Title

Spectrum-Revealing CUR Decomposition

Author

Ekenta, Onyebuchi

Publication Date

2022

Peer reviewed|Thesis/dissertation

Spectrum-Revealing CUR Decomposition

by

Onyebuchi Ekenta

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Mathematics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ming Gu, Chair
Professor James Demmel
Professor Katherine Yelick

Summer 2022

Spectrum-Revealing CUR Decomposition

Abstract

Spectrum-Revealing CUR Decomposition

by

Onyebuchi Ekenta

Doctor of Philosophy in Mathematics

University of California, Berkeley

Professor Ming Gu, Chair

The CUR decomposition is a popular tool for computing a low rank factorization of a matrix in terms of a small number of columns and rows of the matrix. CUR decompositions are favored in some use-cases because they have a higher degree of interpretability and are able to preserve the sparsity of the input matrix. Previous random sampling-based approaches are able to construct CUR decompositions with relative-error bounds with high probability. However, these methods are costly to run on practical datasets. We implement a novel algorithm to compute CUR approximations of sparse matrices. Our method comes with relative error bounds for matrices with rapidly decaying spectrum and runs in time that is nearly linear in m and n.

Introduction

Data analysis is essential to making scientific progress in the modern world. Scientists often rely on data collected through massive experiments involving hundreds of sensors or produced through large simulations to gain insights the structure and behavior of the systems they study. These datasets can reach enormous sizes. Experiments from the LHC can produce petabytes worth of data. The ITER project, the world's largest nuclear fusion project, is expected to produce two petabytes of data every day by the year 2035. Analyzing such data sets often necessitates the use of considerable computing resources. Access to and usage of such computing platforms is a limiting factor for scientific programs conducted across a wide range of disciplines. Having access to more scalable and efficient procedures for data analysis provides researchers with greater flexibility by allowing them to analyze their experiments faster and with more easily accessible computational resources.

Low rank approximation is a common technique in data analysis for reducing the noise and understanding the relationship between data variables. A common approach to producing low rank approximations is computing a truncated singular value decomposition. While this method provides the most accurate approximations, it has drawbacks that make it unsuitable for certain applications. The resulting singular value vectors will typically be dense even for sparse input matrices which can result in excessive costs in storage and processing time. Also, it is sometimes desirable to interpret the singular value vectors as though they were instances of the data set (e.g. eigengenes for gene-based data). But this interpretation becomes difficult when using the SVD as the singular value vectors do not conserve important properties of the matrix such as nonnegativity and sparsity.

The CUR matrix decomposition is an alternative approach that is better suited for handling these issues. CUR decomposition computes a low rank approximation of the form $\mathbf{A} \approx \mathbf{CUR}$ where the matrices $\mathbf{C}$ and $\mathbf{R}$ consist of columns and rows of $\mathbf{A}$. By approximating the matrix in terms of actual columns and rows it both preserves the sparsity of the original matrix and allows for the natural interpretation as instances of the data set. Because of these properties, CUR decompositions have become popular in the data science community where they have been applied to problems such as feature selection, clustering and graph mining. [26, 21, 5]

Some random-sampling based methods come with relative-error guarantees [10, 30, 6]. That is to say, with high probability the CUR decomposition will satisfy

$$\|\mathbf{A} - \mathbf{CUR}\|_F^2 < (1+\epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2$$

where $\mathbf{A}_k$ is the optimal rank k approximation. However these methods require computing sampling $O(k/\epsilon)$ columns and rows to achieve their error guarantees which is often impractical. Some of these methods rely on quantities that are expensive to compute such as singular value vectors and leverage scores. Computing these quantities for large-scale datasets of interest can require the use of large parallel or distributed platforms. [13]

In this paper we introduce the Spectrum-Revealing CUR decomposition method (SR-CUR). This algorithm allows for the computation of relative-error CUR decompositions of

matrices with a rapidly decaying spectrum. Importantly, for sparse matrices, the algorithm runs in time that is nearly linear in m and n, making it feasible to compute large factorizations with modest computing resources. We run experiments verifying the speed and accuracy of our factorizations. Our code is available on Github [11].

Related Work

CUR decomposition

A CUR decomposition approximates a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ as the product of a matrix $\mathbf{C} \in \mathbb{R}^{m \times c}$ consisting of a collection of columns of $\mathbf{A}$, $\mathbf{R} \in \mathbb{R}^{r \times n}$ consisting of a collection of rows of $\mathbf{A}$ and an inner mixing matrix $\mathbf{U}$. The goal is to find a choice of columns and rows which minimizes the factorization error $\|\mathbf{A} - \mathbf{CUR}\|$.

A variety of different approaches have been taken to selecting the columns and rows for the factorization. One approach is to employ deterministic pivoting strategies to make the selections [25, 7, 17, 3]. Another useful approach is to employ the maximum volume principle, meaning one seeks to select the columns and rows that maximizes the absolute determinant of the matrix formed at the intersection of C and R. This principle was employed to develop a CUR decomposition method known as pseudoskeleton approximation [16, 15, 14]. In [27] a similar principle known as simplex volume maximization is used.

In [9] the authors propose a linear time CUR decomposition algorithm. A collection of c columns and r rows are randomly sampled to construct the C and R matrices which are in turn used to compute U. For a given target rank k, by sampling $c = O(\log(1/\delta)\epsilon^{-4})$ columns and $r = O(k\delta^{-2}\epsilon^{-2})$ rows the CUR decomposition will satisfy

$$\|A - CUR\|_2 \leq \|A - A_k\|_2 + \epsilon\|A\|_2$$

with probability $1 - \delta$. Alternatively by sampling $c = O(k \log(1/\delta)\epsilon^{-4})$ columns and $r = O(k\delta^{-2}\epsilon^{-2})$ rows the above will hold for the Frobenius norm with probability $1 - \delta$.

This method was improved in [10] to yield a relative error factorization method. Here C and R are randomly sampled and U is set to be the Moore-Penrose pseudoinverse of their intersection. The sampling probabilities used here depend on the right-singular vectors of A. In [21] a random-sampling based CUR decomposition method was presented which employed statistical leverage scores to construct the sample distribution. In [6] and also in [30] relative-error CUR decompositions running in linear time and requiring $O(k/\epsilon)$ row and column samples are presented.

The RandomizedSVD algorithm introduced in [22, 1] can approximate the singular vectors in $O(mn \log k)$ time. In [28, 29] factorization methods were proposed achieving expected relative error and improved performance compared to existing methods. See [19] for more information.

Rank Revealing Factorizations

Rank-revealing algorithms [7] are low rank matrix approximation algorithms that accurately capture the rank of a given matrix. One such example is the rank revealing QR factorization. Given an $m \times n$ matrix A this produces a factorization of the form

$$AP = QR = Q \begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}$$

where $Q \in \mathbb{R}^{m \times n}$ is orthonormal, $R_{11} \in \mathbb{R}^{k \times k}$, $R_{12} \in \mathbb{R}^{k \times n-k}$ and $R_{22} \in \mathbb{R}^{n-k \times n-k}$. The factorization is called rank-revealing if

$$\frac{\sigma_k(A)}{p(k,n)} \leq \sigma_{min}(R_{11}) \leq \sigma_k(A)$$
$$\sigma_{k+1}(A) \leq \sigma_{\max}(R_{22}) \leq p(k,n)\sigma_{k+1}(A)$$

where $p(k,n)$ is a low-degree polynomial in k and n. A low rank factorization can be obtained from RRQR by neglecting R_{22}. The resulting low rank factorization $\tilde{A}$ satisfies

$$\|A - \tilde{A}\|_2 \leq p(k,n)\sigma_{k+1}(A)$$

. A table of achievable values for $p(k,n)$ can be found in [4]. Rank-revealing methods need not capture the full spectrum of the of the matrix. Mirian and Gu introduced a new low rank approximation scheme that approximated the full spectrum of the matrix. [23]

Randomized Sketching

Sketching is a technique where a large problem is replaced by a much smaller which can inform the solution of the original problem. A good example of how randomization can help improve algorithms in numerical linear algebra can be observed in the problem of linear regression. Consider an $m \times n$ matrix $\mathbf{A}$ with $n \ll m$. Since the system is overdetermined the problem $A\mathbf{x} = \mathbf{b}$ cannot be solved exactly so we seek to solve the minimization problem $\min_{\mathbf{x}} \|\mathbf{Ax} - \mathbf{b}\|$. This problem can be greatly reduced in size by applying a matrix $\mathbf{\Omega}$ that is randomly sampled from some distribution to produce the much smaller problem problem $\min_{\mathbf{x}} \|\mathbf{\Omega Ax} - \mathbf{\Omega b}\|$. In some cases, solving this smaller system yields an approximate solution to the original problem.

The sampling matrix $\mathbf{\Omega}$ can be constructed in a variety of different, the simplest being forming it out of independent Gaussian distributions. One can also construct sparse linear maps or maps that have specialized internal structure. [2]

In addition to least-squares problems, randomized techniques have also gained popularity for the problem of low-rank approximation, where certain methods improve upon their deterministic counterparts. [31, 20, 18].

Our Approach

Overview

Given an $m \times n$ matrix $\mathbf{A}$ our algorithm begins by computing a truncated LU factorization of the matrix. More precisely, for a choice of rank ℓ, we compute a factorization of the form

$$\mathbf{PAQ} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} = \begin{pmatrix} \mathbf{L}_{11} & \\ \mathbf{L}_{21} & \mathbf{I}_{n-k} \end{pmatrix} \begin{pmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ & \mathbf{S} \end{pmatrix}$$

where the submatrix $\mathbf{A}_{11}$ is $\ell \times \ell$. The factorizations are computed using the LUSOL software package [12]. LUSOL provides implementations of threshold complete pivoting (TCP) and threshold rook pivoting (TRP) pivot strategies which use Markowitz pivoting to maintain sparsity of the matrix. Either can be used to produce the initial factorization. This LU decomposition provides us with our initial selection of columns and rows, namely $\mathbf{C} = \begin{pmatrix} \mathbf{A}_{11} \\ \mathbf{A}_{21} \end{pmatrix}$ and $\mathbf{R} = \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \end{pmatrix}$. We also retain the Schur complement $\mathbf{S}$ which will be used in the Spectrum-Revealing Pivoting procedure described later.

After the initial factorization the $\mathbf{L}_{21}$ and $\mathbf{U}_{12}$ matrices are discarded and we are left storing only the factorization $\mathbf{A}_{11} = \mathbf{L}_{11}\mathbf{U}_{11}$. Following this, we run the Spectrum-Revealing Pivoting (SRP) procedure to improve the quality of the factorization. The final CUR decomposition will be computed with the StableCUR procedure.

Estimating Element with the Largest Magnitude

Computing the maximum entry of a matrix can be expensive if we only have indirect access to the matrix, such as through a factorization $\mathbf{A} = \mathbf{BC}$. We implement a method to estimate the maximum value in such cases at a reduced cost. We begin by producing a random matrix sketch via $\mathbf{R} = \mathbf{\Omega A}$ where $\mathbf{\Omega}$ is a $p \times m$ random matrix whose entries are sampled independently from the standard normal distribution. EELM then determines the column of $\mathbf{R}$ of maximum norm and returns the maximum element in the corresponding column of $\mathbf{A}$. When p is small computing the p matrix vector products to compute $\mathbf{\Omega A}$ can be more efficient than computing all entries of $\mathbf{A}$.

Using EELM we are able to quickly identify large elements of the Schur complement which can be used to improve the quality of the factorization. In addition to this, if the value returned by EELM is small we can bound the error of the factorization with high probability. The quality of the estimated maximum is given by the following theorem

Theorem 1 *For $p = \Theta\left(\log(n/\delta)/\epsilon^{-2}\right)$ the estimated value x given by Algorithm 1 satisfies the inequality $x \geq \sqrt{\frac{1-\varepsilon}{mn(1+\varepsilon)}}\|\mathbf{A}\|_F$ with probability at least $1-\delta$.*

Algorithm 1: Estimating Element with Largest Magnitude (EELM)

Input: A matrix $A \in \mathbb{R}^{m \times n}$ and its projection $\mathbf{R} \in \mathbb{R}^{p \times n}$

Output: r, c, m - row index, column index, and value of the largest element

1 $c = \arg\max_{i<j<n} \|\mathbf{R}(:, j)\|_2$

2 Compute the c-th column of $\mathbf{A}$

3 $r = \arg\max_{i \le j \le m} |\mathbf{A}(j, c)|$

4 $m = \mathbf{A}(r, c)$

In particular, we only require $p = O(\log n)$ to yield good approximations with high probability. In practice p is set to some reasonable constant (e.g. 20).

Spectrum Revealing Pivoting

In the SRP procedure we identify rows and columns that can be swapped with the rows and columns chosen during the truncated LU facotorization procedure to improve the quality of the factorization. We apply the maximum volume principle as a heuristic to judge the quality of a selection of rows and columns. Thus, we perform a sequence of row and column swaps with the goal of maximizing $|\det \mathbf{A}_{11}|$. Each swap will transform $\mathbf{A}_{11}$ into a new $\ell \times \ell$ submatrix $\mathbf{A}'_{11}$ whose volume is at least a factor $f > 1$ greater than the volume of $\mathbf{A}_{11}$, where f is a tolerance parameter. We repeat this process until no appropriate swap can be found.

We determine the swap in two phases. First we extend $\mathbf{A}_{11}$ into a $(\ell+1) \times (\ell+1)$ matrix $\overline{\mathbf{A}}_{11}$ by adding a new row $\ell+i$ and column $\ell+j$ to the matrix $\mathbf{A}_{11}$, forming a $(\ell+1) \times (\ell+1)$ matrix $\overline{\mathbf{A}}_{11}$. Then we choose a row $i' \in \{1, \ldots, \ell, \ell+i\}$ and column $j' \in \{1, \ldots, \ell, \ell+i\}$ we wish to remove from $\overline{\mathbf{A}}_{11}$.

Having identified the relevant rows and columns we swap row $\ell+i$ with i' and column $\ell+j$ with column j', thus transforming $\mathbf{A}_{11}$ into a new $\ell \times \ell$ matrix $\mathbf{A}'_{11}$. Note that if either $i' = \ell+i$ or $j' = \ell+j$ then the corresponding swap need not be performed. The following theorem helps guide the selection of the rows and columns.

Theorem 2 *Let $\alpha = \mathbf{S}(i, j)$ and $\beta = \overline{\mathbf{A}}_{11}^{-T}(i', j')$. Then we have,*

$$\left| \frac{\det \mathbf{A}'_{11}}{\det \mathbf{A}_{11}} \right| = |\alpha\beta|$$

.

Thus we have that if $|\alpha\beta| > f$ we know the volume of $\mathbf{A}_{11}$ will increase by at least a factor of f. Ideally, to select the appropriate rows of and columns we'd want to identify the maximum elements of $\mathbf{S}$ and $\overline{\mathbf{A}}_{11}^{-T}$ so as to maximize α and β. However, computing the exact maximum would be too computationally expensive, so we apply EELM to estimate them. To do this SRP must keep track of matrix sketches of $\mathbf{A}_{11}^{-T}$ and $\mathbf{S}$.

Algorithm 2: Spectrum Revealing Pivoting

Input: f - tolerance parameter
$\tilde{\mathbf{A}}_{11}^{-T}$ - $q \times (\ell + 1)$ Sketch of $\overline{\mathbf{A}}_{11}^{-T}$
$\tilde{\mathbf{S}}$ - $p \times n$ Sketch of Schur Complement

```
while True do
    (i, j, α) ← EELM(S, S̃)
    Form Ā_11 = A([1 : k, k + i], [1 : k, k + j])
    (i′, j′, β) ← EELM(A_11^{-T}, Ã_11^{-T})
    if |αβ| < f then
    end
    Swap rows k + i and i′ and columns k + j and j′ and update the LU
     factorization.
    Update S̃
    Recompute random sketch Ã_11^{-T}
end
```

After identifying the swap to be performed the LU factorization and the matrix sketches must be updated to reflect the swap. This means we must update the LU factorization of $\mathbf{A}_{11}$ and the matrix sketches of $\mathbf{S}$ and $\overline{\mathbf{A}}_{11}^{-T}$. The row and column swaps can be implemented with LUSOL's column replacement and row replacement routines. Since $\overline{\mathbf{A}}_{11}^{-T}$ is small it is cheap to simply recompute a new random projection each step. For the Schur complement we iteratively update the projection at each step. Let $\mathbf{\Omega}$ be the random $p \times m - \ell$ matrix used to compute the projection of $\mathbf{S}$. In each iteration we compute the projection of the new Schur complement $\mathbf{\Omega S}'$ as an update to the previous projection $\mathbf{\Omega S}$. Details are provided in the next section.

Updating Sketched Schur Complement

Rank One Update In the cases where only a single swap is performed the swap corresponds to a rank one update of the matrix. That is we have that

$$\mathbf{A}' = \mathbf{A} + \mathbf{v}\mathbf{w}^T$$

for the appropriate choice of $\mathbf{v}$ and $\mathbf{w}$. For example if we are swapping columns j_1 and j_2 then $\mathbf{v} = \mathbf{A}(:, j_2) - \mathbf{A}(:, j_1)$ and $w = \mathbf{e}_{j_1} - \mathbf{e}_{j_2}$ where $\mathbf{e}_k$ is the k-th standard basis vector of $\mathbb{R}^n$. The following theorem allows us describe the update to the Schur complement when the matrix goes under a rank one update.

Theorem 3 *Let* $\mathbf{B} = \mathbf{A} + \mathbf{v}\mathbf{w}^T$. *We partition* $\mathbf{B}$, $\mathbf{v}$ *and* $\mathbf{w}$ *according to the partion for* $\mathbf{A}$. *Let* $\mathbf{S}' = \mathbf{B}_{22} - \mathbf{B}_{21}\mathbf{B}_{11}^{-1}\mathbf{B}_{12}$. *Then we have* $\mathbf{S}' = \mathbf{S} + \mathbf{E}$ *where*

$$\begin{aligned}\mathbf{E} &= \frac{1}{d}\hat{\mathbf{v}}\hat{\mathbf{w}}^T \\ \hat{\mathbf{v}} &= \left(\mathbf{v}_2 - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{v}_1\right) \\ \hat{\mathbf{w}}^T &= \left(\mathbf{w}_2^T - \mathbf{w}_1^T\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\right) \\ d &= \left(1 + \mathbf{w}_1^T\mathbf{A}_{11}^{-1}\mathbf{v}_1\right)\end{aligned}$$

With this, the corresponding update to the sketched Schur complement can be computed via

$$\mathbf{\Omega S}' = \mathbf{\Omega S} + \frac{1}{d}\mathbf{\Omega}\hat{\mathbf{v}}\hat{\mathbf{w}}^T$$

It takes $O(\ell^2)$ steps to compute d. Computing $\hat{\mathbf{v}}$ can be done in $O(m\ell)$ time and computing $\hat{\mathbf{w}}$ takes $O(n\ell)$ time. Computing the quantity $\frac{1}{d}\mathbf{\Omega}\hat{\mathbf{v}}\hat{\mathbf{w}}^T$ and adding it to $\mathbf{\Omega S}$ requires an additional $O(mp+np)$ steps. So the update to the Schur complement takes $O((\ell+p)(m+n))$ time to compute.

Rank Two Update Now we consider the case where rows $\ell+i$ and i' and columns $\ell+j$ and j' are swapped. First we must define some vectors. Let $\mathbf{a}_{21}^T$ and $\mathbf{a}_{12}$ be i-th row of $\mathbf{A}_{21}$ and the j-th column of $\mathbf{A}_{12}$ respectively. Let $\mathbf{s}_c$ and $\mathbf{s}_r^T$ be the the i-th column and the j-th row of the Schur complement. Let $\mathbf{e}_{i'}$ be the i'-th standard basis vector in $\mathbb{R}^\ell$ and $\overline{\mathbf{e}}_i$ be the i-th standard basis vector of $\mathbb{R}^{m-\ell}$. Let $\mathbf{e}_{j'}^T$ be the j'-th standard basis (row) vector of $\mathbb{R}^\ell$ and and $\overline{\mathbf{e}}_j^T$ be he j-th standard (row) vector of $\mathbb{R}^{n-\ell}$. Finally, let $\alpha = \mathbf{S}(i,j)$ and $\beta = \overline{\mathbf{A}}_{11}^{-T}(i',j')$, where $\overline{\mathbf{A}}_{11}$ is as defined previously. Then we have the following theorem characterizing the update to $\mathbf{S}$.

Theorem 4 *After swapping rows $\ell+i$ and i' and columns $\ell+j$ and j' the new Schur complement will be given by*

$$\mathbf{S}' = \mathbf{S} - \frac{1}{\alpha}\mathbf{s}_c\mathbf{s}_r^T + \frac{1}{\beta}\mathbf{v}_c\mathbf{v}_r^T$$

, where $\mathbf{v}_c$ and $\mathbf{v}_r$ are given by

$$\begin{aligned}\mathbf{v}_c &= \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{e}_{i'} + \overline{\mathbf{e}}_i + \frac{\mathbf{a}_{21}^T\mathbf{A}_{11}^{-1}\mathbf{e}_{i'}}{\alpha}\mathbf{s}_c \\ \mathbf{v}_r^T &= \mathbf{e}_{j'}^T\mathbf{A}_{11}^{-1}\mathbf{A}_{12} + \overline{\mathbf{e}}_j^T + \frac{\mathbf{e}_{j'}^T\mathbf{A}_{11}^{-1}\mathbf{a}_{12}}{\alpha}\mathbf{s}_r^T\end{aligned}$$

Given this result we can update the sketch of the Schur complement via

$$\mathbf{\Omega S}' = \mathbf{\Omega S} - \frac{1}{\alpha}\mathbf{\Omega}\mathbf{s}_c\mathbf{s}_r^T + \frac{1}{\beta}\mathbf{\Omega}\mathbf{v}_c\mathbf{v}_r^T$$

Algorithm 3: 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 $\widetilde{\mathbf{A}}_\mathbf{k} = \mathbf{Q}_c\mathbf{B}_k\mathbf{Q}_r$

.

Computing the vectors $\mathbf{s}_c$ and $\mathbf{v}_c$ can be done in $O(m\ell)$ time and computing $\mathbf{s}_r$ and $\mathbf{v}_r$ can be done in $O(n\ell)$ time. The corresponding update to the sketched Schur complement will require an additional $O(mp+np)$ steps to compute. The total time necessary to update the Schur complement is therefore $O((\ell+p)(m+n))$.

Spectrum-Revealing CUR

The end result of SRLU gives a selection of columns and rows to use for the CUR decomposition. The final CUR decomposition is given by the StableCUR procedure described in Algorithm 3 with the above choice of $\mathbf{C}$ and $\mathbf{R}$ as input. Since this method requires the computation of QR factorizations, it comes with an increased memory cost.

Theoretical Analysis

Time Complexity

LU Decomposition

For dense matrices the truncated LU decomposition would take $O(\ell mn)$ time. For sparse matrices the time varies depending on the size and degree of sparsity of the matrix.

The choice of TCP and TRP affects the overall run time of the algorithm. When threshold complete pivoting is used, typically the follow-up spectrum-revealing pivoting steps performs very few or no swaps at all. As a result TCP will typically be the faster choice for smaller matrices. However if the matrix is too large or too dense TCP can become too expensive to compute and so TRP will be the better choice.

When TRP is used, the spectrum-revealing pivoting tends to perform many swaps and the initial factorization will typically only represent a small fraction of the total computation time. Thus the total computation time will typically be dominated by the SRP and StableCUR procedures.

Spectrum-Revealing Pivoting

Running EELM for the Schur complement requires $O(pn)$ time. Recomputing the sketch $\tilde{\mathbf{A}}_{11}$ and estimating the maximum of $\mathbf{A}_{11}^{-T}$ requires $O(q\ell^2)$ time. The LUSOL routines to update the factorization require $O(\ell^2)$ steps. The update of the projected Schur complement requires $O((p+\ell)(m+n))$ steps. Thus the total time for an iteration of one iteration of SRP takes $O(q\ell^2 + (p+\ell)(m+n))$ time.

With a probability of .99 SRP will require at most $O(\ell \log(mn))$ iterations [8]. Thus the total complexity for spectrum-revealing pivoting is $O(\ell \log(mn)(q\ell^2+(p+\ell)(m+n))$. EELM procedure yields good approximations for $p = O(\log n)$ and $q = O(\log \ell)$. Thus the total time complexity of the Spectrum-Revealing Pivoting method is $O\left(\ell \log(mn)(\ell^2 \log \ell + (\log n + \ell)(m+n))\right)$.

StableCUR

The QR factorization of $\mathbf{R}^T$ and $\mathbf{C}$ requires $O(m\ell^2)$ time and $O(n\ell^2)$ time respectively to compute. For dense matrices $\mathbf{B} = \mathbf{Q}_c^T \mathbf{A} \mathbf{Q}_r^T$ would require $O(mn\ell)$ time. But since $\mathbf{A}$ is sparse significant savings are made. If A is sparse, the matrix-vector product $A\mathbf{v}$ can be computed in time $O(\text{nnz}(A))$. Computing $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{Q}_r^T$ can be done in $O(\text{nnz}(A)\ell)$ time. Then, computing $\mathbf{Q}_c^T \tilde{\mathbf{A}}$ requires an additional $O(m\ell^2)$ steps to compute.

Error Bounds

Spectrum Revealing Pivoting

Theorem 5 *For $j \leq k$ and $\gamma = O(fk\sqrt{mn})$, SRP produces a rank k SRLU factorization with*

$$\left\| \Pi_1 \mathbf{A} \Pi_2^T - \hat{\mathbf{L}}\hat{\mathbf{U}} \right\|_2 \leq \gamma \sigma_{k+1}(\mathbf{A})$$

$$\left\| \Pi_1 \mathbf{A} \Pi_2^T - (\widehat{\mathbf{L}}\widehat{\mathbf{U}})_j \right\|_2 \leq \sigma_{j+1}(\mathbf{A}) \left(1 + 2\gamma \frac{\sigma_{k+1}(\mathbf{A})}{\sigma_{j+1}(\mathbf{A})}\right)$$

SR-CUR

Theorem 6 *For $\gamma = O(fk\sqrt{mn})$ the SR-CUR decomposition satisfies*

$$\|\mathbf{A} - \widehat{\mathbf{A}}\|_2 \leq \|\mathbf{A} - \widehat{\mathbf{A}}\|_F \leq \gamma \sigma_{\ell+1}(\mathbf{A}),$$

$$\left\|\mathbf{A} - \widehat{\mathbf{A}}_k\right\|_F^2 \leq \left(1 + \frac{2\gamma^2 \sigma_{\ell+1}^2(\mathbf{A})}{\sum_{i=k+1}^{\text{rank}(\mathbf{A})} \sigma_i^2(\mathbf{A})}\right) \|\mathbf{A} - \mathbf{A}_k\|_F^2,$$

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

with probability .98

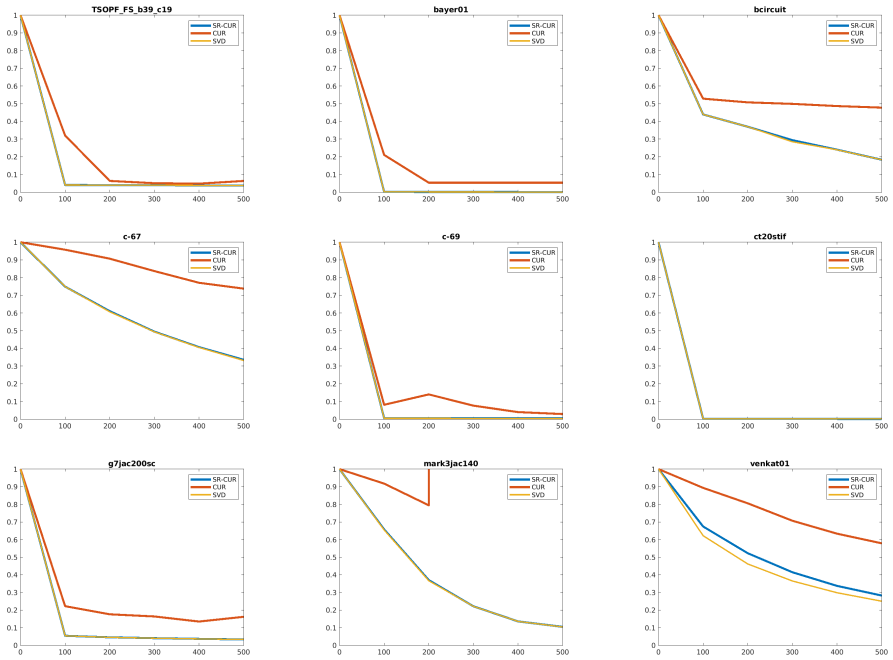

Figure 0.1: Comparison of SVD, SR-CUR and L2-norm sampling CUR

Experiments

SR-CUR

We provide an experiment demonstrating the capacity of SR-CUR to compute nearly optimal CUR decompositions. Drawing from the SuiteSparse Matrix Collection we collect a dataset consisting of matrices exhibiting rapidly decaying spectrum.

We compute factorizations for $k = 100, 200, 300, 400, 500$. We use TRP for the initial factorization and set $f = 2$. For each k we set $\ell = k + 50$ to apply the Stable-CUR algorithm. The SVD computations were computed with Matlab's `svds` function for $k = 500$. We compute the relative Frobenius error $\|A - A_k\|_F / \|A\|_F$ for each factorization. We compare the results against SVD and the L2-norm based random sampling method described in [9] with implementation provided by [24]. The results are shown in Figure 0.1. The time spent on the $k = 500$ factorization is shown in Figure 0.2. To simulate a low resource environment the maximum number of MATLAB threads was set to 4 during these computations.

Conclusion

We provide a high quality implementation of a the SR-CUR decomposition algorithm. We see that for matrices with rapidly decaying spectrum SR-CUR is capable of achieving relative-

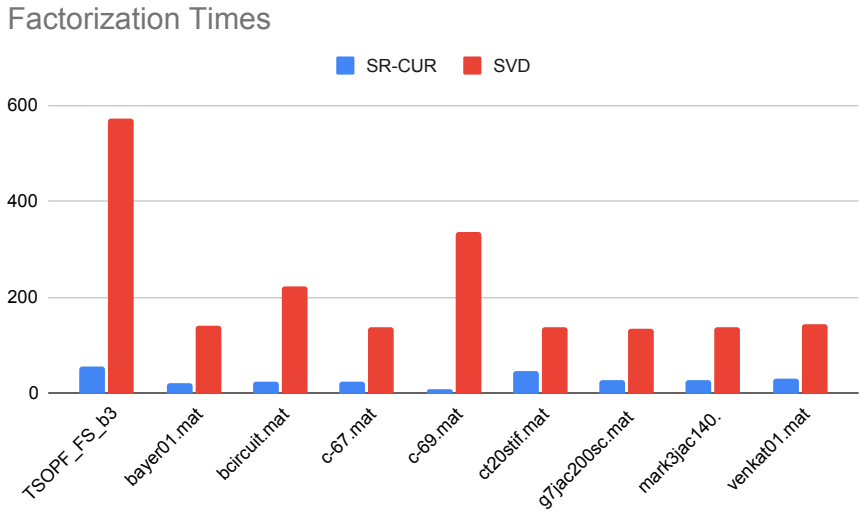

Figure 0.2: Computation time in seconds for Rank 500 Factorizations

error bounds. For sparse matrices, our approach comes with improved complexity bounds that allow it to scale easily to large matrices without the use of parallelization. Our experiments verify that our approach offers substantial benefits over alternative methods.

Bibliography

[1] "A randomized algorithm for the decomposition of matrices". In: *Applied and Computational Harmonic Analysis* 30.1 (2011), pp. 47–68. ISSN: 1063-5203. DOI: https://doi.org/10.1016/j.acha.2010.02.003. URL: https://www.sciencedirect.com/science/article/pii/S1063520310000242.

[2] Nir Ailon and Bernard Chazelle. "The Fast Johnson–Lindenstrauss Transform and Approximate Nearest Neighbors". In: *SIAM Journal on Computing* 39.1 (2009), pp. 302–322. DOI: 10.1137/060673096. eprint: https://doi.org/10.1137/060673096. URL: https://doi.org/10.1137/060673096.

[3] Michael W. Berry, Shakhina A. Pulatova, and G. W. Stewart. "Algorithm 844: Computing Sparse Reduced-Rank Approximations to Sparse Matrices". In: *ACM Trans. Math. Softw.* 31.2 (June 2005), pp. 252–269. ISSN: 0098-3500. DOI: 10.1145/1067967.1067972. URL: https://doi.org/10.1145/1067967.1067972.

[4] Christos Boutsidis, Michael Mahoney, and Petros Drineas. "On selecting exactly k columns from a matrix". In: (Jan. 2008).

[5] Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. "Unsupervised Feature Selection for Principal Components Analysis". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '08. Las Vegas, Nevada, USA: Association for Computing Machinery, 2008, pp. 61–69. ISBN: 9781605581934. DOI: 10.1145/1401890.1401903. URL: https://doi.org/10.1145/1401890.1401903.

[6] Christos Boutsidis and David P. Woodruff. "Optimal CUR Matrix Decompositions". In: *SIAM Journal on Computing* 46.2 (2017), pp. 543–589. DOI: 10.1137/140977898. eprint: https://doi.org/10.1137/140977898. URL: https://doi.org/10.1137/140977898.

[7] Tony F. Chan. "Rank revealing QR factorizations". In: *Linear Algebra and its Applications* 88-89 (1987), pp. 67–82. ISSN: 0024-3795. DOI: https://doi.org/10.1016/0024-3795(87)90103-0. URL: https://www.sciencedirect.com/science/article/pii/0024379587901030.

[8] Cheng Chen et al. "Efficient Spectrum-Revealing CUR Matrix Decomposition". In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, 26–28 Aug 2020, pp. 766–775. URL: https://proceedings.mlr.press/v108/chen20a.html.

[9] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. "Fast Monte Carlo Algorithms for Matrices III: Computing a Compressed Approximate Matrix Decomposition". In: *SIAM Journal on Computing* 36.1 (2006), pp. 184–206. DOI: 10.1137/S0097539704442702. eprint: https://doi.org/10.1137/S0097539704442702. URL: https://doi.org/10.1137/S0097539704442702.

[10] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. "Relative-Error CUR Matrix Decompositions". In: *SIAM Journal on Matrix Analysis and Applications* 30.2 (2008), pp. 844–881. DOI: 10.1137/07070471X. eprint: https://doi.org/10.1137/07070471X. URL: https://doi.org/10.1137/07070471X.

[11] Onyebuchi Ekenta. *sr-cur*. https://github.com/Rioghasarig/sr-cur. 2022.

[12] Philip E. Gill et al. "Maintaining LU factors of a general sparse matrix". In: *Linear Algebra and its Applications* 88-89 (1987), pp. 239–270. ISSN: 0024-3795. DOI: https://doi.org/10.1016/0024-3795(87)90112-1. URL: https://www.sciencedirect.com/science/article/pii/0024379587901121.

[13] Alex Gittens et al. "Matrix Factorization at Scale: a Comparison of Scientific Data Analytics in Spark and C+MPI Using Three Case Studies". In: *CoRR* abs/1607.01335 (2016). arXiv: 1607.01335. URL: http://arxiv.org/abs/1607.01335.

[14] S. Goreinov et al. "How to find a good submatrix". In: 2010.

[15] S.A. Goreinov, E.E. Tyrtyshnikov, and N.L. Zamarashkin. "A theory of pseudoskeleton approximations". In: *Linear Algebra and its Applications* 261.1 (1997), pp. 1–21. ISSN: 0024-3795. DOI: https://doi.org/10.1016/S0024-3795(96)00301-1. URL: https://www.sciencedirect.com/science/article/pii/S0024379596003011.

[16] Sergei Goreinov and E. Tyrtyshnikov. "The maximal-volume concept in approximation by low-rank matrices". In: *Contemporary Mathematics* 208 (Jan. 2001). DOI: 10.1090/conm/280/4620.

[17] Ming Gu and Stanley C. Eisenstat. "Efficient Algorithms for Computing a Strong Rank-Revealing QR Factorization". In: *SIAM Journal on Scientific Computing* 17.4 (1995), pp. 848–869. DOI: https://doi.org/10.1137/0917055.

[18] N. Halko, P. G. Martinsson, and J. A. Tropp. "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions". In: *SIAM Review* 53.2 (2011).

[19] N. Kishore Kumar and Jan Shneider. "Literature survey on low rank approximation of matrices". In: *ArXiv* abs/1606.06511 (2016).

[20] Michael W. Mahoney. "Randomized Algorithms for Matrices and Data". In: *Found. Trends Mach. Learn.* 3.2 (Feb. 2011), pp. 123–224. ISSN: 1935-8237. DOI: 10.1561/2200000035. URL: https://doi.org/10.1561/2200000035.

[21] Michael W. Mahoney and Petros Drineas. "CUR matrix decompositions for improved data analysis". In: *Proceedings of the National Academy of Sciences* 106.3 (2009), pp. 697–702. ISSN: 0027-8424. DOI: 10.1073/pnas.0803205106. eprint: https://www.pnas.org/content/106/3/697.full.pdf. URL: https://www.pnas.org/content/106/3/697.

[22] P.G. Martinsson, V. Rohklin, and M. Tygert. *A Randomized Algorithm for the Approximation of Matrices*. Tech. rep. 2006.

[23] L Miranian and M Gu. "Strong rank revealing LU factorizations". In: *Linear Algebra and its Applications* 367 (2003), pp. 1–16. ISSN: 0024-3795. DOI: https://doi.org/10.1016/S0024-3795(02)00572-4. URL: https://www.sciencedirect.com/science/article/pii/S0024379502005724.

[24] Christopher Schinnerl. *pymf3*. https://github.com/charlespwd/project-title. 2017.

[25] G.W. Stewart. "Four algorithms for the the efficient computation of truncated pivoted QR approximations to a sparse matrix". In: *Numerische Mathematikk* 83.2 (1999), pp. 313–323. DOI: https://doi.org/10.1007/s002110050451.

[26] Jimeng Sun et al. "Less is More: Compact Matrix Decomposition for Large Sparse Graphs (Best research paper award!)" In: *the 2007 SIAM International Conference on Data Mining (SDM), Minneapolis, MN*. Apr. 2007. URL: https://www.microsoft.com/en-us/research/publication/less-is-more-compact-matrix-decomposition-for-large-sparse-graphsbest-research-paper-award/.

[27] Christian Thurau, Kristian Kersting, and Christian Bauckhage. "Deterministic CUR for Improved Large-Scale Data Analysis: An Empirical Study". In: *Proceedings of the 2012 SIAM International Conference on Data Mining (SDM)*, pp. 684–695. DOI: 10.1137/1.9781611972825.59. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611972825.59. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611972825.59.

[28] Shusen Wang and Zhihua Zhang. "A Scalable CUR Matrix Decomposition Algorithm: Lower Time Complexity and Tighter Bound". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 647–655.

[29] Shusen Wang and Zhihua Zhang. "Improving CUR Matrix Decomposition and the Nyström Approximation via Adaptive Sampling". In: *J. Mach. Learn. Res.* 14.1 (Jan. 2013), pp. 2729–2769. ISSN: 1532-4435.

[30] Shusen Wang, Zhihua Zhang, and Tong Zhang. "Towards More Efficient SPSD Matrix Approximation and CUR Matrix Decomposition". In: *Journal of Machine Learning Research* 17.209 (2016), pp. 1–49. URL: http://jmlr.org/papers/v17/15-190.html.

[31] David P. Woodruff. "Sketching as a Tool for Numerical Linear Algebra". In: 10.1–2 (Oct. 2014), pp. 1–157. ISSN: 1551-305X. DOI: 10.1561/0400000060. URL: https://doi.org/10.1561/0400000060.