Odabir i regularizacija linearnih modela s primjenom u aktuarstvu

Petrunić, Ivan

Professional thesis / Završni specijalistički

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:054531

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK
Poslijediplomski specijalistički studij aktuarske matematike

Ivan Petrunić

ODABIR I REGULARIZACIJA LINEARNIH MODELA S PRIMJENOM U AKTUARSTVU

Završni rad

Voditelj završnog rada:
prof. dr. sc. Bojan Basrak

Zagreb, 2020

Sadržaj

Poglavlje 1

Uvod

1.1 Motivacija

Svakim danom je u različitim industrijama, uključujući industriju osiguranja, dostupno sve više podataka. Za donošenje ispravnih odluka, potrebno je razlučiti koji od tih podataka su bitni za odlučivanje, a koji nisu. Trošenje resursa na prikupljanje podataka koji u konačnici nisu relevantni trebalo bi minimizirati. Ukoliko potencijalni kupac proizvoda osiguranja na web stranici jednog osiguratelja mora odgovoriti na dvadeset pitanja kako bi dobio ponudu, a kod drugog osiguratelja samo na deset, drugi osiguratelj će biti u prednosti.

Kod velikog broja podataka želimo napraviti model koji ih što bolje opisuje. Taj model možemo koristiti za interpretaciju, kako bismo vidjeli koje su od opaženih varijabli u najvećoj vezi s opaženim odazivom, te za predviđanja, kako bismo procijenili visinu odaziva za opažanja koja nisu bila uključena pri izgradnji modela.

U poglavljima 2, 3 i 4 opisujemo metode koje možemo koristiti pri ostvarenju opisanih ciljeva, dok u poglavlju 5 ilustriramo praktičnu primjenu tih metoda u slučaju osiguranja od profesionalne odgovornosti liječnika. No prvo uvedimo pojmove koje ćemo koristiti u ostatku rada.[1]

1.2 Osnovni pojmovi

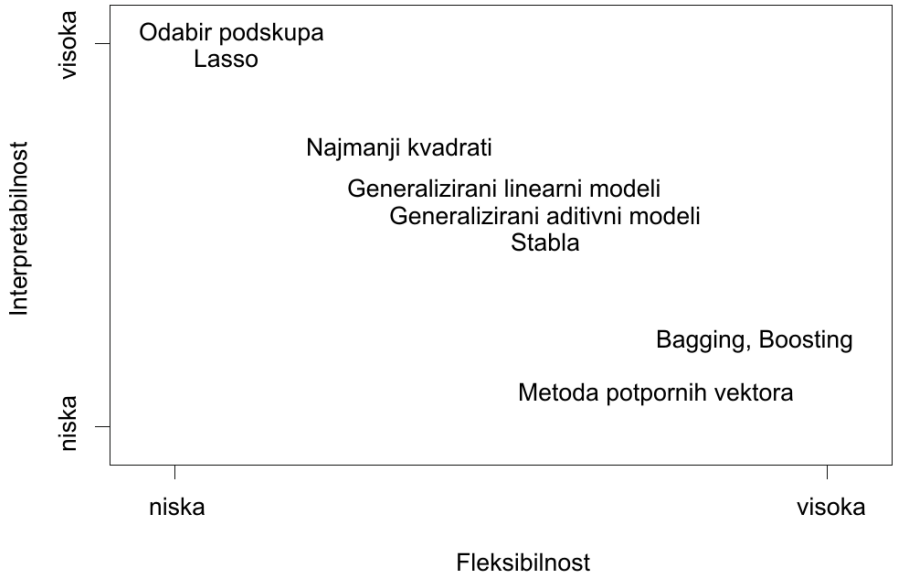
1.2.1 Kompromis između preciznosti predviđanja i interpretabilnosti modela

Kod modeliranja podataka, metode mogu biti više ili manje fleksibilne. Kao što ćemo vidjeti, linearna regresija metodom najmanjih kvadrata je relativno nefleksibilna, jer rezultira

[1] Sljedeće potpoglavlje temelji se na [1], str. 23-24; [2], str. 66-67 i [3], str. 15-36.

samo linearnim funkcijama (poput pravaca u dvodimenzionalnom, odnosno ravnina u trodimenzionalnom slučaju). Fleksibilnije metode se mogu bolje prilagoditi podacima, no i restriktivnije metode imaju svoje prednosti.

Na primjer, ako nas zanima samo zaključivanje (inferencija), linearni model može biti dobar odabir jer je jednostavnije opisati odnos između odaziva Y i prediktora $X_1, X_2, ..., X_p$. Primjerice, ako je model oblika $Y = 3X_1 + 7X_2$, onda možemo reći da, ako se X_2 poveća za jednu jedinicu, onda će se odaziv povećati za 7 jedinica. Nasuprot tome, vrlo fleksibilni pristupi poput korištenja splajnova mogu dovesti do vrlo složenih procjena odaziva, kod kojih veza između pojedinog prediktora i odaziva ne mora biti očita.



Slika 1.1: Prikaz kompromisa između fleksibilnosti i interpretabilnosti za različite metode statističkog učenja. Općenito, što se fleksibilnost neke metode povećava, to se njena interpretabilnost smanjuje. (Izvor: [3], Fig. 2.7)

Na slici 1.1 dana je ilustracija kompromisa između fleksibilnosti i interpretabilnosti za neke metode statističkog učenja. Linearna regresija metodom najmanjih kvadrata, o kojoj će biti riječi u sljedećem poglavlju, je relativno nefleksibilna, ali se može lako interpretirati. Lasso regresija, koja se obrađuje u četvrtom poglavlju, oslanja se na linearni model, no restriktivnija je pri određivanju koeficijenata, pri čemu neke od njih postavlja točno na nulu. Zato je u tom smislu lasso regresija manje fleksibilna od linearne regresije. S druge strane, lakše ju je interpretirati, jer će u konačnom modelu varijabla odaziva biti povezana

samo s malim podskupom prediktora, i to onima kod kojih su procijenjeni koeficijenti različiti od nule.

Generalizirani aditivni modeli (GAM), s druge strane, proširuju linearni model kako bi se u obzir uzeli i određeni nelinearni odnosi. Prema tome, GAM su fleksibilniji od linearne regresije, ali su manje interpretabilni, jer se odnos između svakog prediktora s odazivom sada modelira krivuljama. Još jedno proširenje linearnog modela su generalizirani linearni modeli (GLM). Konačno, nelinearne metode kao što su bagging, boosting i metoda potpornih vektora su vrlo fleksibilni pristupi, no koje je teže interpretirati.

Kada nam je cilj inferencija, očite su prednosti korištenja jednostavnih i relativno nefleksibilnih metoda statističkog učenja. Ipak, u nekim situacijama nas može zanimati samo predviđanje, a ne interpretabilnost nekog modela. Na primjer, ako nam je cilj razviti algoritam koji će dati predviđanje za cijenu neke dionice, naš jedini zahtjev bit će da algoritam daje što preciznije predviđanje, dok nas interpretabilnost uopće ne zanima. U tom kontekstu mogli bismo očekivati da bi bilo najbolje koristiti najfleksibilniji dostupni model. No, to ne mora uvijek biti slučaj. Ponekad ćemo češće dobiti preciznija predviđanja korištenjem manje fleksibilnih metoda. Ovaj fenomen, koji se na prvi pogled može činiti kontraintuitivnim, vezan je uz potencijal da vrlo fleksibilne metode budu pretrenirane (engl. overfitted). O ovom važnom konceptu bit će riječi i u sljedećim odjeljcima.

1.2.2 Nadzirano i nenadzirano učenje

Većina problema statističkog učenja može se svrstati u jednu od sljedećih dviju kategorija: nadzirano ili nenadzirano. Metode statističkog učenja navedene u prethodnom odjeljku spadaju u područje nadziranog učenja. Za svako opažanje prediktora $x_i, i = 1, ..., n$, postoji vezano opažanje odaziva y_i. Želimo prilagoditi model koji odaziv dovodi u vezu s prediktorima, s ciljem preciznog predviđanja odaziva za buduća opažanja (predviđanje), odnosno boljeg razumijevanja odnosa između odaziva i prediktora (inferencija). Mnoge klasične metode statističkog učenja poput linearne i logističke regresije, kao i noviji pristupi kao što su GAM, boosting te metoda potpornih vektora, spadaju u područje nadziranog učenja.

Nasuprot tome, nenadzirano učenje opisuje nešto problematičniju situaciju u kojoj za svako opažanje $i = 1, ..., n$ dobijemo vektor izmjerenih vrijednosti x_i, ali bez odgovarajućeg odaziva y_i. Ne možemo prilagoditi linearni regresijski model, jer nema varijable odaziva koja bi se predviđala. U takvoj situaciji na neki način tapkamo u mraku; kažemo da je riječ o nenadziranom učenju zbog odsustva varijable odaziva koja bi nadzirala našu analizu. Ono što možemo pokušati je utvrditi odnos između varijabli ili odnos između opažanja.

1.2.3 Mjerenje kvalitete prilagodbe modela

Kako bismo mogli ocijeniti rezultate neke metode statističkog učenja na danom skupu podataka, potreban nam je način na koji ćemo izmjeriti koliko se dobro predviđanja te metode poklapaju s opaženim podacima. Drugim riječima, želimo kvantificirati u kojoj mjeri je za neko opažanje predviđeni odaziv blizu stvarnog odaziva. U kontekstu regresije, jedna od najčešće korištenih mjera je srednjekvadratna pogreška (MSE). Za definiciju MSE, uvedimo prvo nekoliko oznaka i pojmova.

Pretpostavimo da imamo kvantitativni odaziv Y i p prediktora $X_1, X_2, ..., X_p$. Pretpostavljamo da između Y i $X = (X_1, X_2, ..., X_p)$ postoji neki odnos koji se može zapisati u obliku

$$Y = f(X) + \epsilon.$$

Ovdje je f fiksna ali nepoznata funkcija od $X_1, X_2, ..., X_p$, dok je ϵ slučajna pogreška s očekivanjem nula.

Neka je dano n opažanja $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$, pri čemu je $x_i = (x_{i1}, x_{i2}, ..., x_{ip})$ vektor izmjerenih vrijednosti prediktora, a y_i izmjereni odaziv u i-tom opažanju. Dodatno, označimo sa $\hat{f}$ procjenu za f dobivenu metodom statističkog učenja. Tada će $\hat{f}(x_i)$ biti predviđanje odaziva koje $\hat{f}$ daje za i-to opažanje. Sada možemo definirati MSE sa

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{f}(x_i))^2. \tag{1.1}$$

Ako je predviđeni odaziv blizu izmjerenog odaziva, MSE će biti mala, dok će u slučaju znatnog odstupanja između predviđenog i opaženog odaziva MSE biti velika.

Opažanja $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ koja smo pomoću neke metode koristili za određivanje procjene funkcije f zovemo još i skup podataka za učenje. Vidimo da se i MSE računa pomoću podataka za učenje koji su korišteni za prilagodbu modela. Prema tome, precizniji naziv bio bi "MSE na podacima za učenje" (engl. training MSE). No, općenito nas ne zanima koliko je neka metoda dobra na podacima za učenje. Puno nas više zanima preciznost predviđanja koja dobijemo kada metodu primijenimo na testnim podacima koji nikada prije nisu bili viđeni. Ilustrirajmo to sljedećim primjerom. Recimo da želimo razviti algoritam koji će predvidjeti visinu štete u osiguranju od nezgode na temelju karakteristika osiguranika (zanimanje, dob, mjesto prebivališta itd.). Metodu možemo trenirati na skupu 200 postojećih osiguranika, za koje imamo podatke o visini štete. No, nas ne zanima koliko dobro će ta metoda predvidjeti visinu štete za nekog od postojećih osiguranika - za njih čak imamo egzaktne podatke. Ono što nas zanima je predviđanje visine štete za novog osiguranika čije podatke metoda nikad nije vidjela. Tada taj algoritam možemo koristiti pri određivanju adekvatne premije koja će pokriti predviđeni iznos štete.

Drugim riječima, pretpostavimo da smo izvršili prilagodbu metode na opažanjima iz skupa za učenje $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, te da smo dobili procjenu $\hat{f}$. Tada možemo

izračunati $\hat{f}(x_1), \hat{f}(x_2), ..., \hat{f}(x_n)$. Ako su te vrijednosti približno jednake $y_1, y_2, ..., y_n$, onda će MSE na skupu za učenje, dana s (1.1), biti mala. No, nas zapravo ne zanima je li $\hat{f}(x_i) \approx y_i$, nego želimo znati je li $\hat{f}(x_0)$ približno jednako y_0, gdje je (x_0, y_0) testno opažanje koje nije bilo korišteno pri treniranju metode statističkog učenja. Želimo odabrati onu metodu koja će dati najmanju testnu MSE, nasuprot najmanjoj MSE na skupu za učenje.

1.2.4 Kompromis između pristranosti i varijance

Neka je (x_0, y_0) novo opažanje koje nije bilo korišteno pri treniranju metode. Označimo sa $\mathcal{T}$ skup podataka za učenje/treniranje. S obzirom da se metoda statističkog učenja prilagođava podacima za učenje, uočimo da ćemo za različite skupove podataka za učenje dobiti različite $\hat{f}$. To znači da je $\hat{f}$ slučajna varijabla, te ima smisla govoriti o očekivanju. Uz oznake uvedene u prethodnom odjeljku, definiramo očekivanu testnu MSE za danu vrijednost x_0 kao $\mathbb{E}_{\mathcal{T}}\left[\left(\hat{f}(x_0) - y_0\right)^2\right]$. To je prosječna vrijednost testne MSE koju bismo dobili kada bismo uzastopce procjenjivali f korištenjem mnogo skupova za učenje, i testirali svaku od dobivenih procjena $\hat{f}$ na x_0. Ukupna očekivana testna MSE može se dobiti tako da izračunamo prosječnu vrijednost $\mathbb{E}_{\mathcal{T}}\left[\left(\hat{f}(x_0) - y_0\right)^2\right]$ nad svim mogućim vrijednostima x_0 u testnom skupu. Još definirajmo pristranost (engl. bias) od $\hat{f}(x_0)$ sa $\text{Bias}(\hat{f}(x_0)) = \mathbb{E}_{\mathcal{T}}\left[\hat{f}(x_0)\right] - y_0$.

Tada se, za danu vrijednost x_0, očekivana testna MSE može prikazati kao zbroj varijance od $\hat{f}(x_0)$ i kvadrata pristranosti od $\hat{f}(x_0)$:

$$\mathbb{E}_{\mathcal{T}}\left[\left(\hat{f}(x_0) - y_0\right)^2\right] = \text{Var}_{\mathcal{T}}\left(\hat{f}(x_0)\right) + \left(\text{Bias}(\hat{f}(x_0))\right)^2. \tag{1.2}$$

Pokažimo to. Kako vrijedi $\text{Var}X = \mathbb{E}[X^2] - (\mathbb{E}X)^2$, odnosno $\mathbb{E}[X^2] = \text{Var}X + (\mathbb{E}X)^2$, to je

$$\mathbb{E}_{\mathcal{T}}\left[\left(\hat{f}(x_0) - y_0\right)^2\right] = \text{Var}_{\mathcal{T}}\left(\hat{f}(x_0) - y_0\right) + \left(\mathbb{E}_{\mathcal{T}}\left[\hat{f}(x_0) - y_0\right]\right)^2.$$

Koristimo svojstvo varijance: ako je a konstanta, onda je $\text{Var}(X + a) = \text{Var}X$:

$$= \text{Var}_{\mathcal{T}}\left(\hat{f}(x_0)\right) + \left(\mathbb{E}_{\mathcal{T}}\left[\hat{f}(x_0) - y_0\right]\right)^2.$$

Zbog linearnosti očekivanja vrijedi

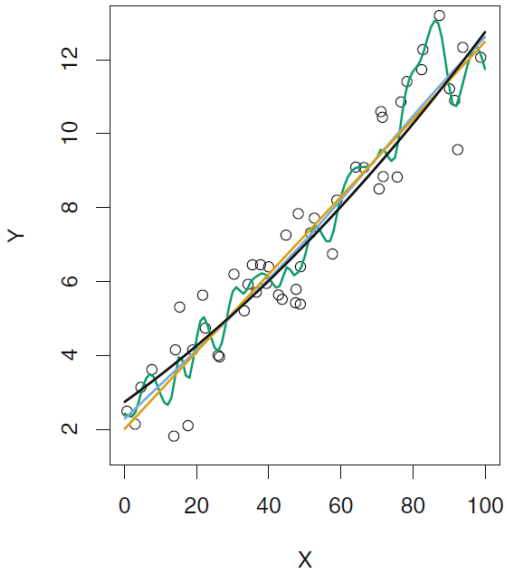
$$= \text{Var}_{\mathcal{T}}\left(\hat{f}(x_0)\right) + \left(\mathbb{E}_{\mathcal{T}}\left[\hat{f}(x_0)\right] - y_0\right)^2.$$

Konačno, iz definicije pristranosti slijedi

$$= \text{Var}_{\mathcal{T}}\left(\hat{f}(x_0)\right) + \left(\text{Bias}(\hat{f}(x_0))\right)^2.$$

Jednadžba (1.2) nam govori da, ako želimo minimizirati testnu pogrešku, onda moramo odabrati metodu statističkog učenja koja će istovremeno imati nisku varijancu i nisku pristranost.

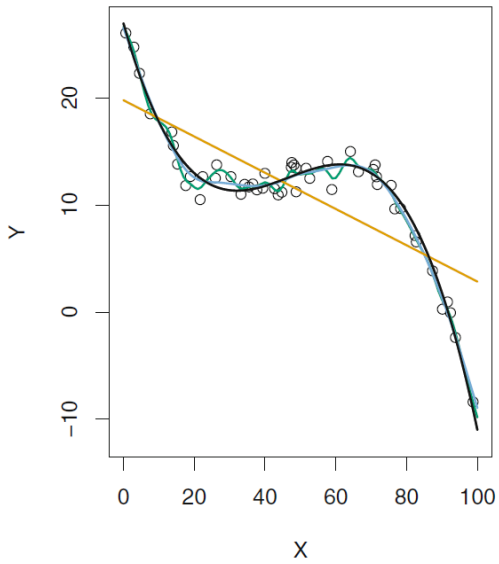
Objasnimo detaljnije pojmove varijance i pristranosti neke metode statističkog učenja. Varijanca nam govori koliko bi se $\hat{f}$ promijenila ako bismo ju procijenili pomoću nekog drugog skupa za učenje. Kao što je već rečeno, različiti skupovi podataka za učenje dovest će do različitih $\hat{f}$. U idealnom slučaju, procjena od f neće previše varirati od jednog do drugog skupa za učenje. No, ako neka metoda ima veliku varijancu, onda male promjene u podacima za učenje mogu dovesti do velikih promjena $\hat{f}$. Općenito će fleksibilnije statističke metode imati veću varijancu. Promotrimo zelenu i narančastu krivulju na slici 1.2. Zelena krivulja vrlo blisko slijedi opažanja. Ima veliku varijancu, jer promjena samo jedne od podatkovnih točaka može dovesti do znatne promjene $\hat{f}$. S druge strane, narančasti pravac linearne regresije je relativno nefleksibilan, te će promjena nekog opažanja vjerojatno dovesti do samo male promjene položaja pravca.



Slika 1.2: Podaci su simulirani iz f, koja je prikazana crnom krivuljom. Prikazane su tri procjene f: pravac linearne regresije (narančasta linija) te dvije prilagodbe kubičnih splajnova (plava i zelena krivulja). (Izvor: [3], Fig. 2.10)

Pristranost se odnosi na pogrešku koja nastane kada problem iz stvarnog života, koji može biti vrlo kompleksan, pokušamo opisati puno jednostavnijim modelom. Na primjer, pretpostavka linearne regresije je da postoji linearan odnos između Y i $X_1, X_2, ..., X_p$. Malo

je vjerojatno da će za ikoji problem iz stvarnog života zaista vrijediti takav linearan odnos, te će provedba linearne regresije vrlo vjerojatno rezultirati pristranošću kod procjene f. Na slici 1.3 je prava f nelinearna, te, ako bismo za modeliranje koristili linearnu regresiju, nikad neće biti moguće doći do preciznog modela, koliko god da smo opažanja za treniranje modela dobili. Drugim riječima, linearna regresija će u ovom primjeru dovesti do velike pristranosti. S druge strane, prava f sa slike 1.2 je vrlo blizu linearnoj funkciji, pa bi, ako nam je dostupno dovoljno podataka, linearna regresija trebala dovesti do precizne procjene. Općenito će fleksibilniji modeli imati manju pristranost.



Slika 1.3: Opis je sličan opisu slike 1.2, s razlikom da se sada koristi f koja je vrlo različita od linearne funkcije. U ovoj situaciji linearna regresija daje lošu prilagodbu podacima. (Izvor: [3], Fig. 2.11)

Generalno možemo reći da, što su fleksibilniji modeli koje koristimo, to će se varijanca povećavati, dok će se pristranost smanjivati. Odnos brzina kojima se te dvije veličine mijenjaju odredit će hoće li se testna MSE smanjivati ili povećavati.

Odnos između pristranosti, varijance i testne MSE koji je dan jednadžbom (1.2) zove se kompromis između pristranosti i varijance (engl. bias-variance trade-off). Da bi neka metoda statističkog učenja dala dobre rezultate na testnom skupu, potrebne su i niska varijanca i niska kvadrirana pristranost. Ovdje govorimo o kompromisu zbog toga što je jednostavno doći do metode s izuzetno niskom pristranošću, ali velikom varijancom (na primjer, provlačenjem krivulje kroz svako opažanje iz skupa za učenje), ili do metode s vrlo niskom varijancom, ali velikom pristranošću (provlačenjem horizontalnog pravca kroz

podatke). Izazov leži u nalaženju metode za koju će i varijanca i kvadrat pristranosti biti niski.

Poglavlje 2

Linearna regresija

U ovom poglavlju želimo ukratko opisati linearni regresijski model, s obzirom da se kasnija poglavlja temelje na proširenjima i modifikacijama tog modela.[1] Linearna regresija je jednostavan pristup za nadzirano učenje. Posebno, ona je koristan alat za predviđanje kvantitativnog odaziva. Linearna regresija je danas u vrlo širokoj primjeni, a istovremeno služi kao polazna točka za novije pristupe: mnogi sofisticirani pristupi statističkog učenja se mogu promatrati kao generalizacije i proširenja linearne regresije.

2.1 Jednostavna linearna regresija

Jednostavna linearna regresija, kao što i ime kaže, je vrlo jednostavan pristup za predviđanje kvantitativnog odaziva Y na temelju jedne prediktorske varijable X. Pretpostavlja se da između X i Y postoji približno linearan odnos. Taj linearan odnos možemo zapisati kao

$$Y \approx \beta_0 + \beta_1 X. \tag{2.1}$$

U jednadžbi (2.1) su β_0 i β_1 dvije nepoznate konstante koje predstavljaju odsječak na y-osi i nagib u linearnom modelu. β_0 i β_1 zajedno zovemo koeficijentima ili parametrima modela. Nakon što pomoću podataka za učenje dobijemo procjene parametara $\hat{\beta}_0$ i $\hat{\beta}_1$, možemo predvidjeti vrijednost Y na temelju neke određene vrijednosti X računajući

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_0 x,$$

gdje $\hat{y}$ označava predviđenu vrijednost od Y na temelju $X = x$. Ovdje koristimo simbol $\hat{}$ kako bismo označili procijenjenu vrijednost nepoznatog parametra, ili predviđenu vrijednost odaziva.

[1] Ovo poglavlje temelji se na [3], str. 59-73.

2.1.1 Procjena parametara

U praksi su β_0 i β_1 nepoznati. Zato, prije nego što možemo koristiti (2.1) za predviđanja, trebamo procijeniti parametre na temelju dostupnih podataka. Označimo s

$$(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$$

n parova opažanja, gdje se svaki par sastoji od jednog mjerenja X i jednog mjerenja Y. Želimo doći do procjena parametara $\hat{\beta}_0$ i $\hat{\beta}_1$ takvih da linearni model (2.1) dobro opisuje dostupne podatke, odnosno da je $y_i \approx \hat{\beta}_0 + \hat{\beta}_1 x_i$ za $i = 1, ..., n$. Drugim riječima, želimo naći odsječak na y-osi $\hat{\beta}_0$ i nagib $\hat{\beta}_1$ takve da odgovarajući pravac leži što bliže n podatkovnim točkama. Postoji više načina za mjerenje ove bliskosti. U najširoj primjeni je pristup koji uključuje minimizaciju kriterija najmanjih kvadrata, koji ćemo opisati u nastavku. Druge pristupe opisat ćemo u Poglavlju 4.

Neka je $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ predviđanje za Y temeljeno na i-toj vrijednosti od X. Tada $e_i = y_i - \hat{y}_i$ predstavlja i-ti rezidual: razliku između i-te opažene vrijednosti odaziva i i-te vrijednosti odaziva predviđene našim linearnim modelom. Definiramo sumu kvadrata reziduala (RSS, od engl. *residual sum of squares*) kao

$$\text{RSS} = e_1^2 + e_2^2 + ... + e_n^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2.2}$$

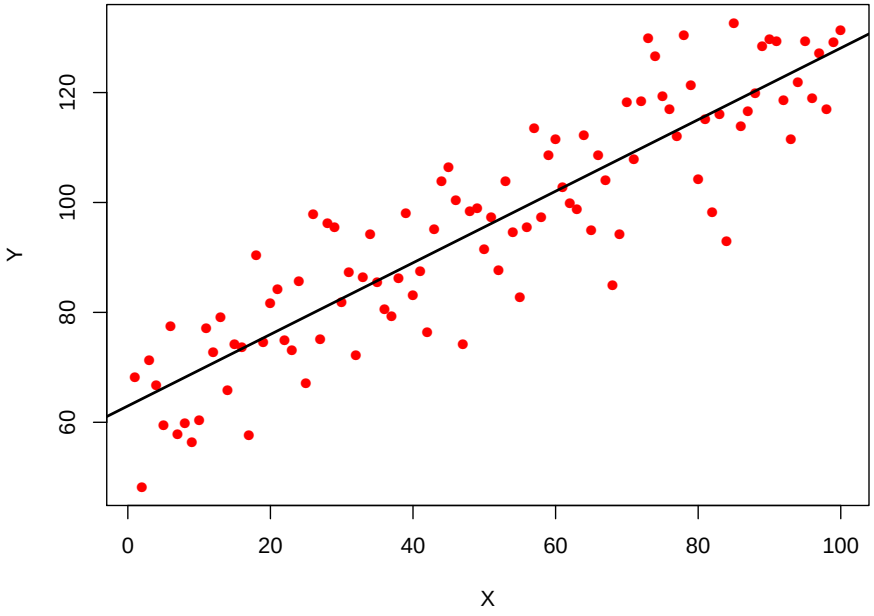
ili, ekvivalentno, kao

$$\text{RSS} = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + ... + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2.$$

Metoda najmanjih kvadrata bira $\hat{\beta}_0$ i $\hat{\beta}_1$ tako da se minimizira RSS. Može se pokazati da se minimum postiže za

$$\begin{aligned} \hat{\beta}_1 &= \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}, \\ \hat{\beta}_0 &= \bar{y} - \hat{\beta}_1 \bar{x}, \end{aligned} \tag{2.3}$$

gdje su $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$ i $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$ uzoračke sredine. Drugim riječima, (2.3) određuje procjene parametara metodom najmanjih kvadrata za jednostavnu linearnu regresiju. Slika 2.1 ilustrira jednostavnu linearnu regresiju Y na X.



Slika 2.1: Jednostavna linearna regresija, gdje su parametri procijenjeni metodom najmanjih kvadrata.

2.1.2 Procjena točnosti modela

Nakon što smo odredili parametre modela, prirodno je pitati se koliko dobro model opisuje podatke. Jedan od načina za procjenu kvalitete prilagodbe je korištenje R^2 statistike. R^2 statistika je omjer koji nam govori koliki postotak varijabilnosti u opažanjima je objašnjen modelom. Kako je riječ o omjeru, vrijednost će uvijek biti između 0 i 1, te neovisna o jedinicama u kojima je izražen Y. R^2 računamo pomoću formule

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}} \tag{2.4}$$

gdje je $\text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2$ ukupna suma kvadrata, dok je RSS definiran u (2.2). TSS mjeri ukupnu varijabilnost odaziva Y, te se može promatrati kao količina varijabilnosti u odazivu prije provođenja regresije. Nasuprot tome, RSS mjeri količinu varijabilnosti koja je neobjašnjena nakon provođenja regresije. Prema tome, TSS − RSS mjeri količinu

varijabilnosti odaziva koja je objašnjena (ili uklonjena) provođenjem regresije, te R^2 mjeri omjer varijabilnosti u Y koji se može objasniti pomoću X. R^2 statistika koja je blizu 1 upućuje na to da je veliki dio varijabilnosti odaziva objašnjen regresijom. Vrijednost koja je blizu 0 upućuje na to da regresija nije objasnila većinu varijabilnosti odaziva; to može biti zbog toga što je linearni model pogrešan, slučajna pogreška velika, ili oboje.

2.2 Višestruka linearna regresija

Jednostavni linearni regresijski model može se proširiti tako da, umjesto jednog, uključuje više prediktora. Ako pretpostavimo da imamo p prediktora, onda će višestruki linearni regresijski model biti oblika

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon, \tag{2.5}$$

gdje X_j predstavlja j-ti prediktor, β_j kvantificira vezu između te varijable i odaziva, dok je ϵ slučajna greška s očekivanjem nula. Možemo interpretirati β_j kao prosječan utjecaj na Y ako se X_j poveća za jednu jedinicu, uz držanje svih ostalih prediktora fiksnima.

2.2.1 Procjena parametara

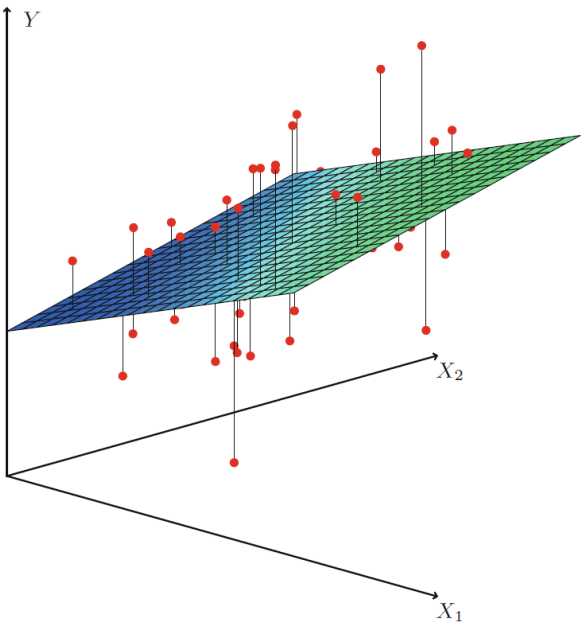
Kao i u slučaju jednostavne linearne regresije, regresijski koeficijenti $\beta_0, \beta_1, ..., \beta_p$ u (2.5) su nepoznati, te ih je potrebno procijeniti. Za dane procjene $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p$ možemo vršiti predviđanja koristeći formulu

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + ... + \hat{\beta}_p x_p.$$

Parametre procjenjujemo pomoću iste metode najmanjih kvadrata koju smo vidjeli u slučaju jednostavne linearne regresije. Biramo $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p$ tako da se minimizira suma kvadrata reziduala

$$\begin{aligned} \text{RSS} &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} \left(y_i - \hat{\beta}_0 - \sum_{j=1}^{p} \hat{\beta}_j x_{ij} \right)^2 . \end{aligned} \tag{2.6}$$

Vrijednosti $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_p$ koje minimiziraju (2.6) su procjene parametara višestruke linearne regresije metodom najmanjih kvadrata. Na slici 2.2 prikazan je slučaj sa $p = 2$ prediktora; tada regresijski pravac postaje ravnina.
Više o linearnoj regresiji može se pročitati u [1], str. 44-56, [2], str. 91-98, te [3], str. 59-126.



Slika 2.2: Višestruka linearna regresija, gdje su parametri procijenjeni metodom najmanjih kvadrata (Izvor: [3], Fig. 3.4)

Poglavlje 3

Unakrsna validacija

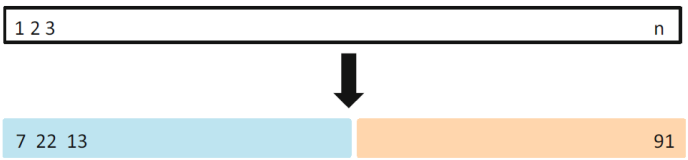
U Poglavlju 1 opisali smo razliku između testne pogreške i pogreške učenja.[1] Testna pogreška je prosječna pogreška koja nastane kada koristimo metodu statističkog učenja za predviđanje odaziva na novo opažanje - drugim riječima, na mjerenje koje nije bilo korišteno pri treniranju metode. Za dani skup podataka, korištenje određene metode statističkog učenja bit će preporučljivo ukoliko ta metoda rezultira malom testnom pogreškom. Testna pogreška može se izračunati ukoliko je dostupan odgovarajući testni skup. To, nažalost, često nije slučaj. Nasuprot tome, pogreška učenja može se lako izračunati primjenom metode statističkog učenja na opažanja korištenja pri njenom treniranju. No, pogreška učenja je često vrlo različita od testne pogreške, i može značajno podcijeniti testnu pogrešku.
U odsustvu vrlo velikog testnog skupa pomoću kojega bi se mogla neposredno procijeniti testna pogreška, možemo koristiti razne tehnike za procjenu te pogreške koristeći dostupne podatke iz skupa za učenje. Neke metode daju procjenu testne pogreške pomoću matematičke prilagodbe pogreške učenja. Ti pristupi bit će opisani u Poglavlju 4. U ovom poglavlju, s druge strane, promatramo klasu modela koji procjenjuju testnu pogrešku tako da prvo odvoje jedan podskup opažanja koji se neće koristiti pri treniranju, i onda dobivenu metodu statističkog učenja testiraju na tom odvojenom podskupu.

3.1 Metoda validacijskog skupa

Pretpostavimo da želimo procijeniti testnu pogrešku vezanu uz prilagodbu određene metode statističkog učenja skupu opažanja. Metoda validacijskog skupa, prikazana na slici 3.1, je vrlo jednostavna strategija za taj zadatak. Skup dostupnih opažanja se podijeli na dva dijela: na skup za učenje te na validacijski skup. Model se prilagođava podacima iz

[1] Ovo poglavlje temelji se na [3], str. 175-183.

skupa za učenje, te se zatim koristi za predviđanje odaziva za opažanja u validacijskom skupu. Rezultirajuća pogreška, koja se u slučaju numeričkog odaziva tipično mjeri uz pomoć MSE, dat će procjenu testne pogreške.



Slika 3.1: Shematski prikaz metode validacijskog skupa. Skup n opažanja je slučajno podijeljen na skup za učenje (lijevo, uključujući 7. 22. i 13. opažanje) te na validacijski skup (desno, uključujući, među ostalim, 91. opažanje). Metoda statističkog učenja se prilagođava skupu za učenje, dok se rezultat vrednuje na validacijskom skupu. (Izvor: [3], Fig. 5.1)

Metoda validacijskog skupa je konceptualno jednostavna i može se lako implementirati. No, ima dva potencijalna nedostatka:

1. Procjena testne pogreške pomoću validacijskog skupa može znatno varirati, u ovisnosti o tome koja su točno opažanja bila uključena u skup za učenje, a koja u validacijski skup.

2. U metodi validacijskog skupa, model se prilagođava samo podskupu podataka - opažanjima koja su uključena u skup za učenje (a ne u validacijski skup). S obzirom da statističke metode obično daju lošije rezultate ako su trenirane na manjem skupu podataka, to upućuje na to da pogreška na validacijskom skupu može precijeniti testnu pogrešku modela koji je prilagođen svim dostupnim podacima.

U nastavku ćemo predstaviti unakrsnu validaciju (engl. cross-validation), koja je profinjenje metode validacijskog skupa s ciljem da se riješe dva navedena nedostatka.

3.2 Pojedinačna unakrsna validacija (LOOCV)

Pojedinačna unakrsna validacija (engl. leave-one-out cross-validation; LOOCV) je blisko povezana s metodom validacijskog skupa iz prethodnog potpoglavlja, no pokušava riješiti nedostatke te metode.

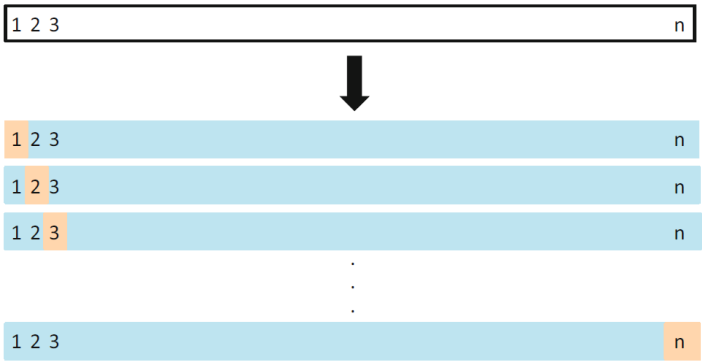
Kao i metoda validacijskog skupa, LOOCV uključuje podjelu skupa opažanja na dva dijela. No, umjesto podjele na dva podjednako velika podskupa, validacijski skup se sastoji

od samo jednog opažanja (x_1, y_1), dok preostala opažanja $\{(x_2, y_2), ..., (x_n, y_n)\}$ čine skup za učenje. Metoda statističkog učenja se prilagođava $n-1$ opažanjima iz skupa za učenje, dok se predviđanje $\hat{y}_1$ vrši za isključeno opažanje, koristeći njegovu vrijednost x_1. S obzirom da (x_1, y_1) nije bio korišten u postupku prilagodbe, $\text{MSE}_1 = (y_1 - \hat{y}_1)^2$ će dati približno nepristranu procjenu testne pogreške. No, iako je MSE_1 nepristrana za testnu pogrešku, riječ je o lošoj procjeni koja je vrlo varijabilna, jer se temelji na samo jednom opažanju (x_1, y_1).

Postupak možemo ponoviti tako da odaberemo (x_2, y_2) za validacijski skup, metodu statističkog učenja treniramo na preostalih $n-1$ opažanja $\{(x_1, y_1), (x_3, y_3)..., (x_n, y_n)\}$, te računamo $\text{MSE}_2 = (y_2 - \hat{y}_2)^2$. Ponavljanje ovog postupka n puta dat će nam n kvadratnih pogrešaka $\text{MSE}_1, ..., \text{MSE}_n$. Procjena testne MSE pomoću LOOCV metode je prosjek tih n procjena:

$$\text{CV}_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \text{MSE}_i. \tag{3.1}$$

Shematski prikaz LOOCV metode dan je na slici 3.2.



Slika 3.2: Shematski prikaz LOOCV metode. Skup n opažanja se opetovano dijeli na validacijski skup, koji se sastoji od jednog opažanja, te na skup za učenje, koji se sastoji od svih preostalih podataka. Testna pogreška se procjenjuje računanjem prosjeka n dobivenih MSE. (Izvor: [3], Fig. 5.3)

LOOCV metoda ima nekoliko značajnih prednosti nad metodom validacijskog skupa. Prvo, znatno je manje pristrana. U LOOCV metodi opetovano prilagođavamo metodu statističkog učenja koristeći skupove za učenje koji se sastoje od $n-1$ opažanja, što je skoro kao veličina cijelog skupa. To je različito od metode validacijskog skupa, gdje se skup za učenje tipično sastoji od polovice originalnih podataka. Posljedica toga je da LOOCV metoda obično neće precijeniti testnu pogrešku u mjeri u kojoj to čini metoda

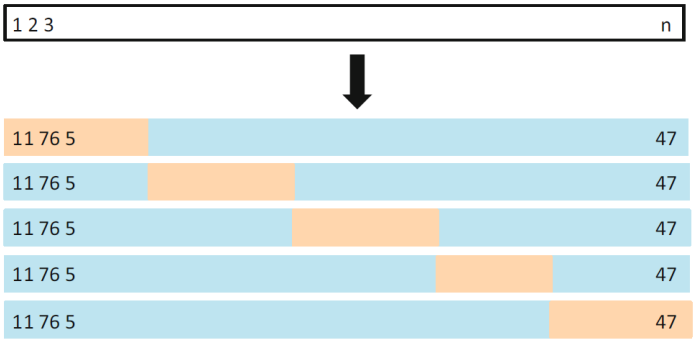
validacijskog skupa. Drugo, ako ponavljamo metodu validacijskog skupa više puta, dobit ćemo drugačije rezultate zbog slučajnosti pri podjeli na skup za učenje i validacijski skup. Nasuprot tome, kod ponavljanja LOOCV metode više puta uvijek ćemo dobiti isti rezultat, jer kod podjele na skup za učenje i validacijski skup nema slučajnosti.

3.3 *k*-struka unakrsna validacija

Alternativa za LOOCV je k-struka unakrsna validacija. U toj metodi se skup opažanja na slučajan način dijeli na k podskupova podjednake veličine. Prvi podskup se tretira kao validacijski skup, dok se metoda statističkog učenja prilagođava preostalim $k-1$ podskupovima. Zatim se srednjekvadratna pogreška, MSE_1, računa na opažanjima iz izdvojenog podskupa. Ovaj postupak ponavlja se k puta: svaki puta se drugi podskup opažanja tretira kao validacijski skup. Ovaj proces rezultira u k procjena testne pogreške $\text{MSE}_1, \text{MSE}_2, ..., \text{MSE}_k$. Računanjem prosjeka tih vrijednosti dolazimo do procjene metodom k-struke unakrsne validacije:

$$\text{CV}_{(k)} = \frac{1}{k}\sum_{i=1}^{k}\text{MSE}_i. \tag{3.2}$$

Metoda k-struke unakrsne validacije ilustrirana je na slici 3.3.



Slika 3.3: Shematski prikaz peterostruke unakrsne validacije. Skup n opažanja je na slučajan način podijeljen na pet disjunktnih podskupova. Svaka od ovih petina podataka služi kao validacijski skup, dok ostatak podataka služi kao skup za učenje. Testna pogreška se procjenjuje računanjem prosjeka pet dobivenih procjena za MSE. (Izvor: [3], Fig. 5.5)

Očito je da je LOOCV poseban slučaj k-struke unakrsne validacije za $k = n$. U praksi se k-struka unakrsna validacija često provodi za $k = 5$ ili $k = 10$. Jedna od očiglednih

prednosti korištenja $k = 5$ ili $k = 10$ umjesto $k = n$ je računska. LOOCV metoda zahtijeva prilagodbu metode statističkog učenja podacima n puta, što može biti računski vrlo zahtjevno, pogotovo ako je broj opažanja n vrlo velik. Nasuprot tome, za provedbu 10-struke unakrsne validacije nužno je provesti prilagodbu metode učenja podacima samo deset puta, što može biti znatno lakše provedivo.

Poglavlje 4

Odabir i regularizacija linearnih modela

U kontekstu regresije, standardni linearni model

$$Y = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p + \epsilon \tag{4.1}$$

se često koristi za opis odnosa između odaziva Y i nezavisnih varijabli $X_1, X_2, ...X_p$.[1] U Poglavlju 2 smo vidjeli da se taj model tipično prilagođava podacima pomoću metode najmanjih kvadrata. U ovom poglavlju promatramo načine na koje se linearni model može poboljšati, tako da se prilagodba pomoću najmanjih kvadrata zamijeni alternativnim metodama. Korištenje alternativnih metoda može rezultirati u preciznijem predviđanju i boljoj interpretabilnosti modela.

- Preciznost predviđanja: Uz pretpostavku da je odnos odaziva i prediktora otprilike linearan, procjene metodom najmanjih kvadrata bit će nepristrane. Ako je $n \gg p$, tj. ako je broj opažanja n znatno veći od broja nezavisnih varijabli p, onda će procjene metodom najmanjih kvadrata često imati nisku varijancu, te će davati dobre rezultate za testna opažanja. No, ako n nije znatno veći od p, onda će u prilagodbi metodom najmanjih kvadrata biti dosta varijabilnosti, što rezultira u pretreniranosti (engl. overfitting) i, posljedično, lošim predviđanjima za buduća opažanja koja nisu korištena u treniranju modela. Ako je pak $p > n$, onda više ne postoji jedinstvena procjena parametara metodom najmanjih kvadrata, te se ta metoda ne može koristiti. Ograničavanjem ili smanjenjem procijenjenih parametara možemo značajno smanjiti varijancu modela uz trošak neznatnog povećanja pristranosti. To može dovesti do značajnog poboljšanja preciznosti predviđanja odaziva za opažanja koja nisu bila korištena u treniranju modela.
- Interpretabilnost modela: Često je slučaj da neke ili čak mnoge nezavisne varijable koje se koriste u višestrukom regresijskom modelu uopće nisu povezane s odazivom.

[1] Ovo poglavlje temelji se na [3], str. 203-238.

Uključivanje tih nerelevantnih varijabli dovodi do nepotrebne kompleksnosti dobivenog modela. Ako te varijable uklonimo, tj. ako odgovarajuće procjene parametara postavimo na nulu, možemo dobiti model koji je puno lakše interpretirati. Vrlo je mala vjerojatnost da će metoda najmanjih kvadrata dati procjene parametara koje su baš jednake nuli. U ovom poglavlju ćemo razmotriti pristupe koji automatski vrše odabir značajki (engl. feature selection) odnosno odabir varijabli; drugim riječima, koje isključuju nerelevantne varijable iz višestrukog regresijskog modela.

Uz metodu najmanjih kvadrata, postoje brojni alternativni pristupi, i klasični i suvremeni, za prilagodbu modela 4.1. U ovom ćemo poglavlju razmotriti dvije važne klase metoda.

- *Odabir podskupa prediktora.* Ovaj pristup uključuje prepoznavanje podskupa od p prediktora za koji vjerujemo da je povezan s odazivom. Zatim prilagođavamo model korištenjem metode najmanjih kvadrata na reduciranom skupu varijabli.

- *Smanjenje koeficijenata.* Ovaj pristup uključuje prilagodbu modela uz korištenje svih p prediktora. No, u odnosu na procjenu metodom najmanjih kvadrata, koeficijenti se smanjuju prema nuli. Posljedica ovog smanjenja koeficijenata (koje se još zove i regularizacija) je redukcija varijance. U ovisnosti o vrsti smanjenja koja se provodi, procjena nekih koeficijenata može biti točno nula. Prema tome, metode smanjenja koeficijenata mogu vršiti i odabir varijabli.

4.1 Odabir podskupa prediktora

U ovom potpoglavlju razmatramo nekoliko metoda za odabir podskupa prediktora, uključujući metodu odabira najboljeg podskupa te metodu postupnog odabira.

4.1.1 Odabir najboljeg podskupa

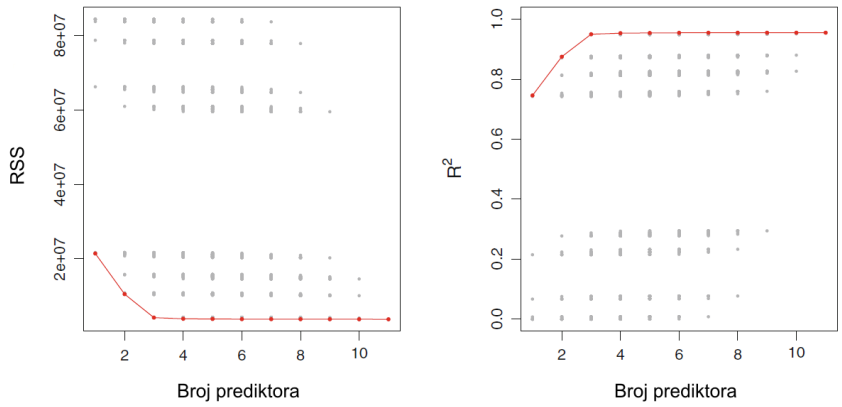
Da bismo proveli odabir najboljeg podskupa, za svaku moguću kombinaciju p prediktora vršimo zasebnu prilagodbu metodom najmanjih kvadrata. Drugim riječima, prilagođavamo svih p modela koji se sastoje od točno jednog prediktora, $\binom{p}{2} = \frac{(p-1)p}{2}$ modela koji se sastoje od točno dva prediktora itd. Zatim promotrimo sve dobivene modele, s ciljem da odaberemo "najbolji".

Problem odabira "najboljeg modela" između 2^p mogućnosti koje se promatraju u metodi odabira najboljeg podskupa nije trivijalan. Postupak je opisan u algoritmu 4.1.

U algoritmu 4.1, u koraku 2 za svaku veličinu podskupa identificiramo najbolji model (na podacima za učenje), kako bismo smanjili problem odabira jednog od 2^p na odabir jednog od $p + 1$ modela. Na slici 4.1, ti modeli čine granicu označenu crvenom bojom.

Algoritam 4.1 Odabir najboljeg podskupa

1. Označimo sa $\mathcal{M}_0$ nulti model, koji ne sadrži niti jedan prediktor. Ovaj model za svako opažanje jednostavno predviđa uzoračko očekivanje.
2. Za $k = 1, 2, ..., p$:
   a) Prilagodi svih $\binom{p}{k}$ modela koji se sastoje od točno k prediktora.
   b) Odaberi najbolji od tih $\binom{p}{k}$ modela, i nazovi ga $\mathcal{M}_k$. Ovdje "najbolji" definiramo kao model s najmanjom RSS, odnosno najvećim R^2.
3. Odaberi najbolji model među $\mathcal{M}_0, ..., \mathcal{M}_p$ koristeći unakrsnu validaciju, C_p (AIC), BIC ili prilagođeni R^2. (Za njihove definicije, vidi odjeljak 4.1.3.)



Slika 4.1: Prikaz RSS i R^2 za svaki mogući model koji se sastoji od jednog podskupa 11 prediktora iz simuliranog skupa podataka. (Izvor: [3], Fig. 6.1)

Kako bismo došli do najboljeg, trebamo odabrati jedan od tih $p+1$ modela. To moramo napraviti pažljivo s obzirom da, što je veći broj uključenih svojstava, RSS tih $p+1$ modela monotono opada, a R^2 monotono raste. Prema tome, ako koristimo RSS i R^2 kao kriterije za odabir najboljeg modela, uvijek ćemo završiti s modelom koji uključuje sve varijable. Ovdje je problem što nizak RSS odnosno visok R^2 upućuje na model s niskom pogreškom učenja, dok nas zanima model s niskom testnom pogreškom. (Obično će pogreška učenja biti znatno manja od testne pogreške, te mala pogreška učenja nikako ne garantira malu testnu pogrešku.) Iz tog razloga za odabir među $\mathcal{M}_0, \mathcal{M}_1, ..., \mathcal{M}_p$ u koraku 3 koristimo unakrsnu validaciju, C_p, BIC ili prilagođeni R^2. Ti pristupi bit će opisani u odjeljku 4.1.3.

Na slici 4.1 prikazana je primjena metode odabira najboljeg podskupa. Svaka točka odgovara jednom modelu koji je prilagođen metodom najmanjih kvadrata uz korištenje drugog podskupa 11 prediktora iz simuliranog skupa podataka. Za svaki model prikazane su vrijednosti RSS i R^2 statistika kao funkcije broja nezavisnih varijabli. Crvene krivulje povezuju najbolje modele za svaku veličinu podskupa. Slika pokazuje, kao što i očekujemo, da su ove vrijednosti sve bolje što više varijabli uključimo u model. No, također vidimo da, počevši od modela s tri varijable, nema značajnog poboljšanja u RSS i R^2 pri dodavanju novih prediktora.

Iako je metoda odabira najboljeg podskupa jednostavna i konceptualno privlačna, računalna provedba može biti problematična. Što više p raste, broj mogućih modela koje je potrebno razmotriti se eksponencijalno povećava. Općenito će postojati 2^p modela koji uključuju podskupove p prediktora. Ako je $p = 10$, onda postoji 1024 mogućnosti koje je potrebno razmotriti; ako je $p = 20$, onda će postojati preko milijun mogućnosti. Posljedično, metoda odabira najboljeg podskupa postat će računalno nepraktična kada je p veći od otprilike 40, čak i uz vrlo brza suvremena računala. Postoje računske prečice, no i one su od ograničene koristi kako se p povećava. Dodatno, one se mogu primijeniti samo u slučaju linearne regresije metodom najmanjih kvadrata. U nastavku ćemo prikazati računalno učinkovitije alternative metodi odabira najboljeg podskupa.

4.1.2 Postupni odabir

Iz računskih razloga, metoda odabira najboljeg podskupa nije primjenjiva za vrlo velike p. Osim toga, kod te metode mogu se pri velikim vrijednostima p javiti i statistički problemi. Što je veći prostor pretraživanja, to je veća vjerojatnost da ćemo naći modele koji se dobro ponašaju na podacima za učenje, iako će dati vrlo loša predviđanja za buduće podatke. Iz tog razloga, veliki prostor pretraživanja može dovesti do pretreniranosti i visoke varijance procjena parametara.

Iz oba ova razloga će postupne metode, koje pretražuju znatno manji skup modela, biti privlačne alternative metodi odabira najboljeg podskupa.

Postupni odabir unaprijed

Postupni odabir unaprijed (engl. forward stepwise selection) je računalno učinkovita alternativa metodi odabira najboljeg podskupa. Dok metoda odabira najboljeg podskupa promatra svih 2^p mogućih modela koji sadrže podskupove p prediktora, metoda postupnog odabira unaprijed promatra znatno manji skup modela. Metoda kreće od modela koji ne sadrži niti jedan prediktor, te zatim dodaje prediktore u model, jedan po jedan, sve dok se u model ne uključe svi prediktori. Posebno, u svakom koraku će u model biti dodana ona nezavisna varijabla koja daje najveće dodatno poboljšanje prilagodbe. Metoda postupnog odabira unaprijed formalnije je opisana u algoritmu 4.2.

Algoritam 4.2 Postupni odabir unaprijed

1. Označimo sa $\mathcal{M}_0$ nulti model, koji ne sadrži niti jedan prediktor.
2. Za $k = 0, 1, ..., p-1$:
   a) Promotri svih $p-k$ modela koji proširuju prediktore u $\mathcal{M}_k$ jednim dodatnim prediktorom.
   b) Odaberi najbolji od tih $p-k$ modela, i nazovi ga $\mathcal{M}_{k+1}$. Ovdje "najbolji" definiramo kao model s najmanjom RSS, odnosno najvećim R^2.
3. Odaberi najbolji model među $\mathcal{M}_0, ..., \mathcal{M}_p$ koristeći unakrsnu validaciju, C_p (AIC), BIC ili prilagođeni R^2. (Za njihove definicije, vidi odjeljak 4.1.3.)

Za razliku od metode odabira najboljeg podskupa, koja je uključivala 2^p modela, metoda postupnog odabira unaprijed uključuje prilagodbu jednog nultog modela, te $p-k$ modela u k-toj iteraciji, za $k = 0, ..., p-1$. Riječ je o ukupno $1 + \sum_{k=0}^{p-1}(p-k) = 1 + \frac{p(p+1)}{2}$ modela. To je značajna razlika: za $p = 20$, metoda odabira najboljeg podskupa uključuje prilagodbu 1.048.576 modela, dok metoda postupnog odabira unaprijed uključuje prilagodbu samo 211 modela.

U koraku 2 b) algoritma 4.2, za dani k moramo prepoznati najbolji među $p-k$ modela koji proširuju $\mathcal{M}_k$ jednim dodatnim prediktorom. To možemo jednostavno postići tako da odaberemo model s najmanjom RSS, odnosno najvećim R^2. No, u koraku 3 moramo odabrati najbolji model iz skupa modela koji imaju različit broj varijabli. To je zahtjevnije, te će biti raspravljeno u odjeljku 4.1.3.

Prednost metode postupnog odabira unaprijed nad metodom odabira najboljeg podskupa je jasna. No, iako metoda postupnog odabira unaprijed daje dobre rezultate u praksi, nije garantirano da će pronaći najbolji model među svih 2^p modela koji sadrže podskupove p prediktora. Na primjer, pretpostavimo da u danom skupu podataka s $p = 3$ prediktora, najbolji model s jednom varijablom sadrži X_1, dok najbolji model s dvije varijable sadrži X_2 i X_3. Tada metoda postupnog odabira unaprijed neće odabrati najbolji model s dvije varijable, jer će $\mathcal{M}_1$ sadržavati X_1, pa će i u $\mathcal{M}_2$ morati biti uključen X_1 s jednom dodatnom varijablom.

Postupni odabir unatrag

Kao i postupni odabir unaprijed, postupni odabir unatrag (engl. backward stepwise selection) je učinkovita alternativa metodi odabira najboljeg podskupa. No, za razliku od postupnog odabira unaprijed, ova metoda kreće od punog modela dobivenog metodom

najmanjih kvadrata koji uključuje svih p prediktora, te zatim iterativno uklanja jedan po jedan najmanje koristan prediktor. Detalji su dani u algoritmu 4.3.

Algoritam 4.3 Postupni odabir unatrag

1. Označimo sa $\mathcal{M}_p$ puni model, koji uključuje svih p prediktora.
2. Za $k = p, p-1, ..., 1$:
   a) Promotri svih k modela koji sadrže sve prediktore u $\mathcal{M}_k$ osim jednoga, tj. koji se sastoje od $k-1$ prediktora.
   b) Odaberi najbolji od tih k modela, i nazovi ga $\mathcal{M}_{k-1}$. Ovdje "najbolji" definiramo kao model s najmanjom RSS, odnosno najvećim R^2.
3. Odaberi najbolji model među $\mathcal{M}_0, ..., \mathcal{M}_p$ koristeći unakrsnu validaciju, C_p (AIC), BIC ili prilagođeni R^2. (Za njihove definicije, vidi odjeljak 4.1.3.)

Poput postupnog odabira unaprijed, metoda postupnog odabira unatrag pretražuje samo $1 + \frac{p(p+1)}{2}$ modela, te se može primijeniti u slučajevima gdje je p prevelik za korištenje metode odabira najboljeg podskupa. Slično kao kod postupnog odabira unaprijed, nije garantirano da će metoda postupnog odabira unatrag dati najbolji model koji se sastoji od podskupa p prediktora.

Hibridni pristupi

Metode odabira najboljeg podskupa, postupnog odabira unaprijed i postupnog odabira unatrag će općenito dati slične no ne i identične modele. Uz njih, dostupne su i hibridne verzije metoda postupnog odabira unaprijed i unatrag. U hibridnoj verziji postupnog odabira unaprijed, varijable se postupno dodaju u model, kao i u uobičajenoj verziji. No, nakon dodavanja svake nove varijable, metoda može iz modela izbaciti one varijable koje ne daju doprinos prilagodbi modela. Ovakav pristup se pokušava približiti metodi odabira najboljeg podskupa, uz zadržavanje računskih prednosti metode postupnog odabira unaprijed.

4.1.3 Odabir optimalnog modela

Metode odabira najboljeg podskupa, postupnog odabira unaprijed i postupnog odabira unatrag rezultirat će skupom modela $\mathcal{M}_0, \mathcal{M}_1, ..., \mathcal{M}_p$, pri čemu se svaki od tih modela sastoji od jednog podskupa p prediktora. Kako bismo implementirali ove metode, potreban nam je način na koji ćemo odrediti koji od tih modela je "najbolji". Kao što smo spomenuli u odjeljku 4.1.1, model koji se sastoji od svih prediktora će uvijek imati najmanju RSS i

najveći R^2, s obzirom da su te veličine povezane s pogreškom učenja. No, mi želimo odabrati model s najmanjom testnom pogreškom. Kao što je već navedeno, pogreška učenja može biti loša procjena testne pogreške. Iz tog razloga RSS i R^2 nisu prikladni za odabir najboljeg modela iz skupa modela s različitim brojem prediktora.

Kako bismo odabrali najbolji model u odnosu na testnu pogrešku, potrebno je testnu pogrešku procijeniti. Postoje dva uobičajena pristupa:

1. Testnu pogrešku možemo indirektno procijeniti tako da prilagodimo pogrešku učenja, kako bi u obzir uzeli pristranost modela zbog pretreniranosti.

2. Testnu pogrešku možemo direktno procijeniti korištenjem metode validacijskog skupa ili metode unakrsne validacije, koje su opisane u Poglavlju 3.

U nastavku ćemo razmotriti oba ova pristupa.

C_p, AIC, BIC i prilagođeni R^2

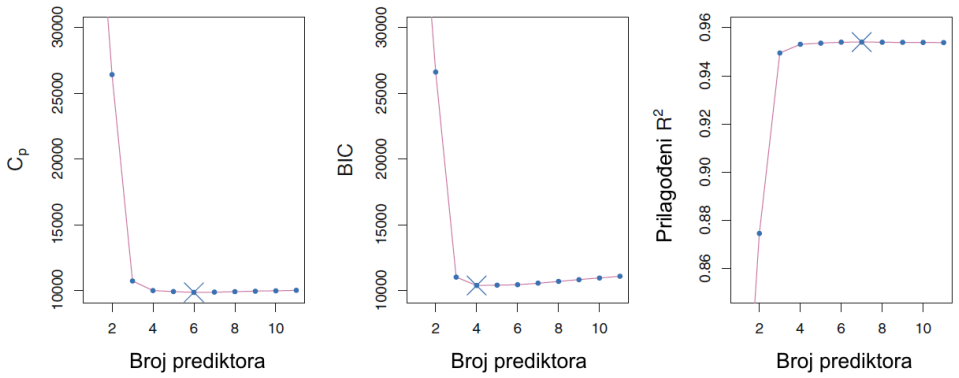
U Poglavlju 1 smo pokazali da će MSE na skupu za učenje općenito podcijeniti testnu MSE. (Primijetimo da je MSE = RSS/n.) Razlog za to je sljedeći: kada prilagođavamo model podacima iz skupa za učenje koristeći metodu najmanjih kvadrata, parametre regresije određujemo upravo tako da RSS učenja (ali ne i testna RSS) bude najmanja moguća. Posebno, pogreška učenja će se smanjivati što više varijabli uključimo u model, no to ne mora vrijediti i za testnu pogrešku. Prema tome, RSS i R^2 na skupu za učenje se ne mogu koristiti za odabir među modelima s različitim brojem varijabli.

Ipak, postoji nekoliko pristupa za prilagodbu pogreške učenja s obzirom na veličinu modela, koji se mogu koristiti za odabir među modelima s različitim brojem varijabli. Promotrit ćemo četiri takva pristupa: C_p, Akaikeov informacijski kriterij (AIC, engl. Akaike information criterion), bayesovski informacijski kriterij (BIC, engl. Bayesian information criterion) te prilagođeni R^2. Slika 4.2 prikazuje C_p, BIC i prilagođeni R^2 za najbolji model svake veličine koji je dobiven primjenom metode odabira najboljeg podskupa na simuliranom skupu podataka.

Za model prilagođen metodom najmanjih kvadrata koji se sastoji od d prediktora, C_p procjena testne MSE računa se pomoću jednadžbe

$$C_p = \frac{1}{n}(\text{RSS} + 2d\hat{\sigma}^2), \tag{4.2}$$

gdje je $\hat{\sigma}^2$ procjena varijance slučajne greške ϵ povezane s mjerenjem odaziva u (4.1). $\hat{\sigma}^2$ se tipično procjenjuje pomoću punog modela koji sadrži sve prediktore. U suštini, C_p statistika na RSS dodaje kaznu u iznosu $2d\hat{\sigma}^2$ kako bi se uzela u obzir činjenica da pogreška učenja obično podcjenjuje testnu pogrešku. Očito je da se kazna povećava s brojem prediktora - namjera je prilagoditi odgovarajuće smanjenje RSS na skupu za učenje



Slika 4.2: Prikaz C_p, BIC i prilagođenog R^2 za najbolji model svake veličine na simuliranom skupu podataka. C_p i BIC su procjene testne MSE. Na srednjem grafu vidimo da se procjena testne pogreške pomoću BIC povećava nakon što uključimo četiri varijable. Druga dva grafa su relativno ravna nakon što se uključe četiri varijable. (Izvor: [3], Fig. 6.2)

kako broj prediktora raste. Može se pokazati da, ako je $\hat{\sigma}^2$ nepristrani procjenitelj za σ^2 u (4.2), onda je C_p nepristrani procjenitelj za testnu MSE. Posljedica toga je da će C_p statistika često poprimati male vrijednosti za modele s niskom testnom pogreškom. Zato ćemo pri određivanju najboljeg modela u nekom skupu odabrati onaj s najnižom vrijednosti C_p. Na slici 4.2 je na temelju C_p statistike odabran model sa 6 prediktora.

AIC je definiran za široku klasu modela koji su prilagođeni metodom maksimalne vjerodostojnosti. U slučaju kada su greške u (4.1) normalno distribuirane, metoda maksimalne vjerodostojnosti je isto što i metoda najmanjih kvadrata. U tom slučaju je AIC dan sa

$$\text{AIC} = \frac{1}{n\hat{\sigma}^2}(\text{RSS} + 2d\hat{\sigma}^2),$$

gdje smo zbog jednostavnosti izostavili jednu aditivnu konstantu. Vidimo da su, u slučaju modela prilagođenih metodom najmanjih kvadrata, AIC i C_p proporcionalni, te je iz tog razloga na slici 4.2 prikazan samo C_p.

BIC je izveden na temelju bayesovskog gledišta, no u konačnici izgleda slično kao C_p (i AIC). Za model prilagođen metodom najmanjih kvadrata s d prediktora, BIC je (do na konstante koje nam nisu relevantne) dan sa:

$$\text{BIC} = \frac{1}{n\hat{\sigma}^2}(\text{RSS} + \log(n)d\hat{\sigma}^2). \tag{4.3}$$

Poput C_p-a, i BIC će težiti poprimanju malih vrijednosti za modele s niskom testnom pogreškom, te ćemo općenito odabrati model s najmanjom vrijednosti BIC-a. Uočimo da

je $2d\hat{\sigma}^2$ iz formule za C_p zamijenjen izrazom $\log(n)d\hat{\sigma}^2$ u formuli za BIC, pri čemu je n broj opažanja. Zbog toga što je $\log n > 2$ za $n > 7$, BIC statistika će općenito imati veću kaznu za modele s velikim brojem varijabli, što će dovesti do odabira manjih modela nego u slučaju kada koristimo C_p. Da je to zaista slučaj vidimo na slici 4.2: BIC je odabrao model sa samo četiri prediktora. U ovom slučaju su krivulje relativno ravne, te se čini da nema velike razlike u točnosti između modela s četiri i modela sa šest varijabli.

Korištenje prilagođene R^2 statistike je još jedan popularan pristup za odabir među modelima s različitim brojem varijabli. Kao što je rečeno u Poglavlju 2, uobičajeni R^2 se definira kao $1 - \text{RSS}/\text{TSS}$, gdje je $\text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2$ ukupna suma kvadrata za odaziv. Zbog toga što se RSS uvijek smanjuje što više varijabli dodajemo u model, to će se R^2 uvijek povećavati s dodavanjem novih varijabli. Za model prilagođen metodom najmanjih kvadrata, prilagođena R^2 statistika se računa kao

$$\text{prilagođeni } R^2 = 1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)}. \tag{4.4}$$

Za razliku od C_p-a, AIC-a i BIC-a, kod kojih mala vrijednost upućuje na model s niskom testnom pogreškom, u slučaju prilagođenog R^2 će velika vrijednost upućivati na model s niskom testnom pogreškom. Maksimizacija prilagođenog R^2 je ekvivalentna minimizaciji $\frac{\text{RSS}}{n-d-1}$. Dok se RSS uvijek smanjuje s povećanjem broja varijabli u modelu, $\frac{\text{RSS}}{n-d-1}$ se zbog prisutnosti d u nazivniku može ili smanjivati ili povećavati.

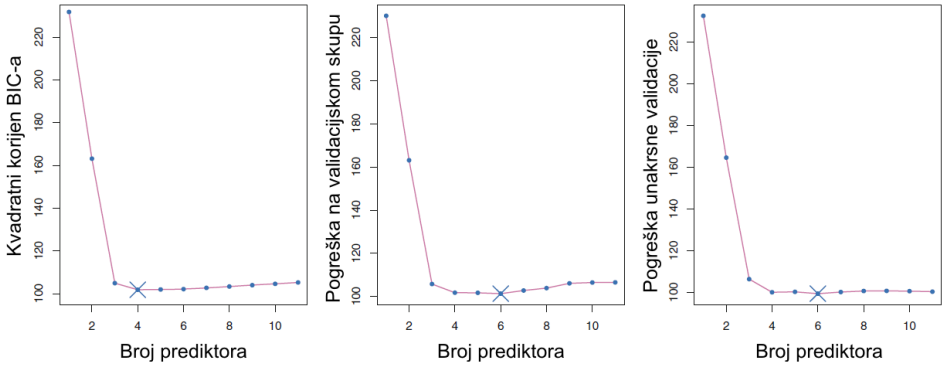
Intuicija u pozadini prilagođenog R^2 je da, jednom kada su sve relevantne varijable uključene u model, dodavanje dodatnih varijabli "šuma" će dovesti do vrlo malog smanjenja RSS. Kako dodavanje varijabli šuma vodi do povećanja d, i $\frac{\text{RSS}}{n-d-1}$ će se povećati, te će se posljedično prilagođeni R^2 smanjiti. Prema tome, model s najvećim prilagođenim R^2 će teoretski imati samo relevantne varijable, a neće imati varijable šuma. Za razliku od R^2 statistike, kod prilagođenog R^2 se plaća cijena uključivanja nepotrebnih varijabli u model. Na slici 4.2 vidimo da je korištenje prilagođene R^2 statistike dovelo do modela sa sedam varijabli, odnosno jednom varijablom više nego u slučaju gdje su korišteni C_p i AIC.

Validacija i unakrsna validacija

Uz upravo opisane pristupe, testnu pogrešku možemo direktno procijeniti korištenjem metode validacijskog skupa i metoda unakrsne validacije prikazane u Poglavlju 3. Za svaki model koji promatramo možemo izračunati pogrešku na validacijskom skupu ili pogrešku unakrsne validacije, te zatim odabrati model s najmanjom procijenjenom testnom pogreškom. Prednost ovog pristupa u odnosu na korištenje AIC-a, BIC-a, C_p-a i prilagođenog R^2 je ta što pruža izravnu procjenu testne pogreške, uz manje pretpostavki o stvarnom modelu u pozadini podataka. Također se može koristiti za širi raspon zadaća vezanih uz odabir modela, čak i u slučajevima kada nije jednostavno utvrditi broj stupnjeva slobode modela

(npr. broj prediktora u modelu) ili kada nije jednostavno odrediti varijancu slučajne greške σ^2.

U prošlosti je provođenje unakrsne validacije bilo računski gotovo neizvedivo za mnoge zadaće s velikim p i/ili velikim n, te su AIC, BIC, C_p i prilagođeni R^2 bili privlačniji pristupi za odabir unutar skupa modela. No, uz današnja brza računala, izračuni potrebni za provođenje unakrsne validacije rijetko predstavljaju problem. Prema tome, unakrsna validacija je danas vrlo atraktivan pristup za odabir unutar skupa promatranih modela.



Slika 4.3: Za simulirani skup podataka, prikaz kvadratnog korijena BIC-a (lijevo), pogreške na validacijskom skupu (sredina) te pogreške unakrsne validacije (desno) u ovisnosti o broju prediktora d, gdje se d kreće od 1 do 11. Na temelju ovih veličina, "najbolji" model označen je križićem. (Izvor: [3], Fig. 6.3)

Za simulirani skup podataka, slika 4.3 prikazuje kvadratni korijen BIC-a, pogrešku na validacijskom skupu te pogrešku unakrsne validacije u ovisnosti o broju prediktora d. U drugom slučaju, skup opažanja je slučajno podijeljen tako da tri četvrtine podataka čine skup za učenje, a jedna četvrtina validacijski skup. U trećem slučaju korištena je desetorostruka unakrsna validacija ($k = 10$). U ovom primjeru, i metoda validacijskog skupa i metoda unakrsne validacije rezultiraju modelom sa šest varijabli. No, sva tri pristupa sugeriraju da su modeli sa četiri, pet i šest varijabli otprilike podjednaki što se tiče njihovih testnih pogrešaka.

Primijetimo da su krivulje procijenjene testne pogreške u srednjem i desnom grafu na slici 4.3 relativno ravne. Dok model s tri varijable očito ima manju procijenjenu testnu pogrešku od modela s dvije varijable, procijenjene testne pogreške modela s 3 do 11 varijabli su slične. Nadalje, ako ponovimo metodu validacijskog skupa koristeći drugu podjelu podataka na skup za učenje i validacijski skup, ili ako ponovimo metodu unakrsne validacije s drugim brojem podskupova k, onda je vrlo vjerojatno da će se promijeniti i mo-

del s najmanjom procijenjenom testnom pogreškom. U tom kontekstu možemo odabrati model korištenjem pravila jedne standardne pogreške.[2] Prvo za svaku veličinu modela izračunamo standardnu pogrešku procijenjene testne MSE, te zatim odaberemo najmanji model za koji je procijenjena testna pogreška unutar jedne standardne pogreške od najniže točke na krivulji. Ovdje je obrazloženje sljedeće: ako se određeni modeli čine podjednako dobrima, onda je ekonomično odabrati najjednostavniji model, odnosno onaj s najmanjim brojem prediktora. U našem primjeru, primjena pravila jedne standardne pogreške na metodu validacijskog skupa ili metodu unakrsne validacije dovest će do odabira modela s tri varijable.

4.2 Metode smanjenja koeficijenata

Metode odabira podskupa prediktora opisane u potpoglavlju 4.1 uključuju korištenje metode najmanjih kvadrata za prilagodbu linearnog modela koji sadrži neki podskup prediktora. Alternativno, možemo vršiti prilagodbu modela koji sadrži svih p prediktora koristeći tehniku koja ograničava ili regularizira procjene koeficijenata, ili ekvivalentno, koja smanjuje koeficijente prema nuli. Pokazat će se da smanjenje procjene koeficijenata može značajno smanjiti njihovu varijancu. Dvije najpoznatije tehnike za smanjenje koeficijenata regresije prema nuli su *ridge regresija* i *lasso*.

4.2.1 Ridge regresija

U Poglavlju 2 smo naveli da metoda najmanjih kvadrata procjenjuje $\beta_0, \beta_1, ..., \beta_p$ korištenjem vrijednosti koje minimiziraju

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

Ridge regresija je vrlo slična metodi najmanjih kvadrata, samo što se koeficijenti procjenjuju minimizacijom malo drugačijeg izraza. Procjene koeficijenata ridge regresije $\hat{\beta}_\lambda^R$ su vrijednosti koje minimiziraju

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 = \text{RSS} + \lambda \sum_{j=1}^{p} \beta_j^2, \tag{4.5}$$

gdje je $\lambda \geq 0$ parametar podešavanja koji se određuje zasebno. Jednadžba (4.5) predstavlja kompromis između dva kriterija. Kao kod metode najmanjih kvadrata, i kod ridge regresije

[2] Za definiciju standardne pogreške, vidjeti [3], str. 65 ili [2], str. 56.

traže se procjene parametara tako da model bude dobro prilagođen podacima, odnosno da je RSS malen. No, drugi sumand, $\lambda \sum_{j=1}^{p} \beta_j^2$, koji se zove kazna smanjenja (engl. shrinkage penalty), je malen kada su $\beta_1, ..., \beta_p$ blizu nuli, te ima za posljedicu smanjenje procjena za β_j prema nuli. Pomoću parametra podešavanja λ može se podesiti u kojoj mjeri ova dva sumanda utječu na procjenu parametara regresije. Kada je $\lambda = 0$, izraz za kaznu nema učinka, te će ridge regresija dati iste procjene parametara kao i metoda najmanjih kvadrata. S druge strane, kako $\lambda \to \infty$, kazna smanjenja se povećava, te će se procjene parametara ridge regresijom približavati nuli. Za razliku od metode najmanjih kvadrata, koja daje samo jedan skup procjena parametara, ridge regresija će dati različit skup procjena parametara, $\hat{\beta}_\lambda^R$, za svaku vrijednost λ. Odabir dobrog λ je od kritične važnosti, te će biti raspravljen u odjeljku 4.2.3, gdje ćemo koristiti unakrsnu validaciju.

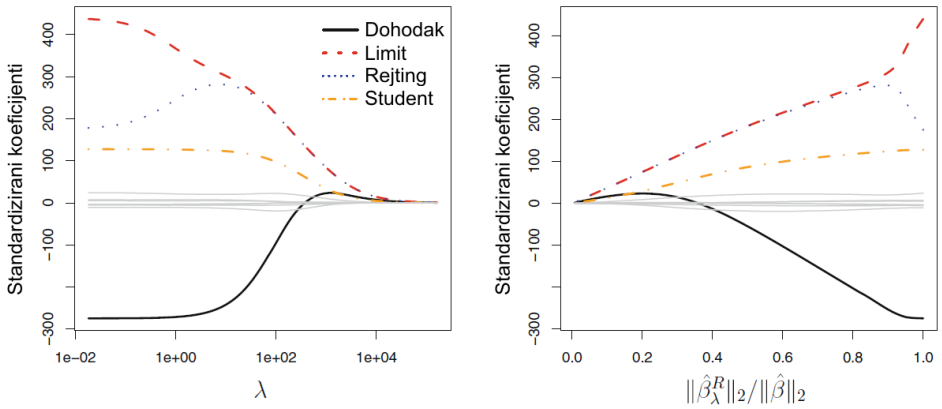
Primijetimo da se u jednadžbi (4.5) kazna smanjenja primjenjuje na $\beta_1, ..., \beta_p$, ali ne i na slobodni član β_0. Želimo smanjiti procijenjenu povezanost svake varijable s odazivom, no ne želimo smanjiti slobodni član, koji je jednostavno mjera srednje vrijednosti odaziva kada su $x_{i1} = x_{i2} = ... = x_{ip} = 0$. Ako pretpostavimo da su prije provođenja ridge regresije nezavisne varijable centrirane tako da im je očekivanje nula, onda će procjena za slobodni član biti oblika $\hat{\beta}_0 = \bar{y} = \sum_{i=1}^{n} y_i/n$.

Primjena na podacima o kreditima

Ilustrirajmo primjenu ridge regresije na skupu podataka o kreditima. Taj skup podataka opisan je u [3], str. 83. Ukratko, u njemu su uključeni podaci o stanju na kreditnoj kartici (dug), numerički prediktori kao što su dohodak, dob, broj kreditnih kartica, kreditni rejting, maksimalno zaduženje (limit), te kategorijalni prediktori kao što su studentski status, spol, etnicitet itd.

Na slici 4.4 prikazane su procjene standardiziranih koeficijenata ridge regresije na skupu podataka o kreditima (opis standardizacije bit će dan u nastavku). Na lijevom grafu svaka krivulja odgovara procjeni koeficijenta ridge regresije za jednu od varijabli, kao funkcija od λ. Na primjer, puna crna krivulja predstavlja procjenu ridge regresijom koeficijenta za Dohodak, kako se λ mijenja. Na krajnjem lijevom rubu grafa je λ praktički nula, te su odgovarajuće procjene koeficijenata ridge regresijom iste kao i procjene metodom najmanjih kvadrata. No, kako λ raste, procjene koeficijenata ridge regresijom smanjuju se prema nuli. Kada je λ vrlo velik, tada su procjene svih koeficijenata ridge regresijom praktički nula - to odgovara nul-modelu koji ne sadrži niti jedan prediktor. Na ovom grafu su Dohodak, Limit, Rejting i Student prikazani različitim bojama, jer su uz ove varijable vezane najveće procjene koeficijenata. Iako se procjene koeficijenata ridge regresijom u cjelini smanjuju kako λ raste, moguće je da neki koeficijenti djelomično rastu kako se λ povećava (kao, na primjer, Rejting i Dohodak).

Desni graf na slici 4.4 prikazuje iste procjene koeficijenata ridge regresijom kao i lijevi graf, no sada su na x-osi umjesto λ dane vrijednosti $\|\hat{\beta}_\lambda^R\|_2/\|\hat{\beta}\|_2$, gdje $\hat{\beta}$ označava vektor



Slika 4.4: Standardizirani koeficijenti ridge regresije na skupu podataka o kreditima, kao funkcije od λ odnosno $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$. (Izvor: [3], Fig. 6.4)

čiji su elementi procjene koeficijenata metodom najmanjih kvadrata. Zapis $\|\beta\|_2$ označava ℓ_2 normu vektora, koja se definira kao $\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$, i koja nam govori koliko je β udaljen od nule. Kako λ raste, ℓ_2 norma od $\hat{\beta}^R_\lambda$ će se smanjivati, a prema tome smanjivat će se i $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$. Vrijednost zadnjeg izraza kretat će se od 1 (kada je $\lambda = 0$, te je procjena koeficijenata ridge regresijom jednaka procjeni metodom najmanjih kvadrata, odnosno $\|\hat{\beta}^R_\lambda\|_2 = \|\hat{\beta}\|_2$) do 0 (kada je $\lambda = \infty$, te je procjena koeficijenata ridge regresijom nul-vektor, čija je ℓ_2 norma jednaka nuli). Prema tome, vrijednosti na x-osi desnog grafa slike 4.4 nam govore u kojoj mjeri su procjene koeficijenata ridge regresije smanjene prema nuli: male vrijednosti upućuju na znatno smanjenje koeficijenata prema nuli.

Kod procjene koeficijenata metodom najmanjih kvadrata vrijedi da, ako pomnožimo prediktor X_j konstantom c, onda će procjena odgovarajućeg koeficijenta biti skalirana konstantom $\frac{1}{c}$. Drugim riječima, neovisno o tome kako je j-ti prediktor skaliran, $\hat{\beta}_j X_j$ će ostati isto. Nasuprot tome, procjene koeficijenata ridge regresijom se mogu značajno promijeniti ukoliko dani prediktor pomnožimo konstantom. Promotrimo, na primjer, varijablu Dohodak koja je mjerena u dolarima. Dohodak bi bilo moguće mjeriti i u tisućama dolara, u kojem slučaju bi se opažene vrijednosti smanjile za faktor 1000. Zbog prisustva sume kvadrata koeficijenata $\sum_{j=1}^{p} \beta_j^2$ u formuli (4.5), takva promjena mjernih jedinica neće jednostavno povući skaliranje procijenjenog koeficijenta za faktor 1000. Dakle, $\hat{\beta}^R_{j,\lambda} X_j$ neće ovisiti samo o vrijednosti λ, nego i o skaliranju j-tog prediktora. Zapravo, vrijednost $\hat{\beta}^R_{j,\lambda} X_j$ može ovisiti čak i o skaliranju drugih prediktora. Iz tog je razloga najbolje provesti ridge

regresiju nakon standardizacije prediktora korištenjem formule

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}, \tag{4.6}$$

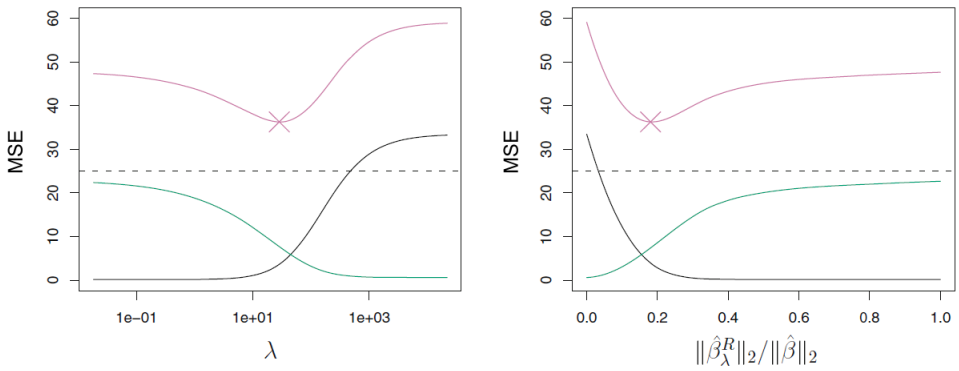
tako da svi prediktori budu izraženi u istom mjerilu. Nazivnik u formuli (4.6) je procijenjena standardna devijacija j-tog prediktora. Prema tome, standardna devijacija svih standardiziranih prediktora će iznositi 1. Iz tog razloga konačna prilagodba neće ovisiti o tome u kojim jedinicama su prediktori mjereni. Na slici 4.4, na y-osi su prikazane procjene standardiziranih koeficijenata ridge regresijom - to jest, procjene koeficijenata dobivene provođenjem ridge regresije korištenjem standardiziranih prediktora.

Prednosti ridge regresije nad metodom najmanjih kvadrata

Prednosti ridge regresije nad metodom najmanjih kvadrata proizlaze iz kompromisa između pristranosti i varijance. Kako se λ povećava, tako se fleksibilnost prilagodbe ridge regresijom smanjuje, što vodi do smanjenja varijance ali povećanja pristranosti. To je ilustrirano na lijevom grafu slike 4.5, koja se temelji na simuliranom skupu podataka sa $p = 45$ prediktora i $n = 50$ opažanja. Zelena krivulja na lijevom grafu slike 4.5 prikazuje varijancu predviđanja ridge regresijom kao funkciju od λ. Kod procjene koeficijenata metodom najmanjih kvadrata, što odgovara ridge regresiji sa $\lambda = 0$, varijanca je visoka dok je pristranost vrlo mala. No, kako λ raste, smanjenje koeficijenata procijenjenih ridge regresijom vodi do značajnog smanjenja varijance predviđanja, uz cijenu malog povećanja pristranosti. Kao što je prethodno navedeno, srednjekvadratna pogreška (MSE, označena ljubičastom bojom) uključuje varijancu i kvadrat pristranosti. Za vrijednosti λ do otprilike 10 varijanca se brzo smanjuje, uz vrlo malo povećanje pristranosti, koja je označena crnom bojom. Prema tome, MSE se znatno smanjuje kako λ raste od 0 do 10. Nakon te točke, smanjenje varijance uslijed povećanja λ se usporava. Istovremeno, koeficijenti zbog smanjenja postaju znatno podcijenjeni, što rezultira velikim povećanjem pristranosti. Najmanja MSE postiže se za otprilike $\lambda = 30$.

Desni graf na slici 4.5 prikazuje iste krivulje kao i lijevi graf, no ovaj puta prikazane u odnosu na kvocijent ℓ_2 norme procjene koeficijenata ridge regresijom i ℓ_2 norme procjene koeficijenata metodom najmanjih kvadrata. Sada, kako se krećemo s lijeva na desno, prilagodba postaje fleksibilnija, te se pristranost smanjuje dok se varijanca povećava.

Općenito, kada je odnos između odaziva i prediktora približno linearan, procjene metodom najmanjih kvadrata imat će nisku pristranost ali moguće visoku varijancu. To znači da mala promjena u podacima za učenje može dovesti do velike promjene u procjeni koeficijenata metodom najmanjih kvadrata. Posebno, kada je broj varijabli p velik skoro kao broj opažanja n, kao u primjeru na slici 4.5, procjene metodom najmanjih kvadrata bit će



Slika 4.5: Kvadrat pristranosti (crno), varijanca (zeleno) i testna srednjekvadratna pogreška (ljubičasto) za predviđanja ridge regresijom na simuliranom skupu podataka kao funkcije od λ odnosno $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$. Ljubičastim križićima označeni su modeli ridge regresije za koje je MSE najmanja. (Izvor: [3], Fig. 6.5)

vrlo varijabilne. Zato ridge regresija daje najbolje rezultate u slučajevima kada procjene metodom najmanjih kvadrata imaju visoku varijancu.

Dodatno, ridge regresija ima značajne računske prednosti nad metodom odabira najboljeg podskupa, kod koje je potrebno pretraživati skup od 2^p modela. Kao što je prethodno spomenuto, takva pretraga može biti računski neizvediva, čak i za umjereno velike p. Nasuprot tome, za svaki fiksni λ, ridge regresija prilagođava samo jedan model, te se postupak prilagodbe modela podacima može provesti dosta brzo.

4.2.2 Lasso regresija

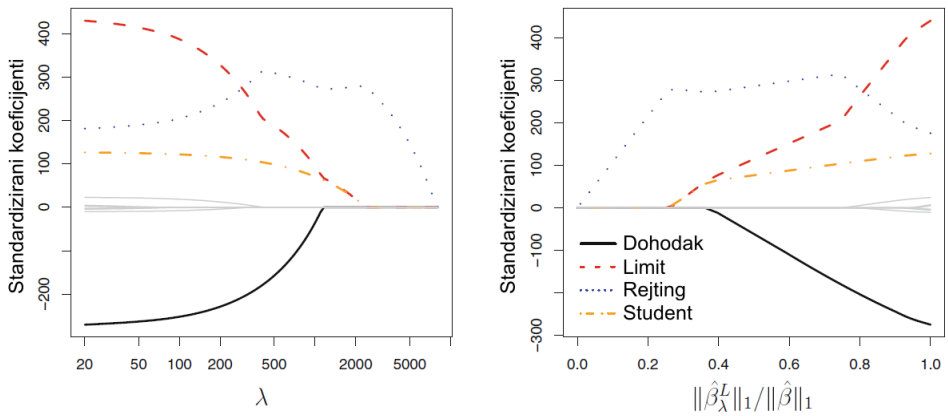
Ridge regresija ima jedan očiti nedostatak. Za razliku od metoda odabira najboljeg podskupa te postupnog odabira unaprijed i unatrag, koje će općenito odabrati modele koji se sastoje samo od podskupa prediktora, ridge regresija će u finalni model uključiti svih p prediktora. Kazna u iznosu $\lambda \sum_{j=1}^{p} \beta_j^2$ iz (4.5) će smanjiti sve koeficijente prema nuli, ali niti jednog od njih neće postaviti točno na nulu (osim u slučaju $\lambda = \infty$). To ne mora biti problem što se tiče preciznosti predviđanja, no može predstavljati problem kod interpretacije modela u situacijama kada je p velik. Na primjer, čini se da su u skupu podataka o kreditima najvažniji prediktori Dohodak, Limit, Rejting i Student. Zato bismo mogli napraviti model koji se sastoji samo od tih prediktora. No, ridge regresija će uvijek rezultirati modelom koji uključuje svih deset prediktora. Povećavanje λ će dovesti do smanjenja veličine koeficijenata, ali neće rezultirati isključenjem niti jedne od varijabli (to jest, niti uz jedan prediktor neće koeficijent biti točno nula).

Lasso regresija (od engl. least absolute shrinkage and selection operator) je relativno nova alternativa ridge regresiji koja je uklonila taj nedostatak. Koeficijenti lasso regresije, $\hat{\beta}^L_\lambda$, minimiziraju izraz

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\left|\beta_j\right| = \text{RSS} + \lambda\sum_{j=1}^{p}\left|\beta_j\right|. \tag{4.7}$$

Usporedimo li (4.7) sa (4.5), vidjet ćemo da su lasso i ridge regresija slično formulirane. Jedina razlika je ta što je β_j^2 u izrazu za kaznu u ridge regresiji zamijenjen s $\left|\beta_j\right|$ u kazni lasso regresije. Možemo reći da lasso regresija koristi ℓ_1 kaznu umjesto ℓ_2 kazne. ℓ_1 norma vektora koeficijenata β dana je s $\|\beta\|_1 = \sum_{j=1}^{p}\left|\beta_j\right|$.

Kao u slučaju ridge regresije, i lasso regresija smanjuje procjene koeficijenata prema nuli. No, u slučaju lasso regresije ℓ_1 kazna ima učinak postavljanja procjena nekih koeficijenata točno na nulu kada je parametar podešavanja λ dovoljno velik. Iz tog razloga lasso regresija vrši odabir prediktora, kao i metoda odabira najboljeg podskupa. Općenito, modele koje daje lasso regresija će biti znatno lakše interpretirati od modela generiranih ridge regresijom. Kažemo da lasso regresija daje rijetke modele (engl. sparse models), to jest modele koji sadrže samo neki podskup prediktora. Kao i kod ridge regresije, odabir dobre vrijednosti za λ je od kritične važnosti. To će biti raspravljeno u odjeljku 4.2.3, gdje ćemo koristiti unakrsnu validaciju.



Slika 4.6: Standardizirani koeficijenti lasso regresije na skupu podataka o kreditima, kao funkcije od λ odnosno $\|\hat{\beta}^L_\lambda\|_1/\|\hat{\beta}\|_1$. (Izvor: [3], Fig. 6.6)

Kao primjer, promotrimo grafove procjena koeficijenata na slici 4.6 koji su dobiveni primjenom lasso regresije na skupu podataka o kreditima. Kada je $\lambda = 0$, lasso regresija

daje isti model kao i metoda najmanjih kvadrata. Kada je λ dovoljno velik, lasso regresija daje nulti model u kojem su procjene svih koeficijenata jednake nuli. No, između ova dva ekstrema se modeli koje daju lasso i ridge regresija dosta razlikuju. Kako se krećemo s lijeva na desno na desnom grafu slike 4.6, uočavamo da isprva lasso regresija rezultira modelom koji sadrži samo prediktor Rejting. Zatim u model gotovo istovremeno ulaze prediktori Student i Limit, a ubrzo nakon toga i Dohodak. Konačno, u model ulaze i preostali prediktori. Prema tome, u ovisnosti o vrijednosti λ, lasso regresija može dati model koji uključuje bilo koji broj prediktora. Nasuprot tome, ridge regresija će uvijek uključivati sve varijable u modelu, iako će veličina procijenjenih koeficijenata ovisiti o odabranom λ.

Drugi zapis ridge i lasso regresije

Može se pokazati da su procjene koeficijenata lasso regresije rješenja zadaće

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{uz uvjet} \quad \sum_{j=1}^{p} |\beta_j| \leq s, \tag{4.8}$$

dok su procjene koeficijenata ridge regresije rješenja zadaće

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{uz uvjet} \quad \sum_{j=1}^{p} \beta_j^2 \leq s. \tag{4.9}$$

Drugim riječima, za svaki $\lambda \geq 0$, postoji s takav da će (4.7) i (4.8) dati iste procjene koeficijenata lasso regresije. Slično, za svaki $\lambda \geq 0$, postoji odgovarajući s takav da će (4.5) i (4.9) dati iste procjene koeficijenata ridge regresije. Kada je $p = 2$, onda (4.8) govori da procjene koeficijenata lasso regresije daju najmanju RSS od svih točaka unutar romba definiranog s $|\beta_1| + |\beta_2| \leq s$. Slično, procjene koeficijenata ridge regresije daju najmanju RSS od svih točaka u krugu definiranom s $\beta_1^2 + \beta_2^2 \leq s$.

O (4.8) možemo razmišljati na sljedeći način. Kada provodimo lasso regresiju, pokušavamo naći skup procjena koeficijenata koji će dovesti do najmanje RSS uz budžet s koji ograničava koliko $\sum_{j=1}^{p} |\beta_j|$ može biti veliko. Kada je s ekstremno velik, taj budžet nije jako restriktivan pa procjene koeficijenata mogu biti velike. Zapravo, ako je s toliko velik da rješenje metode najmanjih kvadrata bude u okviru budžeta, onda će (4.8) jednostavno dati rješenje metode najmanjih kvadrata. Nasuprot tome, ako je s malen, onda i $\sum_{j=1}^{p} |\beta_j|$ mora biti maleno kako bi se izbjeglo prekoračenje budžeta. Isto tako, (4.9) nam govori da, kada provodimo ridge regresiju, tražimo skup procjena koeficijenata takvih da RSS bude što je manja moguća, uz uvjet da $\sum_{j=1}^{p} \beta_j^2$ ne premaši budžet s.

Zapisi (4.8) i (4.9) otkrivaju blisku vezu između lasso regresije, ridge regresije, i odabira najboljeg podskupa. Promotrimo zadaću

$$\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\} \quad \text{uz uvjet} \quad \sum_{j=1}^{p} I(\beta_j \neq 0) \leq s. \tag{4.10}$$

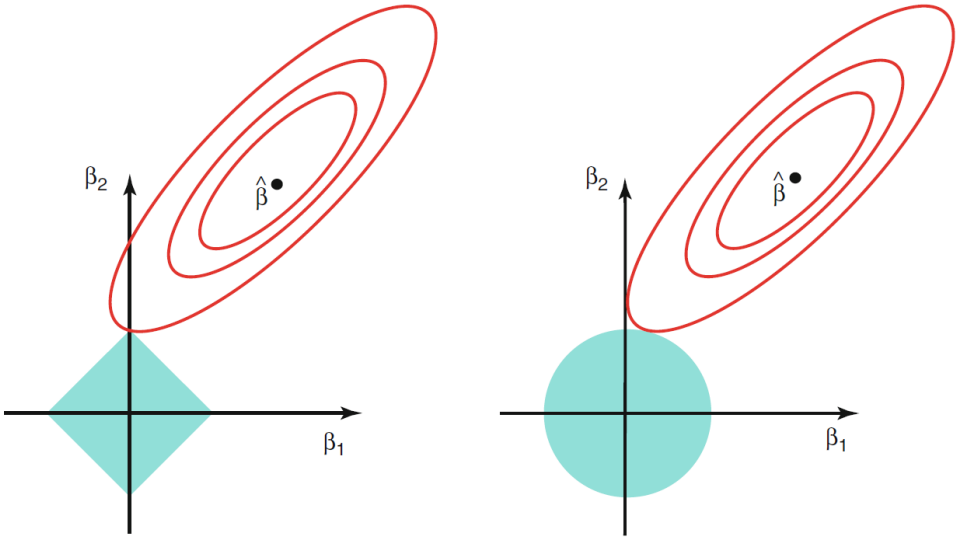
Ovdje je $I(\beta_j \neq 0)$ indikatorska varijabla: ako je $\beta_j \neq 0$ poprima vrijednost 1, a inače joj je vrijednost nula. Tada je (4.10) problem nalaženja skupa procjena koeficijenata takvih da RSS bude što je manja moguća, uz uvjet da najviše s koeficijenata može biti različito od nule. Zadaća (4.10) ekvivalentna je problemu odabira najboljeg podskupa. Na žalost, rješavanje (4.10) je računski neizvedivo kada je p velik, jer uključuje razmatranje svih $\binom{p}{s}$ modela koji sadrže s prediktora. Stoga možemo interpretirati lasso i ridge regresiju kao računski provedive alternative metodi odabira najboljeg podskupa. Naravno, lasso regresija je puno bliža metodi odabira najboljeg podskupa od ridge regresije iz razloga što, za dovoljno malu vrijednost s u (4.8), jedino lasso regresija vrši odabir prediktora.

Odabir prediktora kod lasso regresije

Kako je moguće da lasso regresija, za razliku of ridge regresije, daje procjene koeficijenata od kojih su neke točno jednake nuli? Zapisi (4.8) i (4.9) mogu nam biti od koristi pri traženju odgovora na to pitanje. Situacija je ilustrirana na slici 4.7.
Rješenje metode najmanjih kvadrata označeno je sa $\hat{\beta}$, dok plava područja prikazuju ograničenja lasso regresije u (4.8), odnosno ridge regresije u (4.9). Za dovoljno velik s, područja ograničenja će sadržavati $\hat{\beta}$, te će se procjene lasso i ridge regresijom podudarati s procjenom metodom najmanjih kvadrata. Ta velika vrijednost od s odgovara slučaju kada je $\lambda = 0$ u (4.5) i (4.7). No, na slici 4.7 procjena metodom najmanjih kvadrata leži izvan područja ograničenja, te će zbog toga biti različita od procjena lasso i ridge regresijom.

Svaka elipsa oko $\hat{\beta}$ odgovara jednoj vrijednosti RSS-a. Kako se elipse odmiču od procjene koeficijenata metodom najmanjih kvadrata, tako se RSS povećava. Jednadžbe (4.8) i (4.9) upućuju na to da su procjene koeficijenata lasso i ridge regresijom dane prvom točkom u kojoj jedna elipsa dodirne rub područja ograničenja. Zbog toga što je kod ridge regresije područje ograničenja krug bez oštrih rubova, točka dodira općenito neće biti na jednoj od osi, te će zbog toga sve procjene koeficijenata ridge regresijom biti različite od nule. S druge strane, područje ograničenja lasso regresije ima kuteve na svakoj od osi, te će elipsa često dotaknuti područje ograničenja na jednoj od osi. Kada se to dogodi, jedan od koeficijenata bit će jednak nuli. U višedimenzionalnom slučaju je moguće da više procjena koeficijenata bude istovremeno jednako nuli. Na slici 4.7 dodir se dogodio pri $\beta_1 = 0$, te će rezultirajući model uključivati samo β_2.

Na slici 4.7 smo promotrili samo slučaj kada je $p = 2$. Kada je $p = 3$, područje ograničenja će postati sfera za ridge regresiju, odnosno poliedar za lasso regresiju. Kada



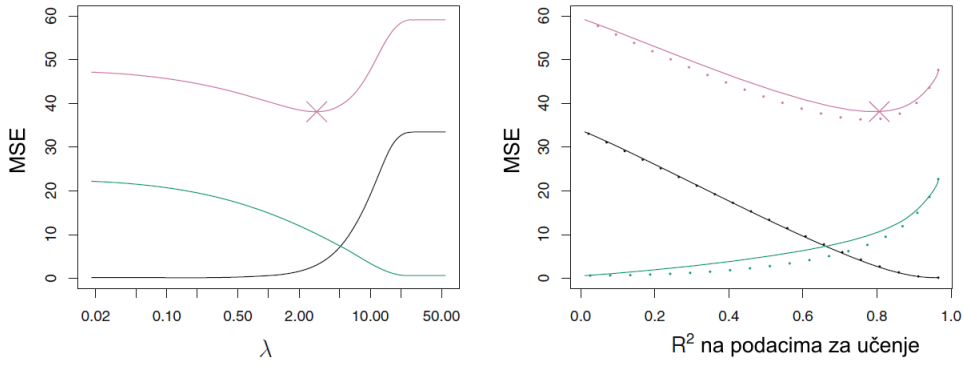
Slika 4.7: Za lasso regresiju (lijevo) i ridge regresiju (desno) plavom bojom su označena područja ograničenja ($|\beta_1| + |\beta_2| \leq s$ odnosno $\beta_1^2 + \beta_2^2 \leq s$), dok su crvene elipse konture RSS-a. (Izvor: [3], Fig. 6.7)

je $p > 3$, područje ograničenja bit će hipersfera za ridge regresiju, odnosno politop za lasso regresiju. No, i u tim slučajevima i dalje vrijede ideje prikazane na slici 4.7. Kada je $p > 2$, lasso regresija će dovesti do selekcije varijabli zbog oštrih rubova poliedra odnosno politopa.

Usporedba lasso i ridge regresije

Očita prednost lasso regresije nad ridge regresijom je u tome što rezultira modelima koje je lakše interpretirati, s obzirom da se sastoje samo od podskupa prediktora. S druge strane, možemo se pitati koji od modela daje bolja predviđanja. Slika 4.8 prikazuje kvadrat pristranosti, varijancu i testnu MSE lasso regresije primijenjene na istim simuliranim podacima koji su korišteni kod slike 4.5. Jasno je da se, kvalitativno, lasso regresija ponaša slično kao i ridge regresija: kako se λ povećava, tako se varijanca modela smanjuje dok se pristranost povećava. Na desnom grafu slike 4.8 točkasto su prikazane prilagodbe ridge regresijom. Podaci su prikazani u odnosu na R^2 na podacima za učenje: to je vrlo koristan način indeksacije modela, koji se može koristiti za usporedbu modela s različitim vrstama regularizacije, kao što je ovdje slučaj. U ovom primjeru, lasso i ridge regresija rezultiraju gotovo istom pristranošću. No, varijanca ridge regresije je nešto manja od varijance lasso regresije. Zbog toga će i minimalna MSE ridge regresije biti nešto manja nego u slučaju

lasso regresije.

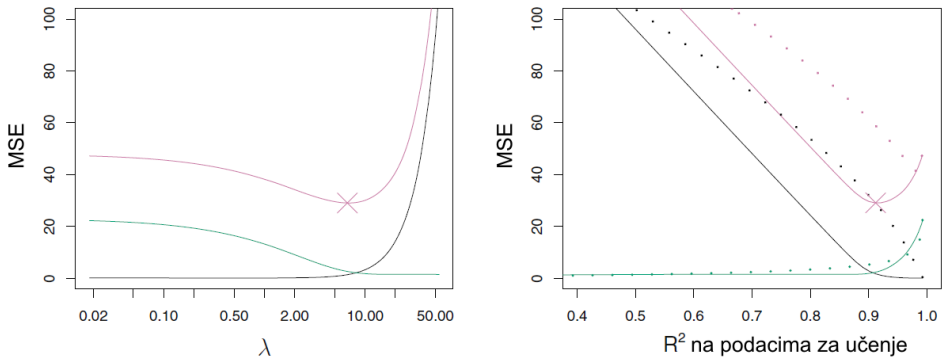


Slika 4.8: Lijevo: grafovi kvadrata pristranosti (crno), varijance (zeleno) i testne MSE (ljubičasto) za lasso regresiju na simuliranom skupu podataka. Desno: usporedba kvadrata pristranosti, varijance i testne MSE između lasso (puna linija) i ridge regresije (točkasto). Grafovi su prikazani u odnosu na R^2 na podacima za učenje, što je česti oblik indeksacije. Križići označavaju lasso model za koji je testna MSE najmanja. (Izvor: [3], Fig. 6.8)

Podaci na slici 4.8 generirani su tako da je svih 45 prediktora povezano s odazivom, odnosno niti jedan od pravih koeficijenata $\beta_1, ..., \beta_{45}$ nije bio jednak nuli. Lasso regresija implicitno pretpostavlja da će neki od koeficijenata biti baš jednaki nuli. Iz tog razloga ne iznenađuje da u ovom kontekstu ridge regresija daje bolje rezultate od lasso regresije. Slika 4.9 ilustrira sličnu situaciju, samo što je sada odaziv funkcija samo dva od 45 prediktora. Sada će lasso regresija davati pretežno bolje rezultate od ridge regresije u smislu pristranosti, varijance i MSE.

Ova dva primjera pokazuju da, općenito, niti jedna od lasso i ridge regresije neće uvijek biti bolja od druge. Generalno, možemo očekivati da će lasso regresija davati bolje rezultate u situacijama kada relativno mali broj prediktora ima značajno velike koeficijente, dok su uz druge prediktore koeficijenti koji su ili vrlo mali ili jednaki nula. Ridge regresija davat će bolje rezultate kada je odaziv funkcija većeg broja prediktora, pri čemu svi imaju podjednako velike koeficijente. No, u slučaju stvarnih podataka, broj prediktora koji su povezani s odazivom nije nikad unaprijed poznat. Pri donošenju odluke o tome koji je pristup bolji za određeni skup podataka možemo koristiti tehnike poput unakrsne validacije.

Kao i kod ridge regresije, kada procjene metodom najmanjih kvadrata imaju vrlo visoku varijancu, korištenje lasso regresije može dovesti do smanjenja varijance uz trošak malog povećanja pristranosti, te konačno do točnijih predviđanja. Za razliku od ridge regresije, lasso regresija vrši odabir varijabli, te rezultira modelima koje je lakše interpretirati.



Slika 4.9: Lijevo: grafovi kvadrata pristranosti (crno), varijance (zeleno) i testne MSE (ljubičasto) za lasso regresiju. Simulirani podaci slični su onima sa slike 4.8, s razlikom da su sada samo dva prediktora povezana s odazivom. Desno: usporedba kvadrata pristranosti, varijance i testne MSE između lasso (puna linija) i ridge regresije (točkasto). Grafovi su prikazani u odnosu na R^2 na podacima za učenje, što je česti oblik indeksacije. Križići označavaju lasso model za koji je testna MSE najmanja. (Izvor: [3], Fig. 6.9)

Jednostavan specijalni slučaj ridge i lasso regresije

Označimo sa $\mathbf{X}$ matricu podataka: to je $n \times p$ matrica, pri čemu je n broj opažanja, p broj prediktora, te čiji (i, j)-ti element je x_{ij}:

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}.$$

Kako bismo dobili bolju intuiciju o ponašanju ridge i lasso regresije, promotrimo jednostavan specijalni slučaj kada je $n = p$, i kada je $\mathbf{X}$ jedinična matrica (t.j. s jedinicama na dijagonali i nulama izvan dijagonale, odnosno $x_{ij} = 1$ za $i = j$, a 0 inače). Kako bismo još pojednostavnili problem, dodatno pretpostavimo da vršimo regresiju bez slobodnog člana (β_0). Uz te pretpostavke, običan problem najmanjih kvadrata svodi se na nalaženje $\beta_1, ..., \beta_p$ koji minimiziraju

$$\sum_{j=1}^{p} (y_j - \beta_j)^2. \tag{4.11}$$

U tom slučaju, rješenje metode najmanjih kvadrata dano je s

$$\hat{\beta}_j = y_j.$$

U ovom kontekstu, provođenje ridge regresije svodi se na traženje $\beta_1, ..., \beta_p$ koji minimiziraju

$$\sum_{j=1}^{p}(y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p}\beta_j^2, \tag{4.12}$$

dok se provođenje lasso regresije svodi na traženje $\beta_1, ..., \beta_p$ koji minimiziraju

$$\sum_{j=1}^{p}(y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p}\left|\beta_j\right|. \tag{4.13}$$

Može se pokazati da su tada procjene koeficijenata ridge regresijom oblika

$$\hat{\beta}_j^R = y_j/(1 + \lambda), \tag{4.14}$$

dok su procjene koeficijenata lasso regresijom oblika

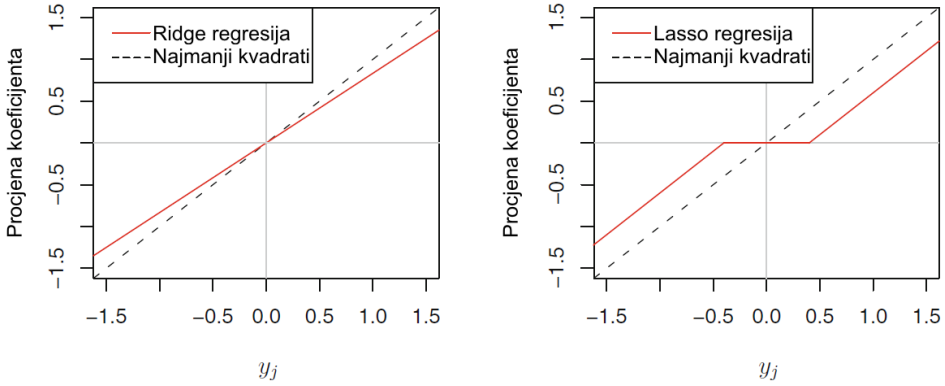
$$\hat{\beta}_j^L = \begin{cases} y_j - \lambda/2 & \text{za } y_j > \lambda/2, \\ y_j + \lambda/2 & \text{za } y_j < -\lambda/2, \\ 0 & \text{za } \left|y_j\right| \leq \lambda/2. \end{cases} \tag{4.15}$$

Situacija je prikazana na slici 4.10. Vidimo da je kod ridge i lasso regresije riječ o dva vrlo različita načina smanjenja koeficijenata. Kod ridge regresije, svaka procjena koeficijenata metodom najmanjih kvadrata se smanjuje u istom omjeru. Nasuprot tome, lasso regresija smanjuje svaku procjenu koeficijenata metodom najmanjih kvadrata za konstantni iznos $\lambda/2$; procjene čija je apsolutna vrijednost manja od $\lambda/2$ postavljene su točno na nulu. Činjenica da su neki koeficijenti lasso regresije postavljeni točno na nulu objašnjava zašto lasso regresija vrši odabir prediktora.

Situacija prikazana na slici 4.10 bit će nešto kompliciranija u slučaju općenitije matrice podataka $\mathbf{X}$, no glavne ideje će i dalje približno vrijediti: ridge regresija će više ili manje sve koeficijente smanjivati u istom omjeru, dok će lasso regresija više ili manje smanjivati sve koeficijente za isti iznos, dok će dovoljno mali koeficijenti biti postavljeni točno na nulu.

4.2.3 Odabir parametra podešavanja

Kao što nam je u potpoglavlju 4.1 bila potrebna metoda za odabir najboljeg modela među promatranima, tako nam je i kod implementacije ridge i lasso regresije potrebna metoda za

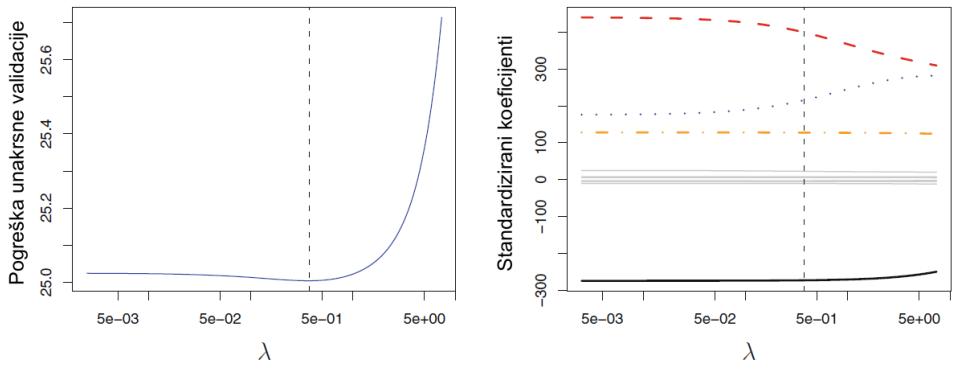


Slika 4.10: Procjene koeficijenata ridge i lasso regresijom u jednostavnom slučaju kada je $n = p$ i $\mathbf{X}$ jedinična matrica. Lijevo: procjene koeficijenata ridge regresijom se prema nuli smanjuju proporcionalno u odnosu na procjene metodom najmanjih kvadrata. Desno: procjene koeficijenata lasso regresijom su u odnosu na procjene metodom najmanjih kvadrata smanjene za fiksni iznos, ili su postavljene na nulu. (Izvor: [3], Fig. 6.10)

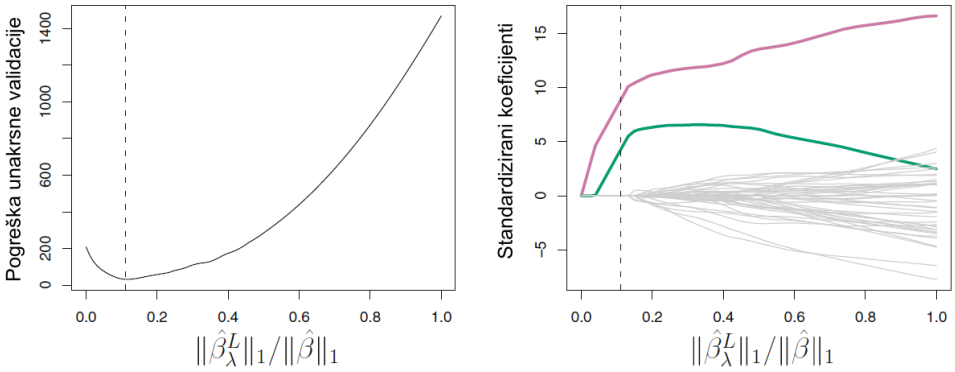
odabir parametra podešavanja λ u (4.5) i (4.7) odnosno, ekvivalentno, vrijednost ograde s u (4.9) i (4.8). U rješenju tog problema može nam pomoći unakrsna validacija: odaberemo mrežu vrijednosti λ, te za svaku vrijednost λ izračunamo pogrešku unakrsne validacije kako je opisano u poglavlju 3. Zatim odaberemo onu vrijednost parametra podešavanja za koju je pogreška unakrsne validacije najmanja. Konačno, izvršimo novu prilagodbu modela pri čemu sada koristimo sva dostupna opažanja i odabranu vrijednost parametra podešavanja.

Slika 4.11 prikazuje odabir λ nakon provođenja pojedinačne unakrsne validacije za ridge regresiju na skupu podataka o kreditima. Crtkanom vertikalnom linijom označena je odabrana vrijednost λ. U ovom slučaju ta je vrijednost relativno mala, što upućuje na to da je za optimalnu prilagodbu podacima potrebno samo malo smanjenje koeficijenata u odnosu na metodu najmanjih kvadrata. Dodatno, iz grafa vidimo da postoji široki raspon vrijednosti od λ koje bi rezultirale sličnom pogreškom. U ovakvim slučajevima možemo jednostavno koristiti metodu najmanjih kvadrata.

Slika 4.12 ilustrira primjenu deseterostruke unakrsne validacije u slučaju lasso regresije na simuliranom skupu podataka sa slike 4.9, gdje su samo dva prediktora povezana s odazivom. Lijevi graf prikazuje pogrešku unakrsne validacije, dok desni graf prikazuje procjene koeficijenata. Vertikalne crtkane linije označavaju mjesta na kojima je pogreška unakrsne validacije najmanja. Dvije obojane linije na desnom grafu predstavljaju prediktore koji su povezani s odazivom (često ih zovemo varijablama signala), dok sive linije predstavljaju prediktore koji nemaju veze s odazivom (često ih zovemo varijablama šuma). Lasso regre-



Slika 4.11: Lijevo: pogreške unakrsne validacije nakon primjene ridge regresije na skupu podataka o kreditima uz različite vrijednosti λ. Desno: procjene koeficijenata kao funkcije od λ. Crtkanom linijom označena je vrijednost λ odabrana unakrsnom validacijom. (Izvor: [3], Fig. 6.12)



Slika 4.12: Lijevo: pogreške deseterostruke unakrsne validacije za lasso regresiju, primijenjene na simuliranom skupu podataka sa slike 4.9 (gdje su samo dva prediktora povezana s odazivom). Desno: odgovarajuće procjene koeficijenata lasso regresijom. Crtkana linija označava prilagodbu lasso regresijom kod koje je pogreška unakrsne validacije najmanja. (Izvor: [3], Fig. 6.13)

sija je varijablama signala ispravno dodijelila znatno veće procjene koeficijenata. Dodatno, najmanja pogreška unakrsne validacije odgovara procjeni koeficijenata kod koje su samo koeficijenti uz varijable signala različiti od nule. Vidimo da je lasso regresija u kombinaciji

s unakrsnom validacijom ispravno identificirala dvije varijable signala u modelu, iako je riječ o problematičnoj situaciji sa $p = 45$ varijabli i samo $n = 50$ opažanja. Nasuprot tome, procjena metodom najmanjih kvadrata, prikazana na desnom rubu desnog grafa slike 4.12, je samo jednoj varijabli signala dodijelila veliku procjenu koeficijenta.

Poglavlje 5

Primjena u aktuarstvu

5.1 Uvod

Aktuar u osiguravajućem društvu dobio je zadatak da revidira cjenik osiguranja od profesionalne odgovornosti liječnika. U tu svrhu želi vidjeti koje varijable su najviše povezane s odazivom, odnosno visinom štete. To će mu pomoći u određivanju cijene koja će odgovarati rizičnosti osiguranika: osiguranici s očekivanim malim iznosima šteta plaćat će manju premiju od rizičnijih osiguranika. (Sličnu situaciju možemo vidjeti kod osiguranja od autoodgovornosti, gdje mladi vozači mogu plaćati znatno veće premije od iskusnijih vozača.)

Od odjela šteta aktuar je dobio podatke o $n = 500$ šteta, od kojih je prvih deset zapisa kako slijedi:

br_stete	god_isk	lok_cij	nastava	osig_sv	sati_tj	spec	steta
1	3	97	0	9.000.000	47	1	1.508.000
2	38	101	0	4.000.000	51	4	1.657.000
3	0	96	0	7.000.000	47	1	1.257.000
4	33	108	1	1.000.000	56	2	687.000
5	22	102	0	2.000.000	51	5	1.745.000
6	42	96	0	7.000.000	47	3	1.731.000
7	13	102	1	2.000.000	52	2	873.000
8	17	104	0	3.000.000	53	3	1.292.000
9	32	103	0	1.000.000	52	3	986.000
10	20	98	0	5.000.000	49	1	939.000

Osim prvog stupca s brojem štete (`br_stete`), dostupne su sljedeće varijable:

- `god_isk`: broj godina iskustva liječnika
- `lok_cij`: indeks cijena u mjestu prebivališta pacijenta. Ukoliko pacijent živi u mjestu s višim troškovima života, mogao bi zahtijevati odštetu u većem iznosu nego

da živi u mjestu s nižim troškovima života, pa je odjel šteta za potrebe analize dostavio i taj podatak.

- `nastava`: 1 ako liječnik sudjeluje u izvođenju nastave, npr. na Medicinskom fakultetu, a 0 inače.
- `osig_sv`: osigurana svota, odnosno visina pokrića koju je liječnik ugovorio. Moguće je ugovoriti iznose od 1.000.000 do 10.000.000 eura, u koracima od 1.000.000 eura. Odabrani iznos odnosi se na ukupno pokriće šteta tijekom jedne godine.
- `sati_tj`: prosječan broj sati koliko liječnik tjedno radi
- `spec`: osiguravajuće društvo je svrstalo liječničke specijalizacije u pet kategorija, od 1 do 5. U kategoriju 1 spadaju specijalizacije kod kojih je vjerojatnost velike štete zbog liječničke pogreške manja (npr. obiteljska medicina), dok u kategoriju 5 spadaju specijalizacije gdje je vjerojatnost velikog iznosa štete veća (npr. kardiokirurgija).
- `steta`: odšteta isplaćena pacijentu (ili nasljednicima u slučaju smrti), uključujući troškove obrade štete

Podaci su generirani u R-u. Kako bi čitatelju izlaganje bilo zanimljivije, i neizvjesnije, opis generiranja ovog skupa podataka bit će dan na kraju, u potpoglavlju 5.7. Ovdje samo napomenimo da je pri generiranju podataka primarni cilj bio ilustracija metoda iz prethodnih poglavlja, a manje realističnost.

Još jedan od zadataka aktuara je razvoj modela koji će dati predviđanje visine štete za danu kombinaciju gore navedenih ulaznih parametara. Ta informacija može biti korisna odjelu za preuzimanje rizika (underwriting), koji mora odlučiti hoće li nekog liječnika primiti u osiguranje ili ne. Ako je predviđena šteta vrlo visoka, odjel može odlučiti da tog liječnika ne primi u osiguranje, ili da zaračuna doplatak na premiju.

Visinu štete aktuar je odlučio modelirati linearnim modelom oblika

$$\texttt{steta} = \beta_0 + \beta_1 \texttt{god\_isk} + \beta_2 \texttt{lok\_cij} + \beta_3 \texttt{nastava} + \beta_4 \texttt{osig\_sv} + \beta_5 \texttt{sati\_tj} + \beta_6 \texttt{spec} + \epsilon.$$

Uočimo da svi prediktori ovise o liječniku, s iznimkom razine cijena u prebivalištu pacijenta, pa se u kontekstu predviđanja vrijednost te varijable može postaviti na 100, ili procijeniti drugačije, npr. ako liječnik prima samo pacijente iz svog mjesta, može se uzeti razina cijena u mjestu ordinacije.

U nastavku ćemo opisati primjenu metoda iz prethodnih poglavlja u ovom konkretnom slučaju.[1] Koristit ćemo programski paket R. Prije toga, možemo vizualno ispitati podatke

[1] U pripremi sljedećih potpoglavlja od koristi je bila [3], str. 113-115, 191 i 244-255.

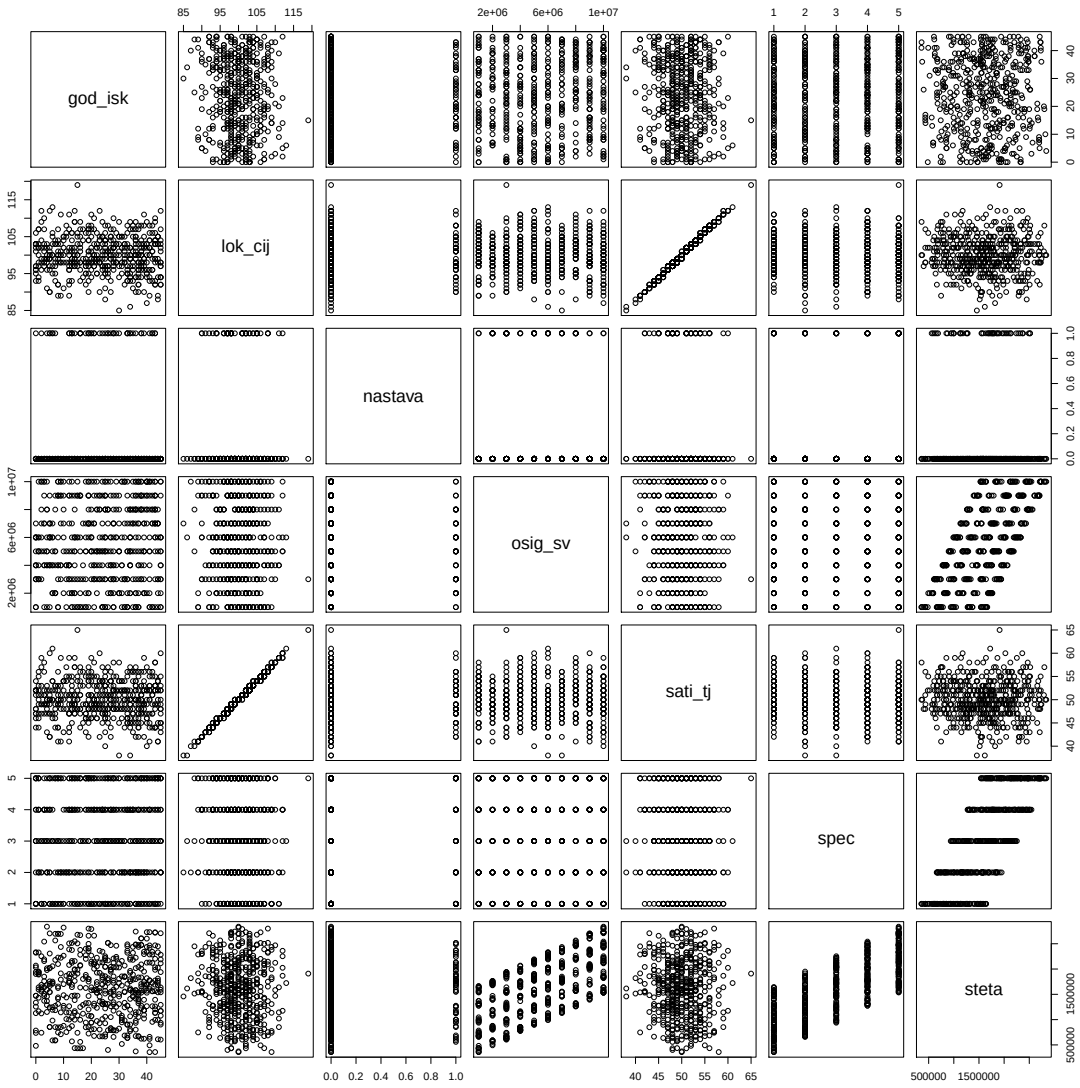
pomoću naredbe `pairs(podaci)`, pri čemu je `podaci` matrica (preciznije: data frame) u koju je aktuar spremio podatke dobivene od odjela šteta. Rezultat ove naredbe je skup dijagrama raspršenja (engl. scatterplot) za svaki par varijabli u `podaci`, prikazan na slici 5.1. Za moguću ovisnost visine štete o drugim varijablama, pogledajmo grafove u zadnjem redu. Uočavamo da bi mogla postojati linearna veza između visine štete i specijalizacije, odnosno visine štete i osigurane svote, te da je visina štete u oba slučaja pozitivno korelirana s ta dva prediktora.

5.2 Višestruka linearna regresija

Za prilagodbu višestrukog linearnog regresijskog modela metodom najmanjih kvadrata, u R-u koristimo funkciju `lm()`.

```
> model1 = lm(steta ~ ., data = podaci)
> summary(model1)

Call:
lm(formula = steta ~ ., data = podaci)

Residuals:
       Min         1Q     Median         3Q        Max
-9.111e-08 -2.180e-10  1.740e-10  6.000e-10  1.659e-09

Coefficients:
              Estimate Std. Error    t value Pr(>|t|)
(Intercept)  1.478e-08  1.487e-08  9.940e-01    0.321
god_isk     -3.000e+03  1.400e-11 -2.144e+14   <2e-16 ***
lok_cij     -5.923e-10  3.861e-10 -1.534e+00    0.126
nastava     -2.389e-10  6.113e-10 -3.910e-01    0.696
osig_sv      1.300e-01  6.580e-17  1.976e+15   <2e-16 ***
sati_tj      1.000e+03  4.814e-10  2.077e+12   <2e-16 ***
spec         3.000e+05  1.285e-10  2.335e+15   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.152e-09 on 493 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 1.624e+30 on 6 and 493 DF,  p-value: < 2.2e-16
```

Na temelju p-vrijednosti, te na razini značajnosti od npr. 5%, zaključujemo da su prediktori koji su povezani s odazivom sljedeći: godine iskustva (`god_isk`), osigurana svota (`osig_sv`), broj sati rada tjedno (`sati_tj`), te specijalizacija (`spec`). Varijable koje se ne čine značajne su razina lokalnih cijena (`lok_cij`, p-vrijednost 0.126), te bavljenje nastavom (`nastava`, p-vrijednost 0.696). Naime, kod određivanja je li neka varijabla u vezi



Slika 5.1: Rezultat naredbe `pairs(podaci)`.

s odazivom, ispitujemo je li koeficijent uz tu varijablu (npr. β) značajno različit od nule. Drugim riječima, testiramo nul-hipotezu

$$H_0 : \beta = 0$$

nasuprot alternativi

$$H_a : \beta \neq 0.$$

U svrhu testiranja koristimo *t*-statistiku (za detalje upućujemo čitatelja na [3], str. 67-8), te promatramo pripadnu p-vrijednost. P-vrijednost je najmanja razina značajnosti uz koju bi H_0 bila odbačena u korist alternative H_a uz vrijednost opažene testne statistike ([2], str. 80). Prema tome, ako je razina značajnosti 0.05 (tj. 5%), u slučaju prediktora `nastava` ne odbacujemo nul-hipotezu da je koeficijent jednak nuli, dok u slučaju prediktora `spec` odbacujemo nul-hipotezu u korist alternative.

Mogli bismo prilagoditi novi model, u kojem bismo uključili samo one prediktore koji su prema prethodno opisanom u vezi s odazivom. Za to u R-u koristimo sljedeće naredbe:

```
> model2 = lm(steta ~ god_isk + osig_sv +  sati_tj + spec, data = podaci)
> summary(model2)
```

Dobivamo sljedeći izlaz:

```
Call:
lm(formula = steta ~ god_isk + osig_sv + sati_tj + spec, data = podaci)

Residuals:
       Min         1Q     Median         3Q        Max
-9.105e-08 -1.270e-10  2.220e-10  5.700e-10  1.749e-09

Coefficients:
              Estimate Std. Error    t value Pr(>|t|)
(Intercept) -6.050e-09  2.422e-09 -2.498e+00   0.0128 *
god_isk     -3.000e+03  1.401e-11 -2.142e+14   <2e-16 ***
osig_sv      1.300e-01  6.570e-17  1.979e+15   <2e-16 ***
sati_tj      1.000e+03  4.591e-11  2.178e+13   <2e-16 ***
spec         3.000e+05  1.285e-10  2.334e+15   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.157e-09 on 495 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 2.429e+30 on 4 and 495 DF,  p-value: < 2.2e-16
```

Vidimo da su koeficijenti uz ove "relevantne" prediktore jednaki kao i u prethodnom modelu. Također uočavamo da je koeficijent uz godine iskustva negativan, što upućuje na negativnu koreliranost s odazivom. Interpretacija toga je da će liječnik koji ima više godina iskustva u prosjeku imati manji iznos štete nego liječnik s manje iskustva, ako sve druge varijable držimo na istim razinama.

5.3 Odabir najboljeg podskupa

Sada nam je cilj u R-u provesti odabir najboljeg podskupa prediktora, kako je opisano u odjeljku 4.1.1. Koristimo funkciju `regsubsets()` iz paketa `leaps`, koja za dani broj

prediktora $k = 1, 2, ..., p$ (u oznakama algoritma 4.1) identificira najbolji model, gdje je "najbolji" onaj s najmanjom RSS-om (prisjetimo se, RSS je definirana u (2.2) kao $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$).

```
> library(leaps)
> model3 = regsubsets(steta ~ ., data = podaci)
> summary(model3)

Subset selection object
Call: regsubsets.formula(steta ~ ., data = podaci)
6 Variables  (and intercept)
...
1 subsets of each size up to 6
Selection Algorithm: exhaustive
         god_isk lok_cij nastava osig_sv sati_tj spec
1  ( 1 ) " "     " "     " "     " "     " "     "*"
2  ( 1 ) " "     " "     " "     "*"     " "     "*"
3  ( 1 ) "*"     " "     " "     "*"     " "     "*"
4  ( 1 ) "*"     " "     " "     "*"     "*"     "*"
5  ( 1 ) "*"     " "     "*"     "*"     "*"     "*"
6  ( 1 ) "*"     "*"     "*"     "*"     "*"     "*"
```

Prema izlazu funkcije, vidimo da se najbolji model s jednim prediktorom sastoji od prediktora `spec`, najbolji model s dva prediktora od `spec` i `osig_sv`, a najbolji model s tri prediktora od `spec`, `osig_sv` i `god_isk`.

U oznakama algoritma 4.1, došli smo do modela $\mathcal{M}_1, ..., \mathcal{M}_6$, no još je potrebno odrediti koji je od njih najbolji (ne uzimajući u obzir nulti model, $\mathcal{M}_0$). Za to će nam od koristi biti funkcija `summary()`, koja za svaki od modela vraća R^2, RSS, prilagođeni R^2, C_p i BIC.

```
> model3sazetak = summary(model3)
> model3sazetak$rsq
[1] 0.5965252 0.9951734 0.9999512 1.0000000 1.0000000 1.0000000
```

Na primjer, vidimo da R^2 statistika monotono raste od 59,7% u slučaju modela s jednim prediktorom, do 100% u slučaju modela s četiri prediktora. Primijetimo da je riječ o ista četiri prediktora (`god_isk`, `osig_sv`, `sati_tj`, `spec`) do kojih smo došli na kraju prethodnog potpoglavlja, nakon promatranja p-vrijednosti.

Promotrimo i druge statistike. Vrijednosti RSS statistike, u ovisnosti o broju prediktora, su sljedeće:

```
> model3sazetak$rss
[1]  6.774e+13  8.103e+11  8.197e+09  8.491e-16  8.451e-16 -1.563e-02
> which.min(model3sazetak$rss)
[1] 6
```

RSS je najmanja za model sa šest prediktora. Vrijednosti prilagođenog R^2 su sljedeće:

```
> model3sazetak$adjr2
[1] 0.5957150 0.9951540 0.9999509 1.0000000 1.0000000 1.0000000
```

Kod modela s četiri i više prediktora, vrijednost prilagođenog R^2 je 100%. Vrijednosti C_p su:

```
> model3sazetak$cp
[1]  3.953e+31  4.729e+29  4.784e+27  5.548e+00  5.184e+00 -9.119e+15
> which.min(model3sazetak$cp)
[1] 6
```

C_p je najmanji za model sa šest prediktora. Konačno, vrijednosti BIC-a su:

```
> model3sazetak$bic
[1]   -441.39  -2648.17  -4938.79 -33697.27 -33693.44         NaN
> which.min(model3sazetak$bic)
[1] 4
```

BIC je najmanji za model s četiri prediktora. Ove su vrijednosti grafički prikazane na slici 5.2.

Promatrajući grafove, vidimo da se kod modela s dva prediktora RSS i C_p značajno smanje u odnosu na model s jednim prediktorom, dok se tek neznatno smanje pri dodavanju trećeg prediktora. Analogno, prilagođeni R^2 se znatno poveća ako modelu od jednog prediktora dodamo drugi, a tek se neznatno poveća ako dodamo i treći prediktor. U slučaju BIC-a, najveće smanjenje može se uočiti ako se modelu s tri prediktora doda četvrti.

To aktuara vodi do zaključka da odjelu za preuzimanje rizika može za potrebe brzih analiza i izračuna preporučiti jednostavan model od samo dva prediktora (specijalizacija i osigurana svota), što je lakše za implementaciju (npr. manje pitanja u upitniku za klijenta). Za detaljnije analize aktuar se može odlučiti za model s četiri prediktora. Iako su RSS i C_p najmanji kod modela sa šest varijabli, poboljšanje uslijed dodavanja još dva prediktora u model je zanemarivo, te se ti modeli neće uzeti u obzir.

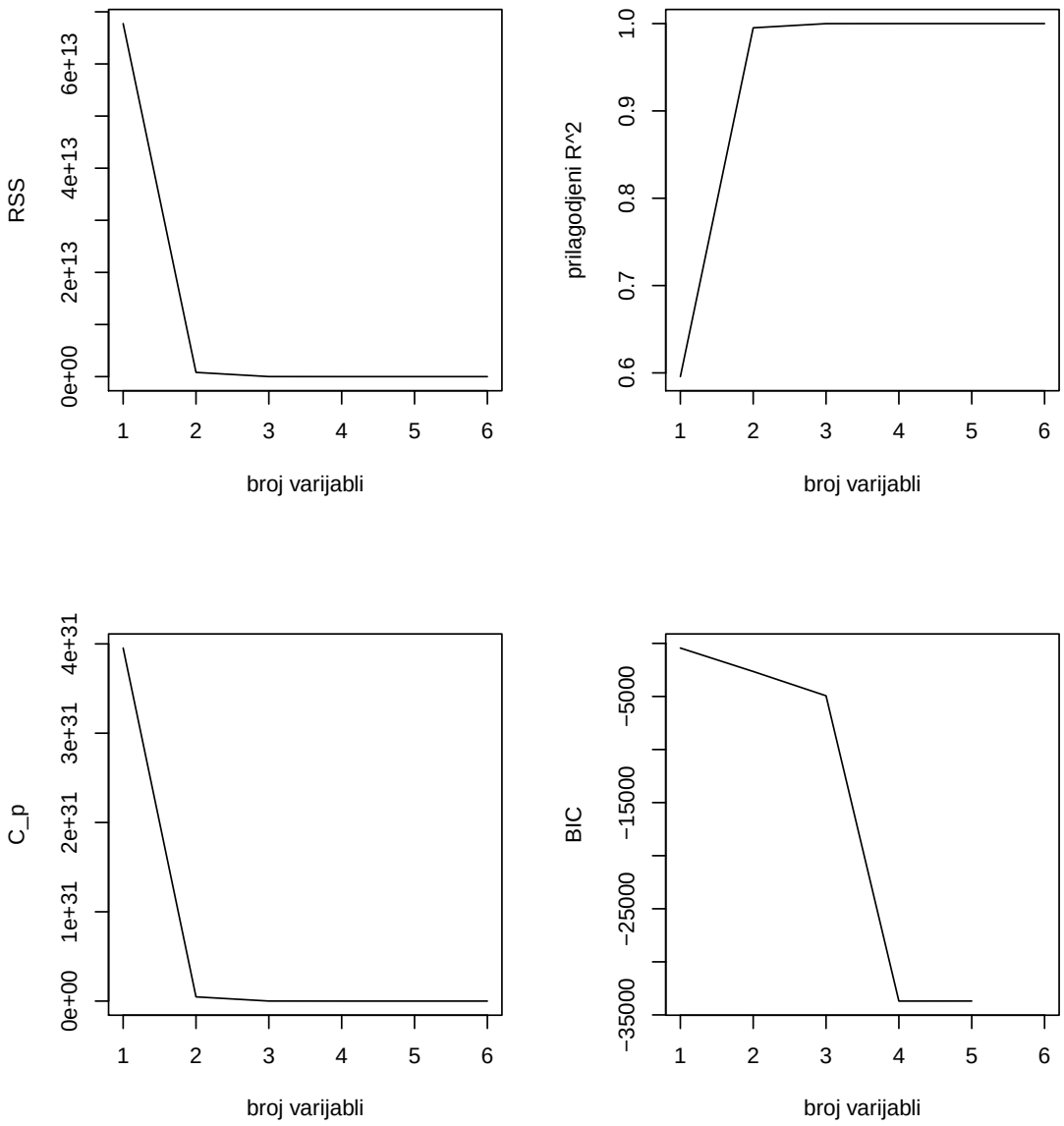
Procjene koeficijenata tih modela možemo očitati u R-u korištenjem funkcije `coef()`. U slučaju modela s dva prediktora, koeficijenti su:

```
> round(coef(model3,2), digits = 2)
(Intercept)     osig_sv        spec
  -14057.51        0.13   299523.73
```

dok su koeficijenti u slučaju modela s četiri prediktora:

```
> round(coef(model3,4), digits = 2)
(Intercept)     god_isk     osig_sv     sati_tj        spec
       0.00    -3000.00        0.13     1000.00   300000.00
```

Primjećujemo da su u drugom slučaju koeficijenti jednaki koeficijentima u modelu na kraju prethodnog potpoglavlja.



Slika 5.2: Vrijednosti statistika RSS, prilagođeni R^2, C_p i BIC u ovisnosti o broju prediktora.

5.4 Postupni odabir

Funkciju `regsubsets()` možemo koristiti i za provođenje postupnog odabira unaprijed, uz dodavanje argumenta `method = "forward"`, odnosno postupnog odabira unatrag, uz dodavanje argumenta `method = "backward"`:

```
> model4 = regsubsets(steta ~ ., data = podaci, method = "forward")
> summary(model4)

Subset selection object
Call: regsubsets.formula(steta ~ ., data = podaci, method = "forward")
6 Variables  (and intercept)
...
1 subsets of each size up to 6
Selection Algorithm: forward
         god_isk lok_cij nastava osig_sv sati_tj spec
1  ( 1 ) " "     " "     " "     " "     " "     "*"
2  ( 1 ) " "     " "     " "     "*"     " "     "*"
3  ( 1 ) "*"     " "     " "     "*"     " "     "*"
4  ( 1 ) "*"     " "     " "     "*"     "*"     "*"
5  ( 1 ) "*"     " "     "*"     "*"     "*"     "*"
6  ( 1 ) "*"     "*"     "*"     "*"     "*"     "*"

> model5 = regsubsets(steta ~ ., data = podaci, method = "backward")
> summary(model5)

Subset selection object
Call: regsubsets.formula(steta ~ ., data = podaci, method = "backward")
6 Variables  (and intercept)
...
1 subsets of each size up to 6
Selection Algorithm: backward
         god_isk lok_cij nastava osig_sv sati_tj spec
1  ( 1 ) " "     " "     " "     " "     " "     "*"
2  ( 1 ) " "     " "     " "     "*"     " "     "*"
3  ( 1 ) "*"     " "     " "     "*"     " "     "*"
4  ( 1 ) "*"     " "     " "     "*"     "*"     "*"
5  ( 1 ) "*"     " "     "*"     "*"     "*"     "*"
6  ( 1 ) "*"     "*"     "*"     "*"     "*"     "*"
```

Uočavamo da se u našem slučaju najbolji modeli koji su odabrani metodama odabira najboljeg podskupa, postupnog odabira unaprijed te postupnog odabira unatrag podudaraju za svaki broj prediktora k. Za ilustraciju skupa podataka gdje to nije slučaj, upućujemo čitatelja na [3], str. 247.

5.5 Metoda validacijskog skupa i unakrsna validacija

U algoritmima 4.1, 4.2 i 4.3 smo napisali da najbolji od modela s različitim brojem prediktora možemo odabrati pomoću C_p-a, BIC-a ili prilagođenog R^2 (što je upravo opisano), ali i pomoću unakrsne validacije. Kako to možemo napraviti, opisujemo u nastavku.

Kako bi metode validacijskog skupa i unakrsne validacije dale dobre procjene testne pogreške, bitno je da u svim koracima prilagodbe modela koristimo samo opažanja iz skupa za učenje, uključujući odabir varijabli. Prema tome, kod određivanja najboljeg modela s danim brojem prediktora moramo koristiti samo opažanja iz skupa za učenje. To je važno, jer ako u koraku odabira najboljeg podskupa koristimo sve dostupne podatke, pogreške na validacijskom skupu i pogreške unakrsne validacije neće biti dobre procjene testne pogreške.

5.5.1 Metoda validacijskog skupa

Za primjenu metode validacijskog skupa, prvo moramo rastaviti opažanja na skup za učenje i na testni skup. U tu svrhu kreiramo slučajni vektor `ucenje` čija je dimenzija jednaka broju opažanja, a koji će imati elemente `TRUE` ako je odgovarajuće opažanje u skupu za učenje, a `FALSE` inače. Vektor `test` imat će vrijednosti `TRUE` ako je opažanje u testnom skupu, a `FALSE` inače. Prije toga, pomoću funkcije `set.seed()` fiksiramo parametar za generator slučajnih brojeva u R-u, kako bi čitatelj mogao dobiti istu podjelu na skup za učenje, odnosno testiranje.

```
> set.seed(1)
> ucenje = sample(c(TRUE, FALSE), nrow(podaci), rep = TRUE)
> test = (!ucenje)
```

Zatim pomoću funkcije `regsubsets()` (vidi potpoglavlje 5.3) provedemo odabir najboljeg podskupa (na razini modela s istim brojem prediktora).

```
> model6 = regsubsets(steta ~ ., data = podaci[ucenje, ])
```

Sintaksa `podaci[ucenje, ]` znači da smo u obzir uzeli samo podatke iz skupa za učenje. Sljedeći je korak računanje pogreške na validacijskom skupu za najbolji model svake veličine. Prvo od podataka za testiranje, korištenjem funkcije `model.matrix()`, napravimo matricu modela (drugi naziv je matrica dizajna). Ukratko: ako imamo n podataka, te ih na primjer modeliramo na sljedeći način:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i \qquad i = 1, ..., n,$$

onda možemo koristiti matrični zapis oblika $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, pri čemu su

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} \\ \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} \text{ te } \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

U tom slučaju matricu $\mathbf{X}$ zovemo matrica modela.[2]

```
> matricamodela = model.matrix(steta ~ ., data = podaci[test, ])
```

Prvih pet redaka te matrice izgleda ovako:

```
> matricamodela[1:5,]

   (Intercept) god_isk lok_cij nastava osig_sv sati_tj spec
2            1      38     101        0 4000000      51    4
5            1      22     102        0 2000000      51    5
9            1      32     103        0 1000000      52    3
10           1      20      98        0 5000000      49    1
16           1       8     100        0 9000000      50    5
```

Sada u `for`-petlji, za svaku veličinu modela `i`, uzmemo koeficijente iz rezultata funkcije `regsubsets()` (spremljene u `model6`) za najbolji model dane veličine (u gornjim oznakama to su elementi matrice $\boldsymbol{\beta}$), pomnožimo ih s odgovarajućim elementima testne matrice modela (u gornjim oznakama $\mathbf{X}$, odnosno `matricamodela`) kako bismo dobili predviđanja za odaziv na testnom skupu ($\hat{y}_i$), te ta predviđanja usporedimo s opaženim vrijednostima (y_i) kako bismo odredili testnu MSE. Prisjetimo se, MSE (srednjekvadratna pogreška) je definirana kao $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. Još napomenimo da je sintaksa za matrično množenje u R-u sljedeća: `A%*%B` za matrični umnožak $\mathbf{AB}$. Dodatno, do imena varijabli u najboljem modelu određene veličine možemo doći pomoću funkcije `names()`. Primjerice, kako bismo vidjeli imena u najboljem modelu s tri prediktora, koristimo sljedeću naredbu:

```
> names(coef(model6, id = 3))
[1] "(Intercept)" "god_isk"     "osig_sv"     "spec"
```

Vidimo da su MSE, počevši od modela s četiri varijable, praktički jednake nuli.

```
> validacijskepogreske = rep(NA, 6)
> for(i in 1:6){
+ koeficijenti_i = coef(model6, id = i)
+ predvidjanje = matricamodela[,names(koeficijenti_i)]%*%koeficijenti_i
+ validacijskepogreske[i] = mean((podaci$steta[test] - predvidjanje)^2)
+ }
> validacijskepogreske
```

[2] Više detalja dostupno je na `https://genomicsclass.github.io/book/pages/expressing_design_formula.html`.

```
[1] 144196610860.15353393554687500000   1715273528.78439378738403320312
[3]     15238858.41943426430225372314            0.00000000000000000035
[5]            0.00000000000000000033            0.00000000000000000030
```

Pogledajmo procjene koeficijenata modela s četiri varijable.

```
> coef(model6, id = 4)
      (Intercept)           god_isk           osig_sv           sati_tj
    -0.0000000014  -3000.0000000000      0.1300000000   1000.0000000000
             spec
300000.0000000002
```

Korištenje metode validacijskog skupa dalo nam je vrijednu informaciju: model s četiri prediktora dobro opisuje podatke. Sada možemo iskoristiti sve dostupne podatke kako bismo dobili preciznije procjene koeficijenata za najbolji model s četiri prediktora.

```
> model7 = regsubsets(steta ~ ., data = podaci)
> coef(model7, id = 4)
      (Intercept)           god_isk           osig_sv           sati_tj
    -0.0000000013  -3000.0000000000      0.1300000000   1000.0000000000
             spec
300000.0000000008
```

No, u slučaju naših podataka, razlike u odnosu na prethodnu procjenu su zanemarive.

5.5.2 Unakrsna validacija

Sada među modelima s različitim brojem prediktora želimo odabrati najbolji model pomoću unakrsne validacije. Koristimo deseterostruku unakrsnu validaciju ($k = 10$). Svako ćemo opažanje svrstati u jedan od deset podskupova, za što ćemo koristiti vektor `podskupovi`.

```
k = 10
set.seed(1)
podskupovi = sample(1:k, nrow(podaci), replace = TRUE)
```

Dodatno, inicijaliziramo matricu u koju ćemo spremati rezultate unakrsne validacije. i-ti redak matrice odgovara i-toj podjeli na skup za učenje i skup za testiranje, dok za tu podjelu, u j-tom stupcu bilježimo testnu MSE za najbolji model s j varijabli.

```
pogreskeUnVal = matrix(NA, k, 6, dimnames = list(NULL, paste(1:6)))
```

Sada u `for`-petlji provodimo unakrsnu validaciju. U i-toj podjeli ($i = 1, ..., 10$), elementi vektora `podskupovi` koji su jednaki i bit će u skupu za testiranje, dok će ostali elementi biti u skupu za učenje. Pomoću modela dobivenog korištenjem podataka iz skupa za učenje napravimo predviđanja za odaziv na skupu za testiranje, te izračunamo testnu pogrešku (predviđanja usporedimo s opaženim vrijednostima). To napravimo za najbolje modele

svih veličina ($j = 1, ..., 6$). Konačno, dobivene testne pogreške spremimo na odgovarajuće, (i, j)-to, mjesto u matrici `pogreskeUnVal`.

```
for(i in 1:k){

# za i-tu podjelu na skup za ucenje i skup za testiranje,
# odredimo najbolje modele koristeci samo podatke iz skupa za ucenje

najboljiModel = regsubsets(steta ~ ., data = podaci[podskupovi != i, ])

# zatim svaki od tih modela primijenimo na skup za testiranje
# (napravimo predvidjanje), i izracunamo MSE

matricamodela = model.matrix(steta ~ ., data = podaci[podskupovi == i, ])

for(j in 1:6){
koeficijenti_j = coef(najboljiModel, id = j)
predvidjanje = matricamodela[,names(koeficijenti_j)] %*% koeficijenti_j
pogreskeUnVal[i,j] = mean((podaci$steta[podskupovi==i] - predvidjanje)^2)
        }

}
```

Tako smo dobili 10×6 matricu, čiji (i, j)-ti element odgovara testnoj MSE za i-tu podjelu na skup za učenje i skup za treniranje za najbolji model s j prediktora:

```
                 1          2        3                              4
 [1,] 440006643052 1638235880 15774282 0.00000000000000000005460424
 [2,] 261247274330 1323455646 10898878 0.00000000000000000001227546
 [3,] 139644247757 1653465737 16734347 0.00000000000000000008316560
 [4,]  45066292324 1889437410 16725957 0.00000000000000000004711496
 [5,]   8458546175 1515977336 18344165 0.00000000000000000001350579
 [6,]   7011899381 2057956586 21078973 0.00000000000000000024018658
 [7,]  42648285119 1733628622 13989525 0.00000000000000000005178795
 [8,] 108971130725 1453564849 15426828 0.00000000000000000013635021
 [9,] 244708853808 1428245226 17586141 0.00000000000000000126643154
[10,] 404060881726 1796574271 18603337 0.00000000000000000038806255
                                    5                              6
 [1,] 0.00000000000000000005555153 0.00000000000000000005672701
 [2,] 0.00000000000000000001338746 0.00000000000000000001482090
 [3,] 0.00000000000000000009223537 0.00000000000000000009223537
 [4,] 0.00000000000000000004844364 0.00000000000000000003922261
 [5,] 0.00000000000000000003437201 0.00000000000000000001824918
 [6,] 0.00000000000000000026210076 0.00000000000000000025616833
 [7,] 0.00000000000000000005074989 0.00000000000000000006955042
 [8,] 0.00000000000000000014707438 0.00000000000000000014247831
 [9,] 0.00000000000000000140560557 0.00000000000000000131740989
[10,] 0.00000000000000000039194931 0.00000000000000000034500131
```

Pomoću funkcije `apply()` izračunamo prosjek svakog stupca matrice. Tako dolazimo do vektora koji na j-tom mjestu ima pogrešku unakrsne validacije za model s j varijabli ($j = 1, ..., 6$), u skladu s formulom (3.2). Vrijednost drugog argumenta ("2") u funkciji `apply()` ispod označava da računamo prosjeke stupaca matrice. Kada bi vrijednost tog argumenta bila "1", računao bi se prosjek redaka.

```
> prosjecneGreske = apply(pogreskeUnVal, 2, mean)
> prosjecneGreske
                                1                                 2
170182405439.6471557617187500000   1649054156.3520154953002929688
                                3                                 4
    16516243.1937167178839445114            0.0000000000000000023
                                5                                 6
           0.0000000000000000025            0.0000000000000000024
```

Dolazimo do istog zaključka kao i na kraju odjeljka 5.5.1, te bi aktuar i na temelju unakrsne validacije odabrao model s četiri prediktora. Prediktori uključeni u model, s odgovarajućim koeficijentima, dani su na kraju odjeljka 5.5.1.

5.6 Ridge i lasso regresija

Za provođenje ridge i lasso regresije koristit ćemo paket `glmnet`. Najvažnija funkcija u tom paketu je `glmnet()`, koja među ostalim služi i za prilagodbu ridge i lasso modela. Ona ima drugačiju sintaksu od prethodno korištenih funkcija: prima matricu `x` i vektor `y`, te se ne koristi sintaksa `y~x`. U nastavku ćemo koristiti ridge i lasso regresiju za predviđanje odaziva (`steta`) na temelju podataka iz skupa `podaci`.

```
x = model.matrix(steta ~ ., data = podaci)[ , -1]
y = podaci$steta
```

"`[ , -1]`" u gornjoj naredbi znači da smo iz matrice modela uklonili prvi stupac, koji se sastoji samo od jedinica, te koji nam u ovom slučaju nije potreban (vidjeti opis matrice modela u odjeljku 5.5.1).

5.6.1 Ridge regresija

Jedan od argumenata funkcije `glmnet()` je `alpha`, koji specificira koji će se model prilagođavati podacima.[3] Ako je `alpha = 0`, koristi se ridge model; ako je `alpha = 1`, koristi se lasso model. Prvo podacima prilagođavamo ridge model.

[3] Više informacija može se pronaći u dokumentaciji paketa, na `https://cran.r-project.org/web/packages/glmnet/glmnet.pdf`.

```
library(glmnet)
mreza = 10^seq(10, -2, length = 100)
ridgeModel = glmnet(x, y, alpha = 0, lambda = mreza)
```

Ako drugačije nije specificirano, funkcija `glmnet()` će provesti ridge regresiju za automatski odabran raspon vrijednosti parametra λ. No, ovdje smo tu funkciju odlučili implementirati za vrijednosti λ koji se kreću od 10^{10} do 10^{-2}, što pokriva puni raspon scenarija od nul-modela koji se sastoji samo od slobodnog člana, do prilagodbe metodom najmanjih kvadrata (napomena: u dokumentaciji paketa `glmnet` piše da je vrijednosti λ potrebno dati u silaznom poretku). Kao što ćemo vidjeti, moguće je provesti prilagodbu modela i za određenu vrijednost parametra λ koja se ne nalazi u vektoru `mreza`.

Uz svaku vrijednost parametra λ vezan je jedan vektor koeficijenata ridge regresije, pohranjen u matrici do koje se može doći pomoću funkcije `coef()`. U našem slučaju, riječ je o matrici sa 7 redaka (po jedan za svaki prediktor, te za slobodni član) i 100 stupaca (po jedan za svaku vrijednost λ).

```
> dim(coef(ridgeModel))
[1]   7 100
```

Očekujemo da će procjene koeficijenata biti manje (u smislu ℓ_2 norme) u slučaju kada je korištena velika vrijednost parametra λ, nego kada je korištena mala vrijednost λ. Na primjer, ako je $\lambda = 3053856$, onda su koeficijenti, te njihova ℓ_2 norma, sljedeći:

```
> ridgeModel$lambda[30]
[1] 3053856

> coef(ridgeModel)[ , 30]
     (Intercept)          god_isk          lok_cij          nastava
1421645.28801856    -115.68463587    -467.82507583   -2520.94402801
         osig_sv          sati_tj             spec
      0.02131014    -555.19417878   49026.25212411

> sqrt(sum((coef(ridgeModel)[-1,30])^2))
[1] 49096.53
```

Nasuprot tome, ako je $\lambda = 2.66$, onda su koeficijenti i njihova ℓ_2 norma kako slijedi:

```
> ridgeModel$lambda[80]
[1] 2.656088

> coef(ridgeModel)[ , 80]
   (Intercept)         god_isk         lok_cij         nastava
 -6607.8204700  -2999.8505872    174.1617002     15.7014383
       osig_sv         sati_tj            spec
     0.1300005    783.7834179 299997.9261886
```

```
> sqrt(sum((coef(ridgeModel)[-1,80])^2))
[1] 300014
```

Uočimo da je uz ovu manju vrijednost parametra λ vezana znatno veća ℓ_2 norma koeficijenata.

Pomoću funkcije `predict()` možemo doći do koeficijenata ridge regresije u slučaju nove vrijednosti λ, na primjer 120:

```
> predict(ridgeModel, s = 120, type = "coefficients")

(Intercept)  -8070.7374346
god_isk      -2998.6939570
lok_cij        222.1257292
nastava         16.9179737
osig_sv          0.1299751
sati_tj        722.9762332
spec        299938.6125817
```

Procjene koeficijenata u ovisnosti o λ smo grafički prikazali na slici 5.3. Zbog razlike u redu veličine koeficijenata, koristimo dva grafa. Pritom smo koristili naredbe:

```
par(mfrow = c(2,1))
plot(ridgeModel, xvar = "lambda", ylim = c(-4500, 301000), lwd = 3,
     ylab = "koeficijenti", xlab = expression(paste("log(",lambda,")")))
text(0, 290000, "spec")
plot(ridgeModel, xvar = "lambda", ylim = c(-4500, 1200),lwd = 3,
     ylab = "koeficijenti", xlab = expression(paste("log(",lambda,")")))
text(c(0,0,16.5,0,0,20.5), c(-2800, 350, -3000, -200, 1000, 1000),
     c("god_isk", "lok_cij", "nastava", "osig_sv", "sati_tj", "spec"))
```

Podijelimo sada podatke, kao i ranije, na skup za učenje i skup za testiranje, kako bismo mogli procijeniti testnu pogrešku ridge i lasso regresije.

```
> set.seed(1)
> ucenje = sample(c(TRUE, FALSE), nrow(podaci), rep = TRUE)
> test = (!ucenje)
```

Ridge regresijski model ćemo prilagoditi podacima iz skupa za učenje, te procijeniti testnu MSE na skupu za testiranje, uz proizvoljno odabran $\lambda = 10$. Ponovno ćemo koristiti funkciju `predict()`, pri čemu sada, kako bismo dobili predviđanja na testnom skupu, umjesto argumenta `type = "coefficients"` koristimo argument `newx`, pomoću kojega specificiramo nove vrijednosti od `x` koje će se koristiti za testiranje.

```
> ridgeModel2 = glmnet(x[ucenje,], y[ucenje], alpha = 0, lambda = mreza)
> ridgePredv = predict(ridgeModel2, s = 10, newx = x[test,])
> mean((ridgePredv - y[test])^2)

[1] 19475.17
```

Testna MSE iznosi 19475.17. U slučaju nul-modela, koji se sastoji samo od slobodnog člana, te koji odgovara ridge modelu s vrlo velikim λ, testna MSE će biti znatno veća:

```
> ridgePredv2 = predict(ridgeModel2, s = 10^10, newx = x[test,])
> mean((ridgePredv2 - y[test])^2)

[1] 343118949956
```

Prema tome, prilagodba ridge regresijskog modela s $\lambda = 10$ vodi do znatno manje testne MSE nego prilagodba modela koji se sastoji samo od slobodnog člana. Promotrimo što je u slučaju prilagodbe metodom najmanjih kvadrata, koja odgovara ridge modelu s $\lambda = 0$. Napomena: da bi nam funkcija `predict()` dala baš rezultate regresije metodom najmanjih kvadrata, potrebno je u poziv funkcije dodati argument `exact = T`.

```
> ridgePredv3 = predict(ridgeModel2, s = 0, newx = x[test,], exact = T,
                x = x[ucenje,], y = y[ucenje])
> mean((ridgePredv3 - y[test])^2)

[1] 8100.141
```

Uz gornju podjelu na skup za učenje i testni skup, odredimo za koji λ će testna MSE biti najmanja.

```
> testnaMSE = rep(NA, 100)
> for(i in 1:100){
      ridgePredv_i = predict(ridgeModel2, s = mreza[i], newx = x[test,])
      testnaMSE[i] = mean((ridgePredv_i - y[test])^2)
      }

> min(testnaMSE)
[1] 8374.529

> which.min(testnaMSE)
[1] 100

> mreza[100]
[1] 0.01
```

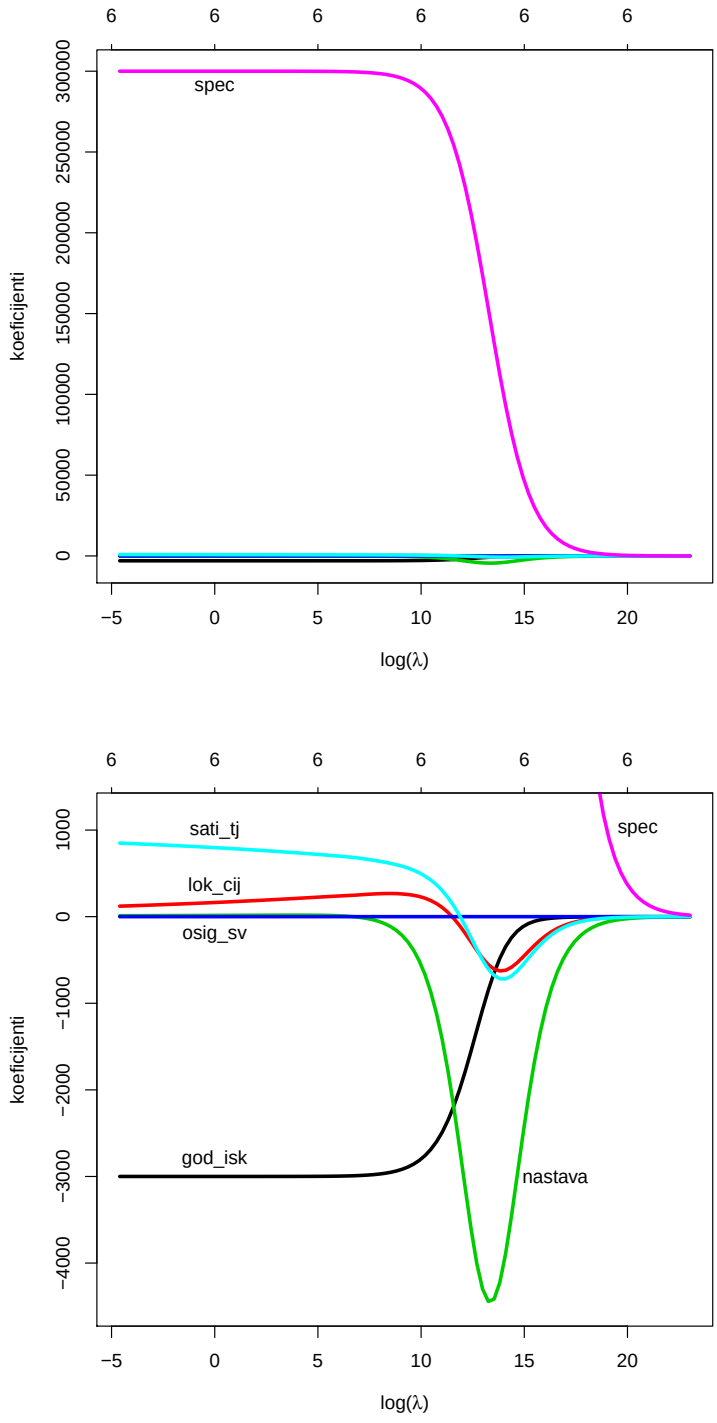
Najmanja testna MSE postiže se za najmanji λ iz vektora `mreza`, koji iznosi 0.01. To vidimo i grafički na slici 5.4. No, za $\lambda = 0$, odnosno u slučaju regresije metodom najmanjih kvadrata, testna MSE je još manja, jer MSE monotono opada kako se λ smanjuje. Strogo govoreći, zaključak bi mogao biti da u našem konkretnom slučaju ridge regresija ne daje bolje rezultate na testnom skupu od metode najmanjih kvadrata. No, pogledajmo pobliže sliku 5.4! Vidimo da su za λ za koje je $\log(\lambda)$ otprilike ≤ 10 srednjekvadratne pogreške relativno malene. Zbog toga bismo mogli argumentirati u korist ridge modela sa $\log(\lambda) = 10$, odnosno $\lambda = 22026$: zadržavamo prednosti modela sa smanjenim koeficijentima u

odnosu na metodu najmanjih kvadrata (što smanjuje rizik pretreniranosti/overfittinga), te još uvijek imamo relativno malu testnu pogrešku. U slučaju ridge modela sa $\lambda = 22026$, procjene koeficijenata su sljedeće:

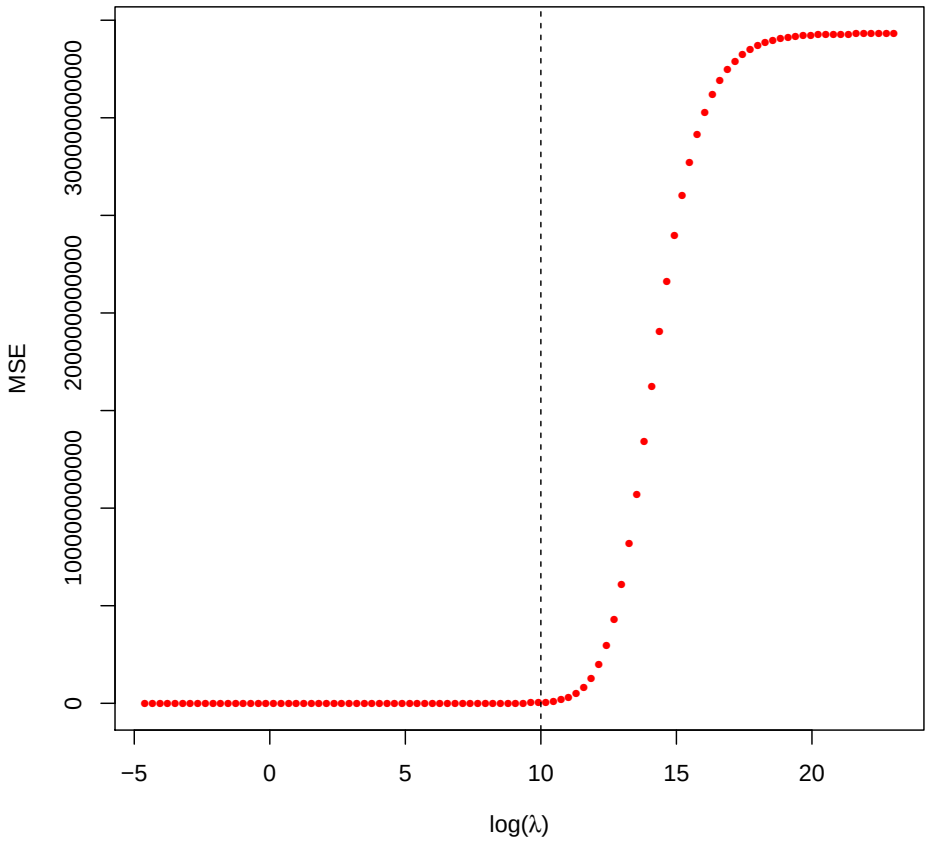
```
> predict(ridgeModel, s = 22026, type = "coefficients")

(Intercept)  56003.5031412
god_isk      -2798.7125472
lok_cij        227.4022818
nastava       -556.4277675
osig_sv          0.1253553
sati_tj        495.5661429
spec        289304.8333583
```

Uočimo da niti jedan od koeficijenata nije jednak nuli, jer ridge regresija ne vrši odabir prediktora. U slučaju ovog modela, testna MSE je sljedeća:

```
> ridgePredv4 = predict(ridgeModel2, s = 22026, newx = x[test,])
> mean((ridgePredv4 - y[test])^2)
[1] 438611030
```



Slika 5.3: Koeficijenti ridge regresije u ovisnosti o λ. Zbog razlike u redu veličine koeficijenata, za prikaz koristimo dva grafa (s različitim skalama na y-osi). Na x osi dan je $\log(\lambda)$. Uočimo: kako λ ide od 10^{-2} do 10^{10}, tako $\log(\lambda)$ ide od -4.61 do 23.03.



Slika 5.4: Prikaz testne MSE u odnosu na $\log(\lambda)$ u slučaju ridge regresije.

5.6.2 Lasso regresija

Za provedbu lasso regresije, također ćemo koristiti funkciju `glmnet()`, samo što će sada biti `alpha = 1` (kako je prethodno navedeno).

```
lassoModel = glmnet(x, y, alpha = 1, lambda = mreza)
lassoModel2 = glmnet(x[ucenje,], y[ucenje], alpha = 1, lambda = mreza)
```

Procjene koeficijenata lasso regresije u ovisnosti o λ prikazane su na slici 5.5. Zbog razlike u redu veličine koeficijenata, koristimo dva grafa. Za kreiranje grafova koristili smo naredbe:

```
par(mfrow = c(2,1))
plot(lassoModel, xvar = "lambda", ylim = c(-4500, 301000), lwd = 3,
     ylab = "koeficijenti", xlab = expression(paste("log(", lambda, ")")))
text(0, 290000, "spec")
plot(lassoModel, xvar = "lambda", ylim = c(-3000, 1200),lwd = 3,
     ylab = "koeficijenti", xlab = expression(paste("log(", lambda, ")")))
text(c(0,6,0,0,-3,14.7), c(-2850, 750, 170, -150, 700, 1000),
     c("god_isk", "lok_cij", "nastava", "osig_sv", "sati_tj", "spec"))
```

Primijetimo na slici 5.5 da su kod lasso regresije procjene nekih koeficijenata točno jednake nuli, što kod ridge regresije nije slučaj.

Kao i kod ridge regresije, promotrimo sada testne MSE o ovisnosti o $\log(\lambda)$ na slici 5.6. Uočavamo da i u slučaju lasso regresije za λ takve da je $\log(\lambda) \leq 10$, srednjekvadratne pogreške postaju relativno malene. Zato sada možemo odabrati lasso model sa $\lambda = 22026$. Procjene koeficijenata tog modela dane su u nastavku.

```
> predict(lassoModel, s = 22026, type = "coefficients")

(Intercept)  97766.393287
god_isk      -1196.345747
lok_cij          .
nastava          .
osig_sv          0.121815
sati_tj          .
spec        285019.179204
```

Za taj model, testna MSE jednaka je

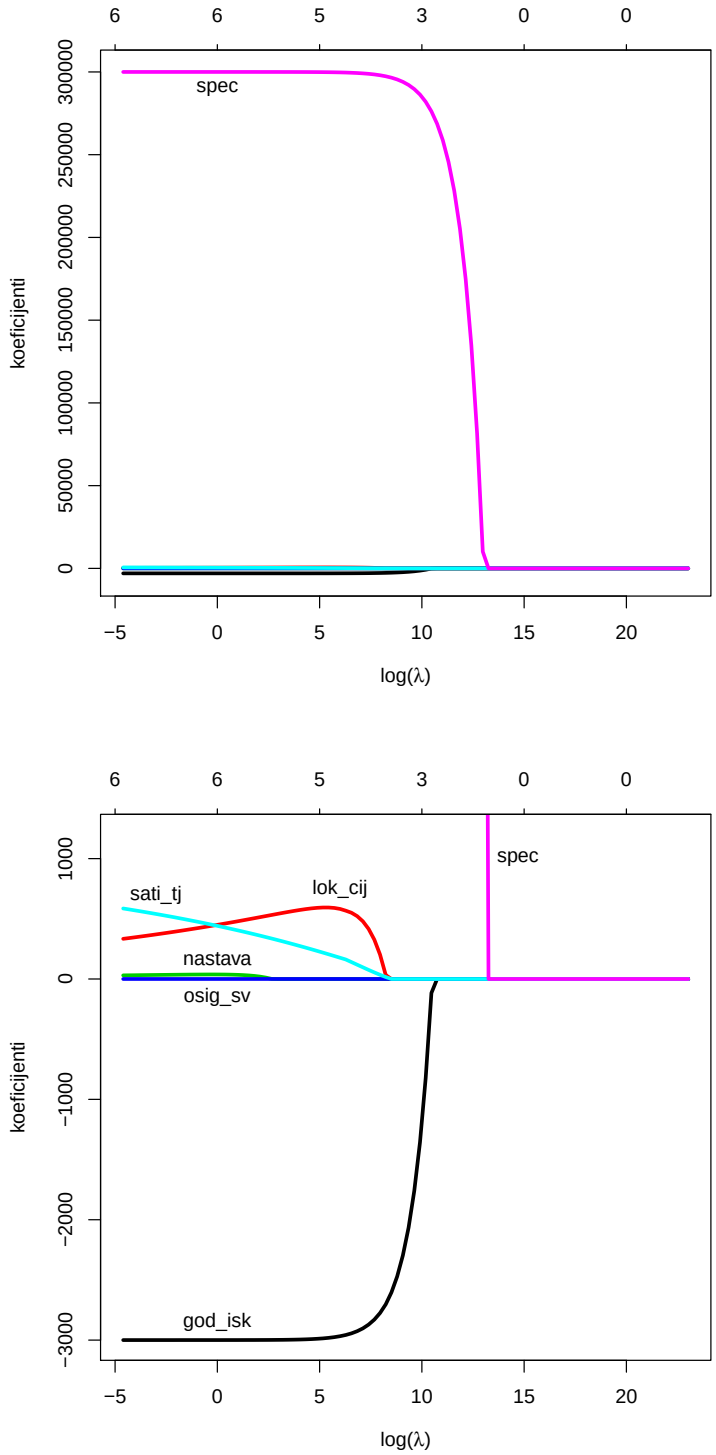
```
> lassoPredv = predict(lassoModel2, s = 22026, newx = x[test,])
> mean((lassoPredv - y[test])^2)
[1] 1396096661
```

Promatrajući procjene koeficijenata, primijetimo najvažniju razliku u odnosu na ridge model s kraja prethodnog odjeljka: procjene koeficijenta uz tri prediktora (`lok_cij`, `nasta-`

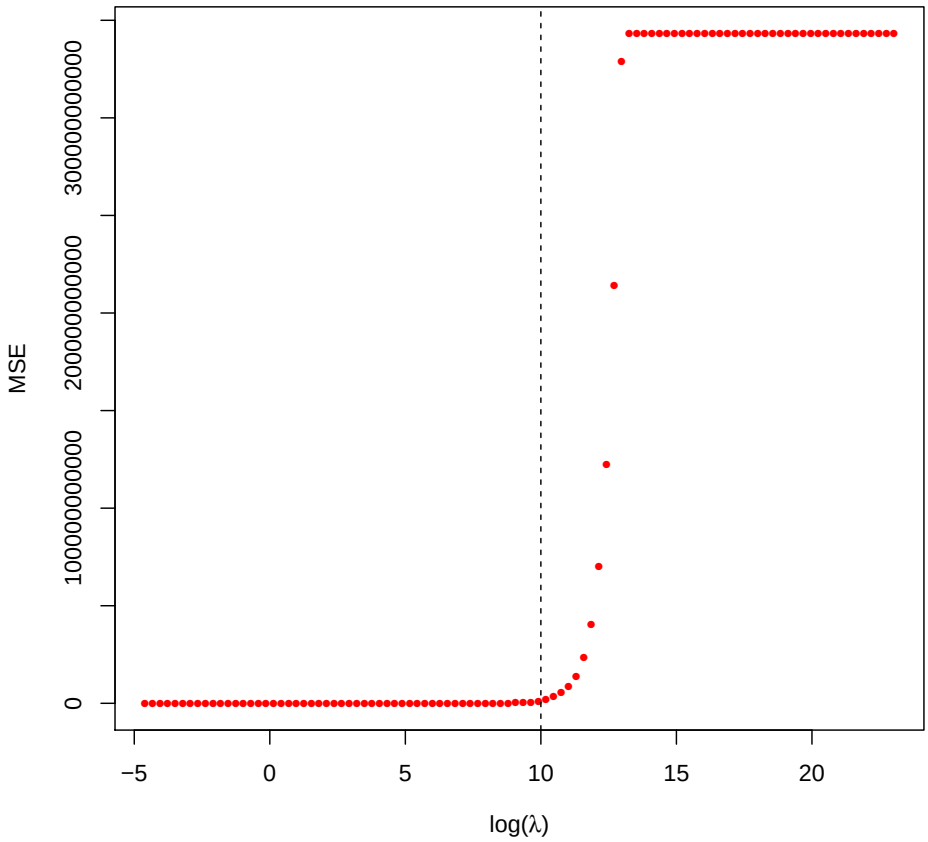
`va` te `sati_tj`) sada su točno jednake nuli (u gornjem izlazu naredbe iz R-a, takve procjene označene su točkom). Drugim riječima, lasso regresija napravila je odabir prediktora. To doprinosi većoj interpretabilnosti modela, te aktuar može odlučiti da u daljnjim analizama više pažnje posveti samo sljedećim odabranim prediktorima: godinama iskustva (`god_isk`), osiguranoj svoti (`osig_sv`) te specijalizaciji (`spec`). Dodatna prednost je ta što aktuar ovaj model može koristiti i za predviđanja na novim podacima zbog relativno male testne srednjekvadratne pogreške, kao što smo vidjeli na slici 5.6. Iako je za $\lambda = 22026$ testna MSE veća nego kod ridge regresije, prednost lasso modela je veća interpretabilnost u odnosu na ridge regresijski model.

Zanimljivo je još primijetiti da su se modeli koje smo odabrali u većini prethodnih potpoglavlja sastojali od četiri prediktora, odnosno uključivali su još broj sati rada tjedno (`sati_tj`). No, prema analizi rezultata lasso regresije, moglo bi se zaključiti da taj prediktor i nije toliko bitan za predviđanje odaziva.

Koji prediktori su stvarno bili u vezi s odazivom pri generiranju podataka? To otkrivamo u nastavku.



Slika 5.5: Koeficijenti lasso regresije u ovisnosti o λ. Zbog razlike u redu veličine koeficijenata, za prikaz koristimo dva grafa (s različitim skalama na y-osi). Na x osi dan je $\log(\lambda)$.



Slika 5.6: Prikaz testne MSE u odnosu na $\log(\lambda)$ u slučaju lasso regresije.

5.7 Generiranje podataka[4]

Sada opisujemo kako su generirani podaci korišteni u prethodnim potpoglavljima.

Prvo napomenimo da smo, zbog toga što su podaci u R-u generirani slučajno, prvo koristili sljedeću naredbu:

```
> set.seed(1)
```

kako bi fiksirali parametar u generatoru slučajnih brojeva u R-u. Na taj način će čitatelj dobiti iste rezultate kao što su ovdje prikazani. Ovu naredbu koristili smo prije svakog poziva funkcija `sample()` i `rnorm()`. Skup generiranih podataka sastoji se od 500 zapisa (`n <- 500`).

Podaci o godinama iskustva generirani su naredbom

```
> god_isk <- sample(0:45, n, replace = TRUE)
```

Drugim riječima, godine iskustva su uniformno odabrane iz skupa $\{0, 1, ..., 45\}$.

Lokalni indeks cijena dolazi iz normalne distribucije s očekivanjem 100 i standardnom devijacijom 5. U našem skupu podataka, vrijednosti se kreću od 85 do 119. U pozadini stoji pretpostavka da su cijene u većini mjesta grupirane oko iste razine, uz manji broj odstupanja, što se može prikladno opisati normalnom distribucijom.

```
> lok_cij <- round(rnorm(n, mean = 100, sd = 5), digits = 0)
> min(lok_cij)
[1] 85
> max(lok_cij)
[1] 119
```

Bavljenje nastavom indicirano je vrijednostima 0 (ne bavi se nastavom) i 1 (bavi se nastavom). Pri tome su vjerojatnosti postavljene tako da je vjerojatnost da se liječnik ne bavi nastavom jednaka 0.9, a da se bavi jednaka 0.1. U našem slučaju, 448 liječnika sudjeluje u izvođenju nastave, dok 52 ne sudjeluje.

```
> nastava <- sample(0:1, n, replace = TRUE, prob = c(0.9, 0.1))
> table(nastava)
nastava
  0   1
448  52
```

S obzirom da svaki liječnik može po svom odabiru ugovoriti visinu pokrića, osigurane svote su uniformno distribuirane na skupu $\{1000000, ..., 10000000\}$.

```
> osig_sv <- sample(1:10, n, replace = TRUE)*1000000
```

Tjedni broj radnih sati je normalno distribuiran s očekivanjem 50 i standardnom devijacijom 4. U našem slučaju, vrijednosti se kreću između 38 i 65.

---

[4]Na ovom mjestu želio bih izraziti zahvalnost kolegici Heidrun König na savjetima pri generiranju podataka i komentarima vezanim uz područje osiguranja od profesionalne odgovornosti liječnika.

```
> sati_tj <- round(rnorm(n, mean = 50, sd = 4), digits = 0)
> min(sati_tj)
[1] 38
> max(sati_tj)
[1] 65
```

Specijalizacija je uniformno distribuirana na skupu $\{1, ..., 5\}$.

```
> spec <- sample(1:5, n, replace = TRUE)
```

Konačno, visina štete određena je na sljedeći način:

```
> steta <- 300000*spec + 0.13*osig_sv - 3000*god_isk + 1000*sati_tj
```

Vidimo da su prediktori koji su povezani s odazivom: specijalizacija, osigurana svota, godine iskustva te broj radnih sati tjedno. Uz godine iskustva stavljen je negativan koeficijent, zbog pretpostavke o negativnoj koreliranosti godina iskustva liječnika i visine štete.

Na kraju smo podatke spremili u data frame `podaci`, koji smo koristili u naredbama iz prethodnih potpoglavlja:

```
podaci <- data.frame(god_isk,lok_cij,nastava,osig_sv,sati_tj,spec,steta)
```

Koliko u prosjeku svaki od prediktora utječe na visinu štete? Promotrimo odnose doprinosa pojedinog prediktora prema prosječnoj šteti (ne uzimajući u obzir predznake).

```
> 300000*mean(spec)/mean(steta)
[1] 0.5634219

> 0.13*mean(osig_sv)/mean(steta)
[1] 0.4502162

> 3000*mean(god_isk)/mean(steta)
[1] 0.04472371

> 1000*mean(sati_tj)/mean(steta)
[1] 0.03108565
```

Broj sati tjedno u prosjeku doprinosi samo oko 3% iznosa visine štete, pa je opravdan zaključak na kraju odjeljka o lasso regresiji, gdje smo taj prediktor isključili iz modela. Također, vidimo da je opravdana preporuka iz potpoglavlja o odabiru najboljeg podskupa da za brze analize koristimo model sa samo dva prediktora (specijalizacija i osigurana svota), jer ta dva prediktora u prosjeku opisuju najveći dio visine odaziva. Iz formule za visinu štete vidimo da je ispravan zaključak iz većine prethodnih potpoglavlja, da preferiramo model s četiri (gore navedena) prediktora. Na kraju, uočimo da su metoda najmanjih kvadrata kao i druge metode dale vrlo dobre procjene koeficijenata modela.

Bibliografija

[1] T. Hastie, R. Tibshirani i Friedman J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2017.

[2] M. Huzak, *Vjerojatnost i matematička statistika, predavanja*, 2006.

[3] G. James, D. Witten, T. Hastie i R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*, Springer, 2017.

Sažetak

U kontekstu linearnih modela, postavlja se pitanje koji od prediktora (kojih može biti vrlo mnogo) su najviše povezani s odazivom. Drugo pitanje koje se javlja je: mogu li se koeficijenti linearnog modela tako prilagoditi, da model daje dobra predviđanja za nove, do sada neviđene podatke?

Da bismo dali odgovore na ta pitanja, prvo uvodimo osnovne pojmove i ideje iz područja statistike i statističkog učenja kao što su kompromis između preciznosti predviđanja i interpretabilnosti modela, nadzirano i nenadzirano učenje, mjerenje kvalitete prilagodbe modela te kompromis između pristranosti i varijance.

Zatim promatramo klasičan linearni regresijski model metodom najmanjih kvadrata, koji će biti temelj za kasnije proširenje.

Nakon toga, opisujemo nekoliko metoda pomoću kojih možemo procijeniti pogrešku na testnom skupu, kao što su metoda validacijskog skupa, pojedinačna unakrsna validacija (LOOCV) te k-struka unakrsna validacija.

U glavnom dijelu rada, prvo promatramo metode odabira podskupa prediktora, kao što su metoda odabira najboljeg podskupa, postupni odabir unaprijed te postupni odabir unatrag. Opisujemo kako odabrati najbolji model korištenjem statistika C_p, BIC i prilagođeni R^2, te pomoću unakrsne validacije. Zatim opisujemo dvije metode smanjenja koeficijenata (regularizacije): ridge i lasso regresiju.

Konačno, opisanu teoriju praktično primjenjujemo u slučaju osiguranja od profesionalne odgovornosti liječnika.

Summary

Selection and Regularization of Linear Models With Actuarial Applications

A question that arises in the context of linear models is: which of the predictors (and there could be many of them) is most closely related to the response? Another question is: could the linear model coefficients be adjusted in a way that the model performs well on previously unseen data?

In order to answer these questions, we first introduce basic concepts and ideas from statistics and statistical learning, such as the trade-off between prediction accuracy and model interpretability, supervised and unsupervised learning, measurement of the quality of fit and the bias-variance trade-off.

We then consider the classical least squares linear regression model, which will be the basis for further extension.

After that, we describe several methods for the test error estimation, such as the validation set approach, the leave-one-out cross-validation (LOOCV), and the k-fold cross-validation.

In the central part of the thesis, we first consider subset selection methods, such as the best subset selection, forward stepwise selection and backward stepwise selection. We describe how to select the best model using the C_p, BIC and adjusted R^2 statistics, as well as using cross-validation. We then describe two shrinkage (or regularization) methods: the ridge regression and the lasso.

Finally, we apply the theory to the case of medical malpractice insurance.

Zahvala

Na ovom mjestu želio bih izraziti zahvalnost mentoru, prof. dr. sc. Bojanu Basraku, na komentarima i savjetima pri pisanju ovog rada, te na spremnosti za vođenje rada “na daljinu”.

Životopis

Ivan Petrunić rođen je u Zagrebu, gdje je završio osnovnu i srednju školu. Na Arhitektonskom fakultetu Sveučilišta u Zagrebu 2011. godine je diplomirao studij arhitekture, te stekao zvanje diplomirani inženjer arhitekture. 2013. godine završio je preddiplomski studij matematike na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta Sveučilišta u Zagrebu. 2015. godine je na Matematičkom odsjeku završio diplomski studij financijske i poslovne matematike, s diplomskim radom iz područja teorije igara i pod mentorstvom doc. dr. sc. Lavoslava Čaklovića. 2018. godine je na Matematičkom odsjeku upisao poslijediplomski specijalistički studij aktuarske matematike.

Od prosinca 2015. do studenog 2018. bio je zaposlen na aktuarskim poslovima u ERGO osiguranju d.d. u Zagrebu. Od prosinca 2018. zaposlen je u području aktuarstva u ERGO Group AG u Düsseldorfu, Njemačka.