bio dosegnut veliki broj tačaka k , potrebna je velika zapremina V . Otud, pristupi oceni gustine raspodele mogu biti formulisani oslanjajući se na neku od ove dve veličine, prema čemu razlikujemo *pristupe zasnovane na kernelima* i *pristupe zasnovane na najbližim susedima*. Metodi zasnovani na kernelima formulišu se u odnosu na zapreminu V , a metodi zasnovani na najbližim susedima u terminima broja k . Preciznije, u slučaju metoda zasnovanim na kernelima, broj tačaka na osnovu kojih se vrši ocena gustine raspodele u tački x zavisi od toga koliko ih ima u okolini tačke x unapred izabrane zapremine V . U slučaju metoda zasnovanim na najbližim susedima, zapremina okoline tačke x u kojoj se nalaze tačke na osnovu kojih se vrši ocena gustine u tački x zavisi od unapred izabranog broja tačaka k .

¹Osvrnamo se na uslove pod kojima ova ocena konvergira stvarnoj gustini raspodele $p(x)$. Naime, ako se fokusiramo na jednu tačku uzorka, njena dovoljno mala okolina će sadržati samo tu tačku, a okolina se može proizvoljno smanjivati. Kako njena zapremina teži nuli, naša ocena teži beskonačnosti. Stoga, u slučaju konačnog uzorka okolina očito ne sme biti proizvoljno mala. Razmotrimo situaciju u kojoj broj tačaka N teži beskonačnosti. Neka je gustina $p(x)$ neprekidna u tački x i, jednostavnosti radi, neka je $p(x) \neq 0$. Neka je \mathcal{R}_N lopta sa centrom u tački x , zapremine V_N , koja sadrži k_N tačaka i koja se koristi za ocenu gustine pri uzorku veličine N . Tada ocena $p_N(x) = \frac{k_N}{NV_N}$ teži vrednosti $p(x)$ ako i samo ako važe uslovi

- $\lim_{N \rightarrow \infty} k_N = \infty$
- $\lim_{N \rightarrow \infty} V_N = 0$
- $\lim_{N \rightarrow \infty} k_N/V_N = 0$

Prvi uslov omogućava da k_N/N teži P_N (verovatnoća regiona \mathcal{R}_N). Drugi uslov omogućava da se okolina na osnovu koje se ocenjuje $p(x)$ smanjuje. U suprotnom bi ovakva ocena gustine uprosečavala vrednost gustine u okolini tačke x . I konačno, ako V_n teži nuli, onda i P_N/V_N mora težiti nuli, inače bi ocena divergirala.

5.2 Metodi zasnovani na kernelima

Veliki broj metoda zasnovanih na instancama počiva na upotrebi *kernela*. Neformalno *kernel* predstavlja funkciju sličnosti. Što je vrednost kernela za neke dve instance veća, to se one mogu smatrati sličnijim. Što je manja, to se te dve instance mogu smatrati različitim. U različitim kontekstima, prave se različite pretpostavke vezane za kernele, pa ovde neće biti formalna definicija, već ce pretpostavke koje se prave biti navedene u odgovarajućim odeljcima.

Kerneli najčešće imaju određene parametre, kojima se finije podešava njihovo ponašanje. Ovi parametri su u vezi sa zapreminom okoline neke tačke u čijoj se okolini ocenjuje gustina raspodele. Iako ih u kontekstu kernela називамо параметрима, u kontekstu algoritama učenja oni zapravo predstavljaju metaparametre!

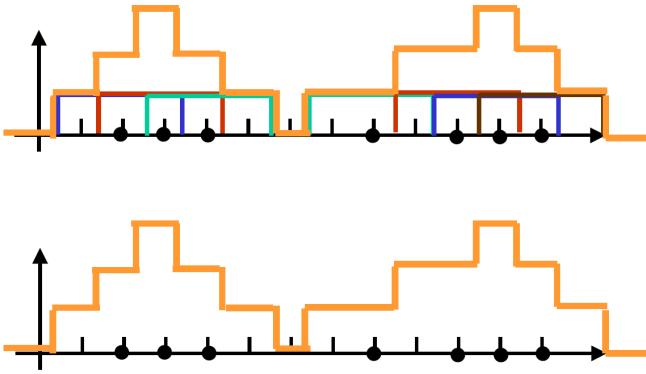
Najavimo jedno svojstvo metoda zanovanih na kernelima koje ih razlikuju od ranije viđenih metoda. Ranije izlagani imetodi izražavaju zavisnost ciljne promenljive od atributa pomoću koeficijenata koji kontrolisu dejstvo promene tog atributa na promenu ciljne promenljive. Tipično, veće vrednosti atributa vode većim vrednostima ciljne promenljive, a male manjim, ili suprotno. Nasuprot tome, pristup zasnovan na kernelima, kao merama sličnosti, se zasniva na bliskosti vrednosti atributa. Ukoliko su vrednosti atributa nove instance bliske vrednostima atributa neke instance iz skupa za obučavanje, i vrednost ciljne promenljive nove instance treba da bude bliska ciljnoj vrednosti te instance, bez obzira da li su vrednosti atributa nove instance same za sebe visoke ili niske. Ovaj pristup vodi *lokalnosti* modela. Ranije diskutovani modeli pokušavaju da okarakterišu globalne zavisnosti, dok u slučaju modela zasnovanog na kernelima, u različitim delovima prostora atributa, te zavisnosti mogu biti vrlo različite, a vrednost ciljne promenljive nove instance će biti određena na osnovu zakonitosti koja važi u njenoj okolini.

5.2.1 Ocena gustine raspodele zasnovana na kernelima

U kontekstu ocene gustine raspodele, obično se polazi od pojma *glačajućeg kernela* (eng. *smoothing kernel*), funkcije jedne promenljive za koju važi:

- $K(x) \geq 0$
- $\int K(x)dx = 1$
- $K(-x) = K(x)$
- K ne raste sa udaljavanjem od nule

Skalirani kernel se definiše kao $K_\sigma = \frac{1}{\sigma^n} K\left(\frac{x}{\sigma}\right)$ za $\sigma > 0$, gde je n dimenzija vektora x . Skalirani kernel zadovoljava sve uslove koje zadovoljava i polazni kernel. Metaparametar σ se naziva *širinom* (eng. bandwidth) kernela.



Slika 5.4: Ocena gustine jednodimenzionalne raspodele pomoću Parzenovih prozora širine 3.

Neka je oblast \mathcal{R} hiperkocka sa centrom u tački x . Definišimo karakterističnu funkciju jedinične hiperkocke sa centrom u koordinatnom početku

$$K(x) = \begin{cases} 1, & |x_i| \leq 1/2, \\ 0, & \text{inače} \end{cases} \quad i = 1, \dots, n$$

Funkcija K je primer kernela i naziva se *Parzenovim prozorom*. Broj tačaka k koji se nalazi u hiperkocki strane σ sa centrom u tački x može se izraziti pomoću odgovarajućeg skaliranog kernela kao

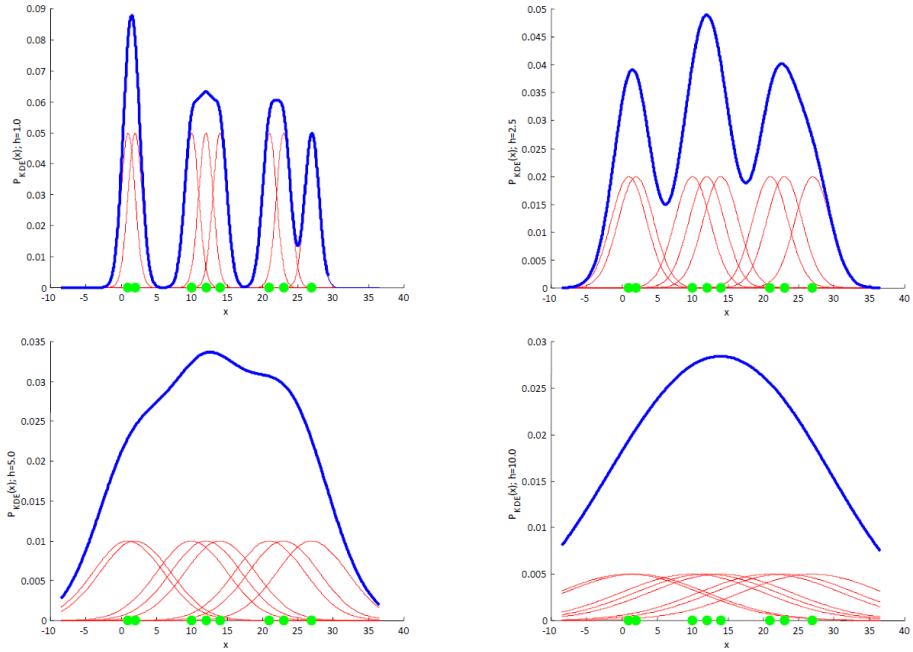
$$k = \sum_{i=1}^N K\left(\frac{x - x_i}{\sigma}\right)$$

Na osnovu ovog zapažanja i prethodno izvedene aproksimacije $p(x) \approx \frac{k}{NV}$ važi

$$p(x) = \frac{1}{N} \frac{1}{\sigma^n} \sum_{i=1}^N K\left(\frac{x - x_i}{\sigma}\right) = \frac{1}{N} \sum_{i=1}^N \frac{1}{\sigma^n} K\left(\frac{x - x_i}{\sigma}\right) = \frac{1}{N} \sum_{i=1}^N K_\sigma(x - x_i)$$

Prikaz ocene jednodimenzionalne gustine raspodele pomoću Parzenovih prozora dat je na slici 5.4

Ovim se dobija prekidna ocena gustine vrlo slična histogramu. Ipak, postoji razlika. Razmotrimo je u jednodimenzionalnom slučaju. Kod histograma su podeoci na x osi, odnosno tačke promene vrednosti histograma, fiksirani, dok u ovom slučaju zavise od podataka, jer se mogu desiti samo u tačkama koje su na rastojanju $\sigma/2$ od neke tačke podataka. Ovaj pristup ima nešto poželjnija svojstva od histograma. Naime, za fiksirane podeoke, odnosno korpice, tačka x koja pripada jednoj korpici, može biti bliže većini tačaka druge korpice, nego



Slika 5.5: Ocena gustine jednodimenzionalne raspodele pomoću Gausovih kerneala različitih širina.

većini tačaka iz svoje. Stoga, tačke iz njene korpice, od kojih je daleko određuju vrednost gustine raspodele u tački x , a tačke iz druge korpice, kojima je bliža, nemaju nikakav uticaj. U slučaju date ocene pomoću kernela, ovaj problem ne postoji. Bliže tačke uvek daju više doprinosova od udaljenijih. Moguće je prevazići i problem prekidnosti korишćenjem neprekidnih kernela, poput Gausovog:

$$K_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$$

Na slici 5.5, prikazana je ocena gustine pomoću Gausovog kernela za različite širine.

Kao i u slučaju regularizacije, izbor metaparametara σ ćemo diskutovati kasnije, ali generalno, mala širina kernela tipično vodi preprilagođavanju, dok velika vodi potprilagođavanju i uprosećavanju informacije. To se vidi i na slici 5.5.

Mana ocene gustine raspodele zasnovane na kernelima je ta što je širina kernela nezavisna od tačke x , pa jedan skup parametara u oblastima visoke gustine može rezultovati preširokim kernelom, čijim se dejstvom usrednjjava struktura u podacima, dok bi u oblastima u kojima je gustina značajno niža taj isti krenel mogao biti premale širine i voditi preprilagođavanju.

5.2.2 Nadaraja-Votson metod za regresiju zasnovanu na kernelima

Kao što je ranije rečeno, regresiona funkcija, čija aproksimacija se traži u problemu regresije, definisana je kao uslovno očekivanje ciljne promenljive pri datim vrednostima atributa:

$$r(x) = \mathbb{E}[y|x] = \int yp(y|x)dy = \int y \frac{p(x,y)}{p(x)} dy$$

Jedan način da se izvede ocena ove funkcije je da se u datom integralu upotrebe ocene gustina raspodela koje figurišu u njemu. Ukoliko su definisani neki glaćajući kerneli na prostoru atributa \mathcal{X} i na skupu vrednosti ciljne promenljive \mathcal{Y} , njihov proizvod predstavlja kernel na prostoru $\mathcal{X} \times \mathcal{Y}$, pa je poslednji integral moguće aproksimirati sledećim modelom:

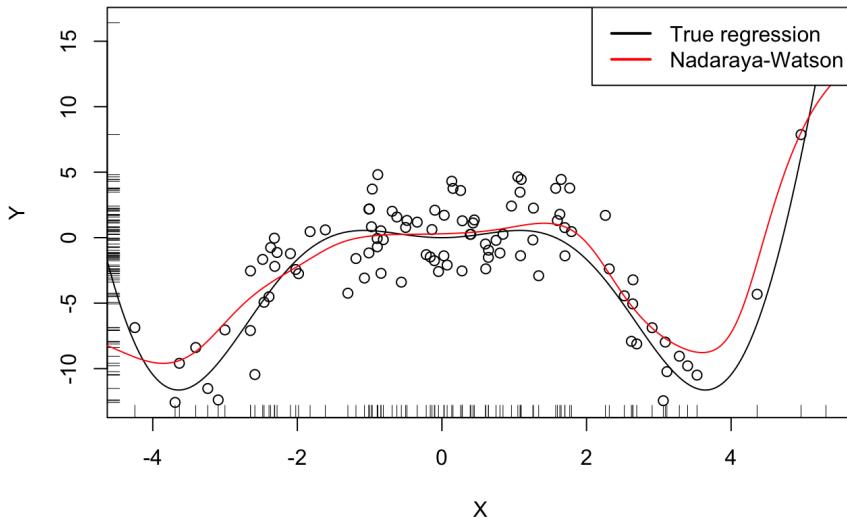
$$\begin{aligned} f_\sigma(x) &= \int y \frac{\sum_{i=1}^N K_\sigma(x - x_i) K_\sigma(y - y_i)}{\sum_{j=1}^N K_\sigma(x - x_j)} dy = \\ &\frac{\sum_{i=1}^N K_\sigma(x - x_i) \int y K_\sigma(y - y_i) dy}{\sum_{j=1}^N K_\sigma(x - x_j)} = \\ &\frac{\sum_{i=1}^N K_\sigma(x - x_i) y_i}{\sum_{j=1}^N K_\sigma(x - x_j)} \end{aligned}$$

gde poslednja jednakost važi na osnovu simetričnosti kernela. Ovo je model regresije Nadaraja-Votson. Razmotrimo, njegov smisao. Težina

$$\frac{K_\sigma(x - x_i)}{\sum_{j=1}^N K_\sigma(x - x_j)}$$

koja se pridružuje vrednosti ciljne promenljive y_i je proporcionalna sličnosti tačke x sa tačkom x_i . Kako se ove težine sabiraju na 1, zaključujemo da je dobijeno rešenje težinski prosek vrednosti y_i , pri čemu se pridaje veća težina sličnijim instancama.

Na slici 5.6, prikazan je primer regresije pomoću ovog metoda. Povećavanje širine kernela vodilo bi manjem, a smanjivanje većem prilagođavanju podacima. Aproksimacije koje smo prikazali u slučaju linearne regresije polinomom izgledale su slično. Ipak, linearni model pretpostavlja formu modela, a korišćenje polinoma, formu baznih funkcija. Pritom, to je samo jedan mogući izbor i možda bi drugačiji izbori vodili boljim rezultatima. Odnosno, parametarski pristup očekuje da smo u stanju da donešemo dobru pretpostavku vezanu za formu modela. Neparametarski pristup to ne zahteva. Ali, s druge strane, izražava rešenje u terminima celog skupa za obučavanje koji je potrebno čuvati i koji se ceo koristi prilikom predviđanja.



Slika 5.6: Primer regresije pomoću metoda Nadaraja-Votson. Tačke su generisane na osnovu crne krive dodavanjem slučajnog šuma. Crvena kriva predstavlja ocenu regresije.

5.2.3 Kernelizovani metod potpornih vektora

U kontekstu metoda potpornih vektora, bitan je pojam pozitivno semidefinitnog kernela (a ne glaćajućeg kernela). Neka je \mathcal{X} neprazan skup i neka je data simetrična funkcija $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. Ako je za svako $n \in \mathbb{N}$ i svako $x_1, \dots, x_n \in \mathcal{X}$ matrica dimenzija $n \times n$ sa elementima $k(x_i, x_j)$ pozitivno semidefinitna, funkcija k je *pozitivno semidefinitan kernel* ili *Mercerov kernel*. U nastavku se pod izrazom kernel podrazumeva Mercerov kernel. Važan podatak je da za svaki kernel k postoji preslikavanje Ψ_k iz \mathcal{X} u neki vektorski prostor \mathcal{H}_k sa skalarnim proizvodom, tako da važi $k(x, y) = \Psi_k(x) \cdot \Psi_k(y)$. Drugim rečima, svaki kernel se može posmatrati kao skalarni proizvod u nekom vektorskem prostoru. Otud se kernel može smatrati merom sličnosti nad elementima skupa \mathcal{X} . Pritom, za taj skup nije prepostavljena nikakva struktura. Odnosno, za prozivoljne objekte nad kojima možemo definisati kernele, možemo imati značajan deo tehničkih pogodnosti koje pruža skalarni proizvod. Kernele imaju određena poželjna svojstva, kao da je linearna kombinacija kernela takođe kernel, da je proizvod kernela takođe kernel i slično. Ova svojstva se mogu upotrebiti za konstrukciju novih kernela od već poznatih. U nastavku se pod pojmom kernela podrazumeva kernel.

U ovom odeljku se forkusiramo na metod potpornih vektora za klasifikaciju. Njegov ključni problem je činjenica da hiperravan ne mora predstavljati adekvatnu granicu među klasama. Oblici granica u praksi mogu biti proizvoljni. Pritom, ovaj problem nije rešen formulacijom sa mekim pojasom, pošto meki pojas pretpostavlja da je hiperravan ugrubo dobar oblik granice, samo što podaci nekada završe sa pogrešne strane. Moglo bi biti potrebno da granica bude kružna, a i tad se može očekivati da podaci nekada završe sa pogrešne strane kružnice. Stoga su ta dva pitanja nezavisna.

Metod potpornih vektora ima sledeću formu modela:

$$f_{w,w_0}(x) = \sum_{i=1}^N \alpha_i y_i (x_i \cdot x) + w_0$$

pri čemu važi

$$w_0 = -\frac{\min_{(x,1) \in \mathcal{D}} w \cdot x + \max_{(x,-1) \in \mathcal{D}} w \cdot x}{2}$$

Uočavamo da se rešenje oslanja na određene instance iz skupa za obučavanje tako što računa skalarni proizvod instance koju treba klasifikovati sa njima. Skalarni prozivod je jedna vrsta kernela i kerneli se mogu posmatrati kao nje-gova uopštenja. Razmotrimo relevantnu definiciju kernela.

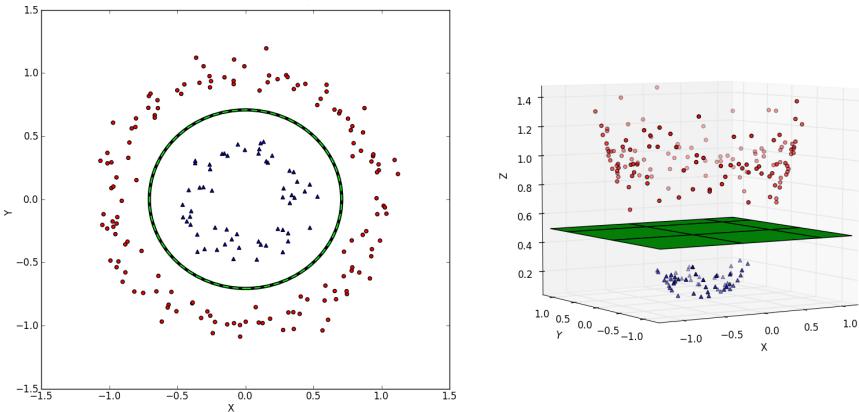
Zahvaljujući činjenici da za svaki kernel postoji preslikavanje Φ u neki vektorski prostor u kojem važi

$$k(x, x') = \Phi(x) \cdot \Phi(x')$$

Šta ako bi u tom prostoru podaci za obučavanje mogli biti razdvojeni pomoću hiperravnih, iako u polaznom prostoru taj oblik nije odgovarajući? Kako je na podacima moguće vršiti različite vrste preprocesiranja, ukoliko bismo znali preslikavanje Φ mogli bismo ga primeniti na date podatke i dobiti novi skup podataka nad kojim bismo mogli primeniti standardni metod potpornih vektora. Pored nepoznavanja tog preslikavanja, ozbiljan problem bi mogao biti i taj što to preslikavanje može slikati podatke u neki beskonačno dimenzionalni prostor, pa podatke ne bismo mogli zapisati. Umesto toga, bolja strategija je zamena skalarnog proizvoda kernelom u svim formulama, što je sa matematičke tačke gledišta ista stvar. U tom slučaju model postaje:

$$f_{w,w_0}(x) = \sum_{i=1}^N \alpha_i y_i K(x_i, x) + w_0$$

I dalje nije jasno da smo ovim na dobitku. Naime, kako ne znamo kako izgledaju podaci u novom prostoru, ne znamo ni da li će metod u njemu raditi išta bolje nego u polaznom prostoru. Razmotrimo jedan konkretan problem i konstruišimo adekvatan kernel za taj problem.



Slika 5.7: Preslikavanje kojim se problem koji je linearno nerazdvojiv u dve dimenzije preslikava u problem koji je linearno razdvojiv u tri dimenzije.

Primer 3 Neka su podaci raspoređeni tako da elementi jedne klase unutar lopte određenog poluprečnika u prostoru atributa, dok su elementi druge klase van te lope, što je u dvodimenzionalnom slučaju prikazano na slici 5.7. Definišimo kernel zasnovan na preslikavanju $\Phi(x) = (x_1, x_2, x_1^2 + x_2^2)$:

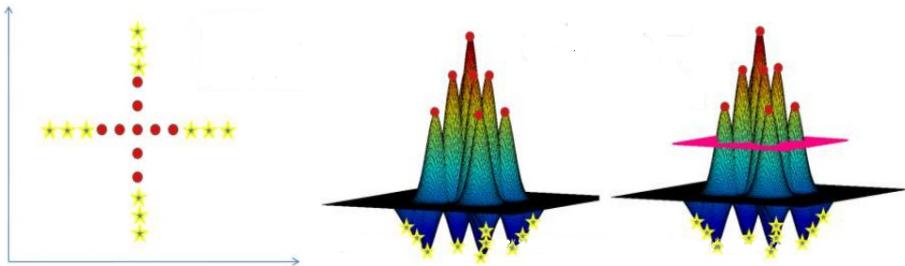
$$k(x, x') = (x_1, x_2, x_1^2 + x_2^2) \cdot (x'_1, x'_2, x'^2_1 + x'^2_2)$$

Ovo preslikavanje očigledno slika podatke u prostor veće dimenzije u kojem se mogu razdvojiti pomoću hiperravnih, kao što je prikazano na slici. U polaznom prostoru, inverzna slika razdvajajuće hiperravni izgleda kao kružnica.

Iako je u prethodnom primeru upotreba kernela bila očigledno korisna, primer je polazio od pretpostavke da nam je rasporedi instanci u prostoru poznat, što je prejaka pretpostavka. Zato je potreban opštiji pristup. Ispostavlja se da je jedna vrsta kernela dovoljna (iako ne nužno uvek najpogodnija). To je takozvani Gausov kernel:

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{\gamma}\right)$$

Veće vrednosti parametra γ vode širim Gausovim zvonima, a manje užim. Objasnimo univerzalnu primenljivost ovog kernela primerom. Na slici 5.8, prikazan je skup tačaka i odgovarajući model koji ih tačno klasificiše. Ključni uvid je da za dovoljno malo γ svaka tačka može biti potporni vektor sa pridruženom okolinom u kojoj nema drugih tačaka i da se množenjem Gausovog zvona znakom klase može postići njena tačna klasifikacija. Ukoliko se dozvole kerneli sa dovoljno malom širinom, ovo je moguće postići na svakom neprotivrečnom skupu podataka, što znači da skup svih modela zasnovanih na Gausovom kernelu



Slika 5.8: Podaci za obučavanje (levo), model u slučaju Gausovog kernela kada je svaka instanca potporni vektor (sredina) i zajedničko dejstvo funkcije sgn na taj model pod pretpostavkom da je slobodni član negativan, usled čega je ljubičasta površ izdignuta (desno).

proizvoljne širine ima beskonačnu Vapnik-Červonenkisovu dimenziju. Ipak, u praksi se nikad ne omogućava korišćenje proizvoljnih vrednosti parametra γ , već se γ koristi kao metaparametar, nalik regularizacionom metaparametru.

Vratimo se na formu modela metoda potpornih vektora:

$$f_{w,w_0}(x) = \sum_{i=1}^N \alpha_i y_i K(x_i, x) + w_0$$

U slučaju linearног kernela, odnosno običnog skalarnog proizvoda, koeficijenti liennarnog modela su se mogli eksplicitno izračunati zahvaljujući distributivnosti sabiranja u odnosu na skalarni prozvod:

$$w = \sum_{i=1}^N \alpha_i y_i x_i$$

U ovom, opštijem slučaju to nije moguće, što znači da je čuvanje instanci neophodno, kako bi se izračunale vrednosti kernela. Ovo je uobičajeno za metode zasnovane na kernelima. Međutim, u ovom kontekstu već pomenuto svojstvo metoda potpornih vektora da model zavisi samo od nekih instanci, dobija na važnosti. Za razliku od drugih metoda zasnovanih na kernelima, u slučaju ovog metoda dovoljno je čuvati samo neke instance, a prilikom predviđanja, zahvaljujući manjem broju instanci za koje se računaju vrednosti kernela, izračunavanje je brže.

Na isti način, odnosno zamenom skalarnog proizvoda kernelom, može se kernelizovati i metod potpornih vektora za regresiju. I šire, mnogi metodi mašinskog učenja kod kojih se model može izraziti u terminima skalarnih proizvoda sainstancama iz skupa za obučavanje, mogu se kernelizovati, što često vodi boljim prediktivnim performansama.

5.3 Metodi zasnovani na najbližim susedima

Kao što metodi zasnovani na kernelima počivaju na upotrebi funkcija sličnosti, tako metodi zasnovani na najbližim susedima počivaju na upotrebi rastojanja. Pritom, veza između rastojanja i sličnosti je tesna, tako da ovo ne predstavlja razliku između ove dve grupe metoda, već sličnost. Tipičan metaparametar ovakvih metoda je broj suseda koji se razmatra. Ipak, u slučaju korišćenja Mahalanobisovog rastojanja, elemente matrice rastojanja takođe predstavljaju metaparametre metoda.²

5.3.1 Ocena gustine raspodele zasnovana na najbližim susedima

Do metoda k najbližih suseda za ocenu gustine raspodele dolazi se kada se umesto fiksiranja zapremine, kao u slučaju Parzenovog prozora, kao princip konstruisanja ocene gustine raspodele uzme fiksiranje broja tačaka k koji treba da budu obuhvaćeni okolinom tačke u kojoj se ocenjuje gustina raspodele. Tada se računa sfera najmanje zapremine V_k koja sadrži k tačaka i dolazi se do ocene

$$p(x) = \frac{k}{NV_k}$$

Primetimo da oblik sfere zavisi od izabrane metrike.

Problem sa datom ocenom gustine je taj što za konačne uzorke sama ne predstavlja validnu gustinu raspodele, zato što integral po x divergira. Recimo, u slučaju $k = 1$ će u tačkama uzorka zapremina najmanje svfere biti nula, a ocena gustine beskonačna. Ipak, ova ocena može biti korisna, makar za izvođenje drugih metoda, kao što će biti urađeno u narednom odeljku.

Male vrednosti broja k vode preprilagođavanju tako što se masa verovatnoće raspoređuje sve više na tačke iz skupa za obučavanje, a sve manje na druge tačke, dok velike vrednosti broja k vode usrednjavanju i gubljenju informacije, tako što se masa verovatnoće raspoređuje ravnomernije nego što bi trebalo, čime se gubi struktura raspodele podataka koja u konkretnom slučaju možda i nije ravnomerna.

5.4 Algoritam k najbližih suseda

Algoritam k najbližih suseda verovatno je najjednostavniji algoritam mašinskog učenja. Može služiti za klasifikaciju sa proizvoljnim brojem klasa, kao i za regresiju. Osnovna prepostavka ovog algoritma je postojanje rastojanja nad prostorom atributa. Najčešće se prepostavlja vektorska reprezentacija instanci i euklidsko rastojanje, ali takve prepostavke nisu neophodne.

²Prisetimo se da smo u slučaju algoritma k najbližih suseda zasnovanom na širokom pojasu ove elemente smatrati parametrima koje smo učili. To nije karakteristično za metaparametre, ali je moguće.

Hladnoća	Curenje iz nosa	Glavobolja	Groznica	Grip
Da	Ne	Blaga	Da	Ne
Da	Da	Ne	Ne	Da
Da	Ne	Jaka	Da	Da
Ne	Da	Blaga	Da	Da
Ne	Ne	Ne	Ne	Ne
Ne	Da	Jaka	Da	Da
Ne	Da	Jaka	Ne	Ne
Da	Da	Blaga	Da	Da

Tabela 5.1: Tabela podataka za problem dijagnostifikovanja gripa.

Procena verovatnoće klase na osnovu vrednosti atributa može se uraditi korišćenjem Bajesove formule i ocene gustine raspodele pomoću k najbližih suseda:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{\frac{k_y}{N_y} \frac{N_y}{V_k}}{\frac{k}{NV_k}} = \frac{k_y}{k}$$

gde je N_y broj podataka koji pripada klasi y , a k_y broj tačaka iz k najbližih suseda koji pripadaju klasi y . Na osnovu izvedene uslovne raspodele, predviđanje se vrši na uobičajen način:

$$f(x) = \operatorname{argmax}_y p(y|x)$$

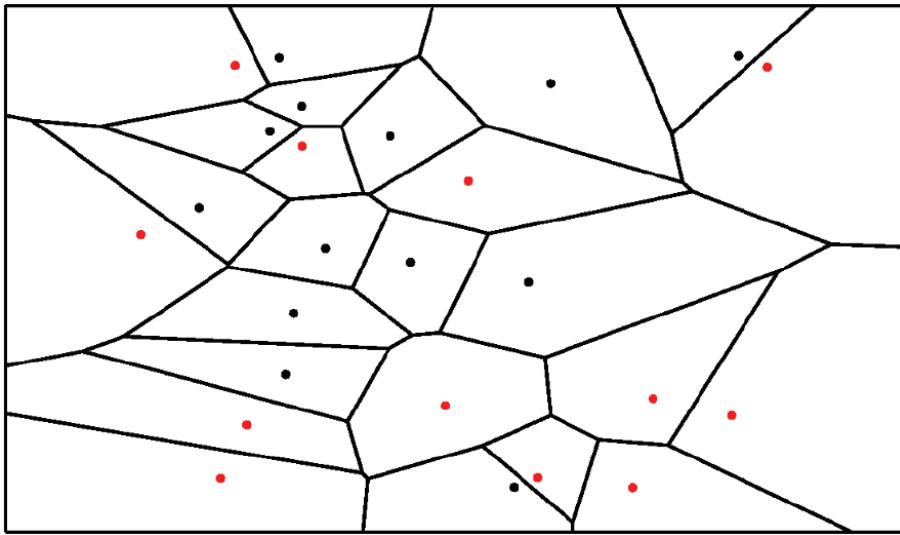
Drugim rečima, algoritam k najbližih suseda klasificuje nepoznatu instancu tako što pronalazi k instanci iz skupa za obučavanje koje su joj najbliže u smislu neke izabrane metrike i pridružuje joj klasu koja se najčešće javlja među tih k instanci. Ovaj algoritam se očito može posmatrati kao diskriminativni probabilistički. Retko predstavlja najbolji izbor za rešavanje nekog problema, ali neretko daje relativno dobre rezultate, a izuzetno lako se implementira i primenjuje.

Primer 4 U tabeli 5.1 prikazan je primer primene ovog algoritma. U odnosu na date podatke, potrebno je klasifikovati instancu (Da, Ne, Blaga, Ne). Kako su vrednosti atributa kategoričke, potrebno je definisati specifičnu funkciju rastojanja. Neka je izabrana funkcija:

$$d(x, x') = \sum_{i=1}^n I(x_i \neq x'_i)$$

Ako se u obzir uzima 1 najbliži sused, najbliži je prvi primer iz tabele, pa je predviđena klasa Ne.

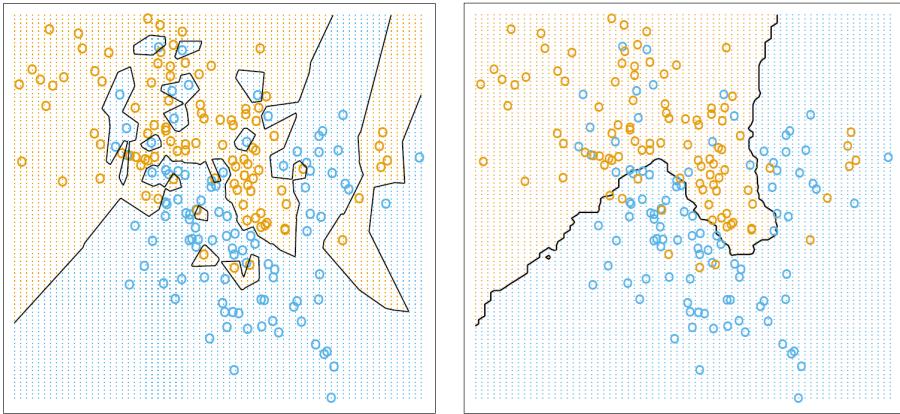
Primetimo da ovaj algoritam nema eksplisitnu formu modela, funkciju greške u odnosu na koju bi se model obučavao, pa ni fazu obučavanja uopšte, osim



Slika 5.9: Način na koji algoritam k najbližih suseda deli prostor atributa na podoblasti različitih klasa.

izbora vrednosti metaparametra k . Zapravo, slično važi i za metod Nadaraja-Votson, pa i za druge (ali ne sve) metode zasnovane na instancama. U slučaju metoda k najbližih suseda, sav rad se obavlja u fazi predviđanja. Kao i u slučaju mnogih metoda zasnovanih na kernelima, kako ne proizvodi model u nekoj funkcionalnoj parametrizovanoj formi, koji se može čuvati radi kasnije primene, potrebno je čuvati sve instance ili, eventualno, neki podskup predstavnika ukoliko je broj instanci preveliki. Iako algoritam ne daje eksplicitan model, on je implicitno definisan skupom instanci i može se vizualizovati. Na slici 5.9 prikazan je način na koji algoritam k najbližih suseda deli prostor atributa na podoblasti koje pripadaju različitim klasama za $k = 1$. U tom slučaju, svakoj instanci je pridružen skup tačaka koje su bliže toj instanci nego drugiminstancama iz skupa za obučavanje. Za razliku od prethodnih algoritama klasifikacije koji su pretpostavljali da je razdvajajuća granica među klasama hiperravan, ovaj algoritam dozvoljava značajno komplikovaniju razdvajajuću granicu među klasama. Štaviše regioni koji pripadaju istoj klasi mogu biti i nepovezani.

Na slici 5.10, dato je poređenje razdvajajućih granica algoritma k najbližih suseda u slučaju $k = 1$ i $k = 15$. Primećuje se da je ona u prvom slučaju komplikovanija nego u drugom. U ekstremnom slučaju, kada je k jednako veličini skupa za obučavanje, ceo prostor se klasificiše u istu – većinsku klasu. Ovo je slično kako ponašanju metoda zasnovanih na kernelima u odnosu na izbor metaparametra σ , tako i ponašanju regularizacije kod parametarskih metoda. To nije slučajno. Svi ti metaparametri igraju vrlo slične uloge. Što je vrednost



Slika 5.10: Podela prostora atributa algoritmom k najbližih suseda, za $k = 1$ i $k = 15$.

k manja, ovaj algoritam je u većoj opasnosti od preprilagođavanja podacima za obučavanje. Što je vrednost k veća, ta opasnost je manja, ali lakše dolazi do potprilagođavanja.

Za algoritam k najbližeg suseda, za $k = 1$, postoji zanimljiv teorijski rezultat. Definišimo prvo pojam *Bajesovog rizika*. Neka je $p(y|x)$ prava (ne ocenjena) uslovna verovatnoća klase za date vrednosti atributa. Klasifikator koji vrši klasifikaciju na sledeći način

$$y = \operatorname{argmax}_y p(y|x)$$

naziva *Bajesov klasifikator*. Pritom, kako raspodela $p(y|x)$ nije poznata, ovaj klasifikator se ne može konstruisati, ali može poslužiti za teorijska razmatranja. Rizik Bajesovog klasifikatora je *Bajesov rizik*. Neka je R stvarni rizik algoritma 1 najbližeg suseda, neka je R^* Bajesov rizik i neka je M broj klasa. Tada važi

$$R^* \leq R \leq R^* \left(2 - \frac{M}{M-1} R^* \right)$$

Ugrubo, stvarni rizik ovog algoritma je ne više od dva puta veći od teorijski najmanjeg rizika koji se može postići bilo kakvim algoritmom.

Bitno je znati kada se algoritam k najbližih suseda ponaša posebno loše. U visokodimenzionalnim prostorima, rastojanja se ponašaju drugačije nego što po intuiciji trodimenzionalnog prostora očekujemo. Razmotrimo primer dva niza zapremina n dimenzionalnih kocki – sa stranicom dužine 1 i stranicom dužine 0.99. Važi

$$\lim_{n \rightarrow \infty} \frac{V_{0.99}^n}{V_1^n} = \lim_{n \rightarrow \infty} \frac{0.99^n}{1^n} = 0$$

Slično se može pokazati i za lopte vrlo bliskih poluprečnika, sa centrom u istoj tački. Zaključujemo da su u visokodimenzionalnim prostorima praktično sve

tačke lopte vrlo daleko od njenog centra. Štaviše da su uglavnom na istom rastojanju od centra (npr. na rastojanju većem od $0.99r$, a manjem od r). Nekoliko teorijskih rezultata pokazuje da je u visokodimenzionalnim prostorima varijacija distanci između nasumice generisanih tačaka mala. Ovi uvidi su obeshrabrujući za primenu algoritma k najbližih suseda, pošto se on zasniva na razlikovanju bliskih i dalekih suseda, a ako su te razlike male, njegova predviđanja nisu pouzdana. Ovo se primećuje i u praksi.

Još jedan problem ovog algoritma u slučaju visokodimenzionalnih prostora je prokletstvo dimenzionalnosti. Ukoliko se tačka klasificiše na osnovu bliskih suseda, očekuje se da postoje bliski susedi. To je moguće ako je prostor gusto popunjeno podacima za obučavanje. Međutim kako dimenzionalnost prostora raste, broj tačaka potreban za održavanje nekog konstantnog stepena gustine raste eksponencijalno u odnosu na dimenziju i nije za očekivati da će dovoljna količina podataka biti na raspolaganju, a ako i jeste, može se ispostaviti da predviđanja budu računski skupa.

Postoje još neki problemi primene ovog algoritma koji, na sreću, nisu fundamentalni, odnosno mogu se otkloniti određenim preprocesiranjima. Prvo pitanje je pitanje atributa koji se mere na različitim skalamama. Razmotrimo konkretnije slučaj euklidske metrike

$$d(x, x') = \sqrt{\sum_{i=1}^n (x_i - x'_i)^2}$$

Ukoliko atributi x_1 i x_2 predstavljaju dužine, ali je prvi meren u milimetrima, a drugi u metrima, razlika $(x_1 - x'_1)^2$ će tipično biti mnogo veća od razlike $(x_2 - x'_2)^2$, prosto zbog različite skale na kojoj se vrednosti izražavaju, a ne zbog razlike u stvarnim dužinama. Otud će atribut x_1 više uticati na ishod klasifikacije, nego atributa x_2 , što ne mora biti opravdano. Otud se pre primene algoritma k najbližih suseda obavezno primenjuje neki vid preprocesiranja koji svodi različite atrbute na istu skalu. Na primer, standardizacija.

Drugo pitanje je pitanje ponavljenih atributa ili njihove visoke koreliraneosti. Iako ponavljanje baš istih atributa nije verovatno, razmatranje tog slučaja očiglednije ukazuje na problem. U praksi problem ne mora biti toliko drastičan, jer atributi nisu ponavljeni, ali visoka korelacija vodi vrlo sličnoj vrsti problema. Neka su podaci opisani pomoću dva atributa. Neka je potom jedan umnožen 100 puta. U euklidskom rastojanju će razlika po prvom atributu figurisati jednom, a po drugom 100 puta. Otud će uticaj prvog atributa na predviđanje biti zanemarljiv u odnosu na uticaj drugog, a oba mogu biti podjednako informativna, ili čak prvi može biti informativniji. Ovakvi problemi se rešavaju ili ublažavaju tehnikama smanjenja dimenzionalnosti podataka poput *analyze glavnih pravaca* (eng. principal component analysis).

Da sumiramo, prednosti algoritma k najbližih suseda su jednostavnost, laka implementacija i primena i proizvoljni oblici granica između klasa. Negativne strane su loše ponašanje predviđanja u visokodimenzionalnim prostorima, neotpornost na ponovljene i visoko korelirane atrbute, nedostatak interpretabil-

nosti, kao i standardne mane algoritama zasnovanih na instancama – potreba za čuvanjem instanci iz skupa za obučavanje i više vreme primene modela.

Algoritam k najbližih suseda se može primeniti i na problem regresije. U tom slučaju, uzimanje većinske vrednosti među vrednostima ciljne promenljive suseda nema smisla, pošto se vrednosti na kontinualne promenljive često neće ponavljati. Prirodniji pristup je za nepoznatu instancu naći k najbližih suseda i uprosećiti vrednosti ciljne promenljive tih instanci. I u slučaju regresije važe prethodne primedbe vezane za prednosti i mane algoritma.

Glava 6

Evaluacija i izbor modela

Evaluacija modela predstavlja kvantifikaciju njegove sposobnosti predviđanja. Ukoliko imamo na raspolaganju konačan broj modela, od kojih je potrebno koristiti jedan, očigledno se postavlja pitanje *izbora modela*, koje se obično rešava tako što se na neki način evaluiraju svi raspoloživi modeli i zabere se najbolji. Iako navedeno zvuči trivijalno, to nije slučaj, o čemu svedoči i činjenica da se u praksi, bilo akademskoj, bilo industrijskoj, često prave greške baš u ovim poslovima. Tehnike kojima se ovi poslovi vrše mogu biti netrivijalne za razumevanje, a nekad i za implementaciju, a često bivaju potcenjene, upravo zbog utiska jednostavnosti koje ove teme na prvi pogled ostavljaju.

Evaluacija modela počiva na *merama kvaliteta modela* i na *tehnikama evaluacije modela*. Kako izbor modela počiva na evaluaciji modela, tehnike koje se koriste su slične, što doprinosi konfuziji i potencijalnim greškama. U nastavku ćemo izbor i evaluaciju modela prikazati paralelno zbog sličnosti tehnika, ali pazeci da uvek naglasimo koja tehnika se za šta koristi.

6.1 Mere kvaliteta modela

Mere kvaliteta modela zavise od vrste problema koji se rešava, kao i od željenih ishoda. Mogu se osmislti za konkretan problem, ali mi ćemo se baviti opštim merama koje se često koriste u različitim kontekstima.

Mere koje se najčešće koriste za klasifikaciju su *tačnost klasifikacije* (eng. *classification accuracy*), *preciznost i odziv* (eng. *precision and recall*), F_1 mera i površina ispod ROC krive (eng. *area under the curve – AUC*).

Praktično sve često korišćenje mere kvaliteta klasifikacije počivaju na *matrici konfuzije* (eng. *confusion matrix*) i pojmovima vezanim za nju. Ovo je matrica C čiji element c_{ij} predstavlja broj elemenata klase i koji su klasifikovani u klasu j . Klasifikacija je očito najbolja kada je ova matrica dijagonalna, što znači da je klasifikacija potpuno ispravna. Nedijagonalni elementi označavaju greške. U slučaju binarne klasifikacije, obično se jedna klasa naziva *pozitivnom*, a druga *negativnom*. Tada matrica konfuzije ima specifičan

Stvarno/Predviđeno		Pozitivno	Negativno
Pozitivno	stvarno pozitivno (TP)		lažno negativno (FN)
Negativno	lažno pozitivno (FP)		stvarno negativno (TN)

Tabela 6.1: Matrica konfuzije

oblik prikazan tabelom 6.1. *Stvarno pozitivne* (eng. *true positive*) instance su pozitivne instance koje su od strane modela prepoznate kao pozitivne. Broj takvih instanci skraćeno označavamo *TP*. *Stvarno negativne* (eng. *true negative*) instance su negativne instance koje su od strane modela prepoznate kao negativne. Broj takvih instanci skraćeno označavamo *TN*. *Lažno pozitivne* (eng. *false positive*) instance su negativne instance koje su od strane modela proglašene pozitivnim. Broj takvih instanci skraćeno označavamo *FP*. *Lažno negativne* (eng. *false negative*) instance su pozitivne instance koje su od strane modela proglašene negativnim. Broj takvih instanci skraćeno označavamo *FN*.

Tačnost klasifikacije predstavlja udeo tačno klasifikovanih instanci u ukupnom broju instanci. U slučaju binarne klasifikacije, može se izraziti kao

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}$$

Iako vrlo intuitivna, tačnost klasifikacije ne mora uvek biti pogodna mera kvaliteta. Jedan razlog je nenačinljivost u slučaju da klase imaju vrlo različit broj instanci. Ukoliko jedna klasi pripada 99% instanci, a drugoj 1%, naizgled impresivna tačnost od 0.99 može biti postignuta tako što će sve instance biti klasifikovane u prvu klasu. Ipak, takav klasifikator je beskorisan. Pritom, klase će često biti neizbalansirane, a čak i ekstremni slučajevi su realistični. Na primer, u slučaju detekcije prevara sa kreditnim karticama, detekcije retkih bolesti i slično.

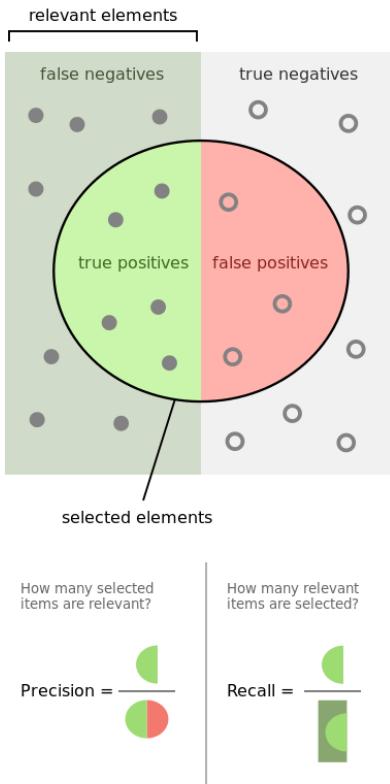
Preciznost i odziv se obično koriste pri evaluaciji sistema za pronalaženje informacija. Ipak i taj problem se može smatrati problemom klasifikacije – razdvajanja bitnih od nebitnih informacija pri čemu se bitne prikazuju korisniku. Preciznost je udeo pozitivnih instanci u svim instancama koje su proglašene pozitivnim, odnosno

$$Prec = \frac{TP}{TP + FP}$$

i odgovara na pitanje koliko puta smo bili u pravu kad smo tvrdili da je nešto relevantno. Odziv je udeo pronađenih pozitivnih instanci u svim pozitivnim instancama, odnosno

$$Rec = \frac{TP}{TP + FN}$$

i odgovara na pitanje koliko relevantnih informacija smo našli od svih relevantnih informacija. Preciznost i odziv su ilustrovani na slici Figure 6.1.



Slika 6.1: Ilustracija pojmove preciznosti i odziva.

Preciznost i odziv pojedinačno nisu korisne mere. Naime, proglašavajući sve instance za pozitivne, odziv je maksimalan, a ne proglašavajući nijednu, ne pravi se nijedna greška, pa je preciznost maksimalna. Stoga ih ima smisla posmatrati samo zajedno. Način na koji se to najčešće radi je tako što se izračuna F_1 mera – njihova harmonijska sredina

$$F_1 = 2 \frac{Prec \cdot Rec}{Prec + Rec}$$

Treba imati u vidu da ova mera nije simetrična u odnosu na izbor neke klase kao pozitivne. U slučaju prethodno pomenutog klasifikatora koji klasificiše sve instance u prvu klasu, ako se ona smatra pozitivnom preciznost je 0.99, a odziv 1, što daje F_1 meru od 0.99, što deluje sjajno. Ali ako se ista mera računa u odnosu na drugu klasu, preciznost je 0.01, a odziv 1, što daje F_1 meru 0.02, što uopšte ne deluje sjajno. Uprosečavanjem ova dva slučaja, dobija se 0.51, što daje bolju sliku o kvalitetu klasifikacije nego tačnost.

Jedna mera koja se češće koristi u slučaju binarne klasifikacije sa neizbalansiranim klasama je površina ispod ROC krive. Pretpostavlja se da klasifikator počiva na nekom modelu koji različitim instancama pridružuje neke skorove. Logistička regresija i metod potpornih vektora su primeri takvih metoda. Obično se takav skor prevodi u klasu nekom vrstom zaokruživanja u odnosu na neki prag. Recimo, u pomenutim metodima, vrednosti linearne modela koje su manje od nule označavaju jednu klasu, a one koje su veće od nule, označavaju drugu. Osnovni smisao površine ispod ROC krive, koja će tek biti definisana, je da se proveri da li ovaj metod dodeljuje niže skorove elementima jedne klase, a više elementima druge klase, nevezano od nekog konkretnog praga – moguće je da izbor praga nije idealan i da bi se čak mogao bolje podešiti, pa ga zanemarujemo. Klasu kojoj odgovaraju niže vrednosti skora nazovimo negativnom a onu kojoj odgovaraju više vrednosti nazovimo pozitivnom. Površina ispod ROC krive je ideo parova instanci iz dve klase takvih da je instanci iz negativne klase pridružen manji skor nego instanci iz pozitivne klase u ukupnom broju parova instanci iz različitih klasa. Formalnije, neka su C_1 i C_2 skupovi instanci iz različitih klasa, neka je $N_1 = |C_1|$ i $N_2 = |C_2|$ i neka je f model koji pridružuje skor instancama. Neka je $r(i)$ funkcija koja svakoj instanci i dodeljuje rang u sortiranom nizu instanci u odnosu na vrednost modela f . Ukoliko su instance izjednačene, za rang se uzima prosek svih indeksa (koji počinju od 1) koji odgovaraju tim instancama u sortiranom nizu. Na primer, ukoliko su skorovi 1, 2, 3, 3, 3, 4, 4, 5, rangovi su 1, 2, 4, 4, 4, 6.5, 6.5, 8. Ako se C_1 smatra negativnom, a C_2 pozitivnom klasom, tada važi

$$AUC = \frac{1}{N_1 N_2} \left(\sum_{i \in C_2} r(i) - \frac{N_2(N_2 + 1)}{2} \right)$$

U zagradi se nalazi razlika sume rangova instanci prve klase i sume rangova koja bi im pripadali da svi imaju manje skorove od svih instanci druge klase. Ukoliko ne bi bilo izjednačenih instanci, važila bi i sledeća jednakost

$$AUC = \frac{|\{(i_1, i_2) | i_1 \in C_1 \wedge i_2 \in C_2 \wedge f(i_1) < f(i_2)\}|}{N_1 N_2}$$

Odavde je očigledno da AUC ocenjuje verovatnoću da pri slučajnom izboru dve instance iz različitih klasa ona iz negativne klase ima manji skor od one iz pozitivne klase. Očito, ako je vrednost AUC velika, postoji neki prag koji dobro razdvaja dve klase. U suprotnom, ne postoji. Minimalna vrednost mere AUC je 0.5. Ukoliko bi se dobila manja vrednost, jednostavnom promenom znaka skora, dobila bi se veća. Ova mera nije osetljiva na neizbalansiranost klasa. Razmotrimo pomenuti klasifikator koji sve instance klasificuje u istu klasu, recimo tako što svima daje vrednost skora 1. Tada svima pripada rang $(N + 1)/2$, pa je AUC

$$\frac{1}{0.99 \cdot 0.01 N^2} \left(\sum_{i=1}^{0.99N} \frac{N+1}{2} - \frac{0.99N(0.99N+1)}{2} \right) =$$

$$\frac{1}{0.99 \cdot 0.01N^2} \left(\frac{0.99N(N+1)}{2} - \frac{0.99N(0.99N+1)}{2} \right) = \\ \frac{1}{0.99 \cdot 0.01N^2} \left(0.99N \frac{0.01N}{2} \right) = 0.5$$

što je minimalna vrednost, pa ukazuje na to da je kvalitet klasifikatora najgori.

Jedna stvar koja bi trebalo da je privukla pažnju je neobičan naziv ove mere. Način na koji je definisana ne uključuje nikakvu krivu. Zapravo, izabrali smo da je definišemo kao normiranu vrednost statistike Man-Vitni-Vilkoksonovog testa zbog jednostavne interpretacije u terminima verovatnoće. Alternativna definicija je drugačija. *ROC kriva* (eng. receiver operating characteristic curve) je kriva u koordinatnom sistemu udela FP instanci (x osa) u skupu svih instanci i udela TP instanci (y osa) u skupu svih instanci koji zaprema jedinični kvadrat. Za svaku vrednost praga na skorovima koje klasifikator daje, definisana je jedna tačka u ovom koordinatnom sistemu, a menjajući ovaj skor od minimalnog (na datim instancama) do maksimalnog, dobija se kriva koja spaja koordinatni početak sa tačkom (1, 1). Površina ispod te krive je jednaka prethodno definisanoj veličini *AUC*, ali intuicija iza takve definicije nije očigledna.

Mere koje se najčešće koriste za regresiju su *srednjekvadratna greška* i njen koren (eng. *mean square error*, *root mean square error*), *srednja relativna greška* izražena u procentima (eng. *mean absolute percentage error*) i *koeficijent determinacije*, poznatiji pod oznakom R^2 . Važi

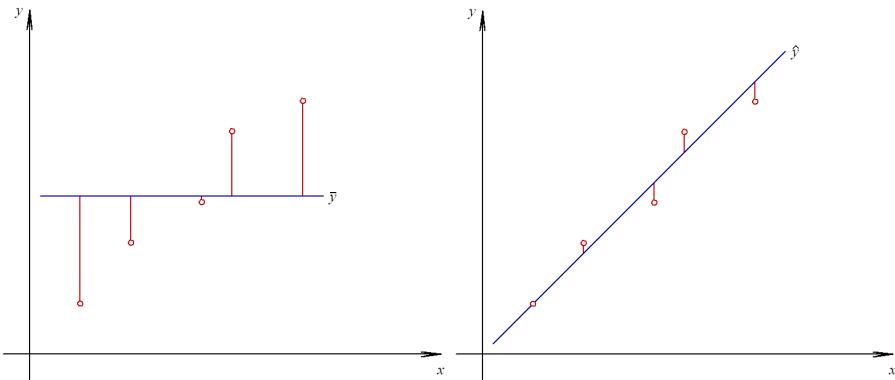
$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - f(x_i))^2$$

Značaj *MSE* je očigledan, pošto se radi o veličini koja se direktno minimizuje u mnogim regresionim problemima. *RMSE*, koren veličine *MSE* je nešto korisniji, pošto je ova mera izražena na istoj skali na kojoj i ciljna promenljiva, što olakšava interpretaciju dobijenih vrednosti. Zapravo, interpretabilnost je važna prednost ove mere. Ukoliko se vrši predviđanje troškova u nekom poslu i procenom kvaliteta modela dobije se *RMSE* = 10.000 u dinarima. Jasno je da su šanse male da model pogreši za milion dinara. Ako nam je greška reda veličine 10.000 – 30.000 dinara prihvatljiva, možemo koristiti model.

Zbog jakog uticaja odudarajućih podataka na ovu grešku i zbog potrebe da se greška nekada izrazi u odnosu na vrednost ciljne promenljive, koristi se i srednja relativna greška, najčešće izražena u procentima, koja se definiše kao

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

Prethodne mere daju informaciju o redu veličine greške u apsolutnim jedinicama ili relativno u odnosu na vrednost ciljne promenljive. Drugi način evaluacije bi bio da se odmeri koliki je napredak ostvaren učenjem u odnosu na



Slika 6.2: Greške kojima se definišu varijansa i koeficijent determinacije. Mogu se posmatrati kao promena promenljive y koju odgovarajući model nije uspeo da objasni promenom promenljive x .

neki trivijalan prediktivni metod koji bi nam bio dostupan i bez učenja. Takođe prediktivnih metoda bi moglo biti raznih. Na primer, slučajno pogađanje. Međutim takav „metod“ nema veze sa podacima. Jedan koji ima, a ipak je trivijalan je uzimanje prosečne vrednosti ciljne promenljive na nekom uzorku za buduće konstantno predviđanje. Takav metod već ima smisla ne samo zbog jednostavnosti, već zato što pokušava da predviđa vrednost ciljne promenljive bez uspostavljanja veze sa vrednostima atributa, pa je zanimljivo videti koliko uspostavljanje takve veze pomaže u predviđanju u odnosu na metod koji takvu vezu ne koristi. Koeficijent determinacije, odnosno R^2 , nekog modela predstavlja udeo varijanse ciljne promenljive koji je objašnjen tim modelom. Šta to znači? Podsetimo se da modele mašinskog učenja, nalik fizičkim modelima, možemo razumeti kao pokušaj da se objasni variranje neke promenljive na osnovu promena u drugim promenljivim. U fizici, ukoliko dođe do promene temperature gasa, doći će i do promene pritiska. Drugim rečima, promena pritiska je objašnjena promenom temperature. Slično možemo posmatrati i modele mašinskog učenja. Udeo objašnjene varijanse ciljne promenljive se može izraziti preko udela preostale varijanse koju ne uspevamo da objasnimo modelom, što je zapravo MSE . Stoga važi

$$R^2 = 1 - \frac{MSE}{\text{var}[y]}$$

Na slici 6.2 prikazane su greške koje pravimo ukoliko za predviđanje koristimo prosek i greške koje pravimo ukoliko za predviđanje koristimo neki model. Srednjekvadratna greška u odnosu na prosek je $\text{var}[y]$, a u odnosu na model je MSE . R^2 upravo poređi ove greške.

6.2 Tehnike evaluacije i izbora modela

Tehnike izbora i evaluacije variraju po svojoj složenosti u zavisnosti od željene pouzdanosti ocene, količine podataka i svojstava algoritama učenja čiji se modeli evaluiraju. Koja god tehnika izbora ili evaluacije da se koristi, mora biti ispoštovano glavno načelo evaluacije modela, a to je da **podaci korišćeni u evaluaciji modela ni na koji način ne smeju biti korišćeni prilikom njegovog obučavanja**. Iako deluje jednostavno, ne može se dovoljno naglasiti potreba za pažnjom pri sprovođenju ovog načela u delo. Naime, greške koje se prave u evaluaciji su najčešće ove prirode, a da ljudi koji evaluaciju vrše toga nisu ni svesni. Probleme evaluacije i izbora modela ćemo u nastavku prikazati u dva slučaja. Prvi je pojednostavljeni slučaj u kojem se pretpostavlja da algoritam koji se koristi nije konfigurabilan. Drugi pretpostavlja konfigurabilnost algoritma.

6.2.1 Izbor modela u slučaju nekonfigurabilnog algoritma učenja

Izbor modela u slučaju nekonfigurabilnog algoritma je prilično jednostavan, pošto u nedostatku konfigurabilnosti, ne postoje ni mogućnosti izbora, osim po pitanju podataka na kojima se obučavanje vrši. Međutim, odgovor na to pitanje je jednostavan – svi podaci su vredni i model koji želimo da koristimo u budućnosti treba obučavati na svim dostupnim podacima.¹ Prvi slučaj uopšte nije realističan, ali omogućava da se koncepti usvoje postepeno. Ovaj model ćemo označiti sa M .

6.2.2 Evaluacija modela u slučaju nekonfigurabilnog algoritma učenja

Ključno pitanje evaluacije modela M je kako ga evaluirati, a da se u evaluaciji ne koriste podaci na kojima je model obučavan, kad je model obučavan na svim dostupnim podacima. Jedna mogućnost bi bila da se model M , namenjen budućoj upotrebi, ne obučava na svim podacima, već samo na delu podataka, kako bi neki preostali za evaluaciju. Ipak, korišćenje manje količine podataka u obučavanju povlače manju pouzdanost modela, što nije poželjno za model koji ćemo koristiti u praksi. Ipak, moguće je napraviti kompromis između ova dva suprotstavljenja zahteva. Naime, model M će svakako biti obučavan na svim podacima. Stoga ne može biti direktno evaluiran. Međutim, taj model može biti aproksimiran modelom M' koji je obučavan na većini podataka. Evaluacijom modela M' aproksimativno se evaluira model M .

Najjednostavnija tehnika evaluacije je *evaluacija pomoću skupa za testiranje*. Ukupni podaci se dele na dva skupa – skup za obučavanje i skup za

¹Osim u slučaju podataka na koje algoritam ne može biti primenjen (na primer, poput podataka koji nemaju vrednosti nekih atributa), odudarajućih podataka za koje se smatra da nisu od značaja i slično.

x_1	x_2	x_3	y
1	9	0	8
0	6	2	1
1	3	1	5
4	9	7	6
1	1	6	7
7	2	3	4
2	9	9	9
3	3	4	6
7	2	1	7
6	5	1	5

Tabela 6.2: Primer podele podataka na skup za obučavanje (plavo) i skup za testiranje (crveno).

testiranje. Skup za obučavanje je obično veći – ugrubo dve trećine (ali ne nužno) i na njemu se određuje model M' , koji se potom primenjuje na skup za testiranje, čime se dobijaju njegova predviđanja, koja se onda nekom merom kvaliteta mogu oceniti u odnosu na tačne vrednosti ciljne promenljive. Podela je ilustrovana tabelom 6.2.

Mana ovakvog pristupa ima više. Prvo, neke instance su izabrane da budu u skupu za obučavanje, a neke da budu u skupu za testiranje. Od načina na koji je izvršena podela zavisiće i ocena kvaliteta. Za različite podele, ova ocena može značajno varirati. Pritom, greška ocene može biti vrlo velika ako se skup za testiranje pristrasno izabere. Na primer, ako sadrži samo instance jedne klase ili, u slučaju regresije, ako sadži samo instance sa najvišom (ili najnižom) vrednošću ciljne promenljive.

Odgovor na neke od prethodnih problema je tehnika *K-slojne unakrsne validacije* (eng. *K-fold cross-validation*). Sprovodi se na sledeći način:

- Podatke \mathcal{D} podeliti na K približno jednakih podskupova, takozvanih *slojeva* (eng. *folds*) $\{S_1, \dots, S_K\}$
- Za $i = 1, \dots, K$
 - Obučiti model na podacima $\mathcal{D} \setminus S_i$
 - Izvršiti predviđanja dobijenim modelom na sloju S_i
- Izračunati ocenu kvaliteta na osnovu svih predviđanja na celom skupu \mathcal{D}

Izlutracija podele podataka pri unakrsnoj validaciji data je tabelom 6.3.

Potrtajmo da se u unakrsnoj validaciji i dalje ne ocenjuje direktno model M , već se formira niz modela M'_1, \dots, M'_K koji svi aproksimiraju model M . Činjenicu da se model M nikad ne evaluira direktno ne treba gubiti izvida, jer to u komplikovanim scenarijima vodi zabuni. Često je pitanje koji od K

x_1	x_2	x_3	y
1	9	0	8
0	6	2	1
1	3	1	5
4	9	7	6
1	1	6	7
7	2	3	4
2	9	9	9
3	3	4	6
7	2	1	7
6	5	1	5

Tabela 6.3: Prikaz podele podataka pri unakrsnoj validaciji. Različite boje predstavljaju različite slojeve.

modela obučenih u unakrsnoj validaciji treba koristiti ili kako ih ukombinovati. Odgovor je nijedan, odnosno nikako. Treba koristiti model M koji oni aproksimiraju.

Napomenimo da se ocena kvaliteta računa tek nakon što su izračunata sva predviđanja. U praksi se često pravi greška da se mere kvaliteta računaju na svakom sloju pojedinačno, pa se na kraju uproseće. To može biti podjednako dobro u slučaju nekih mera poput MSE koje predstavljaju zbirove, ali u slučaju drugih, poput R^2 , ne.

Očigledna prednost unakrsne validacije u odnosu na prethodni metod je manja varijansa ocene greške usled toga što se ocena greške računa na većoj količini podataka. Drugim rečima, ocena je pouzdanija. S druge strane, model se ne obučava jednom, već K puta, što može biti vremenski zahtevno. Takođe, postavlja se pitanje i izbora broja K . U praksi se za K koriste vrednost 5 i 10. U slučaju malih podataka, nekad se koristi vrednost $K = N$ (eng. *leave one out*). Ipak, teorijski je pokazano da ovaj pristup vodi optimističnoj proceni kvaliteta, pa se ne preporučuje. Unakrsnom validacijom nije rešen problem pristrasnog izbora podskupova, a o tome će biti reči kasnije.

6.2.3 Izbor modela u slučaju konfigurabilnog algoritma učenja

Algoritam učenja može biti konfigurabilan po različitim aspektima. Recimo, mogu imati metaparametre poput regularizacionih metaparametara, parametra tolerancije kod metoda potpornih vektora za regresiju, parametara kernela, itd. Takođe, izbor podskupa atributa nad kojim se uči predstavlja vid konfigurabilnosti. U slučaju neuronskih mreža, moguće je birati različite arhitekture koje se razlikuju po različitim aspektima. Čak i izbor algoritma (npr. neuronska mreža ili metod potpornih vektora) se može smatrati konfigurablešću procesa učenja. Skup izbora za sve ove aspekte učenja nazivaćemo

konfiguracijom. Značaj konfigurabilnosti je u tome što za dati skup podataka, različite konfiguracije vode različitim modelima i postavlja se pitanje kako izabrati najbolju konfiguraciju, a time i model. Odgovor nije težak – za različite konfiguracije, potrebno je obučiti modele koji im odgovaraju, evaluirati ih, izabrati konfiguraciju koja daje najbolji model i pomoću njе obučiti finalni model na svim podacima. Ali zašto birati konfiguraciju, a ne baš model, kad je model to što nam treba? Da li smo samo bili neprecizni u izražavanju? Ipak nismo. Model koji se obučava prilikom evaluacije obučava se na delu skupa podataka kako bi ostao deo podataka za testiranje. Štaviše, u unaprjednoj validaciji se obučava K modela, pa se postavlja i pitanje koji je to model koji treba izabrati od tih K koji su dobijeni za istu konfiguraciju. Oba zapažanja su u vezi sa već naglašavanom činjenicom da metod evaluacije nikad i ne evaluira direktno model koji će biti korišćen već njegove aproksimacije. Ako je \mathcal{K} unapred definisani skup konfiguracija, ceo postupak je sledeći:

- Za svaku konfiguraciju $K \in \mathcal{K}$
 - Uraditi evaluaciju algoritma za konfiguraciju K nekim od metoda koji se koristi u slučaju nekonfigurabilnih algoritama i zapamtiti ocenu kvaliteta.
- Pomoću konfiguracije za koju je ocena kvaliteta najbolja, obučiti model M na svim podacima \mathcal{D} .

6.2.4 Evaluacija modela u slučaju konfigurabilnog algoritma učenja

Ova vrsta evaluacije predstavlja jedan od najčešćih izvora grešaka u praksi mašinskog učenja, u kojoj je neophodna – uvek. Osnovna greška sastoji se u tome da se ocena kvaliteta dobijena prilikom izbora modela prijavi kao ocena kvaliteta tog modela. **To je pogrešno.** Naime, osnovno načelo izbora i evaluacije modela je da podaci korišćeni u evaluaciji modela ni na koji način ne smeju biti korišćeni prilikom njegovog obučavanja. Međutim, proces obučavanja uključuje sve korake dolaženja do modela. Kako izbor konfiguracije uslovjava izbor modela, i izbor konfiguracije je deo obučavanja modela. Kako je izbor konfiguracije rađen na osnovu svih podataka, to znači da je navedeno načelo prekršeno! Stoga su potrebne sofisticirane tehnike.

Prva tehnika je analogon evaluacije pomoću skupa za testiranje, ali malo komplikovanija. Radi se o *evaluaciji pomoću skupova za validaciju i testiranje*. U njoj će umesto podele na dva skupa biti korišćenja podela na tri skupa. Tehnika se sprovodi kroz naredne korake:

- Iz podataka izdvojiti skup za testiranje \mathcal{T}
- Na podacima $\mathcal{D} \setminus \mathcal{T}$ izvršiti izbor modela pomoću skupa za testiranje i zapamtiti najbolju konfiguraciju

- Pomoću te konfiguracije obučiti model M' na podacima $\mathcal{D} \setminus \mathcal{T}$ i izvršiti predviđanje na podacima iz skupa \mathcal{T}
- Oceniti kvalitet predviđanja i prijaviti tu ocenu

Dobijena ocena kvaliteta je ocena modela M opisanog u prethodnom odeljku! Skup za testiranje koji se koristi u izboru modela naziva se *validacioni skup*.

Evaluacija pomoću skupova za validaciju i testiranje pati od istih problema kao i evaluacija pomoću skupa za testiranje. Kao što je u tom kontekstu upotrebljena unakrsna validacija, tako može biti upotrebljena i u ovom, samo što je u ovom kontekstu postupak komplikovaniji i naziva se *ugnežđena² unakrsna validacija*. Sprovodi se na sledeći način:

- Podatke \mathcal{D} podeliti na K približno jednakih podskupova (tzv. slojeva) $\{S_1, \dots, S_K\}$
- Za $i = 1, \dots, K$
 - Izvršiti izbor modela pomoću unakrsne validacije na podacima $\mathcal{D} \setminus S_i$ i zapamtiti najbolju konfiguraciju
 - Pomoću te konfiguracije obučiti model M' na podacima $\mathcal{D} \setminus S_i$ i izvršiti predviđanje na podacima iz skupa S_i
- Izračunati ocenu kvaliteta na osnovu svih predviđanja na celom skupu \mathcal{D}

Ugnežđena unakrsna validacija u odnosu na evaluaciju pomoću skupova za validaciju i testiranje ima iste prednosti koje unakrsna validacija ima u odnosu na evaluaciju pomoću skupa za testiranje. Takođe, ima i iste mane. Pritom, one su još izraženije jer je broj obučavanja modela za K struku ugnezđenu unakrsnu validaciju K^2 . Tim pre se koriste male vrednosti parametra K , poput 5 ili 10.

Dobra preporuka je da se svaka metoda evluacije izvede više puta sa sa slučajnim podelama podataka na podskupove i da se rezultati uproseče. Nаравно, praktična primenljivost ovog saveta zavisi od veličine skupa podataka i računskih resursa.

6.3 Napomene vezane za pretprecesiranje

Unakrsna validacija i dalje ne rešava problem pristrasnog izabora skupa za testiranje. Isto važi za ugnežđenu unakrsnu validaciju. Naime, ako se recimo u problemu regresije instance sortiraju po ciljnoj promenljivoj i onda u tom poretku podele na K slojeva, prvi sloj će sadržati instance sa najmanjim vrednostima ciljne promenljive. Takve vrednosti se ne nalaze u preostalih $K - 1$ slojeva i stoga model obučen na njima ekstrapolira kada se primeni na prvi sloj. U takvim slučajevima se ne očekuju dobre performanse. Razlog je što

²Da, možete proveriti u pravopisu da je ovo ispravan oblik ove reči. Alternativno, može i ugnježđena.

x_1	x_2	x_3	y
0	6	2	1
7	2	3	4
1	3	1	5
6	5	1	5
4	9	7	6
3	3	4	6
1	1	6	7
7	2	1	7
1	9	0	8
2	9	9	9

Tabela 6.4: Ilustracija podele podataka stratifikovane po ciljnoj promenljivoj. Različite boje predstavljaju različite podskupove.

su parametri raspodele na test skupu značajno različiti od onih na skupu za obučavanje, pa se od algoritma učenja i ne može očekivati mnogo. Otud se prilikom bilo kakvih podela podataka često vodi računa o tome da svi podskupovi imaju sličnu raspodelu kao i ukupan skup podataka. Pritom, to je teško ili nemoguće postići, osim ako imamo puno podataka male dimenzionalnosti. U tom slučaju i slučajan uzorak često zadovoljava pomenute zahteve. Tehnike koje pokušavaju da zadovolje zahtev da podskupovi imaju istu raspodelu atributa i ciljne promenljive nazivaju se tehnikama *stratifikacije*. Ovakve tehnike predstavljaju vid preprocesiranja podataka. Pojednostavljena varijanta, koju je uvek moguće izvesti je *stratifikacija po ciljnoj promenljivoj*. Podela na K delova stratifikovana po ciljnoj promenljivoj sprovodi se na sledeći način:

- Sortirati instance u odnosu na ciljnu promenljivu. Ako je ciljna promenljiva kategorička, to se može uraditi tako što se svakoj njenoj vrednosti pridruži različit broj.
- Za $i = 1, \dots, K$
 - Instance sa indeksima $i + j * K$ za $j = 0, 1, \dots$, svrstati u podskup P_i .

Na ovaj način svi podskupovi imaju približne raspodele ciljne promenljive. Ilustracija je data tabelom 6.3.

Česta greška u praksi mašinskog učenja je u lošoj primeni različitih vidova preprocesiranja. Takva greška je primena standardizacije na ceo skup podataka, pre podele na podskupove koji se koriste u evaluaciji. Grešku je najlakše uočiti na primeru evaluacije pomoću skupa za testiranje. Prilikom standardizacije, vrednosti atributa na skupu za testiranje utiču na prosek i standardnu devijaciju koji se koriste pri standardizaciji. S druge strane, podaci na kojima će se ubuduće model primenjivati nisu dostupni da daju svoj doprinos ovim veličinama. Jedan kontraargument bi mogao biti da to nije važno jer mašinsko

učenje ionako može raditi samo u slučaju da je raspodela tih budućih podataka jednaka raspodeli podataka za obučavanje. Ipak, i pod tim uslovom, veličina podataka za obučavanje je retko dovoljno velika da nas u budućnosti ništa ne može makar malo iznenaditi, a time i dovesti do veće greške modela. Izbegavanje zajedničke standardizacije celog skupa podataka omogućava baš taj efekat – daje šansu podacima iz skupa za testiranje da nas iznenade i time ocenu greške modela učine realističnijom. Ova diskusija je data na primeru evaluacije pomoću skupa za testiranje i standardizacije. Ipak, ona je opštija i odnosi se i na ostale tehnike evaluacije i na druge tehnike pretprocesiranja, poput smanjenja dimenzionalnosti i izbora podskupa atributa.

Glava 7

Regularizacija

Značaj regularizacije je već naglašen u kontekstu smanjenja fleksibilnosti modela i predupređivanja preprilagođavanja. Iako je to najznačajnija uloga regularizacije, ipak nije jedina. Regularizacija nam može pomoći i u nametanju određene strukture modelu, što može biti vrlo korisno, kao što ćemo se uveriti uskoro, u uključivanju domenskog znanja u model i slično. U nastavku će biti diskutovano nekoliko poznatih i široko primenljivih vrsta regularizacije.

7.1 Proređeni modeli

Jedno specifično svojstvo modela mašinskog učenja na koje se obraća posebna pažnja je *proređenost modela* (eng. *model sparsity*). Model se smatra utoliko proređenijim ukoliko ima veći broj koeficijenata sa vrednošću nula. Ne-kada se za model samo kaže da je proređen, ali ne postoji definisan prag vezan za broj koeficijenata koji bi trebalo da imaju vrednost nula da bi se model smatrao proređenim. Stoga, ovakva kvalifikacija je neformalna, a u strogom smislu može se samo reći da je neki model proređeniji od drugog modela nad istim skupom atributa.

Primetimo da proređenost modela¹ znači da je njegovim obučavanjem obavljen posao *izbora atributa* (eng. *feature selection*), koji predstavlja čest posao u mašinskom učenju. Ovaj posao može biti obavljan odvojenim metodama, ali se obično najefikasnije obavlja ukoliko je tendencija ka proređenim modelima ugrađena u metod obučavanja.

Značaj proređenosti modela (pa time i izbora atributa) je višestruk. Prvo, ukoliko je neki koeficijent modela nula, to znači da postoji nešto u strukturi modela što nije bitno i što se ne mora izračunavati. U slučaju linearog modela, koeficijent sa vrednošću nula znači da neki atribut nije bitan, pa se ne mora ni meriti. Treba imati u vidu da merenje nekih atributa može biti skupo ili nepoželjno iz drugog razloga. Na primer, neke hemijske analize su skupe i poželjno je da model ne zavisi od njihovih rezultata, kako se ne bi vršile. Slično,

¹Evo prethodno pomenute neformalne upotrebe izraza proređenost!

neke medicinske analize su bilo skupe, bilo neugodne ili čak štetne za pacijente. Dalje, ukoliko je neki koeficijent nula, to obično znači da neka od zavisnosti koje model može da izrazi ne postoji, odnosno da je model jednostavniji, što je generalno poželjno svojstvo modela u kontekstu moći generalizacije. I konačno, model sa manje parametara je lakše analizirati i razumeti, odnosno proređeniji model je interpretabilniji.

Treba imati u vidu da se nekada za metod potpornih vektora kaže da daje proređene modele. Ti modeli su proređeni u smislu da ne zavise od svih instanci skupa podataka, već samo od nekih, pošto su Lagranžovi koeficijenti koji im odgovaraju jednaki nuli, ali to nije proređenost u smislu o kojem sada govorimo.

Proređeni modeli se često dobijaju tako što se neki metod učenja modifičuje posebnim vidom regularizacije – najčešće tako što se u minimizacionom problemu kao regularizacioni izraz upotrebi ℓ_1 norma:

$$\|w\|_1 = \sum_{i=1}^n |w_i|$$

Ovaj metod se naziva *lasso* (eng. *lasso – least absolute shrinkage and selection operator*) regularizacijom.

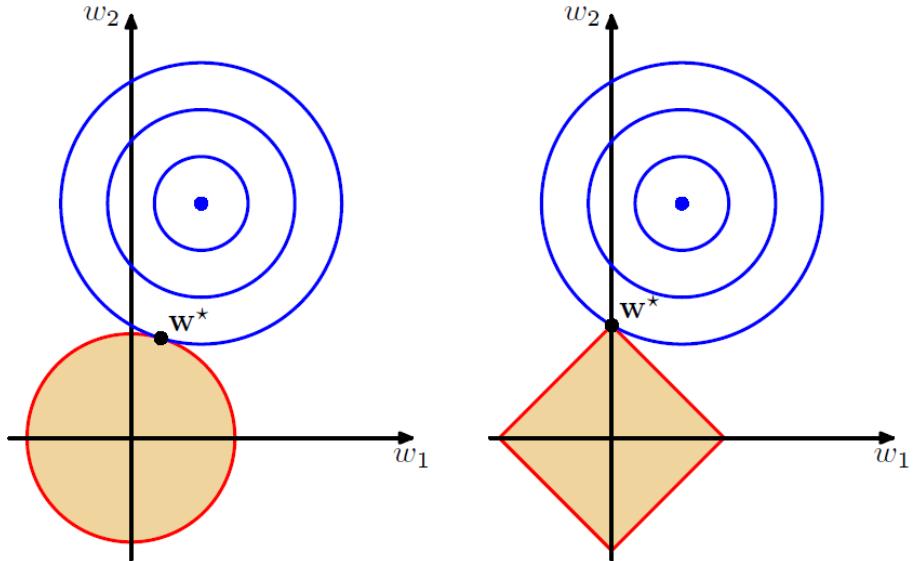
Ključna razlika ℓ_1 regularizacije u odnosu na ℓ_2 regularizaciju je da dok ℓ_2 regularizacija vodi smanjivanju apsolutnih vrednosti koeficijenata, često tako da veliki broj koeficijenata postane mali, ali i dalje različit od nule, ℓ_1 regularizacija vodi tome da neki, manje važni koeficijenti postanu baš jednaki nuli. U specijalnom slučaju linearne regresije u kojem za matricu podataka važi $X^T X = I$, mogu se preciznije okarakterisati efekti ove dve regularizacije. Nalgasimo da dati uslov znači da su atributi normirani i nekorelirani vektori i postoje metode za njegovo obezbeđivanje. Neka je w vektor vrednosti parametara modela koji se dobija bez regularizacije, neka je w' vektor vrednosti parametara koji se dobija pri ℓ_1 regularizaciji, a w'' vektor vrednosti parametara koji se dobijaju pri ℓ_2 regularizaciji. Tada važi:

$$w'_i = \text{sgn}(w_i) \max \left(|w_i| - \frac{\lambda}{2}, 0 \right) \quad w''_i = \frac{w_i}{1 + \lambda} \quad i = 1, \dots, n$$

gde je λ vrednost regularizacionog parametra. Iako prvi izraz na prvi pogled deluje komplikovano, on samo kaže da vrednost parametra u slučaju ℓ_1 regularizacije ostaje istog znaka, ali apsolutne vrednosti umanjene za λ . Ukoliko je lambda dovoljno veliko da $|w_i| - \lambda$ postane negativno, taj koeficijent će imati vrednosti nula. Pod pomenutim specijalnim uslovima, odavde je jasno zašto lasso regularizacija daje proređene modele – koeficijent može postati baš nula ako se od njegove apsolutne vrednosti oduzima konačna vrednost, ali ne i ako se njegova apsolutna vrednost deli konačnom vrednošću. Ipak, intuicija iz ovih izraza nije očigledna. Stoga ćemo se osvrnuti na geometrijsku intuiciju.

Svaki problem oblika

$$\min_w E(w, \mathcal{D}) + \lambda \|w\|_q$$



Slika 7.1: ℓ_2 i ℓ_1 lopta (crveno) i konture srednje greške (plavo). Optimalni parametri u drugom slučaju leže na w_2 osi, što znači da je model 50% proređen.

za $q \in \mathbb{N}^+$ ekvivalentan je problemu oblika

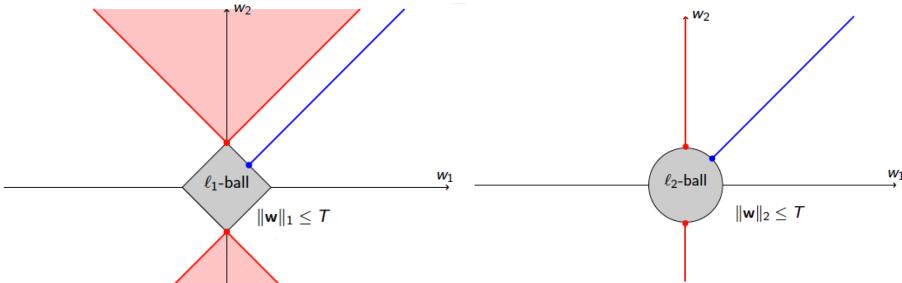
$$\min_w E(w, \mathcal{D})$$

$$\|w\|_q \leq t$$

odnosno, za svaku vrednost metaparametra λ postoji neka vrednost metaparametra t , tako da je rešenje oba problema isto i obratno. Na slici 7.1, prikazani su oblici lopti $\|w\|_q \leq t$ za $q = 1$ i $q = 2$ i konture jednakih vrednosti funkcije E . Kao što se sa slike vidi, zahvaljujući špicastom obliku ℓ_1 lopte, optimalno rešenje u slučaju ℓ_1 regularizacije ima 50% nenula parametara, dok optimalno rešenje u slučaju ℓ_2 lopte ima 100% nenula parametara. Treba primetiti i da je kvalitet rešenja u slučaju ℓ_1 regularizacije nešto gori. Uz dozu opreza, ova dva zapažanja se daju generalizovati. Modeli koji se dobijaju pri ℓ_2 regularizaciji obično nisu proređeni, ali često prave nešto manju grešku od modela koji se dobijaju pri ℓ_1 regularizaciji, dok su ovi drugi obično u manjoj ili većoj meri proređeni.

Na slici 7.2 prikazani su regioni u kojima će ℓ_1 i ℓ_2 regularizacije proizvesti proređene modele za koje je važi $w_1 = 0$, ukoliko je minimum funkcije E sa kružnim konturama u njima. Očigledno, ℓ_2 regularizacija će proizvesti proređeni model samo ukoliko je to baš optimalan model. U svim drugim slučajevima koeficijenti će biti nenula.

Još jedan pogled koji doprinosi razumevanju zašto ℓ_1 regularizacija vodi proređenim modelima, a ℓ_2 ne, je vezan za gradjente ovih normi. Optimiza-



Slika 7.2: Regioni (crveno) u kojima će ℓ_1 i ℓ_2 regularizacija proizvesti model za koji je $w_1 = 0$.

cioni metodi obično počivaju na uzastopnom umanjivanju tekućih parametara modela za neki vektor proporcionalan gradijentu ciljne funkcije. U slučaju ℓ_1 regularizacije, doprinos regularizacionog izraza gradijentu ciljne funkcije je

$$\frac{\partial \|\mathbf{w}\|_1}{\partial w_i} = \text{sgn}(w_i)$$

kad god važi $w_i \neq 0$. U slučaju ℓ_2 regularizacije važi

$$\frac{\partial \|\mathbf{w}\|_2^2}{\partial w_i} = 2w_i$$

Očito, kolika god da je vrednost parametra w_i , ℓ_1 regularizacija podjednako doprinosi smanjenju apsolutne vrednosti koeficijenta w_i . S druge strane, ℓ_1 regularizacija doprinosi utoliko manje, što je vrednost parametra manja, tako da kako se vrednost bliži nuli, regularizacija sve manje pomaže u daljem smanjenju.

Očigledan problem vezan za ℓ_1 regularizaciju je njena nediferencijabilnost. Nediferencijabilnost se u mašinskom učenju često ignoriše, pošto u nekim problemima nije mnogo verovatno da će optimizacioni algoritam zasnovan na gradijentima naleteti na tačku u kojoj je funkcija nediferencijabilna. Čak i ako naleti i preduzme korak u pogrešnom pravcu, taj korak će u mnogim slučajevima biti kompenzovan daljim tokom optimizacije. Ipak, u slučaju ℓ_1 regularizacije, ovaj problem se ne može ignorisati. Naime, tačka u kojoj je funkcija nediferencijabilna je baš tačka rešenja kojoj se teži, a ne neka tačka na koju optimizacija može, a ne mora nabasati, a u koju se ne mora vraćati. Otud se uz ovu vrstu regularizacije koriste i posebni optimizacioni algoritmi.

Još jedna mala laso regularizacija je ta što vodi nestabilnim rešenjima. Naime, ukoliko postoje dva jako korelirana atributa, laso regularizacija će verovatno iz modela isključiti jedan od tih atributa. Kako su atributi jako korelirani, nije velika razlika da li će biti izbačen jedan ili drugi i lako se može desiti da male promene u podacima vode isključivanju različitih atributa. Ovo očito predstavlja problem za interpretaciju modela. Jedno rešenje ovog problema je

korišćenje *elastične mreže* (eng. *elastic net*), što je regularizacija kojoj odgovara izraz

$$\Omega(w) = \mu\|w\|_1 + (1 - \mu)\|w\|_2^2$$

pri čemu važi $\mu \in [0, 1]$.

7.2 Modeli složenije strukture i uključivanje domenskog znanja

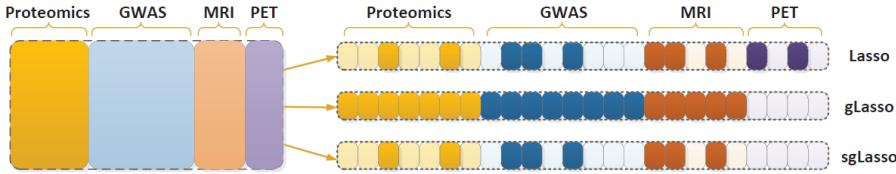
Neka je potrebno predvideti prinos pšenice na jednom podneblju na osnovu nekih atributa, a poznat je model na nekom drugom podneblju. Modeli verovatno neće biti isti, ali mogu biti vrlo slični. Posebno ukoliko je količina podataka na kojima je potrebno odrediti model mala, oslanjanje na već poznati model može biti vrlo korisno. Ovo predstavlja vid uključivanja prethodnog znanja u proces učenja. Dodatno, ne mora ni biti poznat takav model. Moguće je da ekspert za dati domen ima prepostavke o tome kako neki atribut utiče na ciljnu promenljivu. Ukoliko bi mogao ugrubo da kvantifikuje taj uticaj i taka informacija bi mogla biti korisnija nego nikakva. Postavlja se pitanje kako ukljuciti takvo domensko znanje u model. Jedan jednostavan način je sledeći.

Neka G označava grupu atributa i neka w_G označava vektor vrednosti parametara koji odgovaraju tim atributima. Ukoliko je za grupu nekih atributa G ugrubo poznato kako utiču na ciljnu promenljivu, proces učenja može biti unapređen uključivanjem tog domenskog znanja. Neka je w'_G vektor pretpostavljenih vrednosti parametara za attribute iz grupe G . Tada se te vrednosti mogu upotrebiti za konstrukciju sledećeg regularizacionog izraza:

$$\Omega(w) = \|w_G - w'_G\|_2^2$$

Pritom, moguće je u ciljnu funkciju dodati i druge vrste regularizacije zajedno sa ovom. Očigledan je problem kvaliteta pretpostavljenih vrednosti w'_G , ali to nije suštinski problem. Ukoliko je pretpostavka loša, prilikom izbora najbolje konfiguracije biće izabrana vrednost koeficijent λ koja je mala ili jednaka nuli. To je istovremeno i indikacija kvaliteta same pretpostavke ili, u kontekstu problema predviđanja prinosa pšenice, sličnosti zakonitosti između ciljne promenljive i atributa na dva različita podneblja.

Jedna vrlo korisna vrsta regularizacije je *grupna lasso regularizacija* (eng. *group lasso*). Vrlo često, atributi sa mogu grupisati prema nekom kriterijumu. U medicinskim primenama, to recimo mogu biti merenja na osnovu uzorka krvi, rendgenskog snimka, snimanja magnetnom rezonanciom, biopsije i slično. Neke od ovih analiza su neprijatne, neke skupe, na neke se može dugo čekati i slično. Već je istaknuto da je kvalitet proređenih modela što omogućavaju da se merenja koja odgovaraju određenim atributima ne vrše. Ipak, zamislimo da ℓ_1 regularizacija pridruži koeficijent 0 merenju hemoglobina, ali nenula koeficijent merenju triglicerida u krvi. To što nije potrebno meriti hemoglobin, ipak ne znači mnogo, pošto je svejedno potrebno da pacijent da krv i da se izvrše neke



Slika 7.3: Ilustracija dejstva obične laso regularizacije (desno gore), grupne laso regularizacije (desno u sredini) u skladu sa definisanim grupama (levo) i proredene grupne laso regularizacije (desno dole).

analize. Suštinski dobitak bi bio da nijednu od analiza krvni nije potrebno raditi. U suprotnom, u ovom slučaju, mogu se uraditi sve. Slično, ukoliko se uradi snimanje magnetnom rezonancom, nebitno je da li se računaju neki atributi snimka ili svi. Grupna laso regularizacija rešava ovaj problem. Neka je dat skup grupa $\{G_1, \dots, G_M\}$ atributa koje mogu, ali ne moraju biti disjunktnе. Grupna laso regularizacija se vrši upotreбom regularizacionog izraza

$$\Omega(w) = \sum_{i=1}^M \|w_{G_i}\|_2$$

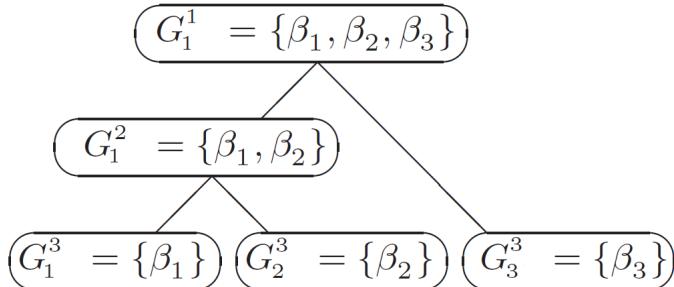
Primetimo da norma nije kvadrirana. Ovakva regularizacija teži tome da anulira norme pojedinačnih grupa, a anuliranjem normi grupa anuliraju se svi koeficijenti u grupi. Pored anuliranja celih grupa, i dalje može biti poželjno da se i u okviru relevantnih grupa model što više proredi. U tom slučaju moguće je koristiti kombinaciju grupne i obične laso regularizacije:

$$\Omega(w) = \mu\|w\|_1 + (1 - \mu) \sum_{i=1}^M \|w_{G_i}\|_2$$

za neko $\mu \in [0, 1]$. Ovaj vid grupne laso regularizacije naziva se *proređena grupna laso* (eng. *sparse group lasso*) regulizacija. Slika 7.3 ilustruje ove vidove regularizacije.

Još jedan kontekst u kojem je grupna laso regularizacija vrlo korisna je postojanje kategoričkih atributa. Kao što je već rečeno kategorički atribut sa C mogućih vrednosti se najčešće uključuje u model tako što se kodira pomoću $C - 1$ binarnih promenljivih. Isključivanje samo nekih od tih binarnih promenljivih deluje nepoželjno sa tačke gledišta interpretabilnosti. Ukoliko je deo binarnih promenljivih uključen u model, a deo nije, teže je zaključiti da li je polazna kategorička promenljiva bitna. S druge strane, korišćenje grupne laso regularizacije, vodiće tome da su sve binarne promenljive uključene u model ili da su sve isključene iz njega.

Još jedan vid regularizacije koja daje specifičnu strukturu modela, a i uključivanje domenskog znanja je *hijerarhijska laso regularizacija* (eng. *tree group lasso*), pri čemu prepostavljamo da je hijerarhija data stablom u čijim



Slika 7.4: Organizacija grupa atributa u vidu stabla.

su listovima (i samo u listovima) atributi. Ovakve hijerarhije nisu neuobičajene. Recimo, medicinske dijagnoze se prirodno organizuju u stabla, pri čemu u korenu stabla može biti najopštija dijagnoza *bolest*, na nešto nižim nivoima mogu biti recimo grupe dijagnoza *bolest pluća*, *bolest srca* i tako dalje, dok u listovima mogu biti konkretnе dijagnoze poput *grip*, *upala pluća*, *artritis* i tako dalje. U ovom slučaju regularizacioni izraz ima istu formu kao i izraz za grupnu laso regularizaciju. Hijerarhijski efekat se postiže tako što svakom čvoru stabla odgovara grupa svih atributa koji su u listovima potomcima datog čvora, kao što je prikazano slikom 7.4. Na taj način, ukoliko se regularizacijom iz modela eliminuju plućne bolesti, automatski su eliminisane i sve pojedinačne plućne bolesti.

7.3 Učenje više poslova odjednom

Kao što je već diskutovano u prethodnom odeljku, nekada se nedovoljna količina podataka može nadomestiti jačim pretpostavkama o modelu. U primeru vezanom za prinos pšenice, pretpostavka je bila vezana za sličnost sa modelom sa nekog drugog podneblja. Scenario koji će biti diskutovan u nastavku možda ne deluje srođno, ali se zapravo oslanja na slične principe.

Nekada podaci na kojima je potrebno uraditi obučavanje mogu biti podeљeni na nekoliko grupa koje uprkos istim atributima i ciljnoj promenljivoj mogu dolaziti iz nešto različitih raspodela. Na primer, problemi predviđanja roda pšenice u Srbiji, Španiji i Australiji sigurno imaju nešto zajedničko, čak toliko da se mogu smatrati jednim problemom, ali postoje i razlike usled značajno različitih klimatskih, zemljишnih i drugih svojstava ta tri podneblja. Slično, predviđanje budućeg stanja pacijenata (na primer, da li će lečenje uspeti ili će pacijentima ponovo biti potreban tretman) u različitim bolnicama, čak i ako su bolnice iste specijalnosti, ima sličnosti, ali može biti i razlika. Na primer, primena pravih procedura će sigurno voditi boljim rezultatima kod pacijenata iz svih bolnica, ali će pacijenti iz bolnice u rudarskom gradu verovatno imati sporiji oporvak, nego pacijenti iz bolnice u centru velikog grada. Razlog

za to je prosto što pacijenti iz prve bolnice u proseku verovatno imaju nepovoljnije životne uslove, što se održava i na njihovo zdravstveno stanje. Kako modelujemo ovakve probleme? Ukoliko je količina podataka velika, verovatno ima smisla napraviti odvojene modele za svaki od potproblema. Ukoliko bismo spojili sve podatke i napravili jedan model, moguće je da bi se različite zakonitosti koje važe u različitim potproblemima uprosećile i da bi ukupni model izgubio na tačnosti. Ipak, često podataka nema dovoljno. Često je raspodela broja podataka neravnomerna. Neki potproblemi mogu imati veliku količinu podataka, a neki malu. Ukoliko bi se svi podaci spojili, specifičnosti potproblema sa malom količinom podataka bi verovatno bile zanemarene u korist specifičnosti potproblema za koje su dostupne velike količine podatka. S druge strane, učenje ne bi bilo uspešno ni u slučaju odvojenog obučavanja, pošto učenje na malom skupu podataka lako može voditi nepouzdanom modelu. Ovo su tipična pitanja *učenja više poslova odjednom* (eng. *multitask learning*). Postoje različite postavke ovakve vrste problema, a i različiti pristupi njihovog rešavanja. Otud narednu formulaciju i diskusiju koja je prati ne treba smatrati najopštijim slučajem.

Pretpostavimo da se skup podataka \mathcal{D} može predstaviti kao unija disjunktivnih skupova \mathcal{D}_i za $i = 1, \dots, T$, gde je T broj različitih, ali srodnih poslova. Svi podaci imaju iste atribute i istu ciljnu promenljivu. Neka je W matrica dimenzija $n \times T$, čije su kolone vektori W_1, \dots, W_T . Jedan jednostavan način zajedničkog obučavanja modela za sve poslove je sledeći:

$$\min_W \sum_{i=1}^T E(W_i, \mathcal{D}_i) + \lambda \sum_{i=1}^T \|W_i - \bar{W}\|_2^2$$

pri čemu važi

$$\bar{W} = \frac{1}{T} \sum_{i=1}^T W_i$$

Drugim rečima, potrebno je minimizovati grešku pri specifičnoj regularizaciji, koja sugerije da nijedan od modela ne treba daleko da odstupi od proseka svih modela. Može se pokazati da je ovo ekvivalentno formulaciji

$$\min_W \sum_{i=1}^T E(W_i, \mathcal{D}_i) + \lambda \sum_{i=1}^{T-1} \sum_{j=i+1}^T \|W_i - W_j\|_2^2$$

u kojoj regularizacija sugerije da svaki model treba da bude blizak svakom drugom. Na ovaj način, vrši se nagodba između minimizacije greške i sličnosti modela, odnosno, svaki model može odstupiti od drugih, ali samo ako to značajno doprinosi smenjenju greške. Na taj način, model koji odgovara poslu sa manjom količinom podataka, može se osloniti na modele sa većom količinom podataka, ali u meri u kojoj mu to potreba za smanjenjem greške dopušta. Kao i obično, relativni značaj ovih faktora se vaga izborom vrednosti parametra λ .

Prethodni model pretpostavlja da su naša apriorna uverenja o sličnosti među različitim poslovima jednaka. Ipak, nekada možemo smatrati da su neka

dva posla sličnija od neka druga dva. U opštem slučaju, moguće je konstruisati graf zavisnosti među poslovima i sličnost poslova i i j kvantifikovati težinama grana grafa α_{ij} i rešiti sledeći problem:

$$\min_W \sum_{i=1}^T E(W_i, \mathcal{D}_i) + \lambda \sum_{i=1}^{T-1} \sum_{j=i+1}^T \alpha_{ij} \|W_i - W_j\|_2^2$$

Nekada se želi da dobijeni modeli budu proređeni. Tada se mogu dodati ℓ_1 regularizacije za svaki od modela. U takvom slučaju bi bilo korisni da svi modeli budu proređeni na isti način. Odnosno, da ako kod jednog pacijenta ne vršimo neku analizu, to ne radimo ni kod drugog. Ovaj problem je već malo komplikovaniji. Neka je $\ell_{p,q}$ norma matrice definisana na sledeći način

$$\|W\|_{p,q} = \left(\sum_{i=1}^n \|W^i\|_p^q \right)^{\frac{1}{q}}$$

gde W^i označava i -tu vrstu matrice W . Tada se rešavanjem sledećeg optimizacionog problema dobija željeno svojstvo jednake proređenosti više modela istovremeno:

$$\min_W \sum_{i=1}^T E(W_i, \mathcal{D}_i) + \lambda \sum_{i=1}^T \|w_i - \bar{w}\|_2^2 + \rho \|W\|_{2,1}$$

Naime, time što će cela ℓ_2 norma vrste W^i biti svedena na nulu, nijedan model neće uključivati merenja odgovarajućeg atributa. Ova regularizacija može pomoći i da se model multinomijalne logističke regresije učini interpretabilnijim.

Glava 8

Optimizacija

Matematička optimizacija se bavi metodama pronalaženja minimuma i maksimuma funkcija. Opšti *problem optimizacije* je obično oblika:

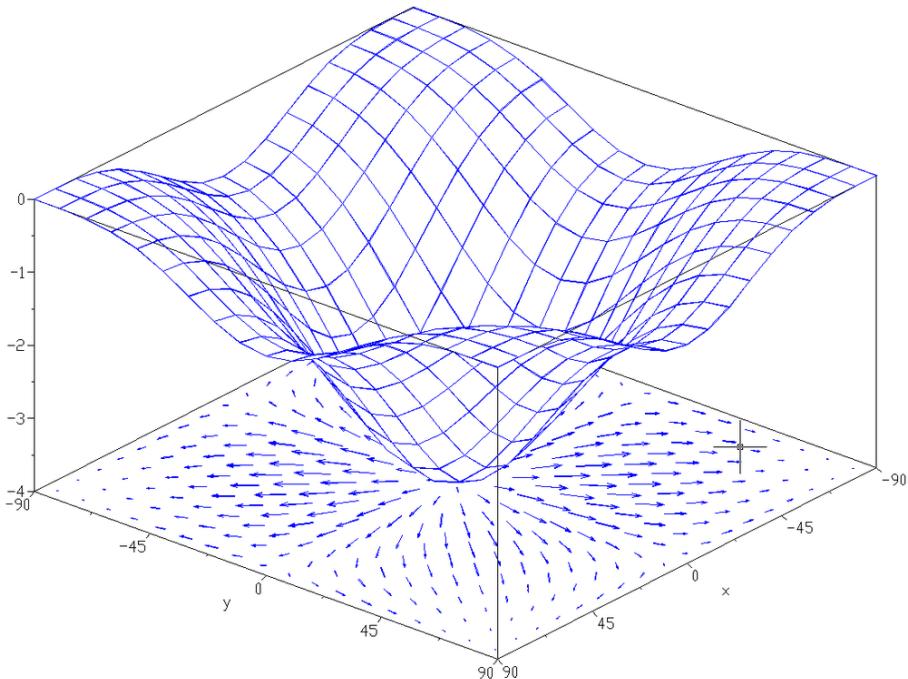
$$\begin{aligned} & \min_{x \in \mathcal{D}} f(x) \\ \text{pri uslovima } & g_i(x) \leq 0 \quad i = 1, \dots, L \end{aligned}$$

pri čemu se funkcija f naziva *ciljnom funkcijom*, skup \mathcal{D} domenom, a uslovi vezani za g_i , takođe i *ograničenjima*. Objekat iz domena koji zadovoljava sva ograničenja, naziva se *dopustivo rešenje*. Potrebno je među svim dopustivim rešenjima naći ono za koje je vrednost ciljne funkcije najmanja. Ova formulacija obuhvata i pronalaženje maksimuma, pošto se pronalaženje maksimuma funkcije f može svesti na pronalaženje minimuma funkcije $-f$. Zato će u nastavku biti reči isključivo o metodama pronalaženja minimuma, odnosno *minimizacije*. Takođe, treba primetiti da se i jednakosna ograničenja lako uklapaju u navedeni okvir. Naime, ograničenje $g(x) = 0$ se može predstaviti pomoću dva ograničenja: $g(x) \leq 0$ i $-g(x) \leq 0$.

U mašinskom učenju se koriste najrazličitije varijante ove opšte forme. Ipak, najčešće se sreću problemi bez ograničenja ili problemi koji se lako mogu transformisati u takve probleme, pa će u nastavku biti diskutovani algoritmi za optimizaciju tog tipa.

8.1 Gradijentni spust

Najjednostavnija i najpoznatija metoda optimizacije prvog reda za diferencijabilne funkcije je *gradijentni spust* (eng. gradient descent). Ova metoda, kao i većina metoda optimizacije, zasniva se na postepenom, iterativnom, približavanju rešenju problema. Gradijent ukazuje na pravac najbržeg uspona, što je ilustrovano slikom 8.1. Stoga, negativna vrednost gradijenta ukazuje na pravac najbržeg spusta. Osnovna ideja gradijentnog spusta je da se, polazeći



Slika 8.1: Gradijenti funkcije u različitim tačkama

od neke nasumice izabrane tačke, nizom koraka u pravcu gradijenta dođe vrlo blizu rešenju. Ako je polazna tačka x_0 , svaka naredna se dobija primenom pravila

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

U vezi sa ovakvim pristupom, postavlja se više pitanja. Prvo je kako se bira dužina koraka α_k koji se preduzima u pravcu suprotnom gradijentu. Postoje različiti pristupi. Jedan, jednostavan izbor je korišćenje konstantne vrednosti koraka $\alpha_k = \alpha$, za neko α , za svako i . Drugi pristup je oslanjanje na Robins-Monroove uslove

$$\sum_{i=0}^{\infty} \alpha_k = \infty \quad \sum_{i=1}^{\infty} \alpha_k^2 < \infty$$

Intuitivno, smisao prvog uslova je da su koraci dovoljno veliki da se može dostići rešenje problema. Smisao drugog uslova je da su koraci dovoljno mali da niz tačaka x_k konvergira rešenju, umesto da osciluje. Jedan od izbora koji zadovoljava ove uslove je $\alpha_k = \frac{1}{k}$. Pored ovih pristupa, postoje i drugi. Drugo pitanje je kada se staje sa izračunavanjem. Kriterijuma zaustavljanja koji se u praksi koriste ima više. Najčešći su zaustavljanje nakon unapred zadatog broja iteracija, nakon što razlika između susednih koraka $\|x_{k+1} - x_k\|$ postane manja od unapred zadate vrednosti ε , nakon što razlika između vrednosti funkcije

u susednim koracima $|f(x_{k+1}) - f(x_k)|$ postane manja od ε ili nakon što ovo razlika u odnosu na polaznu vrednost funkcije $|f(x_{k+1}) - f(x_k)|/|f(x_0)|$ postane manja od ε . Moguće je kombinovati i više ovakvih kriterijuma.

U slučaju konstantnog koraka, moguće je dokazati konvergenciju metoda ka pravom rešenju, ali tek sa određenom nesavladivom greškom, koja je utoliko veća ukoliko je veličina koraka veća. U slučaju oslanjanja na Robins-Monroove uslove, za konveksne funkcije sa Lipšic neprekidnim gradijentom, greška metode $\|x_k - x^*\|$ u koraku k , gde je x^* tačka minimuma, je reda $O(\frac{1}{k})$, što očito implicira konvergenciju. Za jako konveksne funkcije sa Lipšic neprekidnim gradijentom, greška je reda $O(c^k)$ za neko $0 < c < 1$. U slučaju nekonveksnih funkcija, gradijentni spust i njegove varijante prikazane u nastavku konvergiraju, ali navedene brzine konvergencije ne važe. Konvergencija gradijentnog spusta se smatra relativno sporom. Razmotrimo realističnost jake konveksnosti.

Primer 5 *Funkcija greške u problemu linearne regresije $\|Xw - y\|_2^2$ je konveksna. Naime,*

$$(Xw - y)^T(Xw - y) = w^T X^T X w - w^T X^T y - y^T X w + y^T y$$

Matrice oblika $X^T X$ su uvek pozitivno semidefinitne. Pošto je matrica $X^T X$ hesijan funkcije $w^T X^T X w$, onda je ona konveksna funkcija. Ostale funkcije su linearne, pa otud i konveksne. Zbir konveksnih funkcija je konveksna funkcija.

Srednja reška u problemu regularizovane linearne regresije

$$\|Xw - y\|_2^2 + \lambda \|w\|_2^2$$

je jako konveksna. Naime,

$$(Xw - y)^T(Xw - y) + \lambda w^T w = w^T X^T X w - w^T X^T y - y^T X w + y^T y + \lambda w^T w$$

Hesijan ove funkcije je

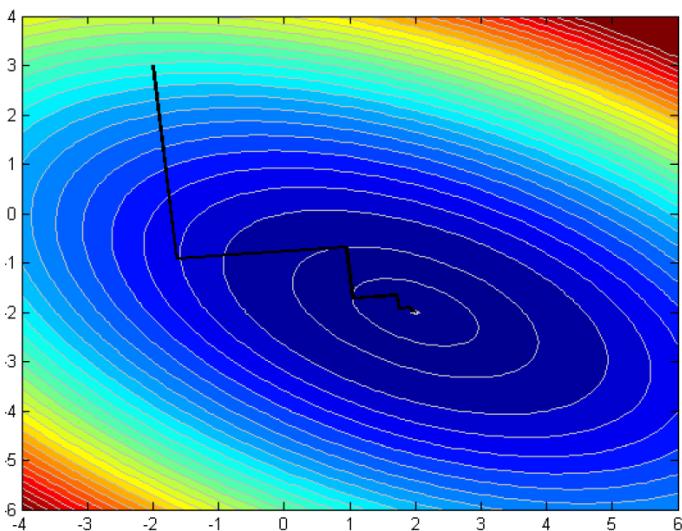
$$X^T X + \lambda I$$

Znamo da je funkcija f jako konveksna ukoliko je matrica $\nabla^2 f(x) - mI$ pozitivno semidefinitna. Za $m = \lambda$, dobija se

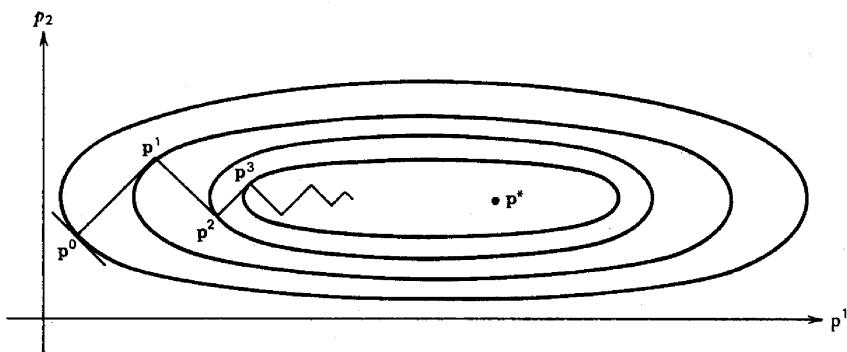
$$X^T X + \lambda I - \lambda I = X^T X$$

što je pozitivno semidefinitna matrica. Stoga, polazna funkcija mora biti jako konveksna. Otud se zaključuje da ℓ_2 regularizacija ne utiče pozitivno samo na prediktivne performanse dobijenog modela, već i na brzinu optimizacije.

Poznato je da je gradijent u svakoj tački normalan na konturu (poput izohipse na geografskoj karti) funkcije sa istom vrednošću koju funkcija ima u toj tački. Ovo ponašanje je ilustrovano slikom 8.2. Imajući ovo u vidu, ne čudi da se gradijentni spust ne ponaša dobro u slučajevima funkcija čije su konture izdužene, kao na slici 8.3. U takvim situacijama, gradijentni spust bira tačke



Slika 8.2: Pravac gradijenta u nekoj tački je normalan na odgovarajuću konturu funkcije.



Slika 8.3: Ponašanje gradijentnog spusta u slučaju funkcije sa izduženim konturnama.

koje leže duž cik-cak putanje ka minimumu i broj koraka do zadovoljavajućeg rešenja može biti veliki. Očito, pravac najbržeg uspona uopšte ne mora biti pravac najbržeg kretanja ka minimumu.

U sumi, prednosti metode gradijentnog spusta su njena jednostavnost i široki uslovi primenljivosti, a mane su spora konvergencija, to što je izabrani pravac samo lokalno optimalan, što dodatno usporava konvergenciju cik-cak kretanjem i to što se u mnogim slučajevima za izračunavanje tog neoptimalnog pravca troši puno vremena.

8.2 Metod inercije

Kao što je rečeno, pri gradijentnom spustu gradijent u nekim situacijama naglo menja pravac, što dovodi do cik-cak kretanja i sporije konvergencije. *Metod intercije* se zasniva na ideji akumuliranja prethodnih gradijenata, pri čemu je značaj starijih gradijenata manji, a novijih veći, a onda se umesto gradijenta u dатој тачки koristi ukupan akumulirani gradijent. Kako prosek nekih vrednosti, manje varira nego same vrednosti, ovakva tehnika dovodi do manjih promena pravca u gradijentu i često do povećanja brzine konvergencije. Metod intercije je definisan na sledeći način:

$$d_0 = 0$$

$$d_{k+1} = \beta_k d_k + \alpha_k \nabla f(x_k)$$

$$x_{k+1} = x_k - d_{k+1}$$

pri čemu važi $0 \leq \beta < 1$. U vektoru d_k se akumuliraju gradijenti prvih k koraka. Pritom, kako se d_k u svakoj iteraciji množi brojem manjim od 1, uticaj ranijih gradijenata eksponencijalno brzo opada, tako da skoriji gradijenti dosta više utiču na pravac koraka. Ovaj metod se često koristi za obučavanje neuronskih mreža.

8.3 Nestorovljev ubrzani gradijentni spust

Nesterovljev ubrzani gradijentni spust je modifikacija metoda intercije, koja predstavlja asimptotski optimalan algoritam prvog reda za konveksne funkcije. Ukoliko je funkcija konvksna sa Lipšić neprekidnim gradijentom, greška je reda $O\left(\frac{1}{k^2}\right)$, naspram $O\left(\frac{1}{k}\right)$ u slučaju običnog gradijentnog spusta, pod istim uslovima.

$$d_0 = 0$$

$$d_{k+1} = \beta_k d_k + \alpha_k \nabla f(x_k - \beta_k d_k)$$

$$x_{k+1} = x_k - d_{k+1}$$

Algoritam definiše specifičan izbor vrednosti α_k i β_k , ali o njemu neće biti reči. Ovaj algoritam je posebno pogodan u slučaju podataka visoke dimenzionalnosti. Naime, u tom slučaju je teško primeniti metode drugog reda, zbog toga što veličina hesijana može biti ogromna. Zbog svoje brzine, nestorovljev algoritam je tada najbolja alternativa. I on se često koristi u obučavanju neuronskih mreža.

8.4 Adam

Najkorišćeniji algoritam za obučavanje neuronskih mreža je Adam (eng. adaptive moment estimation). Zasniva se na ocenama prvog i drugog momenta

gradijenata koje su date narednim formulama:

$$m_0 = 0$$

$$v_0 = 0$$

$$m_{k+1} = \beta_1 m_k + (1 - \beta_1) \nabla f(x_k)$$

$$v_{k+1} = \beta_2 v_k + (1 - \beta_2) \nabla f(x_k) \odot \nabla f(x_k)$$

Očito, ocena prvog momenta m_{k+1} predstavlja akumulirani pravac kretanja, kao što je bio slučaj i sa prethodnim algoritmima. Ocena drugog momenta slično akumulira kvadrat norme gradijenta. Kako ove procene kreću od nule, što je proizvoljna odluka, pristrasne su ka nuli i potrebno ih je korigovati:

$$\hat{m}_{k+1} = m_{k+1} / (1 - \beta_1^{k+1})$$

$$\hat{v}_{k+1} = v_{k+1} / (1 - \beta_2^{k+1})$$

Ažuriranje parametara vrši se na sledeći način:

$$x_{k+1} = x_k - \alpha_{k+1} \frac{\hat{m}_{k+1}}{\sqrt{\hat{v}_{k+1}} + \varepsilon}$$

pri čemu sabiranje vektora $\hat{v}_k + 1$ i skalara ε predstavlja dodavanje tog skalara na sve koordinate vektora i pri čemu je deljenje pokoordinatno.

Suština algoritma je da dužina koraka koji se preduzima zavisi od svojstava tekućeg regiona u kojem se funkcija optimizuje. Ukoliko je tekući korak preveliči u odnosu na širinu minimuma, tipično dolazi do oscilovanja optimizacionog procesa preko minimuma. To dovodi do čestih promena pravca, usled čega dolazi do poništavanja gradijenata pri njihovoј akumulaciji, pa se prvi moment, a time i korak, smanjuje i optimizacioni proces lakše silazi ka minimumu. U slučaju stabilnog kretanja niz neku nizbrdicu, zbog nedostatka promena pravca, pri oceni prvog momenta nema poništavanja, pa je ta ocena velika, a time i korak gradijenta. Koja je uloga normiranja ocenom drugog momenta? Naime, vrednost prvog momenta nekad nije mala zbog poništavnja usled promena pravca, već prosti zato što je norma gradijenta mala, iako je pravac stabilan. U takvoj situaciji deljenje malom ocenom drugog momenta vodi ubrzavanju kretanja što je uvek poželjno ako nema promena pravca. U slučaju velikih gradijenata, čak i kad se osciluje, rezultujuća vrednost može biti velika. Deljenje velikom ocenom drugog momenta vodi smanjenju koraka i bržem zaustavljanju oscilacija.

Jedno važno svojstvo algoritma je da se zahvaljujući tome što su veličine m_k i v_k vektori i tome što se računske operacije ažuriranja vektora x_k izvode pokoordinatno, svaka njegova koordinata ima zasebnu promenljivu veličinu koraka. Ovo je od izuzetnog značaja za obučavanje neuronskih mreža čije konture mogu biti vrlo izdužene, pa otud jednake dužine koraka po svim dimenzijama nisu odgovarajuće.

8.5 Stohastički gradijentni spust

Jedna, vrlo široko primenjena, modifikacija gradijentnog spusta je *stohastički gradijentni spust* (eng. stochastic gradient descent). Analogna modifikacija se može primeniti i na druge diskutovane metode. Stohastički gradijentni spust se intenzivno primenjuje u obučavanju modela mašinskog učenja sa velikim količinama podataka. Modifikacija se sastoji u tome da je umesto gradijenta dovoljno koristiti neki slučajni vektor čije je očekivanje kolinearno sa gradijentom i istog je smera. Ovakva modifikacija ima smisla pre svega kada se funkcija koja se optimizuje može predstaviti kao prosek drugih jednostavnijih funkcija:

$$f(x) = \frac{1}{N} \sum_{i=1}^N f_i(x)$$

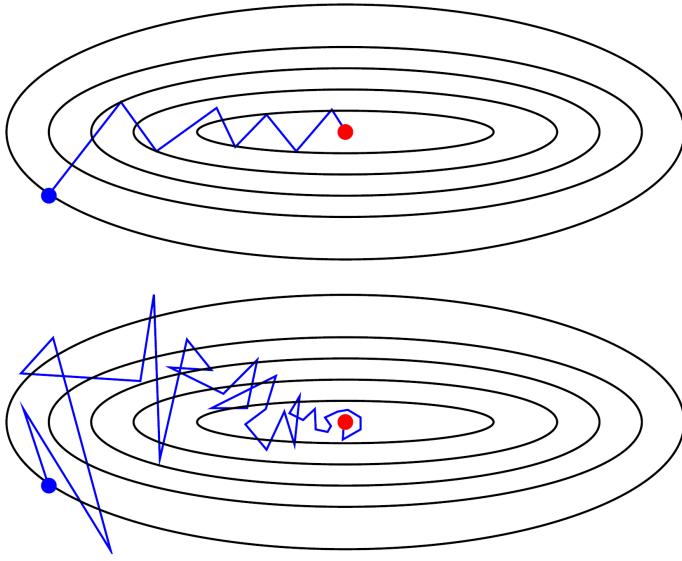
Ovo je tipičan slučaj u kontekstu mašinskog učenja, gde se minimizuje funkcija greške koja je zbir grešaka na pojedinačniminstancama. Tada je pravilo izračunavanja novog koraka moguće zameniti sledećim pravilom:

$$x_{k+1} = x_k - \alpha_k \nabla f_i(x_k)$$

Jasno, kako je funkcija f , prosek funkcija f_i , ako se i bira u skladu sa uniformnom raspodelom, očekivanje slučajnog vektora $\nabla f_i(x)$ je baš $\nabla f(x)$. Obično se i bira tako da bude jednak ($k \bmod N$) + 1, odnosno tako da se u svakom koraku koristi naredna funkcija f_i dok se ne dođe do poslednje, a onda se ponovo nastavlja od prve. Ovaj pristup predstavlja jeftinu aproksimaciju gradijenta. Ipak, ona može biti prilično neprecizna, kao što se može videti sa slike 8.4. Stoga se kao kompromis često, umesto samo jedne od funkcija f_i , koristi prosek nekog podskupa ovih funkcija (eng. *minibatch*). Ovo je praktično uvek pristup koji se koristi u obučavanju neuronskih mreža.

Brzina konvergencije stohastičkog gradijentnog spusta merena *u broju iteracija* je dosta manja nego kod običnog gradijentnog spusta. U slučaju konveksnih funkcija sa Lipšic neprekidnim gradijentom, greška je reda $O\left(\frac{1}{\sqrt{k}}\right)$, a u slučaju jako konveksnih funkcija sa Lipšic neprekidnim gradijentom, greška je reda $O\left(\frac{1}{k}\right)$. Uprkos ovome, u mašinskom učenju, u kojem se danas često koriste ogromne količine podataka, vreme jedne iteracije gradijentnog spusta, koji u svakoj iteraciji koristi sve podatke, je drastično veće nego u slučaju stohastičkog gradijentnog spusta, koji u svakoj iteraciji koristi samo po jednu instancu iz skupa podataka.

U odnosu na gradijentni spust, prednosti stohastičkog gradijentnog spusta su mnogostrukе. Gradijent, koji inače može biti skup za izračunavanje, jeftino se aproksimira. U kontekstu metoda mašinskog učenja nad velikim količinama podataka, to često vodi bržem učenju. Greška aproksimacije gradijenta može poslužiti i kao vid regularizacije, pošto sprečava preciznu konvergenciju ka minimumu, koja u slučaju vrlo fleksibilnih modela ili male količine podataka može voditi ka preprilagođavanju. Manje je podložan problemu redundantnosti podataka prilikom obučavanja. Pod redundantnošću se podrazumeva ponavljanje



Slika 8.4: Ponašanje gradijentnog spusta i stohastičkog gradijentnog spusta.

istih ili sličnih instanci u skupu podataka. Poslednja poenta zahteva opširnije obrazloženje, koje je dato narednim primerom. Mana je očito veći broj iteracija do konvergencije, što u slučaju da korak gradijentnog spusta nije vremenski skup, vodi sporijem zaustavljanju stohastičke varijante u odnosu na izvornu.

Primer 6 Neka se skup podataka sastoji od instanci $\{(x_1, y_1), \dots, (x_N, y_N)\}$. Neka se minimizuje srednjekvadratna greška

$$E(w) = \frac{1}{N} \sum_{i=1}^N (y_i - f_w(x_i))^2$$

Onda je pravilo ažuriranja koeficijenata u skladu sa stohastičkim gradijentnim spustom:

$$w_{k+1} = w_k - 2\alpha_k (y_i - f_{w_k}(x_i)) \nabla f_{w_k}(x_i)$$

dok je u slučaju gradijentnog spusta to

$$w_{k+1} = w_k - 2\alpha_k \frac{1}{N} \sum_{i=1}^N (y_i - f_{w_k}(x_i)) \nabla f_{w_k}(x_i)$$

Ukoliko se ceo skup podataka uveća tako što ponovi za redom M puta u poretku

$$\underbrace{(x_1, y_1), \dots, (x_N, y_N), \dots, (x_1, y_1), \dots, (x_N, y_N)}_{M \times N}$$

pravac koraka u gradijentnom spustu se neće promeniti ni u jednom koraku, pa je stoga i broj koraka u primeni algoritma isti. Kako svaki korak zahteva M puta više vremena, ceo proces M puta duže traje. Stohastički gradijentni spust u ovom slučaju ne zahteva ništa više vremena nego inače, zahvaljujući tome što u svakom koraku koristi samo po jednu instancu, pa korak košta jednako vremena, i što se instance u uvećanom skupu za obučavanje nižu na isti način na koji ih stohastički gradijentni spust i inače smenjuje.

Očigledno, ovo je ekstreman primer redundantnosti podataka, koji se ne očekuje u praksi, ali je ova prednost stohastičkog gradijentnog spusta osetna i u manje ekstremnim slučajevima.

Glava 9

Neuronske mreže i duboko učenje

Neuronske mreže (eng. neural networks) predstavljaju najpopularniju i jednu od najprimjenjenijih metodu mašinskog učenja. Njihove primene su mnogo-brojne i pomeraju domete veštačke inteligencije, računarstva i primenjene matematike. Neke od njih su kategorizacija teksta, medicinska dijagnostika, prepoznavanje objekata na slikama, autonomna vožnja, igranje igara poput igara na tabli (tavla i go) ili video igara, mašinsko prevođenje prirodnih jezika, modelovanje semantike reči prirodnog jezika i slično. Neuronske mreže zapravo predstavljaju parametrizovanu reprezentaciju koja može poslužiti za aproksimaciju drugih funkcija. Kao i u slučaju drugih metoda učenja, pronalaženje odgovarajućih parametara se vrši matematičkom optimizacijom nekog kriterijuma kvaliteta aproksimacije i može biti računski vrlo izazovno.

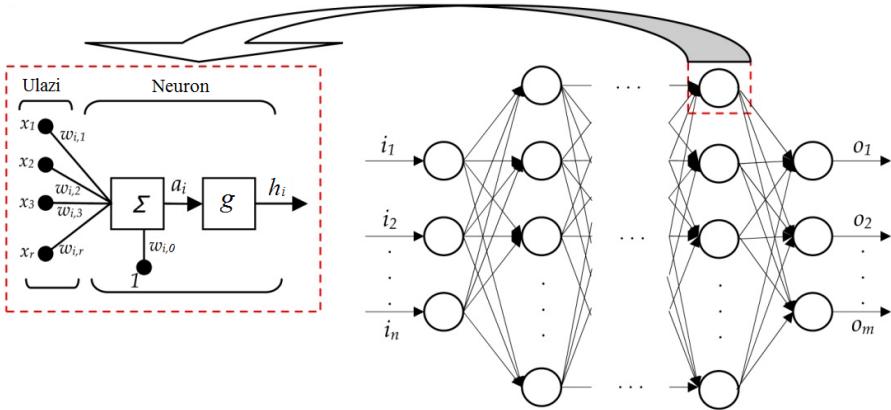
Postoje različite vrste neuronskih mreža. Osnovnu varijantu predstavljaju *neuronske mreže sa propagacijom unapred* (eng. *feed forward neural networks*). U obradi slika i drugih vrsta signala, pa i teksta, vrlo su popularne *konvolutivne neuronske mreže* (eng. *convolutional neural networks*). Za obradu podataka na lik nizovima promenljive dužine, najčešće se koriste *rekurentne neuronske mreže* (eng. *recurrent neural networks*), za obradu podataka koji se mogu predstaviti stablima koriste se *rekuzivne neuronske mreže* (eng. *recursive neural networks*), a za obradu podataka koji se predstavljaju grafovima koriste se *grafovske neuronske mreže* (eng. *graph neural networks*). U nastavku će biti diskutovane prve tri vrste. Poslednje dve su novije od ostalih, pa je i obim njihovih primena manji, ali su vrlo zanimljive zbog svoje izražajne moći.

U svetu njihovih izvanrednih uspeha i velike popularnosti, u laičkim krugovima postoji tendencija poistovećivanja mašinskog učenja, pa čak i veštačke inteligencije sa neuronskim mrežama. Ovakav pogled je prostо pogrešan. Takođe, postoji tendencija da se neuronska mreža razmatra kao prvi izbor metoda učenja nevezano od toga o kom se problemu radi. Ovo bi bio vrlo loš praktičan savet. Stoga, pre nego što predemo na diskusiju neuronskih mreža, naglašavamo u kakvim situacijama su superiorne u odnosu na druge modele. Dok je sasvim moguće da će neuronska mreža preći druge modele i u drugaćijim problemima, problemi zahvaljujući kojima su se neuronske mreže proslavile, imaju

određena zajednička svojstva. To su, velika količina podataka i učenje na osnovu sirove reprezentacije podataka. Male količine podataka u slučaju neuronskih mreža lako vode preprilagodavanju, a učenje nad do sada diskutovanim vektorskim reprezentacijama podataka ne koristi u dovoljnoj meri ključnu prednost neuronskih mreža – da same konstruišu nove atributе nad sirovom reprezentacijom podataka. Naime, iako domenski eksperti nekad mogu pretpostaviti koji su atributi najinformativniji za predviđanje ciljne promenljive, njihovi izbori nekada mogu biti i pogrešni, a neretko lošiji od onoga što bi algoritam učenja mogao da detektuje u sirovoj informaciji ako bi bio primenljiv na nju. Neuronske mreže su karakteristične po tome što postoje njihove varijacije koje su u stanju to da rade. Stoga, ukoliko je skup podataka mali i u vektorskem obliku, nema razloga da očekujemo posebni benefit od primene neuronskih mreža, a moguće je da ćemo imati problema sa njihovim nedostacima. Ukoliko je veliki i u sirovom obliku, verovatno je dobra ideja primeniti neku varijaciju neuronske mreže. Ukoliko je ispunjen samo jedan od ovih uslova, teško je napraviti procenu a priori.

9.1 Neuronske mreže sa propagacijom unapred

Neuronska mreža sa propagacijom unapred (u nastavku samo – neuronska mreža) se sastoji od osnovnih računskih jedinica koje se nazivaju *jedinicama* ili *neuronima*, koje predstavljaju jednostavne parametrizovane funkcije. Svaka jedinica računa linearnu kombinaciju svojih argumenta i nad njom računa neku nelinearnu transformaciju, takozvanu *aktivacionu funkciju* (eng. *activation function*). Ove jedinice su organizovane u slojeve, tako da jedinice jednog sloja primaju kao svoje argumente, odnosno *ulaze*, vrednosti, odnosno *izlaze*, svih jedinica prethodnog sloja i sve jedinice prosleđuju svoje izlaze jedinicama narednog sloja. Svi slojevi čije jedinice prosleđuju svoje izlaze drugim jedinicama se nazivaju *skrivenim slojevima*. Ulazi jedinica prvog sloja se nazivaju ulazima mreže. Izlazi jedinica poslednjeg sloja se nazivaju izlazima mreže. Ukoliko neuronska mreža ima više od jednog skrivenog sloja, naziva se *dubokom neuronskom mrežom* (eng. *deep neural network*). Vrednosti neurona skrivenih slojeva mreže se mogu smatrati novim atributima tih objekata, nad kojima ostatak neuronske mreže uči aproksimaciju ciljne funkcije. Drugim rečima, neuronska mreža konstruiše nove atributе u svojim skrivenim slojevima. Svaki sloj je u stanju da nadograđuje nad prethodnim i tako gradi složenije i složenije atributе. Ovo svojstvo je posebno uočljivo kod konvolutivnih neuronskih mreža i smatra se da je ova mogućnost konstrukcije novih atributa jedan od glavnih razloga za uspešnost dubokih neuronskih mreža.



Slika 9.1: Struktura neuronske mreže sa propagacijom unapred.

9.1.1 Formulacija modela

Formalno, model se definiše na sledeći način:

$$\begin{aligned} h_0 &= x \\ h_i &= g(W_i h_{i-1} + w_{i0}) \quad i = 1, 2, \dots, L \end{aligned}$$

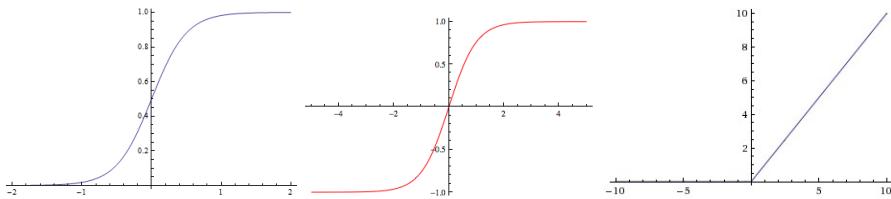
gde je x vektor ulaznih promenljivih, L je broj slojeva, W_i je matrica čija j -ta vrsta predstavlja vektor vrednosti parametara jedinice j u sloju i , w_{i0} predstavlja vektor slobodnih članova linearnih kombinacija koje jedinice i -tog sloja izračunavaju, a g je nelinearna aktivaciona funkcija. Za vektor v , $g(v)$, predstavlja vektor $(g(v_1), g(v_2), \dots, g(v_m))^T$, gde je m dimenzionalnost vektora. Skup svih parametara modela će se ukratko označavati sa w . Važi $f_w(x) = h_L$. Shema neuronske mreže, prikazana je na slici 9.1.

Naredna teorema izražava važno svojstvo neuronskih mreža – da se svaka neprekidna funkcija može proizvoljno dobro aproksimirati neuronskom mrežom sa jednim skrivenim slojem i konačnim brojem neurona.

Teorema 4 (Teorema o univerzalnoj aproksimaciji) *Neka je g ograničena i monotono strogo rastuća neprekidna funkcija. Tada za svaku funkciju $f \in C[0, 1]^n$ i svako $\varepsilon > 0$, postoji broj $m \in \mathbb{N}$, matrica $W \in \mathbb{R}^{m \times n}$, vektor $w_0 \in \mathbb{R}^m$ i vektor $v \in \mathbb{R}^m$, tako da za svako $x \in [0, 1]^n$ važi*

$$|v^T g(Wx + w_0) - f(x)| < \varepsilon$$

Odnosno, skup svih neuronskih mreža sa jednim skrivenim slojem je svuda gust na skupu funkcija neprekidnih na intervalu $[0, 1]^n$. Ova teorema predstavlja osnovno teorijsko opravdanje za korišćenje neuronskih mreža u aproksimaciji funkcija. Ipak, ova teorema samo ustanovljava da postoji neuronska mreža sa datim svojstvima. To nikako ne znači da ju je lako naći.



Slika 9.2: Najčešće korišćene aktivacione funkcije – sigmoidna, tangens hiperbolički i ispravljačka linearna jedinica.

9.1.2 Aktivacione funkcije

U dатој formulaciji modela neuronske mreže nije precizirano koja aktivaciona funkcija se koristi. Moguće je izabrati različite funkcije, ali se u praksi najčešće koristi nekoliko narednih funkcija:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$$

$$rlu(x) = \max(0, x)$$

Grafići sve tri funkcije su prikazani na slici 9.2. Prva, *sigmoidna funkcija* je široko korišćena u mašinskom učenju i dugo je bila najčešće korišćena aktivaciona funkcija u neuronskim mrežama. Ipak, ova funkcija u praksi nije najpogodnija za upotrebu, pre svega zbog problema u optimizaciji. Naime, ona je praktično konstantna osim u okolini nule, pa stoga dovodi praktično do anuliranja gradijenta, što otežava ili praktično onemogućava učenje. Druga, tangens hiperbolički je takođe bila u širokoj upotrebi, obično sa nešto većim uspehom od sigmoidne funkcije sa kojom je vrlo srodnja (važi $\tanh(x) = 2\sigma(2x) - 1$), pre svega zato što je u okolini nule bliska identitetu, što čini model sličnijim linearnom i u nekoj meri olakšava optimizaciju. Treća, *ispravljena linearna jedinica* (eng. *rectified linear unit*) predstavlja trenutno najčešće korišćenu aktivacionu funkciju. Razlog za njenu popularnost su pogodnija svojstva pri optimizaciji, uprkos svojoj nediferencijabilnosti. Naime, šanse da se naleti na tačku nediferencijabilnosti u procesu optimizacije nisu velike, a i ako se to desi, u kasnjem toku optimizacije greška će tipično biti nadomešćena (ovo recimo ne bi važilo za ℓ_1 regularizovane funkcije pošto je tada tačka nediferencijabilnosti baš tačka kojoj se teži). S druge strane, izvod u linearном delu funkcije je konstantno 1, što vodi bržoj konvergenciji nego kad se gradijent smanjuje u nekim delovima domena funkcije. Ipak, ravan deo funkcije levo od nule predstavlja problem za optimizaciju. Naime, instance za koje je vrednost aktivacione funkcije nula, ne doprinose obučavanju, jer se na njima i građenje anulira zahvaljujući konstantnosti funkcije. Stoga se ova aktivaciona funkcija često modifikuje tako da se to izbegne. Jedna modifikovana varijanta je *nakošena ispravljena linearna*

jedinica (eng. *leaky rectified linear unit*), koja levo od nule uzima vrednost αx , za malu vrednost parametra α , poput 0.01.

9.1.3 Izlazne jedinice

Neuronske mreže se mogu koristiti kako za regresiju funkcija sa vrednostima iz \mathbb{R}^n , tako i za klasifikaciju.

U slučaju regresije, jedinice poslednjeg nivoa ne koriste aktivacionu funkciju (kao što je predviđeno i u formulaciji teoreme o univerzalnoj aproksimaciji). Uz pretpostavku da je dat skup parova $\mathcal{D} = \{(x_i, y_i) | i = 1, \dots, N\}$, gde su x_i argumenti, a y_i vrednosti aproksimirane funkcije, aproksimacija se sprovodi rešavanjem narednog optimizacionog problema:

$$\min_w \sum_{i=1}^N (f_w(x_i) - y_i)^2 + \lambda \|w\|_2^2$$

Regularizacija formalno nije neophodna, ali se u praksi uvek koristi neki vid regularizacije (ne nužno ℓ_2), jer su neuronske mreže vrlo fleksibilni modeli, što znači da je problem učenja često loše uslovjen.

U slučaju klasifikacije se na linearne kombinacije jedinica poslednjeg nivoa primenjuje takozvana funkcija *mekog maksimuma* (eng. *softmax*) koja preslikava vektor dimenzije C u vektor iste dimenzije:

$$\text{softmax}(x) = \left(\frac{e^{x_1}}{\sum_{i=1}^C e^{x_i}}, \dots, \frac{e^{x_C}}{\sum_{i=1}^C e^{x_i}} \right)$$

Vrednosti novog vektora se očigledno sumiraju na 1 i stoga se mogu koristiti kao raspodela verovatnoće. Takođe, ova funkcija naglašava razlike među koordinatama polaznog vektora. Najveća pozitivna vrednost će biti transformisana u novu vrednost koja još više odskače od drugih. Za vrednost aproksimacije se uzima kategorija koja odgovara izlazu sa najvišom vrednošću. Minimizacija srednjekvadratne greške nije najbolji pristup ovom problemu, već se kao u slučaju drugih probabilističkih metoda pribegava primeni principa maksimalne verodostojnosti. Potrebno je maksimizovati verovatnoću opaženih vrednosti y_i za date vrednosti x_i . Uz pretpostavku uslovne nezavisnosti vrednosti y_i za date vrednosti x_i , važi:

$$P_w(y_1, \dots, y_N | x_1, \dots, x_N) = \prod_{i=1}^N P_w(y_i | x_i)$$

gde w predstavlja vektor parametara neuronske mreže od kojeg vrednosti na izlazu, koje predstavljaju verovatnoće, očigledno zavise. Za verovatnoću jednog podataka važi

$$P_w(y_i | x_i) = \prod_{j=1}^C \left(\frac{e^{a_j}}{\sum_{k=1}^C e^{a_k}} \right)^{t_{ij}}$$

gde promenljiva t_{ij} ima vrednost 1 ako vrednosti y_i odgovara kategoriji j , a 0 u suprotnom. Kao i pre, umesto maksimizacije date verovatnoće, vrši minimizacije negativne vrednosti logaritma te verovantoće

$$-\log P_w(y_1, \dots, y_N | x_1, \dots, x_N) = -\sum_{i=1}^N \log P_w(y_i | x_i) = -\sum_{i=1}^N \sum_{j=1}^C t_{ij} \log \frac{e^{a_i}}{\sum_{k=1}^C e^{a_k}}$$

U oba slučaja, za rešavanje ovog problema potrebno je koristiti metode matematičke optimizacije.

9.1.4 Algoritam propagacija unazad

Problem optimizacije neuronske mreže je težak zbog svoje nekonveksnosti, što znači da je moguće završiti u lokalnim optimumima ili da neke metode optimizacije nisu lako primenljive ili da sporije rade. U praksi, uvek se koriste gradijentne metode optimizacije. Nekad se koriste i metode drugog reda, koje se oslanjaju na hesijan, ali je to u slučaju većeg broja parametara nemoguće, jer je broj elemenata hesijana kvadratan u odnosu na broj parametara. U slučaju neuronske mreže, već je izračunavanje gradijenta netrivijalan problem i vrši se algoritmom *propagacije unazad* (eng. *backpropagation*) koji će biti opisan u nastavku.

Algoritam propagacije unazad, prikazan na slici 9.3, je jedan od malog broja najznačajnijih algoritama mašinskog učenja. Zasniva se na pravilu izračunavanja parcijalnog izvoda složene funkcije. Za funkcije $g : \mathbb{R}^m \rightarrow \mathbb{R}^n$ i $f : \mathbb{R}^n \rightarrow \mathbb{R}$, parcijalni izvod se računa prema formuli:

$$\partial_i(f \circ g) = \sum_{j=1}^n (\partial_j f \circ g) \partial_i g_j$$

Krećući se po slojevima neuronske mreže od poslednjeg ka prvom, algoritam u svakoj iteraciji obavlja tri posla:

- proširivanje do tada izračunatog parcijalnog izvoda izvodom aktivacione funkcije u skladu sa pravilom za računanje izvoda složene funkcije,
- izračunavanje vrednosti gradijenta po parametrima jedinica na tekućem nivou, zarad čega se do tada izračunati parcijalni izvod množi ulazima jedinica koje ti parametri množe i
- proširivanje do tada izračunatog parcijalnog izvoda izvodom linearne kombinacije po ulazima, u skladu sa pravilom za računanje izvoda složene funkcije.

Trivijalan primer takvog proširivanja parcijalnog izvoda dat je narednim izvođenjem:

$$f(g(h(x)))' = \underbrace{f'(g(h(x)))}_{d} g(h(x))' = \underbrace{f'(g(h(x)))}_{d} \underbrace{g'(h(x))}_{d} h(x)' = \underbrace{f'(g(h(x)))}_{d} \underbrace{g'(h(x))}_{d} \underbrace{h'(x)}_{d}$$

Potpuniji primer rada algoritma, dat je narednim primerom.

```

1  $d = \nabla_{h_L} E$ 
2 repeat
3    $d = d \odot g'(a_k)$ 
4    $\nabla_{w_{k_0}} E(w) = d + \lambda \nabla_{w_{k_0}} \Omega(w)$ 
5    $\nabla_{W_k} E(w) = d h_{k-1}^T + \lambda \nabla_{W_k} \Omega(w)$ 
6    $d = W_k^T d$ 
7    $k = k - 1$ 
8 until  $k = 0;$ 

```

Slika 9.3: Algoritam propagacije unazad. Simbol \odot označava pokoordinatno množenje vektora.

Primer 7 Neka je potrebno naći gradijent funkcije

$$E(w) + \Omega(w)$$

gde je $E(w)$ funkcija odstupanja koju treba minimizovati izračunata za jedan par (x, y) , a $\Omega(w)$ regularizacioni izraz. Recimo, u primeru regresije, važi

$$E(w) = (h_L - y)^2$$

$$\Omega(w) = \|w\|^2$$

gde h_L jasno zavisi od w . Gradijent za ceo skup \mathcal{D} se dobija sumiranjem gradijenata za pojedinačne instance. Izvršavanje algoritma propagacije unazad je prikazano u tabeli 9.1, na primeru izračunavanja gradijenta naredne funkcije:

$$E(w) = (h_2 - y)^2$$

$$f_w(x) = h_2 = \sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$$

Algoritam propagacije unazad nije lako primenljiv na duboke neuronske mreže, zbog velikog broja uzastopnih množenja. Konkretno, zbog toga često dolazi do toga da vrednosti parcijalnih izvoda koje se računaju budu praktično nula (kada se množe brojevi manji od jedan) ili da budu ogromne ili čak da dođe do prekoračenja ili prosto nestabilnosti optimizacionog metoda (kada se množe brojevi veći od jedan). Ovaj problem se naziva problemom nestajućih i eksplodirajućih gradijenata (eng. *exploding and vanishing gradients*). Ovaj problem je blaži ukoliko se kao aktivaciona funkcija koristi nakošena ispravljena linearna jedinica, pošto za pozitivne vrednosti ima vrednost izvoda 1.

9.1.5 Kvaliteti i mane

Kao što je na početku rečeno, neuronske mreže su se izvanredno pokazale u rešavanju praktičnih problema. Za razliku od klasičnih metoda, za koje je

#	Aktivni deo formule	Akumulirano	Izračunato
1	$(\textcolor{red}{h}_2 - y)^2$	$2(h_2 - y)$	
3	$\sigma(w_{20} + \underbrace{w_{21}\sigma(w_{10} + w_{11}x)}_{a_2})$	$2(h_2 - y)\sigma'(a_2)$	
4	$\sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$	$2(h_2 - y)\sigma'(a_2)$	$2(h_2 - y)\sigma'(a_2)$
5	$\sigma(w_{20} + \textcolor{red}{w_{21}}\sigma(\underbrace{w_{10} + w_{11}x}_{a_1}))$	$2(h_2 - y)\sigma'(a_2)$	$2(h_2 - y)\sigma'(a_2)\sigma(a_1)$
6	$\sigma(w_{20} + w_{21}\sigma(\textcolor{red}{w_{10} + w_{11}x}))$	$2w_{21}(h_2 - y)\sigma'(a_2)$	
3	$\sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$	$2w_{21}(h_2 - y)\sigma'(a_2)\sigma'(a_1)$	
4	$\sigma(w_{20} + w_{21}\sigma(\textcolor{red}{w_{10} + w_{11}x}))$	$2w_{21}(h_2 - y)\sigma'(a_2)\sigma'(a_1)$	$2w_{21}(h_2 - y)\sigma'(a_2)\sigma'(a_1)$
5	$\sigma(w_{20} + w_{21}\sigma(w_{10} + \textcolor{red}{w_{11}x}))$	$2w_{21}(h_2 - y)\sigma'(a_2)\sigma'(a_1)$	$2w_{21}2w_{21}(h_2 - y)\sigma'(a_2)\sigma'(a_1)x$
6	$\sigma(w_{20} + w_{21}\sigma(w_{10} + w_{11}x))$	$2w_{11}w_{21}(h_2 - y)\sigma'(a_2)\sigma'(a_1)$	

Tabela 9.1: Ilustracija izvršavanja algoritma propagacije unazad. Kolone predstavljaju redni broj koraka u algoritmu, deo formule po kojem se radi diferenciranje, akumulirani parcijalni izvod i izračunate vrednosti parcijalnih izvoda, za koje se vidi kom parametri odgovaraju po tome koji parametar je istaknut u drugoj koloni.

potrebno pažljivo definisati attribute, neuronske mreže su često u stanju da nelinearnim transformacijama uče korisne attribute i da onda uče nad tim reprezentacijama. Ipak, imaju i značajne mane. Da bi njihovi kvaliteti došli do izražaja potrebna velika količina podataka. Vreme potrebno za optimizaciju mreže može biti vrlo veliko (npr. može se meriti danima) i mogu zahtevati specijalizovani hardver da bi optimizacija bila izvršena u prihvatljivom vremenu. Postoji veliki broj izbora koje je potrebno učiniti pre rešavanja optimizacionog problema, poput broja nivoa mreže, broja jedinica u nivou, izbora vrednosti regularizacionog parametra, izbora algoritma za optimizaciju, itd. Za pravljenje ovih izbora često ne postoje jasne smernice, pa se često određuju empirijski – velikim brojem rešavanja optimizacionog problema dok se ne postignu zadovoljavajući rezultati, što je vremenski vrlo zahtevno. Zbog izrazite fleksibilnosti modela, moguće je da se model preterano prilagodi podacima na kojima je vršena optimizacija i da njegove performanse pri upotrebi budu nezadovoljavajuće. Javljuju se različiti problemi u procesu optimizacije, poput problema nestajućih i eksplodirajućih gradijenata. Zbog svega ovoga upotreba neuronskih mreža zahteva i veliko iskustvo.

9.1.6 Primer primene – prepostavljanje naredne izgovorene reči

Kao primer primene neuronske mreže, poslužio bi bilo koji problem regresije ili klasifikacije u kojem je za svaki vektor podataka x data vrednost y koju treba aproksimirati. Ipak, neki primeri, poput primena u prepoznavanju govora, su posebno zanimljivi.

Primer 8 Prepoznavanje govora, odnosno identifikacije reči koje su izgovorene je izazovan problem iz mnogo razloga, kao što su različite boje različitih

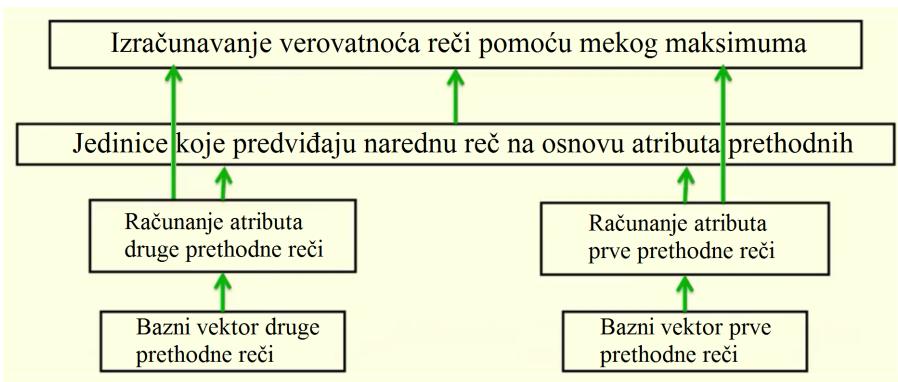
ljudi, različita brzina govora, različit intenzitet glasa, različit akcenat i artikulacija, prisustvo šuma prilikom snimanja, itd. Ljudi nisu svesni težine ovog problema u svakodnevnom govoru, pošto se u velikoj meri oslanjaju na razumevanje konteksta, koje vodi tome da očekuju pojavljivanje nekih reči, pre nego nekih drugih. Analogno tome, sistem za prepoznavanje govora bi mogao značajno poboljšati svoje performanse ukoliko bi mu bila dostupna makar približna informacija o verovatnoćama pojavljivanja reči, recimo na osnovu prethodne dve koje su već identifikovane. Naime, ukoliko sistem koji analizira govor pridružuje približne verovatnoće dvema rečima samo na osnovu njihovog zvučnog zapisa, ali se na osnovu prethodne dve reči može zaključiti da je verovatnoća pojavljivanja jedne od njih nakon prethodne dve približna nuli, to je važan indikator da je izgovorena druga reč.

Predviđanje treće reči na osnovu prve dve u principu je izvodljivo ukoliko je poznat ogroman korpus teksta, iz kojeg se verovatnoće pojavljivanja različitih trojki mogu oceniti na osnovu njihovih frekvencija. Ovo se može uraditi birajući reč koja ima najvišu uslovnu verovatnoću, koja se ocenjuje iz frekvencija f kombinacija reči u korpusu:

$$P(w_3|w_2, w_1) = \frac{P(w_3, w_2, w_1)}{P(w_2, w_1)} \approx \frac{f(w_3, w_2, w_1)}{f(w_2, w_1)}$$

U današnjem dobu, veliki korpusi teksta su dostupni, ali određivanje verovatnoća trojki reči nije pouzdano zbog toga što se mnoge trojke i u velikim korpusima mogu javiti vrlo retko ili ne uopšte, zbog toga što je veličina vokabulara koji se koristi u nekom jeziku vrlo velika. Na primer ako se broj reči koje osoba koristi meri desetinama hiljada, onda se broj svih trojki tih reči meri hiljadama milijardi. Zbog toga bi verovantoće mnogih trojki bile verovatno netačno ocenjene kao izuzetno male ili čak jednake nuli. Osnovna mana ovog pristupa je da tretira reči kao potpuno različite nedeljive celine i ne uočava potencijalne sličnosti među njima, poput recimo sinonimije ili srodnosti po smislu. Na primer, ukoliko čovek na osnovu prethodnog konteksta očekuje da se u nastavku rečenice javi reč novac, lakše će prepoznati i reč pare. Ukoliko očekuje da se u nastavku rečenice javi kućni ljubimac, lakše će prepoznati reči poput psa i mačke. Ukoliko očekuje da se javi dan u nedelji, lakše će prepoznati reči poput ponedeljka i utorka.

Ako se umesto ovog pristupa, prepostavi da je svaka reč okarakterisana numeričkim vektorom atributa određene dimenzije m , koji opisuje njena sintaksna i semantička svojstva, dolazi se do mnogo kompaktnije reprezentacije reči jer broj m može biti drastično manji od ukupnog broja reči. Dodatno, ukoliko se model predviđanja formuliše nad takvim reprezentacijama, on može da nauči da se nakon reči sa nekim svojstvima očekuje reč koja označava kućnog ljubimca i da na osnovu toga pridruži višu verovatnoću rečima pas i mačka koje će imati visoku vrednost atributa kućni ljubimac. Ovakav pristup omogućava i korišćenje dosta dužeg konteksta reči nego što je moguće u pristupu koji se oslanja na same reči. Razlog za to je što se smisao kućnog ljubimca, predstavljen nekom od reči koje označavaju kućne ljubimce, mnogo češće javlja u



Slika 9.4: Shema neuronske mreže koja predviđa narednu reč na osnovu prethodnog konteksta.

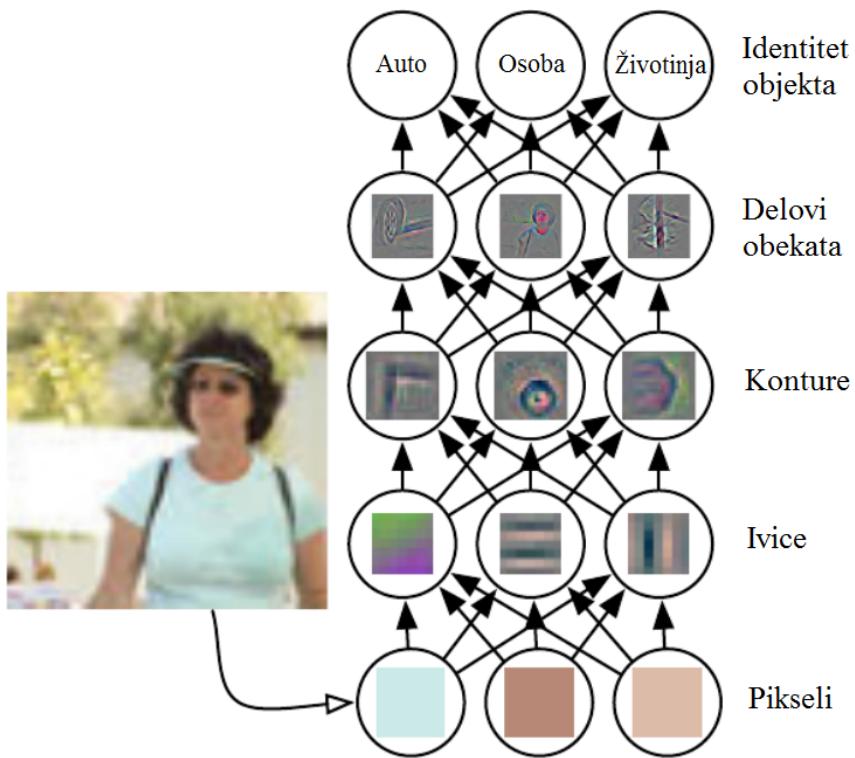
tekstovima od bilo koje pojedinačne reči koja ga označava, pa su i ocene verovatnoća pouzdani. Naravno, postavlja se pitanje, otkud dolaze sintaksna i semantička svojstva reči. Da li ih je potrebno osmisliti unapred i koliko to košta u terminima vremena i novca? Odgovor je da će atributi koji opisuju reči biti definisani u procesu optimizacije, tako što će u tom procesu biti određeni parametri jedinica u skrivenim slojevima mreže, a izlazi tih jedinica će predstavljati vrednosti tih atributa. Vrlo je verovatno da će biti teško ili nemoguće odrediti značenje tako konstruisanih atributa, ali to nije mnogo važno ako je glavni cilj pogoditi narednu reč.

Na slici 9.4, data je shema neuronske mreže koja predviđa sledeću reč na osnovu prethodnog konteksta. Ukoliko je broj reči u vokabularu m , reči se predstavljaju baznim vektorima v_1, v_2, \dots, v_m , prostora \mathbb{R}^m , takvima da je vrednost i -te koordinate vektora v_j , 1 ukoliko je $i = j$, a 0 u suprotnom.

Ovakvi problemi se još bolje rešavaju rekurentnim neuronskim mrežama, koje ne prepostavljaju fiksiranu dužinu konteksta.

9.2 Konvolutivne neuronske mreže

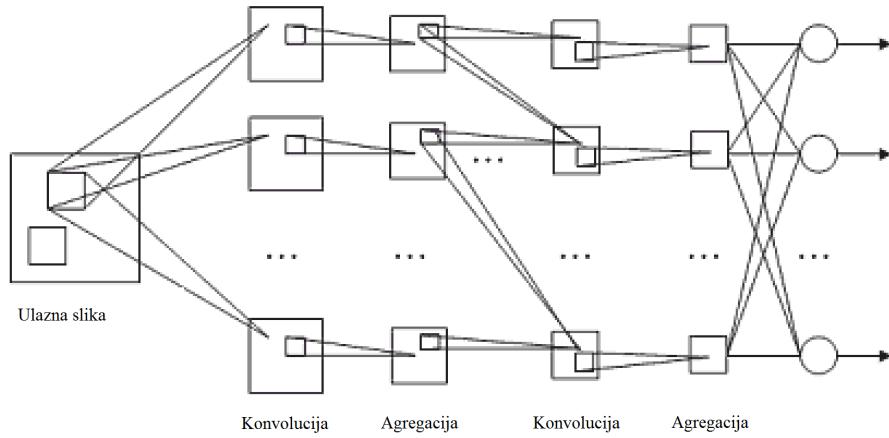
Konvolutivne neuronske mreže se intenzivno koriste u obradi signala poput zvuka i slike, ali takođe i teksta. Zasnivaju se upravo na pomenutoj sposobnosti mreža da konstruišu atributе, ali ne nužno samo iz već datih atributa, već i iz sirovog signala. Nazivaju se konvolutivnim zato što uče filtere, čijom konvolutivnom primenom detektuju određena svojstva signala. U obradi signala su dugo korišćeni filteri dizajnirani od strane inženjera i istraživača (npr. poput filtera za detekciju ivica na slikama). Značaj konvolutivnih mreža je upravo u tome što ne zahtevaju ljudski angažman u definisanju relevantnih svojstava signala, koji verovatno i ne bi proizveo najbolje rezultate, već u zavisnosti od problema koji je potrebno rešiti, same ustanovljavaju koja su svojstva bitna,



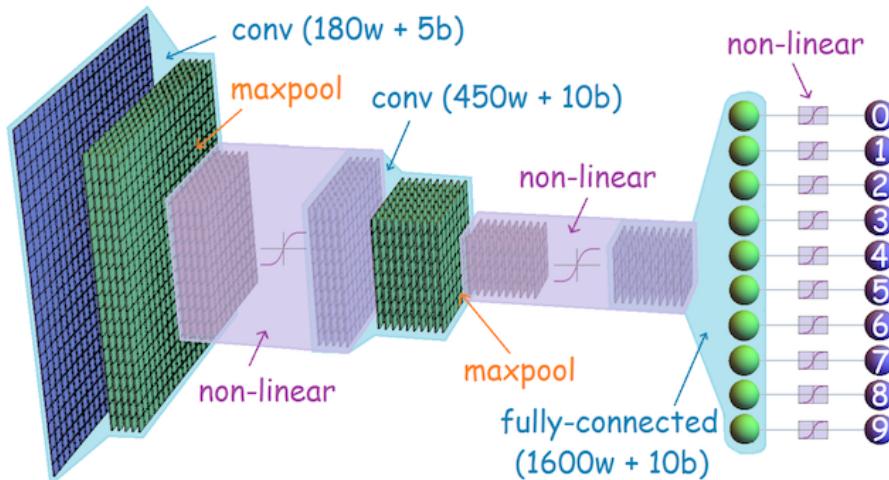
Slika 9.5: Shema konstrukcije složenijih od jednostavnijih atributa u slojevima konvolutivne mreže.

kroz učenje adekvatnih filtera. Ipak, ovakva vrsta fleksibilnosti podrazumeva izazove pri optimizaciji, kao i veliku količinu podataka.

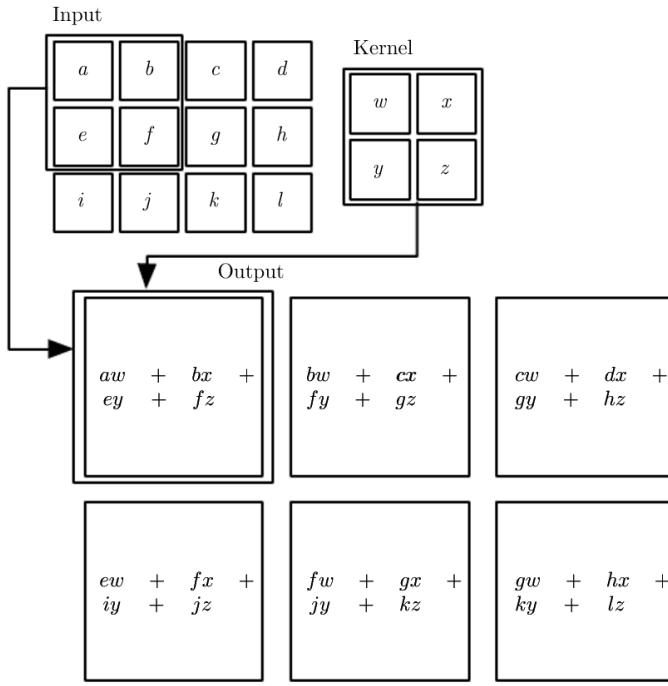
Konvolutivne mreže su praktično uvek duboke neuronske mreže, jer je potrebno od sitnijih detalja, poput uspravnih, kosih i horizontalnih linija, koji obično bivaju detektovani u nižim slojevima mreže, iskonstruisati složenije oblike poput delova lica. Ovo je ilustrovano slikom 9.5. Uobičajena struktura konvolutivne mreže podrazumeva smenjivanje dve vrste slojeva – konvolutivnih slojeva (eng. convolution layer) i slojeva agregacije (eng. pooling layer), kao što je prikazano na slici 9.6, pri čemu je moguće i da se ista vrsta sloja ponovi više puta. Na izlaze poslednjeg od tih slojeva se obično nadovezuje mreža sa propagacijom unapred, koja uči nad atributima koje prethodni slojevi konstruišu. Još jedna ilustracija, koja naglašava manje detalja, ali je nakon inicijalnog razumevanja osnovnih principa, zbog svoje kompaktnosti lakša za vizualizaciju, data je na slici 9.7.



Slika 9.6: Shema konvolutivne mreže.



Slika 9.7: Grublja i kompaktnija shema konvolutivne mreže.



Slika 9.8: Konvolucija ulazne slike sa filterom.

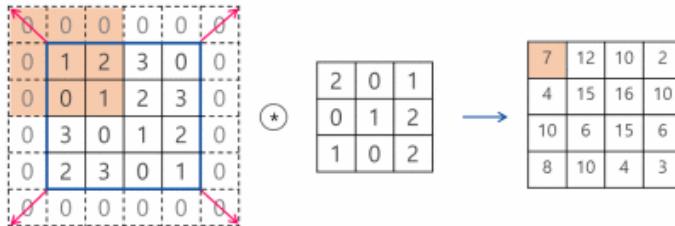
9.2.1 Konvolucija

Neka su f i g dve matrice dimenzija $m \times n$ i $p \times q$. Konvolucija je operacija definisana u diskretnom dvodimenzionalnom slučaju na sledeći način

$$(f * g)_{ij} = \sum_{k=0}^{p-1} \sum_{l=0}^{q-1} f_{i-k, i-l} g_{k, l}$$

Oduzimanje u indeksima izraza $f_{i-k, i-l}$ se u praksi često menja sabiranjem pošto u kontekstu konvolutivnih mreža ne dovodi do razlike. Matrica f je obično ulaz, poput slike, dok je matrica f filter – pomoćna matrica koja izdvaja neku vrstu informacije iz ulaza. Umesto izraza filter, često se koristi i izraz kernel, ali ćemo ga izbegavati jer u ovom kontekstu filter nema svojstva kornela na koja smo navikli. Primer izračunavanja konvolucije dat je na slici 9.8.

Primetimo da formula konvolucije koju smo dali nije definisana za sve indekse $i = 0, \dots, m - 1$ i $j = 0, \dots, n - 1$. Na primer, za $i, j = 0$ i $k, l > 0$, vrednost $f_{i-k, i-l}$ nije definisana. Ukoliko bismo se ograničili na definisane vrednosti, dimenzija konvolucije bi bila manja od dimenzije matrice f , što se vidi i sa slike 9.8. To nije uvek poželjno, i tada se izbegava tako što se vrši proširivanje (eng. padding) matrice f , recimo nulama ili još bolje vrednostima



Slika 9.9: Konvolucija slike proširene nulama sa filterom.

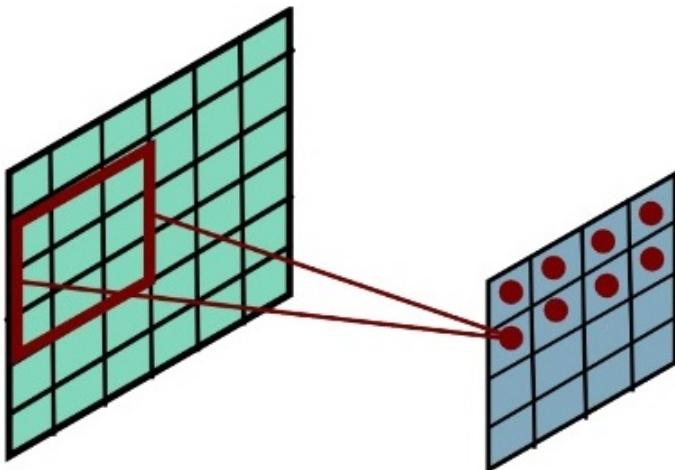
koje su već na obodu, tako da veličina rezultujuće matrice bude jednaka veličini matrice f pre proširivanja. Ovo je prikazano na slici 9.10. Takođe, Prilikom sumiranja, indeksi k i l se ne moraju povećavati za jedan, već i za neki veći korak (eng. *stride*), ali takvim tehničkim detaljima se nećemo dublje baviti.

Konvolutivni sloj ima ulogu konstrukcije novih atributa, poput detekcije nosa, ukoliko je poznato gde su detektovane uspravne, kose i horizontalne linije, što ustanovljava prethodni konvolutivni sloj, recimo na osnovu samih ulaznih podataka. Svaki konvolutivni sloj raspolaže nizom parametrizovanih filtera koji vrše ove poslove. Zbog toga te konvolutivne slojeve nazivamo *više kanalnim*. Na primer, ako jedan filter detektuje vertikalne linije, prirodno je imati paralelno, nad istim prethodnim slojem (ili ulazom) i filter koji detektuje horizontalne linije. Upravo, parametri filtera predstavljaju parametre konvolutivne mreže i bivaju naučeni u procesu obučavanja mreže. Filter se realizuje jedinicama koje su organizovane u niz (u slučaju jednodimenzionalih signala poput zvuka) ili matricu (u slučaju dovodimenzionalih signala poput slike) i *dele vrednosti parametara*. Obično kao ulaze uzimaju vrednosti susednih jedinica iz prethodnog sloja. Na primer, u slučaju slike, to mogu biti vrednosti 3×3 jedinice iz prethodnog sloja. Ovakva organizacija jedinica je prikazana na slici ???. Na taj način, imajući u vidu da sve jedinice u jednom sloju imaju iste koeficijente, dejstvo konvolutivnog sloja se može razumeti kao konvolucija prethodnog sloja sa jedinicom definisanom parametrima tekućeg sloja ili u jednostavnijim terminima, kao prevlačenje filtera duž slike i izračunavanje vrednosti koju filter daje na svakoj poziciji.

U prethodnim razmatranjima nije diskutovano da li filter deluje samo nad tačno jednim kanalom prethodnog sloja ili nad više. Osnovna varijanta prepostavlja tako. Međutim, u praksi se implementiraju i dosta komplikovanije višekanalne konvolucije koje omogućavaju da filter deluje nad više kanala iz prethodnog sloja odjednom. Slika 9.6 to i odražava.

9.2.2 Agregacija

Sloj agregacije ukrupnjuje informacije koje dobija iz prethodnog sloja, obično tako što računa neku jednostavnu funkciju agregacije susednih jedinica prethodnog sloja, poput maksimuma ili proseka. Jednom kanalu konvolucije odgo-



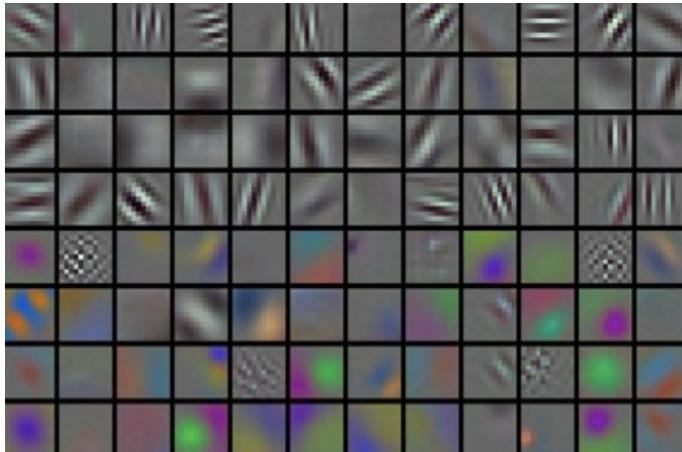
Slika 9.10: Konvolutivni sloj. Svaka jedinica, označena crvenim krugom, kao ulaze uzaze prethodnog sloja koji odgovaraju jedinicama koje se nalaze u crvenom kvadratu. Sve jedinice istog konvolutivnog sloja imaju iste vrednosti parametara i samim tim vrše isti obrazac u prethodnom sloju.

vara jedan kanal agregacije. Ukoliko agregira, na primer, 3×3 piskela, onda je broj izlaza ovog sloja 9 puta manji od broja izlaza prethodnog. Maksimum je najpopularniji izbor funkcije agregacije. Ukoliko 3×3 jedinice prethodnog sloja predstavljaju ulaz u jednu jedinicu sloja agregacije i ako ona računa njihov maksimum, dolazi do zanemarivanje informacije o tome gde je precizno neko svojstvo (poput uspravne linije) pronađeno, ali se ne gubi informacija da je pronađeno. Ovakva vrsta zanemarivanja informacije često ne šteti cilju koji treba postići. Na primer, ako je na slici pronađeno oko, uvo, usta i nos, informacija o tačnoj poziciji najverovatnije nije bitna za odlučivanje da li se na slici nalazi lice. Naime, u realnosti, šanse da slika koja sadrži navedene elemente, ne sadrži lice su izuzetno male. Ipak, ukoliko je potrebno napraviti mrežu koja igra igru u kojoj su pozicije objekata na ekranu bitne, nije poželjno koristiti agregaciju.

Uloga agregacije je smanjenje broja računskih operacija u višim slojevima, a i smanjenje broja parametara u mreži sa propagacijom unapred koja sledi za slojevima konvolucije i agregacije. Sve to rezultuje smanjenjem računske zahtevnosti pri optimizaciji i smanjenjem fleksibilnosti modela.

9.2.3 Interpretabilnost

Najčešća primena konvolutivnih neuronskih mreža je u obradi slika. Pokazale su se izuzetno uspešnim u prepoznavanju objekata na slikama. Kao što je rečeno, niži slojevi konvolutivne mreže detektuju jednostavne oblike. Vrlo



Slika 9.11: Vizualizacija oblika koje jedinice najnižeg sloja mreže prepoznaju.

je lako ustanoviti koji oblik detektuju jedinice prvog konvolutivnog nivoa. To je oblik za koji one daju najviše vrednosti na izlazima. Kako je aktivaciona funkcija monotona, to je oblik koji daje najvišu vrednost linearne kombinacije koju jedinica računa. Ako su koeficijenti jedinice w , oblik je dat kao rešenje problema

$$\max_{\|x\|=1} w \cdot x$$

Normiranje je važno pošto bi u slučaju da je skalarni proizvod pozitivan, povećavanjem intenziteta vektora x , uvek mogla biti dostignuta proizvoljna vrednost skalarnog proizvoda. Pod tim uslovom, lako se ustanovljava (recimo, postavljanjem parcijalnih izvoda po x na 0) da se maksimalna vrednost dostiže za vrednost $w/\|w\|$. Stoga, da bi se prikazao oblik koji jedinica najnižeg konvolutivnog sloja prepoznaće, dovoljno je vizualizovati njene koeficijente u vidu slike čije dimenzije odgovaraju dimenzijama filtera koji ta jedinica predstavlja. Jedan takav prikaz je dat na slici 9.11. Vizualizacija oblika koje prepoznaaju viši nivoi konvolutivne mreže je komplikovanija i zahteva neki vid optimizacije, kako bi se našli ulazi u mrežu za koje ove jedinice daju najviše vrednosti. Na slici 9.12 dat je prikaz oblika koje prpoznaaju jedinice višeg nivoa mreže koja prepoznaće objekte na slikama. Ova vrsta analize je važna zbog toga što pokazuje koja svojstva analiziranih objekata, u ovom slučaju slika, su važna za obavljanje posla koji mreža obavlja. Ova informacija ne mora uvek biti poznata korisniku i može predstavljati novo saznanje.

9.2.4 Kvaliteti, mane i poboljšanja

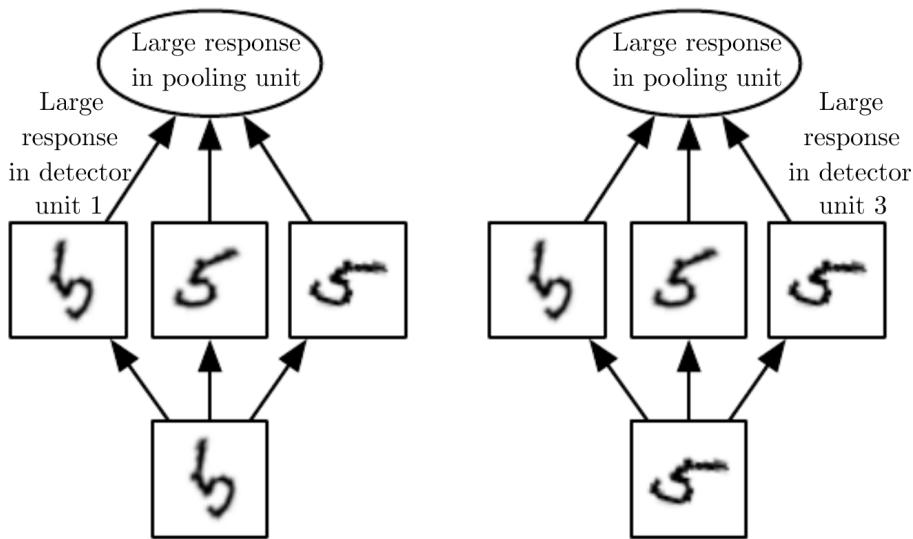
Konvolutivne neuronske mreže su vrhunski model za obradu signala mašinskim učenjem – slika, zvuka i vremenskih serija. Imaju nekoliko glavnih kvaliteta



Slika 9.12: Vizualizacija oblika koje prepoznaju jedinice višeg nivoa mreže – jedrenjak, plišani meda i bokal.

koji su zaslužni za njihove dobre performanse. Prvi su *proređene interakcije*, pod čime se podrazumeva da je svaka jedinica povezana samo sa malim brojem jedinica iz prethodnog sloja, umesto sa svim, kao što u slučaju mreža sa propagacijom unapred. Drugi je *deljenje parametara*, koje se odnosi na to da sve jedinice jednog kanala imaju iste parametre – definisane filterom tog kanala. Zahvaljujući tome, manji je broj parametara, a time i mogućnost preprilagođavanja. Takođe, kada parametri ne bi bili deljeni, kao kod mreže sa propagacijom unapred, učeći različite parametre za različite delove slike, mreža bi učila da nekim delovima ulaza pridaje posebnu semantiku. To može zvučati dobro, ali za detekciju lica negde na slici, ne bi bilo dobro da se nauči da nos treba da bude baš na sredini slike, jer bi mogao biti i na nekom drugom mestu ako foto aparat nije bio centriran na lice. Deljenje parametara omogući da se nauči filter koji traži nos bilo gde na slici. Treći je *neosetljivost na translacije*, odnosno svojstvo da će neki signal koji mreža traži biti nađen bez obzira kako je transliran na slici. Četvrti je *specijalizovanost za topologiju signala*. Naime i prethodno diskutovani modeli mašinskog učenja bi se mogli primenjivati na probleme vezane za zvuk, slike i slično, ali ako bi recimo trebalo da rade nad sirovom reprezentacijom slike u vidu piksela, raspored piksela bi bio potpuno proizvoljan, pošto ovi modeli ne uzimaju u obzir susednost piksela na slici, dok su konvolutivne mreže konstruisane imajući u vidu da to što su pikseli jedni u okolini drugih ima poseban značaj. Još jedan kvalitet, koji nije vezan za performanse je *delimična interpretabilnost*, zahvaljujući tome što se mogu vizualizovati ulazi na koje filteri daju najjači odgovor.

S druge strane, postoje i problemi. Naime, mreža je osetljiva na neke druge transformacije, poput rotacije i skaliranja (homotetije). Takođe, kako mreža sa propagacijom unapred koja se nalazi na kraju niza slojeva konvolutivne mreže uvek mora imati fiksiran broj ulaza, tako i konvolutivni deo mora proizvesti izlaz tačno određenih dimenzija, što zbog fiksiranog broja slojeva agregacije i fiksirane rezolucije agregiranja (npr. 3×3 vrednosti se zamenjuju jednom), znači i da ulazi moraju biti istih dimenzija.



Slika 9.13: Agregacija nad više kovolutivnih kanala odjednom. Ukoliko se u nekom od kovolutivnih kanala pronađe relevantni uzorak, agregacioni sloj prijavljuje da je pronađen. Pritom različiti kovolutivni kanali traže na različite načine transformisan uzorak, čime omogućavaju invarijantnost mreže u odnosu na datu transformaciju.

Postoje određeni pristupi poboljšanja kojima se ublažuju prethodno pomenuti problemi. U slučaju potrebe za neosetljivošću na određene transformacije, to se može donekle postići time što agregacija ne bi bila vršena nad jednim kanalom konvolucije, već nad više. U tom slučaju, različiti kovolutivni kanali mogu naučiti da traže različito transformisane uzorkе od značaja i ukoliko ga jedan od kanala nađe, agregacija maksimumom prijavljuje da je obrazac nađen. Ovo je ilustrovano slikom 9.13. Drugi problem, vezan za veličine ulaza, može se rešiti na više načina. Neki su komplikovani i uključuju izgradnju novih modela i na njih se ne osvrćemo. Neki, poput skaliranja slike, su naivni i vode gubitku informacije. Dodatno, osetljivost na transformacije, uključujući skaliranje, je već konstatovana slabost kovolutivnih mreža. Poluzadovoljavajući pristup bi bio taj da rezolucija aggregacije zavisi od dimenzija ulaza. Obično se ovaj tip aggregacije radi samo na poslednjem nivou. Recimo, ako se za slike dimenzija 1000×500 na poslednjem nivou aggregacije traži maksimum delova dimenzija 3×3 , onda bi se za slike dimenzija 2000×1000 tražio maksimum delova dimenzija 6×6 . Na taj način, dimenzije ulaza u mrežu sa propagacijom unapred ostaju iste. Još jedna mogućnost je da se na poslednjem nivou, aggregacija uradi uprosecavanjem svih vrednosti kanala. Primetimo da su ovakva rešenja moguća zahvaljujući deljenju parametara. Naime, u slučaju kovolutivne mreže, zahvaljujući deljenju parametara, je uvek moguće agregirati vrednosti različitih

jedinica jer imaju istu semantiku.

Pored diskutovanih problema koji su specifični za konvolutivne mreže, problemi diskutovani u slučaju mreža sa propagacijom unapred su takođe prisutni. To što su ublaženi navedenim kvalitetima konvolutivnih mreža, vodi povećanju ambicija i izgradnji većih mreža, sa većim brojem parametara, pa isti problemi ponovo dolaze do izražaja.

Deo II

Dodatak

Glava 10

Matematičko predznanje

Mašinsko učenje je u velikoj meri matematička disciplina. Kako po prirodi teorija koje se bave proučavanjem generalizacije, tako i po sveprisutnosti matematičkog aparata u metodama i primenama, čak i kada su one prvenstveno inženjerske prirode. Oblasti matematike koje su od najvećeg značaja za mašinsko učenje su linearna algebra, analitička geometrija, matematička analiza, verovatnoća i statistika. Pored njih, primenu nalaze i funkcionalna analiza, topologija, hiperbolička geometrija i druge. U nastavku je dat pregled matematičkih tema koje su najznačajnije za mašinsko učenje. Poznavanje nekih osnovnih pojmoveva, poput matrične algebre, determinantni, limesa, izvoda i drugih se podrazumeva. Neke od definicija koje slede mogu biti i opštije, ali je izlaganje namerno prilagođeno potrebama razumevanja mašinskog učenja.

10.1 Sopstvene vrednosti i sopstveni vektori

Kvadratna matrica $A \in \mathbb{R}^{n \times n}$ ima *sopstveni vektor* $x \in \mathbb{R}^n$ i odgovarajuću *sopstvenu vrednost* $\lambda \in \mathbb{R}$, ukoliko važi

$$Ax = \lambda x \quad x \neq 0$$

Drugim rečima, matrica A ne menja pravac sopstvenog vektora (iako mu možda menja smer). Poznato je da različitim sopstvenim vrednostima odgovaraju linearno nezavisni sopstveni vektori, kao i da su sopstvene vrednosti jednake nulama karakterističnog polinoma

$$\det(A - \lambda I)$$

Svakoj sopstvenoj vrednosti λ_i , $i = 1, \dots, m$ odgovara sopstveni potprostor V_i . Ukoliko se dimenzije ovih sopstvenih potprostora sabiraju na dimenziju matrice A , matrica se može svesti na dijagonalnu. Ako je D dijagonalna matrica sopstvenih vrednosti, u kojoj je svaka sopstena vrednost λ_i ponovljena $\dim(V_i)$ puta i ako je X matrica čije su kolone odgovarajući sopstveni vektori, važi

$$AX = XD$$

Onda važi i

$$A = XDX^{-1} \quad D = X^{-1}AX$$

Drugim rečima, linearne preslikavanje indukovano matricom A se može opisati faktorima izduživanja (sopstvenim vrednostima) duž određenih pravaca (sopstvenih vektora). Ukoliko je neka od sopstvenih vrednosti negativna, radi se o promeni smera. Ukoliko pomenuti uslov vezan za dimenzije prostora V_i nije ispunjen, matrica X neće biti invertibilna.

U slučaju relanih simetričnih matrica, uvek je moguće konstruisati odgovarajuću dijagonalnu matricu. Dodatno, tada važi $X^{-1} = X^T$, odnosno $XX^T = X^TX = I$. Drugim rečima, matrica sopstvenih vektora je ortogonalna.

10.2 Definitnost

Kvadratna matrica $A \in \mathbb{R}^{n \times n}$ je *pozitivno semi definitna* ukoliko za svaki vektor $x \in \mathbb{R}^n$ važi

$$x^T Ax \geq 0$$

Analogno se definiše *negativna semi definitnost*. Kvadratna matrica A je *strog pozitivno definitna* ili krace *pozitivno definitna* ukoliko važi

$$x^T Ax > 0$$

Analogno se definiše *negativna definitnost*. Pojam pozitivne definitnosti predstavlja uopštenje pojma pozitivnosti brojeva na matrice. Slično je sa ostalim pojmovima.

Prema Silvesterovom kriterijumu, matrica A je pozitivno definitna ako i samo ako su determinante svih kvadratnih podmatrica koje uključuju element a_{11} pozitivne. Kriterijumi za pozitivnu semidefinitnost i za negativnu definitnost su nešto komplikovaniji i ne diskutujemo ih.

Ukoliko je simetrična matrica A dijagonalno dominantna, odnosno važi

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}| \quad \text{za svako } i$$

i ako su dijagonalni elementi nenegativni, matrica A je pozitivno semidefinitna. Obrnuto ne mora da važi.

Kvadratna simetrična matrica A je pozitivno definitna ako i samo ako ima pozitivne sopstvene vrednosti. Ako je X matrica čije su kolone sopstveni vektori matrice A , znamo da važi $A = XDX^{-1}$. Zbog simetričnosti matrice, sopstveni vektori su ortogonalni, odnosno važi $X^{-1} = X^T$, što znači da važi $x^T Ax = x^T XDX^T x = (X^T x)D(X^T x)$, gde je matrica D dijagonalna. Transformacija $X^T x = y$ predstavlja promenu koordinata i nadalje pišemo $y^T Dy$. Ovaj izraz je pozitivan ako važi

$$\sum_{i=1}^n \lambda_i y_i^2 > 0$$

što je tačno za svako y ako i samo ako su sve sopstvene vrednosti λ_i duž dijagonale matrice D pozitivne. Time smo se uverili u tvrdnju sa početka paragrafa.

Kvadratna matrica A je pozitivno definitna ako i samo ako postoji donjetrougaona matrica L , takva da važi $A = LL^T$. Ovakva faktorizacija matrice A naziva se *Čoleski dekompozicijom*. Pokušaj Čoleski dekompozicije predstavlja računski efikasan način da se proveri pozitivna definitnost matrice.

10.3 Norma i skalarni proizvod

Neka je $X \subseteq \mathbb{R}^n$. *Norma* je funkcija $\|\cdot\| : X \rightarrow \mathbb{R}$ takva da za svako $\alpha \in \mathbb{R}$ i $x, y \in X$ važi:

- $\|\alpha x\| = |\alpha| \|x\|$
- $\|x + y\| \leq \|x\| + \|y\|$
- Ako važi $\|x\| = 0$, onda važi $x = 0$.

Intuitivno, norma vektora se povezuje sa njegovim intenzitetom, odnosno dužinom. Ipak, norme se mogu definisati na različite načine.

Takozvane p norme nad vektorima iz \mathbb{R}^n se definišu na sledeći način:

$$\|x\|_p = \sqrt[p]{\sum_{i=1}^n x_i^p}$$

Za svake dve p norme $\|\cdot\|_a$ i $\|\cdot\|_b$ postoje konstante $0 < c_1 \leq c_2$, takve da za svako $x \in X$ važi

$$c_1 \|x\|_b \leq \|x\|_a \leq c_2 \|x\|_b$$

Ovo svojstvo se naziva *ekvivalentnošću* p normi. Jedna od implikacija je da konvergencija u jednoj p normi znači konvergenciju u bilo kojoj drugoj p normi.

Norme se mogu lako definisati i nad matricama. U slučaju p normi, definicija se oslanja na p norme nad vektorima:

$$\|A\|_p = \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$$

Još jedna često korišćena matrična norma, ekvivalentna matričnim p normama je Frobenijusova norma:

$$\|A\|_F^2 = \sum_{i=1}^m \sum_{j=1}^n a_{ij}^2$$

Ukoliko za neki vektor x važi $\|x\| = 1$ kažemo da je taj vektor *normiran*. Postupak *normiranja* se izvodi deljenjem vektora njegovom normom. Očito, vektor može biti normiran u odnosu na jednu normu, a da nije normiran u odnosu na drugu.

Neka je $X \subseteq \mathbb{R}^n$. Skalarni proizvod je funkcija $\|\cdot\| : X \times X \rightarrow \mathbb{R}$ takva da za svako $\alpha \in \mathbb{R}$ i $x, y, z \in X$ važi:

- $x \cdot y = y \cdot x$
- $x \cdot (y + z) = x \cdot y + x \cdot z$
- $x \cdot (\alpha y) = \alpha(x \cdot y)$
- $x \cdot x \geq 0$
- $x \cdot x = 0 \Leftrightarrow x = \mathbf{0}$

Skalarni proizvod prirodno indukuje normu $x \cdot x = \|x\|$. Takođe, postoji veza između skalarnog proizvoda i ugla između vektora. Naime, važi $x \cdot y = \|x\|\|y\| \cos \angle(x, y)$, odnosno, u slučaju normiranih vektora, skalarni proizvod je kosinus ugla između dva vektora. Ukoliko je kosinus između dva vektora 1, ti vektori su kolinearni. U slučaju normiranih, oni su jednakim. Ukoliko je kosinus 0, vektori su ortogonalni, a ukoliko je -1 , vektori su suprotnih smerova. Otud, skalarni proizvod se može uzeti za meru sličnosti vektora sa rasponom $[-1, 1]$ i u tom svojstvu se često koristi u mašinskom učenju.

10.4 Izvod, parcijalni izvod i gradijent

Za funkciju $f : \mathbb{R} \rightarrow \mathbb{R}$ jedne promenljive, *izvod* $f' : \mathbb{R} \rightarrow \mathbb{R}$ se definiše na sledeći način:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x + h) - f(x)}{h}$$

Očito, izvod predstavlja odnos priraštaja funkcije na granicama nekog intervala i dužine tog intervala kada dužina tog intervala teži nuli. Intuitivno, izvod funkcije u nekoj tački predstavlja nagib funkcije u toj tački ili formalnije koeficijent pravca njene tangente u toj tački.

Za funkciju $f : \mathbb{R}^n \rightarrow \mathbb{R}$ više promenljivih, *parcijalni izvod* $\frac{\partial f}{\partial x_i} : \mathbb{R} \rightarrow \mathbb{R}$ se definiše kao

$$\frac{\partial f(x_1, \dots, x_n)}{\partial x_i} = \lim_{h \rightarrow 0} \frac{f(x_1, \dots, x_i + h, \dots, x_n) - f(x_1, \dots, x_i, \dots, x_n)}{h}$$

Umesto notacije $\frac{\partial}{\partial x_i}$, nekad ćemo koristiti notaciju ∂_i . Parcijalni izvod kvanti-fikuje nagib duž samo jednog koordinatnog pravca.

Izvodi i parcijalni izvodi se umesto izračunavanjem limesa, obično izračunavaju prema poznatim pravilima za izračunavanje izvoda nekih poznatih funkcija i pravila izvoda složene funkcije. U svom najjednostavnijem obliku, ovo pravilo glasi

$$\partial_i(f \circ g) = (\partial_i f \circ g)\partial_i g$$

Ukoliko je funkcija g vektorska, odnosno važi $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$, onda pravilo glasi

$$\partial_i(f \circ g) = \sum_{j=1}^m (\partial_j f \circ g) \partial_i g_j$$

gde g_j označava j -tu koordinatu (vektorske) vrednosti funkcije g .

Vektor svih parcijalnih izvoda funkcije naziva se *gradijentom funkcije* i označava ∇f .

Pored prvih parcijalnih izvoda, od značaja su i drugi parcijalni izvodi

$$\frac{\partial^2 f(x_1, \dots, x_n)}{\partial x_i \partial x_j}$$

Umesto $\frac{\partial^2}{\partial x_i \partial x_j}$ nekada ćemo skraćeno pisati ∂_{ij} . Matricu drugih parcijalnih izvoda funkcije ćemo nazivati *hesijanom funkcije* i označavati $\nabla^2 f$.

Primer 9 Neki primer izračunavanja.

10.5 Konveksnost

Neka je X konveksan skup i neka je realna funkcija f definisana na njemu. Tada za funkciju f kažemo da je *konveksna* ukoliko za svako $\alpha \in [0, 1]$ i svako $x_1, x_2 \in X$ važi

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2)$$

Ukoliko funkcija ispunjava dati uslov sa strogom nejednakostju, funkcija je *strogo konveksna*. Funkcija f je (*strogo*) *konkavna* ukoliko je funkcija $-f$ (strogo) konveksna. Intuitivno, funkcija je konveksna, ako je svaka njena sečica u svakoj tački između tačaka preseka iznad grafika funkcije. Sva naredna razmatranja su data za konveksne funkcije. Obično važe analogna svojstva za strogo konveksne i za konkavne funkcije.

Ukoliko je funkcija f diferencijabilna u tački x , onda je konveksna u tački x ako i samo ako postoji okolina tačke x takva da za svako y iz te okoline važi

$$f(y) \geq f(x) + \nabla f(x)^T(y - x)$$

ili u slučaju jedne promenljive

$$f(y) \geq f(x) + f'(x)(y - x)$$

Drugim rečima, funkcija je konveksna ukoliko je njen grafik iznad tangentne ravni.

Dva puta diferencijabilna funkcija f je strogo konveksna u nekoj tački ukoliko je hesijan $\nabla^2 f(x)$ pozitivno definitan u nekoj okolini te tačke. Hesijan funkcije koja je konveksna u nekoj tački u nekoj okolini te tačke ima pozitivne sopstvene vrednosti.

Konveksne funkcije imaju naredna svojstva koja često mogu olakšati prepoznavanje da je neka funkcija konveksna:

- Ako su f_1, \dots, f_m konveksne funkcije i važi $w_1 \geq 0, \dots, w_m \geq 0$, onda je i funkcija

$$w_1 f_1(x) + \dots + w_m f_m(x)$$

konveksna funkcija.

- Ako je f konveksna funkcija, A matrica i b vektor odgovarajućih dimenzija, onda je i $f(Ax + b)$ konveksna funkcija.
- Ako su f_1, \dots, f_m konveksne funkcije, onda je i funkcija

$$\max\{f_1(x), \dots, f_m(x)\}$$

konveksna funkcija. Isto važi i za supremum nad beskonačnim skupom konveksnih funkcija.

- Kompozicija $f \circ g$ je konveksna ako je funkcija f konveksna i neopadajuća po svim argumentima, a funkcija g konveksna ili ako je funkcija f konveksna i nerastuća po svim argumentima, a g konkavna.

Funkcija diferencijabilna u tački x se naziva *jako konveksnom* ukoliko za svako y iz neke okoline tačke x važi

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{m}{2} \|y - x\|^2$$

za neko $m > 0$. Jaka konveksnost označava stroži uslov od konveksnosti. Ne treba je mešati sa strogom konveksnošću. Lako se pokazuje da je dva puta diferencijabilna funkcija jako konveksna u tački x ukoliko je matrica

$$\nabla^2 f(x) - mI$$

pozitivno semidefinitna za neko $m > 0$.

10.6 Lokalni optimumi

Funkcija u nekoj tački ima (*strogji*) *lokalni minimum* ukoliko u nekoj okolini te tačke uzmia (strogo) veće vrednosti nego u toj tački. Analogno se definiše (*strogji*) *lokalni maksimum* funkcije. Lokalni minimumi i lokalni maksimumi se skupa nazivaju *lokalnim optimumima* funkcije. U nastavku će biti reči o lokalnim minimumima, ali analogni zaključci važe i za stroge lokalne minimume i za lokalne maksimume.

Ako je x optimum funkcije i ako je ona u njemu diferencijabilna, onda važi $\nabla f(x) = 0$. Da bi funkcija imala minimum u tački x mora da važi $\nabla^2 f(x) > 0$ i funkcija mora biti konveksna. Otud hesijan (ako postoji) $\nabla^2 f(x)$ mora biti pozitivno semidefinitan, odnosno sopstvene vrednosti hesijana moraju biti negativne. Ukoliko bi postojala negativna sopstvena vrednost, pravac odgovarajućeg sopstvenog vektora bi bio pravac u kojem vrednost funkcije opada u odnosu na tačku x , pa ona ne bi bila optimum.

10.7 Integral

U ovom odeljku integral nećemo formalno definisati, već će akcenat biti na njegovom smislu. Od ključnog značaja je intuicija uopštenja sume. Sume se formulišu oslanjajući se na diskretne promenljive kao indekse. Ovakva operacija se često vrši nad nizovima – funkcijama diskretne promenljive:

$$\sum_{i=1}^n a_i$$

Ipak, nekad se prirodno nameće potreba sumiranja vrednosti funkcije po kontinualnoj promenljivoj. Ovaj posao se vrši integracijom, koja se u jednodimenzionom slučaju može razumeti kao suma označenih površina pravougaonika između grafika funkcije i koordinatne x ose, kada dužine stranica pravougao-nika koje leže na x osi teže nuli.

Pored sumiranja, integral često služi težinskom uprosečavanju funkcija. Neka je potrebno izračunati prosečnu vrednost rude u nekom rudnom ležištu. Ukoliko bi se ležište moglo podeliti na n homogenih celina od kojih svaka ima vrednost v_i po toni i obuhvata količinu rude q_i izraženu u tonama, prirodno bi bilo uprosečiti vrednosti v_i , ali pridajući veću težinu onoj vrednosti koja se javlja u većoj količini. Takva prosečna vrednost data je izrazom:

$$\sum_{i=1}^n \frac{q_i}{\sum_{j=1}^n q_j} v_i$$

Težina koja se pridaje svakoj vrednosti je jednaka udelu količine takve rude u ležištu. U slučaju da ne postoji homogene celine, već kvalitet i gustina željenog materijala približno neprekidno variraju od tačke do tačke ležišta, prirodno je zamjeniti sumu integralom u kojem su diskretne vrednosti i količine zamjenjene funkcijama $v(x)$ i $q(x)$ koje zavise od lokacije:

$$\int_{\mathcal{D}} v(x)q(x)dx$$

pri čemu je \mathcal{D} oblast koju zauzima ležište. Kako bi izvedena zamena sume integralom imala smisla, potrebno je da funkcija $q(x)$ zadovoljava ključno svojstvo koje zadovoljavaju težine pridružene vrednostima u sumiranju, a to je da se sabiraju na 1:

$$\sum_{i=1}^n \frac{q_i}{\sum_{j=1}^n q_j} = 1$$

Analogno tome, mora važiti

$$\int_{\mathcal{D}} q(x)dx = 1$$

Navedena interpretacija integrala kao uprosečavanja je sveprisutna u mašinskom učenju, pre svega zbog oslanjanja na verovatnoću i pojам matematičkog očekivanja.

10.8 Verovatnoća

Verovatnoća se bavi kvantifikovanjem izglednosti različitih događaja. Postoje dve ključne interpretacije pojma verovatnoće u okviru dve različite škole statistike. *Frekventistička statistika* interpretira verovatnoću nekog događaja kao dugoročnu frekvenciju opažanja tog događaja. Stoga, verovatnoća pisma ili glave pri bacanju novčića jednaka je po 0.5 zato što se u ponovljenim eksperimentima primećuje da se udeo eksperimenata u kojima je ishod pismo bliži 0.5 kako broj ponavljanja eksperimenta raste. S druge strane, *bajesovska statistika* interpretira verovatnoću kao nivo uverenja. Pošto novčić ima dve strane i pošto ne vidimo nikav razlog da jedna strana pada češće nego druga, podjednako očekujemo glavu koliko i pismo. Otud verovatnoća 0.5 za svaki od ishoda. Iako uverenja mogu biti proizvoljna, jednom kada su uverenja za elementarne ishode eksperimenta definisana, verovatnoće komplikovanih događaja se računaju u skladu sa pravilima verovatnoće koja garantuju saglasnost izračunavanja. Konkretno, ako je verovatnoća padanja glave 0.5, verovatnoća padanja tri glave mora biti 0.125. Pravila računanja verovatnoće složenijih događaja (koji se definišu nad elementarnim ishodima) definisana su aksiomama verovatnoće i važe u obe interpretacije. Zapravo, ova pravila definišu (ali ne jedinstveno) pojam *verovatnosne mere*, koju označavamo sa p .

Pored verovatnoće događaja, definiše se i *uslovna verovatnoća* događaja kao

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)}$$

Ukoliko događaj Y predstavlja parne ishode bacanja kockice, a događaj X ishode 4, 5 i 6, ukoliko znamo da je bacanjem kockice dobijen paran broj, verovatnoća da je to neki od brojeva 4, 5 i 6 mora biti $2/3$ pošto to ne može biti 5, a 4 i 6 su dva ishoda od mogućih 2, 4 i 6. Presek događaja $X \cup Y$ označavaćemo i kao XY . Koristeći definiciju uslovne verovatnoće dva puta, izvodimo sledeću jednakost:

$$P(X|Y) = \frac{P(XY)}{P(Y)} = \frac{P(Y|X)P(X)}{P(Y)}$$

poznatu pod nazivom *Bajesova formula*, a koja povezuje uslovne verovatnoće $P(X|Y)$ i $P(Y|X)$ i koja se često koristi u mašinskom učenju.

Važan koncept je koncept nezavisnosti događaja. Intuitivno, dva događaja su nezavisna ukoliko informacija od događanju jednog ne daje nikakvu informaciju od događanju drugog, ili formalno:

$$P(X|Y) = P(X)$$

Na osnovu definicije uslovne verovatnoće, ova relacija se može predstaviti i drugačije

$$P(XY) = P(X)P(Y)$$

Nezavisnost može biti i uslovna:

$$P(XY|Z) = P(X|Z)P(Y|Z)$$

U tom slučaju informacija o događaju X nam ne govori ništa o događaju Y ako imamo informaciju da se desio događaj Z . Primera radi, veličina vokabulara kojim vlada jedna osoba nam na govori ništa o njenoj visini ako su nam poznate njene godine. U suprotnom, bogat vokabular sugerije veću visinu, pošto su deca i niža i imaju manji vokabular od odraslih.

Važne pojmove gustine raspodele i kumulativne funkcije raspodele uvešćemo naopakim redom – onim koji odgovara značaju tih pojmove u mašinskom učenju i verovatno primenjenoj matematici uopšte, iako ne i opštosti pojmove. Nenegativna funkcija $p : \mathbb{R}^n \rightarrow \mathbb{R}$ za koju važi

$$\int_{\mathbb{R}^n} p(x) dx$$

naziva se *gustina raspodele*. Intuitivno, ova funkcija verovatnjim događajima pridružuje više vrednosti, a manje verovatnim niže. Međutim upravo navedena formulacija je problematična jer strogo gledano, svaka pojedinačna tačka je mera nula, pa ima i verovatnoću nula. Ipak, ova intuicija važi na proizvoljno malim intervalima, što dozvoljava da o gustini raspodele razmišljamo na navedeni način. Odgovarajuća *kumulativna funkcija raspodele*¹ je:

$$F(x_1, \dots, x_n) = \int_{-\infty}^{x_1} \dots \int_{-\infty}^{x_n} p(x_1, \dots, x_n) dx_1 \dots dx_n$$

Ona u mašinskom učenju ne igra važnu ulogu, ali daje vezu sa pojmom verovatnosne mere. Naime, može se pokazati da za datu kumulativnu funkciju raspodele F postoji tačno jedna verovatnosna mera P za koju važi

$$F(x_1, \dots, x_n) = P((-\infty, x_1] \times \dots \times (-\infty, x_n])$$

Moguće je definisati i diskretnu funkciju raspodele nad nizom realnih brojeva, ali se u mašinskom učenju prvenstveno oslanjamamo na pojam gustine raspodele nad neprekidnim domenom.

Važan pojam vezan za raspodelu promenljive je pojam *marginalne raspodele*. Ukoliko je poznata zajednička raspodela slučajnih vektora X i Y , marginalna raspodela vektora X je raspodela vektora X dobijena sumiranjem ili integracijom (zavisnosti od toga da li je diskretni ili neprekidni slučaj) po Y . Važi:

$$p(x) = \int_y p(x, y) dy$$

Ova formula ima intuitivnu interpretaciju. Ako nas zanima koliko često se opaža neka vrednost vektora X i znamo koliko se često ona javlja u kombinacijama sa svim različitim vrednostima vektora Y , intuitivno je da učestalost svih pojavljivanja te vrednosti vektora X dobijamo sumirajući učestalosti tih kombinacija.

¹Pojam kumulativne funkcije raspodele je opštiji od ove definicije. Postoje kumulativne funkcije raspodele za koje ne postoji odgovarajuća gustina raspodele.

Slučajna promenljiva je funkcija koja preslikava elementarne ishode u \mathbb{R} . Na primer visina slučajno odabrane osobe iz neke populacije je slučajna promenljiva. Njena svojstva su opisana pridruženom raspodelom verovatnoće njenih vrednosti. Verovatnoća da neprekidna slučajna promenljiva uzme vrednosti iz nekog intervala je verovatnoća ishoda za koje slučajna promenljiva daje vrednosti iz tog intervala. Na osnovu ove veze sa pojmom verovatnosne mere nad nekim prostorom događaja, pojmovi gustine raspodele i kumulativne funkcije raspodele se prirodno definišu za slučajne promenljive. Intuitivniji način razumevanja gustine raspodele slučajne promenljive, bio bi sledeći. Neka je dat uzorak vrednosti slučajne promenljive, recimo uzorak visna različitih ljudi. Sve visine imaju svoje mesto na realnoj pravoj koju možemo izdeliti na podintervale jednakе širine. Lažne slikovitosti radi, zovimo te intervale korpicama (eng. bin). U svaku korpicu upada određeni broj visina. Ukoliko za svaku korpicu nacrtamo pravougaonik visine proporcionalne broju (gle!) visina u njoj, dobijamo *histogram* visina. Smanjujući širinu korpice, a povećavajući broj izmerenih ljudi, histogram postaje sve uglačaniji i teži gustini raspodele. Pažljiv čitalac bi trebalo da je primetio da je primer zasnovan na visinama problematičan jer ljudi nema neprebrojivo mnogo, dok je gustina raspodele definisana nad neprebrojivim domenom. Takav čitalac je slobodan da razmišlja o temperaturama, osim ako se dovoljno razume u fiziku da se zapita o kvantima energije i počne da sumnja u to da matematički pojam neprekidnosti može da se savršeno primeni bilo gde u stvarnom svetu.

Pomenuti pojam histograma je važan za upotrebu mačinskog učenja u praksi kako bi se sumarno vizualizovale vrednosti različitih promenljivih. Ključni problem u njegovoj upotrebi je izbor širine korpice. Ukoliko su korpice vrlo široke, dobija se vrlo gruba slika raspodele vrednosti promenljive. Ukoliko su korpice vrlo uske, ne očekuje se da se u bilo kojoj nađe veliki broj tačaka, možda nijedna. Takav histogram ne daje nikakvu korisnu informaciju, jer se iz njega obično ne može razaznati nikakav karakterističan oblik, a i to što se vidi može značajno zavisi od konkretnog (slučajno generisanog) uzorka. Stoga histogram uvek treba pogledati za veći broj različitih širina korpica.

10.9 Sredina i rasipanje slučajne promenljive

Slučajne promenljive su opisane svojom raspodelom. Ipak, raspodele je često korisno sumirati nekim jednostavnim veličinama, kako bi se bolje razumele. Neke od osnovnih informacija o raspodelama, za koje smo obično zainteresovani, su srednja vrednost i rasipanje slučajne promenljive. Ovi intuitivni pojmovi se formalizuju pojmovima *matematičkog očekivanja* i *varijanse* slučajne promenljive.

Matematičko očekivanje je prosek vrednosti slučajne promenljive otežan njihovom učestalošću, odnosno verovatnoćom. Ako je $p(x)$ gustina raspodele promenljive X , onda se, imajući u vidu diskusiju upotrebe integrala u svrhe

uprosečavanja, matematičko očekivanje ove promenljive može definisati kao:

$$\mathbb{E}[X] = \int_{\mathcal{D}} xp(x)dx$$

gde je \mathcal{D} skup vrednosti koje uzima slučajna promenljiva X . Očekivanje se u praksi aproksimira uzoračkom sredinom:

$$\mu_n = \frac{1}{n} \sum_{i=1}^n x_i$$

Prosek nije jedini način da se definiše ideja sredine. Otud ni matematičko očekivanje nije jedini način definisanja srednje vrednosti. Jedan, često pogodniji, parametar raspodele slučajne promenljive je *medijana*, odnosno vrednost od koje slučajna promenljiva u pola slučajeva uzima manju, a u pola veću vrednost. Formalnije, to je vrednost m , takva da važi

$$\int_{-\infty}^m xp(x)dx = \frac{1}{2}$$

Primetimo da ova definicija pretpotavlja jednodimenzionali slučaj. Postoje višedimenziona uopštenja, ali o njima neće biti reči. Medijana se u praksi aproksimira *uzoračkom medijanom*, koja se može definisati kao element na sredini sortiranog niza vrednosti uzorka ukoliko je broj elemenata u uzorku neparan ili kao aritmetička sredina dve susedne vrednosti na sredini sortiranog niza ukoliko je broj elemenata u uzorku paran. Osnovni kvalitet medijane je njena robusnost, odnosno neosetljivost na određenu količinu šuma u podacima. Razmotrimo uzorak brojeva $\{1, 3, 5, 7, 9\}$. Ukoliko dođe do pogrešnog očitavanja i 1 bude očitano kao 3, dobija se uzorak $\{3, 3, 5, 7, 9\}$. Prosek novodobijenog uzorka je drugi, ali medijana je ista kao kod prvog uzorka. Potrebno je da neki od podataka bude pogrešno očitan kao podatak sa druge strane medijane da bi se ona promenila. Takođe, medijana je u praksi često realističnija slika srednje vrednosti od proseka, ali to naravno zavisi od tačnog smisla pojma sredine koji nam je važan. Na primer, u slučaju analize bogatstva građana jedne zemlje, medijana daje dosta važniju informaciju. Prosek plata je direktno proporcionalan sumi plata (za faktor broja građana čija plata se analizira). Što je ukupno bogatstvo veće, veća je i prosečna vrednost. Ipak, prosek ne govori ništa o raspodeli bogatstva. Na primer, ukoliko 99% građana raspolaže primanjima od 10.000 dinara, dok 1% građana raspolaže primanjima od 9.010.000 dinara mesečno, prosečno primanje je 100.000 dinara, što ostavlja utisak da građani imaju dobar životni standard, što je pogrešno u 99% slučajeva. S druge strane, vrednost medijane je 10.000 dinara, što daje vernu sliku materijalnog stanja građana, osim u 1% slučajeva.

Varijansa se definiše kao prosečno odstupanje od proseka, odnosno

$$\text{var}[X] = \mathbb{E}[X - \mathbb{E}X]^2$$

i aproksimira se uzoračkom varijansom:

$$\sigma_n^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu_n)^2$$

Kao što prosek nije jedina formalizacija pojma sredine, tako ni varijansa nije jedina formalizacija pojma rasipanja slučajne promenljive. Ako se sa m označi medijana, onda bi jedna alternativa mogla biti

$$m[X - m[X]]$$

10.10 Statističke ocene i njihova svojstva

Funkcije uzoraka koje zadovoljavaju određene pretpostavke (koje su u mašinskom učenju tipično zadovoljene) nazivaju se *statistikama*. Prosek, uzoračka varijansa, maksimum i minimum uzorka predstavljaju primere statistika. Statistike često služe za empirijsku aproksimaciju, odnosno *ocenu*, nekih teorijskih veličina. Na primer, rastojanje između dve udaljene tačke, koje predstavlja objektivnu veličinu kojoj nemamo direktni pristup, može se oceniti prosekom većeg broja geodetskih merenja. Svako od tih merenja sadrži grešku o kojoj razmišljamo kao o slučajnoj (nekad je pozitivna, nekad negativna, nekada mala, a nekad velika), ali se njihovim uprosećavanjem greška smanjuje. Iだlje, na osnovu konačnog uzorka ne možemo biti sigurni da je ocena precizna, pa čak i uopste upotrebljiva, pa zbog toga obično ulazimo u analizu svojstava takve ocene. Ocena je *konzistentna* (eng. consistent) ukoliko teži veličini koju ocenjuje kako veličina uzorka teži beskonačnosti, odnosno:

$$\lim_{n \rightarrow \infty} \hat{\theta}_n = \theta^*$$

gde je $\hat{\theta}_n$ ocena na osnovu uzorka veličine n , a θ^* prava vrednost veličine koja se ocenjuje. Ocena je *nepristrasna* ukoliko je očekivanje ocene jednako vrednosti koja se ocenjuje, odnosno:

$$\mathbb{E}[\hat{\theta}] = \theta$$

gde je \mathbb{E} očekivanje u odnosu na raspodelu podataka sa gustinom $p_\theta(x)$ od kojih $\hat{\theta}$ zavisi. Eventualno odstupanje tog očekivanja od prave vrednosti

$$\mathbb{E}[\hat{\theta}] - \theta$$

naziva se *sistematskim odstupanjem*. Pristrasnost ne mora isključivati konzistentnost, pošto se sistematsko odstupanje može smanjivati sa povećanjem veličine uzorka. Među ocenama jednakog sistematskog odstupanja, ocena najmanje varijanse je *najbolja*. Intuitivno je preferirati najbolju nepristrasnu ocenu. Ipak, pokazuje se da pristrasne ocene često mogu biti korisnije jer ukupna greška neke ocene ne zavisi samo od sistematskog odstupanja, već i od varijanse, a pristrasne ocene mogu imati osetno manju varijansu od nepristrasnih.

10.11 Statistički modeli

Statistički model je definisan prostorom događaja i familijom verovatnosnih mera nad tim prostorom događaja. Ključni problem matematičke statistike je izbor jedne iz skupa kandidatnih verovatnosnih mera, koja u nekom smislu najbolje opisuje empirijski opažene podatke. Statistički modeli često pretpostavljaju određena svojstva pomenutih verovatnosnih mera od kojih neka predstavljaju tehničke pogodnosti (čest je slučaj da se iz ovog razloga pretpostavlja normalna raspodela podataka), a neke predstavljaju odluke prilikom modelovanja zavisnosti među različitim promenljivim koje model dovodi u vezu. Prilikom definisanja modela, često se vodi računa o tome da se model definiše tako da je u praksi moguće vršiti izbor verovatnosne mere na osnovu podataka, čak i ako neke od postavljениh pretpostavki nisu realistične. Iako nepovoljne, ovakve pretpostavke su potrebne, jer model koji ne bi zadovoljio ovaj zahtev ne bi bio koristan u praksi.

10.12 Metod maksimalne verodostojnosti

Kao što je rečeno, ključni problem matematičke statistike je izbor jedne iz skupa kandidatnih verovatnosnih mera, koja u nekom smislu najbolje opisuje empirijski opažene podatke. Specifikacija statističkog modela često uključuje konačan broj parametara čije vrednosti je potrebno izabrati kako bi se izabrala verovatnosna mera, odnosno kako bi se precizirao model. Postavlja se pitanje – na koji način je moguće vršiti izbor vrednosti tih parametara kako bi se dobio smislen rezultat. Jedan od ključnih principa na kojima počivaju metodi izbora vrednosti parametara statističkog modela, prisutan i u drugim oblastima statistike (npr. u testiranju statističkih hipoteza) je da *ne verujemo u neverovatne događaje*, odnosno da odbacujemo vrednosti parametara pri kojima bi opaženi podaci bili malo verovatni. Alternativno, prihvatomamo vrednosti parametara pri kojima bi opaženi podaci bili visoko verovatni. Naglasimo intuitivnost ovog principa. Negativan bilans na bankovnom računu koji nekada vidimo kroz sistem elektronskog bankarstva, pored ostalih mogućnosti, može biti objašnjen bilo nepažljivim trošenjem novca, bilo promenom stanja memorije bankovnog servera usled pogotka čestice kosmičkog zračenja u neku memorijsku ćeliju. Iako bismo preferirali da se drugo objašnjenje ispostavi tačnim, racionalan posmatrač bi se ipak oprededio za prvo, upravo zato što deluje verovatnije. Jedan pristup formalizaciji ovakvog principa je *metod maksimalne verodostojnosti* (eng. maximal likelihood). Biće formulisan za neprekidan slučaj, ali je definicija u diskretnom slučaju analogna, samo što se umesto gustine raspodele koristi diskretna funkcija raspodele. Neka je p_θ gustina raspodele određena parametrima θ statističkog modela. Neka je $X = \{x_1, \dots, x_n\}$ skup opažanja. Pod pretpostavkom njihove nezavisnosti, gustina raspodele opaženih događaja

je jednaka:

$$\mathcal{L}(\theta) = \prod_{i=1}^n p_\theta(x_i)$$

Funkcija \mathcal{L} se naziva verodostojnošću parametara θ , a princip izbora vrednosti je

$$\theta^* = \max_{\theta} \mathcal{L}(\theta)$$

U praksi je pogodnije baratati sumama nego proizvodima, kako zbog analitički pogodnjeg diferenciranja, tako i zbog izbegavanja prekoračenja (u slučaju množenja brojeva velikih po apsolutnoj vrednosti) i potkoračenja (u slučaju množnja brojeva malih po apsolutnoj vrednosti). Takođe, konvencija je da se problemi optimizacije češće predstavljaju kao problemi nalaženja minimuma, nego maksimuma. Stoga se prethodni problem često zamenjuje narednim

$$\theta^* = \max_{\theta} \mathcal{L}(\theta) = \min_{\theta} (-\log \mathcal{L}(\theta)) = \min_{\theta} \left(- \sum_{i=1}^n \log p_\theta(x_i) \right)$$

pri čemu druga jednakost važi zahvaljujući tome što je logaritam monotono rastuća funkcija. Funkcija koja se minimizuje je negativna vrednost logaritma verodostojnosti (eng. negative log likelihood) i često se označava skraćenicom NLL.

Nalaženje minimuma često nije moguće izvršiti analitički, pa se stoga često vrši gradijentnim metodama, koje počivaju na izračunavanju gradijenta funkcije $-\log \mathcal{L}(\theta)$ po parametrima θ .