

Tenzorske dekompozicije za prepoznavanje radnje

Tonček, Maja

Master's thesis / Diplomski rad

2020

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:017585>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-29**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Maja Tonček

TENZORSKE DEKOMPOZICIJE ZA
PREPOZNAVANJE RADNJE

Diplomski rad

Voditelj rada:
prof. dr. sc. Zlatko Drmač

Zagreb, veljača, 2020.

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. _____

Sadržaj

Sadržaj	iii
Uvod	1
1 Osnovno o tenzorima	2
1.1 O definiciji	2
1.2 Neki pojmovi i jednostavnije operacije s tenzorima	2
1.3 Matricizacija, vektorizacija te tenzorska i matrična množenja	6
2 Dekompozicije tenzora	13
2.1 CP dekompozicija	13
2.1.1 Osnovno	13
2.1.2 Rang tenzora	15
2.1.3 Jedinstvenost rang dekompozicije	16
2.1.4 Aproksimacije nižeg ranga	18
2.1.5 Računanje CP dekompozicije	19
2.1.6 Općenita primjena	23
2.1.7 Primjena u prepoznavanju radnje	23
2.2 Tuckerova dekompozicija	26
2.2.1 Osnovno	26
2.2.2 n -rang i multirang	28
2.2.3 Računanje Tuckerove dekompozicije	29
2.2.4 (Ne)jedinstvenost	32
2.2.5 Općenita primjena	33
2.2.6 Primjena u prepoznavanju radnje	33
2.3 Vlačna dekompozicija	36
2.4 Prstenasta dekompozicija	36
2.5 Blokovska dekompozicija	38
3 Implementacija algoritma za prepoznavanje radnje	40
3.1 Algoritam Tuckerove <i>ridge</i> regularizacije	41

<i>SADRŽAJ</i>	iv
3.2 Eksperiment	44
3.3 Rezultati	46
Bibliografija	49

Uvod

Prepoznavanje koristi se u mnogim područjima ljudskog djelovanja poput video nadziranja i praćenja okolišnih uvjeta kuće na daljinu, pohrane i dohvaćanja videozapisa, inteligentnih sučelja čovjek-stroj i prepoznavanja identiteta ([76]). Prirodan način prikazivanja videa ili niza snimljenih scena koji predstavljaju neku kretnju je pomoću tenzora, koji se mogu shvatiti kao poopćenja matrica i o čijim će osnovnim svojstvima biti riječ u prvom poglavlju. Takvi tenzori tada mogu biti trodimenzionalni, s dimenzijama koje označavaju piksele scena po visini i širini te vrijeme. Često se želi bolje razlučiti kako se kretnja odvija s obzirom na pojedinu dimenziju, izvući iz danih podataka najbitnije značajke koje na kretnju utječu i organizirati ih u zasebne "prostore". Napomenimo još kako postoje drugačiji načini manevriranja sa spremanjem scena u tenzore, s drugačijim značenjima dimenzija, što ćemo uočiti u primjerima u radu.

Spomenuti zasebni prostori dobivaju se tenzorskim dekompozicijama. U drugom poglavlju prvo ćemo detaljnije opisati najpoznatije CP i Tuckerovu dekompoziciju. Dotaknut ćemo se pojmova rangova dekompozicija, svojstva (ne)jedinственosti dekompozicija te ćemo objasniti najpopularnije algoritme za njihovo računanje CP-ALS (CP), HOSVD te HOOI (Tuckerova). Zatim ćemo predstaviti manje poznatije vlačnu, prstenastu i blokovsku dekompoziciju, koje su posljednjih godina donosile uspješne rezultate u komprimiranju neuronskih mreža za prepoznavanje radnje. Na kraju opisa dekompozicija govorit ćemo o njihovim primjenama, pri čemu ćemo se posebno osvrnuti na područje prepoznavanja radnja.

Posljednje poglavlje ćemo posvetiti jednom pristupu prepoznavanja radnje iz rada [77] koji koristi Tuckerovu regularizaciju. Na početku ćemo objasniti kako su slike u radu prikazane tenzorima pomoću HOG opisnika slika, potom ćemo detaljno predstaviti algoritam, a na kraju ćemo navesti rezultate naše implementacije i usporediti ih s onima iz rada.

Poglavlje 1

Osnovno o tenzorima

1.1 O definiciji

U ovom radu tenzore ćemo shvaćati na drugačiji način od njihove standardne definicije. Ispravno gledano, oni su elementi nekog prostora tenzorskog produkta ([45]) koji se označava s $V_1 \otimes \cdots \otimes V_d$, pri čemu su V_1, \dots, V_d vektorski prostori nad poljem \mathbb{F} (u ostatku rada promatrat ćemo isključivo polje \mathbb{R}). Uz dane baze $B_i = \{b_1^{(i)}, \dots, b_{n_i}^{(i)}\}$ za $V_i, i = 1, \dots, d$, može se pokazati da se svaki tenzor \mathcal{T} iz $V_1 \otimes \cdots \otimes V_d$ može prikazati na jedinstven način kao linearna kombinacija

$$\sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} \cdots \sum_{j_d=1}^{n_d} a_{j_1 \dots j_d} b_{j_1}^{(1)} \otimes \cdots \otimes b_{j_d}^{(d)},$$

pri čemu je $a_{j_1 \dots j_d} \in \mathbb{F}$, a $b_{j_1}^{(1)} \otimes \cdots \otimes b_{j_d}^{(d)}$ je element prostora $V_1 \otimes \cdots \otimes V_d$, za sve j_1, \dots, j_d . Dakle, uz dani skup baza, tenzor \mathcal{T} je u potpunosti određen koordinatama $a_{j_1 \dots j_d}$. Shvatimo li njih kao (j_1, \dots, j_d) -element nekog d -dimenzionalnog polja A iz $\mathbb{F}^{n_1 \times \cdots \times n_d}$, dobivamo reprezentaciju tenzora \mathcal{T} pomoću "hipermatrice" ([45]) A uz odabir skupa baza. Uskoro ćemo se uvjeriti kako je ovakvo shvaćanje tenzora slično uobičajenoj matričnoj teoriji.

1.2 Neki pojmovi i jednostavnije operacije s tenzorima

Prema prethodnome, ne pravimo razliku između tenzora i hipermatrice, tj. pretpostavljamo da je svaki tenzor element iz nekog $\mathbb{R}^{I_1 \times \cdots \times I_N}$.

Definicija 1.2.1. Za svaki tenzor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ broj N zovemo **redom** od \mathcal{X} , brojeve $1, 2, \dots, N$ **modovima** od \mathcal{X} , a brojeve I_1, I_2, \dots, I_N **veličinama** odgovarajućih modova.

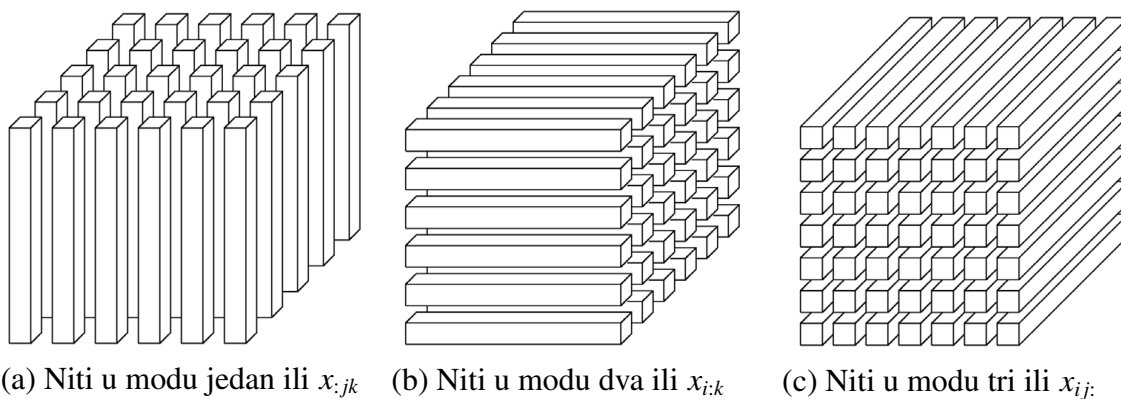
Vektori su tada zapravo tenzori reda jedan, a matrice tenzori reda dva. Tenzori reda barem tri nazivaju se tenzorima višeg reda.

Ono što kod matrica nazivamo stupcima i redcima, kod tenzora zovemo *nitima* u *pojedinom modu*, a dobijemo ih kada fiksiramo sve indekse tenzora osim indeksa koji odgovara modu. Niti tenzora reda tri prikazane su na Slici 1.1. (i_1, \dots, i_N) -ti element tenzora $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ označavat ćemo s $x_{i_1 \dots i_N}$.

Definicija 1.2.2. *Nit u modu m* tenzora $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$, gdje je $m \in \{1, \dots, N\}$, je vektor $x \in \mathbb{R}^{I_m}$ t.d.

$$x(j) = x_{k_1 \dots k_{m-1} j k_{m+1} \dots k_N}, \quad j = 1, \dots, I_m,$$

pri čemu je $k_l \in \{1, \dots, I_l\}$ proizvoljan i fiksiran za svaki $l \in \{1, \dots, N\} \setminus \{m\}$. Vektor x označavamo s $x_{k_1 \dots k_{m-1} : k_{m+1} \dots k_N}$, a u računu ćemo ga kasnije shvaćati kao jednostupčanu matricu iz $M_{I_m, 1}(\mathbb{R})$.



Slika 1.1: Niti tenzora trećeg reda. Preuzeto iz [36].

Fiksiramo li sve osim dva indeksa nekog tenzora, dobivamo *odsječak* tenzora. Odsječci tenzora reda tri ilustrirani su na Slici 1.2.

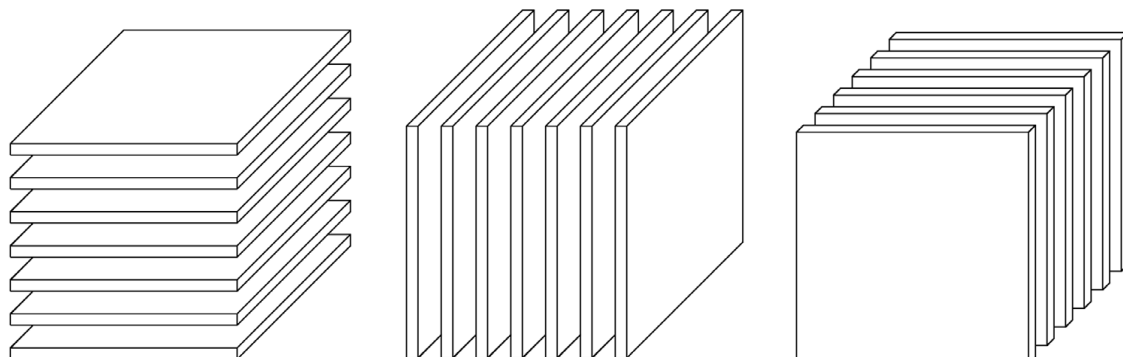
Definicija 1.2.3. *Odsječak tenzora* $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ reda barem dva je matrica $X \in M_{I_m, I_n}$, gdje su $m, n \in \{1, \dots, N\}$, $m < n$, takva da je

$$x(i, j) = x_{k_1 \dots k_{m-1} i k_{m+1} \dots k_{n-1} j k_{n+1} \dots k_N}, \quad i = 1, \dots, I_m, \quad j = 1, \dots, I_n,$$

pri čemu je $k_l \in \{1, \dots, I_l\}$ proizvoljan i fiksiran za svaki $l \in \{1, \dots, N\} \setminus \{m, n\}$. Matricu X označavamo s $X_{k_1 \dots k_{m-1} : k_{m+1} \dots k_{n-1} : k_{n+1} \dots k_N}$.

Definicija 1.2.4. Za tenzor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ kažemo da je:

- *kubični*, ako mu je svaki mod jednake veličine, tj. $I_1 = I_2 = \dots = I_N$.



(a) Horizontalni odsječci ili $X_{i::}$ (b) Lateralni odsječci ili $X_{:j}$ (c) Frontalni odsječci, $X_{::k}$ ili X_k

Slika 1.2: Odsječci tenzora trećeg reda. Preuzeto iz [36].

- **supersimetrični**, ako je kubični i ako za svaku permutaciju p od N elemenata vrijedi

$$x_{i_1 i_2 \dots i_N} = x_{i_{p(1)} i_{p(2)} \dots i_{p(N)}}, \quad i_1, i_2, \dots, i_N = 1, 2, \dots, I_1 = I_2 = \dots = I_N.$$

- **simetrični u modovima iz $S \subseteq \{1, 2, \dots, N\}$** , prema [63], ako su modovi iz S jednakih veličina i ako za svaku permutaciju p na skupu S vrijedi

$$x_{i'_1 i'_2 \dots i'_N} = x_{i_1 i_2 \dots i_N}, \quad \text{gdje je } i'_j = \begin{cases} p(i_j), & j \in S \\ i_j, & \text{inače} \end{cases}$$

za sve $j = 1, 2, \dots, N$ i sve indekse i_1, i_2, \dots, i_N tenzora \mathcal{X} (primjerice, tenzor $\mathcal{X} \in \mathbb{R}^{I \times I \times K}$ reda tri je simetričan u modovima jedan i dva ako su mu svi frontalni odsječci simetrični, tj. za sve $k = 1, 2, \dots, K$ je $X_k = X_k^T$).

- **dijagonalni**, ako vrijedi

$$x_{i_1 i_2 \dots i_N} \neq 0 \Leftrightarrow i_1 = i_2 = \dots = i_N \quad i_n = 1, 2, \dots, I_n, \quad n = 1, 2, \dots, N.$$

Unutarnji produkt dvaju tenzora i norma tenzora su poopćenja Frobeniusovog skalarnog produkta i Frobeniusove matrične norme.

Definicija 1.2.5. *Unutarnji (skalarni) produkt dvaju tenzora $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times \dots \times I_N}$, u oznaci $\langle \mathcal{X}, \mathcal{Y} \rangle$, je broj koji se dobije množenjem tenzora po komponentama te zbrajanjem tih vrijednosti, tj.*

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N} y_{i_1 i_2 \dots i_N}.$$

Norma tenzora $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$, u oznaci $\|\mathcal{X}\|$, je broj koji se dobije korjenovanjem sume kvadrata svih elemenata tenzora, tj.

$$\|\mathcal{X}\| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N}^2} = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}.$$

Sljedeću operaciju definiramo kako bismo mogli objasniti pojam tenzora ranga jedan.

Definicija 1.2.6. *Vanjski produkt N vektora $a^{(1)}, a^{(2)}, \dots, a^{(N)}$, u oznaci $a^{(1)} \circ a^{(2)} \circ \dots \circ a^{(N)}$ (ili \otimes umjesto \circ), gdje $a^{(n)} \in \mathbb{R}^{I_n}, \forall n \in \{1, \dots, N\}$, je tenzor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ t.d. je svaki njegov element produkt odgovarajućih elemenata vektora $a^{(n)}$, tj.*

$$x_{i_1 i_2 \dots i_N} = a_{i_1}^{(1)} a_{i_2}^{(2)} \dots a_{i_N}^{(N)}, \quad i_n = 1, \dots, I_n, \quad n = 1, \dots, N.$$

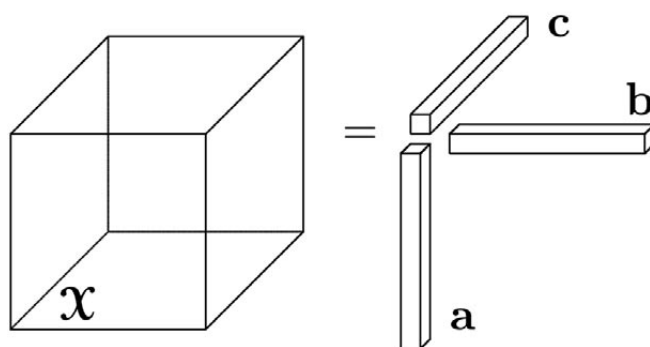
Kada budemo govorili o dekompozicijama tenzora, zapravo ćemo govoriti o načinima njihovog rastavljanja na tenzore ranga jedan. Slijedi definicija takvih tenzora, a ilustracija tenzora reda tri ranga jedan dana je u Slici 1.3.

Definicija 1.2.7. *Za tenzor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ kažemo da je **tenzor ranga jedan** ako se može zapisati kao jedan vanjski produkt N vektora, tj. postoje $a^{(n)} \in \mathbb{R}^{I_n}$ za $n = 1, 2, \dots, N$ tako da*

$$\mathcal{X} = a^{(1)} \circ a^{(2)} \circ \dots \circ a^{(N)}.$$

Zapis po elementima gornjeg izraza je

$$x_{i_1 i_2 \dots i_N} = a_{i_1}^{(1)} a_{i_2}^{(2)} \dots a_{i_N}^{(N)} \quad 1 \leq i_n \leq I_n.$$



Slika 1.3: Tenzor $\mathcal{X} = a \circ b \circ c$ reda tri ranga jedan, čiji je zapis po elementima $x_{ijk} = a_i b_j c_k$. Preuzeto iz [36].

1.3 Matricizacija, vektorizacija te tenzorska i matrična množenja

Matricizacijom i vektorizacijom se pojedini tenzor zapisuje u obliku matrice i vektora. Slijede nešto formalnije definicije tih dvaju pojmova.

Definicija 1.3.1. *Matricizacija u modu n je preslikavanje koje tenzoru $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ reda barem tri pridružuje matricu dimenzije $I_n \times (I_1 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N)$, označenu s $X_{(n)}$, čiji su stupci niti u modu n tenzora \mathcal{X} . Preciznije, (i_n, j) -ti element matrice $X_{(n)}$ jednak je (i_1, i_2, \dots, i_N) -tom elementu tenzora \mathcal{X} , pri čemu*

$$j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^N (i_k - 1)J_k \quad i \quad J_k = \prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m. \quad (1.1)$$

Matricu $X_{(n)}$ nazivamo odmotavanjem ili matricizacijom tenzora \mathcal{X} u modu n .

Ponekad se matricizacija u modu n definira s drugačijim poretkom stupaca, no poredak općenito nije važan sve dok je konzistentan u svim računanjima matricizacije.

Općenitija definicija matricizacije od navedene može se pronaći u [35] gdje se tenzor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ prikazuje matricom veličine $J \times K$, pri čemu su J i K umnošci modova iz skupova \mathcal{R} i \mathcal{C} , respektivno, a \mathcal{R} i \mathcal{C} partitioniraju $\{I_1, I_2, \dots, I_N\}$.

Definicija 1.3.2. *Vektorizacija je preslikavanje koje tenzoru $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ pridružuje vektor dimenzije $I_1 I_2 \dots I_N$, u oznaci $\text{vec}(\mathcal{X})$, koji se dobiva slaganjem niti u modu 1 tenzora \mathcal{X} u jedan stupac.*

Gornja definicija odgovara standardnoj matričnoj vektorizaciji, kod koje su niti u modu jedan upravo stupci matrice, i $\text{vec}(\mathcal{X}) = \text{vec}(X_{(1)})$.

Primjer 1.3.3. *Neka je $\mathcal{X} \in \mathbb{R}^{3 \times 4 \times 2}$ tenzor s frontalnim odsječcima*

$$X_1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix} \quad i \quad X_2 = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}.$$

Tada su odmotavanja tenzora \mathcal{X} u modovima jedan, dva i tri, redom,

$$X_{(1)} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix},$$

$$X_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix} \quad i$$

$$X_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & \cdots & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & \cdots & 21 & 22 & 23 & 24 \end{bmatrix}.$$

Vektorizacija tenzora \mathcal{X} dana je s

$$\text{vec}(\mathcal{X}) = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ 24 \end{bmatrix}.$$

Svaka matricizacija i vektorizacija tenzora je bijektivno preslikavanje, stoga je tenzor moguće zadati i matricom koja predstavlja njegovo odmotavanje u nekom modu ili vektorom.

Slijede neka svojstva norme i skalaranog produkta tenzora povezana s prethodnim dvjema operacijama, koja direktno slijede iz definicija.

Propozicija 1.3.4. *Neka su $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Tada vrijedi:*

- (1) $\|\mathcal{X}\|_F = \|X_{(n)}\|_F$, $n = 1, 2, \dots, N$,
- (2) $\|\mathcal{X}\|_F = \|\text{vec}(\mathcal{X})\|_2$,
- (3) $\langle \mathcal{X}, \mathcal{Y} \rangle = \text{vec}(\mathcal{X})^\top \text{vec}(\mathcal{Y})$.

Nastavljamo s uvođenjem množenja tenzora s matricama i vektorima.

Definicija 1.3.5. *Produkt u modu n tenzora $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ s matricom $U \in \mathbb{R}^{J \times I_n}$ je tenzor iz $\mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$, u oznaci $\mathcal{X} \times_n U$, čiji su elementi dani s*

$$(\mathcal{X} \times_n U)_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \cdots i_n} u_{j i_n} \quad (1.2)$$

za sve $i_l = 1, 2, \dots, I_l$, $l = 1, 2, \dots, n-1, n+1, \dots, N$, $j = 1, 2, \dots, J$.

U sljedećoj propoziciji pokazat ćemo da se sve niti u modu n tenzora $\mathcal{X} \times_n U$ mogu dobiti množenjem matrice U s nitima u modu n tenzora \mathcal{X} .

Propozicija 1.3.6. *Za $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ i $U \in \mathbb{R}^{J \times I_n}$ vrijedi*

$$\mathcal{Y} = \mathcal{X} \times_n U \Leftrightarrow Y_{(n)} = UX_{(n)}. \quad (1.3)$$

Dokaz. Elementi od \mathcal{Y} dani su s (1.2). Prema Definiciji 1.3.1, $(i_1, \dots, i_{n-1}, j, i_{n+1}, \dots, i_N)$ -ti element od \mathcal{Y} jednak je (j, k) -tom elementu odmotavanja $Y_{(n)}$, pri čemu je k dan s (1.1), tj.

$$(Y_{(n)})_{jk} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \dots i_N} u_{j i_n}. \quad (1.4)$$

S druge strane, (j, k) -ti element od $UX_{(n)}$ jednak je $\sum_{l=1}^{I_n} u_{jl}(X_{(n)})_{lk}$. Kako se, ponovo prema Definiciji 1.3.1, (l, k) -ti element odmotavanja $X_{(n)}$ preslikava u $(i_1, \dots, i_{n-1}, l, i_{n+1}, \dots, i_N)$ -ti element od \mathcal{X} , tj. $(X_{(n)})_{lk} = x_{i_1 \dots i_{n-1} l i_{n+1} \dots i_N}$, imamo

$$(UX_{(n)})_{jk} = \sum_{l=1}^{I_n} u_{jl} x_{i_1 \dots i_{n-1} l i_{n+1} \dots i_N}. \quad (1.5)$$

Vrijednosti iz (1.4) i (1.5) su jednake za sve $j = 1, 2, \dots, I_n$ i sve $k = 1, 2, \dots, I_1 \cdots I_{n-1} I_{n+1} \cdots I_N$, pa dolazimo do tražene tvrdnje. \square

Primjer 1.3.7. Uz $\mathcal{X} \in \mathbb{R}^{3 \times 4 \times 2}$ definiran kao u primjeru 1.3.3 i $U = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \in \mathbb{R}^{2 \times 3}$, produkt $\mathcal{X} \times_1 U = \mathcal{Y} \in \mathbb{R}^{2 \times 4 \times 2}$ dan je, u duhu (1.3), odmotavanjem u modu 1

$$\begin{aligned} \mathcal{Y}_{(1)} &= \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix} \cdot \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix} \\ &= \begin{bmatrix} 22 & 49 & 76 & 103 & 130 & 157 & 184 & 211 \\ 28 & 64 & 100 & 136 & 172 & 208 & 244 & 280 \end{bmatrix}, \end{aligned}$$

odnosno frontalnim odsječcima

$$Y_1 = \begin{bmatrix} 22 & 49 & 76 & 103 \\ 28 & 64 & 100 & 136 \end{bmatrix} \quad i \quad Y_2 = \begin{bmatrix} 130 & 157 & 184 & 211 \\ 172 & 208 & 244 & 280 \end{bmatrix}.$$

Poredak kod množenja tenzora s dvjema matricama u različitim modovima nije bitan; to i drugačiji zapis množenja tenzora dvjema matricama u istom modu pokazat ćemo u sljedećoj propoziciji. Također, navedena svojstva vrijede i kod množenja s proizvoljno mnogo matrica.

Propozicija 1.3.8. (a) Za $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, $A \in \mathbb{R}^{J_m \times I_m}$ i $B \in \mathbb{R}^{J_n \times I_n}$, $m \neq n$, vrijedi

$$(\mathcal{X} \times_m A) \times_n B = (\mathcal{X} \times_n B) \times_m A = \mathcal{X} \times_m A \times_n B.$$

(b) Za $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, $A \in \mathbb{R}^{I \times I_n}$ i $B \in \mathbb{R}^{K \times I}$ vrijedi

$$\mathcal{X} \times_n A \times_n B = \mathcal{X} \times_n (BA).$$

Dokaz. (a) Označimo s $\mathcal{Y} = \mathcal{X} \times_m A$ i $\mathcal{Z} = \mathcal{X} \times_n B$, te bez smanjenja općenitosti pretpostavimo da je $m < n$. Tada za sve moguće vrijednosti indeksa, koristeći (1.2), imamo

$$\begin{aligned} (\mathcal{Y} \times_n B)_{i_1 \dots i_{m-1} j_{m+1} \dots i_{n-1} k_{n+1} \dots i_N} &= \sum_{i_n=1}^{I_n} y_{i_1 \dots i_{m-1} j_{m+1} \dots i_n} b_{k_{i_n}} = \sum_{i_n=1}^{I_n} \left(\sum_{i_m=1}^{I_m} x_{i_1 \dots i_n} a_{j_{i_m}} \right) b_{k_{i_n}} \\ &= \sum_{i_m=1}^{I_m} \left(\sum_{i_n=1}^{I_n} x_{i_1 \dots i_n} b_{k_{i_n}} \right) a_{j_{i_m}} = \sum_{i_m=1}^{I_m} z_{i_1 \dots i_{n-1} k_{n+1} \dots i_n} a_{j_{i_m}} \\ &= (\mathcal{Z} \times_m A)_{i_1 \dots i_{m-1} j_{m+1} \dots i_{n-1} k_{n+1} \dots i_N} \end{aligned}$$

(b) Označimo $\mathcal{Y} = \mathcal{X} \times_n A \times_n B$. Tada iz (1.3) slijedi

$$Y_{(n)} = B(\mathcal{X} \times_n A)_{(n)} = BAX_{(n)},$$

stoga $\mathcal{Y} = \mathcal{X} \times_n (BA)$. □

Definicija 1.3.9. *Produkt u modu n tenzora $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ s vektorom $v \in \mathbb{R}^{I_n}$ je tenzor iz $\mathbb{R}^{I_1 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N}$, u oznaci $\mathcal{X} \bar{\times}_n v$, čiji su elementi dani s*

$$(\mathcal{X} \bar{\times}_n v)_{i_1 \dots i_{n-1} i_{n+1} \dots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \dots i_n} v_{i_n} \quad (1.6)$$

za sve $i_l = 1, 2, \dots, I_l$, $l = 1, 2, \dots, n-1, n+1, \dots, N$.

Iz (1.6) pak uočavamo da produkt u nekom modu tenzora s vektorom dobivamo unutar-njim produktom svake niti u zadanoj modu tenzora i vektora.

Primjer 1.3.10. Uz $\mathcal{X} \in \mathbb{R}^{3 \times 4 \times 2}$ iz primjera 1.3.3 i $v = [1 \ 2 \ 3 \ 4]^T \in \mathbb{R}^{4 \times 1}$, imamo

$$\mathcal{X} \bar{\times}_2 v = \begin{bmatrix} 70 & 190 \\ 80 & 200 \\ 90 & 210 \end{bmatrix} \in \mathbb{R}^{3 \times 2}.$$

Kod produkta tenzora s dvama vektorima u različitim modovima bitan je poredak jer se mijenja red tenzora iz međurezultata, tj. za $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, $a \in \mathbb{R}^{I_m}$, $b \in \mathbb{R}^{I_n}$ i $1 \leq m < n \leq N$ vrijedi

$$\mathcal{X} \bar{\times}_m a \bar{\times}_n b = (\mathcal{X} \bar{\times}_m a) \bar{\times}_{n-1} b = (\mathcal{X} \bar{\times}_n b) \bar{\times}_m a.$$

Slijede opisi triju matričnih produkata: Kroneckerovog, Khatri-Raovog i Hadamardovog.

Definicija 1.3.11. Kroneckerov produkt matrica $A = [a_{ij}] \in M_{I,J}(\mathbb{R})$ i $B = [b_{kl}] \in M_{K,L}(\mathbb{R})$, u oznaci $A \otimes B$, je blok matrica definirana sa

$$\begin{aligned} A \otimes B &= \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1J}B \\ a_{21}B & a_{22}B & \cdots & a_{2J}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}B & a_{I2}B & \cdots & a_{IJ}B \end{bmatrix} \\ &= \begin{bmatrix} a_1 \otimes b_1 & a_1 \otimes b_2 & a_1 \otimes b_3 & \cdots & a_J \otimes b_{L-1} & a_J \otimes b_L \end{bmatrix} \in M_{IK, JL}(\mathbb{R}). \end{aligned}$$

Neka svojstva Kroneckerovog produkata dana su u sljedećoj propoziciji.

Propozicija 1.3.12. Za matrice odgovarajućih dimenzija vrijedi:

(a) $(A \otimes B)(C \otimes D) = AC \otimes BD$

(b) $(A \otimes B)^\top = A^\top \otimes B^\top$

(c) ako su A i B inverzne, onda je i $A \otimes B$ inverzna i $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$

(d) ako su A i B ortogonalne, onda je i $A \otimes B$ ortogonalna

(e) $(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$

Pri tome simbol $A^\dagger \in \mathbb{R}^{J \times I}$ označava Moore-Penroseov inverz matrice $A \in \mathbb{R}^{I \times J}$, tj. (jedinstvenu) matricu $X \in \mathbb{R}^{J \times I}$ koja zadovoljava četiri Moore-Penroseova uvjeta: $AXA = A$, $XAX = X$, $(AX)^\top = AX$ i $(XA)^\top = XA$.

Važnost Kroneckerovog produkta posebno se očituje u tome što se pomoću njega može računati umnožak tenzora s nizom matrica. Dokaz propozicije preuzet je iz [60].

Propozicija 1.3.13. Neka su $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ i $A^{(n)} \in \mathbb{R}^{J_n \times I_n}$ za sve $n \in \{1, \dots, N\}$. Tada za bilo koji $n \in \{1, \dots, N\}$ vrijedi

$$\begin{aligned} \mathcal{Y} = \mathcal{X} \times_1 A^{(1)} \times_2 A^{(2)} \cdots \times_N A^{(N)} &\Leftrightarrow \\ Y_{(n)} = A^{(n)} X_{(n)} (A^{(N)} \otimes \cdots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \cdots \otimes A^{(1)})^\top &. \end{aligned}$$

Dokaz. Uočimo prvo da su obje matrice $Y_{(n)}$ i $A^{(n)} X_{(n)} (A^{(N)} \otimes \cdots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \cdots \otimes A^{(1)})^\top$ dimenzija $J_n \times (J_1 J_2 \cdots J_{n-1} J_{n+1} \cdots J_N)$.

Prema Definiciji 1.3.1, (j_1, \dots, j_N) -ti element tenzora \mathcal{Y} jednak je (j_n, k) -tom elementu odmotavanja $Y_{(n)}$, gdje je

$$k = 1 + \sum_{\substack{l=1 \\ l \neq n}}^N (j_l - 1) \prod_{\substack{m=1 \\ m \neq n}}^{l-1} J_m.$$

Pokažemo li da je (j_1, \dots, j_N) -ti element od \mathcal{Y} jednak i (j_n, k) -tom elementu od $A^{(n)}X_{(n)}(A^{(N)} \otimes \dots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \dots \otimes A^{(1)})^\top$, tvrdnja će biti dokazana.

Prema Definiciji 1.3.5, s jedne strane imamo

$$y_{j_1 \dots j_N} = (\mathcal{X} \times_1 A^{(1)} \times_2 A^{(2)} \dots \times_N A^{(N)})_{j_1 \dots j_N} = \sum_{i_1=1}^{I_1} \dots \sum_{i_N=1}^{I_N} x_{i_1 \dots i_N} a_{j_1 i_1}^{(1)} \dots a_{j_N i_N}^{(N)}.$$

S druge strane, označimo li s $M_n = (A^{(N)} \otimes \dots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \dots \otimes A^{(1)})^\top$, imamo

$$\begin{aligned} (A^{(n)}X_{(n)}M_n)_{j_n k} &= A_{j_n}^{(n)}(X_{(n)}M_n)_{:k} \\ &= \sum_{i_n=1}^{I_N} a_{j_n i_n}^{(n)} (X_{(n)}M_n)_{i_n k} \\ &= \sum_{i_n=1}^{I_N} a_{j_n i_n}^{(n)} \sum_{i=1}^{\hat{I}_n} (X_{(n)})_{i_n i} (M_n)_{ik}, \end{aligned} \quad (1.7)$$

pri čemu je $\hat{I}_n = I_1 \dots I_{n-1} I_{n+1} \dots I_N$. Prema Definiciji 1.3.1, $(X_{(n)})_{i_n i} = x_{i_1 \dots i_n}$, gdje je

$$i = 1 + \sum_{\substack{l=1 \\ l \neq n}}^N (i_l - 1) \prod_{\substack{m=1 \\ m \neq n}}^{l-1} I_m.$$

Iz definicije Kroneckerovog produkta može se pokazati da za isti i vrijedi

$$(M_n)_{ik} = \tilde{a}_{i_N j_N}^{(N)} \dots \tilde{a}_{i_{n+1} j_{n+1}}^{(n+1)} \tilde{a}_{i_{n-1} j_{n-1}}^{(n-1)} \dots \tilde{a}_{i_1 j_1}^{(1)},$$

gdje $\tilde{a}_{i_m j_m}^{(m)}$ označava (i_m, j_m) -ti element matrice $A^{(m)\top}$. Koristeći upravo pokazano, (1.7) postaje

$$\sum_{i_1=1}^{I_1} \dots \sum_{i_N=1}^{I_N} x_{i_1 \dots i_N} a_{j_1 i_1}^{(1)} \dots a_{j_N i_N}^{(N)}$$

pa je tvrdnja dokazana. \square

Khatri-Raov produkt se od Kroneckerovog razlikuje u tome što se međusobno množe samo stupci istih indeksa. Koristit ćemo ga kada budemo govorili o matricizacijama tenzora prikazanih pomoću CP dekompozicije i o nužnim uvjetima jedinstvenosti CP dekompozicije.

Definicija 1.3.14. *Khatri-Raov produkt matrica* $A = [a_{ij}] \in M_{I,K}(\mathbb{R})$ i $B = [b_{ij}] \in M_{J,K}(\mathbb{R})$, u oznaci $A \odot B$, je matrica definirana sa

$$A \odot B = [a_1 \otimes b_1 \quad a_2 \otimes b_2 \quad \dots \quad a_K \otimes b_K] \in M_{IJ,K}(\mathbb{R}).$$

Definicija 1.3.15. *Hadamardov produkt matrica* $A = [a_{ij}] \in M_{I,J}(\mathbb{R})$ i $B = [b_{ij}] \in M_{I,J}(\mathbb{R})$, u oznaci $A * B$, je matrica nastala množenjem A i B po koordinatama, tj.

$$A * B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ} \end{bmatrix} \in M_{I,J}(\mathbb{R}).$$

Slijede neka svojstva Khatri-Raovog i Hadamardovog produkta. Posljednje svojstvo se koristi u algoritmu CP-ALS za računanje CP dekompozicije i donosi određeno poboljšanje, o čemu ćemo više reći u sljedećem poglavlju.

Propozicija 1.3.16. *Za matrice odgovarajućih dimenzija vrijedi:*

$$(1) A \odot B \odot C = (A \odot B) \odot C = A \odot (B \odot C)$$

$$(2) (A \odot B)^\top (A \odot B) = A^\top A * B^\top B$$

$$(3) (A^\top)^\dagger = (A^\dagger)^\top$$

$$(4) (A \odot B)^\dagger = [(A^\top A) * (B^\top B)]^\dagger (A \odot B)^\top.$$

Poglavlje 2

Dekompozicije tenzora

Ovdje ćemo detaljno predstaviti tenzorske dekompozicije koje su našle primjenu u prepoznavanju radnje. O ostalim dekompozicijama, među kojima su INDSCAL, PARAFAC2, CANDELINC, DEDICOM, PARATUCK2 i hijerarhijska Tuckerova dekompozicija, može se više pročitati u [36] i [57].

2.1 CP dekompozicija

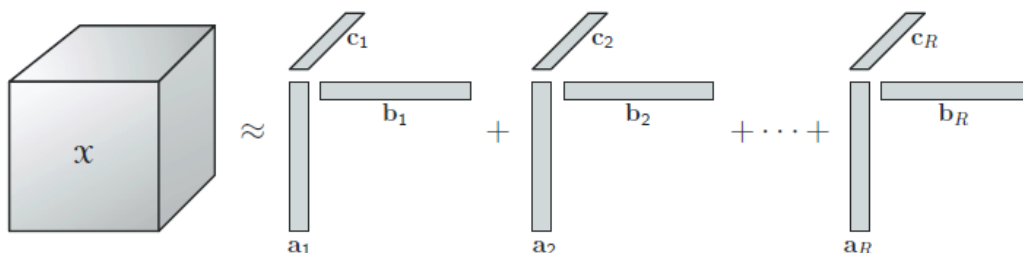
Začetci ove dekompozicije datiraju iz 1927., iz rada američkog matematičara i fizičara Franka L. Hitchcocka, u kojem se po prvi put predlaže ideja o zapisu tenzora pomoću konačne sume tenzora ranga jedan. U radu iz 1944., koji će poslužiti kao inspiracija kasnijim istraživanjima, ponovo se dotiče ta ideja, i zamire sve do 1970-ih, odnosno do otkriivanja primjene u psihometriji, tj. matematičkim modelima za analizu podataka o ljudskim osobnostima. U tom razdoblju je nazivana CANDECOMP, od *canonical decomposition*, i PARAFAC, od *parallel factors*, a danas nosi i ujedinen naziv CP dekompozicija.

2.1.1 Osnovno

Prije nego što definiramo CP dekompoziciju tenzora općenitog reda, približit ćemo je opisujući prvo situaciju za tenzore reda tri, koja je zapravo u praksi često upotrebljavana. Dakle, u CP dekompoziciji nekog tenzora $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ reda tri cilj je \mathcal{X} zapisati izrazom

$$\mathcal{X} \approx \sum_{r=1}^R a_r \circ b_r \circ c_r, \quad (2.1)$$

gdje je R neki prirodan broj, o čijoj vrijednosti ćemo malo kasnije nešto više reći, a tzv. *faktori* a_r , b_r i c_r su redom vektori iz \mathbb{R}^I , \mathbb{R}^J i \mathbb{R}^K za $r = 1, \dots, R$. Skica ove dekompozicije je na Slici 2.1.



Slika 2.1: CP-dekompozicija tenzora reda tri iz [36].

Izraz (2.1) možemo zapisati na nekoliko ekvivalentnih načina. Ako to učinimo po elementima, imamo

$$x_{ijk} \approx \sum_{r=1}^R a_{ir} b_{jr} c_{kr}, \quad \text{za } i = 1, \dots, I, j = 1, \dots, J, k = 1, \dots, K.$$

Organiziramo li faktore a_r , b_r i c_r za $r = 1, 2, \dots, R$ u *faktorske matrice* $A = [a_1 \ a_2 \ \dots \ a_R]$, $B = [b_1 \ b_2 \ \dots \ b_R]$ i $C = [c_1 \ c_2 \ \dots \ c_R]$, tada za odmotavanja u trima modovima tenzora \mathcal{X} vrijedi

$$\begin{aligned} X_{(1)} &\approx A(C \odot B)^\top, \\ X_{(2)} &\approx B(C \odot A)^\top \quad \text{i} \\ X_{(3)} &\approx C(B \odot A)^\top. \end{aligned}$$

Frontalni odsječci tenzora \mathcal{X} , a analogno i horizontalni i lateralni, dani su izrazom

$$X_k \approx A D^{(k)} B^\top, \quad k = 1, \dots, K, \quad (2.2)$$

gdje je $D^{(k)} \equiv \text{diag}(c_k)$, tj. $D^{(k)}$ je dijagonalna matrica koja na dijagonali ima elemente k -tog retka matrice C . Odsječci tenzora se, doduše, ne mogu jednostavno zapisati kada je red tenzora veći od tri.

Kompaktniji zapis izraza (2.1) dan je sa

$$\mathcal{X} \approx \llbracket A, B, C \rrbracket \equiv \sum_{r=1}^R a_r \circ b_r \circ c_r, \quad (2.3)$$

a normaliziranjem faktora i "grupiranjem" dobivenih normi u vektor $\lambda \in \mathbb{R}^R$, uz ažurirane vrijednosti matrica i vektora, izraz (2.1) označava se sa

$$\mathcal{X} \approx \llbracket \lambda; A, B, C \rrbracket \equiv \sum_{r=1}^R \lambda_r a_r \circ b_r \circ c_r.$$

Slijedi definicija CP dekompozicije tenzora općenitog reda.

Definicija 2.1.1. Za tenzor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ reda N kažemo da je njegova CP dekompozicija dana konačnom sumom tenzora ranga jedan, tj.

$$\mathcal{X} \approx \llbracket \lambda; A^{(1)}, A^{(2)}, \dots, A^{(N)} \rrbracket \equiv \sum_{r=1}^R \lambda_r a_r^{(1)} \circ a_r^{(2)} \circ \dots \circ a_r^{(N)}, \quad (2.4)$$

pri čemu je $R \in \mathbb{N}$, $\lambda \in \mathbb{R}^R$, a $A^{(n)} = \begin{bmatrix} a_1^{(n)} & a_2^{(n)} & \dots & a_r^{(n)} \end{bmatrix} \in \mathbb{R}^{I_n \times R}$ za $n = 1, \dots, N$ su faktorske matrice norme jedan. CP dekompozicija može biti zadana i u "nenormaliziranom obliku", u kojem faktori ne moraju nužno biti norme jedan te je vektor λ izostavljen.

Odmotavanje u modu n tenzora iz gornje definicije dano je sa

$$X_{(n)} \approx A^{(n)} \Lambda(A^{(N)}) \odot \dots \odot A^{(n+1)} \odot A^{(n-1)} \odot \dots \odot A^{(1)\top},$$

pri čemu je $\Lambda = \text{diag}(\lambda)$ ili, u nenormaliziranom obliku, $\Lambda = I$.

2.1.2 Rang tenzora

Prisjetimo se za početak matricne dekompozicije na singularne vrijednosti, ili SVD-a. Prema [23], za svaku matricu $A \in \mathbb{R}^{m \times n}$ postoje ortogonalne matrice $U = [u_1 \dots u_m] \in \mathbb{R}^{m \times m}$ i $V = [v_1 \dots v_n] \in \mathbb{R}^{n \times n}$ te dijagonalna matrica $\Sigma \in \mathbb{R}^{m \times n}$ s (jedininstvenim) dijagonalnim elementima $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_p = 0$, pri čemu je $p = \min\{m, n\}$, takva da vrijedi $A = U\Sigma V^\top$. Također, vrijedi

$$r = \text{rang}(A) \quad \text{i} \quad A = \sum_{i=1}^r \sigma_i u_i v_i^\top = \sum_{i=1}^r \sigma_i u_i \circ v_i.$$

Usporedimo li gornju jednakost s (2.4), možemo zaključiti kako je CP dekompozicija generalizacija SVD dekompozicije uz izostanak uvjeta ortogonalnosti i normiranosti. Također, vidimo i da se rang matrice alternativno može definirati kao broj pozitivnih singularnih vrijednosti u SVD-u matrice, odnosno kao broj sumanada u gornjem zapisu matrice, a upravo se analogno definira i rang tenzora. Ipak, pojmovi ranga matrice i ranga tenzora bitno se razlikuju u nekim svojstvima, a CP dekompozicija se u slučaju reda dva ne svodi "posve" na SVD dekompoziciju.

Definicija 2.1.2. Rang tenzora \mathcal{X} , u oznaci $\text{rang}(\mathcal{X})$, je najmanji broj tenzora ranga jedan koji zbrojeni daju neku egzaktnu CP dekompoziciju od \mathcal{X} , pri čemu riječ "egzaktna" označava da u definiciji CP dekompozicije vrijedi jednakost. Sve egzaktne CP dekompozicije od \mathcal{X} s $\text{rang}(\mathcal{X})$ komponenta nazivamo **rang dekompozicijama** tenzora \mathcal{X} .

Objasnimo u ostatku ove sekcije razlike dvaju rangova. Dok je rang svake matrice jedinstven, rang realnog tenzora se može razlikovati s obzirom na to jesu li elementi iz CP dekompozicije iz \mathbb{R} ili iz \mathbb{C} . Primjerice, tenzor \mathcal{X} zadan s frontalnim odsječcima

$$X_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{i} \quad X_2 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

ima rang 3 nad \mathbb{R} i rang 2 nad \mathbb{C} . Rang dekompozicija $\mathcal{X} = \llbracket A, B, C \rrbracket$ nad \mathbb{R} dana je s

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad \text{i} \quad C = \begin{bmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{bmatrix},$$

a rang dekompozicija nad \mathbb{C} s istim oznakama dana je s

$$A = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -i & i \end{bmatrix}, \quad B = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix} \quad \text{i} \quad C = \begin{bmatrix} 1 & 1 \\ i & -i \end{bmatrix}.$$

Zatim, rang matrice moguće je izračunati pomoću konačnog niza elementarnih transformacija, a za računanje ranga tenzora (višeg reda) takav izravan algoritam ne postoji - zapravo je riječ o NP-teškom problemu [46]. Kasnije ćemo vidjeti načine na koje se traži dobra aproksimacija ranga tenzora.

Posljednja razlika između dvaju rangova tiče se pojmova *maksimalnog* i *tipičnog* ranga matrice, odnosno tenzora.

Definicija 2.1.3. *Označimo s riječju objekt matricu ili tenzor. **Maksimalni rang** nekog objekta najveća je vrijednost koju rang tog objekta može postići s obzirom na dimenzije objekta. **Tipični rang** nekog objekta je bilo koja vrijednost koju rang objekta može postići s vjerojatnošću većom od nula kada se elementi objekta biraju na slučajan način iz uniformne neprekidne distribucije uz fiksirane dimenzije objekta.*

Za sve matrice $X \in \mathbb{R}^{I \times J}$ maksimalni rangovi su jednaki tipičnima i iznose $\min\{I, J\}$. Kod tenzora višeg reda ovo ne vrijedi, a realni među njima mogu imati čak i više od jednog tipičnog ranga. Primjeri takvih su tenzori iz $\mathbb{R}^{2 \times 2 \times 2}$ [42], pri čemu je vjerojatnost pojavljivanja tenzora ranga dva jednaka 79%, tenzora ranga tri 21%, a tenzora ranga jedan, iako postoje, 0%. Maksimalni i tipični rangovi tenzora reda tri različitih dimenzija i svojstava mogu se pronaći u [36] i referencama navedenim ondje.

2.1.3 Jedinstvenost rang dekompozicije

Važno svojstvo tenzora višeg reda jest česta jedinstvenost njihovih rang dekompozicija, što omogućuje jedinstvenu interpretaciju odnosa među modovima tenzora. Zanimljivo je da

isto svojstvo ne vrijedi i za tenzore reda dva, odnosno matrice. Primjerice, neka je $X \in \mathbb{R}^{I \times J}$ neka matrica tenzorskog ranga R , i neka je njena rang dekompozicija oblika

$$X = \sum_{r=1}^R a_r \circ b_r = AB^T.$$

Ako je $U\Sigma V^T$ dekompozicija singularnih vrijednosti od X , tada odabirimo $A_1 = U\Sigma$ i $B_1 = V$ te $A_2 = U\Sigma W$ i $B_2 = VW$, pri čemu je W neka ortogonalna matrica iz $\mathbb{R}^{R \times R}$, lako dobivamo dvije različite rang dekompozicije od X , $A_1 B_1^T$ i $A_2 B_2^T$. Mogli bismo reći kako je SVD tip matrice dekompozicije koji jest jedinstven, no jedan od razloga zašto je jedinstven je jaki uvjet ortogonalnosti.

Kada za rang dekompoziciju tenzora \mathcal{X} višeg reda kažemo da je jedinstvena, odnosno da ne postoji niti jedna drugačija kombinacija tenzora ranga jedan koji tvore \mathcal{X} , mislimo na jedinstvenost do na puno slabije uvjete od ortogonalnosti, odnosno do na permutacijsku neodređenost i neodređenost skaliranja. Permutacijska neodređenost dozvoljava proizvoljan poredak tenzora ranga jedan, tj. ako je \mathcal{X} iz $\mathbb{R}^{I \times J \times K}$ dan rang dekompozicijom

$$\mathcal{X} = \sum_{r=1}^R a_r \circ b_r \circ c_r = \llbracket A, B, C \rrbracket, \quad (2.5)$$

tada za bilo koju permutacijsku matricu $\Pi \in \mathbb{R}^{R \times R}$ vrijedi

$$\llbracket A, B, C \rrbracket = \llbracket A\Pi, B\Pi, C\Pi \rrbracket.$$

Neodređenost skaliranja dozvoljava uvjetno skaliranje faktora, tj. za sve realne skalare α_r , β_r i γ_r takve da je $\alpha_r \beta_r \gamma_r = 1$, gdje je $r = 1, \dots, R$, vrijedi

$$\mathcal{X} = \sum_{r=1}^R (\alpha_r a_r) \circ (\beta_r b_r) \circ (\gamma_r c_r).$$

Navedeno vrijedi za tenzore svih redova.

Navest ćemo neke rezultate koji govore o nužnim i dovoljnim uvjetima jedinstvenosti rang dekompozicije. Najpoznatiji uvjet dovoljnosti za tenzore reda tri dao je Kruskal [41][42], a uvjet koristi pojam k -ranga matrice.

Definicija 2.1.4. *k -rang matrice A , u oznaci k_A , je maksimalna vrijednost k tako da je bilo kojih k stupaca matrice A linearno nezavisno.*

Uvjet kaže da je za jedinstvenost rang dekompozicije iz (2.5) dovoljno da vrijedi

$$k_A + k_B + k_C \geq 2R + 2.$$

Prethodno vrijedi i za općenit red tenzora, tj. za jedinstvenost rang dekompozicije tenzora reda N dane sa

$$\mathcal{X} = \sum_{r=1}^R a_r^{(1)} \circ a_r^{(2)} \circ \dots \circ a_r^{(N)} = \llbracket A^{(1)}, A^{(2)}, \dots, A^{(N)} \rrbracket \quad (2.6)$$

dovoljno je da vrijedi

$$\sum_{n=1}^N k_{A^{(n)}} \geq 2R + (N - 1).$$

Taj uvjet je i nužan za jedinstvenost rang dekompozicija tenzora ranga dva i tri, no ne i tenzora višeg ranga. Općenitiji nužan uvjet za jedinstvenost dekompozicije iz (2.5) dan je sa

$$\min\{\text{rank}(A \odot B), \text{rank}(A \odot C), \text{rank}(B \odot C)\} = R,$$

a dekompozicije iz (2.6) sa

$$\min_{n=1, \dots, N} \text{rank}(A^{(1)} \odot \dots \odot A^{(n-1)} \odot A^{(n+1)} \odot \dots \odot A^{(N)}) = R.$$

2.1.4 Aproksimacije nižeg ranga

Osim jedinstvenosti, postoji još jedno svojstvo po kojemu se matrice i tenzori višeg ranga razlikuju, koje je ovog puta više naklonjeno matricama. Prema teoremu Eckarta i Younga iz [23], najbolja aproksimacija neke matrice matricom manjeg ranga k , ili *aproksimacija ranga k* , dana je vodećim k sumandima SVD-a te matrice, tj. za matricu A ranga R sa SVD dekompozicijom

$$A = \sum_{r=1}^R \sigma_r u_r \circ v_r \quad \text{uz} \quad \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_R > 0$$

vrijedi da je matrica ranga $k < R$ koja minimizira $\|A - B\|_2$ dana sa

$$B = \sum_{r=1}^k \sigma_r u_r \circ v_r.$$

Analogna tvrdnja za tenzore višeg reda i CP dekompoziciju ne vrijedi. Primjerice, u radu [34] na primjeru jednog kubičnog tenzora pokazano je da se najbolja aproksimacija ranga jedan tog tenzora ne nalazi kao faktor u najboljoj aproksimaciji ranga dva tenzora. Iz navedenog slijedi da se sumandi najbolje aproksimacije ranga k tenzora višeg reda općenito ne mogu računati sekvencijalno, već se moraju pronaći simultano.

Štoviše, za neke tenzore višeg reda najbolja aproksimacija ranga k uopće ne postoji, odnosno ti tenzori se mogu proizvoljno dobro aproksimirati CP dekompozicijom. Njih zovemo *degeneriranim tenzorima*.

Primjer 2.1.5. *Neka je \mathcal{X} tenzor dan rang dekompozicijom*

$$\mathcal{X} = a_1 \circ b_1 \circ c_2 + a_1 \circ b_2 \circ c_1 + a_2 \circ b_1 \circ c_1,$$

pri čemu su a_1 i a_2 , b_1 i b_2 te c_1 i c_2 linearno nezavisni. \mathcal{X} je degeneriran jer se može proizvoljno dobro aproksimirati tenzorom ranga dva danog sa

$$\mathcal{Y} = \alpha \left(a_1 + \frac{1}{\alpha} a_2 \right) \circ \left(b_1 + \frac{1}{\alpha} b_2 \right) \circ \left(c_1 + \frac{1}{\alpha} c_2 \right) - \alpha a_1 \circ b_1 \circ c_1 \quad \text{uz } \alpha \in \mathbb{R}, \alpha \neq 0,$$

jer vrijedi

$$\|\mathcal{X} - \mathcal{Y}\| = \frac{1}{\alpha} \left\| a_2 \circ b_2 \circ c_1 + a_2 \circ b_1 \circ c_2 + a_1 \circ b_2 \circ c_2 + \frac{1}{\alpha} a_2 \circ b_2 \circ c_2 \right\|,$$

što teži ka nuli za velike α .

Dodatno o primjerima i svojstvima degeneriranih tenzora može se pronaći u referencama iz [36], od kojih jedno pokazuje kako problem nepostojanja najbolje aproksimacije nije rijedak događaj, što ukazuje na važnost njegovog proučavanja.

2.1.5 Računanje CP dekompozicije

Određivanje broja komponenti

Već znamo kako uopće nije lako naći rang tenzora, tj. da je čak riječ o NP-teškom problemu. Iako jest predmet matematičke znatiželje, taj problem ne donosi poteškoće u primjerima u praksi; u njima je, zbog šuma i nesavršenosti u podacima, broj zaista bitnih i smislenih komponenti CP dekompozicije nekog tenzora obično puno manji od njegovog pravog ranga ([66]).

Jedan od načina na koji se u praksi može odrediti taj "korisni" rang tenzora heurističke je prirode - odabere se konačno mnogo različitih vrijednosti za rang te se računaju odgovarajuće CP dekompozicije (nekim algoritmom) i kao konačno rješenje odabere ona dekompozicija koja je najbliža početnom tenzoru po nekoj kombinaciji kriterija. Uobičajeni kriteriji su suma kvadriranih grešaka i postotak objašnjene varijance (explained variance), no kada ima velikog udjela šuma u podacima, što je često slučaj, koristi se i alat CORCONDIA, od Core Consistency Diagnostic, koji se temelji na činjenici da se CP dekompozicija može svesti na Tuckerovu dekompoziciju, o kojoj ćemo uskoro nešto više reći. Ne preporučuje se zadržati se samo na kriteriju poput sume kvadriranih grešaka i iz razloga što početni tenzor može biti degenerirani te za njega mogu postojati aproksimacije ranga nižeg i od onog korisnog koje su mu proizvoljno blizu, odnosno bliže od idealne dekompozicije s korisnim rangom.

Posljednjih godina razvili su se i modeli koji koriste konvolucijske neuronske mreže za određivanje ranga, prema [81].

Nakon fiksiranog broja komponenti

Kada je broj komponenti za CP dekompoziciju fiksiran nekom vrijednošću R , problem traženja CP dekompozicije tenzora $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ (a i tenzora višeg reda) svodi se na problem traženja tenzora $\hat{\mathcal{X}} \in \mathbb{R}^{I \times J \times K}$ oblika

$$\hat{\mathcal{X}} = \sum_{r=1}^R \lambda_r a_r \circ b_r \circ c_r = \llbracket \lambda; A, B, C \rrbracket \quad (2.7)$$

koji najbolje aproksimira \mathcal{X} , tj. za kojeg vrijedi

$$\min_{\hat{\mathcal{X}}} D(\mathcal{X}, \hat{\mathcal{X}}). \quad (2.8)$$

Pri tome funkcija D predstavlja mjeru blizine dvaju tenzora, odnosno kvalitetu aproksimacije \mathcal{X} s $\hat{\mathcal{X}}$. Kada je D jednako normi, imamo

$$\min_{\hat{\mathcal{X}}} \|\mathcal{X} - \hat{\mathcal{X}}\|, \quad (2.9)$$

što je nelinearna i nekonveksna verzija problema najmanjih kvadrata. Zbog tih svojstava mogući su optimizacijski problemi s lokalnim optimumima.

CP-ALS

Najčešći algoritam koji se koristi za rješavanje problema iz (2.9) služi se iterativnom metodom alternirajućih najmanjih kvadrata (engl. Alternating Least Squares, ALS), pa je nazvan CP-ALS. Njegove korake ćemo objasniti za tenzore reda tri, dok se pseudokod za tenzore općenitog reda nalazi u Algoritmu 1.

Na samom početku inicijaliziraju se faktorske matrice. Prema [68], to se može učiniti sofisticiranim metodama poput metode GRAM (Generalized Rank Annihilation Method) ili metode DTD (Direct Trilinear Decomposition), s rezultatima blizu pravog rješenja što dovodi do manjeg broja iteracija CP-ALS-a i uštede u vremenu. Inicijalne faktorske matrice se mogu dobiti i uzimanjem početnih R singularnih vektora SVD dekompozicije odmotavanja tenzora \mathcal{X} u odgovarajućem modu, kada CP-ALS također brže konvergira ka pravom rješenju. No, kod navedenih pristupa teško je ustvrditi je li konačno rješenje uopće stacionarna točka, a kamoli lokalni ili, što nam je cilj, globalni minimum. Taj problem može se dijelom riješiti pomoću više slučajnih inicijalizacija - ako su sva pripadajuća konačna rješenja jednaka, teško je moguće da su ona ista nestacionarna točka ili lokalni optimum.

Nakon inicijalizacije, u CP-ALS-u se zatim fiksiraju dvije od tri matrice A , B i C te se rješava sada linearni problem najmanjih kvadrata za preostalu, treću matricu. Primjerice,

fiksiranjem matrica B i C te zapisom tenzora u obliku odmotavanja u modu jedan, zadatak (2.9) postaje

$$\min_{\hat{A}} \|X_{(1)} - \hat{A}(C \odot B)^\top\|_F,$$

uz $\hat{A} = A \cdot \text{diag}(\lambda)$. Njegovo optimalno rješenje dano je s

$$\hat{A} = X_{(1)}[(C \odot B)^\top]^\dagger,$$

koje se, koristeći svojstvo (4) iz Propozicije 1.3.16 koje povezuje Khatri-Raov i Hadamardov produkt te Moore-Penroseov inverz, može zapisati u naizgled računski pogodnijem obliku

$$\hat{A} = X_{(1)}(C \odot B)[(C^\top C) * (B^\top B)]^\dagger \quad (2.10)$$

jer dopušta računanje pseudoinverza (obično manje) $R \times R$ matrice umjesto (obično veće) $JK \times R$ matrice, iako taj pristup može dovesti do numeričkih problema. Normiranjem stupaca od \hat{A} radi stabilnosti algoritma dobivamo matricu A , tj. postavljamo $\lambda_r = \|\hat{a}_r\|$ i $a_r = \hat{a}_r / \lambda_r$. Navedeni postupak se provodi za sve tri moguće kombinacije fiksniranih matrica i ponavlja sve dok se ne zadovolji neki od kriterija konvergencije. Kriteriji su obično neznatno poboljšanje u ciljnoj funkciji u uzastopnim iteracijama, ne mijenjanje faktorskih matrica, ciljna funkcija blizu nule ili dostignut maksimalan broj iteracija.

Algoritam 1: CP-ALS algoritam za računanje CP dekompozicije s R komponenta tenzora $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$

procedura CP-ALS(\mathcal{X}, R)

inicijaliziraj $A^{(n)} \in \mathbb{R}^{I_n \times R}$ za $n = 1, \dots, N$

ponavljaj

za $n = 1, \dots, N$ **čini**

$V \leftarrow A^{(1)\top} A^{(1)} * \dots * A^{(n-1)\top} A^{(n-1)} * A^{(n+1)\top} A^{(n+1)} * \dots * A^{(N)\top} A^{(N)}$

$A^{(n)} \leftarrow X^{(n)}(A^{(N)} \odot \dots \odot A^{(n+1)} \odot A^{(n-1)} \odot \dots \odot A^{(1)})V^\dagger$

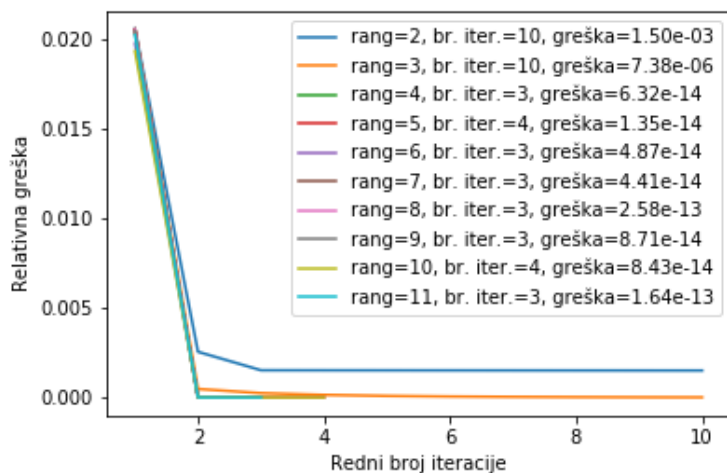
 normaliziraj stupce od $A^{(n)}$ spremajući norme u λ

kraj

dok ne bude konvergiralo;

vрати $\lambda, A^{(1)}, A^{(2)}, \dots, A^{(N)}$

Demonstrirat ćemo na jednostavnom primjeru kako se za različite rangove CP dekompozicije relativna greška mijenja s povećanjem rednog broja iteracija. Relativna greška se računa formulom $\|\mathcal{X} - \hat{\mathcal{X}}\| \setminus \|\mathcal{X}\|$, gdje su \mathcal{X} i $\hat{\mathcal{X}}$ redom originalni tenzor i njegova rekonstrukcija iz CP dekompozicije. Neka tenzor $\mathcal{X} \in \mathbb{R}^{12 \times 12 \times 12}$ čine brojevi $0, 1, 2, \dots, 12^3 - 1$. Ograničimo li maksimalan broj iteracija na 10 i koristimo malu toleranciju greške, u jednom od pokretanja funkcije parafac iz sučelja Tensorly ([38]) dobivamo ponašanje koje se može vidjeti na Slici 2.2.



Slika 2.2: Mijenjanje relativne greške CP dekompozicije ovisno o rednom broju iteracije za različite rangove.

Iako su relativne greške za sve rangove male, za rangove veće od 3 one su ipak bitno manje i dostižu toleranciju greške već u trećoj ili četvrtoj iteraciji, dok se za rangove 2 i 3 CP dekompozicija mora računati sve do maksimalnog broja iteracija. Kako je razlika između relativne greške kod ranga 4 i manjih relativnih grešaka od nje zanemariva, mogli bismo se za aproksimaciju od \mathcal{X} CP dekompozicijom odlučiti za onu ranga 4.

Korake CP-ALS-a nije teško razumjeti, no već smo kod govora o inicijalizaciji mogli uočiti opasnosti koje se javljaju kod ove metode: moguć je veliki broj iteracija potreban za konvergenciju, nije zagarantirana kovergencija (unatoč monotonom smanjivanju vrijednosti ciljne funkcije) te ovisnost konačnog rješenja o inicijalnim matricama. Također, umnožak $X_{(1)}(C \circ B)$ Khatri-Raovog produkta s matriciziranim tenzorom iz (2.10), u užoj literaturi poznat pod kraticom MTTKRP ili "Matricized Tensor Times Khatri-Rao Product", predstavlja usko grlo u računanju [57]. Tehnike kojima se može ubrzati CP-ALS uključuju *enhanced line search* te Tikhonovu regularizaciju, što je spomenuto u [36], a neke sugestije za osiguravanje konvergencije mogu se naći u [66]. U nedavnom istraživanju [10] proširujući randomiziranu metodu najmanjih kvadrata na tenzore dobivena su poboljšanja u brzini kovergencije, memorijskim zahtjevima i robusnosti s obzirom na inicijalizaciju te se potpuno izbjeglo računanje Khatri-Raovog produkta.

Druge metode

Neki algoritmi za računanje CP dekompozicije, za razliku od CP-ALS-a, za cilj ipak imaju rješavanje nelinearne verzije problema najmanjih kvadrata iz (2.9), tj. istodobno optimi-

ziraju sve faktorske matrice. Metode koje se pritom koriste su obična metoda silaska gradijenta i napredniji pristupi poput Gauss-Newton, Levenberg-Marquard ili prigušeni (eng. damped) Gauss-Newtona te nelinearnog konjugiranog gradijenta [32]. Detaljniji pregled algoritama koji računaju CP dekompoziciju i njihova usporedba može se naći u [66] i [57].

2.1.6 Općenita primjena

Prvo područje u kojem je CP dekompozicija našla primjenu bila je psihometrija. U radu Carrolla i Changa [14] analizirale su se razlike među osobama na temelju zvučnih podataka i percepcijama nacija, a u Harshmanovom radu [25], koji je sadržavao i prve rezultate o uvjetima jedinstvenosti, proučavale su se osobine glasova na temelju podataka o ljudima (prva dimenzija) koji bi izgovarali određene glasove (druga dimenzija) uz zabilježena mjerenja izgovora, npr. visinu tona (treća dimenzija).

U kemometriji se CP dekompozicija počela primjenjivati u [6], a pregled nekih istraživanja dan je u [4]. Obrada signala senzora [64] i telekomunikacije [67, 65, 17] također su neka od područja primjene.

CP dekompozicija se otkrila neovisno u neuroznanosti u [51] analiziranjem potencijala u snimkama mozga. U [3] su se za bolju interpretaciju i jedinstvenost višedimenzionalnih fMRI podataka bilinearne metode poput PCA zamijenile CP dekompozicijom. Nemali broj radova tiče se analize EEG podataka i njihove vremenske komponente [49, 50, 52]. U [2, 19, 20] su se CP dekompozicijom nastojali lokalizirati početci epileptičkih napadaja. Neka novija istraživanja pomoću fMRI mjerenja i CP dekompozicije žele bolje objasniti regije mozga i veze među njima [16], a neka klasificiraju snimke mozga pacijenata s različitim bolestima i oštećenjima [26].

Predviđanje linkova na temelju onih iz prošlosti [21], analiza teksta u e-mailovima [8, 58], uočavanje uzoraka ponašanja iz objava na Facebooku [56, 28] te identificiranje običnih i akademskih zajednica ljudi na temelju različitih načina komunikacije i podataka o objavljenim radovima [7, 55] neke su od primjena CP dekompozicije u analizi društvenih mreža.

O primjeni u rudarenju weba, pretraživanju informacija, sustavu preporuka, analizi prometa računalnih mreža, obradi govora, kompresiji i klasifikaciji slika može se pronaći u [36] i [57].

2.1.7 Primjena u prepoznavanju radnje

CP dekompozicija se nije koristila puno u ovom području, za razliku od sljedeće dekompozicije koju ćemo opisati (Tuckerova), no postoje barem dva rada koja su je koristila i koja su ostvarila značajnije rezultate na poznatim datasetovima, a njih ćemo ovdje pobliže opi-

sati. Svi rezultati koje ćemo spominjati u ovim primjenama postignuti su od strane autora članaka.

Nenegativna CP dekompozicija videa i zapis snimki pomoću baznih slika iz [40]

U ovom radu videi su nizovi snimki organizirani u tenzore \mathcal{G} trećeg reda koji se nastoje aproksimirati nenegativnom CP dekompozicijom kao $\mathcal{G} = \sum_{j=1}^K u^j \otimes v^j \otimes w^j$, gdje su $u^j, v^j, w^j \geq 0$. Zapravo, svaka se snimka videa G_t , tj. frontalni odsječak od \mathcal{G} , želi prikazati kao $\sum_{j=1}^K w^{t,j}(u^j \otimes v^j)$, gdje se matrica $B_j = u^j \otimes v^j$ naziva baznom matricom. Tako je svaka snimka B_t linearna kombinacija K baznih slika B_j i K koeficijenata $w^{t,j}$. Na datasetovima je pokazano kako se iza baznih slika i koeficijenata doista krije određeno značenje - bazne slike predstavljaju dijelove tijela u određenim položajima, primjerice uspravnu ili nakošenu glavu ili trup, dok koeficijent označava koliko je taj položaj prisutan u trenutnoj snimci.

Bazne slike se ekstrahiraju CP dekompozicijom iz niza snimki, a koeficijenti se dobivaju tzv. "filtriranjem" snimaka baznim slikama. Tako se uočavaju sve akcije, odnosno nizovi poza ili linearnih kombinacija baznih slika i koeficijenata. Kada se želi prepoznati neka akcija na novom videu, on se također filtrira baznim slikama uz dobivanje $T \cdot K$ koeficijenata $w^{t,j}$. No, kako bi se u obzir uzele razlike u trajanju akcija, umjesto proučavanja jednog vektora svih koeficijenata, proučava se više vektora duljine $N \cdot K$ s pomakom u odabiru koeficijenata: $(w^{1,1}, \dots, w^{N,1}, \dots, w^{1,K}, \dots, w^{N,K}), (w^{2,1}, \dots, w^{N+1,1}, \dots, w^{2,K}, \dots, w^{N+1,K})$ itd. Time se dobiva $(T - N + 1) \cdot K$ vektora značajki kojima se potom nalazi najbliži susjed među videima za treniranje.

Datsetovi na kojima se testirao ovaj algoritam su Weizmann i TUM dataset. Prvi se sastoji od 10 akcija (trčanje, hodanje, preskakivanje, "jumping-jack", skakanje naprijed, skakanje na mjestu, galopiranje u stranu, mahanje jednom i dvjema rukama te saginjanje) koje izvodi 9 ljudi, a drugi prikazuje nekoliko ljudi koji postavljaju stol, prateći lijevu i desnu ruku te trup. Na Weizmann datasetu je uz "leave-one-out" kros-validaciju, 20 baznih slika i parametar $N = 10$ ostvarena točnost od 98%, dok je na TUM datasetu uzimajući sve tri značajke u obzir postignuto 46.6%, izbacujući trup 54%, a ostavljajući samo lijevu ili samo desnu ruku 67% točnosti.

Stroj s potpornim tenzorima višeg ranga iz [39]

Ovdje se dvoklasno klasifikacije tenzorskih podataka rješava metodom stroja s potpornim tenzorima višeg ranga i njegovim dvjema varijacijama. Pri tome se parametri koji određuju razdvajajuću hiperravninu organiziraju u tenzor $\mathcal{W} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$ koji se želi prikazati sumom tenzora ranga jedan, tj. CP dekompozicijom.

Uz podatke za treniranje $\mathcal{X}_i \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M}$ i labela $y_i \in \mathbb{R}$, za $i = 1, \dots, L$, problem stroja s potpornim tenzorima dan je s

$$\min_{\mathcal{W}, b, \xi \geq 0} \frac{1}{2} \langle \mathcal{W}, \mathcal{W} \rangle + C \sum_{i=1}^L \xi_i \quad \text{t.d.} \quad y_i (\langle \mathcal{W}, \mathcal{X}_i \rangle + b) \geq 1 - \xi_i, \quad 1 \leq i \leq L, \quad \xi_i \geq 0, \quad (2.11)$$

pri čemu je b pristranost, $\xi = [\xi_1, \dots, \xi_L]$ vektor varijabli koje mjere pogreške u podacima, a C faktor koji penalizira greške kod treniranja. Gornja funkcija čiji se minimum traži nije konveksna, stoga se problem rješava iterativno, zamjenjivanjem \mathcal{W} s kombinacijom matricizacije u modu j i CP dekompozicije, uz fiksiranje svih parametara različitih j . Nakon raznih supstitucija, (2.11) se može svesti na probleme stroja s potpornim vektorima (SVM):

$$\min_{\tilde{U}^{(j)}, b, \xi \geq 0} \frac{1}{2} \text{vec}(\tilde{U}^{(j)})^\top \text{vec}(\tilde{U}^{(j)}) + C \sum_{i=1}^L \xi_i \quad \text{t.d.} \quad y_i [\text{vec}(\tilde{U}^{(j)})^\top \text{vec}(\tilde{X}_{(j)i}) + b] \geq 1 - \xi_i,$$

$$1 \leq i \leq L, \quad \xi_i \geq 0.$$

Tijekom testiranja testni se tenzori \mathcal{Y} klasificiraju funkcijom $g: \mathbb{R}^{I_1 \times I_2 \times \dots \times I_M} \rightarrow [-1, 1]$, $g(\mathcal{Y}) = \text{sign}(\langle \mathcal{W}, \mathcal{Y} \rangle + b)$.

Jedna od varijacija algoritma uzima u obzir matrice raspršenja unutar i između klasa, što rezultira približavanjem podataka srednjoj vrijednosti pripadne klase te međusobnom udaljavanju klasa, čime ih se može bolje razdvojiti. Druga varijacija ograničuje odozgo udaljenost točaka od razdvajajuće tenzorske ravnine skalarom B uz istodobno maksimiziranje margine. Time se može omogućiti manja raspršenost projekcija točaka na hiperravnine okomite na razdvajajuće hiperravnine, tj. projekcije su kompaktnije pa je klase lakše razdvojiti. Ilustracija ovoga može se pronaći u radu. Obje varijacije se uz vlastite supstitucije svode na SVM problem, s time da druga varijacija još zahtjeva uvjet u tenzorskoj, tj. vektorskoj formi,

$$\frac{1}{2} (\langle \mathcal{W}, \mathcal{X} \rangle + b)^2 \leq \frac{B^2}{2}, \quad \text{tj.} \quad \frac{1}{2} [\text{vec}(\tilde{U}^{(j)})^\top \text{vec}(\tilde{X}_{(j)i}) + b]^2 \leq \frac{B^2}{2}.$$

Metode su se testirale na datasetovima Weizmann i KTH, pri čemu se potonji sastoji od videa sa šest različitih kretnji (boksanje, pljeskanje, mahanje, joggiranje, trčanje, hodanje) koje izvodi 25 osoba. Videi su izrađeni brzinom od 25 sličica po sekundi te su iz njih ekstrahirane bitne značajke. Performanse klasifikatora su se računale "leave-one-person-out" kros-validacijom. Na Weizmann datasetu su za tri klasifikatora najbolje ostvarene točnosti bile 95.6%, 97.8% i 100% uz rang jednak 6, a uz rang jednak 1 rezultati su za oko 5% niži. Na KTH datasetu ostvarene su točnosti 93.3%, 94.7% i 96.7%, također uz rang jednak 6, te uz oko 3% manje rezultate za rang jednak 1. U prvom datasetu ostvarena su bitna poboljšanja u odnosu na dotadašnje najbolje rezultate, a u drugom datasetu to vrijedi u slučaju trećeg klasifikatora.

2.2 Tuckerova dekompozicija

Ovu dekompoziciju razvio je Tucker 1963. u području psihometrije i, uz Levina, profinio u narednim godinama. O Tuckerovoj dekompoziciji može se razmišljati i kao o generalizaciji PCA-a odnosno SVD-a, u što ćemo se uvjeriti u narednom tekstu, a imena pod kojim je sve poznata su *Three-mode factor analysis* (3MFA/Tucker3), *Three-mode PCA* (3MPCA), *N-mode PCA*, *Higher-order SVD* (HOSVD) te *N-mode SVD*.

2.2.1 Osnovno

U Tuckerovoj dekompoziciji tenzor se rastavlja na *jezgreni* tenzor pomnožen s faktorskim matricama uzduž svakog moda. Tako se tenzor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ reda tri može prikazati kao

$$\mathcal{X} \approx \mathcal{G} \times_1 A \times_2 B \times_3 C, \quad (2.12)$$

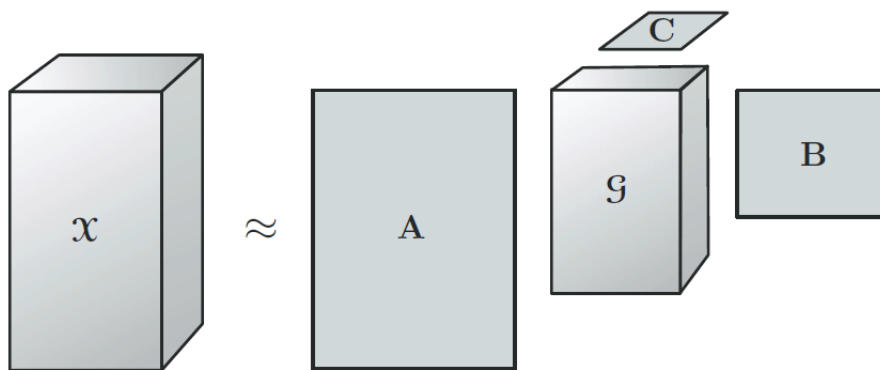
pri čemu je $\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$ jezgreni tenzor, a faktorske matrice su $A \in \mathbb{R}^{I \times P}$, $B \in \mathbb{R}^{J \times Q}$ i $C \in \mathbb{R}^{K \times R}$ sa stupcima koje nazivamo faktorima. Drugi izraz u (2.12) jednak je drugom izrazu u

$$\mathcal{X} \approx \sum_{p=1}^P \sum_{q=1}^Q \sum_{r=1}^R g_{pqr} a_p \circ b_q \circ c_r, \quad (2.13)$$

a koristi se i kompaktan zapis analogan onome kod CP dekompozicije,

$$\mathcal{X} \approx \llbracket G; A, B, C \rrbracket. \quad (2.14)$$

Ilustracija Tuckerove dekompozicije je na Slici 2.3.



Slika 2.3: Tuckerova dekompozicija tenzora reda tri iz [36].

U (2.13) možemo uočiti kako Tuckerova dekompozicija opisuje interakciju svakog faktora iz A (a_p) sa svakim faktorom iz B (b_q) i sa svakim faktorom iz C (c_r), jačina kojeg odnosa je pohranjena u odgovarajućem elementu jezgrenog tenzora \mathcal{G} (g_{pqr}). Ovo se razlikuje od CP dekompozicije iz (2.1) jer se u njoj računa vanjski produkt samo faktora jednakih indeksa.

Ekvivalentni zapis po elementima od (2.12) glasi

$$x_{ijk} \approx \sum_{p=1}^P \sum_{q=1}^Q \sum_{r=1}^R g_{pqr} a_{ip} b_{jq} c_{kr} \quad \text{za } i = 1, \dots, I, j = 1, \dots, J, k = 1, \dots, K,$$

a prema dokazanome u Propoziciji 1.3.13, odmotavanja u trima modovima tenzora \mathcal{X} iz (2.12) dana su sa

$$\begin{aligned} X_{(1)} &\approx AG_{(1)}(C \otimes B)^\top, \\ X_{(2)} &\approx BG_{(2)}(C \otimes A)^\top \quad \text{i} \\ X_{(3)} &\approx CG_{(3)}(B \otimes B)^\top. \end{aligned}$$

Tuckerovom dekompozicijom često se nastoji komprimirati originalni tenzor jer pohranjivanje dekompozicije može zauzeti značajno manje prostora od pohranjivanja samog tenzora, a da se pri tome ne izgube podaci iz tenzora. Tada su brojevi komponenata u faktorskim matricama P , Q i R iz (2.12) bitno manji od I , J i K , redom, a tenzor \mathcal{G} smatramo komprimiranom verzijom od \mathcal{X} . Primjerice, ako imamo tenzor $\mathcal{X} \in \mathbb{R}^{1000 \times 1000 \times 1000}$ s Tuckerovom dekompozicijom koja se sastoji od jezgrenog tenzora $\mathcal{G} \in \mathbb{R}^{10 \times 10 \times 10}$ i faktorskih matrica A , B i C iz $\mathbb{R}^{1000 \times 10}$, tada umjesto spremanja $1000^3 = 10^9$ elemenata od \mathcal{X} spremamo samo $10^3 + 3 \cdot 100 \cdot 10 = 4 \cdot 10^3$ elemenata dekompozicije.

U algoritmima koji nalaze Tuckerovu dekompoziciju nerijetko se pretpostavlja da su faktorske matrice ortogonalne, odakle dijelom dolazi usporedba s PCA-om. Ta pretpostavka nije nužna; čak se i svaka CP dekompozicija, koja ne zahtjeva uvjet ortogonalnosti, može svesti na Tuckerovu dekompoziciju, odabere li se u (2.13) za jezgri tenzor dijagonalni tenzor s jedinicama na dijagonali.

Slijedi formalna definicija ove dekompozicije za tenzor općenitog reda, prema [60].

Definicija 2.2.1. Za tenzor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ reda N kažemo da se nalazi u Tuckerovom formatu ako se može prikazati kao

$$\mathcal{X} = \mathcal{G} \times_1 A^{(1)} \times A^{(2)} \dots \times_N A^{(N)} = \llbracket \mathcal{G}; A^{(1)}, A^{(2)}, \dots, A^{(N)} \rrbracket, \quad (2.15)$$

gdje $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_N}$ nazivamo jezgri tenzorom, a matrice $A^{(n)} \in \mathbb{R}^{I_n \times R_n}$ faktorskim matricama, za $R_n \in \mathbb{N}$ i $n = 1, 2, \dots, N$. Elementarno,

$$x_{i_1 i_2 \dots i_N} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N} a_{i_1 r_1}^{(1)} a_{i_2 r_2}^{(2)} \dots a_{i_N r_N}^{(N)} \quad \text{za } i_n = 1, \dots, I_n, n = 1, \dots, N.$$

Matricizirana verzija od (2.15) dana je, opet prema Propoziciji 1.3.13, sa

$$X_{(n)} = A^{(n)} G_{(n)} (A^{(N)} \otimes \dots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \dots \otimes A^{(1)})^\top.$$

Postoje i važne su varijacije Tuckerove dekompozicije kod kojih je podskup faktorskih matrica jednak identičnim matricama. Primjerice, *Tucker2* dekompozicija tenzora \mathcal{X} reda tri, u kojoj je matrica C iz (2.12) jednaka identiteti $I \in \mathbb{R}^{K \times K}$, dana je s

$$\mathcal{X} = \mathcal{G} \times_1 A \times_2 B = \llbracket \mathcal{G}; A, B, I \rrbracket,$$

a *Tucker1* dekompozicija tenzora \mathcal{X} , u kojoj su B i C identične matrice, dana je s

$$\mathcal{X} = \mathcal{G} \times_1 A = \llbracket \mathcal{G}; A, I, I \rrbracket.$$

Posljednje je ekvivalentno i dvodimenzionalnom PCA jer je $X_{(1)} = AG_{(1)}$.

2.2.2 n -rang i multirang

Slijede definicije dvaju pojmova usko vezanih uz Tuckerovu dekompoziciju.

Definicija 2.2.2. n -rang tenzora $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, koji se označava kao $\text{rang}_n(\mathcal{X})$, je rang odmotavanja tenzora \mathcal{X} u modu n , tj. vrijedi

$$\text{rang}_n(\mathcal{X}) = \text{rang}(X_{(n)}).$$

Drugim riječima, n -rang je dimenzija vektorskog prostora generiranog nitima u modu n tenzora \mathcal{X} .

Definicija 2.2.3. *Multirang* tenzora $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ je N -torka svih n -rangova od \mathcal{X} za $n = 1, 2, \dots, N$, tj. jednak je

$$(\text{rang}_1(\mathcal{X}), \text{rang}_2(\mathcal{X}), \dots, \text{rang}_N(\mathcal{X})).$$

Teorem koji slijedi govori kako za svaki tenzor možemo naći njegovu egzaktnu Tuckerovu dekompoziciju. U komentaru iza tog teorema uvjerit ćemo se i kako za svaki tenzor \mathcal{X} iz Definicije 2.2.1 postoji egzaktna Tuckerova dekompozicija za koju je (R_1, R_2, \dots, R_N) jednako multirangu od \mathcal{X} , tj. vrijedi $R_n = \text{rang}_n(\mathcal{X})$ za $n = 1, 2, \dots, N$. Takva dekompozicija među svim egzaktnim Tuckerovim dekompozicijama od \mathcal{X} ima najmanji mogući jezgri tenzor i smanjenje pohrane tenzora u vidu dekompozicije tada je maksimizirano, prema [60]. Međutim, ponekad su neke komponente multiranga tenzora prevelike da bi se značajnije uštedjelo na pohrani tenzora; u takvim situacijama se računa aproksimativna Tuckerova dekompozicija koja zadovoljava $R_n < \text{rang}_n(\mathcal{X})$ za jedan ili više n -ova.

2.2.3 Računanje Tuckerove dekompozicije

HOSVD

Tucker je 1966. objavio nekoliko metoda za nalaženje Tuckerove dekompozicije, a mi ćemo proučiti jednu koja je kasnije, u radu [18], nazvana *higher-order singular value decomposition* ili HOSVD. Ta metoda temelji se na sljedećem teoremu, a koji opravdava naziv metode, tj. činjenicu da je riječ o generalizaciji SVD-a.

Teorem 2.2.4. [18, Teorem 2] Svaki tenzor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ može se zapisati kao

$$\mathcal{X} = \mathcal{G} \times_1 A^{(1)} \times_2 A^{(2)} \times_3 \dots \times_N A^{(N)}, \quad (2.16)$$

pri čemu je

$$(1) A^{(n)} = \begin{bmatrix} a_1^{(n)} & a_2^{(n)} & \dots & a_{I_n}^{(n)} \end{bmatrix} \in \mathbb{R}^{I_n \times I_n} \text{ ortogonalna matrica,}$$

$$(2) \mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N} \text{ tenzor čiji podtenzori } \mathcal{G}_{i_n=\alpha}, \text{ nastali fiksiranjem } n\text{-tog indeksa na } \alpha, \text{ zadovoljavaju svojstvo potpune ortogonalnosti}$$

$$\langle \mathcal{G}_{i_n=\alpha}, \mathcal{G}_{i_n=\beta} \rangle = 0, \text{ za } \alpha \neq \beta$$

i poredak

$$\|\mathcal{G}_{i_n=1}\|_F \geq \|\mathcal{G}_{i_n=2}\|_F \geq \dots \geq \|\mathcal{G}_{i_n=I_N}\|_F \geq 0,$$

za sve moguće vrijednosti od n . Norme $\|\mathcal{G}_{i_n=i}\|_F$, simbolizirane sa $\sigma_i^{(n)}$, su singularne vrijednosti, a $a_i^{(n)}$ singularni vektori odmotavanja tenzora \mathcal{G} u modu n .

Dokaz. Pretpostavimo da za dva tenzora \mathcal{X} i \mathcal{G} iz $\mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ vrijedi jednakost iz (2.15) uz uvjet da su $A^{(1)}, A^{(2)}, \dots, A^{(N)}$ ortogonalne matrice. Prema Propoziciji 1.3.13, ta jednakost ekvivalentna je

$$X_{(n)} = A^{(n)} G_{(n)} (A^{(N)} \otimes \dots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \dots \otimes A^{(1)})^\top. \quad (2.17)$$

Promotrimo slučaj u kojem bi matrica $A^{(n)}$ bila još i matrica lijevih singularnih vektora od $X_{(n)}$, tj. takva da vrijedi

$$X_{(n)} = A^{(n)} \Sigma^{(n)} B^{(n)\top}, \quad (2.18)$$

pri čemu je $B^{(n)}$ ortogonalna i $\Sigma^{(n)} = \text{diag}(\sigma_1^{(n)}, \sigma_2^{(n)}, \dots, \sigma_{I_n}^{(n)})$ sa

$$\sigma_1^{(n)} \geq \sigma_2^{(n)} \geq \dots \geq \sigma_{R_n}^{(n)} > \sigma_{R_n+1}^{(n)} = \dots = \sigma_{I_n}^{(n)} = 0$$

i $R_n \leq I_n$. Izjednačavanjem (2.17) i (2.18) te korištenjem svojstva (d) iz Propozicije 1.3.12 dobivamo

$$G_{(n)} = \Sigma^{(n)} B^{(n)\top} (A^{(N)} \otimes \cdots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \cdots \otimes A^{(1)}).$$

Odatle slijedi da za proizvoljne ortogonalne matrice $A^{(1)}, A^{(2)}, \dots, A^{(n-1)}, A^{(n+1)}, \dots, A^{(N)}$ matrica $G_{(n)}$ ima ortogonalne retke. Kako su redci od $G_{(n)}$ dani s $\text{vec}(\mathcal{G}_{i_n=1}), \text{vec}(\mathcal{G}_{i_n=2}), \dots, \text{vec}(\mathcal{G}_{i_n=I_n})$, prema Propoziciji 1.3.4 (3) vrijedi

$$\text{vec}(\mathcal{G}_{i_n=\alpha})^\top \text{vec}(\mathcal{G}_{i_n=\beta}) = \langle \mathcal{G}_{i_n=\alpha}, \mathcal{G}_{i_n=\beta} \rangle = 0, \text{ za } \alpha \neq \beta.$$

Također, prema Propoziciji 1.3.4 (2) je

$$\|\mathcal{G}_{i_n=1}\|_F = \|\text{vec}(\mathcal{G}_{i_n=1})\|_2 = \|(G_{(n)})_1\|_2 = \sigma_1^{(n)}.$$

Konstruiramo li matrice $A^{(1)}, \dots, A^{(n-1)}, A^{(n+1)}, \dots, A^{(N)}$ na isti način kao i $A^{(n)}$, tenzor \mathcal{G} će zadovoljiti sve uvjete teorema. Izračunamo li svaki $A^{(n)}$ iz SVD-a od $X_{(n)}$, pomoću svojstva (c) iz Propozicije 1.3.13 tenzor \mathcal{G} dobivamo formulom

$$\mathcal{G} = \mathcal{X} \times_1 A^{(1)\top} \times_2 A^{(2)\top} \times_3 \cdots \times_N A^{(N)\top}.$$

□

Pseudokod HOSVD-a nalazi se u Algoritmu 2. Izjednačimo li $X_{(n)}$ u (2.18) s njezinim skraćenim oblikom SVD-a, odnosno promatramo li samo prvih $\text{rang}_n(\mathcal{X})$ stupaca iz originalnog $A^{(n)}$, dobit ćemo egzaktnu Tuckerovu dekompoziciju od \mathcal{X} s dimenzijom jezgrenog tenzora jednakom multirangu od \mathcal{X} i faktorskim matricama $A^{(n)} \in \mathbb{R}^{I_n \times \text{rang}_n(\mathcal{X})}$. U HOSVD-u vrijedi $R_n \geq \text{rang}_n(\mathcal{X})$ za $n = 1, 2, \dots, N$, a tražimo li Tuckerovu dekompoziciju istom metodom uz $R_n < \text{rang}_n(\mathcal{X})$ za jedan ili više n -ova, dobivamo skraćeni (eng. *truncated*) HOSVD, koji nužno ne daje egzaktnu Tuckerovu dekompoziciju.

Algoritam 2: HOSVD tenzora \mathcal{X} reda $I_1 \times I_2 \times \cdots \times I_N$.

procedura HOSVD($\mathcal{X}, R_1, R_2, \dots, R_N$)

za $n = 1, \dots, N$ **čini**

$A^{(n)} \leftarrow R_n$ vodećih lijevih singularnih vektora od $X_{(n)}$

kraj

$\mathcal{G} \leftarrow \mathcal{X} \times_1 A^{(1)\top} \times_2 A^{(2)\top} \cdots \times_N A^{(N)\top}$

vrti $\mathcal{G}, A^{(1)}, A^{(2)}, \dots, A^{(N)}$

U [57] saznajemo kako se mogu odrediti argumenti R_1, R_2, \dots, R_N skraćenog HOSVD-a, a i sljedećeg algoritma kojeg ćemo opisati. Jedan način se zove DIFFIT, od *difference in fit*, a traži dekompoziciju s vrijednostima R_i koje zadovoljavaju $\sum_{i=1}^N R_i = s$ za različite vrijednosti od s koja najbolje aproksimira tenzor. Navedena verzija zahtijeva računanje više

Tuckerovih dekompozicija pa usporava nalaženje optimalnih argumenata, no osmišljen je i način koji računa samo jednu dekompoziciju, a daje usporedive rezultate s prethodno opisanim.

Prema [60], skraćeni HOSVD može umjesto (R_1, R_2, \dots, R_N) kao ulaz primati toleranciju $\epsilon > 0$ te odbaciti one singularne vektore čije su pripadajuće singularne vrijednosti manje od ϵ .

HOOI

Dekompozicija koju nalazi skraćeni HOSVD nije najbliža početnom tenzoru po normi udaljenosti, ali je dobra polazna točka za inicijalizaciju faktorskih matrica u nekom ALS iterativnom algoritmu. Jedan takav je *higher-order orthogonal iteration* ili HOOI, kojeg opisujemo u ovoj podsekciji.

Slično kao kod CP dekompozicije, kada tražimo aproksimativnu Tuckerovu dekompoziciju tenzora $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, želimo riješiti optimizacijski problem

$$\min_{\mathcal{G}, A^{(1)}, \dots, A^{(N)}} \left\| \mathcal{X} - \llbracket \mathcal{G}; A^{(1)}, A^{(2)}, \dots, A^{(N)} \rrbracket \right\|, \quad (2.19)$$

pri čemu je $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \dots \times R_N}$ i $A^{(n)} \in \mathbb{R}^{I_n \times R_n}$ je stupčano ortogonalna za sve $n = 1, 2, \dots, N$. Detaljnije o razradi ovog problema može se naći u [36] i referencama navedenim ondje, npr. [35]. Ciljna funkcija u (2.19) može se transformirati u

$$\|\mathcal{X}\|^2 - \left\| \mathcal{X} \times_1 A^{(1)\top} \times_2 A^{(2)\top} \dots \times_N A^{(N)\top} \right\|^2.$$

Kako je $\|\mathcal{X}\|^2$ konstanta, problem (2.19) se zbog gornjeg izraza može riješiti metodom alternirajućih kvadrata, tj. može se svesti na niz maksimizacijskih potproblema, od kojih je n -ti jednak

$$\max_{A^{(n)} \in \mathbb{R}^{I_n \times R_n}} \left\| \mathcal{X} \times_1 A^{(1)\top} \times_2 A^{(2)\top} \dots \times_N A^{(N)\top} \right\|, \quad (2.20)$$

pri čemu je $A^{(n)}$ stupčano ortogonalna. Ciljna funkcija u (2.20) u matričnom zapisu jednaka je

$$\left\| A^{(n)\top} W \right\|, \text{ uz } W = X_{(n)}(A^{(N)} \otimes \dots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \dots \otimes A^{(1)}).$$

Rješenje se može odrediti pomoću SVD-a, postavljanjem $A^{(n)}$ na prvih R_n vodećih lijevih singularnih vektora od W . Pseudokod HOOI-ja nalazi se u Algoritmu 3.

Algoritam 3: Metoda HOOI za računanje Tuckerove dekompozicije tenzora $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$.

procedura HOOI($\mathcal{X}, R_1, R_2, \dots, R_N$)
 inicijaliziraj $A^{(n)} \in \mathbb{R}^{I_n \times R_n}$ za $n = 1, \dots, N$ pomoću HOSVD-a
ponavljaj
 za $n = 1, \dots, N$ **čini**
 $\mathcal{Y} \leftarrow \mathcal{X} \times_1 A^{(1)\top} \dots \times_{n-1} A^{(n-1)\top} \times_{n+1} A^{(n+1)\top} \dots \times_N A^{(N)\top}$
 $A^{(n)} \leftarrow R_n$ vodećih lijevih singularnih vektora od $Y_{(n)}$
kraj
dok ne bude pogoršanja u ciljnoj funkciji ili je dostignut maksimalan broj iteracija;
 $\mathcal{G} \leftarrow \mathcal{X} \times_1 A^{(1)\top} \times_2 A^{(2)\top} \dots \times_N A^{(N)\top}$
vrti $\mathcal{G}, A^{(1)}, A^{(2)}, \dots, A^{(N)}$

Metoda HOOI konvergira prema rješenju za koje će se ciljna funkcija iz (2.19) prestati smanjivati, no ne konvergira nužno prema stacionarnoj točki, pa tako ni prema globalnom minimumu. Neke metode za ubrzanje HOOI-ja nalaze se u [5]. Također, u [22] je predstavljena metoda koja osigurava konvergenciju u stacionarnu točku ciljne funkcije kvadratične brzine uz, doduše, skuplje iteracije, a temelji se na Newtonovoj metodi na produktu Grassmannovih mnogostrukosti. Druga poboljšanja bazirana na sličnom principu mogu se naći u [30, 31, 62]. Još metoda može se naći u [57] i [24].

2.2.4 (Ne)jedinственost

Tuckerove dekompozicije općenito nisu jedinstvene. Promotrimo li izraz (2.15) iz definicije Tuckerove dekompozicije, možemo lako provjeriti da za proizvoljne regularne matrice $U \in \mathbb{R}^{P \times P}$, $V \in \mathbb{R}^{Q \times Q}$ i $W \in \mathbb{R}^{R \times R}$ vrijedi

$$\llbracket \mathcal{G}; A, B, C \rrbracket = \llbracket \mathcal{G} \times_1 U \times_2 V \times_3 W; AU^{-1}, BV^{-1}, CW^{-1} \rrbracket.$$

Dakle, dekompozicija se ne mijenja množenjem jezgrenog tenzora regularnim matricama sve dok se množe i pripadajuće faktorske matrice inverzima prvih matrica. Ovime je određenim načinima moguće pojednostaviti jezgreni tenzor tako da većina njegovih elemenata bude jednako nuli, čime se eliminiraju interakcije među odgovarajućim komponentama. Iako je superdijagonalizaciju nemoguće postići, često se nedijagonalne elemente jezgrenog tenzora nastoji jako umanjiti, pa Tuckerova dekompozicija počinje nalikovati na CP dekompoziciju za koju znamo da je esencijalno jedinstvena. Više o problemu nejedinstvenosti Tuckerove dekompozicije može se pronaći u [36] i referencama navedenim ondje.

2.2.5 Općenita primjena

Tuckerova dekompozicija se, kao i CP dekompozicija, prvo koristila u kemometriji i psihometriji, a nakon toga u obradi signala.

Na području računalnog vida prvi put se koristila u obliku metodologije TensorFaces, u kojoj su se fotografije različitih ljudi uzete u različitim uvjetima (osvijetljenje, kut kamere, ekspresija), što uz piksele predstavlja modove, pohranile u tenzor koji se zatim dekomponirao HOSVD-om, a nakon toga se vršilo prepoznavanje novih fotografija. To prepoznavanje se pokazalo značajno preciznijim od dotadašnjeg pristupa pomoću PCA-a. TensorFaces je korisno i za kompresiju jer može zanemariti nebitne efekte poput osvijetljenja, a sačuvati važne značajke lica. Još istraživanja koja se tiču ljudske kretnje, modeliranja ekspresija na licu, komprimiranja slika, stavljanja vodenog žiga na videe i ostalog može se pronaći u [36].

U [57] dan je pregled upotrebe koji uključuje analizu podataka iz soba za chat (korisnik, ključna riječ, vrijeme) radi generiranja umjetne sobe za chat, modeliranje znanstvenih radova (autor, ključna riječ, konferencija) za identificiranje grupa autora koje vežu slične teme i konferencije, proučavanje ponašanja korisnika Twittera tijekom vremena (izvorni korisnik, označeni korisnik, ključna riječ), personaliziranje pretraživanja na internetu pomoću povijesti pretraživanja (korisnik, upit u tražilici, kliknuti link), dizajniranje sustava za predlaganje "tagova", predviđanje kriminalne aktivnosti na nekoj lokaciji i drugo.

Posljednjih godina Tuckerova dekompozicija se koristi kao alat u radu s konvolucijskim neuronskim mrežama, posebice za smanjivanje dimenzionalnosti [33, 78, 27, 69, 11, 43].

2.2.6 Primjena u prepoznavanju radnje

TensorPose i skriveni Markovljevi modeli iz [59]

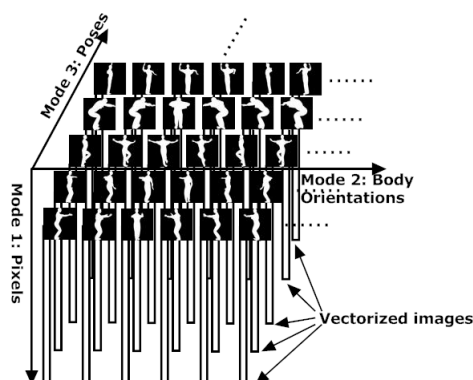
U ovom radu razvilo se okruženje TensorPose slično spomenutom TensorFaceu, u kojem se slike silueta u različitim orijentacijama i pozama vektoriziraju i potom slažu u tenzor tako da prvi mod označava piksele, drugi orijentaciju, a treći pozu, kako je to prikazano na Slici 2.4. Takvom tenzoru $\mathcal{A} \in \mathbb{R}^{N_i \times N_o \times N_p}$ se računa Tucker2 dekompozicija s nereduciranim modovima, tj. $\mathcal{D} \times_1 U_i \times_2 U_o \times_3 U_p$, pri čemu je $\mathcal{D} \in \mathbb{R}^{N_i \times N_o \times N_p}$, a faktorske matrice su kvadratne i ortogonalne, i U_i je jedinična, pa je dekompozicija zapravo $\mathcal{D} \times_2 U_o \times_3 U_p$.

Daljnja analiza oslanja se na to da vrijedi

$$\mathcal{A}(:, i, j) = \mathcal{D} \times_2 u_{o,i} \times_3 u_{p,j}, \quad (2.21)$$

gdje se $\mathcal{A}(:, i, j)$ odnosi na vektoriziranu sliku u orijentaciji i i pozici j , a $u_{o,i}$ i $u_{p,j}$ su, redom, i -ti i j -ti redak od U_o i U_p , tj. vektori koji reprezentiraju i -tu orijentaciju i j -tu pozu. Nakon dobivanja Tuckerove dekompozicije, za neku novu sliku, tj. vektor z , želi se saznati u kojoj je pozici snimljena, pa se računa

$$z = \mathcal{D} \times_2 v_o \times_3 v_p, \quad (2.22)$$



Slika 2.4: Struktura TensorPose iz [59].

gdje je v_o vektor orijentacije, a v_p vektor poze. Ti vektori se dobivaju metodom alternirajućih najmanjih kvadrata tako što se, počevši od inicijalnih vektora orijentacije $v_o^{(0)}$, prvo računa

$$C_o^{(n)} = \mathcal{D} \times_2 v_o^{(n)}, \quad (2.23)$$

odakle se vektor poze $v_p^{(n)}$ dobiva kao rješenje linearne jednadžbe

$$z = C_o^{(n)} v_p^{(n)}. \quad (2.24)$$

Potom se dobiveni vektor koristi kao početna točka u računanjima

$$C_p^{(n)} = \mathcal{D} \times_3 v_p^{(n)} \quad \text{i} \quad z = C_p^{(n)} v_o^{(n+1)}, \quad (2.25)$$

a čitav postupak se ponavlja sve do konvergencije.

Radi uštede u broju ponavljanja ALS-a, autori su za kandidate početnog vektora $v_o^{(0)}$ uzeli u obzir sve retke $\{u_{o,i}\}_{i=1}^{N_o}$ matrice U_o , dobivajući u (2.24) vektore $\{v_{p,i}\}_{i=1}^{N_o}$, među kojima bi za vektor $v_p^{(0)}$ odabrali onaj koji je najbliži nekom od redaka $\{u_{p,i}\}_{i=1}^{N_p}$ matrice U_p . Drugim riječima

$$v_p^{(0)} = \operatorname{argmax} \frac{v_{p,i} \cdot u_{p,j}}{\|v_{p,i}\| \|u_{p,j}\|}, \quad \text{za} \quad i = 1, \dots, N_o, \quad j = 1, \dots, N_p. \quad (2.26)$$

Nakon konvergencije, svakom vektoru z se pridružuje dobivena projekcija na prostor poza, v_p , koja je neovisna o orijentaciji. Tako se kolekciji vektora koji predočavaju tijekom jedne radnje ili geste pridružuje niz poza, a one se potom koriste kao vektori značajki u skrivenom Markovljevom modelu, koji prepoznaje o kojoj je radnji riječ. Model je tipa lijevo-desno, sastoji se od 12 stanja i koristi algoritam maksimizacije očekivanja, a više o njemu može se pročitati u radu.

Dataset se sastojao od slika šest gesta: odgurivanja, širenja desne ruke, sklapanja ruku i čučanja, kaktus poze, povlačenja desne ruke i podizanja lijeve ruke. Gestama je pridruženo

28 različitih poza, a svaka poza je izvedena u 16 orijentacija. Točnost prepoznavanja prosječno po svim gestama je iznosila 99.48%.

Tangencijalni svežnji na specifičnim mnogostrukostima iz [47]

Ovdje se proučavaju složeniji matematički objekti koji se zovu mnogostrukosti (specifične ortogonalne grupe te Stiefelove i Grassmannove mnogostrukosti), te njihova uloga u prepoznavanju radnje. Faktorske matrice koje nastaju Tuckerovom dekompozicijom tenzora trećeg reda, tj. videa koji prikazuje neku radnju, predstavljaju točke na mnogostrukostima. Svakoj takvoj točki pridružen je tangencijalni prostor na kojem se mogu primijeniti tzv. dijagrami mnogostrukosti. Svi tangencijalni prostori zajedno čine svežanj na kojemu se računa određena unutrašnja udaljenost i čime se klasificiraju videi. Ovaj algoritam ne zahtjeva prethodno treniranje.

Testiranje se provodilo na datasetovima Cambridge gesture, UMD Keck body gesture i UCF sport. Cambridge gesture se sastoji od lokaliziranih videa s uniformnim pozadinama, i na njemu je ostvareno 91% točnosti. Videi iz UMD Keck body gesture s radnjama u statičnim i dinamičnim scenama klasificirani su uz točnosti 96% i 92%, redom. Na UCF sport datasetu, koji se sastoji od videa preuzetih s televizija, dakle s dinamičnim okruženjem i neujednačenom pozadinom, ostvareno je 91% točnosti. Navedeno pokazuje kako algoritam vrlo dobro nastupa u kontroliranim, ali i nekontroliranim uvjetima.

3D skeletoni iz [44], [37], i [9]

Osim prepoznavanja radnji iz uobičajenih monokularnih RGB videa, koji zbog raznih ometajućih faktora poput pozadinskog šuma i promjena u osvjetljenju mogu narušiti uspješnost prepoznavanja, posljednjih godina popularan je prikaz ljudske kretnje kao prostorno-vremenske evolucije niza skeletona, gdje je skeleton sustav krutih tijela povezanih zglobovima čije se pozicije dobivaju od dubinskih senzora. U navedenim radovima se uz tenzorsku dekompoziciju koriste razne druge tehnike za prepoznavanje radnje, kao što su linerani dinamički sustavi, kerneli, diskriminantna korelacijska analiza te neuronske mreže. U [44] iz 2019. na datasetovima MSR-Action3D, UTKinect-Action i G3D-Gaming postignuti su rezultati bolji od dotadašnjih najboljih.

Gornje navedeni radovi nisu jedini koji koriste Tuckerovu dekompoziciju za prepoznavanje radnje, ali su među onima koji su privukli najviše pažnje prema broju citiranja i koji su postigli jako dobre rezultate. Čitavo treće poglavlje posvećeno je implementaciji još jednog algoritma koji koristi *ridge* regresiju.

2.3 Vlačna dekompozicija

U praksi su česti tenzori reda znatno većeg od tri, što kod tenzorskih dekompozicija poput Tuckerove stvara problem prokletstva dimenzionalnosti (*curse of dimensionality*) jer broj varijabli koje se trebaju izračunati raste eksponencijalno s redom tenzora [57]. Pretpostavimo li, primjerice, da je dan tenzor \mathcal{X} reda N sa svim modovima jednakim nekom $r > 1$, za pronalazak njegove Tuckerove dekompozicija potrebno je izračunati r^N elemenata jezgrog tenzora što je za velike N , primjerice $N = 100$, velik broj, čak i kada je r mali. Ova i sljedeća dekompozicije dobro aproksimiraju tenzore reda i puno većeg od tri izbjegavajući problem prokletstva dimenzionalnosti.

Tensor-train ili vlačna tenzorska dekompozicija osmišljena je 2011. u radu [53]. U njoj se tenzor rastavlja na matricu, tj. "lokomotivu", na koju se veže niz jezgrenih tenzora reda tri i još jedna matrica, koji predstavljaju "vagone", pri čemu je svaki jezgreni tenzor sa susjednim povezan preko jednog reduciranog moda. Formalnije, primjerice, dekompozicija tenzora \mathcal{X} reda četiri može se zapisati kao

$$\mathcal{X}(i, j, k, l) \approx \sum_{r_1, r_2, r_3} G_1(i, r_1) \mathcal{G}_2(r_1, j, r_2) \mathcal{G}_3(r_2, k, r_3) G_4(r_3, l), \quad (2.27)$$

a njena ilustracija je na Slici 2.5. U slučaju tenzora općenitog reda N , dekompozicija je

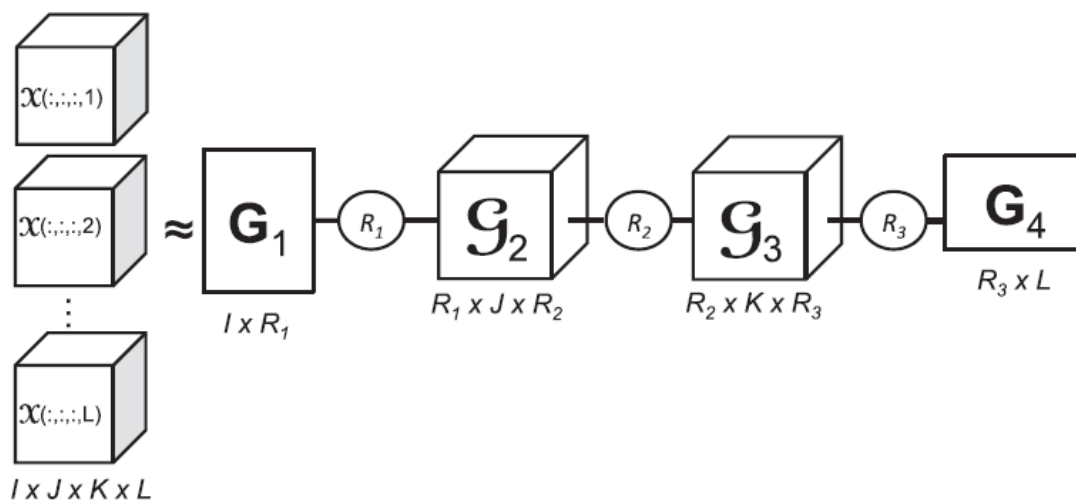
$$\mathcal{X}(i_1, \dots, i_N) \approx \sum_{r_1, \dots, r_{N-1}} G_1(i_1, r_1) \mathcal{G}_2(r_1, i_2, r_2) \cdots \mathcal{G}_{N-1}(r_{N-2}, i_{N-1}, r_{N-1}) G_N(r_{N-1}, i_N). \quad (2.28)$$

Ako su svi originalni modovi tenzora jednaki I , a reducirani jednaki r , broj elemenata koje je potrebno izračunati je $O(NIr^2)$, čime je izbjegnuto prokletstvo dimenzionalnosti.

Pseudokod algoritma za rješavanje dekompozicije ponovo se može naći u [57]. Neka od područja u kojima se primjenjivala vlačna dekompozicija su kvantifikacija neizvjesnosti [79, 13], klasifikacija videa [73], oporavak slika i videa [12] te valne transformacije [53].

2.4 Prstenasta dekompozicija

U radu [80] iz 2016. predstavljena je prstenasta tenzorska dekompozicija (*tensor ring decomposition*), još jedna dekompozicija koja početni tenzor prikazuje nizom povezanih tenzora manjih dimenzija. I njome je izbjegnut problem prokletstva dimenzionalnosti uz jednaku vremensku složenost kao kod vlačne dekompozicije, no za razliku od strogo sekvencijalnog množenja jezgrenih tenzora zbog kojeg je otežan pronalazak optimalne vlačne dekompozicije, prstenasta dekompozicija dopušta ciklične permutacije poretka množenja. Ako imamo tenzor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$, njegova prstenasta dekompozicija je dana nizom tenzora $\mathcal{Z}_k \in \mathbb{R}^{r_k \times n_k \times r_{k+1}}$ reda tri za $k = 1, 2, \dots, d$, pri čemu se $r = (r_1, r_2, \dots, r_d)$ smatra



Slika 2.5: Vlačna dekompozicija tenzora reda četiri iz [57].

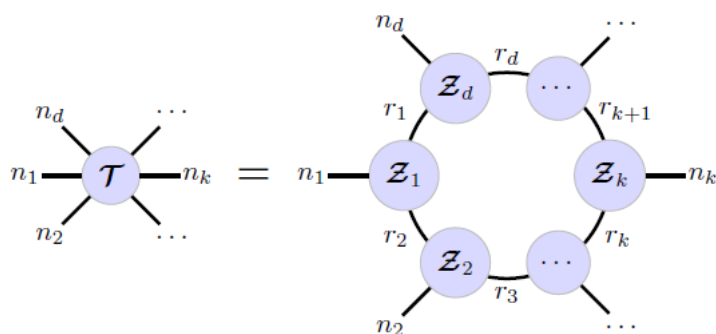
rangom dekompozicije. Zapis po elementima glasi

$$\mathcal{T}(i_1, i_2, \dots, i_d) = \text{tr} \left(Z_1(i_1) Z_2(i_2) \cdots Z_d(i_d) \right) = \text{tr} \left(\prod_{k=1}^d Z_k(i_k) \right), \quad (2.29)$$

pri čemu $Z_k(i_k) \in \mathbb{R}^{r_k \times r_{k+1}}$ označava i_k -ti lateralni odsječak tenzora \mathcal{Z}_k . Svaka dva susjedna tenzora \mathcal{Z}_k i \mathcal{Z}_{k+1} dijele dimenziju r_{k+1} odgovarajućih modova, a za posljednji tenzor vrijedi $r_{d+1} = r_1$, čime je osigurano da je matrica čiji se trag računa kvadratna. Primijetimo kako su kod vlačne dekompozicije prvi i zadnji tenzori zapravo matrice, tj. vrijedi $r_1 = r_{d+1} = 1$, što ograničava sposobnost dobre reprezentacije podataka. Spomenute ciklične permutacije dopuštene su zbog invarijantnosti traga matrice na njima. Ilustracija dekompozicije nalazi se na Slici 2.6.

U [80] predložena su četiri algoritma za nalaženje prstenaste dekompozicije. Algoritam naziva TR-SVD bazira se na računanju d SVD-ova, efikasan je zbog nerekurzivnosti i proizvoljno blizu aproksimira bilo koji tenzor, no može davati drugačiju reprezentaciju ovisno o izboru početnog moda zbog čega ne dovodi uvijek do optimalnog ranga. Ostala tri algoritma bazirana su na ALS principu, računaju rang koji ne ovisi o izboru početnog moda, no zbog rekurzivnosti sporo konvergiraju. Dodatno, stabilni su, dva od njih mogu prilagođavati rang tijekom optimizacije, pri čemu je jedan od ta dva učinkovit bez velikih zahtjeva na vremensku složenost.

Prstenasta dekompozicija je u [80] testirana na nekim sintetičkim podacima, gdje je pokazano da bolje komprimira i reprezentira podatke od vlačne dekompozicije, te na datasetovima COIL-100 i KTH, demonstriranjem učinkovitosti u izdvajanju značajki, klasifikaciji i nenadziranom učenju. U [70] predstavljen je algoritam za rad s nedostajućim

Slika 2.6: Prstenasta dekompozicija tenzora $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ iz [80].

podacima temeljen na prstenastoj dekompoziciji, i pokazana je njegova prednost na nekoliko primjera u odnosu na algoritam koji koristi vlačnu dekompoziciju. Za komprimiranje konvolucijskih neuronskih mreža prstenasta dekompozicija se upotrebljavala u [71, 54].

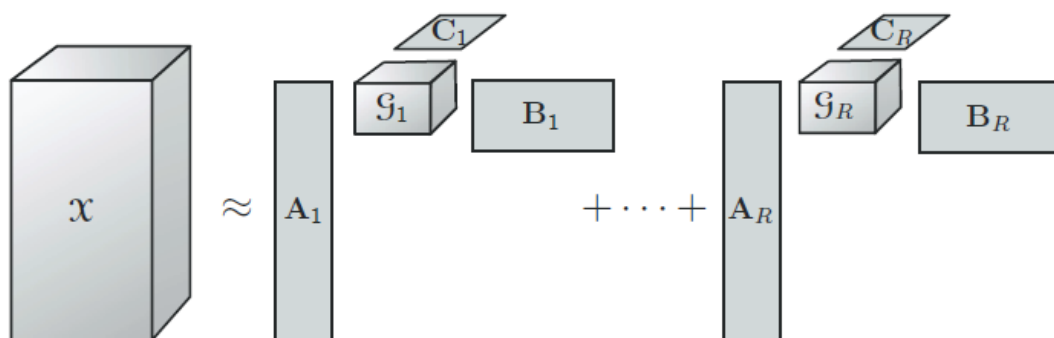
2.5 Blokovska dekompozicija

Blokovska (*block-term*) dekompozicija ujedinjuje CP, koja rastavlja tenzor na sumu tenzora ranga jedan, i Tuckerovu dekompoziciju rastavljanjem tenzora na sumu Tuckerovih dekompozicija. Tenzor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ se tako zapisuje kao

$$\mathcal{X} \approx \sum_{r=1}^R \llbracket \mathcal{G}_r; A_r, B_r, C_r \rrbracket,$$

pri čemu su $\mathcal{G}_r \in \mathbb{R}^{M_r \times N_r \times P_r}$, $A_r \in \mathbb{R}^{I \times M_r}$, $B_r \in \mathbb{R}^{J \times N_r}$ i $C_r \in \mathbb{R}^{K \times P_r}$ za $r = 1, \dots, R$. Ilustracija je na Slici 2.7.

Blokovska dekompozicija se primjenjivala u medicinskim istraživanjima, preciznije za analizu EEG podataka za modeliranje epileptičkih napadaja ([29]) i analizu fibrilacije atrijske ([61, 75]). Kod prepoznavanja ljudske radnje na videima, koristila se u [74] za rješavanje problema redundancije rekurentnih neuronskih mreža. Stvoren je model s manje parametara i sa snažnijim opisom korelacije dimenzija značajki, što je dovelo do jednostavnijeg treniranja i, na datasetu UCF11 YouTube action, boljih rezultata od long short-term memory arhitekture i već spomenutog TT-LSTM-a.



Slika 2.7: Blokovska dekompozicija tenzora reda tri iz [36].

Poglavlje 3

Implementacija algoritma za prepoznavanje radnje

Za učinkovitije prepoznavanje ljudske radnje sa snimljenih scena, one se mogu provesti kroz opisnike značajki koji prepoznaju značajke otporne na promjene u veličini, šumu, osvjetljenju i geometrijskom iskrivljenju. Jedan takav opisnik je histogram orijentiranih gradijenata ili HOG, čiji se izlazni podatak može prikazati kao tenzor. HOG se koristi i u radu [77], u kojem se pomoću Tuckerove dekompozicije i *ridge* regresije kreira algoritam za prepoznavanje radnje, i čije ćemo rezultate ovdje pokušati ponoviti.

Prema [15], računanje HOG-a se sastoji od globalnog normaliziranja slike (radi smanjenja utjecaja iluminacije), računanja gradijenata (koji sadrže informaciju o konturi, silueti i teksturi), računanja gradijentnih histograma za svaku ćeliju te normaliziranja preko blokova ćelija. Histogrami se računaju dijeljenjem slike na manje ćelije i prebrojavanjem gradijentnih orijentacija preko svih piksela u ćeliji, čime se proizvodi enkodiranje lokalnih sadržaja slike otporno na manje promjene u pozici i izgledu. Blokovi su kolekcije ćelija, a njihova normalizacija služi za bolju invarijantnost na iluminaciju i osjenčanost. Detaljnije o dobivanju histograma može se pronaći u [48]. Nakon normalizacije, dobiveni histogrami se mogu, kao u implementaciji iz [1] koju ćemo koristiti, organizirati u tenzor s dimenzijama broj redova blokova, broj stupaca blokova, broj redova ćelija u bloku, broj stupaca ćelija u bloku te broj orijentacija. U [77] to je učinjeno malo drugačije, no mi ćemo slijediti opisani način.

Poznati algoritmi poput k najbližih susjeda, metode potpornog vektora i *ridge* regresije, koji se mogu iskoristiti za klasifikaciju slika, kao ulazni podatak očekuju vektor. Stoga bi za takav algoritam tenzor nastao HOG-om bilo potrebno vektorizirati, što, prema [77], ne mora dovesti do loših rezultata, ali može proizvesti vektore jako velikih dimenzija, što pak može dovesti do problema s overfitanjem i velikim memorijskim zahtjevima. K tome se još gubi i prostorna informacija o slici. U [77] su navedeni i algoritmi prilagođeni radu s ten-

zorima, no svi imaju određene manjkavosti. Primjerice, jedan dekompozira ulazni tenzor u tenzor ranga jedan, čime se gubi diskriminativna informacija zbog opisa svakog moda sa samo jednim vektorom, dok jedan koristi CP dekompoziciju, koja je ograničavajuća u odnosu na Tuckerovu dekompoziciju sjetimo li se zahtjeva za jednakim brojem faktorskih vektora po svim modovima.

3.1 Algoritam Tuckerove *ridge* regularizacije

Objasnimo ukratko algoritam iz rada [77], nazvan Tuckerovom *ridge* regularizacijom ili TuRR. On pretpostavlja da su HOG opisnik i parametri koje ćemo tražiti u tenzorskom obliku, tj. da su tenzori trećeg reda. Zbog prednosti Tuckerove nad CP dekompozicijom, tenzor parametara se, po jedan za svaku klasu ljudske radnje, dekomponira Tuckerovom dekompozicijom u jezgri tenzor i tri faktorske matrice. U svakom krugu alternirajućeg algoritma jezgri tenzor se matricizira u pojedinom modu te se faktorska matrica koja tom modu pripada računa *ridge* regresijom uz fiksiranje ostalih faktorskih i jezgrene matrice. Nakon što se time pronađu sve faktorske matrice, na isti princip se traži jezgri tenzor fiksiranjem svih faktorskih matrica. Ovaj postupak se ponavlja sve do konvergencije.

Uz odgovarajuću regularizaciju, funkcija gubitka najmanjih kvadrata daje usporedive rezultate s drugim funkcijama gubitka. Uz odabir l_2 norme za regularizaciju, tipična *ridge* regresija je dana s

$$\min_{w,b} \sum_{i=1}^n (\langle x_i, w \rangle - y_i + b)^2 + \lambda \|w\|_2^2,$$

gdje je n broj slika za treniranje, x_i vektorizirani opisnik slike, w vektorski parametar, b pristranost, $y = [y_1, \dots, y_n]$ vektor oznaka pripadnosti slika nekoj klasi i λ parametar regularizacije. Kako bismo prilagodili *ridge* regresiju tenzorskim podacima, označimo prvo s $\mathcal{X}_i \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_M}$ HOG opisnik i -te slike za treniranje, gdje $i = 1, \dots, n$, a M je red tenzora, tj. 3. Promatramo li klasu ljudske radnje r , oznaka y_i će biti jednaka 1 ukoliko se \mathcal{X}_i odnosi na klasu r , tj. -1 u suprotnom. Prilagođena *ridge* regresija sada je jednaka

$$\min_{\mathcal{W}, b} \sum_{i=1}^n (\langle \mathcal{X}_i, \mathcal{W} \rangle - y_i + b)^2 + \lambda \|\mathcal{W}\|_F^2, \quad (3.1)$$

pri čemu je $\mathcal{W} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_n}$ tenzorski parametar. Zamijenimo li \mathcal{W} njegovom Tuckerovom dekompozicijom, (3.1) postaje

$$\min_{\mathcal{G}, U_1, \dots, U_M} \sum_{i=1}^n (\langle \mathcal{X}_i, \llbracket \mathcal{G}; U_1, U_2, \dots, U_M \rrbracket \rangle - y_i + b)^2 + \lambda \|\llbracket \mathcal{G}; U_1, U_2, \dots, U_M \rrbracket\|_F^2. \quad (3.2)$$

Funkcija gubitka u (3.2) nije zajednički konveksna (*jointly convex*) za sve novodobivene parametre $\mathcal{G}, U_1, \dots, U_M$, ali može se svesti na alternirajući optimizacijski algoritam

POGLAVLJE 3. IMPLEMENTACIJA ALGORITMA ZA PREPOZNAVANJE RADNIKA

koji pri svakoj iteraciji rješava konveksni optimizacijski problem za po jedan parametar fiksirajući ostale parametre.

Za početak, (3.1) se u matriciziranom obliku u modu k može zapisati kao

$$\min_{W_{(k)}, b} \sum_{i=1}^n (\text{tr}(W_{(k)} X_{(k)}^{i\top}) - y_i + b)^2 + \lambda \|W_{(k)}\|_F^2. \quad (3.3)$$

Označimo li $\tilde{U}_k = U_M \otimes \cdots \otimes U_{k+1} \otimes U_{k-1} \otimes \cdots \otimes U_1$, prema Propoziciji 1.3.13 je $W_{(k)} = U_k G_{(k)} \tilde{U}_k^\top$. Supstituiramo li taj izraz u (3.3) uz fiksirane U_l ($l=1, l \neq k$) i $G_{(k)}$, U_k tražimo kao rješenje od

$$\min_{U_k, b} \sum_{i=1}^n (\text{tr}(U_k G_{(k)} \tilde{U}_k^\top X_{(k)}^{i\top}) - y_i + b)^2 + \lambda \text{tr}(U_k G_{(k)} \tilde{U}_k^\top \tilde{U}_k G_{(k)}^\top U_k^\top). \quad (3.4)$$

Označimo li nadalje $\tilde{X}_k^{i\top} = G_{(k)} \tilde{U}_k^\top X_{(k)}^{i\top}$ i $D_k = G_{(k)} \tilde{U}_k^\top \tilde{U}_k G_{(k)}^\top$, (3.4) se može zapisati kao

$$\min_{U_k, b} \sum_{i=1}^n (\text{tr}(U_k \tilde{X}_k^{i\top}) - y_i + b)^2 + \lambda \text{tr}(U_k D_k U_k^\top), \quad (3.5)$$

a uz $\text{tr}(U_k \tilde{X}_k^{i\top}) + b = [\text{vec}(U_k)^\top b][\text{vec}(\tilde{X}_k^i)^\top \mathbf{1}]^\top = v_k^\top \hat{x}_k^i$ i $\tilde{D}_k = \begin{bmatrix} D_k \otimes I_{d_k \times d_k} & 0 \\ 0 & 1 \end{bmatrix}$, (3.5) je

$$\min_{v_k} \sum_{i=1}^n (v_k^\top \hat{x}_k^i - y_i)^2 + \lambda \text{tr}(v_k^\top \tilde{D}_k v_k). \quad (3.6)$$

Postavimo li $\hat{X}_k = [\hat{x}_k^1, \hat{x}_k^2, \dots, \hat{x}_k^n]$, iz (3.6) konačno dobivamo

$$\min_{v_k} \text{tr}(v_k^\top (\hat{X}_k \hat{X}_k^\top + \lambda \tilde{D}_k) v_k) - 2 \text{tr}(v_k^\top \hat{X}_k y^\top) + \text{tr}(y^\top y). \quad (3.7)$$

Izjednačimo li derivaciju funkcije gubitka iz (3.7) po v_k s nulom, dobivamo

$$\begin{aligned} 2(\hat{X}_k \hat{X}_k^\top + \lambda \tilde{D}_k) v_k - 2 \hat{X}_k y^\top &= 0 \\ \implies v_k &= (\hat{X}_k \hat{X}_k^\top + \lambda \tilde{D}_k)^{-1} \hat{X}_k y^\top. \end{aligned} \quad (3.8)$$

Nakon što ovim postupkom dobijemo sve matrice U_1, U_2, \dots, U_M , računamo tenzor \mathcal{G} , odnosno njegovu matricizaciju u nekom modu, primjerice modu 1. Uz oznaku $U_\otimes = U_M \otimes \cdots \otimes U_1$ vrijedi $\text{vec}(W_{(1)}) = U_\otimes \text{vec}(G_1)$. Odatle se iz (3.3) dobiva

$$\min_{G_{(1)}, b} \sum_{i=1}^n (\text{vec}(G_{(1)})^\top U_\otimes^\top \text{vec}(X_{(k)}^i)^\top - y_i + b)^2 + \lambda \text{vec}(G_{(1)})^\top U_\otimes^\top U_\otimes \text{vec}(G_{(1)}). \quad (3.9)$$

POGLAVLJE 3. IMPLEMENTACIJA ALGORITMA ZA PREPOZNAVANJE RADNIKA

Označimo li $\bar{x}_k^i = [(U_\otimes^\top \text{vec}(X_{(k)}^i))^\top \ 1]$, $g_1 = [\text{vec}(G_1)^\top \ b]$, $D = \begin{bmatrix} U_\otimes^\top U_\otimes & 0 \\ 0 & 1 \end{bmatrix}$ i $\bar{X}_k = [\bar{x}_k^1, \bar{x}_k^2, \dots, \bar{x}_k^n]$, (3.9) se svodi na problem *ridge* regresije za g_1 , tj. na

$$\min_{g_1, b} \text{tr}(g_1^\top (\bar{X}_k \bar{X}_k^\top + \lambda D) g_1) - 2 \text{tr}(g_1^\top \bar{X}_k y^\top) + \text{tr}(y^\top y). \quad (3.10)$$

Izjednačujući derivaciju funkcije u (3.10) s 0, dobivamo

$$\begin{aligned} 2(\bar{X}_k \bar{X}_k^\top + \lambda D) g_1 - 2\bar{X}_k y^\top &= 0 \\ \implies g_1 &= (\bar{X}_k \bar{X}_k^\top + \lambda D)^{-1} \bar{X}_k y^\top. \end{aligned} \quad (3.11)$$

Optimiziranje dekompozicije $[\mathcal{G}; U_1, U_2, \dots, U_M]$ se iterira sve do konvergencije, tj. do dobivanja globalnih rješenja, što je zagarantirano jer smo originalni nekonveksni problem sveli na dva konveksna (3.7) i (3.10) koji imaju globalna rješenja. Koraci su sažeti u Algoritmu 4. Parametri $\mathcal{G}, U_1, \dots, U_M$ se računaju za svaku od c klasa, a prepoznavanje labela testnih slika nalazi se u Algoritmu 5.

Algoritam 4: Tuckerova *ridge* regularizacija

Ulaz: Ulazni tenzori $\mathcal{X}_i \in \mathbb{R}^{d_1 \times \dots \times d_M}$ i labela Y_i , za $i = 1, \dots, n$, dimenzije jezgrenog tenzora R_1, R_2, \dots, R_M i regularizacijski parametar λ .

Izlaz: Parametri $\{U_1^r, U_2^r, \dots, U_M^r, \mathcal{G}^r, b^r\}_{r=1}^c$

za $r = 1, \dots, c$ **čini**

za $i = 1, \dots, n$ **čini**

ako $Y_i = r$ **onda** $y_i = 1$ **inače** $y_i = -1$;

kraj

 Postavi $t = 0$ i inicijaliziraj $\mathcal{G}, U_1, \dots, U_M$ na slučajan način

ponavljaj

$t = t + 1$

za $k = 1, \dots, M$ **čini**

 Izračunaj v_k prema (3.8)

kraj

 Izračunaj g_1 prema (3.11)

dok ne bude konvergiralo;

kraj

vrați skup parametara $\{U_1^r, U_2^r, \dots, U_M^r, \mathcal{G}^r, b^r\}_{r=1}^c$

Algoritam 5: Proces prepoznavanja

Ulaz: Testni tenzori $\mathcal{X}_i \in \mathbb{R}^{d_1 \times \dots \times d_M}$, $i = 1, \dots, n$, i parametri $\{U_1^r, U_2^r, \dots, U_M^r, \mathcal{G}^r, b^r\}_{r=1}^c$.

Izlaz: Predviđene labele y testnih tenzora.

za $r = 1, \dots, c$ **čini**

 Izračunaj tenzorski parametar \mathcal{W}^r klase r pomoću

$$\mathcal{W}^r = \mathcal{G}^r \times_1 U_1^r \times_2 U_2^r \cdots \times_M U_M^r$$

kraj

za $i = 1, \dots, n$ **čini**

 Izračunaj labelu od \mathcal{X}_i pomoću $y_i = \arg \max_r (\langle \mathcal{X}_i, \mathcal{W}^r \rangle + b^r)_{r=1}^c$

kraj

vрати predviđene labele y

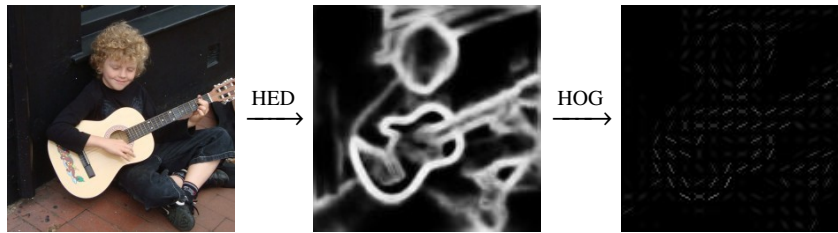
3.2 Eksperiment

U [77] TuRR algoritam se uspoređivao s nekoliko tenzorskih algoritama koji koriste Tuckerovu ili CP dekompoziciju i s jednim vektorskim, a svi se baziraju se na metodi potpornih vektora ili na regresiji. Prosječna točnost po svim klasama nekog dataseta je uzeta kao evaluacijska metrika.

Sve slike su smanjene na veličinu 128×128 , a umjesto Pb operatora koji se koristio u [77] za prepoznavanje granica objekata na slici koristili smo alat Holistically-Nested Edge Detection (HED) iz [72] temeljen na dubokom učenju za kojeg je na nekoliko datasetova pokazano da daje bolje rezultate. Također, u [77] kod računanja HOG-a nije jasna podjela na ćelije, blokovi veličine 16×16 piksela su nepreklapajući i korištena su četiri različita tipa normalizacije (vjerojatno L2-norm, L2-Hys, L1-norm i L1-sqrt). S četiri orijentacije, ondje se dobivaju tenzori dimenzija $16 \times 16 \times 16$. Kako je već rečeno, u ovom radu koristimo implementaciju iz [1] zbog jednostavnosti korištenja. Uz veličine ćelija 8×8 piksela, veličine preklapajućih blokova 2×2 ćelija s pomakom od jedne ćelije i 4 orijentacije, dobivamo tenzore dimenzija $15 \times 15 \times 2 \times 2 \times 4$, koje preoblikujemo u tenzore dimenzija $15 \times 15 \times 16$. Preobrazba slike kroz HED i HOG opisnik može se vidjeti na Slici 3.1.

Datasetovi koje smo koristili su Sport action, People playing musical instruments (PPMI) i Still DB, a slike u tim datasetovima su iz raznih izvora pa se razlikuju po rezoluciji, osvjetljenosti, pozadinskom šumu i sl. Dataset Sport action sastoji se od šest klasa koji opisuju pokrete u sportovima: teniski servis i forhend, udarac u odbojci, obrambeni udarac i bacanje lopte u kriketu te udarac u kriketu. Ukupno je 180 slika za treniranje, po 30 u svakoj klasi, i 120 za testiranje, po 20 u svakoj klasi. Dataset PPMI prikazuje ljude koji sviraju sedam različitih instrumenata: fagot, erhu, flautu, francuski rog, gitaru, saksofon i violinu.

POGLAVLJE 3. IMPLEMENTACIJA ALGORITMA ZA PREPOZNAVANJE RADNJE



Slika 3.1: Rezultat djelovanja HED-a i HOG-a na sliku.

Ima 1400 slika, po 100 za svaku klasu za treniranje i 100 za testiranje. Still DB sadrži slike šest različitih kretnji ljudi: trčanja, hodanja, hvatanja i bacanja lopte, čučanja i udaranja lopte nogom. Ukupno ih je 467, broj slika u klasi varira od 54 (hvatanje) do 101 (trčanje), a za podjelu na slike za treniranje i testiranje poznat je samo omjer 60:40, zbog čega, kao u [77], tu podjelu ponavljamo 10 puta i bilježimo prosječan rezultat.

Algoritam je pisan u Pythonu u okruženju Spyder, za operacije s tenzorima koristi se sučelje Tensorly iz [38] i sve se pokretalo na laptopu s procesorom Intel(R) Core(TM) i3-3217U CPU na 1.80 GHz i 6 GB RAM-a.



Slika 3.2: Primjerci slika iz datasetova Sport Action, PPMI i Still DB.

3.3 Rezultati

Nakon isprobavanja vrijednosti 0.1, 1, 10, 100 i 1000 za regularizacijski parametar λ , neovisno o dimenzijama jezgrenog tenzora i broju iteracija uočavamo da se najbolji rezultati za sve datasetove postižu za $\lambda = 100$, kao u originalnom radu. Zato fiksiramo $\lambda = 100$ i ugađamo dimenzije jezgrenog tenzora R_1 , R_2 i R_3 (od 2 do 15) te broj iteracija (od 1 do 15).

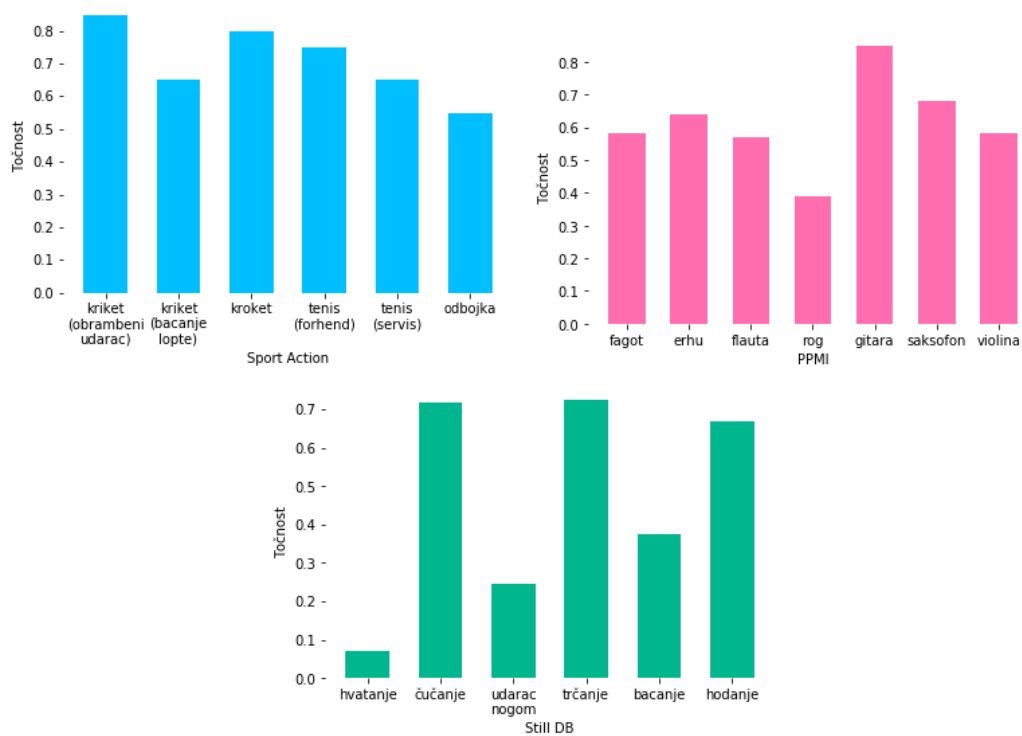
Najbolje dobivene točnosti prikazane su u Tablici 3.1 zajedno s originalnim rezultatima i rezultatima spomenutog vektorskog algoritma (SVR). Za dataset Sport Action je dobiveno bitno poboljšanje. Na druga dva dataseta dobiveni rezultati su podosta manji od originalnih rezultata, pa i od SVR-a, što bi se moglo pripisati problemu s parametrima HOG opisnika koji su se mogli bolje prilagoditi slikama i slučajnoj inicijalizaciji koja ipak bitno utječe na konačni rezultat. Na datasetu Still DB se općenito postižu lošiji rezultati u usporedbi s druga dva dataseta zbog, kako navode autori, većeg broja testnih slika nego kod Sport Action, manjeg broja slika za treniranje nego kod PPMI te zbog velike sličnosti između pojedinih klasa, npr. 'hvatanje' i 'bacanje'. Točnosti po klasama prikazane su na Slici 3.3.

Datasetovi	SVR	TuRR	TuRR iz [77]
Sport action	66.67	72.5	77.50
PPMI	64.57	61.29	71.57
Still DB	48.15	47.10	55.56

Tablica 3.1: Najbolje točnosti algoritama na datasetovima.

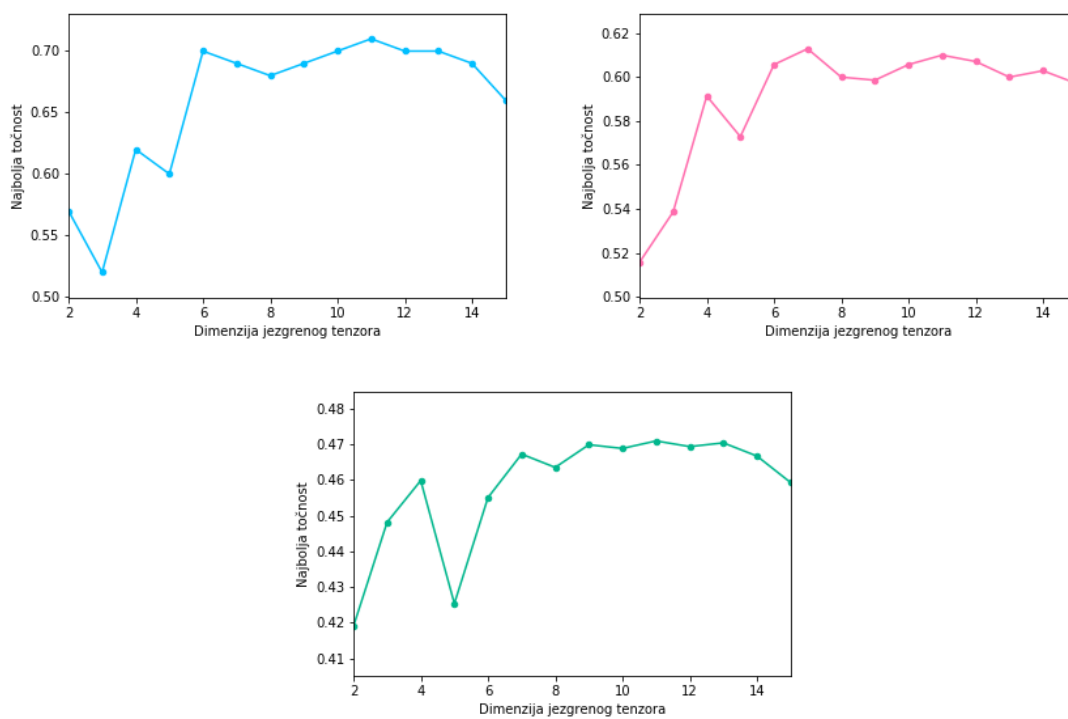
Na Slici 3.4 mogu se vidjeti najbolje dobivene točnosti po vrijednosti dimenzije jezgrenog tenzora za sve datasetove. Uočljivo je da se vrlo visoke, a za dataset PPMI i najveće točnosti, dobivaju već za dimenziju manju od 7. To govori kako se uz redukciju dimenzije originalnih tenzora za više od pola može i dalje ostvariti solidna točnost, tj. sačuvati bitna količina informacije iz slika.

POGLAVLJE 3. IMPLEMENTACIJA ALGORITMA ZA PREPOZNAVANJE RADNJE



Slika 3.3: Najbolje točnosti po klasama triju datasetova.

POGLAVLJE 3. IMPLEMENTACIJA ALGORITMA ZA PREPOZNAVANJE RADNJE



Slika 3.4: Najbolje tačnosti za pojedine dimenzije jezgrenog tenzora po datasetovima Sport Action, PPMI i Still DB.

Bibliografija

- [1] *Histogram of Oriented Gradients*, https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_hog.html, 2020-01-14.
- [2] E. Acar, C. Aykut-Bingol, H. Bingol, R. Bro i B. Yener, *Multiway analysis of epilepsy tensors*, *Bioinformatics* **23** (2007), br. 13, i10–i18.
- [3] A. H. Andersen i W. S. Rayens, *Structure-seeking multilinear methods for the analysis of fMRI data*, *NeuroImage* **22** (2004), br. 2, 728–739.
- [4] C. M. Andersen i R. Bro, *Practical aspects of PARAFAC modeling of fluorescence excitation-emission data*, *Journal of Chemometrics: A Journal of the Chemometrics Society* **17** (2003), br. 4, 200–215.
- [5] C. A. Andersson i R. Bro, *Improving the speed of multi-way algorithms:: Part I. Tucker3*, *Chemometrics and intelligent laboratory systems* **42** (1998), br. 1-2, 93–103.
- [6] C. J. Appellof i E. R. Davidson, *Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents*, *Analytical Chemistry* **53** (1981), br. 13, 2053–2056.
- [7] M. Araujo, S. Papadimitriou, St. Günnemann, C. Faloutsos, P. Basu, A. Swami, E. E. Papalexakis i D. Koutra, *Com2: fast automatic discovery of temporal ('comet') communities*, *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2014, str. 271–283.
- [8] B. W. Bader, M. W. Berry i M. Browne, *Discussion tracking in Enron email using PARAFAC*, *Survey of Text Mining II*, Springer, 2008, str. 147–163.
- [9] P. Barmpoutis, T. Stathaki i S. Camarinopoulos, *Skeleton-Based Human Action Recognition through Third-Order Tensor Representation and Spatio-Temporal Analysis*, *Inventions* **4** (2019), br. 1, 9.

- [10] C. Battaglino, G. Ballard i T. G. Kolda, *A practical randomized CP tensor decomposition*, SIAM Journal on Matrix Analysis and Applications **39** (2018), br. 2, 876–901.
- [11] H. Ben-Younes, R. Cadene, M. Cord i N. Thome, *Mutan: Multimodal tucker fusion for visual question answering*, Proceedings of the IEEE international conference on computer vision, 2017, str. 2612–2620.
- [12] J. A. Bengua, H. N. Phien, H. D. Tuan i M. N. Do, *Efficient tensor completion for color image and video recovery: Low-rank tensor train*, IEEE Transactions on Image Processing **26** (2017), br. 5, 2466–2479.
- [13] D. Bigoni, A. P. Engsig-Karup i Y. M. Marzouk, *Spectral tensor-train decomposition*, SIAM Journal on Scientific Computing **38** (2016), br. 4, A2405–A2439.
- [14] J. D. Carroll i J. J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of “Eckart-Young” decomposition*, Psychometrika **35** (1970), br. 3, 283–319.
- [15] Na. Dalal i B. Triggs, *Histograms of oriented gradients for human detection*, 2005.
- [16] I. Davidson, S. Gilpin, O. Carmichael i P. Walker, *Network discovery via constrained tensor analysis of fmri data*, Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2013, str. 194–202.
- [17] L. De Lathauwer i journal=Signal Processing volume=87 number=2 pages=322–336 year=2007 publisher=Elsevier Castaing, J., *Tensor-based techniques for the blind separation of DS-CDMA signals*.
- [18] L. De Lathauwer, B. De Moor i J. Vandewalle, *A multilinear singular value decomposition*, SIAM journal on Matrix Analysis and Applications **21** (2000), br. 4, 1253–1278.
- [19] M. De Vos, L. De Lathauwer, B. Vanrumste, S. Van Huffel i W. Van Paesschen, *Canonical decomposition of ictal scalp EEG and accurate source localisation: Principles and simulation study*, Computational intelligence and neuroscience **2007** (2007).
- [20] M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmmini i W. Van Paesschen, *Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone*, NeuroImage **37** (2007), br. 3, 844–854.
- [21] D. M. Dunlavy, T. G. Kolda i E. Acar, *Temporal link prediction using matrix and tensor factorizations*, ACM Transactions on Knowledge Discovery from Data (TKDD) **5** (2011), br. 2, 10.

- [22] L. Eldén i B. Savas, *A Newton–Grassmann Method for Computing the Best Multilinear Rank-(r_1, r_2, r_3) Approximation of a Tensor*, SIAM Journal on Matrix Analysis and Applications **31** (2009), br. 2, 248–271.
- [23] G. H. Golub i C. F. Van Loan, *Matrix Computations (3rd Ed.)*, Johns Hopkins University Press, Baltimore, MD, USA, 1996, ISBN 0-8018-5414-8.
- [24] L. Grasedyck, D. Kressner i C. Tobler, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitteilungen **36** (2013), br. 1, 53–78.
- [25] R. A. Harshman et al., *Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multimodal factor analysis*, (1970).
- [26] L. He, X. Kong, P. S. Yu, X. Yang, A. B. Ragin i Z. Hao, *Dusk: A dual structure-preserving kernel for supervised tensor learning with applications to neuroimages*, Proceedings of the 2014 SIAM International Conference on Data Mining, SIAM, 2014, str. 127–135.
- [27] G. Hu, Y. Hua, Y. Yuan, Z. Zhang, Z. Lu, S. S. Mukherjee, T. M. Hospedales, N. M. Robertson i Y. Yang, *Attribute-enhanced face recognition with neural tensor fusion networks*, Proceedings of the IEEE International Conference on Computer Vision, 2017, str. 3744–3753.
- [28] F. Huang, U.N. Niranjan, M. U. Hakeem i A. Anandkumar, *Fast detection of overlapping communities via online tensor methods*, arXiv preprint arXiv:1309.0787 **40** (2013), 43.
- [29] B. Hunyadi, D. Camps, L. Sorber, W. Van Paesschen, M. De Vos, S. Van Huffel i L. De Lathauwer, *Block term decomposition for modelling epileptic seizures*, EURASIP Journal on Advances in Signal Processing **2014** (2014), br. 1, 139.
- [30] M. Ishteva, P. A. Absil, S. Van Huffel i L. De Lathauwer, *Best low multilinear rank approximation of higher-order tensors, based on the Riemannian trust-region scheme*, SIAM Journal on Matrix Analysis and Applications **32** (2011), br. 1, 115–135.
- [31] M. Ishteva, L. De Lathauwer, P. A. Absil i S. Van Huffel, *Differential-geometric Newton method for the best rank-(R_1, R_2, R_3) approximation of tensors*, Numerical Algorithms **51** (2009), br. 2, 179–194.
- [32] A. Jukić, *Dekompozicije tenzora i primjena u izdvajanju značajki*, http://www.lair.irb.hr/ikopriva/Data/PhD_Students/ajukic/AJukic_Dekompozicije_tenzora.pdf, 2020-01-14.

- [33] Y. D. Kim, E. Park, S. Yoo, T. Choi, L. Yang i D. Shin, *Compression of deep convolutional neural networks for fast and low power mobile applications*, arXiv preprint arXiv:1511.06530 (2015).
- [34] T. G. Kolda, *Orthogonal tensor decompositions*, SIAM Journal on Matrix Analysis and Applications **23** (2001), br. 1, 243–255.
- [35] ———, *Multilinear operators for higher-order decompositions.*, Teh. izv., Sandia National Laboratories, 2006.
- [36] T. G. Kolda i Brett W. Bader, *Tensor Decompositions and Applications*, SIAM Rev. **51** (2009), br. 3, 455–500, ISSN 0036-1445, <http://dx.doi.org/10.1137/07070111X>.
- [37] P. Koniusz, A. Cherian i F. Porikli, *Tensor representations via kernel linearization for action recognition from 3d skeletons*, European Conference on Computer Vision, Springer, 2016, str. 37–53.
- [38] J. Kossaifi, Y. Panagakis i M. Pantic, *TensorLy: Tensor Learning in Python*, CoRR **abs/1610.09555** (2016), <http://arxiv.org/abs/1610.09555>.
- [39] I. Kotsia, W. Guo i I. Patras, *Higher rank support tensor machines for visual recognition*, Pattern Recognition **45** (2012), br. 12, 4192–4203.
- [40] B. Krausz i C. Bauckhage, *Action recognition in videos using nonnegative tensor factorization*, 2010 20th International Conference on Pattern Recognition, IEEE, 2010, str. 1763–1766.
- [41] J. B. Kruskal, *Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear algebra and its applications **18** (1977), br. 2, 95–138.
- [42] ———, *Multiway Data Analysis*, North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 1989, str. 7–18, ISBN 0-444-87410-0, <http://dl.acm.org/citation.cfm?id=120565.120567>.
- [43] D. Li, X. Wang i D. Kong, *Deeprebirth: Accelerating deep neural network execution on mobile devices*, Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
- [44] G. Li, K. Liu, W. Ding, F. Cheng i C. Ding, *Nonnegative Tensor-Based Linear Dynamical Systems for Recognizing Human Action from 3D Skeletons*, Mathematical Problems in Engineering **2019** (2019).

POGLAVLJE 3. IMPLEMENTACIJA ALGORITMA ZA PREPOZNAVANJE RADNJE

- [45] L. H. Lim, *Tensors and hypermatrices*, Handbook of Linear Algebra, 2nd ed., L. Hogben Ed., Cleveland, OH, USA, CRC Press (2013), 231–260.
- [46] L. H. Lim i C. Hillar, *Most tensor problems are np hard*, Preprint (2009).
- [47] Y. M. Lui, *Tangent bundles on special manifolds for action recognition*, IEEE Transactions on Circuits and Systems for Video Technology **22** (2011), br. 6, 930–942.
- [48] S. Mallick, *Histogram of Oriented Gradients*, <https://www.learnopencv.com/histogram-of-oriented-gradients/>, 2020-01-14.
- [49] E. Martinez-Montes, P. A. Valdés-Sosa, F. Miwakeichi, R. I. Goldman i M. S. Cohen, *Concurrent EEG/fMRI analysis by multiway partial least squares*, NeuroImage **22** (2004), br. 3, 1023–1034.
- [50] F. Miwakeichi, E. Martinez-Montes, P. A. Valdés-Sosa, N. Nishiyama, H. Mizuhara i Y. Yamaguchi, *Decomposing EEG data into space–time–frequency components using parallel factor analysis*, NeuroImage **22** (2004), br. 3, 1035–1045.
- [51] J. Mocks, *Topographic components model for event-related potentials and some biophysical considerations*, IEEE transactions on biomedical engineering **35** (1988), br. 6, 482–484.
- [52] M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas i S. M. Arnfred, *Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG*, NeuroImage **29** (2006), br. 3, 938–947.
- [53] I. V. Oseledets, *Tensor-train decomposition*, SIAM Journal on Scientific Computing **33** (2011), br. 5, 2295–2317.
- [54] Y. Pan, J. Xu, M. Wang, J. Ye, F. Wang, K. Bai i Z. Xu, *Compressing recurrent neural networks with tensor ring for action recognition*, Proceedings of the AAAI Conference on Artificial Intelligence, sv. 33, 2019, str. 4683–4690.
- [55] E. E. Papalexakis, L. Akoglu i D. Ience, *Do more views of a graph help? community detection and clustering in multi-graphs*, Proceedings of the 16th International Conference on Information Fusion, IEEE, 2013, str. 899–905.
- [56] E. E. Papalexakis, C. Faloutsos i N. D. Sidiropoulos, *Parcube: Sparse parallelizable tensor decompositions*, Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2012, str. 521–536.

- [57] ———, *Tensors for data mining and data fusion: Models, applications, and scalable algorithms*, ACM Transactions on Intelligent Systems and Technology (TIST) **8** (2017), br. 2, 16.
- [58] E. E. Papalexakis, N. D. Sidiropoulos i R. Bro, *From k-means to higher-way co-clustering: Multilinear decomposition with sparse latent factors*, IEEE transactions on signal processing **61** (2012), br. 2, 493–506.
- [59] B. Peng, G. Qian i S. Rajko, *View-invariant full-body gesture recognition from video*, 2008 19th International Conference on Pattern Recognition, IEEE, 2008, str. 1–5.
- [60] L. Periša, *Recompression of Hadamard products of tensors in Tucker format*, Disertacija, Prirodoslovno-matematički fakultet, Sveučilište u Zagrebu, 2017.
- [61] L. N. Ribeiro, A. L.F. de Almeida i V. Zarzoso, *Enhanced block term decomposition for atrial activity extraction in atrial fibrillation ECG*, 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM), IEEE, 2016, str. 1–5.
- [62] B. Savas i L. H. Lim, *Quasi-Newton methods on Grassmannians and multilinear approximations of tensors*, SIAM Journal on Scientific Computing **32** (2010), br. 6, 3352–3393.
- [63] M. D. Schatz, T. M. Low, R. A. Van De Geijn i T. G. Kolda, *Exploiting symmetry in tensors for high performance: Multiplication with symmetric tensors*, SIAM Journal on Scientific Computing **36** (2014), br. 5, C453–C479.
- [64] N. D. Sidiropoulos, R. Bro i G. B. Giannakis, *Parallel factor analysis in sensor array processing*, IEEE Transactions on Signal Processing **48** (2000), br. 8, 2377–2388, ISSN 1941-0476.
- [65] N. D. Sidiropoulos i R. S. Budampati, *Khatri-Rao space-time codes*, IEEE Transactions on Signal Processing **50** (2002), br. 10, 2396–2407.
- [66] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis i C. Faloutsos, *Tensor decomposition for signal processing and machine learning*, IEEE Transactions on Signal Processing **65** (2017), br. 13, 3551–3582.
- [67] N. D. Sidiropoulos, G. B. Giannakis i R. Bro, *Blind PARAFAC receivers for DS-SS-CDMA systems*, IEEE Transactions on Signal Processing **48** (2000), br. 3, 810–823.
- [68] A. Smilde, R. Bro i P. Geladi, *Multi-Way Analysis with Applications in the Chemical Sciences. 2004*, Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

- [69] S. Wang, Y. Shen, W. Chen, T. Xiao i J. Hu, *Automatic recognition of mild cognitive impairment from mri images using expedited convolutional neural networks*, International Conference on Artificial Neural Networks, Springer, 2017, str. 373–380.
- [70] W. Wang, V. Aggarwal i S. Aeron, *Efficient low rank tensor ring completion*, Proceedings of the IEEE International Conference on Computer Vision, 2017, str. 5697–5705.
- [71] W. Wang, Y. Sun, B. Eriksson, W. Wang i V. Aggarwal, *Wide compression: Tensor ring nets*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, str. 9329–9338.
- [72] S. Xie i Z. Tu, *Holistically-nested edge detection*, Proceedings of the IEEE international conference on computer vision, 2015, str. 1395–1403.
- [73] Y. Yang, Denis K. i V. Tresp, *Tensor-train recurrent neural networks for video classification*, Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, str. 3891–3900.
- [74] J. Ye, L. Wang, G. Li, D. Chen, S. Zhe, X. Chu i Z. Xu, *Learning compact recurrent neural networks with block-term tensor decomposition*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, str. 9378–9387.
- [75] V. Zarzoso, *Parameter estimation in block term decomposition for noninvasive atrial fibrillation analysis*, 2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), IEEE, 2017, str. 1–5.
- [76] H. B. Zhang, Y. X. Zhang, B. Zhong, Q. Lei, L. Yang, J. X. Du i D. S. Chen, *A Comprehensive Survey of Vision-Based Human Action Recognition Methods*, Sensors **19** (2019), br. 5, 1005.
- [77] J. Zhang, Y. Han i J. Jiang, *Tucker decomposition-based tensor learning for human action recognition*, Multimedia Systems **22** (2016), br. 3, 343–353.
- [78] Q. Zhang, L. T. Yang, X. Liu, Z. Chen i P. Li, *A tucker deep computation model for mobile multimedia feature learning*, ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) **13** (2017), br. 3s, 39.
- [79] Z. Zhang, X. Yang, I. V. Oseledets, G. E. Karniadakis i L. Daniel, *Enabling high-dimensional hierarchical uncertainty quantification by ANOVA and tensor-train decomposition*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **34** (2014), br. 1, 63–76.

- [80] Q. Zhao, G. Zhou, S. Xie, L. Zhang i A. Cichocki, *Tensor ring decomposition*, arXiv preprint arXiv:1606.05535 (2016).
- [81] M. Zhou, Y. Liu, Z. Long, L. Chen i C. Zhu, *Tensor rank learning in CP decomposition via convolutional neural network*, Signal Processing: Image Communication **73** (2019), 12–21.

Sažetak

U ovom radu dan je pregled osnovnih obilježja multilinearne algebre čiji su središnji objekti tenzori. Potom su detaljno opisane najpopularnije tenzorske dekompozicije CP i Tucker, dotičući se pojma ranga, svojstva (ne)jedinственosti, algoritama za računanje dekompozicija te općenite primjene i primjene u prepoznavanju radnje. Također su opisane i dekompozicije vlačna, prstenasta i blokovska, koje se tek posljednjih godina koriste u području prepoznavanja radnje. Na kraju je implementiran algoritam Tuckerove *ridge* regularizacije, koji kombinacijom Tuckerove dekompozicije i *ridge* regularizacije uspjeva relativno dobro raspoznavati kretnje sa statičnih slika s raznolikim pozadinama.

Summary

In this thesis we first give an overview of the basic concepts in multilinear algebra whose core objects are tensors. Then we introduce the most known and used tensor decompositions CP and Tucker, where we also discuss ranks of these decompositions, their uniqueness property or the lack of it, algorithms for computing the decompositions and the various applications, with a special emphasis on action recognition. We also talk about the tensor train, tensor ring and block term decompositions which have recently been more used in the field of action recognition. The end of the thesis we dedicate to the implementation of Tucker ridge regression, an algorithm for recognizing actions from static images.

Životopis

Rođena sam 3.8.1995. u Slavonskom Brodu. Pohađala sam Osnovnu školu "Viktor Car Emin" u Donjim Andrijevcima i Gimnaziju "Matija Mesić" u Slavonskom Brodu. 2014. godine sam upisala Preddiplomski sveučilišni studij Matematike na Prirodoslovno-matematičkom fakultetu u Zagrebu, a nakon završetka 2017. i Diplomski studij Računarstva i matematike na istom fakultetu.