Efficient Spectrum-Revealing Sparse CUR Decomposition

Onyebuchi Ekenta

University of California, Berkeley

November 3, 2021

Problem Statement

- Given a sparse $m \times n$ matrix A find a rank k matrix $\tilde{A}$, with $k \ll m, n$ that approximates A.
- Approaches that treat the matrices as dense can require vast amounts of computation power [GDR+16]
- **Goal**: Exploit the sparsity of the matrix to compute the factorizations with minimal computational resources.

Outline

- Overview of Previous Low Rank Decomposition Methods
- Introduction to CUR
- Maximum Volume Principle
- Description of Our Method
- Experiments

SVD

$$A = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

Eckart-Young Theorem: If A_k is the sum of the first k terms of the SVD then

$$\|A - A_k\|_2 = \min_{rank(B) \leq k} \|A - B\|_2 = \sigma_{k+1}(A)$$

Also minimizes Frobenius norm.

Drawbacks

- Can be too expensive to compute especially for large sparse matrices
- Can be difficult to interpret.
  - Fix: Approximate A using actual columns and rows

Rank-Revealing QR

- More computationally efficient than the SVD
- Approximates A in terms of actual columns of A
- Still not well-suited for sparse matrices

Rank-Revealing QR

The approximation will satisfy [XGL17]

$$\left\| \mathbf{A} - \hat{\mathbf{A}}_k \right\|_2 \leq \sigma_{k+1}(A) \sqrt{1 + \tau^2 \left(\frac{\sigma_{\ell+1}(A)}{\sigma_{k+1}(A)} \right)^2}.$$

For matrices with rapidly decaying spectrum, can be just as good as SVD.

CUR Decomposition

Description

Choose c columns of A and r rows of A to form matrices C and R. Construct a $c \times r$ inner matrix U to make the approximation,

$$A \approx CUR$$

Advantages

- Preserves sparsity of the input matrix
- Better interpretability

Maximum Volume Principle

Let M be the intersection of C and R. $\text{vol}(M) = \text{abs}(\det M)$.
Maximum Volume Principle: Maximizing the volume of M tends to lead to good approximations.

Randomized CUR

Many randomized CUR algorithms manage to achieve a $(1+\epsilon)$-relative error [KS16]

$$\|A - CUR\|_F \leq (1+\epsilon)\|A - A_k\|_F$$

Drawback: These approaches require sampling $\ell = O(k/\epsilon)$ rows and columns to achieve this error which may not be practical.

Method Outline

Our method works in three phases

1. Compute rank a $\ell > k$ truncated LU factorization to obtain $\mathbf{C}$ and $\mathbf{R}$.
2. Apply *Spectrum Revealing Pivoting* to improve the quality of the factorization and guarantee approximation bounds
3. Apply *StableCUR* to compute final rank k approximation.

Key Benefits

1. Scales to very large matrices
2. Gap-dependent error bounds on Frobenius and Spectral Norms
3. Spectral Revealing Properties
4. Only requires sampling $\ell = k + O(1)$ columns / rows for matrices with rapidly decaying spectrum.

Truncated LU Decompostion

We approximate the matrix via truncated LU decomposition

$$\begin{aligned}
\mathbf{A}(\mathbf{p},\mathbf{q}) &= \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \\
&= \begin{pmatrix} \mathbf{L}_{11} & \\ \mathbf{L}_{21} & I \end{pmatrix} \begin{pmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ & \mathbf{S} \end{pmatrix} \\
&\approx \begin{pmatrix} \mathbf{L}_{11} \\ \mathbf{L}_{21} \end{pmatrix} \begin{pmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \end{pmatrix}
\end{aligned}$$

Produces $\mathbf{C} = \begin{pmatrix} \mathbf{A}_{11} \\ \mathbf{A}_{21} \end{pmatrix}$, $\mathbf{R} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \end{pmatrix}$

LUSOL

Package for maintaining and updating sparse LU factors of a square or rectangular matrix.

Features

- Uses Markowitz Pivoting to maintain sparsity
- Threshold Complete Pivoting and Threshold Rook Pivoting
- Routines to Update the LU Factorization

Threshold Complete Pivoting

- Often yields good rank k approximations even without any further processing from SRP.
- Doesn't have theoretical guarantees on its own
- Fails for very large matrices that are too dense.

Threshold Rook Pivoting

- Typically only takes a second or two, even for very large matrices.
- Rarely gives good approximations on its own so requires SRP
- SRP does "most of the work"

Spectrum Revealing Pivoting Algorithm

- Swap out some of the ℓ selected rows and columns to improve the quality of the factorization.
- Maintains a low-dimensional (e.g. 20) projection, $\mathbf{S}' = \Omega\mathbf{S}$ of the Schur complement to detect promising rows and columns

Estimating Element with Largest Magnitude

Let $\mathbf{R} = \Omega\mathbf{A}$ be a random projection

1. Identify the column c of $\mathbf{R}$ of maximum norm.
2. Compute a_c, column c of A
3. Return m, the maximum element of a_c

Spectrum Revealing Pivoting

One Iteration:

- Estimate the largest element $\alpha = \mathbf{S}(i, j)$ of $\mathbf{S}$
- Create $\overline{\mathbf{A}}_{11}$ by adding row $\ell + i$ and column $\ell + j$ to $\mathbf{A}_{11}$
- Estimate the largest element $\beta = \overline{\mathbf{A}}_{11}^{-T}(a, b)$ of $\overline{\mathbf{A}}_{11}^{-T}$
- Swap rows $a, k + i$ and columns $b, k + j$
- **Theorem** : $\det(\mathbf{A}'_{11}) = \alpha\beta \det(\mathbf{A}_{11})$

Updating ΩS

- Computation of the projected Schur complement ΩS is an expensive operation.
- Swapping a row and column induces a rank 2 update on $\mathbf{S}$ which can be used to update ΩS efficiently.

Updating ΩS

The Schur complement is updated via

$$S \leftarrow S + \begin{bmatrix} \mathbf{s}_c & \mathbf{v}_c \end{bmatrix} \begin{bmatrix} -\alpha^{-1} & 0 \\ 0 & \beta^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{s}_r^T \\ \mathbf{v}_r^T \end{bmatrix}$$

- $\mathbf{s}_c$ and $\mathbf{s}_r^T$ are the j-th column and i-th row of the Schur complement. α, β are as defined before.
- Don't know a simple interpretation of $\mathbf{v}_c$ and $\mathbf{v}_r^T$

Spectrum Revealing Pivoting

Key Benefits

- Comes with proveable guarantees on factorization accuracy
- Scales better for larger / denser matrices than TCP.

Drawbacks

- TCP will be faster for smaller / sparser matrices

Stable CUR

Algorithm 1: StableCUR

Input: A matrix $A \in \mathbb{R}^{m \times n}$, $\mathbf{R} \in \mathbb{R}^{\ell \times n}$ $\mathbf{C} \in \mathbb{R}^{m \times \ell} \in$

Output: $\tilde{\mathbf{A}}_k$

1 Do *QR* factorization on $\mathbf{R}^T$ to obtain a basis of rows of $\mathbf{R}, \mathbf{R} = \mathbf{R}_r \mathbf{Q}_r$

2 Do *QR* factorization on $\mathbf{C}$ to obtain a basis of columns of $\mathbf{C}, \mathbf{C} = \mathbf{Q}_c \mathbf{R}_c$

3 $\mathbf{B} = \mathbf{Q}_c^T \mathbf{A} \mathbf{Q}_r^T$

4 Do *SVD* on $\mathbf{B}$ to Compute $\mathbf{B}_k$

5 $\widetilde{\mathbf{A}}_\mathbf{k} = \mathbf{Q}_c \mathbf{B}_k \mathbf{Q}_r$

CUR Error Bounds

Rank Revealing QR Decomposition [XGL17]

$$\left\|\mathbf{A}-\widehat{\mathbf{A}}_k\right\|_2 \leq \sigma_{k+1}(A)\sqrt{1+\tau^2\frac{\sigma_{\ell+1}(A)^2}{\sigma_{k+1}(A)^2}}.$$

Our Method

$$\left\|\mathbf{A}-\widehat{\mathbf{A}}_k\right\|_2 \leq \sigma_{k+1}(A)\sqrt{1+2\gamma^2\frac{\sigma^2_{\ell+1}(\mathbf{A})}{\sigma^2_{k+1}(\mathbf{A})}}$$

For $\gamma = O(f\ell\sqrt{mn})$[CGZ+20]

CUR Error Bounds

Rank Revealing QR Decomposition [XGL17]

$$\sigma_j(\hat{\mathbf{A}}_k) \geq \frac{\sigma_k(A)}{\sqrt{1+\tau^2\left(\frac{\sigma_{k+1}(A)}{\sigma_j(A)}\right)^2}}$$

Our Method

$$\sigma_j(\mathbf{A}) \geq \sigma_j(\hat{\mathbf{A}}_k) \geq \sigma_j(\mathbf{A})\sqrt{1-2\gamma^2\left(\frac{\sigma_{\ell+1}(\mathbf{A})}{\sigma_j(\mathbf{A})}\right)^2}$$

Experiments

- We compare SRP, TCP, RRQR, and Matlabs svds algorithm.
- Experiments were run on Berkeley's Savio computer with maxNumCompThreads set to 4.
- We compute the relative error $\frac{\|A-\hat{A}_k\|_F}{\|A\|_F}$ of the factorizations.

Small Matrices: 3000-5000 rows

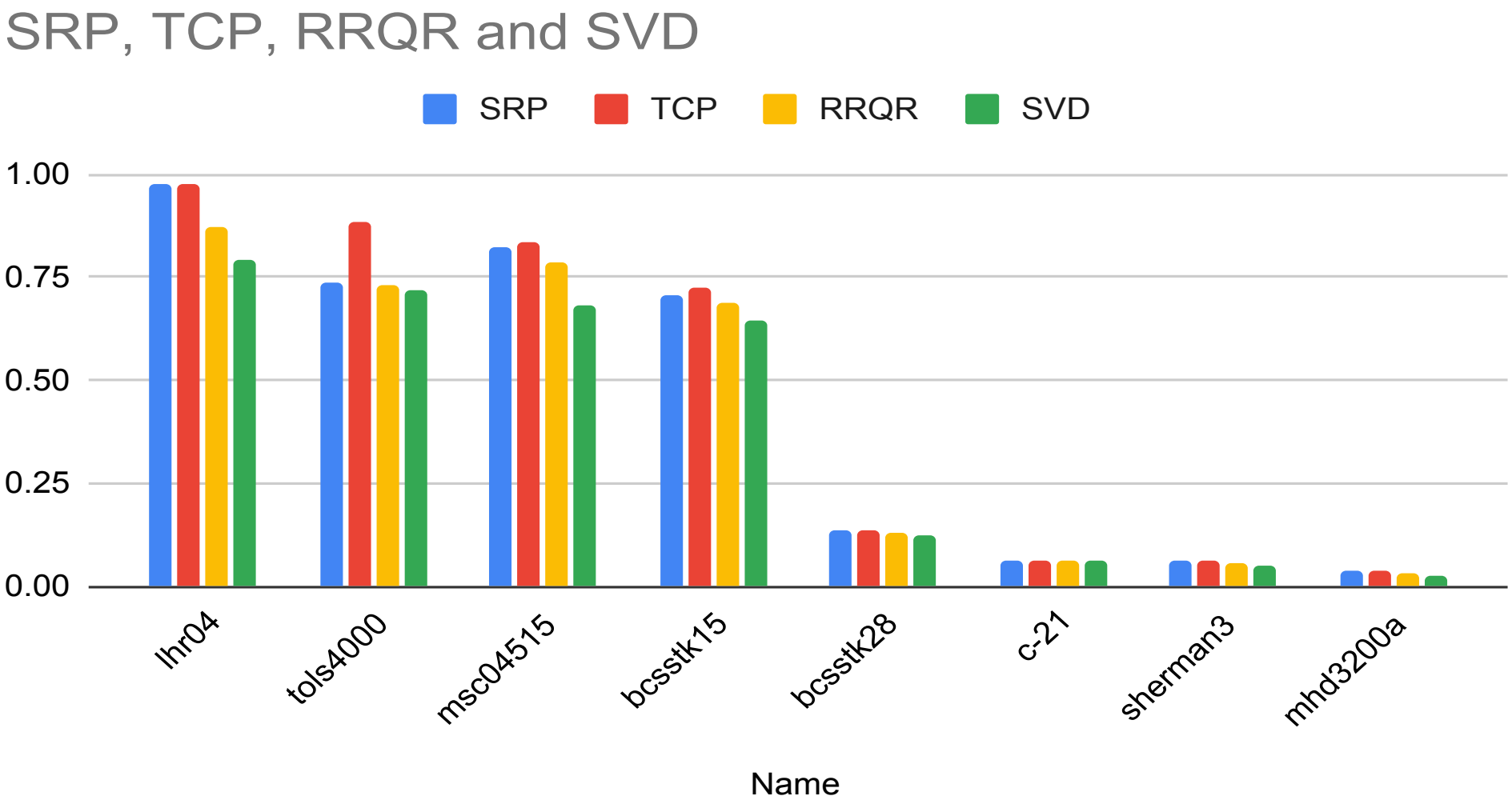
$\ell = k = 100$.

Factored with SRP and TCP.

TCP required no extra swaps from SRP for these matrices.

Name	SRP Swaps
bcsstk15	96
bcsstk28	80
c-21	96
lhr04	0
mhd3200a	99
msc04515	96
sherman3	100
tols4000	100

Small Matrices

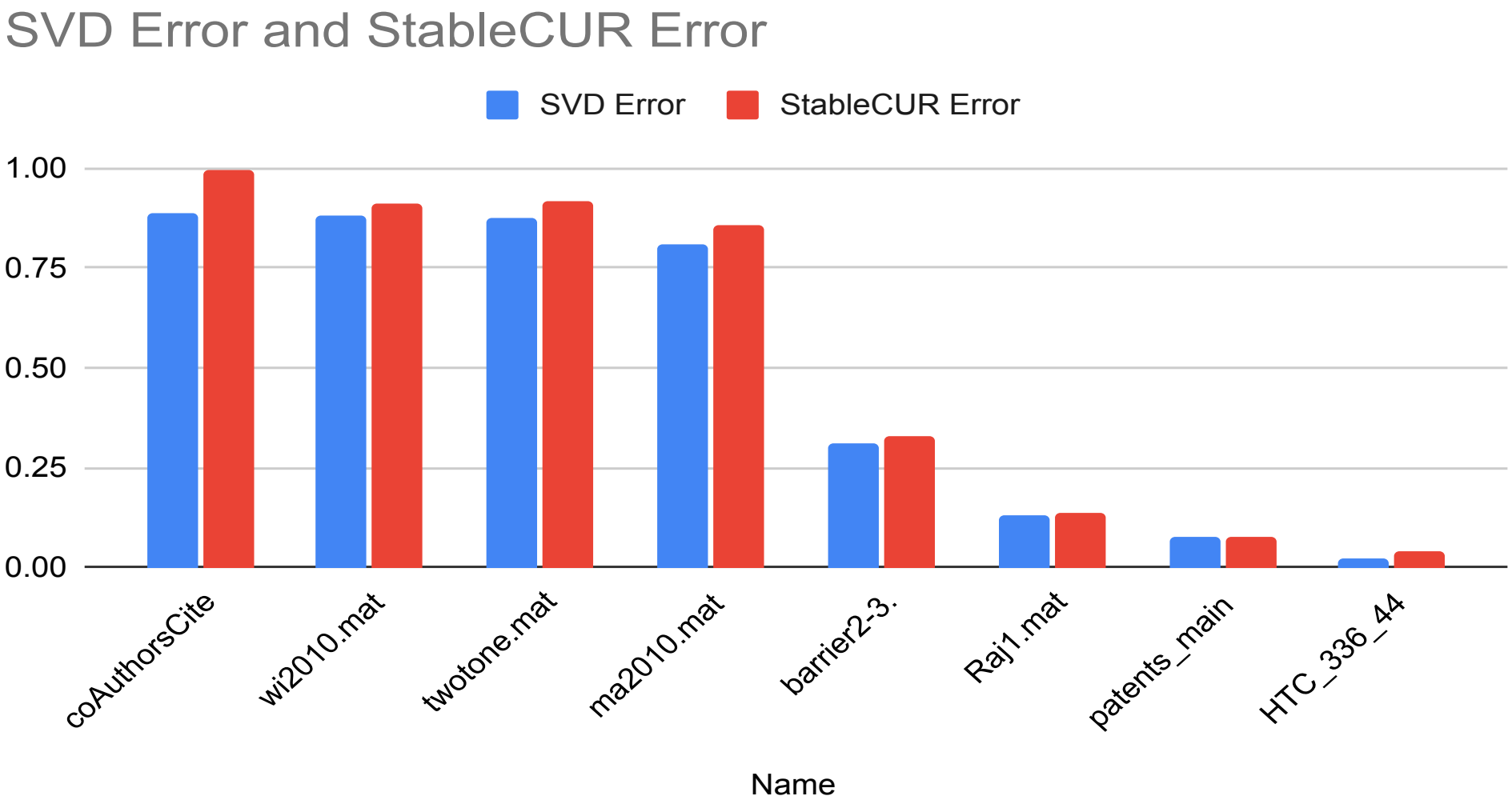


Large Matrices: 100,000 - 300,000 rows

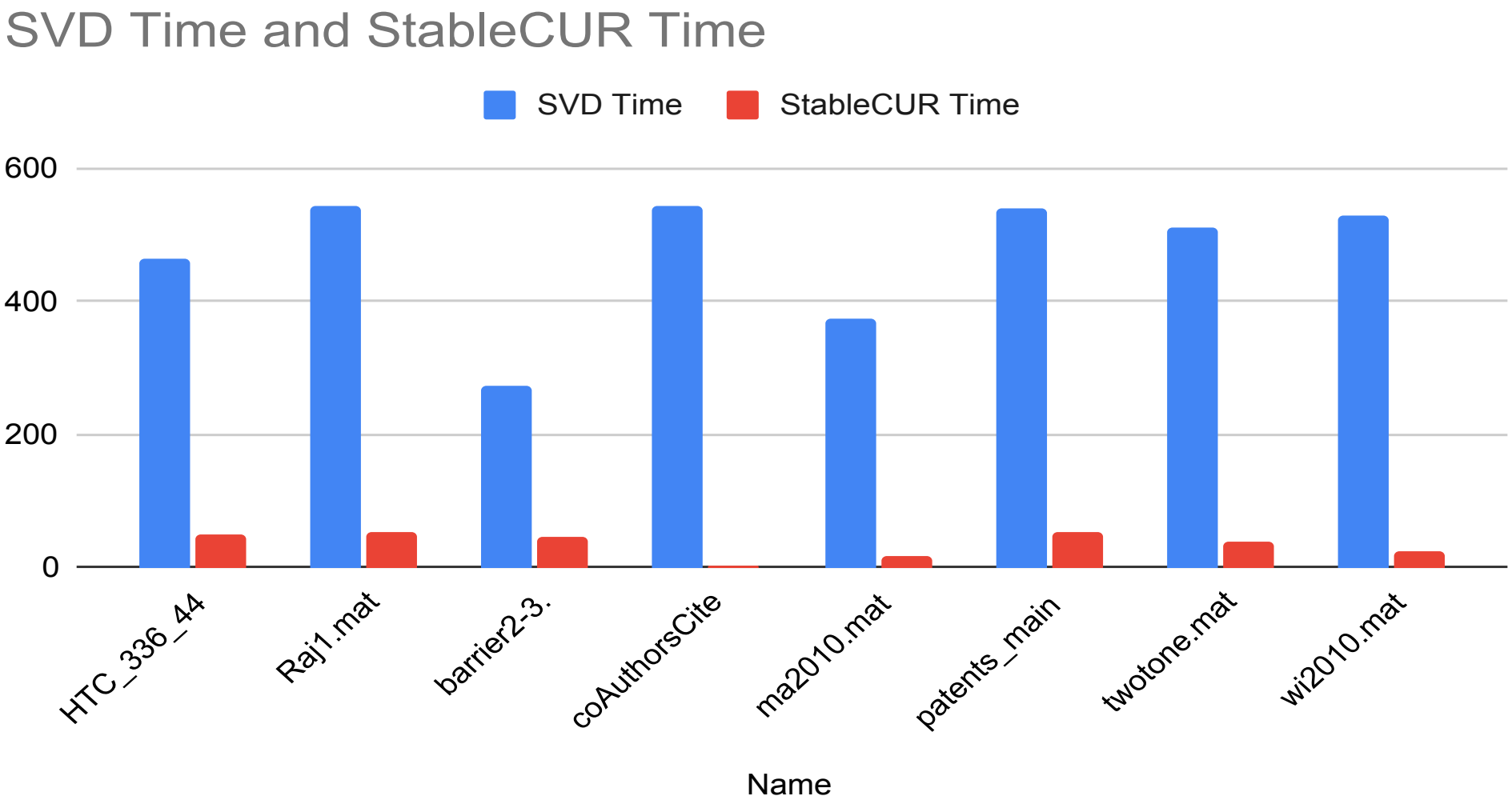
StableCUR with $\ell = 550, k = 500$.

Name	SRP Swaps
HTC_336_4438.mat	550
Raj1.mat	541
barrier2-3.mat	550
coAuthorsCiteseer.mat	0
ma2010.mat	214
patents_main.mat	549
twotone.mat	550
wi2010.mat	212

Large Matrices



Large Matrices



Conclusion

- We present a novel CUR decomposition scheme that allows for efficient low rank estimation of sparse matrices.
- Our method is fast and efficient, able to quickly deliver high quality factorizations in a low resource environment.

[CGZ+20] Cheng Chen, Ming Gu, Zhihua Zhang, Weinan Zhang, and Yong Yu.
Efficient spectrum-revealing cur matrix decomposition.
volume 108 of *Proceedings of Machine Learning Research*, pages 766–775. PMLR, 26–28 Aug 2020.

[GDR+16] Alex Gittens, Aditya Devarakonda, Evan Racah, Michael F. Ringenburg, Lisa Gerhardt, Jey Kottalam, Jialin Liu, Kristyn J. Maschhoff, Shane Canon, Jatin Chhugani, Pramod Sharma, Jiyan Yang, James Demmel, Jim Harrell, Venkat Krishnamurthy, Michael W. Mahoney, and Prabhat.
Matrix factorization at scale: a comparison of scientific data analytics in spark and C+MPI using three case studies.
CoRR, abs/1607.01335, 2016.

[KS16] N. Kishore Kumar and Jan Shneider.
Literature survey on low rank approximation of matrices, 2016.

[XGL17] Jianwei Xiao, Ming Gu, and Julien Langou.
Fast parallel randomized qr with column pivoting algorithms for reliable low-rank matrix approximations.
2017 IEEE 24th International Conference on High Performance Computing (HiPC), Dec 2017.