Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH ARTICLE

# Volume-based subset selection

Alexander Osinsky

[1]Marchuk Institute of Numerical Mathematics of Russian Academy of Sciences, Moscow 119333, Russia

**Summary**

This paper provides a fast algorithm for the search of a dominant (locally maximum volume) submatrix, generalising the existing algorithms from $n \leqslant r$ to $n > r$ submatrix columns, where $r$ is the number of searched rows. We prove the bound on the number of steps of the algorithm, which allows it to outperform the existing subset selection algorithms in either the bounds on the norm of the pseudoinverse of the found submatrix, the bounds on the complexity or both.

**KEYWORDS:**
subset selection, locally maximum volume, optimal design, sparse approximation

## 1 | INTRODUCTION

The paper is dedicated to the problem of finding a "good" subset of $n$ columns in matrix $R$ with $r$ rows and $N$ columns (usually $r \ll N$). Namely, we search for a submatrix, which is far from being degenerate. The nondegeneracy of the submatrix $\hat{R} \in \mathbb{R}^{r \times n}$ is usually measured by the ratios $\left\|\hat{R}^{+}\right\|_2 / \|R^{+}\|_2$ and $\left\|\hat{R}^{+}\right\|_F / \|R^{+}\|_F$ [1, 2]. This problem arises in least squares problems[3] or when trying to find accurate low-rank approximations of a matrix[4]. In case $r = n$ an effective strategy is to search for a locally maximum volume submatrix[5, 6], where the volume of the submatrix $\hat{R}$ for $n \geqslant r$ can be defined as $\mathcal{V}(\hat{R}) = \sqrt{\det \hat{R}^T \hat{R}}$. Here we answer one of the questions posed by Avron and Boutsidis[2], namely, whether there exists a way to efficiently find a locally maximum volume submatrix with $n > r$ columns.

To be precise, we are searching for $c$-locally maximum volume submatrix with a factor $c$ close to 1.

**Definition 1.** A nondegenerate submatrix $\hat{R} \in \mathbb{R}^{r \times n}$ of matrix $R \in \mathbb{R}^{r \times N}$ is said to have $c$-locally maximum volume, if for any submatrix $\tilde{R} \in \mathbb{R}^{r \times n}$, which differs from $\hat{R}$ only in one column,

$$\mathcal{V}\left(\tilde{R}\right) \leqslant c\mathcal{V}\left(\hat{R}\right).$$

The reason for this definition is that it allows to limit the number of steps in the search for such a submatrix, while preserving all of the necessary estimates up to a factor $c$.

Ideas of maximizing rectangular submatrix volume first appeared in D-optimal designs[7, 8]. Greedy column addition was later rediscovered by Michalev and Oseledets[9] and later named rect-maxvol[10]. We rely on their ideas as a basis to construct an algorithm, which allows to find $c$-locally maximum volume submatrix. We prove the general bound on the number of steps, which also improves the bound on the number of steps for the known case $n = r$.

Our algorithm makes column exchanges in the submatrix in just $O(Nn)$ time. Moreover, we were able to limit the number of column exchanges before achieving $c$-locally maximum volume, thus proving a bound on the total complexity, which is $O\left(Nnr \log_c n\right)$. This bound also allows to reduce the known complexity bounds of locally maximum volume search algorithms with $r = n$ like Strong RRQR[5] and Rank-revealing Cholesky[11] factorizations by a factor $\frac{\log N}{\log r}$.

There exist several theorems[10, 12], which bound the error of cross matrix approximations by using locally maximum volume submatrices. It was suggested, that the algorithm rect-maxvol[10] should be used to find these submatrices. However, the use of rect-maxvol algorithm, which greedily adds new columns, does not guarantee that the resulting submatrix has locally maximum volume, nor that it has $c$-locally maximum volume, nor that its volume is in any way related to the maximum volume. On the other hand, the ability to reach $c$-locally maximum volume submatrices allows the new algorithm, which we call dominant, to guarantee the estimates, which could not be reached by rect-maxvol algorithm. In addition to the theorems from[10, 12], which require locally maximum volume submatrices, our algorithm can be also used to find large projective volume submatrices[12] more efficiently, which in practice leads to approximations arbitrarily close to SVD[13].

## 2 | KNOWN ALGORITHMIC BOUNDS FOR THE NORMS OF THE PSEUDOINVERSE

Let us start by looking at the algorithms, which are known to have some guarantees on the ratios $\|\hat{R}^{+}\|_2/\|R^{+}\|_2$ and $\|\hat{R}^{+}\|_F/\|R^{+}\|_F$. The following table is taken from[2]. We also include algorithm dominant for comparison.

**Table 1** Methods for finding a strongly nondegenerate rectangular submatrix $\hat{R} \in \mathbb{R}^{r\times n}$ in the rows $R \in \mathbb{R}^{r\times N}$.

| Method | $\|\hat{R}^{+}\|_F^2/\|R^{+}\|_F^2$ | $\|\hat{R}^{+}\|_2^2/\|R^{+}\|_2^2$ | Complexity |
|---|---|---|---|
| RRQR[5], $n = r$ | $(1 + c(N - r))\,\frac{r\|R^{+}\|_2}{\|R^{+}\|_F}$ | $1 + cr(N - r)$ | $O(Nr^2 \log N/\log c)$ |
| Theorem 3.7 ($\delta = 1/2$) from[2], $n \geqslant 32r\ln(4r)$ | $4N$ | $4N$ | $O(Nr^2 + n\log n)$ |
| Theorem 3.11 ($\delta = 1/2$) from[2], $n = r$ | $c(N - r + 1)$ | $cr(N - r + 1)$ | $O(Nr^3/\log c)$ |
| Theorem 3.5 from[2], $n > r$ | $\frac{(1+\sqrt{\frac{N}{n}})^2}{(1-\sqrt{\frac{r}{n}})^2}$ | $\frac{(1+\sqrt{\frac{N}{n}})^2}{(1-\sqrt{\frac{r}{n}})^2}$ | $O(Nnr^2)$ |
| Theorem 3.1 from[2], $n \geqslant r$ | $\frac{N-r+1}{n-r+1}$ | $r\frac{N-r+1}{n-r+1}$ | $O(Nr^2 + N(N - n)r)$ |
| Corr. 3.3 from[2], $n \geqslant r$ | $\frac{N-r+1}{n-r+1} \cdot \frac{r\|R^{+}\|_2}{\|R^{+}\|_F}$ | $1 + r\frac{N-n}{n-r+1}$ | $O(Nr^2 + N(N - n)r)$ |
| NEW ALGORITHM | | | |
| dominant, $n \geqslant r$ | $\left(\frac{N-r+1}{n-r+1} + \frac{(c-1)n(N-n)}{r(n-r+1)}\right)\frac{r\|R^{+}\|_2^2}{\|R^{+}\|_F^2}$ | $1 + \frac{r+(c-1)n}{n-r+1}\,(N - n)$ | $O(Nnr\log n/\log c)$ |

The first algorithm in this table (strong rank-revealing QR) is based on the idea of finding locally maximum volume (aka dominant) submatrix. Here we improve the complexity bound and generalise this algorithm for the case $n \geqslant r$. Note that apart from the second row, which selects too many columns, dominant has the lowest complexity among all of the subset selection algorithms.

In terms of the optimality of the upper bounds, the following proposition can help.

**Proposition 1.**

$$\sup_{R\in\mathbb{R}^{r\times N}} \min_{\hat{R}\in\mathbb{R}^{r\times n}} \frac{\left\|\hat{R}^{+}\right\|_2^2}{\|R^{+}\|_2^2} \geqslant \sup_{R\in\mathbb{R}^{r\times N}} \min_{\hat{R}\in\mathbb{R}^{r\times n}} \frac{\left\|\hat{R}^{+}\right\|_F^2}{\|R^{+}\|_F^2} = \frac{N - r + 1}{n - r + 1}.$$

*Proof.* The bound

$$\sup_{R\in\mathbb{R}^{r\times N}} \min_{\hat{R}\in\mathbb{R}^{r\times n}} \frac{\left\|\hat{R}^{+}\right\|_F^2}{\|R^{+}\|_F^2} \leqslant \frac{N - r + 1}{n - r + 1}$$

is provided by the fifth row of table 1 (theorem 3.1 from[2]). We now provide an example to prove the inequalities in the opposite direction.

Consider the following matrix $R \in \mathbb{R}^{r \times N}$:

$$R = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & 0 & 1 & 0 & \cdots & \cdots & 0 \\ 0 & \cdots & 0 & \varepsilon & \cdots & \cdots & \varepsilon \end{bmatrix}$$

with $r-1$ ones and $N-r+1$ $\varepsilon$-s. Then, with $\varepsilon < 1/\sqrt{N-r+1}$, we have $\|R^+\|_2^2 = \varepsilon^{-2}/(N-r+1)$ and $\|R^+\|_F^2 = \varepsilon^{-2}/(N-r+1)+r-1$.

All non-degenerate submatrices $\hat{R} \in \mathbb{R}^{r \times n}$ of matrix $R$ are equal up to column permutations and their pseudoinverse is

$$\hat{R}^+ = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & \vdots \\ \vdots & 0 & 1 & 0 \\ 0 & \ddots & 0 & \frac{\varepsilon^{-1}}{n-r+1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \frac{\varepsilon^{-1}}{n-r+1} \end{bmatrix}.$$

Then

$$\frac{\left\|\hat{R}^+\right\|_F^2}{\|R^+\|_F^2} = \frac{\varepsilon^{-2}/(n-r+1)+r-1}{\varepsilon^{-2}/(N-r+1)+r-1} \xrightarrow[\varepsilon \to 0]{} \frac{N-r+1}{n-r+1}$$

and

$$\frac{\left\|\hat{R}^+\right\|_2^2}{\|R^+\|_2^2} = \frac{\varepsilon^{-2}/(n-r+1)}{\varepsilon^{-2}/(N-r+1)} = \frac{N-r+1}{n-r+1}.$$

□

Although the upper bounds from table 1 can be far from the lower bound of proposition 1, as will be shown later, ratios of norms for the locally maximum volume submatrices rarely exceed even the lower bound of $\frac{N-r+1}{n-r+1}$.

Next, we provide an efficient algorithm to search for locally maximum volume submatrices and then show how close its output is to the bounds of proposition 1.

## 3 | LOCALLY MAXIMUM VOLUME SEARCH

First, let us remind the reader of a standard rank-revealing QR decomposition, called QR with column pivoting (for a detailed description, see [14], page 278). The version, which does not require computing QR and only selects the columns is discussed in [15]. It can be written as follows:

**Algorithm 1** Column pivoting

**Require:** Matrix $R \in \mathbb{R}^{r \times N}$, required rank $r$.
**Ensure:** Set of row indices $\mathcal{I}$ of cardinality $r$, containing a submatrix with large volume.

1: $\mathcal{I} := \emptyset$
2: **for** $k := 1$ to $r$ **do**
3:     **for** $i := 1$ to $N$ **do**
4:         $\gamma_i := \|R_{:,i}\|_2^2$
5:     **end for**
6:     Select $j$, corresponding to $\max_j \gamma_j$
7:     $\mathcal{I} := \mathcal{I} \cup \{j\}$
8:     $R := R - \frac{1}{\gamma_i} R_{:,i} R_{:,i}^T R$
9: **end for**

Here we denote by $R_{:,i}$ the $i$-th column of matrix $R$. Similarly, we will denote by $R_{i,:}$ the $i$-th row of matrix $R$.

At each step the column pivoting algorithm adds the column, which has the maximum length in the subspace, orthogonal to the already selected columns. Column pivoting has one useful property: the volume of the output submatrix can't be too far from the maximum.

Let us denote by $R_{\mathcal{I}}$ the submatrix, defined by the (ordered) set of columns $\mathcal{I}$. The following result was proven in [15].

**Proposition 2** ([15])**.** Column pivoting produces a submatrix $R_{\mathcal{I}}$ with the volume not more than $r!$ times smaller than the maximum:

$$\mathcal{V}\left(R_{\mathcal{I}}\right) \geqslant \frac{1}{r!} \max_{\mathcal{J},|\mathcal{J}|=r} \mathcal{V}\left(R_{\mathcal{J}}\right).$$

From this proposition we can infer the bound on the number of steps of strong rank revealing QR (RRQR)[5], see also first row of table 1. Strong RRQR algorithm iteratively replaces the columns of $R_{\mathcal{I}}$ one-by-one. The replacement is chosen so as to maximize the volume of the new submatrix. For the case $r = n$ this algorithm is completely equivalent to the maxvol algorithm[16], see algorithm 2.

---

**Algorithm 2** maxvol[16]

---

**Require:** Matrix $R \in \mathbb{R}^{r\times N}$, starting set of column indices $\mathcal{I}$ of cardinality $r$. Bound $c$.
**Ensure:** The row and column indices of the $c$-locally maximum volume submatrix is written in $\mathcal{I}$.

1: $C := R_{\mathcal{I}}^{-1} R$
2: Select indexes $i$ and $j$ corresponding to $\max_{i,j} \left|C_{i,j}\right|_2$
3: **while** $\left|C_{i,j}\right| > c$ **do**
4: Replace index $i$ with $j$ in the set $\mathcal{I}$
5: Update $C$ correspondingly
6: Select indexes $i$ and $j$ corresponding to $\max_{i,j} \left|C_{i,j}\right|_2$
7: **end while**

---

The idea of the algorithm is that the search for the locally maximum volume in $R$ can be replaced by the search in $C = R_{\mathcal{I}}^{-1} R$, since multiplication by $R_{\mathcal{I}}^{-1}$ changes the volumes of all submatrices by the same factor $\mathcal{V}\left(R_{\mathcal{I}}^{-1}\right)$. On the other hand, current submatrix in $C$ is always equal to identity matrix $C_{\mathcal{I}} = R_{\mathcal{I}}^{-1} R_{\mathcal{I}} = I$, so its volume (absolute value of the determinant) after replacement of the $i$-th column by $C_{:,j}$ is just $\left|C_{i,j}\right|$:

$$\begin{vmatrix} 1 & 0 & C_{1j} & 0 & 0 \\ 0 & 1 & \vdots & \vdots & \vdots \\ \vdots & 0 & C_{ij} & 0 & \vdots \\ \vdots & \vdots & \vdots & 1 & 0 \\ 0 & 0 & C_{rj} & 0 & 1 \end{vmatrix} = C_{ij}.$$

By construction, at each step of the algorithm the volume is increased by a factor $\left|C_{i,j}\right| > c$. Consequently, if the algorithm starts from a submatrix, obtained from column pivoting, the number of steps $s$ is bounded by

$$s \leqslant \log_c r! \leqslant \log_c r^r = r \log_c r,$$

because the starting volume was at most $r!$ smaller than the maximum due to proposition 2.

Update of $C$ in maxvol as well as in RRQR[5] requires $O(Nr)$ operations. Thus, for $c \leqslant r$ the total complexity is $O\left(Nr^2 \log_c r\right)$, which is lower than $O\left(Nr^2 \log_c N\right)$ bound, proven in[5]. The same bound on the number of steps is also applicable to other locally maximum volume search algorithms, e.g., strong rank revealing Cholesky factorization[11].

We now need to generalise the algorithm to the case of $n > r$ columns. To do that, we use a generalisation of the column pivoting strategy, which is called rect-maxvol[10]. It is based on the following lemmas, which we will use to construct our column replacement algorithm. The first one describes the volume change, when the submatrix $\hat{R}$ is extended by a new column.

**Lemma 1** ([10])**.** Let $\hat{R} \in \mathbb{R}^{r\times n}$, $n \geqslant r$, be a submatrix of the matrix $R \in \mathbb{R}^{r\times N}$.

Then, for the submatrix $\widehat{R} \in \mathbb{R}^{r\times(n+1)}$, obtained from $\hat{R}$ by appending the $j$-th column of $R$

$$\frac{\mathcal{V}^2\left(\widehat{R}\right)}{\mathcal{V}^2\left(\hat{R}\right)} = 1 + \left\|C_{:,j}\right\|_2^2,$$

where $C = \hat{R}^+ R$.

The next lemma describes the update of the matrix $C$, when the matrix $\hat{R}$ is extended to $\widehat{R}$ by a new column.

**Lemma 2** ([10])**.** Let $\hat{R} \in \mathbb{R}^{r\times n}$, $n \geqslant r$, be a submatrix of the matrix $R \in \mathbb{R}^{r\times N}$. Let $C = \hat{R}^+ R \in \mathbb{R}^{n\times N}$.
Then, for the matrix $\widehat{R} \in \mathbb{R}^{r\times(n+1)}$, obtained from $\hat{R}$ by appending the $j$-th column of $R$ and $\tilde{C} = \widehat{R}^+ R \in \mathbb{R}^{(n+1)\times N}$,

$$\tilde{C} = \begin{bmatrix} C - \frac{1}{1+\|C_{:,j}\|_2^2} C_{:,j} C_{:,j}^T C \\ \frac{1}{1+\|C_{:,j}\|_2^2} C_{:,j}^T C \end{bmatrix}. \tag{1}$$

Moreover, for any $k$ the new squared length of the $k$-th column $\tilde{C}_{:,k}$ is

$$\|\tilde{C}_{:,k}\|_2^2 = \|C_{:,k}\|_2^2 - \frac{1}{1 + \left\|C_{:,j}\right\|_2^2} \left|C_{:,k}^T C_{:,j}\right|^2. \tag{2}$$

Note that, unlike column pivoting, there is no analogue of proposition 2 for rect-maxvol algorithm. Nevertheless, proposition 2 already gives us a good enough starting submatrix, and we do not need to use rect-maxvol to select more columns. Instead, we are going to arbitrarily extend $r \times r$ submatrix to $n \times r$ and exchange its columns one-by-one, like in maxvol algorithm. Lemmas 1 and 2 can be used to prove the following update criterion.

**Lemma 3.** Let $\hat{R} \in \mathbb{R}^{r\times n}$ be a submatrix in the first $n$ rows of the matrix $R \in \mathbb{R}^{r\times N}$. Then, replacing $i$-th column of $\hat{R}$ by the $j$-th column of $R$ (for $i > n$) changes the squared volume of $\hat{R}$ by a factor

$$B_{ij} = |C_{ij}|^2 + \left(1 + \left\|C_{:,j}\right\|_2^2\right)\left(1 - \|C_{:,i}\|_2^2\right),$$

where $C = \hat{R}^+ R$.

Thus, when $\max_{ij} B_{ij} > c$, we can increase the volume by at least a factor of $\sqrt{c}$, and when $\max_{ij} B_{ij} \leqslant c$ is reached, then, by definition 1, we have a $\sqrt{c}$-locally maximum volume submatrix.

*Proof.* To prove the lemma, we need to derive the fast update of matrix $C$.
The update is done in 2 steps. First, we append the $j$-th column, and then remove the $i$-th column. We will denote the corresponding submatrices $\hat{R} \in \mathbb{R}^{r\times n}$ (initial), $\widehat{R} \in \mathbb{R}^{r\times(n+1)}$ (with the $j$-th column) and $\hat{R}' \in \mathbb{R}^{r\times n}$ (with $j$-th, but without $i$-th column). They are used to form $C = \hat{R}^+ R \in \mathbb{R}^{n\times N}$, $\tilde{C} = \widehat{R}^+ R \in \mathbb{R}^{(n+1)\times N}$ and, finally, $C' = \hat{R}'^+ R \in \mathbb{R}^{n\times N}$.
From (2) the squared length of $j$-th column changes as

$$\left\|\tilde{C}_{:,j}\right\|_2^2 = \left\|C_{:,j}\right\|_2^2 - \frac{\left\|C_{:,j}\right\|_2^4}{1 + \left\|C_{:,j}\right\|_2^2} = \frac{\left\|C_{:,j}\right\|_2^2}{1 + \left\|C_{:,j}\right\|_2^2}.$$

The same can be said when we remove the $i$-th column. To do that, we replace $i$ by $j$ and $C$ by $C'$:

$$\|\tilde{C}_{:,i}\|_2^2 = \frac{\left\|C'_{:,i}\right\|_2^2}{1 + \left\|C'_{:,i}\right\|_2^2}.$$

Rearranging, we get the following:

$$\frac{1}{1 + \left\|C'_{:,i}\right\|_2^2} = 1 - \|\tilde{C}_{:,i}\|_2. \tag{3}$$

We can calculate the length of the $i$-th column of $\tilde{C}$ from $C$ using (2):

$$\|\tilde{C}_{:,i}\|_2^2 = \|C_{:,i}\|_2^2 - \frac{1}{1 + \left\|C_{:,j}\right\|_2^2} \left|C_{:,i}^T C_{:,j}\right|^2.$$

Now we note that $C_{:,i}^T C_{:,j} = C_{ij}$:

$$C_{:,i}^T C_{:,j} = R_{:,i}^T \left(\hat{R}^+\right)^T \hat{R}^+ R_{:,j} = R_{:,i}^T \left(\hat{R}\hat{R}^T\right)^{-1} R_{:,j} = R_{:,i}^+ R_{:,j} = C_{ij},$$

so

$$\|\tilde{C}_{:,i}\|_2^2 = \|C_{:,i}\|_2^2 - \frac{1}{1+\left\|C_{:,j}\right\|_2^2} \left|C_{ij}\right|^2. \tag{4}$$

Using lemma 1 twice we get the total (squared) ratio of volumes:

$$B_{ij} = \frac{\mathcal{V}^2\left(\hat{R}'\right)}{\mathcal{V}^2\left(\hat{R}\right)} \cdot \frac{\mathcal{V}^2\left(\hat{R}\right)}{\mathcal{V}^2\left(\hat{R}\right)} = \left(1+\left\|C_{:j}\right\|_2^2\right) / \left(1+\|C'_{:i}\|_2^2\right) = \left(1+\left\|C_{:j}\right\|_2^2\right)\left(1-\|\tilde{C}_{:i}\|_2^2\right),$$

where the last equality is due to (3). We substitute $\left\|\tilde{C}_{:,i}\right\|_2^2$ from (4) to finally get

$$B_{ij} = \left(1+\left\|C_{:,j}\right\|_2^2\right)\left(1-\|C_{:,i}\|_2^2\right) + |C_{ij}|^2.$$

□

The use of the matrix $B_{ij}$ for the decision to replace the columns leads us to the following algorithm. We call this algorithm dominant.

---

**Algorithm 3** dominant

---

**Require:** Matrix $R \in \mathbb{R}^{M\times r}$, the starting set of row indices $\mathcal{I}$ of cardinality $n$. For example, $\mathcal{I} = \{1, ..., n\}$. Bound $c$.
**Ensure:** The updated set $\mathcal{I}$, corresponding to a dominant submatrix.

$C := R_{\mathcal{I}}^+ R$
**for** $i := 1$ to $n$ **do**
  **for** $j := n+1$ to $M$ **do**
    $B_{i,j} := \left(1+\|C_{:,j}\|_2^2\right)\left(1-\|C_{:,i}\|_2^2\right) + |C_{i,j}|^2$
  **end for**
**end for**
$\{i,j\} := \arg\max_{i,j} B_{i,j}$
**while** $B_{i,j} > c$ **do**
  Update $C$ and $B$
  Replace $j$ with $i$ in $\mathcal{I}$
  $\{i,j\} := \arg\max_{i,j} B_{i,j}$
**end while**

---

To construct the initial matrix $C$, $O(Nnr)$ operations are required. Next we show that matrix $C$ can be updated in $O(Nn)$ operations. Then for $s$ steps the total complexity is $O(Nnr + Nns)$.

Fast updates from $C$ to $\tilde{C}$ are already provided by lemma 2. Now, let us focus on column removal.

Let us denote by $\bar{C} \in \mathbb{R}^{n\times N}$ the first $n$ columns of $\tilde{C}$ (without $n$+1-st column). Then, according to (1), appending the $j$-th column leads to the following update of the $j$-th column itself:

$$\bar{C}_{:,j} = C_{:,j} - \frac{1}{1+\left\|C_{:,j}\right\|_2^2} C_{:,j} C_{:,j}^T C_{:,j} = \frac{1}{1+\left\|C_{:,j}\right\|_2^2} C_{:,j}. \tag{5}$$

For $\bar{C}_{:,i}$ we get from (5) by replacing $i$ by $j$ and $C$ by $C'$:

$$\bar{C}_{:,i} = \frac{1}{1+\left\|C'_{:,i}\right\|_2^2} C'_{:,i} = C'_{:,i}\left(1-\|\tilde{C}_{:,i}\|_2^2\right). \tag{6}$$

Interpreting $\tilde{C}$ as being obtained from $C'$ by adding $i$-th column to $\hat{R}'$, we get from (1):

$$\bar{C} = C' - \frac{1}{1 + \left\|C'_{:,i}\right\|_2^2} C'_{:,i} C'^T_{:,i} C'. \tag{7}$$

We can now substitute $C'_{:,i}$ from (6) and $1 + \|\tilde{C}_{:,i}\|_2^2$ from (3) to express $C'$ in terms of $\bar{C}$ in (7):

$$C' = \bar{C} + \frac{1}{1 - \|\tilde{C}_{:,i}\|_2^2} \bar{C}_{:,i} \cdot \frac{1}{1 + \left\|C'_{:,i}\right\|_2^2} C'^T_{:,i} C'. \tag{8}$$

According to lemma 2, the value $\frac{1}{1+\left\|C'_{:,i}\right\|_2^2} C'^T_{:,i} C'$ is the last row of $\tilde{C}$, so we have enough information to find $C'$ from $\tilde{C}$, and equation (8) clearly involves $O(Nn)$ operations. Pseudocode with the updates can be found in Appendix A.

Next, we prove the bound on the number of steps.

**Proposition 3.** Let the initial set of $n$ columns $\mathcal{I}$ contain the subset of $r$ columns $\mathcal{I}'$, obtained with column pivoting. Then the while loop in algorithm 3 performs

$$s \leqslant 2r \log_c n$$

steps.

*Proof.* By Bine's theorem $\mathcal{V}^2\left(R_\mathcal{I}\right) = \det\left(R_\mathcal{I} R_\mathcal{I}^T\right)$ can be expressed in terms of determinants of its $r \times r$ submatrices as follows:

$$\mathcal{V}\left(R_\mathcal{I}\right) = \sqrt{\sum_{\mathcal{J}' \subset \mathcal{I}, |\mathcal{J}'|=r} \left|\det R_{\mathcal{J}'}\right|^2},$$

so, since one of the submatrices is $R_{\mathcal{I}'}$,

$$\mathcal{V}\left(R_\mathcal{I}\right) \geqslant \sqrt{\left|\det R_{\mathcal{I}'}\right|^2} = \mathcal{V}\left(R_{\mathcal{I}'}\right). \tag{9}$$

On the other hand, Bine's theorem also gives us an upper bound for the maximum volume of all $n \times r$ submatrices. When applied to arbitrary submatrix $R_\mathcal{J}$,

$$\mathcal{V}\left(R_\mathcal{J}\right) = \sqrt{\sum_{\mathcal{J}' \subset \mathcal{J}, |\mathcal{J}'|=r} \left|\det R_{\mathcal{J}'}\right|^2}.$$

Since there are $C_n^r$ submatrices of size $r \times r$,

$$\max_{\mathcal{J}, |\mathcal{J}|=n} \mathcal{V}\left(R_\mathcal{J}\right) \leqslant \sqrt{\max_{\mathcal{J}', |\mathcal{J}'|=r} C_n^r \left|\det R_{\mathcal{J}'}\right|^2} \leqslant \sqrt{C_n^r} \max_{\mathcal{J}', |\mathcal{J}'|=r} \mathcal{V}\left(R_{\mathcal{J}'}\right). \tag{10}$$

Using proposition 2 together with (9) and (10), we obtain

$$\mathcal{V}\left(R_\mathcal{I}\right) \geqslant \mathcal{V}\left(R_{\mathcal{I}'}\right) \geqslant \frac{1}{\sqrt{C_n^r} r!} \max_{\mathcal{J}, |\mathcal{J}|=n} \mathcal{V}\left(R_\mathcal{J}\right).$$

Since `dominant` increases the volume at least $\sqrt{c}$ times after each step, the number of steps $s$ is bounded by

$$s \leqslant \log_{\sqrt{c}}\left(\sqrt{C_n^r} r!\right) \leqslant \log_{\sqrt{c}}\left(\sqrt{nr}\right)^r \leqslant \log_{\sqrt{c}} n^r = 2r \log_c n.$$

□

So, if $c \leqslant n$, the bound on the exchanges dominates the total complexity, which is then $O\left(Nnr \log_c n\right)$. After that $B_{ij} \leqslant c$, so by definition 1 we obtain $\sqrt{c}$-locally maximum volume submatrix.

Finally, we bound the norms of $R_\mathcal{I}^+$ using the following theorem.

**Theorem 1.** For any $c$-locally maximum volume submatrix $\hat{R} \in \mathbb{R}^{r \times n}$ of matrix $R \in \mathbb{R}^{r \times N}$

$$\left\|\hat{R}^+ R\right\|_F^2 \leqslant r + \frac{r + (c^2 - 1)n}{n - r + 1} (N - n).$$

*Proof.* Let us select an arbitrary column $j$. Since for $c$-locally maximum volume submatrices $B_{ij} \leqslant c^2$ for any $i$,

$$\begin{aligned} nc^2 \geqslant \sum_{i=1}^{n} B_{ij} &= \left\|C_{:,j}\right\|_2^2 + \left(1 + \left\|C_{:,j}\right\|_2^2\right)\left(n - \sum_{i=1}^{n} \|C_{:,i}\|_2^2\right) \\ &= \left\|C_{:,j}\right\|_2^2 + \left(1 + \left\|C_{:,j}\right\|_2^2\right)\left(n - \left\|\hat{R}^+\hat{R}\right\|_F^2\right) \\ &= (n - r + 1)\left\|C_{:,j}\right\|_2^2 + (n - r), \end{aligned}$$

so, rearranging and using the fact that $j$ is arbitrary,

$$\max_{j>n} \left\|C_{:,j}\right\|_2^2 \leqslant \frac{r + (c^2 - 1)n}{n - r + 1}.$$

Now, expressing $\left\|\hat{R}^+ R\right\|_F^2$ as the sum of the squared norms of its columns, we get

$$\left\|\hat{R}^+ R\right\|_F^2 \leqslant \left\|\hat{R}^+\hat{R}\right\|_F^2 + \sum_{j=n+1}^{N} \left\|\hat{R}^+ R_{:,j}\right\|_2^2 = r + \sum_{j=n+1}^{N} \left\|C_{:,j}\right\|_2^2 \leqslant r + \frac{r + (c^2 - 1)n}{n - r + 1}(N - n).$$

□

**Corollary 1.** Output submatrix $\hat{R} \in \mathbb{R}^{r\times n}$ from algorithm 3 has the following properties:

$$\begin{aligned} \frac{\left\|\hat{R}^+\right\|_F^2}{\|R^+\|_F^2} &\leqslant \left(1 + \frac{1 + (c-1)n/r}{n - r + 1}(N - n)\right)\frac{r\,\|R^+\|_2^2}{\|R^+\|_F^2}, \\ \frac{\left\|\hat{R}^+\right\|_2^2}{\|R^+\|_2^2} &\leqslant 1 + \frac{r + (c-1)n}{n - r + 1}(N - n). \end{aligned} \tag{11}$$

*Proof.*

$$\left\|\hat{R}^+ R\right\|_F^2 \geqslant \left\|\hat{R}^+\right\|_F^2 \sigma_r^2(R) = \left\|\hat{R}^+\right\|_F^2 / \|R^+\|_2^2, \tag{12}$$

where $\sigma_r(R)$ is the $r$-th (minimum) singular value of $R$. The first inequality in (11) follows from (12) and the fact that $\hat{R}$ has $\sqrt{c}$-locally maximum volume, so, according to theorem 1,

$$\left\|\hat{R}^+ R\right\|_F^2 \leqslant r + \frac{r + (c-1)n}{n - r + 1}(N - n).$$

Since singular values of $\hat{R}^+ R$ are not smaller than singular values of its submatrix $\hat{R}^+\hat{R}$, we conclude that the first $r$ singular values are at least 1. Then the sum of squares of the first $r$ singular values gives us the squared Frobenius norm:

$$\left\|\hat{R}^+ R\right\|_F^2 = \sum_{k=1}^{r} \sigma_k^2\left(\hat{R}^+ R\right) \geqslant \left\|\hat{R}^+ R\right\|_2^2 + r - 1.$$

Consequently,

$$\left\|\hat{R}^+\right\|_2^2 / \|R^+\|_2^2 = \left\|\hat{R}^+\right\|_2^2 \sigma_r^2(R) \leqslant \left\|\hat{R}^+ R\right\|_2^2 \leqslant 1 + \frac{r + (c-1)n}{n - r + 1}(N - n),$$

which proves the second inequality. □

## 4 | RESULTS

We put our bounds in table 2, adding a pair of specific cases.

Note, that the results depend on the constant $c$, which determines how far we want to be from the locally maximum volume. In practice, however, $c = 1$ is often reached in just $n$ exchanges, as illustrated in table 3.

As a test case we consider $R$ to be chosen from RANDSVD ensemble: its singular values are fixed and the singular vectors are random unitary matrices with the Haar measure. Note that the performance of the algorithms maxvol and dominant depends only on right singular vectors $V \in \mathbb{R}^{r\times N}$ of truncated SVD $R = U\Sigma V$, since

$$C = \hat{R}^+ R = \left(U\Sigma\hat{V}\right)^+ U\Sigma V = \left(\hat{V}\right)^+ V.$$

**Table 2** Bounds on the norms of the submatrix and complexity for locally maximum volume search algorithms in the rows $R \in \mathbb{R}^{r \times N}$. It is assumed that the starting submatrix is obtained from column pivoting.

| Method | $\|\hat{R}^+\|_F^2 / \|R^+\|_F^2$ | $\|\hat{R}^+\|_2^2 / \|R^+\|_2^2$ | Complexity |
|---|---|---|---|
| maxvol[16], $n = r$ | $(1 + c(N-r)) \frac{r\|R^+\|_2^2}{\|R^+\|_F^2}$ | $1 + cr(N-r)$ | $O(Nr^2 \log_c r)$ |
| dominant, $n = 2r-1, c = 1 + \frac{r}{n}$ | $\left(\frac{2N-3r+2}{r}\right) \frac{r\|R^+\|_2^2}{\|R^+\|_F^2}$ | $2N - 4r + 3$ | $O(Nr^2 \log r)$ |
| dominant, $n \geqslant r$ | $\left(\frac{N-r+1}{n-r+1} + \frac{(c-1)n(N-n)}{n-r+1}\right) \frac{r\|R^+\|_2^2}{\|R^+\|_F^2}$ | $1 + \frac{r+(c-1)n}{r(n-r+1)}(N-n)$ | $O(Nnr \log_c n)$ |

**Table 3** Number of steps of maxvol and dominant for $c = 1$. Average and maximum is among 100 random generations of $V \in \mathbb{R}^{r \times N}$, $VV^T = I$. $r = 50$, $N = 5000$.

| Algorithm | maxvol ($n = r = 50$) | dominant ($n = 100$) | dominant ($n = 500$) |
|---|---|---|---|
| Steps, average | 1.2 | 81 | 437 |
| Steps, maximum | 7 | 99 | 457 |

According to table 3, there are usually only a few steps needed to be made for $n = r$ and at most $n$ steps needed in case $n > r$. While with more trials it may be possible to find an example, when more than $n$ steps are needed, it should be rare. To be sure, we will limit ourselves by $2n$ swaps in all further experiments, which leads to the total complexity $O\left(Nn^2\right)$.

Let us check our results on a few special cases. According to our estimates, the worst case scenario is $\frac{\|R^+\|_2^2}{\|R^+\|_F^2} \approx 1$, i. e., there is only one small singular value; and the best-case scenario is $\frac{\|R^+\|_2^2}{\|R^+\|_F^2} = 1/r$, i. e., when all singular values are equal. So, these cases would be the most interesting to us.

According to proposition 1, there is a lower bound of $\frac{\|\hat{R}^+\|_{2,F}}{\|R^+\|_{2,F}} \geqslant \frac{N-r+1}{n-r+1}$ for both spectral and Frobenius norm in the worst case. A reasonable question is whether the same bound can be reached in practice by locally maximum volume, since its worst upper bound is about $r$ times higher. To answer this question, we compare average and worst-case performance of dominant. We look at 2 special cases, present in table 2: $n = r$ and $n = 2r - 1$.

To test the methods for the case of equal singular values we set $R = V$, where $V \in \mathbb{R}^{r \times N}$ is a random (in terms of Haar measure) matrix with orthonormal rows. To get any desired distribution of singular values we can then choose $R = \Sigma V$, where $\Sigma$ is a diagonal matrix of singular values. Note that we do not need to multiply by left singular vectors, since they neither affect the algorithm (since the volumes of the submatrices are not affected by multiplication by an unitary matrix) nor the norms of the submatrices or the norm of the matrix $R$ itself.

So, we have 2 cases:

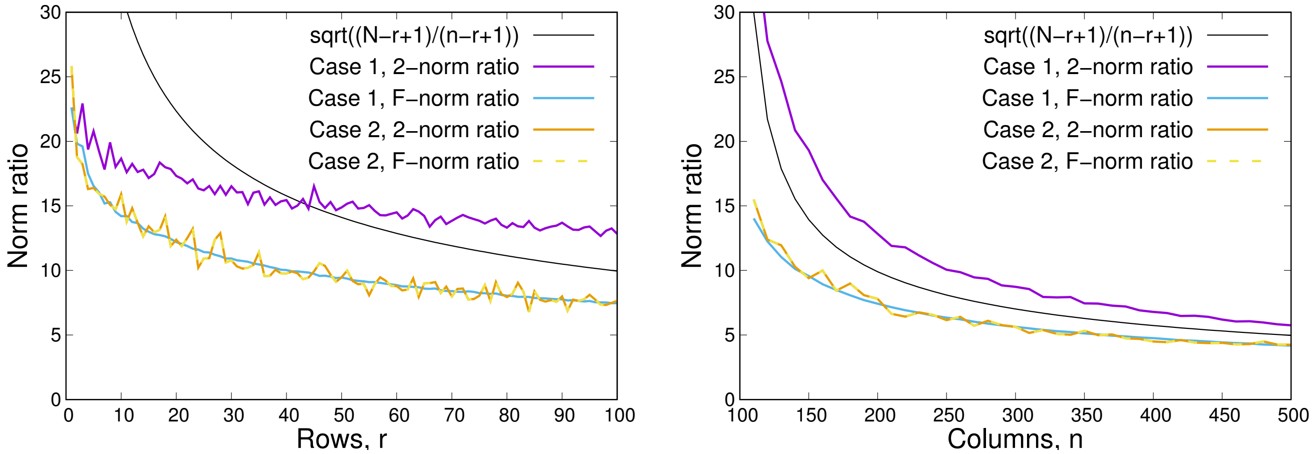
$$\begin{aligned} \text{Case 1}: \quad & \Sigma = I, \\ \text{Case 2}: \quad & \Sigma = \operatorname{diag}\left(1, \ldots, 1, 10^{-10}\right). \end{aligned}$$

As a test case let us use $r = 100$ and $N = 10099$. With such a choice, for $n = r$ we expect to see $\frac{\|\hat{R}^+\|_{2,F}}{\|R^+\|_{2,F}} \sim \sqrt{\frac{N-r+1}{n-r+1}} = 100$ and for $n = 2r - 1$ we expect to see $\frac{\|\hat{R}^+\|_{2,F}}{\|R^+\|_{2,F}} \sim \sqrt{\frac{N-r+1}{n-r+1}} = 10$. On the other hand, bounds from table 2 predict in the worst case (but assuming $2n$ operations is enough to reach dominant submatrix) the ratios to be about 10 times higher for spectral norm in both cases and also 10 times higher for Frobenius norm in Case 2. As expected, for Case 2 the ratios for the spectral and Frobenius norms are almost the same, since $R^+$ is almost of rank 1.

As we can see, the upper bound from table 2, where the ratio is about $\sqrt{Nr} \approx 1000$, never happens, and in practice all ratios are of the same order as the lower bound $\sqrt{\frac{N-n+1}{n-r+1}}$. Case 1 with spectral norm proved to be the worst as the ratio is about 30%

**Table 4** Performance of maxvol and dominant. Average and maximum is among 1000 random generations of $V$. $r = 100$, $N = 10099$.

| Columns | $n = r$ | $n = 2r - 1$ |
|---|---|---|
| $\|\hat{R}^+\|_F / \|R^+\|_F$, Case 1, average | 18.4 | 7.48 |
| $\|\hat{R}^+\|_F / \|R^+\|_F$, Case 1, maximum | 19.1 | 7.52 |
| $\|\hat{R}^+\|_F / \|R^+\|_F$, Case 2, average | 25.7 | 7.56 |
| $\|\hat{R}^+\|_F / \|R^+\|_F$, Case 2, maximum | 38.3 | 8.39 |
| $\|\hat{R}^+\|_2 / \|R^+\|_2$, Case 1, average | 64.9 | 13.0 |
| $\|\hat{R}^+\|_2 / \|R^+\|_2$, Case 1, maximum | 77.7 | 13.6 |
| $\|\hat{R}^+\|_2 / \|R^+\|_2$, Case 2, average | 25.7 | 7.56 |
| $\|\hat{R}^+\|_2 / \|R^+\|_2$, Case 2, maximum | 38.3 | 8.39 |
| $\sqrt{\frac{N-r+1}{n-r+1}}$ | 100 | 10 |




(a) $n = 2r - 1$, $r$ from 1 to 100.

(b) $r = 100$, $n$ from 100 to 500.

**Figure 1** Performance of dominant for $R = \Sigma V$ with random $V \in \mathbb{R}^{r\times N}$, $N = 10000$.

higher than $\sqrt{\frac{N-n+1}{n-r+1}} = 10$. However, the ratio $\|\hat{R}^+\|_2/\|R^+\|_2 \approx 13.6$ is still about twice as low as the best possible upper bound for other algorithms from table 1, namely $\frac{1+\sqrt{N/n}}{1-\sqrt{r/n}} \approx 27.7$.

As another illustration, figure 1 (a) shows how spectral and Frobenius norms are affected by $r$ and figure 1 shows how they change, when $n$ is increased.

## 5 | CONCLUSION

We have presented algorithm dominant, which generalises the search for locally maximum volume submatrix to rectangular submatrices with $n > r$ columns. The suggested algorithm has $O\left(Nnr \log_c n\right)$ complexity, which is similar to the complexity of $O\left(Nr^2 \log_c r\right)$ for the case $n = r$. We have also proven the bounds on the norms of the found submatrix and shown that in practice the bounds are $\frac{\|\hat{R}^+\|_{2,F}}{\|R^+\|_{2,F}} \sim \sqrt{\frac{N-r+1}{n-r+1}}$. Since its actual performance is better than all upper bounds from [2], new algorithm is a perfect choice for subset selection, since it allows to find significantly nondegenerate submatrix much faster than the alternatives.

□

## APPENDIX

## A PSEUDOCODE FOR DOMINANT ALGORITHM

Here we present the pseudocode for the dominant algorithm with fast updates. The set of indexes is stored as the first $n$ elements of the permutation of all $N$ columns. The permutation is updated at the same time as there are column swaps in matrix $C$, using the swap function.

**Require:** Matrix $R \in \mathbb{R}^{r \times N}$, the starting set of columns $\mathcal{I}$ of cardinality $n$.
**Ensure:** The updated set $\mathcal{I}$ corresponding to a dominant submatrix.

$C := R_{\mathcal{I}}^{+} R$
permutation $:= \{1, \dots, N\}$
$C$.swap($\mathcal{I}, \{1, \dots, n\}$, permutation)
$l := 0_N$
**for** $i := 1$ to $N$ **do**
    $l_i := \|C_{:,i}\|_2^2$
**end for**
$B := 0_{n \times N}$
**for** $i := 1$ to $n$ **do**
    **for** $j := n+1$ to $N$ **do**
        $B_{ij} := |C_{ij}|^2 + (1 + l_j)(1 - l_i)$
    **end for**
**end for**
$\{i, j\} := \arg\max_{i,j} B_{ij}$
**while** $B_{ij} > 1$ **do**
    $C_I := \frac{C_{:,i}^T}{1+l_i} C$
    **for** $k := 1$ to $N$ **do**
        $\tilde{l}_k := l_k - |C_{I,k}|^2 (1 + l_i)$
    **end for**
    $C := C - C_{:,i} C_I$
    $C$.swap($i, j$, permutation)
    $C_I$.swap($i, j$)
    $\tilde{l}$.swap($i, j$)
    swap($C_{j,:}, C_I$)
    **for** $k := 1$ to $M$ **do**
        $l_k := \tilde{l}_k + \frac{|C_{I,k}|^2}{1 - \tilde{l}_i}$
    **end for**
    $C := C + \frac{C_{:,i}}{1 - \tilde{l}_i} C_I$
    **for** $i := 1$ to $n$ **do**
        **for** $j := n+1$ to $N$ **do**
            $B_{ij} := |C_{ij}|^2 + (1 + l_j)(1 - l_i)$
        **end for**
    **end for**
    $\{i, j\} := \arg\max_{i,j} B_{ij}$
**end while**
$\mathcal{I} :=$ permutation$[1..n]$

## References


1. de Hoog FR, and Mattheij RMM. Subset selection for matrices. Linear Algebra and its Applications. 2007;**422**(2):349–359.
2. Avron H, and Boutsidis C. Faster Subset Selection for Matrices and Applications. SIAM J Matrix Anal Appl. 2011;**34**(4).
3. Golub GH, Klema V, and Stewart GW. Rank degeneracy and least-squares problems. In: Technical Report TR-456. Baltimore, MD, USA: Department of Computer Science, University of Maryland; 1976. .
4. Boutsidis C, Drineas P, and Magdon-Ismail M. Near-optimal column-based matrix reconstruction. SIAM Journal on Computing. 2013;.
5. Gu M, and Eisenstat SC. Efficient algorithms for computing a strong rank-revealing qr factorization. SIAM J ScI COMPUT. 1996;**17**(4):848–869.
6. Pan CT. On the existence and computation of rank revealing LU factorizations. Linear Algebra and its Applications. 2000;**316**:199–222.
7. Fedorov VV. New York: Academic Press; 1969.
8. Miller AJ, and Nam-Ky Nguyen NK. A Fedorov Exchange Algorithm for D-optimal Design. Applied statistics. 1994;**43**(4):669–677.
9. Mikhalev A, and Oseledets I. Rectangular submatrices of maximum volume and their computation. Doklady Mathematics. 2015;**91**:267–268.
10. Michalev AY, and V OI. Rectangular maximum-volume submatrices and their applications. Linear Algebra and its Applications. 2018;**538**:187–211.
11. Gu M, and Miranian L. Strong rank revealing cholesky factorization. Electronic Transactions on Numerical Analysis. 2004;**17**:76–92.
12. Osinsky AI, and Zamarashkin NL. Pseudo-skeleton approximations with better accuracy estimates. Linear Algebra and its Applications. 2018;**537**:221–249.
13. Zamarashkin NL, and Osinsky AI. On the Accuracy of Cross and Column Low-Rank Maxvol Approximations in Average. Computational Mathematics and Mathematical Physics. 2021;**61**:786–798.
14. Golub GH, and Van Loan CF. Matrix Computations. — 4th ed. Baltimore, MD: The Johns Hopkins University Press; 2013.
15. Çivril A, and Magdon-Ismail M. On selecting a maximum volume sub-matrix of a matrix and related problems. Theoretical Computer Science. 2009;**410**(47-49):4801–4811.
16. Goreinov SA, Oseledets IV, Savostyanov DV, Tyrtyshnikov EE, and Zamarashkin NL. How to find a good submatrix. In: Matrix Methods: Theory, Algorithms, Applications. World Scientific Publishing; 2010. p. 247–256.