Randomized Singular Value Projection

Stephen Becker
UPMC Paris 6
Email: stephen.becker@upmc.fr

Volkan Cevher
Ecole Polytechnique Fédérale de Lausanne
Email: volkan.cevher@epfl.ch

Anastasios Kyrillidis
Ecole Polytechnique Fédérale de Lausanne
Email: anastasios.kyrillidis@epfl.ch

***Abstract*—Affine rank minimization algorithms typically rely on calculating the gradient of a data error followed by a singular value decomposition at every iteration. Because these two steps are expensive, heuristic approximations are often used to reduce computational burden. To this end, we propose a recovery scheme that merges the two steps with randomized approximations, and as a result, operates on space proportional to the degrees of freedom in the problem. We theoretically establish the estimation guarantees of the algorithm as a function of approximation tolerance. While the theoretical approximation requirements are overly pessimistic, we demonstrate that in practice the algorithm performs well on the quantum tomography recovery problem.**

I. Introduction

In many signal processing and machine learning applications, we are given a set of observations $\mathbf{y} \in \mathbb{R}^p$ of a rank-r matrix $\mathbf{X}^\star \in \mathbb{R}^{m\times n}$ as $\mathbf{y} = \mathcal{A}\mathbf{X}^\star + \boldsymbol{\varepsilon}$ via the linear operator $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^p$, where $r \ll \min\{m,n\}$ and $\boldsymbol{\varepsilon} \in \mathbb{R}^p$ is additive noise. As a result, we are interested in the solution of

$$\begin{aligned} \underset{\mathbf{X}\in\mathbb{R}^{m\times n}}{\text{minimize}} \quad & f(\mathbf{X}) \\ \text{subject to} \quad & \operatorname{rank}(\mathbf{X}) \le r, \end{aligned} \tag{1}$$

where $f(\mathbf{X}) := \|\mathbf{y} - \mathcal{A}\mathbf{X}\|_2^2$ is the data error. While the optimization problem in (1) is non-convex, it is possible to obtain robust recovery with provable guarantees via iterative greedy algorithms (SVP) [1], [2] or convex relaxations [3], [4] from measurements as few as $p = \mathcal{O}(r(m+n-r))$.

Currently, there is a great interest in designing algorithms to handle large scale versions of (1) and its variants. As a concrete example, consider quantum tomography (QT), where we need to recover low-rank density matrices from dimensionality reducing Pauli measurements [5]. In this problem, the size of these density matrices grows exponentially with the number of quantum bits. Other collaborative filtering problems, such as the Netflix challenge, also require huge dimensional optimization. Without careful implementations or non-conventional algorithmic designs, existing algorithms quickly run into time and memory bottlenecks.

These computational difficulties typically revolve around two critical issues. First, virtually all recovery algorithms require calculating the gradient $\nabla f(\mathbf{X}) = 2\mathcal{A}^*(\mathcal{A}(\mathbf{X}) - \mathbf{y})$ at an intermediate iterate $\mathbf{X}$, where $\mathcal{A}^*$ is the adjoint of $\mathcal{A}$. When the range of $\mathcal{A}^*$ contains dense matrices, this forces algorithms to use memory proportional to $\mathcal{O}(mn)$. Second, after the iterate is updated with the gradient, projecting onto the low-rank space requires a partial singular value decomposition (SVD). This is usually problematic for the initial iterations of convex algorithms, where they may have to perform full SVD's. In contrast, greedy algorithms [2] fend off the complexity of full SVD's, since they need fixed rank projections, which can be approximated via Lanczos or randomized SVD's [6].

Algorithms that avoid these two issues do exist, such as [7]–[10], and are typically based on the Burer-Monteiro splitting [11]. The main idea in Burer-Monteiro splitting is to remove the non-convex rank constraint by directly embedding it into the objective: as opposed to optimizing $\mathbf{X}$, splitting algorithms directly work with its fixed factors $\mathbf{U}\mathbf{V}^T = \mathbf{X}$ in an alternating fashion, where $\mathbf{U} \in \mathbb{R}^{m\times\hat{r}}$ and $\mathbf{V} \in \mathbb{R}^{n\times\hat{r}}$ for some $\hat{r} \ge r$. Unfortunately, rigorous guarantees are difficult.[1] The work [12] has shown approximation guarantees if $\mathcal{A}$ satisfies a restricted isometry property with constant $\delta_{2r} \le \kappa^2/(100r)$ (noiseless), where $\kappa = \sigma_1(\mathbf{X}^\star)/\sigma_r(\mathbf{X}^\star)$, or $\delta_{2r} \le 1/(3200r^2)$ for a bound independent of κ. The authors suggest that these bounds may be tightened based on the good empirical performance of the algorithm.

In this paper, we merge the gradient calculation and the singular value projection steps into one and show that this not only removes a huge computational burden, but suffers only a minor convergence speed drawback in practice. Our contribution is a natural but non-trivial fusion of the Singular Value Projection (SVP) algorithm in [1] and the approximate projection ideas in [2]. The SVP algorithm is a hard-thresholding algorithm that has been considered in [1], [13]. Inexact steps in SVP have been considered as a heuristic [13] but have not been incorporated into an overall convergence result. A non-convex framework for affine rank minimization (including variants of the SVP algorithm) that utilizes inexact projection operations with provable signal approximation and convergence guarantees is proposed in [2]. Both [1], [2] do not consider splitting techniques in the proposed schemes.

In this work, departing from [1], [2], we engineer the SVP algorithm to operate like splitting algorithms that *directly work with the factors*; this added twist decreases the per iteration requirements in terms of storage and computational complexity. Using this new formulation, each iteration is nearly as fast as in the splitting method, hence removing a

[1] If $\hat{r} \gtrsim \sqrt{p}$, then [11] shows their method obtains a global solution, but this is impractical for large p. Moreover, it is shown that the explicit rank $\hat{r}$ splitting method solves a non-convex problem that has the same local minima as (1) (if $\hat{r} = r$). However, the non-convex problems are not *equivalent* (e.g. $\mathbf{U} = \mathbf{0}$, $\mathbf{V} = \mathbf{0}$ is a stationary point for the splitting problem whereas $\mathbf{X} = \mathbf{0}$ is generally not a stationary point for (1)).

drawback to SVP in relation to splitting methods. Furthermore, we prove that, under some conditions, it is still possible to obtain perfect recovery even if the projections are inexact. Such characterizations have been used for convex [3] and non-convex [1], [2] algorithms to obtain approximation guarantees.

II. Preliminary material

Notation: $\mathcal{P}_\Omega$ is an orthogonal projection onto the closed set Ω when it exists, and $\mathcal{P}_r$ stands for $\mathcal{P}_{\{\mathbf{X}:\mathrm{rank}(\mathbf{X})\leq r\}}$ (which does exist by the Eckart-Young theorem). Computer routine names are typeset with a `typewriter font`.

R-RIP: The Rank Restricted Isometry Property (R-RIP) is a common tool used in matrix recovery [1]–[3]:

Definition 1 (R-RIP for linear operators on matrices [3])**.** *A linear operator $\boldsymbol{\mathcal{A}} : \mathbb{R}^{m\times n} \to \mathbb{R}^p$ satisfies the R-RIP with constant $\delta_r(\boldsymbol{\mathcal{A}}) \in (0,1)$ if, $\forall \mathbf{X} \in \mathbb{R}^{m\times n}$ with $\mathrm{rank}(\mathbf{X}) \leq r$,*

$$(1-\delta_r(\boldsymbol{\mathcal{A}}))\|\mathbf{X}\|_F^2 \leq \|\boldsymbol{\mathcal{A}}\mathbf{X}\|_2^2 \leq (1+\delta_r(\boldsymbol{\mathcal{A}}))\|\mathbf{X}\|_F^2. \quad (2)$$

We use the short notation δ_r to mean $\delta_r(\boldsymbol{\mathcal{A}})$.

Additional convex constraints: Consider the variant

$$\begin{array}{ll}\underset{\mathbf{X}\in\mathbb{R}^{m\times n}}{\text{minimize}} & f(\mathbf{X}) \\ \text{subject to} & \mathrm{rank}(\mathbf{X}) \leq r,\ \mathbf{X}\in\mathcal{C},\end{array} \quad (3)$$

for a convex set $\mathcal{C}$. Our main interests are $\mathcal{C}_+ = \{\mathbf{X} : \mathbf{X} \succeq 0\}$ and the matrix simplex $\mathcal{C}_\Delta = \{\mathbf{X} : \mathbf{X} \succeq 0,\ \mathrm{trace}(\mathbf{X}) = 1\}$. In both cases the constraints are unitarily invariant and the projection onto these sets can be done by taking the eigenvalue decomposition and projecting the eigenvalues. Furthermore, for these specific $\mathcal{C}$, $\mathcal{P}_{\{\mathbf{X}:\mathrm{rank}(\mathbf{X})\leq r\}\cap\mathcal{C}} = \mathcal{P}_\mathcal{C} \circ \mathcal{P}_r$ (see [14]).

Approximate singular value computations: The standard method to compute a partial SVD is the Lanczos method. However, the method is somewhat hard to parallelize and it lacks theoretical bounds of the form used in Theorem 1.

Algorithm 1 `RandomizedSVD`

Finds Q such that $X \approx \mathcal{P}_Q X = QQ^*X$

Require: Function $\mathtt{h} : \widetilde{Z} \mapsto X\widetilde{Z}$
Require: Function $\mathtt{h}^* : \widetilde{Q} \mapsto X^*\widetilde{Q}$
Require: $r \in \mathbb{N}$ // Rank of output
Require: $q \in \mathbb{N}$ // Number of power iterations to perform
$\ell = r + \rho$ // Typical value of ρ is 5
Ω a $n \times \ell$ standard Gaussian matrix
$W \leftarrow \mathtt{h}(\Omega)$
$Q \leftarrow \mathtt{QR}(W)$ // The QR algorithm to orthogonalize W
for $j = 1, 2, \ldots, q$ **do**
  $Z \leftarrow \mathtt{QR}(\mathtt{h}^*(Q))$
  $Q \leftarrow \mathtt{QR}(\mathtt{h}(Z))$
end for
$Z \leftarrow \mathtt{h}^*(Q)$
$(U, \Sigma, V) \leftarrow$ `FactoredSVD`(Q, I_ℓ, Z)
Let Σ_r be the best rank r approximation of Σ
return (U, Σ_r, V)

Algorithm 2 `FactoredSVD`$(\widetilde{U}, \widetilde{D}, \widetilde{V})$

Computes the SVD $U\Sigma V^*$ of the matrix X implicitly given by $X = \widetilde{U}\widetilde{D}\widetilde{V}^*$

$(U, R_U) \leftarrow \mathtt{QR}(\widetilde{U})$
$(V, R_V) \leftarrow \mathtt{QR}(\widetilde{V})$
$(u, \Sigma, v) \leftarrow$ `DenseSVD`$(R_U\widetilde{D}R_V^*)$
return $(U, \Sigma, V) \leftarrow (Uu, \Sigma, Vv)$

As an alternative, we turn to randomized linear algebra. On this front, we restrict ourselves to algorithms that require only multiplications, as opposed to sub-sampling entries/rows/columns, as sub-sampling is not efficient for the application we present. The randomized approach presented in Algorithm 1 has been rediscovered many times, but has seen a recent resurgence of interest due to theoretical analysis [6]:

Theorem 1 (Average Frobenius error)**.** *Suppose $\mathbf{X} \in \mathbb{R}^{m\times n}$, and choose a target rank r and oversampling parameter $\rho \geq 2$ where $\ell := r + \rho \leq \min\{m, n\}$. Calculate Q and $\mathcal{P}_Q$ via `RandomizedSVD` using $q = 0$ and set $\widetilde{\mathbf{X}} = \mathcal{P}_Q\mathbf{X}$. Then*

$$\mathbb{E}\|\mathbf{X} - \widetilde{\mathbf{X}}\|_F^2 \leq (1+\epsilon)\,\|\mathbf{X} - \mathbf{X}_r\|_F^2$$

where $\mathbf{X}_r$ is the best rank r approximation in the Frobenius norm of $\mathbf{X}$, $\widetilde{\mathbf{X}}$ has rank ℓ, and $\epsilon = \frac{r}{\rho-1}$.

The theorem follows from the proof of Thm. 10.5 in [6]. The expectation is with respect to the Gaussian r.v. in `RandomizedSVD`. For the sake of our analysis, we cannot immediately truncate $\widetilde{\mathbf{X}}$ to rank r since then the error bound in [6] is not tight enough. Thus, since $\widetilde{X}$ is rank ℓ, in practice we even observe that $\|\mathbf{X} - \widetilde{\mathbf{X}}\|_F^2 < \|\mathbf{X} - \mathbf{X}_r\|_F^2$, especially for small r, as shown in Figure 3. The figure also shows that using $q > 0$ power iterations is extremely helpful, though this is not taken into account in our analysis since there are no useful theoretical bounds. Note that variants for eigenvalues also exist; we refer to the equivalent of `RandomizedSVD` as `RandomizedEIG`, which has the property that $U = V$ and Σ need not be positive (cf., [6])

III. A projected gradient descent algorithm

Our approach is based on the projected gradient descent:

$$\mathbf{X}_{i+1} = \mathcal{P}_r^\epsilon(\mathbf{X}_{i+1} - \mu_i\nabla f(\mathbf{X}_i)), \quad (4)$$

where $\mathbf{X}_i$ is the i-th iterate, $\nabla f(\cdot)$ is the gradient of the loss function, μ_i is a step-size, and $\mathcal{P}_r^\epsilon(\cdot)$ is the approximate projector onto rank r matrices given by `RandomizedSVD`. If we include a convex constraint $\mathcal{C}$, then the iteration is

$$\mathbf{X}_{i+1} = \mathcal{P}_\mathcal{C}(\mathcal{P}_r^\epsilon(\mathbf{X}_{i+1} - \mu_i\nabla f(\mathbf{X}_i))). \quad (5)$$

In practice, Nesterov acceleration improves performance:

$$\mathbf{Y}_{i+1} = (1+\beta_i)\mathbf{X}_i - \beta_i\mathbf{X}_{i-1} \quad (6)$$
$$\mathbf{X}_{i+1} = \mathcal{P}(\mathbf{Y}_i - \mu_i\nabla f(\mathbf{Y}_i)), \quad (7)$$

Algorithm 3 Efficient implementation of SVP, $\mathcal{K} = \{\mathbb{R}, \mathbb{C}\}$

Require: step-size $\mu > 0$, measurements $\mathbf{y}$, initial points $u_0 \in \mathcal{K}^{m\times r}$, $v_0 \in \mathcal{K}^{n\times r}$, $d_0 \in \mathcal{K}^r$, (opt.) unitarily invariant convex set $\mathcal{C}$

Require: Function $\mathtt{A} : (u, d, v) \mapsto \boldsymbol{\mathcal{A}}(u\,\mathrm{diag}(d)v^*)$

Require: Function $\mathtt{At} : (\mathbf{z}, w) \mapsto \boldsymbol{\mathcal{A}}^*(\mathbf{z})w$

Require: Function $\mathtt{At}^* : (\mathbf{z}, w) \mapsto (\boldsymbol{\mathcal{A}}^*(\mathbf{z}))^*w$

$v_{-1} \leftarrow 0$, $u_{-1} \leftarrow 0$, $d_{-1} \leftarrow 0$

for $i = 0, 1, \ldots$ **do**

  Compute β_i // See text

  $u_y \leftarrow [u_i, u_{i-1}]$, $v_y \leftarrow [v_i, v_{i-1}]$

  $d_y \leftarrow [(1+\beta_i)d_i, -\beta_i d_{i-1}]$

  $\mathbf{z} \leftarrow A(u_y, d_y, v_y) - \mathbf{y}$ // Compute the residual

  Define the functions

  $\mathtt{h} : w \mapsto u_y\,\mathrm{diag}(d_y)v_y^*w - \mu\mathtt{At}(\mathbf{z}, w)$

  $\mathtt{h}^* : w \mapsto v_y\,\mathrm{diag}(d_y)u_y^*w - \mu\mathtt{At}^*(\mathbf{z}, w)$

  $(u_{i+1}, d_{i+1}, v_{i+1}) \leftarrow \mathtt{RandomizedSVD}(\mathtt{h}, \mathtt{h}^*, r)$

  $d_{i+1} \leftarrow \mathcal{P}_{\mathcal{C}}(d_{i+1})$ // Optional

end for

return $X \leftarrow u_i d_i v_i^*$ // If desired

where β_i is chosen $\beta_i = (\alpha_{i-1} - 1)/\alpha_i$ and $\alpha_0 = 1$, $2\alpha_{i+1} = 1 + \sqrt{4\alpha_i^2 + 1}$ [15] (see [2]). Algorithm 3 shows details for low-memory implementation. The implementation of `A` and `At` depends on the structure of $\boldsymbol{\mathcal{A}}$ in the specific problem.

IV. Convergence

We assume the observations are generated by $\mathbf{y} = \boldsymbol{\mathcal{A}}\mathbf{X}^\star + \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon}$ is a noise term (not to be confused with ϵ). In the following theorem, we will assume that $\|\boldsymbol{\mathcal{A}}\|^2 \le mn/p$, which is true for quantum tomography [16]; if $\boldsymbol{\mathcal{A}}$ is a normalized Gaussian, then this assumption holds in expectation.

Theorem 2. *(Iteration invariant) Pick an accuracy $\epsilon = \frac{r}{\rho - 1}$, where ρ is defined as in Theorem 1. Define $\ell = r + \rho$ and let c be an integer such that $\ell = (c-1)r$. Let $\mu_i = \frac{1}{2(1+\delta_{cr})}$ in* (4) *and assume $\|\boldsymbol{\mathcal{A}}\|^2 \le mn/p$ and $f(\mathbf{X}_i) > C^2\|\boldsymbol{\varepsilon}\|^2$, where $C \ge 4$ is a constant. Then the descent scheme* (4) *or* (5) *has the following iteration invariant*

$$\mathbb{E}f(\mathbf{X}_{i+1}) \le \theta f(\mathbf{X}_i) + \tau\|\boldsymbol{\varepsilon}\|^2, \tag{8}$$

in expectation, where

$$\theta \le 12 \cdot \frac{1+\delta_{2r}}{1-\delta_{cr}} \cdot \left(\frac{\epsilon}{1+\delta_{cr}} \cdot \frac{mn}{p} + (1+\epsilon)\frac{3\delta_{cr}}{1-\delta_{2r}} \right),$$

and

$$\tau \le \frac{1+\delta_{2r}}{1-\delta_{cr}} \cdot \left(12 \cdot (1+\epsilon)\left(1 + \frac{2\delta_{cr}}{1-\delta_{2r}}\right) + 8 \right).$$

The expectation is taken with respect to Gaussian random designs in `RandomizedSVD`. If $\theta \le \theta_\infty < 1$ for all iterations, then $\lim_{i\to\infty} \mathbb{E}f(\mathbf{X}_i) \le \max\{C^2, \frac{\tau}{1-\theta_\infty}\}\|\boldsymbol{\varepsilon}\|^2$.

Each call to `RandomizedSVD` draws a new Gaussian r.v., so the expected value does not depend on previous iterations.

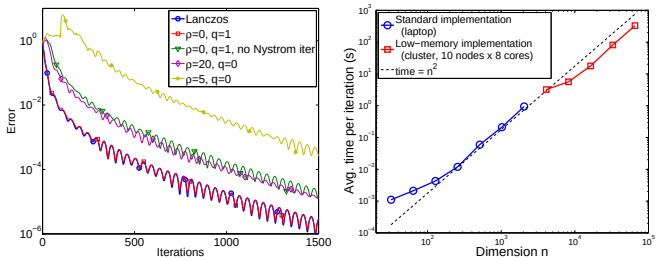

Fig. 1. (Left) Convergence rate as a function of parameters to `RandomizedSVD`/`RandomizedEIG`. (Right) Scaling plot for computation time of `RandomizedEIG`.

The expected value of the function converges linearly at rate θ to within a constant of the noise level, and in particular, it converges to zero when there is no noise.

Unfortunately, the theorem imposes overly pessimistic values for ϵ. The bound on θ should be less than 1 in order to have a contraction. This gives the requirement that $\delta_{cr} \lesssim 1/200$, which is reasonable (cf. [12]). However, it also imposes[2] $\frac{12}{1-\delta_{cr}^2} \cdot \frac{\epsilon\, mn}{p} < \frac{1}{2}$, which means that we need $\epsilon \lesssim \frac{p}{24mn}$. For quantum tomography, $m = n$ and $p = \mathcal{O}(rn)$, so we require $\varepsilon \lesssim r/n$. From Theorem 1, our bound on ϵ is $r/(\rho - 1)$, so we require $\rho \simeq n$, which defeats the purpose of the randomized algorithm (in this case, one would just perform a dense SVD). Surprisingly, numerical examples in the next section show that ρ can be nearly a small constant, so the theory is not sharp.

V. Numerical experiments

We apply the algorithm to the quantum tomography problem, which is a particular instance of (1). For details, we refer to [5]. The salient features are that the variable $\mathbf{X} \in \mathbb{C}^{n\times n}$ is constrained to be Hermitian positive-definite, and that, unlike many low-rank recovery problems, the linear operator $\boldsymbol{\mathcal{A}}$ satisfies the R-RIP: [16] establishes that Pauli measurements (which comprise $\boldsymbol{\mathcal{A}}$) have R-RIP with overwhelming probability when $p = \mathcal{O}(rn\log^6 n)$. In the ideal case, $\mathbf{X}^\star$ is exactly rank 1, but it may have larger rank due to some (non-Gaussian) noise processes, in addition to AWGN $\boldsymbol{\varepsilon}$. Furthermore, it is known that the true solution $\mathbf{X}^\star$ has trace 1, which is also possible to exploit in our algorithmic framework. Each component of the linear operator $\boldsymbol{\mathcal{A}}$ has a special Kronecker product structure, which we exploit in order to keep memory low, using custom parallel code.

Figure 1 (left) plots convergence and accuracy results for a quantum tomography problem with 8 qubits and $p = 4rn$ with $r = 1$. The SVP algorithm works well on noisy problems but we focus here on a noiseless (and truly low-rank) problem in order to examine the effects of approximate SVD/eigenvalue computations. The figure shows that the power method with $q \ge 1$ is extremely effective (if $q = 0$, then $\rho \simeq 20$ still leads to convergence). When p is smaller and the R-RIP is not satisfied, taking ρ or q too small can lead to divergence.

[2]For the details, see the extended version at http://arxiv.org/abs/1303.0167.

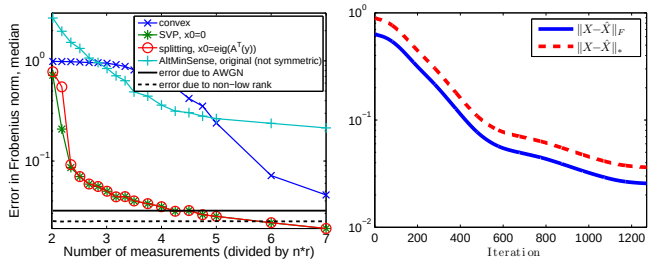

Fig. 2. (Left) Accuracy comparison of several algorithms, as a function of number of samples. (Right) Convergence for the 16-qubit simulation

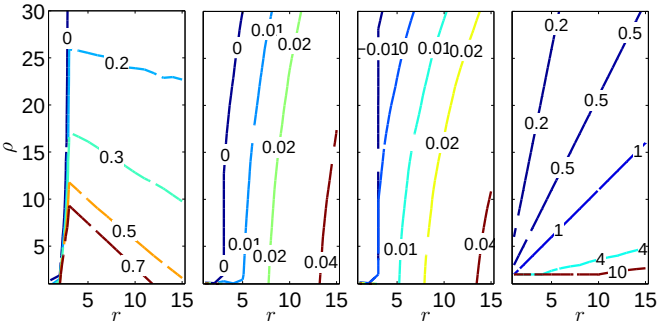

Fig. 3. Left 3 plots are the empirical estimates $\widetilde{\epsilon}$ for $q = 0, 1, 2$ power iterations. Rightmost plot is the theoretical values ϵ from Thm. 1

The right subfigure of Figure 1 shows that the low-memory implementation has time complexity $\mathcal{O}(n^2)$ up to $n = 2^{16}$.

The left subfigure of Figure 2 reports the median error on 10 test problems across a range of p. Here, $\mathbf{X}^\star$ is only approximately low rank and y is contaminated with noise. We compare the convex approach [5], the "AltMinSense" approach [12], and a standard splitting approach. AltMinSense and the convex approach have poor accuracy; the accuracy of AltMinSense can be improved by incorporating symmetry, but this changes the algorithm fundamentally and the theoretical guarantees are lost. The splitting approach, if initialized correctly, is accurate, but lacks guarantees. Furthermore, it is slower in practice due to slower convergence, though for some simple problems it is possible to accelerate using L-BFGS [10].

Figure 3 tests Theorem 1 by plotting the value of

$$\widetilde{\epsilon} = \|\mathbf{X} - \widetilde{\mathbf{X}}\|_F^2 / \|\mathbf{X} - \mathbf{X}_r\|_F^2 - 1$$

(which is bounded by ϵ) for matrices $\mathbf{X}$ that are generated by the iterates of the algorithm. The algorithm is set for $r = 1$ (so $\mathbf{X}$ is the sum of a rank 2 term, which includes the Nesterov term, and the full rank gradient), but the plots consider a range of r and a range of oversampling parameters ρ. Because $\widetilde{\mathbf{X}}$ has rank $\ell = r + \rho$, it is possible for $\widetilde{\epsilon} < 0$, as we observe in the plots when r is small and ρ is large. For two power iterations, the error is excellent. In all cases, the observed error $\widetilde{\epsilon}$ is much better than the bound ϵ from Theorem 1.

Finally, to test scaling to very large data, we compute a 16 qubit state ($n = 65536$), using a known quantum state as input, with realistic quantum mechanical perturbations (global depolarizing noise of level $\gamma = 0.01$; see [5]) as well as AWGN to give a SNR of 30 dB, and $p = 5n = 327680$ measurements. The first iteration uses Lanczos and all subsequent iterations use `RandomizedEIG` using $\rho = 5$ and $q = 3$ power iterations. On a cluster with 10 computers, the mean time per iteration is 401 seconds. After 1270 iterations, $\|\mathbf{X} - \mathbf{X}^\star\|_F = 0.0256$; see Figure 2 (right).

VI. Conclusion

Randomization is a powerful tool to accelerate and scale optimization algorithms, and it can be rigorously included in algorithms that are robust to small errors. In this paper, we leverage randomized approximations to remove memory bottlenecks by merging the two-key steps of most recovery algorithms in affine rank minimization problems: gradient calculation and low-rank projection. Unfortunately, the current black-box approximation guarantees, such as Theorem 1, are too pessimistic to be directly used in theoretical characterizations of our approach. For future work, motivated by the overwhelming empirical evidence of the good performance of our approach, we plan to directly analyze the impact of randomization in characterizing the algorithmic performance.

Acknowledgment: VC and AK's work was supported by MIRG-268398, ERC Future Proof, and SNF 200021-132548. SRB is supported by the Fondation Sciences Mathématiques de Paris. The authors thank Alex Gittens for his insightful comments.

References

[1] R. Meka, P. Jain, and I. S. Dhillon, "Guaranteed rank minimization via singular value projection," in *NIPS*, 2010.
[2] A. Kyrillidis and V. Cevher, "Matrix recipes for hard thresholding methods," *arXiv preprint arXiv:1203.4481*, 2012.
[3] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.
[4] E. Candes and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, pp. 717–772, 2009.
[5] S. Flammia, D. Gross, Y. Liu, and J. Eisert, "Quantum tomography via compressed sensing: error bounds, sample complexity, and efficient estimators," *New J. Phys.*, vol. 14, no. 9, p. 095022, 2012.
[6] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.
[7] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Mathematical Programming Computation*, pp. 1–29, 2010.
[8] B. Recht and C. Ré, "Parallel stochastic gradient algorithms for large-scale matrix completion," *Math. Prog. Comput.*, to appear, 2013.
[9] J. Lee, B. Recht, R. Salakhutdinov, N. Srebro, and J. A. Tropp, "Practical large-scale optimization for max-norm regularization," in *NIPS*, 2011.
[10] S. Laue, "A hybrid algorithm for convex semidefinite optimization," in *ICML*, 2012.
[11] S. Burer and R. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Math. Prog. (series B)*, vol. 95, no. 2, pp. 329–357, 2003.
[12] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *ACM Symp. Theory Comput.*, 2012.
[13] D. Goldfarb and S. Ma, "Convergence of fixed-point continuation algorithms for matrix rank minimization," *Foundations of Computational Mathematics*, vol. 11, no. 2, pp. 183–210, 2011.
[14] S. Becker, V. Cevher, C. Koch, and A. Kyrillidis, "Sparse projections onto the simplex," in *ICML*, to appear, 2013.
[15] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence $\mathcal{O}(1/k^2)$," *Doklady AN SSSR*, translated as *Soviet Math. Docl.*, vol. 269, pp. 543–547, 1983.
[16] Y. K. Liu, "Universal low-rank matrix recovery from Pauli measurements," in *NIPS*, 2011, pp. 1638–1646.