CS 6220

Lecture 7: Blendenpik

February 11, 2020

Lecturer: Anil Damle *Scribe: Ariel Kellison*

1 Introduction

Blendenpik is a randomized least squares solver that uses an iterative least squares algorithm (LSQR) with a preconditioner obtained from random projection methods. The random projection preconditioner generated by Blendenpik yields faster convergence of LSQR than LSQR without a preconditioner, even for ill-conditioned systems. Furthermore, on large matrices, Blendenpik is substantially faster than LAPACK.

1.1 Motivation

Consider the linear least squares problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2, \tag{1}$$

for a large highly overdetermined system: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $m >> n$, and $rank(A) = n$. The unique minimum length solution to (1) is determined by computing x^*:

$$x^* = \text{arg}\min \|x\|_2 \text{ subject to } x \in \text{arg}\min_z \|Az - b\|_2 \tag{2}$$

Given the problem description it is reasonable to wonder if there is superfluous information in the system. In particular, can a problem of smaller size be generated from a subset $\mathcal{S}$ of the rows of A such that

$$x_{\mathcal{S}} = \text{arg}\min_x \|A(\mathcal{S}, :)x - b(\mathcal{S})\|_2 \tag{3}$$

is a good approximation to x^*? Blendenpik achieves such an approximation by generating a problem of smaller size using a unitary transformation (i.e. a Walsh–Hadamard transform, a discrete cosine transform, or a discrete Hartley transform) for blending the rows/column of A, and then taking a random sampling of the rows of the result.

2 Blendenpik

The essence of the Blendenpik algorithm given problem (1) can be summarized in three steps:

1. Pick $\mathcal{S}$ of size r, a subset of the rows of A such that $m \geq r \geq n$

2. Let $A(\mathcal{S},:) = QR$

3. Use R^{-1} as a right preconditioner for (1) in LSQR

The convergence rate of LSQR is related to the condition number of the normal equations matrix A^TA; in many applications, this matrix is ill-conditioned. LSQR applied to a preconditioned system where the right preconditioner R is determined such that $A(:,:) = QR$, converges in a single iteration: the preconditioned normal equations matrix is $(AR^{-1})^T(AR^{-1}) = Q^TQ = I$. To obtain a feasible preconditioner for LSQR, Blendenpik takes a uniform random sampling of the *blended* rows of A.

2.1 Uniform Sampling (Row Picking)

Uniform random sampling of the rows of A is a suitable approach for generating preconditioners when the solution is relatively independent of specific rows of A. The *coherence* of a matrix provides a measure of the dependence of the solution on specific rows. Thus, uniform random sampling of the rows of A is a suitable approach only if the coherence of A is small (that is, when A is incoherent).

Definition 2.1. Let A be an $n \times n$ full rank matrix and let U be an $n \times n$ matrix whose columns form an orthonormal basis for the column space of A. The *coherence* of A is defined as

$$\mu(A) = \max_i \|U(i,:)\|_2^2,$$

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

While the coherence of A is independent of the condition number of A, there is a relationship between the coherence of A, the sample size (r) of the rows of A, and the condition number of the preconditioned system [1]:

Theorem 2.2. *Let A be an $m \times n$ full rank matrix and let S be a subset of $[m]$ of size $r \geq n$ chosen uniformly at random such that $rank(A(\mathcal{S},:)) = n$. Let $\tau = C\sqrt{\frac{m\mu(A)log(r)}{r}}$ where C is a constant defined in the proof. For some $\delta \in (0,1)$, assume that $\delta^{-1}\tau < 1$. With probability of at least $1-\delta$, the sampled matrix $A(\mathcal{S},:)$ is full column rank, and if $A(\mathcal{S},:) = QR$ is a reduced QR factorization of $A(\mathcal{S},:)$, then*

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

Before proving Theorem 2.2, a relationship between the condition number of the preconditioned system and the condition number of a subsampling of an orthonormal basis of A must be established:

Theorem 2.3. *Suppose that r, m, and n are postive integers such that $m \geq r \geq n$. Additionally, suppose that the SVD of $A \in \mathbb{R}^{m\times n}$ is*

$$A_{m\times n} = U_{m\times n}\Sigma_{n\times n}V_{n\times n}^T.$$

Let S be a subset of $[m]$ of size r and suppose that the matrix $U(\mathcal{S}, :) \in \mathbb{R}^{r\times n}$ has full rank. Then there exists a matrix $Q \in \mathbb{R}^{r\times n}$ with orthonormal columns and a matrix $R \in \mathbb{R}^{n\times n}$ such that

$$A(\mathcal{S}, :) = Q_{r\times n}R_{n\times n}$$

and the condition numbers of AR^{-1} and $U(S, :)$ are equal.

The proof of Theorem 2.3 follows from Lemma 2.4.

Lemma 2.4. *Suppose that r, m, and n are postive integers such that $m \geq r \geq n$. Additionally, suppose that the SVD of $A \in \mathbb{R}^{m\times n}$ is*

$$A_{m\times n} = U_{m\times n}\Sigma_{n\times n}V_{n\times n}^T. \tag{4}$$

Let S be a subset of $[m]$ of size r and suppose that the SVD of the matrix $U(\mathcal{S}, :) \in \mathbb{R}^{r\times n}$ is

$$U(\mathcal{S}, :) = \tilde{U}_{r\times n}\tilde{\Sigma}_{n\times n}\tilde{V}_{n\times n}^T. \tag{5}$$

Then there exists a matrix $Q \in \mathbb{R}^{r\times n}$ with orthonormal columns and a matrix $R \in \mathbb{R}^{n\times n}$ such that

$$A(\mathcal{S}, :) = QR.$$

Finally, if $rank(A) = rank(U(\mathcal{S}, :)) = n$, then there exists a unitary matrix $W \in \mathbb{R}^{n\times n}$ such that

$$R = W_{n\times n}\tilde{\Sigma}_{n\times n}\tilde{V}_{n\times n}^T\Sigma_{n\times n}V_{n\times n}^T. \tag{6}$$

Proof. (Of Theorem 2.3) The proof of the existence of matrices R, Q, and W follow from Lemma 2.4; interested readers can refer to Lemma 3 in [4] for the proof of Lemma 2.4. From (4), (6), and the fact that V, $\tilde{V}$ and W are all unitary, it follows that

$$A_{m\times n}R_{n\times n}^{-1} = U_{m\times n}\tilde{V}_{n\times n}\tilde{\Sigma}_{n\times n}W_{n\times n}^T,$$

$$\|(A_{m\times n}R_{n\times n}^{-1})^T(A_{m\times n}R_{n\times n}^{-1})\| = \|\tilde{\Sigma}_{n\times n}\|_2^2,$$

and

$$\|((A_{m\times n}R_{n\times n}^{-1})^T(A_{m\times n}R_{n\times n}^{-1}))^{-1}\| = \|\tilde{\Sigma}_{n\times n}\|_2^2.$$

Furthermore, from (5), it follows that

$$\|U(\mathcal{S}, :)\|_2 = \|\tilde{\Sigma}_{n\times n}\|_2^2.$$

□

Finally, the proof of Theorem 2.2 follows from Lemma 2.5 [2].

Lemma 2.5. *Let U be an $m \times n$ matrix whose columns are orthonormal, and let S be a subset of of $[m]$ of size $r \leq m$ chosen uniformly at random. Then,*

$$E(\|I_{n\times n} - \frac{m}{r}U(S,:)^T U(S,:)\|_2) \leq C\sqrt{\frac{m\mu(U)log(r)}{r}}$$

for some constant C.

Proof. (of Theorem 2.2) Lemma 2.5 yields

$$E(\|I_{n\times n} - (\frac{m}{r})U(S,:)^T U(S,:)\|_2) \leq \tau.$$

Using Markov's inequality it then follows that

$$Pr(\|I_{n\times n} - (\frac{m}{r})U(S,:)^T U(S,:)\|_2 \geq \delta^{-1}\tau) \leq \delta.$$

Thus, with probability $1 - \delta$,

$$\|I_{n\times n} - (\frac{m}{r})U(S,:)^T U(S,:)\|_2 < \delta^{-1}\tau < 1 \tag{7}$$

and $A(S,:)$ is full column rank. Note that if $A(S,:)$ is full-rank, then so is $U(S,:)$ and from Theorem 2.3, $\kappa(AR^{-1}) = \kappa(U(S,:))$. Now, the desired bound on $\kappa(U(S,:))$ follows from a Rayleigh quotient argument: Note that each eigenvalue of the matrix $M = (\frac{m}{r})U(S,:)^T U(S,:)$, is equal to the Rayleigh quotient $R(M,x)$ for some $x \neq 0$. In particular,

$$\begin{aligned}\lambda &= (\frac{m}{r})\frac{x^T x}{x^T x} \\ &= \frac{x^T x + x^T(M - I_{n\times n})x}{x^T x} \\ &= 1 - \eta,\end{aligned}$$

where $\eta = R(I_{n\times n} - M, x)$. From (7) and the observation that $I_{n\times n} - M$ is symmetric, it follows that $|\eta| < \delta^{-1}\tau$. Thus, the eigenvalues of M lie between $1 - \delta^{-1}\tau$ and $1 + \delta^{-1}\tau$. It follows that

$$\kappa(AR^{-1}) = \kappa(U(S,:)) \leq \sqrt{\frac{1 + \delta^{-1}\tau}{1 - \delta^{-1}\tau}}$$

as desired. □

2.2 Preprocessing (Row Blending)

Observe that Theorem 2.2 implies that, to obtain suitable preconditioners with high probability for highly coherent systems, it is necessary to sample a large number of rows of A. Unfortunately, it is both challenging to estimate the coherence of A and computationally impractical to sample a large number of rows of A. This problem can be avoided by blending the rows of A using a unitary transformation (i.e. a Walsh–Hadamard transform, a discrete cosine transform, or a discrete Hartley transform) to decrease the coherence of A while also preserving the condition number of A. In particular, for any unitary $G \in \mathbb{R}^{m \times m}$,

1. $\kappa(GAR^{-1}) = \kappa(AR^{-1})$,
2. right preconditioners for GAR^{-1} are also right preconditioners for AR^{-1}, and
3. in general, $\mu(GA) \neq \mu(A)$.

The following theorem provides a bound on the coherence of GA [1]. Interested readers will see that this is a generalization of Lemma 3 in [3].

Theorem 2.6. *Let $A \in \mathbb{R}^{m \times n}$ be a full rank matrix with $m \geq n$. Let $F \in \mathbb{R}^{m \times m}$ be a unitary matrix, and let D be a diagonal matrix with $Pr(D_{ii} = \pm 1) = \frac{1}{2}$. If $G = FD$, then with probability of at least 0.95*

$$\mu(GA) \leq Cn\eta log(m),$$

where $\eta = \max |F_{ij}|^2$ and C is some constant.

Thus, a suitable choice of G for blending the rows of A is one where the value of $\max |F_{ij}|^2$ is small. Note that, for the minimal value of $\frac{1}{m}$ for η, each entry of F must have a squared absolute value equal to $\frac{1}{m}$. A normalized Discrete Fourier Transform (DFT) matrix and a Walsh-Hadamard Transform (WHT) matrix satisfy this condition. Note that the WHT can be applied only if the number of rows of A is a power of two; padding A with zeros enables application of the WHT to smaller matrices. In practice, the choice of a DFT matrix for F has the drawback of operation-count and memory penalties because it involves complex numbers [1]. The Blendenpik algorithm outlined in Algorithm 1 will involve the WHT. Table 1 provides a brief comparison of the advantages and disadvantages of different row mixing strategies.

2.3 Algorithm and Computational Cost

The three phases (see Section 2) of the Blendenpik algorithm (see Algorithm 1) have different computational cost. The number of LSQR iterations required for convergence for matrices with $m >> n$ (e.g $n \sim \frac{m}{40}$) depends on the coherence and size of the matrix; for ease of comparison here, one can consider a matrix with $m \sim 4E4$. From Section 5 of [1], we see that approximately 40 (60) LSQR iterations are required for incoherent (coherent) matrices. Thus, the most expensive phase is LSQR: each iteration takes $\Theta(mn)$ time. The QR factorization of the blended matrix has the second highest cost, with running time $\Theta(n^3)$. Finally, the blending phase is the least costly, with running time $\Theta(mnlog(m))$.

Table 1: Comparison of the advantages and disadvantages of different row mixing strategies

Transformation	η value	Advantages	Disadvantages
DFT	1/m	Fast application	Poor op. count/ memory.
WHT	1/m	Theoretically optimal; Erratic on coherent matrices in practice.	Can only be applied if m % 2 = 0; Padding causes discontinuous increases in run time/ memory.
DCT	2/m	Exists for all vector sizes. Works well on coherent matrices in practice.	Application is slow (dependent on factorization of m).
DHT	2/m	Exists for all vector sizes. Works well on coherent matrices in practice.	Application is slow (dependent on factorization of m).

Algorithm 1: Blendenpik's Algorithm

```
x = blendenpik($A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^n$)
▷ $m \geq n,\ A$ is nonsingular
▷ parameters: $\gamma$ and transform type (set here as WHT)
// Pad $A$ appropriately for WHT.
$\tilde{m} \leftarrow 2^{\lceil log_2 m \rceil}$
M ← $\begin{bmatrix} A \\ 0 \end{bmatrix} \in \mathbb{R}^{\tilde{m}\times n}$
while not returned do
    // D is a diagonal matrix with ±1 on its diagonal with
       equal probability
    // $F_{\tilde{m}}$ is the transform type, set here as WHT
    M ← $F_{\tilde{m}}(DM)$
    Let $\mathcal{S} \in \mathbb{R}^{\tilde{m}\times\tilde{m}}$ be a random diagonal matrix: $\mathcal{S}_{ii} = \begin{cases} 1 & \text{with probability} \frac{\gamma n}{\tilde{m}} \\ 0 & \text{with probability} 1 - \frac{\gamma n}{\tilde{m}} \end{cases}$
       // Subsample the non-zero rows of $\mathcal{S}M$
    Y ← $\mathcal{S}M$
    Compute $R \in \mathbb{R}^{n\times n}$ where $Y = QR$
    $\kappa_r \leftarrow$ estimate $\kappa(R)$ using LAPACK
    if $\kappa_r > 5\epsilon_{machine}$ then
        x ← LSQR(A, b, R, $10^{-14}$)
        return $x$
    else
        if #iterations>3 then
            failure: compute $x$ using LAPACK
            return $x$
        end
    end
end
```

References

[1] Avron, Maymounkov, and Toledo. A fast randomized algorithm for overdetermined linear least-squares regression. *SIAM J. on Scientific Computing*, 32(3):1217–1236, 2010.

[2] Haim Avron. *Efficient and Robust Hybrid Iterative-Direct Multipurpose Linear Solvers*. PhD thesis, Tel-Aviv University, Israel, October 2010.

[3] Petros Drineas, Michael W. Mahoney, S. Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *Numer. Math.*, 117(2):219–249, February 2011.

[4] 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.