**Computational Linear Algebra Project**
**Spring 2019**

EPFL

# Blendenpik: Randomized Least Squares

—

William Cappelletti
261934

## 1 Introduction

In this report, we give a general review to Blendenpik [1], a randomized least squared solver, which was able to outperform state-of-the-art LS-solver from LAPACK in many dense cases.

The algorithm aims to find an approximate solution to an overdeterminated linear system $Ax = b$. In order to do so, it uses an iterative Least Squares solver, combined with a particular preconditioner obtained through a randomized procedure. More in detail, the algorithm samples randomly among $A$ rows to get a new matrix $\tilde{A}$. The idea is that, if we sample enough rows, and by doing so we do not loose too much information, the solution $x_*$ of the new linear system $\tilde{A}x = b$ should be very close to the optimal solution. This linear system being smaller, we can efficiently compute a QR decomposition of the new martix $\tilde{A}$, and use $R$ as a right preconditioner to drastically improve the convergence of the LS-solver for $\tilde{A}R^{-1}y = b$, $x = R^{-1}y$. As we will prove in Section 3, it turns out that $R$ is a good preconditioner for the original system too. Therefore, Blendenpik applies it to $Ax = b$ and drastically reduces the overall solving time.

As we said, to have a good approximation through sampling, we need to not miss important rows. Intuitively, we would like each row to carry approximately the same "amount of information" about the solution of the linear system (this concept is explained in more detail in Section 2), therefore the algorithm starts by a *row-mixing preprocessing* of the matrix $A$ which should reduce its *coherence*. In fact, simply row sapling would not work for a general matrix $A$, as it could happen, for instance, that all but one amongst its rows define some coordinates of the solution, while the other is the only one which gives the missing information. In such a setting it is evident that with high probability we miss the necessary row and therefore we cannot find a solution close to that of the original system.

## 2 Coherence

Given a matrix $A \in \mathbb{R}^{m\times n}$, its *coherence* is defined as the maximum norm of a row in $Q$, where $Q \in \mathbb{R}^{m\times n}$ forms an orthonormal basis for the column space of $A$; it is always between $n/m$ and 1. More precisely, if we denote by $Q_{(i)}$ the $i$-th row of $Q$, the *coherence* of A is given by

$$\mu(A) = \max_{i\in[m]} ||Q_{(i)}||_2^2 \tag{2.1}$$

Intuitively, the lower coherence is, the less "information" each coordinate carries. By the way it is defined, if $\mu(A)$ is close to $n/m$, then all of $Q$'s rows have the same "importance"; otherwise, if it is close to 1, then there is one row which has more "information".

The concept may become clearer through an example. For instance, by running 10000 simulations in MATLAB we estimate that the average coherence of a random $1000 \times 50$ matrix is 0.072, and we note that the minimum value, as stated earlier, $\mu(A)$ can have is $0.05 = 50/1000$.

On the other and, the matrix $A \in \mathbb{R}^{1000\times 50}$ defined as

$$A = \begin{pmatrix} 1 & \cdots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \cdots & 1 & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

has coherence 1. We can see that in this case, all $A$ rows from 1 to 999 are equivalent in the solution of a linear system, while $A_{(1000)}$ is linearly independent. It is obvious, thinking about QR-decomposition of $A$, that there is only one eigenvector corresponding to the eigenvalue 1 (corresponding to the last row in the decomposition), which would have norm 1.

## 3 Random preconditioner

Now that we got an idea of the algorithm, we justify the choice of $R$ as a preconditioner for $Ax = b$, where $R$ is given by $\mathcal{S}A = QR$, the QR-decomposiotion of the row-sampled matrix $A$.

In particular, we study the relationship between the coherence, the sample size, and the condition number of the preconditioned system, which is described by the following theorem [1, Theorem 3.2].

**Theorem 1.** *Let $A$ be an $m \times n$ full rank matrix, and let $\mathcal{S}$ be a random sampling operator that samples $r \geq n$ rows from $A$ uniformly. Let $\tau = C\sqrt{m\mu(A)\log(r)/r}$, where $C$ is some constant defined in the proof. Assume that $\delta^{-1}\tau < 1$. With probability of at least $1-\delta$, the sampled matrix $\mathcal{S}A$ is full rank, and if $\mathcal{S}A = QR$ is a reduced $QR$ factorization of $\mathcal{S}A$, we have*

$$\kappa(AR^{-1}) \leq \frac{1+\delta^{-1}\tau}{1-\delta^{-1}\tau}. \tag{3.1}$$

We prove this result in three steps. First we bound $||I_n - \frac{m}{r}Q^T\mathcal{S}^T\mathcal{S}Q||_2$ with high probability, following the proof in [2, Lemma 4]. Then, we bound $\kappa(\mathcal{S}Q)$ with high probability using a Rayleight quotient argument. Finally, using the following theorem [3, Theorem 1], we show that $\kappa(AR^{-1}) = \kappa(\mathcal{S}Q)$.

**Theorem 2.** *Suppose that $l, m$ and $n$ are positive integers such that $m \geq l \geq n$. Suppose further that $A$ is a full-rank $m \times n$ matrix and the SVD of $A$ is*

$$A_{m\times n} = U_{m\times n}\Sigma_{n\times n}V^*_{n\times n}. \tag{3.2}$$

*Suppose in addition that $T$ is an $l \times m$ matrix such that the $l \times n$ matrix $TU$ has full rank. Then, there exists an $n \times n$ matrix $P$, and an $l \times n$ matrix $Q$ whose columns are orthonormal, such that*

$$T_{l\times m}A_{m\times n} = Q_{l\times n}P_{n\times n}. \tag{3.3}$$

*Furthermore, if $P$ is any $n \times n$ matrix, and $Q$ is any $l \times n$ matrix whose columns are orthonormal, such that $P$ and $Q$ satisfy (3.3), then the condition numbers of $AP^{-1}$ and $TU$ are equal.*

Before giving the proof to Theorem 1, we state the following theorem [2, Theorem 4], which will be used in the first step of the proof.

**Theorem 3.** *Let $A \in \mathbb{R}^{m\times n}$ with $||A||_2 \leq 1$. Construct $C$ using the* Exactly*(c) algorithm (Algorithm 1) and let the sampling probabilities $p_i$ satisfy*

$$p_i \geq \beta \frac{||A^{(i)}||_2^2}{||A||_F^2} \tag{3.4}$$

*for all $i \in [n]$ for some constant $\beta \in (0,1]$). Let $\epsilon \in (0,1)$ be an accuracy parameter and assume $||A||_F^2 \geq 1/24$. If*

$$c \geq \frac{96||A||_F^2}{\beta\epsilon^2} \log\left(\frac{96||A||_F^2}{\beta\epsilon^2\sqrt{\delta}}\right), \tag{3.5}$$

*then, with probability at least $1-\delta$,*

$$||AA^T - CC^T||_2 \leq \epsilon. \tag{3.6}$$

We remark that (3.5) only comes out from Theorem 3 as a sufficient condition to grant

$$\frac{c}{\log(2c/\sqrt{\delta})} \geq \frac{48}{\beta\epsilon^2}||A||_F^2, \tag{3.7}$$

which is the condition required in its proof and which will be easier to work with.

**Algorithm 1** Row sampling algorithm

**procedure** Exactly$(c, A, \{p_i\}_{i\in n})$ ▷ $A \in \mathbb{R}^{m\times n}, p_i \geq 0$ s.th. $\sum_{i\in[n]} p_i = 1$, $c \leq n$ psitive int.
  Initialize $S \in \mathbb{R}^{m\times c}$ to be an all-zero matrix.
  **for** $t = 1, \ldots, c$ **do**
    Randomly pick $i_t \in [n]$, with $\mathbb{P}\{i_t = i\} = p_i$;
    $S_{i_t t} \leftarrow 1/\sqrt{cp_{i_t}}$
  **end forreturn** $C = AS$
**end procedure**

We can now prove the main theorem.

*Proof of Theorem 1.* We proceed with the three step procedure we stated earlier.

**Step 1.** We prove that $||I_n - \frac{m}{r}Q^T\mathcal{S}^T\mathcal{S}Q||_2 \leq \delta^{-1}\tau$ with probability at least $1-\delta$. We follow the idea of the proof of [2, Lemma 3]. We note that $Q^T\mathcal{S}^T\mathcal{S}Q$ is an approximation of the product $Q^TQ$ and, in particular $||Q||_2 = 1$ and $||Q||_F = \sqrt{n}$. Then, let $\beta = n/(m\mu(A))$ and observe

$$\beta\frac{||Q_{(i)}^2||_2}{||Q||_F^2} = \frac{n||Q_{(i)}^2||_2}{m\mu(A)n} \leq \frac{1}{m} \qquad \forall i \in [n], \tag{3.8}$$

since $||Q_{(i)}^2||_2 \leq \mu(A)$ by definition of the coherence. Then, we would like to apply Theorem 3 with this choice of $\beta$ and bound

$$||I_n - \frac{m}{r}Q^T\mathcal{S}^T\mathcal{S}Q||_2 = ||Q^TQ - \frac{m}{n}Q^T\mathcal{S}^T\mathcal{S}Q||_2 \leq \epsilon = \delta^{-1}\tau.$$

We thus verify Theorem 3 conditions, with $A \leftarrow Q^T$, $C \leftarrow \sqrt{m/r}\mathcal{S}^T$, $\epsilon \leftarrow \delta^{-1}\tau$, $c \leftarrow r/m$ and inverting the roles of $m$ and $n$.

- $Q^T \in \mathbb{R}^{n\times m}$, $||Q^T||_2 \leq 1$ since $Q$ is orthonormal by definition.
- $p_i = 1/m$ for all $i \in [m]$, and by (3.8) we have
$$p_i = 1/m \geq \beta \frac{||Q^2_{(i)}||_2}{||Q||_F^2}.$$
- It is evident that $\sqrt{m/r}\mathcal{S}^T$ is a realization of $\textsc{Exaclty}(c)$, with $c = r/m$ and probabilities $p_i = 1/m$, $\forall i \in [m]$.
- $||Q^T||_F^2 = ||Q||_F^2 = n > 1/24$.
- We now verify (3.7), which will grant (3.5). We plug in our values and observe
$$\frac{48}{\beta\epsilon^2}||Q^T||_F^2 = \frac{48nm\mu(A)\delta^2 r}{nC^2 m\mu(A)\log(r)} = \frac{48\delta^2}{C^2}\frac{r}{log(r)}, \tag{3.9}$$
and
$$\frac{c}{\log(2c/\sqrt{\delta})} = \frac{r}{m\log\left(\frac{2r}{m\sqrt{delta}}\right)} \geq \frac{r}{\log(r)} \tag{3.10}$$
if $m\sqrt{\delta} \geq 2$. Combining (3.9) and (3.10), we have that (3.7) holds if $48\delta^2/C^2 \leq 1/m$ or equivalently if $\delta^2 m \leq C^2/48$. This holds for $\delta$ small enough and $\delta^{-1}\tau < 1$ still holds by choosing $r$ big enough.

Therefore, we just saw that all conditions work, so we can apply Theorem 3 and get that, with probability at least $1-\delta$,
$$||I_n - \frac{m}{r}Q^T\mathcal{S}^T\mathcal{S}Q||_2 = ||Q^TQ - \frac{m}{n}Q^T\mathcal{S}^T\mathcal{S}Q||_2 \leq \delta^{-1}\tau.$$

**Step 2.** We now use the result of Step 1 to prove that $\kappa(\mathcal{S}Q) \leq \frac{1+\delta^{-1}\tau}{1-\delta^{-1}\tau}$ with probability at least $1-\delta$. First, we recall that for any matrix $M$, we have that $||M||_2 = \sigma_{max}(M)$, its greater singular value. Then, we use the Rayleight-quotient definition of the eigenvalues to write
$$\begin{aligned}
\kappa(\mathcal{S}Q) = \frac{\sigma_{max}(\mathcal{S}Q)}{\sigma_{min}(\mathcal{S}Q)} &= \frac{\max_{||v||_2=1} v^TQ^T\mathcal{S}^T\mathcal{S}Qv}{\min_{||v||_2=1} v^TQ^T\mathcal{S}^T\mathcal{S}Qv} = \\
&= \frac{\max_{||v||_2=1} \frac{m}{r}v^TQ^T\mathcal{S}^T\mathcal{S}Qv - 1 + 1}{\min_{||v||_2=1} \frac{m}{r}v^TQ^T\mathcal{S}^T\mathcal{S}Qv - 1 + 1} = \\
&= \frac{1 - \min_{||v||_2=1} v^T\left(I_n - \frac{m}{r}v^TQ^T\mathcal{S}^T\mathcal{S}Q\right)v}{1 - \max_{||v||_2=1} v^T\left(I_n - \frac{m}{r}v^TQ^T\mathcal{S}^T\mathcal{S}Q\right)v} \leq \\
&\leq \frac{1 + ||I_n - \frac{m}{r}v^TQ^T\mathcal{S}^T\mathcal{S}Q||_2}{1 - ||I_n - \frac{m}{r}v^TQ^T\mathcal{S}^T\mathcal{S}Q||_2} \leq \frac{1+\delta^{-1}\tau}{1-\delta^{-1}\tau},
\end{aligned}$$
where in the last line, we use the fact that $I_n - \frac{m}{r}v^TQ^T\mathcal{S}^T\mathcal{S}Q$ is diagonal and thus its singular values correspond to eigenvalues.

**Step 3.** We now apply Theorem 2 to the decomposition $\mathcal{S}A = QR$ to get that
$$\kappa(AR^{-1}) = \kappa(\mathcal{S}Q).$$
$\mathcal{S}Q$ is full rank with high probability as its spectrum is close to the identity by Step 1 and all other hypothesys hold by construction, thus concluding the proof. □

# 4 The algorithm

We implement in MATLAB the Blendnpik algorithm [1, Algorithm 1], using Discrete Cosine Transform (DCT) for the row-mixing process and MINRES instead of LSQR. This choice moves our focus on solving the normal equation $A^TAx = A^Tb$. Furthermore, instead of just preconditioning on the right as in the original algorithm, we have to perform a split preconditioning, using $M = R^TR$. The matlab code is hereby presented.

```
A_ = A' * A;
b_ = A' * b;

[m1, n] = size(A);
m = ceil(m1/1000)*1000;
M = zeros(m,n);
M(1:m1,:) = A;

for iteration = 1:3
        D = randi([0,1],m,1)*2 -1;
        M = dct(D .* M);
        S = rand(m,1) < (gamma * n / m);
        [~,R] = qr(S .* M, 0);

        k = rcond(R);
        if 1 / k > 5 * eps
                x = minres(A_, b_, 1e-9, 1000, R', R);
                return
        end
end
```

The matrix $\mathcal{S}$ corresponds to a diagonal matrix whose elements are 0 with probability $\gamma n/m$ and 1 with p. $1 - \gamma n/m$. It corresponds to the sampling operator and it is included to simplify the QR-decomposition, as the change of some rows with full zeros make the decomposition faster.

# 5 Running times and convergence

We now test the improvement apported by the preconditioning to the running time of the solver and investigate how many row samples we need to obtain the solution. In particular, we start by finding which $\gamma$, the parameter that determines the sampling ratio, would be optimal with the two following matrices $A, B \in \mathbb{R}^{20000\times400}$.

1. $A$: Incoherent, ill-conditioned matrix:

```
rng(11);
U = orth(rand(20000,400));
S = diag(linspace(1,1e5,400));
V = orth(rand(400));
A = U*S*V';
```

2. $B$: Coherent, ill-conditioned matrix:

```
rng(11);
A = [ diag(linspace(1,e5,400)); zeros(19600,400) ];
A = A + 1e-8*ones(20000,400);
```

In order to get a good estimate of the optimal $\gamma$, we run 7 independent tests for each parameter gamma between 1.5 up to 10. We set the tolerance on the residual to $10^{-14}$ for MINRES, but we remark that often the method stagnates with a relative residual of $10^{-10}$. Our results are shown in Figure 1, where we can see that the optimal $\gamma$ is 2.5 for the inchoerent matrix and 5 for the coherent one. Furthermore, the running time for incoherent matrix seems to increase with the augmenting sample size, while the coherent one is slower in the beginning and then stabilize. This is in line with our expectations, as the incoherent one is well approximated by a small subsample of the rows, while to get a good approximation of the coherent one, we need the more samples. These results are similar to those obtained in the original paper [1], where they state that $\gamma = 4$ is almost optimal with most matrices.

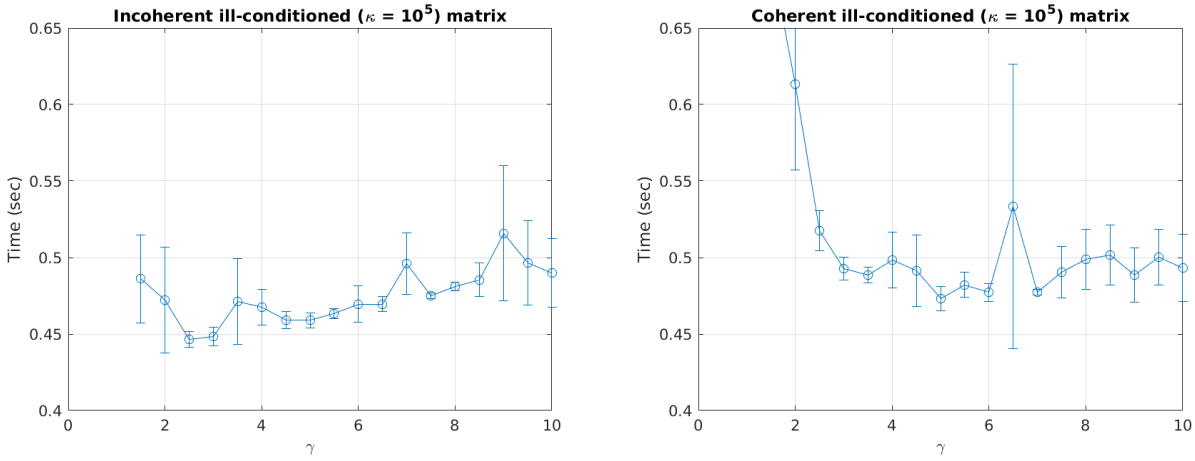


Figure 1: Blendenpick running time as a function of sample size $\gamma$ on both an inchoerent, left, and coherent, right, matrix of size $20000 \times 400$. In both cases the matrix is ill-conditioned ($\kappa = 10^5$).

We then compare the inner convergence of MINRES for the two matrices, computing the preconditoner with sample ratio $\gamma = 4$, the default value for Blendenpick implementation [1]. The reltive residuals at each iteration are shown in Figure 2. We see that the Incoherent matrix has a faster convergence, while the Coherent one is slighlty slower.

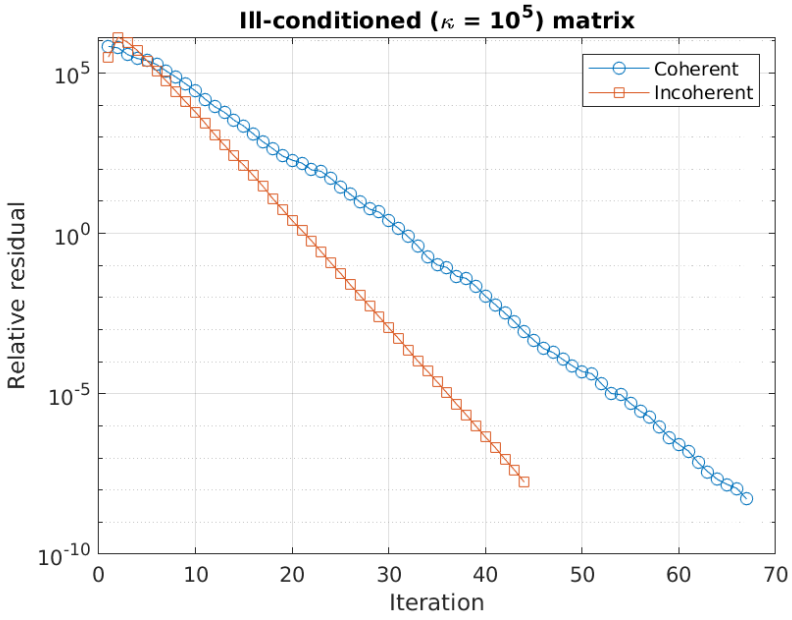


Figure 2: Inner convergence of MINRES at each iteration for the two ill-conditioned matrices.

## References


[1] Haim Avron, Petar Maymounkov, and Sivan Toledo. Blendenpik: Supercharging lapack's least-squares solver. *SIAM Journal on Scientific Computing*, 32(3):1217–1236, 2010.

[2] Petros Drineas, Michael W Mahoney, Shan Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *Numerische mathematik*, 117(2):219–249, 2011.

[3] Vladimir Rokhlin and Mark Tygert. A fast randomized algorithm for overdetermined linear least-squares regression. *Proceedings of the National Academy of Sciences*, 105(36):13212–13217, 2008.