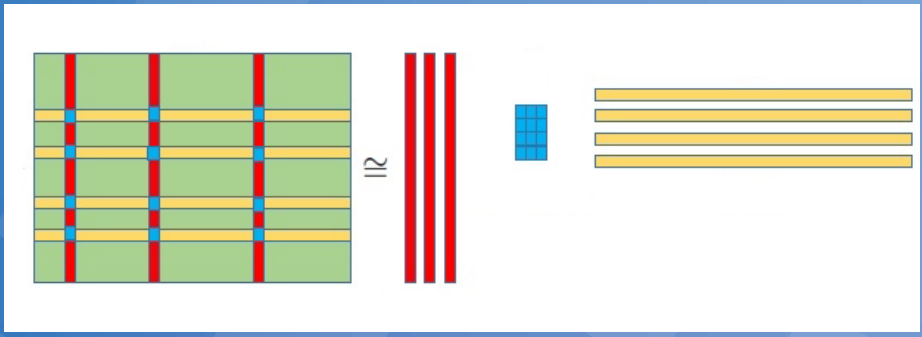
Generalized CUR type Decompositions for Improved Data Analysis

GENERALIZED CUR TYPE DECOMPOSITIONS FOR IMPROVED DATA ANALYSIS



PERFECT YAYRA GIDISU

GENERALIZED CUR TYPE DECOMPOSITIONS FOR IMPROVED DATA ANALYSIS

PERFECT YAYRA GIDISU

This work is financially supported by the European Union's Horizon 2020 Research and Innovation program through the Marie Skłodowska-Curie Grant Agreement No 812912 with the title "Big Data Challenges for Mathematics".

A catalogue record is available from the Eindhoven University of Technology Library

ISBN: 978-90-386-5832-2

Cover design by Nicholas Junior Agbebo, based on the picture from Salman Ahmadi-Asl.

Printed by: Leester Heide Grafisch

GENERALIZED CUR TYPE DECOMPOSITIONS FOR IMPROVED DATA ANALYSIS

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof. dr. S. K. Lenaerts, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op dinsdag 10 oktober 2023 om 16:00 uur

door

PERFECT YAYRA GIDISU

geboren te Accra, Ghana

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof. dr. ir. B. Koren
promotor:	dr. M. E. Hochstenbach
copromoter:	dr. A. Micheletti (University of Milan, Italy)
leden:	prof. dr. N. Gillis (Université de Mons, Belgium)
	prof. dr. M. B. Van Gijzen (TU Delft, Netherlands)
	prof. dr. K. P. L. Veroy-Grepl
adviseur(s):	dr. S. Massei (University of Pisa, Italy)
	dr. S. C. Hess

Het onderzoek of ontwerp dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening

CONTENTS

List of Algorithms

$\mathbb{R}$	set of real numbers
$\mathbb{N}_+$	set of positive natural numbers
$\mathbb{P}, \mathbb{S}$	oblique (interpolatory) projectors
I_m	identity matrix of dimension m
$A(:, j)$	jth column of A
$A(i, :)$	ith row of A
$A(:, \mathbf{p})$	submatrix of A with column indices in vector $\mathbf{p}$
$\|\cdot\|_F$	Frobenius norm, $\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^2}$, $A \in \mathbb{R}^{m \times n}$
$\|\cdot\|_\infty$	vector Inf-norm, $\|x\|_\infty = \max_{1 \le i \le n} \|x_i\|$, $x \in \mathbb{R}^n$
$\|\cdot\|$	vector 2-norm, $\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2}$, $x \in \mathbb{R}^n$
	matrix 2-norm, $\|A\| = \max_{x \in \mathbb{R}^n, \|x\|=1} \|Ax\|$, $A \in \mathbb{R}^{m \times n}$
$(\cdot)^\top$	transpose of a matrix, $(A_{ij})^\top = A_{ji}$
$(\cdot)^{-1}$	inverse of a matrix
$(\cdot)^+$	Moore–Penrose pseudoinverse of a matrix
$\mathsf{diag}(\cdot)$	diagonal or block diagonal matrix

Chapter-related notation

Chapters 2–5

U, V	contain the left and right singular vectors
Σ	contains the singular values of a matrix
$\sigma_i(\cdot)$	ith singular value of a matrix, in nonincreasing order

Chapter 6

W, Z	contain the left and right singular vectors
Ψ	contains the singular values of a matrix
$\psi_i(\cdot)$	ith singular value of a matrix, in nonincreasing order
U, V, X, Y	matrices from the generalized singular vector decomposition
Γ, Σ	contain elements of the generalized singular values of a matrix pair
$\frac{\gamma_i(\cdot)}{\sigma_i(\cdot)}$	ith generalized singular value of a matrix pair, in nonincreasing order

Chapter 7

$\psi_i(\cdot)$	ith singular value of a matrix, in nonincreasing order
W, Z, U, V	matrices from the restricted singular vector decomposition
$\Sigma(\sigma_i)$	correspond to elements of the generalized singular values
α, β, γ	correspond to elements of the restricted singular values
$\frac{\alpha_i(\cdot)}{\beta_i(\cdot)\gamma_i(\cdot)}$	ith restricted singular value of a matrix triplet, in nonincreasing order

1 INTRODUCTION

This thesis concerns one of the most fundamental topics in numerical linear algebra: the low-rank approximation of matrices. This chapter is a brief introduction to this problem.

With the proliferation of big data matrices, dimensionality reduction has become an important tool in many data analysis applications. Approximating a data matrix with one of lower rank is a widely used technique for unsupervised linear dimensionality reduction.

Given a data matrix $A \in \mathbb{R}^{m \times n}$ (without loss of generality, throughout the thesis unless otherwise stated, we assume $m \geq n$), we compute its low-rank approximation by finding rectangular factors U and V such that

$$\underset{m \times n}{A} \approx \underset{m \times k}{U} \underset{k \times n}{V^\top}, \tag{1.1}$$

for some target rank $k \ll \min(m, n)$. In many applications, the data matrix has a low numerical rank and can therefore be well approximated by a low-rank matrix [90]. Data matrices of low rank arise in the discretization of PDEs, machine learning applications, text document analysis, and genomics [11, 41, 68]. Having a low-rank representation of A is advantageous in terms of storage and computational efficiency. For example, multiplying a dense matrix A by a vector $\mathbf{x} \in \mathbb{R}^n$ costs $\mathcal{O}(mn)$ operations; for a rank-k factorization $A \approx UV^\top$, the cost of a matrix-vector multiplication reduces to $\mathcal{O}((m+n)k)$.

There are several methods of low-rank matrix approximation for a given problem; however, the approximated data often consist of derived features that are either no longer interpretable or difficult to interpret in the original context. The singular value decomposition (SVD) is a widely used method for optimally approximating and compressing data. However, it can be challenging for domain experts of the field from which the data are drawn to interpret the singular vectors that result from the decomposition because they are a linear combination of all the data features.

In some applications, it is important to find an approximation that preserves the original characteristics of the data, such as *sparsity*, *nonnegativity*, and *integer values*, while also being *interpretable*. One way to achieve this is through an

interpolative decomposition or a CUR decomposition. This decomposition has first been introduced by Goreinov et al. [52] as the pseudoskeleton decomposition. A CUR decomposition involves using a subset of the rows and columns of a matrix to approximate it. More specifically, in a CUR matrix factorization with a desired rank of k, the matrix $A \in \mathbb{R}^{m \times n}$ is represented by the matrix product $C \cdot U \cdot R$, where $C \in \mathbb{R}^{m \times k}$ and $R \in \mathbb{R}^{k \times n}$ contain a carefully selected subset of the columns and rows of A, respectively, and $U \in \mathbb{R}^{k \times k}$ is computed such that the low-rank approximation to A holds. Since CUR decompositions are constructed from actual data elements, the factors are easily interpretable by practitioners of the field from which the data are drawn, as long as the original data is understandable. Qualitatively, the selected subsets of rows and columns are deemed to capture the most relevant information hidden in the underlying matrix structure.

The original motivation for this research stems from a project of AcomeA SGR. AcomeA SGR is an asset management company that offers, besides traditional systems of investment, an online app (Gimme5) where single investors may invest any amount of money in a range of mutual funds that can be claimed at any time. The app is mostly used by small investors like students, young people, etc. as "piggy banks". The project aims to identify customers (using the app) who have a financial potential larger than their actual investments, to apply target marketing strategies. Financial potential measures the investment capacity of customers in this context, i.e., the amount of money an investor can invest consistently over a long period. The financial potential of an investor provides AcomeA with directions to focus its marketing efforts in the most cost-effective way possible. Rather than trying to reach every customer who could use Gimme5, focusing a marketing plan to fit a smaller and possibly, customers with unexploited financial potential can allow the company to carve out a niche for Gimme5. For this project, we leverage a CUR decomposition for customer profiling and targeted marketing.

The data matrix is structured as follows: each row corresponds to an investor, and each column represents a specific feature that characterizes the investor's investment behavior, socio-demographic background, geolocation data, and estimated financial potential. A CUR decomposition is employed to select a subset of the most relevant columns and rows. The selected rows are the subset of investors that exhibit characteristics or behaviors that are considered influential in understanding the financial potential and investment behavior. Each selected row corresponds to an investor profile, a summarized representation of their investment behaviors, estimated financial potential, and relevant attributes. These profiles provide a concise yet informative snapshot of each investor's characteristics, facilitating easy comparison and analysis. Additionally, the selected rows

are essential for segmenting investors based on shared characteristics. Clustering investors with similar profiles enables the creation of distinct segments, each representing a group of investors with comparable investment behaviors and inferred financial potential. The segments allow for the implementation of targeted marketing strategies tailored to the unique preferences and needs of each group.

The selected columns represent the most informative features that drive investor differentiation. They serve as the building blocks for creating investor profiles. Each column's values contribute to defining an investor's profile, capturing their behavior and characteristics. These columns collectively form a comprehensive representation of an investor's investment tendencies, financial potential, and relevant attributes. Furthermore, they play a crucial role in defining the criteria for segmenting investors. The shared attributes within these columns guide the formation of segments that exhibit similar investment behaviors and financial potential. The segments enable targeted marketing efforts that resonate with specific investor groups.

In summary, we use the selected rows and columns to enhance the understanding of investor behavior and financial potential, enabling us to develop more effective marketing strategies and refine engagement efforts for a diverse investor base.

A CUR decomposition possesses several compelling properties:

- Enhanced data interpretation: One of the notable advantages of a CUR decomposition lies in the identification of subsets of rows and columns that contribute significantly to the structure of the original matrix. These identified subsets often hold valuable insights for data interpretation. In scenarios where understanding the underlying patterns of the data is crucial like in the case of investor profiling, the information about selected rows and columns aids in clarifying the meaningful aspects of the data.

- Preservation of matrix properties: A CUR decomposition preserves important characteristics of the original matrix. This is particularly evident when the matrix possesses specific attributes, such as sparsity or nonnegativity. When the matrix is sparse or nonnegative, the resulting factors C and R also inherit these properties. This feature is crucial when dealing with data that has inherent characteristics, as it ensures that the approximation retains the same structural qualities.

- Memory and computational efficiency: A CUR decomposition stands out for its memory efficiency, making it a practical choice when dealing with large data sets. In particular, when the entries of the matrix itself are available or

can be inexpensively retrieved, then, once the important column and row indices are determined, there is no need to explicitly store the factors C and R. Additionally, the process of selecting the subsets of columns and rows in a CUR decomposition can often be achieved using efficient algorithms, e.g., randomized algorithms. This leads to faster computations compared to some other factorization methods like the SVD while achieving good approximation quality.

Column or row subset selection is a long-standing problem in numerical linear algebra that has recently seen significant advancements in theoretical and algorithmic developments in the fields of numerical linear algebra and theoretical computer science. This problem has a wide range of applications, including scientific computing, model reduction, and statistical data analysis.

The primary focus of the work presented in this thesis is to improve existing deterministic techniques and develop new tools for the analysis of a CUR factorization and its generalizations.

In the first part of this thesis (Chapters 2, 3, 4, and 5), we give a brief review of four different types of low-rank approximations: the singular value decomposition; a rank-revealing QR factorization; column subset selection, which aims at selecting k columns of A that well approximate the column range of A; and a CUR approximation, which uses k rows and columns of A to approximate A. We discuss deterministic algorithms for building low-rank approximations of a matrix using some rows or columns of the matrix itself. We provide a modified version and an extension of an existing CUR decomposition algorithm. We also propose a new column or row selection strategy that iteratively invokes existing column subset selection methods.

A CUR factorization is designed to handle one data matrix at a time. Various practical applications involve multiple data matrices, in which one is tasked with extracting the most discriminative information of one data set relative to others. Additionally, the multiple data sets may also provide information complementary to each other and one is interested in discovering the underlying structure shared by two different representations of a given data. In the second part of the thesis (Chapters 6 and 7), we generalize a CUR decomposition to the setting where we have multiple data sets.

ORGANIZATION OF THE THESIS

The thesis is divided into two parts corresponding to the two broad topics mentioned above. Each part contains a more detailed introductory chapter. Our

contributions are presented in Chapters 3, 4, 5, 6, and 7 which are based on the papers [42–46]. Finally, a conclusion that will summarize the contributions of the thesis, attempt to put them into perspective and give some directions for further research.

Part I

LOW-RANK MATRIX APPROXIMATION BASED ON COLUMNS AND ROWS

The problem of approximating a matrix with a lower-rank matrix plays a fundamental role in data analysis and is widely used for linear dimensionality reduction. The primary objective is to represent a given matrix as the product of two or three matrices with smaller dimensions, thereby constructing a low-rank matrix approximation. This process facilitates data compression, noise reduction, and efficient computation.

This part of the thesis aims to provide an overview of some existing techniques employed for low-rank matrix approximations. These methods encompass a range of approaches, such as the SVD, a pivoted QR factorization, and a CUR decomposition. Each of these techniques involves different mathematical formulations and algorithms, allowing for various trade-offs between accuracy, interpretability, and computational complexity.

We propose new algorithms that leverage a subset of the original matrix's rows and columns to construct a low-rank matrix approximation. By carefully selecting these subsets, we can capture the most influential features of the data while significantly reducing the dimensionality. This approach enhances the efficiency of the approximation process and enables the interpretation and analysis of large-scale data sets that would otherwise be computationally intractable.

Through numerical experiments, we demonstrate the effectiveness and efficiency of our proposed algorithms compared to existing techniques. We showcase their performance on diverse data sets, highlighting their advantages and possible limitations.

2

LOW-RANK MATRIX APPROXIMATIONS

In Chapter 1, we mentioned that the goal in many applications is not just to compute any factorization satisfying equation (1.1), but also to impose additional constraints on the factors U and V. In this chapter, we will describe different specific low-rank matrix decompositions that incorporate these constraints.

2.1 THE SINGULAR VALUE DECOMPOSITION

The SVD gives the best low-rank approximation of a matrix when the approximation error is in terms of Frobenius or spectral norm. Every matrix $A \in \mathbb{R}^{m\times n}$ (without loss of generality $m \geq n$) admits an economy-sized SVD of the form

$$A = U\Sigma V^\top, \tag{2.1}$$

where $U \in \mathbb{R}^{m\times n}$ and $V \in \mathbb{R}^{n\times n}$ are matrices with orthonornal columns, and $\Sigma = \mathsf{diag}(\sigma_1, \ldots, \sigma_n)$ is a diagonal matrix containing the nonincreasing singular values $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$. The orthonormal columns of U and V are the left and right singular vectors, respectively. We can find a truncated SVD (TSVD) of A by setting all but the first $k < \mathsf{rank}(A)$ largest singular values to zero and using only the first k columns of U and V.

Given a target rank k, the problem of minimizing $\|A - A_k\|$ for the spectral or Frobenius norm has a well-known solution given by the Eckart–Young–Mirsky theorem. Here, A_k is a rank-k approximation of A.

Theorem 2.1. *(Eckart–Young–Mirsky Theorem, see, e.g., [63]) Let U_k and V_k contain the first k left and right singular vectors, respectively, and Σ_k be the leading $k \times k$ submatrix of Σ. For $\xi = 2$ or F, it holds that*

$$\min_{\substack{A_k \in \mathbb{R}^{m\times n} \\ rank(A_k) \leq k}} \|A - A_k\|_\xi = \|A - U_k\Sigma_k V_k^\top\|_\xi.$$

The minimal error is achieved by the truncated SVD of A. The minimizer A_k is unique if and only if $\sigma_k \neq \sigma_{k+1}$.

In terms of Frobenius norm, the right-hand side of the above equation equals $\sqrt{\sigma_{k+1}^2 + \cdots + \sigma_n^2}$ and for spectral norm it equals σ_{k+1}.

2.2 RANK-REVEALING QR DECOMPOSITION

A rank-revealing QR (RRQR) factorization is a numerical technique used to compute the QR factorization of a matrix while also revealing its numerical rank and providing information about the importance of its columns. It is particularly valuable when dealing with ill-conditioned or nearly rank-deficient matrices, where standard QR factorization might not provide accurate rank information due to numerical errors.

The goal of an RRQR factorization is to identify a permutation of the columns of the original matrix A such that the resulting matrix T has a clear block structure with a well-defined upper triangular part and a lower triangular part close to zero. This reveals the numerical rank and helps identify the (nearly) linearly dependent columns. Given matrix $A \in \mathbb{R}^{m\times n}$ with $m \geq n$, consider a QR factorization of the form

$$\underset{m\times n}{A}\quad\underset{n\times n}{\Pi}\quad=\quad\underset{m\times n}{Q}\quad\underset{n\times n}{\begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix}},$$

where Q is a matrix with orthonormal columns, $T_{11} \in \mathbb{R}^{k\times k}$ is upper triangular with nonnegative diagonal entries, $T_{12} \in \mathbb{R}^{k\times(n-k)}$, $T_{22} \in \mathbb{R}^{(n-k)\times(n-k)}$, and Π is a permutation matrix chosen to reveal linear dependence among the columns of A. Usually k is selected such that $\|T_{22}\|$ is sufficiently small. By the interlacing property of singular values, for any permutation Π, it holds that

$$\sigma_{\min}(T_{11}) \leq \sigma_k(A) \quad \text{and} \quad \sigma_{\max}(T_{22}) \geq \sigma_{k+1}(A),$$

where $\sigma_{\min}$ and $\sigma_{\max}$ denote the smallest and largest singular values, respectively. Suppose A has a numerical rank k and $\sigma_{k+1} \ll \sigma_k$. The aim in an RRQR factorization is to find a Π such that $\sigma_{\min}(T_{11})$ is maximized and $\sigma_{\max}(T_{22})$ is minimized, i.e.,

$$\sigma_{\min}(T_{11}) \approx \sigma_k(A) \quad \text{and} \quad \sigma_{\max}(T_{22}) \approx \sigma_{k+1}(A).$$

A common method for computing a rank-revealing factorization is a QR factorization with column pivoting. The upper triangular matrix T of a column-pivoted QR factorization satisfies the condition [93, Subsection 2.2, p. 3]

$$|T_{kk}|^2 \geq \sum_{i=k}^{j} |T_{ij}|^2, \; j = k+1, \ldots, n, \quad k = 1, \ldots, n.$$

Given $\mathbf{p}$, a vector of indices, we can express Π as $I(:, \mathbf{p})$. Suppose we partition Q so that

$$A(:, \mathbf{p}) = [Q_1 \;\; Q_2] \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} = Q_1 [T_{11} \;\; T_{12}] + Q_2 [0 \;\; T_{22}], \tag{2.2}$$

where $Q_1 \in \mathbb{R}^{m \times k}$, $Q_2 \in \mathbb{R}^{m \times (n-k)}$, and $\widehat{A}_k := Q_1 [T_{11} \;\; T_{12}]$, we have

$$\|A\Pi - \widehat{A}_k\| \leq \|T_{22}\|$$

to be the error bound of a truncated column-pivoted QR decomposition of A. This implies that Q_1 is an approximation of the range of A and as long as $\|T_{22}\|$ is small, $A(:, \mathbf{p})$ can be approximated by $\widehat{A}_k$. A low-rank approximation of the matrix A can be derived by neglecting the submatrix T_{22} in a column-pivoted QR factorization of A.

To compute a pivoted QR factorization, one may use the column-pivoted Gram–Schmidt algorithm [48]. A QR factorization with column pivoting works well in practice; however, there are examples where it fails to produce a factorization that yields a small $\|T_{22}\|$ (see, e.g., [53]). To circumvent this, QR factorization with other pivoting choices to compute an RRQR factorization have been proposed [14, 15, 48, 53]. The computational complexities of these methods are slightly larger than the standard QR decomposition algorithm.

2.3 THE COLUMN SUBSET SELECTION PROBLEM AND CUR FACTORIZATION

The column subset selection problem is closely connected to a pivoted QR factorization [14, 15, 53, 93]. The aim is to find a subset of the columns of a real matrix, such that, the subset represents the entire matrix well and is far from being rank deficient. That is, determine an index set $\mathbf{p} \in \{1, \ldots, n\}$ containing k distinct entries such that the corresponding k columns $A(:, \mathbf{p})$ represent a good approximation of the column range of A. More precisely, the following is the definition of a column subset selection problem.

Problem 2.2. Given $A \in \mathbb{R}^{m \times n}$ and a positive integer k, select k columns of A to form $C \in \mathbb{R}^{m \times k}$ such that the residual $\|A - CC^+A\|_\xi$, for $\xi = 2$ or F, is minimized over all possible $\binom{n}{k}$ choices for C.

Selecting k out of n (with $k \ll n$) columns such that $\|A - CC^+A\|_\xi$ is minimized requires $\mathcal{O}(n^k)$ time. Finding an exact solution to this problem will be

prohibitively slow for large data sizes and becomes computationally intractable. Therefore, researchers have focused on computing an approximate solution to the problem.

Closely related to the column subset selection problem is a CUR decomposition, which gives a low-rank approximation explicitly expressed in terms of a small number of columns and rows of A. The CUR factorization problem has been widely studied in the literature [8, 10, 17, 31, 32, 51, 52, 72, 83, 93, 94]. One can obtain a CUR decomposition by using a column subset selection algorithm on A and on $A^\top$ to construct C and R, respectively. Existing CUR algorithms use a column subset selection procedure to form the matrix C and R. The methods for constructing a CUR decomposition can be categorized into two broad groups, i.e., randomized and deterministic algorithms. **In this thesis, we will restrict our discussions to *deterministic* algorithms and the selection of exactly k columns and rows**.

A rank-k CUR decomposition of an arbitrary matrix $A \in \mathbb{R}^{m\times n}$ is of the form (In line with [87], we will use the letter M rather than U for the middle matrix because we use the letter U in other decompositions)

$$A \approx CMR := AP \cdot M \cdot S^\top A. \tag{2.3}$$

Here, P is an $n \times k$ (where $k < \min(m, n)$) *index selection matrix* with some columns of the identity matrix I_n that selects certain columns of A. Similarly, S is an $m \times k$ matrix with columns of I_m that selects certain rows of A; so C is $m \times k$ and R is $k \times n$, both of rank k. We construct the $k \times k$ matrix M of full rank such that the decomposition is as close to A as possible. Given matrices C and R of full rank, a standard procedure to determine M is (see, e.g., [83, Sec. 2], where also an alternative is presented) by two consecutive least squares problems:

1: Solve the least squares problem $CX \approx A$ for $X \in \mathbb{R}^{k\times n}$ with solution $X = (C^\top C)^{-1}C^\top A$.
2: Solve the least squares problem $R^\top M^\top \approx X^\top$ for $M \in \mathbb{R}^{k\times k}$ with solution $M = XR^\top(RR^\top)^{-1}$.

Both steps are optimal in terms of the spectral and Frobenius norms. It is important to note that the solution in the spectral norm may not be unique. Similar to the column subset selection problem, obtaining an optimal subset of k columns and rows is a combinatorial problem—a trivial algorithm takes n^k and m^k time, respectively, so existing procedures seek an approximate solution. Consequently, given k, a CUR decomposition is not unique; there are several ways to obtain this form of approximation to A with different techniques of choosing

the representative columns and rows. Many algorithms for this decomposition using the TSVD as a basis have been proposed [10, 33, 72, 83, 94].

The optimal rank-k approximation error in terms of the spectral or Frobenius norm given by

$$\min_{U_k \in \mathbb{R}^{m\times k}} \|A - U_k U_k^+ A\|_\xi, \quad \text{for } \xi = 2 \text{ or } F,$$

provides a lower bound for $\|A - CMR\|_\xi$. Most algorithms proposed in the literature seek to construct C and R such that

$$\|A - CMR\|_\xi \leq \mu \cdot \|A - A_k\|_\xi \tag{2.4}$$

for some modest value of μ. In other words, existing methods for a CUR decomposition aim to find an approximation that is at least as good as the SVD.

2.4 DETERMINISTIC ALGORITHMS FOR A CUR FACTORIZATION

In the following sections, we summarize some well-known and most relevant deterministic index selection algorithms for computing a CUR factorization.

2.4.1 *Leverage scores sampling*

The deterministic column subset selection algorithm proposed by Jolliffe [66] is one of the first column subset selection algorithms. The algorithm corresponds to the largest leverage scores for some target rank k. A leverage score is used to determine the impact of a specific row or column in a matrix on the overall matrix. It can be thought of as representing the influence or "leverage" that a particular row or column has on the rest of the matrix.

Definition 2.3. *Given a data matrix $A \in \mathbb{R}^{m\times n}$, consider any matrix $Q \in \mathbb{R}^{m\times n}$ with orthonormal columns such that* $\text{Range}(Q) = \text{Range}(A)$. *The ith leverage score corresponding to the ith row of A is defined as*

$$\ell(i) := \|Q(i,:)\|^2, \text{ for } i = 1, \ldots, m.$$

Since $\|Q\|_F = \sqrt{n}$, we have that

$$\ell(i) \geq 0 \text{ for all } i \quad \text{and} \quad \sum_{i=1}^{m} \ell(i) = n.$$

Note that the leverage score does not depend on the particular choice of the orthonormal basis matrix Q. Leverage scores are typically computed using the singular vectors of A.

Most leverage-score-based CUR decomposition algorithms involve randomized sampling [10, 31, 33, 72, 94]. Papailiopoulos, Kyrillidis, and Boutsidis [77] provide a provable approximation guarantee for the deterministic leverage score sampling summarized in Algorithm 1. The authors show that, when the leverage scores follow a moderate power-law decay, deterministic sampling can be just as accurate as randomized sampling.

Algorithm 1: Deterministic leverage scores sampling [77]

Data: $A \in \mathbb{R}^{m \times n}$, k, $\theta = k - \varepsilon$, $\varepsilon \in (0, 1)$
Result: Column index set $\mathbf{p} \in \mathbb{N}_{+}^{c}$
1 Compute $V \in \mathbb{R}^{n \times k}$ (top right singular vectors of A)
2 Sort in a nonincreasing order $\ell_i = \|V(i,:)\|^2$ for $i = 1, \ldots, n$
3 Find index c such that: $c = \text{argmin}_c \left(\sum_{i=1}^{c} \ell_i > \theta \right)$
4 If $c < k$, set $c = k$
5 Index set $\mathbf{p}$ contains the indices of c largest leverage scores

Given a stopping parameter θ and target rank k, the deterministic leverage score sampling algorithm selects at least $c \geq k$ column indices. It is worth noting that an upper bound on the number of column indices c that this algorithm can select is not immediate. In [77], the authors show how the stopping parameter $\theta = k - \varepsilon$ for some $\varepsilon \in (0, 1)$ directly controls the number of output columns c. We refer the reader to [77] for more details on the theoretical analysis.

Leverage-score type procedures for constructing a CUR decomposition generally involve oversampling of rows and columns beyond the specified target rank k. Drineas, Mahoney, and Muthukrishnan [33] develop a sampling procedure using normalized leverage scores, which provide a "near optimal" approximation guarantee, i.e., the CUR approximation has an error close to that of the best rank-k approximation. The scores are computed from the k leading singular vectors of A, which are then used to form a probability distribution for identifying the "highly influential" rows and columns to include in the factorization [33, 72]. Mahoney and Drineas [72] demonstrate that, given a target rank k and an error

parameter ε, their leverage-score based algorithm can provide an approximation of a matrix A using a random subset of columns $C \in \mathbb{R}^{m\times c}$ and rows $R \in \mathbb{R}^{m\times r}$. The authors show that for some $0 < \varepsilon < 1$, with a constant probability, the approximation satisfies a relative error bound: $\|A - CMR\|_F \leq (2+\varepsilon)\|A - A_k\|_F$ when the number of sampled columns and rows are $c = r = \mathcal{O}(k\,\varepsilon^{-2}\log k)$. Here, A_k represents the optimal rank-k approximation of A and $M = C^+AR^+$. Wang and Zhang [94] also discuss similar ideas, proposing a CUR factorization using an adaptive sampling method that relies on leverage scores based on residuals computed during the process of constructing the decomposition.

2.4.2 *MaxVol Algorithm*

The MaxVol algorithm [50, 51] is a commonly used deterministic method for obtaining a CUR factorization of a matrix. This technique involves selecting a subset of rows and columns from the input matrix to create a smaller, well-conditioned submatrix. The MaxVol scheme is a search method that aims to find the submatrix with the largest volume in a tall-thin matrix. Here, the volume of a matrix is defined as the absolute value of its determinant. The MaxVol method is first introduced by Goreinov et al. [51], who used it to construct a rank-k CUR factorization called the pseudoskeleton approximation. Oseledets and Tyrtyshnikov [75] later developed a cross-approximation scheme that alternates between selecting rows and columns using the MaxVol algorithm.

Algorithm 2: MaxVol: Approximation to dominant submatrix [50]

Data: $U \in \mathbb{R}^{m\times k}$ with $m > k$, convergence tolerance δ (default 0.01)
Result: $\mathbf{s} \in \mathbb{N}_+^k$ indices
$\mathbf{s} \leftarrow k$ first indices of pivoted rows from LU decomposition of U
repeat
  Set $\widehat{U} \leftarrow U(\mathbf{s},:)$ and $B \leftarrow U\widehat{U}^{-1}$
  Find the element of maximum absolute value in B: $(i,j) \leftarrow \operatorname{argmax}|b_{ij}|$
  if $|b_{ij}| > 1$, swap rows i and j in B: $\mathbf{s}(j) = i$
until $\forall\,(i,j)$: $|b_{ij}| < 1 + \delta$;

Given a tall-thin matrix $U \in \mathbb{R}^{m\times k}$, the MaxVol procedure searches for k row indices such that the resulting $k \times k$ upper submatrix $\widehat{U}$ is dominant in U [50]. This means that $|U\widehat{U}^{-1}|_{ii} \leq 1$. While the dominant property does not necessarily imply that $\widehat{U}$ has the maximum volume, it does guarantee that $\widehat{U}$ is locally optimal; meaning that replacing any row in $\widehat{U}$ with a row from U that is not

already present in $\widehat{U}$ will not increase the volume. The procedure for finding a dominant submatrix using the MaxVol algorithm is outlined in Algorithm 2.

A useful initialization step for the MaxVol algorithm is to use the pivoted rows from the LU decomposition of the input matrix as the starting point [50]. The parameter δ is the convergence tolerance to find pivot elements; this parameter serves as a stopping criterion and should be sufficiently small (a good choice can be 0.01) [50]. Note that by swapping the rows as done in Line 5, the volume of the upper submatrix in B is increased, and also in U until convergence. The most expensive part of the iterations is Line 3; this needs a $k \times k$ matrix inversion and $\mathcal{O}(mk^2)$ operations for the matrix multiplication. Goreinov et al. [50] describe a speed optimization process that avoids expensive matrix multiplications and inversions. We refer the reader to [50] for a more detailed explanation of the MaxVol approach.

2.4.3 *Discrete empirical interpolation method*

The discrete empirical interpolation method (DEIM) is a deterministic greedy index selection algorithm originally presented in [16], as a discrete variant of the empirical interpolation method proposed in [5] in the context of model order reduction for nonlinear dynamical systems. Sorensen and Embree [83] show that this procedure is a viable index selection algorithm for constructing a CUR factorization. The DEIM scheme plays an important role in all the newly proposed methods in this thesis.

The procedure works on the columns of specified basis vectors sequentially. The basis vectors must be linearly independent. Assuming we have a full-rank basis matrix $U \in \mathbb{R}^{m \times k}$ with $k < m$, to select k rows from U, the DEIM procedure constructs an index vector $\mathbf{s} \in \mathbb{N}_+^k$ such that it has nonrepeating values in $\{1, \ldots, m\}$. Defining the selection matrix S as an $m \times k$ identity matrix indexed by $\mathbf{s}$, i.e., $S = I(:, \mathbf{s})$ and $\mathbf{x}(\mathbf{s}) = S^\top \mathbf{x}$ (cf. [83]), we have an *interpolatory projector* defined through the DEIM procedure as

$$\mathbb{S} = U(S^\top U)^{-1} S^\top.$$

We can show that $S^\top U$ is nonsingular (see [83, Lemma 3.2]). The term *interpolatory projector* stems from the fact that for any $\mathbf{x} \in \mathbb{R}^m$, we have

$$(\mathbb{S}\mathbf{x})(\mathbf{s}) = S^\top \mathbb{S}\mathbf{x} = S^\top U(S^\top U)^{-1} S^\top \mathbf{x} = S^\top \mathbf{x} = \mathbf{x}(\mathbf{s}),$$

implying the projected vector $\mathbb{S}\mathbf{x}$ matches $\mathbf{x}$ in the $\mathbf{s}$ entries [83].

Although the DEIM index selection procedure is basis dependent, if the interpolation indices are determined, the DEIM interpolatory projector is independent of the choice of basis spanning the space $\text{Range}(U)$.

Proposition 2.4. *([16, Def. 3.1, (3.6)]). Let Q be an orthonormal basis of $\text{Range}(U)$. Then,*

$$U(S^\top U)^{-1}S^\top = Q(S^\top Q)^{-1}S^\top.$$

This proposition allows us to take advantage of the special properties of a matrix with orthonormal columns in cases where our input basis matrix is not (see Proposition 6.8 and Theorem 7.5).

We will now describe the DEIM index selection process. To select the indices contained in $\mathbf{s}$, the columns of U are considered successively. The first interpolation index corresponds to the index of the entry with the largest magnitude in the first basis vector. The rest of the interpolation indices are selected by removing the direction of the interpolatory projection in the previous basis vectors from the subsequent one and finding the index of the entry with the largest magnitude in the residual vector.

To form $\mathbf{s}$, let $\mathbf{u}_j$ denote the jth column of U and U_j be the matrix containing the first j columns of U. Similarly, let $\mathbf{s}_j$ contain the first j entries of $\mathbf{s}$, and let $S_j = I(:, \mathbf{s}_j)$. More precisely, we define s_1 such that $|\mathbf{u}_1(s_1)| = \|\mathbf{u}_1\|_\infty$ and the jth interpolatory projector $\mathbb{S}_j$ as

$$\mathbb{S}_j = U_j(S_j^\top U_j)^{-1}S_j^\top.$$

To select s_j, remove the $\mathbf{u}_1, \ldots, \mathbf{u}_{j-1}$ components from $\mathbf{u}_j$ by projecting $\mathbf{u}_j$ onto indices $\{s_1, \ldots, s_{j-1}\}$; thus,

$$\mathbf{r}_j = \mathbf{u}_j - \mathbb{S}_{j-1}\mathbf{u}_j. \tag{2.5}$$

Then take the index of the entry with the largest magnitude in the residual, i.e., s_j, such that

$$|\mathbf{r}_j(s_j)| = \|\mathbf{r}_j\|_\infty.$$

As noted in [16], in case of a tie, e.g., $|(\mathbf{r}_j)_i| = |(\mathbf{r}_j)_\ell|$ for $i \neq \ell$, the smaller index is picked. This process will never produce duplicate indices as noted in the paper "A DEIM-induced CUR decomposition" [83, p. A1458]. In a nutshell, we find the indices via a nonorthogonal Gram–Schmidt-like process (oblique projections) on the **u**-vectors. Since the input vectors are linearly independent, the residual vector **r** is guaranteed to be nonzero.

Algorithm 3: Discrete empirical interpolation index selection method [16]

Data: $U \in \mathbb{R}^{m\times k}$ with $k \leq m$ (full rank)
Result: Indices $\mathbf{s} \in \mathbb{N}_+^k$ with non-repeating entries
$\mathbf{s}(1) = \text{argmax}_{1\leq i\leq m} \ |(U(:,1))_i|$
for $j = 2, \ldots, k$ **do**
  $U(:,j) = U(:,j) - U(:,1:j-1)\cdot(U(\mathbf{s},1:j-1) \setminus U(\mathbf{s},j))$
  $\mathbf{s}(j) = \text{argmax}_{1\leq i\leq m} \ |(U(:,j))_i|$
end

This DEIM algorithm forces the selection matrix S to find k linearly independent rows of U such that the local growth of $\|(S^\top U)^{-1}\|$ is kept modest via a greedy search [16, p. 2748] as implemented in Algorithm 3 [1]. It is worth mentioning that, the DEIM index selection process is limited by the rank of the basis matrix; i.e., the number of indices to be selected can be no more than the number of basis vectors available.

In [34], Drmac and Gugercin propose a column-pivoted QR factorization-based DEIM called QDEIM, which is much simpler than the original DEIM with a bound that is in the same order of magnitude as the DEIM projection error bound. The availability of a column-pivoted QR implementation in many open-source packages makes this algorithm an efficient alternative index selection scheme.

Algorithm 4: QDEIM index selection scheme [34]

Data: $U \in \mathbb{R}^{m\times k}$ with $k \leq m$ (full rank)
Result: Indices $\mathbf{s} \in \mathbb{N}_+^k$ with non-repeating entries
$[\sim, \sim, \text{pivot}] = \text{qr}(U^\top)$ (column-pivoted QR on $U^\top$)
$\mathbf{s} = \text{pivot}(1:k)$

To construct a rank-k DEIM-type CUR decomposition, one first applies Algorithm 3 to the k leading right and left singular vectors to obtain the column and

1 Note that the forward and back slash operators used in the algorithms are Matlab-type notation for solving linear systems and least-squares problems.

row indices for forming the factors C and R, respectively. Similarly, using the QDEIM approach [34] as summarized in Algorithm 4, one can apply a column-pivoted QR procedure on the transpose of the leading k right and left singular vectors to find the indices for constructing the factors C and R. Then, given full-rank matrices C and R, the middle matrix is computed as $M = C^{+}AR^{+}$.

Using the results of the error analysis of the DEIM procedure in [83] (cf. also Section 4.2.4), a CUR factorization is formed such that

$$\|A - CMR\| \leq 2^k \left(\sqrt{\tfrac{mk}{3}} + \sqrt{\tfrac{nk}{3}} \right) \sigma_{k+1}.$$

Analogously, using the results from [34] for the QDEIM scheme, one can derive the k column and row indices for constructing C and R such that

$$\|A - CMR\| \leq \sqrt{\tfrac{4^k+6k-1}{3}} \left(\sqrt{\tfrac{m-k+1}{\sigma_{\min}(U)}} + \sqrt{\tfrac{n-k+1}{\sigma_{\min}(V)}} \right) \sigma_{k+1},$$

where U and V contain the k-leading left and right singular vectors. Notice that both bounds feature an exponential dependence on k and depend on the dimension of the matrix.

Although the computation of an (approximate) SVD is a prerequisite for the CUR algorithms discussed and proposed in this thesis, it is important to highlight that there exist alternative procedures for constructing a CUR factorization that do not rely on the SVD. These alternative methods are often designed to reduce computational complexity. The works of [7, 23, 24, 85, 93] present deterministic algorithms for column subset selection and a CUR factorization that leverages a column-pivoted QR decomposition. The column-pivoted QR decomposition is a cheaper alternative to the SVD.

2.4.4 *Rank-revealing QR factorization*

The classical truncated column-pivoted QR factorization is another approach for computing a CUR factorization [93]. Notice that from (2.2), $Q_1 T_{11}$ equals the first k columns of $A(:, \mathbf{p})$, so we can form $C = Q_1 T_{11}$. To construct a CUR decomposition, one may apply a Gram–Schmidt-based pivoted QR algorithm to the matrices A and $A^{\top}$ to obtain the matrices C and R, respectively [7, 85].

Alternatively, Voronin and Martinsson [93] show how to compute a CUR factorization via a two-sided interpolative decomposition (ID), which in turn can be constructed from a column-pivoted QR factorization. Given matrix $A \in \mathbb{R}^{m \times n}$,

an $m \times k$ matrix C whose columns constitute a subset of those of A, and a $k \times n$ matrix $\widetilde{M}$, such that

- some size-k subset of the columns of $\widetilde{M}$ form the $k \times k$ identity matrix, and
- no entry of $\widetilde{M}$ has absolute value greater than 1,

$C\widetilde{M}$ is an interpolative decomposition of A [21, 93]. One could equally approximate the matrix as $A \approx \widehat{M}R$, where R contains k rows of A and the properties of $\widehat{M}$ are analogous to those of $\widetilde{M}$. Both interpolative and CUR decompositions share the common goal of identifying appropriate column and/or row subsets of A such that the selected columns and/or rows adequately span the column and/or row spaces of the original matrix. Constructing an ID and a CUR decomposition via a two-sided interpolative decomposition [93] are summarized in Algorithms 5 and 6.

Algorithm 5: Rank-k interpolative decomposition [93]

Data: $A \in \mathbb{R}^{m \times n}$ and desired rank $k \leq m$
Result: A rank-k ID $A \approx CX^\top$
1 Perform a rank-k column-pivoted QR; $A\Pi = Q_1 T_1$
2 Define the ordered set $\mathbf{p}$ via $I(:, \mathbf{p}) = \Pi$
3 Partition T_1: $T_{11} = T_1(:, 1:k)$, $T_{12} = T_1(:, k+1:n)$
4 $C = A(:, \mathbf{p}(1:k))$ and $X = \Pi \cdot [I_k \;\; T_{11} \backslash T_{12}]^\top$

Algorithm 6: Rank-k CUR-ID [93]

Data: $A \in \mathbb{R}^{m \times n}$ and desired rank $k \leq m$
Result: A rank-k CUR decomposition $A \approx CMR$
1 Perform Algorithm 5 on A so that $A \approx CX_c^\top$
2 Perform Algorithm 5 on $C^\top$ so that $C^\top \approx C^\top(:, \mathbf{s}(1:k))\, X_r^\top$
3 Construct $R = A(:, \mathbf{s}(1:k))$ and $M = X_c^\top / R$

CONTRIBUTIONS OF PART I

In [83], it has been demonstrated that the discrete empirical interpolation method can be used to select indices to formulate a CUR factorization. However, the DEIM scheme has two notable limitations: First, the number of column or row

indices that can be selected is limited to the rank of the input matrix. Secondly, the DEIM procedure locally selects the index corresponding to the largest magnitude element of a vector. A potentially problematic case is if multiple entries in the vector have nearly the same magnitude. Thus, the DEIM scheme faces a difficult choice when the local maximizer is (nearly) nonunique.

Chapter 3 of the thesis presents an extension to overcome the first limitation. The proposed procedure allows for choosing a number of indices greater than the rank of the input matrix. For this algorithm, we combine the strength of deterministic leverage scores and the DEIM schemes.

In Chapter 4, a modification of the DEIM method is introduced to tackle the second drawback. We present new methods for approximating large matrices that are based on the DEIM scheme but designed to be less greedy and more efficient. The proposed techniques are based on the rank-revealing QR factorization and the concept of the maximum volume. The new schemes may also be a good solution where the DEIM procedure faces a difficult choice when the local maximizer is (nearly) nonunique because the optimization is done over more indices instead of just one.

Chapter 5 introduces new columns or rows selection strategies that iteratively invoke the DEIM algorithm. These iterative subselection procedures can potentially result in a better CUR approximation accuracy and also allow for flexibility in the index selection process. In the sense that, the method can be used with different criteria for selecting the subset of columns and rows, depending on the specific problem or application. Thus, it allows for a flexible and customizable approach to a CUR decomposition.

3

A HYBRID DEIM AND LEVERAGE SCORES BASED METHOD FOR CUR INDEX SELECTION

The discrete empirical interpolation method is shown to be a viable index selection strategy for formulating a CUR factorization. A notable drawback of the original DEIM algorithm is that the number of column or row indices that can be selected is limited to the rank of the input matrix. We propose a new variant of the DEIM scheme, which we call L-DEIM, a combination of the strength of deterministic leverage score sampling and the DEIM procedure. This method allows for the selection of a number of indices greater than the rank of the input matrix. Since the DEIM technique requires (approximate) singular vectors as input matrices, L-DEIM is particularly attractive when computing a rank-k SVD approximation is expensive even for moderately small k; this is because the L-DEIM procedure uses a lower-rank SVD approximation instead of the full rank-k SVD. We empirically demonstrate the performance of L-DEIM, which despite its efficiency, may achieve comparable results to the original DEIM and even better approximations than some state-of-the-art methods.

Adapted from [43]

3.1 INTRODUCTION

Large matrices often represent data sets, which can be difficult to manage due to their size, particularly with the growth of the internet and the rise of industrial data. Examples of these data sets include text documents, customer databases, stocks, and financial transactions. Dimension reduction is often necessary for data analysis, and in many applications, interpretable dimension reduction is required, which can be achieved through a CUR decomposition. This factorization as mentioned in Chapter 1 involves selecting a subset of column and row indices of the original data matrix, which can be accomplished by using the DEIM algorithm. While the DEIM algorithm has proven to be useful in many scenarios, it has certain limitations; one of which is that it is not possible to select more indices than the number of available basis vectors. However, there are application scenarios where oversampling is beneficial. Consider the problem of class identification. In some cases, the number of classes in a data set may be larger than the rank of the matrix; this can occur when the number of derived features or time samples

per observation is small compared to the number of classes. For example, in the case of time series data, if there are a large number of classes but only a few time samples per observation, the DEIM algorithm can only detect as many classes as there are time samples. As data sets continue to grow in size, it is becoming more common for the number of observations to be much larger than the number of features or time samples per observation. An example of this is the identification of different beat morphologies in electrocardiogram (ECG) tracings, where there may be billions of tracings with only a few hundred samples per beat, but a large number of distinct beat morphologies (see, e.g., [61]).

In an attempt to address this limitation of the DEIM scheme, we propose a new extension called L-DEIM. The L-DEIM scheme combines the strengths of deterministic leverage scores sampling [77] and the DEIM procedure [83] (also see, Algorithms 1 and 3). Our new approach is an alternative deterministic index selection method which allows for oversampling of indices beyond the rank of the input basis matrix. Since the DEIM scheme relies on the SVD or its approximation, the proposed approach may be particularly attractive in a setting (for example big data problems) where we want a rank-$\widehat{k}$ CUR decomposition and computing a rank-$\widehat{k}$ SVD approximation is expensive even for moderately small $\widehat{k}$. This new algorithm allows us to select $\widehat{k}$ indices without having to compute the full rank-$\widehat{k}$ SVD by using a lower-rank SVD approximation instead. It may be viewed as an approach to reuse the same information to further improve the approximation.

3.2 L-DEIM

We now introduce the new deterministic index selection scheme for a CUR factorization, which allows for oversampling of column and row indices. Our starting point is the method from the earlier work [83], which derives a rank-$\widehat{k}$ CUR factorization by applying DEIM to the leading $\widehat{k}$ singular vectors. Given the promising results of this algorithm compared to other state-of-the-art methods for a CUR approximation, our proposed algorithm builds on the DEIM procedure.

Constructing a rank-$\widehat{k}$ CUR decomposition using the L-DEIM scheme requires a rank-k SVD where $k < \widehat{k}$. The integer k is the number of available (approximate) singular vectors, while $\widehat{k}$ is the number of indices to be selected. To select the $\widehat{k}$ indices, the proposed method performs the original DEIM to find the first k indices while keeping the residual singular vector in each index selection step of the DEIM procedure. The residual singular vector is the error between the input singular vector and its approximation from interpolating the previous singular vectors at the selected indices (also see (2.5)). At the end of the iteration,

using the idea of leverage scores, we compute the 2-norm of the rows of the residual singular vectors to select the additional $\widehat{k} - k$ indices. The procedure is summarized in Algorithm 7. Note that the vectors of U in Line 6 of Algorithm 7 are the residual singular vectors and not the original singular vectors.

Algorithm 7: L-DEIM index selection

Data: $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{n \times k}$, target rank = $\widehat{k}$, with $k \leq \widehat{k} \leq \min(m, n)$
Result: column and row indices $\mathbf{p}, \mathbf{s} \in \mathbb{N}_{+}^{\widehat{k}}$, respectively, with non-repeating entries

1 $\mathbf{s}(1) = \text{argmax}_{1 \leq i \leq m} \ |(U(:, 1))_i|$
2 **for** $j = 2, \ldots, k$ **do**
3 $\quad U(:, j) = U(:, j) - U(:, 1:j-1) \cdot (U(\mathbf{s}, 1:j-1) \setminus U(\mathbf{s}, j))$
4 $\quad \mathbf{s}(j) = \text{argmax}_{1 \leq i \leq m} \ |(U(:, j))_i|$
5 **end**
6 Compute $\ell_i = \|U(i, :)\| \quad$ for $i = 1, \ldots, m$
7 Remove entries in ℓ corresponding to the indices in $\mathbf{s}$
8 $\mathbf{s}' = \widehat{k} - k$ indices corresponding to $\widehat{k} - k$ largest entries of ℓ
9 $\mathbf{s} = [\mathbf{s};\ \mathbf{s}']$
10 Perform 1–9 on V to get index set $\mathbf{p}$

From Algorithm 7, if $\widehat{k} = k$ then the algorithm reduces to the standard DEIM (i.e., Algorithm 3). We note that if the target rank is not specified, given k, we can select at least k indices but the upper bound on the number of indices to be selected is not immediate; we can select an arbitrary number of indices. Similar to leverage scores sampling, the L-DEIM allows for oversampling of columns and rows. Given the target rank $\widehat{k}$, the DEIM procedure involves $\mathcal{O}((m+n)\,\widehat{k}^2)$ while the L-DEIM scheme involves $\mathcal{O}((m+n)\,k^2)$ for selecting the first k indices and $\mathcal{O}((m+n)\,k)$ for selecting the additional $\widehat{k} - k$ indices.

L-DEIM projection error bound

To provide an analysis for the L-DEIM projection error bound, we begin by revisiting some properties of the DEIM index selection algorithm.

Given a full-rank matrix $V \in \mathbb{R}^{n \times k}$ with $k < n$, we construct the index set $\mathbf{p} \in \mathbb{N}_{+}^{k}$ by choosing k distinct rows from V. Suppose $P = I(:, \mathbf{p})$ is an $n \times k$ selection matrix, it can be shown that $V^\top P$ is nonsingular [83, Lemma 3.2]. The interpolatory projector defined through the DEIM index selection scheme is $\mathbb{P} = P(V^\top P)^{-1} V^\top$.

Suppose, however, that we want to select $\widehat{k} > k$ distinct row indices from V to form the index set $\mathbf{p} \in \mathbb{N}_+^{\widehat{k}}$. Then, letting $P = I(:, \mathbf{p}) \in \mathbb{R}^{n \times \widehat{k}}$, we have that $V^\top P$ is no longer square. Assuming $V^\top P$ is of full row rank, we can use the Moore–Penrose pseudoinverse of $V^\top P$ to define an interpolatory projector $\mathbb{P} = P(V^\top P)^+ V^\top$. The proof of the L-DEIM projection error for $\mathbb{P}$ closely follows that presented in [83].

Proposition 3.1. *Given $A \in \mathbb{R}^{m \times n}$ and $V \in \mathbb{R}^{n \times k}$ with orthonormal columns where $k < n$, let $P \in \mathbb{R}^{n \times \widehat{k}}$ with $k < \widehat{k} < n$ be a selection matrix and $V^\top P$ be of full-rank. Let $\mathbb{P} = P(V^\top P)^+ V^\top$. Then,*

$$\|A - A\mathbb{P}\| \leq \|A(I - VV^\top)\| \, \|(V^\top P)^+\|.$$

In particular, if V contains k leading right singular vectors of A, then

$$\|A - A\mathbb{P}\| \leq \sigma_{k+1} \cdot \|(V^\top P)^+\|.$$

Proof. We have that $V^\top \mathbb{P} = V^\top P(V^\top P)^+ V^\top = V^\top$, which implies that $V^\top(I - \mathbb{P}) = 0$. Therefore,

$$\begin{aligned} \|A - A\mathbb{P}\| &= \|A(I - \mathbb{P})\| = \|A(I - VV^\top)(I - \mathbb{P})\| \\ &\leq \|A(I - VV^\top)\| \, \|I - \mathbb{P}\|. \end{aligned}$$

Note that, since $k < n$, we know that $\mathbb{P} \neq 0$ and $\mathbb{P} \neq I$, and hence (see, e.g., [89])

$$\|I - \mathbb{P}\| = \|\mathbb{P}\| = \|P(V^\top P)^+ V^\top\| = \|(V^\top P)^+\|.$$

Given that V contains the k leading right singular vectors we have $\|A(I - VV^\top)\| = \sigma_{k+1}$. ∎

Let us now consider the operation on the left-hand side of A. Given the row-selection projection $\mathbb{S} = U(S^\top U)^+ S^\top$, where $U \in \mathbb{R}^{m \times k}$ contains the k leading left singular vectors and $S \in \mathbb{R}^{m \times \widehat{k}}$, such that $S^\top U$ is of full rank, analogous to Proposition 3.1 we have the following projection error bound:

$$\|A - \mathbb{S}A\| \leq \sigma_{k+1} \cdot \|(S^\top U)^+\|.$$

We will now use Proposition 3.1 to find a bound for the approximation error of an L-DEIM based CUR factorization of A. The following proposition is a slight generalization of [83, Lemma 4.2] and we closely follow the proof technique of

Sorensen and Embree [83]. As in [83] we first show in the following proposition that the error bounds of the interpolatory projection of A onto the chosen rows and columns equally apply to the orthogonal projections of A onto the same row and column spaces.

Proposition 3.2. *(Slight generalization of [83, Lemma 4.2]) Given the selection matrices P, S, let $C = AP$ and $R = S^\top A$. Let $C \in \mathbb{R}^{m\times\widehat{k}}$ and $R \in \mathbb{R}^{\widehat{k}\times n}$ be full rank matrices with $\widehat{k} < \min(m,n)$, and $V^\top P$ and $S^\top U$ be of full row and column rank, respectively. We have the bound for the orthogonal projections of A onto the column and row spaces:*

$$\|(I - CC^+)A\| \le \sigma_{k+1} \cdot \|(V^\top P)^+\|,$$
$$\|A(I - R^+R)\| \le \sigma_{k+1} \cdot \|(S^\top U)^+\|.$$

Proof. This proof is a minor modification of that of [83, Lemma 4.2]; we closely follow their proof technique. With $C = AP$ of full rank, we have $C^+ = (P^\top A^\top AP)^{-1}(AP)^\top$. With this, the orthogonal projection of A onto Range(C) can be stated as

$$CC^+A = (AP(P^\top A^\top AP)^{-1}P^\top A^\top)A.$$

Let $\Pi_P = P(P^\top A^\top AP)^{-1}P^\top A^\top A$, note that $\Pi_P P = P$ since Π_P is an oblique projector on Range(P). We can rewrite CC^+A as $A\Pi_P$. Hence, the error in the orthogonal projection of A will be $(I - CC^+)A = A(I - \Pi_P)$. Since

$$\Pi_P \mathbb{P} = P(P^\top A^\top AP)^{-1}P^\top A^\top AP(V^\top P)^+V^\top = \mathbb{P},$$

we have

$$A(I - \Pi_P) = A(I - \Pi_P)(I - \mathbb{P}) = (I - CC^+)A(I - \mathbb{P}),$$

Consequently

$$\begin{aligned}\|(I - CC^+)A\| &= \|A(I - \Pi_P)\| = \|(I - CC^+)A(I - \mathbb{P})\| \\ &\le \|(I - CC^+)\|\ \|A(I - \mathbb{P})\|.\end{aligned}$$

With C being nonsquare, we have $\|I - CC^+\| = 1$ (see, e.g., [89]) and $\|A(I - \mathbb{P})\| \le \sigma_{k+1} \cdot \|(V^\top P)^+\|$ from Proposition 3.1. It follows that

$$\|(I - CC^+)A\| \le \sigma_{k+1} \cdot \|(V^\top P)^+\|.$$

In a similar vein, with $R = S^\top A$ and $R^+ = R^\top(RR^\top)^{-1}$ we have that $R^+ = A^\top S(S^\top AA^\top S)^{-1}$ and the error in the orthogonal projection of A is $A(I - R^+R) = (I - \Pi_S)A$, where $\Pi_S = AA^\top S(S^\top AA^\top S)^{-1}S^\top$. Thus,

$$(I - \Pi_S)A = (I - \mathbb{S})(I - \Pi_S)A = (I - \mathbb{S})A(I - R^+R)$$

and

$$\|A(I - R^+R)\| \leq \|(I - \mathbb{S})A\| \, \|(I - R^+R)\| \leq \sigma_{k+1} \cdot \|(S^\top U)^+\|.$$

■

This result helps to prove an error bound for the L-DEIM CUR approximation error. For the following theorem, we again closely follow the approach of [83] which also follows a procedure in [72]. Let $M = (C^\top C)^{-1} C^\top A R^\top (RR^\top)^{-1} = C^+ A R^+$.

Theorem 3.3. *(Generalization of [83, Thm. 4.1]) Given $A \in \mathbb{R}^{m \times n}$ and the k leading left and right singular vectors U and V, respectively, let $P \in \mathbb{R}^{n \times \widehat{k}}$ and $S \in \mathbb{R}^{m \times \widehat{k}}$ be selection matrices so that $C = AP$ and $R = S^\top A$ are of full rank. Assuming $V^\top P$ and $S^\top U$ are of full rank, then with the error constants*

$$\eta_p := \|(V^\top P)^+\|, \qquad \eta_s := \|(S^\top U)^+\|,$$

we have

$$\|A - CMR\| \leq (\eta_p + \eta_s) \cdot \sigma_{k+1}.$$

Proof. By the definition of M, we have

$$A - CMR = A - CC^+ A R^+ R = (I - CC^+)A + CC^+ A(I - R^+R).$$

Using the triangle inequality, it follows that [72, 83]

$$\begin{aligned} \|A - CMR\| &= \|A - CC^+ A R^+ R\| \\ &\leq \|(I - CC^+)A\| + \|CC^+\| \, \|A(I - R^+R)\|, \end{aligned}$$

and the fact that CC^+ is an orthogonal projector with $\|CC^+\| = 1$,

$$\|A - CMR\| \leq \sigma_{k+1} \cdot \|(V^\top P)^+\| + \|(S^\top U)^+\| \cdot \sigma_{k+1}.$$

■

Notice that the difference between the error bounds in Theorem 3.3 and [83, Thm. 4.1] is the Lebesgue constant for the discrete interpolation, i.e., η_s and η_p for the L-DEIM and the DEIM scheme. In [83, Lemma 4.4], the authors present an upper bound on the DEIM selection scheme error constants. We show here that this bound holds for the error constants from the L-DEIM selection scheme.

Given a matrix with orthogonal columns $U \in \mathbb{R}^{m\times k}$, we define the selection of k and $\widehat{k}$ rows of U (where $\widehat{k} > k$) by the DEIM and L-DEIM scheme, respectively, as

$$\begin{bmatrix} S_D^\top \\ S_L^\top \end{bmatrix} U =: \begin{bmatrix} Z_D \\ Z_L \end{bmatrix} =: Z,$$

where S_D selects k rows and S_L selects additional $(\widehat{k}-k)$ rows. The error constant of the DEIM procedure is $\sigma_{\min}^{-1}(Z_D)$ while that of the L-DEIM algorithm is $\sigma_{\min}^{-1}(Z) \le \sigma_{\min}^{-1}(Z_D)$. This implies that any upper bound for $\sigma_{\min}^{-1}(Z_D)$ is also an upper bound for $\sigma_{\min}^{-1}(Z)$. Therefore, given that $k < \widehat{k} < \min(m,n)$, we have

$$\eta_p := \|(V^\top P)^+\| < \sqrt{\tfrac{nk}{3}}\, 2^k\,, \quad \eta_s := \|(S^\top U)^+\| < \sqrt{\tfrac{mk}{3}}\, 2^k.$$

For constructive proofs and further explanation of the bound on the Lebesgue constant for the discrete interpolation, we refer the reader to [83, Lemma 4.4].

3.3 NUMERICAL EXPERIMENTS

In our first experiment, with the target rank $\widehat{k}$ fixed, we investigate how the CUR approximation accuracy of L-DEIM varies as k increases. The aim is to observe how fast the accuracy of a rank-$\widehat{k}$ L-DEIM CUR approximation using $k < \widehat{k}$ singular vectors approaches that of a rank-$\widehat{k}$ DEIM CUR approximation and the optimal rank-$\widehat{k}$ approximation. Similar to [83], we generate a sparse, nonnegative matrix $A \in \mathbb{R}^{m\times n}$, with $m = 10000$ and $n = 300$, of the form

$$A = \sum_{j=1}^{10} \frac{2}{j}\, \mathbf{x}_j\, \mathbf{y}_j^\top + \sum_{j=11}^{300} \frac{1}{j}\, \mathbf{x}_j\, \mathbf{y}_j^\top,$$

where $\mathbf{x}_j \in \mathbb{R}^m$ and $\mathbf{y}_j \in \mathbb{R}^n$ are sparse vectors with random nonnegative entries (i.e., $\mathbf{x}_j = \mathsf{sprand}(m, 1, 0.025)$ and $\mathbf{y}_j = \mathsf{sprand}(n, 1, 0.025)$. We fix $\widehat{k} = 30$ and vary k from $1, \ldots, 30$, i.e., $\widehat{k} = 30k, \cdots, \widehat{k} = k$, and similarly for $\widehat{k} = 20$. We generate 100 random matrices of A and Fig. 3.1 reports the average relative errors $\|A - CMR\|/\|A\|$. We observe that the approximation quality of the L-DEIM procedure is comparable to that of the DEIM scheme when k is at least $\frac{\widehat{k}}{2}$.

Using real data sets, we perform some experiments to compare the approximation quality and runtimes of the new method L-DEIM with the existing deterministic methods discussed in Chapter 2. We use the relative error $\|A - CMR\|/\|A\|$

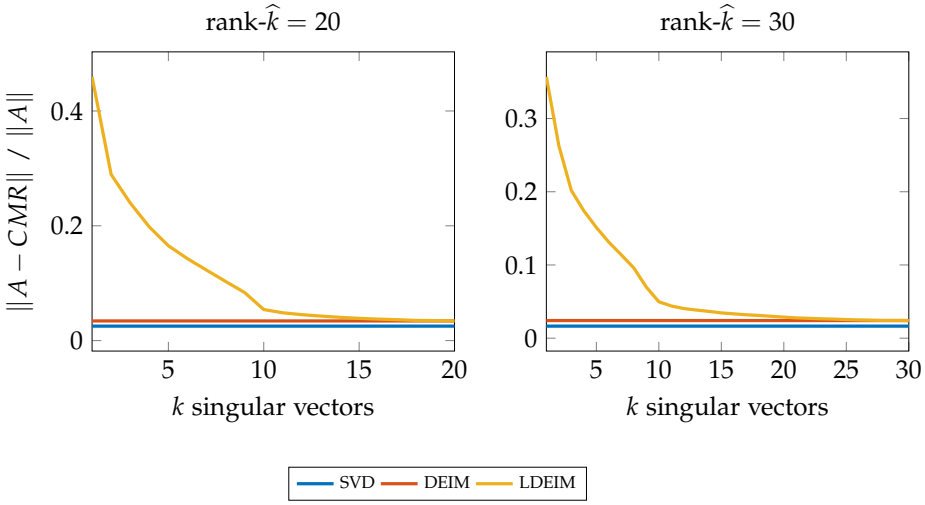


Figure 3.1: The relative approximation errors of the L-DEIM scheme for varying k singular vectors in constructing a rank-$\widehat{k}$ CUR factorization compared with the standard DEIM and the optimal rank-$\widehat{k}$ approximation.

and runtimes for selecting the column and row indices as the evaluation criteria. The application domains of the data sets are Internet term document analysis, genomics, and image analysis. The first is the cancer genetics data set gse10072 from the National Institutes of Health [71]. This data set has 107 patients described by 22283 probes. There are 58 patients with tumors and 49 without. We center the 22283×107 genetics data matrix by subtracting the mean of each row from the entries in that row. We also use the cifar-10 data set [70], a collection of 60000 32×32 pixel color images in ten different classes, with 6000 images per class. The data set is divided into 50000 training set images and 10000 testing set images. We use the training data set. The ten classes in the cifar-10 dataset are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The cifar-10 dataset is commonly used as a benchmark dataset for image classification tasks in machine learning research. Each image in the cifar-10 data set has been reshaped into a 1D array of length 3072, resulting in a 50000×3072 dense matrix. Our last data set is a 29610×29610 low-rank sparse matrix from the SuiteSparse Matrix collection [26] called g7jac100. This data matrix is one of the matrices from an "Overlapping Generations Model" used to study the social security systems of the G7 nations compiled by CEPII, a research institute in Paris.

Table 3.1: Various examples and dimensions considered.

Exp.	Name	Matrix	m	n
1	gse10072	Sparse	22283	107
2	cifar-10	Dense	50000	3072
3	g7jac100	Sparse	29610	29610

For the first data set, due to its small size, we compute the full SVD for all the algorithms using the Matlab in-built function svd while for the other two data sets, we compute only the leading k or $\widehat{k}$ singular vectors using the function svds in Matlab. The reported runtimes include the time taken to compute the leading singular vectors, except for the gse10072 dataset, where we computed the full SVD for all methods and did not include this time.

From Fig. 3.2, we see that the approximation quality of the proposed method L-DEIM can be as good as the original DEIM while the L-DEIM enjoys favorable runtimes. Both DEIM and L-DEIM have considerably lower approximation errors than the other methods. It is also worth pointing out that although the execution times for both the leverage scores sampling and the L-DEIM are not significantly different for the last two data sets, the improvement in approximation error of L-DEIM over the leverage scores is quite significant. We observe from the results not presented here that computing the leverage scores using only the leading two singular vectors yields better approximation results, although not as good as the other index selection methods evaluated here.

Table 3.2 reports the computational times of the various algorithms evaluated in each of three phases: 1) computing the partial the SVD, 2) generating the $\widehat{k}$ indices, 3) forming the CUR approximation for computing a rank-500 approximation of the g7jac100. For the DEIM and QDEIM-based CUR factorization, we compute the first 500 singular vectors while for the L-DEIM and leverage score sampling, we compute the first 250 singular vectors. It is evident that compared to the DEIM method, the L-DEIM approach achieves a significant reduction in computational time not only in computing a rank-k SVD instead of a rank-$\widehat{k}$ SVD but also in selecting the indices. This is because selecting the additional k indices (suppose $\widehat{k} = 2k$) involves $\mathcal{O}((m+n)\,k^2)$ operations for the DEIM scheme compared to $\mathcal{O}((m+n)\,k)$ operations for the L-DEIM procedure.

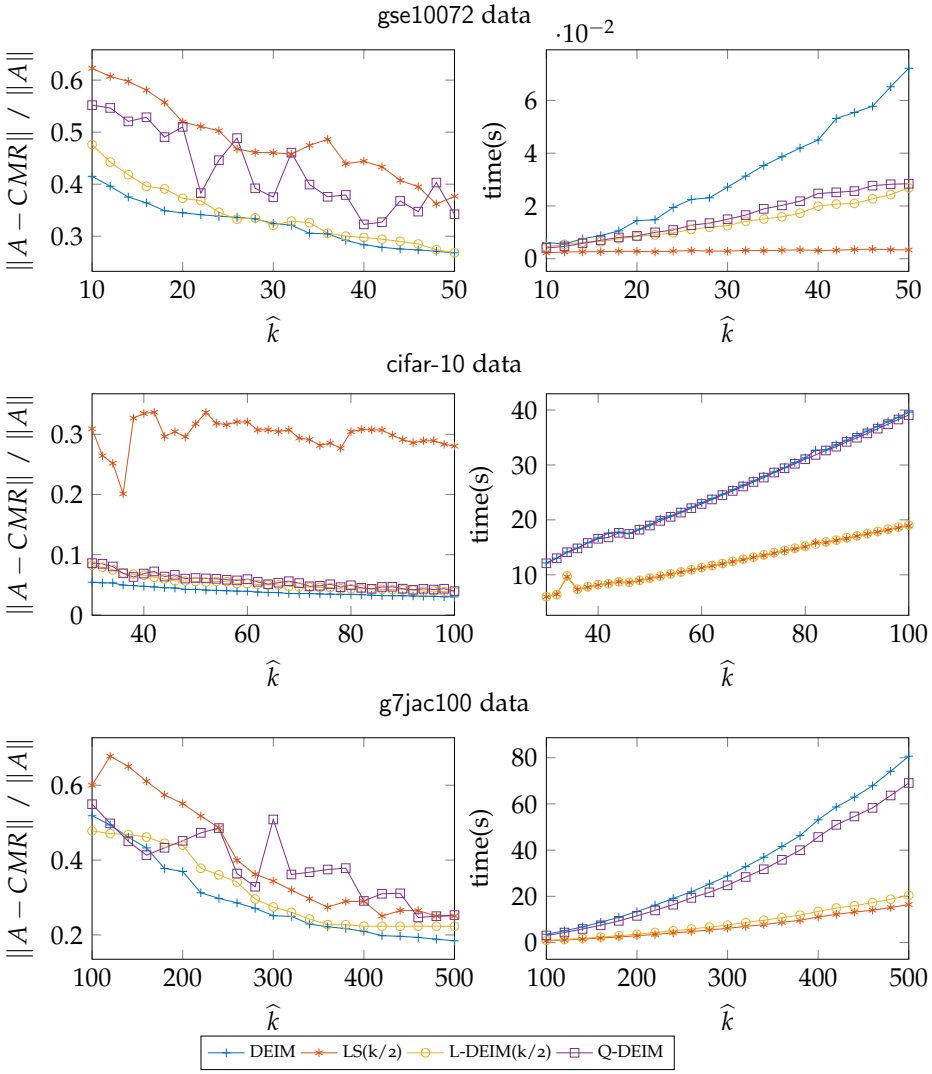


Figure 3.2: The relative approximation errors (first column) and runtimes (second column) of the L-DEIM scheme compared with the standard DEIM, QDEIM, and leverage scores sampling (LS) techniques using the three real data sets. For the L-DEIM and LS methods, we use $k = \widehat{k}/2$ singular vectors.

Table 3.2: Computational times of the various algorithms in each of three phases: 1) computing the partial the SVD, 2) generating the $\widehat{k}$ indices, 3) forming the CUR approximation using the g7jac100 data.

Method	Computational time(s)		
	SVD	$\widehat{k}$ index selection	CMR
DEIM	69.87	16.46	82.79
QDEIM	69.87	4.26	80.30
LDEIM	17.69	4.18	38.13
LS	17.69	0.06	40.45

3.4 FINAL CONSIDERATIONS

We have presented a new extension of the DEIM index selection algorithm (L-DEIM) to identify additional indices for constructing a rank-$\widehat{k}$ CUR decomposition using a lower-rank SVD approximation. This technique is especially useful in a setting (for example big data problems) where computing a full rank-$\widehat{k}$ SVD is relatively expensive. Since L-DEIM allows for oversampling, we only compute a smaller set of left and right singular vectors, hence, we can reduce computational complexity and memory resources. The algorithm may be viewed not only as an extension of DEIM but also as an alternative index selection method for a CUR factorization. The L-DEIM procedure may also be suitable for point selection in the context of model order for nonlinear dynamical systems. Although the proposed algorithm is computationally more efficient than the original DEIM, experiments show that the approximation accuracy of both methods may be comparable when the target rank $\widehat{k}$ is at most twice the available k singular vectors. For all results presented in Section 3.3, we assume that given a target rank $\widehat{k}$, $2k = \widehat{k}$ in Algorithm 7. From the first experiment (also experiments not presented here), if $k \ll \frac{1}{2}\widehat{k}$ in Algorithm 7, then the rank-$\widehat{k}$ CUR approximation quality of the L-DEIM procedure which uses k singular vectors may generally be worse than the rank-$\widehat{k}$ CUR factorization quality of the standard DEIM scheme which requires $\widehat{k}$ singular vectors. However, we stress that the L-DEIM is considerably cheaper and allows for oversampling of indices.

The model order reduction community has delved deeper into the idea of index oversampling concerning the DEIM and QDEIM approaches. Several references, including [74, 80, 100], have explored this concept. In light of this, future studies comparing the L-DEIM with these proposed extensions of the DEIM procedure

within the framework of a CUR decomposition as well as model order reduction applications are necessary.

4

BLOCK DISCRETE EMPIRICAL INTERPOLATION METHODS

This chapter presents two block variants of the discrete empirical interpolation method for a CUR factorization. The block DEIM algorithms are based on the concept of the maximum volume of submatrices and a rank-revealing QR factorization. We also present a version of the block DEIM procedures, which allows for adaptive choice of block size. The results of the experiments indicate that the block DEIM algorithms exhibit comparable accuracy for low-rank matrix approximation compared to the standard DEIM procedure. However, the block DEIM algorithms also demonstrate potential computational advantages, showcasing increased efficiency in terms of computational cost.

Adapted from [46]

4.1 INTRODUCTION

In this chapter, we propose two different block DEIM procedures and an adaptive variant of the block techniques. There are three motivating factors for a block DEIM scheme:

- The DEIM algorithm may be considered a *greedy* algorithm for finding a submatrix with the maximum absolute determinant in a thin-tall matrix: it locally selects the index corresponding to the largest magnitude element of a vector. A block DEIM method shares the same principle but may be less greedy since the optimization is done over more indices instead of just one.
- The proposed block DEIM procedures will generally have higher flop counts than the standard DEIM algorithm. However, in practice, we expect the block DEIM schemes to have higher flop performance than the classical DEIM algorithm as they are mainly based on level 3 BLAS building blocks, which perform matrix-matrix operations.
- A block DEIM scheme may be a good solution where the DEIM procedure faces a difficult choice when the local maximizer is (nearly) nonunique. Given the basis vectors $\mathbf{v}_i$ for $i = 1, \ldots, k$, indeed, in [83, Footnote 3] it is already hinted that a potentially problematic case is if multiple entries in a vector being considered have nearly the same magnitude, e.g., $|(\mathbf{v}_1)_\ell| \approx |(\mathbf{v}_1)_j|$ for $\ell \neq j$, then the DEIM scheme may sometimes make a

relatively arbitrary choice. However, in cases where the nonselected large-magnitude entries in $\mathbf{v}_1$ are not influential in the subsequent $\mathbf{v}_i$ vectors, their corresponding indices may never be picked, despite their equal importance (see Example 4.1).

Our selection criterion for the block-b case will consist of maximizing the modulus of the determinant (volume) over many possible $b \times b$ submatrices. By default, we set b to 2, but one can choose any value for b as long as it is within the range $2 \leq b < k$. We also discuss the option of taking adaptive values of b. To motivate the discussion, let us first consider a small illustrative example.

Example 4.1. Consider the matrix of left singular vectors (where, e.g., $\varepsilon = 10^{-15}$)

$$U = \begin{bmatrix} \frac{1}{3}\sqrt{3}+\varepsilon & 0 \\ \frac{1}{3}\sqrt{3} & \frac{1}{2}\sqrt{2}+\varepsilon \\ \frac{1}{3}\sqrt{3} & -\frac{1}{2}\sqrt{2} \end{bmatrix}, \quad \text{corresponding to singular values } \sigma_1,\ \sigma_2.$$

It may easily be checked that independent of the σ_i, standard DEIM will pick the first index, followed by the second. However, it is clear that the second and third indices are a better choice, especially if σ_2 is close to σ_1. For instance, if $A = U \cdot \begin{bmatrix} 1 & 0 \\ 0 & 0.99 \end{bmatrix}$, working with a determinant of 2×2 submatrices gives a better result. Since

$$\left| \det \begin{bmatrix} \frac{1}{3}\sqrt{3} & \frac{1}{2}\sqrt{2}+\varepsilon \\ \frac{1}{3}\sqrt{3} & -\frac{1}{2}\sqrt{2} \end{bmatrix} \right| > \left| \det \begin{bmatrix} \frac{1}{3}\sqrt{3}+\varepsilon & 0 \\ \frac{1}{3}\sqrt{3} & \frac{1}{2}\sqrt{2}+\varepsilon \end{bmatrix} \right|,$$

a block DEIM variant with block size 2 picks the more appropriate indices 2 and 3. Additionally, as noted in Section 2.4.3, the DEIM scheme attempts to minimize the quantity $\|(S^\top U)^{-1}\|$, where S is an index selection matrix. From our simple example above, we have that

$$\left\| \left(\begin{bmatrix} \frac{1}{3}\sqrt{3} & \frac{1}{2}\sqrt{2}+\varepsilon \\ \frac{1}{3}\sqrt{3} & -\frac{1}{2}\sqrt{2} \end{bmatrix} \right)^{-1} \right\| < \left\| \left(\begin{bmatrix} \frac{1}{3}\sqrt{3}+\varepsilon & 0 \\ \frac{1}{3}\sqrt{3} & \frac{1}{2}\sqrt{2}+\varepsilon \end{bmatrix} \right)^{-1} \right\|.$$

This implies that the rows selected by the block DEIM schemes we will discuss yield a smaller quantity $\|(S^\top U)^{-1}\|$ compared to the DEIM procedure.

The outline of the chapter is as follows: Section 4.2 describes how the DEIM scheme can be blocked using either the concept of the maximum determinant or volume of submatrices or a column-pivoted QR factorization. We discuss the newly proposed block DEIM algorithms and their computational complexities

and state a well-known error bound for the approximation. Section 4.3 reports the results from numerical experiments evaluating the computational efficiency and approximation quality of the various block DEIM procedures. Section 4.4 summarizes this chapter's key points and results. Some main properties of the methods are summarized in Table 4.1.

Table 4.1: Overview of the various variants with their properties. All methods have the SVD as a basis. The quantity η appears in error upper bounds and is discussed in Section 4.2.4.

Method \ Properties	SVD	Block	Adapt.	Tunable	Greediness	η	Speed
Standard DEIM	+	−	−	−	High	Medium	Moderate
B-MaxVol (Section 4.2.1)	+	+	−	+	Low	Low	Fast
B-RRQR (Section 4.2.2)	+	+	−	+	Low	Low	Fast
B-Adaptive (Section 4.2.3)	+	+	+	+	Medium	Medium	Variable

4.2 BLOCK DEIM

This section introduces three new block variants of the DEIM procedure. We combine the index selection algorithms discussed in Section 2.4 to design our block DEIM algorithms. The block DEIM algorithms select a b-size set of indices at each step using a block of singular vectors and then update the subsequent block of vectors using the oblique projection technique in the standard DEIM procedure. The main difference between the block DEIM and the traditional DEIM scheme resides in the b-size indices selection. In the standard DEIM, we select the indices by processing one singular vector at a time, each iteration step produces an index whilst the block version processes a block of singular vectors at a time, and each step provides an index set of size b. As a result, this may lead to a different selection of indices. For ease of presentation, we assume that the number of singular vectors k is a multiple of the block size b and $2 \leq b < k$.

4.2.1 *Block DEIM based on maximum volume*

As the first block variant, we introduce a block DEIM-MaxVol procedure (B-DEIM-MaxVol). For the standard DEIM method, each next index is picked greedily based on the maximal absolute value of an oblique projected singular vector. For this block variant, in every step, we greedily pick a fixed number (b) of indices based on an (approximate) maximal volume (absolute value of determinant) of a $b \times b$ submatrix of the projected singular vectors. To this end, we exploit the

MaxVol scheme [50] described in Chapter 2 to efficiently find a submatrix with the approximately maximal determinant (with a given tolerance δ) in a thin-tall matrix. This gives the pseudocode as displayed in Algorithm 8.

Algorithm 8: Block DEIM index selection based on MaxVol

Data: $U \in \mathbb{R}^{m\times k}$, $V \in \mathbb{R}^{n\times k}$, $k \leq \min(m,n)$, block size b (with $b \mid k$), convergence tolerance δ (default 0.01)
Result: Indices $\mathbf{p}, \mathbf{s} \in \mathbb{N}_+^k$ with non-repeating entries

1 **for** $j = 1, \ldots, k/b$ **do**
2 $\quad \mathbf{s}((j-1)b+1:jb) = \mathsf{MaxVol}(U(:,(j-1)b+1:jb), \delta)$
3 $\quad \mathbf{p}((j-1)b+1:jb) = \mathsf{MaxVol}(V(:,(j-1)b+1:jb), \delta)$
4 $\quad$ Let cols $= jb+1:jb+b$
5 $\quad U(:,\text{cols}) = U(:,\text{cols}) - U(:,1:jb)\cdot(U(\mathbf{s},1:jb) \setminus U(\mathbf{s},\text{cols}))$
6 $\quad V(:,\text{cols}) = V(:,\text{col}) - V(:,1:jb)\cdot(V(\mathbf{p},1:jb) \setminus V(\mathbf{p},\text{cols}))$
7 **end**

In Lines 5 and 6 of Algorithm 8, we use the oblique projection technique as in the standard DEIM scheme to update the subsequent blocks of singular vectors.

We will now describe how Algorithm 8 selects the row indices, the selection of the column indices follows similarly. Algorithm 8 starts from the leading-b dominant left singular vectors U_b, and the first set of index vector $\mathbf{s}_b$ corresponds to the b row indices of the dominant submatrix in U_b, which is obtained by applying Algorithm 2 to U_b. Let $\mathbf{s} = [\mathbf{s}_b]$, $S_b = I_m(:,\mathbf{s}_b)$, and define an oblique projection operator as $\mathbb{S}_b = U_b(S_b^\top U_b)^{-1}S_b^\top$.

Suppose we have $(j-1)b$ indices, so that

$$\mathbf{s}_{(j-1)b} = \begin{bmatrix} s_1 \\ \vdots \\ s_{(j-1)b} \end{bmatrix}, \quad S_{(j-1)b} = I_m(:,\mathbf{s}_{(j-1)b}), \quad U_{(j-1)b} = [\mathbf{u}_1, \ldots, \mathbf{u}_{(j-1)b}],$$

and

$$\mathbb{S}_{(j-1)b} = U_{(j-1)b}\left(S_{(j-1)b}^\top U_{(j-1)b}\right)^{-1} S_{(j-1)b}^\top.$$

Let $U_{jb} = U(:, jb+1:jb+b)$. Compute the residual $E_{jb} = U_{jb} - \mathbb{S}_{(j-1)b}U_{jb}$ (see Line 5 of the algorithm), and select the next set of b indices by applying Algorithm 2 to E_{jb}. It is worth noting that, using this oblique projection operator $\mathbb{S}_{(j-1)b}$ on U_{jb} ensures that the $\mathbf{s}_{(j-1)b}$ entries in E_{jb} are zero, which guarantees nonrepeating indices. At the end of the iteration, the algorithm returns a column and row index set of size k.

The following are two potential benefits of Algorithm 8 compared to standard DEIM.

- Approximation-wise, the greedy selection is not column-by-column, but carried out on a block of columns, possibly leading to a better pick of indices. Since the value of b is modest (typically 2), this procedure is still very affordable.
- Computationally, the block procedure may have some benefits over the vector-variant, for instance in the work for the oblique projection.

The computational cost of this algorithm is dominated by two calls of the MaxVol procedure and the block updates. The initialization step of Algorithm 2 requires a permutation of the input matrix, which can be done via the LU factorization. Given an $n \times b$ matrix, the LU decomposition requires $\mathcal{O}(nb^2)$ operations. Furthermore, the dominant cost of each iteration in the MaxVol algorithm is the multiplication of an $n \times b$ matrix and a $b \times b$ matrix, for a cost of $\mathcal{O}(nb^2)$ operations. Let $\kappa \in \mathbb{N}_+$ denote the number of iterations performed. The computational complexity of the MaxVol procedure is $\mathcal{O}(\kappa nb^2)$. The cost of the two calls of MaxVol is $\mathcal{O}(\kappa(m+n)b^2)$ and the block updates cost $\mathcal{O}((m+n)kb)$. Given that we have k/b iterations in Algorithm 8, the total cost of the B-DEIM-MaxVol algorithm is $\mathcal{O}((m+n)(\kappa bk + k^2))$. A crude bound on the number of iterations in Algorithm 2 is $\kappa \leq (\log|\det(\widehat{U}_{\mathsf{dom}})| - \log|\det(\widehat{U}_{\mathsf{ini}})|)/\log(1+\delta)$, where $\widehat{U}_{\mathsf{ini}}$ is the submatrix at the initialization step and $\widehat{U}_{\mathsf{dom}}$ is the dominant submatrix in Algorithm 2, respectively [50].

4.2.2 *Block DEIM based on RRQR*

The MaxVol algorithm seeks to find a good approximation of a submatrix with the maximum volume in a given input matrix. It is noted in [34, Remark 2.3] that the pivoting in an RRQR factorization can be interpreted as a greedy volume maximizing scheme. Practical experience shows that the index selection via a column-pivoted QR factorization may be more computationally efficient than the MaxVol method.

Additionally, Khabou et al. [67] propose a block LU factorization with panel rank-revealing pivoting (LU-PRRP) procedure by employing a strong rank-revealing QR (RRQR) panel factorization. The algorithm computes the block LU-PRRP as follows: at each step of the block factorization, a block of columns is factored by performing a strong RRQR factorization [53] on its transpose. Pivoting is done by applying the permutation matrix returned on the entire original matrix

followed by an update of the trailing matrix. As noted in [83], the DEIM index selection procedure is equivalent to the index selection of partially pivoted LU decomposition.

With this knowledge, we propose an alternative block DEIM variant (B-DEIM-RRQR), which combines the components of the standard DEIM scheme and an RRQR factorization, implemented in Algorithm 9.

Algorithm 9: Block DEIM index selection based on RRQR

Data: $U \in \mathbb{R}^{m\times k}$, $V \in \mathbb{R}^{n\times k}$, $k \leq \min(m,n)$, block size b (with $b \mid k$)
Result: Indices $\mathbf{p}, \mathbf{s} \in \mathbb{N}_+^k$ with non-repeating entries

for $j = 1, \ldots, k/b$ **do**
  Perform a column-pivoted QR on $U(:,(j-1)b+1:jb)^\top$ and $V(:,(j-1)b+1:jb)^\top$, giving the permutations $\Pi_{\mathbf{s}}$ and $\Pi_{\mathbf{p}}$
  $\mathbf{s}((j-1)b+1:jb) = \Pi_{\mathbf{s}}(1:b)$
  $\mathbf{p}((j-1)b+1:jb) = \Pi_{\mathbf{p}}(1:b)$
  Let cols $= jb+1:jb+b$
  $U(:,\text{cols}) = U(:,\text{cols}) - U(:,1:jb)\cdot(U(\mathbf{s},1:jb) \setminus U(\mathbf{s},\text{cols}))$
  $V(:,\text{cols}) = V(:,\text{col}) - V(:,1:jb)\cdot(V(\mathbf{p},1:jb) \setminus V(\mathbf{p},\text{cols}))$
end

During each step of the index selection process, we use a block size, b, and a block of singular vectors. Firstly, we perform a QR decomposition with column pivoting on the transpose of the leading-b singular vectors, selecting indices corresponding to the first b pivots as the initial set of indices. We then update the next block of singular vectors based on the updating method used in the B-DEIM-Maxvol algorithm and perform a column-pivoted QR on the transpose of the updated block (repeat these two steps until all k indices are selected).

The B-DEIM-RRQR may be viewed as a hybrid standard DEIM and QDEIM scheme [34]. The QDEIM selects k indices by performing one column-pivoted QR while our B-DEIM-RRQR algorithm selects the k indices by performing k/b rounds of column-pivoted QR. Note that when the block size $b = k$, this B-DEIM-RRQR algorithm is just the QDEIM scheme.

The cost of the B-DEIM-RRQR scheme is dominated by two QR factorizations and block updates. Given an $n \times b$ matrix, a QR factorization requires $\mathcal{O}(nb^2)$ operations hence the cost of the two QR decompositions here is $\mathcal{O}((m+n)b^2)$ and the block updates cost $\mathcal{O}((m+n)kb)$. Since there are k/b iterations in Algorithm 9, the total cost of the B-DEIM-RRQR scheme is therefore $\mathcal{O}((m+n)(kb+k^2))$.

4.2.3 *Adaptive block DEIM*

To combine the strength of standard DEIM with a block version, we also consider adaptive choices for the block size b. In particular, we propose the following variant called AdapBlock-DEIM: perform block DEIM if for the singular vector $\mathbf{v}_j$ being considered the two largest elements are (nearly) equal (see Example 4.1), i.e.,

$$\begin{cases} \text{block DEIM} & \text{if } |(\mathbf{v}_j)_i| \approx |(\mathbf{v}_j)_\ell| \quad \text{for } i \neq \ell, \\ \text{standard DEIM} & \text{otherwise.} \end{cases}$$

In Algorithm 10, we show an implementation of the adaptive block DEIM using the block DEIM variants discussed in Section 4.2. The parameter ρ is the desired lower bound on the ratio $|(\mathbf{v}_j)_i| \,/\, |(\mathbf{v}_j)_\ell|$ for $i \neq \ell$. We note that although our criterion for switching from a standard DEIM scheme to a block DEIM method is based on how close the two largest entries (magnitude) in the vector being considered are, other criteria can be used.

4.2.4 *Error bounds*

The following proposition restates a known theoretical error bound for a CUR approximation, which holds for the block DEIM algorithms proposed in this chapter. A detailed constructive proof is in [83]; we provide the necessary details here for the reader's convenience. Let $P \in \mathbb{R}^{n\times k}$ and $S \in \mathbb{R}^{m\times k}$ be matrices with some columns of the identity indexed by the indices selected by employing any of the three block DEIM algorithms.

Proposition 4.2. *[83, Thm. 4.1] Given $A \in \mathbb{R}^{m\times n}$ and a target rank k, let $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{n\times k}$ contain the leading k left and right singular vectors of A, respectively. Suppose $C = AP$ and $R = S^\top A$ are of full rank, and $V^\top P$ and $S^\top U$ are nonsingular. Then, with $M = C^+ A R^+$, a rank-k CUR decomposition constructed by either of the block DEIM schemes presented in this chapter satisfies*

$$\|A - CMR\| \leq (\eta_{\mathbf{s}} + \eta_{\mathbf{p}})\, \sigma_{k+1} \quad \textit{with} \quad \eta_{\mathbf{s}} < \sqrt{\tfrac{nk}{3}}\, 2^k\,, \quad \eta_{\mathbf{p}} < \sqrt{\tfrac{mk}{3}}\, 2^k,$$

where $\eta_{\mathbf{p}} = \|(V^\top P)^{-1}\|$, $\eta_{\mathbf{s}} = \|(S^\top U)^{-1}\|$.

Algorithm 10: Adaptive block DEIM index selection

Data: $U \in \mathbb{R}^{m\times k}$, $V \in \mathbb{R}^{n\times k}$, $k \le \min(m,n)$, block size b, ρ (default 0.95), tolerance δ (default 0.01), method (MaxVol or QR)
Result: Indices $\mathbf{p}, \mathbf{s} \in \mathbb{N}_+^k$ with non-repeating entries

$j = 1$
while $j \le k$ **do**
  if $j > 1$ **then**
    $\widetilde{\mathbf{u}} = U(\mathbf{s}(1:j-1), 1:j-1) \setminus U(\mathbf{s}(1:j-1), j)$
    $U(:,j) = U(:,j) - U(:,1:j-1)\cdot\widetilde{\mathbf{u}}$
  end if
  $[\mathbf{u}, \mathsf{ind}] = \mathsf{sort}(|U(:,j)|)$ (in descending order)
  if $(j+b-1 > k)$ **or** $(\mathbf{u}(2) < \rho\cdot\mathbf{u}(1))$ **then** $\mathbf{s}(j) = \mathsf{ind}(1)$; $j = j+1$
  else
    cols $= j+b-1$
    if $j > 1$ **then**
      $\widetilde{U} = U(:, j+1:\text{cols})$; $\widehat{U} = U(\mathbf{s}(1:j-1), j+1:\text{cols}))$
      $\widetilde{U} = \widetilde{U} - U(:,1:j-1)\cdot(U(\mathbf{s}(1:j-1), 1:j-1) \setminus \widehat{U}$
      $U(:, j+1:\text{cols}) = \widetilde{U}$
    end if
    if method = MaxVol
      $\mathbf{s}(j:\text{cols}) = \mathsf{MaxVol}(U(:, j:\text{cols}), \delta)$
    else
      Perform a column-pivoted QR on $U(:, j:\text{cols})^\top$, giving
      permutation $\Pi_{\mathbf{s}}$
      $\mathbf{s}(j:\text{cols}) = \Pi_{\mathbf{s}}(1:b)$
    end if
    $j = j+b$
  end if
end
Repeat the procedure on V to get indices $\mathbf{p}$

Proof. Let $\mathbb{P} = P(V^\top P)^{-1}V^\top$ and $\mathbb{S} = U(S^\top U)^{-1}S^\top$ be oblique projectors. Note that $V^\top\mathbb{P} = V^\top$ and $\mathbb{S}U = U$, implying that $V^\top(I-\mathbb{P}) = 0$ and $(I-\mathbb{S})U = 0$. Using $M = C^+AR^+$, we have

$$\begin{aligned}\|A - CMR\| &= \|A - CC^+AR^+R\| = \|(I-CC^+)A + CC^+A(I-R^+R)\| \\ &\le \|(I-CC^+)A\| + \|CC^+\|\,\|A(I-R^+R)\|.\end{aligned}$$

Leveraging the fact that CC^+ is an orthogonal projector, $\|CC^+\| = 1$ and [83, Lemma 4.1 and 4.2]

$$\begin{aligned}\|(I - CC^+)A\| &\leq \|A(I - \mathbb{P})\| = \|A(I - VV^\top)(I - \mathbb{P})\| \\ &\leq \|(V^\top P)^{-1}\| \, \|A(I - VV^\top)\|, \\ \|A(I - R^+R)\| &\leq \|(I - \mathbb{S})A\| = \|(I - \mathbb{S})(I - UU^\top)A\| \\ &\leq \|(S^\top U)^{-1}\| \, \|(I - UU^\top)A\|,\end{aligned}$$

we have that

$$\|A - CMR\| \leq \|(V^\top P)^{-1}\| \, \|A(I - VV^\top)\| + \|(S^\top U)^{-1}\| \, \|(I - UU^\top)A\|.$$

Since U and V contain the leading k left and right singular vectors, respectively, $\|(I - UU^\top)A\| = \|A(I - VV^\top)\| = \sigma_{k+1}$. Hence

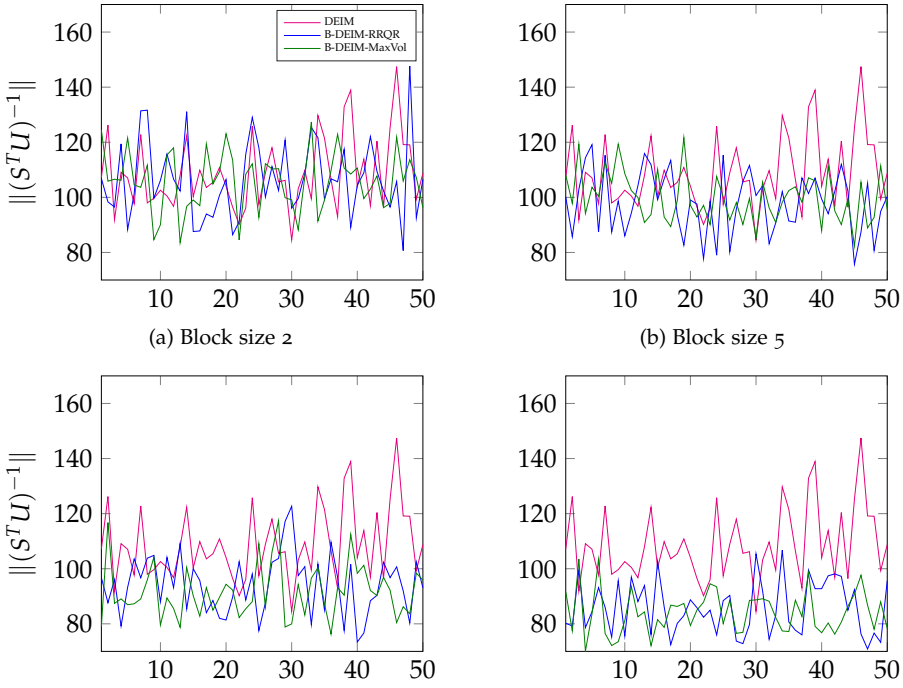
$$\|A - CMR\| \leq (\|(V^\top P)^{-1}\| + \|(S^\top U)^{-1}\|) \cdot \sigma_{k+1}.$$

■

Proposition 4.2 suggests that the quality of an index selection method may be assessed using the error constants $\eta_{\mathbf{s}}$ and $\eta_{\mathbf{p}}$ [83]. Fig. 4.1 illustrates the difference in the values of $\eta_{\mathbf{s}} = \|(S^\top U)^{-1}\|$ computed by DEIM, B-DEIM-RRQR, and B-DEIM-MaxVol using 50 randomly generated matrices with orthonormal columns of size 10000×100. In most cases, the block DEIM variants provide smaller values of $\eta_{\mathbf{s}}$; this may be an indication that the indices picked by the block DEIM variants are better for approximating than those selected by the DEIM scheme. We also observe that the values of $\eta_{\mathbf{s}}$ become smaller as we increase the block sizes.

4.3 NUMERICAL EXPERIMENTS

In this section, we conduct several sets of illustrative experiments to show the effectiveness of the block DEIM variants, i.e., B-DEIM-MaxVol, B-DEIM-RRQR, and AdapBlock-DEIM, proposed in this chapter using synthetic and real data sets. We evaluate the algorithms by applying them to data analysis problems in several application domains: recommendation system analysis, model order reduction, economic modeling, and optimization. We use real-world sparse and dense data matrices with sizes ranging from small to large scale. An overview is presented in Table 4.2. We compare the performance of our algorithms for constructing a CUR approximation with three state-of-the-art deterministic algorithms: DEIM [83],



(a) Block size 2

(b) Block size 5

(c) Block size 10

(d) Block size 20

Figure 4.1: Comparison of the value $\eta_s = \|(S^\top U)^{-1}\|$ in DEIM, B-DEIM-RRQR, and B-DEIM-MaxVol with different block sizes using 50 random matrices with orthonormal columns of size 10000×100.

QDEIM [34], and MaxVol [50]. All these algorithms require the leading k right and left singular vectors to construct a rank-k CUR factorization. We use these two evaluation criteria: the rank-k approximation relative error $\|A - CMR\| \,/\, \|A\|$; the computational efficiency, i.e., the runtime scaling for the rank parameter k. Here, the runtime measures the time it takes each algorithm to select the desired number of column and row indices. We do not consider the run time for computing the singular vectors since all the methods we consider require the SVD. However, it is important to note that the total cost of selecting the indices may be dominated by the computational cost of the SVD. Our experiments are not meant to be exhaustive; however, they provide clear evidence that the block

DEIM schemes proposed in this chapter may provide a comparable low-rank approximation while being computationally more efficient.

In the implementation, we perform the column-pivoted QR factorization and the truncated SVD using the MATLAB built-in functions qr and svds [3], respectively (and svd for small cases). For the MaxVol algorithm, we use a MATLAB implementation by [69] made available on GitHub[1]. Unless otherwise stated, in all the experiments we use as default block size $b = 5$ for small/mid-size matrices and $b = 10$ for large-scale matrices, the AdapBlock-DEIM method parameter $\rho = 0.95$, and the MaxVol scheme convergence tolerance $\delta = 0.01$.

Table 4.2: Various examples and dimensions considered.

Exp.	Domain	Matrix	m	n
1	Recommendation system	Dense	14116	100
2	Economic modeling	Sparse	29610	29610
3	Optimization	Sparse	29920	29920
4	Model order reduction	Sparse	23412	23412
5	Structural engineering	Sparse	22044	22044
6	Synthetic	Dense	2000	4000

Experiment 4.3. In this first set of experiments, we aim to evaluate how our proposed block-DEIM variants compared with the existing deterministic methods mentioned earlier on a small matrix. Our data set is from the recommendation system analysis domain, where one is usually interested in making service or purchase recommendations to users. One of the most common techniques for recommendation systems is collaborative filtering, which involves recommending to users items that customers with similar preferences liked in the past. The Jester data set is often used as a benchmark for recommendation system research [47]. This data matrix consists of 73421 users and their ratings for 100 jokes. We only consider users who have ratings for all 100 jokes resulting in a 14116×100 matrix. We center the matrix by subtracting the mean of each column from all entries in that column.

Based on the observations from Fig. 4.2, we can conclude that the block DEIM methods generally provide slightly more accurate approximations compared to state-of-the-art methods. It is also important to note that the error of the MaxVol and QDEIM approximations do not always decrease monotonically as the rank k increases. When considering the runtime, both the B-DEIM-MaxVol and B-DEIM-RRQR algorithms demonstrate significantly lower computational times compared

1 github.com/bokramer/CURERA/blob/master/maxvol.m

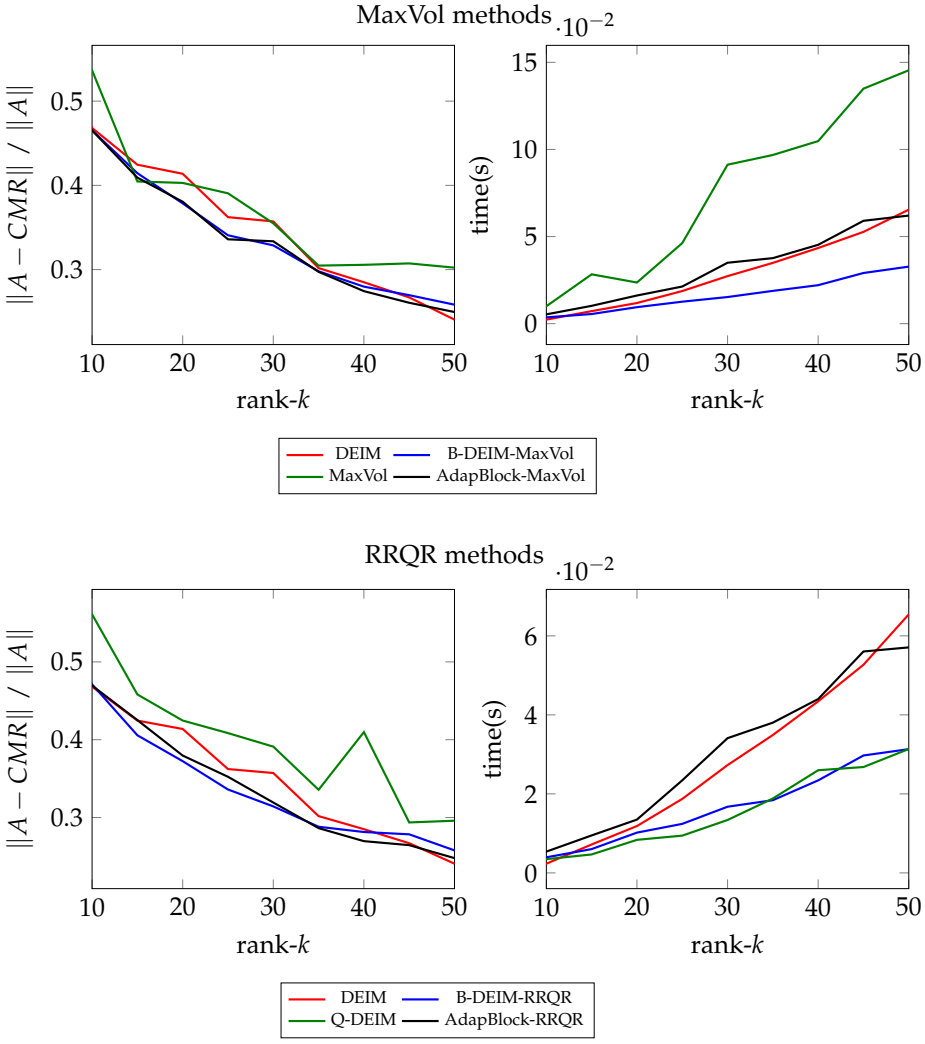


Figure 4.2: Relative approximation errors (left) and runtimes (right) as a function of k for the block DEIM CUR approximation algorithms compared with some standard CUR approximation algorithms using the Jester data set.

to the original DEIM scheme. In these small/mid-scale experiments, the adaptive variants of the block DEIM methods do not seem to improve runtimes compared

to the DEIM procedure. There could be several reasons for this observation: the adaptive variants of block DEIM methods involve additional computations and operations compared to the standard DEIM procedure. These additional steps may introduce computational overhead that offsets the potential gains in runtime. In small/mid-scale scenarios, the overhead might outweigh the benefits. On the other hand, it is evident that the B-DEIM-MaxVol algorithm and its adaptive variant are more efficient than the standard MaxVol approach. By using the block DEIM variants, we gain improvements in both accuracy and speed compared to the standard MaxVol method. Additionally, the B-DEIM-RRQR method proves to be equally efficient as the QDEIM procedure while providing a more accurate approximation.

Experiment 4.4. In the subsequent series of experiments, we turn our attention to evaluating the performance of our proposed block-DEIM variants when dealing with large-scale data. With a block size of $b = 10$ selected for this particular set of experiments, our primary goal is to gain insights into how our block-DEIM approaches tackle the challenges presented by large-scale data sets and to assess their effectiveness and efficiency in this context. To conduct these evaluations, we use a set of standard test matrices specifically designed for sparse matrix problems. These data matrices are sourced from the publicly available SuiteSparse Matrix Collection [26]. The diverse nature of these matrices allows us to assess the effectiveness of our approaches across various problem domains.

The first test matrix, referred to as g7jac100, is derived from the "Overlapping Generations Model" used to study the social security systems of the G7 nations. It is a sparse matrix with dimensions 29610×29610 and contains 335972 numerically nonzero entries. Notably, this matrix has a low rank of 21971. The second matrix, named net100, originates from an optimization problem. It has dimensions of 29920×29920 and contains 2033200 numerically nonzero entries. Similar to the previous matrix, net100 also possesses a low rank, specifically 26983. The Abacus-shell-ld matrix, associated with model order reduction, has dimensions 23412×23412 and represents a rank-2048 structure. It contains 218484 nonzero entries. Lastly, we have pkustk01, a symmetric positive-definite matrix derived from a civil engineering problem. This matrix has dimensions of 22044×22044, a low rank of 3732, and consists of 979380 nonzero entries.

In the case of large-scale data sets, similar to the small/mid-scale experiments, in Figs. 4.3, 4.4, 4.5, and 4.6 the block DEIM variants maintain comparable reconstruction errors as the existing methods. This finding aligns with our observations from the small/mid-scale experiments. However, there are notable differences in terms of algorithm efficiency. Unlike the small/mid-scale cases, where the

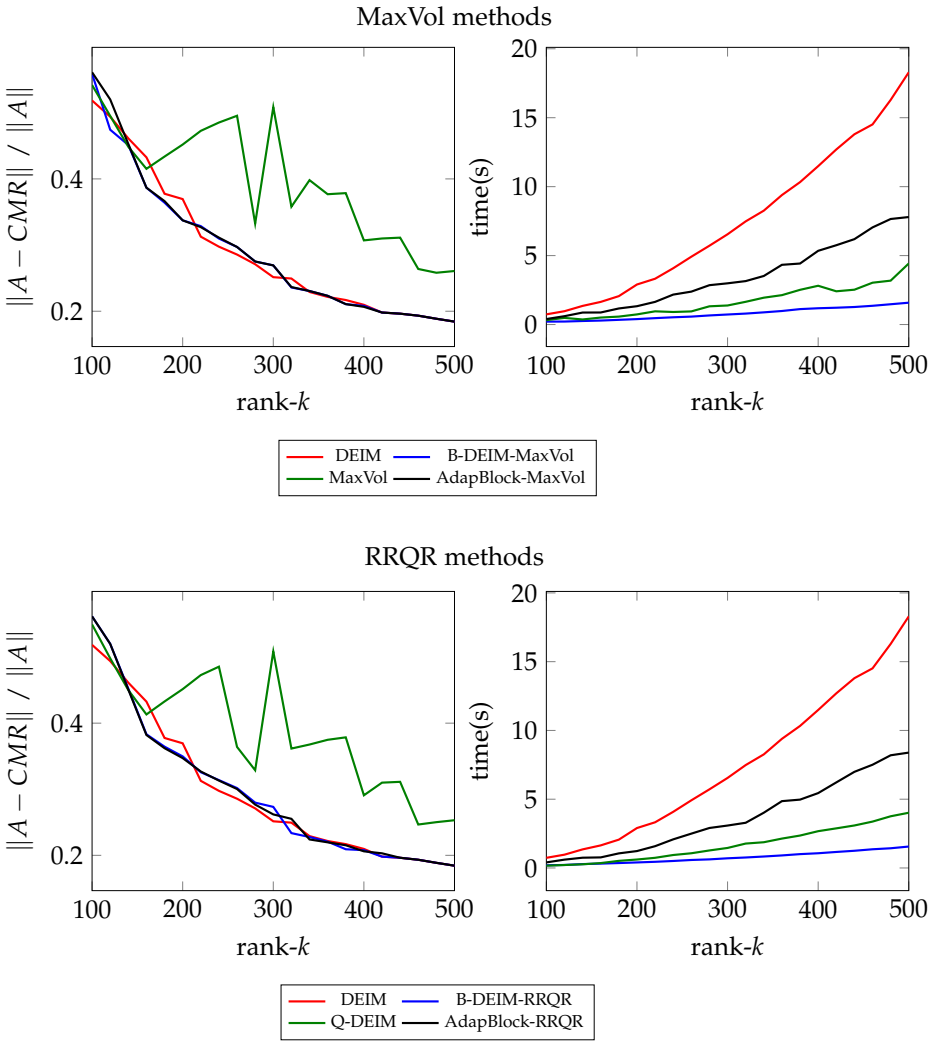


Figure 4.3: Relative approximation errors (left) and runtimes (right) as a function of k for the block DEIM CUR approximation algorithms compared with some standard CUR approximation algorithms using the g7jac100 sparse matrix.

adaptive variants have similar runtimes as the DEIM scheme, in the large-scale experiments, the adaptive variants demonstrate better computational efficiency

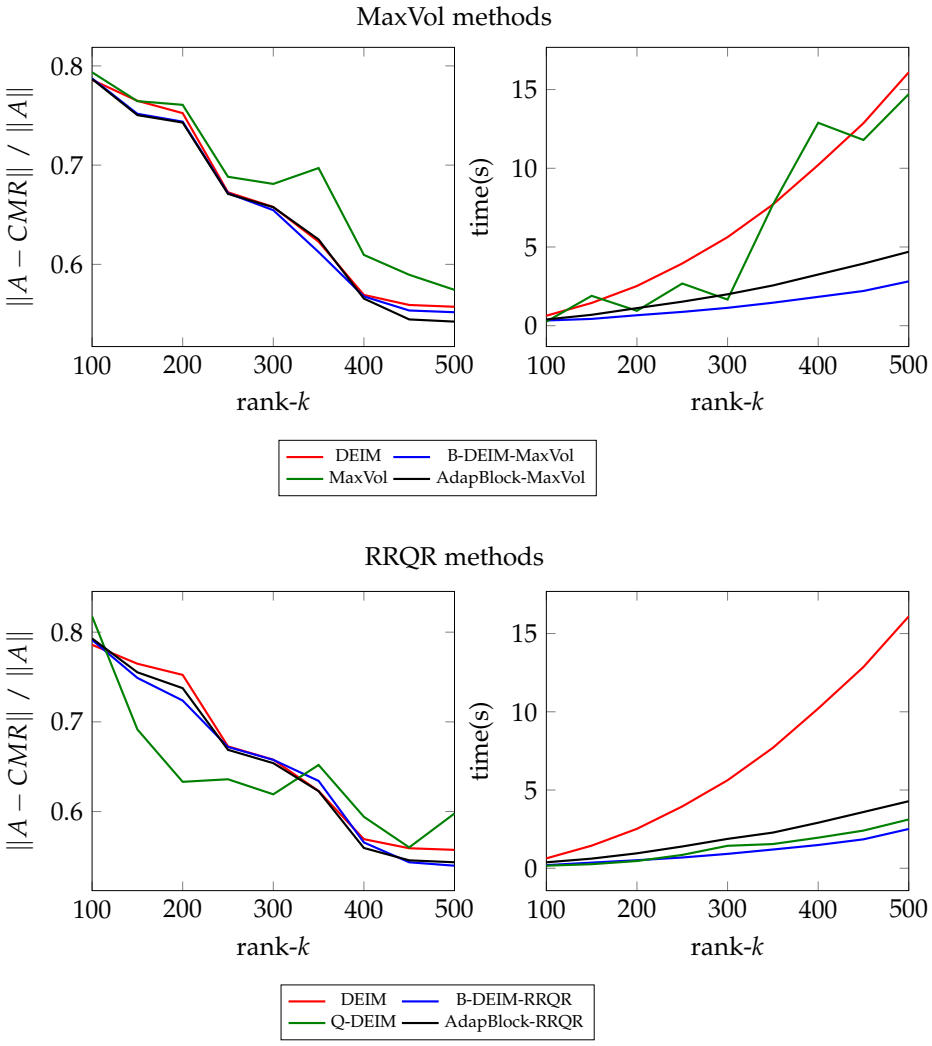


Figure 4.4: Relative approximation errors (left) and runtimes (right) as a function of k for the block DEIM CUR approximation algorithms compared with some standard CUR approximation algorithms using the net100 sparse matrix.

than the standard DEIM scheme. On the other hand, the B-DEIM-MaxVol and B-DEIM-RRQR schemes showcase better speed efficiency overall. These block

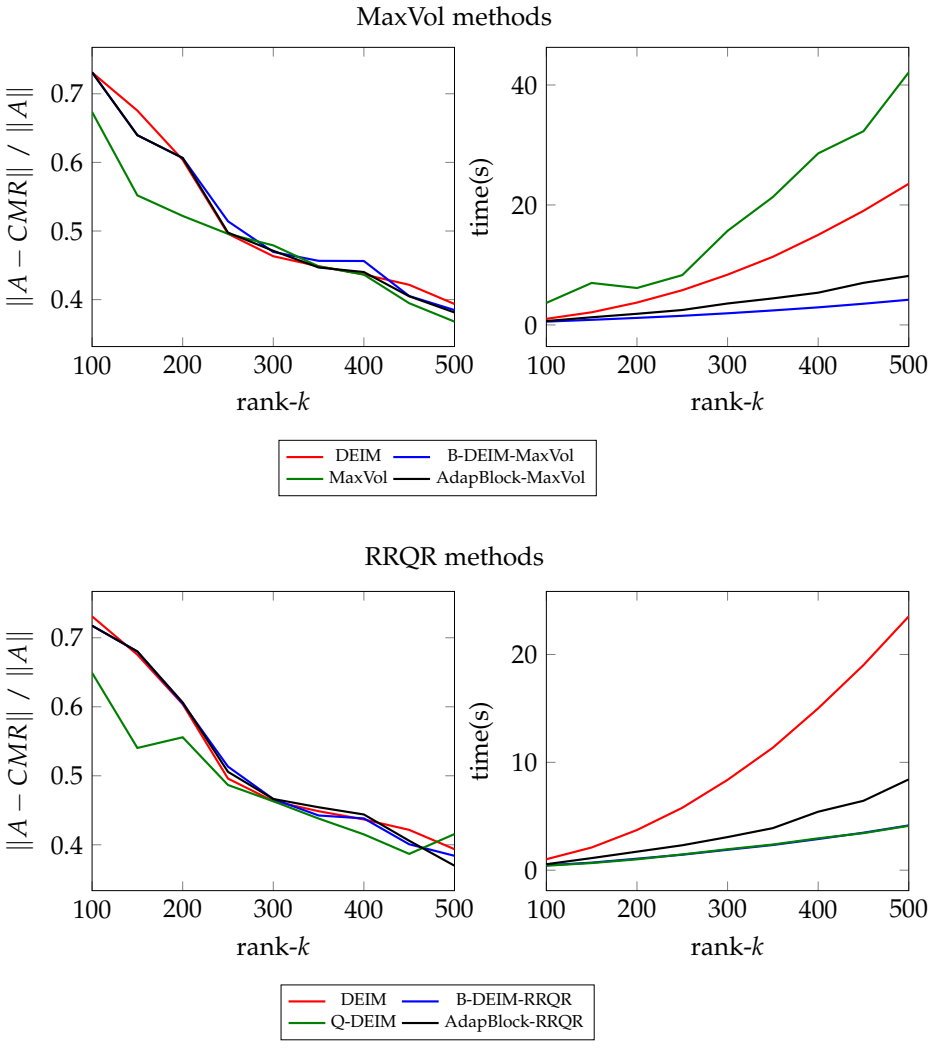


Figure 4.5: Relative approximation errors (left) and runtimes (right) as a function of k for the block DEIM CUR approximation algorithms compared with some standard CUR approximation algorithms using the Abacusa-shell-ld sparse matrix.

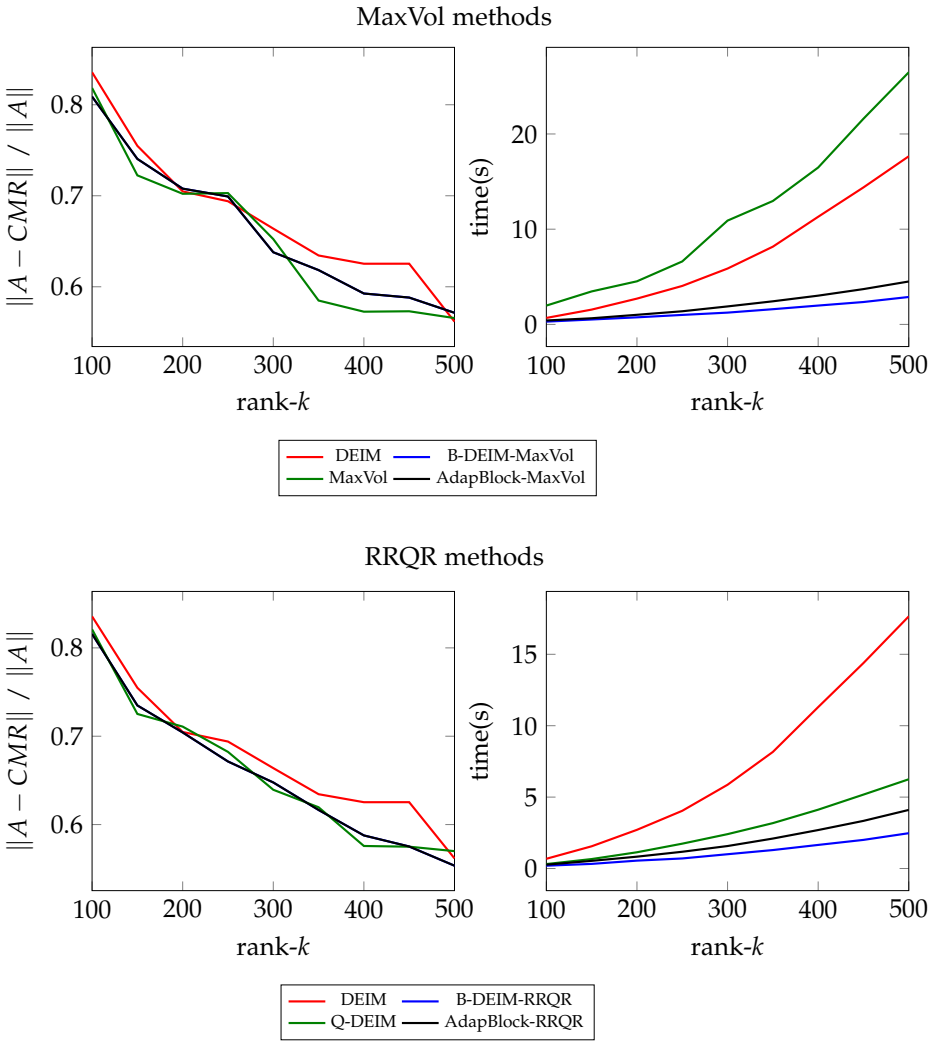


Figure 4.6: Relative approximation errors (left) and runtimes (right) as a function of k for the block DEIM CUR approximation algorithms compared with some standard CUR approximation algorithms using the pkustk01 sparse matrix.

DEIM variants prove to be effective in achieving a balance between accuracy and computational efficiency in the context of large-scale data. Consistent with previ-

ous findings, the MaxVol algorithm generally remains the least efficient method, with one exception in the case of the gijac100 data set. These results highlight the importance of considering the specific characteristics and requirements of the data sets when selecting an appropriate algorithm.

Experiment 4.5. Following the experiments in [93], our test matrix in this experiment is a full-rank data set $A \in \mathbb{R}^{2000\times 4000}$ that has the structure of the SVD, i.e., $A = U\Sigma V^\top$. The matrices U and V have random orthonormal columns obtained via a QR factorization of a random Gaussian matrix, and the diagonal matrix Σ has entries that are logspace ranging from 1 to 10^{-3}.

Using two of the block DEIM algorithms proposed: the B-DEIM-MaxVol and B-DEIM-RRQR, in Fig. 4.7, we investigate how varying block sizes, i.e., $b = (2, 5, 10, 20)$ may affect their approximation quality and computational efficiency. For each fixed block size, maintaining the properties of A, we generate five different test cases and compute the averages of the evaluation criteria for the range of k values.

We observe that for increasing block sizes, both algorithms become considerably faster. On the other hand, the approximation quality of the varying block sizes may not degrade significantly. In this experiment, given the various values of k, the errors are almost similar irrespective of the block size.

4.4 FINAL CONSIDERATIONS

This chapter presents various block variants of the discrete empirical interpolation method for computing CUR decompositions. We exploit the advantages of the classical DEIM procedure, a column-pivoted QR decomposition, and the concept of maximum determinant or volume of submatrices to develop these block variants. We have then presented a version of the block DEIM, which allows for an adaptive choice of block size.

We perform the following procedures in the block DEIM based on RRQR; at each iteration step, we compute a QR factorization with column pivoting on the transpose of a block of singular vectors to obtain the indices corresponding to the first b columns. Then, we update the next block of vectors using the interpolatory projection technique in the DEIM algorithm (repeat these two steps until all indices are selected). A similar procedure is used in the block DEIM based on MaxVol; the difference here is instead of using a column-pivoted QR decomposition, we use the MaxVol method.

Numerical experiments illustrate that the accuracy of a CUR factorization using the newly proposed block DEIM procedures is comparable to the classical

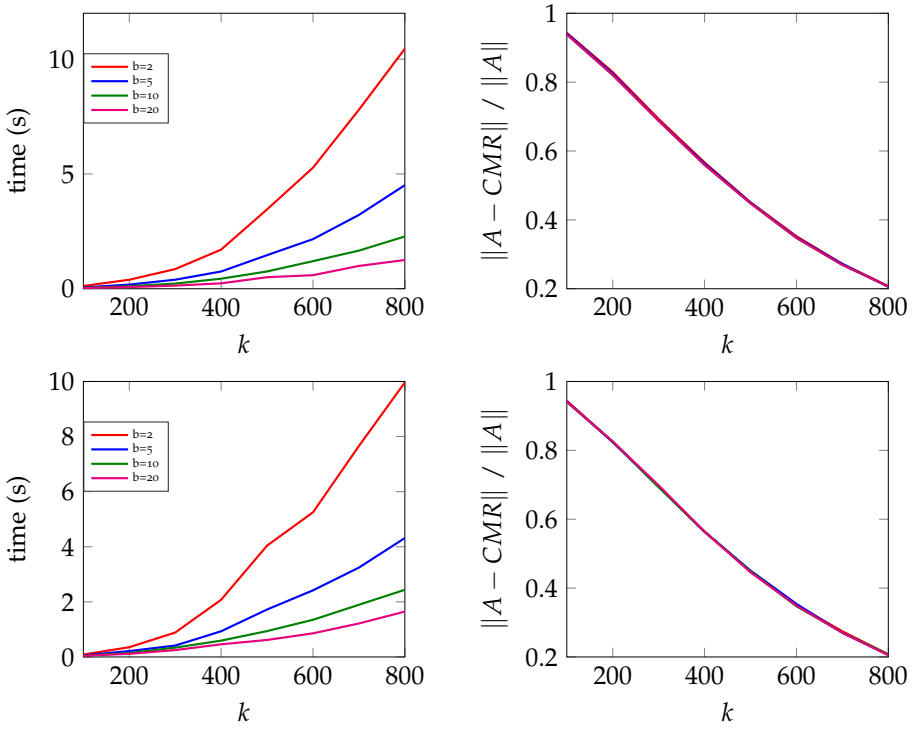


Figure 4.7: Runtimes and average approximation errors for the B-DEIM-RRQR (up) and B-DEIM-MaxVol (down) CUR approximation algorithms as a function of k for varying block sizes using large matrices of size 2000×4000.

DEIM, MaxVol, and QDEIM schemes. The experiments also demonstrate that the block variants, regarding computational speed, may have an advantage over the standard DEIM and MaxVol algorithms. Relative to the QDEIM algorithm, the B-DEIM-RRQR scheme sometimes yields lesser approximation errors while maintaining comparable runtimes. Using the B-DEIM-RRQR and B-DEIM-MaxVol methods, we have also illustrated how increasing the block size improves the speed of the algorithms but may not necessarily degrade the approximation quality significantly. Table 4.1 displays a schematic overview of some properties of the various methods.

5

A DEIM-CUR FACTORIZATION WITH ITERATIVE SVDS

A CUR factorization is often used as a substitute for the singular value decomposition, especially when a concrete interpretation of the singular vectors is challenging. Moreover, if the original data matrix possesses properties like nonnegativity and sparsity, a CUR decomposition can better preserve them. An essential aspect of this approach is the methodology used for selecting a subset of columns and rows from the original matrix. This study investigates the effectiveness of *one-round sampling* and iterative subselection techniques and introduces new iterative subselection strategies based on iterative SVDs. Our contribution aims to improve the approximation quality of the DEIM scheme by iteratively invoking it in several rounds, in the sense that we select subsequent columns and rows based on the previously selected ones. That is, we modify A after each iteration by removing the information that has been captured by the previously selected columns and rows. We also discuss how iterative procedures for computing a few singular vectors of large data matrices can be used with the new iterative subselection strategies.

Adapted from [44]

5.1 ADAPTIVE SAMPLING FOR COLUMN SUBSET SELECTION PROBLEM

The iterative subselection strategies proposed in this chapter are related to the so-called *adaptive sampling* for column subset selection. In this section, we provide an overview of the adaptive sampling technique proposed by Deshpande et al. [28]. The authors introduce a probabilistic method that iteratively selects a subset of columns in multiple rounds to construct a rank-k approximation of a matrix. This approach has been demonstrated to provide improved accuracy and flexibility compared to *one-round sampling* methods. One-round sampling methods refer to selection schemes that obtain all k columns in a single round.

The adaptive sampling method of [28] as summarized in Algorithm 11 involves alternating between two steps in each round: selecting a subset of columns and updating the probability distribution over all columns. The selection of columns in each round is influenced by the columns picked in previous rounds. Suppose we aim to select k subsets of columns from matrix A. The process begins with an initial probability distribution and randomly selects $c < k$ columns

to form a matrix C. The selection of columns is based on the norms of the columns, as described in [28, 37]. Each column j is chosen with a probability $\text{pr}_i^{(j)} = \|E_{i-1}^{(j)}\|^2 \,/\, \|E_{i-1}\|_F^2$ (as in Line 5 of Algorithm 11). After selecting c columns, the probabilities are updated based on the chosen columns, and c new columns are sampled and added to the matrix C. This iterative process continues until all k columns are selected. Note that assigning zero probability to previously selected indices, as described in Line 4, represents a sampling without replacement strategy.

Algorithm 11: Adaptive sampling for column subset selection [28]

Data: $A \in \mathbb{R}^{m\times n}$, target rank k, # rounds t, columns per round c
Result: $C \in \mathbb{R}^{m\times tc}$

1 $\mathbf{p} = [\,]$; $\quad E_0 = A$
2 **for** $i = 1, \ldots, t$ **do**
3 **for** $j = 1, \ldots, n$ **do**
4 **if** $j \in \mathbf{p}$ **then** $\text{pr}_j^{(i)} = 0$ (sample without replacement)
5 **else** $\text{pr}_i^{(j)} = \|E_{i-1}^{(j)}\|^2 \,/\, \|E_{i-1}\|_F^2$
6 **end**
7 $\mathbf{p}_i$ = set of c indices sampled according to pr_i
8 $\mathbf{p} = [\mathbf{p}\ \ \mathbf{p}_i]$
9 $C = A(:, \mathbf{p})$; $\quad E_i = A - CC^+A$
10 **end**

The authors present a detailed explanation and theoretical analysis of this adaptive sampling technique, emphasizing its advantages and diverse applications [28]. The algorithm improves the accuracy of a CUR decomposition compared to one-round sampling methods as demonstrated in [28, 29, 79, 94]. Moreover, it allows for flexibility by accommodating different criteria for selecting column and row subsets based on specific problem requirements, which we will also discuss in Section 5.2. In their approach (Algorithm 11), a constant number of columns is selected per iteration, and the residual is computed as $E = A - CC^+A$.

Nevertheless, one notable drawback of this adaptive sampling technique is its probabilistic nature, introducing randomness into the index selection process. This randomness may sacrifice predictability to some extent. Applying the algorithm multiple times to the same data may lead to inconsistent factorizations, yielding varying CUR decompositions across runs. This lack of predictability may hamper the reproducibility of the decomposition. Moreover, the unpredictable nature may

impact downstream tasks reliant on the CUR decomposition, potentially leading to inconsistent outcomes. As a result, the practical use of a CUR decomposition by this type of probabilistic algorithm may be compromised.

To address this limitation, in the next section, we introduce a modified approach based on the DEIM scheme to implement the iterative subselection algorithm proposed by Deshpande et al. [28] and also propose a new iterative subselection strategy. By incorporating the DEIM scheme, our method provides a deterministic technique for iterative subselection of column indices, in contrast to the original methods that employed a probabilistic approach [10, 28, 29, 79, 94]. This deterministic nature enhances the reproducibility of the CUR decomposition, making it more suitable for practical applications.

5.2 SMALL-SCALE DEIM TYPE CUR WITH ITERATIVE SVDS

In this section, we introduce new index-picking schemes for constructing a CUR decomposition. The standard DEIM scheme determines an SVD of A once, after which the indices are picked iteratively "locally optimal". The new methods that we present now compute an SVD in every iteration. The algorithms adaptively select columns and rows of A in several rounds. In each iteration, we modify A by removing the information that has been captured by the previously selected columns and rows. The time complexities of the various methods after t rounds are summarized in Table 5.1. This includes the computational time for computing an SVD and an updated A (residual) matrix in every round.

Table 5.1: Summary of the dominant work of the different algorithms after t rounds. The time complexity column excludes the computational cost of the DEIM scheme as it is approximately the same for all algorithms.

Method	Matrix	SVD	Time		
			svd	X or M	Residual (E)
CADP-CX (Algorithm 12) **DADP-CX** (Algorithm 13) **DADP-CUR** (Algorithm 14)	Small	Full	$\mathcal{O}(tmn^2)$	$\mathcal{O}(tmnk)$	$\mathcal{O}(tmnk)$
Large: DADP-CX (Algorithm 16)	Large	Few	$\mathcal{O}(mn \cdot \mathrm{nr}_{\mathrm{in}})$	$\mathcal{O}(mnk)$	–

The complexity estimates and the other descriptions are similar for the first three algorithms. It is important to highlight that when constructing a CUR factorization using Algorithms 12 and 13, one needs to perform almost twice the number of SVDs, X, and E, compared to what is required by Algorithm 14. For

the large-scale algorithm, estimating the precise time complexity of computing a low-rank SVD using an iterative method (refer to Algorithm 15), as done in Algorithm 16, can be challenging. The iterative approach involves a series of matrix-vector multiplications and orthogonalization steps. The computational cost of the Lanczos algorithm is typically dominated by matrix-vector multiplications, which is $\mathcal{O}(mn)$ times the total number of inner iterations (denoted by nr_{in} in the table) needed. It is worth noting that we do not explicitly compute the residual in Algorithm 16 as done in Algorithms 12, 13, and 14.

5.2.1 *A DEIM based adaptive sampling for column subset selection*

We present a deterministic variant of the iterative subselection scheme discussed in Section 5.1. The newly proposed algorithm (**CADP-CX**) builds upon the original adaptive sampling algorithm [28] by leveraging the benefits of the DEIM technique. The procedure is summarized in Algorithm 12. The method involves iteratively selecting a constant number of column indices, denoted as c, from A in multiple rounds. We start by computing the leading $c < k$ singular vectors of A. Next, we apply the DEIM scheme (Algorithm 3) to these singular vectors, resulting in the first set of c indices. We then update A by computing the residual matrix E using the interpolative decomposition (as described in Line 7 of Algorithm 12). Next, we compute the leading c singular vectors of E and apply the DEIM procedure again to obtain the next set of c indices. This process is repeated until we have selected all k required indices.

Algorithm 12: DEIM based adaptive sampling for column subset selection

Data: $A \in \mathbb{R}^{m\times n}$, target rank k, columns per round c (with $c \mid k$)
Result: $C \in \mathbb{R}^{m\times k}$

1 $\mathbf{p} = [\,]; \quad E_0 = A$
2 **for** $i = 1, \ldots, k/c$ **do**
3 Compute $[\sim, \sim, V] = \mathsf{svd}(E_{i-1})$
4 $V(\mathbf{p}, :) = 0$
5 $\mathbf{p}_i = \mathsf{deim}(V(:, 1:c))$ (Iteratively pick c indices)
6 $\mathbf{p} = [\mathbf{p} \;\; \mathbf{p}_i]$
7 $C = A(:, \mathbf{p}); \quad X = C \backslash A; \quad E_i = A - CX$
8 **end**

One consequence of using the DEIM scheme here is that each column of A has a chance of being selected in subsequent iterations, even if it was selected

in a previous iteration. This could result in the selection of previously chosen columns in subsequent iterations. Line 4 of Algorithm 12 is a possible sample without-replacement strategy that can alleviate this problem. Since the DEIM procedure selects the index corresponding to the entry of the largest magnitude in a given vector, when these indices are zeroed out after being chosen, it guarantees that they will not be selected again.

With regards to the memory and computational complexity, computing the residual in Line 7 involves a full iteration over the matrix, which has a space complexity of $\mathcal{O}(mn)$. Given that we use the DEIM procedure and select c columns per iteration, in terms of computational complexity, a full SVD requires $\mathcal{O}(mn^2)$, one run of the DEIM algorithm requires $\mathcal{O}(mc^2)$, and computing the residual in Line 7 costs $\mathcal{O}(mnk)$. The overall time complexity after t rounds is $\mathcal{O}(tmn^2 + tmc^2 + tmnk)$.

5.2.2 *A new iterative subselection method*

In Algorithm 13, we introduce a new iterative subselection strategy (**DADP-CX**) for a CUR factorization, which differs from the method employed in our new Algorithm 12 and the adaptive sampling procedures in previous works [10, 28, 79, 94]. In contrast to the previous strategy, which selects a fixed number of columns or rows in each iteration, this new strategy dynamically adjusts the selection schedule based on the decay of the singular values of the data (the relative magnitudes of the singular values).

The motivation behind this new approach is to adapt the subselection process according to the significance of the singular values. By considering the decay pattern of the singular values, we can prioritize the selection of columns or rows that contribute the most to the data's overall structure and information. The decay of singular values provides valuable information about the significance of different components in the data. By leveraging this information, the iterative subselection strategy can adapt to the specific characteristics of the data and prioritize the selection of columns or rows that contribute the most to its structure. This adaptability allows for a more data-driven selection process.

With a user-defined threshold $\delta \in (0, 1]$, the small-scale version of our method begins by computing the leading singular vectors corresponding to the singular values greater than the threshold multiplied by the largest singular value of A, i.e., all $\sigma_i > \delta \cdot \sigma_1$. Let b denote the number of singular values satisfying this condition. Additionally, we introduce an extra parameter ℓ to establish an upper limit on the number of indices per round, taking into account the number of singular

Algorithm 13: Singular value decay-based iterative DEIM with one-sided projected residual

Data: $A \in \mathbb{R}^{m \times n}$, desired rank k, threshold parameter $\delta \in (0, 1]$, upper limit ℓ

Result: Low-rank CUR decomposition $A_k = A(:, \mathbf{p}) \cdot M \cdot A(\mathbf{s}, :)$

1 Set $E = A$; $\mathbf{p} = [\,]$; $\mathbf{s} = [\,]$
2 **while** $\mathsf{length}(\mathbf{p}) < k$ **do**
3 Compute $[\sim, \Sigma, V] = \mathsf{svd}(E)$
4 $V(\mathbf{p}, :) = 0$
5 Let b be the last index $i \leq k - \mathsf{length}(\mathbf{p})$ with $\sigma_i > \delta\sigma_1$
6 $c = \min(b, \ell)$; $\mathbf{p}_c = \mathsf{deim}(V(:, 1:c))$
7 $\mathbf{p} = [\mathbf{p}\ \ \mathbf{p}_c]$; $C = A(:, \mathbf{p})$; $X = C \backslash A$
8 $E = A - CX$ (Update matrix)
9 **end**
10 Repeat steps 1–9 on $A^\top$ to find the row indices $\mathbf{s}$
11 $M = A(:, \mathbf{p}) \setminus (A \,/\, A(\mathbf{s}, :))$

values that exceed the threshold. Consequently, we select the first $c = \min(b, \ell)$ column indices denoted by $\mathbf{p}_c$ by applying the DEIM scheme to the leading c right singular vectors (V_c).

Subsequently, we proceed to construct an interpolative decomposition using the chosen column indices and compute the residual matrix E by subtracting this approximation from A, i.e., $E = A - CC^{+}A$. To determine the next set of indices, we repeat the aforementioned process on E. Thus, we compute the leading singular vectors of E corresponding to singular values greater than δ times the largest singular value of E and repeat the procedure mentioned earlier. As previously mentioned, since the DEIM scheme selects indices corresponding to entries with the largest magnitude, we set the entries in the right singular vectors that correspond to the previously selected indices to zero. This prevents the selection of duplicate indices. We continue this procedure until all k indices are selected. We expect that the multiple passes through A would lead to a reduced approximation error. It is worth mentioning that in Line 10, there is no need to compute the initial SVD of $A^\top$ since we can store the initial left singular vectors from the SVD of A.

In addition to the new selection strategy described in Algorithm 13, we also define an alternative way to compute the residual in the index selection process, which is presented in Algorithm 14. The newly proposed iterative subselection

algorithms (Algorithms 12 and 13) and existing adaptive sampling procedures such as those outlined in [10, 28, 29, 79, 94] define the residual as the error incurred by projecting the matrix A onto either the column space of C or the row space of R, i.e., $E = A - CC^{+}A$ or $E = A - AR^{+}R$, respectively. In contrast, this new method (**DADP-CUR**) defines the residual as the error incurred by simultaneously projecting A onto both the column space of C and the row space of R. This means computing a CUR factorization at each step using only the selected columns and rows.

Algorithm 14: Singular value decay-based iterative DEIM with two-sided projected residual

Data: $A \in \mathbb{R}^{m\times n}$, desired rank k, threshold parameter $\delta = (0, 1]$, upper limit ℓ

Result: Low-rank CUR decomposition $A_k = A(:, \mathbf{p}) \cdot M \cdot A(\mathbf{s}, :)$

Set $E = A$; $\mathbf{p} = [\,]$; $\mathbf{s} = [\,]$

while $\mathsf{length}(\mathbf{p}) < k$ **do**

  $[U, \Sigma, V] = \mathsf{svd}(E)$

  Set $V(\mathbf{p}, :) = 0$, $U(\mathbf{s}, :) = 0$

  Let b be the last index $i \leq k - \mathsf{length}(\mathbf{p})$ with $\sigma_i > \delta\,\sigma_1$

  $c = \min(b, \ell)$; $\mathbf{p}_c = \mathsf{deim}(V(:, 1:c))$; $\mathbf{s}_c = \mathsf{deim}(U(:, 1:c))$

  $\mathbf{p} = [\mathbf{p}\ \ \mathbf{p}_c]$, $\mathbf{s} = [\mathbf{s}\ \ \mathbf{s}_c]$

  $M = A(:, \mathbf{p}) \setminus (A\ /\ A(\mathbf{s}, :))$

  $E = A - A(:, \mathbf{p}) \cdot M \cdot A(\mathbf{s}, :)$ (Update matrix)

end

By considering the simultaneous projection onto the column and row spaces, we aim to use a residual that provides a more accurate representation of the error in the CUR factorization. It takes into account the combined effect of selecting specific columns and rows on capturing the underlying structure and information in the data. This approach offers several potential advantages. It allows for a more comprehensive assessment of the error in the index selection process, considering the contributions from both the columns and rows. Furthermore, it ensures that the residual accurately reflects the approximation quality obtained by a CUR factorization using the selected columns and rows. Additionally, it has the potential to reduce computational costs compared to Algorithm 13, as the latter approach involves performing nearly twice the number of SVDs required by Algorithm 14. Without showing specifics, it is worth mentioning that Algorithm 12 can be adapted using the newly defined residual.

Note that when $\delta = 0$, both Algorithms 13 and 14 are equivalent to the DEIM-type CUR factorization. In terms of time complexity, suppose we need t iterations in Algorithms 13 and 14 to select all k columns and rows. The cost of solving E is $\mathcal{O}(tmnk)$. The cost of an SVD and one run of the DEIM scheme are $\mathcal{O}(mn^2)$ and $\mathcal{O}(nc^2)$, respectively, where c is the maximum number of columns selected per iteration. Therefore, the overall cost of the algorithms is $\mathcal{O}(tmn^2 + t(m+n)c^2 + tmnk)$. However, constructing C and R using Algorithm 13 requires two runs of it. Thus, its cost is almost twice that of Algorithm 14.

For small matrices, these iterative subselection techniques may be worthwhile as the costs are modest and the quality of the approximations may increase. However, these schemes may be especially interesting for large matrices, for which an SVD may be too expensive, and iterative methods are used to compute the left and right singular vectors. We will study this situation in the next section.

5.3 LARGE-SCALE DEIM TYPE CUR WITH ITERATIVE SVDS

For large-scale matrices, taking an SVD every round in Algorithms 12, 13, and 14 will usually be prohibitively expensive. Indeed, even one (reduced) SVD will be too costly, which means that the standard DEIM-type CUR decomposition is generally not affordable. However, the proposed algorithm is suitable for large-scale data, as approximating the largest singular vectors by iterative (Krylov) methods is usually a relatively easy task. Additionally, here, we do not explicitly compute the residual matrix as done in the proposed algorithms; this is done implicitly in the computation of the approximate singular vectors. Furthermore, instead of computing the full SVD as we do in Algorithms 12, 13, and 14, we now carry out:

1: Approximate $\widehat{U}$ and $\widehat{V}$ of E.

This can efficiently be carried out by an implicitly restarted version of Lanczos bidiagonalization (see, e.g., [3]). The idea is as follows. Let $k < \widehat{k}$ be the minimal and maximal dimension of the subspaces. We first carry out $\widehat{k}$ steps of Lanczos bidiagonalization summarized by the matrix equations

$$E\widehat{V}_{\widehat{k}} = \widehat{U}_{\widehat{k}} B_{\widehat{k}}, \quad E^\top \widehat{U}_{\widehat{k}} = \widehat{V}_{\widehat{k}} B_{\widehat{k}}^\top + \beta_{\widehat{k}} \widehat{\mathbf{v}}_{\widehat{k}+1} \mathbf{e}_{\widehat{k}}^\top,$$

where $B_{\widehat{k}}$ is bidiagonal. The singular values of $B_{\widehat{k}}$ are approximations to those of E, and the singular vectors lead to approximations to those of E. With the SVD $B_{\widehat{k}} = W\widehat{\Sigma}Z^\top$, we get

$$E(\widehat{V}_{\widehat{k}} Z) = (\widehat{U}_{\widehat{k}} W)\widehat{\Sigma}, \quad E^\top (\widehat{U}_{\widehat{k}} W) = (\widehat{V}_{\widehat{k}} Z)\widehat{\Sigma} + \beta_{\widehat{k}} \widehat{\mathbf{v}}_{\widehat{k}+1} (W^\top \mathbf{e}_{\widehat{k}})^\top,$$

For any upper triangular matrix $\widehat{\Sigma}$ an elegant implicit restart procedure is possible; here $\widehat{\Sigma}$ is even diagonal. Order the singular values in the desired way; in this case nonincreasingly. Partition the transformed basis, redefining $\widehat{U}_k$ and $\widehat{V}_k$:

$$(\widehat{U}_{\widehat{k}}W) =: [\widehat{U}_k \ \ \widehat{U}_{\widehat{k}-k}], \quad (\widehat{V}_{\widehat{k}}Z) =: [\widehat{V}_k \ \ \widehat{V}_{\widehat{k}-k}], \quad \widehat{\Sigma} = \begin{bmatrix} \widehat{\Sigma}_k & \\ & \widehat{\Sigma}_{\widehat{k}-k} \end{bmatrix}, \tag{5.1}$$

and redefine $B_k = \widehat{\Sigma}_k$, $\beta_{k+1} := \beta_{\widehat{k}+1}$, $\widehat{\mathbf{v}}_{k+1} := \widehat{\mathbf{v}}_{\widehat{k}+1}$, and $\mathbf{f}_k := W^\top \mathbf{e}_{\widehat{k}}$. We can now conveniently restart from the decomposition

$$E\widehat{V}_k = \widehat{U}_k B_k, \quad E^\top \widehat{U}_k = \widehat{V}_k B_k^\top + \beta_k \widehat{\mathbf{v}}_{k+1}\mathbf{f}_k^\top.$$

The pair $(\widehat{U}_k, \widehat{V}_k)$ may be viewed as a pair of approximate invariant spaces with error $\|\mathbf{f}_k\|$. The spaces are expanded with Lanczos bidiagonalization to dimension $\widehat{k}$, after which the selection procedure is carried out again. This scheme is repeated until the quantify $\|\mathbf{f}_k\|$ is sufficiently small. We summarize the method in Algorithm 15. Note that MATLAB built-in function svds is a different implementation of a related technique.

Algorithm 15: Implictly restarted Lanczos bidiagonalization [3]

Data: $E \in \mathbb{R}^{m\times n}$, desired rank k, initial vector $\mathbf{v}_1$, minimum and maximum dimension $k < \widehat{k}$, tolerance tol

Result: Approximation to k largest singular triplets $(\sigma_i, \mathbf{u}_i, \mathbf{v}_i)$, giving best low-rank approximation $E_k = \widehat{U}_k\widehat{\Sigma}_k\widehat{V}_k^\top$

Generate $E\widehat{V}_k = \widehat{U}_k B_k$, $E^\top\widehat{U}_k = \widehat{V}_k B_k^\top + \beta_{k+1}\widehat{\mathbf{v}}_{k+1}\mathbf{f}_k^\top$

for $i = 1, 2, \ldots$ **do**

  Expand to $E\widehat{V}_{\widehat{k}} = \widehat{U}_{\widehat{k}} B_{\widehat{k}}$, $E^\top\widehat{U}_{\widehat{k}} = \widehat{V}_{\widehat{k}} B_{\widehat{k}}^\top + \beta_{\widehat{k}+1}\widehat{\mathbf{v}}_{\widehat{k}+1}\mathbf{f}_{\widehat{k}}^\top$

  Determine SVD $B_{\widehat{k}} = W\widehat{\Sigma}Z^\top$

  Partition according to (5.1), restart with $\widehat{U}_k$, $\widehat{V}_k$,

    redefining $B_k := \widehat{\Sigma}_k$, $\beta_{k+1} := \beta_{\widehat{k}+1}$, $\widehat{\mathbf{v}}_{k+1} := \widehat{\mathbf{v}}_{\widehat{k}+1}$, $\mathbf{f}_k := W^\top \mathbf{e}_{\widehat{k}}$

  Stop if $\|\mathbf{f}_k\| \le$ tol

end

In Algorithm 16 we provide the large-scale version of Algorithm 13 by employing Algorithm 15. Without showing details, it is worth noting that this can also be adapted for Algorithms 12 and 14. It is important to note that the threshold parameter δ and the upper limit ℓ on the number of indices to be selected per round are incorporated within the implementation of Algorithm 15. The efficiency of this method is because for the procedure in Algorithm 15, we do not need

Algorithm 16: Large-scale: Singular value decay-based iterative DEIM with one-sided projected residual

Data: $A \in \mathbb{R}^{m \times n}$, desired rank k, threshold parameter $\delta \in (0, 1]$, upper limit ℓ
Result: Low-rank CUR decomposition $A_k = A(:, \mathbf{p}) \cdot M \cdot A(\mathbf{s}, :)$

1 Set $E = A$; $\mathbf{p} = [\,]$;
2 **while** $\mathsf{length}(\mathbf{p}) < k$ **do**
3 Compute $[\sim, \Sigma, V] = \mathsf{svds}(E)$ (Algorithm 15)
4 finding σ_b, the last index $i \leq k - \mathsf{length}(\mathbf{p})$ with $\sigma_i > \delta\,\sigma_1$ or
5 at most σ_ℓ. Let $c = \min(b, \ell)$
6 $V(\mathbf{p}, :) = 0$; $\mathbf{p}_c = \mathsf{deim}(V(:, 1 : c))$
7 $\mathbf{p} = [\mathbf{p}\ \ \mathbf{p}_c]$; $C = A(:, \mathbf{p})$
8 Update an incremental QR decomposition $C = QT$
9 $E@(\mathbf{x})\ \mathbf{y} = A\mathbf{x};\ \mathbf{y} = \mathbf{y} - Q(Q^\top \mathbf{y})$; (Update matrix implicitly)
10 with transpose $E^\top@(\mathbf{x})\ \mathbf{y} = \mathbf{x} - Q(Q^\top \mathbf{x});\ \mathbf{y} = A^\top \mathbf{y}$
11 **end**
12 Repeat steps 1–9 on $A^\top$ to find the row indices $\mathbf{s}$
13 $M = A(:, \mathbf{p}) \setminus (A\ /\ A(\mathbf{s}, :))$

the matrix E in explicit form; only matrix-vector products (MVs) with E and $E^\top$ are necessary (see Line 9). The routine of Algorithm 15 also takes several MVs that depend on the distribution of the singular value and the starting vector. The cost of computing the singular values and vectors in Line 3 depends on the total number of inner iterations of Algorithm 15. The number of iterations required by the Lanczos algorithm depends on the size of the matrix. In a matrix-vector product $E\mathbf{x}$ for a vector $\mathbf{x}$, the component $A\mathbf{x}$ costs $\mathcal{O}(mn)$ for a full matrix. In the case of a sparse matrix with d nonzeros entries per row, the cost reduces to $\mathcal{O}(md)$. The aggregated cost of Line 8 is only $\mathcal{O}(mk^2)$. In Line 9, the computation of $Q(Q^\top \mathbf{x})$ requires $\mathcal{O}(mk)$ operations. The cost of solving the least squares problem $M = C^+ A R^+$ would be $\mathcal{O}(mnk)$, which is relatively expensive. Nevertheless, it is important to highlight that this step is necessary for all CUR methods as the final step.

As previously mentioned, it is not necessary to compute the initial SVD of $A^\top$ in this case, as we can simply retain the initial left singular vectors obtained from the SVD of A. The value of δ will typically depend on the data set. A value close to 1 may be favorable for the approximation result but is more expensive since Algorithm 15 needs to be carried out approximately k times. However, having

$\delta = 1$ implies that we select one index per iteration, and thus, we need just the first right and left singular vectors of E corresponding to the largest singular value. We can reduce the computational cost by specifying an earlier convergence criterion for finding the approximate leading right and left singular vectors. We use Wedin's theorem for this. The theorem bounds the distance between subspaces and the proof is in (cf., e.g., [86, pp. 260–262]).

Theorem 5.1. *(Wedin's Theorem) Given $E \in \mathbb{R}^{m\times n}$, let*

$$[U_1\ U_2\ U_3]^\top\ E\ [V_1\ V_2] = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \\ 0 & 0 \end{bmatrix},$$

be the SVD of E (where the singular values are not necessarily nonincreasing). The singular subspaces of interest are in the column spaces of U_1 and V_1. Let the inexact/approximate singular subspaces be in the column spaces of $\widehat{U}_1$ and $\widehat{V}_1$ in the decomposition

$$[\widehat{U}_1\ \widehat{U}_2\ \widehat{U}_3]^\top\ \widehat{E}\ [\widehat{V}_1\ \widehat{V}_2] = \begin{bmatrix} \widehat{\Sigma}_1 & 0 \\ 0 & \widehat{\Sigma}_2 \\ 0 & 0 \end{bmatrix}.$$

Now let Φ be the matrix of canonical angles between $\mathrm{Range}(U_1)$ *and* $\mathrm{Range}(\widehat{U}_1)$*, and Θ be the matrix of canonical angles between* $\mathrm{Range}(V_1)$ *and* $\mathrm{Range}(\widehat{V}_1)$*. Given the residuals $F_1 = E\widehat{V}_1 - \widehat{U}_1\widehat{\Sigma}_1$, $F_2 = E^\top\widehat{U}_1 - \widehat{V}_1\widehat{\Sigma}_1$, suppose that there is a number $\alpha > 0$ such that*

$$\min|\sigma(\widehat{\Sigma}_1) - \sigma(\Sigma_2)| \geq \alpha \quad \textit{and} \quad \sigma_{\min}(\widehat{\Sigma}_1) \geq \alpha.$$

Then

$$\sqrt{\|\sin\Phi\|_F^2 + \|\sin\Theta\|_F^2} \leq \frac{\sqrt{\|F_1\|_F^2 + \|F_2\|_F^2}}{\alpha}.$$

Theorem 5.1 shows that the computed singular vectors extracted by the projection method are optimal up to the factor in the right-hand side of the above inequality. This implies that any change in the entries of the computed singular vectors is bounded by this factor. Note that Σ_2 is unknown. For our context, we use $\widehat{\Sigma}_2$ as an approximation to Σ_2. Since we are only concerned with approximating the first leading right singular vector $\widehat{\mathbf{v}}_1$, we approximate $\alpha \approx \widehat{\sigma}_1 - \widehat{\sigma}_2$. Let $m_1(\widehat{\mathbf{v}}_1)$ and $m_2(\widehat{\mathbf{v}}_1)$ denote the largest and second-largest entries in $\widehat{\mathbf{v}}_1$, respectively, and

let $\mathbf{f}_2 = E^\top \widehat{\mathbf{u}}_1 - \widehat{\sigma}_1 \widehat{\mathbf{v}}_1$ be the residual vector (associated with residual matrix F_2). The above svds routine results in $\mathbf{f}_1 = E\widehat{\mathbf{v}}_1 - \widehat{\sigma}_1 \widehat{\mathbf{u}}_1 = 0$ (associated with residual matrix F_1). The DEIM algorithm selects the index corresponding to the largest element in the magnitude of a vector. Therefore, when $\delta = 1$, one can set an early convergence criterion to find the first singular vector that corresponds to the largest singular value, using the following approximate bound:

$$m_1(\widehat{\mathbf{v}}_1) - m_2(\widehat{\mathbf{v}}_1) \lesssim 2\,\frac{\|\mathbf{f}_2\|}{\widehat{\sigma}_1 - \widehat{\sigma}_2}.$$

Remark 5.2. *In [83, Thm. 4.1], Sorensen and Embree provide a theoretical error bound that applies to a general class of CUR factorizations (see Proposition 4.2). We note that this bound also holds for our proposed methods in this chapter. A detailed constructive proof of this bound is in [83], but we provide the necessary details in Section 4.2.4 for the reader's convenience.*

5.4 NUMERICAL EXPERIMENTS

We conduct numerical experiments to evaluate the empirical performance of the DEIM scheme [83], the QDEIM procedure [34], the MaxVol method [50], and the iterative subselection techniques discussed in this chapter. The following are the iterative subselection methods we evaluate:

CADP-CX: refers to Algorithm 12.

DADP-CX: represents Algorithm 13.

DADP-CUR: corresponds to Algorithm 14.

CADP-CUR: denotes the adapted version of Algorithm 12 with the residual defined as $E = A - CMR$.

To assess the effectiveness of our algorithms, we test them on various data analysis tasks in different application domains, such as analyzing recommendation systems, categorizing text and retrieving information, and image compression. Our evaluation includes synthetic and real-world data matrices, both sparse and dense, with varying sizes ranging from small to large scale, which we summarize in Table 5.2. In the implementation, we perform the column-pivoted QR factorization and the reduced SVD using the MATLAB built-in functions qr and svd, respectively. For Algorithms 15 and 16, we use our implementation of the Lanczos bidiagonalization method of [3] by incorporating the threshold parameter δ and

upper limit ℓ on the number of singular vectors to be computed. Unless otherwise stated, in all the experiments we use as default the number of rounds $t = 10$, the parameter $\delta = 0.8$, and upper limit $\ell = k/10$.

Table 5.2: Various examples and dimensions considered.

Exp.	Domain	Matrix	m	n
1	Synthetic	Sparse	100000	300
2	Text categorization	Sparse	139	15210
3	Text categorization	Sparse	8293	18933
4	Image compression	Dense	13500	5000
5	Recommendation system	Dense	14116	100
6	Image compression	Dense	50000	3072

Experiment 5.3. In our first experiment, we investigate how different choices of δ and the number of rounds t affect the approximation accuracy of the various iterative subselection strategies. We use the relative approximation error $\|A - CMR\| \,/\, \|A\|$ as the evaluation metric. For this experiment just as in [83], we generate a sparse, nonnegative matrix $A \in \mathbb{R}^{m\times n}$, with $m = 100000$ and $n = 300$, of the form

$$A = \sum_{j=1}^{10} \frac{2}{j}\, \mathbf{x}_j\, \mathbf{y}_j^\top + \sum_{j=11}^{300} \frac{1}{j}\, \mathbf{x}_j\, \mathbf{y}_j^\top,$$

where $\mathbf{x}_j \in \mathbb{R}^m$ and $\mathbf{y}_j \in \mathbb{R}^n$ are sparse vectors with random nonnegative entries (i.e., $\mathbf{x}_j = \mathsf{sprand}(m, 1, 0.025)$ and $\mathbf{y}_j = \mathsf{sprand}(n, 1, 0.025)$.

From Fig. 5.1 we observe that increasing the number of rounds t or δ does not necessarily lead to a monotonic decrease in the approximation errors. The result implies that to get the optimum advantage of using the iterative subselection strategies one needs to carefully choose the parameter δ or the number of rounds. For this experiment, we also observe that using the residual $E = A - CMR$ instead of $A - CC^+A$ for the iterative subselection yields better approximation errors in the delta strategy while it produces worse approximation errors in the constant number of columns strategy.

Experiment 5.4. Our next experiment is to demonstrate that the iterative subselection techniques yield better approximation results than one-round sampling. We perform the experiment using four real data sets and report the relative approximation error $\|A - CMR\| \,/\, \|A\|$ of each algorithm on each data set.

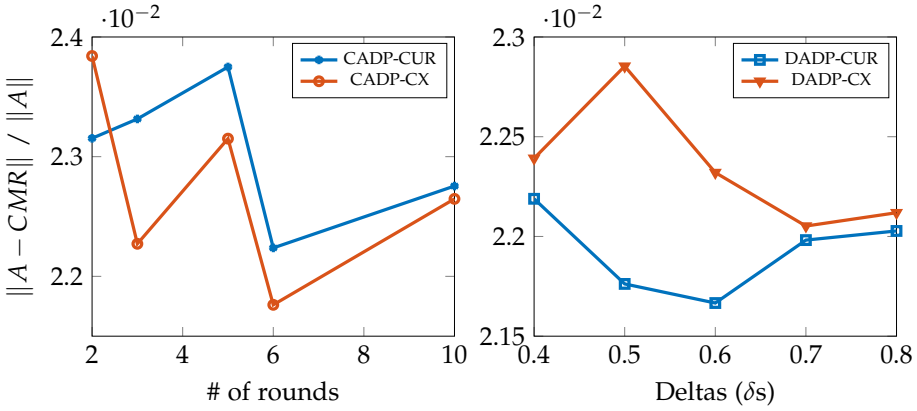


Figure 5.1: Relative approximation errors for the various iterative subselection DEIM CUR approximation algorithms for $k = 30$. The right figure represents selecting a constant number of columns and rows per iteration and the left is the delta strategy. In all cases, increasing the number of rounds or delta does not lead to a monotonic decrease in the approximation errors.

The first two data sets are relevant to applications in text categorization and information retrieval. In such data analysis problems, a "bag of words" approach is commonly employed to represent documents. We opt for the Reuters-21578 text categorization collection, which comprises documents that were featured on Reuters' newswire in 1987. This data set is extensively used as a benchmark in the text classification community, consisting of 21578 documents categorized into 135 categories. For our experiment, we use the preprocessed data set, which has 18933 unique terms and 8293 documents [12]. We normalize the rows of the sparse matrix, which has dimensions 8293×18933, to have a unit length. The second data set, the Internet term document data, is from the Technion Repository of Text Categorization Datasets (TechTC) [40]. We use the test set 26, which consists of a collection of 139 documents on two topics with 15210 terms describing each document[1]. As in [83], the 139×15210 TechTC matrix rows are scaled to have a unit 2-norm.

The third data set is the Gisette data [54]. Gisette is a handwritten digit recognition problem. The problem is to separate the highly confusable digits '4' and '9'. The digits have been size-normalized and centered in a fixed-size image of dimension 28×28. The resulting data set is of dimension 13500×5000.

1 http://gabrilovich.com/resources/data/techtc/

The next data set pertains to the recommendation system analysis field, where the primary objective is to provide service or purchase suggestions to users. Collaborative filtering is a commonly used technique in recommendation systems, which involves recommending items to users that were previously liked by customers with comparable preferences. The Jester data set is frequently employed as a benchmark in recommendation system research [47]. The data set comprises 73421 users and their ratings for 100 jokes. We limit our analysis to users who have provided ratings for all 100 jokes, resulting in a 14116×100 matrix. We normalize the matrix by subtracting the mean of each column from all the entries in that column.

From Fig. 5.2, we can see that in all cases our iterative subselection-based CUR algorithm has a lower approximation error than all the *one-round* deterministic index selection algorithms considered. We also observe that the approximation error of the QDEIM and the MaxVol techniques do not always decrease monotonically with increasing k values. We choose the number of rounds for the CADP-CUR and CADP-CX algorithms to be $t = 10$, and the parameter $\delta = 0.8$ and upper limit $\ell = k/10$ for the DADP-CUR and DADP-CX algorithms. The results of all four proposed iterative subselection algorithms are comparable.

Experiment 5.5. As stated in Section 5.3, when dealing with large-scale matrices, performing a full (even reduced) SVD in each iteration of algorithms 12, 13, and 14 can often become excessively costly. To evaluate the efficiency of the proposed algorithms, which compute the full SVD, compared to their respective large-scale versions that employ Lanczos bidiagonalization to find a limited number of singular vectors (refer to Algorithm 16), we conduct experiments using a large-scale data set. Specifically, we used the cifar-10 data set, which consists of 60000 color images sized 32×32 pixels, divided into ten different classes with 6000 images per class. The data set is divided into 50000 training set images and 10000 test set images. We focus on the training data set containing 50000 images. Each image in the data set has been reshaped into a 1D array of length 3072, resulting in a dense matrix of size 50000×3072. For our analysis, we approximate this matrix using a rank-100 approximation.

Table 5.3 presents the results obtained from running the various algorithms. We observe that the large-scale variants (i.e., the various adaptations of Algorithm 16), which employ an iterative method for computing a few SVDs, demonstrate higher efficiency while maintaining comparable approximation quality compared to the algorithms that compute the full SVD. Notably, both for the full SVD and the iterative SVD routines, the algorithms with the residual defined as $E = A - CMR$ exhibit greater efficiency than those with the residual computed

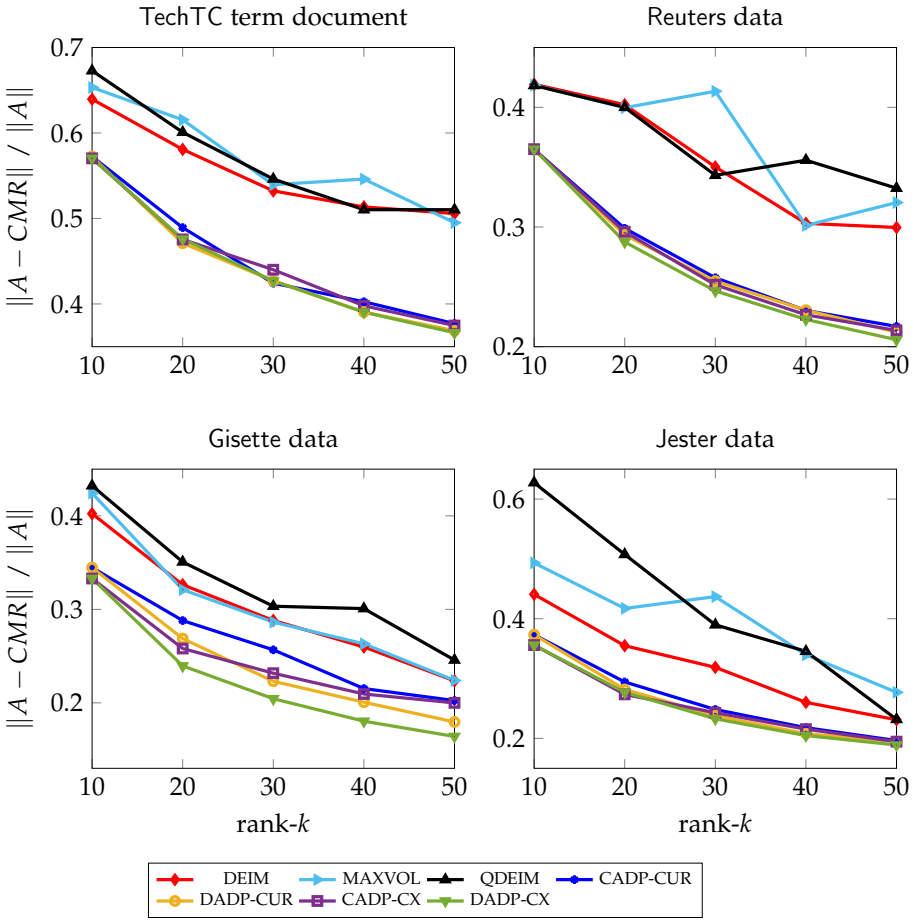


Figure 5.2: Relative approximation errors as a function of k for the various iterative subselection DEIM CUR approximation algorithms compared with some standard CUR approximation algorithms using real data sets.

as $E = A - CC^+A$. Therefore, our new approach to computing the residual for the iterative subselection proves to be more efficient than the existing method while maintaining similar approximation accuracy for this experiment.

Table 5.3: Comparison of large-scale iterative subselection algorithms (iterative method for computing few SVDs) and small-scale iterative subselection algorithms (full SVD computation) on the cifar-10 data set approximation.

Method	Full SVD		Iterative SVD	
	Relative error	Runtime (s)	Relative error	Runtime (s)
CADP-CUR	0.028	$3.24 \cdot 10^2$	0.028	$1.61 \cdot 10^2$
DADP-CUR	0.027	$5.06 \cdot 10^2$	0.027	$1.66 \cdot 10^2$
CADP-CX	0.026	$6.21 \cdot 10^2$	0.026	$5.40 \cdot 10^2$
DADP-CX	0.026	$1.15 \cdot 10^3$	0.026	$5.78 \cdot 10^2$

5.5 FINAL CONSIDERATIONS

New approaches for selecting subsets of columns and rows using iterative subselection strategies have been presented. The first one is a DEIM adaptation of the so-called adaptive sampling [28] for column subset selection. This procedure follows a fixed selection schedule, choosing a predetermined number of columns or rows in each iteration. In contrast, the second proposed iterative subselection strategy dynamically adjusts the selection schedule based on the decay of the singular values of the data. This approach aims to prioritize the selection of columns or rows that contribute the most to the overall structure and information of the data. By considering the significance of singular values and leveraging their decay pattern, the algorithm can adapt to the unique characteristics of the data, resulting in a more data-driven selection process.

Additionally, we also introduce an alternative approach for computing the residual in the index selection process. The first two iterative subselection algorithms we propose, i.e., Algorithm 12 and Algorithm 13, as well as existing adaptive sampling procedures [10, 28, 29, 79, 94], define the residual as the error resulting from projecting the matrix A onto either the column space of C or the row space of R, i.e., $E = A - CC^+A$ or $E = A - AR^+R$, respectively. In contrast, our new method defines the residual as the error incurred by simultaneously projecting A onto both the column space of C and the row space of R. This entails computing a CUR factorization at each step using only the selected columns and rows.

We have also discussed how iterative procedures for computing a few singular vectors of large data matrices can be used with the newly proposed strategies. We have presented an adaptation of Algorithm 13 for the large-scale case in Algorithm 16, which can straightforwardly be adapted for Algorithm 12 and

Algorithm 14. To the best of our knowledge, Algorithm 16 is the first DEIM-type algorithm for large-scale data sets.

For each of the iterative subselection strategies proposed in this chapter, we invoke the DEIM index selection method. However, we note that other deterministic index selection schemes such as the QDEIM technique [34] and the MaxVol procedure [50] may be employed. We have demonstrated through empirical analysis that the proposed methods in this chapter can produce better approximation results than the traditional method of *one-round sampling* of all columns and rows.

Overall, the proposed techniques may be useful for improving the accuracy of a CUR decomposition, but may also introduce additional complexities that need to be carefully addressed. The choice of whether to use the proposed iterative subselection methods or not may depend on the specific problem or application, as well as the trade-offs between accuracy, complexity, and computational resources.

Part II

GENERALIZATIONS OF CUR DECOMPOSITION

Over the decades, several generalizations of the SVD corresponding to the product or quotient of two to three matrices have been proposed. The most commonly known generalization is the generalized SVD (GSVD), also referred to as the quotient SVD of a matrix pair (A, B) [22], which corresponds to the SVD of AB^{-1} if B is square and nonsingular. Another generalization is the restricted singular value decomposition (RSVD) of a matrix triplet (A, B, G) [97] which shows the SVD of $B^{-1}AG^{-1}$ if B and G are square and nonsingular.

Similarly, in the next two chapters of the thesis, we have proposed two generalizations of an SVD-based CUR decomposition: in Chapter 6, a generalized CUR (GCUR) decomposition of a matrix pair (A, B) [42] and in Chapter 7, a restricted SVD-based CUR (RSVD-CUR) decomposition of a matrix triplet (A, B, G) [45]. The RSVD-CUR is more general than the GCUR decomposition. One can derive a GCUR decomposition from an RSVD-CUR factorization given special choices of the matrices B or G.

6

GENERALIZED CUR DECOMPOSITION FOR MATRIX PAIRS

In this chapter, we propose a generalized CUR decomposition for matrix pairs (A, B). Given matrices A and B with the same number of columns, such a decomposition provides low-rank approximations of both matrices simultaneously in terms of a subset of their rows and columns. We obtain the indices for selecting the subset of rows and columns of the original matrices using the discrete empirical interpolation method on the generalized singular vectors. When B is square and nonsingular, there are close connections between the GCUR of (A, B) and the DEIM-induced CUR of AB^{-1}. When B is the identity, the GCUR decomposition of A coincides with the DEIM-induced CUR decomposition of A. We also show a similar connection between the GCUR of (A, B) and the CUR of AB^{+} for a nonsquare but full-rank matrix B. While a CUR decomposition acts on one data set, a GCUR factorization jointly decomposes two data sets. The algorithm may be suitable for applications where one is interested in extracting the most discriminative features from one data set relative to another data set. In numerical experiments, we demonstrate the advantages of the new method over the standard CUR approximation for recovering data perturbed with colored noise and subgroup discovery.

Adapted from [42]

6.1 INTRODUCTION

Given a matrix pair A and B with the same number of columns: A is $m \times n$, B is $d \times n$ and of full rank, we introduce a generalized CUR decomposition. The intuition behind this GCUR decomposition is that we can view it as a *CUR decomposition of A relative to B*. As we will see in Proposition 6.4, when B is square and nonsingular, the GCUR decomposition has a close connection with the CUR of AB^{-1}. The GCUR is also applicable to nonsquare matrices B; see the examples in Section 6.4. We also show in Proposition 6.4 that if B is nonsquare but of a full rank, we still have a close connection between the CUR decomposition of AB^{+} and the GCUR decomposition. Another motivation for this GCUR decomposition comes from a footnote remark by Mahoney and Drineas [72, p. 700]: "For data sets in which a low-dimensional subspace obtained by the SVD failed to capture

category separation, CUR decompositions performed correspondingly poorly." This is evident in Experiment 6.14.

Inspired by the work of Sorensen and Embree [83], we present a GCUR decomposition algorithm based on the discrete empirical interpolation method. In [83], the authors used DEIM as an index selection technique for constructing the C and R factors of a CUR decomposition. The DEIM algorithm independently selects the column and row indices based on the right and left singular vectors of a data matrix A, respectively. Our new GCUR method uses the matrices obtained from the GSVD instead. Besides using DEIM on the GSVD for index selection, we can also use other CUR-type index selection strategies for the GCUR (see also Section 6.5). The proposed method can be used in situations where a low-rank matrix is perturbed with noise and the covariance of the noise is not a multiple of the identity matrix. It may also be appropriate for applications where one is interested in extracting the most discriminative information from a data set of interest relative to another data set.

The following simple example shows that using the matrices obtained from the GSVD instead of the SVD can lead to more accurate results when approximating data with colored noise. Unlike white noise, colored noise is correlated. In discrete time, the noise samples of colored noise need not be independent. In terms of the Fourier transform, some frequencies are more present than others. As in [57, p. 55] and [78], we use the term "colored noise" for the noise of which the covariance matrix is not a multiple of the identity.

Example 6.1. We consider a full-rank matrix A_E representing low-rank data and want to try to recover an original low-rank matrix perturbed by colored noise. Our test matrix A_E is a rank-2 matrix A of size 3×3 perturbed by additive colored noise E with a given desired covariance structure, i.e., $A_E = A + E$. We take

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 2 \\ 1 & 1 & 2 \end{bmatrix}, \quad E^\top E = \begin{bmatrix} 1.0 & 0.8 & 0.3 \\ 0.8 & 1.0 & 0.8 \\ 0.3 & 0.8 & 1.0 \end{bmatrix}.$$

We generate the colored noise as an additive white Gaussian noise multiplied by the Cholesky factor (R) of the desired covariance matrix. The matrix A_E is, as a result, a sum of a rank-2 matrix and a correlated Gaussian noise matrix. We compute the SVD of both A and A_E. The k dominant left singular vectors of A are denoted by W_k, while those of A_E are $\widetilde{W}_k$. We also compute the GSVD of (A_E, R) and denote the k dominant left generalized singular vectors by U_k. Since we are interested in recovering A, we examine the angle between the leading

k-dimensional exact left singular subspace Range(W_k) and its approximations Range($\widetilde{W}_k$) and Range(U_k). We generate 1000 different test cases and take the average of the subspace angles. The subspace angles are computed using the Matlab inbuilt function subspace.

Table 6.1 shows the results for $k = 2$ and three different noise levels. We observe that the approximations obtained using the GSVD in terms of subspace angles are more accurate than those from the SVD: about 40% gain in accuracy. This illustrates the potential advantage of using generalized singular vectors in the presence of colored noise.

Table 6.1: The average angle between the leading two-dimensional exact singular subspace Range(W_2) (which is the range of A) and its approximations Range($\widetilde{W}_2$) and Range(U_2) for different values of the noise level ε. The subspaces Range(U_2) and Range($\widetilde{W}_2$) are from the SVD of A_E and the GSVD of (A_E, R), respectively.

ε	Method	Subspace angle
$5 \cdot 10^{-2}$	SVD	$1.7 \cdot 10^{-2}$
	GSVD	$1.2 \cdot 10^{-2}$
$5 \cdot 10^{-3}$	SVD	$1.7 \cdot 10^{-3}$
	GSVD	$1.1 \cdot 10^{-3}$
$5 \cdot 10^{-4}$	SVD	$1.7 \cdot 10^{-4}$
	GSVD	$1.1 \cdot 10^{-4}$

Inspired by this example, we expect that the GCUR compared to the CUR may produce better approximation results in the presence of nonwhite noise, as it is based on the GSVD instead of the SVD. We show in Section 6.4 that the GSVD and the GCUR may provide equally good approximation results even when we use an inexact Cholesky factor.

The outline of the chapter is as follows: Section 6.2 gives a brief introduction to the GSVD. We also discuss the truncated GSVD and its approximation error bounds. Section 6.3 introduces the new GCUR decomposition with an analysis of its error bounds. In Algorithm 17, we present a DEIM type GCUR decomposition algorithm. Results of numerical experiments are presented in Section 6.4, followed by conclusions in Section 6.5.

6.2 GENERALIZED SINGULAR VALUE DECOMPOSITION

The GSVD appears throughout this chapter since it is a key building block of the proposed algorithm. This section gives a brief overview of this decomposition. The original proof of the existence of the GSVD has first been introduced by Van Loan in [91]. Paige and Saunders [76] later presented a more general formulation without any restrictions on the dimensions except for both matrices to have the same number of columns. Other formulations and contributions to the GSVD have been proposed in [84, 88, 92]. For our applications in this chapter, let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{d\times n}$ with both $m \geq n$ and $d \geq n$. Following the formulation of the GSVD proposed by Van Loan [91], there exist matrices $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{d\times d}$ with orthonormal columns and a nonsingular $X \in \mathbb{R}^{n\times n}$ such that

$$\begin{aligned} U^\top A X &= \Gamma = \operatorname{diag}(\gamma_1, \ldots, \gamma_n), \qquad \gamma_i \in [0,1], \\ V^\top B X &= \Sigma = \operatorname{diag}(\sigma_1, \ldots, \sigma_n), \qquad \sigma_i \in [0,1], \end{aligned} \tag{6.1}$$

where $\gamma_i^2 + \sigma_i^2 = 1$. Although traditionally the ratios γ_i/σ_i are in nondecreasing order, for our purpose we will instead maintain a nonincreasing order. Thus, $1 \geq \gamma_1 \geq \cdots \geq \gamma_n \geq 0$ and $0 \leq \sigma_1 \leq \cdots \leq \sigma_n \leq 1$. The matrices U and V contain the left generalized singular vectors of A and B, respectively, and, similarly, X contains the right generalized singular vectors and is identical for both decompositions. While the SVD provides two sets of linearly independent basis vectors, the GSVD of (A, B) gives three new sets of linearly independent basis vectors (the columns of U, V, and X) so that the two matrices A and B are diagonal when transformed to these new bases. We note that only the reduced GSVD is needed, so that we can assume that $U \in \mathbb{R}^{m\times n}$, $V \in \mathbb{R}^{d\times n}$, and Γ and Σ are $n \times n$.

Our analysis is based on the following formulation of the GSVD presented in [92]. Let $Y := X^{-T}$ in the GSVD of (6.1). Then $A = U\Gamma Y^\top$ and $B = V\Sigma Y^\top$. Let us characterize matrix Y. (In fact, Matlab's `gsvd` routine renders Y instead of X.) Since

$$A = U\,\Gamma\,Y^\top, \qquad B = V\,\Sigma\,Y^\top, \tag{6.2}$$

this implies that we have the following congruence transformations:

$$A^\top A = Y(\Gamma^\top\Gamma)Y^\top, \qquad B^\top B = Y(\Sigma^\top\Sigma)Y^\top.$$

From the above, it follows that $A^\top A$ has the same inertia as $\Gamma^\top\Gamma$ and the same holds for $B^\top B$ and $\Sigma^\top\Sigma$ (here this mainly gives information on the number of

zero eigenvalues). We also see that, provided A and B are of full rank, these similarity transformations hold:

$$\begin{aligned}(B^\top B)(A^\top A)^{-1} &= Y(\Sigma^\top\Sigma)(\Gamma^\top\Gamma)^{-1}Y^{-1} = Y\,\mathrm{diag}(\sigma_i^2/\gamma_i^2)\,Y^{-1},\\ (A^\top A)(B^\top B)^{-1} &= Y(\Gamma^\top\Gamma)(\Sigma^\top\Sigma)^{-1}Y^{-1} = Y\,\mathrm{diag}(\gamma_i^2/\sigma_i^2)\,Y^{-1}.\end{aligned} \tag{6.3}$$

The columns of the matrix Y are therefore the eigenvectors for both $(A^\top A)(B^\top B)^{-1}$ and its inverse $(B^\top B)(A^\top A)^{-1}$. The GSVD avoids the explicit formation of the cross-product matrices $A^\top A$ and $B^\top B$ (see also Experiment 6.14).

Truncated GSVD. In some practical applications it could be of interest to approximate both matrices (A, B) by other matrices (A_k, B_k), said truncated, of a specific rank k. To define the truncated GSVD (TGSVD), let us partition the following matrices:

$$U = [U_k\ \widehat{U}],\ V = [V_k\ \widehat{V}],\ Y = [Y_k\ \widehat{Y}],\ \Gamma = \mathrm{diag}(\Gamma_k, \widehat{\Gamma}),\ \Sigma = \mathrm{diag}(\Sigma_k, \widehat{\Sigma}). \tag{6.4}$$

For use in Section 6.3, we define TGSVD for (A, B) as (cf. [57, (2.34)])

$$A_k := U_k\Gamma_k Y_k^\top, \qquad B_k := V_k\Sigma_k Y_k^\top, \tag{6.5}$$

where $k < n$. Then it follows that $A - A_k = \widehat{U}\,\widehat{\Gamma}\,\widehat{Y}^\top$. The following proposition is useful for understanding the error bounds for the GCUR. The first and second statements of the following proposition are from [56]; while the third statement may not be present in the literature yet, it is straightforward.

Proposition 6.2. *Let $A = U\,\Gamma\,X^{-1} = U\,\Gamma\,Y^\top$ as in (6.1), with $Y = X^{-T}$. Let $\psi_i(A)$ and $\psi_i(Y)$ be the ith singular value of matrix A and Y, respectively, and let $\psi_{\min}$ be the minimum singular value. Then, for $i = 1, \ldots, n$ (see, e.g., [56, pp. 495–496]),*

$$\gamma_i \cdot \psi_{\min}(Y) \le \psi_i(A) = \psi_i(U\,\Gamma\,Y^\top) \le \psi_i(\Gamma)\ \|Y\| = \gamma_i \cdot \|Y\|,$$

so

$$\frac{\psi_i(A)}{\|Y\|} \le \gamma_i = \psi_i(\Gamma) = \psi_i(U^\top A Y^{-T}) \le \psi_i(A)\ \|Y^{-1}\|.$$

Moreover,

$$\gamma_{k+1} \cdot \psi_{\min}(\widehat{Y}) \le \|A - A_k\| \le \gamma_{k+1} \cdot \|\widehat{Y}\|.$$

Proof. This follows from (6.2) and the well-known property that, for the product of two matrices, we have $\psi_i(A)\,\psi_{\min}(B) \le \psi_i(AB) \le \psi_i(A)\,\|B\|$ (see, e.g., [64, p. 89]). ∎

The results above are relevant tools for the analysis and understanding of the GCUR and its error bounds, which we will introduce in Section 6.3.

6.3 GENERALIZED CUR DECOMPOSITION AND ITS APPROXIMATION PROPERTIES

In this section, we describe the proposed GCUR decomposition and provide a theoretical analysis of its error bounds.

6.3.1 *GCUR decomposition*

We now introduce a new generalized CUR decomposition of matrix pairs (A, B), where A is $m \times n$ $(m \geq n)$ and B is $d \times n$ $(d \geq n)$ and of full rank. This GCUR is inspired by the truncated GSVD for matrix pairs, as reviewed in Section 6.2. We now define a GCUR decomposition (cf. (2.3)).

Definition 6.3. *Let A be $m \times n$ and B be $d \times n$ and of full rank, with $m \geq n$ and $d \geq n$. A GCUR decomposition of (A, B) of rank k is a matrix approximation of A and B expressed as*

$$\begin{aligned} A_k &:= C_A M_A R_A \quad = AP M_A S_A^\top A \,, \\ B_k &:= C_B M_B R_B \quad\;\; = BP M_B S_B^\top B. \end{aligned} \tag{6.6}$$

Here $S_A \in \mathbb{R}^{m \times k}$, $S_B \in \mathbb{R}^{d \times k}$, and $P \in \mathbb{R}^{n \times k}$ are index selection matrices $(k < n)$.

It is key that *the same* columns of A and B are selected; this gives a coupling between the decomposition of A and B.

The matrices C_A, C_B and R_A, R_B are subsets of the columns and rows, respectively, of the original matrices. In the rest of the chapter, we will mainly focus our analysis on the matrix A; we can perform a similar analysis for the matrix B (see also the comments at the end of this section). We have the vectors $\mathbf{s}_A, \mathbf{p}$ containing the indices of the selected rows and columns such that $C_A = AP$ and $R_A = S_A^\top A$, where $S_A = I(:, \mathbf{s}_A)$ and $P = I(:, \mathbf{p})$. The choice of $\mathbf{p}$ and $\mathbf{s}_A$ is based on the transformation matrices from the rank-k truncated GSVD. Given P and S_A, we construct the middle matrix M_A as $(C_A^\top C_A)^{-1} C_A^\top A R_A^\top (R_A R_A^\top)^{-1}$.

The following proposition establishes a connection between the DEIM-GCUR of (A, B) and the DEIM-CUR of AB^{-1} and AB^{+} for a square and nonsingular B and a nonsquare but full-rank B, respectively. It is worth noting that this proposition holds for DEIM-based CUR and GCUR algorithms. For alternative ways of constructing CUR and GCUR decompositions (see Chapter 2), these properties may not hold. Let the vector $\mathbf{s}_B$ contain the indices of selected rows of B.

Proposition 6.4. *(i) If B is a square and nonsingular matrix, then the selected row and column indices from a CUR decomposition of AB^{-1} are the same as index vectors $\mathbf{s}_A$ and $\mathbf{s}_B$ obtained from a GCUR decomposition of (A, B), respectively.*

(ii) Moreover, in the special case where $B = I$, a GCUR decomposition of A coincides with a CUR decomposition of A in that the factors C and R of A are the same for both methods: The first line of (6.6) is equal to (2.3).

(iii) In addition, if B is nonsquare but of a full rank, we have a connection as in (i) between the indices from a CUR decomposition of AB^{+} and the index vectors s_A and s_B obtained from a GCUR decomposition of (A, B).

Proof. (i) We start with the GSVD (6.2). If B is square and nonsingular, then the SVD of AB^{-1} can be expressed in terms of the GSVD of (A, B) and is equal to $U(\Gamma\Sigma^{-1})V^{\top}$ [48]. Therefore, the row index selection matrix from the SVD of AB^{-1} is equal to S_A from the GSVD of (A, B), and, similarly, the column index selection matrix obtained from the SVD of AB^{-1} is equal to S_B since they are determined using U and V, respectively.

(ii) If $B = I$, then from the second line of (6.2), we have that $Y = V\Sigma^{-1}$. This implies that the indices of the largest entries in the columns of Y are the same as those of V. In this special case of $B = I$, we have $AB^{-1} = A$, so then, the left and right singular vectors of A are contained in the U and V matrices from the GSVD of (A, I), respectively. Hence, the selection matrix P in (6.6) obtained by performing DEIM on Y is the same as the selection matrix P in (2.3) obtained by applying DEIM to the right singular vectors of A.

(iii) If B is nonsquare but of full-rank n, then we still have a similar connection between the GSVD of (A, B) and the SVD of AB^{+} because of the following. Since the factors in the reduced GSVD $B = V\Sigma Y^{\top}$ are of full rank, we have $B^{+} = Y^{-T}\Sigma^{-1}V^{\top}$. This means that $AB^{+} = U\Gamma\Sigma^{-1}V^{\top}$, so the index vectors $\mathbf{s}_A$ and $\mathbf{s}_B$ from GCUR of (A, B) are equivalent to the selected column and row indices from CUR of AB^{+}, respectively. ∎

Although we can obtain indices of a CUR decomposition of AB^{-1} using the GCUR of (A, B), the converse does not hold. We emphasize that we need the GSVD for the GCUR decomposition and cannot use the SVD of AB^{-1} or AB^{+} instead since the GCUR decomposition requires the Y matrix from (6.5) to find the column indices. While we used the generalized singular vectors here, in principle one could use other vectors, e.g., an approximation to the generalized singular vectors.

To build the decomposition, it is relevant to know the dominant rows and columns of A and B in their rank-k approximations. Given that A_k and B_k are

rank-k approximations of A and B, respectively, how should the columns and rows be selected? Algorithm 17 is a summary of how the DEIM index selection can be used to construct a GCUR decomposition. We note that if we are only interested in approximating the matrix A from the pair (A, B), we can omit line 4 as well as the second part of line 5; thus saving computational cost.

Algorithm 17: DEIM type GCUR decomposition

Data: $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{d\times n}$ (with $m \geq n$ and $d \geq n$), desired rank k
Result: A rank-k GCUR decomposition $A_k = A(:, \mathbf{p}) \cdot M_A \cdot A(\mathbf{s}_A, :)$, $B_k = B(:, \mathbf{p}) \cdot M_B \cdot B(\mathbf{s}_B, :)$

1 $[U, V, Y] = \mathsf{gsvd}(A, B)$ (according to nonincreasing GSVs)
2 $\mathbf{p} = \mathsf{deim}(Y(:, 1:k))$
3 $\mathbf{s}_A = \mathsf{deim}(U(:, 1:k))$
4 $\mathbf{s}_B = \mathsf{deim}(V(:, 1:k))$
5 $M_A = A(:, \mathbf{p}) \setminus (A \,/\, A(\mathbf{s}_A, :))$, $M_B = B(:, \mathbf{p}) \setminus (B \,/\, B(\mathbf{s}_B, :))$

Remark 6.5. The pseudocode in Algorithm 17 assumes the matrices from the GSVD (i.e., U, V, and Y) correspond to a nonincreasing order of the generalized singular values. This implies that we select the most "dominant" parts of A and the least "dominant" parts of B. As a consequence of this assumption, the relative approximation error of matrix A in (6.6) tends to be relatively modest. However, it is important to note that the same cannot be guaranteed for the relative approximation errors of matrix B. Since the least "dominant" parts of matrix B are chosen for approximation, the resulting relative errors may be more significant.

In terms of computational complexity, the dense GSVD method requires $\mathcal{O}((m+d)n^2)$ work, and the three runs of DEIM together require $\mathcal{O}((m+n+d)k^2)$ work, so the overall complexity of the algorithm is dominated by the construction of the GSVD. (This might suggest iterative GSVD approaches; see Section 6.5.)

In some applications, one might be interested in a *generalized interpolative decomposition*, of which the column and row versions are of the form

$$A \approx C_A \widetilde{M}_A,\ B \approx C_B \widetilde{M}_B \qquad \text{or} \qquad A \approx \widehat{M}_A R_A,\ B \approx \widehat{M}_B R_B. \tag{6.7}$$

Here $\widetilde{M}_A = C_A^{+} A$ is $k \times n$, and $\widehat{M}_A = A R_A^{+}$ is $m \times k$; similar remarks hold for $\widetilde{M}_B$ and $\widehat{M}_B$. As noted in [83], since the DEIM index selection algorithm identifies the

row and column indices independently, this form of decomposition is relatively straightforward. The column version of a generalized interpolative decomposition can be obtained by ignoring Lines 3 and 4 in Algorithm 17 and computing Line 5 as $\widetilde{M}_A$ and $\widetilde{M}_B$. On the other hand, the row version can be implemented by omitting Line 2 and replacing Line 5 with $\widehat{M}_A$ and $\widehat{M}_B$.

In generalizing a DEIM-induced CUR decomposition, we also look for a generalization of the related theoretical results. While the results presented in [83] express the error bounds in terms of the optimal rank-k approximation, for our GCUR factorization, the most relevant quantity is the rank-k GSVD approximation. In the following subsection, we present theoretical results for bounding the GCUR approximation error.

6.3.2 *Error bounds in terms of the SVD approximation*

The error bounds for any rank-k matrix approximation are usually expressed in terms of the rank-k SVD approximation error since it provides the optimal low-rank approximation. We will show a result of this type in the following proposition and also discuss its limitations.

We introduce the following notation: Let $A = W\Psi Z^\top = W_k\Psi_k Z_k^\top + W_\perp \Psi_\perp Z_\perp^\top$ be the SVD of A, where Z_k contains the largest k right singular vectors. Let Q_k be an $n \times k$ matrix with orthonormal columns.

It turns out in both [83] and this section that $\|A(I - Q_kQ_k^\top)\|$ is a central quantity in the analysis. In the DEIM-induced CUR decomposition work [83], the right singular vectors are contained in Q_k, but here we study this quantity for general Q_k. In our context, we are particularly interested in Q_k as the orthogonal basis of the matrix Y_k in (6.5). Denote $\mathcal{Q}_k = \operatorname{span}(Q_k)$ and $\mathcal{Z}_k = \operatorname{span}(Z_k)$.

Proposition 6.6. *Let Q_k be an $n \times k$ matrix with orthonormal columns, Z_k contain the largest k right singular vectors of A, and $\psi_i(A)$ be the ith singular value of A. Then*

$$\psi_{k+1}^2(A) \le \|A(I - Q_kQ_k^\top)\|^2 \le \psi_{k+1}^2(A) + \|A\|^2 \cdot \sin^2(\mathcal{Z}_k, \mathcal{Q}_k).$$

More precisely, we have

$$\|A(I - Q_kQ_k^\top)\|^2 \le \psi_{k+1}^2(A) + \sum_{j=1}^{k} \psi_j(A)^2 \cdot \sin^2(\mathbf{z}_j, \mathcal{Q}_k).$$

Proof. The lower bound follows from the SVD; the optimal $\mathcal{Q}_k$ is $\mathcal{Z}_k$. We can derive the upper bounds from

$$\begin{aligned}\|A(I - Q_k Q_k^\top)\|^2 &= \|W_k \Psi_k Z_k^\top (I - Q_k Q_k^\top)\|^2 + \|W_\perp \Psi_\perp Z_\perp^\top (I - Q_k Q_k^\top)\|^2 \\ &\leq \|A\|^2 \cdot \sin^2(\mathcal{Z}_k, \mathcal{Q}_k) + \psi_{k+1}^2 \cdot \sin^2(\mathcal{Z}_\perp, \mathcal{Q}_k) \\ &\leq \|A\|^2 \cdot \sin^2(\mathcal{Z}_k, \mathcal{Q}_k) + \psi_{k+1}^2.\end{aligned}$$

Furthermore, more specifically,

$$\|A(I - Q_k Q_k^\top)\|^2 = \sum_{j=1}^{k} \psi_j^2(A)\ |\mathbf{z}_j^\top (I - Q_k Q_k^\top)|^2 + \|W_\perp \Psi_\perp Z_\perp^\top (I - Q_k Q_k^\top)\|^2.$$

■

The significance of this result is that $\|A(I - Q_k Q_k^\top)\|$ may be close to $\psi_{k+1}(A)$ when $\mathcal{Q}_k$ captures the largest singular vectors of A well. For instance, in the standard CUR, Q_k is equivalent to Z_k, so the quantity $\sin^2(\mathcal{Z}_k, \mathcal{Q}_k)$ equals zero. If the matrix B from (6.2) is close to the identity or is a scaled identity, we expect that $\sin^2(\mathcal{Z}_k, \mathcal{Q}_k)$ will be approximately zero. However, this sine will generally not be small, as we illustrate by the following example.

Example 6.7. Let $A = \mathrm{diag}(1, 2, 3)$, and let $B = \mathrm{diag}(1, 20, 300)$. Denote by $\mathbf{e}_j$ the jth standard basis vector. Then clearly $Z_1 = \mathbf{z}_1 = \mathbf{e}_3$, while the largest right generalized singular vector $\mathbf{q}_1$ is equal to the largest right singular vector of $AB^{-1} = \mathrm{diag}(1, 0.1, 0.01)$, and hence $Q_1 = \mathbf{q}_1 = \mathbf{e}_1$. This implies that $\sin(\mathcal{Z}_1, \mathcal{Q}_1) = \sin(\mathbf{z}_1, \mathbf{q}_1)$ is large.

6.3.3 *Error bounds in terms of the GSVD approximation*

With the above results in mind, instead of using the rank-k SVD approximation error, we will derive error bounds for $\|A - CMR\|$ (see (6.6)) in terms of the error bounds of a rank-k GSVD approximation of A (see Proposition 6.2). Since we are focusing on matrix A, we drop the subscript A in (6.6). The matrices C and R are of full-rank k determined by the row and column index selection matrices S and P, respectively, and $M = C^+ A R^+$. From Algorithm 17, we know that S and P are derived using the k columns of the matrices U and Y, respectively, corresponding to the largest generalized singular value (see (6.5)).

We use the interpolatory projector given in Proposition 2.4. Therefore instead of Y (see (6.2)), we use its orthonormal basis Q to exploit the properties of an orthogonal matrix.

We will now analyze the approximation error between A and its interpolatory projection $A\mathbb{P}$. The proof of the error bounds for the proposed method closely follows the one presented in [83]. The second inequality of the first statement of Proposition 6.8 is in [83, Lemma 4.1]. The first inequality of the first statement is new but completely analogous. In the second statement, we use the GSVD. For the analysis, we need the following QR-decomposition of Y (see (6.4)):

$$[Y_k \;\; \widehat{Y}] = Y = QT = [Q_k \;\; \widehat{Q}] \begin{bmatrix} T_k & T_{12} \\ 0 & T_{22} \end{bmatrix} = [Q_k T_k \;\; Q\widehat{T}], \tag{6.8}$$

where we have defined

$$\widehat{T} := \begin{bmatrix} T_{12} \\ T_{22} \end{bmatrix}. \tag{6.9}$$

This implies that

$$A = A_k + \widehat{U}\,\widehat{\Gamma}\,\widehat{Y}^\top = U_k \Gamma_k Y_k^\top + \widehat{U}\,\widehat{\Gamma}\,\widehat{Y}^\top = U_k \Gamma_k T_k^\top Q_k^\top + \widehat{U}\,\widehat{\Gamma}\,\widehat{T}^\top Q^\top.$$

Proposition 6.8. *(Generalization of [83, Lemma 4.1]) Given $A \in \mathbb{R}^{m\times n}$ and $Q_k \in \mathbb{R}^{n\times k}$ with orthonormal columns where $k < n$, let $P \in \mathbb{R}^{n\times k}$ be a selection matrix and $Q_k^\top P$ be nonsingular and $\psi_{\min}$ be the minimum singular value. Let $\mathbb{P} = P(Q_k^\top P)^{-1} Q_k^\top$. Then*

$$\psi_{\min}(A(I - Q_k Q_k^\top))\, \|(Q_k^\top P)^{-1}\| \le \|A - A\mathbb{P}\| \le \|A(I - Q_k Q_k^\top)\|\, \|(Q_k^\top P)^{-1}\|.$$

In particular, if Q_k is an orthonormal basis for Y_k, the first k columns of Y, then

$$\gamma_{k+1} \cdot \psi_{\min}(T_{22}) \cdot \|(Q_k^\top P)^{-1}\| \le \|A - A\mathbb{P}\| \le \gamma_{k+1} \cdot \|T_{22}\| \cdot \|(Q_k^\top P)^{-1}\|.$$

Proof. We have that $Q_k^\top \mathbb{P} = Q_k^\top P (Q_k^\top P)^{-1} Q_k^\top = Q_k^\top$, which implies $Q_k^\top (I - \mathbb{P}) = 0$. Therefore,

$$\begin{aligned} \|A - A\mathbb{P}\| = \|A(I - \mathbb{P})\| &= \|A(I - Q_k Q_k^\top)(I - \mathbb{P})\| \\ &\le \|A(I - Q_k Q_k^\top)\|\, \|I - \mathbb{P}\|, \end{aligned}$$

and also

$$\|A(I - Q_k Q_k^\top)(I - \mathbb{P})\| \ge \psi_{\min}(A(I - Q_k Q_k^\top))\, \|I - \mathbb{P}\|\,.$$

Note that, since $k < n$, we know that $\mathbb{P} \neq 0$ and $\mathbb{P} \neq I$, and hence (see, e.g., [89])

$$\|I - \mathbb{P}\| = \|\mathbb{P}\| = \|(Q_k^\top P)^{-1}\|.$$

With $A = U\,\Gamma\,Y^\top$, $A_k = U_k\Gamma_k Y_k^\top$, $Y = QT$, and $Y_k = Q_k T_k$, we have

$$\begin{aligned} A\,Q_k\,Q_k^\top &= \begin{bmatrix} U_k & \widehat{U} \end{bmatrix} \begin{bmatrix} \Gamma_k & 0 \\ 0 & \widehat{\Gamma} \end{bmatrix} \begin{bmatrix} T_k^\top & 0 \\ T_{12}^\top & T_{22}^\top \end{bmatrix} \begin{bmatrix} I_k \\ 0 \end{bmatrix} Q_k^\top \\ &= U_k\Gamma_k T_k^\top Q_k^\top + \widehat{U}\,\widehat{\Gamma}\,T_{12}^\top Q_k^\top, \end{aligned}$$

and hence

$$\begin{aligned} A\,(I - Q_k Q_k^\top) &= (A - A_k) - \widehat{U}\,\widehat{\Gamma}\,T_{12}^\top Q_k^\top \\ &= \widehat{U}\,\widehat{\Gamma}\,\widehat{T}^\top Q^\top - \widehat{U}\,\widehat{\Gamma}\,T_{12}^\top Q_k^\top = \widehat{U}\,\widehat{\Gamma}\,T_{22}^\top \widehat{Q}^\top. \end{aligned}$$

This implies

$$\|A\,(I - Q_k Q_k^\top)\| \le \gamma_{k+1} \cdot \|T_{22}\| \quad \text{and} \quad \|A\,(I - Q_k Q_k^\top)\| \ge \gamma_{k+1} \cdot \psi_{\min}(T_{22}).$$

■

Let us now consider the operation on the left-hand side of A. Given the set of interpolation indices $\{s_1, \ldots, s_k\}$ determined from U_k, $S = [\mathbf{e}_{s_1}, \ldots, \mathbf{e}_{s_k}]$ and for a nonsingular $S^\top U_k$, we have the DEIM interpolatory projector $\mathbb{S} = U_k(S^\top U_k)^{-1}S^\top$. Since U_k consists of the dominant k left generalized singular vectors of A and has orthonormal columns, it is not necessary to perform a QR-decomposition as we did in Proposition 6.8.

The following proposition is analogous to Proposition 6.8. The results are similar to those in [83, p. A1461] except that here we use the approximation error of the GSVD instead of the SVD.

Proposition 6.9. *Given $U_k \in \mathbb{R}^{m\times k}$ with orthonormal columns where $k < m$, let $S \in \mathbb{R}^{m\times k}$ be a selection matrix and $S^\top U_k$ be nonsingular. Furthermore, let $\mathbb{S} = U_k(S^\top U_k)^{-1}S^\top$. Then, with $\widehat{T}$ as in (6.9),*

$$\gamma_{k+1} \cdot \psi_{\min}(\widehat{T}) \cdot \|(S^\top U_k)^{-1}\| \le \|A - \mathbb{S}A\| \le \gamma_{k+1} \cdot \|\widehat{T}\| \cdot \|(S^\top U_k)^{-1}\|.$$

Proof. We have

$$\|A - \mathbb{S}A\| = \|(I - \mathbb{S})A\| = \|(I - \mathbb{S})(I - U_k U_k^\top)A\|.$$

Similar to before, since $k < m$, we know that $\mathbb{S} \neq 0$ and $\mathbb{S} \neq I$, and hence

$$\|I - \mathbb{S}\| = \|\mathbb{S}\| = \|(S^\top U_k)^{-1}\|.$$

Since $(I - U_k U_k^\top)A = A - U_k\Gamma_k Y_k^\top = \widehat{U}\,\widehat{\Gamma}\,\widehat{Y}^\top = \widehat{U}\,\widehat{\Gamma}\,\widehat{T}^\top Q^\top$, we get

$$\|(I - U_k U_k^\top)A\| = \|A - A_k\| \le \gamma_{k+1} \cdot \|\widehat{T}\|$$

and $\|(I - U_k U_k^\top)A\| \geq \gamma_{k+1} \cdot \psi_{\min}(\widehat{T})$, from which the result follows. ■

We will now use Propositions 6.8 and 6.9 to find a bound for the approximation error of the GCUR of A relative to B. As in [83], we first show in the following proposition that the error bounds of the interpolatory projection of A onto the chosen rows and columns apply equally to the orthogonal projections of A onto the same row and column spaces.

Proposition 6.10. *(Generalization and slight adaptation of [83, Lemma 4.2]) Given the selection matrices P, S, let $C = AP$ and $R = S^\top A$. Suppose that $C \in \mathbb{R}^{m\times k}$ and $R \in \mathbb{R}^{k\times n}$ are full-rank matrices with $k < \min(m, n)$ and that $Q_k^\top P$ and $S^\top U_k$ are nonsingular. With $\widehat{T}$ and T_{22} as in (6.8)–(6.9), we have the bound for the orthogonal projections of A onto the column and row spaces:*

$$\|(I - CC^+)A\| \leq \gamma_{k+1} \cdot \|T_{22}\| \cdot \|(Q_k^\top P)^{-1}\|,$$
$$\|A(I - R^+R)\| \leq \gamma_{k+1} \cdot \|\widehat{T}\| \cdot \|(S^\top U_k)^{-1}\|.$$

Proof. This proof is a minor modification of that of [83, Lemma 4.2]; we closely follow their proof technique. With $C = AP$ of full rank, we have $C^+ = (P^\top A^\top AP)^{-1}(AP)^\top$. With this, the orthogonal projection of A onto Range(C) can be stated as $CC^+A = (AP(P^\top A^\top AP)^{-1}P^\top A^\top)A$.
Let $\Pi_P = P(P^\top A^\top AP)^{-1}P^\top A^\top A$, and note that $\Pi_P P = P$ since Π_P is an oblique projector on Range(P). We can rewrite CC^+A as $A\Pi_P$. Hence, the error in the orthogonal projection of A will be $(I - CC^+)A = A(I - \Pi_P)$. Since $\Pi_P \mathbb{P} = \mathbb{P}$, we have

$$A(I - \Pi_P) = A(I - \Pi_P)(I - \mathbb{P}) = (I - CC^+)A(I - \mathbb{P}).$$

Therefore

$$\begin{aligned}\|(I - CC^+)A\| = \|A(I - \Pi_P)\| &= \|(I - CC^+)A(I - \mathbb{P})\| \\ &\leq \|(I - CC^+)\| \, \|A(I - \mathbb{P})\|.\end{aligned}$$

With C being nonsquare, $\|I - CC^+\| = 1$ (see, e.g., [89]), and $\|A(I - \mathbb{P})\| \leq \gamma_{k+1} \cdot \|T_{22}\| \cdot \|(Q_k^\top P)^{-1}\|$ from Proposition 6.8, we have

$$\|(I - CC^+)A\| \leq \gamma_{k+1} \cdot \|T_{22}\| \cdot \|(Q_k^\top P)^{-1}\|.$$

In a similar vein, with $R = S^\top A$ and $R^+ = R^\top(RR^\top)^{-1}$, we have $R^+ = A^\top S(S^\top AA^\top S)^{-1}$, and the error in the orthogonal projection of A is $A(I - R^+R) = (I - \Pi_S)A$, where $\Pi_S = AA^\top S(S^\top AA^\top S)^{-1}S^\top$, so that

$$(I - \Pi_S)A = (I - \mathbb{S})(I - \Pi_S)A = (I - \mathbb{S})A(I - R^+R)$$

and

$$\|A(I - R^+R)\| \leq \|(I - \mathbb{S})A\| \, \|(I - R^+R)\| \leq \gamma_{k+1} \cdot \|\widehat{T}\| \cdot \|(S^\top U_k)^{-1}\|.$$

■

This result helps to prove an error bound for the GCUR approximation error. For the following theorem, we again closely follow the approach of [83] which also follows a procedure in [72]. As stated in Definition 6.3, the middle matrix can be computed as $M = (C^\top C)^{-1} C^\top A R^\top (RR^\top)^{-1} = C^+ A R^+$.

Theorem 6.11. *(Generalization of [83, Thm. 4.1]) Given $A \in \mathbb{R}^{m \times n}$ and Y_k, U_k from (6.5), let P and S be selection matrices so that $C = AP$ and $R = S^\top A$ are of full rank. Let $Q_k \in \mathbb{R}^{n \times k}$ be the Q-factor of Y_k and $\widehat{T}$ and T_{22} as in (6.8)–(6.9). Assuming $Q_k^\top P$ and $S^\top U_k$ are nonsingular, then with the error constants*

$$\eta_p := \|(Q_k^\top P)^{-1}\|, \qquad \eta_s := \|(S^\top U_k)^{-1}\|,$$

we have

$$\|A - CMR\| \leq \gamma_{k+1} \cdot (\eta_p \cdot \|T_{22}\| + \eta_s \cdot \|\widehat{T}\|) \leq \gamma_{k+1} \cdot (\eta_p + \eta_s) \cdot \|\widehat{T}\|. \tag{6.10}$$

Proof. By the definition of M, we have

$$A - CMR = A - CC^+AR^+R = (I - CC^+)A + CC^+A(I - R^+R).$$

Using the triangle inequality, it follows that

$$\begin{aligned} \|A - CMR\| &= \|A - CC^+AR^+R\| \\ &\leq \|(I - CC^+)A\| + \|CC^+\| \, \|A(I - R^+R)\|, \end{aligned}$$

and the fact that CC^+ is an orthogonal projection with $\|CC^+\| = 1$,

$$\|A - CMR\| \leq \gamma_{k+1} \cdot \|T_{22}\| \cdot \|(Q_k^\top P)^{-1}\| + \|(S^\top U_k)^{-1}\| \, \|\widehat{T}\| \cdot \gamma_{k+1}.$$

■

The last line of Theorem 6.11 can be related to the results in [83, Thm. 4.1]; both theorems have the factors η_p and η_s. In [83], the error of a CUR approximation of A is within a factor of $\eta_p + \eta_s$ of the best rank-k approximation, obtained from the SVD. Theorem 6.11 provides a bound in terms of $\gamma_{k+1} \leq 1$ from the GSVD (6.1) and the additional factors $\|\widehat{T}\|$ and $\|T_{22}\|$. The results presented in this section suggest that a good index selection procedure that yields small quantities

$\|(Q_k^\top P)^{-1}\|$ and $\|(S^\top U_k)^{-1}\|$ is desirable. To bound $\|\widehat{T}\|$, we start by restating the results of [57, Thm. 2.3] for the GSVD of (A, B). Defining

$$L := \begin{bmatrix} A \\ B \end{bmatrix}, \quad \text{it follows that} \quad \|X^{-1}\| = \|L\| \leq \|A\| + \|B\|.$$

We know from (6.2) that $X^{-T} = Y$, so we can restate the above inequality as $\|Y\| \leq \|A\| + \|B\|$. Given the partitioning and QR factorization of Y in (6.8), we have that

$$\|\widehat{T}\| = \|Q\widehat{T}\| = \|\widehat{Y}\| \leq \|Y\| = \|L\| \leq \|A\| + \|B\|.$$

We note that we can exploit the tighter bound $\|L\| \leq (\|A\|^2 + \|B\|^2)^{1/2}$ to improve the bound on $\widehat{T}$ accordingly.

We note that where these results have been presented for matrix A in (6.6), similar results can be obtained for B. The following error bound for the approximation of B is analogous to (6.10). As noted in Definition 6.3, the selection matrix P is similar to the GCUR decomposition of A and B. Therefore, we have the quantity $\|(Q_k^\top P)^{-1}\|$ in the error bound of both factorizations:

$$\begin{aligned} \|B - C_B M_B R_B\| &\leq \|\widehat{\Sigma}\| \cdot (\|(Q_k^\top P)^{-1}\| \cdot \|T_{22}\| + \|(S_B^\top V_k)^{-1}\| \cdot \|\widehat{T}\|) \\ &\leq (\|(Q_k^\top P)^{-1}\| + \|(S_B^\top V_k)^{-1}\|) \cdot \|\widehat{T}\|. \end{aligned}$$

Recall that the entries of Σ are sorted in nondecreasing order with the maximum element being 1. Therefore, $\|\widehat{\Sigma}\| = 1$. It is worth noting that these bounds hold regardless of the approach used to select the row and column indices. Since the GCUR algorithm presented in this chapter is DEIM based, [83] provides deterministic bounds for the error constants:

$$\|(Q_k^\top P)^{-1}\| < \sqrt{\tfrac{nk}{3}}\, 2^k, \quad \|(S_A^\top U_k)^{-1}\| < \sqrt{\tfrac{mk}{3}}\, 2^k, \quad \text{and} \quad \|(S_B^\top V_k)^{-1}\| < \sqrt{\tfrac{dk}{3}}\, 2^k.$$

We refer the reader to [83, Lemma 4.4] for the constructive proofs and we will give an example with the various quantities in Experiment 6.12.

6.4 NUMERICAL EXPERIMENTS

We now present the results of a few numerical experiments to illustrate the performance of GCUR for low-rank matrix approximation. For the first two experiments, we consider a case where a data matrix A is corrupted by a random additive noise E and the covariance of this noise (the expectation of $E^\top E$) is

not a multiple of the identity matrix. We are therefore interested in a method that can take the actual noise into account. Traditionally, a prewhitening matrix R^{-1} (where R is the Cholesky factor of the noise's covariance matrix) may be applied to the perturbed matrix [57] so that one can use SVD-based methods on the transformed matrix. With a GSVD formulation, the prewhitening operation becomes an integral part of the algorithm [58]; we do not need to explicitly compute R^{-1} and transform the perturbed matrix. We show in the experiments that using SVD-based methods without prewhitening the perturbed data yields less accurate approximation results of the original matrix.

For the last two experiments, we consider a setting with two data sets collected under different conditions, e.g., treatment and control experiment, where the former has distinct variation caused by the treatment: signal-free and signal recordings with the signal-free data set containing only noise. We are interested in exploring and identifying patterns and discriminative features that are specific to one data set.

Experiment 6.12. This experiment is an adaptation of experiments in [57, section. 3.4.4, p. 66] and [83, Example. 6.1]; see also the motivating example in Section 6.1. We construct matrix A to be of a known modest rank. We then perturb this matrix with a noise matrix $E \in \mathbb{R}^{m\times n}$ whose entries are correlated. Given $A_E = A + E$, we evaluate and compare the GCUR and the CUR decomposition on A_E in terms of recovering the original matrix A. Specifically, the performance of each decomposition is assessed based on the spectral norm of the relative matrix approximation error, i.e., $\|A - \widetilde{A}\|/\|A\|$, where $\widetilde{A}$ is the approximated low-rank matrix. We present the numerical results for four noise levels; thus, $E = \varepsilon \frac{\|F\|}{\|A\|} F$, where ε is the parameter for the noise level and F is a randomly generated correlated noise. We first generate a sparse, nonnegative rank-50 matrix $A \in \mathbb{R}^{m\times n}$, with $m = 100000$ and $n = 300$, of the form

$$A = \sum_{j=1}^{10} \frac{2}{j}\, \mathbf{x}_j\, \mathbf{y}_j^\top + \sum_{j=11}^{50} \frac{1}{j}\, \mathbf{x}_j\, \mathbf{y}_j^\top,$$

where $\mathbf{x}_j \in \mathbb{R}^m$ and $\mathbf{y}_j \in \mathbb{R}^n$ are sparse vectors with random nonnegative entries (i.e., $\mathbf{x}_j = \mathsf{sprand}(m, 1, 0.025)$ and $\mathbf{y}_j = \mathsf{sprand}(n, 1, 0.025)$, just as in [83]. Unlike [83], we then perturb the matrix with a correlated Gaussian noise E whose entries have zero mean and a Toeplitz covariance structure (in MATLAB desired-cov(F) = $\mathsf{toeplitz}(0.99^0, \ldots, 0.99^{n-1})$, $R = \mathsf{chol}(\text{desired-cov}(F))$, and $F = \mathsf{randn}(m, n) \cdot R$) and $\varepsilon \in \{0.05, 0.1, 0.15, 0.2\}$. We compute the SVD of A_E and the GSVD of (A_E, R) to get the input matrices for the CUR and the GCUR decomposition,

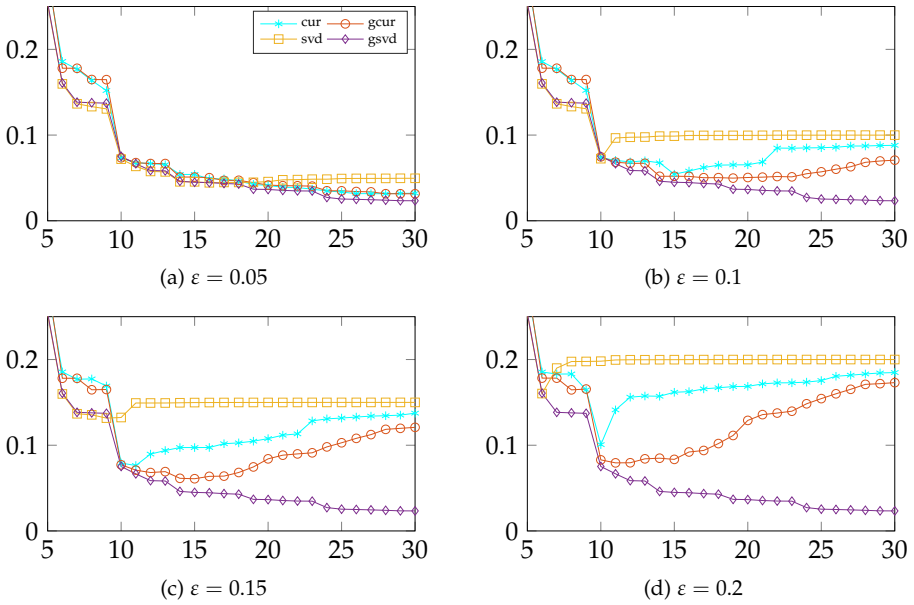
respectively. Figures 6.1a, 6.1b, 6.1c, and 6.1d compares the relative errors of the proposed DEIM-GCUR (see Algorithm 17) and the DEIM-CUR for reconstructing the low-rank matrix A for different noise levels. We observe that for higher noise levels, the GCUR technique gives a more accurate low-rank approximation of the original matrix A. The DEIM-GCUR scheme seems to perform distinctly well for higher noise levels and moderate values of k. As indicated in Section 6.3, the GCUR method is slightly more expensive since it requires the computation of the TGSVD instead of the TSVD. We observe that, as k approaches $\operatorname{rank}(A)$, the relative error of the TGSVD continues to decrease; this is not true for the GCUR. We may attribute this phenomenon to the fact that the relative error is saturated by the noise, considering we pick actual columns and rows of the noisy data. Since ε indicates the relative noise level, it is therefore natural that for increasing k, the quality of the TSVD approximation rapidly approaches ε. For this experiment, we assume that an estimate of the noise covariance matrix is known, and therefore we have the exact Cholesky factor; we stress that this may not always be the case.

Therefore, we now show an example where we use an inexact Cholesky factor $\widehat{R}$. We derive $\widehat{R}$ by multiplying all off-diagonal elements of the exact Cholesky factor R by factors that are uniformly random from the interval $[0.9, 1.1]$. Here, the experiment setup is the same as described above with the difference that we compute the GSVD of $(A_E, \widehat{R})$ instead. In Figs. 6.2a and 6.2b, we observe that the GCUR and the GSVD still deliver good approximation results even for an inexact Cholesky factor $\widehat{R}$, which may imply that we do not necessarily need the exact noise covariance.

We conclude this experiment with an illustration of the various quantities in Theorem 6.11. In Fig. 6.3, we see that the upper bound in Theorem 6.11 may be a rather crude bound on the true GCUR error. As in [83, Figure 4], the quantities η_S and η_P may differ considerably in magnitude. While $\|T_{22}\|$ steadily decreases, $\|\widehat{T}\|$ seems to stabilize as k increases.

Experiment 6.13. For this experiment, we maintain all properties of matrix A_E mentioned in the preceding experiment except for the column size, which we reduce to 10000 (i.e., $A_E \in \mathbb{R}^{10000\times 300}$) and instead of a sparse nonnegative matrix A, we generate a dense random matrix A. As in [83, Example 6.2], we also modify A so that there is a significant drop in the 10th and 11th singular values. The matrix A is now of the form

$$A = \sum_{j=1}^{10} \frac{1000}{j}\, \mathbf{x}_j\, \mathbf{y}_j^\top + \sum_{j=11}^{50} \frac{1}{j}\, \mathbf{x}_j\, \mathbf{y}_j^\top,$$



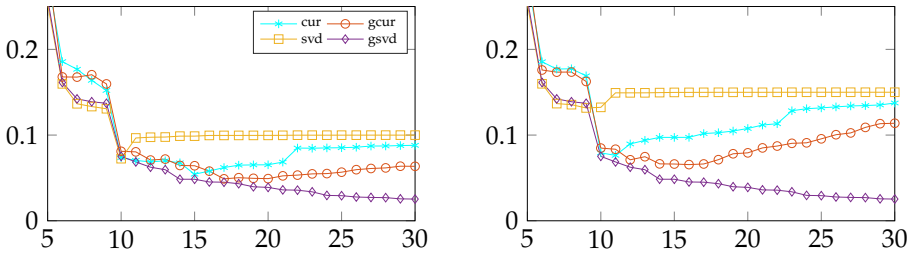
(a) $\varepsilon = 0.05$ (b) $\varepsilon = 0.1$

(c) $\varepsilon = 0.15$ (d) $\varepsilon = 0.2$

Figure 6.1: Accuracy of the DEIM-GCUR approximations compared with the standard DEIM-CUR approximations in recovering a sparse, nonnegative matrix A perturbed with correlated Gaussian noise (Experiment 6.12) using exact Cholesky factor of the noise covariance. The relative errors $\|A - \widetilde{A}_k\|/\|A\|$ (on the vertical axis) as a function of rank k (on the horizontal axis) for $\varepsilon = 0.05$, 0.1, 0.15, 0.2, respectively.

For each fixed ε and k, we repeat the process 100 times and then compute the average relative error. The results in Table 6.2 show that the advantage of the GCUR over the CUR remains even when singular values of the original matrix A decrease more sharply. We observe that the difference in the relative error of the GCUR and the CUR is quite significant when the rank of the recovered matrix $\widetilde{A}$ is lower than that of A (i.e., $k \ll 50$).

The higher the noise level, the more advantageous the GCUR scheme may be over the CUR one. Especially for moderate values of k, such as $k = 10$, the GCUR approximations are of better quality than those based on the CUR. For higher values of k, such as $k = 30$, the approximation quality of the CUR and GCUR methods become comparable since they both start to pick up the noise in the data columns. In this case, the GCUR does not improve on the CUR. Since it is



(a) $\varepsilon = 0.1$ (b) $\varepsilon = 0.15$

Figure 6.2: Accuracy of the DEIM-GCUR approximations compared with the standard DEIM-CUR approximations in recovering a sparse, nonnegative matrix A perturbed with correlated Gaussian noise (Experiment 6.12) using an inexact Cholesky factor of the noise covariance. The relative errors $\|A - \widetilde{A}_k\|/\|A\|$ (on the vertical axis) as a function of rank k (on the horizontal axis) for $\varepsilon = 0.15$, 0.1, respectively.

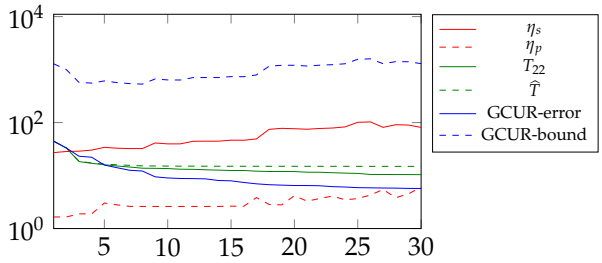


Figure 6.3: Various quantities from Theorem 6.11. Error constants $\eta_P = \|(Q_k^\top P)^{-1}\|$ (red dashed) and $\eta_S = \|(S_A^\top U_k)^{-1}\|$ (red solid); multiplicative factors $\|T_{22}\|$ (green solid) and $\|\widehat{T}\|$ (green dashed); the GCUR true error $\|A_E - (CMR)_{\text{gcur}}\|$ of approximating A_E in Experiment 6.12 (blue solid) and its upper bound (blue dashed).

a discrete method, picking indices for columns instead of generalized singular vectors, we see that the GCUR method yields higher reconstruction errors than the TGSVD approach.

Experiment 6.14. Our next experiment is adapted from [1]. We create synthetic data sets that give an intuition for settings where the GSVD and the GCUR may resolve the problem of subgroups. Consider a data set of interest (target data), A, containing 400 data points in a 30-dimensional feature space. This data set

Table 6.2: Comparison of the qualities $\|A - \widetilde{A}_k\| / \|A\|$ of the TSVD, TGSVD, CUR, and GCUR approximations as a function of index k and noise level ε in Experiment 6.13. The relative errors are the averages of 100 test cases.

k	Method \ ε	0.05	0.1	0.15	0.2
10	TSVD	0.008	0.045	0.150	0.200
	TGSVD	0.002	0.003	0.005	0.007
	CUR	0.052	0.118	0.141	0.186
	GCUR	0.053	0.088	0.112	0.134
15	TSVD	0.050	0.100	0.150	0.200
	TGSVD	0.009	0.017	0.026	0.035
	CUR	0.049	0.097	0.146	0.196
	GCUR	0.046	0.091	0.138	0.185
20	TSVD	0.050	0.100	0.150	0.200
	TGSVD	0.011	0.023	0.034	0.015
	CUR	0.050	0.099	0.149	0.199
	GCUR	0.049	0.097	0.146	0.198
30	TSVD	0.050	0.100	0.150	0.200
	TGSVD	0.016	0.031	0.047	0.063
	CUR	0.050	0.100	0.150	0.199
	GCUR	0.050	0.099	0.149	0.199

has four subgroups (blue, yellow, orange, and purple), each of 100 data points. The first 10 columns for all 400 data points are randomly sampled from a normal distribution with a mean of 0 and a variance of 100. The next 10 columns of two of the subgroups (blue and orange) are randomly sampled from a normal distribution with a mean of 0 and a unit variance, while the other two subgroups (yellow and purple) are randomly sampled from a normal distribution with a mean of 6 and a unit variance. The last 10 columns of subgroups blue and yellow are sampled from a normal distribution with a mean of 0 and a unit variance, and those of purple and orange are sampled from a normal distribution with a mean of 3 and a unit variance.

One of the goals of the SVD (or the related concept principal component analysis) in dimension reduction is to find a low-dimensional rotated approximation of a data matrix while maximizing the variances. We are interested in reducing the dimension of A. If we project the data onto the two leading right singular vectors, we are unable to identify the subgroups because the variation along the first 10

columns is significantly larger than in any other direction, so some combinations of those columns are selected by the SVD.

Suppose we have another data set B (a background data set), whose first 10 columns are sampled from a normal distribution with a mean of 0 and a variance of 100. The next 10 columns are sampled from a normal distribution with a mean of 0 and a variance of 9, and the last 10 columns are sampled from a normal distribution with a mean of 0 and a unit variance. The choice of the background data set is key in this context. Generally, the background data set should have the structure we would like to suppress in the target data, which usually corresponds to the direction with a high variance but is not of interest for the data analysis [1]. With the new data, one way to extract discriminative features for clustering the subgroups in A is to maximize the variance of A while minimizing that of B, which leads to a trace ratio maximization problem [18]

$$\widehat{U} := \operatorname*{argmax}_{U \in \mathbb{R}^{n\times k},\, U^\top U = I_k} \operatorname{Tr}\left[(U^\top B^\top B\, U)^{-1}(U^\top A^\top A\, U)\right],$$

where $n = 30$ and $k = 5$ or $k = 10$. By doing this, the first dimensions are less likely to be selected because they also have a high variance in data set B. Instead, the middle and last dimensions of A are likely to be selected, as they have the dimensions with the lowest variance in B, thereby allowing us to separate all four subgroups. The solution $\widehat{U} \in \mathbb{R}^{n\times k}$ to the above problem is given by the k (right) eigenvectors of $(B^\top B)^{-1}(A^\top A)$ corresponding to the k largest eigenvalues (cf. [39, pp. 448–449]); this corresponds to the ("largest") right generalized singular vectors of (A, B) (the transpose of (6.3)). As seen in Fig. 6.4, projecting A onto the leading two right generalized singular vectors produces a much clearer subgroup separation (top-right figure) than projecting onto the leading two right singular vectors (top-left figure). Therefore, we can expect that a CUR decomposition based on the SVD will also perform not very well with the subgroup separation. In the bottom figures is a visualization of the data using the first two important columns selected using the DEIM-CUR (left figure) and the DEIM-GCUR (right figure). To a large extent, the GCUR is able to differentiate the subgroups, while the CUR fails to do so. We investigate this further by comparing the performance of subset selection via DEIM-CUR on A and DEIM-GCUR on (A, B) (Algorithm 17) in identifying the subgroup or class representatives of A; we select a subset of the columns of A (5 and 10) and compare the classification results of each method. We center the data sets by subtracting the mean of each column from all the entries in that column. Given the class labels of the subgroups, we perform a 10-fold cross-validation (i.e., split the data points into 10 groups and for each unique group take the group as testing data set and the rest as training data set [65,

p. 181]) and apply two classifiers on the reduced data set: ECOC (Error-Correcting Output Codes) [30] and *classification tree* [4] using the functions fitcecoc and fitctree with default parameters as implemented in MATLAB. It is evident from Table 6.3 that the TGSVD and the GCUR achieve the lowest classification error rate, e.g., for reducing the dimension from 30 to 10— 0% and 6.3%, respectively, using the ECOC classifier and 0% and 9.5%, respectively, using the tree classifier. The standard DEIM-CUR method achieves the worst classification error rate.

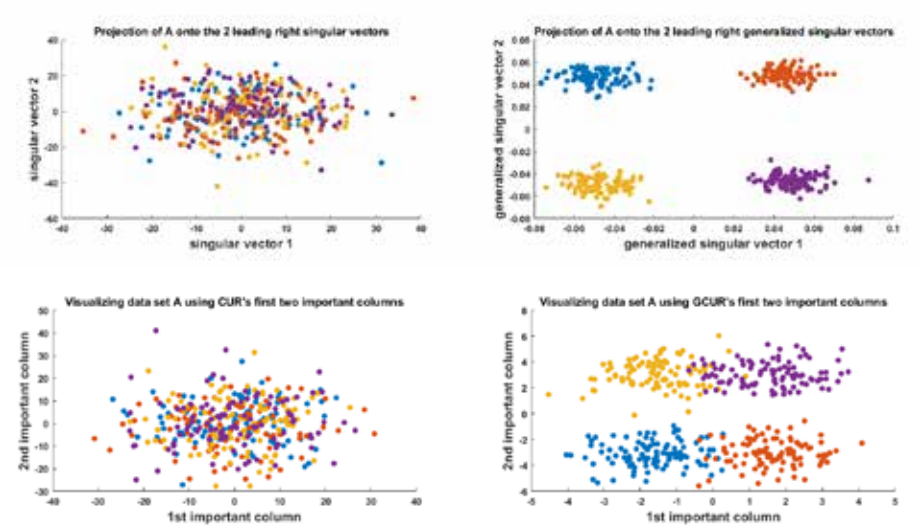


Figure 6.4: (Top left) We project the synthetic data containing four subgroups onto the first two dominant right singular vectors. The lower-dimensional representation using the SVD does not effectively separate the subgroups. In the bottom-left figure, we visualize the data using the first two columns selected by DEIM-CUR. (Top right) We illustrate the advantage of using GSVD by projecting the data onto the first two dominant right generalized singular vectors corresponding to the two largest generalized singular values. In the bottom-right figure, we visualize the data using the first two columns selected by DEIM-GCUR. The lower-dimensional representation of the data using the GSVD-based methods clearly separates the four clusters, while the SVD-based methods fail to do so.

Experiment 6.15. We will now investigate the performance of the GCUR compared to the CUR on higher-dimensional public data sets. The data sets consist of

Table 6.3: k-Fold loss is the average classification loss overall 10-fold using SVD, GSVD, CUR, and GCUR as dimension reduction in Experiment 6.14. The second and third columns give information on the number of columns selected from the data set using the CUR and GCUR plus the number of singular and generalized singular vectors considered for the ECOC classifier, likewise for the fifth and sixth columns for the tree classifier.

Method	k-Fold Loss		Method	k-Fold Loss	
	5	10		5	10
TSVD+ECOC	0.638	0.490	TSVD+Tree	0.693	0.555
TGSVD+ECOC	0	0	TGSVD+Tree	0	0
CUR+ECOC	0.793	0.485	CUR+Tree	0.793	0.540
GCUR+ECOC	0.055	0.063	GCUR+Tree	0.075	0.095

single-cell RNA expression levels of bone marrow mononuclear cells (BMMCs) from an acute myeloid leukemia (AML) patient and two healthy individuals. We have data on the BMMCs before the stem-cell transplant and the BMMCs after the stem-cell transplant. We preprocess the data sets as described by the authors in [9][1] keeping the 1000 most variable genes measured across all 16856 cells (patient-035: 4501 cells and two healthy individuals; one of 1985 cells and the other of 2472 cells). The data from the two healthy patients are combined to create a background data matrix of dimension 4457×1000, and we use the patient-035 data set as the target data matrix of dimension 4501×1000. Both data matrices are sparse: The patient-035 data matrix has 1628174 nonzeros, i.e., about 36% of all entries are nonzero, and the background data matrix has 1496229 nonzeros, i.e., about 34% of all entries are nonzero. We are interested in exploring the differences in the AML patient's BMMC cells pre- and post-transplant. We perform SVD, GSVD, CUR and GCUR on the target data (AML patient-035) to see if we can capture biologically meaningful information relating to the treatment status. For the GSVD and the GCUR procedure, the background data are taken into account. As evident in Fig. 6.5, the GSVD and the GCUR produce almost linearly separable clusters that correspond to pre- and posttreatment cells. These methods evidently capture the biologically meaningful information relating to the treatment and are more effective at separating the pre- and post-transplant cell samples compared to the other two. For the SVD and the CUR, we observe that both cell types follow a similar distribution in the space spanned by the first

1 https://github.com/PhilBoileau/EHDBDscPCA/blob/master/analyses/.

three dominant right singular vectors and the first three important gene columns, respectively. Both methods fail to separate the pre- and posttransplant cells.

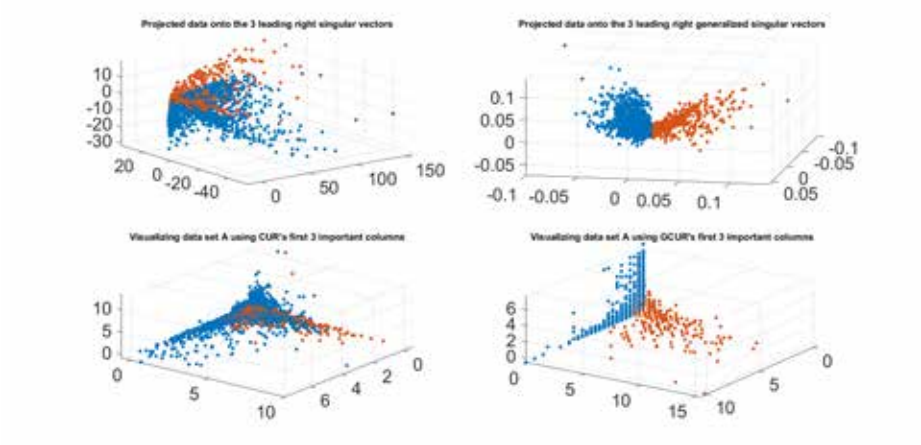


Figure 6.5: Acute myeloid leukemia patient-o35 scRNA-seq data.(Top left) A 3-D projection of the patient's BMMCs on the first three dominant right singular vectors. In the bottom-left figure, we visualize the data using the first three genes selected by DEIM-CUR. The lower-dimensional representation using the SVD-based methods does not effectively give a discernible cluster of the pre- and post-transplant cells. (Top right) We illustrate the advantage of using GSVD by projecting the patient's BMMCs onto the first three dominant right generalized singular vectors corresponding to the three largest generalized singular values. In the bottom-right figure, we visualize the data using the first three genes selected by DEIM-GCUR. The lower-dimensional representation using the GSVD-based methods produces linearly separable clusters.

6.5 FINAL CONSIDERATIONS

We develop a new method, the DEIM-induced GCUR factorization with pseudocode in Algorithm 17. It is an extension of the DEIM-CUR decomposition for matrix pairs. Just as the CUR decomposition has an interpolative decomposition (see, e.g., [93]) associated with it, there is a *generalized interpolative decomposition* (see (6.7)) associated with the GCUR decomposition.

When B is square and nonsingular, there are close connections between the GCUR of (A, B) and the DEIM-induced CUR of AB^{-1}. When B is the identity, the

GCUR decomposition of A coincides with the DEIM-induced CUR decomposition of A. There exists a similar connection between the CUR of AB^+ and the GCUR of (A, B) for a nonsquare but full-rank matrix B.

While a CUR decomposition acts on one data set, a GCUR factorization decomposes two data sets together. An implication of this is that we can use it in selecting discriminative features of one data set relative to another. For subgroup discovery and subset selection in a classification problem where two data sets are available, the new method can perform better than the standard DEIM-CUR decomposition as shown in the numerical experiments. The GCUR algorithm can also be useful in applications where a data matrix suffers from nonwhite (colored) noise. The GCUR algorithm can provide more accurate approximation results compared to the DEIM-CUR algorithm when recovering an original matrix with low rank from data with colored noise. For the recovery of data perturbed with colored noise, we need the Cholesky factor of an estimate of the noise covariance. However, as shown in the experiments, even for an inexact Cholesky factor, the GCUR may still give good approximation results. We note that, while the GSVD always provides a more accurate result than the SVD regardless of the noise level, the GCUR decomposition is particularly attractive for higher noise levels and moderate values of the rank of the recovered matrix compared to the CUR factorization. In other situations, both methods may provide comparable results. In addition, the GCUR decomposition is a discrete method, so choosing indices for columns and rows instead of the generalized singular vectors leads to higher reconstruction errors than the GSVD approach.

Although we used the generalized singular vectors here, in principle one could use other vectors, e.g., an approximation to the generalized singular vectors. We have extended the existing theory concerning the DEIM-CUR approximation error to this DEIM GCUR factorization; we derived the bounds of a rank-k GCUR approximation of A in terms of a rank-k GSVD approximation of A.

Instead of the DEIM procedure for the index selection from the GSVD, it might be possible to use alternative index selection strategies. For a CUR decomposition, one alternative approach to DEIM is to perform a QR factorization with column pivoting [34] on the transpose of the matrices from the truncated SVD. In [50, 51], the authors propose a CUR factorization where C and R are selected by searching for submatrices of maximal volume in the singular vector matrices. Another popular approach is selection via leverage score sampling [33, 72]. From experiment results not reported here, we observe that except for leverage scores sampling, using the above strategies in the context of the GCUR produce competitive results.

Computationally, the DEIM-GCUR algorithm requires the input of the GSVD, which is of the same complexity but more expensive than the SVD required for DEIM-CUR. For the case where we are only interested in approximating the matrix A from the pair (A, B), we can omit some of the lines in Algorithm 17, thus saving computational cost. In the case that the matrices A and B are so large, a full GSVD may not be affordable; in this case, we can consider iterative methods (see, e.g., [62, 99, 102]).

While in this work we used the GCUR method in applications such as extracting information from one data set relative to another, we expect that its promise may be more general.

7

A RESTRICTED SVD TYPE CUR DECOMPOSITION FOR MATRIX TRIPLETS

We present a new restricted SVD-based CUR (RSVD-CUR) factorization for matrix triplets (A, B, G) that aims to extract meaningful information by providing a low-rank approximation of the three matrices using a subset of their rows and columns. The proposed method employs the discrete empirical interpolation method to select the subset of rows and columns based on the orthogonal and nonsingular matrices obtained through a restricted singular value decomposition of the matrix triplet. We explore the relationships between a DEIM type RSVD-CUR factorization, a DEIM type CUR factorization, and a DEIM type generalized CUR decomposition, and provide an error analysis that establishes the accuracy of the RSVD-CUR decomposition within a factor of the approximation error of a rank-k restricted singular value decomposition of the given matrices.

The RSVD-CUR factorization can be used in applications that require approximating one data matrix relative to two other given matrices. We discuss two of such applications, namely multi-view dimension reduction and data perturbation problems where a correlated noise matrix is added to the input data matrix. Our numerical experiments demonstrate the advantages of the proposed method over the standard CUR approximation in these scenarios.

Adapted from [45]

7.1 INTRODUCTION

Given matrices $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times \ell}$, and $G \in \mathbb{R}^{d\times n}$ of compatible dimensions, we generalize a DEIM type CUR [83] method to develop a new coupled CUR factorization of a matrix triplet. This decomposition is based on the knowledge of the restricted singular value decomposition (RSVD). We call our proposed factorization an RSVD based CUR factorization. Note that this RSVD does not stand for randomized SVD (see, e.g., [55]). Both CUR decomposition and RSVD algorithms have been well-studied. However, to the best of our knowledge, this work is the first to combine both methods. The RSVD has been around for over three decades now; this new method introduces a new type of exploitation of the RSVD.

In recent times, real-world data sets often contain multiple representations or viewpoints, each providing unique and complementary information. One of the motivations for the RSVD-CUR factorization comes from the canonical correlation analysis (CCA) (see Section 7.2). CCA is a popular method for analyzing the relationships between two sets of variables, and it has broad applications in various fields [59, pp. 443–454]. CCA aims to find linear combinations of variables from each data set that exhibit the highest correlation with each other. These linear combinations, represented by the canonical vectors, form a basis for the correlated subspaces of the data sets. The first canonical vector pair has the highest correlation, and subsequent pairs have decreasing correlations. Using an RSVD-CUR factorization in this context, our goal is to extract subsets of columns or rows from these data matrices by exploiting the canonical vectors of each matrix that maximize the correlations between them. As we will see in Sections 7.2 and 7.3, the canonical vectors are equivalent to matrices from the RSVD. We believe that the RSVD-CUR factorization can be useful for multi-view dimension reduction and integration of information from multiple views in multi-view learning, a rapidly growing area of machine learning that involves using multiple perspectives to improve generalization performance [96]. Similar to CCA, the RSVD-CUR factorization can handle two-view cases and may also be employed as a supervised feature selection technique in multi-label classification problems, where one view comes from the data and the other from the class labels.

Another motivation for an RSVD-CUR factorization stems from applications where the goal is to select a subset of rows and columns of one data set relative to two other data sets. An example is a data perturbation problem of the form $A_E = A + BFG$ where BFG is a correlated noise matrix (see, e.g., [6, 97]) and the goal is to recover the low-rank matrix A from A_E given the structure of B and G. Conventionally, when faced with this kind of perturbation problem, to use an SVD-based method, a prewhitening step is required to make the additive colored noise a white noise using B^{-1} and G^{-1}. However, with the RSVD formulation, the prewhitening operation becomes an integral part of the algorithm. It is worth pointing out that one does not necessarily need to know the exact noise covariance matrices; the RSVD and RSVD-CUR may still deliver good approximation results given inexact covariance matrices (see Experiment 7.7). An example of an inexact covariance matrix is when we approximate the population covariance matrix by a sample covariance matrix.

A further incentive for proposing an RSVD-CUR factorization arises within the framework of generalized Gauss-Markov models with constraints. Considering the ordinary or total least squares problem of the form $A\mathbf{x} \approx \mathbf{b}$, in many applica-

tions, it is desirable to reduce the number of variables that are to be considered or measured in the future. For instance, the modeler may not be interested in a predictor such as $A\mathbf{x}$ with all redundant variables but rather $A\widehat{\mathbf{x}}$, where $\widehat{\mathbf{x}}$ has at most k nonzero entries. The position of the nonzero entries determines which columns of A, i.e., which variables to use in the model for approximating the response vector $\mathbf{b}$. How to pick these columns is the problem of subset selection, and one can use a CUR factorization algorithm. Consider the setting of generalized Gauss-Markov models with constraints, i.e.,

$$\min_{\mathbf{x},\mathbf{y},\mathbf{f}} \|\mathbf{y}\|^2 + \|\mathbf{f}\|^2 \quad \text{subject to} \quad \mathbf{b} = A\mathbf{x} + B\mathbf{y}, \quad \mathbf{f} = G\mathbf{x}, \tag{7.1}$$

where $A, B, G, \mathbf{b}$ are given. Notice that where $B = I$ and $G = 0$, this formulation is a generalization of the traditional least squares problem. Since this equation involves three matrices, an appropriate tool for its analysis will be the RSVD [27, 57]. For variable subset selection in this problem, the RSVD-CUR may be a suitable method that incorporates the error and the constraints (more details in Section 7.5).

A short review of CCA is provided in Section 7.2. Section 7.3 gives a brief overview of the RSVD. Section 7.4 introduces the new RSVD-CUR decomposition. In this section, we also discuss some error bounds. Algorithm 19 summarizes the procedure of constructing a DEIM type RSVD-CUR decomposition. Results of numerical experiments using synthetic and real data sets are presented in Section 7.5, followed by conclusions in Section 7.6.

7.2 CANONICAL CORRELATION ANALYSIS

This section briefly discusses CCA, one of our motivations for the proposed RSVD-CUR approximation. CCA is one of the most widely used and valuable techniques for multi-data processing. It is used to analyze the mutual relationships between two sets of variables and finds a wide range of applications across many different fields. In a web classification problem, usually, a web document can be described by either the words occurring on the page (this, for instance, can be matrix B) or the words contained in the anchor text of links pointing to this page (and can be taken as matrix G). In a genome-wide association study, CCA is used to discover the genetic associations between genotype data of single nucleotide polymorphisms (SNPs) contained in B and phenotype data of gene expression levels contained in G [20]. In information retrieval, CCA is used to embed both the search space (e.g., images) and the query space (e.g., text) into a shared

low dimensional latent space so that the closeness between the queries and the candidates can be quantified [82]. Other applications include fMRI data analysis [38], natural language processing, and speech recognition [2].

Let $B \in \mathbb{R}^{m\times\ell}$, $G \in \mathbb{R}^{d\times n}$ with $m = d$, be of full column rank and $k \leq \min(\ell, n)$. CCA seeks to find the linear combinations of the form $B\mathbf{z}_i$ and $G\mathbf{w}_i$ for $i = 1, \ldots, k$ that maximize the pairwise correlations across the two matrices [59, p. 443]. We can define the canonical correlations $\rho_1(B, G), \ldots, \rho_k(B, G)$ of the matrix pair (B, G) as [49]

$$\rho_i(B,G) = \max_{\substack{B\mathbf{z}\neq\mathbf{0},\, G\mathbf{w}\neq\mathbf{0} \\ B\mathbf{z}\perp\{B\mathbf{z}_1,\ldots,B\mathbf{z}_{i-1}\} \\ G\mathbf{w}\perp\{G\mathbf{w}_1,\ldots,G\mathbf{w}_{i-1}\}}} \rho(B\mathbf{z}, G\mathbf{w}) =: \rho(B\mathbf{z}_i, G\mathbf{w}_i) := \frac{\mathbf{z}_i^\top B^\top G\mathbf{w}_i}{\|B\mathbf{z}_i\|\,\|G\mathbf{w}_i\|}. \quad (7.2)$$

We have that $\rho_1(B, G) \geq \cdots \geq \rho_k(B, G)$. The vectors of unit length $B\mathbf{z}_i/\|B\mathbf{z}_i\|$ and $G\mathbf{w}_i/\|G\mathbf{w}_i\|$ are referred to as the canonical vectors of (B, G) and the canonical weights are $\mathbf{z}_i/\|B\mathbf{z}_i\|$ and $\mathbf{w}_i/\|G\mathbf{w}_i\|$. As discussed in [49], there are several equivalent ways to formulate CCA. We show a Lagrange multiplier formulation which is suitable for our context and will serve as a motivation for the proposed decomposition. The Lagrange multiplier function of the above constrained optimization problem is [49]

$$f(\mathbf{z}, \mathbf{w}, \lambda, \mu) = \mathbf{z}^\top B^\top G\mathbf{w} - \tfrac{1}{2}\,\lambda\,(\|B\mathbf{z}\|^2 - 1) - \tfrac{1}{2}\,\mu\,(\|G\mathbf{w}\|^2 - 1).$$

Differentiating the above with respect to $\mathbf{z}$ and $\mathbf{w}$ gives

$$\begin{aligned} B^\top G\mathbf{w} - \lambda\, B^\top B\mathbf{z} &= \mathbf{0}\,, \\ G^\top B\mathbf{z} - \mu\, G^\top G\mathbf{w} &= \mathbf{0}. \end{aligned}$$

We note that $\rho_i(B, G)$ is not affected by the rescaling of $B\mathbf{w}_i$ and $G\mathbf{z}_i$. Thus, there is an optimal solution satisfying the constraints $\mathbf{w}^\top G^\top G\mathbf{w} = 1$ and $\mathbf{z}^\top B^\top B\mathbf{z} = 1$. Hence, we solve the systems for that particular optimal solution. Premultiplying the above equations by $\mathbf{z}^\top$ and $\mathbf{w}^\top$, respectively, together with the constraints $\mathbf{w}^\top G^\top G\mathbf{w} = 1$ and $\mathbf{z}^\top B^\top B\mathbf{z} = 1$, we have that $\lambda = \mu$ and

$$\begin{bmatrix} & B^\top G \\ G^\top B & \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{w} \end{bmatrix} = \lambda \begin{bmatrix} B^\top B & \\ & G^\top G \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{w} \end{bmatrix}. \quad (7.3)$$

The canonical weights and correlations are the generalized eigenvectors and eigenvalues, respectively, of this generalized eigenvalue problem. We will show in the next section how this problem relates to the RSVD of matrix triplets which we exploit for our proposed RSVD-CUR factorization.

7.3 RESTRICTED SVD

The RSVD of matrix triplets, as notably studied in [27, 97], is an essential building block for the proposed decomposition in this chapter. We give a brief overview of this matrix factorization here. The RSVD may be viewed as a decomposition of a matrix relative to two other matrices of compatible dimensions. Given a matrix triplet $A \in \mathbb{R}^{m\times n}$ (where, without loss of generality, $m \geq n$), $B \in \mathbb{R}^{m\times \ell}$, and $G \in \mathbb{R}^{d\times n}$, following the formulation in [97], there exist orthogonal matrices $U \in \mathbb{R}^{\ell\times\ell}$ and $V \in \mathbb{R}^{d\times d}$, and nonsingular matrices $Z \in \mathbb{R}^{m\times m}$ and $W \in \mathbb{R}^{n\times n}$ such that

$$A = Z\, D_A\, W^\top, \quad B = Z\, D_B\, U^\top, \quad G = V\, D_G\, W^\top, \tag{7.4}$$

which implies

$$\begin{bmatrix} A & B \\ G & \end{bmatrix} = \begin{bmatrix} Z & \\ & V \end{bmatrix} \begin{bmatrix} D_A & D_B \\ D_G & \end{bmatrix} \begin{bmatrix} W & \\ & U \end{bmatrix}^\top,$$

where $D_A \in \mathbb{R}^{m\times n}$, $D_B \in \mathbb{R}^{m\times \ell}$, and $D_G \in \mathbb{R}^{d\times n}$ are quasi-diagonal matrices [1]. We refer the reader to [97] for detailed proof of the above decomposition. In the case of $m < n$, it is logical to take the transpose of the matrix triplet and interchange the position of B and G to ensure compatible dimensions, i.e., $(A^\top, G^\top, B^\top)$. With respect to the theory, applications and our experiments, we focus on the so-called *regular* matrix triplet (A, B, G), i.e., B is of full row rank and G is of full column rank [98].

Algorithms for the computation of the RSVD are still an active field of research; some recent works include [22, 101]. As noted in [27], the RSVD can be computed via a double GSVD (see Eq. (6.1)). The following is a practical procedure to construct the RSVD using the GSVD. For ease of presentation, we first assume that $m = \ell$ and $d = n$ so that B and G are nonsingular. Then, we have the following expression as the RSVD from two GSVDs:

1 A quasi-diagonal matrix, in this work, is a matrix that is diagonal after removing all zero rows and columns.

$$\begin{aligned}
\begin{bmatrix} A & B \\ G & \end{bmatrix} &= \begin{bmatrix} U_1 & \\ & V_1 \end{bmatrix} \begin{bmatrix} \Gamma_1 & U_1^\top B \\ \Sigma_1 & \end{bmatrix} \begin{bmatrix} Y_1^\top & \\ & I \end{bmatrix} \\
&= \begin{bmatrix} U_1 & \\ & V_1 \end{bmatrix} \begin{bmatrix} \Gamma_1 \Sigma_1^{-1} & U_1^\top B \\ I & \end{bmatrix} \begin{bmatrix} \Sigma_1 Y_1^\top & \\ & I \end{bmatrix} \\
&= \begin{bmatrix} U_1 Y_2 & \\ & V_1 \end{bmatrix} \begin{bmatrix} \Sigma_2^\top & \Gamma_2^\top \\ V_2 & \end{bmatrix} \begin{bmatrix} V_2^\top \Sigma_1 Y_1^\top & \\ & U_2^\top \end{bmatrix} \\
&= \begin{bmatrix} U_1 Y_2 & \\ & V_1 V_2 \end{bmatrix} \begin{bmatrix} \Sigma_2^\top \Gamma_G & \Gamma_2^\top \\ \Gamma_G & \end{bmatrix} \begin{bmatrix} Y_1 \Sigma_1 V_2 \Gamma_G^{-1} & \\ & U_2 \end{bmatrix}^\top .
\end{aligned}$$

In these four steps, we have first computed the GSVD of (A, G), i.e., $A = U_1 \Gamma_1 Y_1^\top$ and $G = V_1 \Sigma_1 Y_1^\top$. Note that Σ_1 is nonsingular since G is nonsingular. Next, we compute the GSVD of the transposes of the pair $(U_1^\top B, \Gamma_1 \Sigma_1^{-1})$, so that $U_1^\top B = Y_2 \Gamma_2^\top U_2^\top$ and $\Gamma_1 \Sigma_1^{-1} = Y_2 \Sigma_2^\top V_2^\top$. Moreover, Γ_G is a nonsingular scaling matrix that one can freely select (see, e.g., [101]). In this square case, we have $\Sigma_2^\top = \Sigma_2$, but we keep this notation for consistency with the nonsquare case we will discuss next.

In some of our applications of interest (see Experiment 7.8), we have that $\ell = d > m \geq n$. In this case, we get the following modifications:

$$\begin{aligned}
&\begin{bmatrix} U_1 & \\ & V_1 \end{bmatrix} \begin{bmatrix} \Gamma_1 & U_1^\top B \\ \begin{bmatrix} \Sigma_1 \\ 0_{d-n,n} \end{bmatrix} & \end{bmatrix} \begin{bmatrix} Y_1^\top & \\ & I \end{bmatrix} \\
&= \begin{bmatrix} U_1 & \\ & V_1 \end{bmatrix} \begin{bmatrix} \Gamma_1 \Sigma_1^{-1} & U_1^\top B \\ \begin{bmatrix} I \\ 0_{d-n,n} \end{bmatrix} & \end{bmatrix} \begin{bmatrix} \Sigma_1 Y_1^\top & \\ & I \end{bmatrix} \\
&= \begin{bmatrix} U_1 Y_2 & \\ & V_1 \end{bmatrix} \begin{bmatrix} \Sigma_2^\top & \Gamma_2^\top \\ \begin{bmatrix} V_2 \\ 0_{d-n,n} \end{bmatrix} & \end{bmatrix} \begin{bmatrix} V_2^\top \Sigma_1 Y_1^\top & \\ & U_2^\top \end{bmatrix} \\
&= \begin{bmatrix} U_1 Y_2 & \\ & V_1 \widehat{V}_2 \end{bmatrix} \begin{bmatrix} \Sigma_2^\top \Gamma_G & \Gamma_2^\top \\ \begin{bmatrix} \Gamma_G \\ 0_{d-n,n} \end{bmatrix} & \end{bmatrix} \begin{bmatrix} Y_1 \Sigma_1 V_2 \Gamma_G^{-1} & \\ & U_2 \end{bmatrix}^\top .
\end{aligned}$$

In these steps, we use $\widehat{V}_2 = \mathsf{diag}(V_2, I_{d-n})$. Algorithm 18 summarizes the procedure for computing the RSVD of the so-called *regular* matrix triplets (A, B, G).

Algorithm 18: RSVD via a double GSVD

Data: $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times \ell}$, $G \in \mathbb{R}^{d\times n}$, $\quad m \geq n,\ m \leq \ell,\ d \geq n$
Result: $Z \in \mathbb{R}^{m\times m}$, $W \in \mathbb{R}^{n\times n}$, $U \in \mathbb{R}^{\ell\times \ell}$, $V \in \mathbb{R}^{d\times d}$ $D_A \in \mathbb{R}^{m\times n}$, $D_B \in \mathbb{R}^{m\times \ell}$, and $D_G \in \mathbb{R}^{d\times n}$ (see (7.4))

Compute $[U_1, V_1, Y_1, \Gamma_1, \Sigma_1] = \mathsf{gsvd}(A, G)$
Set $\Sigma_1 = \Sigma_1(1:n, :)$
Compute $[U_2, V_2, Y_2, \Gamma_2, \Sigma_2] = \mathsf{gsvd}(B^\top U_1,\ (\Gamma_1\Sigma_1^{-1})^\top)$
$\mathbf{a} = \mathsf{diag}(\Sigma_2) \quad (\in \mathbb{R}^n)$
$\Gamma_G = \mathsf{diag}(a_i\,(a_i^2+1)^{-1/2}), \quad (i = 1, \ldots, n)$
if $d = n$, $\quad D_G = \Gamma_G$; $\quad V = V_1V_2$; **end**
if $d > n$, $\quad V = V_1 \cdot \mathsf{diag}(V_2, I_{d-n})$; $\quad D_G = [\Gamma_G;\ 0_{d-n,n}]$; **end**
$Z = U_1Y_2$; $\quad W = Y_1\Sigma_1V_2\Gamma_G^{-1}$; $\quad U = U_2$; $\quad D_A = \Sigma_2^\top\Gamma_G$; $\quad D_B = \Gamma_2^\top$

In the two GSVD steps, we emphasize that we maintain the traditional nondecreasing ordering of the generalized singular values in both GSVDs. That is, the diagonal entries of Γ_1 and Γ_2 are in nondecreasing order while those of Σ_1 and Σ_2 are in nonincreasing order. Note that Σ_1 is again nonsingular because G is of full rank. With reference to (7.4), we define $Z := U_1Y_2$, $W := Y_1\Sigma_1V_2\Gamma_G^{-1}$, $V := V_1\widehat{V}_2$, $U := U_2$, $D_A := \Sigma_2^\top\Gamma_G$, $D_B := \Gamma_2^\top$, and $D_G := \begin{bmatrix}\Gamma_G\\ 0_{d-n,n}\end{bmatrix}$. The quasi-diagonal matrices D_A, D_B, and D_G have the following structure:

$$D_A = \begin{bmatrix} D_1 \\ 0_{m-n,n} \end{bmatrix}, \quad D_B = \begin{bmatrix} D_2 & 0_{n,m-n} & 0_{n,\ell-m} \\ 0_{m-n,n} & I_{m-n} & 0_{m-n,\ell-m} \end{bmatrix}, \quad D_G = \begin{bmatrix} \Gamma_G \\ 0_{d-n,n} \end{bmatrix},$$
$$D_1 = \begin{bmatrix} \alpha_1 & & \\ & \ddots & \\ & & \alpha_n \end{bmatrix}, \quad D_2 = \begin{bmatrix} \beta_1 & & \\ & \ddots & \\ & & \beta_n \end{bmatrix}, \text{ and } \quad \Gamma_G = \begin{bmatrix} \gamma_1 & & \\ & \ddots & \\ & & \gamma_n \end{bmatrix}. \tag{7.5}$$

Note that, in view of the assumption that B and G are of full rank and $m \geq n$, $1 > \alpha_i \geq \alpha_{i+1} > 0$, $1 > \gamma_i \geq \gamma_{i+1} > 0$, and $0 < \beta_i \leq \beta_{i+1} < 1$.

Remark 7.1. *Let $\Sigma_2 = \mathsf{diag}(\sigma_1, \ldots, \sigma_n)$, for $i = 1, \ldots, n$. As mentioned earlier, Γ_G is a scaling matrix one can freely choose. Given Σ_2, we may, for instance, choose $\gamma_i = \sigma_i\,(\sigma_i^2+1)^{-1/2}$, which are nonzero and ordered nonincreasingly (since the function*

$t \mapsto t\,(t^2+1)^{-1/2}$ is strictly increasing). This implies that $\alpha_i = \sigma_i^2\,(\sigma_i^2+1)^{-1/2}$. Given that $\beta_i^2 + \sigma_i^2 = 1$ from the second GSVD, we have that $\alpha_i^2 + \beta_i^2 + \gamma_i^2 = 1$ for $i = 1, \ldots, n$. Note that $\frac{\alpha_i}{\beta_i \gamma_i} \geq \frac{\alpha_{i+1}}{\beta_{i+1}\gamma_{i+1}}$, which follows from the fact that $\alpha_i/\gamma_i = \sigma_i$, which are nonincreasing.

We now state a connection of the RSVD with CCA, which is one of the motivations for our proposed decomposition. In [27], De Moor and Golub show a relation of the RSVD to a generalized eigenvalue problem. The related generalized eigenvalue problem of the RSVD of the matrix triplet $(B^\top G, B^\top, G)$ with $m = d$ as shown in [27, Sec. 2.2] is

$$\begin{bmatrix} & B^\top G \\ G^\top B & \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{w} \end{bmatrix} = \lambda \begin{bmatrix} B^\top B & \\ & G^\top G \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{w} \end{bmatrix}.$$

The above problem is exactly the generalized eigenvalue problem of the $\mathsf{cca}(B, G)$ (see, e.g., [49]). Note that matrices $B^\top B$ and $G^\top G$ can be interpreted as covariance matrices. In applications where these covariance matrices are (nearly) singular, one may use the RSVD instead to find a solution without explicitly solving the generalized eigenvalue problem.

7.4 RSVD-CUR DECOMPOSITION AND ITS APPROXIMATION PROPERTIES

This section describes the proposed RSVD-CUR decomposition and provides theoretical bounds on its approximation errors.

7.4.1 *A restricted SVD based CUR decomposition*

We now introduce a new RSVD-CUR decomposition of a matrix triplet (A, B, G) with $A \in \mathbb{R}^{m\times n}$ (where, without loss of generality, $m \geq n$), $B \in \mathbb{R}^{m\times \ell}$, and $G \in \mathbb{R}^{d\times n}$ where B and G are of full rank. This RSVD-CUR factorization is guided by the knowledge of the RSVD for matrix triplets reviewed in Section 7.3. We now define a rank-k RSVD-CUR approximation; cf. (2.3).

Definition 7.2. *Let A be $m \times n$, B be $m \times \ell$, and G be $d \times n$. A rank-k RSVD-CUR approximation of (A, B, G) is defined as*

$$\begin{aligned} A &\approx A_k := C_A\, M_A\, R_A := AP\, M_A\, S^\top A\,, \\ B &\approx B_k := C_B\, M_B\, R_B := BP_B\, M_B\, S^\top B\,, \\ G &\approx G_k := C_G\, M_G\, R_G := GP\, M_G\, S_G^\top G. \end{aligned} \tag{7.6}$$

Here $S \in \mathbb{R}^{m\times k}$, $S_G \in \mathbb{R}^{d\times k}$, $P \in \mathbb{R}^{n\times k}$, *and* $P_B \in \mathbb{R}^{\ell\times k}$ $(k \le \min(m,n,d,\ell))$ *are index selection matrices with some columns of the identity that select rows and columns of the respective matrices.*

It is key that *the same* rows of A and B are picked and *the same* columns of A and G are selected; this gives a coupling among the decompositions. The matrices $C_A \in \mathbb{R}^{m\times k}$, $C_B \in \mathbb{R}^{m\times k}$, $C_G \in \mathbb{R}^{d\times k}$, and $R_A \in \mathbb{R}^{k\times n}$, $R_B \in \mathbb{R}^{k\times \ell}$, $R_G \in \mathbb{R}^{k\times n}$ are subsets of the columns and rows, respectively, of the given matrices. Let the vectors $\mathbf{s}$, $\mathbf{s}_G$, $\mathbf{p}$, and $\mathbf{p}_B$ contain the indices of the selected rows and columns so that $S = I(:,\mathbf{s})$, $S_G = I(:,\mathbf{s}_G)$, $P = I(:,\mathbf{p})$, and $P_B = I(:,\mathbf{p}_B)$. The choice of $\mathbf{s}$, $\mathbf{s}_G$, $\mathbf{p}$, and $\mathbf{p}_B$ is guided by the knowledge of the orthogonal and nonsingular matrices from the rank-k RSVD. Given the column and row index vectors, following [72, 83, 85], we compute the middle matrices as $M_A = (C_A^\top C_A)^{-1} C_A^\top A R_A^\top (R_A R_A^\top)^{-1}$, and similarly for M_B and M_G. There are several index selection strategies proposed in the literature for finding the "best" row and column indices. The approaches we employ are the DEIM [16] and the QDEIM [34] algorithms, which are introduced in Chapter 2.

A DEIM type CUR decomposition requires singular vectors or approximate singular vectors. In this chapter, we apply the DEIM procedure or its variant QDEIM to the nonsingular and orthogonal matrices from the RSVD instead. In an SVD-based CUR factorization, the left and right singular vectors serve as bases for the column and row spaces of matrix A, respectively. In our new context, the columns of matrices Z and W from (7.4) may be viewed as bases for the column and row spaces, respectively, of A relative to the column space of B and the row space of G. The procedure for constructing a DEIM type RSVD-CUR is summarized in Algorithm 19.

Algorithm 19: DEIM type RSVD-CUR decomposition

Data: $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times \ell}$, $G \in \mathbb{R}^{d\times n}$, desired rank k

Result: A rank-k RSVD-CUR decomposition $A_k = A(:,\mathbf{p}) \cdot M_A \cdot A(\mathbf{s},:)$, $B_k = B(:,\mathbf{p}_B) \cdot M_B \cdot B(\mathbf{s},:)$, $G_k = G(:,\mathbf{p}) \cdot M_G \cdot G(\mathbf{s}_G,:)$

1 Compute rank-k RSVD of (A,B,G) to obtain W,Z,U,V (see (7.4))

2 $\mathbf{p} = \mathsf{deim}(W)$ (perform DEIM on the matrices from the RSVD)

3 $\mathbf{s} = \mathsf{deim}(Z)$

4 $M_A = A(:,\mathbf{p}) \setminus (A \,/\, A(\mathbf{s},:))$

5 $\mathbf{p}_B = \mathsf{deim}(U)$ (optional)

6 $\mathbf{s}_G = \mathsf{deim}(V)$ (optional)

7 Compute M_B and M_G as in Line 4 if needed

In this implementation, the user is supposed to specify k. We note that one can also determine k by comparing the decaying restricted singular values $\frac{\alpha_i}{\beta_i \gamma_i}$ against a given threshold. In some applications, the explicit approximation of B or G may not be necessary. Thus, Lines 5, 6, and 7 in Algorithm 19 should only be implemented if necessary.

Remark 7.3. *(also see Remark 6.5)* In Line 1 of Algorithm 19, the columns of W, Z, U, and V corresponds to the k largest restricted singular values $\frac{\alpha_i}{\beta_i \gamma_i}$. This implies that we select the most "dominant" parts of A and the least "dominant" parts of B and G, so in (7.6) the relative approximation error of A tends to be modest, while this may not be the case for the relative approximation errors of B and G.

In many applications, as we will see in Section 7.5, one is interested in selecting only the key columns or rows and not the explicit $A \approx C_A M_A R_A$ factorization. An interpolative decomposition aims to identify a set of skeleton columns or rows of a matrix. A CUR factorization may be viewed as evaluating the ID for both the column and row spaces of a matrix simultaneously. The following are the column and row versions of an RSVD-ID factorization of a matrix triplet:

$$\begin{aligned} &A \approx C_A \widetilde{M}_A, \quad B \approx C_B \widetilde{M}_B, \quad G \approx C_G \widetilde{M}_G, \quad \text{or} \\ &A \approx \widehat{M}_A R_A, \quad B \approx \widehat{M}_B R_B, \quad G \approx \widehat{M}_G R_G. \end{aligned} \tag{7.7}$$

Here, $\widetilde{M}_A = C_A^+ A$ is $k \times n$ and $\widehat{M}_A = A R_A^+$ is $m \times k$; analogous remarks hold for $\widetilde{M}_B$, $\widetilde{M}_G$, $\widehat{M}_B$, and $\widehat{M}_G$. Notice that in Algorithm 19, the key column and row indices of the various matrices are picked independently. This algorithm can therefore be restricted to select only column indices if we are interested in the column version of the RSVD-ID factorization or select only row indices if we are interested in the row version.

De Moor and Golub [27] show the relation between the RSVD and the SVD and its other generalizations. We indicate in the following proposition the corresponding connection between a DEIM type RSVD-CUR and the (generalized) CUR decomposition [42, 83].

Proposition 7.4. *(i) If B and G are nonsingular matrices, then the selected row and column indices from a CUR decomposition of $B^{-1}AG^{-1}$ are the same as index vectors $\mathbf{p}_B$ and $\mathbf{s}_G$, respectively, obtained from an RSVD-CUR decomposition of (A, B, G).*

(ii) Furthermore, in the particular case where $B = I$ and $G = I$, an RSVD-CUR decomposition of A coincides with a CUR decomposition of A, in that the factors C and R of A are the same for both methods: the first line of (7.6) is equal to (2.3).

(iii) Lastly, given a special choice of $B = I$, an RSVD-CUR decomposition of A and G coincides with a GCUR decomposition of (A, G) (see (6.6)), in that the factors C_A, C_G and R_A, R_G of A and G are the same for both methods. In the dual case that $G = I$, similar remarks hold.

Proof. (i) We start with the RSVD (7.4). If B and G are nonsingular, then the SVD of $B^{-1}AG^{-1}$ can be expressed in terms of the RSVD of (A, B, G), and is equal to $U(D_B^{-1}D_A D_G^{-1})V^\top$ given that $B^{-1} = UD_B^{-1}Z^{-1}$ and $G^{-1} = W^{-T}D_G^{-1}V^\top$ [27]. Consequently, the row and column index vectors from a CUR factorization of $B^{-1}AG^{-1}$ are equal to the vectors $\mathbf{s}_G$ and $\mathbf{p}_B$, respectively, from an RSVD-CUR of (A, B, G) since they are obtained by applying DEIM to matrices V and U, respectively.

(ii) If $B = I$ and $G = I$, from (7.4), $I = ZD_B U^\top$ and $I = VD_G W^\top$ which implies $UD_B^{-1} = Z$ and $W^\top = D_G^{-1}V^\top$. Hence, we find that $A = UD_B^{-1}D_A D_G^{-1}V^\top$ which is an SVD of A. Therefore the selection matrices P, S from CUR of A (2.3) are equal to the selection matrices $P = P_B$, $S = S_G$ from an RSVD-CUR of (A, I, I) (7.6).

(iii) If $B = I$, again from (7.4), $I = ZD_B U^\top$, which implies that $UD_B^{-1} = Z$. Then $A = UD_B^{-1}D_A W^\top$, $G = VD_G W^\top$ which is (up to a diagonal scaling) the GSVD of the matrix pair (A, G); see (6.1) [27]. Thus, the column and row selection matrices from GCUR of (A, G) (see (6.6)) are the same as the column and row selection matrices P, S, S_G from (7.6), respectively. ∎

7.4.2 *Error analysis*

We begin by analyzing the error of a rank-k RSVD of a matrix triplet $A \in \mathbb{R}^{m\times n}$ (where without loss of generality $m \geq n$), $B \in \mathbb{R}^{m\times \ell}$, and $G \in \mathbb{R}^{d\times n}$ (where B and G are of full rank). Given our applications of interest in Section 7.5, we consider the case $\ell = d \geq m \geq n$. To define a rank-k RSVD, let us partition the following matrices

$$U = [U_k\ \widehat{U}]\,,\ V = [V_k\ \widehat{V}]\,,\ W = [W_k\ \widehat{W}]\,,\ Z = [Z_k\ \widehat{Z}]\,,$$
$$D_A = \mathsf{diag}(D_{A_k}, \widehat{D}_A)\,,\ D_B = \mathsf{diag}(D_{B_k}, \widehat{D}_B)\,,\ D_G = \mathsf{diag}(D_{G_k}, \widehat{D}_G),$$

where $\widehat{D}_A \in \mathbb{R}^{(m-k)\times(n-k)}$, $\widehat{D}_B \in \mathbb{R}^{(m-k)\times(\ell-k)}$, and $\widehat{D}_G \in \mathbb{R}^{(d-k)\times(n-k)}$. We define a rank-$k$ RSVD of (A, B, G) as

$$A_k := Z_k D_{A_k} W_k^\top, \quad B_k := Z_k D_{B_k} U_k^\top, \quad G_k := V_k D_{G_k} W_k^\top, \tag{7.8}$$

where $k < n$. It follows that

$$A - A_k = \widehat{Z}\,\widehat{D}_A\,\widehat{W}^\top, \quad B - B_k = \widehat{Z}\,\widehat{D}_B\,\widehat{U}^\top, \quad G - G_k = \widehat{V}\,\widehat{D}_G\,\widehat{W}^\top. \tag{7.9}$$

The following statements are a stepping stone for the error bound analysis of an RSVD-CUR. Denote the ith singular value of A by $\psi_i(A)$. Let $A - A_k = \widehat{Z}\,\widehat{D}_A\,\widehat{W}^\top$ as in (7.9). Then, for $i = 1, \ldots, n$, $\psi_i(\widehat{Z}\,\widehat{D}_A\,\widehat{W}^\top) \le \psi_i(\widehat{D}_A)\,\|\widehat{Z}\|\,\|\widehat{W}\|$ (see, e.g., [63, p. 346]). Since the diagonal elements of $\widehat{D}_A$ are in nonincreasing order, we have $\|A - A_k\| \le \psi_1(\widehat{D}_A)\,\|\widehat{Z}\|\,\|\widehat{W}\| \le \alpha_{k+1} \cdot \|\widehat{Z}\|\,\|\widehat{W}\|$ (see (7.5) for the structure of D_A).

Similarly, we have that $\|B - B_k\| = \|\widehat{Z}\,\widehat{D}_B\,\widehat{U}^\top\| \le \|\widehat{Z}\|$ and $\|G - G_k\| = \|\widehat{V}\,\widehat{D}_G\,\widehat{W}^\top\| \le \gamma_{k+1} \cdot \|\widehat{W}\|$. The first inequality follows from the fact $\widehat{U}$ has orthonormal columns and the diagonal elements of $\widehat{D}_B$ are in nondecreasing order with a maximum value of 1, so we have that $\psi_1(\widehat{D}_B) = 1$ (see (7.5) for the structure of D_B) and $\|\widehat{U}\| = 1$. The second equality is a result of the fact that $\widehat{V}$ has orthonormal columns and the diagonal entries of $\widehat{D}_G$ are in nonincreasing order, therefore, $\psi_1(\widehat{D}_G) = \gamma_{k+1}$ (see (7.5) for the structure of D_G) and $\|\widehat{V}\| = 1$.

We now introduce some error bounds of an RSVD-CUR decomposition in terms of the error of a rank-k RSVD. The analysis closely follows the error bound analysis in [83] and Chapter 6 for the DEIM type CUR and DEIM type GCUR methods with some necessary modifications. As with a DEIM type GCUR method, here also, the lack of orthogonality of the vectors in W and Z from the RSVD necessitates some additional work. We take QR factorizations of W and Z to obtain their respective orthonormal bases to facilitate the analysis, introducing terms in the error bound associated with the triangular matrix in the QR factorizations.

For the analysis, we use the following QR decompositions of the nonsingular matrices from the RSVD (see (7.4))

$$\begin{aligned}
[Z_k \;\; \widehat{Z}] = Z = Q_Z T_Z = [Q_{Z_k} \;\; \widehat{Q}_Z] \begin{bmatrix} T_{Z_k} & T_{Z_{12}} \\ 0 & T_{Z_{22}} \end{bmatrix} = [Q_{Z_k} T_{Z_k} \;\; Q_Z \widehat{T}_Z], \\
[W_k \;\; \widehat{W}] = W = Q_W T_W = [Q_{W_k} \;\; \widehat{Q}_W] \begin{bmatrix} T_{W_k} & T_{W_{12}} \\ 0 & T_{W_{22}} \end{bmatrix} = [Q_{W_k} T_{W_k} \;\; Q_W \widehat{T}_W],
\end{aligned} \tag{7.10}$$

where we have defined

$$\widehat{T}_Z := \begin{bmatrix} T_{Z_{12}} \\ T_{Z_{22}} \end{bmatrix}, \qquad \widehat{T}_W := \begin{bmatrix} T_{W_{12}} \\ T_{W_{22}} \end{bmatrix}. \tag{7.11}$$

This implies that

$$\begin{aligned} A &= A_k + \widehat{Z}\,\widehat{D}_A\,\widehat{W}^\top = Z_k D_{A_k} W_k^\top + \widehat{Z}\,\widehat{D}_A\,\widehat{W}^\top \\ &= Q_{Z_k} T_{Z_k} D_{A_k} T_{W_k}^\top Q_{W_k}^\top + Q_Z \widehat{T}_Z\,\widehat{D}_A\,\widehat{T}_W^\top Q_W^\top, \\ B &= B_k + \widehat{Z}\,\widehat{D}_B\,\widehat{U}^\top = Z_k D_{B_k} U_k^\top + \widehat{Z}\,\widehat{D}_B\,\widehat{U}^\top = Q_{Z_k} T_{Z_k} D_{B_k} U_k^\top + Q_Z \widehat{T}_Z\,\widehat{D}_B\,\widehat{U}^\top, \\ G &= G_k + \widehat{V}\,\widehat{D}_G\,\widehat{W}^\top = V_k D_{G_k} W_k^\top + \widehat{V}\,\widehat{D}_G\,\widehat{W}^\top = V_k D_{G_k} T_{W_k}^\top Q_{W_k}^\top + \widehat{V}\,\widehat{D}_G\,\widehat{T}_W^\top Q_W^\top. \end{aligned} \tag{7.12}$$

Given $Q_W \in \mathbb{R}^{n\times k}$ with orthonormal columns, from [83] and Algorithm 17 as well as here, we have that the quantity $\|A(I - Q_{W_k} Q_{W_k}^\top)\|$ is key in the error bound analysis. Here, we have that $\|A(I - Q_{W_k} Q_{W_k}^\top)\|$ may not be close to $\psi_{k+1}(A)$ since the matrix Q_{W_k} is from the RSVD, therefore we provide a bound on this quantity in terms of the error in the RSVD.

Let P be an index selection matrix derived from performing the DEIM scheme on matrix W_k. Suppose Q_{W_k} is an orthonormal basis for Range(W_k), with $W_k^\top P$ and $Q_{W_k}^\top P$ being nonsingular, we have the interpolatory projector $P(W_k^\top P)^{-1} W_k^\top = P(Q_{W_k}^\top P)^{-1} Q_{W_k}^\top$ (see Proposition 2.4 and also [16, Def. 3.1, (3.6)]). With this equality, we exploit the special properties of an orthogonal matrix by using the orthonormal bases of the nonsingular matrices from the RSVD instead for our analysis. Let $Q_{W_k}^\top P$ and $S^\top Q_{Z_k}$ be nonsingular so that $\mathbb{P} = P(Q_{W_k}^\top P)^{-1} Q_{W_k}^\top$ and $\mathbb{S} = Q_{Z_k}(S^\top Q_{Z_k})^{-1} S^\top$ are oblique projectors.

In the following theorem, we provide bounds on the coupled CUR decompositions of A, B, and G in terms of the RSVD quantities. The upper bounds contain both multiplicative factors (the η's) and the α_{k+1}, γ_{k+1} (both bounded by 1), and T-quantities, which are from the error of the truncated RSVD as defined in (7.8) and (7.9).

Theorem 7.5. *(Generalization of [83, Thm. 4.1] and Theorem 6.11) Given A, B, and G as in Definition 7.2 and $Z_k \in \mathbb{R}^{m\times k}$, $W_k \in \mathbb{R}^{n\times k}$, $U_k \in \mathbb{R}^{\ell\times k}$, and $V_k \in \mathbb{R}^{d\times k}$ from (7.8), let $Q_{Z_k} \in \mathbb{R}^{m\times k}$, $Q_{W_k} \in \mathbb{R}^{n\times k}$ be the Q-factors of Z_k, W_k, respectively, and $\widehat{T}_Z$, $T_{Z_{22}}$, $\widehat{T}_W$, and $T_{W_{22}}$ as in (7.10)–(7.11). Suppose $Q_{W_k}^\top P$, $U_k^\top P_B$, $S_G^\top V_k$, and $S^\top Q_{Z_k}$ are nonsingular, then with the error constants (see Remark 7.1 for details on the quantities α_{k+1} and γ_{k+1})*

$\eta_p := \|(Q_{W_k}^\top P)^{-1}\|$, $\eta_s := \|(S^\top Q_{Z_k})^{-1}\|$, $\eta_{p_B} := \|(U_k^\top P_B)^{-1}\|$, $\eta_{s_G} := \|(S_G^\top V_k)^{-1}\|$,

we have

$$\begin{aligned}
\|A - C_A M_A R_A\| &\le \alpha_{k+1} \cdot (\eta_p \cdot \|\widehat{T}_Z\| \, \|T_{W_{22}}\| + \eta_s \cdot \|T_{Z_{22}}\| \, \|\widehat{T}_W\|) \\
&\le \alpha_{k+1} \cdot (\eta_p + \eta_s) \cdot \|\widehat{T}_W\| \, \|\widehat{T}_Z\| \,, \\
\|B - C_B M_B R_B\| &\le \eta_{p_B} \cdot \|T_{Z_{22}}\| + \eta_s \cdot \|\widehat{T}_Z\| \le (\eta_{p_B} + \eta_s) \cdot \|\widehat{T}_Z\| \,, \\
\|G - C_G M_G R_G\| &\le \gamma_{k+1} \cdot (\eta_p \cdot \|\widehat{T}_W\| + \eta_{s_G} \cdot \|T_{W_{22}}\|) \\
&\le \gamma_{k+1} \cdot (\eta_p + \eta_{s_G}) \cdot \|\widehat{T}_W\|.
\end{aligned} \tag{7.13}$$

Proof. We will prove the result for $\|A - C_A M_A R_A\|$; the results for $\|B - C_B M_B R_B\|$ and $\|G - C_G M_G R_G\|$ follow similarly. We first show the bounds on the errors between A and its interpolatory projections $\mathbb{P}A$ and $A\mathbb{S}$, i.e., the selected rows and columns. Then, using the fact that these bounds also apply to the orthogonal projections of A onto the same column and row spaces [83, Lemma 4.2], we prove the bound on the approximation of A by an RSVD-CUR.

Given the projector $\mathbb{P}$, we have that $Q_{W_k}^\top \mathbb{P} = Q_{W_k}^\top P (Q_{W_k}^\top P)^{-1} Q_{W_k}^\top = Q_{W_k}^\top$, which implies $Q_{W_k}^\top (I - \mathbb{P}) = 0$. Therefore the error in the oblique projection of A is (cf. [83, Lemma 4.1])

$$\begin{aligned}
\|A - A\mathbb{P}\| &= \|A(I - \mathbb{P})\| = \|A(I - Q_{W_k} Q_{W_k}^\top)(I - \mathbb{P})\| \\
&\le \|A(I - Q_{W_k} Q_{W_k}^\top)\| \, \|I - \mathbb{P}\|.
\end{aligned}$$

Note that, since $k < n$, $\mathbb{P} \neq 0$ and $\mathbb{P} \neq I$, it is well known that (see, e.g., [89])

$$\|I - \mathbb{P}\| = \|\mathbb{P}\| = \|(Q_{W_k}^\top P)^{-1}\|.$$

Using the partitioning of A in (7.12), we have

$$\begin{aligned}
A \, Q_{W_k} \, Q_{W_k}^\top &= [Q_{Z_k} \;\; \widehat{Q}_Z] \begin{bmatrix} T_{Z_k} & T_{Z_{12}} \\ 0 & T_{Z_{22}} \end{bmatrix} \begin{bmatrix} D_{A_k} & 0 \\ 0 & \widehat{D}_A \end{bmatrix} \begin{bmatrix} T_{W_k}^\top & 0 \\ T_{W_{12}}^\top & T_{W_{22}}^\top \end{bmatrix} \begin{bmatrix} I_k \\ 0 \end{bmatrix} Q_{W_k}^\top \\
&= Q_{Z_k} T_{Z_k} D_{A_k} T_{W_k}^\top Q_{W_k}^\top + Q_Z \widehat{T}_Z \widehat{D}_A \, T_{W_{12}}^\top Q_{W_k}^\top,
\end{aligned}$$

and hence

$$\begin{aligned} A\,(I - Q_{W_k} Q_{W_k}^\top) &= (A - A_k) - Q_Z \widehat{T}_Z \widehat{D}_A\, T_{W_{12}}^\top Q_{W_k}^\top \\ &= Q_Z \widehat{T}_Z \widehat{D}_A\, \widehat{T}_W^\top Q_W^\top - Q_Z \widehat{T}_Z \widehat{D}_A\, T_{W_{12}}^\top Q_{W_k}^\top \\ &= Q_Z \widehat{T}_Z \widehat{D}_A\, T_{W_{22}}^\top \widehat{Q}_W^\top. \end{aligned}$$

This implies

$$\|A\,(I - Q_{W_k} Q_{W_k}^\top)\| \le \|\widehat{D}_A\|\,\|\widehat{T}_Z\|\,\|T_{W_{22}}\| \le \alpha_{k+1} \cdot \|\widehat{T}_Z\|\,\|T_{W_{22}}\|,$$

and

$$\|A(I - \mathbb{P})\| \le \alpha_{k+1} \cdot \|(Q_{W_k}^\top P)^{-1}\|\,\|\widehat{T}_Z\|\,\|T_{W_{22}}\|.$$

Let us now consider the operation on the left side of A. Given that $S^\top Q_{Z_k}$ is nonsingular, we have the DEIM interpolatory projector $\mathbb{S} = Q_{Z_k}(S^\top Q_{Z_k})^{-1} S^\top$. It is known that (see [83, Lemma 4.1])

$$\begin{aligned} \|A - \mathbb{S}A\| = \|(I - \mathbb{S})A\| &= \|(I - \mathbb{S})(I - Q_{Z_k} Q_{Z_k}^\top)A\| \\ &\le \|(I - \mathbb{S})\|\,\|(I - Q_{Z_k} Q_{Z_k}^\top)A\|. \end{aligned}$$

Similar to before, since $k < m$, we know that $\mathbb{S} \neq 0$ and $\mathbb{S} \neq I$ hence

$$\|I - \mathbb{S}\| = \|\mathbb{S}\| = \|(S^\top Q_{Z_k})^{-1}\|.$$

In the same setting as earlier, we have the following expansion

$$\begin{aligned} Q_{Z_k} Q_{Z_k}^\top A &= [Q_{Z_k} \;\; 0] \begin{bmatrix} T_{Z_k} & T_{Z_{12}} \\ 0 & T_{Z_{22}} \end{bmatrix} \begin{bmatrix} D_{A_k} & 0 \\ 0 & \widehat{D}_A \end{bmatrix} \begin{bmatrix} T_{W_k}^\top & 0 \\ T_{W_{12}}^\top & T_{W_{22}}^\top \end{bmatrix} \begin{bmatrix} Q_{W_k}^\top \\ \widehat{Q}_W^\top \end{bmatrix} \\ &= Q_{Z_k} T_{Z_k} D_{A_k} T_{W_k}^\top Q_{W_k}^\top + Q_{Z_k} T_{Z_{12}} \widehat{D}_A\, \widehat{T}_W^\top Q_W^\top. \end{aligned}$$

We observe that

$$\begin{aligned} (I - Q_{Z_k} Q_{Z_k}^\top)A &= (A - A_k) - Q_{Z_k} T_{Z_{12}} \widehat{D}_A\, \widehat{T}_W^\top Q_W^\top \\ &= Q_Z \widehat{T}_Z\, \widehat{D}_A\, \widehat{T}_W^\top Q_W^\top - Q_{Z_k} T_{Z_{12}} \widehat{D}_A\, \widehat{T}_W^\top Q_W^\top \\ &= \widehat{Q}_Z\, T_{Z_{22}} \widehat{D}_A\, \widehat{T}_W^\top Q_W^\top. \end{aligned}$$

Consequently,

$$\begin{aligned}\|(I - Q_{Z_k} Q_{Z_k}^\top)A\| &= \|\widehat{Q}_Z T_{Z_{22}} \widehat{D}_A \widehat{T}_W^\top Q_W^\top\| \le \|\widehat{D}_A\| \|T_{Z_{22}}\| \, \|\widehat{T}_W\| \\ &\le \alpha_{k+1} \cdot \|T_{Z_{22}}\| \, \|\widehat{T}_W\|,\end{aligned}$$

and

$$\|(I - \mathbb{S})A\| \le \alpha_{k+1} \cdot \|(S^\top Q_{Z_k})^{-1}\| \, \|T_{Z_{22}}\| \, \|\widehat{T}_W\|.$$

Suppose that C_A and R_A are of full rank. Given the orthogonal projectors $C_A C_A^+$ and $R_A^+ R_A$ and computing M_A as $(C_A^\top C_A)^{-1} C_A^\top A R_A^\top (R_A R_A^\top)^{-1} = C_A^+ A R_A^+$, we have (see [72, (6)])

$$A - C_A M_A R_A = A - C_A C_A^+ A R_A^+ R_A = (I - C_A C_A^+)A + C_A C_A^+ A (I - R_A^+ R_A).$$

Using the triangle inequality, it follows that [72, 83]

$$\begin{aligned}\|A - C_A M_A R_A\| &= \|A - C_A C_A^+ A R_A^+ R_A\| \\ &\le \|(I - C_A C_A^+)A\| + \|C_A C_A^+\| \, \|A(I - R_A^+ R_A)\|.\end{aligned}$$

Leveraging the fact that $C_A C_A^+$ is an orthogonal projector so $\|C_A C_A^+\| = 1$, and [83, Lemma 4.2]

$$\|(I - C_A C_A^+)A\| \le \|A(I - \mathbb{P})\|, \quad \|A(I - R_A^+ R_A)\| \le \|(I - \mathbb{S})A\|,$$

as a variant of Theorem 6.11 we have

$$\begin{aligned}\|A - C_A M_A R_A\| &\le \alpha_{k+1} \cdot (\|\widehat{T}_Z\| \|T_{W_{22}}\| \|(Q_{W_k}^\top P)^{-1}\| + \|(S^\top Q_{Z_k})^{-1}\| \|T_{Z_{22}}\| \|\widehat{T}_W\|) \\ &\le \alpha_{k+1} \cdot (\|(Q_{W_k}^\top P)^{-1}\| + \|(S^\top Q_{Z_k})^{-1}\|) \cdot \|\widehat{T}_Z\| \, \|\widehat{T}_W\|.\end{aligned}$$

The last inequality follows directly from the fact that the norms of the submatrices of $\widehat{T}_{Z_{22}}$ and $\widehat{T}_{W_{22}}$ are at most $\|\widehat{T}_Z\|$ and $\|\widehat{T}_W\|$, respectively. ■

Theorem 7.5 suggests that to keep the approximation errors as small as possible, a good index selection procedure that provides smaller quantities $\|(U_k^\top P_B)^{-1}\|$, $\|(S_G^\top V_k)^{-1}\|$, $\|(Q_{W_k}^\top P)^{-1}\|$, and $\|(S^\top Q_{Z_k})^{-1}\|$ would be ideal. The DEIM procedure may be seen as an attempt to attain exactly that. Meanwhile, the quantity

$\alpha_{k+1} \cdot \|\widehat{T}_Z\| \, \|\widehat{T}_W\|$ is a result of the error of the rank-k RSVD. Additionally, we would like to point out that due to the connection of QDEIM with strong rank-revealing QR factorization (matrix volume maximization), the upper bounds for the error constants can be reduced to the theoretically best available one if the QDEIM procedure for the selection of the indices is used instead of the DEIM algorithm [34, 35].

Comparing the results of the decomposition of A in Theorem 7.5 to [83, Thm. 4.1], we have that the σ_{k+1} in [83, Thm. 4.1] is replaced by the error in the RSVD through $\|(I - Q_{Z_k} Q_{Z_k}^\top)A\|$ and $\|A(I - Q_{W_k} Q_{W_k}^\top)\|$. Here, $\|(Q_{W_k}^\top P)^{-1}\|$, and $\|(S^\top Q_{Z_k})^{-1}\|$ are computed using the orthonormal bases of the nonsingular matrices from the RSVD of A rather than the singular vectors. Compared with the results in Theorem 6.11 where all quantities are a result of the GSVD, in Theorem 7.5 we have an additional $\|\widehat{T}_Z\|$, and all quantities are a result of the RSVD.

7.5 NUMERICAL EXPERIMENTS

In this section, we first evaluate the performance of the proposed RSVD-CUR decomposition for reconstructing a data matrix perturbed with nonwhite noise. We then show how the proposed algorithm can be used for feature selection in multi-view classification problems. In Experiment 7.8, we only care about the key columns of B and G so we do not explicitly compute the RSVD-CUR factorization. We use the DEIM and the QDEIM index selection scheme interchangeably.

In Experiments 7.6 and 7.7, we consider two popular covariance structures [95] for the correlated noise. The first covariance matrix has a compound symmetry structure (CSS), which means the covariance matrix has constant diagonal and constant off-diagonal entries. A homogeneous compound symmetry covariance matrix is of the form

$$\mathrm{CSS}_{\mathrm{cov}} = \nu^2 \begin{bmatrix} 1 & \xi & \xi & \xi \\ \xi & 1 & \xi & \xi \\ \xi & \xi & 1 & \xi \\ \xi & \xi & \xi & 1 \end{bmatrix}, \tag{7.14}$$

where ν and ξ (with $-1 < \xi < 1$) denote the variance and correlation coefficient, respectively. The second covariance matrix we use has a first-order autoregressive structure (AR(1)), which implies the matrix has a constant diagonal and the off-diagonal entries decaying exponentially; it assumes that the correlation between

any two elements gets smaller the further apart they are separated (e.g., in terms of time or space). A homogeneous AR(1) covariance matrix is of the form

$$\mathrm{AR}(1)_{\mathrm{cov}} = \nu^2 \begin{bmatrix} 1 & \xi & \xi^2 & \xi^3 \\ \xi & 1 & \xi & \xi^2 \\ \xi^2 & \xi & 1 & \xi \\ \xi^3 & \xi^2 & \xi & 1 \end{bmatrix}. \tag{7.15}$$

Experiment 7.6. In our first experiment, we address the problem of matrix perturbation (The case $A_E = A + E$ of unstructured noise was already considered in Chapter 6), specifically $A_E = A + BFG$, where F is a Gaussian random matrix and B and G are the Cholesky factors of non-diagonal noise covariance matrices. The goal is to reconstruct a low-rank matrix A from A_E assuming that the noise covariance matrices or their estimates are known. The requirement that the noise covariance matrices or their estimates should be known is not always trivial. We evaluate and compare a rank-k RSVD-CUR and CUR decomposition of A_E in terms of reconstructing matrix A. The approximation quality of each decomposition is assessed by the relative matrix approximation error, i.e., $\|A - \widetilde{A}\| \,/\, \|A\|$, where $\widetilde{A}$ is the reconstructed low-rank matrix. As an adaptation of the first experiment in [83, Ex. 6.1] we generate a rank-100 sparse nonnegative matrix $A \in \mathbb{R}^{10000\times 1000}$ of the form

$$A = \sum_{j=1}^{10} \frac{2}{j}\, \mathbf{x}_j\, \mathbf{y}_j^\top + \sum_{j=11}^{100} \frac{1}{j}\, \mathbf{x}_j\, \mathbf{y}_j^\top,$$

where $\mathbf{x}_j \in \mathbb{R}^{10000}$ and $\mathbf{y}_j \in \mathbb{R}^{1000}$ are random sparse vectors with nonnegative entries (in Matlab, $\mathbf{x}_j$=sprand$(10000, 1, 0.025)$ and $\mathbf{y}_j$=sprand$(1000, 1, 0.025)$). We then perturb A with a nonwhite noise matrix BFG (see, e.g., [57, p. 55]). The matrix $F \in \mathbb{R}^{10000\times 1000}$ is random Gaussian noise. We assume that $B \in \mathbb{R}^{10000\times 10000}$ is the Cholesky factor of a symmetric positive definite covariance matrix with compound symmetry structure (7.14) (with diagonal entries 4 and off-diagonal entries 1), and $G \in \mathbb{R}^{1000\times 1000}$ is the Cholesky factor of a symmetric positive definite covariance matrix with first-order autoregressive structure (7.15) (with diagonal entries 1 and the off-diagonal entries related by a multiplicative factor of 0.99). The resulting perturbed matrix we use is of the form $A_E = A + \varepsilon \frac{\|A\|}{\|BFG\|} BFG$, where ε is the noise level.

Given a noise level, we compare the naive approaches (rank-k SVD-based methods), which do not take the structure of the noise into account with the methods (rank-k RSVD-based methods), which consider the actual noise. That

is, we compute the SVD of A_E and the RSVD of (A_E, B, G) to obtain the input matrices for a CUR and an RSVD-CUR decomposition, respectively. For each noise level, we generate ten random cases and take the average of the relative errors for varying k values.

Figure 7.1 summarizes the results of three noise levels $(0.1, 0.15, 0.2)$. We observe that to approximate A, the RSVD-CUR factorization enjoys a considerably lower average approximation error than the CUR decomposition. The error of the former is at least half of the latter. Meanwhile, the average relative error of an RSVD-CUR approximation unlike that of the RSVD (monotonically decreasing) approaches ε after a certain value of k. This situation is natural because the RSVD-CUR routine picks actual columns and rows of A_E, so the relative error is likely to be saturated by the noise. The rank-k SVD of A_E fails in approximating A for the given values of k. Its average relative error rapidly approaches ε; this is expected since the covariance of the noise is not a multiple of the identity. The truncated SVD of A_E gives the optimal rank-k approximation to A_E. However, here, the goal is to reconstruct the noise-free matrix A hence, we measure the rank-k approximation error against the unperturbed A. To heuristically explain why the SVD fails here, let us consider the ideal situation of a perturbation matrix E whose entries are normally distributed with zero mean and standard deviation ν (white noise). We have that the expected value of $\|E\|$ is approximately $\nu\sqrt{m}$ [57]. Hence if the singular values of unperturbed A decay gradually to zero, then the singular values of the perturbed matrix A_E are expected to decrease monotonically (by definition) until they tend to settle at a level $\tau = \nu\sqrt{m}$ determined by the errors in A_E [57]. Note that given that the rank of A is k, the singular values (ψ_i) of A_E will plateau at $i = k + 1$. We can therefore expect to estimate A in terms of the rank-k SVD A_E [57]. In this experiment, given the structure of the noise $E = BFG$, the expected value of $\|E\|$ is not approximately $\nu\sqrt{m}$. As a result, the rank-k SVD of A_E fails to estimate A. We need algorithms that can take the actual noise into account, which is typically done by the "prewhitening" process. Thus, given the nonsingular error equilibration matrices B and G, we have that the error in $B^{-1}A_E G^{-1}$ is equilibrated, i.e., uncorrelated with zero mean and equal variance (white noise). One may then approximate A by applying the truncated SVD to the matrix $B^{-1}A_E G^{-1}$, followed by a "dewhitening" by a means of left and right multiplication with B and G, respectively [57]. However, it is worth noting that even when B and G are nonsingular it may be computationally risky to work with their inverses since they may be close to singular, and so in general it makes sense to reformulate the problem with no inverses. Using the RSVD formulation, this "prewhitening" and "dewhitening" process becomes an integral part of the algorithm.

Another observation worth pointing out is that a rank-k CUR of A_E yields a more accurate approximation of A compared to the rank-k SVD of A_E, although the column and row indices are selected using the singular vectors of A_E. This behavior can be attributed to the fact that the SVD is a linear combination of all the perturbed data points, so the total noise is included in the rank-k approximation. On the other hand, with the CUR approximation, the C and R factors are actual columns and rows of A_E, so selecting k columns and rows of A_E implies that our approximation would only contain part and not the total noise.

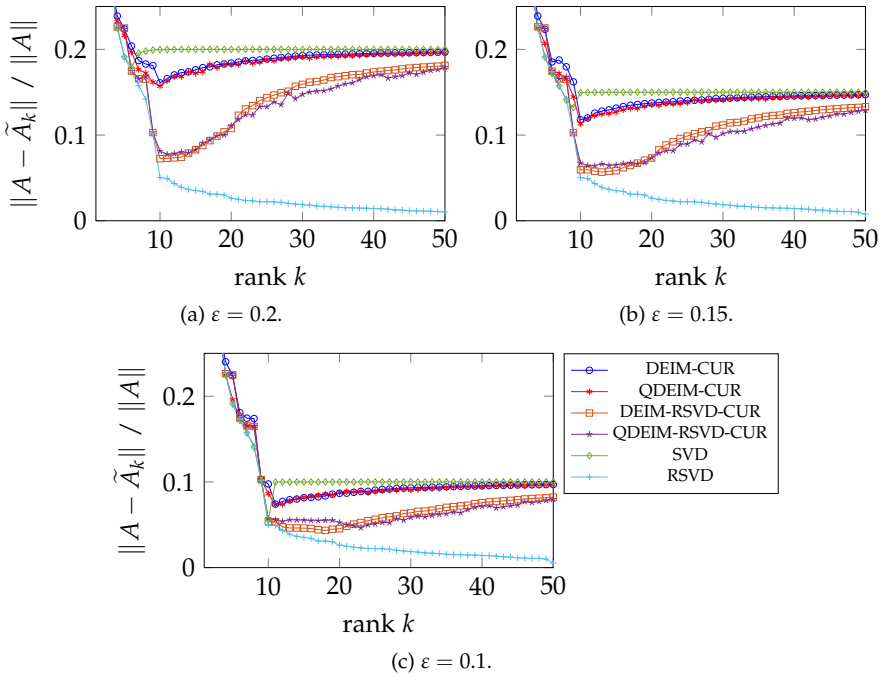
We also observe that the approximation accuracy of the QDEIM index selection method is comparable to that of the DEIM scheme, although the former is simpler and more efficient.

It is also worth noting that the improved performance of an RSVD-CUR approximation compared to a CUR factorization is particularly more attractive for higher noise levels with modest k values, i.e., when k is significantly less than $\text{rank}(A)$. One can observe from Fig. 7.1 that the range of k values that yield the lowest approximation error measured against unperturbed A is between 10 and 20. Using the following k values $(10, 15, 20)$, in Table 7.1, we investigate the significant improvement of a DEIM-RSVD-CUR approximation over a DEIM-CUR factorization. With noise level 0.1, we see that for the lowest error of a DEIM-CUR, which corresponds to $k = 15$, a DEIM-RSVD-CUR produces about a 39% reduction in the error. The improvement is even more significant if the noise level is 0.2, where the rank-10 RSVD-CUR approximation reduces the rank-10 CUR decomposition error (lowest error of the CUR) by about 50%.

Table 7.1: The approximation quality of a DEIM RSVD-CUR approximations compared with DEIM-CUR approximations in recovering a sparse, nonnegative matrix A perturbed with nonwhite noise. The average relative errors $\|A - \widetilde{A}_k\| \,/\, \|A\|$ for k values (10, 15, 20).

Noise	**Method** \ k	10	15	20
$\varepsilon = 0.1$	CUR-$\widetilde{A}_k$	0.100	0.084	0.089
	RSVD-CUR-$\widetilde{A}_k$	0.064	0.051	0.049
$\varepsilon = 0.2$	CUR-$\widetilde{A}_k$	0.162	0.177	0.184
	RSVD-CUR -$\widetilde{A}_k$	0.080	0.084	0.106

In Fig. 7.2, using the DEIM-RSVD-CUR decomposition of A_E with noise level $\varepsilon = 0.1$, we show the various quantities in Theorem 7.5. We observe that the upper bound in Theorem 7.5 may be rather pessimistic, and the true DEIM-RSVD-CUR



(a) $\varepsilon = 0.2$. (b) $\varepsilon = 0.15$.

(c) $\varepsilon = 0.1$.

Figure 7.1: The approximation quality of RSVD-CUR approximations compared with CUR approximations in recovering a sparse, nonnegative matrix A perturbed with nonwhite noise. The average relative errors $\|A - \widetilde{A}_k\| \,/\, \|A\|$ (on the vertical axis) as a function of rank k (on the horizontal axis) for $\varepsilon = 0.2$, 0.15, 0.1, respectively.

error may be substantially lower in practice. As in [83, Fig. 4], the magnitude of the quantities η_s and η_p may vary. We see that $\|\widehat{T}_W\|$ and $\|\widehat{T}_Z\|$ seem to stabilize as k increases and both quantities are close to $\|A_E\|$.

Experiment 7.7. In this experiment, we investigate the use of inexact Cholesky factors $\widehat{B}$ and $\widehat{G}$. In particular, we assume that the exact noise covariance matrices $B^\top B$ and $G^\top G$ are unknown, and we compare the performance of the proposed RSVD-CUR decomposition with that of a CUR factorization in reconstructing A from A_E. To generate the inexact Cholesky factors $\widehat{B}$ and $\widehat{G}$, we multiply the off-diagonal elements of the exact Cholesky factors B and G by uniform random numbers from the interval $[0.9, 1.1]$. We maintain the experimental setup described in Experiment 7.6 using noise levels $\varepsilon = 0.1, 0.2$, except that here we

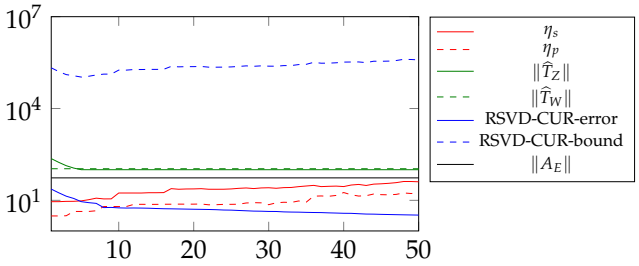
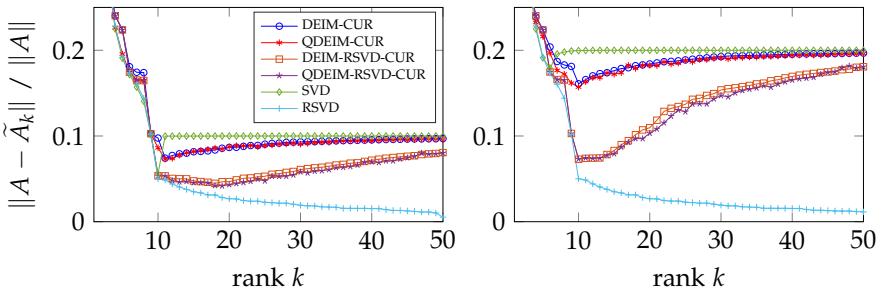


Figure 7.2: Various quantities from Theorem 7.5 using the DEIM index selection scheme: error constants $\eta_p = \|(Q_W^\top P)^{-1}\|$ (red dashed) and $\eta_s = \|(S_A^\top Q_Z)^{-1}\|$ (red solid); multiplicative factors $\|\widehat{T}_Z\|$ (green solid) and $\|\widehat{T}_W\|$ (green dashed); an RSVD-CUR true error $\|A_E - (CMR)_{\mathsf{rsvd-cur}}\|$ of approximating A_E in Experiment 7.6 (blue solid) and its upper bound (blue dashed).

compute the RSVD of $(A_E, \widehat{B}, \widehat{G})$ to obtain the input matrices for the RSVD-CUR decomposition. Our results, presented in Figures 7.3a and 7.3b, suggest that the RSVD and RSVD-CUR factorization still provide good approximation results even when using inexact Cholesky factors. This finding may indicate that we may not necessarily need the exact noise covariance.



(a) $\varepsilon = 0.1$. (b) $\varepsilon = 0.2$.

Figure 7.3: The approximation quality of RSVD-CUR factorizations using inexact Cholesky factors of the noise covariance matrices compared with CUR decompositions in recovering a sparse, nonnegative matrix A perturbed with nonwhite noise. The average relative errors $\|A - \widetilde{A}_k\| \;/\; \|A\|$ (on the vertical axis) as a function of rank k (on the horizontal axis) for $\varepsilon = 0.1$, 0.2, respectively.

Experiment 7.8. This experiment demonstrates the effectiveness of an RSVD-ID (as defined in (7.7)) in discovering the underlying class structure that two views of a data set share. We demonstrate that using an RSVD-ID as a feature selection technique in multi-view classification problems can improve classification accuracy. Typically, multi-view classification problems aim to improve classification accuracy by integrating information from different views into a unified representation. Two traditional approaches for dimension reduction in such problems are concatenation and separation. The concatenation strategy merges different views into a new feature space and applies traditional feature selection algorithms such as an interpolative decomposition on the merged set, while the separation strategy performs feature selection separately on each view. The concatenation approach may overlook the unique statistical properties of each feature set, while the separation strategy may miss important relationships between views. Given that multiple views of data can offer complementary information, it is reasonable to develop a feature selection algorithm that leverages all views and exploits their relationships.

Consider the two views/feature sets as matrices B and G. We are primarily interested in the key columns of B and G rather than their explicit CUR factorization. An RSVD-ID may serve as an unsupervised feature selection method for two-view data sets, leveraging the correlation between the views. While we are interested in a subset of the columns of B and G, the problem involves three matrices and requires the use of RSVD-ID. Specifically, we use $A := B^\top G$ as the cross-correlation between the two views, B as View1, and G as View2. We compare the classification test error rate of the QDEIM-type RSVD-ID scheme with that of the QDEIM-type ID algorithm. (Recall the relationship between $\mathrm{cca}(B, G)$ and the RSVD of $(B^\top G, B^\top, G)$ as discussed in Section 7.3.) We generate several reduced feature sets and compare their classification performance as follows:

(i) Two reduced feature sets are created by applying the QDEIM-ID procedure on each view separately, i.e., the separation strategy. We label the reduced feature sets as ID-View1 and ID-View2.

(ii) Another two reduced feature sets are created by performing QDEIM-type RSVD-ID on the two views, i.e., RSVD-ID of $(A, B^\top, G)$. We label them RSVD-ID-View1 and RSVD-ID-View2, which are the RSVD-ID selected features of views 1 and 2, respectively.

(iii) The feature set labeled Fused-ID is a concatenation of the two reduced feature sets from (i), i.e., [ID-View1, ID-View2].

(iv) Concat-ID is another feature set formed by applying the QDEIM-ID scheme on the column concatenation of both views. Note that here, the concatenation of the views is done before the dimension reduction is performed, i.e., the concatenation strategy.

(v) Finally, we concatenate the two reduced feature sets from (ii) to get Fused-RSVD-ID, i.e., [RSVD-ID-View1, RSVD-ID-View2].

To ensure a fair comparison, we present the results of the single views (i) and (ii) in one table and the feature fusion results (iii), (iv), and (v) in a separate table. This allows us to investigate the impact of incorporating complementary information from all views on the classification performance of each feature set and to determine which method yields the best results.

To demonstrate the effectiveness of our approach, we use the handwritten digits data set from the UCI repository, which contains features of handwritten numerals (0–9) extracted from Dutch utility maps [36]. The data set consists of 2000 digits, with 200 instances for each of the ten classes. We extract three types of feature sets: Fourier descriptors, Karhunen–Loève coefficients, and image vectors. The Fourier set contains 76 two-dimensional shape descriptors, the Karhunen–Loève feature set consists of 64 features, and the 'pixel' feature set was obtained by dividing the image of 30×48 pixels into 240 tiles of 2×3 pixels. We combine these three feature sets to form three experiments of a two-view classification. In the first experiment, we use the Fourier coefficients of the character shapes (fou) and the Karhunen–Loève coefficients (kar) as view-1 and view-2, respectively. The second experiment uses the pixel averages in 2×3 windows (pix) as the first view and the Fourier coefficients of the character shapes (fou) as the second view. Finally, the third experiment takes the pixel averages in 2×3 windows (pix) as the first view and the Karhunen–Loève coefficients (kar) as the second view. Table 7.2 summarizes the basic traits of the various data sets. We normalize all the data sets to have a zero center and a standard deviation of one.

Table 7.2: Summary characteristics of multi-view data sets used in the experiments.

Data set	**Samples**	**View 1** (B)	**View 2** (G)
Digits (fou vs. kar)	2000	76	64
Digits (pix vs. fou)	2000	240	76
Digits (pix vs. kar)	2000	240	64

In each experiment, we randomly split the normalized data into 75% training and 25% testing data. For the randomization of the experiments, we perform

20 cases using different random seeds. Tables 7.3 and 7.4 reports the average classification test error rate of the default k-nearest neighbor (k-NN) classifier in MATLAB for varying reduced dimensions.

Table 7.3: The average classification test error rate over 20 different random train-test splits of the 'pixel', Fourier descriptors and Karhunen–Loève feature sets using QDEIM-ID and QDEIM-RSVD-ID as a dimension reduction method for a k-nearest neighbor classifier in Experiment 7.8.

Data/Method	Rank-k	ID-View 1	RSVD-ID-View1	ID-View2	RSVD-ID-View2
B=pix vs. G=fou	20	0.15	**0.10**	0.33	**0.19**
	30	0.10	**0.07**	0.28	**0.19**
B=fou vs. G=kar	20	0.33	**0.18**	0.17	**0.07**
	30	0.28	**0.19**	0.13	**0.06**
B=pix vs. G=kar	20	0.15	**0.08**	0.17	**0.04**
	30	0.10	**0.06**	0.14	**0.04**

Table 7.4: The average classification test error rate over 20 different random train-test splits of fused feature sets using QDEIM-ID and QDEIM-RSVD-ID as a dimension reduction method for a k-nearest neighbor classifier in Experiment 7.8.

Data/Method	Rank-k	Fused-ID	Concat-ID	Fused-RSVD-ID
B=pix vs. G=fou	20	0.11	0.13	**0.06**
	30	0.09	0.12	**0.04**
B=fou vs. G=kar	20	0.14	0.15	**0.03**
	30	0.10	0.13	**0.02**
B=pix vs. G=kar	20	0.10	**0.05**	0.06
	30	0.07	**0.04**	**0.04**

From the results, we observe that a QDEIM-RSVD-ID method consistently performs better than a QDEIM-ID scheme. In particular, from the classification results using single views, the QDEIM-RSVD-ID significantly improves the worse QDEIM-ID single view results, as seen in columns 3 and 4 of Table 7.3. We notice that using information from multiple views indeed reduces the classification test error rate. Furthermore, in Table 7.4, we observe that feature fusion from a QDEIM-type RSVD-ID approximation usually gives the least test error rate

compared with the two other approaches involving the QDEIM-ID scheme, i.e., method (iii) and (iv).

Experiment 7.9. In certain applications, selecting the "best" feature subset is not enough; cost considerations associated with those features also need to be taken into account. For instance, in medical diagnosis, medical tests incur a cost and risk, whereas symptoms observed by patients or medical practitioners are usually cost-free. Thus, reducing monetary costs and sparing patients from unpleasant or dangerous clinical tests (which can be quantified as costly variables) is essential. In image analysis, feature acquisition processes' time and space complexities generally constitute the computational cost of features. As such, reducing this cost by selecting only relevant and ideally "inexpensive" variables is a typical modeler's goal.

In this experiment, we evaluate the efficacy of the CUR, the GCUR, and the RSVD-CUR methods in selecting relevant features that enhance prediction accuracy while keeping feature acquisition costs low in the presence of correlated noise. We demonstrate how RSVD-CUR factorization can be used in this context using the Thyroid disease data set [81] from the UCI repository. The problem is to determine whether a patient referred to the clinic has hypothyroidism. The data set comprises three classes: normal (not hypothyroid), hyperfunction, and subnormal functioning. Given that 92% of the patients are not hyperthyroid, a good classifier must have an accuracy significantly higher than 92%.

This 21-dimensional data set has a separate training and testing set. The training set consists of 3772 samples and the testing set consists of 3428 instances. The data set comes with an intrinsic cost associated with 20 input features, which we used to construct a diagonal matrix $G \in \mathbb{R}^{20\times 20}$. We, therefore, dropped the feature that does not have an associated cost. Table 7.5 show the Thyroids data set attributes. For our purpose, we add a small correlated noise perturbation to the normalized training data set, e.g., $\varepsilon = \|A_E - A\| / \|A\| = 0.001$, where matrix A represents the original 20-dimensional training data set. (Note that we only perturb the training data set.) We assume that the lower triangular matrix $B \in \mathbb{R}^{3772\times 3772}$ is the Cholesky factor of a symmetric positive definite matrix with first-order autoregressive structure (7.15) (with diagonal entries 1 and the off-diagonal entries related by a multiplicative factor of 0.99). So the perturbation matrix $E = B\widetilde{E}$, where $\widetilde{E} \in \mathbb{R}^{3772\times 20}$ is a random Gaussian matrix.

We evaluate the performance of the three algorithms based on the cost of features selected and their classification test error rates. The DEIM index selection procedure is used in this experiment. The standard CUR decomposition selects ten columns of A_E without considering the noise filter B and the cost matrix

Table 7.5: Thyroid disease data features and their associated costs.

Feature	Cost	Feature	Cost
age	1.00	query_hyperthyroid	1.00
sex	1.00	lithium	1.00
on_thyroxine	1.00	goitre	1.00
query_on_thyroxine	1.00	tumor	1.00
on_antithyroid_medication	1.00	hypopituitary	1.00
sick	1.00	psych	1.00
pregnant	1.00	TSH	22.78
thyroid_surgery	1.00	T3	11.41
I131_treatment	1.00	TT4	14.51
query_hypothyroid	1.00	T4U	11.41

G. The GCUR method selects a subset of ten columns of A_E relative to G but does not consider the noise. The RSVD-CUR method selects ten columns of A_E by incorporating all available prior information from (A_E, B, G). To ensure the validity of the results, we perform ten cases using different random seeds. Table 7.6 presents the average total cost of the selected features and the average classification test error rate, where we use the default k-nearest neighbor (k-NN) classifier in MATLAB. The results shown in Table 7.6 indicate that methods

Table 7.6: Average classification test error rate and total cost of selected variables using data set in Experiment 7.9. The k-NN model is trained using the perturbed training data set.

Method/Criteria	Error rate		Cost
	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-3}$	
RSVD-CUR	0.07	0.07	10
GCUR	0.09	0.26	10
CUR	0.11	0.11	23.51
All features	0.07	0.07	76.11

incorporating cost information (i.e., GCUR and RSVD-CUR) lead to lower total costs of features. The features selected by the RSVD-CUR result in the lowest classification error rate, possibly because the RSVD-CUR considers the noise structure during feature selection. When $\varepsilon = 10^{-4}$, the average classification error rate of the RSVD-CUR is approximately 28% and 57% lower than that of the GCUR and the CUR, respectively. Furthermore, the error rates of the RSVD-CUR

and the CUR are less sensitive to perturbation levels compared to the GCUR (the error rate of the GCUR-selected features increases drastically from 0.09 to 0.26 as the noise level increases from 0.0001 to 0.001). Notably, the error rate of using the full 20-dimensional feature set is similar to that of the RSVD-CUR's 10-dimensional feature set with lower cost. When $\varepsilon = 10^{-3}$, both the RSVD-CUR and the GCUR select the "pregnant" feature (with lower cost) as the most important, while the CUR selects "TT4" (with higher cost) as the most important, resulting in higher feature cost compared to the GCUR and the RSVD-CUR.

General Gauss-Markov model with constraints. We briefly describe another possible application of the RSVD-CUR factorization here.

The RSVD-CUR decomposition may be used as a subset selection procedure in the general Gauss-Markov linear models with constraints problem (7.1). "This problem formulation admits ill-conditioned or rank-deficient $B \in \mathbb{R}^{m\times\ell}$ and $G \in \mathbb{R}^{d\times n}$ (usually with $d \leq n$) matrices" [27]. The matrix B may be considered a noise filter and G may represent prior information about the unknown components of $\mathbf{x}$ or may reflect the fact that certain components of $\mathbf{x}$ are more important or less costly than others [27]. Minimizing $\|\mathbf{y}\|^2 + \|\mathbf{f}\|^2$ reflects that the goal is to explain as much in terms of the columns of A (i.e., minimize $\|\mathbf{y}\|^2$), taking into consideration the prior information on the structure of the noise as well as the preference of the modeler to use more of one predictor than others in explaining the phenomenon [27]. It is easy to see that the problem has a solution if the linear system

$$\begin{bmatrix} A & B \\ G & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{f} \end{bmatrix}$$

is consistent.

In many applications, it is desirable to reduce the number of variables that are to be considered or measured in the future. As a result, it would be appropriate to use a variable subset selection method that incorporates all available prior information (i.e., B and G). Since this problem (7.1) involves three matrices, the RSVD-CUR is a suitable procedure for variable subset selection. One may argue that a CUR decomposition of $B^{-1}AG^{-1}$ may be used if B and G are nonsingular. However, since the above problem admits an ill-conditioned or rank-deficient B, such a formulation may not always be valid. Suppose we want to select k columns of A and the corresponding columns of G, the above linear system reduces to

$$\begin{bmatrix} A_k & B \\ G_k & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{b}} \\ \widehat{\mathbf{f}} \end{bmatrix},$$

where $A_k = A(:, \mathbf{p})$ and $G_k = G(:, \mathbf{p})$. The index vector $\mathbf{p}$ is obtained by applying Algorithm 19 to (A, B, G).

Example 7.10. The following examples are adapted from [25]. We give results for three problems with $m = \ell = 1000$ and $n = d = 100$. We denote by randn a matrix or vector from the standard normal distribution and by randsvd(κ) a random matrix with spectral norm condition number κ and geometrically distributed singular values; generated by the routine randsvd in Matlab's gallery. We consider problems where either one of the matrices is ill conditioned. For all problems, we take $\mathbf{b} =$ randn and $\mathbf{f} =$ randn. Table 7.7 reports the average relative errors of 100 test cases for each problem using the original and reduced system. We compute the errors as $\|(\mathbf{b}, \mathbf{f}) - (\widehat{\mathbf{b}}, \widehat{\mathbf{f}})\|$. In this RSVD-CUR type approach for a Gauss–Markov application, typical behavior of slowly decaying decomposition error is observed.

Table 7.7: Average relative errors of the original and reduced system for varying k values for the various problems of Example 7.10.

Problem \ k			10	20	30
A	B	G			
randsvd(10)	randsvd(10^4)	randn	0.29	0.27	0.25
randsvd(10^6)	randsvd(10)	randn	0.29	0.27	0.25
randsvd(10^4)	randsvd(10^4)	randsvd(10)	0.28	0.27	0.25
randn	randn	randn	0.29	0.27	0.25

7.6 FINAL CONSIDERATIONS

We have proposed a new low-rank matrix decomposition method, referred to as RSVD-CUR factorization, which extends the CUR decomposition to matrix triplets (A, B, G). The construction of the C and R factors is performed using the DEIM or QDEIM index selection procedure, although other selection methods such as a maximum volume algorithm [50] may be used instead. We have discussed the relationship between this DEIM type RSVD-CUR and a DEIM type CUR or GCUR for nonsingular B and G. When $B = G = I$, the RSVD-CUR decomposition of A coincides with a CUR decomposition of A. Additionally, when $B = I$, a DEIM type RSVD-CUR of (A, B, G) corresponds to a DEIM type GCUR of (A, G), and similarly for the transpose of (A, B) when $G = I$. The RSVD-CUR factorization can be applied to feature fusion and feature subset selection in multi-view classification and multi-label classification problems. In data perturbation

problems, the RSVD-CUR approximation can provide more accurate results than a CUR factorization when reconstructing a low-rank matrix from a correlated noise-perturbed data matrix. The RSVD-CUR factorization can also be used as a subset selection technique in generalized Gauss-Markov problems with constraints. The experiments in Section 7.5 demonstrate the effectiveness of the RSVD-CUR factorization in these applications.

CONCLUSIONS & OUTLOOK

A CUR factorization is used as an alternative to the well-known singular value decomposition. There are several advantages to using a CUR factorization over the SVD. Firstly, it uses a subset of the original matrix's rows and columns, hence, it can preserve important structures in the original matrix, while eliminating noise or irrelevant data. Additionally, because the C and R matrices provide explicit subsets of the original matrix's columns and rows, the resulting factorization can provide a more interpretable representation of the original data. This can be especially useful in applications such as collaborative filtering, where the factors correspond to user preferences and item features.

Unlike the SVD, which is a unique decomposition for any given matrix, a CUR factorization, in general, is non-unique. This is because the C and R factors in the CUR approximation can be chosen in many different ways, as long as they contain appropriate subsets of the original matrix's columns and rows. Additionally, the middle matrix can be scaled or computed in various ways without affecting the validity of the factorization. An essential aspect of the CUR factorization is the methodology used for selecting a subset of columns and rows from the original matrix. In this work, we have focused on developing new tools for CUR factorization and improving existing techniques.

One provably appropriate technique for index selection in constructing a CUR factorization is the discrete empirical interpolation method. In Chapter 3 we presented a new extension of the DEIM index selection algorithm called L-DEIM, which can identify additional indices for constructing a rank-$\widehat{k}$ CUR decomposition using a lower-rank SVD approximation. The method combines the strength of deterministic leverage score sampling and the DEIM scheme. The proposed procedure is particularly useful in scenarios such as big data problems where computing a full rank-$\widehat{k}$ SVD is expensive. The L-DEIM scheme can be viewed not only as an extension of DEIM but also as an alternative index selection method for a CUR factorization. Although the L-DEIM algorithm can be more computationally efficient than the original DEIM scheme, experiments suggest that the approximation accuracy of both methods is comparable when the target rank $\widehat{k}$ is at most twice the available k singular vectors.

Chapter 4 introduces several block versions of the discrete empirical interpolation method for computing CUR decompositions. These variants leverage the benefits of the classic DEIM procedure, a column-pivoted QR decomposition, and

the idea of maximum determinant or volume of submatrices. We also present a version of block DEIM that allows for flexible block size selection.

The block DEIM based on RRQR works as follows: at each iteration step, we compute a QR factorization with column pivoting on the transpose of a block of singular vectors to identify the indices corresponding to the first b columns. Next, we update the next block of vectors using the interpolatory projection technique from the DEIM procedure. We repeat these two steps until all indices are selected. The block DEIM based on MaxVol follows a similar procedure, except instead of using a column-pivoted QR decomposition, we use the MaxVol method.

The block DEIM variants provide effective alternatives for constructing a CUR decomposition, exhibiting comparable accuracy and potentially improved computational efficiency when compared to some existing methods.

In Chapter 5 new techniques for selecting subsets of columns and rows using an iterative subselection strategy are introduced. Additionally, we discuss how iterative procedures for computing a few singular vectors of large data matrices can be used with these iterative subselection techniques. The new approaches adaptively employ the DEIM scheme and are shown through empirical analysis to produce better approximation results than the traditional DEIM procedure of one-round sampling of all columns and rows. The proposed methods can be useful techniques for improving the accuracy of a CUR decomposition. However, they can also introduce additional complexities that need to be addressed carefully. Whether to use iterative subselection schemes or not depends on the specific problem or application, as well as the trade-offs between accuracy, complexity, and computational resources.

Chapter 6 presents a new low-rank matrix approximation method called a generalized CUR factorization, which is an extension of the DEIM-CUR decomposition for matrix pairs. It decomposes two data sets together and may be used in selecting discriminative features of one data set relative to another. The GCUR algorithm may also be useful in applications where a data matrix suffers from nonwhite (colored) noise and the goal is to recover the unperturbed matrix given some information about the noise. We compare the new method with the standard DEIM-CUR decomposition in numerical experiments and show that it can perform better in subgroup discovery and subset selection in a classification problem. The GCUR algorithm provides more accurate approximation results compared to the DEIM-CUR algorithm when recovering an original matrix with low rank from data with colored noise given some known information about the noise.

In Chapter 7 we introduced the RSVD-CUR factorization, a new low-rank matrix decomposition method that extends the CUR decomposition to matrix triplets

(A, B, G). We explored the relationship between the DEIM type RSVD-CUR and a DEIM type CUR or GCUR for nonsingular B and G. The RSVD-CUR factorization can be employed for feature fusion and feature subset selection in multi-view classification and multi-label classification problems. It can also provide more accurate results than a CUR factorization when reconstructing a low-rank matrix from data with structured colored noise perturbation problems. In addition, the RSVD-CUR factorization can be used as a subset selection technique in generalized Gauss-Markov problems with constraints. Our experiments confirm the efficacy of the RSVD-CUR factorization in these applications.

OUTLOOK

The findings presented in this dissertation rely on specific assumptions and choices made to effectively address the research questions. While the methods discussed have been found to have limitations, we have also identified their strengths and practical applications. Moving forward, we suggest potential areas for further research that stem from the previous chapters.

In Chapter 3, Algorithm 7 extracts at least $\widehat{k} \geq k$ column of A given rank-k singular vectors. However, an upper bound on the number of output columns $\widehat{k}$ is not immediate. It would be an idea to investigate such an upper bound. Could the stopping criteria depend on the decay of the "leverage scores"?

In Chapter 5, we introduced a basic iterative subselection strategy that requires the user to specify a parameter δ to determine the number of selected columns or rows in each iteration. However, there are still several open questions regarding this approach. One of the primary questions is how to efficiently determine the optimal value of δ. Additionally, it remains unclear whether varying δ in each iteration can lead to improved theoretical bounds or better empirical performance. These open questions present exciting opportunities for future research in the field.

The proposed methods in Chapters 6 and 7 are based on the generalized and restricted SVD, both of which are computationally intensive tasks. Future research could focus on developing more efficient algorithms for the GCUR and RSVD-CUR, which could make the methods even more practical for large-scale problems. For the standard CUR, randomized algorithms and methods that do not require the SVD have been proposed. It could be interesting to explore such methods and pass-efficient algorithms for the GCUR and the RSVD-CUR factorizations.

We presented possible applications for the GCUR and the RSVD-CUR decompositions. We expect that the promise of the GCUR and the RSVD-CUR methods may be more general than the applications considered in this thesis. Future research could focus on exploring various applications and demonstrating the effectiveness of the proposed method in real-world scenarios. Additionally, the standard CUR decomposition has been shown to be useful in several machine learning methods. It could be interesting to explore how the two proposed factorizations could be integrated with other machine learning techniques.

The CUR decomposition has been extended to tensors [73]. The extension of the CUR decomposition to tensors involves finding a set of factor matrices that can be multiplied together to reconstruct the original tensor. This is a challenging problem due to the increased complexity of tensors, which have multiple modes or dimensions. However, several techniques have been proposed to address this issue [13, 19, 73]. These methods use the same principles as the original CUR decomposition but adapt them to the tensor setting.

Recently, He et al. [60] extended the generalized singular value decomposition from matrix pairs to tensor pairs. This extension has opened up new possibilities for tensor decomposition, as it allows for the simultaneous decomposition of two tensors. Given the promising results of the GCUR for matrix pairs, it would be interesting to explore whether the GCUR can be adapted to tensor pairs using the GSVD for tensors. This could lead to improved low-rank tensor approximation methods, which could be used in a variety of applications, such as image and video processing, scientific data analysis, and machine learning. Further research in this area could help advance the field of tensor decomposition and lead to breakthroughs in high-dimensional data analysis.

BIBLIOGRAPHY

[1] A. Abid, M. J. Zhang, V. K. Bagaria, and J. Zou. "Exploring patterns enriched in a dataset with contrastive principal component analysis." *Nat. Commun.* 09 (2018), pp. 1–7.

[2] R. Arora and K. Livescu. "Multi-view CCA-based acoustic features for phonetic recognition across speakers and domains." In: *Proc. Int. Conf. Acoust. Speech Signal Proces.* 2013, pp. 7135–7139.

[3] J. Baglama and L. Reichel. "Augmented implicitly restarted Lanczos bidiagonalization methods." *SIAM J. Sci. Comput.* 27.1 (2005), pp. 19–42.

[4] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. "A comparison of decision tree ensemble creation techniques." *IEEE Trans. Pattern Anal. Mach. Intell.* 29.1 (2006), pp. 173–180.

[5] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. "An empirical interpolation method: Application to efficient reduced-basis discretization of partial differential equations." *Comptes Rendus Math.* 339 (2004), pp. 667–672.

[6] A. Beck. "The matrix-restricted total least-squares problem." *Signal Process.* 87.10 (2007), pp. 2303–2312.

[7] M. W. Berry, S. A. Pulatova, and G. W. Stewart. "Algorithm 844: Computing sparse reduced-rank approximations to sparse matrices." *ACM Trans. Math. Softw.* 31.2 (2005), pp. 252–269.

[8] J. Bien, Y. Xu, and M. W. Mahoney. "CUR from a sparse optimization viewpoint." *Adv. Neural Inf. Process Syst.* 23 (2010), pp. 217–225.

[9] P. Boileau, N. S. Hejazi, and S. Dudoit. "Exploring high-dimensional biological data with sparse contrastive principal component analysis." *Bioinformatics* 36.11 (2020), pp. 3422–3430.

[10] C. Boutsidis and D. Woodruff. "Optimal CUR matrix decompositions." *SIAM J. Comput.* 46.2 (2017), pp. 543–589.

[11] J. P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. "Metagenes and molecular pattern discovery using matrix factorization." *Proc. Natl. Acad. Sci.* 101.12 (2004), pp. 4164–4169.

[12] D. Cai, X. He, and J. Han. "Document clustering using locality preserving indexing." *IEEE Trans. Knowl. Data Eng.* 17.12 (2005), pp. 1624–1637.

[13] H. Cai, K. Hamm, L. Huang, and D. Needell. "Mode-wise tensor decompositions: Multi-dimensional generalizations of CUR decompositions." *J. Mach. Learn. Res.* 22.1 (2021), pp. 8321–8356.

[14] T. F. Chan. "Rank revealing QR factorizations." *Linear Algebra Appl.* 88 (1987), pp. 67–82.

[15] S. Chandrasekaran and I. C. F. Ipsen. "On rank-revealing factorisations." *SIAM J. Matrix Anal. Appl.* 15.2 (1994), pp. 592–622.

[16] S. Chaturantabut and D. C. Sorensen. "Nonlinear model reduction via discrete empirical interpolation." *SIAM J. Sci. Comput.* 32 (2010), pp. 2737–2764.

[17] C. Chen, M. Gu, Z. Zhang, W. Zhang, and Y. Yu. "Efficient spectrum-revealing CUR matrix decomposition." In: *Proc. Artif. Intell. Statistics Conf.* PMLR. 2020, pp. 766–775.

[18] J. Chen, G. Wang, and G. B. Giannakis. "Nonlinear dimensionality reduction for discriminative analytics of multiple datasets." *IEEE Trans. Signal Process.* 67.3 (2019), pp. 740–752.

[19] J. Chen, Y. Wei, and Y. Xu. "Tensor CUR decomposition under T-product and its perturbation." *Numer. Funct. Anal. Optim.* 43.6 (2022), pp. 698–722.

[20] X. Chen, L. Han, and J. Carbonell. "Structured sparse canonical correlation analysis." In: *Proc. Artif. Intell. Statistics Conf.* 2012, pp. 199–207.

[21] H. Cheng, Z. Gimbutas, P. G. Martinsson, and V. Rokhlin. "On the compression of low rank matrices." *SIAM J. Sci. Comput.* 26.4 (2005), pp. 1389–1404.

[22] D. Chu, L. De Lathauwer, and B. De Moor. "On the computation of the restricted singular value decomposition via the cosine-sine decomposition." *SIAM J. Matrix Anal. Appl.* 22.2 (2000), pp. 580–601.

[23] A. Civril and M. Magdon-Ismail. "On selecting a maximum volume sub-matrix of a matrix and related problems." *Theor. Comput. Sci.* 410.47-49 (2009), pp. 4801–4811.

[24] A. Civril and M. Magdon-Ismail. "Exponential inapproximability of selecting a maximum volume sub-matrix." *Algorithmica* 65 (2013), pp. 159–176.

[25] A. J. Cox and N. J. Higham. "Row-wise backward stable elimination methods for the equality constrained least squares problem." *SIAM J. Matrix Anal. Appl.* 21.1 (1999), pp. 313–326.

[26] T. A Davis and Y. Hu. "The University of Florida sparse matrix collection." *ACM TOMS* 38.1 (2011), pp. 1–25.

[27] B. L. De Moor and G. H. Golub. "The restricted singular value decomposition: properties and applications." *SIAM J. Matrix Anal. Appl.* 12.3 (1991), pp. 401–425.

[28] A. Deshpande, L. Rademacher, S. S. Vempala, and G. Wang. "Matrix approximation and projective clustering via volume sampling." *Theory Comput.* 2.1 (2006), pp. 225–247.

[29] A. Deshpande and S. Vempala. "Adaptive sampling and fast low-rank matrix approximation." In: *Proceedings of the 10th RANDOM APPROX*. 2006, pp. 292–303.

[30] T. G. Dietterich and G. Bakiri. "Solving multiclass learning problems via error-correcting output codes." *J. Artif. Intell. Res.* 2 (1994), pp. 263–286.

[31] P. Drineas and R. Kannan. "Pass efficient algorithms for approximating large matrices." In: *SODA*. Vol. 3. 2003, pp. 223–232.

[32] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. *Polynomial time algorithm for column-row based relative-error low-rank matrix approximation*. Tech. rep. 2006-04, DIMACS, 2006.

[33] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. "Relative-error CUR matrix decompositions." *SIAM J. Matrix Anal. Appl.* 30 (2008), pp. 844–881.

[34] Z. Drmac and S. Gugercin. "A new selection operator for the discrete empirical interpolation method—Improved a priori error bound and extensions." *SIAM J. Sci. Comput.* 38.2 (2016), pp. A631–A648.

[35] Z. Drmac and A. K. Saibaba. "The discrete empirical interpolation method: Canonical structure and formulation in weighted inner product spaces." *SIAM J. Matrix Anal. Appl.* 39.3 (2018), pp. 1152–1180.

[36] R. Duin. *Multiple Features*. UCI Machine Learning Repository. DOI: 10.24432/C5HC70.

[37] A. Frieze, R. Kannan, and S. Vempala. "Fast Monte-Carlo algorithms for finding low-rank approximations." *J. ACM* 51.6 (2004), pp. 1025–1041.

[38] O. Friman, M. Borga, P. Lundberg, and H. Knutsson. "Adaptive analysis of fMRI data." *NeuroImage* 19.3 (2003), pp. 837–845.

[39] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, 2013.

[40] E. Gabrilovich and S. Markovitch. "Text categorization with many redundant features: Using aggressive feature selection to make SVMs competitive with C4.5." In: *The 21st International Conference on Machine Learning*. 2004, p. 41.

[41] Y. Gao and G. Church. "Improving molecular cancer class discovery through sparse non-negative matrix factorization." *Bioinformatics* 21.21 (2005), pp. 3970–3975.

[42] P. Y. Gidisu and M. E. Hochstenbach. "A generalized CUR decomposition for matrix pairs." *SIAM J. Math. Data Science* 4.1 (2022). **Winner of a 2023 SIAM Student Paper Prize**, pp. 386–409.

[43] P. Y. Gidisu and M. E. Hochstenbach. "A hybrid DEIM and leverage scores based method for CUR index selection." In: *Progress in Industrial Mathematics at ECMI 2021*. Springer International Publishing, 2022, pp. 147–153.

[44] P. Y. Gidisu and M. E. Hochstenbach. *A DEIM-CUR factorization with iterative SVDs*. In Preparation. 2023.

[45] P. Y. Gidisu and M. E. Hochstenbach. *A restricted SVD type CUR decomposition for matrix triplets*. To appear in SIAM J. Sci. Comput. arXiv: 2204.02113 [math.NA].

[46] P. Y. Gidisu and M. E. Hochstenbach. *Block discrete empirical interpolation methods*. arXiv: 2208.02213 [math.NA].

[47] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. "Eigentaste: A constant time collaborative filtering algorithm." *Inform. Retrieval* 4.2 (2001), pp. 133–151.

[48] G. H. Golub and C. F. Van Loan. *Matrix Computations*. 4th. Baltimore: Johns Hopkins University Press, 2012.

[49] G. H. Golub and H. Zha. "The canonical correlations of matrix pairs and their numerical computation." In: *Linear Algebra for Signal Processing*. Springer, 1995, pp. 27–49.

[50] S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. "How to find a good submatrix." In: *Matrix Methods: Theory, Algorithms, And Applications*. World Scientific, Singapore, 2010, pp. 247–256.

[51] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. "A theory of pseudoskeleton approximations." *Linear Algebra Appl.* 261.1-3 (1997), pp. 1–21.

[52] S. A. Goreinov, N. L. Zamarashkin, and E. E. Tyrtyshnikov. "Pseudoskeleton approximations by matrices of maximal volume." *Math. Notes* 62.4 (1997), pp. 515–519.

[53] M. Gu and S. C. Eisenstat. "Efficient algorithms for computing a strong rank-revealing QR factorization." *SIAM J. Sci. Comput.* 17.4 (1996), pp. 848–869.

[54] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. *Gisette*. UCI Machine Learning Repository. 2008. DOI: 10.24432/C5HP5B.

[55] N. Halko, P. G. Martinsson, and J. A. Tropp. "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions." *SIAM Review* 53.2 (2011), pp. 217–288.

[56] P. C. Hansen. "Regularization, GSVD and Truncated GSVD." *BIT* 29.3 (1989), pp. 491–504.

[57] P. C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, Philadelphia, 1998.

[58] P. C. Hansen and S. H. Jensen. "Subspace-based noise reduction for speech signals via diagonal and triangular matrix decompositions: Survey and analysis." *EURASIP J. Adv. Signal Process.* 2007 (2007), pp. 1–24.

[59] W. K. Härdle and L. Simar. *Applied Multivariate Statistical Analysis*. 4th. Berlin, Heidelberg: Springer, 2015.

[60] Z. H. He, M. K. Ng, and C. Zeng. "Generalized singular value decompositions for tensors and their Applications." *Numer. Math. Theor. Meth. Appl.* 14.3 (2021), pp. 692–713.

[61] E. P. Hendryx, B. M. Rivière, and C. G. Rusin. "An extended DEIM algorithm for subset selection and class identification." *Mach. Learn.* 110.4 (2021), pp. 621–650.

[62] M. E. Hochstenbach. "A Jacobi–Davidson type method for the generalized singular value problem." *Linear Algebra Appl.* 431.3-4 (2009), pp. 471–487.

[63] R. A. Horn and C. R. Johnson. *Matrix Analysis*. 2nd. Cambridge University Press, 2012.

[64] A. S. Householder. *The Theory of Matrices in Numerical Analysis*. Dover Publications, New York, 2013.

[65] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning*. Vol. 112. Springer, New York, 2013.

[66] I. T. Jolliffe. "Discarding variables in a principal component analysis I: Artificial data." *Appl. Statist.* 21.2 (1972), pp. 160–173.

[67] A. Khabou, J. W. Demmel, L. Grigori, and M. Gu. "LU factorization with panel rank revealing pivoting and its communication avoiding version." *SIAM J. Matrix Anal. Appl.* 34.3 (2013), pp. 1401–1429.

[68] H. Kim and H. Park. "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis." *Bioinformatics* 23.12 (2007), pp. 1495–1502.

[69] B. Kramer and A. A. Gorodetsky. "System identification method via CUR factorization of Hankel matrix." *SIAM J. Sci. Comput.* 40.2 (2018), pp. A848–A866.

[70] A. Krizhevsky, G. Hinton, et al. *Learning multiple layers of features from tiny images*. Tech. rep. CIFAR, 2009.

[71] M. T. Landi, T. Dracheva, M. Rotunno, J. D. Figueroa, H. Liu, A. Dasgupta, F. E. Mann, J. Fukuoka, M. Hames, A. W. Bergen, et al. "Gene expression signature of cigarette smoking and its role in lung adenocarcinoma development and survival." *PloS one* 3.2 (2008), p. e1651.

[72] M. W. Mahoney and P. Drineas. "CUR matrix decompositions for improved data analysis." *Proc. Natl. Acad. Sci. USA* 106 (2009), pp. 697–702.

[73] M. W. Mahoney, M. Maggioni, and P. Drineas. "Tensor-CUR decompositions for tensor-based data." *SIAM J. Matrix Anal. Appl.* 30.3 (2008), pp. 957–987.

[74] K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton. "Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns." *IEEE Control Systems Magazine* 38.3 (2018), pp. 63–86.

[75] I. Oseledets and E. Tyrtyshnikov. "TT-cross approximation for multidimensional arrays." *Linear Algebra Appl.* 432.1 (2010), pp. 70–88.

[76] C. C. Paige and M. A. Saunders. "Towards a generalized singular value decomposition." *SIAM J. Numer. Anal.* 18 (1981), pp. 398–405.

[77] D. Papailiopoulos, A. Kyrillidis, and C. Boutsidis. "Provable deterministic leverage score sampling." In: *Proc. 20th ACM SIGKDD Conf. Knowl. Discovery Data Mining*. 2014, pp. 997–1006.

[78] H Park. "ESPRIT direction-of-arrival estimation in the presence of spatially correlated noise." *SIAM J. Matrix Anal. Appl.* 15.1 (1994), pp. 185–193.

[79] S. Paul, M. Magdon-Ismail, and P. Drineas. "Column selection via adaptive sampling." *Adv. Neural Inf. Process Syst.* 28 (2015), pp. 406–414.

[80] B. Peherstorfer, Z. Drmac, and S. Gugercin. "Stability of discrete empirical interpolation and gappy proper orthogonal decomposition with randomized and deterministic sampling points." *SIAM J. Sci. Comput.* 42.5 (2020), pp. A2837–A2864.

[81] R. Quinlan. *Thyroid Disease*. UCI Machine Learning Repository. 1987. DOI: 10.24432/C5D010.

[82] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos. "A new approach to cross-modal multimedia retrieval." In: *Proc. Int. Conf. Multimedia, ACM*. 2010, pp. 251–260.

[83] D. C. Sorensen and M. Embree. "A DEIM induced CUR factorization." *SIAM J. Sci. Comput.* 33.3 (2016), pp. A1454–A1482.

[84] G. W. Stewart. "Computing the CS decomposition of a partitioned orthonormal matrix." *Numer. Math.* 40.3 (1982), pp. 297–306.

[85] G. W. Stewart. "Four algorithms for the efficient computation of truncated pivoted QR approximations to a sparse matrix." *Numer. Math.* 83 (1998), pp. 313–323.

[86] G. W. Stewart and J. G. Sun. *Matrix Perturbation Theory*. Academic press, 1990.

[87] G. Strang. *Linear Algebra and Learning from Data*. SIAM, Philadelphia, 2019.

[88] J. G. Sun. "Perturbation analysis for the generalized singular value problem." *SIAM J. Numer. Anal.* 20.3 (1983), pp. 611–625.

[89] D. B. Szyld. "The many proofs of an identity on the norm of oblique projections." *Numer. Algorithms* 42 (2006), pp. 309–323.

[90] M. Udell and A. Townsend. "Why are big data matrices approximately low rank?" *SIAM J. Math. Data Science* 1.1 (2019), pp. 144–160.

[91] C. F. Van Loan. "Generalizing the singular value decomposition." *SIAM J. Numer. Anal.* 13.1 (1976), pp. 76–83.

[92] C. F. Van Loan. "Computing the CS and the generalized singular value decompositions." *Numer. Math.* 46.4 (1985), pp. 479–491.

[93] S. Voronin and P. G. Martinsson. "Efficient algorithms for CUR and interpolative matrix decompositions." *Adv. Comput. Math.* 43 (2017), pp. 495–516.

[94] S. Wang and Z. Zhang. "Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling." *J. Mach. Learn. Res.* 14 (2013), pp. 2729–2769.

[95] R. Wicklin. *Simulating Data with SAS*. SAS Institute, 2013.

[96] C. Xu, D. Tao, and C. Xu. *A survey on multi-view learning*. 2013. arXiv: 1304.5634 [cs.LG].

[97] H. Zha. "The restricted singular value decomposition of matrix triplets." *SIAM J. Matrix Anal. Appl.* 12.1 (1991), pp. 172–194.

[98] H. Zha. "A numerical algorithm for computing the restricted singular value decomposition of matrix triplets." *Linear Algebra Appl.* 168 (1992), pp. 1–25.

[99] H. Zha. "Computing the generalized singular values/vectors of large sparse or structured matrix pairs." *Numer. Math.* 72.3 (1996), pp. 391–417.

[100] Y. B. Zhou. "Model reduction for nonlinear dynamical systems with parametric uncertainties." PhD thesis. Massachusetts Institute of Technology, 2012.

[101] I. N. Zwaan. *Towards a more robust algorithm for computing the restricted singular value decomposition*. 2020. arXiv: 2002.04828 [math.NA].

[102] I. N. Zwaan and M. E. Hochstenbach. *Generalized Davidson and multidirectional-type methods for the generalized singular value decomposition*. 2017. arXiv: 1705.06120 [math.NA].

INDEX

SUMMARY

In data analysis applications and machine learning, the data set is often represented by a matrix. Data matrices are usually large and in many cases, a key step in the analysis is to approximate the data using a few features and/or a few data points so that one can easily manipulate, understand, and interpret the data. The most common and optimal approach for this approximation is the truncated singular value decomposition. However, an alternative approach is to identify a good subset of columns and rows in the data matrix; a CUR factorization. A CUR decomposition is a substitute for the SVD, especially when a concrete interpretation of the singular vectors is challenging. Moreover, if the original data matrix possesses properties like nonnegativity and sparsity, a CUR decomposition can better preserve them. This dissertation introduces three new index selection methods for constructing CUR decompositions. Additionally, the thesis proposes two new procedures that extend a traditional CUR decomposition to handle matrix pairs and triplets.

Several methods have been proposed in the literature to carefully select a subset of rows and columns from the given matrix that results in a small approximation error. One such procedure is a DEIM-induced CUR approximation. This procedure applies the DEIM index selection algorithm to the right and left singular vectors of the matrix to identify the indices of the columns and rows to be selected. The DEIM procedure originated from the context of model reduction of nonlinear dynamical systems. It locally selects an index corresponding to the entry of the largest magnitude in a given vector. A notable limitation of the DEIM scheme is that the number of indices that can be selected is limited to the number of available singular vectors. Given the promising results seen in previous works, we build off of the DEIM procedure to develop a technique that allows for the selection of additional indices (L-DEIM). The L-DEIM scheme performs the original DEIM to find the first set of indices up to the number of singular vectors available and then exploits the idea of leverage scores to select the additional indices. Additionally, we develop several block variants of the DEIM scheme. The block DEIM procedures share the same principle as the standard DEIM, but they are a bit less greedy since the optimization is done over more indices instead of just one. These block variants may generally be more computationally efficient than the standard DEIM. Additionally, a block DEIM scheme may be a good solution in situations where DEIM faces a difficult choice since the local

maximizer is (nearly) nonunique. Furthermore, we propose an iterative variant of the DEIM scheme. This procedure aims to improve the approximation quality of the DEIM scheme by adaptively invoking it; in the sense that we modify the original matrix after each iteration by removing the information that has been captured by the previously selected columns and rows.

Although a CUR decomposition can be used in a vast number of applications, it may still not be suitable for some applications. For example, when a matrix is perturbed with non-white noise and one is interested in recovering the unperturbed matrix. Another application is in a setting where we have two matrices and the goal is to find a low-rank representation of one relative to the other. For this reason, we formulate two generalizations of a CUR decomposition to cope with the problem of simultaneous decomposition of matrix pairs and matrix triplets.

The first generalization, a generalized CUR decomposition for matrix pairs provides low-rank approximations of *two matrices simultaneously*, in terms of some of their rows and columns; where the same columns of the two matrices are selected to give a coupling between the two decompositions. In contrast to the CUR decomposition techniques that use singular vectors, we develop a technique based on the matrices from the generalized singular value decomposition to select the representative rows and columns of the two matrices. We derive an error bound for the approximation quality of this generalized CUR decomposition in terms of the error of the GSVD. We prove that a CUR decomposition can be considered a special case of a generalized CUR decomposition. The development of this method has opened up new possibilities for generalizing CUR decomposition to more than one matrix.

Our second generalization, a restricted SVD-based CUR (RSVD-CUR) decomposition for matrix triplets provides a low-rank matrix approximation for three matrices using a subset of their columns and rows. Here, we present a method based on the restricted singular value decomposition to select the representative columns and rows. We derive an error bound on the accuracy of this RSVD-CUR decomposition in terms of the error of the restricted singular value decomposition. An RSVD-CUR factorization may be suitable for applications where one is interested in approximating one data matrix relative to two other given matrices. Two key applications that we discuss are the multi-view dimension reduction (where one has multiple information rather than a single representation of a data problem, e.g. computer vision tasks where an image can be described with color, shape, and texture features of high dimensions) and (structured) data perturbation problems with non-white noise matrix where we want to recover the unperturbed data.

PUBLICATIONS

[1] P. Y. Gidisu and M. E. Hochstenbach. "A generalized CUR decomposition for matrix pairs." *SIAM J. Math. Data Science* 4.1 (2022). **Winner of a 2023 SIAM Student Paper Prize**, pp. 386–409.

[2] P. Y. Gidisu and M. E. Hochstenbach. "A hybrid DEIM and leverage scores based method for CUR index selection." In: *Progress in Industrial Mathematics at ECMI 2021*. Springer International Publishing, 2022, pp. 147–153.

[3] P. Y. Gidisu and M. E. Hochstenbach. *A DEIM-CUR factorization with iterative SVDs*. In Preparation. 2023.

[4] P. Y. Gidisu and M. E. Hochstenbach. *A restricted SVD type CUR decomposition for matrix triplets*. To appear in SIAM J. Sci. Comput. arXiv: 2204.02113 [math.NA].

[5] P. Y. Gidisu and M. E. Hochstenbach. *Block discrete empirical interpolation methods*. arXiv: 2208.02213 [math.NA].

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to God for His unwavering love, mercy, and grace that have sustained me throughout this journey. To Him belongs all the praise, honor, and glory for eternity.

I am indebted to Prof. La Torre Davide for sharing the PhD application opportunity and encouraging me to pursue it.

My heartfelt thanks go to dr. M.E. Hochstenbach, my supervisor, for providing me with the invaluable opportunity to embark on this PhD under his guidance. I am grateful for his mentorship, support, and the thought-provoking problems he presented to me. Michiel's extensive knowledge in numerical linear algebra has greatly benefited me and his guidance on research writing and presentation.

I would like to thank the committee members for providing a detailed review of the thesis and some helpful discussions. This has helped improve the quality of the document.

I extend my gratitude to Enna, Gaby, Jonelleke, Lut, and Diane for their invaluable administrative support.

The experience within the CASA group has been made special by the warm atmosphere and camaraderie. I would like to thank my old and new office mates for the delightful conversations and memorable outings.

I also appreciate the support and meaningful interactions I have experienced with my co-promoter Prof. Alessandra Micheletti, my industry supervisor, Giuseppe Codazzi, and the BigMath group.

In May 2023, I had the opportunity to spend two weeks at Virginia Tech: Prof. Mark Embree has been a great host and I appreciated the many discussion on "CUR stuff" we had.

Last but certainly not least, my deepest thanks go to my family and friends. Their unwavering encouragement and prayers throughout this entire journey have been instrumental in my progress and successful completion of this project. I dedicate this thesis to the memory of my late mum (Magdalene Afua Dah).

CURRICULUM VITAE

Perfect Yayra Gidisu, born on April 28, 1991, hails from Kpando, Ghana. She completed her Economics and Political Science undergraduate studies with honors at the University of Ghana in 2013. Seeking further academic achievements, she relocated to Italy and earned a Master's degree in Economics and Finance with distinction, specializing in Quantitative Finance, from the University of Milan in 2018. During her Master's program, Perfect received the University of Milan Merit Scholarship on two occasions and secured a coveted position in an exchange program at Peking University HSBC Business School in China. Before commencing her Master's degree, she gained valuable professional experience working in various roles at UT Bank in Ghana for three years.

In March 2019, Perfect commenced a doctoral research project at the Eindhoven University of Technology in the Netherlands in collaboration with AcomeA Sgr, an asset management company based in Milan. The research project is supervised by dr. M.E. Hochstenbach and G. Codazzi. Her research is part of the "BigMath: Big Data Challenges for Mathematics" project, which is funded by the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie Grant Agreement No 812912.

Throughout her doctoral studies, she presented her research findings at several seminars and conferences. Additionally, she was recognized as one of the four finalists for the 2023 KWG PhD prize and her paper "A Generalized CUR Decomposition for Matrix Pairs" (this is Chapter 6), has been selected as one of the winners of the 2023 SIAM Student Paper Prize. During her pursuit of a PhD, Perfect actively engaged in academic enrichment by attending two notable summer schools: "High-Performance Data Analytics" in Aussois in June 2019, and "School on The Mathematics of Machine Learning" at Centro De Giorgi in Pisa in January 2023.