A DEIM Induced CUR Factorization

D. C. SORENSEN† and M. EMBREE‡
e-mail: sorensen@rice.edu, embree@vt.edu

July 21, 2014

Abstract

We derive a CUR matrix factorization based on the Discrete Empirical Interpolation Method (DEIM). For a given matrix $A$, such a factorization provides a low rank approximate decomposition of the form $A \approx CUR$, where $C$ and $R$ are subsets of the columns and rows of $A$, and $U$ is constructed to make $CUR$ a good approximation. Given a low-rank singular value decomposition $A \approx VSW^T$, the DEIM procedure uses $V$ and $W$ to select the columns and rows of $A$ that form $C$ and $R$. Through an error analysis applicable to a general class of CUR factorizations, we show that the accuracy tracks the optimal approximation error within a factor that depends on the conditioning of submatrices of $V$ and $W$. For large-scale problems, $V$ and $W$ can be approximated using an incremental QR algorithm that makes one pass through $A$. Numerical examples illustrate the favorable performance of the DEIM-CUR method, compared to CUR approximations based on leverage scores.

1 Introduction

This work presents a new CUR matrix factorization based upon the Discrete Empirical Interpolation Method (DEIM). A CUR factorization is a low rank approximation of a given matrix $A \in \mathbb{R}^{m \times n}$ of the form $A \approx CUR$, where $C = A(:, q) \in \mathbb{R}^{m \times k}$ is a subset of the columns of $A$ and $R = A(p, :) \in \mathbb{R}^{k \times n}$ is a subset of the rows of $A$. The $k \times k$ matrix $U$ is constructed to assure that $CUR$ is a good approximation to $A$. Assuming the best rank-$k$ singular value decomposition (SVD) $A \approx VSW^T$ is available, the algorithm uses the DEIM index selection procedure, $q = \text{DEIM}(V)$ and $p = \text{DEIM}(W)$, to determine $C$ and $R$. The resulting approximate factorization is nearly as accurate as the best rank-$k$ SVD, with

$$
\|A - CUR\| \leq (\eta_p + \eta_q) \sigma_{k+1},
$$

where $\sigma_{k+1}$ is the first neglected singular value of $A$, and $\eta_p = \|V(p, :)^{-1}\|$, $\eta_q = \|W(q, :)^{-1}\|$. Here and throughout, $\|\cdot\|$ denotes the vector 2-norm and the matrix norm it induces, and $\|\cdot\|_F$ is the Frobenius norm. We

\footnote{This work was supported in part by AFOSR grant FA9550-12-1-0155 and by NSF grant CCF-1320866.}

\footnote{Department of Computational and Applied Mathematics, MS 134, Rice University, Houston, Texas 77005-1892.}

\footnote{Department of Mathematics, Virginia Tech, Blacksburg, Virginia 24061-0123}
use MATLAB notation to index vectors and matrices, so that, e.g., $A(p, :) \in \mathbb{R}$ denotes the $k$ rows of $A$ whose indices are specified by the entries of the vector $p \in \mathbb{N}^k$, while $A(:, q)$ denotes the $k$ columns of $A$ indexed by $q \in \mathbb{N}^k$.

The CUR factorization is an important tool for handling large-scale data sets, offering two advantages over the SVD: when $A$ is sparse, so too are $C$ and $R$, unlike the matrices of singular vectors; and the columns and rows that comprise $C$ and $R$ are representative of the data (e.g., sparse, nonnegative, integer valued, etc.). The following simple example, adapted from Mahoney and Drineas [19, Fig. 1b], illustrates the latter advantage. Construct $A \in \mathbb{R}^{2 \times n}$ so that its first $n/2$ columns have the form

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and the remaining $n/2$ columns have the form

$$\frac{\sqrt{2}}{2} \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

where in both cases $x_1 \sim N(0, 1)$ and $x_2 \sim N(0, 4^2)$ are independent samples of normal random variables, i.e., the columns of $A$ are drawn from two different multivariate normal distributions. Figure 1 shows that the two left singular vectors, though orthogonal by construction, fail to represent the true nature of the data; in contrast, the first two columns selected by the DEIM-CUR procedure give a much better overall representation. While trivial in this two-dimensional case, one can imagine the utility of such approximations for high-dimensional data. We shall illustrate the advantages of CUR approximations with further computational examples in Section 6.

CUR-type factorizations originated with “pseudoskeleton” approximations [11] and pivoted, truncated QR decompositions [20]; in recent years many new algorithms have been proposed in the numerical linear algebra and theoretical computer science literatures. Some approaches seek to maximize the volume of the decomposition [11].

Figure 1: Comparison of singular vectors (left, scaled, in red) and DEIM-CUR columns (right, in blue) for a data set drawn from two multivariate normal distributions having different principal axes.
Numerous other algorithms instead use leverage scores \[^1\] to approximate the SVD \[^1\]. These methods typically first compute a singular value decomposition \[^1\] of \( \mathbf{A} = \mathbf{VSW}^T \) (or an approximation to it), with \( \mathbf{V} \in \mathbb{R}^{m \times n}, \mathbf{W} \in \mathbb{R}^{n \times n} \). The leverage score for the \( j \)th row (\( k \)th column) of \( \mathbf{A} \) is the squared two-norm of the \( j \)th row of \( \mathbf{V} \) (\( k \)th row of \( \mathbf{W} \)). When scaled by the number of singular vectors, these leverage scores give probability distributions for randomly sampling the columns and rows to form \( \mathbf{C} \) and \( \mathbf{R} \). This approach leads to probabilistic bounds on \( \| \mathbf{A} - \text{CUR} \|_F \) \[^8\] \[^19\]. In cases where \( \mathbf{A} \) has small singular values (precisely the case where one would seek a low-rank factorization), the singular vectors can be sensitive to perturbations to \( \mathbf{A} \), making the leverage scores unstable \[^15\]. Thus leverage scores are often computed using only the leading few singular vectors, but the choice of how many vectors to keep can be somewhat ad hoc.

The algorithm described in Sections 2 and 3 is entirely deterministic and involves few (if any) parameters. The method is supported by error analysis presented in Section 4, which can also be applied to a broad class of CUR factorizations. In Section 5 we propose a novel incremental QR algorithm for approximating the SVD (and potentially also approximating leverage scores). Section 6 illustrates the performance of this new CUR factorization on several examples.

## 2 CUR Factorization

We are concerned with large matrices \( \mathbf{A} \in \mathbb{R}^{m \times n} \) that represent nearly low-rank data, which can therefore be expressed as

\[
\mathbf{A} = \text{CUR} + \mathbf{F},
\]

with \( \| \mathbf{F} \| \) small relative to \( \| \mathbf{A} \| \). The matrix \( \mathbf{C} \in \mathbb{R}^{m \times k} \) is formed by extracting \( k \) columns from \( \mathbf{A} \), and \( \mathbf{R} \in \mathbb{R}^{k \times n} \) from \( k \) rows of \( \mathbf{A} \). List the indices of the selected rows and columns in the vectors \( \mathbf{p}, \mathbf{q} \in \mathbb{N}^k \), so that \( \mathbf{C} = \mathbf{A}(\cdot, \mathbf{q}) \) and \( \mathbf{R} = \mathbf{A}(\mathbf{p}, \cdot) \). Our choice for \( \mathbf{p} \) and \( \mathbf{q} \) is guided by knowledge of the rank-\( k \) SVD (or an approximation to it). Before detailing the method for selecting these indices, we discuss how, given \( \mathbf{p} \) and \( \mathbf{q} \), one should construct \( \mathbf{U} \) so that \( \text{CUR} \) satisfies desirable approximation properties.

As motivation, suppose for the moment that \( \mathbf{A} \) has exact rank \( k \), and \( \mathbf{C} \) and \( \mathbf{R} \) are full-rank subsets of the columns and rows of \( \mathbf{A} \). Now let \( \mathbf{Y} \in \mathbb{R}^{m \times k} \) and \( \mathbf{Z} \in \mathbb{R}^{n \times k} \) be any matrices that satisfy \( \mathbf{Y}^T \mathbf{C} = \mathbf{RZ} = \mathbf{I} \in \mathbb{R}^{k \times k} \). Then \( \mathbf{CY}^T \) is a projector onto \( \text{Ran}(\mathbf{C}) = \text{Ran}(\mathbf{A}) \) and \( \mathbf{(ZR)}^T \) is a projector onto \( \text{Ran}(\mathbf{R}^T) = \text{Ran}(\mathbf{A}^T) \), where \( \text{Ran}(\cdot) \) denotes the range (column space). It follows that \( \mathbf{CY}^T \mathbf{A} = \mathbf{A} \) and \( \mathbf{(ZR)}^T \mathbf{A}^T = \mathbf{A}^T \). Putting \( \mathbf{U} \equiv \mathbf{Y}^T \mathbf{AZ} \) gives

\[
\text{CUR} = \mathbf{CY}^T \mathbf{AZR} = \mathbf{AZR} = \mathbf{A}.
\]

Thus, any choice of \( \mathbf{Y} \) and \( \mathbf{Z} \) that satisfies \( \mathbf{Y}^T \mathbf{C} = \mathbf{RZ} = \mathbf{I} \) gives a \( \mathbf{U} \) such that \( \text{CUR} \) exactly recovers \( \mathbf{A} \). In general different choices for \( \mathbf{Y} \) and \( \mathbf{Z} \) give different \( \mathbf{U} = \mathbf{Y}^T \mathbf{AZ} \).

Now consider the general case (\ref{2.1}). Once \( \mathbf{p}, \mathbf{q}, \mathbf{Y}, \) and \( \mathbf{Z} \) have been specified, then

\[
\mathbf{U} = \mathbf{Y}^T \mathbf{AZ} \quad \text{and} \quad \mathbf{F} \equiv \mathbf{A} - \text{CUR}.
\]

One might design \( \mathbf{Y} \) and \( \mathbf{Z} \) so that \( \text{CUR} \) matches the selected columns \( \mathbf{C} = \mathbf{A}(\cdot, \mathbf{q}) \) and rows \( \mathbf{R} = \mathbf{A}(\mathbf{p}, \cdot) \) of \( \mathbf{A} \) exactly. This can be accomplished with interpolatory projectors, which we discuss in detail in the next section.

\[^1\]We use the nonstandard notation \( \mathbf{VSW}^T \) for the SVD to avoid conflicts with the standard CUR notation.
For now, let $P = I(:,p) \in \mathbb{R}^{m \times k}$ and $Q = I(:,q) \in \mathbb{R}^{n \times k}$ be submatrices of the identity, so that $P^T a = a(p)$ and $b^T Q = b(q)^T$ for arbitrary vectors $a$ and $b$ of appropriate dimensions. Now define $Y^T = (P^T C)^{-1} P^T$ and $Z = Q(RQ)^{-1}$ (presuming $P^T C$ and $RQ$ are invertible). Then since $C = A(:,q)$ and $R = A(p,:)$,

$$P^T C = C(p,:) = A(p,q) \quad \text{and} \quad RQ = R(:,q) = A(p,q),$$

so

$$U = Y^T A Z = (P^T C)^{-1} P^T AQ(RQ)^{-1} = A(p,q)^{-1} A(p,q) A(p,q)^{-1} = A(p,q)^{-1}.$$

This CUR approximation matches the $q$ columns and $p$ rows of $A$, 

$$A(:,q) = \text{CUR}(:,q) \quad \text{and} \quad A(p,:) = C(p,:) UR,$$

and, in our experiments, usually delivers a very good approximation. However, a CUR factorization with better theoretical approximation properties results from orthogonal projection, as originally suggested by Stewart [20, p. 320]; see also, e.g., Mahoney and Drineas [19].

Given a selection of indices $p$ and $q$, again put

$$C = A(:,q) \quad \text{and} \quad R = A(p,:).$$

Assume that $C$ and $R$ both have full rank $k$, and now let $Y^T = C^I = (C^T C)^{-1} C^T$ and $Z = R^I = R^T (R R^T)^{-1}$ denote left and right inverses of $C$ and $R$. These choices also satisfy $Y^T C = I$ and $RZ = I$, but now $C Y^T = CC^I$ and $ZR = R^I R$ are orthogonal projectors. We compute

$$U = Y^T A Z = C^I A R^I,$$

yielding a CUR factorization that can be viewed as a two step process: first the columns of $A$ are projected onto $\text{Ran}(C)$, then the result is projected onto the row space of $R$:

1) $M = CC^I A,$  
2) $\text{CUR} = MR^I R.$

Both steps are optimal with respect to the 2-norm error, which is the primary source of the excellent approximation properties of this approach.

Several strategies for selecting $p$ and $q$ have been proposed\footnote{In the theoretical computer science literature, one often takes $C$ and/or $R$ to have rank larger than $k$, but then builds $U$ with rank $k$. By selecting these extra columns and/or rows, one seeks to get within some factor $1 + \varepsilon$ of the optimal approximation; see, e.g., [8].}

The approach presented in the next section is simple to implement and has complexity $mk$ and $nk$ to select the indices $p$ and $q$, provided the leading $k$ right and left singular vectors of $A$ are available. Thus the overall complexity is dominated by the construction of the rank-$k$ SVD $A \approx VSW^T$, where $V^T V = W^T W = I \in \mathbb{R}^{k \times k}$ and $S = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)$ is the $k \times k$ matrix of dominant singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$.

### 3 DEIM

The DEIM point selection algorithm was first presented in [6] in the context of model order reduction of nonlinear dynamical systems, and is a discrete variant of the Empirical Interpolation Method originally proposed in [4].

The DEIM procedure operates on the singular vector matrices $V$ and $W$ independently to select the row indices $p$ and column indices $q$. We explain the process for selecting $p$; applying the same steps to $W$ yields $q$. To derive the method, we elaborate upon the interpolatory projectors introduced in the last section.
Definition 3.1 Given a full rank matrix $V \in \mathbb{R}^{m \times k}$ and a set of distinct indices $p \in \mathbb{N}^k$, the interpolatory projector for $p$ onto $\text{Ran}(V)$ is
\[ P \equiv V(P^T V)^{-1}P^T, \]  
(3.1)
where $P = I(:, p) \in \mathbb{R}^{m \times k}$, provided $P^T V$ is invertible.

In general $P$ is an oblique projector, and it has an important property not generally enjoyed by orthogonal projectors: for any $x \in \mathbb{R}^m$,
\[ (Px)(p) = P^TPx = P^TV(P^TV)^{-1}P^Tx = P^Tx = x(p), \]
so the projected vector $Px$ matches $x$ in the $p$ entries, justifying the name “interpolatory projector.”

The DEIM algorithm processes the columns of

\[ V = [v_1 \ v_2 \ \cdots \ v_k] \]

one at a time, starting from the leading singular vector $v_1$. Each step processes the next singular vector to produce the next index. The first index $p_1$ corresponds to the largest magnitude entry in $v_1$:
\[ |v_1(p_1)| = \|v_1\|_\infty. \]

Now define $p_1 \equiv |p_1|$, and let
\[ P_1 \equiv v_1(P_1^Tv_1)^{-1}P_1^T \]
denote the interpolatory projector for $p_1$ onto $\text{Ran}(v_1)$. The second index $p_2$ corresponds to the largest entry in $v_2$, after the interpolatory projection in the $v_1$ direction has been removed:
\[ r_2 \equiv v_2 - P_1v_2 \]
\[ |r_2(p_2)| = \|r_2\|_\infty. \]

Notice that $r_2(p_1) = 0$, since $P_1v_2$ matches $v_2$ in the $p_1$ position, a consequence of interpolatory projection. This property ensures the process will never produce duplicate indices.

Now suppose we have $j - 1$ indices, with
\[ p_{j-1} \equiv \begin{bmatrix} p_1 \\ \vdots \\ p_{j-1} \end{bmatrix}, \quad P_{j-1} \equiv I(:, p_{j-1}), \quad V_{j-1} \equiv [v_1 \ \cdots \ v_{j-1}], \quad P_{j-1} \equiv V_{j-1}(P_{j-1}^TV_{j-1})P_{j-1}^T. \]

To select $p_j$, remove from $v_j$ its interpolatory projection onto the indices $p_{j-1}$ and take the largest remaining entry:
\[ r_j \equiv v_j - P_{j-1}v_j \]
\[ |r_j(p_j)| = \|r_j\|_\infty. \]

Implementations should not explicitly construct these projectors; see the pseudocode in Algorithm 1 for details.
Input: \( V \), an \( m \times k \) matrix  
Output: \( p \), an integer vector with \( k \) distinct entries in \( \{1 : m\} \)

\[
v = V(:, 1) \\
[\sim, p_1] = \text{max}(|v|) \\
p = [p_1] \\
\text{for } j = 2, 3, \ldots, k \\
v = V(:, j) \\
c = V(p, 1 : j - 1)^{-1}v(p) \\
r = v - V(:, 1 : j - 1)c \\
[\sim, p_j] = \text{max}(|r|) \\
p = [p; p_j]
\]

end

Algorithm 1: DEIM point selection algorithm.

Since the DEIM algorithm processes the singular vectors sequentially, from most to least significant, it introduces new singular vector information in a coherent manner as it successively selects the \( k \) indices. Contrast this to index selection strategies based on leverage scores, where all singular vectors are incorporated at once via row norms of \( V \) and \( W \); to account for the fact that higher singular vectors are less significant, such approaches often instead compute leverage scores using only a few of the leading singular vectors.\(^3\)

For the interpolatory projector \( P_j \) to exist at the \( j \)th step, the matrix \( P_{j-1}^T V_{j-1} \) must be nonsingular. The linear independence of the columns of \( V \) assures this. In the following, \( e_j \) denotes the \( j \)th column of the identity matrix.

**Lemma 3.1** Let \( P_j = [e_{p_1}, e_{p_2}, \ldots, e_{p_j}] \) and let \( V_j = [v_1, v_2, \ldots, v_j] \) for \( 1 \leq j \leq k \). If \( \text{rank}(V) = k \), then \( P_j^T V_j \) is nonsingular for \( 1 \leq j \leq k \).

**Proof:** Suppose \( P_{j-1}^T V_{j-1} \) is nonsingular and let \( r_j = v_j - V_{j-1}(P_{j-1}^T V_{j-1})^{-1}P_{j-1}^T v_j \). Then \( \| r_j \|_{\infty} > 0 \), for otherwise \( 0 = v_j - V_{j-1}e_{j-1} \), in violation of the assumption that \( \text{rank}(V) = k \).

Hence

\[
0 < |e_{p_j}^T r_j| = |e_{p_j}^T v_j - e_{p_j}^T V_{j-1}(P_{j-1}^T V_{j-1})^{-1}P_{j-1}^T v_j|,
\]

where \( p_j \) is the \( j \)th DEIM interpolation point. Now, the matrix

\[
P_j^T V_j = \begin{bmatrix} P_{j-1}^T V_{j-1} & P_{j-1}^T V_j \\ e_{p_j}^T V_{j-1} & e_{p_j}^T v_j \end{bmatrix} = \begin{bmatrix} I_{j-1} & 0 \\ e_{p_j}^T V_{j-1}(P_{j-1}^T V_{j-1})^{-1} & 1 \end{bmatrix} \begin{bmatrix} P_{j-1}^T V_{j-1} & P_{j-1}^T V_j \\ 0 & v_j \end{bmatrix}
\]

\(^3\)A potential limitation of the DEIM approach is that \( r_j \) could have multiple entries that have nearly the same magnitude, but only one index is selected at the \( j \)th step; if the other large-magnitude entries in \( r_j \) are not significant in subsequent \( r_k \) vectors, the corresponding indices will not be selected. One can imagine modifications of the selection algorithm to account for such situations, e.g., by processing multiple singular vectors at a time.
Lemma 4.1

Assume

\[
\nu_j = e_{p_j}^T v_j - e_{p_j}^T v_{j-1} (P_{j-1}^T v_{j-1})^{-1} P_{j-1}^T v_j.
\]

Inequality (3.2) implies \( \nu_j \neq 0 \) and hence equation (3.3) implies \( P_j^T v_j \) is nonsingular. Since \( e_{p_1}^T v_1 \neq 0 \), this argument provides an inductive proof that \( P_j^T v_j \) is nonsingular for \( 1 \leq j \leq k \).

\[\blacksquare\]

4 CUR Approximation Properties

While the theory presented in this section was designed to bound \( \| A - \text{CUR} \| \) for the DEIM-CUR method, the analysis applies to any CUR factorization with full rank \( C \in \mathbb{R}^{m \times k} \) and \( R \in \mathbb{R}^{k \times n} \), and \( U = C'AR' \), regardless of the procedure used for selecting the columns and rows.\(^4\)

Consider a CUR factorization that uses row indices \( p \in \mathbb{N}^k \) and column indices \( q \in \mathbb{N}^k \), and set

\[
P = I(:, p) = [e_{p_1}, \ldots, e_{p_k}] \in \mathbb{R}^{m \times k}, \quad Q = I(:, q) = [e_{q_1}, \ldots, e_{q_k}] \in \mathbb{R}^{n \times k}.
\]

The first step in this analysis bounds the mismatch between \( A \) and its interpolatory projection \( PA \).

\textbf{Lemma 4.1} Assume \( P^T V \) is invertible and let \( \mathcal{P} = V(P^T V)^{-1} P^T \) be the interpolatory projector (3.1). If \( V^T V = I \), then any \( A \in \mathbb{R}^{m \times n} \) satisfies

\[
\| A - P A \| \leq \| (P^T V)^{-1} \| (I - VV^T) A \|.
\]

Additionally, if \( V \) consists of the leading \( k \) left singular vectors of \( A \), then

\[
\| A - P A \| = \| (I - \mathcal{P}) A \| \leq \| (P^T V)^{-1} \| \sigma_{k+1}.
\]

\textbf{Proof:} First note that \( \mathcal{P} V = V(P^T V)^{-1} P^T V = V \), so that \( (I - \mathcal{P}) V = 0 \). Therefore

\[
\| A - P A \| = \| (I - \mathcal{P}) A \| = \| (I - \mathcal{P})(I - VV^T) A \| \leq \| (I - \mathcal{P}) \| \| (I - VV^T) A \|.
\]

It is well known that

\[
\| (I - \mathcal{P}) \| = \| \mathcal{P} \| = \| (P^T V)^{-1} \|
\]

so long as \( \mathcal{P} \neq 0 \) or \( I \) (see, e.g., [21]). This establishes the first result. The second follows from the fact that

\[
\| (I - VV^T) A \| = \| A - VSW^T \| = \sigma_{k+1}
\]

when \( V \) consists of the leading \( k \) left singular vectors of \( A \).

\[\blacksquare\]

Now let \( VSW^T \approx A \) be a rank-\( k \) SVD of \( A \). (The singular vectors play a crucial role in this analysis, even if \( p \) and \( q \) were selected using some scheme that did not use them.) In addition to the interpolatory projector \( \mathcal{P} = V(P^T V)^{-1} P^T \) that operates on the left of \( A \), we shall also use \( Q = Q(W^T Q)^{-1} W^T \), which operates on the right of \( A \). Assuming that \( P^T V \) and \( W^T Q \) are invertible, define the error constants

\[
\eta_p \equiv \| (P^T V)^{-1} \|, \quad \eta_q \equiv \| (W^T Q)^{-1} \|.
\]

\(^4\)We are grateful to Ilse Ipsen for pointing out the general applicability of this analysis.
Lemma 4.1 implies
\[
\|A(I - Q)\| \leq \eta_q \sigma_{k+1} \quad \text{and} \quad \|(I - \mathcal{P})A\| \leq \eta_p \sigma_{k+1}.
\] (4.1)

The next lemma shows that these bounds on the error of the interpolatory projection of A onto the select columns and rows also apply to the orthogonal projections of A onto the same column and row spaces.

**Lemma 4.2** Suppose the row and column indices p and q give full rank matrices C = A(:, q) = AQ ∈ R^{m×k} and R = A(p, :) = PA ∈ R^{k×n}, with finite error constants η_p and η_q, and suppose that k < min{m, n}. Then
\[
\|(I - CC^T)A\| \leq \eta_q \sigma_{k+1} \quad \text{and} \quad \|A(I - R^TR)\| \leq \eta_p \sigma_{k+1}.
\]

**Proof:** Using the formula C = AQ, we have C^T = (C^T C)^{-1} C = AQ(Q^T A^T AQ)^{-1}, so the orthogonal projection of A onto Ran(C) is
\[
CC^T A = (AQ(Q^T A^T AQ)^{-1} Q^T A^T) A = A(Q(Q^T A^T AQ)^{-1} Q^T A^T A).
\]

Hence the error in the orthogonal projection of A is
\[
(I - CC^T)A = A(I - \Phi), \quad \text{where} \quad \Phi = Q(Q^T A^T AQ)^{-1} Q^T A^T A. \tag{4.2}
\]

Note that Φ is an oblique projector onto Ran(Q), so ΦQ = Q. Therefore, ΦQ = Q, since
\[
\Phi Q = \Phi Q(W^T Q)^{-1} W^T = Q(W^T Q)^{-1} W^T = Q.
\]

This implies that
\[
A(I - \Phi) = A(I - \Phi)(I - Q) = (I - CC^T)A(I - Q),
\]
and so from (4.2) we have
\[
\|(I - CC^T)A\| = \|A(I - \Phi)\|
= \|(I - CC^T)A(I - Q)\|
\leq \|I - CC^T\|\|A(I - Q)\|
\leq \eta_q \sigma_{k+1}.
\]

The last line follows from the bound (4.1) and the fact that \|I - CC^T\| = 1, since CC^T is an orthogonal projector and k < min{m, n}.

A similar argument shows that
\[
A(I - R^TR) = (I - \Psi)A
\]
where \(\Psi = AA^T P (P^T AA^T P)^{-1} P^T\), and also that
\[
(I - \Psi)A = (I - \mathcal{P})(I - \Psi)A = (I - \mathcal{P})A(I - R^TR),
\]
from which follows the error bound
\[
\|A(I - R^TR)\| \leq \|(I - \mathcal{P})A\|\|I - R^TR\| \leq \eta_p \sigma_{k+1}.
\]

The main result on approximation of A by CUR readily follows from combining this last lemma with a basic CUR analysis technique used by Mahoney and Drineas [19, eq. (6)].
Theorem 4.1 Given $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $1 \leq k < \min\{m, n\}$, let $\mathbf{C} = \mathbf{A}(\cdot, \mathbf{q}) \in \mathbb{R}^{m \times k}$ and $\mathbf{R} = \mathbf{A}(\mathbf{p}, \cdot) \in \mathbb{R}^{k \times n}$ with finite error constants $\eta_p$ and $\eta_q$, and set $\mathbf{U} = \mathbf{C}^t \mathbf{A} \mathbf{R}^t$. Then

$$\|\mathbf{A} - \mathbf{CUR}\| \leq (\eta_p + \eta_q)\sigma_{k+1}.$$ 

Proof: From the definitions,

$$\mathbf{A} - \mathbf{CUR} = \mathbf{A} - \mathbf{CC}^t \mathbf{AR}^t = (\mathbf{I} - \mathbf{CC}^t)\mathbf{A} + \mathbf{CC}^t\mathbf{A}(\mathbf{I} - \mathbf{R}^t\mathbf{R}).$$

Applying Lemma 4.2,

$$\|\mathbf{A} - \mathbf{CUR}\| \leq \|(\mathbf{I} - \mathbf{CC}^t)\mathbf{A}\| + \|\mathbf{CC}^t\|\|\mathbf{A}(\mathbf{I} - \mathbf{R}^t\mathbf{R})\|$$

$$\leq \eta_q\sigma_{k+1} + \eta_p\sigma_{k+1}$$

$$= (\eta_p + \eta_q)\sigma_{k+1},$$

since $\|\mathbf{CC}^t\| = 1$. $\blacksquare$

Theorem 4.1 shows that CUR is within a factor of $\eta_p + \eta_q$ of the optimal rank-$k$ approximation, hence these error constants suggest a way to assess a wide variety of column/row selection schemes. The quality of the approximation is controlled by the conditioning of the selected $k$ rows of the dominant $k$ (exact) singular vectors. If those singular vectors are available as part of the column/row selection process, then Theorem 4.1 provides an a posteriori bound requiring only the fast ($O(k^3)$) computation of $\eta_p$ and $\eta_q$, and thus could suggest methods for adjusting either $k$ or the point selection process to reduce the error constants. In this context, notice that if $\mathbf{VSW}^T$ is only an approximation to the optimal rank-$k$ SVD with $\mathbf{V}$ and $\mathbf{W}$ having orthonormal columns (as computed, for example, using the incremental QR algorithm described in the next section), the preceding analysis gives

$$\|\mathbf{A} - \mathbf{CUR}\| = \|(\mathbf{I} - \mathbf{CC}^t)\mathbf{A}\| + \|\mathbf{A}(\mathbf{I} - \mathbf{R}^t\mathbf{R})\|$$

$$= \|\mathbf{A}(\mathbf{I} - \mathbf{Q})\| + \|(\mathbf{I} - \mathbf{P})\mathbf{A}\|$$

$$= \|\langle\mathbf{W}^T\rangle^{-1}\|\|\mathbf{A}(\mathbf{I} - \mathbf{WW}^T)\| + \|\langle\mathbf{P}^T\rangle^{-1}\|\|\mathbf{A}(\mathbf{I} - \mathbf{VV}^T)\|,$$

showing how $\sigma_{k+1}$ in Theorem 4.1 is replaced by the error in the approximate SVD through $\|\mathbf{A}(\mathbf{I} - \mathbf{WW}^T)\|$ and $\|\langle\mathbf{W}^T\rangle^{-1}\|\|\mathbf{A}(\mathbf{I} - \mathbf{WW}^T)\|$. In this case $\langle\mathbf{W}^T\rangle^{-1}$ and $\langle\mathbf{P}^T\rangle^{-1}$ are computed using the approximate singular vectors in $\mathbf{V}$ and $\mathbf{W}$, rather than the exact singular vectors in the theorem. Alternatively, if one has probabilistic bounds for $\eta_p$ and $\eta_q$, then Theorem 4.1 immediately gives a probabilistic bound for $\|\mathbf{A} - \mathbf{CUR}\|$.

Numerical examples in Section 6 compare how the error constants evolve as $k$ increases for the DEIM-CUR factorization and several other factorizations based on leverage scores.

4.1 Interpretation of the bound for DEIM-CUR

For DEIM-CUR, we can ensure the hypotheses of Theorem 4.1 are satisfied and bound the error constants. Suppose the DEIM points are selected using the exact rank-$k$ SVD $\mathbf{A} \approx \mathbf{VSW}^T$. Lemma 3.1 ensures that the matrices $\mathbf{P}^T \mathbf{V}$ and $\mathbf{W}^T \mathbf{Q}$ are invertible, so $\eta_p$ and $\eta_q$ are finite. The DEIM strategy also gives full rank $\mathbf{C}$ and $\mathbf{R}$ matrices, presuming $k \leq \text{rank}(\mathbf{A})$. To see this, note that for any unit vector $\mathbf{y} \in \mathbb{R}^k$,

$$\mathbf{C} \mathbf{y} = \mathbf{A} \mathbf{Q} \mathbf{y} = \mathbf{VSW}^T \mathbf{Q} \mathbf{y} + \mathbf{E} \mathbf{Q} \mathbf{y},$$
where \( E = A - VSW^T \). Since \( VT^E = 0 \),
\[
\|Cy\|^2 = \|AQy\|^2 = \|VSW^TQy\|^2 + \|EQy\|^2.
\]

Since \( \|W^TQy\| \geq \|y\|/\|(W^TQ)^{-1}\| = 1/\eta_q \),
\[
\|Cy\| \geq \|VSW^TQy\| \geq \sigma_k/\eta_q > 0.
\]
Thus \( C \) must be full rank. A similar argument shows \( R \) to be full rank as well.

The examples in Section [6] illustrate that \( \eta_p \) and \( \eta_q \) are often quite modest for the DEIM-CUR approach, e.g., \( O(100) \). However, worst-case bounds permit significant growth in \( k \) that is generally not observed in practice.

**Lemma 4.3 (Chaturantabut and Sorensen [6, Lemma 3.2])** For the DEIM selection scheme derived above,
\[
\eta_p \leq \frac{(1 + \sqrt{2m})^{k-1}}{\|v_1\|_\infty}, \quad \eta_q \leq \frac{(1 + \sqrt{2n})^{k-1}}{\|w_1\|_\infty},
\]
where \( v_1 \) and \( w_1 \) denote the first columns of \( V \) and \( W \).

## 5 Incremental QR Factorization

The DEIM point selection process presumes access to the first \( k \) left and right singular vectors of \( A \in \mathbb{R}^{m \times n} \). If either \( m \) or \( n \) is of modest size (say \( \leq 1000 \)) and \( A \) can be stored as a dense matrix, library software for computing the “economy sized” SVD, e.g., \([V, S, W] = \text{svd}(A, \text{‘econ’})\) in MATLAB, usually performs very well. For larger scale problems, the leading \( k \) singular vectors can be computed using iterative methods, such as the Krylov subspace-based ARPACK software [18] (which is used by MATLAB’s \texttt{svds} command), PROPACK [17], or IRLBA [1], or the Jacobi–Davidson algorithm [14]. Randomized SVD algorithms provide an appealing alternative for large problems [13].

Another attractive possibility is an **incremental** low rank \( A \approx QR \) approximation, where \( Q \in \mathbb{R}^{n \times k} \) has orthonormal columns and \( R \in \mathbb{R}^{k \times m} \) is upper triangular. (In this section only, \( Q \) and \( R \) denote different quantities from elsewhere in the paper.) Take the dense (economy sized) SVD \( R = \hat{V}SW^T \), and put \( V = Q\hat{V} \) to get
\[
A \approx QR = VSW^T.
\]

Incremental algorithms for building the QR factorization and SVD have been proposed by Stewart [20], Baker, Gallivan, and Van Dooren [2] and many others, as surveyed in [3]; these ideas are also closely related to rank-revealing QR factorizations [12]. Algorithm [2] differs from those of Stewart in its use of internal pivoting and threshold truncation in place of Stewart’s column pivoting. This distinction enables a one pass algorithm that is closely related to [2, Algorithm 1].

The proposed method is presented in Algorithm [2] which proceeds at each step by orthogonalizing a column of \( A \) against the previously orthogonalized columns. The rank of the resulting factors is controlled through an update-and-delete procedure that is illustrated in Figure [2]. After orthogonalizing a column of \( A \), the algorithm checks if any row of \( R \) has small relative norm; if such a row exists, the corresponding column of \( Q \) makes little
Input:  \( A \), an \( m \times n \) matrix
        \( tol \), a positive scalar controlling the accuracy of the factorization

Output:  \( Q \), an \( m \times k \) matrix with orthonormal columns
         \( R \), a \( k \times n \) rectangular matrix
         with \( A \approx QR \)

Choose \( k \ll \min(m, n) \)

Compute the QR factorization \( A(:,1:k) = QR \), with \( Q \in \mathbb{R}^{n \times k} \) and \( R \in \mathbb{R}^{k \times m} \)

\[ \text{rownorms}(i) = \| R(i,:) \|_2^2 \text{ for } i = 1, \ldots, k \]

\( j = k + 1 \)

while \( j \leq n \)

\( a = A(:,j); \quad r = Q^T a; \quad f = a - Qr; \quad \rho = \| f \|; \quad q = f / \rho \)

\( Q = [Q, q]; \quad R = \begin{bmatrix} R & r \\ 0 & \rho \end{bmatrix} \)

\[ \text{rownorms}(i) = \text{rownorms}(i) + r(i)^2 \text{ for } i = 1, \ldots, k \]

\[ \text{rownorms}(k+1) = \rho^2; \]

\[ \text{FnormR} = \text{sum} \left( \text{rownorms} \right); \]

\[ [\sigma, i_{\text{min}}] = \text{min} \left( \text{rownorms}(1:k+1) \right); \]

if \( \sigma > (tol^2) \ast (\text{FnormR} - \text{rownorms}(i_{\text{min}})) \)

\% no deflation

\[ k = k + 1; \]

else

\% deflation required

if \( i_{\text{min}} < k + 1 \)

\[ R(i_{\text{min}},:) = R(k+1,:); \quad Q(:,i_{\text{min}}) = Q(:,k+1) \]

\[ \text{rownorms}(i_{\text{min}}) = \text{rownorms}(k+1) \]

end

\% delete the minimum norm row of \( R \)

\( Q = Q(:,1:k); \quad R = R(1:k,:) \)

end

\( j = j + 1 \)

end

Algorithm 2: Incremental QR low rank approximate factorization

contribution to the factorization, so that column of \( Q \) and row of \( R \) can be deleted at only a small loss of accuracy
in the factorization. (Future columns of \( A \) will not be orthogonalized against the vector deleted from \( Q \), so this
direction can re-emerge if a later column in \( A \) warrants it.)
Robust implementations of Algorithm 2 should replace the classical Gram–Schmidt operations
\[ r = Q^T a, \quad f = a - Qr \]
with a re-orthogonalization step, as suggested by Daniel, Gragg, Kaufman, and Stewart [7]:
\[
\begin{align*}
 r &= Q^T a \\
 f &= a - Qr \\
 c &= Q^T f \\
 f &= f - Qc \\
 r &= r + c \\
\end{align*}
\]
\begin{align*}
 \rho &= ||f|| \\
 q &= f / \rho.
\end{align*}

The extra steps (5.1)–(5.3) generally provide a \( Q \) that is numerically orthogonal to working precision. Pathological cases are easily overcome with some additional slight modifications; see [10] for a complete analysis. Because this algorithm uses the classical Gram–Schmidt method, one can easily block it for parallel efficiency.
5.1 Incremental QR Error Bounds

At step $j$ the truncation criterion in Algorithm 2 will delete row $r_j^T = e_j^T R_j$ if

$$
\|r_i\| \leq \text{tol} \cdot \|\hat{R}_j\|_F,
$$

where $r_i^T$ is the row of minimum norm and $\hat{R}_j$ denotes $R_j$ with the $i$th row deleted. This strategy has a straightforward error analysis.

Lemma 5.1 Let $R_j$ be the triangular factor at step $j$ of Algorithm 2 and $Q_j$ the corresponding orthonormal columns in the approximate QR factorization $A_j \approx Q_j R_j$, where $A_j$ consists of the first $j$ columns of $A$. Then

$$
\|A_j - Q_j R_j\|_F \leq \text{tol} \cdot d_j \cdot \|R_j\|_F,
$$

where $d_j$ is the number of row deletions that have been made up to and including step $j$. (Note that $Q_j \in \mathbb{R}^{m \times (j-d_j)}$, $R \in \mathbb{R}^{(j-d_j) \times j}$, and $d_n = \min\{m, n\} - k$, where $k = \text{rank}(Q_n R_n)$.)

Proof: The proof shall be by induction. Let $E_j = A_j - Q_j R_j$ and assume

$$
\|E_j\|_F \leq \text{tol} \cdot d_j \cdot \|R_j\|_F. \hspace{1cm} (5.4)
$$

Orthogonalize column $j + 1$ of $A$ using Gram–Schmidt to obtain

$$
A_{j+1} = Q_{j+1} R_{j+1} + [E_j, 0].
$$

If no deflation occurs at this step, the bound holds trivially since

$$
\|E_{j+1}\|_F = \|E_j\|_F \leq \text{tol} \cdot d_j \cdot \|R_j\|_F \leq \text{tol} \cdot d_{j+1} \cdot \|R_{j+1}\|_F,
$$

because $d_{j+1} = d_j$ and $\|R_j\| \leq \|R_{j+1}\|_F$.

Suppose $R_j$ has dimension $k \times j$ (i.e., $k = j - d_j$). Let $i$ be the index of the row of minimum norm and let $\hat{R}_{j+1}$ be obtained by deleting the $i$th row of $R_{j+1}$. If $r_i^T = e_i^T R_{j+1}$ satisfies $\|r_i^T\| \leq \text{tol} \cdot \|\hat{R}_{j+1}\|_F$ then deflation occurs. Deleting row $i$ of $R_{j+1}$ and column $i$ of $Q_{j+1}$ replaces $Q_{j+1}$ and $R_{j+1}$ with $\hat{Q}_{j+1}$ and $\hat{R}_{j+1}$. Then

$$
\hat{Q}_{j+1} \hat{R}_{j+1} = Q_{j+1}(R_{j+1} - e_i r_i^T),
$$

and

$$
A_{j+1} = Q_{j+1}(R_{j+1} - e_i r_i^T) + [E_j, 0] + Q_{j+1} e_i r_i^T.
$$

Hence the deletion gives the overall error

$$
E_{j+1} = A_{j+1} - \hat{Q}_{j+1} \hat{R}_{j+1} = [E_j, 0] + Q_{j+1} e_i r_i^T.
$$

Therefore, when $i < k + 1$, the inductive assumption (5.4) implies

$$
\|E_{j+1}\|_F \leq \|E_j\|_F + \|r_i^T\| \leq \text{tol} \cdot (d_j \cdot \|R_j\|_F + \|\hat{R}_{j+1}\|_F) \leq \text{tol} \cdot (d_j + 1) \cdot \|\hat{R}_{j+1}\|_F.
$$
since $\hat{R}_{j+1}$ contains row $k + 1$ of $R_{j+1}$, which must have a norm larger than the row marked for deletion. Since row $k + 1$ of $\hat{R}_{j+1}$ consists of just one nonzero element,

$$\|\hat{R}_{j+1}\|_F^2 \geq \|\hat{R}_j\|_F^2 + \rho_{k+1,j+1}^2 \geq \|R_j\|_F^2,$$

where $\rho_{k+1,j+1}$ is the element $R_{j+1}(k + 1, j + 1)$ and $\hat{R}_j$ is the matrix $R_j$ with $i$th row deleted. If $i = k + 1$, then the last row of $R_{j+1}$ is deleted and the desired inequality must hold, since $R_j$ is a submatrix of $\hat{R}_{j+1}$. At the end of this process, replace $R_{j+1}$ and $Q_{j+1}$ with $\hat{R}_{j+1}$ and $\hat{Q}_{j+1}$ to obtain the approximation

$$\|A_{j+1}\| \leq tol \cdot d_{j+1} \cdot \|R_{j+1}\|_F,$$

since $d_{j+1} = d_j + 1$.

The error bound for the base case $j = 1$ clearly holds, completing the induction.

The approximate QR factorization that results from this algorithm could be used directly for the approximation of leverage scores. The perturbation theory of Ipsen and Wentworth [15] describes how the tolerance in our algorithm will affect the accuracy of the resulting leverage scores.

6 Computational Examples

This section presents some computational evidence illustrating the excellent approximation properties indicated by the error analysis. For each of our three examples, we compare the accuracy of the DEIM-CUR factorization with several schemes based on leverage scores. To remove random variations from our experiments, in most cases we select columns and rows having the highest leverage scores; for the first example, we include results for random leverage score sampling.

Example 1. Low-rank approximation of a sparse, nonnegative matrix

The first example builds a matrix $A \in \mathbb{R}^{300,000 \times 300}$ of the form

$$A = \sum_{j=1}^{10} \frac{2}{j} x_j y_j^T + \sum_{j=11}^{300} \frac{1}{j} x_j y_j^T,$$

(6.1)

where $x_j \in \mathbb{R}^{300,000}$ and $y_j \in \mathbb{R}^{300}$ are sparse vectors with random nonnegative entries (in MATLAB, $x_j = \text{sprand}(30000, 1, 0.025)$ and $y_j = \text{sprand}(300, 1, 0.025)$). In this instantiation, $A$ has 15,971,584 nonzeros, i.e., about 18% of all entries are nonzero. The form (6.1) is not a singular value decomposition, since $\{x_j\}$ and $\{y_j\}$ are not orthonormal sets; however, due to the sparsity of these vectors this decomposition suggests the structure of the SVD: the singular values decay like $1/j$, and with the first ten singular values weighted more heavily to give a notable drop between $\sigma_{10}$ and $\sigma_{11}$.

Figure 3 compares the error $\|A - CUR\|$ for DEIM-CUR and methods that take $C$ and $R$ as the columns and rows of $A$ with the highest leverage scores. These scores are computed using either all right and left singular vectors (300 of each), or using only the leading ten right and left singular vectors. Both approaches perform rather worse than DEIM-CUR, which closely tracks the optimal value $\sigma_{k+1}$. 

14
To gain insight into these results, we examine the interpolation constants $\eta_p$ and $\eta_q$ for all three approaches. Figure 4 shows that these interpolation constants are largest for leverage scores based on all the singular vectors; using only ten singular vectors improves the interpolation constant, and also the performance of the approximation (as seen in Figure 3). The DEIM-CUR method improves the error constants, and likewise the quality of the approximation.

A CUR factorization can also be obtained by randomly sampling columns and rows of $A$, with the probability of selection weighted by leverage scores [19]. We apply this approach on the current example, selecting $k = 30$...
Figure 5: Accuracy of CUR approximations for (6.1) generated by randomly sampling rows and columns with probability weighted by leverage scores computed from the leading ten singular vectors. All ten trials (gray lines) perform similarly to the deterministic “LS (10)” approach, and worse than the DEIM-CUR approximation.

Now modify (6.1) to give a more significant drop between $\sigma_{10}$ and $\sigma_{11}$:

$$A = \sum_{j=1}^{10} \frac{1000}{j} x_j y_j^T + \sum_{j=11}^{300} \frac{1}{j} x_j y_j^T.$$  \hspace{1cm} (6.2)

As seen in Figures 5 and 7, the DEIM-CUR approach again delivers an excellent approximation, while selecting the rows and columns with highest leverage scores does not perform nearly as well. (In Figure 7 note the significant jump in the “LS (10)” error constant $\eta_q$ corresponding to those $k$ values where $\|A - C_k U_k R_k\|/\sigma_{k+1}$ is large.)

**Example 2. TechTC term document data**

The second example, adapted from Mahoney and Drineas \[19\], computes the CUR factorization of a term document matrix with data drawn from the Technion Repository of Text Categorization Datasets (TechTC) \[9\]. The rows of the data matrix correspond to websites (consolidated from multiple webpages), while the columns correspond to “features” (words from the text of the webpages). The $(j,k)$ entry of $A$ reflects the importance of the feature text on the given website; most entries are zero. For this experiment we use TechTC-100 test set 26, which concatenates a data set relating to Evansville, Indiana (id 10567) with another for Miami, Florida (id 11346). Following Mahoney and Drineas \[19\], we omit all features with four or fewer characters from the data set, leaving a matrix with 139 rows and 15,170 columns. Each row of $A$ is then scaled to have unit 2-norm. Ideally a CUR factorization not only gives an accurate low-rank approximation to $A$, but also selects rows corresponding to representative webpages from each geographic area, and columns corresponding to meaningful features.
Figure 6: Accuracy of CUR approximations using $k$ columns and rows, for DEIM-CUR and two leverage score strategies for the sparse, nonnegative matrix (6.2). "LS (all)" selects rows and columns having the highest leverage scores computed using all 300 singular vectors; "LS (10)" uses the leading 10 singular vectors.

Figure 7: Error constants $\eta_p = \| (P_k^T V_k)^{-1} \|$ and $\eta_q = \| (W_k Q_k)^{-1} \|$ for columns and rows selected using two leverage score strategies (left plot) and the DEIM algorithm (right plot), for the sparse matrix $A$ given in (6.2).

Figure 8 compares DEIM-CUR approximations to row and column selection based on highest leverage scores (from all singular vectors, or the two leading singular vectors). The DEIM-CUR method is generally more accurate than those based on leverage scores, but since the singular values decay slowly, the accuracy of all approaches is limited. How do the DEIM columns (features) compare to those with the highest leverage scores? Figure 9 shows the leverage scores associated with each column of $A$ (based on the two leading singular vectors), along with the first 30 columns selected by DEIM. While the columns with highest leverage scores were found by DEIM, there are DEIM columns with marginal leverage scores, and vice versa. This data is more easily parsed in Table 1, which
Figure 8: Accuracy of CUR factorizations for the TechTC example, selecting rows and columns using top leverage scores for all singular vectors and the leading two singular vectors, and DEIM.

Figure 9: The columns selected by DEIM for the TechTC example, compared to leverage scores from the leading two singular vectors.

lists the features corresponding to the first 20 DEIM columns. (For ease of comparison, we normalize leverage scores so that the maximum value is one.) The leading features identified by DEIM, including “evansville” (first DEIM point), “florida” (second), “miami” (sixth), and “indiana” (nineteenth) indeed reveal the key geographic terms. These terms scored at least as high when ranked by leverage scores based on two leading singular vectors; when all singular vectors are taken into account, the scores generally drop relative to other features. Overall, one notes that DEIM selects a significantly different set of indices than valued by leverage scores, and, as seen in Figure 8, tends to provide a somewhat better low-rank approximation.
Table 1: The features selected by DEIM-CUR for the TechTC data set, compared to the (scaled) leverage scores using the leading two singular vectors, and all singular vectors.

| DEIM rank | DEIM index | LS (2) rank | LS (2) score | LS (all) rank | LS (all) score | feature |
|-----------|------------|-------------|--------------|---------------|----------------|---------|
| 1         | 10973      | 1           | 1.000        | 4             | 0.875          | evansville |
| 2         | 1          | 2           | 0.741        | 8             | 0.726          | florida   |
| 3         | 1547       | 13          | 0.031        | 2             | 0.948          | spacer    |
| 4         | 109        | 8           | 0.055        | 66            | 0.347          | contact   |
| 5         | 209        | 12          | 0.040        | 32            | 0.458          | service   |
| 6         | 50         | 4           | 0.116        | 6             | 0.739          | miami     |
| 7         | 824        | 46          | 0.007        | 5             | 0.809          | chapter   |
| 8         | 1841       | 33          | 0.010        | 20            | 0.537          | health    |
| 9         | 171        | 5           | 0.113        | 13            | 0.617          | information |
| 10        | 234        | 16          | 0.026        | 37            | 0.436          | events    |
| 11        | 595        | 84          | 0.004        | 15            | 0.576          | church    |
| 12        | 60         | 15          | 0.026        | 67            | 0.347          | email     |
| 13        | 945        | 10          | 0.047        | 30            | 0.474          | services  |
| 14        | 1670       | 129         | 0.002        | 1             | 1.000          | bullet    |
| 15        | 216        | 35          | 0.009        | 38            | 0.430          | music     |
| 16        | 78         | 3           | 0.246        | 24            | 0.492          | south     |
| 17        | 213        | 19          | 0.018        | 110           | 0.259          | their     |
| 18        | 138        | 14          | 0.030        | 43            | 0.408          | please    |
| 19        | 6110       | 7           | 0.060        | 95            | 0.280          | indiana   |
| 20        | 1152       | 70          | 0.005        | 152           | 0.221          | member    |

Example 3. Tumor detection in genetics data

Our final example uses the GSE10072 cancer genetics data set from the National Institutes of Health, previously investigated by Kundu, Nambirijan, and Drineas [16]. The matrix $A \in \mathbb{R}^{22,283 \times 107}$ contains data for 22,283 probes applied to 107 patients. The $(j,k)$ entry of $A$ reflects how strongly patient $k$ responded to probe $j$. This experiment seeks probes that segment the population into two clusters: the 58 patients with tumors, and the 49 without.\footnote{The data is available from http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE10072} To center the data, we subtract the mean of each row from all entries in that row. As shown in [16], the leading two principal vectors of this matrix segment the population very well.

Like the TechTC data, the singular values of $A$ decay slowly, as seen in Figure 10. Once again the DEIM-CUR procedure produces a more accurate low-rank approximation than obtained by selecting the rows and columns with highest leverage scores, whether those are computed using all the singular vectors, or just the leading two or ten.

Table 2 reports the first 15 rows selected by the DEIM-CUR process, along with the corresponding leverage scores based on two, ten, and all singular vectors. Do the probes selected by DEIM discriminate the patients with tumors (“sick”) from those without (“well”)? To investigate, for each selected probe we count the number of large positive entries corresponding to sick and well patients.\footnote{In particular, we call an entry of the mean-centered matrix $A$ large if its value exceeds one. Of the 22,283 probes, for only 23 probes do at least 30 of the 58 sick patients have such large entries; for only 95 probes do at least 30 of the 49 well patients have large entries. There is no overlap between the probes that are strongly expressed by the sick and well patients.} Some but not all of the DEIM-CUR probes effectively
select only sick or well patients. Contrast these results with Table 3, which shows the probes with highest leverage scores (drawn from the leading two singular vectors). Only four of these probes were also selected by the DEIM procedure (even if we continue the DEIM procedure to select the maximum number, \( n = 107 \), of indices). This discrepancy is quite different from the good agreement between DEIM and leverage score indices for the TechTC data in Table 1, despite the similar dimensions and the comparably slow decay of the singular values.

While the rows selected from leverage scores did not produce as accurate an approximation, \( \|A - C_k U_k R_k\| \), as DEIM, these probes do a much more effective job of discriminating patients with tumors from those without. Indeed, for 14 of the top 15 probes, the tumor-free patients express strongly, while the patients with tumors do not; in the remaining case, the opposite occurs.

7 Conclusions

The Discrete Empirical Interpolation Method (DEIM) is an index selection procedure that gives simple, deterministic CUR factorizations of matrix \( A \). Since DEIM utilizes (approximate) singular vectors, we propose an effective one-pass incremental approximate QR factorization that can efficiently compute dominant singular vectors for data sets with rapidly decaying singular values; this method could prove useful in a variety of other settings. The accuracy of the resulting rank-\( k \) CUR factorization can be bounded in terms of \( \sigma_{k+1} \), the error in the best rank-\( k \) approximation to \( A \). Our analysis of the 2-norm error \( \|A - \text{CUR}\| \) applies to all CUR approximations that use the optimal central factor \( U = C^I A R^I \), and hence can give insight into the performance of other index selection algorithms, such as leverage scores or uniform random sampling. Numerical examples illustrate that the DEIM-CUR approach can deliver very good low rank approximations, compared to row selection based on dominant leverage scores.
**Acknowledgements**

We thank Inderjit Dhillon, Petros Drineas, Ilse Ipsen, and Michael Mahoney for a number of helpful discussions.

Table 2: Genetics example: the probes selected by DEIM-CUR, compared to the (scaled) leverage scores using the leading two singular vectors, ten singular vectors, and all singular vectors.

| DEIM index | probe set | gene name | number sick | number well | LS (2) rank | LS (2) score | LS (10) rank | LS (10) score | LS (all) rank | LS (all) score |
|------------|-----------|-----------|-------------|-------------|-------------|--------------|--------------|---------------|---------------|----------------|
| 1          | 210081_at | AGER      | 2           | 45          | 1           | 1.000        | 45           | 0.504         | 386           | 0.123          |
| 2          | 214895_at | ADAM10    | 8           | 3           | 211         | 0.173        | 1171         | 0.108         | 3344          | 0.036          |
| 3          | 209156_at | COL6A2    | 5           | 6           | 15156       | 0.005        | 252          | 0.245         | 708           | 0.091          |
| 4          | 211653_at | AKR1C2    | 18          | 1           | 6440        | 0.017        | 11           | 0.656         | 146           | 0.185          |
| 5          | 214777_at | IGKV4-1   | 37          | 3           | 281         | 0.148        | 19           | 0.607         | 106           | 0.209          |
| 6          | 219612_at | FGG       | 17          | 17          | 2591        | 0.039        | 2            | 0.956         | 4             | 0.825          |
| 7          | 204304_at | PROM1     | 16          | 4           | 992         | 0.073        | 70           | 0.417         | 32            | 0.345          |
| 8          | 203824_at | TSPAN8    | 17          | 4           | 9687        | 0.011        | 21           | 0.582         | 31            | 0.355          |
| 9          | 204748_at | PTGS2     | 18          | 14          | 424         | 0.118        | 13           | 0.624         | 42            | 0.313          |
| 10         | 201909_at | RPS4Y1    | 21          | 34          | 8232        | 0.013        | 3            | 0.913         | 5             | 0.736          |
| 11         | 214774_at | TOX3      | 34          | 0           | 95          | 0.262        | 49           | 0.492         | 102           | 0.210          |
| 12         | 211074_at | FOLR1     | 17          | 4           | 9482        | 0.011        | 926          | 0.124         | 213           | 0.159          |
| 13         | 210096_at | CYP4B1    | 6           | 44          | 8           | 0.797        | 65           | 0.431         | 54            | 0.284          |
| 14         | 204475_at | MMP1      | 27          | 0           | 34          | 0.406        | 24           | 0.564         | 21            | 0.465          |
| 15         | 214612_at | MAGEA     | 16          | 0           | 489         | 0.110        | 134          | 0.323         | 35            | 0.339          |

Table 3: Genetics example: the probes with top (scaled) leverage scores, derived from the first two singular vectors.

| LS (2) rank | index $q_j$ | LS (2) score | probe set | gene name | number sick | number well | number sick | number well | DEIM rank |
|-------------|-------------|--------------|-----------|-----------|-------------|-------------|-------------|-------------|-----------|
| 1           | 9565        | 1.000        | 210081_at | AGER      | 2           | 45          | 1           | 1           | 1         |
| 2           | 13766       | 0.922        | 214387_at | SFTPC     | 6           | 48          | —           | —           | —         |
| 3           | 11135       | 0.907        | 211735_at | SFTPC     | 5           | 48          | 73          | —           | —         |
| 4           | 9361        | 0.899        | 209875_at | SPP1      | 50          | 2           | —           | —           | —         |
| 5           | 5509        | 0.896        | 205982_at | SFTPC     | 5           | 48          | —           | —           | —         |
| 6           | 9103        | 0.835        | 209613_at | ADH1B     | 2           | 47          | —           | —           | —         |
| 7           | 14827       | 0.834        | 215454_at | SFTPC     | 0           | 46          | —           | —           | —         |
| 8           | 9580        | 0.797        | 210096_at | CYP4B1    | 6           | 44          | 13          | —           | —         |
| 9           | 4239        | 0.754        | 204712_at | WIFI      | 5           | 43          | 70          | —           | —         |
| 10          | 3507        | 0.724        | 203980_at | FABP4     | 2           | 44          | —           | —           | —         |
| 11          | 18594       | 0.717        | 219230_at | TMEM100   | 2           | 38          | —           | —           | —         |
| 12          | 9102        | 0.684        | 209612_at | ADH1B     | 2           | 46          | —           | —           | —         |
| 13          | 13514       | 0.626        | 214135_at | CLDN18    | 3           | 47          | —           | —           | —         |
| 14          | 5393        | 0.626        | 205866_at | FCN3      | 0           | 39          | —           | —           | —         |
| 15          | 4727        | 0.614        | 205200_at | CLEC3B    | 0           | 39          | —           | —           | —         |
References

[1] James Baglama and Lothar Reichel. An implicitly restarted block Lanczos bidiagonalization method using Leja shifts. *BIT*, 53:285–310, 2013.

[2] C. G. Baker, K. A. Gallivan, and P. Van Dooren. Low-rank incremental methods for computing dominant singular subspaces. *Linear Algebra Appl.*, 436:2866–2888, 2012.

[3] Christopher G. Baker. A block incremental algorithm for computing dominant singular subspaces. Master’s thesis, Florida State University, 2004.

[4] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen, and Anthony T. Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Acad. Sci. Paris, Ser. I*, 339:667–672, 2004.

[5] Christos Boutsidis and David P. Woodruff. Optimal CUR matrix decompositions. arXiv:1405.7910 [cs.DS], 2014.

[6] Saifon Chaturantabut and Danny C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.*, 32:2737–2764, 2010.

[7] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart. Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization. *Math. Comp.*, 30:772–795, 1976.

[8] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM J. Matrix Anal. Appl.*, pages 844–881, 2008.

[9] Evgeniy Gabrilovich and Shaul 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 (ICML)*, pages 321–328, 2004.

[10] Luc Giraud, Julien Langou, Miroslav Rozložník, and Jasper van den Eshof. Rounding error analysis of the classical Gram–Schmidt orthogonalization process. *Numer. Math.*, 101:87–100, 2005.

[11] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra Appl.*, 261:1–21, 1997.

[12] Ming Gu and Stanley C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM J. Sci. Comput.*, 17:848–869, 1996.

[13] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53:217–288, 2011.

[14] Michiel E. Hochstenbach. A Jacobi–Davidson type SVD method. *SIAM J. Sci. Comput.*, 23:606–628, 2001.

[15] Ilse C. F. Ipsen and Thomas Wentworth. Sensitivity of leverage scores. arXiv:1402.0957 [math.NA], 2014.
[16] Abhisek Kundu, Srinivas Nambirajan, and Petros Drineas. Identifying influential entries in a matrix. arXiv:1310.3556 [cs.nA], 2013.

[17] Rasmus Munk Larsen. PROPACK: Software for large and sparse SVD calculations. http://sun.stanford.edu/~rmunk/PROPACK/ 2005. Version 2.1.

[18] R. B. Lehoucq, D. C. Sorensen, and C. Yang. ARPACK Users’ Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods. SIAM, Philadelphia, 1998.

[19] Michael W. Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. Proc. Nat. Acad. Sci., 106:697–702, 2009.

[20] G. W. Stewart. Four algorithms for the efficient computation of truncated QR approximations to a sparse matrix. Numer. Math., 83:313–323, 1999.

[21] Daniel B. Szyld. The many proofs of an identity on the norm of oblique projections. Numer. Alg., 42:309–323, 2006.

[22] Christian Thurau, Kristian Kersting, and Christian Bauckhage. Deterministic CUR for improved large-scale data analysis: an empirical study. In Proceedings of the 2012 SIAM International Conference on Data Mining, pages 684–695, 2012.

[23] Shusen Wang and Zhihua Zhang. Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling. J. Machine Learning Res., 14:2729–2769, 2013.