Numeričke metode za tekstualnu analizu

Matanović, Lucijan

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Lucijan Matanović

NUMERIČKE METODE ZA TEKSTUALNU ANALIZU

Diplomski rad

Voditelj rada:
doc. dr. sc. Ivana Šain Glibić

Zagreb, veljača 2025.

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 obitelji, prijateljima i kolegama koji su mi bili oslonac tijekom studija te mentorici doc. dr. sc. Ivani Šain Glibić na stručnom vodstvu i vrijednim savjetima

Sadržaj

Uvod

U današnje doba, dostupne su nam velike količine podataka tako da možemo reći da su podaci svuda oko nas. Podaci su postali jedan od glavnih orijentira u mnogim područjima, od znanstvenih istraživanja do svakodnevnog poslovanja. Postoje različite vrste podataka, no svi podaci na kraju imaju zajedničko to da ih je najlakše analizirati ako ih imamo u brojčanom obliku. U ovom radu bavit ćemo se tekstualnim oblikom podataka koji je često nestrukturiran što može otežati izdvajanje korisnih informacija. Naglasak u radu je na razvoju i primjeni numeričkih metoda u kojima tražimo prikladne matrične faktorizacije. Početni korak je pripremiti podatke, odnosno napraviti pretvorbu teksta u matričnu formu jer će nam takav pristup omogućiti lakšu obradu i analizu. U našem slučaju imat ćemo dostupnu kolekciju dokumenata gdje stupcem želimo reprezentirati dokument, a retkom korištene riječi. Rad istražuje primjenu različitih metoda matričnih fatkorizacija poput Latent Semantic Indexing (LSI), klasteriranja, nenegativne matrične faktorizacije i LGK bidijagonalizacije. Jedan od ciljeva rada je razviti efikasne tehnike za ekstrakciju i klasifikaciju informacija što može značajno unaprijediti pretraživanje informacija. Rad je podijeljen u tri ključna dijela: najprije obrađujemo osnovne rezultate vezane za rad s matricama i vektorskim prostorima, zatim obrađujemo numeričke metode, te na kraju vršimo usporedbu tih metoda kako bi se procijenila njihova učinkovitost.

Sadržaj rada oslanja se na djelo Fundamentals of Algorithms: Matrix Methods in Data Mining and Pattern Recognition, 2019, odnosno [4].

Poglavlje 1

Osnovne definicije i motivacija

1.1 Osnovne definicije i teoremi

U ovom potpoglavlju uvodimo osnovne definicije i pojmove koji će nam biti od koristi. Definicije, iskazi teorema i propozicija su preuzeti iz [2], [5] te [8].

Definicija 1.1.1. *Neka su A i $P = \{P_1, \ldots, P_n\}$ skupovi. Kažemo da je P particija od A ako vrijede sljedeća svojstva:*

(1) $\emptyset \notin P$,

(2) $\bigcup_{i=1}^{n} P_i = A$,

(3) $P_i \bigcap P_j \neq \emptyset$, *za* $i \neq j$.

Definicija 1.1.2. *Neka je V vektorski prostor nad poljem $\mathbb{F}$, pri čemu je $\mathbb{F}$ polje $\mathbb{R}$ ili $\mathbb{C}$. Preslikavanje $s : V \times V \to \mathbb{F}$ koje svakom uređenom paru vektora pridružuje skalar $s(a, b) = \langle a, b\rangle \in \mathbb{F}$ naziva se skalarno množenje na prostoru V ako su ispunjena sljedeća svojstva:*

(1) $\langle a, a\rangle \geq 0$, *za sve* $a \in V$, *pri čemu je* $\langle a, a\rangle = 0$ *ako i samo ako je* $a = 0_V$;

(2) $\langle a, b\rangle = \overline{\langle b, a\rangle}$[1], *za sve* $a, b \in V$;

(3) $\langle \lambda a, b\rangle = \lambda\langle a, b\rangle$, *za sve* $a, b \in V$, $\lambda \in \mathbb{F}$;

(4) $\langle a + b, c\rangle = \langle a, c\rangle + \langle b, c\rangle$, *za sve* $a, b, c \in V$.

[1] Broj $\overline{\alpha}$ predstavlja konjugirano kompleksni broj od α.

Skalar $\langle a, b\rangle$ zove se skalarni produkt vektora a i b. Uređeni par (V, s) nazivamo unitarni prostor nad poljem $\mathbb{F}$.

Primijetimo da u definiciji skalarnog množenja tražimo svojstva homogenosti i linearnosti u samo prvom argumentu. Koristeći defincijska svojstva pokaže se da vrijede slična svojstva i za drugi argument.

Propozicija 1.1.1. *Neka je V unitarni prostor nad $\mathbb{F}$. Tada je*

(3') $\langle a, \lambda b\rangle = \overline{\lambda}\langle a, b\rangle$, *za sve* $a, b \in V$, $\lambda \in \mathbb{F}$,

(4') $\langle a, b + c\rangle = \langle a, b\rangle + \langle b, c\rangle$, *za sve* $a, b, c \in V$.

Definicija 1.1.3. *Neka su a i b vektori unitarnog prostora V. Kažemo da su vektori a i b međusobno ortogonalni ako vrijedi $\langle a, b\rangle = 0$.*

Teorem 1.1.1 (**Nejednakost Cauchy-Schwarz-Bunjakowskog**). *Neka je V unitarni prostor. Vrijedi*

$$|\langle a, b\rangle|^2 \leq |\langle a, a\rangle||\langle b, b\rangle|$$

za sve a, b ∈ V. Jednakost se postiže ako i samo ako je $\{a, b\}$ linearno zavisan skup.

Definicija 1.1.4. *Neka je V vektorski prostor nad poljem $\mathbb{F}$, pri čemu je $\mathbb{F} = \mathbb{R}$ ili $\mathbb{F} = \mathbb{C}$. Preslikavanje*

$$\|\cdot\| : V \to \mathbb{R}$$

koje svakom vektoru $a \in V$ pridružuje realni broj $\|a\|$ sa svojstvima:

(1) $\|a\| \geq 0$, *pri čemu je* $\|a\| = 0$ *ako i samo ako je* $a = 0_V$,

(2) $\|\lambda a\| = |\lambda| \cdot \|a\|$.

(3) $\|a + b\| \leq \|a\| + \|b\|$,

za sve a, b ∈ V, $\lambda \in \mathbb{F}$, naziva se norma na prostoru V. Uređeni par $(V, \|\cdot\|)$ zove se normirani prostor.

Prirodno je zapitati se možemo li općenito u unitarnom prostoru definirati preslikavanje koje će zadovoljavati svojstva norme kako bismo time dobili normirani prostor. Naime, odgovor je potvrdan jer u unitarnim prostorima normu možemo *inducirati* skalarnim produktom što pokazuje sljedeća propozicija.

Propozicija 1.1.2. *Neka je $(V, \langle\cdot,\cdot\rangle)$ unitarni prostor. Preslikavanje*

$$a \mapsto \sqrt{\langle a, a\rangle}$$

s prostora V u polje $\mathbb{R}$ je norma na prostoru V.

Dokaz. Označimo s $\|a\| = \sqrt{\langle a, a\rangle}$, za $a \in V$. Provjerimo da preslikavanje $\|\cdot\| : V \to \mathbb{R}$ zadovoljava sva tri svojstva koje posjeduje norma. Svojstvo *(1)* skalarnog produkta (pozitivna definitnost) direktno povlači i svojstvo *(1)* koje imamo kod normi. Pokažimo svojstvo *(2)*, za $a \in V$ i $\lambda \in \mathbb{C}$ imamo

$$\|\lambda a\| = \sqrt{\langle \lambda a, \lambda a\rangle} = \sqrt{\lambda\overline{\lambda}\langle a, a\rangle} = \sqrt{|\lambda|^2\langle a, a\rangle} = |\lambda|\sqrt{\langle a, a\rangle} = |\lambda|\|a\|.$$

Za pokazivanje svojstva (3) koristimo se svojstvima hermitske simetričnosti i aditivnosti skalarnog produkta. Vrijedi

$$|a + b|^2 = \langle a + b, a + b\rangle = \langle a, a\rangle + \langle a, b\rangle + \langle b, a\rangle + \langle b, b\rangle = \|a\|^2 + 2\mathrm{Re}\langle a, b\rangle + \|b\|^2.$$

Kako je $\mathrm{Re}\langle a, b\rangle \leq |\mathrm{Re}\langle a, b\rangle| \leq |\langle a, b\rangle|$, a koristeći rezultat Teorema 1.1.1 vrijedi i: $|\langle a, b\rangle| \leq \|a\|\|b\|$. □

U idućih nekoliko primjera konstruirat ćemo unitarne prostore ako imamo dane vektorske prostore. Dakle, konstruirat ćemo preslikavanje za koje ćemo provjeriti da zadovoljava svojstva iz definicije unitarnog prostora.

Primjer 1.1.1. *Na realnom vektorskom prostoru* $\mathbb{R}^n$ *za* $x = (x_1, \ldots, x_n)$ *i* $y = (y_1, \ldots, y_n)$ *iz* $\mathbb{R}^n$ *definiramo*

$$\langle x, y\rangle = x_1y_1 + \cdots + x_ny_n = \sum_{i=1}^{n} x_iy_i.$$

Da se zaista radi o unitarnom prostoru slijedi iz osnovih svojstava zbrajanja i množenja.

Primjer 1.1.2. *Na kompleksnom vektorskom prostoru* $\mathbb{C}^n$*, za* $x = (x_1, \ldots, x_n)$ *i* $y = (y_1, \ldots, y_n)$ *iz* $\mathbb{C}^n$ *stavimo*

$$\langle x, y\rangle = x_1\overline{y_1} + \cdots + x_n\overline{y_n} = \sum_{i=1}^{n} x_i\overline{y_i}.$$

Primjer 1.1.3. *Na vektorskom prostoru matrica* $M_{mn}(\mathbb{R})$*, za matrice* $A = \left[a_{ij}\right]$, $B = \left[b_{ij}\right] \in M_{mn}(\mathbb{R})$ *skalarno množenje definiramo kao sumu umnožaka svih odgovarajućih koeficijenata na istim pozicijama*

$$\langle A, B\rangle = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}b_{ij}.$$

Ovo skalarno množenje možemo zapisati i sažetije pomoću traga umnoška matrica

$$\langle A, B\rangle = tr\left(AB^T\right).$$

Definicija 1.1.5. *Skup vektora* $S = \{a_1, a_2, \ldots, a_k\}$ *unitarnog prostora* V *je ortonormiran ako je ortogonalan i* $\|a_i\| = 1$*, za sve* $i = 1, \ldots, k$*, to jest ako je*

$$\langle a_i, a_j \rangle = \delta_{i,j},$$

za sve $i, j = 1, \ldots, k$*. Posebno, ako je* S *baza prostora* V *i uz to ortonormiran skup,* S *nazivamo ortonormiranom bazom.*

U praksi je kod unitarnih prostora vrlo korisno imati ortonormiranu bazu s obzirom na to da se svaki račun koji obuhvaća skalarno množenje vektora zapisanih u toj bazi bitno pojednostavljuje. Naime, ako imamo vektore

$$x = \sum_{i=1}^{n} x_i a_i,\ y = \sum_{i=1}^{n} y_i a_i$$

koji su zapisani u nekoj općenitoj bazi nekog unitarnog prostora V dimenzije n, tada pri računanju skalarnog produkta

$$\langle x, y \rangle = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i \overline{y_j} \langle a_i, a_j \rangle$$

imamo n^2 pribrojnika. U slučaju da imamo ortonormiranu bazu $\{a_1, a_2, \ldots, a_n\}$, znamo da je $\langle a_i, a_j \rangle = \delta_{i,j}$ čime će se prethodna jednakost svesti na najviše n pribrojnika koji ne moraju biti jednaki 0

$$\langle x, y \rangle = \sum_{i=1}^{n} x_i \overline{y_i}.$$

Osim navedenog, važna karakteristika ortonormiranih baza je da vrlo jednostavno dolazimo do prikaza vektora u takvoj bazi, odnosno ako imamo ortonormiranu bazu $\{e_1, e_2, \ldots, e_n\}$, te vektor v tada nas zanimaju koeficijenti $v_1, v_2, \ldots, v_n$ takvi da vrijedi

$$v = v_1 e_1 + \cdots + v_n e_n.$$

U ovom slučaju nećemo morati rješavati sustav s n jednadžbi i n nepoznanica jer ukoliko prethodnu jednakost redom skalarno pomnožimo s $e_1, \ldots, e_n$ dobivamo da je općenito

$$v_i = v e_i,\ i = 1, \ldots, n.$$

Napomena 1.1.1. *Koristimo oznaku* $[\{a_1, a_2, \ldots, a_k\}]$ *kako bismo opisali prostor koji čini skup svih linearnih kombinacija vektora* $a_1, a_2, \ldots, a_k$*.*

Teorem 1.1.2. *Neka je V unitarni prostor te $\{a_1, a_2, \ldots, a_k\}$ linearno nezavisan podskup od V. Tada postoji ortonormirani podskup $\{e_1, e_2, \ldots, e_k\}$ u V takav da je*

$$[\{a_1, a_2, \ldots, a_k\}] = [\{e_1, e_2, \ldots, e_k\}],$$

za sve $j = 1, \ldots, k$.

Prethodni teorem može se dokazati korištenjem matematičke indukcije konstruirajući vektore $e_1, \ldots, e_n$ kao

$$e_1 = \frac{a_1}{\|a_1\|},$$

$$e_{j+1} = \frac{a_{j+1} - \sum_{i=1}^{j} \langle a_{l+1}, e_i \rangle e_i}{\|a_{j+1} - \sum_{i=1}^{j} \langle a_{l+1}, e_i \rangle e_i\|}, \ j = 1, \ldots, k-1$$

Dani postupak naziva se *Gram-Schmidtov postupak ortogonalizacije*. Sljedeći korolar direktna je posljedica prethodnog teorema.

Korolar 1.1.1. *U svakom konačnodimenzionalnom netrivijalnom prostoru postoji ortonormirana baza.*

U nastavku ćemo uvesti osnovne definicije vezane uz svojstva matrica kako bismo uspostavili temelje za daljnju analizu i primjenu.

Definicija 1.1.6. *Za matricu $A \in M_n(\mathbb{F})$ kažemo da je unitarna ako vrijedi $A^*A = AA^* = I_n$. U slučaju da je $\mathbb{F} = \mathbb{R}$ i $A^TA = AA^T = I$ kažemo da je matrica A ortogonalna.*

Definicija 1.1.7. *Matrična norma je svaka funkcija $\|\cdot\| : \mathbb{F}^{m\times n} \to \mathbb{R}$ koja zadovoljava sljedeća svojstva:*

(1) $\|A\| \geq 0$, *za sve $A \in \mathbb{F}^{m\times n}$, a jednakost vrijedi ako i samo ako je $A = 0$,*

(2) $\|\alpha A\| = |\alpha| \, \|A\|$, *za svaki $\alpha \in \mathbb{R}$, za sve $A \in \mathbb{F}^{m\times n}$,*

(3) $\|A + B\| \leq \|A\| + \|B\|$, *za sve $A, B \in \mathbb{F}^{m\times n}$*

Za matričnu normu ćemo reći da je konzistentna ako vrijedi

(4) $\|AB\| \leq \|A\| \, \|B\|$

kad god je matrični produkt AB definiran.

Od koristi će nam biti Frobeniusova matrična norma koju za $A \in \mathbb{C}^{m\times n}$ definiramo kao

$$\|A\|_F = (tr(A^*A))^{\frac{1}{2}} = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}.$$

Koristeći svojstvo nejednakosti trokuta koje vrijedi kod normi, može se pokazati da je Frobeniusova norma jedna konzistentna matrična norma.

Definicija 1.1.8. *Za simetričnu matricu $A \in \mathbb{C}^{n\times n}$ kažemo da je pozitivno semidefinitna, ako vrijedi*

$$x^TAx \geq 0,$$

za svaki $x \in \mathbb{R}^n$. Ako vrijedi stroga nejednakost kažemo da je A pozitivno definitna matrica.

1.2 Matrične faktorizacije

U ovom dijelu detaljno ćemo obraditi *QR* i *SVD* dekompoziciju, dvije matrične dekompozicije od velike važnosti za naš rad.

QR dekompozicija

Definicija 1.2.1. *Neka je zadana matrica $A \in \mathbb{R}^{m\times n}$ koja ima puni stupčani rang. QR faktorizacija je rastav oblika*

$$A = QR = Q\begin{bmatrix}R_0\\0\end{bmatrix},$$

pri čemu je $Q \in \mathbb{R}^{m\times m}$ ortogonalna matrica, te $\mathbb{R} \in \mathbb{R}^{n\times n}$ gornjetrokutasta matrica s pozitivnim elementima na dijagonali.

U praksi se za dobivanje QR faktorizacije koriste Givensove rotacije ili Householderovi reflektori. Givensove rotacije je pogodnije koristiti u slučaju da nam početna matrica ima puno nula jer rotacijama onda ciljano poništavamo pojedinačne elemente. S obzirom na to da će naša glavna matrica biti upravo takva efikasnije će biti koristiti Givensove rotacije. Možemo napomenuti da je Gram-Schmidtov postupak ortogonalizacije također opcija za dobivanje spomenute faktorizacije, no ipak numerički nestabilniji i posljedično manje precizniji.

Givensove rotacije

Krenimo od rotacije u ravnini, tada je Givensova rotacija dana u obliku matrice

$$G(\varphi) = \begin{bmatrix}\cos\varphi & -\sin\varphi\\ \sin\varphi & \cos\varphi\end{bmatrix}$$

koja svaki vektor $x \in \mathbb{R}^2$ rotira za kut φ u pozitivnom smjeru. Iz ovog je primjera jasno da imamo ortogonalnu matricu, a ovo svojstvo će biti prisutno u svakoj Givensovoj rotaciji. Koristeći istu logiku, u $\mathbb{R}^n$ definiramo Givensovu rotaciju u (i, j) kao

$$G(i,j,\varphi) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & \cos\varphi & \cdots & -\sin\varphi & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & \sin\varphi & \cdots & \cos\varphi & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}.$$

Djelujemo li matricom $G(i, j, \varphi)$ na neki vektor $x \in \mathbb{R}^n$, tada vektoru x mijenjamo samo i-tu i j-tu komponentu, odnosno označimo li s $y = G(i, j, \varphi)\, x$ dobiveni vektor, tada je

$$y_k = \begin{cases} x_i \cos\varphi - x_j \sin\varphi, & \text{ako } k = i, \\ x_i \sin\varphi + x_j \cos\varphi, & \text{ako } k = j, \\ x_k, & \text{ako } k \neq i, j \end{cases}.$$

Tražimo trigonometrijske vrijednosti $\cos\varphi$ i $\sin\varphi$ uz koje ćemo poništiti j-tu komponentu vektora y, dakle postavljamo uvjet

$$y_j = x_i \cos\varphi - x_j \sin\varphi = 0.$$

Uz smislenu pretpostavku da je $x_j \neq= 0$ imamo

$$\begin{aligned} x_i + \operatorname{ctg}\varphi x_j &= 0, \\ \operatorname{ctg}\varphi &= -\frac{x_i}{x_j}. \end{aligned}$$

Koristeći trigonometrijske identite $\sin^2\varphi + \cos^2\varphi = 1$ i $1 + \operatorname{ctg}^2\varphi = \frac{1}{\sin^2\varphi}$ možemo dobiti izraze za $\sin^2\varphi$ i $\cos^2\varphi$ preko komponenata x_i i x_j

$$\sin^2\varphi = \frac{x_j^2}{x_i^2 + x_j^2},\ \cos^2\varphi = \frac{x_i^2}{x_i^2 + x_j^2}.$$

Preostaje još za za odrediti predznake koje ćemo namjestiti tako da u konačnosti y_i bude pozitivan

$$\sin\varphi = -\frac{x_j}{\sqrt{x_i^2 + x_j^2}},\ \cos\varphi = \frac{x_i}{\sqrt{x_i^2 + x_j^2}}.$$

Na kraju dobivamo izraz i za y_i

$$\begin{aligned} y_i &= x_i \cos\varphi - x_j \sin\varphi \\ &= \frac{x_i}{\sqrt{x_i^2 + x_j^2}} x_i + \frac{x_j}{\sqrt{x_i^2 + x_j^2}} x_j \\ &= \sqrt{x_i^2 + x_j^2} \end{aligned}$$

Do QR faktorizacije matrice $A \in \mathbb{R}^{m\times n}$ onda možemo doći uzastopnim korištenjem Givensovih rotacija na svim ispoddijagonalnim netrivijalnim elementima. Najčešće se ove eliminacije rade po stupcima. Matrični element (i, j) ćemo poništiti Givensovom rotacijom $G_j\left(i-1, i, \varphi_{i,j}\right)$. Uzastopnim primjenjivanjem Givensovih rotacija dolazimo do gornjetrokutaste matrice R

$$\begin{aligned} &G_n\left(n, n+1, \varphi_{n,n+1}\right) \cdots G_n\left(m-1, m, \varphi_{m-1,m}\right) \cdots \\ &G_1\left(n-1, n, \varphi_{n-1,n}\right) \cdots G_1\left(m-1, m, \varphi_{m-1,m}\right) A := Q^{-1}A = R \end{aligned}$$

Matrica koju smo označili s Q^{-1} je ortogonalna i regularna matrica kao produkt ortogonalnih i regularnih. Inverz ortogonalne i regularne matrice je ponovno ortogonalna i regularna, pa označimo li s $Q = \left(Q^{-1}\right)^{-1}$ tada imamo

$$A = QR,$$

što je upravo QR faktorizacija početne matrice A.

SVD dekompozicija

Singularna dekompozicija matrice jedna je od najkorištenijih dekompozicija u numeričkoj linearnoj algebri. Sada smo spremni izreći spomenuti teorem koji ćemo ujedno i dokazati obzirom da ćemo ga iskoristiti u glavnom dijelu ovog rada.

Teorem 1.2.1 (**Singularna dekompozicija matrice**, [2]). *Ako je $A \in \mathbb{C}^{m\times n}$, tada postoje unitarne matrice $U \in \mathbb{C}^{m\times m}$ i $V \in \mathbb{C}^{n\times n}$, takve da je*

$$U^*AV = \Sigma,\ \Sigma = diag\left(\sigma_1, \sigma_2, \ldots \sigma_{min\{m,n\}}\right),$$

pri čemu vrijedi $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{min\{m,n\}} \geq 0$.
Brojeve $\sigma_1, \sigma_2, \ldots \sigma_{min\{m,n\}}$ zovemo singularne vrijednosti matrice A. Stupce matrice U zovemo lijevi, a stupce matrice V desni singularni vektori matrice A.

Dokaz. Kako je jedinična sfera u $\mathbb{C}^n$ ograničen i zatvoren skup, on je kompaktan, pa svaka neprekidna funkcija na njemu dostiže minimum i maksimum. Funkcija $f(x) = \|Cx\|_2$ je neprekidna, pa postoji jedinični vektor $v \in \mathbb{C}^n$, takav da je

$$\|Cv\|_2 = \max\{\|Cx\|_2 : \|x\|_2 = 1, x \in \mathbb{C}^n\}.$$

Ako je $\|Cv\|_2 = 0$, onda je $C = 0$ i faktorizacija u iskazu teorema je trivijalna uz $\Sigma = 0$ i s proizvoljnim matricama U i V reda m i n, respektivno.

Ako je $\|Cv\|_2 > 0$, stavimo $\sigma_1 = \|Cv\|_2$ i formirajmo jedinični vektor

$$u_1 = \frac{Cv}{\sigma_1} \in \mathbb{C}^m.$$

Nadopunimo u_1 s $m-1$ vektora do baze u $\mathbb{C}^m$ i onda primijenimo Gramm-Schimdtov proces ortogonalizacije, tako da dobijemo ortonormiranu bazu $u_1, \ldots, u_m$ za $\mathbb{C}^m$. Drugim riječima, dobili smo unitrarnu matricu $U_1 = [u_1, u_2, \ldots, u_m]$. Slično za $v_1 = v$ postoji $n-1$ ortonormiranih vektora $v_2, v_3, \ldots, v_n \in \mathbb{C}^n$, takvih da je matrica $V_1 = [v_1, v_2, \ldots, v_n]$ unitarna. Tada je

$$C_1 = U_1^* C V_1 = \begin{bmatrix} u_1^* \\ u_2^* \\ \vdots \\ u_m^* \end{bmatrix} \begin{bmatrix} Cv_1 & Cv_2 & \cdots & Cv_n \end{bmatrix} = \begin{bmatrix} u_1^* \\ u_2^* \\ \vdots \\ u_m^* \end{bmatrix} \begin{bmatrix} \sigma_1 u_1 & Cv_2 & \cdots & Cv_n \end{bmatrix}$$

$$= \begin{bmatrix} \sigma_1 & u_1^* C v_2 & \cdots & u_1^* C v_n \\ 0 & u_2^* C v_2 & \cdots & u_2^* C v_n \\ \vdots & \vdots & \ddots & \vdots \\ 0 & u_m^* C v_2 & \cdots & u_m^* C v_n \end{bmatrix} = \begin{bmatrix} \sigma_1 & z^* \\ 0 & C_2 \end{bmatrix},$$

gdje je $z \in \mathbb{C}^{n-1}$, $C_2 \in \mathbb{C}^{(m-1)\times(n-1)}$. Za jedinični vektor

$$y = \frac{1}{\sqrt{\sigma_1^2 + z^* z}} \begin{bmatrix} \sigma_1 \\ z \end{bmatrix},$$

zbog unitarne invarijantnosti euklidske norme vrijedi

$$\|C(V_1 y)\|_2^2 = \|(U_1^* C V_1) y\|_2^2 = \|C_1 y\|_2^2 = \frac{\left(\sigma_1^2 + z^* z\right) + \|C_2 z\|_2^2}{\sigma_1 + z^* z} \geq \sigma_1^2 + z^* z,$$

a ovo je striktno veće od σ_1^2 ako je $z \neq 0$. Pošto je to u surpotnosti s maksimalnošću od σ_1, zaključujemo da je $z = 0$. Stoga je

$$C_1 = U_1^* C V_1 = \begin{bmatrix} \sigma_1 & 0 \\ 0 & C_2 \end{bmatrix},$$

Sada ponavljamo isti postupak za matricu $C_2 \in \mathbb{C}^{(m-1)\times(n-1)}$. Na taj način dobivamo unitarne matrice U i V kao produkt unitarnih matrica koje su dobijene nakon svakog koraka. Ako je $m \geq n$ taj postupak vodi do dijagonale matrice Σ.

Ako je $m \leq n$, u zadnjem koraku radimo s matricom $C_m \in \mathbb{C}^{1\times(n-m+1)}$. Za C_m postoji unitarna matrica takva da je

$$C_m V_m = \begin{bmatrix} \|C_m\|_2 & 0 & \dots & 0 \end{bmatrix}$$

pa će lijeve i desne komponente unitarne matrice u zadnjem koraku biti $U_m = I_1$ i V_m, respektivno. □

Sljedeći paragraf do Napomene 1.2.1 je izveden iz [7].

Neka je $A = U\Sigma V^T$ singularna dekompozicija matrice $A \in \mathbb{R}^{m\times n}$, tada o stupcima matrice V možemo razmišljati kao o svojstvenim vektorima matrice $C := A^T A$ jer je

$$A^T A = \left(U\Sigma V^T\right)^T U\Sigma V^T = V\Sigma^2 V^T, \tag{1.1}$$

gdje smo u drugoj jednakosti koristili činjenicu da je U unitarna. Nadalje, množenjem zdesna matricom V imamo

$$A^T AV = V\Sigma^2 \implies CV = V\Sigma^2.$$

Također, možemo uočiti da su svojstvene vrijednosti matrice C upravo kvadrirane singularne vrijednosti početne matrice A. Označimo li s $a_1, a_2, \dots, a_n$ stupce matrice A, tada matrica C na mjestu (i, j) ima skalarni produkt $\langle a_i, a_j\rangle$. Iz ovog razloga o matrici C možemo razmišljati kao o korelacijskoj matrici između stupaca. O stupcima matrice U možemo razmišljati na sličan način ako definiramo matricu $D = AA^T$. Svojstveni vektori matrice D će odgovarati stupcima matrice U, dok će svojstvene vrijednosti matrice D biti iste kao i kod matrice C. Možemo primijetiti da su ovako definirane matrice C i D pozitivno definitne što će implicirati da imamo realne pozitivne svojstvene vrijednosti.
Napomenimo da ovo nije način na koji se trebaju određivati matrice U, V i Σ, već samo služi za bolje razumijevanje.

Napomena 1.2.1. *Koristit ćemo oznake $U(:, 1{:}r)$ i $U(1{:}r, :)$ što će predstavljati odabir prvih r stupaca odnosno odabir prvih r redaka matrice U.*

Označimo li s $r = \min(m, n)$, tada matricu A možemo zapisati koristeći $\hat{U} = U(:, 1{:}r)$, $\hat{\Sigma} = \operatorname{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$ te $\hat{V} = V(:, 1{:}r)$ kao

$$A = \hat{U}\hat{\Sigma}\hat{V}^T. \tag{1.2}$$

Ovakva dekompozicija je ekonomičnija u odnosu na dekompoziciju iz Teorema 1.2.1 iz razloga što nam je potrebno manje memorije za spremanje matrica $\hat{U}$,$\hat{\Sigma}$,$\hat{V}$, a ne gubimo nikakve korisne informacije o strukturi matrice A.

Osim spomenute Frobeniusove matrične norme, od značaja u dokazu sljedećeg teorema bit će nam matrična 2 norma koja se definira preko najveće singularne vrijednosti matrice A

$$||A||_2 = \sigma_{max}(A).$$

U sljedećem teoremu vidimo važnost i jednu direktnu primjenu singularne dekompozicije.

Teorem 1.2.2 (**Ekhard, Young, Mirsky**). *[2] Neka je $A = U\Sigma V^*$ singularna dekompozicija matrice $A \in \mathbb{C}^{m\times n}$ ranga r. Neka je $k \leq r$ i*

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^*.$$

Tada je

$$\min_{\text{rang}(K)=k} ||A - K||_F = ||A - A_k||_F = \sigma_{k+1}. \tag{1.3}$$

Dokaz. Koristeći SVD dekompoziciju uvedenu u Teoremu 1.2.1 znamo da je

$$U^* C_k V = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k, 0, \ldots, 0),$$

pa nadalje imamo

$$||C - C_k||_2 = ||U^*(C - C_k)V||_2 = \text{diag}(0, \ldots, 0, \sigma_{k+1}, \ldots, \sigma_{min\{m,n\}}) = \sigma_{k+1}.$$

Ovime smo dokazali drugu jednakost u (1.3), pokažimo da vrijedi i prva.

Uzmimo proizvoljnu matricu $K \in \mathbb{C}^{m\times n}$ ranga k. Budući da je slika linearnog operatora K k-dimenzionalni potprostor prostora $\mathbb{C}^n$ možemo primijeniti teorem o rangu i defektu. Iz toga zaključujemo da je jezgra operatora $n-k$ dimenzionalni potprostor. Uzmimo ortonormiranu bazu $x_1, \ldots, x_{n-k}$ koja razapinje jezgru linearnog operatora. Vrijedi sljedeće

$$[\{x_1, \ldots, x_{n-k}\}] \cap [\{v_1, \ldots, v_{k+1}\}] \neq \{0\},$$

jer bismo u suprotnom mogli izgenerirati prostor veće dimenzije nego u odnosu na početni $\mathbb{C}^n$ što naravno ne možemo. Ovo nam daje opravdanje da pronađemo jedinični vektor z koji pripada presjeku. Kako vektor z pripada presjeku, posebno pripada i jezgri tako da je $Kz = 0$. Također, zaključujemo da je

$$Cz = \sum_{i=1}^{k+1} \sigma_i (v_i^* z) u_i.$$

Sada imamo sljedeći niz nejednakosti

$$||C - K||_2^2 \geq ||(C - K)z||_2^2 = ||Cz||_2^2 = \sum_{i=1}^{k+1} \sigma_i^2 |v_i^* z|^2 \geq \sigma_{k+1}^2.$$

Zadnje nejednakost vrijedi jer se radi o konveksnoj sumi brojeva σ_i, $1 \leq i \leq k+1$, a konveksnost sume slijedi iz činjenice da je $0 \leq |v_i^* z|^2 \leq 1$ i

$$\sum_{i=1}^{k+1} |v_i^* z|^2 = \|z\|_2^2 = 1.$$

□

Dakle, ukoliko želimo pronaći matricu nižeg ranga koja najbolje aproksimira početnu po Frobeniusovoj normi to možemo učiniti direktno ako imamo određenu singularnu dekompoziciju.

1.3 Matrična reprezentacija dokumenata

Sadržaj ove sekcije je rađen prema [4]. U cilju nam je niz dokumenata prikazati nekom matricom A. Matricu A ćemo nazivati dokument matricom, gdje će svaki stupac te matrice predstavljati jedan dokument. Najprije ćemo iz dokumenata izvući sve riječi koje se pojavljuju. Obzirom da ne želimo raditi razliku između riječi koje počinju malim, odnosno velikim slovom sve ćemo riječi staviti da kreću malim slovom. Završetak riječi smatramo da se dogodio u trenutku kada imamo pojavu praznog stringa, odnosno *spacea*. Iz tog razloga sljedeći korak će biti uklanjanje pravopisnih znakova poput točke, zareza, dvotočke i sličnih. Zatim uklanjamo riječi koje ne obogaćuju teskt korisnim informacijama. Ovakve riječi nazivat ćemo *stopwordsima*. Imamo dostupno 1.460 dokumenata gdje je svaki dokument reprezentiran s nekoliko rečenica, te 112 upita koji su također sastavljeni u obliku rečenica. Primjeri nekih, naravno ne svih stopwordsa koje uklanjamo su sljedeći:

> *a, the, able, about, above, after, againt, against, all, allow, almost, alone, already, also, although, always, am, among, amongst, an, and, another, any, anybody, anyhow, anyone, anything, anyway, anyways, anywhere, apart, appear, appropriate, are, aren't, around, as, aside, ask, ...*

Naposljetku, želimo objediniti različite riječi koje upućuju na isti pojam u jednu riječ. Konkretno ovo bi mogli napraviti tako da kažemo da 2 različite riječi predstavljaju istu riječ ukoliko imaju isti korijen riječi. Ovaj način grupiranja riječi u jednu riječ nazivamo *stemming*. Prije stemminga imali smo 10.625 jedinstvenih riječi, nakon primjene stemminga pali smo na 6.819 različitih korijena riječi. Pogledajmo u sljedećem primjeru što smo dobili kao rezultat stemminga.

Primjer 1.3.1. *U Tablici 1.1 prikazane su neke grupe riječi koje su reprezentirane jednom ključnom riječi.*

bez stemminga	**sa stemmingom**
cover	cover
covers	
covering	
covered	
originally	origin
origins	
originator	
original	
originate	
originated	
originals	
origin	
originating	
originators	adapt
market	market
marketing	
build	build
building	
buildings	

Tablica 1.1: Prikaz logike dobivanja ključnih riječi

U ovom postupku smanjenja riječi treba pripaziti da ne dođe do gubitka sadržaja. Ovime smo došli do finalnog skupa svih riječi koje nazivamo ključne riječi. Ovaj dio pretprocesiranja podataka napravljen je u Pythonu 3.11, dok ćemo nastavak računa raditi u Octave-u 8.3 i Matlabu 23.2.

Ono s čime nastavljamo jest s pretpostavkom da imamo m različitih dokumenata i n ključnih riječi. Prezentirat ćemo 3 različite ideje kako bismo mogli popuniti dokument matricu A. Najtrivijalniji način bi bio popuniti je na indikatorski način, gdje bismo na mjestu $a_{i,j}$ imali vrijednost 1 ako postoji pojavljivanje i-te ključne riječi u j-tom dokumentu, dok bismo u suprotnom stavili vrijednost 0. Ovakvo popunjavanje nije najsretnije jer očito ne dajemo na važnosti broju pojavljivanja riječi unutar dokumenta.

Iduća ideja bila bi prebrojati pojavljivanja ključnih riječi u svakom od dokumenata, odnosno imati frekvencije pojavljivanja ključnih riječi unutar dokumenta. Međutim, ovakvo dodijeljivanje težina također nije najbolje rješenje jer bi tada težine bile dodijeljene samo u odnosu na riječi koje se pojavljuju u tom dokumentu. Htjeli bismo težine odrediti na skupu svih ključnih riječi, jer želimo veću težinu dati onim riječima koje se rijetko pojavljuju u

dokumentima te manju težinu onima koji se često pojavljuju u dokumentima.

Primjer 1.3.2. *U Tablici 1.2 prikazane su frekvencije pojavljivanja najučestalijih riječi u dokumentima. U slučaju pojavljivanja takvih riječi u dokumentu ili upitu njima želimo smanjiti važnost jer nam ta riječ ne ukazuje dovoljno o samom sadržaju. A i u slučaju da ukazuje, možemo zaključiti da na temelju te riječi u većini slučajeva nećemo uvidjeti razliku između dokumenata.*

Riječ	**Zastupljenost**
use	43,9%
inform	42,05%
librari	35,75%
system	33,84%
develop	25,07%
research	23,42%
studi	23,36%
one	21,85%
present	21,51%

Tablica 1.2: Relativna frekvencija pojavljivanja najučestalijih korijena riječi u dokumentima

Konkretno, riječ *use* nalazi se u gotovo svakom drugom dokumentu te bismo htjeli smanjiti značajnost pojavljivanja te riječi. Odnosno, želimo dati težinu svakoj ključnoj riječi u ovisnosti o tome u koliko dokumenata se ista pojavila.

Ovo bismo mogli postići tako da definiramo sljedeće preslikavanje $h : (1, \ldots, m) \times (1, \ldots, n) \to \mathbb{R}$

$$h(i, j) = f_{ij} \log (n/n_i), \tag{1.4}$$

gdje je m broj ključnih riječi, n broj dokumenata, f_{ij} broj pojavljivanja i-te riječi u j-tom dokumentu te n_i broj dokumenata u kojima možemo pronaći riječ i. Ovako definirano preslikavanje h će biti nenegativno i jednako nuli u slučaju da se riječ pojavljuje u svim dokumentima. Ovime smo konstruirali i treći način kako možemo popuniti dokument matricu. Obzirom da ćemo imati matricu velike dimenzije, od koristi će nam biti čim veći broj pojavljivanja nula kako bismo si računanje učinili efikasnijim. Kroz rad ćemo usporediti koja konstrukcija dokument matrice daje najbolje rezultate.

1.4 Udaljenost između dokumenata

Nakon što smo svaki dokument prikazali jednim stupcem, odnosno vektorom sada bismo htjeli da za zadani upit q pronađemo sve relevantne dokumente. Zadani upit također možemo prikazati pomoću jednog stupca, gdje bi retci (ključne riječi) odgovarali retcima matrice kojom smo reprezentirali sve dokumente. Dakle, ono što bi prvo trebalo napraviti je očitati ključne riječi iz upita. Za određivanje sličnih dokumenata možemo promatrati kosinus kuta koji vektor q zatvara sa svakim od dokumenata koristeći definiciju skalarnog produkta. Za danu razinu tolerancije $tol > 0$ možemo odrediti sve dokumente a_j tako da vrijedi

$$\cos\left(\angle\left(q, a_j\right)\right) = \frac{q^T a_j}{\|q\|_2 \|a_j\|_2} > tol. \tag{1.5}$$

Najprije primijetimo da će svi dokumenti, pa tako i upit imati nenenegativne vrijednosti u svim retcima što će doprinijeti da je

$$\angle\left(q, a_j\right) \in \left[0, \frac{\pi}{2}\right] \tag{1.6}$$

iz čega opravdavamo smislenost pronalaženja relevantnih dokumenata preko skalarnog produkta. Nadalje, definirat ćemo dva mjerenja kojima ćemo utvrđivati uspješnosti pronalaska. Prvim mjerenjem ćemo gledati omjer broja vraćenih relevantnih dokumenata D_r i ukupnog broja vraćenih dokumenata D_t. Ovako definiran broj nazivat ćemo preciznost. Drugim mjerenjem će nas zanimati broj dobiven omjerom broja vraćenih relevantnih dokumenata D_r i stvarnog broja relevantnih dokumenata u čitavoj bazi N_r. Taj broj ćemo nazivati pokrivenost. Dakle, imat ćemo

$$P = \frac{D_r}{D_t},$$

$$R = \frac{D_r}{N_r}$$

gdje će P označavati preciznost, a R pokrivenost.

Možemo primijetiti da smanjenjem tolerancije u izrazu (1.5) ćemo kao rezultat dobiti manju preciznost i veću pokrivenost.
Ovime smo definirali osnovne pojmove koji će nam biti potrebni u nastavku rada.

Poglavlje 2

Numeričke metode u analizi teksta

2.1 Osnovna metoda

Ovo je poglavlje motivirano [6] i [4], odakle su velikim dijelom preuzeti iskazi i njihovi dokazi. U ovom potpoglavlju izvest ćemo osnovne rezultate koristeći ideju prezentiranu u Potpoglavlju 1.4. Ovi rezultati će nam biti orijentir u budućnosti kada ćemo raditi naprednije metode da vidimo što smo i koliko postigli s kojom metodom. Uzet ćemo jedan konkretan upit za koji ćemo odrediti razine preciznosti i pokrivenosti za sve veće tolerancije. Podsjetimo se da u našoj bazi podataka [1] radimo s 6.820 ključnih riječi, 1.460 dokumenata i 112 upita. Dokument matricu ćemo popuniti na način opisan jednadžbom (1.4). Nakon popunjavanja, dokument matricu ćemo normirati na način da svaki stupac bude duljine 1 po 2 - normi.

Upit za koji ćemo računati pokrivenosti i preciznosti je sljedeći:

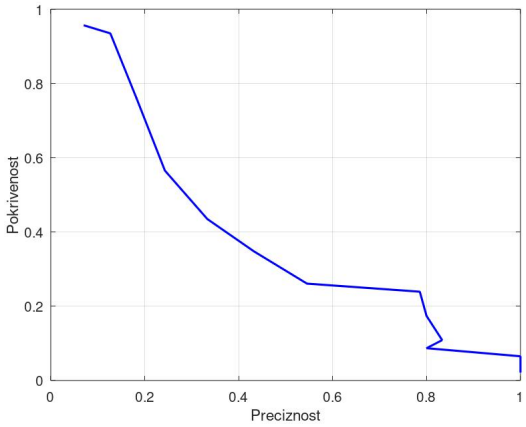
What problems and concerns are there in making up descriptive titles? What difficulties are involved in automatically retrieving articles from approximate titles? What is the usual relevance of the content of articles to their titles?

Kako bi upit bio usporediv s dokumentima on mora proći iste transformacije koju je prošla i dokument matrica. Upit želimo zapisati u istoj formi kao i dokument, odnosno preko onih ključnih riječi koje su se javljale u dokumentima. Postojat će neke ključne riječi koje su se javile unutar upita, a da se nisu javile unutar skupa ključnih riječi kojima su opisani dokumenti. Međutim takve riječi nisu bile dio niti jednog dokumenta pa nam te riječi neće biti od koristi. Upit ćemo na kraju i normirati kako bismo imali efikasnije računanje.

Rekli smo da ćemo dokument proglasiti referentnim ako je kosinus kuta između dokumenta i upita dovoljno velik. Kako smo dokumente a_j i upit q normirali imamo

$$\cos\left(\angle\left(q, a_j\right)\right) = q^T a_j.$$

Primijetimo da je sada za odrediti kosinus kuta dovoljno odrediti skalarni produkt, odnosno imati točno onoliko zbrajanja i množenja koliko imamo ključnih riječi. Sama složenost računanja će naravno ovisiti i o broju dokumenata te koliko imamo pozitivnih vrijednosti u zapisu dokument matrice i upita. Na Slici 2.1 prikazujemo odnos preciznosti i pokrivenosti za spomenuti upit kako povećavamo toleranciju. Možemo primijetiti da povećanjem



Slika 2.1: Odnos preciznosti i pokrivenosti za jedan upit

tolerancije istovremeno doprinosimo sve većoj preciznosti i sve manjoj pokrivenosti što je očekivano. Treba pripaziti da potpuna pokrivenost doprinosi vraćanju prevelikog broja dokumenata što očitavamo iz preciznosti, isto tako potpuna preciznost doprinosi iznimno niskoj pokrivenosti.

2.2 LSI metoda

Nastavljamo s istom tematikom, zanima nas i dalje kako se pokrivenost i preciznost mijenjaju za sve veću toleranciju, ali sada koristeći LSI metodu. Glavna pretpostavka LSI metode je da postoji skrivena semantička struktura u podacima, skrivena iz razloga što postoje raznolike riječi koje opisuju sličan pojam. Metoda se bazira na projiciranju podataka na nižedimenzionalni prostor pri čemu se gube manje bitni detalji, dok zadržavamo semantičku relevantnost podataka. To također omogućava da se grupiraju slični dokumenti čak i ako ne koriste iste riječi, već semantički srodne izraze. Projiciranje možemo napraviti na više načina, no nama će u ovom potpoglavlju zanimljiv slučaj biti upravo pronalazak matrice K manjeg ranga koja će po Frobeniusovoj matričnoj normi biti najbliža početnoj matrici A. Konstrukciju ovakve matrice K nam omogućava Teorem 1.2.2. Konkretno, dokument matricu $A \in \mathbb{R}^{m\times n}$ ćemo zapisati koristeći memorijski manje zahtjevnu SVD

dekompoziciju

$$A = \hat{U}\hat{\Sigma}\hat{V}^T,$$

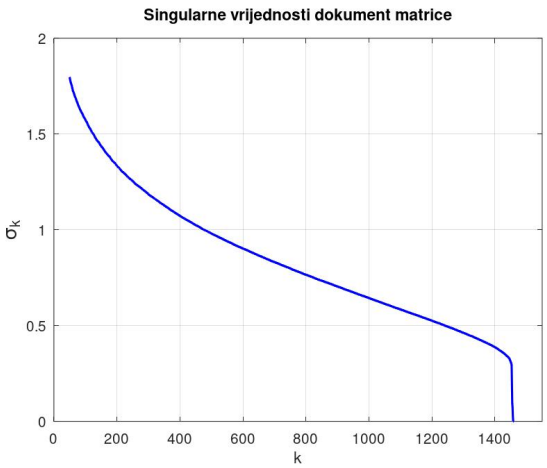
gdje koristimo oznake iz (1.2). Tada za singularne vrijednosti vrijedi $\sigma_i > 0$, za $i = 1, \ldots, r$, te $\sigma_j = 0$ za $j = r+1, \ldots, k$. Za $k < r$ možemo dobiti matricu A_k ranga k kao

$$A_k := U(:, 1{:}k)\,\Sigma(1{:}k, 1{:}k)\,V(:, 1{:}k)^T = U_k\Sigma_k V_k^T =: U_k H_k.$$

Da se zaista radi o matrici ranga k vidi se iz činjenice da prvih k stupaca matrice U čine jedan ortogonalan skup vektora kojim ćemo približno opisati svaki od dokumenata. Neka je $H = (h_1, h_2, \ldots, h_k)$ zapis matrice H preko njenih stupaca. Kako je $A \approx U_k H_k$, imamo $a_j \approx U_k h_j$, tj.u stupcu h_j možemo očitati skalare za približan zapis dokumenta j u novoj ortogonalnoj bazi. Kako bi i dalje mogli promatrati izraz (1.5), moramo i upit q projicirati na isti nižedimenzionalni prostor u koji smo smjestili i dokumente. Dakle, novi reducirani upit zapisan u novoj bazi bit će oblika $q_k = U_k^T q$. Zapise dokumenata i upita ćemo ponovno normirati na duljinu 1 pa će izraz (1.5) postati

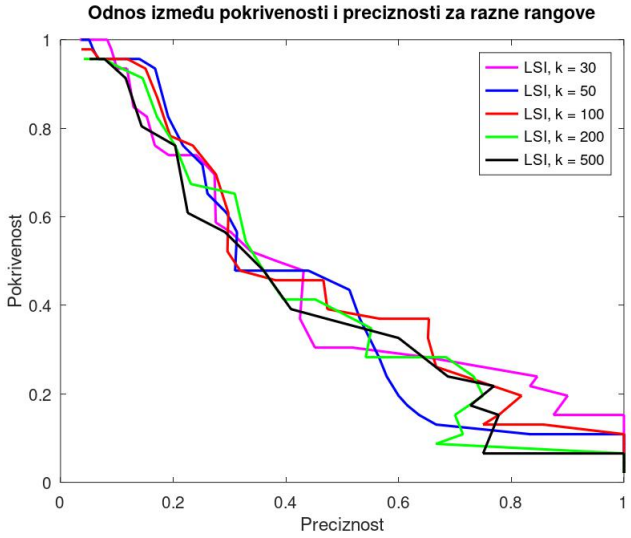
$$\cos\theta_j = q_k^T h_j.$$

Ovime smo postigli da se računanje odvija u k-dimenzionalnom prostoru čime smanjujemo vrijeme potrebno za određivanje udaljenosti između dokumenta i upita. Zanimat će nas koja će biti zadovoljavajuća dimenzija aproksimativne dokument matrice, odnosno rang k za kojeg bismo postigli zadovoljavajuće rezultate. Rezultate ćemo interpretirati preko već definiranih pojmova preciznosti i pokrivenosti. Grafički na Slici 2.2 prikazujemo pad singularnih vrijednosti kako bismo pokušali očitati koji bi nam rang k bio zadovoljavajući.



Slika 2.2: Prikaz pada singularnih vrijednosti

Možemo primijetiti da nemamo velike skokove u nizu singularnih vrijednosti tako da nam je teško iz ovog grafa očitati optimalni rang. Upravo iz tog razloga ćemo za optimalni rang uzeti onaj za koji dobivamo najbolje rezultate. Na Slici 2.3 prikazujemo odnos pokrivenosti i preciznosti za aproksimativne matrice rangova: 50, 100, 200 i 500. Na slici ne



Slika 2.3: Kretanje pokrivenosti i preciznosti za različite rangove aproksimativne dokument matrice

vidimo značajniju razliku između rezultata dobivenih uzimanjem različitih rangova. Kada bismo se fokusirali na rang 50 i rang 500, zanimljivo za primijetiti je da je plava krivulja koja predstavlja rang 50 u više navrata iznad crne koja predstavlja rang 500. Iz slike možemo očitati da najstabilnije rezultate dobivamo za rozi graf što predstavlja rang 30. Osim navednog razloga, bitna nam je i efikasnost, odnosno brzina računanja za aproksimativnu dokument matricu. Zato za aproksimativnu dokument matricu uzimamo da je ranga 30 te uspoređujemo rezultat u odnosu na osnovnu metodu. Ono što nas isto zanima jest koliko je aproksimativna matrica udaljena od početne dokument matrice u Frobeniusovoj normi, odnosno koliku aproksimacijsku grešku imamo. Tu informaciju možemo najlakše uvidjeti iz relativne greške

$$\frac{\|A - A_{30}\|_F}{\|A\|_F} \approx 0.30.$$

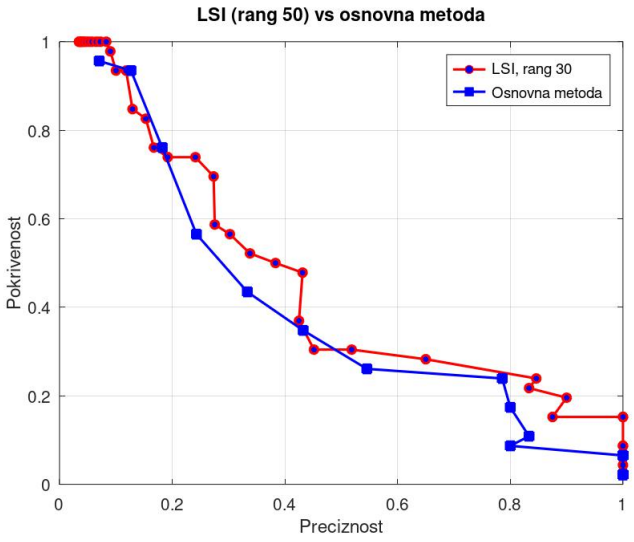
Vidimo da greška nije mala, no unatoč tome dobivamo bolje rezultate nego u osnovnoj metodi. Iz ovog razloga se možemo zapitati jesmo li odabrali dobar način za odabir baze koja će najbolje reprezentirati početnu dokument matricu. Mi smo bazu odabrali koristeći singularne vektore, odnosno onu bazu koja stoji uz najveće singularne vrijednosti. S druge

strane, obzirom na to da smo dobili zadovoljavajuće rezultate smislenije je za zaključiti da Frobeniusova norma možda nije najbolja mjera za gledati koliko smo uspješni s aproksimacijom. Ono što je zanimljivo za vidjeti koji su to najvažniji smjerovi u prostoru dokumenata, te uočiti najveće komponente po apsolutnoj vrijednosti tih smjerova. Tako za prva 2 najdominantnija smjera $U(:,1)$ i $U(:,2)$ gledamo najveće komponente po apsolutnoj vrijednosti. Koristeći upit u Octaveu-u `find`$(abs(U(:,k)) > 0.15)$ dolazimo do traženih komponenti za $k = 1, 2$, pa uparivanjem indeksa dolazimo i do ključnih riječi prikazanih u Tablici 2.1.

U(:, 1)	U(:, 2)
countless	cryptarithmet
conflict	newark
electron	conflict
patient	proverbi
	belong
	zeroon

Tablica 2.1: Tablica najvažnijih ključnih riječi

Usporedba između osnovne metode i LSI metode prikazana je na Slici 2.4.



Slika 2.4: Osnovna metoda nasuprot LSI metode

Možemo uočiti da dobivamo bolje rezultate LSI metodom uzimajući rang 30 u odnosu na osnovnu metodu.

S obzirom na to koliko imamo efikasnije računanje u LSI metodi, a dobivamo rezultate koji su i bolji od osnovne metode zaključujemo da u slučaju ovog upita LSI metoda nam je puno bolja. Konkretno, vrijeme potrebno za određivanje preciznosti i pokrivenosti je skoro pa 5 puta manje od osnovne metode. Osnovnoj metodi je potrebno okvirno 2.4 sekunde, dok je LSI metodi za rang 30 potrebno 0.51 sekundi. Ovu analizu trebamo uzeti s velikom dozom opreznosti jer je rađena na samo jednom upitu.

2.3 Klasteriranje

Klasteriranje je metoda analize podataka koja se koristi za grupiranje skupa objekata ili podataka u tzv. klastere na temelju sličnosti ili zajedničkih karakteristika. Cilj klasteriranja je otkrivanje skrivenih obrazaca ili struktura. U klasteriranju, mjera udaljenosti je način kvantificiranja sličnosti ili razlike između objekata unutar skupa podataka. Klasteriranje se koristi za grupiranje sličnih objekata, a mjera udaljenosti određuje koliko su ti objekti međusobno udaljeni što direktno utječe na formiranje klastera. Najčešće korištene mjere udaljenosti su: Euklidska, Manhattanska i kosinusna. Mi ćemo pri klasteriranju dokumenata koristiti Euklidsku udaljenost, no ovdje moramo biti oprezni. Naime, veliku ulogu pri računanju Euklidske udaljenosti ima duljina vektora tako da je važno prije samog klasteriranja normirati dokumente da budu duljine 1 jer u našem slučaju važniji nam je smjer vektora nasuprot njegovoj duljini. Opravdajmo korištenje Euklidske udaljenost, odnosno pokažimo relaciju koja povezuje edukidsku udaljenost i kosinusnu udaljenost. Krenimo od pretpostavke da imamo dva vektora a i b takvih da vrijedi $||a||_2 = ||b||_2 = 1$, tada imamo

$$||a - b||_2^2 = ||a||_2^2 - 2ab + ||b||_2^2 = 2 - 2ab.$$

Korjenovanjem dobivamo izraz za izračun Euklidske udaljenosti između dvaju vektora

$$d(a, b) = \sqrt{2 - 2ab}.$$

Iz ovoga primjećujemo da Euklidska udaljenost ovisi samo o kutu između vektora, a ne i o njihovoj duljini i to na način da imamo sve manju Euklidsku udaljenost što je kut između vektora manji.

Smisleno za pretpostaviti je da u slučaju kolekcije dokumenata postoje dokumenti sličnog sadržaja. Htjeli bismo od skupa dokumenata učiniti jednu particiju, gdje ćemo elemente te particije zvati klasteri. Sadržajno slične dokumente želimo grupirati u isti klaster, te bismo za svaki klaster htjeli imati njegovog predstavnika kojeg ćemo nazivati centroid. Jedan način za definiranje centroida jest uzeti prosječan dokument tog klastera. Broj klastera može biti proizvoljan, te će broj klastera predstavljati kardinalitet baze kojom ćemo aproksimativno reprezentirati dokumente. Dakle, ovdje je idejno slično kao u LSI,

želimo smanjiti dimenziju, no sada ćemo bazu za prostor dokumenata konstruirati koristeći centroide klastera.

Neka su zadani dokumenti $a_1, a_2, \ldots, a_n \in \mathbb{R}^m$, te k željeni broj klastera. Želimo odrediti particiju $\pi = (\pi_1, \ldots, \pi_k)$ koja će biti takva da minimizira funkciju cilja

$$Q(\pi) = \sum_{i=1}^{k} \sum_{s=1}^{s_i} \|a_s^{(i)} - m_i\|_2^2, \tag{2.1}$$

gdje je s_i kardinalitet od $\pi_i = \{a_1^{(i)}, \ldots, a_{s_i}^{(i)}\}$, te $m_i = \frac{i}{s_i} \sum_{s=1}^{s_i} a_s^{(i)}$ centroid i-te particije. Dakle, Q minimiziramo po svim k-particijama skupa od n elemenata. U nastavku prezentiramo iterativni algoritam k-sredina koji nam iz iteracije u iteraciju konstruira novu particiju i to takvu da je vrijednost funkcije cilja iz (2.1) sve manja.

Algoritam k-sredina

(1) Zadaj početnu particiju $\pi_j^{(0)}$ i izračunaj početne centroide $m_j^{(0)}$, gdje je $j = 1, \ldots, k$ i $t = 0$

(2) Svaki dokument a_i neka potraži sebi najbliži centroid i pridruži se novom skupu, odnosno kreiraj novu particiju takvu da

$$\pi_j^{(t+1)} = \left\{ a \in \{a_1, \ldots, a_n\} : \|a - m_j^{(t)}\|_2 = \min_{\ell \in 1,\ldots,k} \|a - m_\ell\|_2 \right\}$$

(3) Izračunaj centroide $m_j^{(t+1)}$ skupova $\pi_j^{(t+1)}$, $j = 1, \ldots, k$.

(4) Provjeri kriterij zaustavljanja dan na vrijednost funkcije cilja. Ako je kriterij zadovoljen, kao konačnu partiicju uzmi $\pi_j^{(t+1)}$, $j = 1, \ldots, k$. i stani. U suprotnom, $t = t + 1$ i vrati se na 2.korak

Pokazujemo tvrdnju od maloprije da će ovakvim konstrukcijama particija vrijednost ciljne funkcije biti sve manja.

Propozicija 2.3.1. *[3] Vrijede sljedeće nejednakosti* $Q\left(\pi^{(t)}\right) \geq Q\left(\pi^{(t+1)}\right) \geq 0$*, gdje je* Q *funkcija cilja iz* (2.1)*, te particija* π *konstruirana algoritmom k-sredina.*

Dokaz. Imamo sljedeći niz jednakosti i nejednakosti

$$\begin{aligned} Q\left(\pi^{(t)}\right) &= \sum_{j=1}^{k} \sum_{a \in \pi_j^{(t)}} \|a - m_j^{(t)}\|_2^2 = \sum_{j=1}^{k} \sum_{i=1}^{k} \sum_{a \in \pi_j^{(t)} \cap \pi_i^{(t+1)}} \|a - m_j^{(t)}\|_2^2 \\ &\geq \sum_{j=1}^{k} \sum_{i=1}^{k} \sum_{a \in \pi_j^{(t)} \cap \pi_i^{(t+1)}} \|a - m_i^{(t)}\|_2^2 = \sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{a \in \pi_j^{(t)} \cap \pi_i^{(t+1)}} \|a - m_i^{(t)}\|_2^2 \\ &= \sum_{i=1}^{k} \sum_{a \in \pi_i^{(t+1)}} \|a - m_i^{(t)}\|_2^2 \geq \sum_{i=1}^{k} \sum_{a \in \pi_i^{(t+1)}} \|a - m_i^{(t+1)}\|_2^2 = Q\left(\pi^{(t+1)}\right), \end{aligned}$$

gdje zadnja nejednakost proizlazi iz činjenice:

$$\min_{c \in \mathbb{R}^k} \sum_{a \in \pi_i^{(t+1)}} \|a - c\|_2^2 = m_i^{(t+1)}.$$

□

Pretpostavimo da nakon algortima *k-sredina* imamo matricu $C_k \in \mathbb{R}^{m \times k}$ centroida kojoj normiramo stupce tako da budu duljine 1. Tada nam ta matrica može predstavljati aproksimativni prostor dokumenata. Sada bismo trebali odrediti aproksimativni zapis dokumenata u takvoj bazi, odnosno matricu $G_k \in \mathbb{R}^{m \times n}$ takvu da matrica $C_k \hat{G}_k$ po Frobeniusovoj normi bude čim bliža početnoj matrici A, odnosno

$$\min_{\hat{G}_k} \|A - C_k \hat{G}_k\|_F.$$

Ovo je diskretni matrični problem najmanjih kvadrata, pa će nam od koristi biti najprije dobiti ortonormiranu bazu za stupce matrice C_k što možemo koristeći QR dekompoziciju čime imamo

$$C_k = P_k R,\ P_k \in \mathbb{R}^{m \times k},\ R \in \mathbb{R}^{k \times k}.$$

Time dobivamo sljedeće

$$\min_{G_k} \|A - P_k G_k\|_F. \tag{2.2}$$

Zapišemo li svaki stupac matrice $A - P_k G_k$ odvojeno možemo uočiti da je matrični problem najmanjih kvadrata ekvivalentan s n nezavisnih standardnih problema najmanjih kvadrata

$$\min_{g_j} \|a_j - P_k g_j\|_2,\ j = 1, \ldots, n,$$

gdje g_j predstvavlja j-ti stupac matrice G. S obzirom na to da stupci matrice P_k čine jednu ortonormiranu bazu za $\mathbb{R}^k$, znamo da je onda rješenje prethodnog problema upravo $g_j = P_k^T a_j$, odnosno rješenje problema (2.2) možemo zapisati kao

$$G_k = P_k^T A.$$

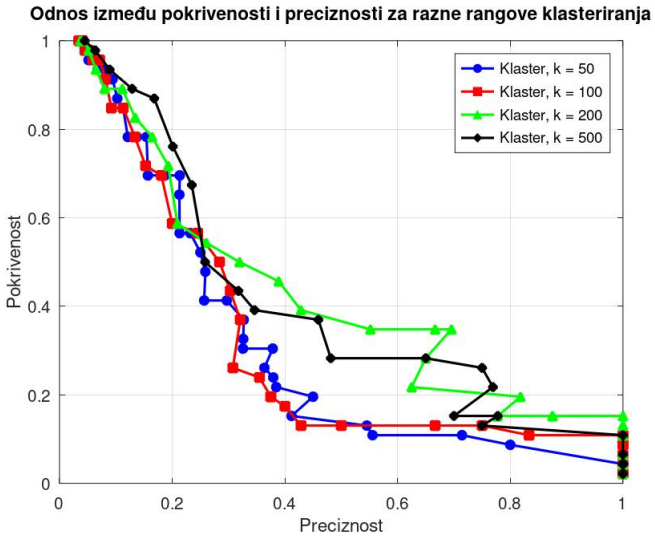
Kako bi upit bio usporediv s novim zapisima dokumenata imamo

$$q^T A \approx q^T P_k G_k = \left(P_k^T q\right)^T G_k = q_k^T G_k,$$

gdje $q_k = P_k^T q$. Tada kosinus kuta između upita q i dokumenta a_j u niždimenzionalnom prostoru možemo odrediti kao

$$\frac{q_k^T g_j}{\|q_k\|_2 \|g_j\|_2}.$$

Na Slici 2.5 za različite brojeve klastera prikazujemo odnos preciznosti i pokrivenosti za sve veće razine tolerancije.

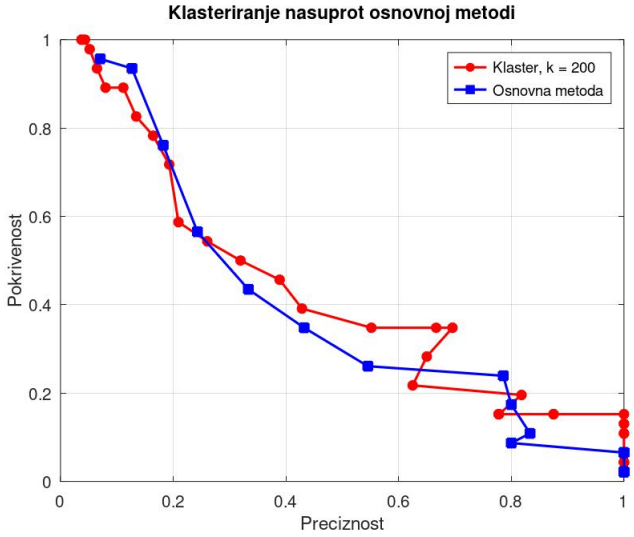


Slika 2.5: Odnos preciznosti i pokrivenosti u ovisnosti o broju zadanih klastera

Ovdje možemo primijetiti da u većini slučaja najbolji rezultat dobivamo kada imamo aproksimaciju koristeći 200 klastera. No, vidimo da ni koristeći aproksimaciju s 500 klastera ne dobivamo puno lošiji rezultat, štoviše u nekim dijelovima imamo i uspješniji rezultat. Obzirom na to da ako uzmemo aproksimaciju koristeći 500 klastera nismo puno smanjili dimenziju. Iz ovog razloga odlučili smo se za aproksimaciju koristiti 200 klastera, što je prosječno po 7 dokumenata u jednom klasteru. Kao i maloprije, možemo pogledati koliko iznosi aproksimacijska grešku u odnosu na početnu matricu

$$\frac{\|A - A_{200}\|_F}{\|A\|_F} \approx 0.32,$$

gdje A_{200} predstavlja aproksimativnu matricu koristeći 200 klastera. U idućem grafu na Slici 2.6 uspoređujemo rezultat dobiven klasteriranjem u odnosu na osnovnu metodu.



Slika 2.6: Metoda s 200 klastera nasuprot osnovnoj metodi

2.4 Nenegativna matrična faktorizacija

Obzirom da su svi elementi dokument matrice $A \in \mathbb{R}^{m\times n}$ nenegativni imamo zadovoljen nužni uvjet kako bi postojala nenegativna matrična faktorizacija dokument matrice. Spomenuta faktorizacija je obično aproksimativna gdje za $k < \min(m, n)$ tražimo nenegativne matrice $W \in \mathbb{R}^{m\times k}$ i $H \in \mathbb{R}^{k\times n}$ koje minimiziraju neku funkciju pogreške između A. Najčešće korištena funkcija pogreške zasnovana je na Frobeniusovoj normi

$$f(W,H) = \|A - WH\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left(a_{ij} - \sum_{p=1}^{k} w_{ip}h_{pj}\right)^2}.$$

S obzirom na to da je funkcija pogreške definirana koristeći matričnu normu, znamo da je ta funkcija ograničena odozdo s 0 te da će norma biti upravo jednaka 0 u slučaju da je $A = WH$. Uočimo da će nam problem minimizacije biti jednostavniji ako se odlučimo promatrati kvadrat funkcije f, odnosno

$$g(W,H) := f^2(W,H) = \sum_{i=1}^{m}\sum_{j=1}^{n}\left(a_{ij} - \sum_{p=1}^{k} w_{ip}h_{pj}\right)^2.$$

Odlučimo li fiksirati argument H, tada imamo konveksnu funkciju $g(W)$ obzirom da imamo kvadratnu funkciju u elementima matrice W. Odlučimo li fiksirati argument W dolazimo do sličnog zaključka da je i funkcija $g(H)$ konveksna. Međutim, konveksnost funkcije u

svakom od njenih argumenata pojedinačno ne povlači nužno da je funkcija globalno konveksna. Ovakav slučaj imamo kod naše funkcije g. Deriviramo li funkciju redom po W i H imamo

$$\frac{\partial g}{\partial w_{ip}} = -2\sum_{j=1}^{n}\left(a_{ij} - \sum_{q=1}^{k} w_{iq}h_{qj}\right)h_{pj} \to \nabla_W g = -2(A - WH)H^T,$$
$$\frac{\partial g}{\partial h_{pj}} = -2\sum_{j=1}^{m}\left(a_{ij} - \sum_{q=1}^{k} w_{iq}h_{qj}\right)w_{pj} \to .\nabla_H g = -2W^T(A - WH).$$

Deriviramo li još jednom matrične prikaze iznad, možemo dobiti blok Hessijan matricu koja ima sljedeći oblik

$$\begin{aligned}\nabla^2 g\,(W,H) &= \begin{bmatrix}\nabla^2_{W,W}g & \nabla^2_{W,H}g\\ \nabla^2_{H,W} & \nabla^2_{H,H}\end{bmatrix}\\ &= \begin{bmatrix}2HH^T & -2\,(A - WH)\\ -2\,(A - WH)^T & 2W^TW\end{bmatrix}\end{aligned}$$

Iz ovoga možemo zaključiti da funkcija g neće biti konveksna, pa time neće postojati globalni minimum, tako da ćemo se pokušati aproksimativno približiti nekom od lokalnih minimuma. Kako bismo došli do željene faktorizacije koristit ćemo ugrađenu funkciju `nmf` u Matlabu.

Pretpostavimo da imamo određenu aproksimativnu faktorizaciju dokument matrice:

$$A = WH,\ W \geq 0, H \geq 0.$$

U stupcu j matrice H možemo očitati aproksimativni zapis j-tog dokumenta u bazi koja je reprezentirana stupcima matrice W. Kao i maloprije, najprije nas zanima kako izgleda zapis upita q u novoj bazi. Rješavamo ponovo problem najmanjih kvadrata $\min_{\hat{q}} \|q - W\hat{q}\|_2$, ovaj problem ćemo najelegantnije riješiti ukoliko odredimo QR dekompoziciju matrice W. Tada je

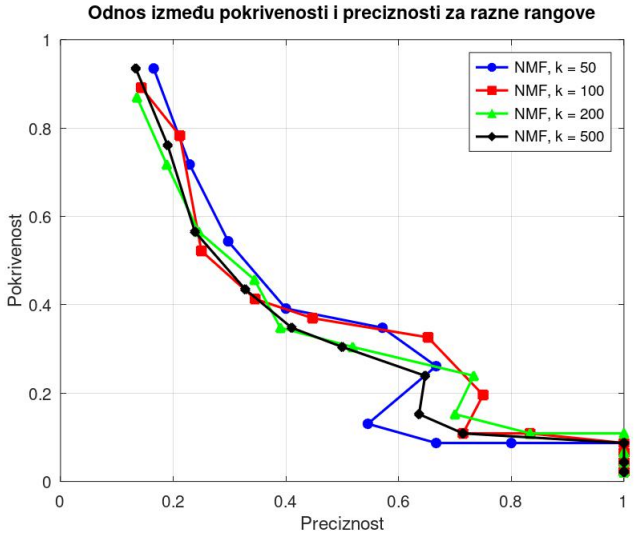
$$W = QR, \quad Q \in \mathbb{R}^{m\times k}, \quad R \in \mathbb{R}^{k\times k},$$

pa upit u novoj bazi možemo zapisati kao

$$\hat{q} = R^{-1}Q^T q.$$

Znamo da je vrijednost kosinusa kuta između dokumenta j i danog upita q zapisanih u novoj bazi upravo

$$\frac{\hat{q}^T h_j}{\|\hat{q}\|_2\|h_j\|_2}.$$



Slika 2.7: Odnos preciznosti i pokrivenosti u ovisnosti o rangu

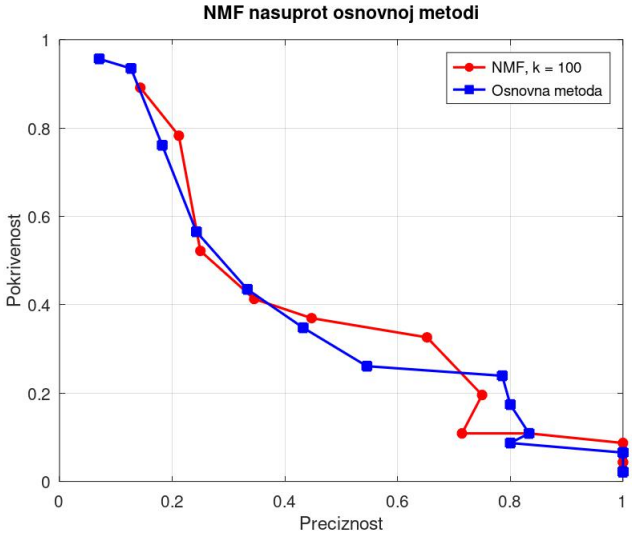
Na Slici 2.7 prikazujemo odnos između preciznosti i pokrivenost za različite rangove k.

Zanimljivo za primijetiti je da ne vidimo neku drastičnu razliku između rezultata dobivenih uzimajući matricu ranga 50 i ranga 500. Ovo ima smisla jer $\sigma_{50} = 1.29$, a $\sigma_{500} = 1.05$ iz čega primjećujemo da su singularne vrijednosti blizu jedna drugoj pa je posljedično greška koju napravimo slična. Međutim, s 2.7 možemo primijetiti da rang 100 daje najstabilnije rzultate, zbog čega ćemo ga koristiti za usporedbu s osnovnom metodom.

Možemo primijetiti da nenegativnom matričnom faktorizacijom u usporedbi s osnovnom metodom u većini slučajeva imamo veću preciznost i pokrivenost. Iznimka je slučaj kada tražimo preciznost od približno 80%, tada osnovnom metodom možemo ostvariti veću pokrivenost. Jedan od razloga zašto se ovo dogodilo može biti u tome što namećemo ograničenje nenegativnosti kod NMF faktorizacije.

2.5 Bidijagonalizacija

Do sada smo u ovom poglavlju opisali tri metode za poboljšanje osnovne vektorske metode. U sve tri metode smo najprije tražili aproksimativnu matricu nižeg ranga koja nam je predstavljala dokument matricu A. U svakoj metodi smo na drugačiji način određivali bazu za aproksimativnu dokument matricu. Nedostatak ovih metoda je u tome što dođe li do dodavanja, odnosno brisanja dokumenata moramo ažurirati trenutnu aproksimativnu matricu što nije efikasno. Iz ovog razloga, u ovom ćemo potpoglavlju opisati metodu ko-



Slika 2.8: Nenegativna matrična faktorizacija nasuprot osnovnoj metodi

jom će aproksimativna dokument matrica biti izračunata za svaki upit. Ovime nemamo problem ako dođe do promjene kolekcije dokumenata, dok s druge strane količina posla za podudaranje svakog upita postaje veća. Definirajmo najprije pojam gornje bidijagonalne matrice i bidijagonalne forme.

Definicija 2.5.1. *Matrica $A \in \mathbb{R}^{m\times n}$ je gornje bidijagonalna ako vrijedi $a_{i,j} = 0$, za sve $i > j$, te $a_{k,k+2}$ za $k = 1, \ldots, n-2$.*

Definicija 2.5.2. *Za matricu $A \in \mathbb{R}^{m\times(n+1)}$ kažemo da ima bidijagonalnu formu ako ju možemo zapisati na sljedeći način*

$$A = P \begin{bmatrix} \hat{B} \\ 0 \end{bmatrix} W^T,$$

gdje su P i W ortogonalne matrice, a $\hat{B}$ gornje bidijagonalna matrica.

Dakle, dopustimo li dijagonalnoj matrici da ima vrijednosti različite od nule iznad glavne dijagonale, tada će takva matrica biti gornje bidijagonalna. Želimo odrediti bidijagonalnu formu matrice $A \in \mathbb{R}^{m\times n}$. Ovaj problem nije značajno različit od problema dobivanja QR dekompozicije. Podsjetimo se, kod QR dekompozicije htjeli smo doći do unitarne matrice Q te gornjetrokutaste matrice R. No, u ovom slučaju imamo nešto stroži uvjet jer ne dopuštamo vrijednosti koje su različite od nule na pozicijama koje su od glavne dijagonale udaljene za barem 2 stupca. Ovaj problem možemo riješiti koristeći Householderove transformacije kojima ćemo redom poništiti željene elemente unutar stupca, te

zatim i željene elemente unutar retka. U nastavku ilustriramo dolazak do bidijagonalne forme na matrici $A \in \mathbb{R}^{6\times5}$. Najprije, množenjem matrice A slijeva matricom $P_1^T \in \mathbb{R}^{6\times5}$ poništimo sve elemente ispod glavne dijagonalne prvog stupca (pozicije na kojima je došlo do promjene ćemo označavati s *):

$$P_1^T A = P_1^T \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix}.$$

Množeći sada zdesna matricu $P_1^T A$ Householderovom transformacijom $W_1 \in \mathbb{R}^{5\times5}$ poništimo elemente prvog retka na pozicijama od 3 do 5. Kako bismo ovo postigli, a prvi stupac ostao nepromijenjen odabiremo sljedeću blok matricu

$$\mathbb{R}^{5\times5} \ni W_1 = \begin{bmatrix} 1 & 0 \\ 0 & Z_1 \end{bmatrix},$$

gdje je Z_1 Householderova transformacija. Ovime imamo

$$P_1^T A W_1 = \begin{bmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix} = \begin{bmatrix} \times & * & 0 & 0 & 0 \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{bmatrix} W_1 =: A_1$$

Ponhištimo sada ispoddijagonalne elemente drugog stupca tako da elementi prvog stupca ostanu nepromijenjeni. Ovo ćemo ostvariti uzimanjem matrice P_2

$$\mathbb{R}^{6\times6} \ni P_2 = \begin{bmatrix} 1 & 0 \\ 0 & \hat{P} \end{bmatrix},$$

gdje je $\hat{P} \in \mathbb{R}^{5\times5}$ Householderov reflektor. Ovim množenjem dobivamo

$$P_2^T A_1 = \begin{bmatrix} \times & \times & 0 & 0 & 0 \\ \times & * & * & * & * \\ \times & 0 & * & * & * \\ \times & 0 & * & * & * \\ \times & 0 & * & * & * \\ \times & 0 & * & * & * \end{bmatrix}.$$

Nakon, množenjem zdesna sljedećom blok matricom

$$W_2 = \begin{bmatrix} I_2 & 0 \\ 0 & Z_2 \end{bmatrix}, \qquad I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Ovom matricom poništavamo elemente u drugom retku pazeći da ne uništimo tek uvedene nule

$$P_2^T A_1 W_2 = \begin{bmatrix} \times & \times & 0 & 0 & 0 \\ 0 & \times & * & 0 & 0 \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \\ 0 & 0 & * & * & * \end{bmatrix} =: A_2$$

Primjenjujemo li isti postupak, konačno dobivamo

$$P^T A W = \begin{bmatrix} \times & \times & & & \\ & \times & \times & & \\ & & \times & \times & \\ & & & \times & \times \\ & & & & \times \\ & & & & \end{bmatrix} = \begin{bmatrix} \hat{B} \\ 0 \end{bmatrix}. \tag{2.3}$$

U općenitom slučaju,

$$P = P_1 P_2 \cdots P_n \in \mathbb{R}^{m \times m}, \quad W = W_1 W_2 \cdots W_{n-2} \in \mathbb{R}^{(n+1)\times(n+1)}$$

su ovo množenja Householderovih transformacija, te

$$\hat{B} = \begin{bmatrix} \beta_1 & \alpha_1 & & & \\ & \beta_2 & \alpha_2 & & \\ & & \ddots & \ddots & \\ & & & \beta_n & \alpha_n \\ & & & & \beta_{n+1} \\ & & & & \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}$$

je gornje bidijagonalna.

Propozicija 2.5.1. *[4] Označimo li stupce matrice P u bidijagonalnoj kompoziciji* (2.3) *s p_i, $i = 1, 2, \ldots, m$. Tada vrijedi*

$$p_1 = \beta_1 a_1, \quad W = \begin{bmatrix} 1 & 0 \\ 0 & Z \end{bmatrix}$$

gdje je a_1 prvi stupac od A te je $Z \in \mathbb{R}^{n\times n}$ ortogonalna.

Dokaz. Prva jednažba proizlazi direktno iz činjenice da je $P^T a_1 = \beta_1 e_1$. Druga tvrdnja proizlazi iz činjenice da W_i ima strukturu

$$W_i = \begin{bmatrix} I_1 & 0 \\ 0 & Z_i \end{bmatrix},$$

gdje su $I_i \in \mathbb{R}^{i\times i}$ matrice identiteta, a Z_i ortogonalne. □

Odaberimo $A = \begin{bmatrix} b & C \end{bmatrix}$, gdje o $b \in \mathbb{R}^{m\times 1}$ možemo razmišljati kao o jednom upitu te o $C \in \mathbb{R}^{m\times n}$ kao dokument matrici. Na toj matrici želimo primijeniti proceduru za bidijagonalizaciju, pa ćemo koristeći rezultat prethodne propozicije i jednadžbu (2.3) dobiti sljedeće

$$P^T AW = P^T \begin{bmatrix} b & C \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & Z \end{bmatrix} = \begin{bmatrix} P^T b & P^T AZ \end{bmatrix} = \begin{bmatrix} \beta_1 e_1 & B \\ 0 & 0 \end{bmatrix}, \tag{2.4}$$

gdje je

$$B = \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_n & \alpha_n \\ & & & \beta_{n+1} \end{bmatrix} \in \mathbb{R}^{(n+1)\times n}.$$

Ovakav prikaz će nam u nastavku pomoći pri alternativnom opisu postupka bidijagonalizacije koji nam omogućuje da izračunamo dekompoziciju (2.3) na rekurzivan način. Dio posljednje jednažbe (2.4) može se zapisati kao

$$P^T A = \begin{bmatrix} BZ^T \\ 0 \end{bmatrix}, \quad BZ^T \in \mathbb{R}^{(n+1)\times n}.$$

Nadalje, transponiramo li prethodnu jednakost imamo

$$A^T \begin{bmatrix} p_1 & p_2 & \cdots & p_{n+1} \end{bmatrix} = ZB^T = \begin{bmatrix} z_1 & z_2 & \dots & z_n \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta_1 & & & & & & \\ & \alpha_2 & \beta_3 & & & & & \\ & & \ddots & \ddots & & & & \\ & & & & \beta_i & & & \\ & & & & \alpha_i & & & \\ & & & & & \ddots & \ddots & \\ & & & & & & \alpha_n & \beta_{n+1} \end{bmatrix}$$

Izjednačavanjem stupca i (za $i \geq 2$) s obje strane, dobivamo

$$A^T p_i = \beta_i z_{i-1} + \alpha_i z_i,$$

što možemo zapisati kao

$$\alpha_i z_i = A^T p_i - \beta_i z_{i-1} \tag{2.5}$$

Slično, izjednačavanjem po stupcima imamo

$$AZ = A\begin{bmatrix} z_1 & z_2 & \cdots & z_n \end{bmatrix} = PB = \begin{bmatrix} p_1 & p_2 & \dots & p_{n+1} \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta_1 & & & & & & \\ & \alpha_2 & \beta_3 & & & & & \\ & & \ddots & \ddots & & & & \\ & & & & \beta_i & & & \\ & & & & \alpha_i & & & \\ & & & & & \ddots & \ddots & \\ & & & & & & \alpha_n & \beta_{n+1} \end{bmatrix},$$

iz čega dobivamo

$$Az_i = \alpha_i p_i + \beta_{i+1} p_{i+1},$$

što možemo zapisati kao

$$\beta_{i+1} p_{i+1} = Az_i - \alpha_i p_i. \tag{2.6}$$

Nalazimo se na korak do rekurzije, a početni uvjet iz prethodne propozicije pomaže nam u njezinom konačnom definiranju.

LGK bidijagonalizacija za upit q

(1) $\beta_1 p_1 = q,\ z_0 = 0$

(2) za $i = 1 : k$ izračunaj

$\alpha_i z_i = A^T p_i - \beta_i z_{i-1}$

$\beta_{i+1} p_{i+1} = Az_i - \alpha_i p_i$

Način odabira koeficijenata α_{i-1} i β_i nije preciziran u algoritmu. Kako bismo osigurali da vektori p_i i z_i budu jedinične duljine, koeficijente određujemo tako da vrijedi $\|p_i\| = \|z_i\| = 1$, odnosno $\beta_i = \|p_i\|$ i $\alpha_i = \|z_i\|$. Vektore p_i i z_i organiziramo u matrice kako slijedi

$$P_{k+1} = \begin{bmatrix} p_1 & p_2 & \dots & p_{k+1} \end{bmatrix},\ Z_k = \begin{bmatrix} z_1 & z_2 & \dots & z_k \end{bmatrix}$$

Sažetak izvođenja prikazan je u sljedećoj propoziciji.

Propozicija 2.5.2. *Neka je $AZ_k = P_{k+1}B_{k+1}$ rezultat k koraka LGK rekurzije, te neka je*

$$B_{k+1} = Q_{k+1}\begin{bmatrix} \hat{B}_k \\ 0 \end{bmatrix}$$

QR dekompozicija bidijagonalne matrice B_{k+1}. Tada imamo aproksimaciju ranga k

$$A \approx W_k Y_k^T \tag{2.7}$$

gdje je

$$W_k = P_{k+1} Q_{k+1} \begin{bmatrix} I_k \\ 0 \end{bmatrix}, \quad Y_k = Z_K \hat{B}_k^T.$$

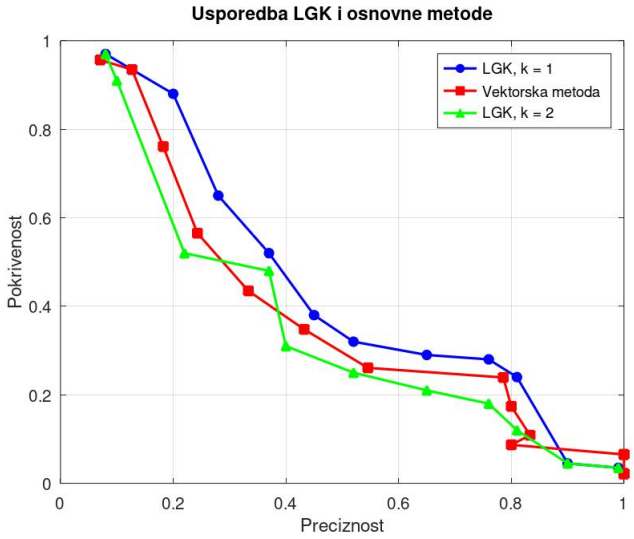
Stupci vektori matrice W_k čine ortogonalnu, približnu bazu za dokumente koji su blizu upita q. Umjesto izračunavanja koordinata stupaca matrice A u odnosu na ovu bazu, sada biramo izračunavanje projekcije upita u odnosu na ovu bazu

$$\hat{q} = W_k W_k^T q \in \mathbb{R}^m$$

Kao i do sada, izračunat ćemo vrijednost sljedećeg izraza, budući da smo upravo na taj način definirali sličnost između dokumenta a_j i upita q

$$\cos(\varphi_j) = \frac{\hat{q}^T a_j}{\|\hat{q}^T\|_2 \|a_j\|_2}.$$

Na Slici 2.9 prikazujemo usporedbu između osnovnog vektorskog modela i LGK bidijagonalizacije za $k = 1, 2$.



Slika 2.9: LGK bidijagonalizacija nasuprot osnovnoj metodi

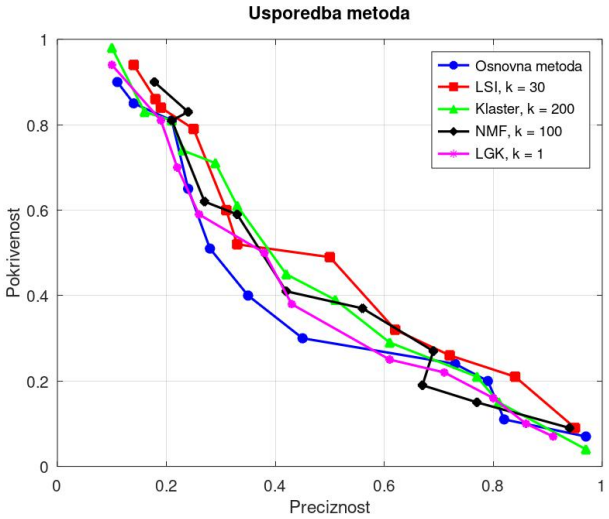
Možemo primijetiti da korištenjem LGK bidijagonalizacije uzimajući $k = 1$ imamo najbolje rezultate. Zanimljivo za uočiti je da uzimajući $k = 2$ dobivamo slične rezultate kao u vektorskoj metodi. Ovo je iz razloga što uzimajući sve veći k aproksimacijska matrica je sve bliže početnoj dokument matrici.

Poglavlje 3

Usporedba obrađenih metoda

Prvo je važno napomenuti da su rezultati iz drugog poglavlja temeljeni isključivo na jednom upitu. U ovom dijelu ćemo odrediti prosječnu preciznost i pokrivenost na skupu svih upita za svaku od obrađenih metoda, kako bismo dobili širu osnovu za donošenje zaključaka vezanih uz ovaj skup podataka. Za neke upite q ne postoje relevantni dokumenti, te za takve, pojmovi preciznosti i pokrivenosti nisu dobro definirani. Iz tog razloga pokrivenost i preciznost je računata samo za one upite koji imaju barem jedan relevantni dokument u bazi dokumenata. Također, treba imati na umu da će se ove metode primijeniti samo na jednoj kolekciji tekstova, što zahtijeva oprez u tumačenju rezultata.

Na Slici 3.1 prikazujemo prosječnu preciznost i pokrivenost na skupu upita za obrađene metode.



Slika 3.1: Usporedba metoda

Možemo uočiti da sve metode daju bolje prosječne rezultate od osnovnog vektorskog modela. Međutim, treba biti svjestan da posljedično imamo veću složenost pri računanju. U slučaju LSI, klasteriranja i nenegativne matrične faktorizacije, dodatni izračuni mogu se obaviti izvanmrežno, odnosno odvojeno od usklađivanja upita. Problem kod ovih metoda nastaje kada se učestalo mijenja skup dokumenata, jer tada mora doći do ponovnog iračuna aproksimacije, što može biti skupo. Metoda temeljena na LGK bidijagonalizaciji, s druge strane, izvodi dodatne proračune u vezi s usklađivanjem upita. Stoga se može učinkovito koristiti u situacijama gdje je skup dokumenata podložan čestim promjenama.

Na kraju, važno je napomenuti da link na podatke korištene u [4] više nije dostupan, zbog čega su u ovom radu korišteni alternativni podaci. Osim toga, moguće je da podaci nisu očišćeni jednako temeljito kao u [4], što bi moglo objasniti zašto je u [4] razlika između osnovne vektorske metode i naprednijih pristupa izraženija nego u našim rezultatima.

Bibliografija

[1] *CISI (a dataset for Information Retrieval)*, https://www.kaggle.com/datasets/dmaso01dsta/cisi-a-dataset-for-information-retrieval/data.

[2] Zlatko Drmac, Vjeran Hari, Miljenko Marušic, Mladen Rogina, Sanja Singer i Saša Singer, *Numericka analiza*, PMF-Matematicki odjel, Sveucilište u Zagrebu (2003).

[3] Zlatko Drmač, *Prezentacije iz kolegija Matrične i tenzorske metode u analizi podataka*.

[4] Lars Eldén, *Matrix methods in data mining and pattern recognition*, second., Fundamentals of Algorithms, sv. 15, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2019, ISBN 978-1-611975-85-7. MR 3999331

[5] Zrinka Franušić i Juraj Šiftar, *Linearna algebra*, Matematički odsjek, PMF (2022.).

[6] Gene H. Golub i Charles F. Van Loan, *Matrix computations*, fourth., Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 2013, ISBN 978-1-4214-0794-4; 1-4214-0794-9; 978-1-4214-0859-0. MR 3024913

[7] Nick Higham, *What Is the Singular Value Decomposition*, 2020, https://nhigham.com/2020/10/13/what-is-the-singular-value-decomposition/.

[8] Goran Muić i Mirko Primc, *Vektorski prostori - skripta*, https://www.pmf.unizg.hr/_download/repository/vp%5B1%5D.pdf.

Sažetak

Rad se bavi analizom različitih numeričkih metoda za obradu tekstualnih podataka. Kroz rad se istražuju osnovni vektoriski model, latentna semantička indeksacija (LSI), klasteriranje, nenegativna matrična faktorizacija (NMF) te LGK bidijagonalizacija. Za svaku od metoda evaluirani su pojmovi preciznosti i pokrivenosti u pretraživanju dokumenata za zadane upite.

Najprije smo obradili osnovni vektorski model koji pruža jednostavan i intuitivan pristup koji je često nedovoljno precizan. Zatim smo se uvjerili da nešto složenije matrične faktorizacije poput LSI i NMF mogu pridonijeti boljim rezultatima jer identificiraju semantičke veze među dokumentima. Klasteriranjem grupiramo slične dokumente u jedan klaster, dok LGK bidijagonalizacija pokazuje efikasnost u dinamičnim skupovima podataka gdje je skup dokumenata sklon čestim izmjenama. Kroz rad smo se uvjerili da sofisticiranije metode donose bolje rezultate, no ipak uz povećanu računalnu složenost.

Zaključno, rad naglašava važnost odabira odgovarajuće metode ovisno o specifičnostima skupa podataka i zahtjevima korisnika.

Summary

This paper focuses on the analysis of various numerical methods for processing textual data. It explores basic vector models, Latent Semantic Indexing (LSI), clustering, Non-negative Matrix Factorization (NMF), and LGK bidiagonalization. For each method, the concepts of precision and recall in document retrieval for given queries were evaluated.

We first examined the basic vector model, which offers a simple and intuitive approach, though often insufficiently precise. We then demonstrated that more complex matrix factorizations such as LSI and NMF can yield better results by identifying semantic relationships among documents. Clustering groups similar documents into a single cluster, while LGK bidiagonalization shows efficiency in dynamic datasets where the document collection is subject to frequent changes. Throughout the study, we confirmed that more sophisticated methods provide better results, albeit with increased computational complexity.

In conclusion, the paper emphasizes the importance of selecting an appropriate method depending on the specifics of the dataset and user requirements.

Životopis

Lucijan Matanović rođen je 30.08.1999. u Zagrebu. U Zagrebu je završio osnovnu školu (OŠ Voltino) i srednju (Gimnazija Tituša Brezovačkog) školu. Nakon srednje škole, 2018. godine, upisuje preddiplomski sveučilišni studij edukacije matematike na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta u Zagrebu kojeg završava 2021. godine. Svoje obrazovanje nastavlja upisom na diplomski studij Financijske i poslovne matematike, a od srpnja 2023. godine zasniva radni odnos kao analitičar podataka u Konzumu.