Randomized Least Squares

Maximo Cravero Baraja
Ecole Polytechnique Federale de Lausanne
maximo.craverobaraja@epfl.ch

Abstract

This report aims to provide an overview of the Blendenpik algorithm, which is a randomized version of a least squares solver geared towards large scale, highly overdetermined systems of the form $\mathrm{Ax} = \mathrm{b}$. Theoretical and practical results will be used to illustrate the importance of coherence and row-mixing to enable a uniform sampling of the rows which guarantees high quality solutions with high probability.

1 Background

The setting which is considered throughout this report is that of a highly overdetermined system $Ax = b$, with $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $m \gg n$. The aim is to obtain the optimal value of x, namely x^*, in the least squares sense i.e. $x^* = argmin_{x \in \mathbb{R}^n} ||b - Ax||_2$. Traditional least squares solvers such as LAPACK make use of a QR decomposition, where we can write:

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix} = [Q_1 \quad Q_2] \begin{bmatrix} R \\ 0 \end{bmatrix} \qquad Q \in \mathbb{R}^{m \times m}, Q_1 \in \mathbb{R}^{m \times n}, Q_2 \in \mathbb{R}^{m \times m-n}, R \in \mathbb{R}^{n \times n}$$

Q is an orthogonal matrix (or unitary in case A is complex), i.e. its rows and columns are orthonormal: $QQ^T = Q^TQ = I \implies Q^T = Q^{-1}$. R is an upper triangular matrix (in this case because $m > n$ and because we assume A is full rank). The linear system can thus be rewritten by using $Ax = b \implies Q^TAx = Q^Tb$, meaning:

$$||x - Ab||_2 = ||Q^Tb - Q^TAx||_2 = \left|\left| \begin{bmatrix} Q_1^Tb - Rx \\ Q_2^Tb \end{bmatrix} \right|\right|$$

This holds because $Q^TA = R$. Thus minimizing in the least squares sense using the QR decomposition boils down to solving $x = R^{-1}Q_1^Tb$. The total cost associated with solving the least squares problem in this manner consists of the cost of a QR decomposition of A, and the matrix inverse and multiplications to compute x. Since m is assumed to be large in these settings, especially given the increasing prevalence of big data applications, this can be extremely costly.

The objective of the Blendenpik algorithm is to apply uniform random sampling and random projections to define a preconditioner which reduces the number of number of iterations and thus computational cost associated with solving large scale systems using methods such as MINRES or LSQR. This requires careful consideration of the implications of random row sampling, as well as preprocessing steps to ensure a good solution can still be obtained with high probability.

A natural attempt at reducing the cost of solving an overdetermined system would be to sample a certain number of rows $\mathcal{R}$ and obtain the solution $x_{\mathcal{R}}^* = argmin_{x \in \mathbb{R}^n} ||A_{\mathcal{R}}x - b_{\mathcal{R}}||$. Under certain conditions on A and $\Omega(n \log(m) \log(n \log(m)))$ uniformly randomly sampled rows, this leads to a residual such that $||Ax_{\mathcal{R}} - b||_2 / ||Ax_{opt} - b||_2 \leq 1 + \epsilon$. This has two problems (1):

1. Standard stability analysis of linear least squares algorithms is done in terms of backward error.

2. Running time depends on ϵ^{-1}

An alternative approach is to define a preconditioner based on the uniformly sampled rows, denoted by $A_{\mathcal{R}}$, and use this in conjunction with a standard iterative method such as LSQR or MINRES. However, depending on the matrix A, the resulting subsampled matrix may be ill-conditioned/ rank deficient, leading to failure in the iterative subspace method. Such an approach works *only* if the coherence of the matrix is small. This can be enforced via a randomized row-mixing preprocessing phase. Before further elaborating on these randomized sampling and row-mixing approaches, it is necessary to define some terminology. This will then be used to derive bounds on the condition number of the preconditioned overdetermined system.

2 Key Concepts

2.1 Coherence

Coherence defines a notion of the degree to which the importance of the rows is concentrated onto a small number of rows. This is undesirable in the case of uniform sampling as we would need to sample a large number of rows to ensure these are picked. Therefore we prefer matrices with 'mixed' information, i.e. all rows are more or less equally important. Such matrices have a low coherence value (lower bounded by $\frac{n}{m}$), whereas badly behaved matrices have high coherence values (upper bounded by 1). Given a matrix $A \in \mathbb{R}^{m\times n}$ (where $m > n$) and a matrix U such that it forms an orthonormal basis of the column space of A, the coherence is defined as:

$$\mu(A) = \max_i ||U_{i,:}||_2^2$$

Lemma 1. *The coherence $\mu(A)$ of a matrix $A \in \mathbb{R}^{m\times n}$ has the following bounds:*

$$\frac{n}{m} \leq \mu(A) \leq 1$$

Proof. First we prove the upper bound. We can define a basis with orthonormal columns by using the QR decomposition, namely $A = QR = [Q_1 Q_2]R$. Since Q is orthogonal, we know that the row norms are all equal to 1. Further, we only require the first n columns from Q to span the column space of A, which corresponds to the Q_1 matrix. Consequently we see that the row norms are now *at most* 1, i.e. $\mu(A) \leq 1$. Now to prove the lower bound of the coherence, we consider the Frobenius norm of Q_1. The squared Frobenius norm of Q_1 is $||Q_1||_F^2 = \sum_{i\in[m],j\in[n]} Q_{1_{i,j}}^2 = n$, which follows from the fact that the vector 2-norm of each column is equal to 1. If $\mu(A) < \frac{n}{m}$, that is, the maximum squared row norm is less than $\frac{n}{m}$, then that implies that $||Q_1||_F^2 < \sum_{i\in[m]} \frac{n}{m} = n$, which is a contradiction as $||Q_1||_F^2 = n$. Thus we see that $\mu(A) \geq \frac{n}{m}$ and conclude that $\frac{n}{m} \leq \mu(A) \leq 1$. □

Generally speaking the coherence number of randomly generated matrices is quite low. In a basic experiment that was performed using $A_i \in \mathbb{R}^{1000}$, $i \in [1000]$, an average coherence of 0.072 was obtained, which is quite good seeing as the lower bound on such a matrix is $\frac{50}{1000} = 0.05$.

```
rand('state', 42);
n_matrices = 1000; n_rows = 1000; n_cols = 50;

% b) average coherences
coherences = [];
for n=1:n_matrices
    A = rand(n_rows, n_cols);
    if ~(rank(A) == n_cols)
        fprintf("rank deficient random matrix with rank: %i\n", rank(A));
    end
    [Q, R] = qr(A, 0);
    coherence = max(sum(Q .^ 2, 2));
    coherences = [coherences, coherence];
end
```

```
mean_coherence = mean(coherences);
fprintf("Mean coherence of random matrices: %d\n", mean_coherence);
```

In order to obtain the worst possible coherence, one can simply define a rank deficient matrix. For instance, in the case of the random matrices $A_i \in \mathbb{R}^{1000\times 50}$, taking any one of these matrices and setting the first column to $[0, ..., 0]^T$ will yield a QR decomposition for which the first column of Q will be $[1, 0, ..., 0]^T$, meaning the coherence is automatically equal to 1 i.e. the worst possible value.

2.2 Unitary Seed Transforms - Discrete Cosine Transform

Todo

2.3 Randomized Row-Mixing

Randomized row-mixing preprocessing phase to reduce coherence of a matrix. This enables us to apply uniform sampling to the mixed matrix, as this only works well for matrices with low coherence values.

Todo

3 Path to Blendenpik

Noting these definitions, we can formally prove bounds on a matrix A preconditioned via randomized techniques. General thought process of the procedure:

1. Given an overdetermined system $Ax = b$
2. Don't know (or don't want to compute) its coherence number, but we still want to apply random uniform sampling approach to avoid computing QR decomposition of original A matrix. For this we need to define a good preconditioner which results in a good condition number $\kappa(AR^{-1})$.
3. To guarantee a good coherence number, we apply a unitary and randomized seed transform $\mathcal{F} = FD$, where F is a unitary matrix and D is a diagonal matrix with IID Rademacher random variables.
4. Given our incoherent (low coherence), randomly row-mixed matrix $\mathcal{F}A$, we can now apply uniform random row sampling and obtain a matrix $\tilde{A}$ with a good condition number with high probability. Now we can compute the QR decomposition $\tilde{A} = QR$ and use R as a preconditioner for A with a standard method like MINRES or LSQR.

3.1 Proofs

Since the point of the Blendenpik algorithm is to reduce the required number of iterations until convergence via a suitable preconditioner, it is useful to define an upper bound on the condition number $\kappa(AR^{-1})$, which is guaranteed with high probability. Todo: finalize proof, refer to Moritz's report for current progress

Theorem 2. *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 a constant. Further assume $\delta^{-1}\tau < 1$. With probability 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*

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

Theorem 7 from (2) and Theorem 1 from (3) are used in this proof.

Theorem 3. *(3) 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 that the SVD of A is*

$$A_{m\times n} = U_{m\times n}\Sigma_{n\times n}V_{n\times n*}$$

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

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

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 the latter equation, then the condition numbers of AP^{-1} and TU are equal.

Theorem 4. *(2) Suppose $A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{n \times p}$, and $c \leq n$. Construct C and R with Algorithm 6, using the `EXPECTED(c)` algorithm. If the sampling probabilities $\{p_i\}_{i=1}^{n}$ used by the algorithm are of the form (44) or (45), then*

$$\mathbf{E}\left[\|AB - CR\|_F\right] \leq \frac{1}{\sqrt{\beta c}}\|A\|_F\|B\|_F$$

If, in addition, $B = A^T$, then

$$\mathbf{E}\left[\left\|AA^T - CC^T\right\|_2\right] \leq O(1)\sqrt{\frac{\log c}{\beta c}}\|A\|_F\|A\|_2.$$

3.2 Algorithm

We apply the Blendenpik algorithm using two different iterative solvers, namely MINRES and LSQR. In the case of MINRES we are required to use the normal equations $A^T Ax = A^T b$ as a symmetric, square matrix is required, unlike for LSQR.

Algorithm 1 Blendenpik Algorithm (DCT)

```
$x = \mathbf{blendenpik}(A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^n, \gamma)$
$\tilde{m} \leftarrow \lceil m/1000 \rceil \times 1000$
$M \leftarrow \begin{bmatrix} A \\ 0 \end{bmatrix} \in \mathbb{R}^{\tilde{m} \times n}$
while true
    1. $M \leftarrow DCT(DM)$, where $D \in \mathbb{R}^{\tilde{m} \times \tilde{m}}$ is diagonal with $D_{ii} = \pm 1$ with equal probability
    2. Define diagonal matrix $\mathcal{S} \in \mathbb{R}^{\tilde{m} \times \tilde{m}}$ with $S_{ii} = \begin{cases} 1 & \text{with probability } \gamma n/\tilde{m} \\ 0 & \text{with probability } 1 - \gamma n/\tilde{m} \end{cases}$
    3. Compute reduced $QR$ decomposition $\mathcal{S}M = QR$ with $R \in \mathbb{R}^{n \times n}$
    4. if $\kappa_{est}(R)^{-1} > 5\epsilon_{machine}$
        (a) $x \leftarrow \begin{cases} \text{MINRES applied to } A^T A, A^T b \text{ with } R^{-T} \text{ and } R^{-1} \text{ as left and right preconditioners} \\ \text{LSQR applied to } A, b \text{ using R as a preconditioner} \end{cases}$
        (b) return
    5. endif
end while
```

4 Experiments

We define matrices A_1 with:

```
rng(11);
U = orth(rand(20000,400));
S = diag(linspace(1,1e5,400));
V = orth(rand(400));
A1 = U*S*V';
```

Matrix A_2 is given by:

```
rng(11);
A2 = [ diag(linspace(1,e5,400)); zeros(19600,400) ];
A2 = A2 + 1e-8*ones(20000,400);
```

In the following figures we show the required number of iterations to converge for different values of γ. The b vector is randomly initialized.

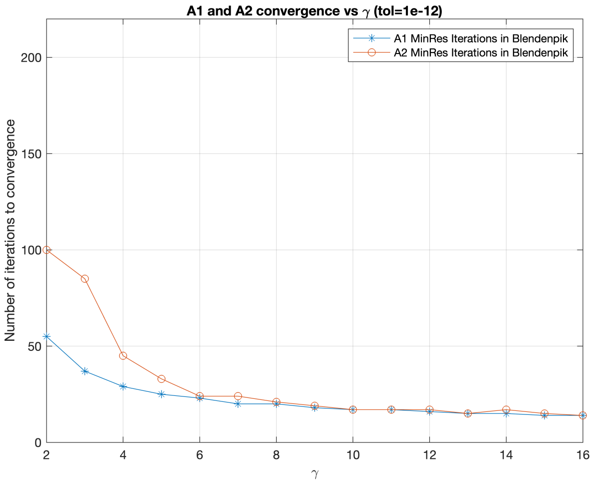

Figure 1: Convergence of A_1 and A_2 as function of γ

The following figure investigates the convergence of the inner LSQR/MINRES steps for the two matrices defined above. Generate these plots

References

[1] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós, “Faster least squares approximation,” *CoRR*, vol. abs/0710.1435, 2007. [Online]. Available: http://arxiv.org/abs/0710.1435

[2] P. Drineas, M. Mahoney, and S. Muthukrishnan, “Relative-error cur matrix decompositions,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, pp. 844–881, 05 2008.

[3] V. Rokhlin and M. Tygert, “A fast randomized algorithm for overdetermined linear least-squares regression,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, pp. 13 212–7, 09 2008.

A Appendix

Optionally include extra information (complete proofs, additional experiments and plots) in the appendix. This section will often be part of the supplemental material.