Metoda parcijalnih najmanjih kvadrata: Regresijski model

Sente, Tamara

Master's thesis / Diplomski rad

2016

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:915875

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-18**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

Sveučilište u Zagrebu
Prirodoslovno-matematički fakultet
Matematički odsjek

Tamara Sente

Metoda parcijalnih najmanjih kvadrata: Regresijski model

Diplomski rad

Voditelj rada:
Izv.prof.dr.sc. Miljenko Huzak

Zagreb, rujan 2016.

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. _____

Zahvaljujem se prof.dr.sc. Miljenku Huzaku na izvanrednom mentorstvu te izdvojenim višesatnim konzultacijama i strpljenju prilikom izrade ovog rada.

Sadržaj

1 Uvod **2**

2 Osnovni pojmovi i rezultati **3**
2.1 Linearni modeli više varijabli 3
2.2 Metoda glavnih komponenti - PCA 7
2.3 Dekompozicija singularnih vrijednosti - SVD 10

3 Metoda parcijalnih najmanjih kvadrata - PLS **11**
3.1 Procijenjene vrijednosti 16

4 Regresijski model - PLSR **18**
4.1 Procjena parametara 20
4.2 Konzistentnost procijenjenih parametara 23

5 Algoritmi **26**
5.1 NIPALS 27
5.2 SIMPLS 28
5.3 Usporedba algoritama 29

6 Primjer **30**
6.1 Univarijatan $\mathbf{Y}$ 30
6.1.1 Multilinearna regresija 31
6.1.2 PCR 33
6.1.3 PLSR 37
6.1.4 Usporedba rezultata 42
6.2 Multivarijatan $\mathbf{Y}$ 43
6.2.1 PLS1 44
6.2.2 PLS2 55
6.2.3 Usporedba rezultata 59

Literatura **60**

Sažetak **62**

Summary **63**

Životopis **64**

1 Uvod

PLS metoda (metoda parcijalnih najmanjih kvadrata) je popularna metoda za modeliranje na industrijskom polju. Iako se danas najčešće koristi u kemometriji, prvotno je korištena 1966. godine na području društvenih znanosti – preciznije ekonomiji, kada je dani model predstavio Herman Wold. Danas se najčešće koristi kao model i često ga se koristi u kombinaciji s regresijskim modelima, no prvotno je predstavljen kao algoritam za traženje svojstvenih vrijednosti – NIPALS, koji podsjeća na metodu potencija.

PLS je statistička metoda koja podsjeća na PCA (metoda glavnih komponenti) i MLR (multilinearna regresija). Cilj PLS metode je kao i kod PCA da reduciramo broj komponenti danih varijabli na manji skup komponenti koje nisu međusobno korelirane. Ono što je specifično za PLS je da, za razliku od PCA, maksimizira kovarijancu između zavisne ($\mathbf{X}$) i nezavisne ($\mathbf{Y}$) matrične varijable. Veza između PLS metode i MLR-a je PLSR (PLS regresija), gdje prvotno PLS metodom reduciramo broj komponenti danih varijabli te s njima (umjesto s originalnim) ulazimo u regresijski model.

U ovom radu će se obraditi teme linearnih modela više varijabli (MLR), metode glavnih komponenti (PCA) – koje ćemo usporediti s PLS metodom, odnosno s PLSR, te dekompozicija singularnih vrijednosti (SVD) koji je potreban za implementaciju algoritama.

U poglavlju 3 će se obraditi model za PLS metode, a u istom poglavlju je objašnjeno i na koji način se procjenjuju vrijednosti u praksi. S obzirom da se PLS metoda koristi kada imamo dvije matrične varijable, najčešće se koristi u kombinaciji s regresijskim modelom. Zbog toga je u poglavlju 4 objašnjen PLSR u kojem su također prikazani model i procijenjene vrijednosti.

Kako PLS više pridonosi praksi nego li teoriji, značajno je kako implementirati dani model. U poglavlju 5 su prikazani najpoznatiji algoritmi te ih se međusobno uspoređuje.

Također, u ovome radu će se u zadnjem poglavlju obraditi primjeri na kojima će se usporediti PLSR s PCR i PLR te će se obraditi usporedba nekih PLS algoritama.

2 Osnovni pojmovi i rezultati

Za razumijevanje ovog rada potrebno je poznavanje jednostavne linearne regresije, a u ovom poglavlju će se detaljnije objasniti pojmovi linearnog modela više varijabli, metode glavnih komponenti te dekompozicije singularnih vrijednosti – koji su potrebni za razumijevanje teme koja se obrađuje u radu.

2.1 Linearni modeli više varijabli

Linearni modeli više varijabli su linearni modeli koji imaju više od jedne varijable odziva

$$Y_1, Y_2, \ldots, Y_q.$$

$Y^\top = (Y_1, Y_2, \ldots, Y_q)$ je q-dimenzionalni vektor odziva (i zapisujemo ga kao vektor - stupac). Odnono, zapisujemo q varijabli odziva kao slučajan vektor. Neka je

$$Y_{1\cdot}, Y_{2\cdot}, \ldots, Y_{n\cdot} \tag{1}$$

slučajni uzorak duljine n za vektor odziva Y, pri čemu je $Y_{i\cdot}^\top = (Y_{i1}, Y_{i2}, \ldots, Y_{iq})$ i-to opažanje od Y.
Stavimo

$$Y_{\cdot j} = \begin{bmatrix} Y_{1j} \\ Y_{2j} \\ \vdots \\ Y_{nj} \end{bmatrix}$$

(time smo opisali slučajni uzorak za j-tu komponentu od Y, Y_j). Te vektore možemo staviti u matricu

$$\mathbf{Y} = \begin{bmatrix} Y_{\cdot 1} & Y_{\cdot 2} & \ldots & Y_{\cdot q} \end{bmatrix} = \begin{bmatrix} Y_{1\cdot}^\top \\ Y_{2\cdot}^\top \\ \vdots \\ Y_{n\cdot}^\top \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \ldots & Y_{1q} \\ Y_{21} & Y_{22} & \ldots & Y_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \ldots & Y_{nq} \end{bmatrix}.$$

Time smo slučajni uzorak (1) zapisali u matričnom obliku.

Kao i kod regresijskog modela, označimo sa

$$X^\top = (X_1, \ldots, X_p)$$

vektor p varijabli poticaja (i pritom stavimo $X_1 = 1$, zbog regresijskog modela sa slobodnim članom). Slično, neka je sa

$$\mathbf{X} = \begin{bmatrix} X_{\cdot 1} & X_{\cdot 2} & \ldots & X_{\cdot p} \end{bmatrix} = \begin{bmatrix} X_{1\cdot}^\top \\ X_{2\cdot}^\top \\ \vdots \\ X_{n\cdot}^\top \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & \ldots & X_{1p} \\ X_{21} & X_{22} & \ldots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \ldots & X_{np} \end{bmatrix}$$

dana matrica dizajna.

Linearni model više varijabli zapisujemo ovako

$$Y^\top = X^\top B + \boldsymbol{\varepsilon}^\top, \tag{2}$$

gdje je $Y^\top$ q-dimenzionalni vektor odziva, $X^\top$ p-dimenzionalni vektor poticaja, $B \in M_{p,q}$ matrica parametara modela, te $\boldsymbol{\varepsilon}$ q-dimenzionalni slučajni uzorak (koji predstavlja slučajnu pogrešku) za koji pretpostavljamo:

(E1) $\mathbb{E}[\boldsymbol{\varepsilon}] = 0$,

(E2) postoji kovarijacijska matrica $\operatorname{cov}(\boldsymbol{\varepsilon}) = \Sigma \in M_q$ čije elemente označavamo $\Sigma = [\sigma_{jj'}]$.

Slučajni uzorak (1) iz linearnog modela (2) se sada može zapisati

$$\mathbf{Y} = \mathbf{X}B + \mathbf{E}, \tag{3}$$

gdje je

$$\mathbf{E} = \begin{bmatrix} \varepsilon_{\cdot 1} & \varepsilon_{\cdot 2} & \dots & \varepsilon_{\cdot q} \end{bmatrix} = \begin{bmatrix} \varepsilon_{1\cdot}^\top \\ \varepsilon_{2\cdot}^\top \\ \vdots \\ \varepsilon_{n\cdot}^\top \end{bmatrix} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \dots & \varepsilon_{1q} \\ \varepsilon_{21} & \varepsilon_{22} & \dots & \varepsilon_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_{n1} & \varepsilon_{n2} & \dots & \varepsilon_{nq} \end{bmatrix}.$$

Pretpostavljamo još

(E3) $\operatorname{cov}(\varepsilon_{\cdot j}) = \sigma_{jj} I_n$,

gdje je I_n jedinična matrica reda n (iz ovog uvjeta zapravo slijedi da su različite komponente vektora $\boldsymbol{\varepsilon}_{\cdot j}$ nekorelirane).
Iz uvjeta (E1) slijedi

$$\mathbb{E}[\mathbf{E}] = 0,$$

dok iz (E2) i (E3) slijedi

$$\operatorname{cov}(\varepsilon_{ij}, \varepsilon_{i'j'}) = \sigma_{jj'}\delta_{ii'}, \quad i, i' = 1, \dots, n,\ j, j' = 1, \dots, q,$$

gdje je δ Kroneckerov simbol.

Da bismo procijenili B i Σ iz modela (2), napišimo slučajni uzorak (3) iz matričnog u vektorskom obliku

$$\underbrace{\begin{bmatrix} Y_{\cdot 1} \\ Y_{\cdot 2} \\ \vdots \\ Y_{\cdot q} \end{bmatrix}}_{nq\times 1} = \underbrace{\begin{bmatrix} \mathbf{X} & & & \\ & \mathbf{X} & & \\ & & \ddots & \\ & & & \mathbf{X} \end{bmatrix}}_{\substack{nq\times pq \\ \text{blok - dijagonalna matrica}}} \underbrace{\begin{bmatrix} B_{\cdot 1} \\ B_{\cdot 2} \\ \vdots \\ B_{\cdot q} \end{bmatrix}}_{pq\times 1} + \underbrace{\begin{bmatrix} \varepsilon_{\cdot 1} \\ \varepsilon_{\cdot 2} \\ \vdots \\ \varepsilon_{\cdot q} \end{bmatrix}}_{nq\times 1}. \tag{4}$$

Također,

$$\operatorname{cov}\begin{bmatrix}\varepsilon_{\cdot 1}\\ \varepsilon_{\cdot 2}\\ \vdots\\ \varepsilon_{\cdot q}\end{bmatrix} = \mathbb{E}\left(\begin{bmatrix}\varepsilon_{\cdot 1}\\ \varepsilon_{\cdot 2}\\ \vdots\\ \varepsilon_{\cdot q}\end{bmatrix}\begin{bmatrix}\varepsilon_{\cdot 1}^\top & \varepsilon_{\cdot 2}^\top & \cdots & \varepsilon_{\cdot q}^\top\end{bmatrix}\right) = \begin{bmatrix}\sigma_{11}I_n & \sigma_{12}I_n & \cdots & \sigma_{1q}I_n\\ \sigma_{21}I_n & \sigma_{22}I_n & \cdots & \sigma_{2q}I_n\\ \vdots & \vdots & \ddots & \vdots\\ \sigma_{q1}I_n & \sigma_{q2}I_n & \cdots & \sigma_{qq}I_n\end{bmatrix}.$$

Nadalje, definirajmo preslikavanje

$$\operatorname{vec}\colon M_{r,s} \to \mathbb{R}^{r\cdot s},\ \operatorname{vec}(A) = \begin{bmatrix}A_{\cdot 1}\\ A_{\cdot 2}\\ \vdots\\ A_{\cdot s}\end{bmatrix},$$

za $A = \begin{bmatrix}A_{\cdot 1} & A_{\cdot 2} & \cdots & A_{\cdot s}\end{bmatrix} \in M_{r,s}$.

Uočimo da je ovo preslikavanje izomorfizam vektorskih prostora.

Ukoliko je $\mathbf{Y}$ slučajna matrica, imamo $\operatorname{vec}(\mathbb{E}\mathbf{Y}) = \mathbb{E}[\operatorname{vec}(\mathbf{Y})]$, te po definiciji stavimo $\operatorname{cov}(\mathbf{Y}) := \operatorname{cov}(\operatorname{vec}(\mathbf{Y}))$.

Nadalje, definirajmo Kroneckerov produkt:
za $A = [a_{ij}] \in M_{p,q}$, $B \in M_{r,s}$, stavimo

$$A \otimes B := [a_{ij}\cdot B] = \begin{bmatrix}a_{11}B & a_{12}B & \cdots & a_{1q}B\\ a_{21}B & a_{22}B & \cdots & a_{2q}B\\ \vdots & \vdots & \ddots & \vdots\\ a_{p1}B & a_{p2}B & \cdots & a_{pq}B\end{bmatrix} \in M_{pr,qs}.$$

Lema 2.1. *Neka su $A \in M_{p,q}$, $X \in M_{q,r}$, $B \in M_{r,s}$ tada vrijedi*

$$\operatorname{vec}(AXB) = (B^\top \otimes A)\operatorname{vec}(X).$$

Sada direktno slijedi $\operatorname{cov}(\operatorname{vec}(\mathbf{E})) = \Sigma \otimes I_n$, a (4) možemo zapisati u ekvivalentnom obliku

$$\operatorname{vec}(\mathbf{Y}) = (I_q \otimes \mathbf{X})\operatorname{vec}(B) + \operatorname{vec}(\mathbf{E}).$$

Za matricu $A \in M_{n,s}$ označimo sa $L(A)$ potprostor od $\mathbb{R}^n$ razapet stupcima matrice A.

Teorem 2.2. *Neka je $Y = X\boldsymbol{\theta} + \boldsymbol{\varepsilon}$ višestruki linearni regresijski model takav da je $\mathbb{E}\boldsymbol{\varepsilon} = 0$, $\operatorname{cov}(\boldsymbol{\varepsilon}) = V > 0$, X je punog ranga.*

(i) *Ukoliko je $L(VX) = L(XU)$ za neku regularnu matricu U, tada je $\widehat{\boldsymbol{\theta}} = (X^\top X)^{-1}X^\top Y$ LS-procjenitelj u odnosu na skalarni produkt $\langle a, b\rangle := (V^{-1}a, b)$ u $\mathbb{R}^n$,*

$$\langle a, b\rangle = \sum_i \sum_j [V^{-1}]_{ij} a_i b_j.$$

(ii) *Ako je $L(VX) \leq L(X)$, tada je $l^\top\widehat{\boldsymbol{\theta}} = l^\top(X^\top X)^{-1}X^\top Y$ BLUE (*Best Linear Unbiased Estimator*, najbolji linearni nepristrani procjenitelj) za $L(\boldsymbol{\theta}) := l^\top\boldsymbol{\theta} = (l, \boldsymbol{\theta})$.*

Uz naše je pretpostavke

$$(\Sigma \otimes I_q)(I_q \otimes \mathbf{X}) = (I_q \otimes \mathbf{X})\underbrace{(\Sigma \otimes I_p)}_{\text{regularna}},$$

pa je ispunjen uvjet (i) (a time i uvjet (ii)) teorema 2.2.

Zbog toga slijedi

$$\begin{aligned}
\operatorname{vec}(\widehat{B}) &= \widehat{\operatorname{vec}(B)} \\
&= \left[(I_q \otimes \mathbf{X})^\top (I_q \otimes \mathbf{X})\right]^{-1} (I_q \otimes \mathbf{X})^\top \operatorname{vec}(\mathbf{Y}) \\
&= \left(I_q \otimes (\mathbf{X}^\top\mathbf{X})^{-1}\right)\left(I_q \otimes \mathbf{X}^\top\right)\operatorname{vec}(\mathbf{Y}) \\
&= \left(I_q \otimes (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\right)\operatorname{vec}(\mathbf{Y}) \\
&= \operatorname{vec}\left((\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}\right)
\end{aligned}$$

$$\Rightarrow \widehat{B}_{LS} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}.$$

2.2 Metoda glavnih komponenti - PCA

Pretpostavimo da imamo opservaciju $Y \in \mathbb{R}^q$, pri čemu je q velik broj. Želimo reducirati dimenziju te opservacije, no da zadržimo dovoljno informacija za donošenje relevantnih zaključaka, tj želimo minimizirati gubitak potrebnih informacija. Analiza glavnih komponenti (PCA - *Principal Component Analysis*) pronalazi linearne kombinacije originalnih varijabli koje su najbolji linearni predviditelji (prediktori) svih originalnih varijabli. Odaberemo malen broj linearnih kombinacija komponenti od Y tako da imaju sposobnost reproducirati što veći broj komponenti od Y.

Neka su $Y = (Y_1, \ldots, Y_q)^\top$, $X = (X_1, \ldots, X_p)^\top$ zadani slučajni vektori čije komponente imaju konačnu varijancu. Označimo

$$\mu_X = \mathbb{E}X = 0,\ \mu_Y = \mathbb{E}Y = 0$$

$$V_{YX} = \operatorname{cov}(Y, X) = \mathbb{E}[(Y - \mu_Y)(X - \mu_X)^\top] = \mathbb{E}[YX^\top]$$
$$V_{XY} = \operatorname{cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)^\top] = \mathbb{E}[XY^\top] = V_{YX}^\top$$

Posebno, definiramo

$$V_{XX} = \operatorname{cov}(X) = \operatorname{cov}(X, X) = \mathbb{E}[XX^\top] = \Sigma_X$$

$$V_{YY} = \operatorname{cov}(Y) = \operatorname{cov}(Y, Y) = \mathbb{E}[YY^\top] = \Sigma_Y.$$

Definicija 2.3. ***Najbolji linearni prediktor*** *od Y uz dano X je q-dimenzionalni slučajni vektor $\widehat{Y}$ koji je afina funkcija od X, tj. $\widehat{Y} = \widehat{\boldsymbol{\beta}^\top} X$, tako da*

$$\mathbb{E}\left[\left(Y - \widehat{\boldsymbol{\beta}}^\top X\right)^\top \left(Y - \widehat{\boldsymbol{\beta}}^\top X\right)\right] = \min_{\boldsymbol{\beta} \in M_{p,q}} \mathbb{E}\left[\left(Y - \boldsymbol{\beta}^\top X\right)^\top \left(Y - \boldsymbol{\beta}^\top X\right)\right].$$

U oznaci, $\widehat{Y} = P[Y \mid X]$.

Uvjet gornje definicije možemo zapisati kao

$$\|Y - \widehat{\boldsymbol{\beta}}^\top X\| = \min_{\boldsymbol{\beta} \in M_{p,q}} \|Y - \boldsymbol{\beta}^\top X\|,$$

gdje je $\|U\| = \sqrt{\mathbb{E}[UU^\top]}$ norma definirana za slučajne vektore s konačnim kovarijacijskim matricama.

Lema 2.4. *Vrijedi da je*

$$\widehat{Y} = P[Y \mid X] = \boldsymbol{\mu}_Y + \widehat{\boldsymbol{\beta}}^\top (X - \boldsymbol{\mu}_X),$$

gdje je $\widehat{\boldsymbol{\beta}}$ *(bilo koje) rješenje jednadžbe*

$$V_{XX}\boldsymbol{\beta} = V_{XY}.$$

Ukoliko je V_{XX} *pozitivno definitna (tj. regularna) matrica, tada je*

$$\widehat{\boldsymbol{\beta}} = V_{XX}^{-1} V_{XY}.$$

Od sada pa nadalje pretpostavljamo da je ispunjen drugi uvjet gornje leme.

Lema 2.5. *Neka je* G *simetrična, pozitivno definitna, a* P *bilo koja simetrična matrica, obje reda* q*. Tada postoji dijagonalna matrica* Λ *i matrica* A*, obje reda* q*, tako da vrijedi*

$$G^{-1}PA = A\Lambda,\ A^\top GA = I,\ G^{-1} = AA^\top$$

Teorem 2.6. *Neka je* G *pozitivno definitna simetrična matrica, a* P *simetrična, obje reda* q*. Tada ne-nul vektori* $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_q \in \mathbb{R}^q$ *zadovoljavaju uvjete*

(I) **(I1)** $\dfrac{\mathbf{a}_1^\top P \mathbf{a}_1}{\mathbf{a}_1^\top G \mathbf{a}_1} = \max\limits_{\mathbf{a}} \dfrac{\mathbf{a}^\top P \mathbf{a}}{\mathbf{a}^\top G \mathbf{a}},$

(I2) *za svaki* $i \geq 2$*,* $i \leq q$*, vrijedi da je* $\mathbf{a}_i^\top G \mathbf{a}_j = 0$*, za svaki* $j \leq i-1$*, i*

$$\frac{\mathbf{a}_i^\top P \mathbf{a}_i}{\mathbf{a}_i^\top G \mathbf{a}_i} = \max\left\{ \frac{\mathbf{a}^\top P \mathbf{a}}{\mathbf{a}^\top G \mathbf{a}} \colon \mathbf{a}^\top G \mathbf{a}_j = 0,\ j \leq i-1 \right\},$$

ako i samo ako zadovoljavaju uvjet

(II) *za svaki* $i = 1, \ldots, q$*,* $(\mathbf{a}_i, \phi_i)$ *je svojstveni par od* $G^{-1}P$*, pri čemu su* $\phi_1 \geq \phi_2 \geq \ldots \geq \phi_q$ *i vrijedi* $\mathbf{a}_i^\top G \mathbf{a}_j = 0$ *za* $i \neq j$*.*

Neka je $Y = (Y_1, \ldots, Y_q)^\top$ q-dimenzionalni slučajni vektor. Želimo naći nove koordinate

$$\mathbf{a}_1^\top Y, \mathbf{a}_2^\top Y, \ldots, \mathbf{a}_q^\top Y$$

tako da imaju neka određena svojstva.

Pretpostavimo $\mathbb{E}Y = \boldsymbol{\mu}$, $\operatorname{cov}(Y) = \Sigma > 0$. Koordinatni vektori $\mathbf{a}_1, \ldots, \mathbf{a}_q$ biraju se tako da su ortogonalni u odnosu na skalarni produkt $\langle \mathbf{a}, \mathbf{b} \rangle := \mathbf{b}^\top \Sigma \mathbf{a} = (\Sigma \mathbf{a}, \mathbf{b})$, tj.

$$\mathbf{a}_i^\top \Sigma \mathbf{a}_j = 0, \ i \neq j.$$

Odavde za $i \neq j$ slijedi

$$\operatorname{cov}\left(\mathbf{a}_i^\top Y, \mathbf{a}_j^\top Y\right) = \mathbf{a}_i^\top \operatorname{cov}(Y, Y) \mathbf{a}_j = \mathbf{a}_i^\top \Sigma \mathbf{a}_j = 0,$$

tj. nove komponente nisu korelirane. Nadalje, koordinate se biraju tako da sekvencijalno daju optimalnu predikciju od Y (uz dani uvjet ortogonalnosti). Prema tome, $\mathbf{a}_1$ se bira tako da

$$\|Y - P[Y \mid \mathbf{a}_1^\top Y]\| = \min_{\mathbf{a} \neq \mathbf{0}} \|Y - P[Y \mid \mathbf{a}^\top Y]\|,$$

a za $i > 1$, $\mathbf{a}_i$ se bira tako da $\mathbf{a}_i^\top \Sigma \mathbf{a}_j = 0$ za $j = 1, 2, \ldots, i-1$ te

$$\|Y - P[Y \mid \mathbf{a}_i^\top Y]\| = \min\{\|Y - P[Y \mid \mathbf{a}^\top Y]\| \colon \mathbf{a} \in \mathbb{R}^q \backslash \{\mathbf{0}\}, \ \mathbf{a}^\top \Sigma \mathbf{a}_j = 0, \ j < i\}.$$

Može se pokazati kao u [1] da je minimizacija gornje funkcije ekvivalentna maksimizaciji funkcije

$$\mathbf{a} \mapsto \frac{\mathbf{a}^\top \Sigma^2 \mathbf{a}}{\mathbf{a}^\top \Sigma \mathbf{a}},$$

tj. treba naći vektore $\mathbf{a}_1, \ldots, \mathbf{a}_q \in \mathbb{R}^q$ takve da

(i) $\dfrac{\mathbf{a}_1^\top \Sigma^2 \mathbf{a}_1}{\mathbf{a}_1^\top \Sigma \mathbf{a}_1} = \max\limits_{\mathbf{a} \neq \mathbf{0}} \dfrac{\mathbf{a}^\top \Sigma^2 \mathbf{a}}{\mathbf{a}^\top \Sigma \mathbf{a}},$

(ii) za $i > 1$, $\mathbf{a}_i^\top \Sigma \mathbf{a}_j = 0$, $j = 1, \ldots, i-1$, te

$$\frac{\mathbf{a}_i^\top \Sigma^2 \mathbf{a}_i}{\mathbf{a}_i^\top \Sigma \mathbf{a}_i} = \max\left\{ \frac{\mathbf{a}^\top \Sigma^2 \mathbf{a}}{\mathbf{a}^\top \Sigma \mathbf{a}} \colon \mathbf{a}^\top \Sigma \mathbf{a}_j = 0, \ j = 1, \ldots, i-1 \right\},$$

što je po teoremu 2.6 ekvivalentno tome da su $\mathbf{a}_1, \ldots, \mathbf{a}_q$ svojstveni vektori od Σ koji odgovaraju padajućem nizu svojstvenih vrijednosti $\phi_1 \geq \phi_2 \geq \ldots \geq \phi_q > 0$ i $\mathbf{a}_1, \ldots, \mathbf{a}_q$ su međusobno ortogonalni u odnosu na skalarni produkt $\langle \cdot, \cdot \rangle$.

2.3 Dekompozicija singularnih vrijednosti - SVD

Glavni problem ovog rada će se svesti na tražnje svojstvenih, odnosno singularnih vrijednosti i vektora. Jedan od najpoznatijih načina traženja singularnih vrijednosti i vektora je dekompozicija singularnih vrijednosti. Iskazat ćemo osnovne teoreme koji su bitni za ovaj rad, a njihovi dokazi se mogu pronaći u [16] i [17].

Teorem 2.7. *Neka je A proizvoljna matrica tipa $\mathbb{R}^{m\times n}$. Tada postoje ortogonalna $m\times m$ matrica U, ortogonalna $n\times n$ matrica V i dijagonalna $m\times n$ matrica Σ tako da vrijedi $A = U\Sigma V^\top$, gdje je*

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \text{ ili } \Sigma = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_m & 0 & \cdots & 0 \end{pmatrix}.$$

Dijagonalni elementi matrice Σ su jedinstveno određeni matricom A i uređeni su tako da je

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p > \sigma_{p+1} = \cdots = \sigma_{min(m,n)} = 0,$$

gdje je $p = r(A)$ rang matrice A. Kažemo da je $A = U\Sigma V^\top$ singularna dekompozicija matrice (SVD) A.

Stupce matrice U (oznaka u_i) zovemo **lijevi singularni vektori**, stupce matrice V (oznaka v_i) zovemo **desni singularni vektori**, a dijagonalne elemente σ_i matrice Σ **singularne vrijednosti**.

Teorem 2.8. *Ako A ima puni rang, onda je rješenje problema najmanjih kvadrata*

$$\min_x \|Ax - b\|_2$$

jednako $x = V\Sigma^{-1}U^\top b$.

3 Metoda parcijalnih najmanjih kvadrata - PLS

Cilj PLS metode je kreirati ortogonalne vektore maksimizirajući kovarijancu između dva različita skupa.

Neka su dane dvije centrirane matrične varijable $\mathbf{X}, \mathbf{Y}$, tj. očekivanja su im nula. Pretpostavimo da je $\mathbf{X} \in \mathbb{R}^{n\times p}$ te da je $\mathbf{Y} \in \mathbb{R}^{n\times q}$, gdje su p i q jako veliki brojevi koji predstavljaju broj varijabli u svakom skupu, a n predstavlja veličinu uzorka. Odnosno pretpostavljamo da je $\mathbf{X} = \begin{bmatrix} X_{\cdot 1} & X_{\cdot 2} & \dots & X_{\cdot p} \end{bmatrix}$ slučajna matrica reda (n, p), čiji je svaki redak $X_{i\cdot}^\top$ slučajan uzorak duljine p, a $X_{i\cdot}$ su nezavisno jednako distribuirane, $\forall i = 1, \dots, n$. Uočimo da se onda matrica $\mathbf{X}$ može zapisati kao

$$\mathbf{X} = \begin{bmatrix} X_{\cdot 1} & X_{\cdot 2} & \dots & X_{\cdot p} \end{bmatrix} = \begin{bmatrix} X_{1\cdot}^\top \\ X_{2\cdot}^\top \\ \vdots \\ X_{n\cdot}^\top \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1p} \\ X_{21} & X_{22} & \dots & X_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{np} \end{bmatrix}.$$

Također, pretpostavljamo da je $\mathbf{Y} = \begin{bmatrix} Y_{\cdot 1} & Y_{\cdot 2} & \dots & Y_{\cdot q} \end{bmatrix}$ slučajna matrica reda (n, q), čiji je svaki redak $Y_{j\cdot}^\top$ slučajan uzorak duljine q, a $Y_{j\cdot}$ su nezavisno jednako distribuirane, $\forall j = 1, \dots, n$. Slično kao za $\mathbf{X}$, $\mathbf{Y}$ možemo zapisati kao:

$$\mathbf{Y} = \begin{bmatrix} Y_{\cdot 1} & Y_{\cdot 2} & \dots & Y_{\cdot q} \end{bmatrix} = \begin{bmatrix} Y_{1\cdot}^\top \\ Y_{2\cdot}^\top \\ \vdots \\ Y_{n\cdot}^\top \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} & \dots & Y_{1q} \\ Y_{21} & Y_{22} & \dots & Y_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ Y_{n1} & Y_{n2} & \dots & Y_{nq} \end{bmatrix}.$$

Neka je $X_{\cdot i}$ i-ti stupac od matrice $\mathbf{X}$. On je slučajan uzorak duljine n za i-tu komponentu mjerenog obilježja X, koji je p dimenzionalan slučajni vektor. Dodatno, neka je $X_{1\cdot}, \dots, X_{n\cdot}$ slučajan uzorak za X. Neka je $Y_{\cdot j}$ j-ti stupac od matrice $\mathbf{Y}$, on je slučajan uzorak duljine n za j-tu komponentu mjerenog obilježja Y, koji je q dimenzionalan slučajni vektor. Dodatno, neka je $Y_{1\cdot}, \dots, Y_{n\cdot}$ slučajan uzorak za Y. Iz početnih pretpostavki slijedi da je $\mu_X = \mathbb{E}X = 0$, $\mu_Y = \mathbb{E}Y = 0$. Označimo sa

$$V_{YX} = \operatorname{cov}(Y, X) = \mathbb{E}[(Y - \mu_Y)(X - \mu_X)^\top] = \mathbb{E}[YX^\top]$$

$$V_{XY} = \operatorname{cov}(X, Y) = \mathbb{E}[(X - \mu_X)(Y - \mu_Y)^\top] = \mathbb{E}[XY^\top] = V_{YX}^\top.$$

Posebno, definiramo

$$V_{XX} = \operatorname{cov}(X) = \operatorname{cov}(X, X) = \mathbb{E}[XX^\top] = \Sigma_X$$

$$V_{YY} = \text{cov}(Y) = \text{cov}(Y, Y) = \mathbb{E}[YY^\top] = \Sigma_Y.$$

Kako bismo uspjeli zadržati što više upotrebljivih informacija o skupovima $\mathbf{X}$ i $\mathbf{Y}$ u realnom vremenu korisno je stvoriti tako zvane latentne varijable tako da se reducira broj početnih varijabli svakog skupa na manji broj. Ova ideja je prirodna, jer su varijable često jako korelirane. Dakle, želimo kreirati mali broj novih varijabli koje imaju svojstvo da u nekom smislu najbolje predviđaju originalne varijable, ali za oba skupa istovremeno. Analiza parcijalnih najmanjih kvadrata (PLS) pronalazi linearne kombinacije originalnih varijabli koje su najbolji linearni predviditelji (prediktori) svih originalnih varijbli, tako da se kreira matrica čiji su stupci međusobno ortogonalni vektori, koji su dobiveni maksimiziranjem kovarijance između tih različitih skupova varijabli $\mathbf{X}$ i $\mathbf{Y}$. Posebno, uočimo da se $\mathbf{X}$ i $\mathbf{Y}$ mogu prikazati u obliku:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{E} \tag{5}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^\top + \mathbf{F} \tag{6}$$

gdje su $\mathbf{T} \in \mathbb{R}^{n\times l}$ i $\mathbf{U} \in \mathbb{R}^{n\times k}$, $\mathbf{P} \in \mathbb{R}^{p\times l}$ i $\mathbf{Q} \in \mathbb{R}^{q\times k}$ definirane kao i gore, a $\mathbf{E} \in \mathbb{R}^{n\times p}$ i $\mathbf{F} \in \mathbb{R}^{n\times q}$ zovemo matrice residuala.

Želimo odabrati matricu $\mathbf{T} = [T_1, \ldots, T_l]$ tako da su joj stupci međusobno ortogonalni i da se mogu prikazati kao linearna kombinacija varijable $\mathbf{X}$. Svaki stupac T_i možemo prikazati preko originalne matrice $\mathbf{X}$ tako da $\forall i = 1, \ldots, l$ vrijedi

$$T_i = \mathbf{X}w_i.$$

S druge strane, svaki stupac T_i možemo prikazati preko težinskih vektora (w_i). Tada vrijedi da je $\mathbf{T} = \mathbf{X}W$, gdje je $W = [w_1, \ldots, w_l]$.

Na sličan način želimo odabrati i matricu $\mathbf{U} = [U_1, \ldots, U_k]$ tako da su joj stupci ortogonalni i da se mogu prikazati kao linearna kombinacija varijable $\mathbf{Y}$. Svaki stupac U_i možemo prikazati preko originalne matrice $\mathbf{Y}$ tako da $\forall j = 1, \ldots, k$ vrijedi

$$U_j = \mathbf{Y}z_j,$$

odnosno, svaki stupac U_j možemo prikazati preko težinskih vektora (z_j). Tada vrijedi da je $\mathbf{U} = \mathbf{Y}Z$, gdje je $Z = [z_1, \ldots, z_k]$.

Želimo pronaći malen broj linearnih kombinacija komponenti od $\mathbf{X}$ koje imaju sposobnost reproducirati komponente od $\mathbf{X}$, ali i istovremeno reproduciraju što veći broj komponenti od $\mathbf{Y}$. Zbog toga, pomoću PLS metode nalazimo ortogonalnu projekciju za koju je kovarijanca između $\mathbf{X}$ i $\mathbf{Y}$ maksimalna. S obzirom da smo pretpostavili da vrijedi (5) slijedi da je cilj maksimizirati kovarijancu između $\mathbf{T}$ i $\mathbf{U}$, gdje se oni mogu prikazati pomoću

početnih varijabli $\mathbf{X}$ i $\mathbf{Y}$.

Prema pretpostavci $Y = (y_1, \ldots, y_q) \in \{Y_{1.}, \ldots, Y_{n.}\}$ je q-dimenzionalan slučajan vektor i $X = (x_1, \ldots, x_p) \in \{X_{1.}, \ldots, X_{n.}\}$ p-dimenzionalan slučajan vektor. Želimo naći nove koordinate:

$$w_1^\top X, w_2^\top X, \ldots, w_p^\top X$$

$$z_1^\top Y, z_2^\top Y, \ldots, z_q^\top Y$$

tako da imaju neka određena svojstva. Prisjetimo se da prema pretpostavci za X i za Y vrijedi $\mathbb{E}X = \mathbb{E}Y = 0$, $\operatorname{cov}(X) = V_{XX} > 0$ i $\operatorname{cov}(Y) = V_{YY} > 0$. Želimo da su koordinatni vektori $z_1, \ldots, z_q$ ortogonalni u odnosu na skalarni produkt $(z_i, z_j) = z_j^\top V_{YY} z_i$ te želimo da su koordinatni vektori $w_1, \ldots, w_p$ ortogonalni su odnosu na skalarni produkt tako da kovarijanca između transformiranih $\mathbf{X}$ i $\mathbf{Y}$ bude maksimalna.

Neka je $Z = [z_1, \ldots, z_q]$ dana matrica kao što je opisano gore. Tada prema lemi 2.5 postoji Λ_Y dijagonalna matrica tako da vrijedi

$$V_{YY}^{-1} Z = Z\Lambda_Y,\ Z^\top V_{YY} Z = I_q,\ ZZ^\top = V_{YY}^{-1}.$$

Za tako zadanu matricu Z tražimo $w_1, \ldots, w_p$ tako da vrijedi

$$w_1 = \arg\max_{w \in \mathbb{R}^n} F(w)$$
$$w_i \perp w_1, \ldots, w_{i-1},\ w_i = \arg\max_{w \perp [w_1, \ldots, w_{i-1}]} F(w),\ \forall i = 2, \ldots, p$$

gdje je funkcija F dana s

$$\begin{aligned} F(w) &= \sum_{i=1}^{q} \left(\operatorname{cov}(X^\top w, Y^\top z_i) \right)^2 = \sum_{i=1}^{q} \left(w^\top \operatorname{cov}(X^\top, Y^\top) z_i \right)^2 \\ &= \sum_{i=1}^{q} \left(w^\top V_{XY} z_i \right)^2 = \sum_{i=1}^{q} \left(w^\top V_{XY} z_i z_i^\top V_{YX} w \right) \\ &= w^\top V_{XY} V_{YY}^{-1} V_{YX} w. \end{aligned}$$

Varijance pojedinih komponenti mogu biti nesumjerljive. Kako bi se to izbjeglo bolje je promatrati standardizirane varijable $\tilde{\mathbf{Y}} = \mathbf{Y}\Sigma_Y^{-\frac{1}{2}}$. Slijedi da za svaki Y iz $\{Y_{1.}, \ldots Y_{n.}\}$, $\tilde{Y} = \Sigma_Y^{-\frac{1}{2}} Y$ vrijedi

$$V_{\tilde{Y}\tilde{Y}} = \operatorname{cov}(\Sigma_Y^{-\frac{1}{2}} Y) = \Sigma_Y^{-\frac{1}{2}} \operatorname{cov}(Y) \Sigma_Y^{-\frac{1}{2}} = \Sigma_Y^{-\frac{1}{2}} \Sigma_Y \Sigma_Y^{-\frac{1}{2}} = I_q.$$

Uočimo da je

$$V_{X\tilde{Y}} = \operatorname{cov}(X, \tilde{Y}) = \operatorname{cov}(X, \Sigma_Y^{-\frac{1}{2}} Y) = \operatorname{cov}(X, Y)\Sigma_Y^{-\frac{1}{2}} = V_{XY}\Sigma_Y^{-\frac{1}{2}},$$

tj. da je $V_{XY} = V_{X\tilde{Y}}\Sigma_Y^{\frac{1}{2}}$. Iz ovoga slijedi da je

$$\begin{aligned} F(w) &= w^\top V_{XY} V_{YY}^{-1} V_{YX} w = w^\top V_{X\tilde{Y}} \Sigma_Y^{\frac{1}{2}} V_{YY}^{-1} \Sigma_Y^{\frac{1}{2}} V_{\tilde{Y}X} w \\ &= w^\top V_{X\tilde{Y}} I_q V_{\tilde{Y}X} w = w^\top V_{X\tilde{Y}} V_{\tilde{Y}X} w. \end{aligned}$$

Prema teoremu 2.2 slijedi da su vektori $w_1, \ldots, w_p$ svojstveni vektori matrice $V_{X\tilde{Y}} V_{\tilde{Y}X}$ koji odgovaraju padajućem nizu svojstvenih vrijednosti

$$\lambda_1^X \geq \lambda_2^X \geq \ldots \lambda_p^X > 0.$$

Analogno, neka je $W = [w_1, \ldots, w_p]$ dana matrica te neka je Λ_X dijagonalna matrica (u skladu s lemom 2.5) tako da vrijedi

$$V_{XX}^{-1} W = W\Lambda_X,\ W^\top V_{XX} W = I_p,\ WW^\top = V_{XX}^{-1}.$$

Tražimo $z_1, \ldots, z_q$ tako da vrijedi

$$\begin{aligned} z_1 &= \arg\max_{z \in \mathbb{R}^n} G(z) \\ z_i \perp z_1, \ldots, z_{j-1},\ z_j &= \arg\max_{z \perp [z_1, \ldots, z_{j-1}]} G(z),\ \forall j = 2, \ldots, q \end{aligned}$$

gdje je funkcija G dana s

$$\begin{aligned} G(z) &= \sum_{j=1}^{p} \left(\operatorname{cov}(Y^\top z, X^\top w_j) \right)^2 = \sum_{j=1}^{p} \left(z^\top \operatorname{cov}(Y^\top, X^\top) w_j \right)^2 \\ &= \sum_{j=1}^{p} \left(z^\top V_{YX} w_j \right)^2 = \sum_{j=1}^{p} \left(z^\top V_{YX} w_j w_j^\top V_{XY} z \right) \\ &= z^\top V_{YX} V_{XX}^{-1} V_{XY} z. \end{aligned}$$

Slično vrijedi, kao i gore, za $\tilde{\mathbf{X}} = \mathbf{X}\Sigma_X^{-\frac{1}{2}}$, slijedi da za svaki X iz $\{X_{1.}, \ldots X_{n.}\}$, $\tilde{X} = \Sigma_X^{-\frac{1}{2}} X$ vrijedi

$$V_{\tilde{X}\tilde{X}} = \operatorname{cov}(\Sigma_X^{-\frac{1}{2}} X) = \Sigma_X^{-\frac{1}{2}} \operatorname{cov}(X) \Sigma_X^{-\frac{1}{2}} = \Sigma_X^{-\frac{1}{2}} \Sigma_X \Sigma_X^{-\frac{1}{2}} = I_p.$$

Uočimo da je

$$V_{\tilde{X}Y} = \operatorname{cov}(\tilde{X}, Y) = \operatorname{cov}(\Sigma_X^{-\frac{1}{2}} X, Y) = \Sigma_X^{-\frac{1}{2}} \operatorname{cov}(X, Y) = \Sigma_X^{-\frac{1}{2}} V_{XY},$$

tj. da je $V_{XY} = \Sigma_X^{\frac{1}{2}} V_{\tilde{Y}X}$. Iz ovoga slijedi da je

$$\begin{aligned} G(z) &= z^\top V_{YX} V_{XX}^{-1} V_{XY} z = z^\top V_{Y\tilde{X}} \Sigma_X^{\frac{1}{2}} V_{XX}^{-1} \Sigma_X^{\frac{1}{2}} V_{\tilde{X}Y} z \\ &= z^\top V_{Y\tilde{X}} I_p V_{\tilde{X}Y} z = z^\top V_{Y\tilde{X}} V_{\tilde{X}Y} z. \end{aligned}$$

Prema teoremu 2.2 slijedi da su vektori $z_1, \ldots, z_q$ svojstveni vektori matrice $V_{Y\tilde{X}} V_{\tilde{X}Y}$ koji odgovaraju padajućem nizu svojstvenih vrijednosti

$$\lambda_1^Y \geq \lambda_2^Y \geq \ldots \lambda_p^Y > 0.$$

Uočimo da u prvom slučaju promatramo realnu funnkciju $\sum_{j=1}^p F(w_j)$ - koja predstavlja zbroj maksimuma, dok je u drugom slučaju $\sum_{i=1}^q G(z_i)$. Iz

$$\begin{aligned} \sum_{j=1}^p F(w_j) &= \sum_{j=1}^p w_j V_{XY} V_{YY}^{-1} V_{YX} w_j = \sum_{j=1}^p \mathrm{tr}(V_{YY}^{-1} V_{YX} w_j w_j^\top V_{XY}) \\ &= \mathrm{tr}(V_{YY}^{-1} V_{YX} V_{XX}^{-1} V_{XY}) = \mathrm{tr}(V_{XY} V_{YY}^{-1} V_{YX} V_{XX}^{-1}) \\ &= \sum_{i=1}^q \mathrm{tr}(V_{XX}^{-1} V_{YX} z_i z_i^\top V_{XY}) = \sum_{i=1}^q z_i V_{YX} V_{XX}^{-1} V_{XY} z_i^\top \\ &= \sum_{i=1}^q G(z_i) \end{aligned}$$

slijedi da je dovoljno promatrati samo jedan slučaj optimizacije, jer oba daju jednako optimalan rezultat.

3.1 Procijenjene vrijednosti

Uočimo da u praksi dosta često nemamo zadane matrice ortogonalnih projekcija i matrice koeficijenata te ih je potrebno procijeniti iz početnih matrica $\mathbf{X}$ i $\mathbf{Y}$. Tada ćemo prikazati naše podatke u obliku:

$$\hat{\mathbf{X}} = \mathbf{T}\mathbf{P}^\top$$
$$\hat{\mathbf{Y}} = \mathbf{U}\mathbf{Q}^\top$$

gdje su $\mathbf{T} \in \mathbb{R}^{n\times l}$ i $\mathbf{U} \in \mathbb{R}^{n\times k}$ ortogonalne projekcije koje zovemo latentne matrice, $\mathbf{P} \in \mathbb{R}^{p\times l}$ i $\mathbf{Q} \in \mathbb{R}^{q\times k}$ predstavljaju matrice koeficijenata.

Analogno, potrebno je procijeniti kovarijacijske matrice kako bismo mogli riješiti dani problem. Prisjetimo se da su uzoračke kovarijance redom dane s

$$\widehat{V_{XX}} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^\top = \frac{1}{n-1}\sum_{i=1}^{n} X_{i.}X_{i.}^\top$$

$$\widehat{V_{YY}} = \frac{1}{n-1}\mathbf{Y}\mathbf{Y}^\top = \frac{1}{n-1}\sum_{i=1}^{n} Y_{i.}Y_{i.}^\top$$

$$\widehat{V_{XY}} = \frac{1}{n-1}\mathbf{X}\mathbf{Y}^\top = \frac{1}{n-1}\sum_{i=1}^{n} X_{i.}Y_{i.}^\top$$

$$\widehat{V_{YX}} = \frac{1}{n-1}\mathbf{Y}\mathbf{X}^\top = \frac{1}{n-1}\sum_{i=1}^{n} Y_{i.}X_{i.}^\top = \widehat{V_{XY}}^\top$$

Tada želimo da su koordinatni vektori $w_1, \ldots, w_p$ ortogonalni s obzirom na skalarni produkt $(w_i, w_j) = w_j^\top \widehat{V_{XX}} w_i = \frac{1}{n-1} w_j^\top \mathbf{X}\mathbf{X}^\top w_i$.

Pretpostavimo da je $\widehat{\Lambda}_Y$ procjenjena dijagonalna matrica Λ_Y i neka je $Z = [z_1, \ldots, z_q]$ dana matrica tako da vrijedi

$$(\mathbf{Y}^\top\mathbf{Y})^{-1}Z = \frac{1}{(n-1)}Z\widehat{\Lambda}_Y,\ Z^\top\mathbf{Y}^\top\mathbf{Y}Z = (n-1)I_q,\ ZZ^\top = (n-1)(\mathbf{Y}^\top\mathbf{Y})^{-1}.$$

Tada se vrijednosti funkcije F u točki w mogu procijeniti funkcijom uzorka

$$\begin{aligned}\widehat{F(w)} &= \sum_{i=1}^{q}\left(\widehat{\mathrm{cov}}(X^\top w, Y^\top z_i)\right)^2 = w^\top \widehat{V_{XY}}\widehat{V_{YY}^{-1}}\widehat{V_{YX}}w \\ &= \frac{1}{n-1}w^\top\mathbf{X}^\top\mathbf{Y}(\mathbf{Y}^\top\mathbf{Y})^{-1}\mathbf{Y}^\top\mathbf{X}w.\end{aligned}$$

Promatrane standardizirane varijable $\tilde{\mathbf{Y}} = \mathbf{Y}\widehat{\Sigma_Y^{-\frac{1}{2}}}$. Slijedi da za svaki Y iz $\{Y_{1.}, \dots Y_{n.}\}$, $\tilde{Y} = \widehat{\Sigma_Y^{-\frac{1}{2}}}Y$ vrijedi

$$\mathbf{X}\tilde{\mathbf{Y}}^\top = (n-1)\widehat{V_{X\tilde{Y}}} = (n-1)\widehat{V_{XY}}\widehat{\Sigma_Y^{-\frac{1}{2}}} = \mathbf{X}\mathbf{Y}^\top\widehat{\Sigma_Y^{-\frac{1}{2}}},$$

tj. da je $\mathbf{X}\mathbf{Y}^\top = \mathbf{X}\tilde{\mathbf{Y}}^\top\widehat{\Sigma_Y^{\frac{1}{2}}}$. Iz ovoga slijedi da je $\widehat{F(w)} = \frac{1}{n-1}w^\top\mathbf{X}^\top\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\top\mathbf{X}w$.

Prema teoremu 2.2 slijedi da su vektori $w_1, \dots, w_p$ svojstveni vektori matrice $\mathbf{X}^\top\tilde{\mathbf{Y}}\tilde{\mathbf{Y}}^\top\mathbf{X}$ koji odgovaraju padajućem nizu svojstvenih vrijednosti $\lambda_1^X \geq \lambda_2^X \geq \dots \lambda_p^X > 0$.

Analogno, želimo da su koordinatni vektori $z_1, \dots, z_q$ ortogonalni s obzirom na skalarni produkt $(z_i, z_j) = z_j^\top\widehat{V_{YY}}z_i = \frac{1}{n-1}z_j^\top\mathbf{Y}\mathbf{Y}^\top z_i$. Pretpostavimo da je $\widehat{\Lambda}_Y$ procijenjena dijagonalna matrica i neka je $W = [w_1, \dots, w_p]$ dana matrica tako da vrijedi

$$(\mathbf{X}^\top\mathbf{X})^{-1}W = \frac{1}{(n-1)}W\widehat{\Lambda}_X,\ W^\top\mathbf{X}^\top\mathbf{X}W = (n-1)I_p,\ WW^\top = (n-1)(\mathbf{X}^\top\mathbf{X})^{-1}.$$

Tada se vrijednosti funkcije G u točki z mogu procijeniti funkcijom uzorka

$$\begin{aligned}\widehat{G(z)} &= \sum_{j=1}^{p}\left(\widehat{\mathrm{cov}}(Y^\top z, X^\top w_j)\right)^2 = z^\top\widehat{V_{YX}}\widehat{V_{XX}^{-1}}\widehat{V_{XY}}z \\ &= \frac{1}{n-1}z^\top\mathbf{Y}^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}z.\end{aligned}$$

Slično vrijedi, kao i gore, za standardizirane matrice $\tilde{\mathbf{X}} = \mathbf{X}\widehat{\Sigma_X^{-\frac{1}{2}}}$, slijedi da za svaki X iz $\{X_{1.}, \dots X_{n.}\}$, $\tilde{X} = \widehat{\Sigma_X^{-\frac{1}{2}}}X$ vrijedi $\widehat{G(z)} = \frac{1}{n-1}z^\top\mathbf{Y}^\top\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top\mathbf{Y}z$.

Prema teoremu 2.2 slijedi da su vektori $z_1, \dots, z_q$ svojstveni vektori matrice $\mathbf{Y}^\top\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top\mathbf{Y}$ koji odgovaraju padajućem nizu svojstvenih vrijednosti $\lambda_1^Y \geq \lambda_2^Y \geq \dots \lambda_p^Y > 0$.

4 Regresijski model - PLSR

Neka su dane dvije centralizirane matrične varijable $\mathbf{X}, \mathbf{Y}$, tj. očekivanje im je nula. Pretpostavimo da je $\mathbf{X} \in \mathbb{R}^{n\times p}$ te da je $\mathbf{Y} \in \mathbb{R}^{n\times q}$, gdje su p i q jako veliki brojevi koji predstavljaju broj varijabli u svakom skupu, a n predstavlja veličinu uzorka.

U prethodnom odjeljku je prikazana PLS metoda pomoću koje možemo prikazati $\mathbf{X}$ i $\mathbf{Y}$ u novim koordinatama. PLS metoda se najčešće koristi kako bi se riješili problemi linearne regresije.

Prisjetimo se da naše matrične varijable možemo prikazati u obliku :

$$\mathbf{X} = \mathbf{T}\mathbf{P}^\top + \mathbf{E}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^\top + \mathbf{F}.$$

PLS metodom pronalazimo $\mathbf{T} = [T_1, \ldots, T_l]$ tako da su $\{T_i\}_{i=1}^l$ dobri linearni prediktori za $\mathbf{Y}$ te pretpostavljamo da postoji linearna ovisnost između vektora T_i i U_j, odnosno da se $\mathbf{T}$ i $\mathbf{U}$ mogu prikazati u obliku

$$\mathbf{U} = \mathbf{T}\mathbf{D} + \mathbf{H} \tag{7}$$

gdje je $\mathbf{D} \in \mathbb{R}^{l\times k}$ matrica koeficijenata i $\mathbf{H}$ predstavlja matricu reziduala.

Prisjetimo se, cilj je prikazati početne matrice u obliku (5) tako da su matrice $\mathbf{T}$ i $\mathbf{U}$ što više korelirane te da se mogu prikazati preko početnih matrica $\mathbf{X}$ i $\mathbf{Y}$.

Cilj je pomoću PLS regresije procijeniti pretpostavljeni linearni odnos $\mathbf{X}$ i $\mathbf{Y}$

$$\mathbf{Y} = \mathbf{X}B + \boldsymbol{\varepsilon},$$

gdje je $\boldsymbol{\varepsilon} \in \mathbb{R}^{n\times q}$ matrica slučajnih grešaka, a matrica $B_{PLS} \in \mathbb{R}^{l\times q}$ predstavlja matricu parametara modela. Dodatno, pretpostavljamo da matrica slučajnih grešaka $\boldsymbol{\varepsilon}$ zadovoljava Gauss-Markovljeve uvjete.

Pokazali smo u lemi 2.4 da je najbolji linearni predviditelj od $\mathbf{Y}$ uz dano $\mathbf{X}$ je $\hat{\mathbf{Y}} \in \mathbb{R}^{n\times q}$

$$\hat{\mathbf{Y}} = P[\mathbf{Y} \mid \mathbf{X}] = \hat{B^\top}\mathbf{X},$$

gdje je $\hat{B}$ rješenje jednadžbe

$$V_{XX}\hat{B} = V_{XY}.$$

U slučaju da je V_{XX} pozitivno definitna, odnosno regularna, onda

$$\hat{B} = V_{XX}^{-1}V_{XY}.$$

Može nastati problem pri računanju B pri velikoj koreliranosti matrica $\mathbf{X}$ i $\mathbf{Y}$.

Uz početne pretpostavke $\mathbf{Y}$ možemo prikazati u obliku nakon eventualnog reduciranja dimenzije

$$\begin{aligned}\mathbf{Y} &= \mathbf{U}\mathbf{Q}^\top + \mathbf{F} \\ &= (\mathbf{T}\mathbf{D} + \mathbf{H})\mathbf{Q}^\top + \mathbf{F} \\ &= \mathbf{T}\mathbf{D}\mathbf{Q}^\top + (\mathbf{H}\mathbf{Q}^\top + \mathbf{F}) \\ &= \mathbf{T}\mathbf{C}^\top + \mathbf{F}^*\end{aligned}$$

gdje je $\mathbf{C} = \mathbf{Q}\mathbf{D}^\top \in \mathbb{R}^{q\times l}$, a $\mathbf{F}^* = \mathbf{H}\mathbf{Q}^\top + \mathbf{F} \in \mathbb{R}^{n\times q}$ predstavlja matricu residuala. Uočimo da je na ovako prikazan način dovoljno izračunati matricu $\mathbf{T}$, odnosno vrijedi (uz eventualno reduciranje dimenzije od $\mathbf{X}$)

$$\begin{aligned}\mathbf{X} &= \mathbf{T}\mathbf{P}^\top + \mathbf{E} \\ \mathbf{Y} &= \mathbf{T}\mathbf{C}^\top + \mathbf{F}^*.\end{aligned}$$

4.1 Procjena parametara

Kod PLS regresije promatramo slučaj PLS metode, gdje prvo reduciramo dimenziju matrične varijable $\mathbf{Y}$, pa u ovisnosti o $\mathbf{Y}$ reduciramo matričnu varijablu $\mathbf{X}$ na l glavnih komponenti tako da su transformacije od $\mathbf{X}$ i $\mathbf{Y}$ maksimalno korelirane. Nakon toga radimo regresijski model s transformiranim $\mathbf{X}$.

U prethodnom poglavlju smo pokazali da se matrična varijabla $\mathbf{X}$ može PLS metodom procijeniti pomoću latentne matrice $\mathbf{T}$ i matrice koeficijenata $\mathbf{P}$ u obliku

$$\hat{\mathbf{X}} = \mathbf{T}\mathbf{P}^\top.$$

Pomoću PLS metode ćemo dobiti $\mathbf{T} = [T_1, \ldots, T_l]$, gdje T_i predstavljaju linearnu kombinaciju matrične varijable $\mathbf{X}$. Danom metodom ćemo dobiti w_i ($W = [w_1, \ldots, w_l]$), računajući svojstvene vektore od $\mathbf{X}^\top\mathbf{Y}\mathbf{Y}^\top\mathbf{X}$ i P_i ($\mathbf{P} = [P_1, \ldots, P_l]$) koji se može dobiti kao regresijska matrica koeficijenata matrične varijable $\mathbf{X}$ na T_i, u obliku

$$\begin{aligned}\mathbf{P}^\top &= (\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{X}\\ \mathbf{P} &= \mathbf{X}^\top\mathbf{T}(\mathbf{T}^\top\mathbf{T})^{-1}.\end{aligned}$$

U članku [8] je definirana matrica $\mathbf{R}$ kao

$$\mathbf{R} = W(\mathbf{P}^\top W)^{-1}, \tag{8}$$

te je pokazano da vrijedi $\mathbf{T} = \mathbf{X}\mathbf{R}$ i da stupci od W i $\mathbf{R}$ razapinju isti prostor.

Uočimo da vrijedi

$$\begin{aligned}\mathbf{R}^\top\mathbf{P} &= \mathbf{R}^\top\mathbf{X}^\top\mathbf{T}(\mathbf{T}^\top\mathbf{T})^{-1}\\ &= \mathbf{T}^\top\mathbf{T}(\mathbf{T}^\top\mathbf{T})^{-1}\\ &= I_l.\end{aligned}$$

Dodatno vrijedi $\mathbf{R}^\top\mathbf{P}W = W$ i da je $\mathbf{R}\mathbf{P}^\top$ matrica ranga l:

$$r(\mathbf{R}\mathbf{P}^\top) = r(\mathbf{P}^\top\mathbf{R}) = r(\mathbf{R}^\top\mathbf{P}) = r(I_l) = l.$$

Također uočimo da vrijedi sljedeće

$$\begin{aligned}\mathbf{R}\mathbf{P}^\top\mathbf{R}\mathbf{P}^\top &= W(\mathbf{P}^\top W)^{-1}\mathbf{P}^\top W(\mathbf{P}^\top W)^{-1}\mathbf{P}^\top\\ &= W(\mathbf{P}^\top W)^{-1}\mathbf{P}^\top\\ &= \mathbf{R}\mathbf{P}^\top,\end{aligned}$$

odnosno vrijedi $(\mathbf{R}\mathbf{P}^\top)^2 = \mathbf{R}\mathbf{P}^\top$.

Ovime smo pokazali da je $\mathbf{R}\mathbf{P}^\top$ projektor na potrostor $L(W)$.

S druge strane, kod PCA regresije koristeći metodu glavnih komponenti nad matricom $\mathbf{X}$, $\mathbf{X}$ možemo procijeniti u obliku

$$\hat{\mathbf{X}} = \mathbf{T}\mathbf{P}^\top,$$

gdje stupci matrice $\mathbf{T} = [T_1, \ldots, T_l]$ predstavljaju nove koordinate, a matrica $\mathbf{P} = [P_1, \ldots, P_l]$ predstavlja matricu koeficijenata. Tada radimo regresijski model s transformiranim $\mathbf{X}$.

Neka je dan slučajan uzorak $Y_{1.}, \ldots, Y_{n.}$. Želimo prikazati matricu $\mathbf{Y}$ pomoću matrice $\mathbf{X}$. Pretpostavljamo da se ona može opisati pomoću linearnog modela

$$\mathbf{Y} = \mathbf{X}B + \boldsymbol{\varepsilon},$$

gdje je $\boldsymbol{\varepsilon} \in \mathbb{R}^{n\times q}$ matrica slučajnih grešaka, a matrica $B \in \mathbb{R}^{l\times q}$ predstavlja matricu parametara modela. Dodatno, pretpostavljamo da matrica slučajnih grešaka $\boldsymbol{\varepsilon}$ zadovoljava Gauss-Markovljeve uvjete.

Iz leme 2.4 slijedi da je najbolji linearni predviditelj od $\mathbf{Y}$ uz dano $\mathbf{X}$

$$\hat{B}_{OLS} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}.$$

Uz redukciju dimenzije od $\mathbf{X}$ nekom od gore navedenih metoda vrijedi,

$$\begin{aligned}\mathbf{Y} &= \mathbf{X}B + \boldsymbol{\varepsilon}\\ &= \mathbf{T}\mathbf{P}^\top B + \boldsymbol{\varepsilon}^*\\ &= \mathbf{T}A + \boldsymbol{\varepsilon}^*\end{aligned}$$

iz čega ćemo procijeniti parametar B, gdje je $\boldsymbol{\varepsilon}^* = \mathbf{E}B + \boldsymbol{\varepsilon} \in \mathbb{R}^{n\times q}$ matrica slučajnih grešaka. Iz ovoga slijedi da vrijedi $A = \mathbf{P}^\top B$, gdje je A matrica parametara dobivena metodom najmanjih kvadrata.

Primjenom gornjih tvrdnji slijedi da je procijenjeni parametar A dan s

$$\hat{A} = (\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{Y}$$

$$\hat{A} = \mathbf{P}^\top\hat{B}.$$

Izjednačavanjem lijeve i desne strane vidimo da vrijedi

$$\mathbf{P}^\top\hat{B} = (\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{Y}. \tag{9}$$

Prilikom PLS regresije, gdje prvo reduciramo dimenziju od $\mathbf{Y}$, pa onda reduciramo dimenziju od $\mathbf{X}$ tako da kovarijanca od transformiranih $\mathbf{X}$ i $\mathbf{Y}$ bude maksimalna, možemo jednadžbu (9) pomnožiti s matricom $\mathbf{R}$ definiranom kao u (8) te ćemo dobiti

$$\mathbf{R}\mathbf{P}^\top \hat{B} = \mathbf{R}(\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{Y}.$$

S obzirom da smo pokazali da je $\mathbf{R}\mathbf{P}^\top$ projektor na potprostor $L(W)$, gdje su retci W dobiveni kao svojstveni vektori od $\mathbf{X}^\top\mathbf{Y}\mathbf{Y}^\top\mathbf{X}$, slijedi da vrijedi

$$\begin{aligned}\hat{B}^l_{PLS} &= \mathbf{R}(\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{Y} \\ &= W(\mathbf{P}^\top W)^{-1}(\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{Y}.\end{aligned}$$

S druge strane, kod PCA regresije, gdje reduciramo samo dimenziju od $\mathbf{X}$, možemo jednadžbu (9) pomnožiti s matricom $\mathbf{P}(\mathbf{P}^\top\mathbf{P})^{-1}$, gdje je $\mathbf{P}$ dobiven PCA metodom nad $\mathbf{X}$, te ćemo dobiti

$$\mathbf{P}(\mathbf{P}^\top\mathbf{P})^{-1}\mathbf{P}^\top\hat{B} = \mathbf{P}(\mathbf{P}^\top\mathbf{P})^{-1}(\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{Y}.$$

Analogno kao i za $\mathbf{R}\mathbf{P}^\top$, pokaže se da je $\mathbf{P}(\mathbf{P}^\top\mathbf{P})^{-1}\mathbf{P}^\top$ projektor na potprostor $L(W)$, gdje su retci W dobiveni kao svojstveni vektori od $\mathbf{X}^\top\mathbf{X}$. Slijedi da vrijedi

$$\hat{B}^l_{PCA} = \mathbf{P}(\mathbf{P}^\top\mathbf{P})^{-1}(\mathbf{T}^\top\mathbf{T})^{-1}\mathbf{T}^\top\mathbf{Y}$$

Razlika između $\hat{B}^l_{PLS}$ i $\hat{B}^l_{PCA}$, je ta da se kod PCA od $\mathbf{X}$ neovisno o $\mathbf{Y}$ umjesto matrice W u procijenjenom parametru $\hat{B}_{PCA}$ pojavljuje $\mathbf{P}$. Valja napomenuti da su u slučaju metode glavnih komponenti od $\mathbf{X}$ matrice W i $\mathbf{P}$ jednake. Kod PLS metode, nakon što reduciramo dimenziju od $\mathbf{Y}$, tražimo nove komponente pomoću kojih ćemo opisati $\mathbf{X}$, ali tako da su maksimalno korelirane s transformacijama od Y, pa se zato u procijenjenom parametru $\hat{B}_{PLS}$ pojavljuje matrica W.

4.2 Konzistentnost procijenjenih parametara

Neka vrijede sljedeće pretpostavke.

Pretpostavka 4.1. *Neka postoje svojstveni vektori $v_1, \dots, v_l$ od V_{XX} pridružene različitim svojstvenim vrijednostima $\lambda_1, \dots, \lambda_l$ tako da vrijedi $V_{XY} = \sum_{j=1}^{l} \alpha_j v_j$, gdje su $\alpha_1, \dots, \alpha_l$ matrice koeficijenata različitih od nule.*

Pretpostavka 4.2. *Neka se svaki redak od $\mathbf{X}$, gdje je $\mathbf{X}^\top = (X_{1.}, \dots, X_{n.})$ može prikazati kao $X_{i.} = \sum_{j=1}^{m} {u_i}^j \rho^j + \sigma_1 z_i$, za neku konstantu σ_1, za $\forall i = 1, \dots, n$, gdje vrijedi:*

- *$\rho^j, j = 1, \dots, m \leq p$ su međusobno ortogonalni vektori dobiveni glavnim komponentama s normama $|\rho^1| \geq |\rho^2| \geq \dots \geq |\rho^m|$*
- *multiplikatori ${u_i}^j \sim N(0,1)$ su međusobno nezavisne jednako distribuirane slučajne varijable, bez obzira na indekse i i j*
- *vektori bijelog šuma $z_i \sim N(0, I_p)$ su međusobno nezavisni te su nezavisni od $\{{u_i}^j\}$*
- *$p(n)$, $m(n)$ i $\{\rho^j = \rho^j(n), j = 1, \dots, m\}$ su funkcije koje ovise o n, a norme vektora dobivenih metodom glavnih komponenti konvergiraju kao niz $\rho(n) = (\|\rho^1(n)\|, \dots, \|\rho^j(n)\|, \dots) \to \rho = (\rho_1, \dots, \rho_j, \dots)$.*

Pretpostavka 4.3. *Neka je dana veza između Y i X tako da je $Y = XB + \sigma_2 e$, gdje je $e \sim N(0, I_n)$, $\|B\|_2 < \infty$ i σ_2 je konstanta.*

Lema 4.4. *Ako vrijede gornje pretpostavke, te ako vrijedi da $p/n \to 0, n \to \infty$, tada:*

- $\|\widehat{V_{XX}} - V_{XX}\|_2 = O_p(\sqrt{p/n})$
- $\|\widehat{V_{XY}} - V_{XY}\|_2 = O_p(\sqrt{p/n})$

Lema 4.5. *Ako vrijede gornje pretpostavke te ako vrijedi da $p/n \to 0, n \to \infty$, tada $\forall k \in \mathbb{N}$:*

- $\|\widehat{V_{XX}}^k \widehat{V_{XY}} - V_{XX}^k V_{XY}\|_2 = O_p(\sqrt{p/n})$
- $\|\widehat{V_{XY}}^\top \widehat{V_{XX}}^k \widehat{V_{XY}} - V_{XY}^\top V_{XX}^k V_{XY}\|_2 = O_p(\sqrt{p/n})$

Teorem 4.6. *Vrijedi:*

(a) ako $p/n \to 0, n \to \infty$, *tada* $\|\hat{B}_{PLS} - B\|_2 \to 0$ *po vjerojatnosti*

(b) ako $p/n \to c, c > 0, n \to \infty$, *tada* $\exists d > 0$ *tako da* $\|\hat{B}_{PLS} - B\|_2 \to d$ *po vjerojatnosti.*

Dokaz. Dokaz slijedi iz članka [8].
Dokažimo prvo tvrdnju pod (a).
Prema članku vrijedi $\hat{B}_{PLS} = \hat{R}(\hat{R}^\top \widehat{V_{XX}} \hat{R})^{-1} \hat{R}^\top \widehat{V_{XY}}$, gdje je
$\hat{R} = (\widehat{V_{XY}}, \ldots, \widehat{V_{XX}}^{l-1} \widehat{V_{XY}})$.
Prvo dokažimo da $\hat{B}_{PLS} \to R(R^\top V_{XX} R)^{-1} R^\top V_{XY}$ po vjerojatnosti. Koristeći nejednakost trokuta i Hölderovu nejednakost dobijemo da je

$$\begin{aligned}
&\|\hat{R}(\hat{R}^\top \widehat{V_{XX}} \hat{R})^{-1} \hat{R}^\top \widehat{V_{XY}} - R(R^\top V_{XX} R)^{-1} R^\top V_{XY}\|_2 \leq \\
&\leq \|\hat{R} - R\|_2 \|(\hat{R}^\top \widehat{V_{XX}} \hat{R})^{-1} \hat{R}^\top \widehat{V_{XY}}\|_2 + \|R\|_2 \|(\hat{R}^\top \widehat{V_{XX}} \hat{R})^{-1} \\
&- (R^\top V_{XX} R)^{-1} R\|_2 \|\hat{R}^\top \widehat{V_{XY}}\|_2 + \|R\|_2 \|(R^\top V_{XX} R)^{-1}\|_2 \|\hat{R}^\top \widehat{V_{XY}} - R^\top V_{XY}\|_2
\end{aligned} \tag{10}$$

Slijedi da treba pokazati da $\|\hat{R} - R\|_2 \to 0$, $\|(\hat{R}^\top \widehat{V_{XX}} \hat{R})^{-1} - (R^\top V_{XX} R)^{-1} R\|_2 \to 0$ i $\|\hat{R}^\top \widehat{V_{XY}} - R^\top V_{XY}\|_2 \to 0$.

Prva tvrdnja slijedi iz matrične norme, definicije R te leme 4.5, odnosno vrijedi

$$\|\hat{R} - R\|_2 \leq \sqrt{l} \max_{1 \leq k \leq l} \|\widehat{V_{XX}}^{k-1} \widehat{V_{XY}} - V_{XX}^{k-1} V_{XY}\|_2 \leq \epsilon.$$

Druga tvrdnja slijedi iz nejednakosti

$$\|(A + E)^{-1} - A^{-1}\|_2 \leq \|E\|_2 \|A^{-1}\|_2 \|(A + E)^{-1}\|_2$$

za bilo koje matrice A i E istog reda. Iz navedene nejednakosti slijedi

$$\begin{aligned}
\|(\hat{R}^\top \widehat{V_{XX}} \hat{R})^{-1} &- (R^\top V_{XX} R)^{-1}\|_2 \leq \\
&\leq \|\hat{R}^\top \widehat{V_{XX}} \hat{R} - R^\top V_{XX} R\|_2 \|\hat{R}^\top \widehat{V_{XX}} \hat{R}\|_2 \|R^\top V_{XX} R\|_2 \\
&\leq \Big(\|\hat{R}^\top - R^\top\|_2 \|\widehat{V_{XX}} \hat{R}\|_2 + \|R^\top\|_2 \|V_{XX}\|_2 \|\hat{R} - R\|_2 + \\
&+ \|R^\top\|_2 \|\widehat{V_{XX}} - V_{XX}\|_2 \|R\|_2 \Big) \|\hat{R}^\top \widehat{V_{XX}} \hat{R}\|_2 \|R^\top V_{XX} R\|_2
\end{aligned}$$

Zbog nesingularnosti matrica $\hat{R}^\top \widehat{V_{XX}} \hat{R}$ i $R^\top V_{XX} R$ slijedi da su im norme konačne. Sada tvrdnja slijedi direktno iz leme 4.4 i 4.5.

Za treću tvrdnju je potrebno primijeniti nejednakost trokuta i Hölderovu nejednakost, iz čega dobijemo.

$$\|\hat{R}^\top \widehat{V_{XY}} - R^\top V_{XY}\|_2 \leq \|\hat{R}^\top - R^\top\|_2 \|\widehat{V_{XY}}\|_2 \|R^\top\|_2 \|\widehat{V_{XY}} - V_{XY}\|_2$$

Sada primjenimo prvu tvrdnju i lemu 4.4, pa tvrdnja slijedi drektno.

Za kraj primjenimo da je $B = V_{XX}^{-1} V_{XY} = R(R^\top V_{XX} R)^{-1} R^\top V_{XY}$, čime je dokazan (a) dio teorema.

Još preostaje pokazati (b) tvrdnju teorema.
Pretpostavimo suprotno da $\|\hat{B}_{PLS} - B\| \to 0$ po vjerojatnosti. Prisjetimo se da vrijedi

$$\widehat{V_{XX}} \hat{B_{PLS}} - \widehat{V_{XY}} = 0$$

$$V_{XX} B - V_{XY} = 0.$$

Oduzimanjem godnjih jednadžbi dobijemo da vrijedi

$$0 = \widehat{V_{XX}}(\hat{B_{PLS}} - B) + (\widehat{V_{XX}} - V_{XX})B + (V_{XY} - \widehat{V_{XY}}).$$

Kako smo pretpostavili da $\|\hat{B}_{PLS} - B\|_2 \to 0$ po vjerojatnosti, onda slijedi da i $\|(\widehat{V_{XX}} - V_{XX})B + (V_{XY} - \widehat{V_{XY}})\|_2 \to 0$, što je u kontradikciji. ■

Posljedica ovog teorema je da je matrica parametara B procijenjena PLS metodom konzistentna pod jako strogim uvijetima, tj. generalno nije konzistentan procjenitelj za jako velike p i jako male n.

5 Algoritmi

Cilj je pronaći ortogonalne projekcije $\mathbf{T}$ i $\mathbf{U}$ od matričnih (vektorskih) varijabli $\mathbf{X}$ i $\mathbf{Y}$. Pokazali smo u prethodnim poglavljima da je dovoljno pronaći matricu $\mathbf{T}$, odnosno vektore $w_1, \ldots, w_p$ tako da vrijedi

$$w_1 = \arg\max_{w \in \mathbb{R}^n} w^\top \mathbf{X}^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{X} w$$
$$w_i \perp w_1, \ldots, w_{i-1},\ w_i = \arg\max_{w \perp [w_1, \ldots, w_{i-1}]} w^\top \mathbf{X}^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{X} w,\ \forall i = 2, \ldots, p$$

gdje je $\mathbf{W} = [w_1, \ldots, w_p]$.

Prema teoremu 2.2 slijedi da su vektori $w_1, \ldots, w_p$ svojstveni vektori matrice $\mathbf{X}^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{X}$ koji odgovaraju padajućem nizu svojstvenih vrijednosti

$$\lambda_1^X \geq \lambda_2^X \geq \ldots \lambda_p^X > 0.$$

Postoji niz algoritama kojima se može riješiti gornji problem. Svaki od njih se svodi na to da u svakom koraku tražimo svojstveni vektor koji odgovara najvećoj svojstvenoj vrijednosti, kako bismo pronašli vektore koji pripadaju matrici ortogonalnih projekcija $\mathbf{T}$. Tada u svakom koraku iteracije provodimo postupak deflacije kako bismo iz podataka uklonili varijaciju povezanu s procijenjenom komponentom. Algoritam se završava nakon što pronađe sve potrebne komponente.

Za pronalazak svojstvenih vektora se najčešće (pa tako i u R-u) koristi poznata SVD, dok se u mnogim originalnim algoritmima koristi metoda potencija - s obzirom da u svakom koraku tražimo novi svojstveni vektor koji odgovara najvećoj svojstvenoj vrijednosti.

Kao što smo rekli, poznate su mnoge varijante istog algoritma, ali postoje i različiti tipovi algoritama – razlikuju se po odabiru matrice kojoj tražimo svojstvene vektore te izboru matrica nad kojima vršimo postupak deflacije.

Najpoznatiji algoritmi su NIPALS i SIMPLS, a u praksi se često koriste i Kernelov algoritam, Wide Kernelov, bidiagonalni, Krylov PLS1 i mnogi drugi algoritmi.

U ovom radu ćemo obraditi NIPALS i SIMPLS algoritme, a usporedit ćemo ih s Kernelovim i Wide Kernelovim algoritmom.

5.1 NIPALS

NIPALS (Nonlinear Iterative Partial Least Squares) je algoritam kojeg je uveo H. Wold [3]. Koristi se kada imamo velike skupove podataka te kada je potrebno izračunati samo prvih nekoliko komponenti, odnosno svojstvenih vektora. Prilikom svakog koraka iteracije zbog kompjuterske numeričke ograničenosti se gubi ortogonalnost. Iz tog razloga se pri svakom koraku primjenjuje Gram-Schmidtov postupak.

Valja napomenuti da postoje dvije verzije primjene algoritma- PLS1 i PLS2. U slučaju kada imamo univarijantan $\mathbf{Y}$ ove dvije verzije su jednake. U slučaju multivarijantnog $\mathbf{Y}$ ove metode se razlikuju. Imamo mogućnost primijeniti PLS algoritam na svaku varijablu $\mathbf{Y}$ posebno - PLS1, ili možemo primijeniti PLS algoritam na sve varijable od $\mathbf{Y}$ istovrmeno -PLS2. S obzirom da je na neki način PLS1 varijanta PLS2, u ovom radu ćemo samo prikazati algoritam za PLS2.

Algorithm 1: NIPALS algoritam

Data: $\mathbf{X}$, $\mathbf{Y}$
Result: B

$\mathbf{E} = \mathbf{X}$;
$\mathbf{F} = \mathbf{Y}$;
k_0 ; // broj komponenti
/* inicijaliziramo prazne matrice */
$W = ()$;
$\mathbf{T} = ()$;
$\mathbf{P} = ()$;
for $i = 1 : k_0$ **do**
    $\mathbf{S} = \mathbf{E}^\top \mathbf{F}$;
    $[u, s, v] = svd(\mathbf{S})$;
    $w = u(:, 1)$; // prvi lijevi svojstveni vektor
    $t = \mathbf{E}w$;
    $p = \mathbf{E}^\top t / t^\top t$;
    $q = \mathbf{F}^\top t / t^\top t$;
    $\mathbf{E} = \mathbf{E} - tp^\top$;
    $\mathbf{F} = \mathbf{F} - tq^\top$;
    $\mathbf{T} = [\mathbf{T}, t]$;
    $W = [W, w]$;
    $\mathbf{P} = [\mathbf{P}, p]$;
$\mathbf{R} = W(\mathbf{P}^\top W)^{-1}$;
$B = \mathbf{R}(\mathbf{T}^\top \mathbf{T})^{-1} \mathbf{T}^\top \mathbf{Y}$

5.2 SIMPLS

SIMPLS (Statistically Inspired Modification of PLS) je algoritam kojeg je uveo Sijmen de Jong [4]. U većini slučajeva daje jednake rezultate kao i NIPALS, a prednost mu je što je nešto brži i što uistinu maksimizira kovarijancu između $\mathbf{X}$ i $\mathbf{Y}$. Dodatno, za razliku od NIPALS-a koji u svakom koraku iteracije računa novi $\mathbf{X}$, kod SIMPLS-a se uvijek koristi početni $\mathbf{X}$, a postupak deflacije se vrši na matrici $\mathbf{S}$.

Algorithm 2: SIMPLS algoritam

Data: $\mathbf{X}$, $\mathbf{Y}$
Result: B

k_0 ; // broj komponenti
$\mathbf{V} = zeros(p, k_0)$;
/* inicijaliziramo prazne matrice */
$W = ()$;
$\mathbf{W} = ()$;
$\mathbf{Q} = ()$;
$\mathbf{S} = \mathbf{X}^\top \mathbf{Y}$;
for $i = 1 : k_0$ **do**
    $[u, s, v] = svd(\mathbf{S})$;
    $w = u(:, 1)$; // prvi lijevi svojstveni vektor
    $t = \mathbf{X}w$;
    $p = \mathbf{X}^\top t / t^\top t$;
    $q = \mathbf{Y}^\top t / t^\top t$;
    $v = p$;
    if $i > 1$ **then**
        $v = v - (v^\top p)$;
    $\mathbf{S} = \mathbf{S} - v(v^\top \mathbf{S})$;
    $W = [W, w]$;
    $\mathbf{Q} = [\mathbf{Q}, q]$;
    $\mathbf{V}(: i) = v$;
$B = W\mathbf{Q}^\top$

5.3 Usporedba algoritama

U R-u pomoću funkcije *plsr* možemo odrediti koji od algoritama želimo koristiti. Za korištenje je automatski postavljen **Kernelov algoritam** - *kernelpls*. Za razliku od orginalnog NIPALS-a u ovom slučaju izvodimo SVD nad matricom $\mathbf{X}^\top\mathbf{Y}\mathbf{Y}^\top\mathbf{X}$, umjesto nad matricom $\mathbf{X}^\top\mathbf{Y}$. Dodatno, kod ovog algoritma se ne vrši deflacijski postupak nad $\mathbf{Y}$. U usporedbi s originalnim NIPALS algoritmom, jednako je brz i stabilan te daje jednake rezultate. Inače, NIPALS je algoritam koji se u R-u može pozvati putem *plsr* funkcijom - *oscorepls*. U slučajevima kada podaci imaju velik broj varijabli preporuča se **Wide Kernelov algoritam** - *widekernelpls*. U ovom slučaju traženje svojstvenih vrijednosti, odnosno izvođenje SVD-a se vrši nad matricom $\mathbf{X}\mathbf{X}^\top\mathbf{Y}\mathbf{Y}^\top$. Iako je nestabilan, ovaj algoritam daje jednake rezultate kao NIPALS algoritam. Jedan od popularnijih algoritama uz najboznatijeg NIPALS-a je SIMPLS. U slučaju kada promatramo $\mathbf{Y}$ samo s jednom varijablom, SIMPLS algoriritam daje jednake rezultate kao NIPALS i Kernelov algoritam. No, u slučaju kada promatramo multivarijantan $\mathbf{Y}$, algoritmi mogu dati različite rezultate.

6 Primjer

U ovom odjeljku ćemo pokazati primjenu naše metode na podacima *gasoline* preuzetim s [6]. Iz istog izvora je preuzet i dodatni primjer koji je također obrađen u ovom poglavlju.

Primjeri su obrađeni u programskom jeziku R za koji je bilo potrebno instalirati paket *pls*.

U prvom primjeru će $\mathbf{Y}$ predstavljati vektorsku varijablu , a u drugom će $\mathbf{Y}$ predstavljati matričnu varijablu. Na oba primjera ćemo prvo pokazati kako izgleda multilinerana regresija, nakon toga ćemo primijeniti PCR (PCA regresija) i PLSR (PLS regresija) na danim podacima, a za kraj ćemo usporediti dane podatke.

6.1 Univarijatan Y

Dani su nam podaci **NIR** – koji predstavljaju matričnu varijablu $\mathbf{X}$ koja sadrži 60 observacija i 401 komponentu. Matričnu, odnosno u ovom slučaju vektorsku varijablu $\mathbf{Y}$ predstavlja **octane** koji ima 60 observacija.

Općenito, *NIR* (Near infrared spectroscopy) je tehnika mjerenja za mnoge višekomponentne kemijske sustave, uključujući i proizvode od naftnih rafinerija i petrokemija, prehrambenih proizvoda (čaj, voće, mlijeko, vino, rakija, meso, kruh, sir, itd.), lijekova i produkta izgaranja. *Octane* (oktanski broj) je standardna mjera pridružena zrakoplovnom gorivu i nafti. Što je veći oktanski broj, više kompresije gorivo može izdržati prije detoniranja (paljenja).

Cilj je pomoću regresijskih metoda koje su objašnjene u prethodnim poglavljima predvidjeti oktanski broj na temelju mjerenih vrijednosti NIR-a koje su nam poznate.

Prvotno je potrebno učitati podatke i podjeliti naš skup na dva dijela – jedan na kojem ćemo učiti (trenirati) algoritam i jedan na kojem ćemo ga testirati. Kako bi algoritam mogao što više naučiti na danim podacima, dijelimo skup na 80% i 20% podataka.

```
> data(gasoline)
> head(gasoline)
> gasTrain <- gasoline[1:50,]
> gasTest <- gasoline[51:60,]
```

Gdje ispis danog koda izgleda ovako.

```
  octane NIR.900 nm NIR.902 nm NIR.904 nm NIR.906 nm
1  85.30   -0.050193   -0.045903   -0.042187   -0.037177
2  85.25   -0.044227   -0.039602   -0.035673   -0.030911
```

3	88.45	−0.046867	−0.04126	−0.036979	−0.031458
4	83.40	−0.046705	−0.04224	−0.038561	−0.034513
5	87.90	−0.050859	−0.045145	−0.041025	−0.036357
6	85.50	−0.048094	−0.042739	−0.038812	−0.034017

6.1.1 Multilinearna regresija

Prvo ćemo nad danim podacima napraviti multilinearnu regresiju, pomoću funkcije *lm* koja je već implementirana u R-u.

```
> gas.mlr <- lm(octane ~NIR, data = gasTrain)
> summary(gas.mlr)


Call:
lm(formula = octane ~ NIR, data = gasTrain)

Residuals:
ALL 50 residuals are 0: no residual degrees of freedom!

Coefficients: (352 not defined because of singularities)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    136.74         NA      NA       NA
NIR900 nm    -2276.35         NA      NA       NA
NIR902 nm      144.74         NA      NA       NA
NIR904 nm     -144.97         NA      NA       NA
NIR906 nm    -1513.80         NA      NA       NA
NIR908 nm     1446.99         NA      NA       NA
NIR910 nm    -1738.91         NA      NA       NA
NIR912 nm       70.65         NA      NA       NA
NIR914 nm     2251.01         NA      NA       NA
NIR916 nm    -1957.10         NA      NA       NA
NIR918 nm     3469.49         NA      NA       NA
NIR920 nm     -234.08         NA      NA       NA
NIR922 nm     1173.40         NA      NA       NA
NIR924 nm      713.15         NA      NA       NA
NIR926 nm    -9810.39         NA      NA       NA
NIR928 nm     6290.77         NA      NA       NA
NIR930 nm     2505.59         NA      NA       NA
NIR932 nm    -1759.53         NA      NA       NA
NIR934 nm    -3535.56         NA      NA       NA
NIR936 nm     3016.39         NA      NA       NA
```

NIR938 nm	−3668.12	NA	NA	NA
NIR940 nm	−3915.94	NA	NA	NA
NIR942 nm	−1865.34	NA	NA	NA
NIR944 nm	5043.27	NA	NA	NA
NIR946 nm	5012.19	NA	NA	NA
NIR948 nm	366.13	NA	NA	NA
NIR950 nm	6190.04	NA	NA	NA
NIR952 nm	5007.08	NA	NA	NA
NIR954 nm	−4948.98	NA	NA	NA
NIR956 nm	−9088.82	NA	NA	NA
NIR958 nm	−857.98	NA	NA	NA
NIR960 nm	1763.81	NA	NA	NA
NIR962 nm	6683.33	NA	NA	NA
NIR964 nm	−4576.83	NA	NA	NA
NIR966 nm	6813.55	NA	NA	NA
NIR968 nm	−218.69	NA	NA	NA
NIR970 nm	−2716.90	NA	NA	NA
NIR972 nm	−6473.85	NA	NA	NA
NIR974 nm	−545.76	NA	NA	NA
NIR976 nm	1984.39	NA	NA	NA
NIR978 nm	−5877.66	NA	NA	NA
NIR980 nm	−3157.50	NA	NA	NA
NIR982 nm	12960.88	NA	NA	NA
NIR984 nm	15035.18	NA	NA	NA
NIR986 nm	−1908.27	NA	NA	NA
NIR988 nm	−10649.08	NA	NA	NA
NIR990 nm	4534.41	NA	NA	NA
NIR992 nm	−9257.32	NA	NA	NA
NIR994 nm	−10704.91	NA	NA	NA
NIR996 nm	10516.97	NA	NA	NA
NIR998 nm	NA	NA	NA	NA
NIR1000 nm	NA	NA	NA	NA

U gornjem ispisu nisu ispisane sve varijable, jer su vrijednosti koje poprimaju *NA*. Ovo se dogodi zbog toga što matrica $\mathbf{X}^\top\mathbf{X}$ nije regularna. Postoji mogućnost korištenja pseudoinverza – Moore - Pseudo inverz, no i u tom slučaju nećemo dobiti dovoljno zadovoljavajuće rješenje zato što među podacima postoji prevelika koreliranost. Kako bismo na najbolji mogući način odabrali koje komponente će nam ući u regresijski model koristimo PCR i PLSR.

6.1.2 PCR

Pomoću PCA metode tražimo nove koordinate i radimo regresijski model između naših varijabli. Na skupu za treniranje primjenjujemo PCA regresiju (PCR) funkcijom *pcr* koja je već implementirana u R-u. S obzirom da želimo reducirati dimenziju, uzet ćemo u obzir samo 10 novih komponenata koje u najvećoj mjeri opisuju matričnu varijablu **X**. Dodatno na istom skupu ćemo primijeniti i *cross validation*, tako da u svakom koraku nećemo uzeti u obzir jednu observaciju, nego ćemo na njoj testirati izbor novih komponenti. Ovaj princip se inače zove *leave-one-out* (LOO).

```
> gas.pcr <- pcr(octane ~ NIR, data = gasTrain,
    + validation ="LOO", ncomp = 10)
> summary(gas.pcr)


Data:    X dimension: 50 401
         Y dimension: 50 1
Fit method: svdpc
Number of components considered: 10

VALIDATION: RMSEP
Cross-validated using 50 leave-one-out segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps
CV           1.545    1.472    1.483   0.2894   0.2522
adjCV        1.545    1.471    1.482   0.2879   0.2518
       5 comps  6 comps  7 comps  8 comps  9 comps
CV      0.2622   0.2681   0.2386   0.2328   0.2416
adjCV   0.2618   0.2677   0.2373   0.2323   0.2411
       10 comps
CV       0.2423
adjCV    0.2415

TRAINING: % variance explained
        1 comps  2 comps  3 comps  4 comps  5 comps
X         79.86    88.12    93.54    96.54    97.74
octane    16.99    21.36    97.00    97.71    97.73
        6 comps  7 comps  8 comps  9 comps  10 comps
X         98.38    98.75    99.06    99.28     99.42
octane    97.77    98.47    98.54    98.62     98.83
```

Gore je dan ispis priloženog koda. Iz ispisa vidimo da imamo ispise CV koji predstavlja RMSEP (*Root Mean Squared Error of Prediction*) prilikom cross

validacije i ajdCV koji je u našem slučaju predstavlja istu stvar kao i CV, a zapravo označava pristrani RMSEP. Vidimo da tri komponente u velikom postotku objašnjavaju obje varijable. Pogledajmo sada s kojim postatkom svaka komponenta objašnjava naše varijable.

```
> explvar(gas.pcr)
```

```
    Comp 1     Comp 2     Comp 3     Comp 4     Comp 5
79.8586603  8.2639500  5.4171903  3.0034945  1.1963215
    Comp 6     Comp 7     Comp 8     Comp 9    Comp 10
 0.6397503  0.3691514  0.3127762  0.2171267  0.1417888
```

Vidimo da se većina podataka može objasniti s tri odnosno s četiri komponente. Pri određivanju prikladnog broja komponenti koritit ćemo RMSE (Root Mean Square Error) koji je dan formulom

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}.$$

Vidimo da RMSE zapravo računa prosječnu grešku između prave i procijenjene vrijednosti.

Pogledat ćemo kolika je procijenjena greška koja se postiže našim modelom za prvih deset komponenti.

```
> RMSEP(gas.pcr, estimate="train", intercept= F)
```

```
 1 comps   2 comps   3 comps   4 comps   5 comps
  1.3796    1.3427    0.2624    0.2290    0.2283
 6 comps   7 comps   8 comps   9 comps  10 comps
  0.2263    0.1871    0.1832    0.1776    0.1640
```

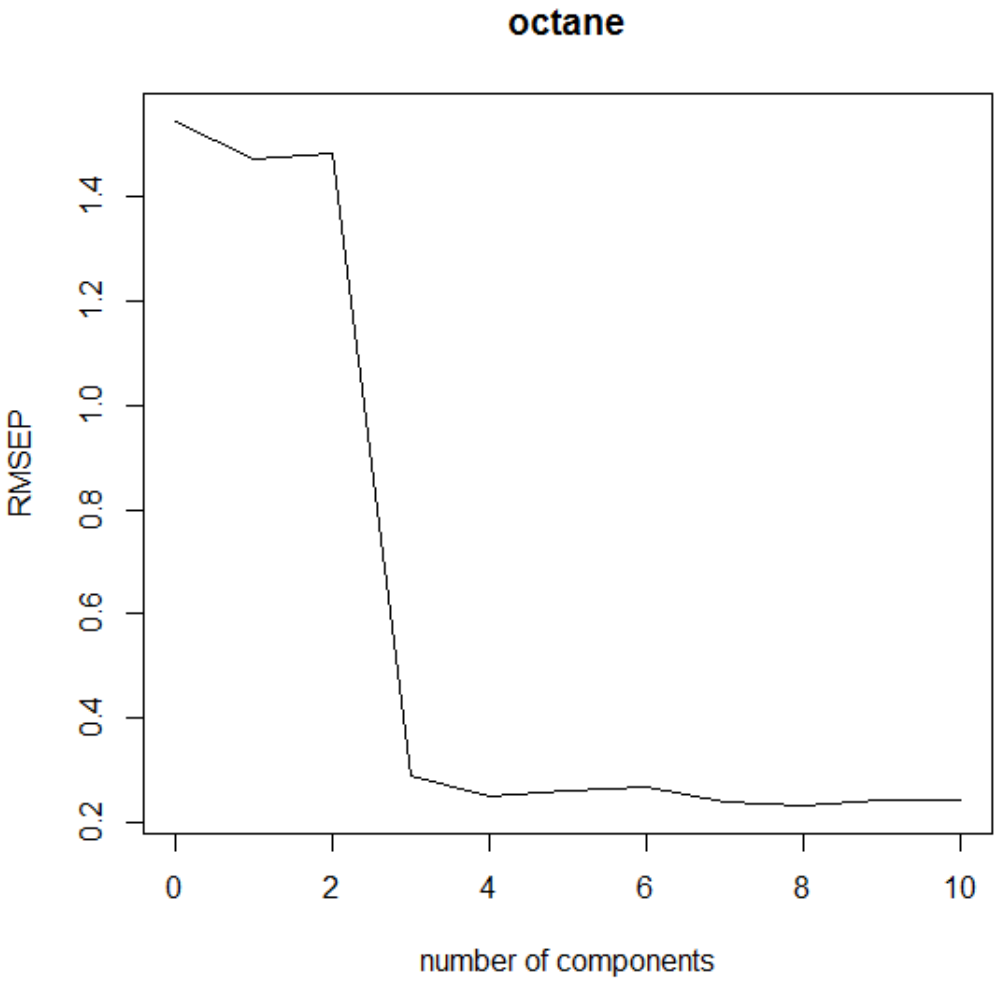
Vidimo da se veliki skok događa kada prelazimo s dvije na tri komponente. Iz toga je i za očekivati da će treća komponenta imati jako mali RMSEP.

```
> RMSEP(gas.pcr, estimate="train", comp=3)
```

```
[1]  0.7474
```

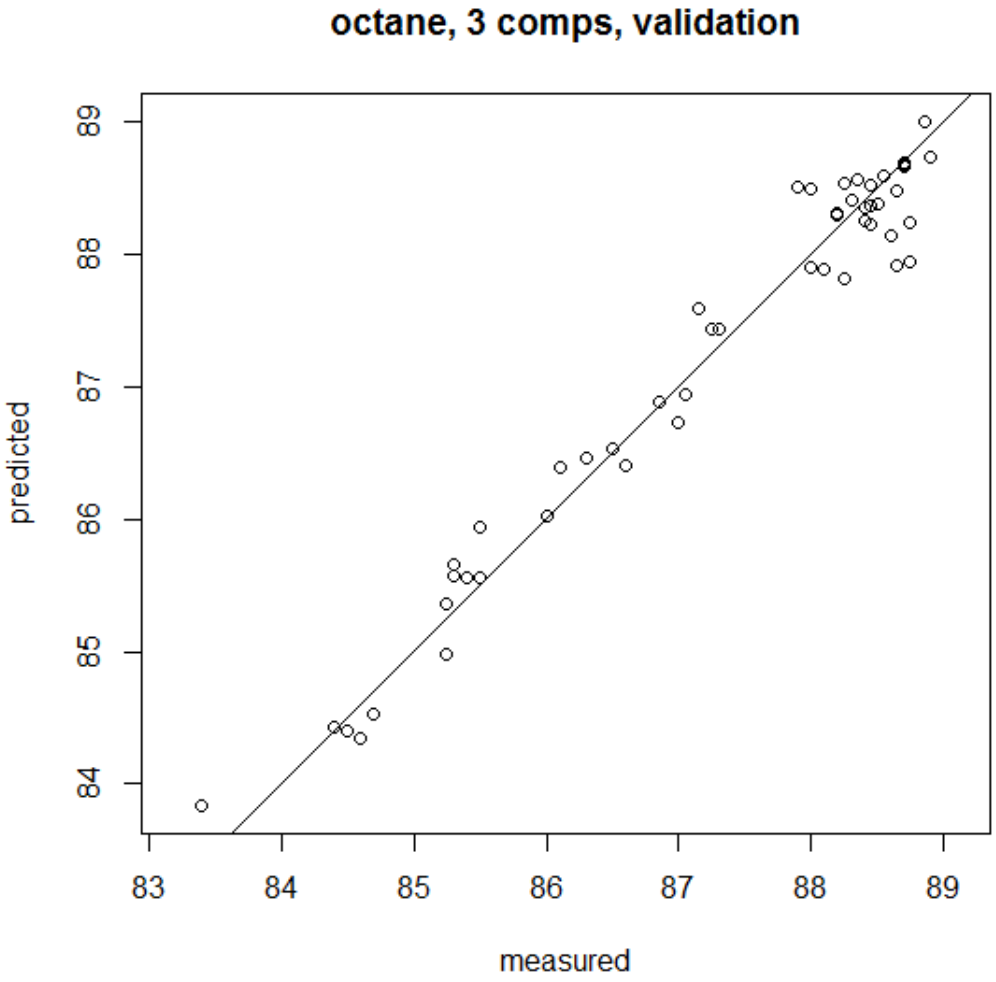
Vidimo da samo treća komponenta ima duplo manji RMSEP nego prve dvije komponente zajedno. Iako se još manji RMSEP postiže s četiri komponente te se ne mijenja značajno nakon toga, iz svih gore navedenih razloga ćemo se odlučiti za model s tri glavne komponente. Našu tvrdnju ćemo dodatno potkrijepiti grafički.

```
> plot(gas.pcr, "validation", estimate = "CV")
```



Nakon što smo procijenili prikladan broj komponenti prikazat ćemo grafički koliko dobro naš model opisuje dane podatke.

```
> plot(gas.pcr, ncomp = 3, asp = 1, line = TRUE)
```



Vidimo da se naši podaci kreću oko danog pravca –identitete, pa možemo zaključiti da naš model s tri glavne komponente dobro opisuje dane podatke.

Sljedeći korak je predviđanje 'nepoznatih' vrijednosti varijable octane.

```
> predict(gas.pcr, ncomp = 3, newdata = gasTest)

, , 3 comps

      octane
51 87.63119
52 87.17090
53 87.84391
54 84.44888
55 84.95272
```

```
56  84.63236
57  86.88466
58  86.50888
59  88.75387
60  86.63756
```

S obzirom da su nam poznate točne vrijednosti varijable *octane*, usporedit ćemo predviđene vrijednosti s originalnim.

```
> RMSEP(gas.pcr, newdata = gasTest)

(Intercept)      1 comps      2 comps      3 comps
     1.5369       1.3226       1.2568       0.4634
    4 comps      5 comps      6 comps      7 comps
     0.2241       0.2283       0.2600       0.2795
    8 comps      9 comps     10 comps
     0.2434       0.2290       0.2881
```

Vidimo da greška predviđanja varijable octane pomoću PCR-a s tri komponente iznosi 0.4634, što je duplo veća greška od one koju smo predviđali.

6.1.3 PLSR

Kao i kod PCR-a, uzet ćemo u obzir samo 10 novih komponenata koje u najvećoj mjeri opisuju obje matrične varijable $\mathbf{X}$ i $\mathbf{Y}$, a primijenjujemo funkciju *plsr* koja je već implementirana u R-u te ćemo kao i kod PCR-a koristili cross validation – LOO.

```
> gas.pls <- plsr( octane ~ NIR, data = gasTrain,
    + validation ="LOO", ncomp = 10)
> summary( gas.pls )

Data:    X dimension: 50 401
         Y dimension: 50 1
Fit method: kernelpls
Number of components considered: 10

VALIDATION: RMSEP
Cross-validated using 50 leave-one-out segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps
CV           1.545    1.357   0.2966   0.2524   0.2476
adjCV        1.545    1.356   0.2947   0.2521   0.2478
       5 comps  6 comps  7 comps  8 comps  9 comps
CV      0.2398   0.2319   0.2386   0.2316   0.2449
```

```
adjCV    0.2388    0.2313    0.2377    0.2308    0.2438
       10 comps
CV       0.2673
adjCV    0.2657

TRAINING: % variance explained
        1 comps  2 comps  3 comps  4 comps  5 comps
X         78.17    85.58    93.41    96.06    96.94
octane    29.39    96.85    97.89    98.26    98.86
        6 comps  7 comps  8 comps  9 comps  10 comps
X         97.89    98.38    98.85    99.02     99.19
octane    98.96    99.09    99.16    99.28     99.39
```

Sljedeći korak je vidjeti u kojem postotku svaka komponenta opisuje matričnu varijablu **X**.

```
> explvar( gas.pls )

Comp 1      Comp 2      Comp 3      Comp 4      Comp 5
78.1707683  7.4122245   7.8241556   2.6577773   0.8768214
    Comp 6      Comp 7      Comp 8      Comp 9      Comp 10
0.9466384  0.4921537  0.4723207   0.1688272   0.1693770
```

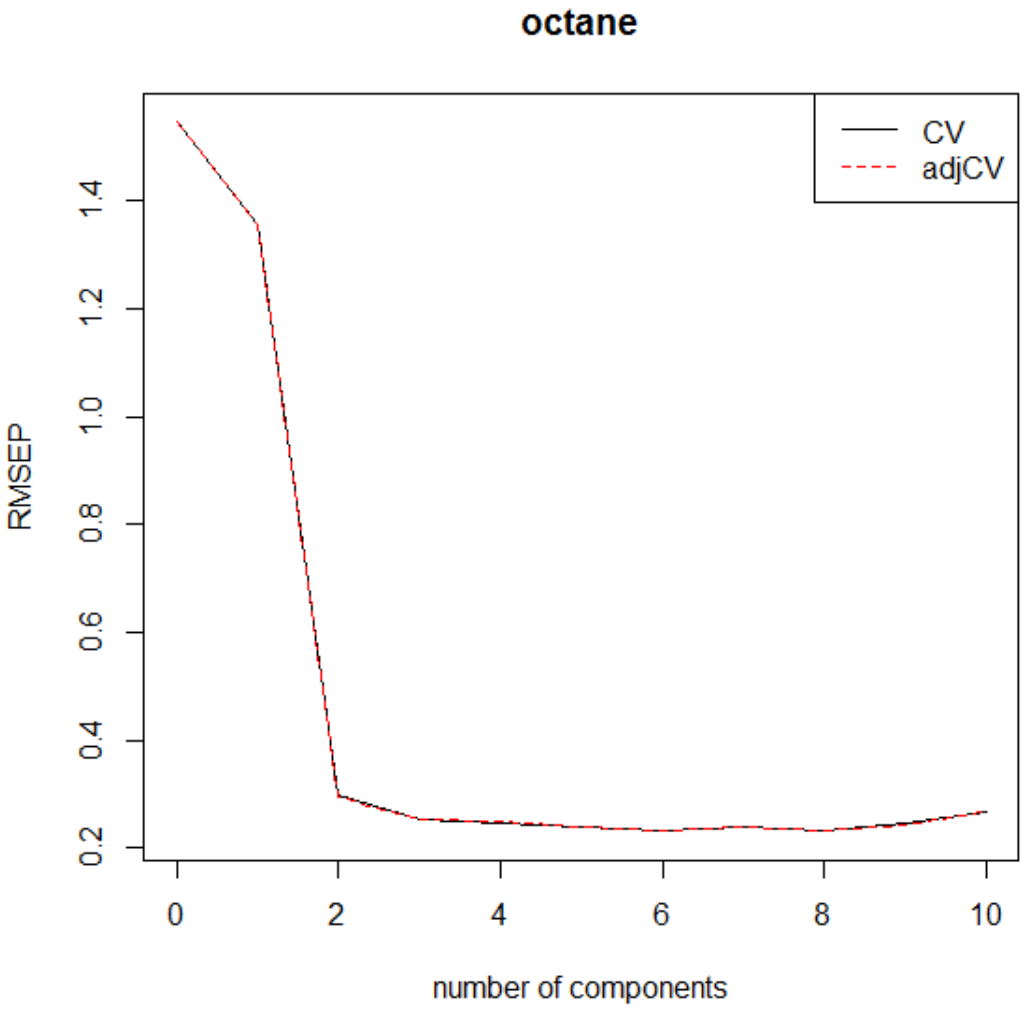
Vidimo da prva komponenta najviše opisuje dane varijable, te da druga i treća varijabla pridonose sa 7,41% odnosno 7,82% pri objašnjavanju početne varijable. Iz ovoga možemo pretpostaviti da će biti dovoljno uzeti dvije ili tri komponente za regresijski model.

Valja uočiti da je u slučaju PLS medote **Y** varijabla objašnjena s čak 96%, dok je s dvije komponente koje su odabrane PCA metodom **Y** varijabla objašnjena s 21%. Također, predviđeni RMSE s dvije komponente je daleko manji kod PLS metode nego kod PCA. Naše tvrdnje ćemo potkrijepiti pomoću RMSEP kao i u slučaju PCA.

```
> RMSEP (gas.pls, estimate="train", intercept= F)

 1 comps   2 comps   3 comps   4 comps   5 comps
  1.2724    0.2688    0.2197    0.1997    0.1615
 6 comps   7 comps   8 comps   9 comps  10 comps
  0.1544    0.1445    0.1390    0.1288    0.1178
```

Prikažimo grafički kako se mijenja RMSEP u odnosu na broj novih komponenti.

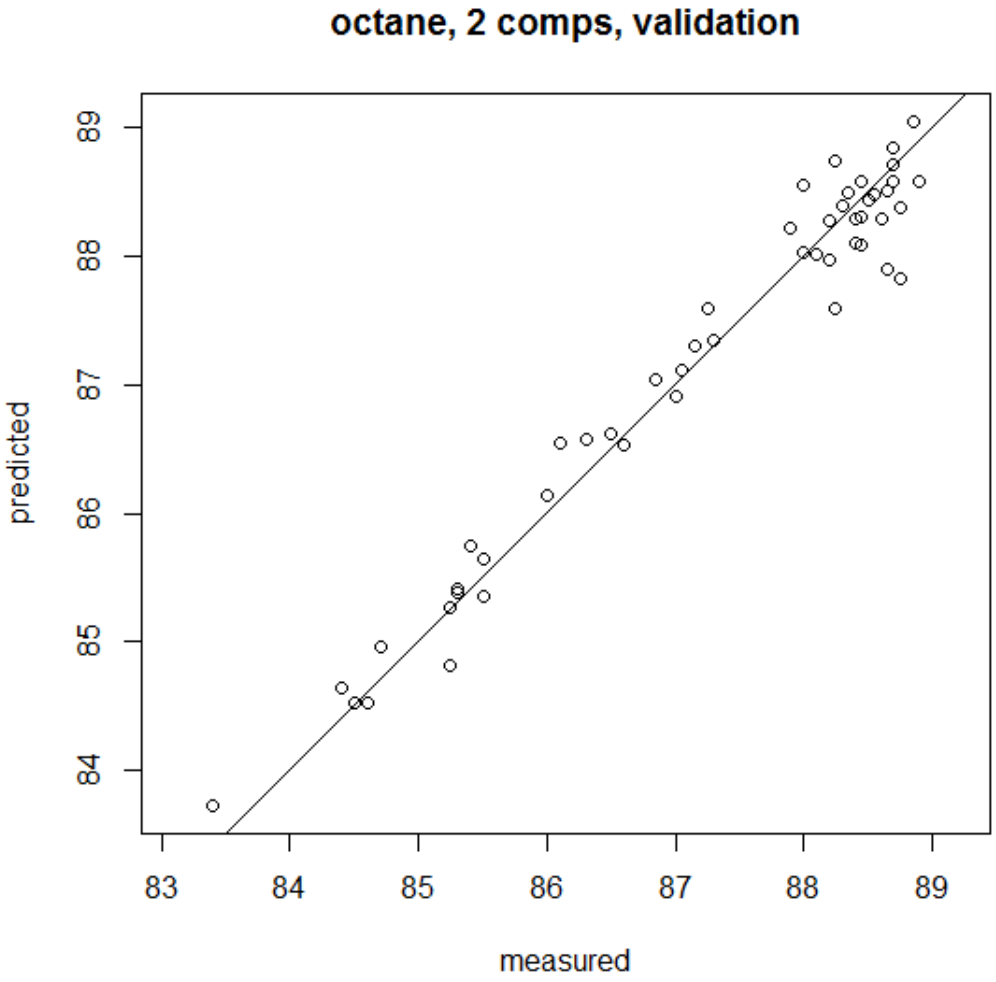
```
> plot(gas.pls, "validation", estimate = "CV")
```



Iz grafičkog prikaza se također vidi da se za izbor od dvije nove komponente RMSEP značajno ne mijenja, te je dovoljno promatrati dvije komponente.

Prije same predikcije ćemo analizirati naš model. Za početak ćemo usporediti prave i predviđene vrijednosti octanea u train setu, u odnosu na tri nove komponente koje smo dobili pomoću PLS metode. Želimo da se predviđeni podaci nalaze na identiteti.

```
> plot(gas.pls, ncomp = 2, asp = 1, line = TRUE)
```



Iz grafičkog prikaza vidimo da se dobiveni prediktivni podaci nalaze oko zadanog pravca, te da naš model dobro opisuje dane podatke.

Sljedeći korak je predviđanje vrijednosti varijable *octane* iz skupa za testiranje.

```
> predict ( gas.pls , ncomp = 2 , newdata = gasTest )

, , 2 comps

      octane
51 87.94125
52 87.25242
53 88.15832
54 84.96913
```

```
55 85.15396
56 84.51415
57 87.56190
58 86.84622
59 89.18925
60 87.09116
```

S obzirom da znamo koje su točne vrijednosti varijable *octane*, možemo usporediti predviđene i prave vrijednosti dane varijable.

```
> gasTest$octane

 [1] 88.10 87.60 88.35 85.10 85.10 84.70 87.20 86.60
 [9] 89.60 87.10
```

Iz danog ispisa vidimo da su predviđene vrijednosti varijable octane, relativno blizu pravim vrijednostima. Našu tvrdnju ćemo potvrditi tako što ćemo izračunati RMSEP.

```
> RMSEP(gas.pls, newdata = gasTest)

(Intercept)      1 comps      2 comps      3 comps
4 comps
     1.5369       1.1696       0.2445       0.2341
0.3287
    5 comps      6 comps      7 comps      8 comps
9 comps
     0.2780       0.2703       0.3301       0.3571
0.4090
   10 comps
     0.6116
```

Vidimo da RMSEP za dvije komponente iznosi 0.2445, što je dovoljno blizu procijenjenoj vrijednosti dobivenoj cross-validacijom.

6.1.4 Usporedba rezultata

Sljedeći korak je usporediti PLS regresiju (PLSR) s klasičnom PCA regresijom (PCR). Valja primijetiti da kod PLSR-a (kao što smo već rekli), prvotno tražimo nove koordinate koje će što bolje opisivati matrične varijable $\mathbf{X}$ i $\mathbf{Y}$, odnosno PLS model nam ovisi o obje varijable. Nakon toga prikažemo obje matrične varijable u novim koordinatima te radimo linearni regresijski model (više varijabli) između varijabli $\mathbf{X}$ i $\mathbf{Y}$. Za razliku od PLSR-a, kod PCR-a tražimo nove koordinate ovisno samo o matričnoj varijabli $\mathbf{X}$. Tada prikažemo obje varijable $\mathbf{X}$ i $\mathbf{Y}$ u novim varijablama te kao i kod PLSR-a radimo linearni regresijski model (više varijabli). S obzirom da PLSR pri kreiranju novih koordinata, uzima u obzir i varijablu $\mathbf{Y}$, logično je pretpostaviti da će predviđene vrijednosti tom metodom dati bolje rezultat.

Nakon provedenih metoda nad istim tipom podataka, možemo uočiti da pomoću PLS metode postižemo manju grešku (0.2445) u predviđanju pomoću samo dvije komponente, nego li pomoću PCA metode s tri komponente (0.4634). Dodatno, u ovom slučaju kod PCA metode je potrebno čak četiri komponente (0.2241) da se postigne jednako mala greška kao kod PLS-a.

6.2 Multivarijatan Y

Dani su nam podaci **vehicles** – koji predstavljaju 30 automobila s 16 specifikacija (*diesel* – tip dizelovog goriva, *turbo* – ubrizgavanje goriva , *two.doors* – vozilo s dvoja vrata, *hatchback* – leteća vrata, *wheel.base* – baza kotača, *length* – duljina, *width* – širina, *height* – visina, *curb.weight* – ukupna težina auta s punom opremom, *eng.size* – veličina motora, *horsepower* – broj konjskih snaga, *symbol* – ocjena rizika osigurnja, *peak.rpm* – broj okretaja u minuti, *price* – cijena dolarima, *city.mpg* – potrošnja goriva u gradu, *highway.mpg* – potrošnja goriva na autocesti).

Od danih specifikacija načinit ćemo matrične varijable **X** i **Y**. Specifikacije *diesel, turbo, two.doors, hatchback, wheel.base, length, width, height, curb.weight, eng.size, horsepower, symbol, peak.rpm* će predstavljati 13 komponenti matrične varijable **X** s 30 observacija. Dok će specifikacije *price, city.mpg* i *highway.mpg* predstavljati komponente matrične varijable **Y**, koja će imati 30 opservacija.

Cilj je pomoću regresijskih metoda PLS1 i PLS2 predvidjeti varijable na temelju ostalih mjerenih vrijednosti koje su nam poznate.

Prvotno je potrebno učitati podatke i podjeliti naš skup na dva dijela – jedan na kojem ćemo učiti (trenirati) algoritam i jedan na kojem ćemo ga testirati. Kako bi algoritam mogao što više naučiti na danim podacima, dijelimo skup na 80% i 20% podataka.

```
> data(vehicles)
> head(vehicles)
> dim(vehicles)
> carTrain <- vehicles[1:24,]
> carTest <- vehicles[25:30,]
```

Gdje ispis danog koda izgleda ovako.

	diesel	turbo	two.doors	hatchback	wheel.base
alfaromeo	0	0	1	1	94.5
audi	0	0	0	0	105.8
bmw	0	0	1	0	101.2
chevrolet	0	0	0	0	94.5
dodge1	0	1	1	1	93.7
dodge2	0	0	0	1	93.7

	length	width	height	curb.weight	eng.size
alfaromeo	171.2	65.5	52.4	2823	152
audi	192.7	71.4	55.7	2844	136
bmw	176.8	64.8	54.3	2395	108

```
chevrolet   158.8   63.6    52.0          1909         90
dodge1      157.3   63.8    50.8          2128         98
dodge2      157.3   63.8    50.6          1967         90
            horsepower peak.rpm price symbol city.mpg
alfaromeo          154     5000 16500      1        19
audi               110     5500 17710      1        19
bmw                101     5800 16430      2        23
chevrolet           70     5400  6575      0        38
dodge1             102     5500  7957      1        24
dodge2              68     5500  6229      1        31
            highway.mpg
alfaromeo            26
audi                 25
bmw                  29
chevrolet            43
dodge1               30
dodge2               38
```

6.2.1 PLS1

Prvo ćemo za $\mathbf{Y}$ provesti PLS1 metodu, odnosno provest ćemo PLSR za svaku komponentu posebno, redom *price, city.mpg* i *highway.mpg*. Odredit ćemo broj komponenti koje su potrebne za što bolji prikaz varijabli, procijenit ćemo dane varijable metodom PLSR te ćemo odrediti pogrešku dobivenu metodom.

price

```
> car.pls11 <- plsr(price ~ diesel + turbo + two.doors
              + hatchback + wheel.base + length + width
              + height + curb.weight + eng.size +
               horsepower + symbol + peak.rpm ,
          data= carTrain, ncomp = 10, validation="LOO")
> summary(car.pls11)

Data:    X dimension: 24 13
         Y dimension: 24 1
Fit method: kernelpls
```

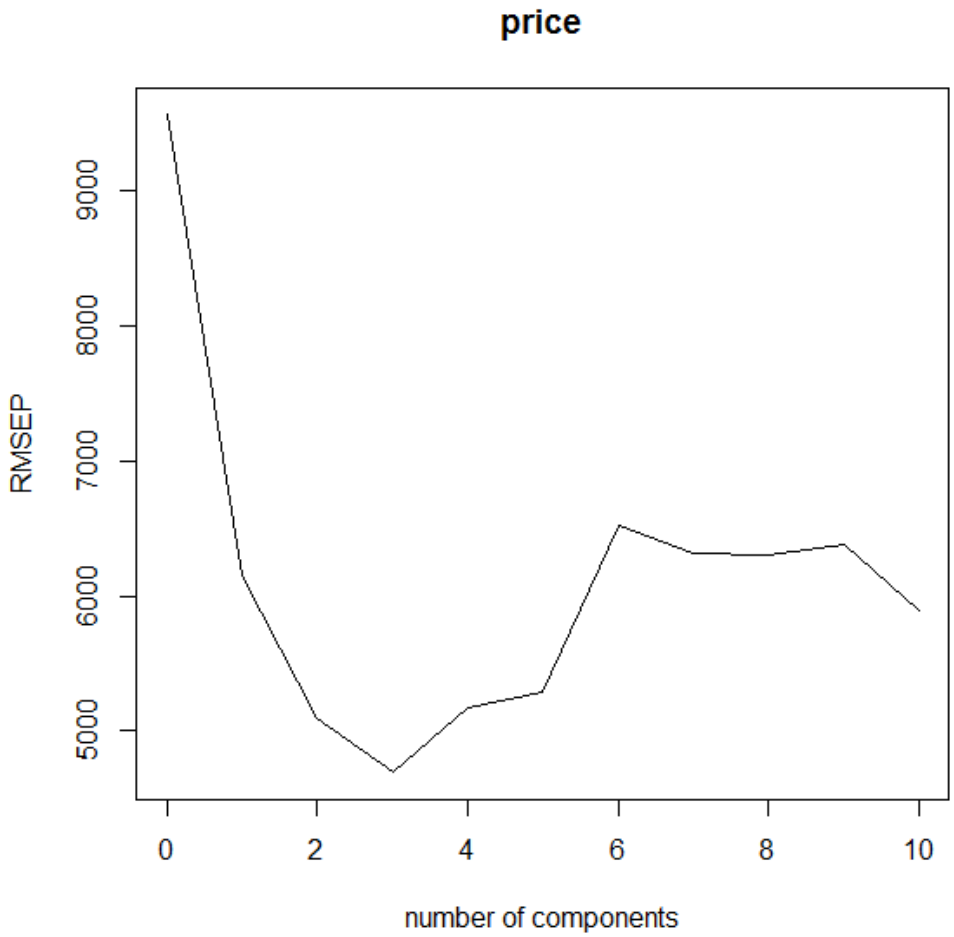
```
Number of components considered: 10

VALIDATION: RMSEP
Cross-validated using 24 leave-one-out segments.
       (Intercept)  1 comps  2 comps  3 comps
CV            9558     6144     5094     4692
adjCV         9558     6140     5073     4662
       4 comps  5 comps  6 comps  7 comps  8 comps
CV        5169     5292     6517     6319     6307
adjCV     5121     5243     6431     6242     6220
       9 comps  10 comps
CV        6377     5891
adjCV     6286     5810

TRAINING: % variance explained
       1 comps  2 comps  3 comps  4 comps  5 comps
X        68.48    99.86    99.97    99.99    100.0
price    65.78    79.83    87.63    89.08     89.3
       6 comps  7 comps  8 comps  9 comps  10 comps
X       100.00   100.00   100.00   100.00    100.00
price    90.79    91.41    93.44    94.26     94.63
```

Vidimo da tri, odnosno četiri, komponente dovoljno dobro opisuju varijable $\mathbf{X}$ i $\mathbf{Y}$. Iz ovoga vidimo da se se procijenjuju jako velike vrijednosti RMSEP. Sljedeći korak je kao i u prethodnom primjeru izračunati RMSEP na skupu za treniranje.

```
> RMSEP(car.pls11, estimate="train", intercept= F)

 1 comps   2 comps   3 comps   4 comps   5 comps
    5358      4113      3221      3026      2997
 6 comps   7 comps   8 comps   9 comps  10 comps
    2779      2685      2346      2194      2123
```

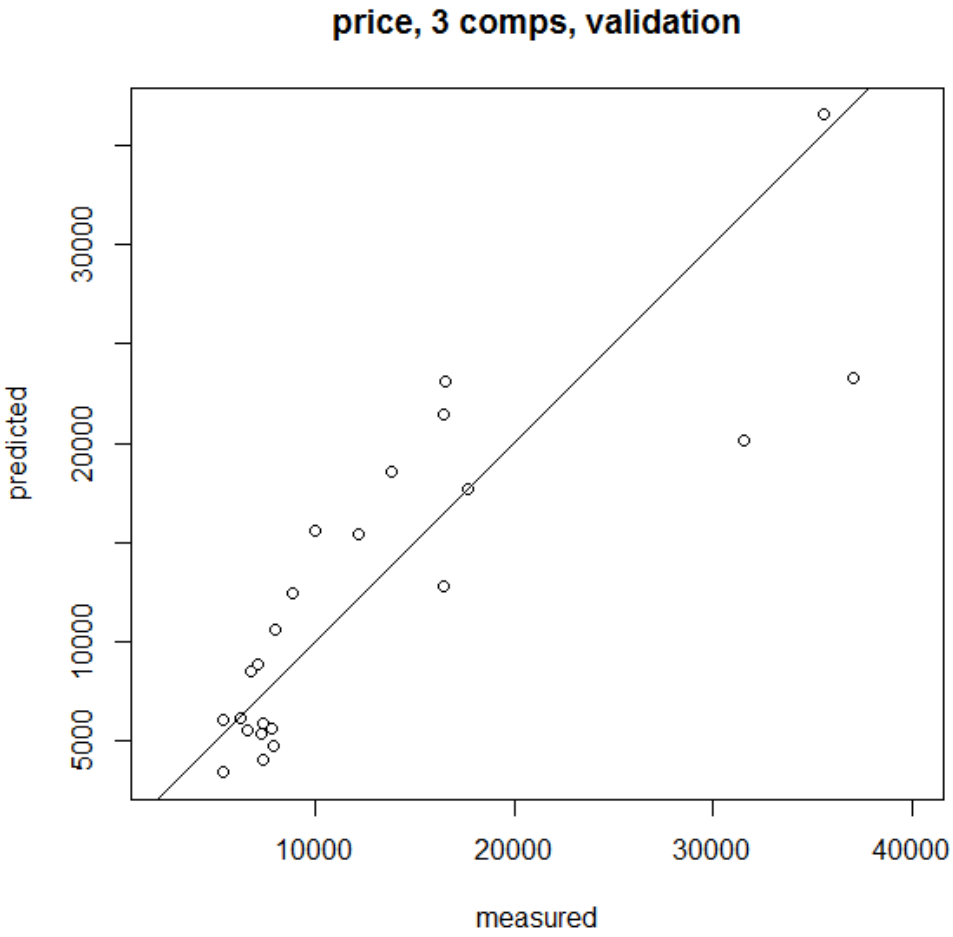
Vidimo da se najveći skok smanjenja RMSEP-a događa kada prelazimo s dvije na tri komponente.

```
> plot(car.pls11, "validation", estimate = "CV")
```



Kao što smo i predviđali, iz grafa je vidljivo da su tri komponente najbolji izbor za modeliranje nad našim podacima. Sljedeći korak je pokazati da su podaci dobro opisani modelom, tj. da se nalaze što bliže dijagonali prvog kvadranta, a što vidimo iz sljedećeg grafa. S obzirom da podaci, više manje prate dani pravac (iako ne dovoljno dobro, te pretpostavljamo da će to biti rezultat jako velike vrijednosti RMSEP-a).

```
> plot(car.pls11, ncomp = 3, asp = 1, line = TRUE)
```




```
> predict ( car.pls11 , ncomp = 3 , newdata = carTest )

, , 3 comps

                price
toyota4       7760.714
volkswagen1   4911.454
volkswagen2  16674.648
volvo1       18515.043
volvo2       22593.449
volvo3       18218.645

> RMSEP(car.pls11 , newdata = carTest)
```

```
(Intercept)      1 comps      2 comps      3 comps
       5225         2681         4284         3873
    4 comps      5 comps      6 comps      7 comps
       3759         3994         3755         3689
    8 comps      9 comps     10 comps
       3095         2661         1987
```

city.mpg

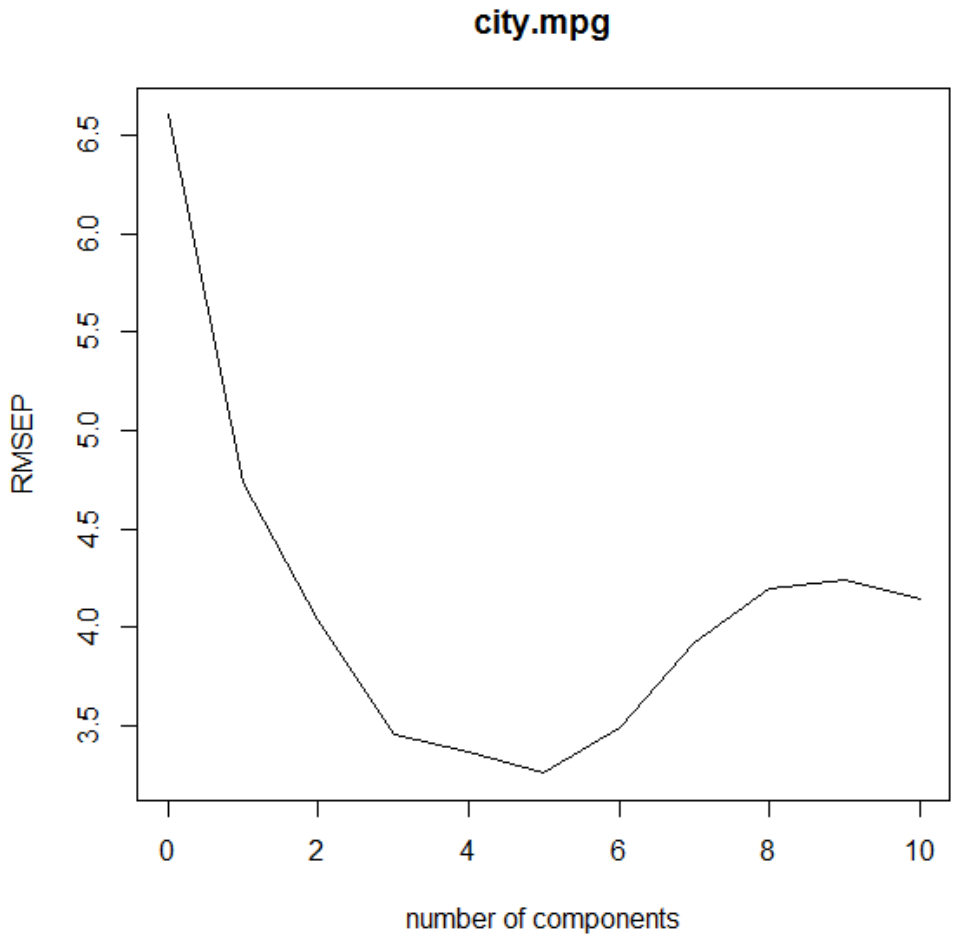
Sljedeća komponenta $\mathbf{Y}$ koja ima ulogu vektorske varijable $\mathbf{Y}$ je *city.mpg*. Kao i do sada u regresijski model ulazi 13 komponenti matrične varijable $\mathbf{X}$.

```
> car.pls12 <- plsr(city.mpg ~ diesel + turbo +
            two.doors + hatchback + wheel.base +
            length + width + height + curb.weight +
            eng.size + horsepower + symbol + peak.rpm,
          data= carTrain, ncomp = 10, validation="LOO")
> summary(car.pls12)

Data:    X dimension: 24 13
         Y dimension: 24 1
Fit method: kernelpls
Number of components considered: 10

VALIDATION: RMSEP
Cross-validated using 24 leave-one-out segments.
       (Intercept)  1 comps  2 comps  3 comps
CV           6.606    4.732    4.032    3.451
adjCV        6.606    4.724    4.021    3.440
       4 comps  5 comps  6 comps  7 comps  8 comps
CV       3.365    3.254    3.491    3.921    4.196
adjCV    3.349    3.238    3.466    3.884    4.154
       9 comps  10 comps
CV       4.241    4.138
adjCV    4.192    4.095

TRAINING: % variance explained
          1 comps  2 comps  3 comps  4 comps  5 comps
X           67.96    99.86    99.97    99.99   100.00
city.mpg    56.55    69.32    79.19    83.09    85.04
```

```
           6 comps  7 comps  8 comps  9 comps
X            100.00   100.00   100.00     100.0
city.mpg      87.01    87.91    88.66      89.5
           10 comps
X            100.00
city.mpg      89.66
```

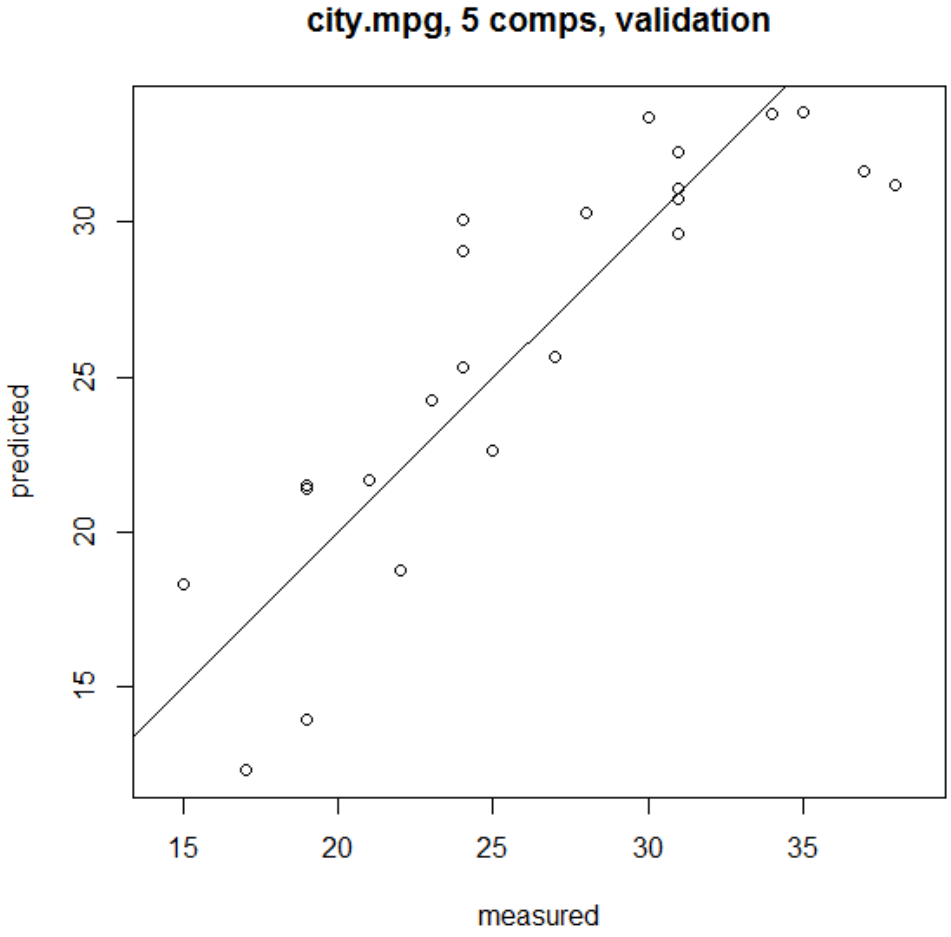
```
> RMSEP(car.pls12, estimate="train", intercept= F)
 1 comps  2 comps  3 comps  4 comps  5 comps
   4.173    3.507    2.888    2.604    2.449
 6 comps  7 comps  8 comps  9 comps 10 comps
   2.282    2.201    2.132    2.052    2.036
```

```
> plot(car.pls12, "validation", estimate = "CV")
```



Vidimo da su predviđene greške nešto manje nego u prethodnom primjeru. Također vidimo da se najveći skok RMSEP-a vidi s dvije na tri varijable. Usprkost tome, prema grafu slijedi da će broj biti najbolje za regresijski model ući s 5 komponenti.

```
> plot(car.pls12, ncomp = 3, asp = 1, line = TRUE)
```




```
> predict ( car.pls12 , ncomp = 3 , newdata = carTest )
, , 5 comps

             city.mpg
toyota4      29.71152
volkswagen1  31.98912
volkswagen2  23.31514
```

```
volvo1      21.16717
volvo2      16.04023
volvo3      23.12700

> RMSEP(car.pls12 , newdata = carTest)

(Intercept)      1 comps      2 comps      3 comps
      6.423        5.381        4.004        3.121
    4 comps      5 comps      6 comps      7 comps
      3.569        3.271        3.365        3.183
    8 comps      9 comps     10 comps
      3.160        3.232        3.240
```

Za kraj gledamo koliko naš model opisuje dane podatke. Vidimo da se naši podaci kreću oko identitete. Također određujemo procijenjene vrijednosti i vidimo da je za tri komponente RMSEP nešto manji nego li za pet komponenti, kako smo predviđali.

highway.mpg

Za kraj preostaje pokazati odnos između *highway.mpg* i matrične varijable **X** pomoću PLS metode.

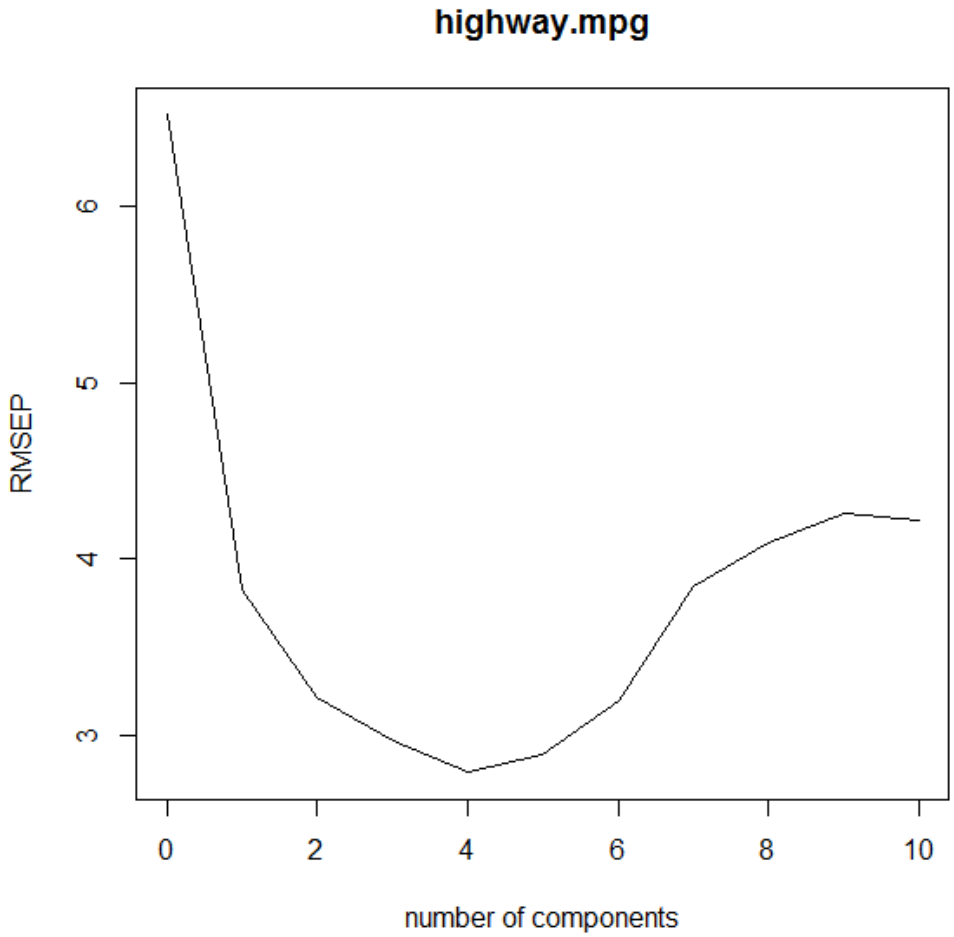
```
> car.pls13 <- plsr(highway.mpg ~ diesel + turbo +
              two.doors + hatchback + wheel.base +
              length + width + height + curb.weight +
              eng.size + horsepower + symbol + peak.rpm,
          data= carTrain, ncomp = 10, validation="LOO")
> summary(car.pls13)

Data:    X dimension: 24 13
         Y dimension: 24 1
Fit method: kernelpls
Number of components considered: 10

VALIDATION: RMSEP
Cross-validated using 24 leave-one-out segments.
       (Intercept)  1 comps  2 comps  3 comps
CV           6.518    3.822    3.212    2.968
adjCV        6.518    3.815    3.203    2.959
       4 comps  5 comps  6 comps  7 comps  8 comps
```

```
CV          2.789     2.892     3.195     3.842     4.090
adjCV       2.778     2.880     3.174     3.805     4.048
       9 comps  10 comps
CV       4.255    4.220
adjCV    4.206    4.174

TRAINING: % variance explained
              1 comps  2 comps  3 comps  4 comps
X               70.80    99.86    99.97    99.99
highway.mpg     70.16    79.98    84.37    86.89
              5 comps  6 comps  7 comps  8 comps
X              100.00   100.00   100.00   100.00
highway.mpg     87.43    88.25    88.76    89.24
              9 comps  10 comps
X              100.00    100.00
highway.mpg     89.63     89.73
```

```
> RMSEP(car.pls13, estimate="train", intercept= F)

 1 comps  2 comps  3 comps  4 comps  5 comps
   3.412    2.795    2.469    2.262    2.215
 6 comps  7 comps  8 comps  9 comps  10 comps
   2.141    2.094    2.049    2.011     2.001
```

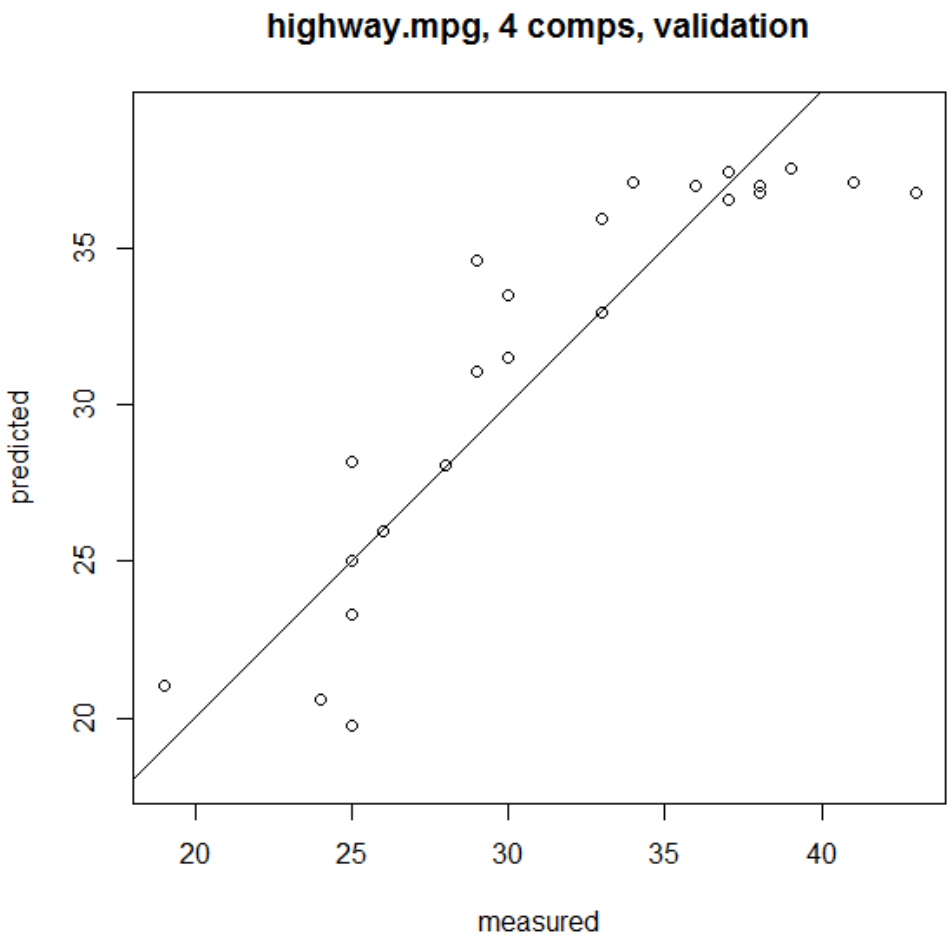
Ko i u svim primjerima do sada izračunali smo predviđeni RMSEP te smo izračunali RMSEP na skupu za treniranje. Iz gore izračunatih vrijednosti možemo zaključiti (jednako obrazloženje kao i u svim primjerima do sada) da će u model biti dovoljno ući s četiri varijable. Ovu našu tvrdnju potvrđuje i graf, iz kojeg isčitavamo da se najmanji RMSEP na skupu za treniranje postiže za četiri komponente.

```
> plot(car.pls13, "validation", estimate = "CV")
```



Sljedeći korak je prikazati kako naš model opisuje dane podatke. Kao i u svim primjerima do sad, naši podaci relativno prate dan pravac. Za kraj nam preostaje predvidjeti vrijednosti *highway.mpg* iz skupa za testiranje i izračunati vrijednosti RMSEP-a na skupu za testiranja. Dani rezultat nam govori da se na skupu za testiranje pokazalo da je greška za RMSEP nešto manja ako uzmemo tri umjesto četiri komponente.

```
> plot(car.pls13, ncomp = 4, asp = 1, line = TRUE)
```




```
> predict ( car.pls13 , ncomp = 4 , newdata = carTest )

, , 4 comps

            highway.mpg
toyota4        33.18953
volkswagen1    35.97803
volkswagen2    29.52233
volvo1         27.01297
volvo2         21.89062
volvo3         25.49739

> RMSEP(car.pls13 , newdata = carTest)
```

```
(Intercept)      1 comps      2 comps      3 comps
      7.603        6.270        5.104        4.698
    4 comps      5 comps      6 comps      7 comps
      4.897        4.852        4.972        4.866
    8 comps      9 comps     10 comps
      4.877        5.047        4.898
```

6.2.2 PLS2

Sljedeći korak je provesti PLS2 metodu. Kao što je opisano u poglavlju Algoritmi, PLS2 metoda istovremeno računa matricu koeficijenata za sve tri komponente koje želimo procijeniti PLSR metodom.

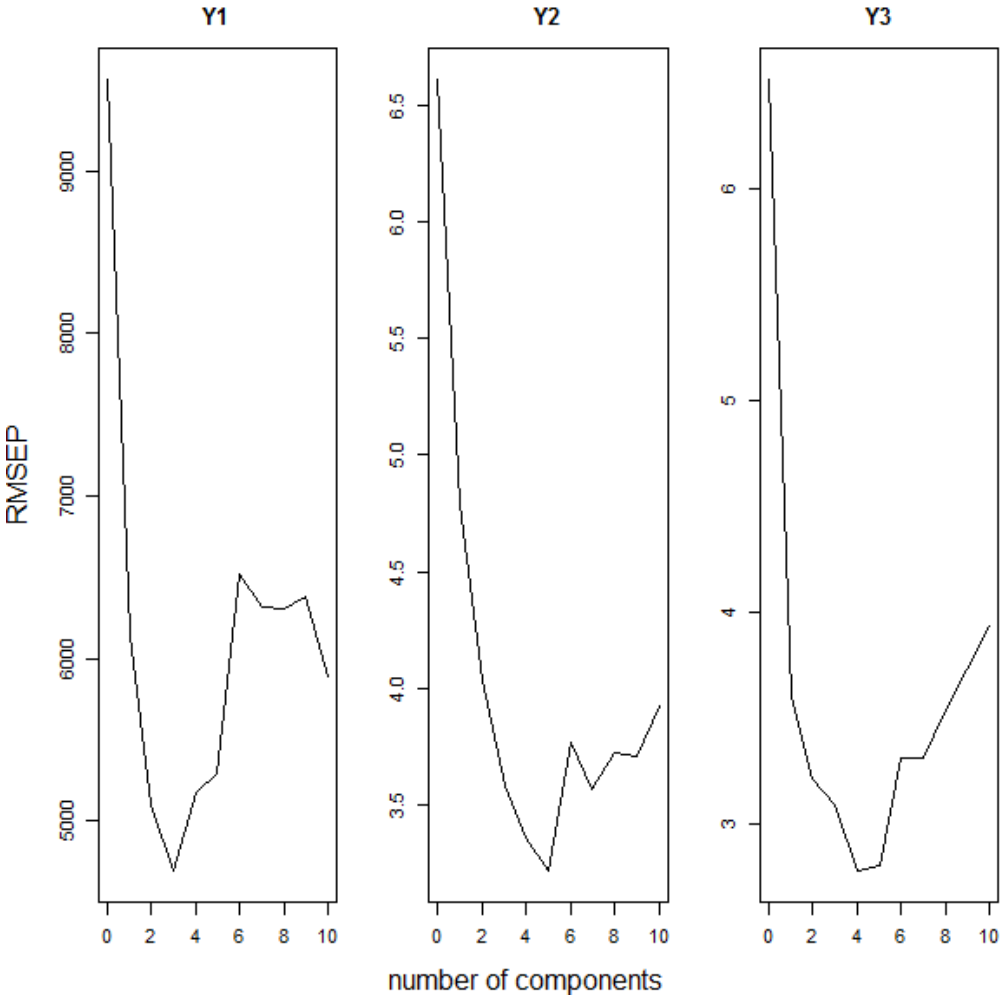
```
> car.pls2 <- plsr(matrix(
        c(price, city.mpg, highway.mpg), 24,3) ~. ,
        data= carTrain, ncomp = 10, validation="LOO")
> summary(car.pls2)

Data:    X dimension: 24 13
         Y dimension: 24 3
Fit method: kernelpls
Number of components considered: 10

VALIDATION: RMSEP
Cross-validated using 24 leave-one-out segments.

Response: Y1
       (Intercept)  1 comps  2 comps  3 comps
CV            9558     6144     5094     4692
adjCV         9558     6140     5073     4662
       4 comps  5 comps  6 comps  7 comps  8 comps
CV        5169     5292     6517     6319     6307
adjCV     5121     5243     6431     6242     6220
       9 comps  10 comps
CV        6377      5891
adjCV     6286      5810

Response: Y2
       (Intercept)  1 comps  2 comps  3 comps
CV           6.606    4.787    4.034    3.583
adjCV        6.606    4.781    4.023    3.564
       4 comps  5 comps  6 comps  7 comps  8 comps
```

```
CV        3.360     3.219     3.771     3.569     3.723
adjCV     3.345     3.192     3.700     3.530     3.685
       9 comps  10 comps
CV        3.711     3.928
adjCV     3.666     3.884

Response: Y3
       (Intercept)  1 comps  2 comps  3 comps
CV           6.518    3.599    3.212    3.092
adjCV        6.518    3.591    3.203    3.078
       4 comps  5 comps  6 comps  7 comps  8 comps
CV       2.776    2.804    3.308    3.308    3.542
adjCV    2.765    2.790    3.269    3.281    3.508
       9 comps  10 comps
CV       3.728    3.935
adjCV    3.691    3.894

TRAINING: % variance explained
    1 comps  2 comps  3 comps  4 comps  5 comps
X     68.48    99.86    99.97    99.99   100.00
Y1    65.78    79.83    87.63    89.08    89.30
Y2    55.57    69.29    75.93    81.31    83.65
Y3    74.43    79.98    82.13    86.45    86.90
    6 comps  7 comps  8 comps  9 comps  10 comps
X    100.00   100.00   100.00   100.00    100.00
Y1    90.79    91.41    93.44    94.26     94.63
Y2    87.56    87.88    88.45    88.95     89.05
Y3    88.23    88.38    88.90    88.91     88.99

> RMSEP(car.pls2, estimate="train", intercept= F)

Response: Y1
 1 comps   2 comps   3 comps   4 comps   5 comps
    5358      4113      3221      3026      2997
 6 comps   7 comps   8 comps   9 comps  10 comps
    2779      2685      2346      2194      2123

Response: Y2
 1 comps   2 comps   3 comps   4 comps   5 comps
   4.220     3.509     3.106     2.737     2.560
 6 comps   7 comps   8 comps   9 comps  10 comps
```

```
   2.233       2.204       2.151       2.105       2.095

Response: Y3
 1 comps   2 comps   3 comps   4 comps   5 comps
   3.159     2.795     2.641     2.300     2.261
 6 comps   7 comps   8 comps   9 comps  10 comps
   2.143     2.130     2.081     2.081     2.072
```

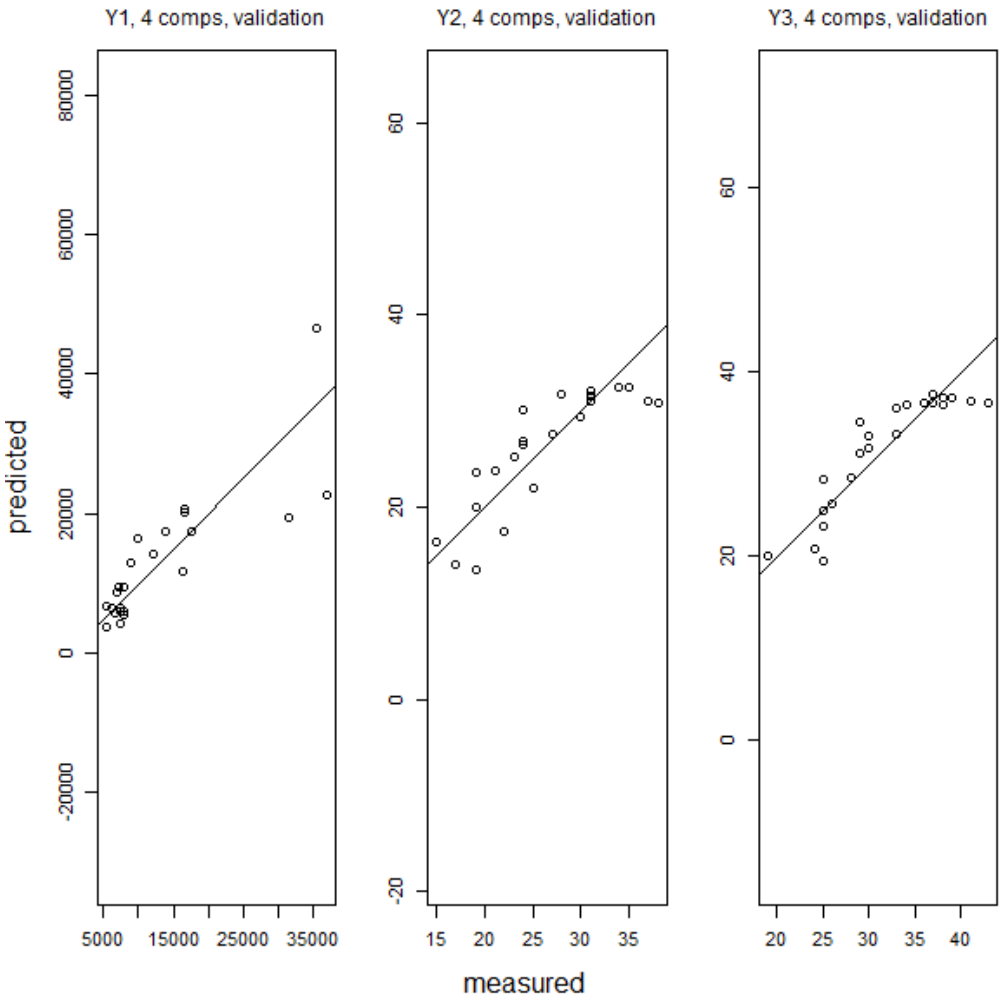
S obzirom da je $\mathbf{Y}$ multivarijantan iz danog ispisa je teško isčitati potreban broj komponenti. Zbog toga ćemo nacrtati grafove kao i u prethodnim primjerima kako bismo otkrili koji broj komponenti je najbolji izbor za dane podatke.

```
> plot(car.pls2, "validation", estimate = "CV")
```



Iz danih grafova vidimo da je za prvu komponentu dovoljno uzeti tri komponente, za drugu komponentu je dovoljno uzeti pet komponenti, a za zadnju četiri komponente. Iz gornjih podataka vidimo da u postotku nema velikih odstupanja objašnjavanja naših varijabli te da se vrijednosti RMSEP ne mijenjaju značajno. Iz tog razloga se odlučujemo za izbor od četiti komponente.

```
> plot(car.pls2, ncomp = 4, asp = 1, line = TRUE)
```



Iz gornjeg grafa vidimo da pretpostavljeni linearni model nije sasavim adekvatan za opis danih podataka. Tvrdnju ćemo potkrijepiti s visokim vrijednosti RMSEP na skupu za testiranje.

```
> y <- predict(car.pls2, ncomp = 4, newdata = carTest)
> y
```

```
, , 4 comps

                    Y1        Y2        Y3
toyota4       7423.680  29.44524  33.18372
volkswagen1   5255.961  32.09227  35.98561
volkswagen2  16932.434  24.18414  29.66130
volvo1       18237.267  22.30272  27.22024
volvo2       19757.930  16.31139  22.11341
volvo3       17380.071  21.91794  25.35285

>rms <- function(x, y) sqrt(mean((x-y)^2))

> rms(carTest$price, y[,1,])
> rms(carTest$city.mpg, y[,2,])
> rms(carTest$highway.mpg, y[,3,])

[1] 3758.864
[1] 3.550748
[1] 4.899416
```

6.2.3 Usporedba rezultata

Gornje metode PLS1 i PLS2 smo provodili na istim podacima kako bismo mogli usporediti konačne rezultate. Ono što smo pokazali je da se na ovom primjeru ove metode ne razlikuju značajno. Ako usporedimo RMSEP koji dobijemo za komponentu *price* – PLS1 (3873) i PLS2 (3758), *city.pmg* – PLS1 (3.27) i PLS2 (3.55) te *highway.mpg* – PLS1 (4.89) i PLS2 (4.89), vidimo da obe metode daju podjednake rezultate. Usprkost tome u praksi se više koristi PLS1 zbog jednostavnijeg razumijevanja i implemetacije.

Literatura

[1] R. Christensen, *Advanced Linear Modeling*, Springer, 2007

[2] R. Wehrens, *Chemometrics with R - Multivariate Data Analysis in the Natural Sciences and Life Sciences*, Springer, 2011

[3] H. Wold, *Estimation of principal components and related models by iterative least squares*, In Multivariate Analysis (Ed., P.R. Krishnaiah), Academic Press, 1966, 391-420

[4] S. Jong, *SIMPLS: an alternative approach to partial least squares regression*, Chemometrics and Intelligent Laboratory Systems, 18, 1992, 251-263

[5] R. Rosipal, N. Krämer, *Overview and Recent Advances in Partial Least Squares*, 34-42
`http://staff.ustc.edu.cn/~zwp/teach/Reg/overview_pls.pdf`

[6] B. H. Mevik, R. Wehrens, *Introduction to the pls Package*, 2015
`https://cran.r-project.org/web/packages/pls/vignettes/pls-manual.pdf`

[7] S. Maitra, J. Yan , *Principle Component Analysis and Partial Least Squares: Two Dimension Reduction Techniques for Regression*, 79-90, 2008
`https://www.casact.org/pubs/dpp/dpp08/08dpp76.pdf`

[8] H. Chun, S. Keles , *Sparse Partial Least Squares Regression for Simultaneous Dimension Reduction and Variable Selection*, 2010
`http://www.stat.wisc.edu/~keles/Papers/SPLS_Nov07.pdf`

[9] T. Y. Liu, L. Trinchera, A. Tenenhaus, D. Wei, A. O. Hero , *Globally Sparse PLS Regression*, 2013
`http://web.eecs.umich.edu/~hero/Preprints/liu_GSIMPLS_Springer13.pdf`

[10] P. H. Garthw, *An lnterpretation of Partial Least Squares* , 1993
`http://avesbiodiv.mncn.csic.es/estadistica/pls2.pdf`

[11] R. D. Tobias , *An Introduction to Partial Least Squares Regression*
`http://www.ats.ucla.edu/stat/sas/library/pls.pdf`

[12] S. Hall , *Implementation and Verification of a Robust PLS Regression Algorithm*
http://publications.lib.chalmers.se/records/fulltext/199254/199254.pdf

[13] B. M. Wise , *Properties of Partial Least Squares (PLS) Regression, and differences between Algorithms*
http://www.eigenvector.com/Docs/Wise_pls_properties.pdf

[14] H. Risvik , *Principal Component Analysis (PCA) and NIPALS algorithm*
http://folk.uio.no/henninri/pca_module/pca_nipals.pdf

[15] Ö. Yeniay, A. Göktas, *A comparison of partial squares regression with other prediciton methods*, 2002
http://www.hjms.hacettepe.edu.tr/uploads/ce7bdb8f-5f89-4f03-9822-a20e2ee35d03.pdf

[16] Z. Drmač, *Numerička Analiza 1*, 2009
https://web.math.pmf.unizg.hr/~drmac/NA1-3.pdf

[17] S. Singer, *Numerička Analiza - 25 predavanje*, 2009
https://web.math.pmf.unizg.hr/~singer/num_anal/NA_0910/25.pdf

Sažetak

PLS metoda (metoda parcijalnih najmanjih kvadrata) je statistička metoda koju je 1966. godine predstavio Herman Wold kao algoritam sličan metodi potencija na polju ekonometrije. U današnjici se dana metoda više koristi na području kemometrije, gdje se koristi u kombinaciji s regresijskim modelom za predviđanje nezavisne varijable. Cilj metode je reducirati dimenziju broja komponenti kojima ćemo ući u regresijski model, zbog čega nas podjeća na PCA (metoda glavnih komponenti). Iako je PCA poznatiji u teoriji, u praksi se češće koristi PLS metoda jer pri redukciji komponenti u regresijskom modelu uzima u obzir zavisnu i nezavisnu varijablu. Osim samog modela, važna je i implementacija te se u radu spominju i dva najpoznatija algoritma: NIPALS i SIMPLS.

Summary

PLS method (partial least squares method) is a statistical method that was originated in 1966 by Herman Wold as an algorithm alike to the power method in the field of econometrics. In the present, given method is being used in the field of chemometrics, where is used in combination with the regression model to predict the independent variable. The aim is to reduce the dimension of the number of components and performs least squares regression on these components, instead of on the original data – which is similar to the PCA (Principal Component Analysis). Although PCA is much more known in theory, PLS method is used more often in practice, because in the reduction of components and regression model combines components from dependent and independent variable. In addition to the modelling, implementation takes the important part in real life, that is why the two most famous algorithm NIPALS and SIMPLS are metioned in this paper.

Životopis

Tamara Sente rođena je 24. ožujka 1993. godine u Zagrebu. Osnovnu školu završava u Zagrebu te potom upisuje Gornjogradsku gimnaziju. Godine 2011. upisuje matematiku na Prirodoslovno-matematičkom fakultetu Sveučilišta u Zagrebu. Preddiplomski studij Matematike završa 2014. godine te tako stječe bacc. univ. math. Nakon toga upisuje dipomski studij matematike, smjer Matematička statistika.