Notes on Adaptive Rank Approximation

Sergey Voronin

Abstract: We discuss the efficient construction of various low rank approximations via randomized algorithms when the singular value distribution of the matrix is not known.

1. Introduction

The idea behind randomized algorithms for constructing low rank approximations is to apply the factorization to a smaller matrix. Given $\mathbf{A} \in \mathbb{R}^{m\times n}$, we can construct samples of the column space of $\mathbf{A}$ via the computation $\mathbf{Y} = \mathbf{AG}$ with $\mathbf{G}$ a Gaussian random matrix where $\mathbf{Y}$ is of size $m \times (k+p)$ with k the desired rank and p a small oversampling parameter. We can then construct a matrix with orthonormal columns (ON) $\mathbf{Q}$ of size $m \times (k+p)$ via a QR factorization of $\mathbf{Y}$. If $\mathbf{Y}$ captures a good portion of the range of $\mathbf{A}$ (assuming k is sufficiently large), then we expect that $\mathbf{QQ}^*\mathbf{A} \approx \mathbf{A}$. Note that while $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}$, the matrix product $\mathbf{QQ}^*$ multiplies out to the identity only in the case that $\mathbf{Q}$ is a square orthogonal matrix. Notice that $P = \mathbf{QQ}^*$ is a projector onto the range of $\mathbf{Q}$ and hence onto that of $\mathbf{Y}$ (where we assume that $\mathbf{Q}$ is obtained via a compact QR factorization of $\mathbf{Y}$). To see this, notice that $P^2 = P$ and for any $v \in \mathcal{R}(\mathbf{Q})$, $Pv = v$. Note also that in the case that $\mathbf{Y}$ captures the entire range of $\mathbf{A}$ we have equality, as per the lemma below:

Lemma 1.1. *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and let $\mathbf{Q} \in \mathbb{R}^{m\times r}$ be an orthonormal matrix. Then the following are equivalent:*

(1) $\mathcal{R}(\mathbf{A}) \subseteq \mathcal{R}(\mathbf{Q})$ *(the range of $\mathbf{A}$ is a subset of the range of $\mathbf{Q}$)*
(2) $\mathbf{A} = \mathbf{QQ}^T\mathbf{A}$

Proof. Assume (1) holds. Then this implies that there exists a matrix $\mathbf{S} \in \mathbb{R}^{r\times n}$ such that $\mathbf{A} = \mathbf{QS}$. It follows that:

$$\mathbf{QQ}^T\mathbf{A} = \mathbf{QQ}^T\mathbf{QS} = \mathbf{QS} = \mathbf{A}$$

since $\mathbf{Q}^T\mathbf{Q} = I$. Hence, (1) $\implies$ (2). Next, assume (2) holds. Then:

$$\mathbf{A} = \mathbf{QQ}^T\mathbf{A} = \mathbf{Q}\left(\mathbf{Q}^T\mathbf{A}\right) \implies \mathcal{R}(\mathbf{A}) \subseteq \mathcal{R}(\mathbf{Q})$$

Hence, (2) $\implies$ (1). □

Given that $\mathbf{QQ}^*\mathbf{A} \approx \mathbf{A}$, we perform a factorization of $\mathbf{Q}^*\mathbf{A}$ which is of size $(k+p) \times n$. Assuming $k \ll \min(m, n)$, the matrix $\mathbf{Q}^*\mathbf{A}$ is substantially smaller than $\mathbf{A}$. For example, we can perform the SVD of rank $k+p$ of $\mathbf{B} = \mathbf{Q}^*\mathbf{A}$ and then multiply by $\mathbf{Q}$ to get an approximate SVD of $\mathbf{A}$:

$$\mathbf{B} = \tilde{\mathbf{U}}\mathbf{D}\mathbf{V}^* \implies \mathbf{A} \approx (\mathbf{Q}\tilde{\mathbf{U}})\mathbf{D}\mathbf{V}^*$$

Notice that instead of performing the SVD of $\mathbf{B}$, we can also perform e.g. a pivoted QR factorization of $\mathbf{B}$ to get an approximate QR of $\mathbf{A}$:

$$\mathbf{BP} = \tilde{\mathbf{Q}}\mathbf{R} \implies \mathbf{AP} \approx (\mathbf{Q}\tilde{\mathbf{Q}})\mathbf{R}$$

The main computational challenge is in the construction of an ON matrix $\mathbf{Q}$ such that $\mathbf{QQ}^*\mathbf{A} \approx \mathbf{A}$. The simplest approach to follow is to do as above, forming $\mathbf{Y} = \mathbf{AG}$. The problem is that without advanced knowledge of the singular value distribution of $\mathbf{A}$, it is hard to guess an optimal k in the size of the $n \times (k+p)$ matrix $\mathbf{G}$. If k is chosen too small relative to the numerical rank of $\mathbf{A}$, then $\mathbf{QQ}^*\mathbf{A}$ will not be close to $\mathbf{A}$. On the other hand, if k is chosen too large, then the matrix

$\mathbf{B} = \mathbf{Q}^*\mathbf{A}$, once computed (in itself an expensive operation because of the QR on $\mathbf{Y}$) will be large and not much savings could be obtained by doing the factorization of choice on $\mathbf{B}$ instead of $\mathbf{A}$. Notice that the bound $\|(\mathbf{I}-\mathbf{Q}\mathbf{Q}^*)\mathbf{A}\|$ is exactly the error bound between the obtained factorization and the original matrix, since the factorization of $\mathbf{Q}^*\mathbf{A}$ is exact.

2. Discussion of Algorithms to construct $\mathbf{Q}$

The first algorithm we discuss produces $\mathbf{Q}$ such that

$$\|(\mathbf{I} - \mathbf{Q}\mathbf{Q}^*)\mathbf{A}\| < \epsilon, \tag{1}$$

but only with a certain very high probability (that is, it can potentially fail). The next two algorithms we discuss guarantee (1) on exit.

2.1. **Algorithm H.**

The algorithm pasted in Figure 1 is algorithm 4.2 from [1]:

Figure 1. Algorithm H

Algorithm 4.2: Adaptive Randomized Range Finder

Given an $m \times n$ matrix $\boldsymbol{A}$, a tolerance ε, and an integer r (e.g. $r = 10$), the following scheme computes an orthonormal matrix $\boldsymbol{Q}$ such that (4.2) holds with probability at least $1 - \min\{m, n\}10^{-r}$.

Draw standard Gaussian vectors $\boldsymbol{\omega}^{(1)}, \ldots, \boldsymbol{\omega}^{(r)}$ of length n.
For $i = 1, 2, \ldots, r$, compute $\boldsymbol{y}^{(i)} = \boldsymbol{A}\boldsymbol{\omega}^{(i)}$.
$j = 0$.
$\boldsymbol{Q}^{(0)} = [\,]$, the $m \times 0$ empty matrix.
while $\max\left\{\left\|\boldsymbol{y}^{(j+1)}\right\|, \left\|\boldsymbol{y}^{(j+2)}\right\|, \ldots, \left\|\boldsymbol{y}^{(j+r)}\right\|\right\} > \varepsilon/(10\sqrt{2/\pi})$,
    $j = j + 1$.
    Overwrite $\boldsymbol{y}^{(j)}$ by $\left(\mathbf{I} - \boldsymbol{Q}^{(j-1)}(\boldsymbol{Q}^{(j-1)})^*\right)\boldsymbol{y}^{(j)}$.
    $\boldsymbol{q}^{(j)} = \boldsymbol{y}^{(j)} / \left\|\boldsymbol{y}^{(j)}\right\|$.
    $\boldsymbol{Q}^{(j)} = [\boldsymbol{Q}^{(j-1)}\ \boldsymbol{q}^{(j)}]$.
    Draw a standard Gaussian vector $\boldsymbol{\omega}^{(j+r)}$ of length n.
    $\boldsymbol{y}^{(j+r)} = \left(\mathbf{I} - \boldsymbol{Q}^{(j)}(\boldsymbol{Q}^{(j)})^*\right)\boldsymbol{A}\boldsymbol{\omega}^{(j+r)}$.
    for $i = (j+1), (j+2), \ldots, (j+r-1)$,
        Overwrite $\boldsymbol{y}^{(i)}$ by $\boldsymbol{y}^{(i)} - \boldsymbol{q}^{(j)}\left\langle \boldsymbol{q}^{(j)},\ \boldsymbol{y}^{(i)}\right\rangle$.
    end for
end while
$\boldsymbol{Q} = \boldsymbol{Q}^{(j)}$.

We now describe how this algorithm works. Notice that the first three steps can be rewritten as:
Input: $\mathbf{Q} \in \mathbb{R}^{m\times r}, \mathbf{y} \in \mathbb{R}^m$ such that $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}_r$.
Iteration:

$$\begin{aligned} \bar{\mathbf{y}} &= (\mathbf{I} - \mathbf{Q}\mathbf{Q}^*)\mathbf{y} \\ \mathbf{q} &= \frac{\bar{\mathbf{y}}}{\|\bar{\mathbf{y}}\|} \\ \bar{\mathbf{Q}} &= [\mathbf{Q}, \mathbf{q}] \end{aligned}$$

Output: $\bar{\mathbf{Q}} \in \mathbb{R}^{m\times(r+1)}$.

The lemma below proves that on output, the range of the matrix $\bar{\mathbf{Q}}$ has expanded and orthonormality of $\mathbf{Q}$ has been preserved.

Lemma 2.1. *The output of the above procedure is a matrix* $\bar{\mathbf{Q}} \in \mathbb{R}^{m\times(r+1)}$ *such that:*

- *(a)* $\mathcal{R}(\bar{\mathbf{Q}}) = span(\mathcal{R}(\mathbf{Q}) \cup \mathbf{y})$.
- *(b)* $\bar{\mathbf{Q}}^*\bar{\mathbf{Q}} = \mathbf{I}_{r+1}$.

Note: we must have $\mathbf{y} \notin \mathcal{R}(\mathbf{Q})$ *as otherwise* $\mathbf{Q}\mathbf{Q}^*\mathbf{y} = \mathbf{y}$ *and* $\bar{\mathbf{y}} = (\mathbf{I} - \mathbf{Q}\mathbf{Q}^*)\mathbf{y} = \mathbf{0}$.

Proof. To show part (a), we must show that:
(a1) if $\mathbf{z} \in \text{span}(\mathcal{R}(\mathbf{Q}) \cup \mathbf{y})$ then $\mathbf{z} \in \mathcal{R}(\bar{\mathbf{Q}})$ and
(a2) if $\mathbf{z} \in \mathcal{R}(\bar{\mathbf{Q}})$ then $\mathbf{z} \in \text{span}(\mathcal{R}(\mathbf{Q}) \cup \mathbf{y})$.

Let us consider (a1). Since, $\mathbf{z} \in \text{span}(\mathcal{R}(\mathbf{Q}) \cup \mathbf{y})$ it follows that $\mathbf{z} = \alpha\mathbf{p} + \beta\mathbf{y}$ for some $\alpha, \beta \in \mathbb{R}$ where $\mathbf{p} \in \mathcal{R}(\mathbf{Q}) \implies \mathbf{p} = \mathbf{Q}\mathbf{r}$ for some vector $\mathbf{r}$. Also, since (in the iteration step) $\bar{\mathbf{y}} = (\mathbf{I} - \mathbf{Q}\mathbf{Q}^*)\mathbf{y}$, we have that $\mathbf{y} = \bar{\mathbf{y}} + \mathbf{Q}\mathbf{Q}^*\mathbf{y}$ and since $\mathbf{q} = \frac{\bar{\mathbf{y}}}{\|\bar{\mathbf{y}}\|}$, we have that $\bar{\mathbf{y}} = \|\bar{\mathbf{y}}\|\mathbf{q}$. Thus:

$$\begin{aligned}\mathbf{z} &= \alpha\mathbf{p} + \beta\mathbf{y} = \alpha\mathbf{Q}\mathbf{r} + \beta(\bar{\mathbf{y}} + \mathbf{Q}\mathbf{Q}^*\mathbf{y}) = \mathbf{Q}(\alpha\mathbf{r} + \beta\mathbf{Q}^*\mathbf{y}) + \beta\bar{\mathbf{y}} = \mathbf{Q}(\alpha\mathbf{r} + \beta\mathbf{Q}^*\mathbf{y}) + \beta\|\bar{\mathbf{y}}\|\mathbf{q} \\ &= [\mathbf{Q}, \mathbf{q}] \begin{bmatrix} \alpha\mathbf{r} + \beta\mathbf{Q}^*\mathbf{y} \\ \beta\|\bar{\mathbf{y}}\| \end{bmatrix} \in \mathcal{R}([\mathbf{Q}, \mathbf{q}]) = \mathcal{R}(\bar{\mathbf{Q}}).\end{aligned}$$

Next, we consider (a2). Since $\mathbf{z} \in \text{span}(\mathcal{R}(\bar{\mathbf{Q}}))$, $\mathbf{z} = \bar{\mathbf{Q}}\mathbf{w}$ for some vector $\mathbf{w}$ which can be split into two parts w_1, w_2. It follows that:

$$\begin{aligned}\mathbf{z} &= [\mathbf{Q}, \mathbf{q}] \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \mathbf{Q}\mathbf{w}_1 + \mathbf{q}\mathbf{w}_2 = \mathbf{Q}\mathbf{w}_1 + \frac{\bar{\mathbf{y}}}{\|\bar{\mathbf{y}}\|}\mathbf{w}_2 = \mathbf{Q}\mathbf{w}_1 + \frac{\mathbf{w}_2}{\|\bar{\mathbf{y}}\|}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^*)\mathbf{y} \\ &= \mathbf{Q}\left(\mathbf{w}_1 - \frac{\mathbf{w}_2}{\|\bar{\mathbf{y}}\|}\mathbf{Q}^*\mathbf{y}\right) + \frac{\mathbf{w}_2}{\|\bar{\mathbf{y}}\|}\mathbf{y} \in \text{span}(\mathcal{R}(\mathbf{Q}) \cup \mathbf{y})\end{aligned}$$

Thus, (a) is established. To show (b), we compute:

$$\bar{\mathbf{Q}}^*\bar{\mathbf{Q}} = [\mathbf{Q}, \mathbf{q}]^*[\mathbf{Q}, \mathbf{q}] = \begin{bmatrix} \mathbf{Q}^*\mathbf{Q} & \mathbf{Q}^*\mathbf{q} \\ \mathbf{q}^*\mathbf{Q} & \mathbf{q}^*\mathbf{q} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_r & 0 \\ 0 & 1 \end{bmatrix}$$

where we have used that

$$\mathbf{Q}^*\mathbf{q} = (\mathbf{q}^*\mathbf{Q})^* = \mathbf{Q}^*\frac{\bar{\mathbf{y}}}{\|\bar{\mathbf{y}}\|} = \mathbf{Q}^*\frac{1}{\|\bar{\mathbf{y}}\|}(\mathbf{I} - \mathbf{Q}\mathbf{Q}^*)\mathbf{y} = \mathbf{Q}^*\frac{\mathbf{y}}{\|\bar{\mathbf{y}}\|} - \mathbf{Q}^*\frac{\mathbf{y}}{\|\bar{\mathbf{y}}\|} = 0$$

□

Next, let us look at the remaining steps on lines $10 - 14$ of the algorithm. Here we pick a random Gaussian vector $\mathbf{w}$ and set $y^{(j+r)} = (\mathbf{I} - \bar{\mathbf{Q}}\bar{\mathbf{Q}}^*)\mathbf{A}\mathbf{w}$ where $\bar{\mathbf{Q}}$ is the updated matrix at the j-th iteration. This of course implies that $y^{(j+r)} \in \mathcal{R}\left((\mathbf{I} - \bar{\mathbf{Q}}\bar{\mathbf{Q}}^*)\mathbf{A}\right) = \mathcal{R}\left((\mathbf{I} - \mathbf{Q}^{(j)}(\mathbf{Q}^{(j)})^*)\mathbf{A}\right)$. We now prove the following claim by induction, which allows us to make sense of the while loop stopping criterion.

Lemma 2.2. *At each iteration* $j > 1$ *we have that :*

$$y^{(j+1)}, y^{(j+2)}, \ldots, y^{(j+r)} \in \mathcal{R}\left((\mathbf{I} - \mathbf{Q}^{(j)}(\mathbf{Q}^{(j)})^*)\mathbf{A}\right) \tag{2}$$

Proof. We proceed by induction. First, note that the claim trivially holds for $j = 0$. Since $Q^{(0)} = 0$, we have to check the claim that $y^{(1)}, \ldots, y^{(r)} \in \mathcal{R}(A)$, which holds by line 2 of Algorithm H. Notice that the claim holds also for $j = 1$ since on line 13, we have:

$$y^{(i)} = y^{(i)} - q^{(1)}\langle q^{(1)}, y^{(i)}\rangle = y^{(i)} - q^{(1)}(q^{(1)})^*y^{(i)} = (I - Q^{(1)}(Q^{(1)})^*)y^{(i)} = (I - Q^{(1)}(Q^{(1)})^*)Aw^{(i)}$$

Suppose $j > 1$ and

$$y^{(j)}, y^{(j+1)}, \ldots, y^{(j+r-1)} \in \mathcal{R}\left((\mathbf{I} - \mathbf{Q}^{(j-1)}(\mathbf{Q}^{(j-1)})^*)\mathbf{A}\right)$$

We would like to show that this implies that (2) holds. By assumption, prior to line 13 of the algorithm (the for loop update), we have that for every $i = (j+1), (j+2), \ldots, (j+r-1)$,

$$\mathbf{y}^{(i)} = \left(\mathbf{I} - \mathbf{Q}^{(j-1)}(\mathbf{Q}^{(j-1)})^*\right)\mathbf{A}\boldsymbol{\mu}^{(i)}$$

for some vector $\boldsymbol{\mu}^{(i)}$ by the recursive assumption. The update on line 13 gives:

$$\begin{aligned}
\mathbf{y}^{(i)} &= \mathbf{y}^{(i)} - \mathbf{q}^{(j)}\langle \mathbf{q}^{(j)}, \mathbf{y}^{(i)}\rangle = \mathbf{y}^{(i)} - \mathbf{q}^{(j)}(\mathbf{q}^{(j)})^*\mathbf{y}^{(i)} \\
&= \left(\mathbf{I} - \mathbf{Q}^{(j-1)}(\mathbf{Q}^{(j-1)})^*\right)\mathbf{A}\boldsymbol{\mu}^{(i)} - \mathbf{q}^{(j)}(\mathbf{q}^{(j)})^*\left(\mathbf{I} - \mathbf{Q}^{(j-1)}(\mathbf{Q}^{(j-1)})^*\right)\mathbf{A}\boldsymbol{\mu}^{(i)} \\
&= \left(\mathbf{I} - \mathbf{Q}^{(j-1)}(\mathbf{Q}^{(j-1)})^*\right)\mathbf{A}\boldsymbol{\mu}^{(i)} - \mathbf{q}^{(j)}(\mathbf{q}^{(j)})^*\mathbf{A}\boldsymbol{\mu}^{(i)} \\
&= \left(\mathbf{I} - \mathbf{Q}^{(j-1)}(\mathbf{Q}^{(j-1)})^* - \mathbf{q}^{(j)}(\mathbf{q}^{(j)})^*\right)\mathbf{A}\boldsymbol{\mu}^{(i)} = \left(\mathbf{I} - \mathbf{Q}^{(j)}(\mathbf{Q}^{(j)})^*\right)\mathbf{A}\boldsymbol{\mu}^{(i)}
\end{aligned}$$

where we have used that $(\mathbf{q}^{(j)})^*\mathbf{Q}^{(j-1)} = 0$ and that $\mathbf{Q}^{(j)} = [\mathbf{Q}^{(j-1)}, \mathbf{q}^{(j)}]$, so that:

$$\mathbf{Q}^{(j)}(\mathbf{Q}^{(j)})^* = \left[\mathbf{Q}^{(j-1)}, \mathbf{q}^{(j)}\right]\begin{bmatrix}(\mathbf{Q}^{(j-1)})^* \\ (\mathbf{q}^{(j)})^*\end{bmatrix} = \mathbf{Q}^{(j-1)}(\mathbf{Q}^{(j-1)})^* + \mathbf{q}^{(j)}(\mathbf{q}^{(j)})^*$$

Thus, after the update, (projecting the new $\mathbf{y}^{(i)}$ away from previously discovered elements of the range of $\mathbf{A}$), the vectors remain in the range of $\left(\mathbf{I} - \mathbf{Q}^{(j)}(\mathbf{Q}^{(j)})^*\right)\mathbf{A}$. □

Given, Lemma 2.2, the whole algorithm now makes sense. The algorithm continues until all of the $\mathbf{y}^{(i)}$ for $i = j+1, j+2, \ldots, j+r$ become small in norm. At the point all these become small, since we have shown all the $\mathbf{y}^{(i)} \in \mathcal{R}\left((\mathbf{I} - \mathbf{Q}^{(j)}(\mathbf{Q}^{(j)})^*)\mathbf{A}\right)$, it must be the case that $\left((\mathbf{I} - \mathbf{Q}^{(j)}(\mathbf{Q}^{(j)})^*)\mathbf{A}\right)\boldsymbol{\mu}^{(i)} \approx 0$ for some set of vectors $\boldsymbol{\mu}^{(i)}$. Given r is large enough, this implies with high probability that $\|(\mathbf{I} - \mathbf{Q}^{(j)}(\mathbf{Q}^{(j)})^*)\mathbf{A}\| \approx 0$ which means that $\mathbf{A} \approx \mathbf{Q}^{(j)}(\mathbf{Q}^{(j)})^*)\mathbf{A}$ and we have constructed the desired $\mathbf{Q}$ which captures much of the range of $\mathbf{A}$. Of course, the result is probabalistic. We could be really unlucky and it could simply be the case that the vectors $\boldsymbol{\mu}^{(i)}$ all have small norm, but this chance decreases rapidly if the number of vectors sampled r is large enough (e.g. $r = 10$).

2.2. **Algorithms 2 and 3.** In contrast to Algorithm H in Figure 1, it is possible to construct an algorithm such that on output, given $\epsilon > 0$, we have that (1) holds. This is an important improvement over Algorithm H, where on output, the norm in (1) is guaranteed to be small only with a given probability. Another improvement, is the ability, in Algorithm 2, to block computations, updating $\mathbf{Q}$ with a block of new vectors in a time, instead of using a single vector. The two algorithms we will discuss are presented side by side below in Figure 2.2 taken from [2]. The first being the single vector version and the second being the blocked method. For both methods, on exit, we have that (1) holds. Let us first analyze the simpler single vector method.

Lemma 2.3. *At the end of iteration j of Algorithm 1, we have:*

$$\mathbf{A}^{(j)} = (\mathbf{I} - \mathbf{Q}_j\mathbf{Q}_j^*)\mathbf{A} \quad \textit{and} \quad \mathbf{B}_j = \mathbf{Q}_j^*\mathbf{A}$$

Proof. The results can be established by induction. Notice that after the first iteration, $\mathbf{q}_1 \in \mathcal{R}(\mathbf{A})$, $\mathbf{Q}_1 = [\mathbf{q}_1]$, $\mathbf{B}_1 = [\mathbf{b}_1] = [\mathbf{q}_1^*\mathbf{A}] = \mathbf{Q}_1^*\mathbf{A}$. Next, since $\|\mathbf{q}_j\| = 1$, we have that $(\mathbf{I} - \mathbf{q}_j\mathbf{q}_j^*)\alpha\mathbf{q}_j = \alpha\mathbf{q}_j - \alpha\mathbf{q}_j = 0$, which implies that $(\mathbf{I} - \mathbf{q}_j\mathbf{q}_j^*) \perp \mathcal{R}(\mathbf{q}_j)$ and that $\mathbf{q}_i \perp \mathbf{q}_j$ for $i \neq j$. In particular, $\mathbf{q}_2 \perp \mathbf{q}_1$, so after the second iteration,

$$\mathbf{A}^{(2)} = \mathbf{A}^{(1)} - \mathbf{q}_2\mathbf{q}_2^*\mathbf{A}^{(1)} = \mathbf{A}^{(1)} - \mathbf{q}_2\mathbf{q}_2^*(A - \mathbf{q}_1\mathbf{q}_1^*) = (\mathbf{I} - \mathbf{q}_1\mathbf{q}_1^*)\mathbf{A} = (\mathbf{I} - \mathbf{Q}_1\mathbf{Q}_1^*)\mathbf{A}.$$

Let us assume that $\mathbf{A}^{(j)} = (\mathbf{I} - \mathbf{Q}_j\mathbf{Q}_j^*)\mathbf{A}$. It follows that:

$$\begin{aligned}\mathbf{A}^{(j+1)} &= \mathbf{A}^{(j)} - \mathbf{q}_{(j+1)}\mathbf{q}_{(j+1)}^*\mathbf{A}^{(j)} = (\mathbf{I} - \mathbf{q}_{(j+1)}\mathbf{q}_{(j+1)}^*)\mathbf{A}^{(j)} = (\mathbf{I} - \mathbf{q}_{(j+1)}\mathbf{q}_{(j+1)}^*)(\mathbf{I} - \mathbf{Q}_j\mathbf{Q}_j^*)\mathbf{A} \\ &= (\mathbf{I} - \mathbf{Q}_j\mathbf{Q}_j^* - \mathbf{q}_{(j+1)}\mathbf{q}_{(j+1)}^*)\mathbf{A} = (\mathbf{I} - \mathbf{Q}_{j+1}\mathbf{Q}_{j+1}^*)\mathbf{A}\end{aligned}$$

Similarly, if we assume $\mathbf{B}_j = \mathbf{Q}_j^*\mathbf{A}$, then we have:

$$\mathbf{B}_{(j+1)} = \begin{bmatrix} \mathbf{Q}_j^*\mathbf{A} \\ \mathbf{q}_{(j+1)}^*\mathbf{A}^{(j)} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_j^*\mathbf{A} \\ \mathbf{q}_{(j+1)}^*(\mathbf{I} - \mathbf{Q}_j\mathbf{Q}_j^*)\mathbf{A} = \mathbf{q}_{(j+1)}^*\mathbf{A} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_j^* \\ \mathbf{q}_{(j+1)}^* \end{bmatrix}\mathbf{A} = \mathbf{Q}_{(j+1)}^*\mathbf{A}$$

□

Notice that we quit Algorithms 1 and 2 precisely when $\|A^{(j)}\| < \epsilon$ is small, so since we have shown that $\mathbf{A}^{(j)} = (\mathbf{I} - \mathbf{Q}_j\mathbf{Q}_j^*)\mathbf{A}$, we have on exit that (1) holds.

Algorithm 1

(1) $\mathbf{Q}_0 = [\,]$; $\mathbf{B}_0 = [\,]$; $\mathbf{A}_0 = \mathbf{A}$; $j = 0$;
(2) **while** $\|\mathbf{A}^{(j)}\| > \varepsilon$
(3) $\quad j = j + 1$
(4) $\quad$ Pick a unit vector $\mathbf{q}_j \in \mathrm{ran}(\mathbf{A}^{(j-1)})$.
(5) $\quad \mathbf{b}_j = \mathbf{q}_j^*\mathbf{A}^{(j-1)}$
(6) $\quad \mathbf{Q}_j = [\mathbf{Q}_{j-1}\ \mathbf{q}_j]$
(7) $\quad \mathbf{B}_j = \begin{bmatrix} \mathbf{B}_{j-1} \\ \mathbf{b}_j \end{bmatrix}$
(8) $\quad \mathbf{A}^{(j)} = \mathbf{A}^{(j-1)} - \mathbf{q}_j\mathbf{b}_j$
(9) **end while**
(10) $k = j$.

function $[\mathbf{Q}, \mathbf{B}] = \mathtt{randbQB}(\mathbf{A}, \varepsilon, b)$

(1) $\mathbf{A}^{(0)} = \mathbf{A}$
(2) **for** $i = 1, 2, 3, \ldots$
(3) $\quad \mathbf{\Omega}_i = \mathtt{randn}(n, b)$
(4) $\quad \mathbf{Q}_i = \mathtt{qr}(\mathbf{A}\mathbf{\Omega}_i, 0)$
(4') $\quad \mathbf{Q}_i = \mathtt{qr}(\mathbf{Q}_i - \sum_{j=1}^{i-1}\mathbf{Q}_j\mathbf{Q}_j^*\mathbf{Q}_i, 0)$
(5) $\quad \mathbf{B}_i = \mathbf{Q}_i^*\mathbf{A}^{(i-1)}$
(6) $\quad \mathbf{A}^{(i)} = \mathbf{A}^{(i-1)} - \mathbf{Q}_i\mathbf{B}_i$
(7) $\quad$ **if** $\|\mathbf{A}^{(i)}\| < \varepsilon$ **then stop**
(8) **end for**
(9) Set $\mathbf{Q} = [\mathbf{Q}_1\ \cdots\ \mathbf{Q}_i]$ and $\mathbf{B} = [\mathbf{B}_1^*\ \cdots\ \mathbf{B}_i^*]^*$.

FIGURE 2. Algorithms 1 and 2.

Next, we discuss the block algorithm on the right of Figure 2.2. Note that in the blocked scheme on the right, line (4′) is not necessary in exact arithmetic, it is simply a reorthogonalization step. Let us define: $\bar{\mathbf{Q}}_i = [\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_i]$, $\bar{\mathbf{B}}_i = [\mathbf{B}_1^*, \mathbf{B}_2^*, \ldots, \mathbf{B}_i^*]^*$, and $\bar{\mathbf{Y}}_i = \mathbf{A}[\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_i] = [\mathbf{A}\mathbf{\Omega}_1, \ldots, \mathbf{A}\mathbf{\Omega}_i] = [\mathbf{Y}_1, \ldots, \mathbf{Y}_i]$. Let us now make an important assumption which we later prove: the orthonormality of the columns of $\bar{\mathbf{Q}}_i$. This implies that $\mathbf{Q}_i^*\mathbf{Q}_j = 0$ when $i \neq j$. Notice also an important relation for $A^{(i)}$ which we will use:

$$\mathbf{A}^{(i)} = \mathbf{A}^{(i-1)} - \mathbf{Q}_i\mathbf{B}_i = \mathbf{A}^{(i-1)} - \mathbf{Q}_i\mathbf{Q}_i^*\mathbf{A}^{(i-1)} = (\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^*)\,\mathbf{A}^{(i-1)}$$

Lemma 2.4. *In exact arithmetic the following relations hold for the blocked algorithm:*

$$\mathbf{A}^{(i)} = \mathbf{A} - \bar{\mathbf{Q}}_i\bar{\mathbf{Q}}_i^*\mathbf{A} \quad \textit{and} \quad \bar{\mathbf{B}}_i = \bar{\mathbf{Q}}_i^*\mathbf{A}$$

Proof. In each case we use the orthonormality of $\bar{\mathbf{Q}}_i$ and repeared substitution via the recursive relations for $\mathbf{A}^{(i)}$ and $\mathbf{B}_i$. For the first equation:

$$\begin{aligned}\mathbf{A}^{(i)} &= (\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^*)\,\mathbf{A}^{(i-1)} = (\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^*)\left(\mathbf{I} - \mathbf{Q}_{i-1}\mathbf{Q}_{i-1}^*\right)\mathbf{A}^{(i-2)} \\ &= \left(\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^* - \mathbf{Q}_{i-1}\mathbf{Q}_{i-1}^* - \cdots - \mathbf{Q}_1\mathbf{Q}_1^*\right)\mathbf{A}^{(0)} = \mathbf{A}(\mathbf{I} - \bar{\mathbf{Q}}_i\bar{\mathbf{Q}}_i^*)\end{aligned}$$

For the second equation:

$$\mathbf{B}_i = \mathbf{Q}_i^*\mathbf{A}^{(i-1)} = \mathbf{Q}_i^*\left(\mathbf{I} - \mathbf{Q}_{i-1}\mathbf{Q}_{i-1}^*\right)\mathbf{A}^{(i-2)} = \mathbf{Q}_i^*\mathbf{A}^{(i-2)} = \cdots = \mathbf{Q}_i^*\mathbf{A}^{(0)} = \mathbf{Q}_i^*\mathbf{A}$$

It follows that:

$$\bar{\mathbf{Q}}_i^*\mathbf{A} = [\mathbf{Q}_1, \ldots, \mathbf{Q}_i]^*\mathbf{A} = \begin{bmatrix}\mathbf{Q}_1^* \\ \vdots \\ \mathbf{Q}_i^*\end{bmatrix}\mathbf{A} = \begin{bmatrix}\mathbf{Q}_1^*\mathbf{A} \\ \vdots \\ \mathbf{Q}_i^*\mathbf{A}\end{bmatrix} = \begin{bmatrix}\mathbf{B}_1^* \\ \vdots \\ \mathbf{B}_i^*\end{bmatrix} = \bar{\mathbf{B}}_i$$

□

Next, we prove a more general lemma concerning the algorithm. We first describe some constructions useful for the proof. We will use the matrix $\mathbf{P}_i = \sum_{k=1}^{i}\mathbf{Q}_k\mathbf{Q}_k^*$, as the orthogonal projection onto the range of $\bar{\mathbf{Y}}_i = \mathbf{A}[\mathbf{\Omega}_1, \ldots, \mathbf{\Omega}_i] = [\mathbf{Y}_1, \ldots, \mathbf{Y}_i]$. Notice that the $\mathbf{P}_i$ is constructed from the standard orthogonal matrix projection formula in linear algebra: $\mathbf{P} = \mathbf{M}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$. Given $\mathbf{Y}_k$ for $1 \le k \le i$, if we take $[\mathbf{Q}_k, \mathbf{R}_k] = qr(\mathbf{Y}_k)$ then plugging in the factorization for $\mathbf{M} = \mathbf{Q}_k\mathbf{R}_k$ we obtain

$$\mathbf{P}_{(i,k)} = \mathbf{Q}_k\mathbf{R}_k\left((\mathbf{Q}_k\mathbf{R}_K)^T(\mathbf{Q}_k\mathbf{R}_k)\right)^{-1}(\mathbf{Q}_k\mathbf{R}_k)^T = \mathbf{Q}_k\mathbf{Q}_k^T$$

We get $\mathbf{P}_i = \sum_{k=1}^{i}\mathbf{P}_{(i,k)}$. Since $\mathbf{P}_i$ is a projection matrix, it satisfies the following properties:

$$\mathbf{P}_i^* = \mathbf{P}, \quad \mathbf{P}_i\mathbf{x} = \mathbf{x} \text{ for } \mathbf{x} \in \mathcal{R}(\bar{\mathbf{Y}}_i), \quad \mathbf{P}_i^2 = \mathbf{P}$$

We will also make use of the following lemma:

Lemma 2.5. *Given,* $\mathbf{P}_i = \sum_{k=1}^{i}\mathbf{Q}_k\mathbf{Q}_k^*$ *and if* $\bar{\mathbf{Q}}_i = [\mathbf{Q}_1, \ldots, \mathbf{Q}_i]$ *has orthonormal columns, then the following holds:*

$$\mathcal{R}(\mathbf{P})_i = \mathcal{R}\left(\sum_{k=1}^{i}\mathbf{Q}_k\mathbf{Q}_k^*\right) = \sum_{k=1}^{i}\mathcal{R}(\mathbf{Q}_k\mathbf{Q}_k^*) = \sum_{k=1}^{i}\mathcal{R}(\mathbf{Q}_k) = \mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_k])$$

We can now state and prove the following lemma. In it, we will prove that $\bar{\mathbf{Q}}_i$ has orthonormal columns, so all of the above results will hold and be used in the proof of the other parts of the lemma:

Lemma 2.6. *For the blocked randQB algorithm presented above, the following holds:*

- *(a)* $\bar{\mathbf{Q}}_i = [\mathbf{Q}_1, \ldots, \mathbf{Q}_i]$ *has orthonormal columns*
- *(b)* $\mathbf{A}^{(i)} = (\mathbf{I} - \mathbf{P}_i)\mathbf{A}$
- *(c)* $\mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_i]) = \mathcal{R}([\mathbf{A}\mathbf{\Omega}_1, \ldots, \mathbf{A}\mathbf{\Omega}_i])$

Proof. The fact the all claims hold for $i = 1$ can be verified directly from the algorithm statements. We proceed by induction. Let us suppose all claims hold up to $(i-1)$. Let's begin with (a) and (b). We assume that: $\bar{\mathbf{Q}}_{i-1} = [\mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}]$ has orthonormal columns and that $\mathbf{A}^{(i-1)} = (\mathbf{I} - \mathbf{P}_{i-1})\mathbf{A}$. We like to show that the same holds true for $\bar{\mathbf{Q}}_i$ and that $\mathbf{A}^{(i)} = (\mathbf{I}-\mathbf{P}_i)\mathbf{A}$. Since $\mathbf{Q}_i = qr(\mathbf{A}^{(i-1)}\mathbf{\Omega}_i)$, it follows that $\mathbf{Q}_i$ has orthonormal columns and that:

$$\mathcal{R}(\mathbf{Q}_i) = \mathcal{R}(\mathbf{A}^{(i-1)}\mathbf{\Omega}_i) \subseteq \mathcal{R}(\mathbf{A}^{(i-1)}) \subseteq \mathcal{R}(\mathbf{I} - \mathbf{P}_{i-1}) = \mathcal{R}(\mathbf{P}_{i-1})^{\perp} = \mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}])^{\perp}$$

Hence, it immediately follows that since $\mathcal{R}(\mathbf{Q}_i) \subseteq \mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}])^{\perp}$, that $\mathbf{Q}_i^*\mathbf{Q}_j = 0$ for $i \neq j$ and so (a) is established. For part (b), we repeatedly substitute the recursive assumption:

$$\begin{aligned}
\mathbf{A}^{(i)} &= (\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^*)\mathbf{A}^{(i-1)} = (\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^*)(\mathbf{I} - \mathbf{P}_{i-1})\mathbf{A} \\
&= (\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^* - \mathbf{P}_{i-1} + \mathbf{Q}_i\mathbf{Q}_i^*\mathbf{P}_{i-1})\mathbf{A} = (\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^* - \mathbf{P}_{i-1})\mathbf{A} \\
&= (\mathbf{I} - \mathbf{Q}_i\mathbf{Q}_i^* - \sum_{k=1}^{i-1}\mathbf{Q}_k\mathbf{Q}_k^*)\mathbf{A} = (\mathbf{I} - \sum_{k=1}^{i}\mathbf{Q}_k\mathbf{Q}_k^*)\mathbf{A} = (\mathbf{I} - \mathbf{P}_i)\mathbf{A}
\end{aligned}$$

By induction, this establishes (b).

To establish (c), we must show that

$$\mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_i]) \subseteq \mathcal{R}(\mathbf{A\Omega}_1, \ldots, \mathbf{A\Omega}_i) \quad \text{and} \quad \mathcal{R}(\mathbf{A\Omega}_1, \ldots, \mathbf{A\Omega}_i) \subseteq \mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_i])$$

We recall that $\mathbf{A}^{(i-1)} = (\mathbf{I}-\mathbf{P}_{i-1})\mathbf{A}$ and $\mathbf{Q}_i = qr(\mathbf{A}^{(i-1)}\mathbf{\Omega}_i)$. Let us assume (induction assumption) that $\mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}]) = \mathcal{R}([\mathbf{A\Omega}_1, \ldots, \mathbf{A\Omega}_{i-1}])$. It follows that:

$$\begin{aligned}
\mathcal{R}(\mathbf{Q}_i) &= \mathcal{R}(\mathbf{A}^{(i-1)}\mathbf{\Omega}_i) = \mathcal{R}((\mathbf{I} - \mathbf{P}_{i-1})\mathbf{A\Omega}_i) = \mathcal{R}(\mathbf{A\Omega}_i - \mathbf{P}_{i-1}\mathbf{A\Omega}_i) \\
&\subseteq \mathcal{R}(\mathbf{A\Omega}_i) + \mathcal{R}(\mathbf{P}_{i-1}\mathbf{A\Omega}_i) \subseteq \mathcal{R}(\mathbf{A\Omega}_i) + \mathcal{R}(\mathbf{P}_{i-1}) = \mathcal{R}(\mathbf{A\Omega}_i) + \mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}])
\end{aligned}$$

Thus, we have $\mathcal{R}(\mathbf{Q}_i) \subseteq \mathcal{R}(\mathbf{A\Omega}_i) + \mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}])$ and we apply this below:

$$\begin{aligned}
\mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_i]) &= \mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}]) + \mathcal{R}(\mathbf{Q}_i) \\
&\subseteq \mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}]) + \mathcal{R}(\mathbf{A\Omega}_i) + \mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}]) = \mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_{i-1}]) + \mathcal{R}(\mathbf{A\Omega}_i) \\
&= \mathcal{R}([\mathbf{A\Omega}_1, \ldots, \mathbf{A\Omega}_{i-1}]) + \mathcal{R}(\mathbf{A\Omega}_i) = \mathcal{R}([\mathbf{A\Omega}_1, \ldots, \mathbf{A\Omega}_i])
\end{aligned}$$

This shows that $\mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_i]) \subseteq \mathcal{R}([\mathbf{A\Omega}_1, \ldots, \mathbf{A\Omega}_i])$. To complete the proof of (c) we show the other direction. We must only show that $\mathcal{R}(\mathbf{A\Omega}_i) \in \mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_i])$. Let $\mathbf{z} \in \mathcal{R}(\mathbf{A\Omega}_i)$ which implies that $\mathbf{z} = \mathbf{A\Omega}_i\mathbf{x}$. Then:

$$\begin{aligned}
\mathbf{z} &= (\mathbf{I} - \mathbf{P}_{i-1} + \mathbf{P}_{i-1})\mathbf{A\Omega}_i\mathbf{x} = (\mathbf{I} - \mathbf{P}_{i-1})\mathbf{A\Omega}_i\mathbf{x} + \mathbf{P}_{i-1}\mathbf{A\Omega}_i\mathbf{x} = \mathbf{A}^{(i-1)}\mathbf{\Omega}_i\mathbf{x} + \mathbf{P}_{i-1}\mathbf{A\Omega}_i\mathbf{x} \\
&\subseteq \mathcal{R}(\mathbf{A}^{(i-1)}\mathbf{\Omega}_i) + \mathcal{R}(\mathbf{P}_{i-1}\mathbf{A\Omega}_i) \subseteq \mathcal{R}(\mathbf{Q}_i) + \mathcal{R}(\mathbf{P}_{i-1}) \subseteq \mathcal{R}([\mathbf{Q}_1, \ldots, \mathbf{Q}_i])
\end{aligned}$$

where we have used Lemma 2.5. □

2.3. **Power sampling scheme.** A slight modification which gives noticeably better results in practice is the power iteration. Here, we form $\mathbf{Y}$ a bit differently, via $\mathbf{Y} = (\mathbf{AA}^*)^q\mathbf{A\Omega}$ with $q \geq 1$ (when $q = 0$, we recover the original method). Note that $\mathbf{A}$ and $(\mathbf{AA}^*)^q\mathbf{A}$ have the same eigenvectors and related eigenvalues. Plugging in the SVD, $\mathbf{A} = \mathbf{U\Sigma V}^*$ we have:

$$\begin{aligned}
&\mathbf{AA}^* = \mathbf{U\Sigma}^2\mathbf{U}^* \implies (\mathbf{AA}^*)^2 = \mathbf{U\Sigma}^2\mathbf{U}^*\mathbf{U\Sigma}^2\mathbf{U}^* = \mathbf{U\Sigma}^4\mathbf{U}^* \\
&\implies (\mathbf{AA}^*)^q = \mathbf{U\Sigma}^{2q}\mathbf{U}^* \\
&\implies (\mathbf{AA}^*)^q\mathbf{A} = \mathbf{U\Sigma}^{2q}\mathbf{U}^*\mathbf{U\Sigma V}^* = \mathbf{U\Sigma}^{2q+1}\mathbf{V}^*
\end{aligned}$$

When $\mathbf{A}$ is such that its trailing singular values decay slowly, the matrix $(\mathbf{AA}^*)^q\mathbf{A}$ has much more rapid decay of singular values. This approach results (in expectation, see section 10.4 HMT) a smaller value of $\|(\mathbf{I}-\mathbf{QQ}^*)\mathbf{A}\|$ in the case that $\mathbf{A}$ is such that its tail singular values are significant.

That is, for matrices whose singular values decay very fast this is probably not necessary but for many practical cases this gives significant improvement. This idea can be easily adapted to Algorithms 1 and 2, in the sampling step. For example, we can replace line (4) of Algorithm 2 with the power sampling scheme.

References

1. Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Review **53** (2011), no. 2, 217–288.
2. P.-G. Martinsson and S. Voronin, *A randomized blocked algorithm for efficiently computing rank-revealing factorizations of matrices*, ArXiv e-prints (2015).