Team random_state=42

Final Project for NLA course 2023

https://github.com/justfollowthesun/ce-retrieval

Problem

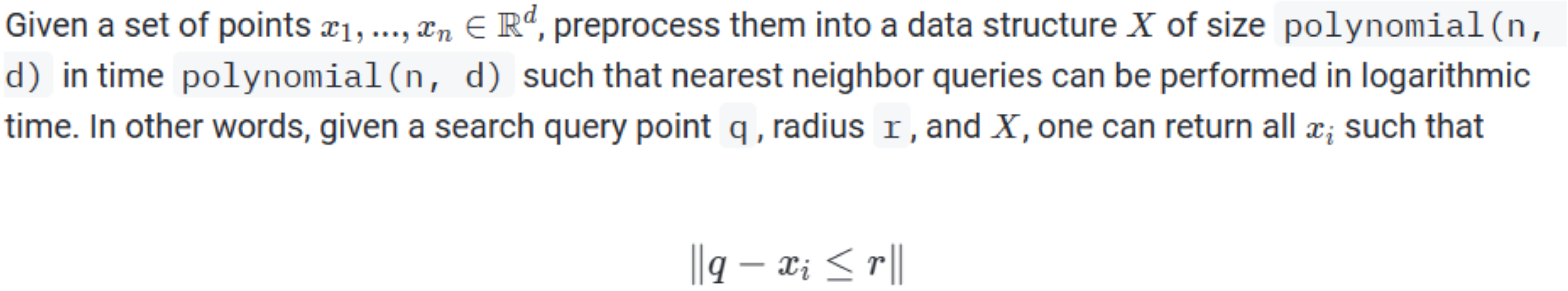
Pairwise sentence scoring tasks have wide applications in NLP. They can be used in information retrieval, question answering, duplicate question detection, or clustering. An approach that sets new state-of-the-art performance for many tasks including pairwise sentence scoring is BERT. Both sentences are passed to the network and attention is applied across all tokens of the inputs. This approach, where both sentences are simultaneously passed to the network, is called cross-encoder.

A downside of cross-encoders is the extreme computational overhead for many tasks. For example, clustering of 10,000 sentences has a quadratic complexity with a cross-encoder and would require about 65 hours with BERT[1].

[1] Sentence-BERT: Sentence Embeddings using Siamese BERT-Network, doi:https://aclanthology.org/D19-1410.pdf

Formal statement about the problem

Nearest Neighbor Search definition



Given a set of points $x_1, ..., x_n \in \mathbb{R}^d$, preprocess them into a data structure X of size `polynomial(n, d)` in time `polynomial(n, d)` such that nearest neighbor queries can be performed in logarithmic time. In other words, given a search query point `q`, radius `r`, and X, one can return all x_i such that

$$\|q - x_i \leq r\|$$

Nearest Neighbor Search, https://calvinfeng.gitbook.io/machine-learning-notebook/supervised-learning/search/nearest_neighbor_search

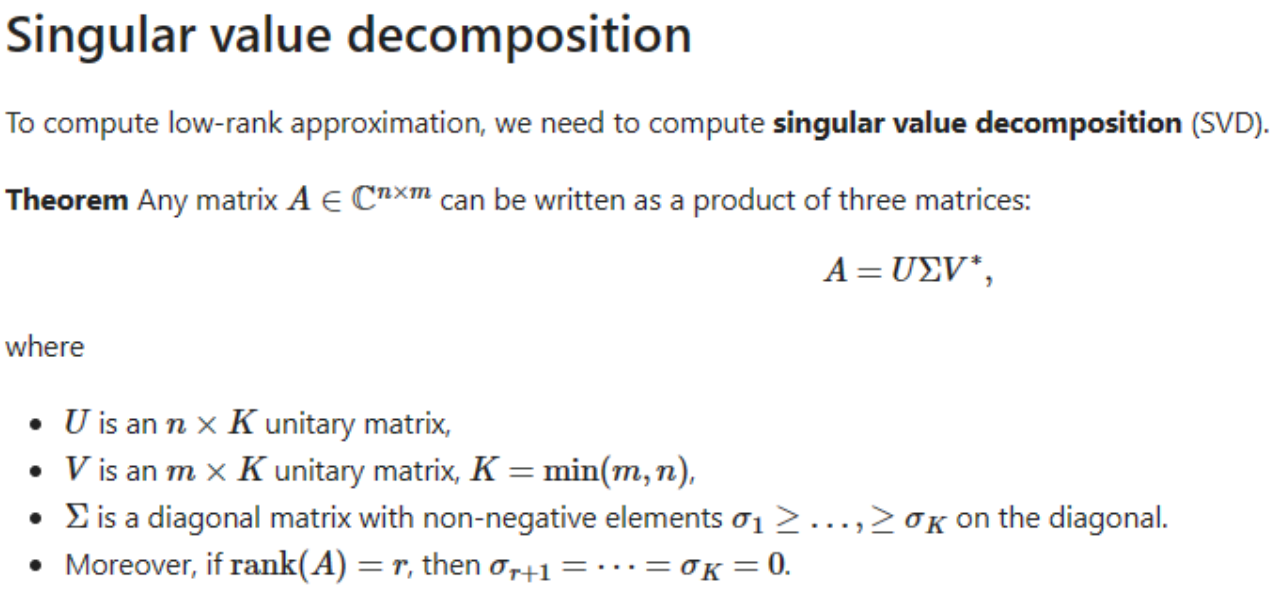
The existing solutions

An approach[2] that avoids the use of a dual-encoder for retrieval, relying solely on the cross-encoder. Retrieval is made efficient with CUR decomposition, a matrix decomposition approach that approximates all pairwise cross-encoder distances from a small subset of rows and columns of the distance matrix.

[2] Efficient Nearest Neighbor Search for Cross-Encoder Models using Matrix Factorization, doi:https://aclanthology.org/2022.emnlp-main.140.pdf

NLA techniques in project

1 SVD



Singular value decomposition

To compute low-rank approximation, we need to compute **singular value decomposition** (SVD).

Theorem Any matrix $A \in \mathbb{C}^{n\times m}$ can be written as a product of three matrices:

$$A = U \Sigma V^*,$$

where

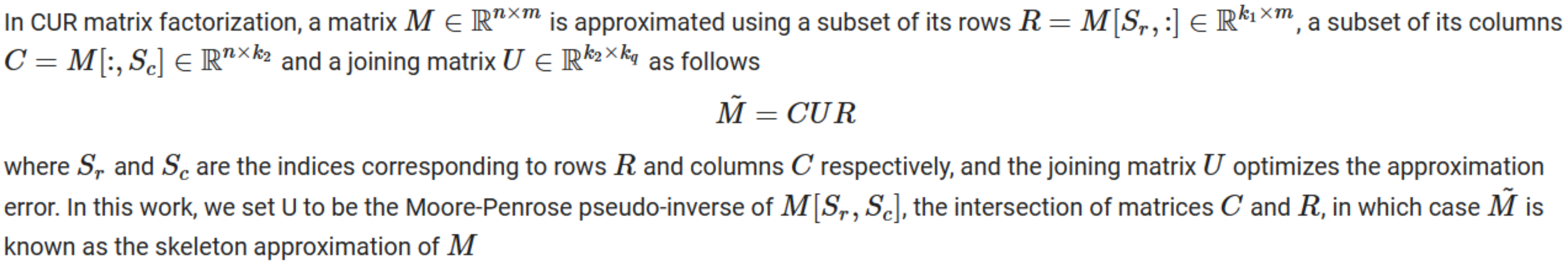
- U is an $n \times K$ unitary matrix,
- V is an $m \times K$ unitary matrix, $K = \min(m, n)$,
- Σ is a diagonal matrix with non-negative elements $\sigma_1 \geq \ldots, \geq \sigma_K$ on the diagonal.
- Moreover, if $\mathrm{rank}(A) = r$, then $\sigma_{r+1} = \cdots = \sigma_K = 0$.

The SVD is expensive to compute, it may not be significantly more expensive than alternative factorizations. However, the SVD is expensive to update when a row or column is added to or removed from the matrix, as happens repeatedly in signal processing applications.

Images source : https://github.com/oseledets/nla2023/blob/main/lectures/lecture-4/lecture-4.ipynb
What Is a Rank-Revealing Factorization?, https://nhigham.com/2021/05/19/what-is-a-rank-revealing-factorization/

NLA techniques in project

2 CUR decomposition



In CUR matrix factorization, a matrix $M \in \mathbb{R}^{n \times m}$ is approximated using a subset of its rows $R = M[S_r, :] \in \mathbb{R}^{k_1 \times m}$, a subset of its columns $C = M[:, S_c] \in \mathbb{R}^{n \times k_2}$ and a joining matrix $U \in \mathbb{R}^{k_2 \times k_q}$ as follows

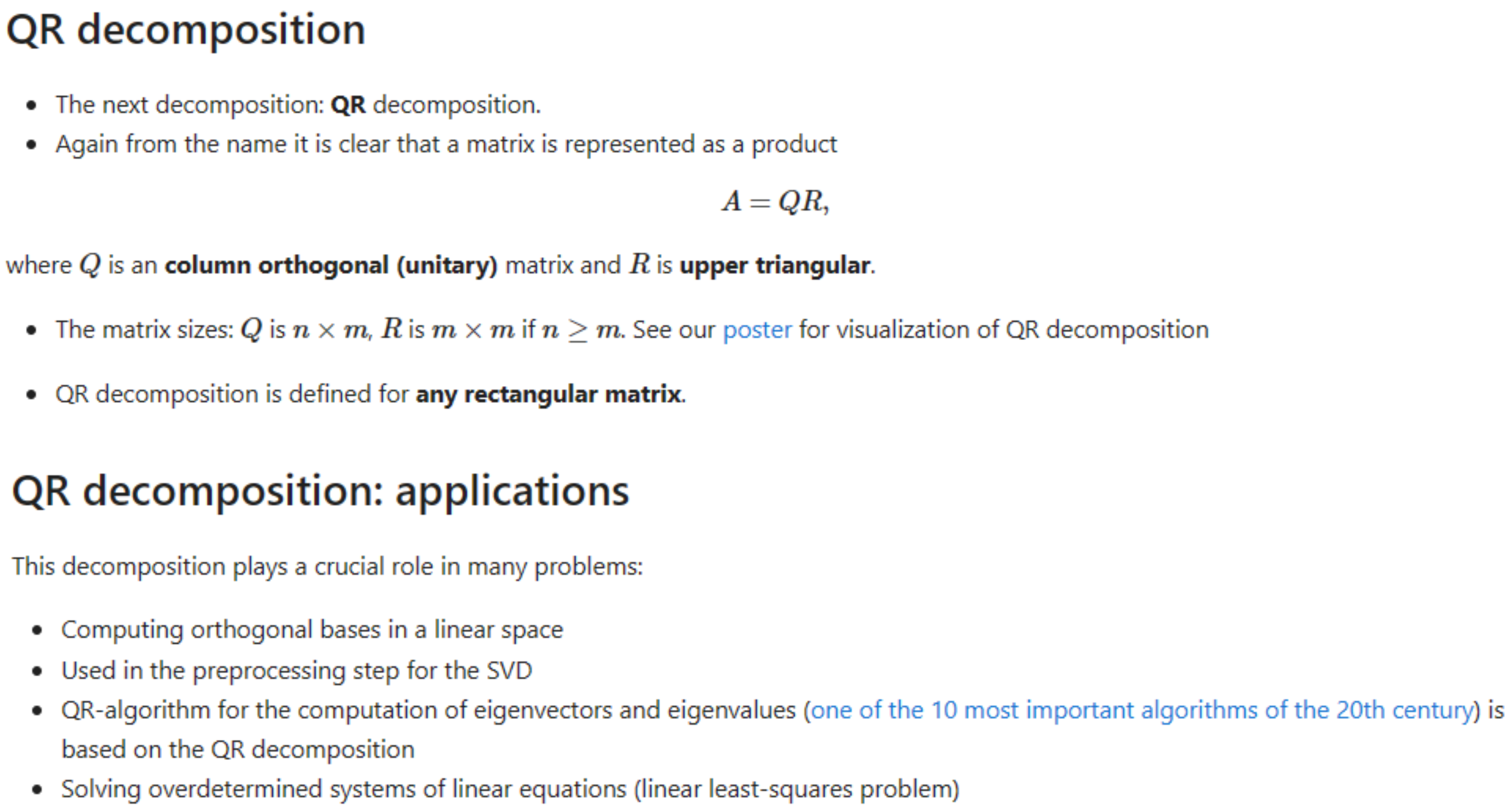
$$\tilde{M} = CUR$$

where S_r and S_c are the indices corresponding to rows R and columns C respectively, and the joining matrix U optimizes the approximation error. In this work, we set U to be the Moore-Penrose pseudo-inverse of $M[S_r, S_c]$, the intersection of matrices C and R, in which case $\tilde{M}$ is known as the skeleton approximation of M

CUR matrix decompositions for improved data analysis, https://doi.org/10.1073/pnas.0803205106
Efficient Nearest Neighbor Search for Cross-Encoder Models using Matrix Factorization, doi:https://aclanthology.org/2022.emnlp-main.140.pdf

Techniques comparison

3 QR decomposition



QR decomposition

- The next decomposition: **QR** decomposition.
- Again from the name it is clear that a matrix is represented as a product

$$A = QR,$$

where Q is an **column orthogonal (unitary)** matrix and R is **upper triangular**.

- The matrix sizes: Q is $n \times m$, R is $m \times m$ if $n \geq m$. See our poster for visualization of QR decomposition

- QR decomposition is defined for **any rectangular matrix**.

QR decomposition: applications

This decomposition plays a crucial role in many problems:

- Computing orthogonal bases in a linear space
- Used in the preprocessing step for the SVD
- QR-algorithm for the computation of eigenvectors and eigenvalues (one of the 10 most important algorithms of the 20th century) is based on the QR decomposition
- Solving overdetermined systems of linear equations (linear least-squares problem)

Images source : https://github.com/oseledets/nla2023/blob/main/lectures/lecture-7/lecture-7.ipynb

The proposed solution

Implement Singular Value Decomposition (SVD) and QR decomposition within the approach procedure, and subsequently, analyze and contrast the outcomes acquired from experiments.

Experimental pipeline

1

Download data

ZeShEL was constructed using Wikias from FANDOM and is licensed under the Creative Commons Attribution-Share Alike License (CC-BY-SA).

2

Tokenize and compute score matrix

We used bert-base_uncased model for tokenization then compute cross-encoder scores for all item in the data.

3

Evaluate cross encoder model

We retrieve k-entities for each test mention using matrix decomposition approximation and re-rank them using a cross-encoder model

4

Quality and time measurement for each approaches

We measured quality and time for both CUR and SVD approaches in score matrix factorization.

Results and comparisons.

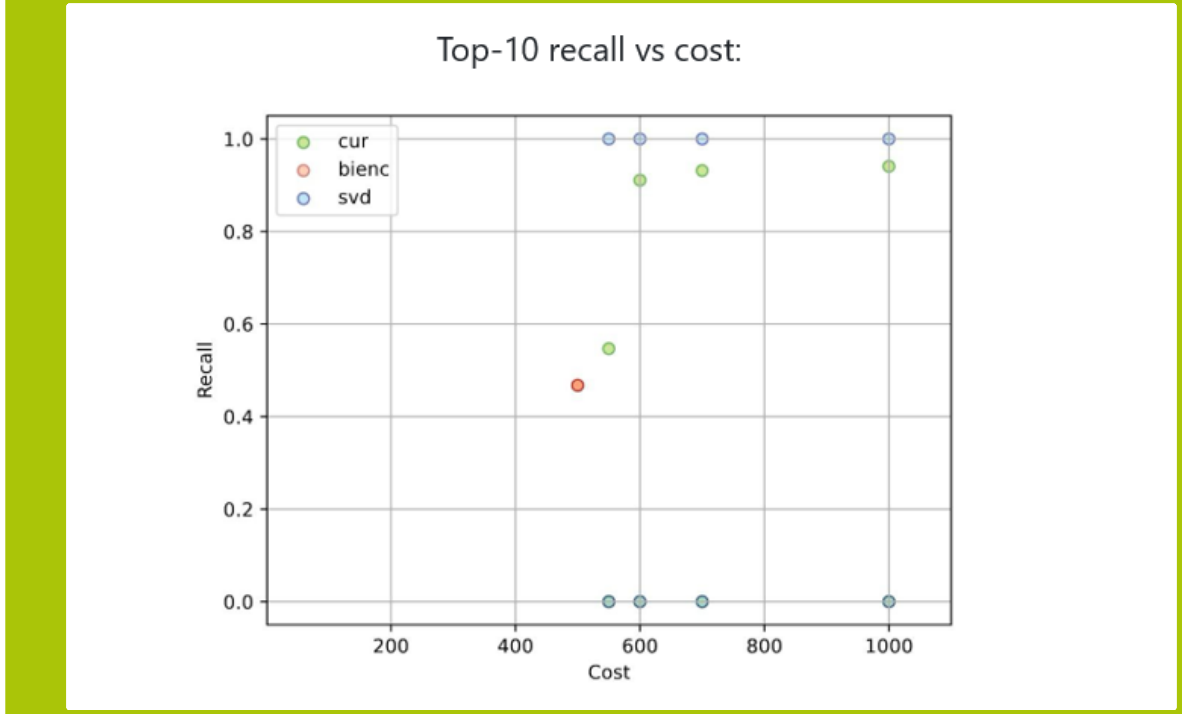


Figure 1 : Comparison of top-10 recall and its cost for each approaches

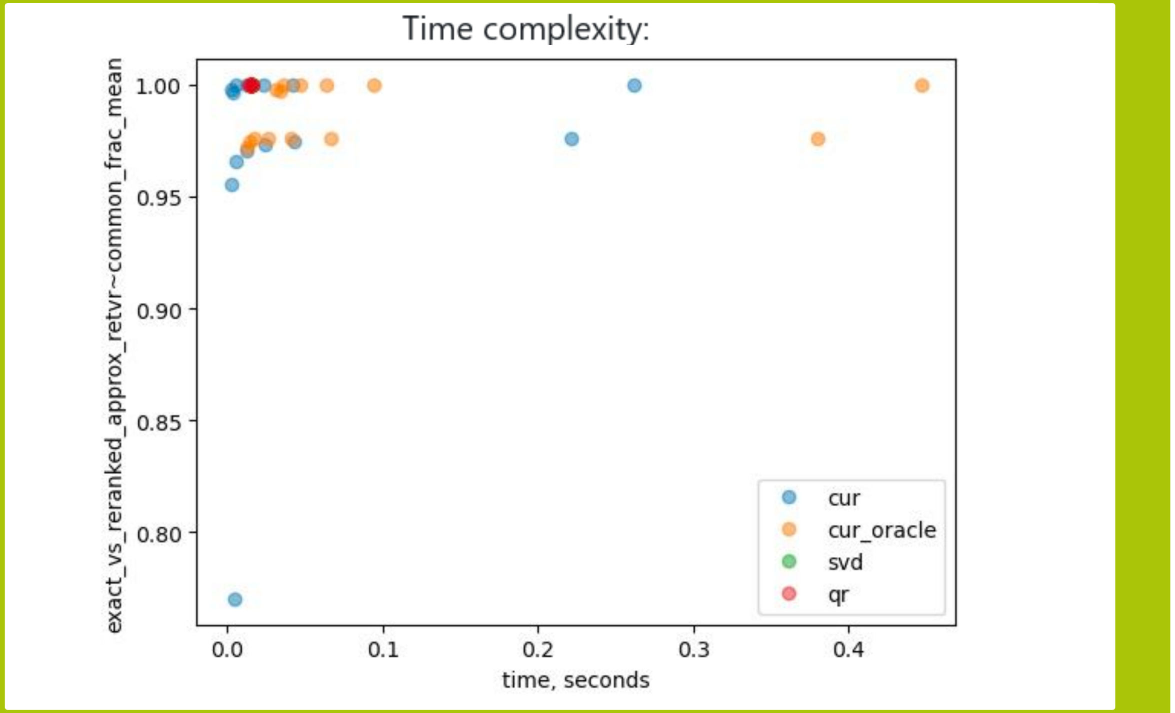


Figure 2 : Comparison of time complexity for each approaches

Results and comparisons.

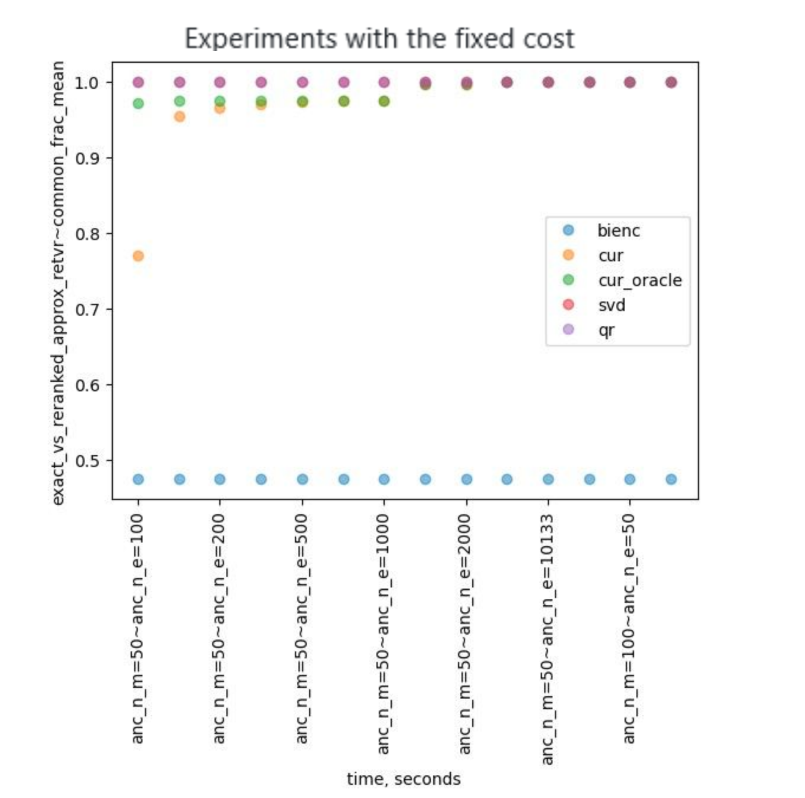


Figure 3 : Comparison of experiment with the fixed cost for each approaches

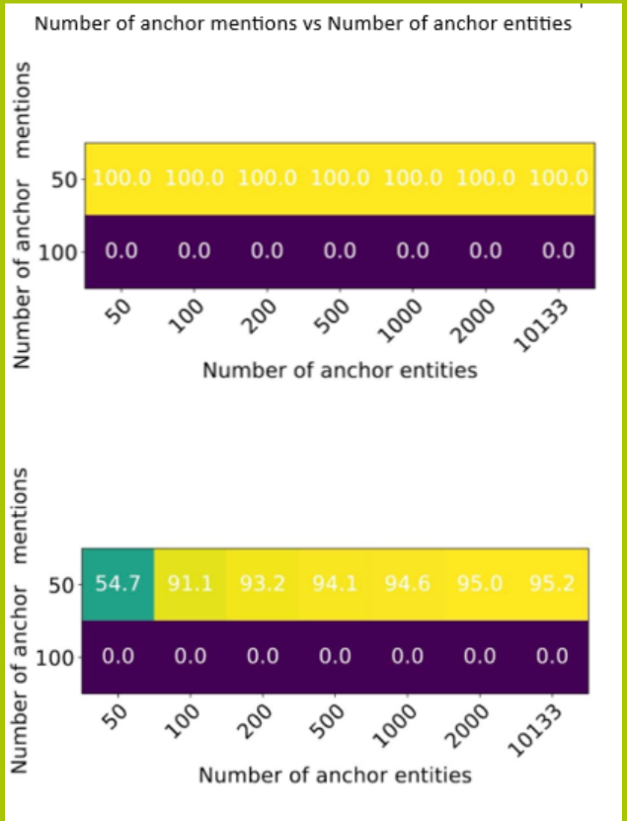


Figure 4 : Comparison between number of anchor mentions and number of anchor entities

SVD shows the highest decomposition quality (which additionally follows from the Eckart-Young theorem), but is not quite optimal in terms of time complexity

Results and comparisons.

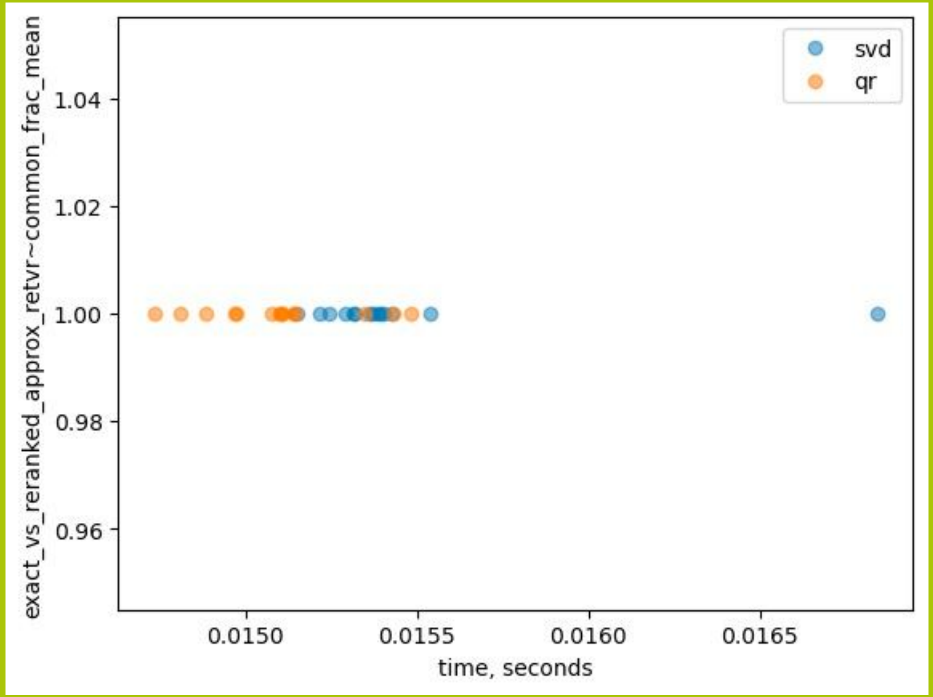


Figure 5 : Comparison of time complexity between SVD and QR decomposition

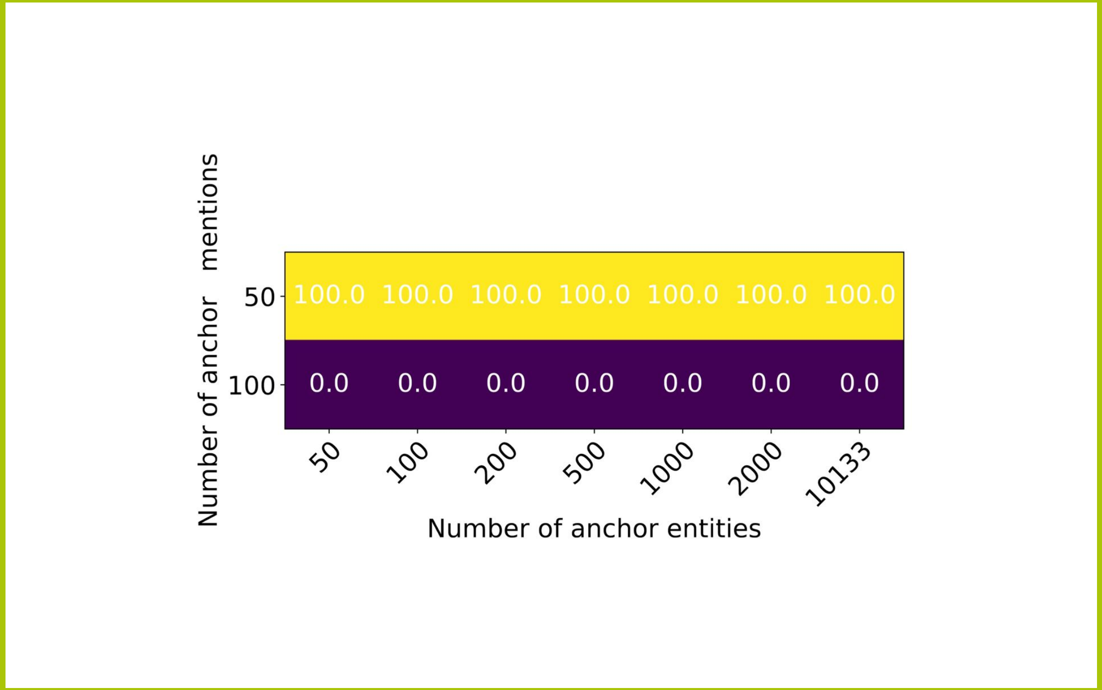


Figure 6 : Comparison between number of anchor mentions and number of anchor entities of QR decomposition

Members

- Ivan Anisimov : Reproducing the results of the original work, writing the readme
- Ivan Borisov : writing svd
- Andrey Safronov : adding svd, reproducing results
- Rattamet Boonwong : presentation
- Adithya Shetty : added QR

thx.

Skoltech