FAST SVD

Making your recommendation systems faster

by Borisov Denis, Kalenyuk Akim, and Krasnov Michail

Introduction. **SVD Algorithm (Singular Value Decomposition) is widely used for recommendation purposes. However, it can be quite slow on large datasets.**

That is why, in our team project, we decided to consider various ways of accelerating our algorithm.

The main information source for us was the paper Faster "Matrix Completion Using Randomized SVD" by Xu Feng, Wenjian Yu, and Yaohang Li.

In this paper authors presented the method to accelerate matrix completion using faster randomized singular value decomposition.

In this presentation we will observe

- **Singular Value Decomposition (SVD)**
- **Randomized Algorithms**
- **Randomized SVD (rSVD)**
- **Algorithms Overview**
- **from rSVD to rSVT (another presentation)**
- **Our results**
- **What is next?**

But let's start

From the beginning

Singular Value Decomposition

Singular Value Decomposition (SVD) is a matrix factorization method used to uncover the hidden structure within data. It decomposes a matrix into three components: U, Σ (Sigma), and V^T, where Σ contains singular values that quantify the significance of data, and U and V represents the left and right singular vectors matrices.

SVD finds crucial application in Matrix Completion, where partial data is observed. Singular Value Thresholding, built upon SVD, plays a pivotal role in addressing this issue. By truncating singular values, it reduces matrix rank while preserving essential information, allowing accurate completion of missing data.

SVD's adaptability and efficiency make it an invaluable tool for diverse scientific fields, providing a pathway to uncover valuable insights and address data recovery challenges.

However,

SVD is not perfect!

Indeed, SVD is lack of speed on large datasets.

Randomized algorithms

Many standard algorithms for dealing with matrices are expensive or intractable in large-scale machine learning and statistics. For example, given an m×n matrix A where both m and n are large, a method such as singular value decomposition (SVD) will require memory and time which is superlinear in m and n. The main idea of randomized algorithms is to build up random, but good approximation for further computations.

What about Random SVD?

- Conventional alg. for dense matrix - $O(n^2 k)$

- Conventional alg. for sparse matrix - $O(N * k)$, where N - number of non-zero elements

- Random SVD method reducing the constant

Random SVD algorithm

- The goal of a randomized range finder is to produce an orthonormal matrix Q with as few columns as possible such that $\|(I-QQ)A\| \leq \varepsilon$ for the tolerance ε .
- The first idea is to watch on low rank matrix in reduced space. The easiest way is using of random projections to obtain main actions of matrix

rSV
D

Randomized Singular Value Decomposition is an approach to computing the Singular Value Decomposition (SVD) of a matrix using randomization techniques.

It's designed to approximate the dominant singular values and corresponding singular vectors of a matrix, often with significantly reduced computational cost compared to traditional SVD methods. rSVD is particularly useful for large and sparse matrices.

Generate a random Gaussian (or other suitable) matrix and multiply it with the original matrix. This creates a smaller, dense matrix, preserving the matrix's structure while reducing its size.

Compute the SVD of the smaller dense matrix. Since it's smaller, the SVD calculation is less computationally expensive.

Use the results from the SVD of the smaller matrix to approximate the dominant singular values and vectors of the original matrix.

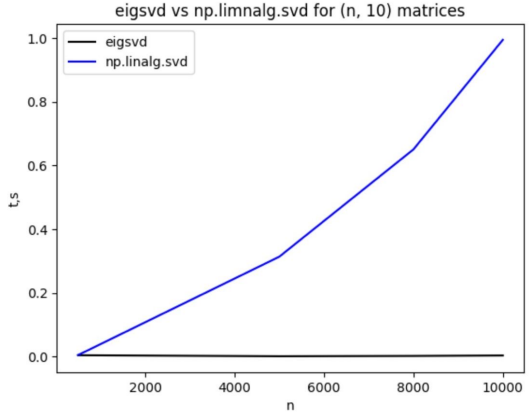
Repeat the process multiple times to refine the approximations:)

Input: $\mathbf{A} \in \mathbb{R}^{m \times n}$, rank parameter k, power parameter p
Output: $\mathbf{U} \in \mathbb{R}^{m \times k}$, $\mathbf{S} \in \mathbb{R}^{k \times k}$, $\mathbf{V} \in \mathbb{R}^{n \times k}$

$\mathbf{\Omega} = \text{randn}(n, k + s)$
$\mathbf{Q} = \text{orth}(\mathbf{A\Omega})$
for $i = 1, 2, \cdots, p$ **do**
    $\mathbf{G} = \text{orth}(\mathbf{A}^{\mathrm{T}}\mathbf{Q})$
    $\mathbf{Q} = \text{orth}(\mathbf{AG})$
end for
$\mathbf{B} = \mathbf{Q}^{\mathrm{T}}\mathbf{A}$
$[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \text{svd}(\mathbf{B})$
$\mathbf{U} = \mathbf{QU}$
$\mathbf{U} = \mathbf{U}(:, 1:k), \mathbf{S} = \mathbf{S}(1:k, 1:k), \mathbf{V} = \mathbf{V}(:, 1:k).$

EigSVD

EigSVD is an algorithm designed to extract the underlying structure of a matrix, focusing on its Singular Value Decomposition (SVD). The primary goal is to efficiently compute the SVD of a given matrix, particularly when dealing with sparse matrices. By utilizing eigendecomposition, EigSVD aims to identify the dominant singular values and their corresponding vectors.



EigSVD serves as a fundamental tool in applications such as matrix completion, where the objective is to efficiently factorize large, potentially sparse matrices. This approach allows for accurate approximation of missing data, reducing the computational complexity while preserving essential matrix characteristics. The essence of EigSVD lies in its ability to reveal the latent patterns within complex data, making it a valuable asset in various scientific and engineering fields.

Input: $\mathbf{A} \in \mathbb{R}^{m \times n}$ $(m \geq n)$

Output: $\mathbf{U} \in \mathbb{R}^{m \times n}$, $\mathbf{S} \in \mathbb{R}^{n \times n}$, $\mathbf{V} \in \mathbb{R}^{n \times n}$

1: $\mathbf{B} = \mathbf{A}^{\mathrm{T}}\mathbf{A}$

2: $[\mathbf{V}, \mathbf{D}] = \mathrm{eig}(\mathbf{B})$

3: $\mathbf{S} = \mathrm{sqrt}(\mathbf{D})$

4: $\mathbf{U} = \mathbf{A}\mathbf{V}\mathbf{S}^{-1}$

rSVD-PI

The rSVD-PI (Randomized Singular Value Decomposition with Power Iteration) algorithm is used to quickly calculate the SVD (Singular Value Decomposition) of large matrices. It is based on a combination of random projection and power iterations. First, a random projection of the original matrix onto a subspace of a smaller dimension is performed, then power-law iterations are performed to obtain an approximate SVD decomposition. The rsvd-PI algorithm allows you to speed up SVD calculations, especially for large matrices.

Input: $\mathbf{A} \in \mathbb{R}^{m \times n}$, rank parameter k, power parameter p
Output: $\mathbf{U} \in \mathbb{R}^{m \times k}$, $\mathbf{S} \in \mathbb{R}^{k \times k}$, $\mathbf{V} \in \mathbb{R}^{k \times n}$

$\mathbf{\Omega} = \mathrm{randn}(n, k+s)$
$\mathbf{Q} = \mathbf{A\Omega}$
for $i = 0, 1, 2, 3, \cdots, p$ **do**
  if $i < p$ **then** $[\mathbf{Q}, \sim] = \mathrm{lu}(\mathbf{Q})$
  else $[\mathbf{Q}, \sim, \sim] = \mathrm{eigSVD}(\mathbf{Q})$ **break**
  $\mathbf{Q} = \mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{Q})$
end for
$\mathbf{B} = \mathbf{Q}^{\mathrm{T}}\mathbf{A}$
$[\mathbf{V}, \mathbf{S}, \mathbf{U}] = \mathrm{eigSVD}(\mathbf{B}^{\mathrm{T}})$
$ind = s+1 : k+s$
$\mathbf{U} = \mathbf{QU}(:, ind)$, $\mathbf{S} = \mathbf{S}(ind, ind)$, $\mathbf{V} = \mathbf{V}(:, ind)$.

rSVD-BKI

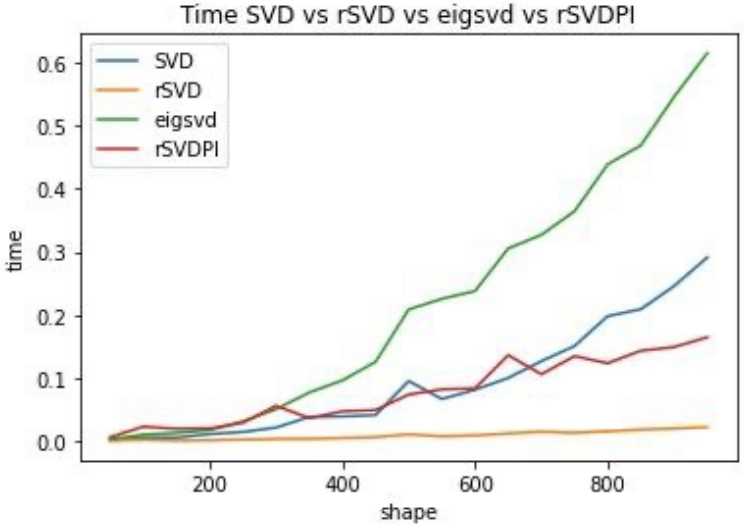
The rSVD-BKI algorithm (Randomized Singular Value Decomposition with Block Krylov Iterations) is also used to quickly calculate large-size SVD matrices. It is based on a combination of random projection and block Krylov methods. Unlike rSVD-PI, rSVD-BKI works with blocks of matrices, which allows you to take into account the structure of the matrix and use it for more efficient calculations. The rSVD-BKI algorithm also allows you to speed up SVD calculations, especially for matrices with a block structure.

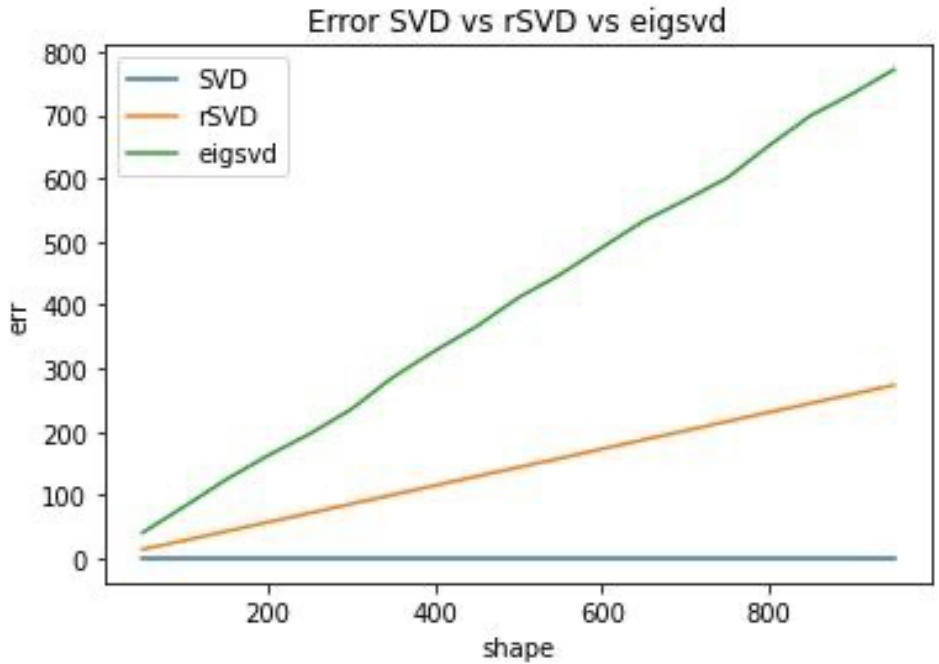
Input: $\mathbf{A} \in \mathbb{R}^{m \times n}$, rank parameter k, power parameter p
Output: $\mathbf{U} \in \mathbb{R}^{m \times k}$, $\mathbf{S} \in \mathbb{R}^{k \times k}$, $\mathbf{V} \in \mathbb{R}^{k \times n}$

$\mathbf{\Omega} = \mathrm{randn}(n, k+s)$
$[\mathbf{H}_0, \sim] = \mathrm{lu}(\mathbf{A\Omega})$
for $i = 1, 2, 3, \cdots, p$ **do**
  if $i < p$ **then** $[\mathbf{H}_i, \sim] = \mathrm{lu}(\mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{H}_{i-1}))$
end for
$\mathbf{H} = [\mathbf{H}_0, \mathbf{H}_1, ..., \mathbf{H}_p]$
$\mathbf{Q} = \mathrm{orth}(\mathbf{H})$
$\mathbf{B} = \mathbf{Q}^{\mathrm{T}}\mathbf{A}$
$[\mathbf{V}, \mathbf{S}, \mathbf{U}] = \mathrm{eigSVD}(\mathbf{B}^{\mathrm{T}})$
$ind = (k+s)(p+1) - k + 1 : (k+s)(p+1)$
$\mathbf{U} = \mathbf{QU}(:, ind)$, $\mathbf{S} = \mathbf{S}(ind, ind)$, $\mathbf{V} = \mathbf{V}(:, ind)$.

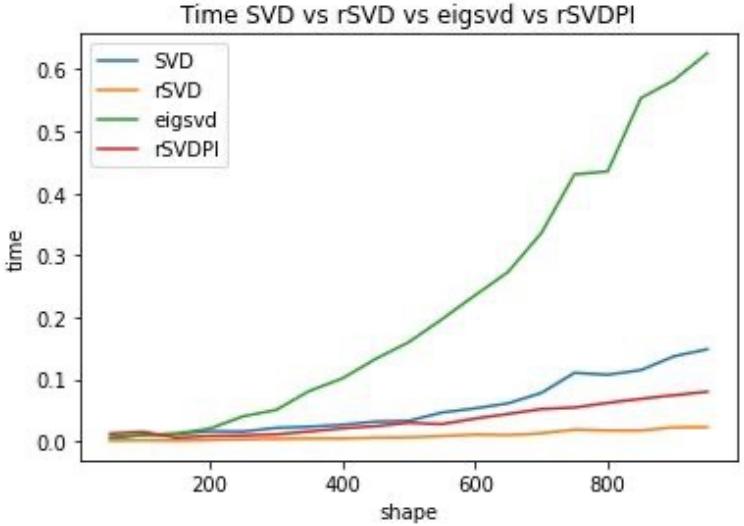
Results overview

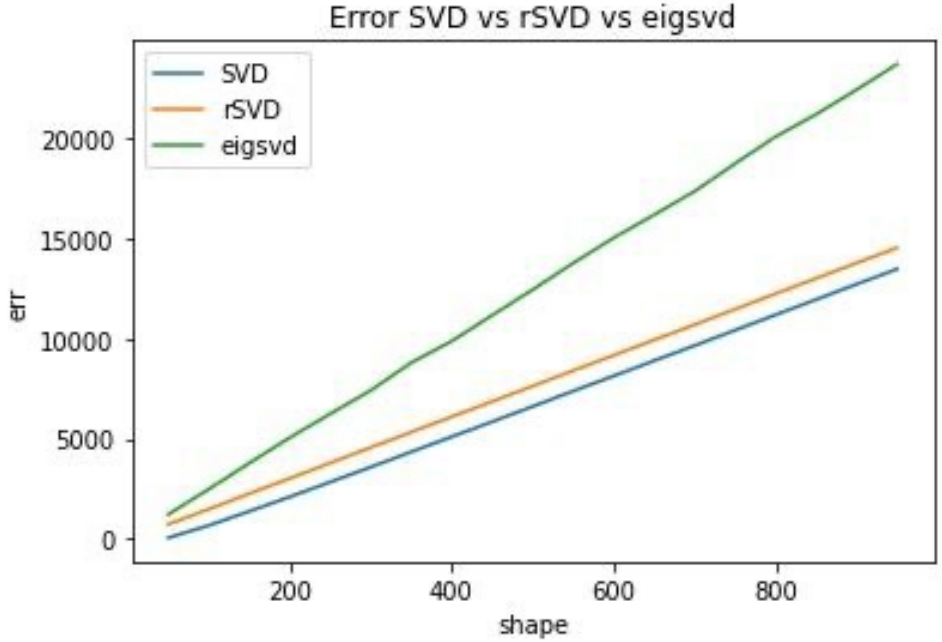
Dense matrices





Sparse matrices





For what is it?

Extra presentation is attached!:)

Summary, conclusion, and plans

- **We have implemented several different randomized algorithms for processing both dense and sparse matrices.**
- **An investigation was conducted to demonstrate the acceleration achieved by our implemented algorithms based on the matrix size and to assess the associated errors.**
- **Despite our plans to implement the fast SVT algorithm, due to time constraints, we were unable to complete this endeavor.**
- **An intriguing avenue for future work would involve applying these developed algorithms to real-world matrices, extending their applicability beyond artificially generated data.**

Thank you for the listening!