Randomized Iterative Methods for Matrix Approximation

Joy Azzam, Benjamin W. Ong, and Allan A. Struthers(✉)

Department of Mathematical Sciences, Michigan Technological University,
Houghton, USA
{atazzam,ongbw,struther}@mtu.edu

Abstract. Standard tools to update approximations to a matrix A (for example, Quasi-Newton Hessian approximations in optimization) incorporate computationally expensive one-sided samples AV. This article develops randomized algorithms to efficiently approximate A by iteratively incorporating cheaper two-sided samples $U^\top AV$. Theoretical convergence rates are proved and realized in numerical experiments. A heuristic accelerated variant is developed and shown to be competitive with existing methods based on one-sided samples.

Keywords: Matrix approximation · Randomized algorithms · Two-sided samples · Quasi-Newton

1 Introduction and Motivation from Optimization

Effective nonlinear optimization algorithms require 1st derivative information, $\nabla f(x)$, while superlinear convergence requires some 2nd derivative approximation [12]. For example, standard Quasi-Newton (QN) methods such as BFGS (complete gradient and an approximate Hessian generated from gradient differences) have superlinear terminal convergence. Limited-Memory (LM) QN methods [11], such as LBFGS, which approximate the Hessian efficiently by storing only the most recent gradient differences, are widely used in large-scale optimization. This article formulates randomized QN like algorithms which can be used to approximate Hessians (as well as general matrices) with reduced cost.

Alternatively, consider Stochastic Gradient Descent (SGD) [14], which is a common dimension reduction technique in statistics and Machine learning. SGD minimizes the average of cost functions $f_i : \mathbb{R}^m \to \mathbb{R}$,

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x),$$

by approximating $\nabla f \approx s^{-1}\sum_{i=1}^{s} \nabla f_i$. Here, f_i is associated with the i-th entry of a large (n entry) data or training set. The SGD approximation is simply $\nabla f \approx F\,p$, where F is the matrix with ith column ∇f_i and the sparse vector p has

© Springer Nature Switzerland AG 2022
G. Nicosia et al. (Eds.): LOD 2021, LNCS 13164, pp. 226–240, 2022.
https://doi.org/10.1007/978-3-030-95470-3_17

s non-zero entries of $1/s$ at sampled indices. Our algorithms generalize SGD by incorporating flexible sampling in $\mathbb{R}^n$ along with sampling in the parameter space $\mathbb{R}^m$ to provide flexibility to tune our algorithms to computational hardware.

Section 2 introduces the fundamental problem, two-sided samples and terminology. Section 3 reviews randomized one-sided Quasi-Newton algorithms while Sect. 4 develops our randomized, two-sided Quasi-Newton algorithms. Section 5 provides probabilistic convergence rates and error estimates. Section 6 numerically demonstrates the convergence of the algorithms. Section 7 incorporates an inner block power iteration to accelerate our two-sided algorithms and compares the result to one-sided algorithms based on similar heuristics.

2 Fundamental Problem, Samples, and Terminology

The fundamental problem our algorithms addresses is how to efficiently construct a sequence of approximations to a matrix, $A \in \mathbb{R}^{m\times n}$, from a stream of incomplete and possibly noisy data. Specifically, we develop and analyze algorithms to *iteratively* embed aggregate information from

$$U^\top A\,V \in \mathbb{R}^{s_1\times s_2}, \quad U \in \mathbb{R}^{m\times s_1}, \quad V \in \mathbb{R}^{n\times s_2}.$$

These weighted linear combinations of the rows and columns of the data A are called two-sided samples. This is in contrast to weighted linear combinations of the rows, $U^\top A$, or weighted linear combinations of the columns, $A\,V$, which we refer to as one-sided samples. Two-sided samples have been used before in non-iterative algorithms: [9] compares Schatten-p norm estimates (pth root of the sum of the pth power of the singular values) using two-sided samples, $U^\top A\,V$, (termed a bi-linear sketch) to estimates using one-sided samples, $A\,V$. Large eigenvalues estimates using two-sided random projectors are examined in [1]; and two-sided samples are used in [2] to tighten bounds on low-rank approximations. We follow their lead by simply counting sample entries to estimate data cost: $m\,s_2$ for the one-sided samples, $A\,V$, and $s_1\,s_2$ for the two-sided sample, $U^\top A\,V$. Algorithms using two-sided samples are a subset of randomized numerical linear algebra. An overview of existing algorithms and applications is provided by the extensive list of articles citing the comprehensive review [7]. The algorithms in [1,7] expend significant up front effort computing projections $\varOmega$ and $\varPsi$ so that the projected matrix, $\varOmega^\top A\,\varPsi$, approximates A on dominant eigenvalues with the goal that uniform random sampling of $\varOmega^\top A\,\varPsi$ yields good approximations to A. In contrast, our algorithms produce an improving sequence of approximations by iteratively embedding small randomized two-sided samples, $U^\top A\,V$, with sample dimensions $s_1 \times s_2$ that can be chosen to suit available hardware. Throughout we compare algorithms using the cost estimates (respectively $s_1\,s_2$ and $m\,s_2$ for the samples $U^\top A\,V$ and $A\,V$) from [9].

We use notation motivated by QN algorithms in non-linear optimization. SPD means symmetric positive definite and W is an SPD weight matrix. X^+ denotes the Moore-Penrose pseudo-inverse of X; $\langle X, Y\rangle_F = \mathrm{Tr}\left[\mathrm{X}^\top \mathrm{Y}\right]$ and

$\|X\|_F^2 = \langle X, X\rangle_F$ are the Frobenius inner product and norm. For conforming SPD weights W_1 and W_2 the weighted Frobenius norm (with special case $X = X^\top$ and $W = W_1 = W_2$ written $F(W^{-1})$) is

$$\|X\|^2_{F(W_1^{-1}, W_2^{-1})} = \|W_1^{-1/2} X\, W_2^{-1/2}\|_F^2.$$

The W-weighted projector $\mathcal{P}$, which projects onto the column space of $W\,U$,

$$\mathcal{P} = P_{W^{-1},U} = W\,U(U^\top\, W\,U)^{-1}U^\top, \tag{1}$$

satisfies $\mathcal{P}\,W = W\,\mathcal{P}^\top = \mathcal{P}\,W\,\mathcal{P}^\top$ and $W^{-1}\mathcal{P} = \mathcal{P}^\top W^{-1} = \mathcal{P}^\top\, W^{-1}\,\mathcal{P}$.

3 Randomized One-Sided Quasi-Newton Algorithms

Our iterative approximations to A using two-sided samples are motivated by the one-sided sampled algorithms in [6] and QN optimization algorithms. Classical QN schemes for SPD matrices A are formulated as constrained minimum change updates for $B \approx A$ or $H \approx A^{-1}$ in weighted Frobenius norms [12]: the constraint enforces the new information while the minimum change condition stabilizes the update. The one-sided sampled update algorithms in [5,6] are given by the KKT [8,12] equations (with particular choices of weight W) for the quadratic programs

$$B_{k+1} = \arg\min_B \left\{ \frac{1}{2}\|B - B_k\|^2_{F(W^{-1})} \mid B\,U_k = A\,U_k \text{ and } B = B^\top \right\}, \tag{2}$$

$$H_{k+1} = \arg\min_H \left\{ \frac{1}{2}\|H - H_k\|^2_{F(W^{-1})} \mid U_k = H\,A\,U_k \text{ and } H = H^\top \right\}. \tag{3}$$

The analytical updates defined by Eqs. (2) and (3), are

$$B_{k+1} = B_k + \mathcal{P}_B(A - B_k) + (A - B_k)\mathcal{P}_B^\top - \mathcal{P}_B(A - B_k)\mathcal{P}_B^\top, \tag{4}$$

$$H_{k+1} = H_k + \mathcal{P}_H(A^{-1} - H_k) + (A^{-1} - H_k)\mathcal{P}_H^\top - \mathcal{P}_H(A^{-1} - H_k)\mathcal{P}_H^\top, \tag{5}$$

where the weighted projectors $\mathcal{P}_B$ and $\mathcal{P}_H$ defined by Eq. (1) are

$$\mathcal{P}_B = P_{W^{-1},U_k} = W\,U_k(U_k^\top\, W\,U_k)^{-1}U_k^\top,$$
$$\mathcal{P}_H = P_{W^{-1},A\,U_k} = W\,A\,U_k(U_k^\top\, A\,W\,A\,U_k)^{-1}U_k^\top\, A.$$

Note, these are two different updates using the same one-sided sample $A\,U_k$ which are not simply connected by the Sherman-Morrison-Woodbury (SMW) formula. In Eq. (4), B_{k+1} is an improved approximation to A while in Eq. (5), H_{k+1} is an improved approximation to A^{-1}. Familiar algorithms are obtained by selecting different weights, W. Block DFP [15] is Eq. (4) with $W = A$

$$B_{k+1} = (I_n - \mathcal{P}_{\mathrm{DFP}})\; B_k(I_n - \mathcal{P}_{\mathrm{DFP}}^\top) + \mathcal{P}_{\mathrm{DFP}}\, A,$$

where $\mathcal{P}_{\mathrm{DFP}} = P_{A^{-1},U_k} = A\,U_k(U_k^\top\, A\,U_k)^{-1}U_k^\top$. Block BFGS [5,6] is the result of inverting Eq. (4) with $W = A^{-1}$ using the SMW formula

$$B_{k+1} = B_k - B_kU_k\left(U_k^\top B_kU_k\right)^{-1} U_k^\top B_k + A\,U_k\left(U_k^\top AU_k\right)^{-1} U_k^\top\, A.$$

QN algorithms are commonly initialized with multiples of the identity.

4 Randomized Two-Sided Quasi-Newton Algorithms

A general algorithm (defined by SPD weight matrices W_1 and W_2) to approximate non-square matrices and two distinct algorithms specialized to symmetric matrices are developed. As with one-sided sampled algorithms, different weights give different algorithms. Algorithms and theorems are developed for a generic initialization B_0.

4.1 General Two-Sided Sampled Update

Analogous to Eq. (2), our first algorithm is defined by the minimization

$$B_{k+1} = \arg\min_{B} \left\{ \frac{1}{2} \| B - B_k \|^2_{F(W_1^{-1}, W_2^{-1})} \mid U_k^\top B\, V_k = U_k^\top A\, V_k \right\}. \tag{6}$$

Solving the KKT equations for Eq. (6) gives the self-correcting update

$$B_{k+1} = B_k + P_{W_1^{-1}, U_k} (A - B_k) P^\top_{W_2^{-1}, V_k}. \tag{7}$$

Since this update explicitly corrects the projected residual sample $R_k = U_k^\top (A - B_k)\, V_k$, it decreases the weighted Frobenius norm $\|A - B_k\|^2_{F(W_1^{-1}, W_2^{-1})}$ unless the approximation is correct on the sampled spaces, i.e., $U_k^\top (A - B_k)\, V_k = 0$.

Given $A \in \mathbb{R}^{m \times n}$, initial approximation $B_0 \in \mathbb{R}^{m \times n}$, two-sided sample sizes $\{s_1, s_2\}$, and SPD weights $\{W_1, W_2\}$, Eq. (7) generates a sequence $\{B_k\}$ that converges monotonically to A in the appropriate weighted Frobenius norm. Pseudocode is provided in Algorithm 1: boxed values give the two-sided sample size per iteration; double boxed values the total for all iterations. For symmetric A, the independent left and right hand sampling fails to preserve symmetry.

```
Require: $B_0 \in \mathbb{R}^{m \times n}$, SPD $W_1 \in \mathbb{R}^{m \times m}$, $W_2 \in \mathbb{R}^{n \times n}$, $\{s_1, s_2\} \in \mathbb{N}$.
repeat $\{k = 0, 1, \ldots\}$
    Sample $U_k \sim N(0,1)^{m \times s_1}$ and $V_k \sim N(0,1)^{n \times s_2}$
    Compute $R_k = U_k^\top A\, V_k - U_k^\top\, B_k V_k \in \mathbb{R}^{s_1 \times s_2}$ ..................... $\boxed{s_1\, s_2}$
    Update $B_{k+1} = B_k + W_1 U_k (U_k^\top W_1 U_k)^{-1} R_k (V_k^\top W_2 V_k)^{-1} V_k^\top W_2$
until convergence
return $B_{k+1}$ ........................................... $\boxed{\boxed{(k+1)\,(s_1\, s_2)}}$
```

Algorithm 1: NS: Non-Symmetric Two-Sided Sampling

4.2 Symmetric Update

Unsurprisingly, the fully symmetrized general algorithm ($A = A^\top, B_0 = B_0^\top$, $V_k = U_k$ and $W = W_1 = W_2$) give symmetric approximations. Pseudocode is provided in Algorithm 2 with sample counts boxed as before.

```
Require: $B_0 \in \mathbb{R}^{n\times n}$ satisfying $B_0^\top = B_0$, SPD $W \in \mathbb{R}^{n\times n}$, $s \in \mathbb{N}$.
repeat $\{k = 0, 1, \ldots\}$
    Sample $U_k \sim \mathcal{N}(0,1)^{n\times s_1}$
    Compute $R_k = U_k^\top A\,U_k - U_k^\top B_k U_k \in \mathbb{R}^{s_1\times s_1}$ ........................ $\boxed{s_1^2}$
    Compute $\tilde{P}_k = W\,U_k(U_k^\top W\,U_k)^{-1}$
    Update $B_{k+1} = B_k + \tilde{P}_k\,R_k\tilde{P}_k^\top$
until convergence
return $B_{k+1}$ ............................................ $\boxed{\boxed{(k+1)\left(s^2\right)}}$
```

Algorithm 2: SS1: Symmetric Two-sided Sampling

Remark 1. Algorithm 2 can give non-SPD updates from SPD input *e.g.*

$$W = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \quad A = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \quad B = \begin{bmatrix}1 & 0\\ 0 & 9\end{bmatrix}, \quad \text{and} \quad U = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix}.$$

4.3 Multi-step Symmetric Updates

Enforcing symmetry for $A = A^\top$ and $B_0 = B_0^\top$ with an internal step

$$\begin{aligned} B_{k+1/2} &= B_k + P_{W^{-1},U_k}(A - B_k)P_{W^{-1},V_k}^\top \\ B_{k+1} &= \frac{1}{2}\left(B_{k+1/2} + B_{k+1/2}^\top\right) \end{aligned}$$

gives convergence comparable to Algorithm 2. However, the two-step algorithm

$$\begin{aligned} B_{k+1/3} &= B_k + P_{W_1^{-1},U_k}(A - B_k)P_{W_2^{-1},V_k}^\top \\ B_{k+2/3} &= B_{k+1/3} + P_{W_2^{-1},V_k}(A - B_{k+1/3}^\top)P_{W_1^{-1},U_k}^\top \\ B_{k+1} &= \frac{1}{2}\left(B_{k+2/3} + B_{k+2/3}^\top\right), \end{aligned} \tag{8}$$

has superior convergence properties and requires no additional data since

$$P_{W_2^{-1},V_k} A\,P_{W_1^{-1},U_k}^\top = \left(P_{W_1^{-1},U_k} A\,P_{W_2^{-1},V_k}^\top\right)^\top.$$

Pseudocode is provided in Algorithm 3 with sample counts boxed as before.

5 Convergence Analysis

Our convergence results rely on properties of randomly generated projectors. In our experiments, we orthogonalize square matrices with entries drawn from $N(0,1)$ to generate rotations from a rotationally invariant distribution [16]. Our algorithms use symmetric rank s projectors defined by an SPD weight W

$$\hat{z} = W^{1/2}U(U^\top WU)^{-1}U^\top W^{1/2},$$

```
Require: $B_0 \in \mathbb{R}^{n\times n}$ satisfying $B_0 = B_0^\top$, SPD $W \in \mathbb{R}^{m\times m}$, $\{s_1, s_2\} \in \mathbb{N}$.
repeat $\{k = 0, 1, \ldots\}$
    Sample $U_k \sim N(0,1)^{n\times s_1}$ and $V_k \sim N(0,1)^{n\times s_2}$
    Compute residual, $R_k = U_k^\top A\, V_k - U_k^\top B_k V_k \in \mathbb{R}^{s_1\times s_2}$ ............ $\boxed{s_1\, s_2}$
    Compute $B_{k+1/3} = B_k + W\, U_k (U_k^\top W\, U_k)^{-1} R_k (V_k^\top W\, V_k)^{-1} V_k^\top W$
    Compute $R_{k+1/3} = (U_k^\top A\, V_k)^\top - V_k^\top B_{k+1/3}\, U_k \in \mathbb{R}^{s_2\times s_1}$
    Compute $B_{k+2/3} = B_{k+1/3} + W\, V_k (V_k^\top W\, V_k)^{-1} R_{k+1/3} (U_k^\top W\, U_k)^{-1} U_k^\top W$
    Update $B_{k+1} = \frac{1}{2}(B_{k+2/3} + B_{k+2/3}^\top)$
until convergence
return $B_{k+1}$ ........................................ $\boxed{\boxed{(k+1)\,(s_1\, s_2)}}$
```

Algorithm 3: SS2: Two-Step Symmetric Two-Sided Sampling

where U is simply the first s columns of a random rotation. The smallest and largest eigenvalues λ_1 and λ_n of the expectation $\mathbf{E}[\hat{z}]$ of these random projections determines convergence of our algorithms with optimal rates when $\lambda_1 = \lambda_n$.

Definition 1. *A random matrix, $\hat{X} \in \mathbb{R}^{m\times n}$, is rotationally invariant if the distribution of $Q_m \hat{X} Q_n$ is the same for all rotations $Q_i \in \mathcal{O}(i)$.*

Proposition 1. *Let $\mathcal{Z}$ be any distribution of real, rank s projectors in $\mathbb{R}^n$. Then,*

$$0 \leq \lambda_{\min}(\boldsymbol{E}[\hat{z}]) \leq \frac{s}{n} \leq \lambda_{\max}(\boldsymbol{E}[\hat{z}]) \leq 1, \quad \hat{z} \in \mathcal{Z}.$$

Further, if $\hat{z}$ is rotationally invariant, then $\boldsymbol{E}[\hat{z}] = \frac{s}{n} I_n$.

Proposition 2. *For $R \in \mathbb{R}^{m\times n}$ and conforming symmetric projections $\hat{y}, \hat{z}$,*

$$\begin{aligned} \langle R\,\hat{z}, R\,\hat{z}\rangle_F &= \langle R, R\,\hat{z}\rangle_F \\ \langle \hat{y}\,R\,\hat{z}, \hat{y}\,R\,\hat{z}\rangle_F &= \langle \hat{y}\,R\,\hat{z}, R\,\hat{z}\rangle_F = \langle \hat{y}\,R\,\hat{z}, R\rangle_F \end{aligned} \tag{9}$$

Proposition 3. *For any $R \in \mathbb{R}^{m\times n}$ and conforming symmetric positive semi-definite matrices S_1, S_2, and (in the special case $m = n$) S we have the bounds:*

$$\begin{aligned} \lambda_{\min}(S_1)\langle R, R\rangle_F &\leq \langle S_1\, R, R\rangle_F \leq \lambda_{\max}(S_1)\langle R, R\rangle_F, \\ \lambda_{\min}(S_2)\langle R, R\rangle_F &\leq \langle R, R\, S_2\rangle_F \leq \lambda_{\max}(S_2)\langle R, R\rangle_F, \\ \lambda_{\min}(S)^2\langle R, R\rangle_F &\leq \langle S\, R, R\, S\rangle_F \leq \lambda_{\max}(S)^2\langle R, R\rangle_F. \end{aligned}$$

Remark 2. Convergence results for Algorithms 1 to 3. are for $\mathbf{E}[\|B - A\|_F^2]$. Such results dominate similar results for $\|\mathbf{E}[B - A]\|_F^2$ since

$$\|\mathbf{E}\,[B - A]\|_F^2 = \mathbf{E}\left[\|B - A\|_F^2\right] - \mathbf{E}\left[\|B - \mathbf{E}\,[B]\|_F^2\right].$$

Theorem 1 (Convergence of Algorithm 1 - NS). *For $A \in \mathbb{R}^{m\times n}$ and $B_0 \in \mathbb{R}^{m\times n}$ with $W_1 \in \mathbb{R}^{m\times m}$ and $W_2 \in \mathbb{R}^{n\times n}$ fixed SPD weights. If $U_k \in \mathbb{R}^{m\times s_1}$*

and $V_k \in \mathbb{R}^{n\times s_2}$ are random, independently selected orthogonal matrices with full column rank (with probability one), then B_k from Algorithm 1 satisfies

$$\boldsymbol{E}\left[\|B_{k+1}-A\|^2_{F(W_1^{-1},W_2^{-1})}\right] \le (\rho_{NS})^k \boldsymbol{E}\left[\|B_0-A\|^2_{F(W_1^{-1},W_2^{-1})}\right],$$

where $\rho_{NS} = 1-\lambda_{\min}(\boldsymbol{E}[\hat{y}])\,\lambda_{\min}(\boldsymbol{E}[\hat{z}])$, with

$$\hat{y}_k = W_1^{1/2}U_k(U_k^\top W_1U_k)^{-1}U_k^\top W_1^{1/2}, \quad \hat{z}_k = W_2^{1/2}V_k(V_k^\top W_2V_k)^{-1}V_k^\top W_2^{1/2}.$$

Proof. Define the kth residual as $R_k := W_1^{-1/2}(B_k-A)W_2^{-1/2}$. With some algebraic manipulation, Eq. (7) can be re-written as $R_{k+1} = R_k - \hat{y}_kR_k\hat{z}_k$. Computing the squared Frobenius norm of both sides,

$$\begin{aligned}\langle R_{k+1},R_{k+1}\rangle_F &= \langle R_k-\hat{y}_kR_k\hat{z}_k, R_k-\hat{y}_kR_k\hat{z}_k\rangle_F\\ &= \langle R_k,R_k\rangle_F - \langle R_k,\hat{y}_kR_k\hat{z}_k\rangle_F - \langle \hat{y}_kR_k\hat{z}_k,R_k\rangle_F + \langle \hat{y}_kR_k\hat{z}_k,\hat{y}_kR_k\hat{z}_k\rangle_F\\ &= \langle R_k,R_k\rangle_F - \langle \hat{y}_kR_k\hat{z}_k,R_k\hat{z}_k\rangle_F,\end{aligned}$$

where we have made use of Proposition 2. Taking the expected value with respect to independent samples U_k (leaving V_k and R_k fixed) gives

$$\begin{aligned}&\mathbf{E}\left[\|R_{k+1}\|_F^2 \mid V_k,R_k\right] = \langle R_k,R_k\rangle_F - \langle \mathbf{E}[\hat{y}_k]R_k\hat{z}_k,R_k\hat{z}_k\rangle_F\\ &\le \langle R_k,R_k\rangle_F - \lambda_{\min}(\mathbf{E}[\hat{y}_k])\langle R_k\hat{z}_k,R_k\hat{z}_k\rangle_F \le \langle R_k,R_k\rangle_F - \lambda_{\min}(\mathbf{E}[\hat{y}_k])\langle R_k,R_k\hat{z}_k\rangle_F,\end{aligned}$$

where we applied Proposition 3 to the symmetric positive semi-definite matrix $\mathbf{E}[\hat{y}_k]$, and used Eq. (9). Taking the expected value with respect to independent samples V_k and leaving R_k fixed gives

$$\begin{aligned}\mathbf{E}[\|R_{k+1}\|_F^2 \mid R_k] &\le \langle R_k,R_k\rangle_F - \lambda_{\min}(\mathbf{E}[\hat{y}_k])\langle R_k,R_k\mathbf{E}[\hat{z}_k]\rangle_F\\ &\le \langle R_k,R_k\rangle_F - \lambda_{\min}(\mathbf{E}[\hat{y}_k])\lambda_{\min}(\mathbf{E}[\hat{z}_k])\langle R_k,R_k\rangle_F.\end{aligned}$$

Taking the full expectation and noting $\mathbf{E}[\|R_{k+1}\|_F^2] = \mathbf{E}[\|B_k-A\|^2_{F(W_1^{-1},W_2^{-1})}$

$$\begin{aligned}\mathbf{E}[\|R_{k+1}\|_F^2] &\le \mathbf{E}\left[\langle R_k,R_k\rangle_F\right] - \lambda_{\min}(\mathbf{E}[\hat{y}_k])\lambda_{\min}(\mathbf{E}[\hat{z}_k])\mathbf{E}\left[\langle R_k,R_k\rangle_F\right]\\ &= (1-\lambda_{\min}(\mathbf{E}[\hat{y}_k])\lambda_{\min}(\mathbf{E}[\hat{z}_k]))\mathbf{E}[\langle R_k,R_k\rangle_F]\end{aligned}$$

gives the result by unrolling the recurrence. Note, independence of U_k and V_k justifies $\mathbf{E}[\langle \hat{y}_kR_k\hat{z}_k,R_k\hat{z}_k\rangle_F] = \langle \mathbf{E}[\hat{y}_k]R_k\hat{z}_k,R_k\hat{z}_k\rangle_F$.

Theorem 2 (Convergence of Algorithm 2 - SS1). *Let $A, W \in \mathbb{R}^{n\times n}$ be fixed SPD matrices and $U_k \in \mathbb{R}^{n\times s}$ be a randomly selected matrix having full column rank with probability 1. If $B_0 \in \mathbb{R}^{n\times n}$ is an initial guess for A with $B_0 = B_0^\top$, then B_k from Algorithm 2 satisfies*

$$\boldsymbol{E}[\|B_{k+1}-A\|^2_{F(W^{-1})}] \le (\rho_{SS1})^k\boldsymbol{E}[\|B_0-A\|^2_{F(W^{-1})}],$$

where $\rho_{SS1} = 1-\lambda_{\min}(\boldsymbol{E}[\hat{z}])^2$ and $\hat{z}_k = W^{1/2}U_k(U_k^\top W\,U_k)^{-1}U_k^\top W^{1/2}$.

Proof. Following similar steps outlined in the proof in Theorem 1, we arrive at

$$\langle R_{k+1}, R_{k+1}\rangle_F = \langle R_k, R_k\rangle_F - \langle R_k, \hat{z}_k R_k \hat{z}_k\rangle_F.$$

Taking the expected value with respect to U_k leaving R_k fixed we have

$$\begin{aligned}
&\mathbf{E}\left[\|R_{k+1}\|_F^2 \mid R_k\right] = \langle R_k, R_k\rangle_F - \mathbf{E}\left[\langle R_k, \hat{z}_k R_k \hat{z}_k\rangle_F\right]\\
&= \langle R_k, R_k\rangle_F - \mathbf{E}\left[\mathrm{Tr}[\mathrm{R_k^\top \hat{z}_k R_k \hat{z}_k}]\right]\\
&= \langle R_k, R_k\rangle_F - \mathrm{Tr}\left[\mathbf{E}\left[\mathrm{R_k \hat{z}_k R_k \hat{z}_k}\right]\right] \leq \langle \mathrm{R_k}, \mathrm{R_k}\rangle_{\mathrm{F}} - \mathrm{Tr}\left[\mathbf{E}\left[\mathrm{R_k \hat{z}_k}\right]^2\right],
\end{aligned}$$

where the inequality arises from application of Jensen's Inequality. Simplifying and applying Proposition 3,

$$\begin{aligned}
&\mathbf{E}[\|R_{k+1}\|_{F(W^{-1})}^2 \mid R_k] \leq \langle R_k, R_k\rangle_F - \mathrm{Tr}\left[\mathbf{E}\left[\mathrm{R_k \hat{z}_k}\right]^2\right]\\
&= \langle R_k, R_k\rangle_F - \mathrm{Tr}\left[\mathrm{R_k}\mathbf{E}\left[\mathrm{\hat{z}_k}\right]\mathrm{R_k}\mathbf{E}\left[\mathrm{\hat{z}_k}\right]\right]\\
&= \langle R_k, R_k\rangle_F - \langle \mathbf{E}[\hat{z}_k]R_k, R_k\mathbf{E}[\hat{z}_k]\rangle_F \leq \langle R_k, R_k\rangle_F - \lambda_{\min}(\mathbf{E}[\hat{z}_k])^2\langle R_k, R_k\rangle_F.
\end{aligned}$$

Taking the full expectation and un-rolling the recurrence yields the desired result.

Theorem 3 (Convergence of Algorithm 3 - SS2). *Let A, U_k, V_k and B_0 be defined as in Theorem 1, and let W be a fixed SPD matrix then B_k from Algorithm 3 satisfies*

$$\boldsymbol{E}\left[\|B_k - A\|_{F(W^{-1})}^2\right] \leq (\rho_{SS2})^k \boldsymbol{E}\left[\|B_0 - A\|_{F(W^{-1})}^2\right],$$

where

$$\rho_{SS2} = 1 - 2\lambda_{\min}(\boldsymbol{E}[\hat{y}])\lambda_{\min}(\boldsymbol{E}[\hat{z}]) + \lambda_{\min}(\boldsymbol{E}[\hat{y}])^2\lambda_{\min}(\boldsymbol{E}[\hat{z}])^2.$$

Proof. Let R_k be the kth residual R_k, and $\hat{y}_k, \hat{z}_k$ be projectors as in Theorem 1 with $W = W_1 = W_2$. Eq. (8) can be re-written in terms of R_k as follows.

$$R_{k+1/3} = R_k - \hat{y}_k R_k \hat{z}_k, \quad R_{k+2/3}^\top = R_{k+1/3}^\top - \hat{z}_k R_{k+1/3}^\top \hat{y}_k,$$
$$R_{k+1} = \frac{1}{2}\left(R_{k+2/3} + R_{k+2/3}^\top\right).$$

Theorem 1 gives

$$\mathbf{E}\left[\left\|R_{k+1/3}\right\|_F^2\right] \leq (\rho_{\mathrm{NS}})\mathbf{E}\left[\|R_k\|_F^2\right],$$

and a repeated application of Theorem 1 gives

$$\mathbf{E}\left[\left\|R_{k+2/3}\right\|_F^2\right] \leq (\rho_{\mathrm{NS}})\mathbf{E}\left[\|R_{k+1/3}\|_F^2\right] \leq (\rho_{\mathrm{NS}})^2\mathbf{E}\left[\|R_k\|_F^2\right].$$

Lastly, we observe via the triangle inequality that

$$\begin{aligned}
&\mathbf{E}\left[\|R_{k+1}\|_F^2\right] = \mathbf{E}\left[\left\|\frac{1}{2}\left(R_{k+2/3} + R_{k+2/3}^\top\right)\right\|_F^2\right]\\
&\leq \frac{1}{2}\mathbf{E}\left[\left\|R_{k+2/3}\right\|_F^2\right] + \frac{1}{2}\mathbf{E}\left[\left\|R_{k+2/3}^\top\right\|_F^2\right] = (\rho_{\mathrm{NS}})^2\mathbf{E}\left[\|R_k\|_F^2\right],
\end{aligned}$$

Un-rolling the loop for k iterations gives the desired result.

With the relevant rate ρ below error bounds for Algorithms 1 to 3 are

$$\|R_{k+1}\|^2_{F(W_1^{-1},W_2^{-1})} \leq \rho\|R_k\|^2_{F(W_1^{-1},W_2^{-1})} \tag{10}$$

where $y_1 = \lambda_{\min}(\mathbf{E}[\hat{y}])$, $z_1 = \lambda_{\min}(\mathbf{E}[\hat{z}])$, and

$$\rho_{\mathrm{NS}}(y_1, z_1) = 1 - y_1 z_1, \quad \rho_{\mathrm{SS1}}(z_1) = 1 - z_1^2, \quad \rho_{\mathrm{SS2}}(y_1, z_1) = (1 - y_1 z_1)^2. \tag{11}$$

Since any symmetric rank s random projection $\hat{z}$ on $\mathbb{R}^n$ satisfies $0 \leq z_1 \leq \frac{s}{n} \leq z_n \leq 1$ and rotationally invariant distributions, e.g. UU^+ with $U \sim N(0,1)^{n\times s}$, further satisfy $\mathbf{E}[\hat{z}] = \frac{s}{n}$, minimizing the various convergence rates ρ over the appropriate domains gives the following optimal rates.

Corollary 1. *The optimal convergence rates for Algorithms 1 to 3 are obtained attained for U_k and V_k sampled from rotationally invariant distributions,*

$$\rho^{opt}_{NS} = 1 - \frac{s_1}{m}\frac{s_2}{n}, \quad \rho^{opt}_{SS1} = 1 - \left(\frac{s_2}{n}\right)^2, \quad \rho^{opt}_{SS2} = \left(1 - \frac{s_1}{m}\frac{s_2}{n}\right)^2. \tag{12}$$

Remark 3. Theorems 1 to 3 all assume the weight matrix W and distributions are fixed. All our non-accelerated numerical experiments use fixed weights and sample from fixed rotationally invariant distributions.

Remark 4. Corollary 1 is an extremely strong result. Consider for simplicity $s_1 = s_2 = s$. Although the convergence rates are $\sim 1-\left(\frac{s}{n}\right)^2$, only $s\times s$ aggregated pieces of information are used each iteration. If a one-sided sampled algorithm uses $s \times n$ pieces of information, e.g. [6], our algorithm can take $\frac{n}{s}$ iterations with the *same amount of information*. Consequently the error decrease after $\frac{n}{s}$ iterations, is comparable to convergence rates of one-sided sampled QN methods.

$$\left(1 - \frac{s^2}{n^2}\right)^{n/s} \approx 1 - \frac{n}{s}\cdot\frac{s^2}{n^2},$$

Lower bounds on the convergence rates (analogous to the upper bounds in Theorems 1 to 3 but using the upper bounds in Proposition 3) are easily derived. For example, the two-sided error bound for Algorithm 1 is

$$\rho_{\mathrm{NS}}(y_m, z_n)\mathbf{E}[\|R_k\|^2_F] \leq \mathbf{E}[\|R_{k+1}\|^2_F] \leq \rho_{\mathrm{NS}}(y_1, z_1)\mathbf{E}[\|R_k\|^2_F],$$

where as before $y_1 \leq y_2 \leq \cdots \leq y_m$ is the spectrum of $\mathbf{E}[\hat{y}]$, $z_1 \leq z_2 \leq \cdots \leq z_n$ is the spectrum of $\mathbf{E}[\hat{z}]$ and the explicit form for ρ_{NS} is in Eq. (11). We collect the similar results for Algorithms 1 to 3 in Corollary 2.

Corollary 2 (Two-Sided Convergence Rates). *Given the assumptions of Theorems 1 to 3 the explicit formulas Eq.* (11) *for ρ give two-sided bounds,*

$$\rho_{NS}(y_m, z_n)^k \le \frac{\boldsymbol{E}\left[\|B_{k+1} - A\|^2_{F(W_1^{-1}, W_2^{-1})}\right]}{\|B_0 - A\|^2_{F(W_1^{-1}, W_2^{-1})}} \le \rho_{NS}(y_1, z_1)^k$$

$$\rho_{SS1}(z_n)^k \le \frac{\boldsymbol{E}\left[\|B_{k+1} - A\|^2_{F(W^{-1})}\right]}{\|B_0 - A\|^2_{F(W^{-1})}} \le \rho_{SS1}(z_1)^k$$

$$\rho_{SS2}(y_n, z_n)^k \le \frac{\boldsymbol{E}\left[\|B_{k+1} - A\|^2_{F(W^{-1})}\right]}{\|B_0 - A\|^2_{F(W^{-1})}} \le \rho_{SS2}(y_1, z_1)^k$$

where y_1, y_m, z_1, z_n are the extreme eigenvalues of $\boldsymbol{E}[\hat{y}]$ and $\boldsymbol{E}[\hat{z}]$.

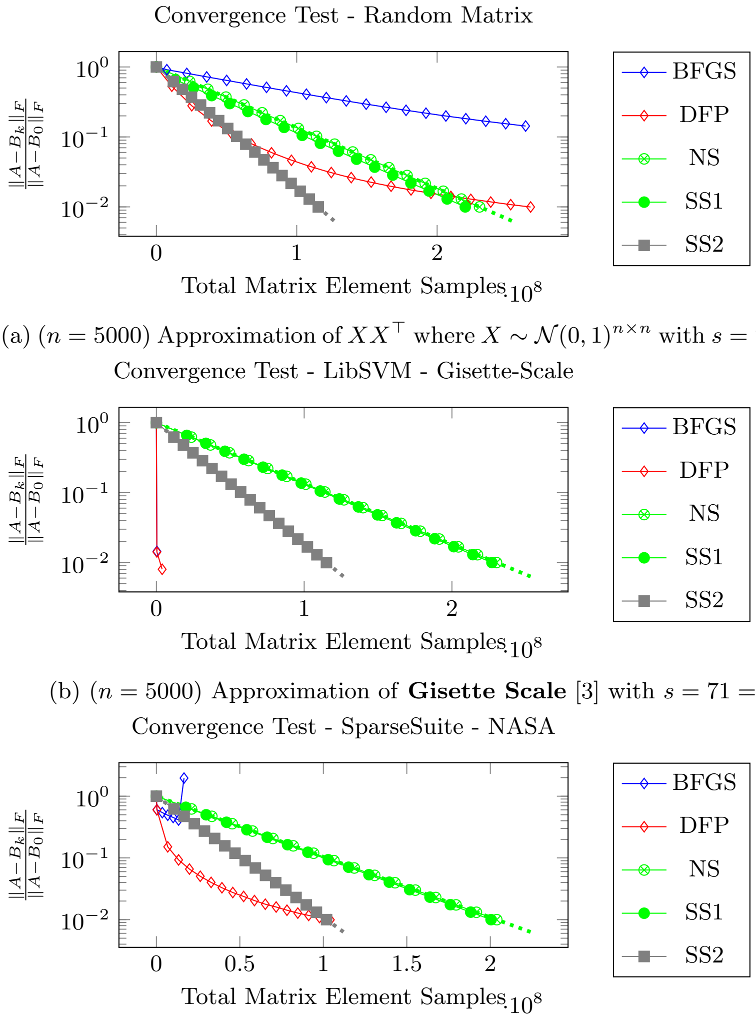
Remark 5. If $\hat{y}$ and $\hat{z}$ are rotationally invariant, the upper and lower probabilistic bounds in Corollary 2 coincide since $z_1 = z_n = \frac{s_1}{n}$ and $y_1 = y_m = \frac{s_2}{m}$. Algorithms 1 to 3 all use rotationally invariant distributions and converge predictably at the expected rate. The algorithms still converge with other distributions provided the smallest eigenvalue of the expectation is positive.

6 Numerical Results

Algorithm 1 to 3 were implemented in the MATLAB framework from [6] and tested on representative SPD matrices from the same article: $A = XX^\top$ with $X \sim \mathcal{N}(0,1)^{n \times n}$; the Gisette-Scale ridge regression matrix from [3]; and the NASA matrix from [4]. The author's website [13] contains MATLAB scripts and similar results for all matrices from [6]. Computations were performed on Superior, the HPC facility at Michigan Technological University.

Many metrics can be used to objectively compare algorithmic costs. Common metrics include number of FLOPS, total memory used, communication overhead, and for matrix-free black-box procedures the number of individual matrix-vector products $A\,v$. As noted by [9] the analogous metric for a black box procedure to compute the matrix product required for our $s_1 \times s_2$ two-sided sample $U^\top A\,V$ is the number of sampled entries, $s_1\,s_2$. As an explicit example, for $f : \mathbb{R}^m \to \mathbb{R}$, Forward-Forward mode [10] Algorithmic Differentiation (AD) simultaneously computes $f(x)$ and a directional 2nd derivative $u^\top \nabla^2 f(x)\,v$. With sufficient shared-memory processors, AD can efficiently compute $f(x)$ and the two-sided sample $U^\top \nabla^2 f(x)\,V$ of the Hessian with cost $s_1\,s_2$.

Algorithms in [6] are for symmetric matrices. We compare the convergence of (unweighted i.e. $W = I$) Algorithms 1 to 3 (sample size $s = \lceil\sqrt{n}\rceil$ matching [6]) to the one sided algorithms in [6]: Fig. 1a for $A = X\,X^\top$ with $X \sim \mathcal{N}(0,1)^{5000 \times 5000}$; Fig. 1b for Gisette-Scale; and Fig. 1c) for NASA. Our algorithms achieve the theoretical convergence rates from Eq. (12) (dotted lines). Weighted algorithms DFP and BFGS use $B_0 = I$, un-weighted Algorithms 1 to 3 use $B_0 = 0$. Runs were terminated after $5n^2$ iterations or when the relative



(a) ($n = 5000$) Approximation of $XX^\top$ where $X \sim \mathcal{N}(0,1)^{n \times n}$ with $s = 71 = \lceil\sqrt{5000}\rceil$.

(b) ($n = 5000$) Approximation of **Gisette Scale** [3] with $s = 71 = \lceil\sqrt{5000}\rceil$.

(c) ($n = 4700$) Approximation of **NASA4704** [4]. $s = 69 = \lceil\sqrt{4704}\rceil$.

Fig. 1. Two-sided sampled algorithm performance with theoretical rate as dots. BFGS and DFP can be: a) comparable; b) superior; or c) stall/diverge.

residual norm fell below 0.01. The one-sided algorithms DFP and BFGS have target dependent weight matrices: DFP is Eq. (4) with weight $W = A$ while BFGS is the Sherman-Morrison-Woodbury inversion of Eq. (5) with $W = A^{-1}$. Figure 1a shows our algorithms outperforming both BFGS and DFP for small tolerances. Figure 1b shows enhanced initial convergence for DFP and BFGS, but Fig. 1c demonstrates that BFGS may not converge. In contrast, our two-sided algorithms converge consistently, achieving the theoretical convergence rates.

7 Heuristic Accelerated Schemes

Algorithm 2 corrects the symmetric projected residual $U_k^\top(A - B_k)U_k$ at each stage; significant corrections occur if U_k aligns with large eigenvalues of R_k. Block power iteration is a standard heuristic [7] to enhance alignment.

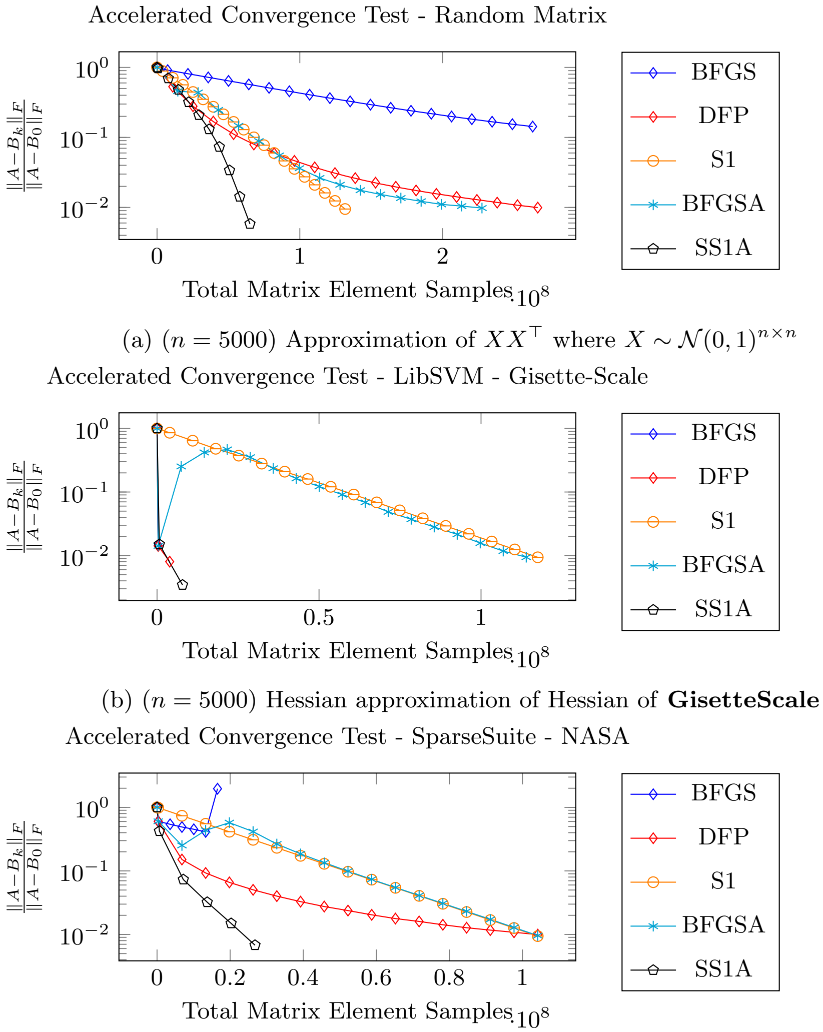
Incorporating p steps of a block power iteration to enrich U_k produces the hybrid algorithm in Algorithm 2: the loop from line 4 to line 9 enriches a random U by multiplying by the residual and re-orthogonalizing p times. As before, work estimates are boxed on the right (p block power iterations require $p\,n\,s$ and the square symmetric sample requires s^2) with the total double boxed. Although the inner iteration requires significantly more matrix samples per iteration, conventional wisdom [7] suggests one or two inner iterations are likely to be beneficial. Our experiments show Algorithm 2 is competitive for $p = 2$.

```
**Require:** $B_0 \in \mathbb{R}^{n\times n}$ satisfying $B_0^\top = B_0$, SPD $W \in \mathbb{R}^{n\times n}$, $s \in \mathbb{N}$.
 1: **repeat** $\{k = 0, 1, \ldots\}$
 2:     Sample $U_{0,k} \sim \mathcal{N}(0,1)^{n\times s}$
 3:     $B_{0,k} = B_k$
 4:     **loop** $\{i = 1, 2, \ldots, p\}$
 5:         $\Lambda = AU_{i-1,k} - B_{i-1,k}U_{i-1,k}$
 6:         $\Sigma = \Lambda(U_{i-1,k}^\top W\,U_{i-1,k})^{-1}U_{i-1,k}^\top W$
 7:         $B_{i,k} = B_{i-1,k} + \Sigma + \Sigma^\top - W\,U_{i-1,k}(U_{i-1,k}^\top W\,U_{i-1,k})^{-1}U_{i-1,k}^\top \Sigma$
 8:         $U_{i,k} = \Lambda$
 9:     **end loop** ........................................ $\boxed{p\,n\,s}$
10:     Compute $R_k = U_{p,k}^\top A\,U_{p,k} - U_{p,k}^\top B_{p,k}U_{m,k} \in \mathbb{R}^{s\times s}$ .......... $\boxed{s^2}$
11:     Compute $\tilde{P}_k = W\,U_{p,k}(U_{p,k}^\top W\,U_{p,k})^{-1}$
12:     Update $B_{k+1} = B_k + \tilde{P}_k\,R_k\tilde{P}_k^\top$
13: **until** convergence
14: **return** $B_{k+1}$ .............................. $\boxed{\boxed{(k+1)(p\,n\,s + s^2)}}$
```

Algorithm 4: SS1A: Accelerated Symmetric Approximation

Acceleration Convergence Results

Algorithm 4 (rotationally invariant samples with $p = 2$) is compared to BFGS-A (the result of applying the SMW formula to the accelerated method AdaRBFGS from [6]) and the three one-sided, non-accelerated algorithms: S1 and DFP defined by Eq. (4) with weights $W = I$ and $W = A$ (respectively), and BFGS defined by applying the SMW formula to Eq. (5) with weight $W = A^{-1}$. We use the same test matrices, initialization, and termination conditions described in Sect. 6: Fig. 2a shows results for $A = XX^\top$; Fig. 2b shows results for the Hessian matrix Gisette Scale [3]; and Fig. 2c shows results for NASA4704 [4]. SS1A matches or outperforms all other algorithms for the three matrices. As before [13] contains MATLAB scripts and results for all matrices from [6]. Both



(a) ($n = 5000$) Approximation of $XX^\top$ where $X \sim \mathcal{N}(0,1)^{n\times n}$

(b) ($n = 5000$) Hessian approximation of Hessian of **GisetteScale**

(c) ($n = 4700$) Approximation of **NASA4704**

Fig. 2. Accelerated SS1A algorithm outperforms accelerated BFGS algorithms.

BFGS-A and SS1A adaptively sample their update spaces. BFGS-A samples from the columns of the Cholesky decomposition of B_k while SS1-A effectively samples from a small power of the residual $A - B_k$. Comparing BFGS to BFGS-A and S1 to SS1-A shows the benefits of adaptivity. The hybrid SS1A performs consistently well.

8 Conclusions and Future Work

Algorithms 1 to 3 iteratively generate matrix approximations to a fixed target from two-sided samples. Rotationally invariant sampling gives optimal theoretical convergence in general and predicted convergence rates are experimentally verified for several real world matrices, with comparable performance to existing one-sided algorithms. A hybrid method combining simultaneous iteration (to enrich a subspace) with the two-sided sampled update is developed and shown to be competitive with existing one-sided accelerated schemes.

The algorithms systematically make minimal changes and drive weighted residual norms for a fixed A monotonically to zero. Such self-correcting algorithms can potentially approximate slowly changing matrices, $A(x)$. For example, QN optimization algorithms have a slowly changing Hessian target $\nabla_x^2 f(x_k)$ while solvers for stiff ODEs $y'(x) = f(y(x))$ have a slowly changing Jacobian target $\nabla_y f(y(x_k))$. The two-sided sampled matrix approximation algorithms and theory presented in the article provide a general foundation for these and other applications. Efficient factorized updates, compact low rank approximations, inverse approximation, and sparse matrix sampling are all planned.

References

1. Andoni, A., Nguyen, H.L.: Eigenvalues of a matrix in the streaming model, pp. 1729–1737. SIAM (2013). https://doi.org/10.1137/1.9781611973105.124
2. Avron, H., Clarkson, K.L., Woodruff, D.P.: Sharper bounds for regularized data fitting. In: Jansen, K., Rolim, J.D.P., Williamson, D., Vempala, S.S. (eds.) Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Leibniz International Proceedings in Informatics (LIPIcs), vol. 81, pp. 27:1–27:22. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, Dagstuhl (2017). https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2017.27
3. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 27:1–27:27 (2011)
4. Davis, T.A., Hu, Y.: The university of florida sparse matrix collection. ACM Trans. Math. Softw. **38**(1), 1:1–1:25 (2011). https://doi.org/10.1145/2049662.2049663
5. Gao, W., Goldfarb, D.: Block BFGS methods. SIAM J. Optim. **28**(2), 1205–1231 (2018). https://doi.org/10.1137/16M1092106
6. Gower, R.M., Richtárik, P.: Randomized Quasi-Newton updates are linearly convergent matrix inversion algorithms. SIAM J. Matrix Anal. Appl. **38**(4), 1380–1409 (2017). https://doi.org/10.1137/16M1062053
7. Halko, N., Martinsson, P., Tropp, J.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. **53**(2), 217–288 (2011). https://doi.org/10.1137/090771806
8. Karush, W.: Minima of Functions of Several Variables with Inequalities as Side Conditions. ProQuest LLC, Ann Arbor, MI (1939). Thesis (SM)-The University of Chicago
9. Li, Y., Nguyen, H.L., Woodruff, D.P.: On Sketching Matrix Norms and the Top Singular Vector, pp. 1562–1581. SIAM (2014). https://doi.org/10.1137/1.9781611973402.114

10. Naumann, U.: The Art of Differentiating Computer Programs. Society for Industrial and Applied Mathematics (2011). https://doi.org/10.1137/1.9781611972078
11. Nocedal, J.: Updating Quasi-Newton matrices with limited storage. Math. Comput. **35**(151), 773–782 (1980). https://doi.org/10.2307/2006193
12. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering, 2nd edn. Springer, New York (2006). https://doi.org/10.1007/978-0-387-40065-5
13. Ong, B., Azzam, J., Struthers, A.: Randomized iterative methods for matrix approximation - supplementary material and software repository (2021). https://www.mathgeek.us/publications.html
14. Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. **22**(3), 400–407 (1951). https://doi.org/10.1214/aoms/1177729586
15. Schnabel, R.: Quasi-newton methods using multiple secant equations. Computer Science Technical Reports 244, 41 (1983). https://scholar.colorado.edu/csci_techreports/244/
16. Stewart, G.: The efficient generation of random orthogonal matrices with an application to condition estimators. SIAM J. Numer. Anal. **17**(3), 403–409 (1980). https://doi.org/10.1137/0717034