Tensor Image Clustering

Implementacija u Matlab-u

Karlo Grozdanić, Mislav Jelašić, Luka Karlić

Sveučilište u Zagrebu
Prirodoslovno-matematički fakultet, Matematički odsjek

23. veljače 2022.

Uvod

- Digitalne slike prirodno spremamo u obliku matrica (tenzora drugog reda)
- Želimo kategorizirati slike prema sadržaju
- Koristimo tenzorski pristup za analizu
- Implementiramo spektralno klasteriranje kao kombinaciju redukcije dimenzije i tradicionalnog algoritma
- Rezultati se mogu proširiti na tenzore viših dimenzija

Algebra tenzora

- Tenzor reda k je realni multilinearni funkcional nad k vektorskih prostora: $T : \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_k} \to \mathbb{R}$
- Skup svih tenzora reda k, $\mathcal{T}^k$ je vektorski prostor sa standardnim operacijama zbrajanja po točkama i skalarnog množenja
- Za tenzore $S \in \mathcal{T}^k$ i $T \in \mathcal{T}^l$ definiramo njihov tenzorski produkt:

$$S \otimes T : \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_{k+l}} \to \mathbb{R},$$

$$S \otimes T(a_1, ..., a_{k+l}) = S(a_1, ..., a_k)T(a_{k+1}, ..., a_{k+l})$$

- Tenzori prvog reda su dualni vektori na $\mathbb{R}^{n_1}$, oznake $\mathcal{T}^1 = \mathcal{R}^{n_1}$
- Prostor tenzora drugog reda je produkt dva prostora tenzora prvog reda, tj. $\mathcal{T}^2 = \mathcal{R}^{n_1} \otimes \mathcal{R}^{n_2}$

Funkcija cilja

- Dano je m točaka $\mathcal{X} = \{X_1, ..., X_m\}$, $X_i \in \mathcal{M} \in \mathcal{R}^{n_1} \otimes \mathcal{R}^{n_1}$
- Formiramo graf susjedstva i pripadnu matricu težina:

$$S_{ij} = \begin{cases} 1, & \textit{ako se } X_i \textit{ nalazi među p najbližih susjeda od } X_j \\ & \textit{ili se } X_j \textit{ nalazi među p najbližih susjeda od } X_i, \\ 0, & \textit{inače}. \end{cases}$$

- Želimo pronaći optimalnu particiju $\mathcal{G} = \mathcal{G}_1 \cup \mathcal{G}_2$, $\mathcal{G}_1 \cap \mathcal{G}_2 = \emptyset$:

$$\min_{\mathcal{G}_1, \mathcal{G}_2} \sum_{i \in \mathcal{G}_1} \sum_{j \in \mathcal{G}_2} S_{ij}$$

- Svakoj točki X_i pridružujemo oznaku pripadnosti $y_i \in \{-1, 1\}$ podgrafu $\mathcal{G}_1$, odnosno $\mathcal{G}_2$:

$$\min_{\mathbf{y} \in \{-1,1\}^m} \sum_{i,j} (y_i - y_j)^2 S_{ij}$$

Metoda

- Uzimamo za y_i realne brojeve i pretpostavljamo da je preslikavanje $X_i \mapsto y_i$ linearno, tj. $y_i = u^T X_i v$:

$$\min_{U,V} \sum_{i,j} ||U^T X_i V - U^T X_j V||^2 S_{ij}$$

- Raspisivanjem izraza dobivamo optimizacijski problem:

$$\begin{cases} \min_{U,V} \frac{\mathrm{tr}\left(U^T(D_V - S_V)U\right)}{\mathrm{tr}\left(U^T D_V U\right)}, \\ \min_{U,V} \frac{\mathrm{tr}\left(V^T(D_U - S_U)V\right)}{\mathrm{tr}\left(V^T D_U V\right)}, \end{cases}$$

gdje su D_V, S_V, D_U, S_U matrice koje ovise o X_i, U, V, S_{ij}

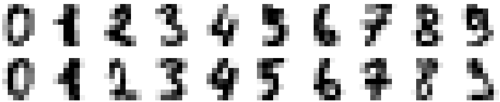
- Rješenje iterativnom metodom generaliziranih sv. vektora:

$$(D_U - S_U)\mathbf{v} = \lambda D_U \mathbf{v}$$

$$(D_V - S_V)\mathbf{u} = \lambda D_V \mathbf{u}$$

Podatci

- Znamenke

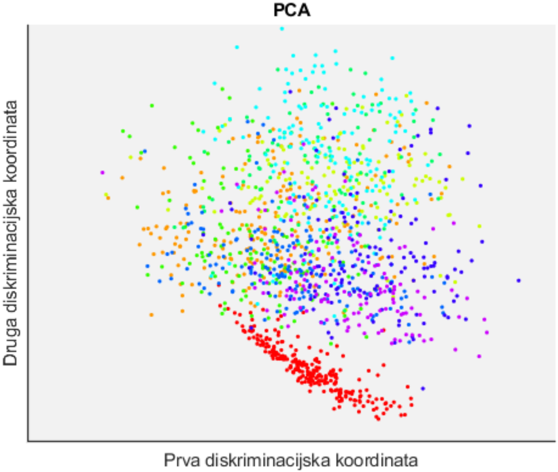


- Lica

- Preprocesuirana lica

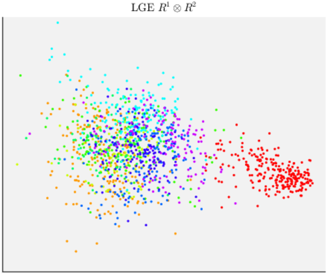
Metode - PCA

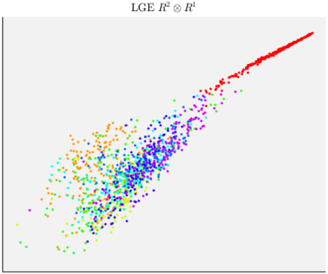
- k-means s prvih 10 disk.koord. daje točnost: **60.28%**



Metode - Linear Graph Embedding

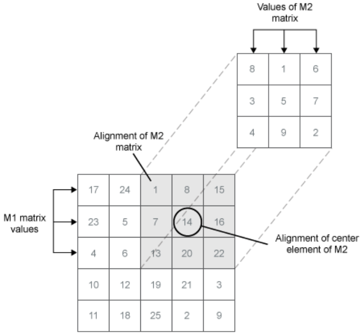
- k-means na 13×13 prostoru daje: **52.49%**





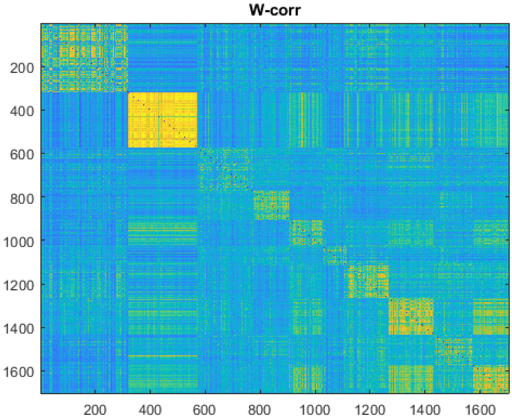
Metode - TensorImage, dobar W

- gledanje euklidske udaljenosti između slika neće dati dobre rezultate
- promašaj ako su slike malo pomaknute
- unakrsna korelacija bolji izbor!



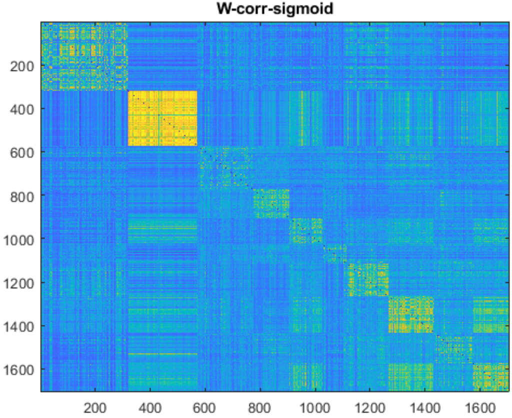
Metode - TensorImage, dobar W

- nije dovoljno strogo



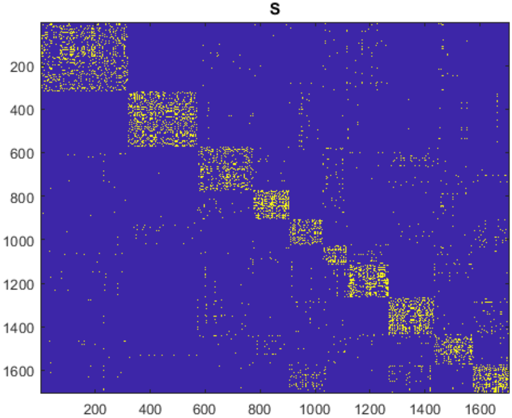
Metode - TensorImage, dobar W

- Primijenimo sigmoid funkciju, bolje!



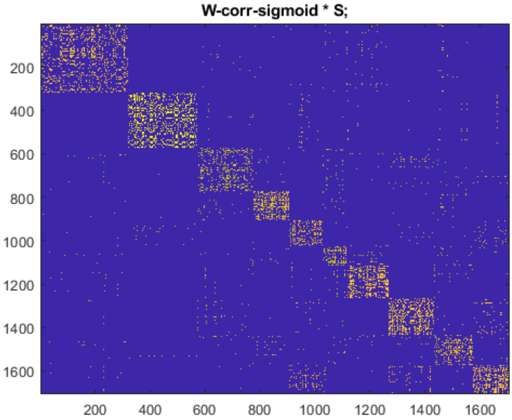
Metode - TensorImage, S

- Matrica susjedstva S
- Bilo bi lijepo još ubaciti informaciju o udaljenost

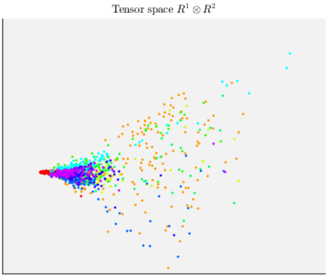


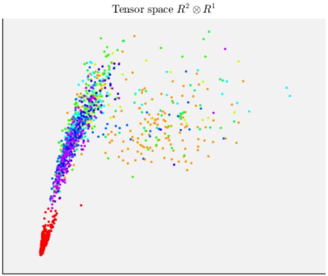
Metode - TensorImage, $S \circ W_{corr+sigmoid}$

- Konačna matrica



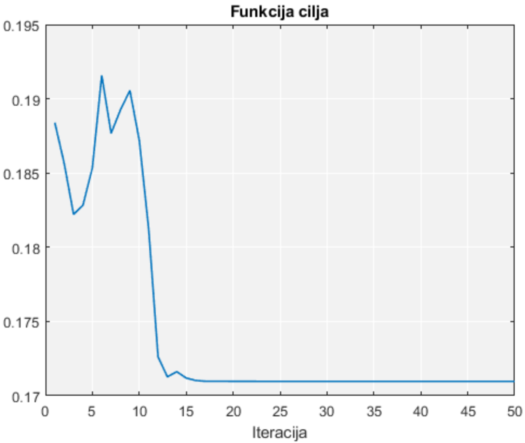
Metode - TensorImage, projekcija





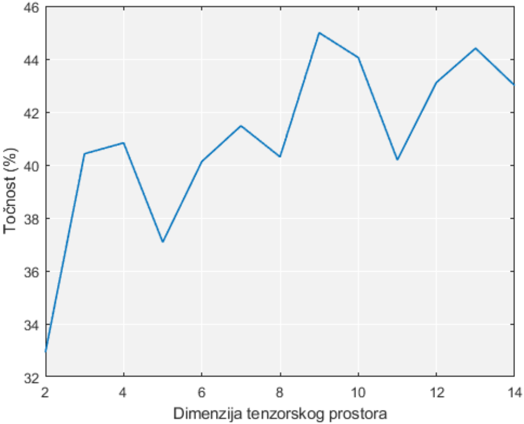
Metode - TensorImage, funkcija cilja

- Stagnacija nakon nekog vremena - kriterij zaustavljanja!



Metode - TensorImage, odnos dimenzije prostora i točnosti

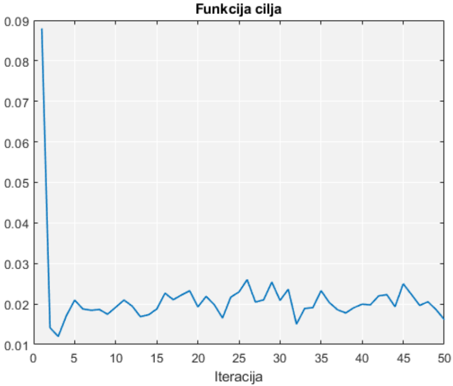
- Problem nalaženja optimalne dimenzije
- korištena matrica: $S \circ W_{corr+sigmoid}$



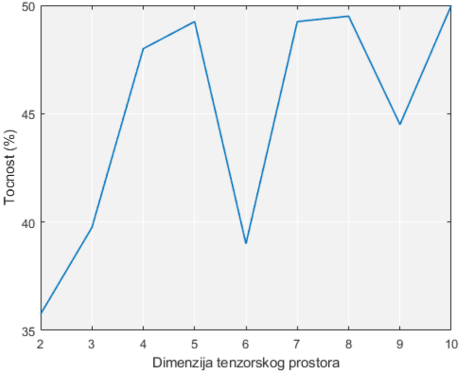
Yale podaci: lica

- novi načini dobivanja matrice udaljenosti
- metrika: kosinus vs. euklidska
- knn (uz euklidsku)
- težina? 'heat kernel'/kosinus

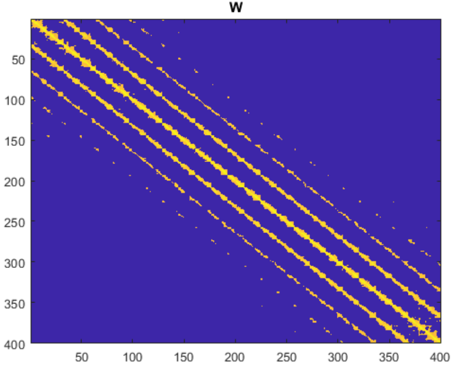
knn = 20, k = 5, metrika: cosine, d = 2



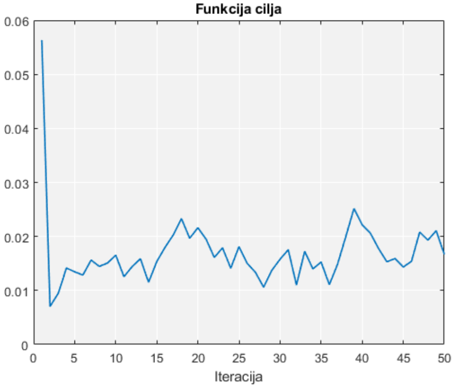
knn = 20, k = 8, metrika: cosine, d = 2



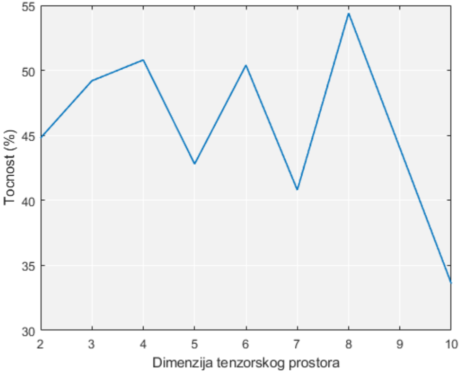
knn = 20, k = 8, metrika: cosine, d = 2



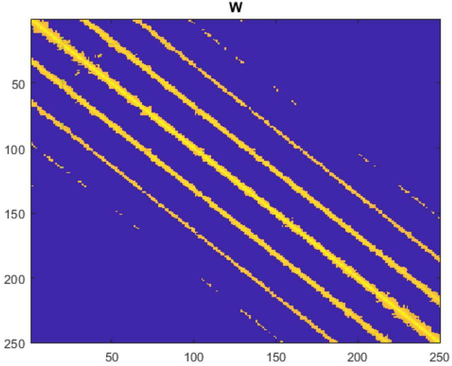
knn = 20, k = 5, metrika: cosine, d = 2



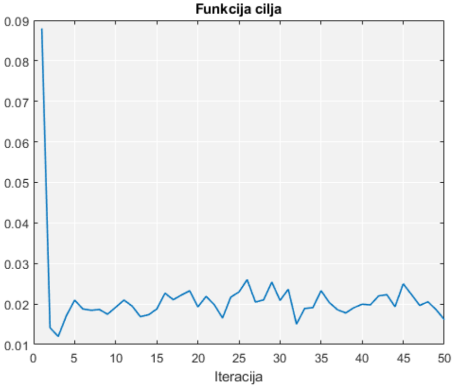
knn = 20, k = 5, metrika: cosine, d = 2



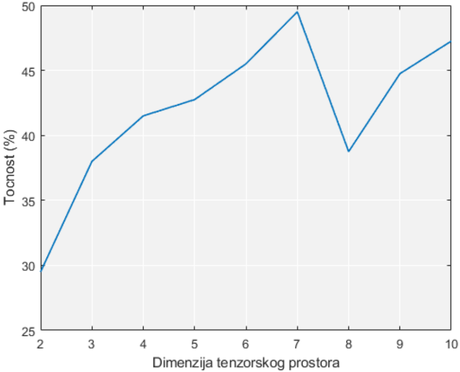
knn = 20, k = 5, metrika: cosine, d = 2



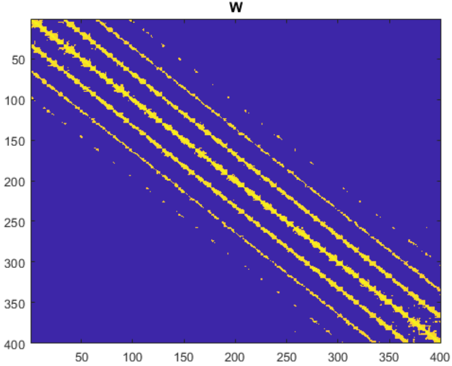
knn = 20, k = 8, metrika: euclid, d = 2



knn = 20, k = 8, metrika: euclid, d = 2



knn = 20, k = 8, metrika: euclid, d = 2



Usporedba s NCut algoritmom

Table: Točnost TensorImage metode naspram NCut metodi u ovisnosti o parametrima matrice susjedstva

	max % TensorImage	**max % Ncuts**
knn=20, cosine, k=5, d=2	55%	77.6%
knn=20, cosine, k=8, d=2	50%	72.8%
knn=20, euclid, k=5, d=2	56%	81.25%
knn=20, euclid, k=8, d=2	50%	80.25%
knn=20, euclid, k=8, d=5	60%	75%
knn=20, euclid, k=8, d=30	55%	72%

Karlo, Mislav i Luka kad saznaju da Ncut i dalje dominira...

Hvala na pažnji!