Metoda parcijalnih najmanjih kvadrata

Jurić Fot, Sanjin

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Sanjin Jurić Fot

METODA PARCIJALNIH NAJMANJIH KVADRATA

Diplomski rad

Voditelj rada:
doc. dr. sc. Azra Tafro

Zagreb, veljača, 2023.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Antunu Štivičiću

Sadržaj

Uvod

Korisne statističke metode često se razviju izvan statističke zajednice. U ovom ćemo se radu baviti jednom metodom koja svoje korijene vuče iz kemometrije, takozvanom metodom parcijalnih najmanjih kvadrata (PLS). Jedno područje spektrografije bavi se predviđanjem kemijskih sastava tvari preko valnih spektara koje one reflektiraju. Ako se signali za svaku valnu duljinu shvate kao kovarijate, javlja se problem njihove koreliranosti, budući da je broj valnih duljina često veći od broja opažanja.

Problem multikoreliranosti se najčešće rješava regresijom nad glavnim komponentama (PCR) ili Ridgeovom regresijom (RR). PCR prvo pronalazi malen broj nekoreliranih komponenti iz kojih će dobro moći rekonstruirati kovarijate pa iz njih linearnom regresijom predviđa zavisnu varijablu, dok RR uvodi u minimizacijski problem najmanjih kvadrata pribrojnik za L_2 regularizaciju.

PLS rješava taj problem tako što pronalazi nove varijable kojima će modelirati i kemijske i spektralne varijable, dakle varijable koje će istovremeno sadržavati informacije o kovarijatama i zavisnoj varijabli. Kao i u PCR-u, nove varijable su nekorelirane te su nastale linearnom transformacijom originalnih kovarijata, ali za razliku od PCR-a, čije komponente maksimiziraju objašnjenje varijance kod kovarijata, one sadrže i neposredne informacije o zavisnoj varijabli. U literaturi se PLS najčešće predstavlja kao algoritam; pojavljuju se čak dva različita algoritma, za koja ćemo u ovom radu pokazati da su ekvivalentna (po uzoru na [4]).

U prvom poglavlju obradit ćemo linearnu regresiju, jedan od najpoznatijih i najraširenijih statističkih modela uopće; definirat ćemo minimizacijski problem na kojem se ona zasniva, a zatim ga riješiti kako bismo pronašli formulu za njezine koeficijente. U drugom poglavlju bavit ćemo se analizom glavnih komponenti; definirat ćemo glavne komponente minimizacijskim problemom te pokazati da su svojstveni vektori kovarijacijske matrice njegovo jedinstveno rješenje. U trećem poglavlju uvodimo metodu parcijalnih najmanjih kvadrata kao dva algoritma koja kombiniraju metode iz prva dva poglavlja te dokazujemo njihovu ekvivalentnost, rekurzivnu formule za težine i rezultate vezane uz zaustavni kriterij. Konačno, u četvrtom poglavlju primjenjujemo sve tri metode na podatke simulirane u Python-u i analiziramo njihove rezultate.

Poglavlje 1

Linearna regresija

U ovom se poglavlju bavimo jednim od najpoznatijih statističkih predikcijskih modela - linearnom regresijom, čiji je glavni cilj pronalazak linearne transformacije kojom će jednu varijablu maksimalno (u smislu norme) "približiti" drugoj. Većina rezultata u ovom poglavlju preuzeta je iz [7].

1.1 Formulacija problema

Pretpostavimo da su $\boldsymbol{y} \in \mathbb{R}^q$ i $\boldsymbol{x} \in \mathbb{R}^p$ slučajni vektori konačne varijance definirani na istom vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$. Model linearne regresije tada će biti oblika

$$\boldsymbol{y} = \boldsymbol{\beta}'\boldsymbol{x} + \boldsymbol{\alpha}, \tag{1.1}$$

pri čemu vektor $\boldsymbol{\alpha} \in \mathbb{R}^q$ i matricu $\boldsymbol{\beta} \in \mathbb{R}^{p\times q}$ tražimo metodom *najmanjih kvadrata*, odnosno minimizacijom kvadratne greške. Pritom razlikujemo dva slučaja. Ukoliko su nam poznate distribucije od $\boldsymbol{x}$ i $\boldsymbol{y}$, kažemo da je riječ o *populacijskom* slučaju i tada nam je cilj minimizirati očekivanje kvadratne greške, tj. pronaći $\boldsymbol{\alpha} \in \mathbb{R}^q$ i $\boldsymbol{\beta} \in \mathbb{R}^{p\times q}$ td.

$$\phi(\boldsymbol{\alpha}, \boldsymbol{\beta}) := \mathbb{E}\left[(\boldsymbol{y} - \boldsymbol{\beta}'\boldsymbol{x} - \boldsymbol{\alpha})^2\right] \longrightarrow \min. \tag{1.2}$$

S druge strane, ako nam distribucije nisu poznate i imamo samo realizaciju uzorka duljine n danu s

$$\mathbf{X} = \begin{pmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_p \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \in \mathbb{R}^{n\times p},$$

$$\mathbf{Y} = \begin{pmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 & \cdots & \boldsymbol{y}_q \end{pmatrix} = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1q} \\ y_{21} & y_{22} & \cdots & y_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nq} \end{pmatrix} \in \mathbb{R}^{n\times q}, \tag{1.3}$$

riječ je o *uzoračkom* slučaju, i tada minimiziramo ukupnu kvadratnu grešku (ili njezin prosjek, što je ekvivalentno) u odnosu na opažanja, tj. tražimo matricu

$$\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}_1 & \boldsymbol{\beta}_2 & \cdots & \boldsymbol{\beta}_q \end{pmatrix} \in \mathbb{R}^{(p+1)\times q}$$

koja minimizira

$$\left\|\mathbf{Y} - \tilde{\mathbf{X}}\boldsymbol{\beta}\right\|_F{}^1 = \sum_{k=1}^{q} \|\boldsymbol{y}_k - \tilde{\mathbf{X}}\boldsymbol{\beta}_k\|_2^2 = \sum_{k=1}^{q}\sum_{j=1}^{n}\left(y_{jk} - \sum_{l=1}^{p}\tilde{\mathbf{X}}_{jl}\boldsymbol{\beta}_{lk}\right)^2, \tag{1.4}$$

pri čemu je

$$\tilde{\mathbf{X}} = \begin{pmatrix} \mathbb{1} & \mathbf{X} \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \cdots & x_{np} \end{pmatrix} \in \mathbb{R}^{n\times(p+1)}.^2$$

[1] $|\cdot|_F$ označava Frobeniusovu matričnu normu koja se definira kao

$$\|A\|_F = \sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2.$$

[2] Alternativno smo mogli modelirati stvar sličnije populacijskom slučaju, odnosno tražiti $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{p\times q}$ i $\hat{\boldsymbol{\alpha}} \in \mathbb{R}^{n\times q}$ koji minimiziraju

$$\|\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\alpha}}\|_F,$$

ali tada bismo morali uvesti restrikciju da su svi reci od $\hat{\boldsymbol{\alpha}}$ jednaki. U našem uzoračkom modelu prvi redak matrice $\hat{\boldsymbol{\beta}}$ igra ulogu $\boldsymbol{\alpha}$ iz populacijskog modela jer se množi sa stupcem jedinica u $\tilde{\mathbf{X}}$.

U ovom ćemo se poglavlju pretežito baviti populacijskom verzijom problema, dok će analogni rezultati za uzorački slučaj biti izneseni bez dokaza, jer je metoda rješavanja oba problema jednaka - budući da se radi o glatkim funkcijama, njihov minimum tražimo deriviranjem. Osim toga, uzorački je slučaj računski nešto jednostavniji po pitanju oznaka budući da ima dimenziju manje. Ali prije negoli krenemo s izvodima za $\boldsymbol{\alpha}$ i $\boldsymbol{\beta}$, razmotrit ćemo neke uvjete pod kojima modeliranje najmanjim kvadratima ima više smisla.

Jedina restrikcija koju smo dosad uveli na naše varijable je konačnost njihovih varijanci. Ona je dovoljna kako bi naš problem imao rješenje. Drugim riječima, ukoliko neka od varijabli ima beskonačni drugi moment može se dogoditi da je $\phi(\boldsymbol{\alpha}, \boldsymbol{\beta}) = +\infty$ za sve $\boldsymbol{\alpha} \in \mathbb{R}^q$ i $\boldsymbol{\beta} \in \mathbb{R}^{p\times q}$.

Iako se linearnom regresijom u praksi često modeliraju stvari i bez sljedeće pretpostavke, primjena najmanjih kvadrata ima puno više smisla ukoliko pretpostavimo da su $\boldsymbol{x}$ i $\boldsymbol{y}$ *u srednjem linearno povezane*, odnosno ako je

$$\mathbb{E}(\boldsymbol{y}|\boldsymbol{x}) = \boldsymbol{\beta}'\boldsymbol{x} + \boldsymbol{\alpha}.$$

Kad to napišemo u obliku

$$\boldsymbol{y} = \boldsymbol{\beta}'\boldsymbol{x} + \boldsymbol{\alpha} + \boldsymbol{e},$$

iz definicije uvjetnog očekivanja slijedi da je

$$\mathbb{E}(\boldsymbol{e}) = \mathbb{E}(\boldsymbol{y} - \mathbb{E}(\boldsymbol{y}|\boldsymbol{x})) = \boldsymbol{0}. \tag{1.5}$$

Nadalje, ako se odlučimo još postrožiti pretpostavke, smisleno je zahtijevati da za $\mathbb{P}$-g.s. $\boldsymbol{x} \in \mathbb{R}^p$ vrijedi

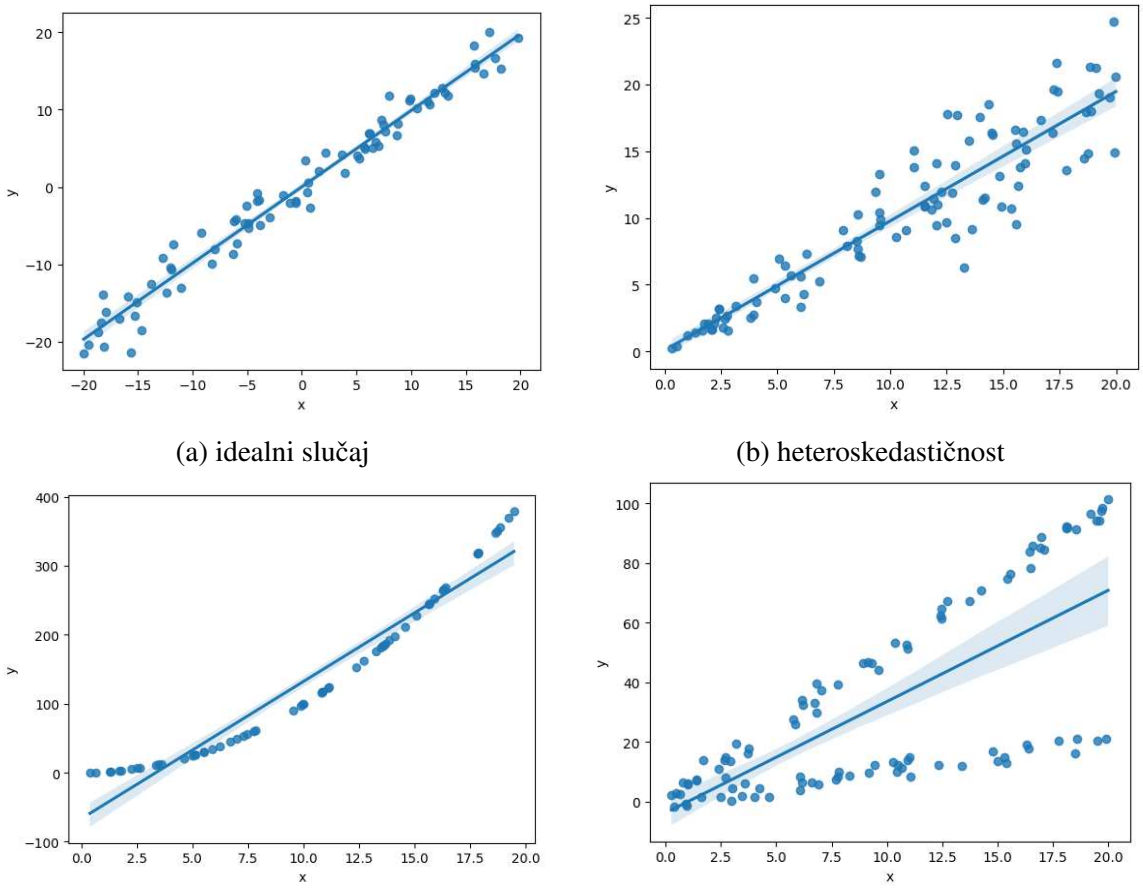
$$\mathbb{E}(\boldsymbol{e}|\boldsymbol{x}) = \boldsymbol{0} \tag{1.6}$$

$$\mathrm{Var}(\boldsymbol{e}|\boldsymbol{x}) = \boldsymbol{\sigma}^2, \quad \text{za neki } \boldsymbol{\sigma}^2 \in \mathbb{R}^{q\times q}. \tag{1.7}$$

Kasnije ćemo pokazati da 1.6 vrijedi ako vrijedi 1.5 te su $\boldsymbol{x}$ i $\boldsymbol{y}$ normalne (što će slijediti iz činjenice da su $\boldsymbol{x}$ i $\boldsymbol{e}$ nekorelirani). S druge strane, hoće li vrijediti 1.7 ili ne, ovisi isključivo o distribuciji podataka. Ta se pretpostavka zove *homoskedastičnost* i ona garantira da će točke svuda uniformno "odskakati" od pravca, što je svakako poželjno svojstvo.

U slučaju modeliranja sa stvarnim podacima zadovoljenje svake od ovih pretpostavki ide u prilog korištenju linearne regresije. Ipak, u nastavku ćemo se ograničiti samo na pretpostavku o konačnosti varijanci, koja nam je nužna kako bi matematički izvodi koji slijede imali smisla jer ona garantira egzistenciju rješenja u 1.2. Rezultati koji slijede vrijede i bez ostalih pretpostavki, iako je te pretpostavke uvijek dobro imati na umu kad odlučujemo hoćemo li za modeliranje koristiti linearnu regresiju ili ne.

Na Slikama 1.1 vidimo nekoliko mogućih odnosa varijabli $\boldsymbol{x}$ i $\boldsymbol{y}$. U slučajevima pod (a) i (b) one su u srednjem linearno povezane, dok su pod (c) i (d) povezane na neki drugi



(a) idealni slučaj

(b) heteroskedastičnost

(c) kvadratna ovisnost

(d) dva pravca

Slika 1.1: Razni slučajevi primjene linearne regresije

način. Pod (*a*) vrijedi homoskedastičnost, dok pod (*b*) ne - varijanca greške se povećava kako x raste. Slika (*c*) prikazuje deterministički odnos varijabli: $y = x^2$. Iako i tad vrijedi $\mathbb{E}(e) = 0$ te su x i e nekorelirane, one nisu nezavisne (za dani x točno znamo koji je e) pa ne vrijedi 1.6. Konačno, slučaj pod (*d*) bilo bi smislenije modelirati s dva pravca nego s jednim, pri čemu bi neka (potencijalno dosad neizmjerena) varijabla z određivala kojem pravcu koje opažanje pripada.

1.2 Izvod koeficijenata

Neka su, dakle,

$$\boldsymbol{x} = (x_1, \ldots, x_p) \in \mathbb{R}^p \quad \text{i} \quad \boldsymbol{y} = (y_1, \ldots, y_q) \in \mathbb{R}^q$$

slučajni vektori čije komponente imaju konačne druge momente, tj.

$$\mathbb{E}\left(x_i^2\right) < +\infty,\ i = 1, \dots p$$
$$\mathbb{E}\left(y_j^2\right) < +\infty,\ j = 1, \dots q.$$

Uvodimo sljedeće oznake:

$$\begin{aligned}\boldsymbol{\mu}_{\boldsymbol{x}} &:= \mathbb{E}(\boldsymbol{x})\\ \boldsymbol{\mu}_{\boldsymbol{y}} &:= \mathbb{E}(\boldsymbol{y})\\ \boldsymbol{\Sigma}_{\boldsymbol{xy}} &:= \mathrm{Cov}(\boldsymbol{x}, \boldsymbol{y})\\ \boldsymbol{\Sigma}_{\boldsymbol{xx}} &:= \mathrm{Var}(\boldsymbol{x})\\ \boldsymbol{\Sigma}_{\boldsymbol{yy}} &:= \mathrm{Var}(\boldsymbol{y}).\end{aligned}$$

Definicija 1.2.1. ***Najbolji linearni predviditelj za y uz dano x*** *je slučajni vektor* $\hat{\boldsymbol{y}} \in \mathbb{R}^q$ *koji je afina fukncija od* $\boldsymbol{x}$ *i koji zadovoljava*

$$\mathbb{E}\left(\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2\right) = \mathbb{E}\left(\left\|\boldsymbol{y} - \hat{\boldsymbol{\beta}}'\boldsymbol{x} - \hat{\boldsymbol{\alpha}}\right\|_2^2\right) = \min_{\boldsymbol{\alpha}\in\mathbb{R}^q,\ \boldsymbol{\beta}\in\mathbb{R}^{p\times q}} \mathbb{E}\left(\left\|\boldsymbol{y} - \boldsymbol{\beta}'\boldsymbol{x} - \boldsymbol{\alpha}\right\|_2^2\right).$$

Najboljeg linearnog predviditelja označavat ćemo $P[\boldsymbol{y}|\boldsymbol{x}]$. *Nadalje, pogreška predviđanja zove se* ***rezidual*** *i označava s*

$$\boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x}) := \boldsymbol{y} - P[\boldsymbol{y}|\boldsymbol{x}].$$

Prije negoli iskažemo i dokažemo teorem koji daje zatvorenu formulu za $\boldsymbol{\alpha}$ i $\boldsymbol{\beta}$, dokazat ćemo jednu korisnu lemu.

Lema 1.2.2. *Za sve slučajne vektore* $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}, \boldsymbol{u}, \boldsymbol{v}$ *te neslučajne matrice i vektore* $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \boldsymbol{a}$ *i* $\boldsymbol{b}$ *za koje su dolje navedeni matrični produkti i zbrojevi dobro definirani vrijedi sljedeće:*

(i) $\mathrm{Cov}(\mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{y} + \boldsymbol{a}, \boldsymbol{z}) = \mathbf{A}\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{z}) + \mathbf{B}\mathrm{Cov}(\boldsymbol{y}, \boldsymbol{z})$

(ii) $\mathrm{Cov}(\boldsymbol{x}, \mathbf{C}\boldsymbol{u} + \mathbf{D}\boldsymbol{v} + \boldsymbol{b}) = \mathrm{Cov}(\boldsymbol{x}, \boldsymbol{u})\mathbf{C}' + \mathrm{Cov}(\boldsymbol{x}, \boldsymbol{v})\mathbf{D}'$

Dokaz. Dokazat ćemo jednakost (*i*) tako lijevu stranu raspišemo po koordinatama.

$$\begin{aligned}[\mathrm{Cov}(\mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{y} + \boldsymbol{a}, \boldsymbol{z})]_{ij} &= \mathrm{Cov}([\mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{y} + \boldsymbol{a}]_i, z_j)\\ &= \mathrm{Cov}([\mathbf{A}\boldsymbol{x}]_i + [\mathbf{B}\boldsymbol{y}]_i + a_i, z_j)\\ &= \mathrm{Cov}\left(\sum_k \mathbf{A}_{ik}x_k + \sum_l \mathbf{B}_{il}y_l + a_i, z_j\right)\end{aligned}$$

Sad korištenjem linearnosti kovarijance slučajnih varijabli dobivamo da je

$$\begin{aligned}[\mathrm{Cov}(\mathbf{A}\boldsymbol{x}+\mathbf{B}\boldsymbol{y}+\boldsymbol{a},\boldsymbol{z})]_{ij} &= \sum_k \mathbf{A}_{ik}\mathrm{Cov}(x_k,z_j)+\sum_l \mathbf{B}_{il}\mathrm{Cov}(y_l,z_j)+\underbrace{\mathrm{Cov}(a_i,z_j)}_{0}\\ &= [\mathbf{A}\mathrm{Cov}(\boldsymbol{x},\boldsymbol{z})]_{ij}+[\mathbf{B}\mathrm{Cov}(\boldsymbol{y},\boldsymbol{z})]_{ij}\\ &= [\mathbf{A}\mathrm{Cov}(\boldsymbol{x},\boldsymbol{z})+\mathbf{B}\mathrm{Cov}(\boldsymbol{y},\boldsymbol{z})]_{ij}.\end{aligned}$$

Tvrdnja pod (*ii*) dokazuje se analogno elementarnim matričnim računom. □

Teorem 1.2.3. *Najbolji linearni predviditelj za* $\boldsymbol{y}$ *uz dano* $\boldsymbol{x}$ *je*

$$P[\boldsymbol{y}|\boldsymbol{x}] = \boldsymbol{\mu}_{\boldsymbol{y}} + \hat{\boldsymbol{\beta}}'(\boldsymbol{x}-\boldsymbol{\mu}_{\boldsymbol{x}}),$$

pri čemu je $\hat{\boldsymbol{\beta}}$ *rješenje jednadžbe*

$$\boldsymbol{\Sigma}_{\boldsymbol{xx}}\hat{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{xy}}. \tag{1.8}$$

Ako je $\boldsymbol{\Sigma}_{\boldsymbol{xx}}$ *regularna, tada je jedinstveno rješenje od 1.8 jednako*

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{xy}}.$$

Dokaz. Raspisivanjem funkcije ϕ iz Definicije 1.2.1 po komponentama dobivamo sljedeće:

$$\begin{aligned}\phi(\boldsymbol{\alpha},\boldsymbol{\beta}) = \mathbb{E}\left(\|\boldsymbol{y}-\boldsymbol{\beta}'\boldsymbol{x}-\boldsymbol{\alpha}\|_2^2\right) &= \sum_{i=1}^{q}\mathbb{E}\left[\left(y_i-\alpha_i-\sum_{j=1}^{p}\beta_{ji}x_j\right)^2\right]\\ &= \sum_{i=1}^{q}\mathbb{E}\left[\left(\underbrace{y_i-\mu_{y_i}-\sum_{j=1}^{p}\beta_{ji}(x_j-\mu_{x_j})}_{\boldsymbol{s}_i}-\underbrace{\left(\alpha_i-\mu_{y_i}+\sum_{j=1}^{p}\beta_{ji}\mu_{x_j}\right)}_{\boldsymbol{t}_i}\right)^2\right]\\ &= \sum_{i=1}^{q}\left[\mathbb{E}\left(\boldsymbol{s}_i^2\right)+2\cdot\mathbb{E}(\boldsymbol{s}_i\boldsymbol{t}_i)+\mathbb{E}\left(\boldsymbol{t}_i^2\right)\right].\end{aligned}$$

Budući da je $\boldsymbol{t}$ neslučajan vektor, vrijedi $\mathbb{E}\left(\boldsymbol{t}_i^2\right) = \boldsymbol{t}_i^2$. Korištenjem linearnosti matematičkog očekivanja sad imamo

$$\begin{aligned}\mathbb{E}(\boldsymbol{s}_i\boldsymbol{t}_i) = \boldsymbol{t}_i\,\mathbb{E}(\boldsymbol{s}_i) &= \boldsymbol{t}_i\,\mathbb{E}\left(y_i-\mu_{y_i}-\sum_{j=1}^{p}\beta_{ji}(x_j-\mu_{x_j})\right)\\ &= \boldsymbol{t}_i\left[\mathbb{E}(y_i-\mu_{y_i})-\sum_{j=1}^{p}\beta_{ji}\,\mathbb{E}(x_j-\mu_{x_j})\right] = 0,\end{aligned}$$

pa je

$$\begin{aligned}\phi(\boldsymbol{\alpha},\boldsymbol{\beta}) &= \sum_{i=1}^{q}\mathbb{E}\left(\boldsymbol{s}_i^2\right)+\sum_{i=1}^{q}t_i^2\\ &= \sum_{i=1}^{q}\mathbb{E}\left[\left(y_i-\mu_{y_i}-\sum_{j=1}^{p}\beta_{ji}(x_j-\mu_{x_j})\right)^2\right]+\sum_{i=1}^{q}\left(\alpha_i-\mu_{y_i}+\sum_{j=1}^{p}\beta_{ji}\mu_{x_j}\right)^2\end{aligned}$$

Budući da je ϕ glatka funkcija definirana na otvorenom skupu $\mathbb{R}^q \times \mathbb{R}^{p\times q}$ (izomorfnom s $\mathbb{R}^{(p+1)\times q}$), točku minimuma tražimo među stacionarnim točkama, tj. rješenjima sustava

$$\phi'(\boldsymbol{\alpha},\boldsymbol{\beta}) = \mathbf{0},$$

koji se sastoji od prvog podsustava

$$\frac{\partial}{\partial\alpha_i}\phi(\boldsymbol{\alpha},\boldsymbol{\beta}) = 2\left(\alpha_i-\mu_{y_i}+\sum_{j=1}^{p}\beta_{ji}\mu_{x_j}\right)=0,\quad 1\le i\le q, \tag{1.9}$$

i drugog podsustava

$$\begin{aligned}\frac{\partial}{\partial\beta_{ik}}\phi(\boldsymbol{\alpha},\boldsymbol{\beta}) &\overset{3}{=} -2\,\mathbb{E}\left[\left(y_i-\mu_{y_i}-\sum_{j=1}^{p}\beta_{ji}(x_j-\mu_{x_j})\right)\cdot(x_k-\mu_{x_k})\right]\\ &\quad+2\left(\alpha_i-\mu_{y_i}+\sum_{j=1}^{p}\beta_{ji}\mu_{x_j}\right)\cdot\mu_{x_k}\\ &\overset{1.9}{=} -2\,\mathbb{E}\left[\left(y_i-\mu_{y_i}-\sum_{j=1}^{p}\beta_{ji}(x_j-\mu_{x_j})\right)\cdot(x_k-\mu_{x_k})\right]\\ &= -2\,\mathbb{E}\left[(y_i-\mu_{y_i})(x_k-\mu_{x_k})\right]-\sum_{j=1}^{p}\beta_{ji}\,\mathbb{E}[(x_j-\mu_{x_j})(x_k-\mu_{x_k})]\\ &= -2\left([\boldsymbol{\Sigma_{yx}}]_{ik}-\sum_{j=1}^{p}\beta'_{ij}[\boldsymbol{\Sigma_{xx}}]_{jk}\right)=0.\end{aligned} \tag{1.10}$$

Iz 1.10 slijedi da za rješenje $\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}$ cijelog sustava vrijedi

$$\boldsymbol{\Sigma_{yx}} = \hat{\boldsymbol{\beta}}'\boldsymbol{\Sigma_{xx}} \iff \boldsymbol{\Sigma_{xy}} = \boldsymbol{\Sigma_{xx}}\hat{\boldsymbol{\beta}},$$

[3] Ovdje u pozadini zapravo koristimo Lebesgueov teorem o dominiranoj konvergenciji. Naime, derivacija je u suštini limes, a očekivanje integral pa ih možemo zamijeniti zbog konačnih drugih momenata od $\boldsymbol{x}$ i $\boldsymbol{y}$.

što je upravo matrična jednadžba 1.8. Iz 1.9 slijedi da za rješenje $\boldsymbol{\alpha} = \hat{\boldsymbol{\alpha}}$ istog sustava vrijedi

$$\hat{\boldsymbol{\alpha}} = \boldsymbol{\mu}_y - \hat{\boldsymbol{\beta}}'\boldsymbol{\mu}_x.$$

Uvrstimo li to u najbolji linearni predviditelj, dobivamo

$$P[\boldsymbol{y}|\boldsymbol{x}] = \hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\beta}}'\boldsymbol{x} = \boldsymbol{\mu}_y + \hat{\boldsymbol{\beta}}(\boldsymbol{x} - \boldsymbol{\mu}_x),$$

što je i trebalo pokazati. □

Napomena 1.2.4. *U uzoračkom slučaju 1.3 najbolji je linearni predviditelj od* $\boldsymbol{y}$ *uz dano* $\boldsymbol{x}$ *dan s*

$$P[\boldsymbol{y}|\boldsymbol{x}] = \overline{\boldsymbol{y}} + (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\mathbf{Y}(\boldsymbol{x} - \overline{\boldsymbol{x}}),$$

pri čemu su $\overline{\boldsymbol{x}} \in \mathbb{R}^p$ *i* $\overline{\boldsymbol{y}} \in \mathbb{R}^q$ *redom prosjeci od* $\boldsymbol{x}$ *i* $\boldsymbol{y}$ *(prosjeci stupaca matrica* $\mathbf{X}$ *i* $\mathbf{Y}$*). Do toga se dolazi traženjem stacionarnih točaka u 1.4. Poveznica s populacijskim modelom je sljedeća:* $\overline{\boldsymbol{y}}$ *je procjenitelj od* $\boldsymbol{\mu}_y$*, uzoračka varijanca* $\mathbf{S}_{xx} = \frac{1}{n-1}\mathbf{X}'\mathbf{X}$ *je procjenitelj od* Σ_{xx}*, dok je uzoračka kovarijanca* $\mathbf{S}_{xy} = \frac{1}{n-1}\mathbf{X}\mathbf{Y}$ *procjenitelj od* Σ_{xy}*. Stoga je smisleno da upravo* $\overline{\boldsymbol{y}}$ *procjenjuje* $\hat{\boldsymbol{\alpha}} = \boldsymbol{\mu}_y$*, a da*

$$\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy} = \left(\frac{1}{n-1}\mathbf{X}'\mathbf{X}\right)^{-1}\frac{1}{n-1}\mathbf{X}\mathbf{Y} = (n-1)(\mathbf{X}'\mathbf{X})^{-1}\frac{1}{n-1}\mathbf{X}\mathbf{Y} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}\mathbf{Y}$$

bude procjenitelj za $\hat{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}$*.*

1.3 Svojstva linearne regresije

Odsad ćemo pretpostavljati da je $\boldsymbol{\Sigma}_{xy}$ pozitivno definitna pa i invertibilna. Prema Teoremu 1.2.3 tada je najbolji linearni predviditelj od $\boldsymbol{y}$ uz dano $\boldsymbol{x}$ jednak

$$P[\boldsymbol{y}|\boldsymbol{x}] = \boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x).$$

Propozicija 1.3.1. *Za najbolji linearni predviditelj* $P[\boldsymbol{y}|\boldsymbol{x}]$ *i pripadni rezidual* $\boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x})$ *vrijedi:*

(i) $\mathbb{E}(P[\boldsymbol{y}|\boldsymbol{x}]) = \boldsymbol{\mu}_y$ *i posljedično* $\mathbb{E}(\boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x})) = \mathbf{0}$*,*

(ii) $\mathrm{Var}(P[\boldsymbol{y}|\boldsymbol{x}]) = \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}$*,*

(iii) $\mathrm{Var}(\boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x})) = \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy}$*,*

(iv) $\mathrm{Var}(\boldsymbol{y}) = \mathrm{Var}(P[\boldsymbol{y}|\boldsymbol{x}]) + \mathrm{Var}(\boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x}))$,

(v) $\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x})) = \mathbf{0}$.

Dokaz. Primjenom linearnosti matematičkog očekivanja imamo da je

$$\mathbb{E}(P[\boldsymbol{y}|\boldsymbol{x}]) = \mathbb{E}(\boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x)) = \boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\,\mathbb{E}(\boldsymbol{x} - \boldsymbol{\mu}_x) = \boldsymbol{\mu}_y,$$

otkud slijedi (*i*). Primjenom svojstava kovarijance iz Leme 1.2.2 lako dobivamo (*ii*):

$$\begin{aligned}
\mathrm{Var}(P[\boldsymbol{y}|\boldsymbol{x}]) &= \mathrm{Var}(\boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x)) \\
&= \mathrm{Cov}(\boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x), \boldsymbol{\mu}_y + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x)) \\
&= \mathrm{Cov}(\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{x}, \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{x}) = \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\mathrm{Var}(\boldsymbol{x})\boldsymbol{\Sigma}_{xx}^{-1}\Sigma_{yx}' \\
&= \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xx}\boldsymbol{\Sigma}_{xx}^{-1}\Sigma_{xy} = \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy},
\end{aligned}$$

a vrlo slično i (*iii*):

$$\begin{aligned}
\mathrm{Var}(\boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x})) &= \mathrm{Var}(\boldsymbol{y} - \boldsymbol{\mu}_y - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x)) \\
&= \mathrm{Cov}(\boldsymbol{y} - \boldsymbol{\mu}_y - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x), \boldsymbol{y} - \boldsymbol{\mu}_y - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x)) \\
&= \mathrm{Var}(\boldsymbol{y}) - \mathrm{Cov}(\boldsymbol{y}, \boldsymbol{x})\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{y}) + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\mathrm{Var}(\boldsymbol{x})\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy} \\
&= \boldsymbol{\Sigma}_{yy} - 2\boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy} + \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy} \\
&= \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\Sigma_{xy}.
\end{aligned}$$

Sada (*iv*) slijedi jednostavno iz prethodne dvije formule:

$$\mathrm{Var}(P[\boldsymbol{y}|\boldsymbol{x}]) + \mathrm{Var}(\boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x})) = \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\Sigma_{xy} + \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}\Sigma_{xy} = \boldsymbol{\Sigma}_{yy} = \mathrm{Var}(\boldsymbol{y}),$$

dok za (*v*) opet koristimo Lemu 1.2.2:

$$\begin{aligned}
\mathrm{Cov}(\boldsymbol{x}, \boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x})) &= \mathrm{Cov}(\boldsymbol{x}, \boldsymbol{y} - \boldsymbol{\mu}_y - \boldsymbol{\Sigma}_{yx}\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_x)) \\
&= \mathrm{Cov}(\boldsymbol{x}, \boldsymbol{y}) - \mathrm{Var}(\boldsymbol{x})\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy} \\
&= \boldsymbol{\Sigma}_{xy} - \boldsymbol{\Sigma}_{xx}\boldsymbol{\Sigma}_{xx}^{-1}\boldsymbol{\Sigma}_{xy} \\
&= \mathbf{0}.
\end{aligned}$$

□

Korolar 1.3.2. *Ako su* $\boldsymbol{x}$ *i* $\boldsymbol{y}$ *normalne, tada je* $\mathbb{E}[\boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x})|\boldsymbol{x}] = \mathbf{0}$.

Dokaz. Ako su $\boldsymbol{x}$ i $\boldsymbol{y}$ onda je i $\boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x})$ normalan kao linearna kombinacija dviju normalnih varijabli, a kako je kod normalnih varijabli nezavisnost ekvivalentna nekoreliranosti, iz Propozicije 1.3.1 (*v*) i (*i*) redom imamo

$$\mathbb{E}[\boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x})|\boldsymbol{x}] = \mathbb{E}[\boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x})] = \mathbf{0}.$$

□

Neka je za $n \in \mathbb{N}$ $\mathcal{L}_n^2$ Hilbertov prostor svih n-dimenzionalnih slučajnih vektora s konačnim drugim momentom koji je opskrbljen skalarnim produktom

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle = \mathbb{E}(\boldsymbol{v}'\boldsymbol{w}), \quad \boldsymbol{v}, \boldsymbol{w} \in \mathcal{L}_n^2,$$

i pripadnom induciranom normom

$$\|\boldsymbol{v}\| = \sqrt{\mathbb{E}\,(\boldsymbol{v}'\boldsymbol{v})}, \quad \boldsymbol{v} \in \mathcal{L}_n^2. \tag{1.11}$$

Tada je očito $\boldsymbol{x} \in \mathcal{L}_p^2$ i $\boldsymbol{y} \in \mathcal{L}_q^2$. Promotrimo linearni potprostor od $\mathcal{L}_q^2$ definiran s

$$\mathcal{K}_{\boldsymbol{x}} = \{\boldsymbol{\alpha} + \boldsymbol{\beta}'\boldsymbol{x} : \boldsymbol{\alpha} \in \mathbb{R}^q, \boldsymbol{\beta} \in \mathbb{R}^{p\times q}\}.$$

Uočimo,

$$\mathrm{Var}(\boldsymbol{\alpha} + \boldsymbol{\beta}'\boldsymbol{x}) \overset{\text{L.1.2.2}}{=} \boldsymbol{\beta}'\Sigma_{\boldsymbol{x}\boldsymbol{x}}\boldsymbol{\beta} < +\infty,$$

pa je zaista $\mathcal{K}_{\boldsymbol{x}} \subseteq \mathcal{L}_q^2$. Nadalje, uistinu je riječ i o vektorskom potprostoru jer je

$$c_1(\boldsymbol{\alpha}_1 + \boldsymbol{\beta}_1'\boldsymbol{x}) + c_2(\boldsymbol{\alpha}_2 + \boldsymbol{\beta}_2'\boldsymbol{x}) = (c_1\boldsymbol{\alpha}_1 + c_2\boldsymbol{\alpha}_2) + (\boldsymbol{\beta}_1 + \boldsymbol{\beta}_2)'\boldsymbol{x} \in \mathcal{K}_{\boldsymbol{x}}$$

za sve $c_1, c_2 \in \mathbb{R}$, $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2 \in R^q$ i $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in \mathbb{R}^{p\times q}$. Sad je iz definicije najboljeg linearnog predviditelja i norme jasno da je

$$P[\boldsymbol{y}|\boldsymbol{x}] = \mathcal{P}_{\mathcal{K}_{\boldsymbol{x}}}(y),$$

pri čemu je $\mathcal{P}_{\mathcal{K}}(y) : \mathcal{L}_q^2 \to \mathcal{L}_q^2$ operator projekcije na $\mathcal{K}_{\boldsymbol{x}}$. Otud trivijalno slijedi idući rezultat.

Korolar 1.3.3. *Za sve neslučajne matrice* $\mathbf{A} \in \mathbb{R}^{q\times p}$ *i vektore* $\boldsymbol{b} \in \mathbb{R}^q$ *vrijedi*

$$P[\mathbf{A}\boldsymbol{x} + \boldsymbol{b}|\boldsymbol{x}] = \mathbf{A}\boldsymbol{x} + \boldsymbol{b}.$$

Sljedeća propozicija govori da bijekcijske linearne transformacije kovarijata ne utječu na najboljeg linearnog predviditelja.

Propozicija 1.3.4. *Za svaku regularnu matricu* $\mathbf{A} \in \mathbb{R}^{p\times p}$ *i vektor* $\boldsymbol{b} \in \mathbb{R}^p$ *vrijedi*

$$P[\boldsymbol{y}|\mathbf{A}\boldsymbol{x} + \boldsymbol{b}] = P[\boldsymbol{y}|\boldsymbol{x}].$$

Dokaz. Neka su $\mathbf{A}$ i $\boldsymbol{b}$ kao u iskazu. Dovoljno je pokazati da je $\mathcal{K}_{\mathbf{A}\boldsymbol{x}+\boldsymbol{b}} = \mathcal{K}_{\boldsymbol{x}}$.

Neka je $\boldsymbol{v} = \boldsymbol{\alpha} + \boldsymbol{\beta}' x \in \mathcal{K}_{\boldsymbol{x}}$ proizvoljan. Tada imamo:

$$\begin{aligned}\boldsymbol{v} &= \boldsymbol{\alpha} + \boldsymbol{\beta}' x = \boldsymbol{\alpha} + \boldsymbol{\beta}' x + \boldsymbol{\beta}'\mathbf{A}^{-1}\boldsymbol{b} - \boldsymbol{\beta}'\mathbf{A}^{-1}\boldsymbol{b} \\ &= \boldsymbol{\alpha} + \boldsymbol{\beta}'\mathbf{A}^{-1}(\mathbf{A}\boldsymbol{x} + \boldsymbol{b}) - \boldsymbol{\beta}'\mathbf{A}^{-1}\boldsymbol{b} \\ &= \boldsymbol{\alpha} - \boldsymbol{\beta}'\mathbf{A}^{-1} + \boldsymbol{\beta}'\mathbf{A}^{-1}(\mathbf{A}\boldsymbol{x} + \boldsymbol{b}) \in \mathcal{K}_{\mathbf{A}\boldsymbol{x}+\boldsymbol{b}}.\end{aligned}$$

Slično, za proizvoljni $\boldsymbol{w} = \boldsymbol{\gamma} + \boldsymbol{\delta}(\mathbf{A}\boldsymbol{x} + \boldsymbol{b}) \in \mathcal{K}_{\mathbf{A}\boldsymbol{x}+\boldsymbol{b}}$ imamo

$$\boldsymbol{w} = \boldsymbol{\gamma} + \boldsymbol{\delta}(\mathbf{A}\boldsymbol{x} + \boldsymbol{b}) = (\boldsymbol{\gamma} + \boldsymbol{\delta}\boldsymbol{b}) + \boldsymbol{\delta}\mathbf{A}\boldsymbol{x} \in \mathcal{K}_{\boldsymbol{x}},$$

otkud slijedi druga inkluzija, a time i tražena tvrdnja. □

Napomena 1.3.5. *Uočimo da prilikom dokazivanja inkluzije* $\mathcal{K}_{\mathbf{A}\boldsymbol{x}+\boldsymbol{b}} \subseteq \mathcal{K}_x$ *nismo koristili regularnost od* $\mathbf{A}$. *To znači da linearna transformacija od* $\boldsymbol{x}$ *ne može poboljšati predviđanje od* $\boldsymbol{y}$ *u smislu najmanjih kvadrata, tj. za sve* $\mathbf{A}$ *i* $\boldsymbol{b}$ *je*

$$\mathbb{E}\left(\|\boldsymbol{e}(\boldsymbol{y}|\mathbf{A}\boldsymbol{x} + \boldsymbol{b})\|_2^2\right) \geq \mathbb{E}\left(\|\boldsymbol{e}(\boldsymbol{y}|\boldsymbol{x})\|_2^2\right).$$

To je i logično, jer linearnom transformacijom ne dobivamo nikakve nove informacije, već ih samo možemo izgubiti. Taj je gubitak proporcionalan defektu od A: ukoliko je $\mathbf{A}$ *regularan, ne gubimo ništa, dok primjerice za* $\mathbf{A} = \mathbf{0}$ *gubimo sve.*

Poglavlje 2

Analiza glavnih komponenti

Pretpostavimo da su x_1, y_2, y i ε slučajne varijable koje zadovoljavaju

$$y = x_1 + 2x_2 + \varepsilon,$$

pri čemu su očekivanja svih varijabli 0 te su x_1, x_2 i ε svi međusobno nezavisni. Tada je

$$\begin{aligned} \mathrm{Cov}(x_1, y) &= \mathbb{E}[x_1(x_1 + 2x_2 + \varepsilon)] = \mathrm{Var}(x_1) \\ \mathrm{Cov}(x_2, y) &= \mathbb{E}[x_2(x_1 + 2x_2 + \varepsilon)] = 2\mathrm{Var}(x_2) \end{aligned}$$

pa imamo

$$\Sigma_{xx} = \begin{pmatrix} \mathrm{Var}(x_1) & 0 \\ 0 & \mathrm{Var}(x_2) \end{pmatrix}, \quad \Sigma_{xy} = \begin{pmatrix} \mathrm{Var}(x_1) \\ 2\mathrm{Var}(x_2) \end{pmatrix}.$$

Stoga sustav1.8 iz Teorema 1.2.3

$$\boldsymbol{\Sigma}_{xx}\boldsymbol{\beta} = \boldsymbol{\Sigma}_{xy} \iff \begin{pmatrix} \mathrm{Var}(x_1) & 0 \\ 0 & \mathrm{Var}(x_2) \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \mathrm{Var}(x_1) \\ 2\mathrm{Var}(x_2) \end{pmatrix}$$

ima jedinstveno rješenje $\hat{\boldsymbol{\beta}} = (1, 2)'$, te je najbolji linearni predviditelj dan s

$$\hat{y} = x_1 + 2x_2.$$

Takav model ima sljedeću interpretaciju: ukoliko se vrijednost varijable x_1 poveća za 1, predviđanje od y povećat će se za 1, a ako povećamo x_2 za 1, $\hat{y}$ raste za 2.

Zamislimo sad sljedeći scenarij: Neka su x_1, x_2, y i ε slučajne varijable koje zadovoljavaju

$$\begin{aligned} y &= x_1 + 2x_2 + \varepsilon \\ x_1 &= \frac{1}{2}x_2. \end{aligned} \tag{2.1}$$

Kao i ranije pretpostavljamo da je očekivanje svih varijabli 0 te da je ε nezavisan od x_1 i x_2. Uvodimo oznaku $\sigma^2 := \mathrm{Var}(x_1) > 0$. Tada je

$$\begin{aligned} \mathrm{Cov}(x_1, y) &= \mathbb{E}[x_1 y] = \mathbb{E}[x_1(x_1 + 2x_2 + \varepsilon)] = \mathbb{E}\left[x_1\left(x_1 + 4x_1 + \varepsilon\right)\right] = 5\sigma^2 \\ \mathrm{Cov}(x_2, y) &= \mathbb{E}[x_2 y] = \mathbb{E}[x_2(x_1 + 2x_2 + \varepsilon)] = \mathbb{E}[2x_1(x_1 + 4x_1 + \varepsilon)] = 10\sigma^2 \\ \mathrm{Cov}(x1, x2) &= \mathbb{E}[x_1 x_2] = \mathbb{E}\left[2x_1^2\right] = 2\sigma^2 \\ \mathrm{Var}(x_2) &= \mathbb{E}\left[x_2^2\right] = \mathbb{E}\left[4x_1^2\right] = 4\sigma^2. \end{aligned}$$

Stoga je

$$\Sigma_{\boldsymbol{xx}} = \begin{pmatrix} \sigma^2 & 2\sigma^2 \\ 2\sigma^2 & 4\sigma^2 \end{pmatrix}, \quad \Sigma_{\boldsymbol{xy}} = \begin{pmatrix} 5\sigma^2 \\ 10\sigma^2 \end{pmatrix},$$

te je vektor $\hat{\boldsymbol{\beta}}$ iz Teorema 1.2.3 rješenje sustava

$$\begin{pmatrix} \sigma^2 & 2\sigma^2 \\ 2\sigma^2 & 4\sigma^2 \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} 3\sigma^2 \\ 6\sigma^2 \end{pmatrix}.$$

Taj je sustav ekvivalentan jednadžbi

$$\beta_1 = 3 - 2\beta_2, \tag{2.2}$$

pa ima beskonačno mnogo rješenja. Za koje god se rješenje $\hat{\boldsymbol{\beta}}$ odlučimo pri konstrukciji modela, gornja interpretacija više neće važiti. Povećanje varijable x_1 dovest će do povećanja y za 5 (jer će se zbog 2.1 x_2 povećati za 2), dok će povećanje x_2 za 1 povisiti y za $\frac{5}{2}$. Međutim, $\hat{\boldsymbol{\beta}} = \left(5, \frac{5}{2}\right)'$ očito ne zadovoljava 2.2 niti ima smisla s obzirom na odnos varijabli. Ali čak ni sva rješenja od 2.2 ne opisuju dobro stvarne ovisnosti. Ako primjerice stavimo $\boldsymbol{\beta} = (-1, 2)'$, naš će model sugerirati da su x_1 i y negativno korelirane, dok je u stvarnosti obrnuto.

U praksi ćemo rijetko naići na to da su varijable x_1 i x_2 savršeno korelirane. Čak i kad se susretnemo s nečim takvim, to je lako riješiti: možemo izbaciti bilo koju od tih varijabli iz modela bez ikakvog gubitka informacija. Problem nastaje kad su dvije varijable visoko korelirane, ali im je korelacija manja od 1. Tada ne možemo izbaciti nijednu bez nekog

gubitka informacija. Nadalje, u uzoračkom slučaju je tada matrica $\mathbf{X}'\mathbf{X}$ "skoro" singularna, a invertiranje takvih matrica je numerički nestabilno (v. str. 192 u [1]).

Jedno od rješenja za visoku koreliranost kovarijata leži u tzv. algoritmima za smanjenje dimenzije, čiji je cilj transformirati p-dimenzionalni vektor $\boldsymbol{x}$ u k-dimenzionalni vektor $\boldsymbol{u}$, $k \leq p$, čije će komponente biti nekorelirane, a da se pritom izgubi minimalno informacija. (U našem bismo primjeru 2.1 mogli staviti $k = 1$, $u_1 = x_1$, no situacija obično nije tako jednostavna.) U ovom poglavlju bavimo se jednim takvim algoritmom zvanim analiza glavnih komponenti (engl. *Principal component analysis*, PCA). Rezultati koji slijede preuzeti su većinom iz [7].

2.1 Formulacija problema

Pretpostavimo da opažamo vektor $x \in \mathbb{R}^p$, pri čemu je p velik. Željeli bismo broj varijabli reducirati na neki manji broj (nekih) varijabli, a da pri toj transformaciji izgubimo minimalno informacija.

Analiza glavnih komponenti pronalazi te nove varijable kao linearnu kombinaciju postojećih s ciljem da one budu najbolji linearni predviditelji komponenata od $\boldsymbol{x}$. Te se *glavne komponente* definiraju tako da su međusobno nekorelirane i imaju sukcesivno sve manju sposobnost predviđanja komponenata od $\boldsymbol{x}$ [7]. Preciznije, želimo pronaći matricu $\boldsymbol{\xi} = \begin{pmatrix}\boldsymbol{\eta}_1 & \cdots & \boldsymbol{\eta}_p\end{pmatrix}' \in \mathbb{R}^{p\times p}$ koja transformira čitav koordinatni sustav, odnosno daje novi koordinatni prikaz od $\boldsymbol{x}$ oblika

$$\boldsymbol{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_p \end{pmatrix} = \boldsymbol{\xi}\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}'\boldsymbol{\eta}_1 \\ \vdots \\ \boldsymbol{x}'\boldsymbol{\eta}_p \end{pmatrix}.$$

s određenim poželjnim svojstvima. Pretpostavimo da je $x \in \mathcal{L}_p^2$, te da ima pozitivno definitnu kovarijacijsku matricu. Stavimo

$$\boldsymbol{\mu} := \mathbb{E}(\boldsymbol{x}), \quad \boldsymbol{\Sigma} := \mathrm{Var}(\boldsymbol{x}) > 0.$$

Ukoliko potprostore S_i od $\mathbb{R}^p$ definiramo sljedećom rekurzijom

$$\begin{aligned} S_0 &= \mathbb{R}^p \\ S_i &= \left\{\boldsymbol{v} \in \mathbb{R}^p : \mathrm{Cov}(\boldsymbol{x}'\boldsymbol{v}, \boldsymbol{x}'\boldsymbol{\eta}_j) = 0, \;\; 1 \leq j \leq i\right\}, \end{aligned}$$

koordinatne vektore $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_p$ biramo tako da zadovoljavaju sljedeće uvjete:

$$\begin{aligned} \boldsymbol{\eta}_i &\in S_{i-1} \\ \mathbb{E}\left[\|\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{\eta}_i)\|_2^2\right] &= \min_{\boldsymbol{v}\in S_{i-1}} \mathbb{E}\left[\|\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})\|_2^2\right], \;\; 1 \leq i \leq p, \end{aligned} \tag{2.3}$$

Iz definicije potprostora S_i vidimo da će novonastale komponente u_i biti nekorelirane. Uvjeti 2.3 u biti garantiraju da ćemo u svakom koraku izabrati komponentu koja linearnom regresijom najbolje predviđa $\boldsymbol{x}$, pod uvjetom da je nekorelirana s prethodnim komponentama. Iduća lema daje nužan i dovoljan uvjet na vektore $\boldsymbol{\eta}_i$ kako bi potonji uvjet bio ispunjen.

Lema 2.1.1. *Komponente vektora*

$$\boldsymbol{u} = (u_1, \ldots, u_p)' = (\boldsymbol{x}'\boldsymbol{\eta}_1, \ldots, \boldsymbol{x}'\boldsymbol{\eta}_p)$$

međusobno su nekorelirane ako i samo ako za sve $i, j \in \{1, \ldots, p\}$ td. $i \neq j$ vrijedi

$$\boldsymbol{\eta}_i \boldsymbol{\Sigma} \boldsymbol{\eta}_j = 0.$$

Dokaz. Za $i \neq j$ vrijedi

$$\mathrm{Cov}(u_i, u_j) = \mathrm{Cov}(\boldsymbol{x}'\boldsymbol{\eta}_i, \boldsymbol{x}'\boldsymbol{\eta}_j) = \mathrm{Cov}(\boldsymbol{\eta}_i'\boldsymbol{x}, \boldsymbol{\eta}_j'\boldsymbol{x}) \overset{\text{L.1.2.2}}{=} \boldsymbol{\eta}_i' \mathrm{Var}(\boldsymbol{x}) \boldsymbol{\eta}_j = \boldsymbol{\eta}_i' \boldsymbol{\Sigma} \boldsymbol{\eta}_j,$$

otkud slijedi tvrdnja. □

Ukoliko definiramo skalarni produkt

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle_{\boldsymbol{\Sigma}} := \langle \boldsymbol{v}, \boldsymbol{\Sigma}\boldsymbol{w} \rangle = \boldsymbol{v}'\boldsymbol{\Sigma}\boldsymbol{w},$$

uvjeti 2.3 postaju:

(1) Prvi koordinatni vektor $\boldsymbol{\eta}_1$ određujemo s

$$\mathbb{E}\left[\|\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{\eta}_1)\|_2^2 \right] = \min_{\boldsymbol{v} \in \mathbb{R}^p} \mathbb{E}\left[\|\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})\|_2^2 \right]. \tag{2.4}$$

(2) Za $i \geq 2$ se $\boldsymbol{\eta}_i$ bira tako da je $\boldsymbol{\eta}_i \perp_{\boldsymbol{\Sigma}} \boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{i-1}$ i vrijedi

$$\mathbb{E}\left[\|\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{\eta}_i)\|_2^2 \right] = \min_{\boldsymbol{v} \perp_{\boldsymbol{\Sigma}} \boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{i-1}} \mathbb{E}\left[\|\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})\|_2^2 \right], \quad 1 \leq i \leq p, \tag{2.5}$$

pri čemu $\perp_{\boldsymbol{\Sigma}}$ označava okomitost u odnosu na $\langle \cdot, \cdot \rangle_{\boldsymbol{\Sigma}}$. Preostaje pokazati da takvi koordinantni vektori postoje te ih onda i pronaći.

2.2 Koordinatni vektori

Teorem 2.2.1. *Uvjeti 2.4 i 2.5 ekvivalentni su sljedećem:*

(1) $\boldsymbol{\eta}_1 \neq \mathbf{0}$ *i*

$$\frac{\boldsymbol{\eta}_1'\boldsymbol{\Sigma}^2\boldsymbol{\eta}_1}{\boldsymbol{\eta}_1'\boldsymbol{\Sigma}\boldsymbol{\eta}_1} = \max_{\boldsymbol{v}\neq\mathbf{0}} \frac{\boldsymbol{v}'\boldsymbol{\Sigma}^2\boldsymbol{v}}{\boldsymbol{v}'\boldsymbol{\Sigma}\boldsymbol{v}}, \tag{2.6}$$

(2) za $i \geq 2$ *je* $\boldsymbol{\eta}_i \perp_{\boldsymbol{\Sigma}} \boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{i-1}$ *i vrijedi*

$$\frac{\boldsymbol{\eta}_i'\boldsymbol{\Sigma}^2\boldsymbol{\eta}_i}{\boldsymbol{\eta}_i'\boldsymbol{\Sigma}\boldsymbol{\eta}_i} = \max_{\boldsymbol{v}\perp_{\boldsymbol{\Sigma}}\boldsymbol{\eta}_1,\ldots,\boldsymbol{\eta}_{i-1}} \frac{\boldsymbol{v}'\boldsymbol{\Sigma}^2\boldsymbol{v}}{\boldsymbol{v}'\boldsymbol{\Sigma}\boldsymbol{v}}. \tag{2.7}$$

Dokaz. Primijetimo da se minimizacija u 2.4 i 2.5 uvijek provodi za $\boldsymbol{v} \neq \mathbf{0}$ jer prema definiciji najboljeg linearnog predviditelja za sve $\boldsymbol{v} \in \mathbb{R}^p$ vrijedi

$$\mathbb{E}\left[\|\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})\|_2^2\right] = \mathbb{E}\left[\|\boldsymbol{x} - P[\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v}]\|_2^2\right] \leq \mathbb{E}\left[\|\boldsymbol{x} - \boldsymbol{\mu}\|_2^2\right] = \mathbb{E}\left[\|\boldsymbol{x} - P[\boldsymbol{x}|\boldsymbol{x}'\mathbf{0}]\|_2^2\right].$$

Korištenjem svojstva linearnog operatora traga

$$\operatorname{tr}(\mathbf{AB}) = \operatorname{tr}(\mathbf{BA}) \tag{2.8}$$

i linearnosti matematičkog očekivanja skupa s prethodnim rezultatima dobivamo da za svaki $\boldsymbol{v} \neq \mathbf{0}$ vrijedi

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})\|_2^2\right] &= \mathbb{E}\left[\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})'\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})\right] = \operatorname{tr}\left(\mathbb{E}\left[\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})'\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})\right]\right) \\
&= \mathbb{E}\left[\operatorname{tr}(\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})'\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v}))\right] \overset{2.8}{=} \mathbb{E}\left[\operatorname{tr}(\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})')\right] \\
&= \operatorname{tr}\left(\mathbb{E}\left[\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})'\right]\right) \overset{1.3.1(i)}{=} \operatorname{tr}(\operatorname{Var}(\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v}))) \\
&\overset{1.3.1(iii)}{=} \operatorname{tr}\left(\boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{x}} - \boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{v}'\boldsymbol{x}}\boldsymbol{\Sigma}_{\boldsymbol{v}'\boldsymbol{x},\boldsymbol{v}'\boldsymbol{x}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{v}'\boldsymbol{x},\boldsymbol{x}}\right) \overset{1.2.2}{=} \operatorname{tr}\left(\boldsymbol{\Sigma} - \boldsymbol{\Sigma}\boldsymbol{v}(\boldsymbol{v}'\boldsymbol{\Sigma}\boldsymbol{v})^{-1}\boldsymbol{v}'\boldsymbol{\Sigma}\right) \\
&\overset{\text{lin. tr.}}{=} \operatorname{tr}\boldsymbol{\Sigma} - (\boldsymbol{v}'\boldsymbol{\Sigma}\boldsymbol{v})^{-1}\operatorname{tr}\left(\boldsymbol{\Sigma}\boldsymbol{v}\boldsymbol{v}'\boldsymbol{\Sigma}\right) \overset{2.8}{=} \operatorname{tr}\boldsymbol{\Sigma} - (\boldsymbol{v}'\boldsymbol{\Sigma}\boldsymbol{v})^{-1}\operatorname{tr}\left(\boldsymbol{v}'\boldsymbol{\Sigma}^2\boldsymbol{v}\right) \\
&= \operatorname{tr}\boldsymbol{\Sigma} - \frac{\boldsymbol{v}'\boldsymbol{\Sigma}^2\boldsymbol{v}}{\boldsymbol{v}'\boldsymbol{\Sigma}\boldsymbol{v}}.
\end{aligned}$$

Stoga je minimizacija $\boldsymbol{v} \to \mathbb{E}\left[\|\boldsymbol{e}(\boldsymbol{x}|\boldsymbol{x}'\boldsymbol{v})\|_2^2\right]$ ekvivalentna maksimizaciji funkcije

$$\boldsymbol{v} \to \frac{\boldsymbol{v}'\boldsymbol{\Sigma}^2\boldsymbol{v}}{\boldsymbol{v}'\boldsymbol{\Sigma}\boldsymbol{v}},$$

otkud slijedi tvrdnja teorema. □

Rješenje minimizacijskog problema danog s 2.6 i 2.7 daje sljedeći teorem, čiji se dokaz može pronaći u [8].

Teorem 2.2.2. *Neka su* $\mathbf{P}$ *i* $\mathbf{G}$ *kvadratne matrice istog reda p takve da je* $\mathbf{P}$ *simetrična, a* $\mathbf{G}$ *još i pozitivno definitna matrica. Tada su sljedeće tvrdnje* (*i*) *i* (*ii*) *ekvivalentne:*

(i) Vektori $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_p$ *su takvi da je*

(1) $\boldsymbol{\eta}_1 \neq \mathbf{0}$ *i*

$$\frac{\boldsymbol{\eta}_1' \mathbf{P} \boldsymbol{\eta}_1}{\boldsymbol{\eta}_1' \mathbf{G} \boldsymbol{\eta}_1} = \max_{\boldsymbol{v} \neq \mathbf{0}} \frac{\boldsymbol{v}' \mathbf{P} \boldsymbol{v}}{\boldsymbol{v}' \mathbf{G} \boldsymbol{v}}, \tag{2.9}$$

(2) za $i \geq 2$ *je* $\boldsymbol{\eta}_i \perp_{\mathbf{G}} \boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{i-1}$ *i vrijedi*

$$\frac{\boldsymbol{\eta}_i' \mathbf{P} \boldsymbol{\eta}_i}{\boldsymbol{\eta}_i' \mathbf{G} \boldsymbol{\eta}_i} = \max_{\boldsymbol{v} \perp_{\mathbf{G}} \boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_{i-1}} \frac{\boldsymbol{v}' \mathbf{P} \boldsymbol{v}}{\boldsymbol{v}' \mathbf{G} \boldsymbol{v}}. \tag{2.10}$$

(ii) Vektori $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_p$ *su takvi da je za* $i = 1, \ldots, p$ $\boldsymbol{\eta}_i$ *svojstveni vektor od* $\mathbf{G}^{-1}\mathbf{P}$ *koji odgovara svojstvenoj vrijednosti* ν_i*, pri čemu je* $\nu_1 \geq \nu_2 \geq \cdots \geq \nu_p$ *i* $\boldsymbol{\eta}_i' \mathbf{G} \boldsymbol{\eta}_j = 0$ *za* $i \neq j$*.*

Nadalje, matrica $\mathbf{G}^{-1}\mathbf{P}$ *ima p realnih svojstvenih vrijednosti (računajući kratnosti) te stoga* (*i*) *ima rješenje.*

Ako primijenimo ovaj teorem za $\mathbf{G} = \boldsymbol{\Sigma}$ i $\mathbf{P} = \boldsymbol{\Sigma}^2$, dobivamo da su traženi koordinatni vektori zapravo svojstveni vektori od $\mathbf{G}^{-1}\mathbf{P} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}^2 = \boldsymbol{\Sigma}$. Time smo sljedeću definiciju učinili smislenom.

Definicija 2.2.3. ***Glavne komponente*** *slučajnog vektora* $\boldsymbol{x} \in \mathbb{R}^p$ *su slučajne varijable* $u_i := \boldsymbol{x}' \boldsymbol{\eta}_i$*, pri čemu su* $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_p$ *svojstveni vektori od* $\boldsymbol{\Sigma}$ *koji odgovaraju svojstvenim vrijednostima* ν_i *u silaznom nizu* $\nu_1 \geq \cdots \geq \nu_p$*.*

Iduća propozicija pokazuje da su $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_p$ ortogonalni.

Propozicija 2.2.4. *Neka su* $(\boldsymbol{\eta}_i, \nu_i)$*,* $i = 1, \ldots, p$ *svojstveni parovi pozitivno definitne matrice* $\boldsymbol{\Sigma}$*. Tada je ekvivalentno:*

(i) $i \neq j \implies \boldsymbol{\eta}_i' \boldsymbol{\Sigma} \boldsymbol{\eta}_j = 0$*,*

(ii) $i \neq j \implies \boldsymbol{\eta}_i' \boldsymbol{\eta}_j = 0$*,*

(iii) $i \neq j \implies \boldsymbol{\eta}_i' \boldsymbol{\Sigma}^{-1} \boldsymbol{\eta}_j = 0$*.*

Dokaz. Budući da je $\boldsymbol{\Sigma} > 0$, vrijedi da je $\nu_i > 0$ za svaki $i = 1, \ldots, p$. Stoga je

$$\Sigma^{-1}\boldsymbol{\eta} = \frac{1}{\nu_i}\boldsymbol{\eta}_i,$$

pa su $\boldsymbol{\eta}, \boldsymbol{\Sigma}\boldsymbol{\eta}_i$ i $\boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_i$ kolinearni, što daje tvrdnju. □

Koordintani vektori su, dakle, međusobno okomiti pa i linearno nezavisni. Ubuduće ćemo pretpostavljati da su oni i normirani, tj. da čine ortonormiranu bazu od $\mathbb{R}^p$.

2.3 Regresija na glavnim komponentama

Vratimo se inicijalnom problemu: za dani $\boldsymbol{x} \in \mathbb{R}^p$ želimo predviđati $\boldsymbol{y} \in \mathbb{R}$. [1] Označimo $\boldsymbol{\sigma} = \mathrm{Cov}(\boldsymbol{x}, y) \in \mathbb{R}^p$. Kako je $\{\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_K\}$ ortonormirana baza, imamo

$$\boldsymbol{\sigma} = \sum_{k=1}^{K} \langle \boldsymbol{\sigma}, \boldsymbol{\eta}_k \rangle \boldsymbol{\eta}_k, \tag{2.11}$$

pa je koeficijent $\boldsymbol{\beta}_{LS}$ u najboljem linearnom predviditelju dan s

$$\boldsymbol{\beta}_{LS} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma} \overset{2.11}{=} \boldsymbol{\Sigma}^{-1}\left(\sum_{k=1}^{K} \langle \boldsymbol{\sigma}, \boldsymbol{\eta}_k \rangle \boldsymbol{\eta}_k\right) = \sum_{k=1}^{K} \langle \boldsymbol{\sigma}, \boldsymbol{\eta}_k \rangle \boldsymbol{\Sigma}^{-1}\boldsymbol{\eta}_k = \sum_{k=1}^{K} \langle \boldsymbol{\sigma}, \boldsymbol{\eta}_k \rangle \frac{1}{\nu_k}\boldsymbol{\eta}_k. \tag{2.12}$$

Kako bismo dobili što stabilnijeg procjenitelja za $\boldsymbol{\beta}_{LS}$, pokušat ćemo maksimalno smanjiti broj pribrojnika u 2.12. Za početak možemo izbaciti sve $\boldsymbol{\eta}_k$ takve da je $\langle \boldsymbol{\sigma}, \boldsymbol{\eta}_k \rangle = 0$. Nadalje, pokazat ćemo da za višestruke svojstvene vrijednosti možemo imati samo jedan pribrojnik; preciznije, ako je $\nu_{i_1} = \nu_{i_2} = \cdots = \nu_{i_r} = \nu$, tada postoji vektor $\boldsymbol{\eta}$ takav da je

$$\sum_{k=1}^{r} \langle \boldsymbol{\sigma}, \boldsymbol{\eta}_{i_k} \rangle \frac{1}{\nu}\boldsymbol{\eta}_{i_k} = \langle \boldsymbol{\sigma}, \boldsymbol{\eta} \rangle \frac{1}{\nu}\boldsymbol{\eta}. \tag{2.13}$$

Vidimo da je $\boldsymbol{\eta}$ kolinearan sa sumom na lijevoj strani pa postoji $\alpha \in \mathbb{R}$ takav da je

$$\boldsymbol{\eta} = \alpha \sum_{k=1}^{r} \langle \boldsymbol{\sigma}, \boldsymbol{\eta}_{i_k} \rangle \boldsymbol{\eta}_{i_k}.$$

[1] U ostatku rada pretpostavljat da je y slučajna varijabla, jer je generalizacija za slučajne vektore jasna (istom metodom kojom predviđamo jednodimenzionalne varijable možemo predviđati vektorske komponente y_i), a oznake su dosta jednostavnije.

2.13 tada postaje

$$\sum_{k=1}^{r}\langle\boldsymbol{\sigma},\boldsymbol{\eta}_{i_k}\rangle\boldsymbol{\eta}_{i_k} = \left\langle \alpha\sum_{k=1}^{r}\langle\boldsymbol{\sigma},\boldsymbol{\eta}_{i_k}\rangle\boldsymbol{\eta}_{i_k},\boldsymbol{\sigma}\right\rangle\cdot\alpha\cdot\sum_{k=1}^{r}\langle\boldsymbol{\sigma},\boldsymbol{\eta}_{i_k}\rangle\boldsymbol{\eta}_{i_k}.$$

Uzimanjem normi obiju strana dobivamo:

$$\alpha^2 = \frac{1}{\sum_{k=1}^{r}\langle\boldsymbol{\sigma},\boldsymbol{\eta}_{i_k}\rangle^2},$$

odnosno

$$\boldsymbol{\eta} = \pm\frac{\sum_{k=1}^{r}\langle\boldsymbol{\sigma},\boldsymbol{\eta}_{i_k}\rangle\boldsymbol{\eta}_{i_k}}{\sqrt{\sum_{k=1}^{r}\langle\boldsymbol{\sigma},\boldsymbol{\eta}_{i_k}\rangle^2}}.$$

Uočimo, $\boldsymbol{\eta}$ je normiran (u normi $\|\cdot\|_2$). Također, kao element linearne ljuske od $\{\boldsymbol{\eta}_{i_k},\ldots,\boldsymbol{\eta}_{i_r}\}$ ortogonalan je na preostale vektore baze te je ujedno svojstveni vektor pridružen svojstvenoj vrijednosti ν. Nadalje, $\boldsymbol{\eta}$ je jedinstven do na predznak (jer u gornji smo račun krenuli samo s nužnim uvjetom 2.13). Time dobivamo (gotovo) jedinstven rastav

$$\boldsymbol{\beta}_{LS} = \sum_{k=1}^{M}\frac{\langle\boldsymbol{\sigma},\boldsymbol{\eta}_k\rangle}{\nu_k}\boldsymbol{\eta_k}, \tag{2.14}$$

u kojem su svi pribrojnici netrivijalni, $\boldsymbol{\eta}_k$ su ortonormirani, a ν_k međusobno različiti.

Definicija 2.3.1. *Svojstveni vektori $\boldsymbol{\eta_k}$ koji se pojavljuju u 2.14 nazivaju se **relevantni svojstveni vektori**, dok se pripadni koeficijenti $u_k = \langle\boldsymbol{\eta_k},\boldsymbol{x}-\boldsymbol{\mu_x}\rangle$ nazivaju **relevantne komponente** od $\boldsymbol{x}$ za predviđanje y.*

Tada imamo sljedeći rastav od $\boldsymbol{x}$:

$$\begin{aligned}\boldsymbol{x} &= \boldsymbol{x}-\boldsymbol{\mu_x}+\boldsymbol{\mu_x} = \boldsymbol{\mu_x}+\sum_{k=1}^{K}\langle\boldsymbol{x}-\boldsymbol{\mu_x},\boldsymbol{\eta}_k\rangle\boldsymbol{\eta}_k\\ &= \boldsymbol{\mu_x}+\sum_{k=1}^{M}\langle\boldsymbol{x}-\boldsymbol{\mu_x},\boldsymbol{\eta}_k\rangle\boldsymbol{\eta}_k+\sum_{k=M+1}^{K}\langle\boldsymbol{x}-\boldsymbol{\mu_x},\boldsymbol{\eta}_k\rangle\boldsymbol{\eta}_k\\ &= \boldsymbol{\mu_x}+\sum_{k=1}^{M}u_k\boldsymbol{\eta}_k+\boldsymbol{e},\end{aligned}$$

pri čemu je

$$\boldsymbol{e} = \sum_{k=M+1}^{K}\langle\boldsymbol{x}-\boldsymbol{\mu_x},\boldsymbol{\eta}_k\rangle\boldsymbol{\eta}_k,$$

dok je linearni model oblika

$$\begin{aligned}\hat{y} &= \mu_y + \boldsymbol{\beta}_{LS}(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}) \\ &= \sum_{k=1}^{M} \frac{\langle \boldsymbol{\sigma}, \boldsymbol{\eta}_k \rangle}{\nu_k} \langle \boldsymbol{\eta}_k, \boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}} \rangle \\ &= \sum_{k=1}^{M} \gamma_k u_k,\end{aligned}$$

gdje je

$$\gamma_k = \frac{\langle \boldsymbol{\sigma}, \boldsymbol{\eta}_k \rangle}{\nu_k},\ k = 1, \ldots, M.$$

Takav se model zove regresija na glavnim komponentama (engl. *Partial component regression*, PCR). On se po predviđanjima koja daje u teoriji ne razlikuje od linearne regresije, ali je zbog nekoreliranosti komponenata u_i puno interpretabilniji. Ukoliko su početne veličine $\boldsymbol{x}$ i $\boldsymbol{y}$ normalne, tada su takve i u_i pa su one i nezavisne. Stoga, kad povećamo u_i za 1, to neće (vjerojatnosno) utjecati na preostale glavne komponente pa očekujemo porast od y za γ_i. Osim toga, budući da su glavne (pa tako i relevantne) komponente sortirane po važnosti, gornju sumu možemo "odrezati" tako da ona ide od 1 do k za neki $k < m$, čime dobivamo malo drugačiji model od klasične linearne regresije, s potencijalno boljom sposobnosti generalizacije.

Poglavlje 3

Parcijalni najmanji kvadrati

Ovo poglavlje čini srž cijelog rada i bavi se algoritmom za modeliranje poznatim pod nazivom metoda parcijalnih najmanjih kvadrata (skraćeno PLS od engl. *Partial least squares*). Slično kao i PCA, metoda se primjenjuje kada imamo puno (potencijalno koreliranih) kovarijata koje želimo svesti na manji broj relevantnih komponenti. Simulacije su pokazale da PLS dostiže najmanju kvadratnu grešku s manjim brojem faktora nego PCA [4]. Među ostalim prednostima u odnosu na PCA ističu se i jedinstven odabir glavnih komponenti te manja složenost algoritma.

PLS vuče svoje korijene iz ranih 1980-ih. Najviše su mu doprinijeli Svante Wold i Harald Martens koji su ga razvijali kao kalibracijsku metodu za predviđanje kemijskih varijabli, kod kojih je zavisna varijabla neka kemijska tvar (protein ili mast), dok kovarijate predstavljaju apsorpcije mjerene pod različitim valnim duljinama nekim spektografskim instrumentom.

Prvo ćemo predstaviti algoritam i pokušati ga intuitivno razumjeti, a potom ga nizom rezultata povezati s modelima iz prethodna dva poglavlja. Većina teorema koje ćemo dokazati, kao i glavni tok misli, preuzeti su iz [5].

3.1 Algoritam

Iako većina rezultata vrijedi i bez pretpostavke normalnosti, konkretnosti radi pretpostavit ćemo da imamo sljedeći slučaj:

$$\begin{pmatrix} \boldsymbol{x} \\ y \end{pmatrix} \sim N\left\{ \begin{pmatrix} \boldsymbol{\mu_x} \\ \mu_y \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma} & \boldsymbol{\sigma} \\ \boldsymbol{\sigma'} & \sigma_y^2 \end{pmatrix} \right\}, \tag{3.1}$$

gdje je $\boldsymbol{x} = (x_1, \ldots, x_K)$ slučajni K-dimenzionalni vektor, a y slučajna varijabla.

Populacijska verzija PLS algoritma tada glasi ovako:

1. Inicijaliziraj početne reziduale:

$$\boldsymbol{e}_0 = \boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}$$
$$f_0 = y - \mu_y.$$

Za $a = 1, 2, \ldots K$:

2. Izračunaj komponentu t_a kao linearnu kombinaciju od $\boldsymbol{e}_{a-1}$, pri čemu za koeficijente uzmi vektor korelacije s f_{a-1}:

$$\boldsymbol{w}_a = \mathrm{Cov}(\boldsymbol{e}_{a-1}, f_{a-1}) \tag{3.2}$$
$$t_a = \langle \boldsymbol{w}_a, \boldsymbol{e}_{a-1} \rangle. \tag{3.3}$$

3. Metodom najmanjih kvadrata odredi koeficijente uz komponente za $\boldsymbol{x}$ i y [1]:

$$\boldsymbol{p}_a = \mathrm{Cov}(\boldsymbol{e}_{a-1}, t_a)/\mathrm{Var}(t_a) \tag{3.4}$$
$$q_a = \mathrm{Cov}(f_{a-1}, t_a)/\mathrm{Var}(t_a). \tag{3.5}$$

4. Izračunaj nove reziduale:

$$\boldsymbol{e}_a = \boldsymbol{e}_{a-1} - \boldsymbol{p}_a t_a$$
$$\boldsymbol{f}_a = \boldsymbol{f}_{a-1} - q_a t_a.$$

Prateći korake algoritma induktivno dobivamo sljedeći rastav od $\boldsymbol{x}$:

$$\boldsymbol{x} = \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{e}_0 = \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{p}_1 t_1 + \boldsymbol{e}_1 = \cdots = \boldsymbol{\mu}_{\boldsymbol{x}} + \boldsymbol{p}_1 t_1 + \cdots + \boldsymbol{p}_a t_a + \boldsymbol{e}_a. \tag{3.6}$$

Slično dobivamo i za y:

$$y = \mu_y + q_1 t_1 + \cdots + q_a t_a + f_a. \tag{3.7}$$

U a-tom koraku, dakle, imamo rastav od $\boldsymbol{x}$ i y na komponente t_a i njihove greške $\boldsymbol{e}_a$ i f_a, redom. U koraku $a + 1$ pokušavamo obje te greške smanjiti projicirajući ih na t_{a+1}, koji nije ništa drugo doli projekcija f_a na $\boldsymbol{e}_a$, pa očekujemo da komponente t_a, osim što dobro opisuju $\boldsymbol{x}$, ujedno sadržavaju bitnu informaciju za predviđanje y; u biti paralelno rastavljamo $\boldsymbol{x}$ i y na glavne komponente - slično kao dvostruki PCA, ali uz uvjet da su dobivene komponente za obje varijable jednake i $\boldsymbol{x}$-izmjerive.

Pokažimo da su t_a doista $\boldsymbol{x}$-izmjerivi. Uočimo da je t_a po definiciji $\boldsymbol{e}_{a-1}$-izmjeriva funkcija pa je stoga i $\boldsymbol{e}_a$ $\boldsymbol{e}_{a-1}$-izmjeriv kao linearna kombinacija $\boldsymbol{e}_{a-1}$-izmjerivih varijabli. Budući

[1] Algoritam svoje ime duguje upravo ovom koraku.

da je $\boldsymbol{e}_0$ $\boldsymbol{x}$-izmjeriva, indukcijom zaključujemo da su sve t_a $\boldsymbol{x}$-izmjerive, što smo i htjeli. Nadalje, svaki $\boldsymbol{e}_a$ je kao greška projekcije najmanjih kvadrata nužno nekoreliran s $\boldsymbol{t}_a$ pa su stoga $t_1, t_2, t_3, \ldots$ svi međusobno nekorelirani, što je također poželjno svojstvo naših komponenti.

Posljedično tome, imamo da je

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{e}_{a-1}, f_{a-1}) &\overset{3.6}{=} \mathrm{Cov}(\boldsymbol{e}_{a-1}, y - \mu_y - q_1 t_1 - \cdots - q_{a-1} t_{a-1}) \\ &= \mathrm{Cov}(\boldsymbol{e}_{a-1}, y) - \sum_{k=1}^{a-1} q_a \underbrace{\mathrm{Cov}(\boldsymbol{e}_{a-1}, t_k)}_{0} \\ &= \mathrm{Cov}(\boldsymbol{e}_{a-1}, y).\end{aligned}$$

Koristeći rastave 3.6 i 3.7 analogno dobivamo da je

$$\begin{aligned}\mathrm{Cov}(\boldsymbol{e}_{a-1}, t_a) &= \mathrm{Cov}(\boldsymbol{x}, t_a) \\ \mathrm{Cov}(\boldsymbol{f}_{a-1}, t_a) &= \mathrm{Cov}(\boldsymbol{y}, t_a).\end{aligned}$$

Stoga u 3.2 umjesto f_{a-1} možemo staviti y, u 3.4 možemo zamijeniti $\boldsymbol{e}_{a-1}$ s $\boldsymbol{x}$, a u 3.5 f_{a-1} s y, čime dobivamo ekvivalentan algoritam u kojem je:

$$\begin{aligned}\boldsymbol{w}_a &= \mathrm{Cov}(\boldsymbol{e}_{a-1}, y) \\ \boldsymbol{p}_a &= \mathrm{Cov}(\boldsymbol{x}, t_a)/\mathrm{Var}(t_a) \\ q_a &= \mathrm{Cov}(y, t_a)/\mathrm{Var}(t_a).\end{aligned}$$

Takvi zapisi daju malo drugačiju interpretaciju. U koraku s "parcijalnim najmanjim kvadratima" primjerice umjesto procjene preostale greške $\boldsymbol{e}_{a-1}$ pomoću novonastale varijable t_a procjenjujemo čitavu varijablu $\boldsymbol{x}$. Međutim, kako je t_a okomit na $t_1, \ldots, t_{a-1}$ (isto kao i $\boldsymbol{e}_{a-1}$), neminovno je da njegovo dodavanje u sumu popravi procjenu upravo u smjeru greške.

Osvrnimo se odmah i na uzoračku verziju algoritma. Pretpostavimo da imamo realizaciju slučajnog uzorka varijabli iz 3.1 duljine N koji je dan s

$$\mathbf{X} = \begin{pmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_K\end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1K} \\ x_{21} & x_{22} & \cdots & x_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NK}\end{pmatrix} \in \mathbb{R}^{N\times K}, \quad \boldsymbol{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N\end{pmatrix} \in \mathbb{R}^N. \tag{3.8}$$

U principu se uzorački algoritam ne razlikuje od populacijskog ni u čemu drugom osim što se umjesto očekivanja i varijance koriste njihovi procjenitelji - prosjek i uzoračka varijanca. Ipak, potpunosti radi (i zato što će nam biti koristan za usporedbu s alternativnim algoritmom u idućem odjeljku), precizno ćemo navesti i njegove korake.

Prvo ćemo oduzeti prosjeke $\overline{\boldsymbol{x}}_1, \dots \overline{\boldsymbol{x}}_K, \overline{\boldsymbol{y}}$ od varijabli $\boldsymbol{x}_1, \dots \boldsymbol{x}_K, \boldsymbol{y}$ redom kako bismo inicijalizirali matricu $\mathbf{E}_0$ i vektor $\boldsymbol{f}_0$.

$$\mathbf{E}_0 = \mathbf{X} - \underbrace{\begin{pmatrix} \overline{\boldsymbol{x}}_1 & \overline{\boldsymbol{x}}_2 & \dots & \overline{\boldsymbol{x}}_K \end{pmatrix}}_{:=\overline{\mathbf{X}}} = \mathbf{X} - \overline{\mathbf{X}} \in \mathbb{R}^{N\times K}, \tag{3.9}$$

$$\boldsymbol{f}_0 = \boldsymbol{y} - \overline{\boldsymbol{y}} \in \mathbb{R}^N. \tag{3.10}$$

Zatim za $a = 1, 2, \dots$ ponavljamo:

1. Izračunamo težine i komponente zajedničke $\boldsymbol{x}$ i y:

$$\begin{aligned} \boldsymbol{w}_a &= \mathbf{E}'_{a-1}\boldsymbol{f}_{a-1} \\ \boldsymbol{t}_a &= \mathbf{E}_{a-1}\boldsymbol{w}_a. \end{aligned}$$

2. Korištenjem najmanjih kvadrata računamo $\boldsymbol{p}_a$ i q_a:

$$\begin{aligned} \boldsymbol{p}_a &= \frac{\mathbf{E}'_{a-1}\boldsymbol{t}_a}{\boldsymbol{t}'_a\boldsymbol{t}_a} \quad \left(= \frac{\mathbf{X}'\boldsymbol{t}_a}{\boldsymbol{t}'_a\boldsymbol{t}_a}\right) \\ q_a &= \frac{\boldsymbol{f}'_{a-1}\boldsymbol{t}_a}{\boldsymbol{t}'_a\boldsymbol{t}_a} \quad \left(= \frac{\boldsymbol{y}'\boldsymbol{t}_a}{\boldsymbol{t}'_a\boldsymbol{t}_a}\right). \end{aligned}$$

3. Na kraju svakog koraka računamo nove greške:

$$\begin{aligned} \mathbf{E}_a &= \mathbf{E}_{a-1} - \boldsymbol{t}_a\boldsymbol{p}'_a \\ \boldsymbol{f}_a &= \boldsymbol{f}_{a-1} - \boldsymbol{t}_a q'_a. \end{aligned}$$

Uočimo, u koraku A uzoračkog algoritma dobivamo rastave

$$\begin{aligned} \mathbf{X} &= \overline{\mathbf{X}} + \boldsymbol{t}_1\boldsymbol{p}'_1 + \dots + \boldsymbol{t}_A\boldsymbol{p}'_A + \mathbf{E}_A \\ \boldsymbol{y} &= \overline{\boldsymbol{y}} + \boldsymbol{t}_1 q_1 + \dots + \boldsymbol{t}_A q_A + \boldsymbol{f}_A, \end{aligned}$$

usporedive s 3.6 i 3.7 redom.

Nadalje, koristeći induktivno ortogonalnost greške nastale primjenom najmanjih kvadrata - slično kao i u populacijskom slučaju - možemo vidjeti da su $\boldsymbol{t}_1, \dots, \boldsymbol{t}_a$ međusobno okomiti za svaki a te je

$$\begin{aligned} \mathcal{P}_{ta}(\boldsymbol{x}_k - \overline{\boldsymbol{x}}_k) &= p_{1k}\boldsymbol{t}_1 + \dots + p_{ak}\boldsymbol{t}_a, \quad k = 1, \dots, K, \\ \mathcal{P}_{ta}(\boldsymbol{y} - \overline{\boldsymbol{y}}) &= q_1\boldsymbol{t}_1 + \dots + q_a\boldsymbol{t}_a, \end{aligned} \tag{3.11}$$

pri čemu smo s $\mathcal{P}_{ta}$ označili projektor na $[\{\boldsymbol{t}_1, \dots, \boldsymbol{t}_a\}]$, dok koeficijent p_{ik} označava k-tu koordinatu vektora $\boldsymbol{p}_i$ za $i = 1, \dots a$.

Pretpostavimo sada da smo za dani uzorak 3.8 izvršili A koraka algoritma i time "istrenirali" naš predikcijski model. Ako je $\boldsymbol{x}_0 = (x_{01}, \ldots, x_{0K})'$ novo opažanje, definiramo $\boldsymbol{e}_0 = \boldsymbol{x}_0 - \overline{\boldsymbol{x}}_0$, a zatim za $a = 1, \ldots, A$ računamo vrijednosti komponenti i grešaka redom ovako:

$$\begin{aligned} t_{a0} &= \boldsymbol{e}'_{a-1}\boldsymbol{w}_a \\ \boldsymbol{e}_a &= \boldsymbol{e}_{a-1} - \boldsymbol{t}_{a0}\boldsymbol{p}_a. \end{aligned} \tag{3.12}$$

Tada je predviđanje u koraku A dano s

$$\hat{y}_{A0} = \overline{\boldsymbol{y}} + \sum_{a=1}^{A} t_{a0} q_a = \overline{\boldsymbol{y}} + \sum_{a=1}^{A} t_{a0} \frac{\boldsymbol{t}'_a \boldsymbol{y}}{\boldsymbol{t}'_a \boldsymbol{t}_a}.$$

3.2 Alternativni algoritam

Drugi algoritam za PLS koji ćemo obraditi uveo je Harald Martens u [11]. Glavna razlika u odnosu na algoritam izložen u prošlom odjeljku leži u tome što se koristi multivarijatna regresija kako bi se pronašli koeficijenti q. Drugim riječima, u koraku a umjesto predviđanja $\boldsymbol{f}_{a-1}$ preko $\boldsymbol{t}_a$ predviđamo čitav $\boldsymbol{y}$ korištenjem svih dotad izračunatih komponenti $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_a$. Time umjesto novog koeficijenta q_a (kojeg smo ranije dodali prethodno izračunatim $q_1, \ldots, q_{a-1}$) dobivamo čitav niz koeficijenata $q_{a1}, \ldots, q_{aa}$, koji "žive" lokalno samo u tom koraku.

Pokazat ćemo da su, usprkos tim razlikama, algoritmi ekvivalentni, u smislu da daju ista predviđanja, što je glavni rezultat ovog odjeljka. Zatim ćemo koristeći tu činjenicu izvesti rekurzivnu formulu za težine $\boldsymbol{w}_a$ i relativno jednostavnu formulu za predviđanje. Ti su rezultati preuzeti iz [4].

Novi algoritam uvest ćemo u njegovoj uzoračkoj verziji (prelazak na populacijski slučaj isti je kao kod primarnog algoritma). Pretpostavimo da imamo realizaciju uzorka danu s 3.8. Greške se inicijaliziraju kao i ranije [2]:

$$\begin{aligned} \mathbf{E}_0^* &= \mathbf{X} - \overline{\mathbf{X}} \\ \boldsymbol{f}_0^* &= \boldsymbol{y} - \overline{\boldsymbol{y}}. \end{aligned}$$

[2] Varijable koje se odnose na alternativni algoritam označavat ćemo zvjezdicom kako bismo ih razlikovali od onih iz primarnog algoritma.

Zatim u koracima $a = 1, 2, \ldots$ računamo sljedeće koeficijente:

$$\begin{aligned}
\boldsymbol{p}_a^* &= \mathbf{E}_{a-1}^{*\prime} \boldsymbol{f}_{a-1}^* \\
\boldsymbol{t}_a^* &= \frac{\mathbf{E}_{a-1}^* \boldsymbol{p}_a^*}{\boldsymbol{p}_a^{*\prime} \boldsymbol{p}_a^*} \quad \left(= \frac{\mathbf{X}^* \boldsymbol{p}_a^*}{\boldsymbol{p}_a^{*\prime} \boldsymbol{p}_a} \right) \qquad (3.13) \\
\mathbf{T}_a^* &= \begin{pmatrix} \boldsymbol{t}_1^* & \ldots & \boldsymbol{t}_a^* \end{pmatrix} \\
\boldsymbol{q}_a^* &= (\mathbf{T}_a^{*\prime} \mathbf{T}_a^*)^{-1} \mathbf{T}_a^{*\prime} \boldsymbol{y} \qquad (3.14) \\
\mathbf{E}_a^* &= \mathbf{E}_{a-1}^* - \boldsymbol{t}_a^* \boldsymbol{p}_a^{*\prime} \\
\boldsymbol{f}_a^* &= \boldsymbol{y} - \sum_{k=1}^{a} \boldsymbol{t}_k^* q_{ak}^*. \qquad (3.15)
\end{aligned}$$

Glavna razlika među algoritmima je, dakle, u 3.14. Međutim, postoji još jedna suptilnija razlika: u alternativnom algoritmu prvo (nekom formulom) definiramo vektor $\boldsymbol{p}_a^*$, a zatim $\boldsymbol{t}_a^*$ računamo metodom najmanjih kvadrata kako bismo preko $\boldsymbol{p}_a^*$ predvidjeli $\mathbf{E}_{a-1}^*$, dok je u primarnom algoritmu obrnuto. Stoga su u drugom algoritmu vektori $\boldsymbol{p}_1^*, \ldots, \boldsymbol{p}_a^*$ ortogonalni, baš kao i vektori $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_a$ u prvom.

Ako je $\boldsymbol{x}_0 = (x_{01}, \ldots, x_{0K})'$ novo opažanje, stavljamo $\boldsymbol{e}_0^* = \boldsymbol{x}_0 - \overline{\boldsymbol{x}}_0$, a zatim, kao u 3.12, za $a = 1, \ldots, A$ računamo

$$\begin{aligned}
t_{a0}^* &= \frac{\boldsymbol{e}_{a-1}^{*\prime} \boldsymbol{p}_a^*}{\boldsymbol{p}_a^{*\prime} \boldsymbol{p}_a} \\
\boldsymbol{e}_a^* &= \boldsymbol{e}_{a-1}^* - t_{a0}^* \boldsymbol{p}_a^* = \boldsymbol{e}_{a-1}^* - \sum_{k=1}^{a} t_{k0}^* \boldsymbol{p}_k^*. \qquad (3.16)
\end{aligned}$$

Predviđanje od y_0 tad glasi

$$\hat{y}_{A0}^* = \overline{y} + \sum_{a=1}^{A} t_{a0}^* q_{Aa}^*. \qquad (3.17)$$

Kasnije ćemo izvesti puno jednostavniju formulu za predviđanje. Ali prvo idemo dokazati najavljenu ekvivalenciju algoritama.

Teorem 3.2.1. *Uz ranije uvedene oznake u oba algoritma, za $a = 1, 2, \ldots$ vrijedi*

(a) $\boldsymbol{p}_a^* = \boldsymbol{w}_a$,

(b) $[\{\boldsymbol{t}_1^*, \ldots, \boldsymbol{t}_a^*\}] = [\{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_a\}]$,

(c) $\boldsymbol{f}_a^* = \boldsymbol{f}_a$,

(d) $\hat{\boldsymbol{y}}^*_{a0} = \hat{\boldsymbol{y}}_{a0}$.

Dokaz. Neka su bez smanjenja općenitosti $\overline{\mathbf{X}} = \mathbf{0}$ i $\overline{\boldsymbol{y}} = \mathbf{0}$. Nadalje, neka su $\boldsymbol{\mathcal{P}}^*_{ta}$ i $\boldsymbol{\mathcal{P}}_{ta}$ projektori na prostore $[\{\boldsymbol{t}^*_1, \ldots, \boldsymbol{t}^*_a\}]$ i $[\{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_a\}]$ redom te neka je $\boldsymbol{I}_N$ jedinična matrica u $\mathbb{R}^{N\times N}$. Tada iz 3.11 imamo

$$\mathbf{E}_a = (\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta})\mathbf{X} \tag{3.18}$$

$$\boldsymbol{f}_a = (\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta})\boldsymbol{y}, \tag{3.19}$$

dok 3.14 - 3.15 daju

$$\mathbf{E}^*_a = \mathbf{X} - \sum_{k=1}^{a} \boldsymbol{t}^*_k \boldsymbol{p}^{*\prime}_k \tag{3.20}$$

$$\boldsymbol{f}^*_a = (\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}^*_{ta})\boldsymbol{y}. \tag{3.21}$$

Pokazat ćemo (a) i (b) zajedno korištenjem matematičke indukcije. Za $a = 1$ imamo

$$\mathbf{E}_0 = \mathbf{X} = \mathbf{E}^*_0$$
$$\boldsymbol{f}_0 = \boldsymbol{y} = \boldsymbol{f}^*_0,$$

pa je

$$\boldsymbol{p}^*_1 = \mathbf{E}^{*\prime}_0 \boldsymbol{f}_0 = \mathbf{E}'_0 \boldsymbol{f}_0 = \boldsymbol{w}_0,$$

što daje bazu za (a), dok bazu za (b) dokazuje relacija

$$t^*_1 = \frac{\mathbf{E}^*_0 \boldsymbol{p}^*_1}{\boldsymbol{p}^{*\prime}_1 \boldsymbol{p}_1} = \frac{\mathbf{E}^*_0 \boldsymbol{w}^*_1}{\boldsymbol{w}^{*\prime}_1 \boldsymbol{w}_1} \propto t_1.$$

Pretpostavimo sada da za neki $a \in \mathbb{N}$ vrijede (a) i (b). Tada je $\boldsymbol{\mathcal{P}}_{\boldsymbol{ta}} = \boldsymbol{\mathcal{P}}^*_{\boldsymbol{ta}}$ pa vrijedi $\boldsymbol{f}_a = \boldsymbol{f}^*_a$, što dokazuje (c). Nadalje, budući da je $(\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta})$ projektor na $[\{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_a\}^\perp$, vrijedi

$$\boldsymbol{f}_a \perp [\{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_a\}], \tag{3.22}$$

$$\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta} = (\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta})^2 \tag{3.23}$$

$$= (\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta})'. \tag{3.24}$$

Otud slijedi

$$\begin{aligned}
\boldsymbol{p}^*_{a+1} = \mathbf{E}^{*\prime}_a \boldsymbol{f}^*_a &= \left(\mathbf{X} - \sum_{k=1}^{a} \boldsymbol{t}^*_k \boldsymbol{p}^{*\prime}_k\right)' \boldsymbol{f}_a = \left(\mathbf{X}' - \sum_{k=1}^{a} \boldsymbol{p}^*_k \boldsymbol{t}^{*\prime}_k\right) \boldsymbol{f}^*_a \\
&= \mathbf{X}'\boldsymbol{f}_a - \sum_{k=1}^{a} \underbrace{\langle \boldsymbol{t}^*_k, \boldsymbol{f}_a \rangle}_{0} \boldsymbol{p}^*_k \overset{3.22}{=} \mathbf{X}'\boldsymbol{f}_a \overset{3.19}{=} \mathbf{X}'(\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta})\boldsymbol{y} \\
&\overset{3.23}{=} \mathbf{X}'(\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta})^2 \boldsymbol{y} = \mathbf{X}'(\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta})(\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta})\boldsymbol{y} \\
&\overset{3.24}{=} [(\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta})\mathbf{X}]'(\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta})\boldsymbol{y} = \mathbf{E}'_a \boldsymbol{f}_a = \boldsymbol{w}_{a+1},
\end{aligned} \tag{3.25}$$

što dokazuje (a). Uvrštavanjem 3.18 i 3.20 u definicije komponenti u primarnom i alternativnom algoritmu redom dobivamo

$$\boldsymbol{t}_{a+1} = \mathbf{E}_a \boldsymbol{w}_{a+1} = (\boldsymbol{I}_N - \boldsymbol{\mathcal{P}}_{ta})\mathbf{X}\boldsymbol{w}_{a+1} = \mathbf{X}\boldsymbol{w}_{a+1} - \boldsymbol{\mathcal{P}}_{ta}\mathbf{X}\boldsymbol{w}_{a+1} \tag{3.26}$$

$$\begin{aligned}\boldsymbol{t}^*_{a+1} \propto \mathbf{E}^*_a \boldsymbol{p}^*_{a+1} &= \left(\mathbf{X} - \sum_{k=1}^{a} \boldsymbol{t}^*_k \boldsymbol{p}^{*\prime}_k\right)\boldsymbol{p}^*_{a+1} = \mathbf{X}\boldsymbol{p}^*_{a+1} - \sum_{k=1}^{a} \underbrace{\langle \boldsymbol{p}^*_k, \boldsymbol{p}^*_{a+1}\rangle}_{0}\, \boldsymbol{t}^*_k \\ &= \mathbf{X}\boldsymbol{p}^*_{a+1} = \mathbf{X}\boldsymbol{w}_{a+1}.\end{aligned} \tag{3.27}$$

Budući da je vektor $\boldsymbol{\mathcal{P}}_{ta}\mathbf{X}\boldsymbol{w}_{a+1}$ linearna kombinacija projekcija komponenti od $\mathbf{X}$ na $[\{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_a\}]$, po pretpostavci se nalazi u $[\{\boldsymbol{t}^*_1, \ldots, \boldsymbol{t}^*_a\}]$. Uvrštavanjem 3.27 u 3.26 sad vidimo da je $\boldsymbol{t}_{a+1} \in [\{\boldsymbol{t}^*_1, \ldots, \boldsymbol{t}^*_{a+1}\}]$, što dokazuje (b).

Zbog jednakosti tih ljuski za svaki a postoji regularna matrica $\mathbf{D}$ takva da je $\mathbf{T}^*_a = \mathbf{T}_a\mathbf{D}$, pri čemu je $\mathbf{T}_a = \begin{pmatrix}\boldsymbol{t}_1 & \cdots & \boldsymbol{t}_a\end{pmatrix}$. Iz načina na koji stvaramo predikcijske komponente $\boldsymbol{t}_k$ i $\boldsymbol{t}^*_k$ u 3.12 i 3.16 redom, jasno je da vrijedi

$$(t^*_{10}, \ldots, t^*_{a0}) = (t_{10}, \ldots, t_{a0})\mathbf{D}.$$

Budući da je $\mathbf{D}$ regularna, po Propoziciji 1.3.4 je $\hat{\boldsymbol{y}}^*_{a0} = \hat{\boldsymbol{y}}_{a0}$ pa slijedi (d). □

Korolar 3.2.2. *Za svaki $a \leq K$ su težine $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_a$ međusobno ortogonalne.*

Dokaz. Tvrdnja slijedi iz Teorema 3.2.1 (a) i činjenice da su $\boldsymbol{p}^*_1, \ldots, \boldsymbol{p}^*_a$ ortogonalni. □

Teorem 3.2.3. *Rekurzija za težine u primarnom algoritmu je dana s*

$$\begin{aligned}\boldsymbol{w}_1 &= \mathbf{E}'_0 \boldsymbol{f}_0 \\ \boldsymbol{w}_{a+1} &= \boldsymbol{s} - \mathbf{W}_a(\mathbf{W}'_a \boldsymbol{S} \mathbf{W}_a)^{-1}\mathbf{W}'_a \boldsymbol{s},\end{aligned} \tag{3.28}$$

pri čemu su

$$\boldsymbol{s} = \mathbf{X}'\boldsymbol{y}, \quad \boldsymbol{S} = \mathbf{X}'\mathbf{X}, \quad \mathbf{W}_a = \begin{bmatrix}\boldsymbol{w}_1 & \cdots \boldsymbol{w}_a\end{bmatrix}.$$

Dokaz. Uvrstimo li Teorem 3.2.1 (a) u 3.13, dobivamo

$$\boldsymbol{t}^*_a = \frac{\mathbf{E}^*_{a-1}\boldsymbol{w}_a}{\boldsymbol{w}_a{}'\boldsymbol{w}_a},$$

što možemo matrično zapisati kao

$$\mathbf{T}^*_a = \mathbf{X}\mathbf{W}_a\mathbf{C}_a, \tag{3.29}$$

gdje su

$$\mathbf{W}_a = \begin{bmatrix} \boldsymbol{w}_1 & \cdots \boldsymbol{w}_a \end{bmatrix} \in \mathbb{R}^{K\times a},$$
$$\mathbf{C}_a = \begin{pmatrix} \frac{1}{\|\boldsymbol{w}_1\|_2} & 0 & \dots & 0 \\ 0 & \frac{1}{\|\boldsymbol{w}_2\|_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\|\boldsymbol{w}_a\|_2} \end{pmatrix} \in \mathbb{R}^{a\times a}.$$

Uvrštavanje 3.29 u 3.14 daje

$$\begin{aligned} \boldsymbol{q}_a^* &= (\mathbf{T}_a^{*\prime}\mathbf{T}_a^*)^{-1}\,\mathbf{T}_a^{*\prime}\boldsymbol{y} = ((\mathbf{X}\mathbf{W}_a\mathbf{C}_a)'\mathbf{X}\mathbf{W}_a\mathbf{C}_a)^{-1}(\mathbf{X}\mathbf{W}_a\mathbf{C}_a)'\boldsymbol{y} \\ &= (\mathbf{C}_a\mathbf{W}_a'\mathbf{X}'\mathbf{X}\mathbf{W}_a\mathbf{C}_a)^{-1}\mathbf{C}_a\mathbf{W}_a'\mathbf{X}'\boldsymbol{y} \\ &= \mathbf{C}_a^{-1}(\mathbf{W}_a'\mathbf{X}'\mathbf{X}\mathbf{W}_a)^{-1}\mathbf{C}_a^{-1}\mathbf{C}_a\mathbf{W}_a'\mathbf{X}'\boldsymbol{y} \\ &= \mathbf{C}_a^{-1}(\mathbf{W}_a'\mathbf{S}\mathbf{W}_a)^{-1}\mathbf{W}_a'\boldsymbol{s}, \end{aligned} \tag{3.30}$$

gdje su

$$\boldsymbol{s} = \mathbf{X}'\boldsymbol{y}, \quad \boldsymbol{S} = \mathbf{X}'\mathbf{X}.$$

Iz 3.25 imamo da je

$$\boldsymbol{w}_{a+1} = \mathbf{X}'(\mathbf{I}_N - \boldsymbol{\mathcal{P}}_{ta}^*)\boldsymbol{y}. \tag{3.31}$$

Kad u matrični zapis projektora $\boldsymbol{\mathcal{P}}_{ta}^*$ uvrstimo 3.29, dobijemo

$$\begin{aligned} \boldsymbol{\mathcal{P}}_{ta}^* &= \mathbf{T}_a^*(\mathbf{T}_a^{*\prime}\mathbf{T}_a^*)^{-1}\mathbf{T}_a^{*\prime} = \mathbf{X}\mathbf{W}_a\mathbf{C}_a((\mathbf{X}\mathbf{W}_a\mathbf{C}_a)'\mathbf{X}\mathbf{W}_a\mathbf{C}_a)^{-1}(\mathbf{X}\mathbf{W}_a\mathbf{C}_a)' \\ &= \mathbf{X}\mathbf{W}_a\mathbf{C}_a\mathbf{C}_a^{-1}(\mathbf{W}_a'\mathbf{X}'\mathbf{X}\mathbf{W}_a)^{-1}\mathbf{C}_a^{-1}\mathbf{C}_a\mathbf{W}_a'\mathbf{X}' \\ &= \mathbf{X}\mathbf{W}_a(\mathbf{W}_a'\mathbf{S}\mathbf{W}_a)^{-1}\mathbf{W}_a'\mathbf{X}'. \end{aligned} \tag{3.32}$$

Konačno, uvrštavanjem 3.32 u 3.31 dobivamo

$$\begin{aligned} \boldsymbol{w}_{a+1} &= \mathbf{X}'(\mathbf{I}_N - \mathbf{X}\mathbf{W}_a(\mathbf{W}_a'\mathbf{S}\mathbf{W}_a)^{-1}\mathbf{W}_a'\mathbf{X}')\boldsymbol{y} \\ &= \boldsymbol{s} - \mathbf{S}\mathbf{W}_a(\mathbf{W}_a'\mathbf{S}\mathbf{W}_a)^{-1}\mathbf{W}_a'\boldsymbol{s}. \end{aligned}$$

□

Napomena 3.2.4. *U populacijskom slučaju vrijedi sljedeća rekurzija za težine:*

$$\begin{aligned} \boldsymbol{w}_1 &= \boldsymbol{\sigma} \\ \boldsymbol{w}_{a+1} &= \boldsymbol{\sigma} - \boldsymbol{\Sigma}\mathbf{W}_a(\mathbf{W}_a'\boldsymbol{\Sigma}\mathbf{W}_a)^{-1}\mathbf{W}_a'\boldsymbol{\sigma}. \end{aligned} \tag{3.33}$$

Na kraju ovog odjeljka dat ćemo pojednostavljenu formulu za računanje predviđanja.

Teorem 3.2.5. *Za oba PLS algoritma, predviđanje u koraku A glasi*

$$\hat{\boldsymbol{y}}_{A0} = \overline{\boldsymbol{y}} + \boldsymbol{\beta}'_{A,PLS}(\boldsymbol{x}_0 - \overline{\boldsymbol{x}}),$$

gdje je

$$\boldsymbol{\beta}_{A,PLS} = \mathbf{W}_a(\mathbf{W}'_a\mathbf{S}\mathbf{W}_a)^{-1}\mathbf{W}'_a\boldsymbol{s}. \tag{3.34}$$

Dokaz. Iz 3.17 i Teorema 3.2.1 (d) imamo da je

$$\hat{\boldsymbol{y}}_{A0} = \hat{\boldsymbol{y}}^*_{A0} = \overline{\boldsymbol{y}} + (t^*_{10}, \ldots, t^*_{A0})\boldsymbol{q}^*_A.$$

Slično kao što smo u dokazu Teorema 3.2.3 dobili $\mathbf{T}^*_a = \mathbf{X}\mathbf{W}_a\mathbf{C}_a$, iz 3.17 lako dobijemo

$$(t^*_{10}, \ldots, t^*_{A0}) = (\boldsymbol{x}_0 - \overline{\boldsymbol{x}})'\mathbf{W}_A\mathbf{C}_A,$$

pa imamo

$$\begin{aligned}
\hat{\boldsymbol{y}}_{A0} &= \overline{\boldsymbol{y}} + (\boldsymbol{x}_0 - \overline{\boldsymbol{x}})'\mathbf{W}_A\mathbf{C}_A\boldsymbol{q}^*_A \\
&\overset{3.30}{=} \overline{\boldsymbol{y}} + (\boldsymbol{x}_0 - \overline{\boldsymbol{x}})'\mathbf{W}_A\mathbf{C}_A\mathbf{C}_a^{-1}(\mathbf{W}'_a\mathbf{S}\mathbf{W}_a)^{-1}\mathbf{W}'_a\boldsymbol{s} \\
&= \overline{\boldsymbol{y}} + (\boldsymbol{x}_0 - \overline{\boldsymbol{x}})'\mathbf{W}_A(\mathbf{W}'_a\mathbf{S}\mathbf{W}_a)^{-1}\mathbf{W}'_a\boldsymbol{s}.
\end{aligned}$$

□

Korolar 3.2.6. *Za $A = K$ vrijedi*

$$\boldsymbol{\beta}_{A,PLS} = \boldsymbol{\beta}_{LS},$$

pri čemu je

$$\boldsymbol{\beta}_{LS} = \mathbf{S}^{-1}\boldsymbol{s}$$

vektor iz najboljeg linearnog procjenitelja.

Dokaz. Kad je $A = K$, matrica S je punog ranga pa iz 3.34 imamo

$$\boldsymbol{\beta}_{A,PLS} = \mathbf{W}_a(\mathbf{W}'_a\mathbf{S}\mathbf{W}_a)^{-1}\mathbf{W}'_a\boldsymbol{s} = \mathbf{W}_a\mathbf{W}_a^{-1}\mathbf{S}^{-1}\mathbf{W}'^{-1}_a\mathbf{W}'_a\boldsymbol{s} = \boldsymbol{S}^{-1}\boldsymbol{s} = \boldsymbol{\beta}_{LS}.$$

□

Napomena 3.2.7. *U populacijskom slučaju za vektor procjenitelja vrijedi sljedeća formula:*

$$\boldsymbol{\beta}_{A,PLS} = \mathbf{W}_A(\mathbf{W}'_A\boldsymbol{\Sigma}\mathbf{W}_A)^{-1}\mathbf{W}'_A\boldsymbol{\sigma}. \tag{3.35}$$

3.3 Veza između PLS i PCA

U Korolaru 3.2.2 pokazali smo da su težine $\boldsymbol{w}_a$ ortogonalni K-dimenzionalni vektori. Budući da su netrivijalni, to povlači da su nezavisni pa regresijski vektor $\boldsymbol{\beta}_{i,PLS}$ iz 3.35 ovisi isključivo o njihovoj linearnoj ljusci. Prema Korolaru 3.2 je $\boldsymbol{\beta}_{K,PLS} = \boldsymbol{\beta}_{LS}$. Glavno je pitanje ovog odjeljka postoji li možda neki $A < K$ takav da je $\boldsymbol{\beta}_{A,PLS} = \boldsymbol{\beta}_{LS}$ i, ako postoji, o čemu on ovisi. Pokazat će se da je A jednak broju relevantnih svojstvenih vektora koje smo definirali na kraju prošlog poglavlja, čime dobivamo potencijalni zaustavni kriterij našeg algoritma.

U svrhu dokazivanja te tvrdnje za početak uvodimo sljedeće oznake:

$$S_A := [\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_A\}],$$
$$M := \max\{n \in \mathbb{N} : \boldsymbol{w}_n \neq 0\}.$$

Uočimo, zbog nezavisnosti od $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_A\}$ nužno vrijedi $M < K$. Nadalje, neka su $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_K$ relevantni svojstveni vektori od $\boldsymbol{x}$ za predvidanje y iz Definicije 2.14, te neka su $\nu_1, \ldots, \nu_K$ pripadne svojstvene vrijednosti. Slijedi niz teorema koji će nas dovesti do najavljene tvrdnje.

Teorem 3.3.1. *(a)* $S_A = [\{\boldsymbol{\sigma}, \boldsymbol{\Sigma}\boldsymbol{\sigma}, \boldsymbol{\Sigma}^2\boldsymbol{\sigma}, \ldots, \boldsymbol{\Sigma}^{A-1}\boldsymbol{\sigma}\}]$.

(b) $M = \min\{A \in \mathbb{N} : \boldsymbol{\Sigma}^A\boldsymbol{\sigma} \in S_A\}$.

(c) $M = \min\{A \in \mathbb{N} : \boldsymbol{\beta}_{LS} \in S_A\}$.

Dokaz. (a) Ovo ćemo dokazati indukcijom. Činjenica da je

$$\boldsymbol{w}_1 = \mathrm{Cov}(\boldsymbol{e}_0, f_0) = \mathrm{Cov}(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}, y - \mu_y) = \boldsymbol{\sigma}$$

dokazuje bazu. Pretpostavimo sad da skupovi $\{\boldsymbol{\sigma}, \boldsymbol{\Sigma}\boldsymbol{\sigma}, \boldsymbol{\Sigma}^2\boldsymbol{\sigma}, \ldots, \boldsymbol{\Sigma}^{A-1}\boldsymbol{\sigma}\}$ i $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_A\}$ razapinju isti potprostor. Rekurziju za težine 3.33 možemo promotriti na sljedeći način:

$$\boldsymbol{w}_{A+1} = \boldsymbol{\sigma} - \boldsymbol{\Sigma}\boldsymbol{W}_A \underbrace{(\mathbf{W}_A'\boldsymbol{\Sigma}\mathbf{W}_A)^{-1}\mathbf{W}_A'\boldsymbol{\sigma}}_{\boldsymbol{\alpha}} = \boldsymbol{\sigma} - \boldsymbol{\Sigma}\boldsymbol{W}_A\boldsymbol{\alpha}, \tag{3.36}$$

pri čemu je $\boldsymbol{\alpha} \in \mathbb{R}^K$ neki niz koeficijenata. Budući da su po pretpostavci

$$\boldsymbol{w}_1, \ldots, \boldsymbol{w}_A \in [\{\boldsymbol{\sigma}, \boldsymbol{\Sigma}\boldsymbol{\sigma}, \boldsymbol{\Sigma}^2\boldsymbol{\sigma}, \ldots, \boldsymbol{\Sigma}^{A-1}\boldsymbol{\sigma}\}],$$

stupci matrice $\boldsymbol{\Sigma}\boldsymbol{W}_A$ su nužno u $[\{\boldsymbol{\Sigma}\boldsymbol{\sigma}, \boldsymbol{\Sigma}^2\boldsymbol{\sigma}, \ldots, \boldsymbol{\Sigma}^A\boldsymbol{\sigma}\}]$ pa je i

$$\boldsymbol{\Sigma}\boldsymbol{W}_A\boldsymbol{\alpha} \in [\{\boldsymbol{\Sigma}\boldsymbol{\sigma}, \boldsymbol{\Sigma}^2\boldsymbol{\sigma}, \ldots, \boldsymbol{\Sigma}^A\boldsymbol{\sigma}\}]. \tag{3.37}$$

Otud slijedi da je

$$\boldsymbol{w}_{A+1} = \boldsymbol{\sigma} - \boldsymbol{\Sigma}\boldsymbol{W}_A\boldsymbol{\alpha} \in [\{\boldsymbol{\sigma}, \boldsymbol{\Sigma}\boldsymbol{\sigma}, \boldsymbol{\Sigma}^2\boldsymbol{\sigma}, \ldots, \boldsymbol{\Sigma}^A\boldsymbol{\sigma}\}],$$

iz čega dobivamo

$$[\{\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{A+1}\}] \subseteq [\{\boldsymbol{\sigma}, \boldsymbol{\Sigma}\boldsymbol{\sigma}, \boldsymbol{\Sigma}^2\boldsymbol{\sigma}, \ldots, \boldsymbol{\Sigma}^A\boldsymbol{\sigma}\}].$$

Ukoliko $\boldsymbol{w}_{A+1} \neq \mathbf{0}$, imamo jednakost jer su oba potprostora tada nužno istih dimenzija (obojici prostora iz pretpostavke se dimenzija povećala za 1 dodavanjem novog člana).

S druge strane, ako je $\boldsymbol{w}_{A+1} = \mathbf{0}$, 3.36 postaje

$$\boldsymbol{\sigma} = \boldsymbol{\Sigma}\boldsymbol{W}_A\boldsymbol{\alpha},$$

pa je prema 3.37

$$\boldsymbol{\sigma} \in [\{\boldsymbol{\Sigma}\boldsymbol{\sigma}, \boldsymbol{\Sigma}^2\boldsymbol{\sigma}, \ldots, \boldsymbol{\Sigma}^A\boldsymbol{\sigma}\}],$$

što znači da je dodavanjem novog člana u skup $\{\boldsymbol{\sigma}, \boldsymbol{\Sigma}\boldsymbol{\sigma}, \boldsymbol{\Sigma}^2\boldsymbol{\sigma}, \ldots, \boldsymbol{\Sigma}^{A-1}\boldsymbol{\sigma}\}$ uvedena zavisnost te se dimenzija njegove ljuske, kao ni one skupa težina, nije povećala pa su po pretpostavci te ljuske ostale međusobno jednake.

(b) Iz (a) imamo da je M maksimalna dimenzija prostora koji nastaju takvim iteracijama pa slijedi tvrdnja.

(c) Pokazat ćemo ekvivalenciju:

$$\boldsymbol{\beta}_{LS} \in S_A \iff \boldsymbol{\Sigma}^A\boldsymbol{\sigma} \in S_A.$$

Ako za neki $A \in \mathbb{N}$ vrijedi da je

$$\boldsymbol{\Sigma}^A\boldsymbol{\sigma} \in S_A,$$

tada prema (a) postoje koeficijenti $\alpha_1, \ldots, \alpha_A$ takvi da je

$$\boldsymbol{\Sigma}^A\boldsymbol{\sigma} = \sum_{k=1}^{A} \alpha_k \boldsymbol{\Sigma}^{k-1}\boldsymbol{\sigma}.$$

Neka je $m := \min\{k : \alpha_k \neq 0\}$. Množenjem gornje relacije sa Σ^{-m} dobivamo

$$\boldsymbol{\Sigma}^{A-m}\boldsymbol{\sigma} = \alpha_m\boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma} + \sum_{k=m+1}^{A} \alpha_k \boldsymbol{\Sigma}^{k-1}\boldsymbol{\sigma}.$$

Budući da $\alpha_m \neq 0$, ovo nam daje

$$\boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma} = \frac{1}{\alpha_m}\left(\boldsymbol{\Sigma}^{A-m}\boldsymbol{\sigma} - \sum_{k=m+1}^{A} \alpha_k \boldsymbol{\Sigma}^{k-1}\boldsymbol{\sigma}\right),$$

pa je

$$\boldsymbol{\beta}_{LS} \in S_A.$$

Obrnuto, ako za neki $A \in \mathbb{N}$ vrijedi da je $\boldsymbol{\beta}_{LS} \in S_A$, tada postoje koeficijenti $\alpha_1, \ldots, \alpha_A$ takvi da je

$$\boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma} = \sum_{k=1}^{A} \alpha_k \boldsymbol{\Sigma}^{k-1}\boldsymbol{\sigma}.$$

Neka je $m := \max\{k : \alpha_k \neq 0\}$. Množenjem gornje relacije sa $\boldsymbol{\Sigma}^{A-m+1}$ dobivamo

$$\boldsymbol{\Sigma}^{A-m}\boldsymbol{\sigma} = \sum_{k=1}^{M-1} \alpha_k \boldsymbol{\Sigma}^{A-m+k}\boldsymbol{\sigma} + \alpha_m \boldsymbol{\Sigma}^{A}\sigma,$$

pa je

$$\boldsymbol{\Sigma}^{A}\sigma = \frac{1}{\alpha_m}\left(\boldsymbol{\Sigma}^{A-m}\boldsymbol{\sigma} - \sum_{k=1}^{M-1} \alpha_k \boldsymbol{\Sigma}^{A-m+k}\boldsymbol{\sigma}\right) \in S_A.$$

□

Teorem 3.3.2. *(a) M je jednak broju svojstvenih vrijednosti ν_k takvih da je $\langle \boldsymbol{\eta_k}, \sigma\rangle \neq 0$ za barem jedan svojstveni vektor $\boldsymbol{\eta_k}$ pridružen ν_k.*

(b) M je jednak broju relevantnih komponenti od $\boldsymbol{x}$ za predviđanje y.

(c) Relevantni svojstveni vektori $\boldsymbol{\eta_1}, \ldots, \boldsymbol{\eta}_M$ iz 2.14 također razapinju S_M.

Dokaz. (a) Za neki $a \in \mathbb{N}$ (kojeg ćemo definirati kasnije) promotrimo sljedeći sustav linearnih jednadžbi s nepoznanicama $c_1, \ldots, c_a$:

$$\sum_{k=1}^{a} c_k \boldsymbol{\Sigma}^{k-1}\boldsymbol{\sigma} = \mathbf{0}. \tag{3.38}$$

Raspisom analognim 2.12 lako dobijemo da je

$$\boldsymbol{\Sigma}^{-k}\boldsymbol{\sigma} = \sum_{j=1}^{K} \frac{\langle \boldsymbol{\sigma}, \boldsymbol{\eta}_j\rangle}{\nu_j^k}\boldsymbol{\eta_j}.$$

Uvrštavanjem toga u sustav 3.38 dobivamo

$$\mathbf{0} = \sum_{k=1}^{a} c_k \boldsymbol{\Sigma}^{k-1}\boldsymbol{\sigma} = \sum_{k=1}^{a} c_k \left(\sum_{j=1}^{K} \frac{\langle \boldsymbol{\sigma}, \boldsymbol{\eta}_j \rangle}{\nu_j^k} \boldsymbol{\eta}_j \right) = \sum_{j=1}^{K} \boldsymbol{\eta}_j \left(\sum_{k=1}^{a} \frac{c_k}{\nu_j^k} \langle \boldsymbol{\sigma}, \boldsymbol{\eta}_j \rangle \right).$$

Budući da su $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_K$ nezavisni, taj je sustav ekvivalentan sljedećem:

$$\sum_{k=1}^{a} \frac{c_k}{\nu_j^k} = 0, \quad \text{za sve } j \text{ takve da } \langle \boldsymbol{\sigma}, \boldsymbol{\eta}_j \rangle \neq 0. \tag{3.39}$$

Neka je J broj jednadžbi u sustavu 3.39. Stavit ćemo da je $a = J$. Tada sustav postaje kvadratni pa možemo promotriti determinantu njegove matrice koja je dana s

$$\begin{pmatrix} 1 & \frac{1}{\nu_1} & \frac{1}{\nu_2} & \cdots & \frac{1}{\nu_J} \\ 1 & \frac{1}{\nu_1^2} & \frac{1}{\nu_2^2} & \cdots & \frac{1}{\nu_J^2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \frac{1}{\nu_1^J} & \frac{1}{\nu_2^J} & \cdots & \frac{1}{\nu_J^J} \end{pmatrix}.$$

Riječ je o poznatoj Vandermondeovoj matrici, koja je regularna ako i samo ako su svi $\nu_1, \ldots, \nu_J$ međusobno različiti, što nam daje sljedeći niz ekvivalencija:

$$\begin{aligned} a \leq J &\iff \nu_j \text{ u 3.39 su međusobno različiti.} \\ &\iff \text{sustav 3.39 ima jedinstveno rješenje } c_1 = \cdots = c_a = 0. \\ &\iff \text{sustav 3.38 ima jedinstveno rješenje } c_1 = \cdots = c_a = 0. \\ &\iff \boldsymbol{\sigma}, \boldsymbol{\Sigma}^{-1}\sigma, \ldots, \boldsymbol{\Sigma}^{a-1}\boldsymbol{\sigma} \text{ su nezavisni.} \end{aligned}$$

Sad po Teoremu 3.3.1 (b) imamo da je $J = M$.

(b) Slijedi iz (a) i Definicije 2.3.1.

(c) Dokaz započinjemo već korištenom relacijom

$$\boldsymbol{\Sigma}^{k-1}\boldsymbol{\sigma} = \sum_{j=1}^{K} \nu_j^{k-1} \langle \boldsymbol{\sigma}, \boldsymbol{\eta}_j \rangle \boldsymbol{\eta}_j.$$

Na isti način kao u 2.14 iz sume na desnoj strani možemo izbaciti pribrojnike kod kojih je $\langle \boldsymbol{\sigma}, \boldsymbol{\eta}_j \rangle = 0$ te ju dodatno profiniti tako da imamo samo jedan pribrojnik po svojstvenoj vrijednosti. Koristeći Teorem 3.3.1(a) dobivamo da se svaki vektor baze od S_A može izraziti pomoću M relevantnih svojstvenih vektora pa stoga i oni čine bazu.

□

Teorem 3.3.3. $\boldsymbol{\beta_{A,PLS} = \beta_{LS}} \iff A = M$.

Dokaz. Kombiniranjem 3.33 i 3.35 dobivamo

$$\boldsymbol{w}_{A+1} = \boldsymbol{\sigma} - \boldsymbol{\Sigma\beta}_{A,PLS}.$$

Stoga je $\boldsymbol{\beta}_{A,PLS} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\sigma}$ ako i samo ako je $\boldsymbol{w}_{A+1} = 0$. □

3.4 Faktorski modeli

Prisjetimo se PCR modela s kraja prethodnog poglavlja. Pokazali smo da vrijedi

$$\boldsymbol{x} = \boldsymbol{\mu_x} + \sum_{k=1}^{M} u_k \boldsymbol{\eta}_k + \boldsymbol{e}, \tag{3.40}$$

pri čemu su

$$\begin{aligned} u_k &= \langle \boldsymbol{x} - \boldsymbol{\mu_x}, \boldsymbol{\eta}_k \rangle, \quad k = 1, \ldots, M \\ \boldsymbol{e} &= \sum_{k=M+1}^{K} \langle \boldsymbol{x} - \boldsymbol{\mu_x}, \boldsymbol{\eta}_k \rangle \boldsymbol{\eta}_k, \end{aligned} \tag{3.41}$$

i

$$y = \sum_{k=1}^{M} \gamma_k u_k + e_0, \tag{3.42}$$

gdje je

$$\gamma_k = \frac{\langle \boldsymbol{\sigma}, \boldsymbol{\eta_k} \rangle}{\nu_k}, \; k = 1, \ldots, M.$$

Sljedeći teorem pokazuje vezu relevantnih komponenti s koeficijentima $\boldsymbol{p}_k$ i q_k iz originalnog PLS algoritma.

Teorem 3.4.1. *(a) U rastavima 3.40 i 3.42 sve su varijable $u_1, \ldots, u_M, e_0$ i $\boldsymbol{e}$ međusobno nekorelirane i normalne.*

(b) Kad je $A = M$, 3.6 je ekvivalentan 3.40, dok je 3.7 ekvivalentan 3.42. Preciznije:

(i) $\boldsymbol{e}_M = \boldsymbol{e}$ *i* $f_M = e_0$ *pa je*

$$\sum_{a=1}^{M} \boldsymbol{p}_a t_a = \sum_{k=1}^{M} \boldsymbol{\eta}_k u_k, \tag{3.43}$$

$$\sum_{a=1}^{M} q_a t_a = \sum_{k=1}^{M} \gamma_k u_k. \tag{3.44}$$

(ii) Vektori $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M$ *razapinju* S_M.

(iii) U Hilbertovom prostoru normalnih varijabli s očekivanjem nula, opremljenim skalarnim produktom $\langle x, y\rangle = \mathbb{E}(xy)$*, ortogonalni skupovi* $\{t_1, \ldots, t_M\}$ *i* $\{u_1, \ldots, u_M\}$ *razapinju isti potprostor* U_M.

Dokaz. (a) Bez smanjenja općenitosti pretpostavit ćemo da je $\boldsymbol{\mu}_{\boldsymbol{x}} = \mathbf{0}$ i $\mu_y = 0$. Prema definiciji relevantnih svojstvenih vektora imamo da je $\langle \boldsymbol{\eta}_k, \boldsymbol{\sigma}\rangle = 0$ kad je $k > M$. Stoga je

$$\begin{aligned}\operatorname{Cov}(\boldsymbol{e}, y) &\overset{\mu_y=0}{=} \mathbb{E}(\boldsymbol{e}y) \overset{3.41}{=} \mathbb{E}\left(\sum_{k=M+1}^{K} \langle \boldsymbol{x}, \boldsymbol{\eta}_k\rangle \boldsymbol{\eta}_k \cdot y\right) \\ &= \sum_{k=M+1}^{K} \langle \mathbb{E}(\boldsymbol{x}y), \boldsymbol{\eta}_k\rangle \boldsymbol{\eta}_k = \sum_{k=M+1}^{K} \langle \boldsymbol{\sigma}, \boldsymbol{\eta}_k\rangle \boldsymbol{\eta}_k = \mathbf{0}.\end{aligned}$$

Slično, zbog ortogonalnosti svojstvenih vektora za $k \leq M$ imamo

$$\begin{aligned}\operatorname{Cov}(\boldsymbol{e}, u_k) = \mathbb{E}(\boldsymbol{e}, u_k) &= \mathbb{E}\left(\sum_{j=M+1}^{K} \langle \boldsymbol{x}, \boldsymbol{\eta}_j\rangle \boldsymbol{\eta}_j \langle \boldsymbol{x}, \boldsymbol{\eta}_k\rangle\right) \\ &= \sum_{j=M+1}^{K} \mathbb{E}(\langle \boldsymbol{x}, \boldsymbol{\eta}_j\rangle\langle \boldsymbol{x}, \boldsymbol{\eta}_k\rangle)\boldsymbol{\eta}_j = \sum_{j=M+1}^{K} \mathbb{E}(\boldsymbol{\eta}_j' \boldsymbol{x}\boldsymbol{x}' \boldsymbol{\eta}_k)\boldsymbol{\eta}_j \\ &= \sum_{j=M+1}^{K} [\boldsymbol{\eta}_k' \, \mathbb{E}(\boldsymbol{x}\boldsymbol{x}')\boldsymbol{\eta}_j]\boldsymbol{\eta}_j = \sum_{j=M+1}^{K} [\boldsymbol{\eta}_k' \Sigma \boldsymbol{\eta}_j]\boldsymbol{\eta}_j \\ &= \sum_{j=M+1}^{K} [\boldsymbol{\eta}_k' \nu_j \boldsymbol{\eta}_j]\boldsymbol{\eta}_j = \mathbf{0}.\end{aligned}$$

Korištenjem istog trika dobivamo da je za $j, k \leq M$ takve da $j \neq k$

$$\operatorname{Cov}(u_j, u_k) = \mathbb{E}(u_j u_k) = \mathbb{E}(\langle \boldsymbol{x}, \boldsymbol{\eta}_j\rangle\langle \boldsymbol{x}, \boldsymbol{\eta}_k\rangle) = 0.$$

(b) Za ostatak dokaza poslužit ćemo se alternativnim algoritmom koji nam daje rastave

$$\boldsymbol{x} = \boldsymbol{\mu_x} + \boldsymbol{p}_1 t_1^* + \cdots + \boldsymbol{p}_a t_a^* + \boldsymbol{e}_a^*, \tag{3.45}$$

$$y = \mu_y + q_{A1}^* t_1 + \cdots + q_{AA}^* t_A + f_A^*. \tag{3.46}$$

Pritom je po Teoremu 3.2.1(a)

$$t_a^* = \frac{\langle \boldsymbol{w}_a, \boldsymbol{x} \rangle}{\langle \boldsymbol{w}_a, \boldsymbol{w}_a \rangle}, \tag{3.47}$$

pa se zapravo radi o projekciji vektora $\boldsymbol{x}$ na potprostor razapet ortogonalnim vektorima $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_a$. Neka je sada $A = M$. Po Teoremu 3.3.2 (c), u gornjem algoritmu $\boldsymbol{x}$ projiciramo i na ljusku vektora $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_M$ pa usporedbom 3.40 i 3.45 dobivamo da je $\boldsymbol{e}_M^* = \boldsymbol{e}$. Po Teoremu 3.2.1 (c) je $f_a^* = f_a$ općenito.

Prvo ćemo dokazati (*iii*). Neka je $U_M = [\{u_1, \ldots, u_A\}]$. Budući da je $u_k = \langle \boldsymbol{\eta}, \boldsymbol{x} \rangle$, 3.47 i Teorem 3.3.1(c) daju da je $U_M = [\{t_1^*, \ldots, t_A^*\}]$. Prema populacijskoj verziji Teorema 3.2.1 (b) vrijedi da je $[\{t_1^*, \ldots, t_A^*\}] = [\{t_1, \ldots, t_A\}]$, što dokazuje tvrdnju.

Nastavljamo s dokazom (*i*). Iz PLS algoritma je jasno da je svaka koordinata vektora $(\boldsymbol{p}_1 t_1 + \cdots \boldsymbol{p}_M t_M)$ projekcija pripadne koordinate od $\boldsymbol{x}$ na U_M, dok je $(q_1 t_1 + \cdots + q_M t_M)$ projekcija od y. Kad usporedimo to s 3.40 i 3.42, jasno je da pripadni reziduali moraju biti isti, tj. $\boldsymbol{e}_M = \boldsymbol{e}$ i $f_M = f$.

Konačno, za proizvoljan $j \leq M$ imamo

$$\begin{aligned} \sum_{a=1}^{M} \boldsymbol{p}_a t_a &= \sum_{k=1}^{M} \boldsymbol{\eta}_k u_k \quad \Big| \cdot t_j \\ \boldsymbol{p_j} t_j^2 &= \sum_{k=1}^{M} \boldsymbol{\eta}_k u_k t_j \quad \Big| \mathbb{E}(\cdot) \\ \boldsymbol{p_j}\, \mathbb{E}(t_j^2) &= \sum_{k=1}^{M} \boldsymbol{\eta}_k\, \mathbb{E}(u_k t_j). \end{aligned}$$

Budući da je $\mathrm{Var}(t_j) > 0$, imamo da je $\boldsymbol{p}_j \in [\{\boldsymbol{\eta_1}, \ldots, \boldsymbol{\eta}_M\}]$. Stoga je

$$[\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_M\}] \subseteq [\{\boldsymbol{\eta_1}, \ldots, \boldsymbol{\eta}_M\}].$$

Koristeći ortogonalnost vektora $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_M$ i činjenicu da je $\mathrm{Var}(u_j) > 0$ za svaki $j \leq M$, analogno dobijemo i $\boldsymbol{\eta}_j \in [\{\boldsymbol{\eta_1}, \ldots, \boldsymbol{\eta}_M\}]$, otkud slijedi druga inkluzija, a samim time i (*ii*).

□

Poglavlje 4

Usporedba metoda na podacima

Za kraj rada testirat ćemo kako se metode opisane u prethodnim poglavljima ponašaju u primjeni. Podaci će biti simulirani tako da zadovoljavaju pretpostavku da su kovarijatni vektor $\boldsymbol{x}$ i zavisna varijabla $\boldsymbol{y}$ normalni te u srednjem linearno povezani. Jednostavnosti radi ćemo, kao i u prethodnom poglavlju pretpostaviti da je y slučajna varijabla (a ne vektor) te da je $\mathrm{Var}(x_i) = 1$ za sve komponente od $\boldsymbol{x}$, dok ćemo među eksperimentima varirati sljedeće parametre:

1. dimenziju K vektora $\boldsymbol{x}$, $K \in \{5, 10, 50, 100, 200\}$,
2. koeficijent korelacije ρ među kovarijatama, $\rho \in \{0.01, 0.1, 0.2, 0.5, 0.9, 0.99\}$,
3. veličinu uzorka za treniranje modela $N \in \{K, 2K, 3K, 5K, 10K\}$,
4. broj komponenti n koje se koriste za predviđanje u PLS i PCR algoritmu, $n \in \{\lceil 0.01K \rceil, \lceil 0.1K \rceil, \lceil 0.25K \rceil, \lceil 0.5K \rceil, \lceil 0.8K \rceil\}$.

Dakle, prvo simuliramo uzorak duljine N varijabli

$$\begin{aligned} x_i &\sim N(0,1), \quad i = 1, \ldots, K, \\ \beta_i &\sim N(0,25)^1, \quad i = 1, \ldots, K, \\ \varepsilon &\sim N(0,1), \end{aligned} \tag{4.1}$$

a zatim konstruiramo uzorak zavisne varijable y formulom

$$y = \boldsymbol{\beta x} + \epsilon.$$

[1] Pokazalo se da su rezultati puno stabilniji ukoliko su koeficijenti β_i po apsolutnoj vrijednosti u prosjeku veći od $\mathrm{Var}(x_i) = 1$. Time na neki način "učvršćujemo" linearnu vezu.

Pritom će u svakom eksperimentu vrijediti i

$$\mathrm{Cov}(x_i, x_j) = \rho,\ i \neq j;$$

dakle, nećemo provoditi eksperimente u kojima kovarijacijska matrica Σ_{xx} sadrži više od dvije vrijednosti. Također, broj glavnih komponenti u PCR-u će u svakom eksperimentu biti jednak broju koraka koje smo napravili u konstrukciji PLS modela.

U Tablici 4.1 vidimo dio rezultata nastalih takvim eksperimentiranjem.

Model \ n	2	20	40	100	160
linreg	116.20	116.20	116.20	116.20	116.20
pcr	2235.12	1967.13	1753.66	1212.92	486.72
pls	1203.17	256.73	168.04	110.45	97.42

Tablica 4.1: MSE za $K = N = 500,\ \rho = 0.5$.

Jedan razuman način validacije naših modela bio bi da koristeći distribucije iz kojih smo generirali podatke izračunamo populacijski $\boldsymbol{\beta}$ pa za grešku modela uzmemo koliko se njegov koeficijent razlikuje od populacijskog, tj. ako je naš model dan s $y = \hat{\boldsymbol{\beta}}\boldsymbol{x}$, njegovu grešku možemo računati kao

$$\mathrm{Err}(\mathrm{Model}) = \left\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\right\|_2 .$$

Tako su, primjerice, modeli validirani u [2].

Međutim, kako je prilikom visoke korelacije kovarijata taj $\boldsymbol{\beta}$ "nestabilan", mi ćemo umjesto toga testirati naše modele na velikom testnom uzorku. Preciznije, prvo ćemo izgenerirati vrlo velik uzorak [2], zatim trenirati model na njegovom podskupu veličine N, a na ostatku izračunati srednju kvadratnu grešku MSE (engl. *Mean squared error*), što će nam uz R^2 biti primarna metrika za validaciju modela. Takav pristup ima dvije prednosti:

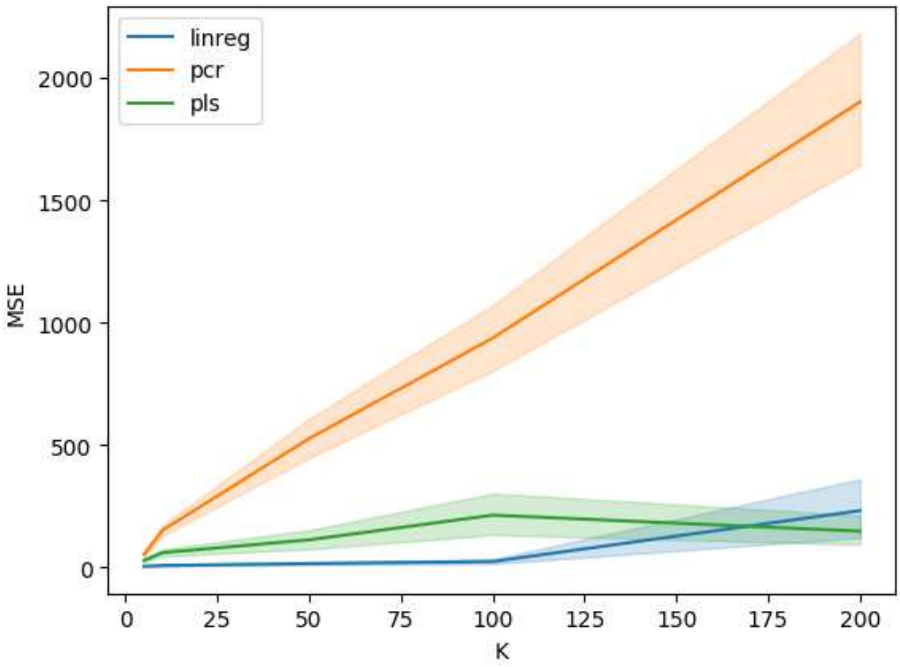
1. Metrike poput kvadratne greške su interpretabilnije od udaljenosti do stvarnog $\boldsymbol{\beta}$.
2. Tako se stvari rade u praksi (jer ne znamo stvarne distribucije pa ni vrijednost populacijskog koeficijenta); na danom uzorku istreniramo model, zatim ga validiramo na testnom skupu pa šaljemo u produkciju.

Za cijeli je gore navedeni postupak korišten programski jezik Python [13]: podaci su simulirani pomoću paketa NumPy [3], dok je za trening i validaciju modela korišten scikit-learn [12]. Za organizaciju i spremanje podataka zaslužan je paket Pandas [10], dok su grafovi nacrtani pomoću paketa Matplotlib [6] i Seaborn [14]. Puni kod može se pronaći na [9].

[2] Za svaki ρ simulirat ćemo uzorak veličine 100 000.

4.1 Rezultati

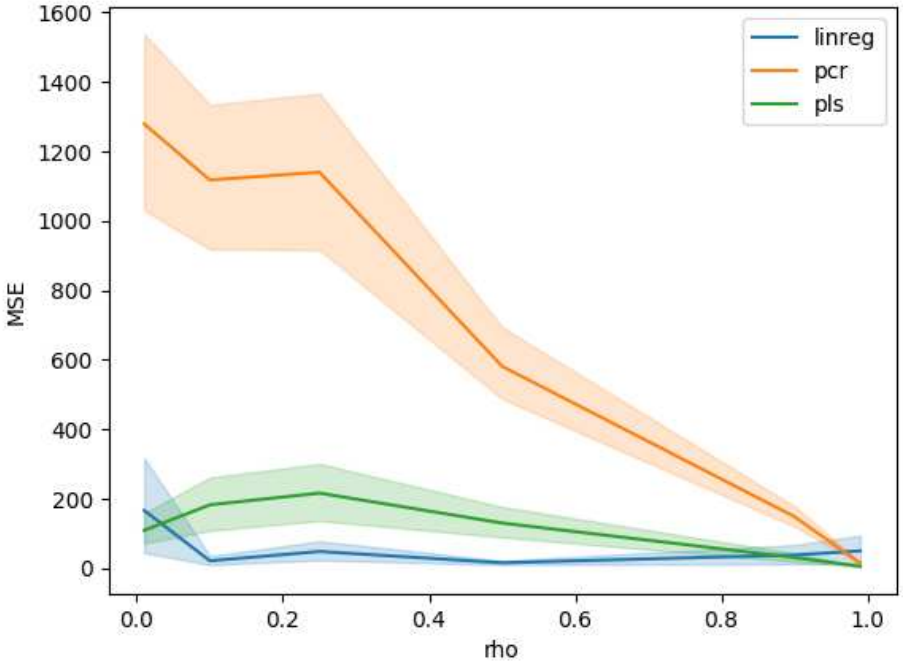
Ukoliko je $\rho = 0$, tada je i $y \sim N(0, 1)$, pri čemu Var(y) raste s ρ i K. Stoga je za očekivati da će i MSE rasti porastom dimenzije K, što pokazuje Slika 4.1 [3]. S druge strane, iako porast koeficijenta korelacije ρ povećava varijancu zavisne varijable, on također omogućava PCR i PLS algoritmima da s manje komponenti stvore što bolje rastave pa točnost tih modela raste porastom zavisnosti među kovarijatama, što možemo vidjeti na Slici 4.2. Na istoj slici vidimo i da se PLS puno bolje ponaša od PCR-a kad je ρ malen. To je upravo zbog toga što PLS paralelno rastavlja $\boldsymbol{x}$ i y, dok PCA nema nikakvu informaciju o veličini parametara β_i te ne može pri izboru glavnih komponenti znati koje varijable x_i najviše utječu na y.



Slika 4.1: Ovisnost srednje kvadratne greške o K.

Nadalje, smisleno je promatrati i kako se PCR i PLS ponašaju s obzirom na relativan broj komponenti koje koriste za predviđanje (Slika 4.3). Tu se PCR pokazuje puno osjetljivijim, i to poglavito kod manjih razina korelacije. To je očekivano, jer tad PCA s malo glavnih komponenti ne može dobro reproducirati ni $\boldsymbol{x}$, a kamoli y. Na grafovima možemo

[3] Na svim grafovima vrijednosti na y-osi predstavljaju prosječni MSE, dok su pruge oko linija širine standarne devijacije. Primjerice, kad promatramo presjek pravca $K = 100$ s pravcem PCR, njegova y-koordinata odgovarat će prosjeku MSE po svim ρ, N i n kad je $K = 100$, dok će širina pruge na tom mjestu biti standarna devijacija tih podataka.



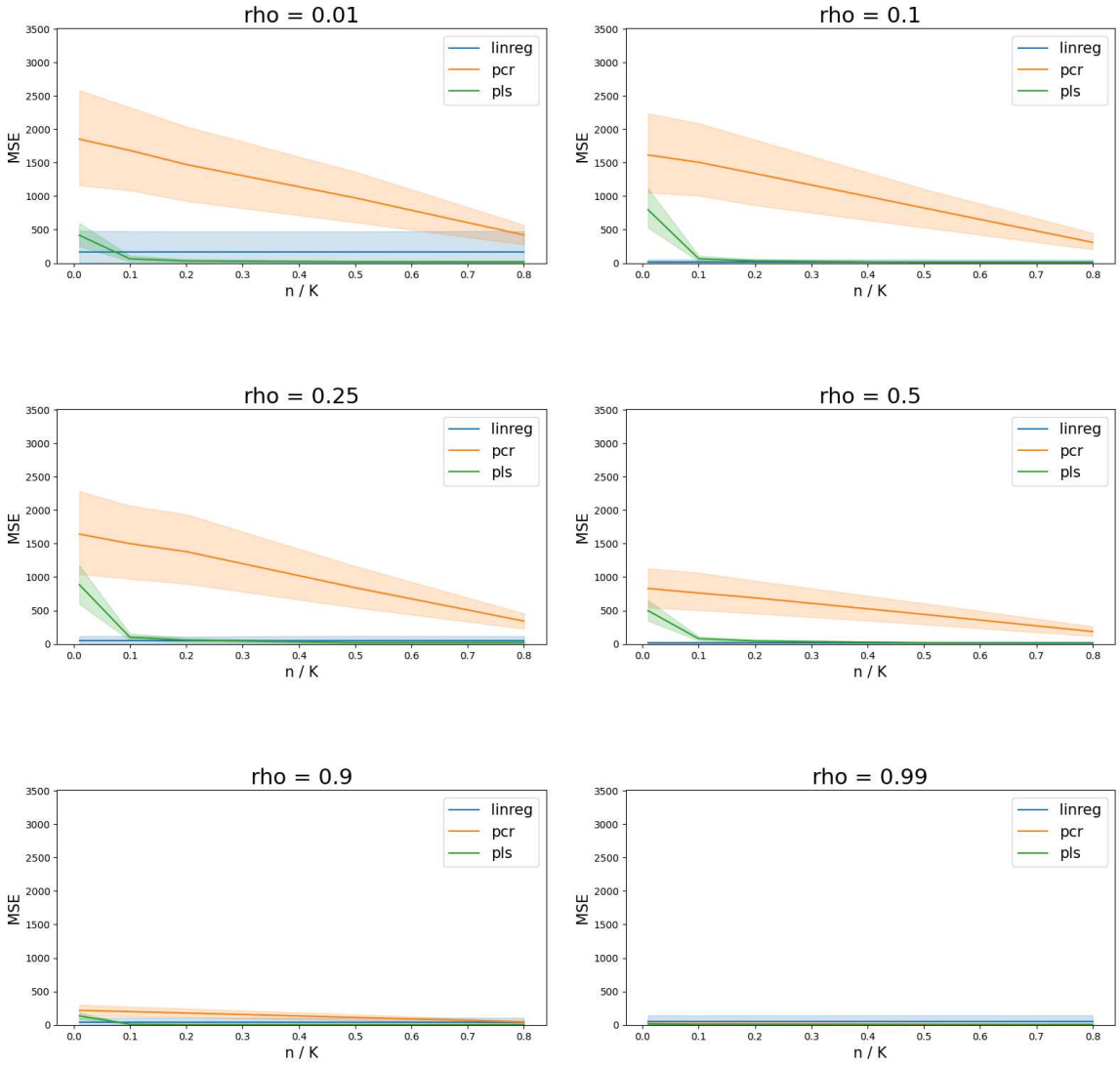
Slika 4.2: Ovisnost srednje kvadratne greške o ρ.

uočiti i kako MSE u PLS-u naglo padne s $^n/_K = 0.01$ na $^n/_K = 0.1$. To je zato što je za većinu K [4] $\lceil 0.01K \rceil = 1$, a PLS u jednom koraku stvara samo komponentu koja iz e_0 predviđa f_0, dok u 10 koraka može puno bolje shvatiti odnos varijabli. Ipak, i taj jedan korak čini ga boljim od PCR-a s jednom komponentom jer, za razliku od njega, barem ima neku informaciju o y. Zanimljivo je da nakon tog prvog koraka PLS na mnogim grafovima bolje predviđa od linearne regresije. Taj je fenomen uočljiv i u Tablici 4.1

Kako bismo to objasnili, pogledajmo grafove na Slici 4.4, koji prikazuju kako relativna veličina uzorka utječe na uspješnost modela. Tu se najosjetljivijom pokazala linearna regresija (poglavito kad je $N < 2K$ i ρ jako visok), dok su PLS i PCR vrlo robustni. Razlog za to možemo pronaći u osjetljivosti invertiranja uzoračke kovarijacijske matrice te činjenici da je za dvostruko veći uzorak ona puno bliže populacijskoj $\mathbf{\Sigma}_{xx}$. Tablica 4.1 prikazuje upravo slučaj $N = K$, kritičan za linearnu regresiju. Budući da je za svaki $^n/_K$ u grafovima na Slici 4.3 jedan eksperiment s $N = K$, on podiže prosjek MSE-a za linearnu regresiju iznad PLS-ovog.

Zanimljivo je uočiti i da su grafovi na 4.3 svi međusobno slični, osim što se PCR greška značajno smanjuje povećanjem koreliranosti. Isto vrijedi i za grafove na 4.4. Dakle, osim već navedenih opaski (koje su se dale naslutiti iz prirode metoda), ne uočavamo nikakve

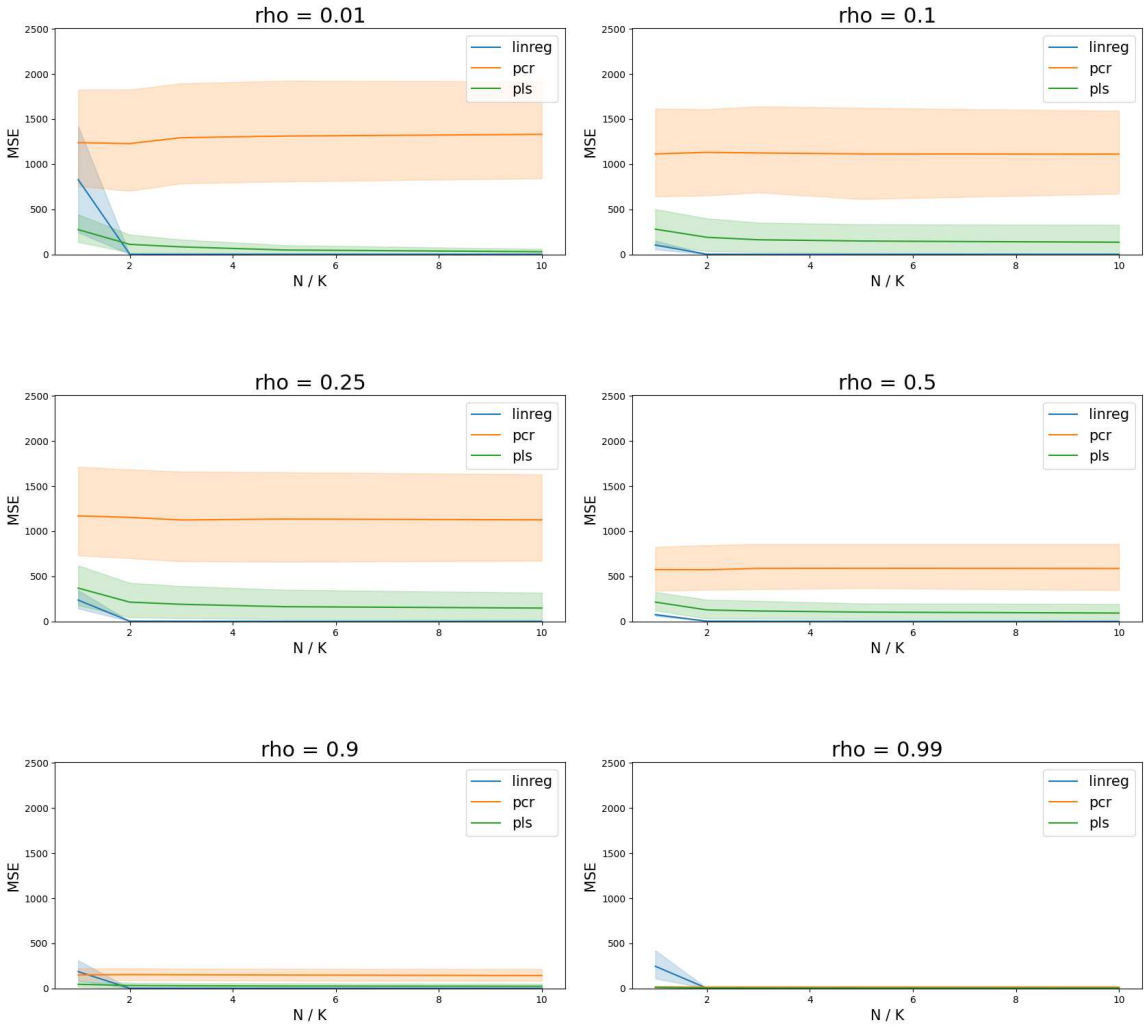
[4] Za sve K osim za 200.



Slika 4.3: Ovisnost srednje kvadratne greške o omjeru n/K.

nove veze među hiperparametrima K, ρ i N niti imamo ikakvih prevelikih neobjašnjivih oscilacija na grafovima. To je zato što su sve tri metode gotovo deterministične [5], a testni skup je vrlo velik pa je razina stohastičnosti u eksperimentima minimalna.

[5] Implementacija PCA-a u paketu scikit-learn sadrži slučajni faktor kako bi brže pronalazila glavne komponente na velikim uzorcima.



Slika 4.4: Ovisnost srednje kvadratne greške o omjeru N/k.

Poglavlje 5

Dodatak

src/config.py

```
import numpy as np

RANDOM_STATE = 43
RNG = np.random.RandomState(RANDOM_STATE)

GLOBAL_PATH_TO_REPO = "~/developer/PLS"
```

---

src/EDistribution.py

```
from enum import Enum
import numpy as np
from src.config import RNG


class EDistribution(str, Enum):
    normal = "normal"

    @staticmethod
    def get_sample(
        K: int,
        distribution_type: str,
        sample_size: int,
        rho: float,
```

```
        mean: float = 0,
    ):

        if distribution_type == EDistribution.normal:
            cov = rho * np.ones([K, K])
            cov = cov + np.diag(1 - rho * np.ones(K))
            X = RNG.multivariate_normal(
                mean=np.zeros(K) * mean,
                cov=cov,
                size=sample_size,
            )

        return X
```

---

src/EModel.py

```
from enum import Enum

import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.decomposition import PCA
from sklearn.cross_decomposition import PLSRegression
from sklearn.pipeline import make_pipeline
from sklearn.metrics import mean_squared_error

from src.config import RANDOM_STATE


class EModel(str, Enum):
    linreg = "linreg"
    pcr = "pcr"
    pls = "pls"

    @staticmethod
    def train(
        model_name: str,
        X: np.array,
        y: np.array,
```

```
    n_components: int | None = None,
):

    if model_name not in list(EModel):
        raise ValueError(
            f"""No such model.
            Available models are {list(EModel)}"""
        )

    if model_name == EModel.linreg:
        model = LinearRegression()

    elif model_name == EModel.pcr:
        model = make_pipeline(
            PCA(
                n_components=n_components,
                random_state=RANDOM_STATE,
            ),
            LinearRegression(),
        )

    elif model_name == EModel.pls:
        model = PLSRegression(n_components=n_components)

    model.fit(X, y)

    return model

@staticmethod
def train_and_evaluate_all_models(
    X_train: np.array,
    X_test: np.array,
    y_train: np.array,
    y_test: np.array,
    n_components: int,
):

    """Trains LS, PLS and PCR models on X_train
    and y_train and evaluates them on X_test and y_test.
```

```
        Args:
            n_components (int): Number of relevant
                components to use for predicition in PCR and PLS.

        Returns:
            dict: Contains model names with
            their respective R2 and MSE values.
        """

        score_dict = {"R2": {}, "MSE": {}}

        for model_name in list(EModel):
            model = EModel.train(
                model_name=model_name,
                X=X_train,
                y=y_train,
                n_components=n_components,
            )
            score_dict["R2"][
                model_name.value
            ] = model.score(X_test, y_test)
            preds_test = model.predict(X_test)
            score_dict["MSE"][
                model_name.value
            ] = mean_squared_error(preds_test, y_test)

        return score_dict
```

---

src/get_xy_sample.py

```
import numpy as np

from src.EDistribution import EDistribution
from src.config import RNG


def get_xy_sample(
```

```
    distribution_type: EDistribution,
    beta: np.array,
    rho: float,
    sample_size: int,
) -> tuple[np.array, np.array]:

    """Generates sample of X and y such that
        y = X*beta + eps, eps ~ N(0, 1)

    Args:
        beta (np.array): Linear
            transformation vector.
        rho (float): Covariance
            between covariates.

    Returns:
        tuple: X-sample, y-sample.
    """

    K = beta.shape[0]
    cov = rho * np.ones([K, K])
    cov = cov + np.diag(1 - rho * np.ones(K))

    K = beta.shape[0]
    X = EDistribution.get_sample(
        K=K,
        distribution_type=distribution_type,
        sample_size=sample_size,
        rho=rho,
    )

    error = RNG.normal(loc=0, scale=1, size=X.shape[0])
    y = np.matmul(X, beta) + error

    return X, y
```

src/save_results.py

```python
import sys

sys.path.append(".")

import pandas as pd
from sklearn.model_selection import train_test_split
from tqdm import tqdm
from itertools import product
import numpy as np

from src.EModel import EModel
from src.EDistribution import EDistribution
from src.get_xy_sample import get_xy_sample
from src.config import (
    RNG,
    RANDOM_STATE,
    GLOBAL_PATH_TO_REPO,
)
from src.EDistribution import EDistribution


save_results_config = {
    "Ks": [200, 100, 50, 10, 5],
    "N_K_ratios": [10, 5, 3, 2, 1],
    "n_K_ratios": [0.01, 0.1, 0.2, 0.5, 0.8],
    "rhos": [0.01, 0.1, 0.25, 0.5, 0.9, 0.99],
    "sample_size": 100_000,
}


def save_results(
    distribution_type: EDistribution,
    csv_name: str | None = None,
    config: dict = save_results_config,
):

    if csv_name is None:
        csv_name = f"{distribution_type}"
```

```
save_path = f"{GLOBAL_PATH_TO_REPO}/data/{csv_name}.csv"

index_columns = [
    "K",
    "n_K_ratio",
    "n",
    "N_K_ratio",
    "M",
    "rho",
]
columns = ["model", "R2", "MSE"]
df = pd.DataFrame(None, columns=index_columns + columns)

Ks = config["Ks"]
N_K_ratios = config["N_K_ratios"]
n_K_ratios = config["n_K_ratios"]
rhos = config["rhos"]
sample_size = config["sample_size"]

loader = tqdm(
    product(Ks, rhos), total=len(Ks) * len(rhos)
)

for K, rho in loader:

    beta = RNG.normal(loc=0, scale=5, size=K)
    X, y = get_xy_sample(
        distribution_type=distribution_type,
        beta=beta,
        rho=rho,
        sample_size=sample_size,
    )

    for N_K_ratio in N_K_ratios:

        M = K * N_K_ratio

        (
```

```
            X_train,
            X_test,
            y_train,
            y_test,
        ) = train_test_split(
            X,
            y,
            random_state=RANDOM_STATE,
            train_size=M,
        )

        for n_K_ratio in n_K_ratios:

            n = int(np.ceil(K * n_K_ratio))

            score_dict = (
                EModel.train_and_evaluate_all_models(
                    X_train=X_train,
                    X_test=X_test,
                    y_train=y_train,
                    y_test=y_test,
                    n_components=n,
                )
            )
            score_df = pd.DataFrame(score_dict)
            score_df.index.name = "model"
            score_df = score_df.reset_index()
            for col_name, value in zip(
                index_columns,
                [K, n_K_ratio, n, N_K_ratio, M, rho],
            ):
                score_df[col_name] = value
            df = pd.concat([df, score_df])
            loader.set_postfix(
                **score_dict["R2"]
                | {"rho": rho, "K": K, "n": n}
            )

    df.to_csv(save_path, index=False)
```

```
    df.to_csv(save_path, index=False)


if __name__ == "__main__":
    save_results(
        distribution_type=EDistribution.normal,
    )
```

Bibliografija

[1] Drmač, Zlatko, Hari Vjeran, Miljenko Marušić, Mladen Rogina, Saša Singer i Sanja Singer: *Numerička analiza*, 2003.

[2] Göktaş, Atila i Özge Akkuş: *Comparison of partial least squares with other prediction methods via generated data*. Journal of Statistical Computation and Simulation, 90(16):3009–3024, 2020. https://doi.org/10.1080/00949655.2020.1793342.

[3] Harris, Charles R., K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke i Travis E. Oliphant: *Array programming with NumPy*. Nature, 585(7825):357–362, rujan 2020. https://doi.org/10.1038/s41586-020-2649-2.

[4] Helland, Inge S.: *On the structure of the partial least squares regression*. Communications in Statistics - Simulation and Computation, 17:581–607, 1988.

[5] Helland, Inge S.: *Partial Least Squares Regression and Statistical Models*. Scandinavian Journal of Statistics, 17:97–114, 1990.

[6] Hunter, J. D.: *Matplotlib: A 2D graphics environment*. Computing in Science & Engineering, 9(3):90–95, 2007.

[7] Huzak, Miljenko: *Analiza glavnih komponenti*. https://web.math.pmf.unizg.hr/nastava/ps/files/PSpredP3.pdf, preuzeto 13.12.2022.

[8] Huzak, Miljenko: *Diskriminacija i alokacija*. https://web.math.pmf.unizg.hr/nastava/ps/files/PSpredP2.pdf, preuzeto 21.12.2022.

[9] Jurić Fot, Sanjin: *PLS*. https://github.com/ninjas77/PLS, 2022.

[10] McKinney, Wes *et al.*: *Data structures for statistical computing in python*. U *Proceedings of the 9th Python in Science Conference*, svezak 445, stranice 51–56. Austin, TX, 2010.

[11] Næs, Tormod i Harald Martens: *Comparison of prediction methods for multicollinear data*. Communications in Statistics - Simulation and Computation, 14:545–576, 1985.

[12] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot i E. Duchesnay: *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12:2825–2830, 2011.

[13] Van Rossum, Guido i Fred L. Drake: *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009, ISBN 1441412697.

[14] Waskom, Michael L.: *seaborn: statistical data visualization*. Journal of Open Source Software, 6(60):3021, 2021. `https://doi.org/10.21105/joss.03021`.

Zahvale

Prije svega zahvaljujem svojoj mentorici doc. dr. sc. Azri Tafro na strpljenju i pomoći tijekom izrade ovog rada. Veliko hvala dugujem i ekipi *Matemakripti* s pub kviza, ekipi iz *bookclub*-a (onima koji su u presjeku, jasno, dvaput), Sonatašinima te svima ostalima čije mi je prijateljstvo obogatilo studij. Zahvaljujem i svim profesorima i asistentima s faksa na znanju koje su mi prenijeli u proteklih pet godina, znanju bez kojeg bi pisanje ovakvog rada bilo neizvedivo. Za kraj, zahvaljujem cijeloj svojoj obitelji na velikoj ljubavi i potpori, a posebice svojoj sestrični Teni.

Sažetak

Metoda parcijalnih najmanjih kvadrata (PLS) statistička je metoda koja se koristi za analizu odnosa dvaju slučajnih veličina $\boldsymbol{x}$ i $\boldsymbol{y}$, najčešće za predviđanje $\boldsymbol{y}$ preko $\boldsymbol{x}$. Slična je analizi glavnih komponenata (PCA), utoliko što se obje metode bave smanjenjem dimenzije, tj. pronalaze skup novih nekoreliranih varijabli koje će sadržavati maksimalno informacija o početnima; ali za razliku od PCA-a, koji pronalazi komponente za maksimalno objašnjenje varijance u $\boldsymbol{x}$, komponente nastale PLS-om objašnjavaju varijancu u obje veličine, te se stoga mogu koristiti kao ulazi u neki prediktivni model, kao što je linearna regresija. Metoda parcijalnih najmanjih kvadrata posebno je korisna kada je omjer broja opažanja i dimenzije od $\boldsymbol{x}$ malen te kada su kovarijate visoko korelirane. U ovom smo radu opisali metodu i izveli ključne teorijske rezultate vezane uz nju te pokazali kako ona u određenim slučajevima predviđa bolje od linearne regresije i regresije na glavnim komponentama.

Summary

Partial least squares (PLS) is a statistical method used for analyzing the relationship between two random variables $\boldsymbol{x}$ and $\boldsymbol{y}$, most commonly to predict $\boldsymbol{y}$ from $\boldsymbol{x}$. It is similar to principal components analysis (PCA) - both methods deal with dimension reduction, i.e. they construct a set of new uncorrelated variables that will contain maximum information about the initial ones; but unlike PCA, which finds the components that maximally explain the variance in $\boldsymbol{x}$, the components generated by PLS explain the variance in both variables, and can therefore be used as inputs to some predictive model, such as linear regression. PLS is especially useful when the ratio of the number of observations to the dimension of $\boldsymbol{x}$ is small and when the explanatory variables are highly correlated. In this paper we described the method, derived crucial theoretical results related to it, and showed that in certain cases it performs better than linear regression and principal component regression (PCR).

Životopis

Rođen sam 13. siječnja 1999. godine u Zagrebu. Nakon završene osnovne škole upisao sam IV. gimnaziju, smjer s dvojezičnim njemačkim jezikom. Paralelno sam pohađao i srednju glazbenu školu Pavla Markovca, u kojoj sam maturirao 2019.

2017. godine sam upisao preddiplomski studij matematike na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta u Zagrebu, koji sam završio 2020. Iste godine sam upisao diplomski sveučilišni studij Matematička statistika na istom fakultetu.