Dropping convexity for faster semi-definite optimization

Srinadh Bhojanapalli
TTI Chicago
bsrinadh@utexas.edu

Anastasios Kyrillidis
UT Austin
anastasios@utexas.edu

Sujay Sanghavi
UT Austin
sanghavi@mail.utexas.edu

Abstract

A matrix $X \in \mathbb{R}^{n \times n}$ is positive semi-definite (PSD) if and only if it can be written as the product $UU^\top$, for some matrix U. This paper uses this observation in optimization: specifically, we consider the minimization of a convex function f over the PSD cone $X \succeq 0$, but via gradient descent on $f(UU^\top)$, which is a non-convex function of U. We focus on the case where U is set to be an $n \times r$ matrix for some $r \leq n$, and correspondingly f satisfies restricted strong convexity.

We propose a novel step size and show that updating U via gradient descent results in linear convergence to the top-r components of the optimum of f; provided we start from a point which has constant relative distance to the optimum. We also develop an initialization scheme for the "first-order oracle" setting.

1 Introduction

This paper considers the following optimization problem[1]:

$$\underset{X \in \mathbb{R}^{n \times n}}{\text{minimize}} \; f(X) \quad \text{subject to} \;\; X \succeq 0, \tag{1}$$

where $f : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$ is a convex and smooth function, and $X \succeq 0$ denotes the convex set over positive semi-definite matrices in $\mathbb{R}^{n \times n}$. In this paper we are interested in solving (1) via the parametrization:

$$\underset{U \in \mathbb{R}^{n \times r}}{\text{minimize}} \quad f(UU^\top) \quad \text{where } r \leq n. \tag{2}$$

This is equivalent to (1) when $r = n$, and otherwise is an approximation.

Note that the new problem has a very specific kind of non-convexity, arising because of representing X as $UU^\top$. In particular, when $r = n$, this means that we are taking the original convex semi-definite optimization problem, and deliberately making it non-convex via this representation. We would choose $r < n$ for computational reasons (as smaller r means lower computational complexity for gradient descent), or statistical reasons (to prevent over-fitting).

Motivation. Problems like (1) commonly arise in optimization in general; within the machine learning domain, a non-exhaustive list of applications includes matrix completion [8, 15, 16, 9], affine rank minimization [14, 2], covariance / inverse covariance selection [13, 17, 23, 12], phase retrieval [21, 6] and sparse PCA [11], just to name a few.

[1] We refer the reader to [3] for a more detailed description of this problem and of our algorithm.

Our motivation for studying the $UU^\top$ parametrization comes from large-scale problem instances. In problems where for example r is much smaller than n, U will be a much smaller matrix than X, making it easier to update, store and iteratively optimize over. Even for the case where $r = n$, standard approaches to solving (1), like projected gradient descent and its accelerated/second-order variants, involve enforcing the $X \succeq 0$ constraint at every iteration; this step can often constitute the primary computational load of the overall iteration.

In contrast, the $UU^\top$ reformulation in (2) automatically encodes the PSD constraint. Applying gradient descent on $f(UU^\top)$ does not require any eigenvalue computation, but the problem is now non-convex. In this paper, we design an efficient initialization procedure, and then prove that updating U via gradient descent converges (fast) to optimal (or near-optimal) solutions.

Contributions. There has been a wide range of works that consider solving (1) in the factorized form for specific f instances and achieve linear convergence rates [15, 21, 24, 25]. To the best of our knowledge, this is the first paper that solves the re-parametrized problem (2) with the same convergence rate guarantees for *general convex functions* f. We assume the *first order oracle* model for access to f; that is, for any matrix X we can obtain the value $f(X)$ and the gradient $\nabla f(X)$. We study how gradient descent, over U, performs in solving (2); this leads to *factored gradient descent* algorithm and corresponds to the update rule

$$U^{+} = U - \eta \nabla f(UU^T) \cdot U.$$

Let $X^\star$ be the solution to (1), and let $X_r^\star$ be the best rank-r approximation (*i.e.*, the top-r spectral components) of $X^\star$. Our contributions in this work can be summarized as follows:

- (i) *Step size rule:* Our main algorithmic contribution is a special choice of the step size η. The crucial insight here is that η needs to depend not only on the convexity parameters of f (as is the case in standard convex optimization) but *also* on the top singular value of the unknown optimum. Section 4 describes the precise step size rule, and also the intuition behind it (via consideration of the second derivative with respect to U).
- (ii) *Correctness and convergence under restricted strong convexity:* For our main result, we consider the case where f has *restricted strong convexity (RSC)*, *i.e.*, f satisfies strong-convexity-like conditions, but only over rank-r matrices. We show that when f has RSC, and we use the step size rule as above, U converges geometrically (*i.e.*, with linear rate) to a region close to $X_r^\star$, when initialized from constant relative distance.
- (iii) *Initialization:* We focus on the case where we only have access to f via the first-order oracle: specifically, we initialize based on the gradient at zero, *i.e.*, $\nabla f(0)$. We show that, for certain condition numbers of f, this yields a constant relative error initialization.

1.1 Related work

We briefly describe the work that utilizes factorization in the Burer and Monteiro [4, 5] sense. [4, 5] popularized the idea of solving classic SDPs by representing the solution as a product of two factor matrices. The main idea in such representation is to remove the positive semi-definite constraint by directly embedding it into the objective. For linear objective f, they establish convergence guarantees to the optimum but do not provide convergence rates.

Specialized algorithms – for objectives beyond the linear case – that utilize such factorization include matrix completion solvers [15], non-negative matrix factorization schemes [19], phase retrieval methods [21, 7, 6] and sparse PCA algorithms [18]. Restricted to the case of matrix completion, [15] shows linear convergence (with $O(\log(1/\varepsilon))$ steps) in solving (2). [24, 25] study the problem of recovering a low-rank PSD matrix from linear measurements. Both these approaches admit linear convergence to the optimal solution by employing a careful initialization step. Nevertheless, both [24, 25] only apply to simple quadratic loss objectives and not to generic convex functions f.

For generic smooth convex functions, [22] use ideas from sparse approximation to greedily refine U factors via rank-1 updates; however, no convergence rate guarantee is provided. Based on similar ideas, [18] propose a sub-linearly convergent (i.e., $O(1/\varepsilon)$ rate) framework, where the rank-1 update is followed by a nonlinear improvement of the current solution using the L-BFGS algorithm.

At the time of submission, we became aware of the work of Chen and Wainwright [10]. There, the authors propose a first-order optimization framework for the problem (1), where the same parametrization technique is used to efficiently accommodate the PSD constraint. Withal, the proposed algorithmic solution can accommodate extra constraints on X. Their results are of the same flavor with ours: under proper assumptions, one can prove local convergence with $O(1/\varepsilon)$ or $O(\log(1/\varepsilon))$ rate and for f instances that even fail to be locally convex.

2 Preliminaries

Assumptions. We will investigate the performance of non-convex gradient descent for functions f that satisfy strong convexity and restricted strong convexity.

Definition 2.1. *Let $f : \mathbb{S}_+^n \to \mathbb{R}$ be a convex differentiable function. Then, f is m-strongly convex if for any $X, Y \in \mathbb{S}_+^n$, the following holds:*

$$f(Y) \geq f(X) + \langle \nabla f(X), Y - X \rangle + \tfrac{m}{2} \|Y - X\|_F^2. \tag{3}$$

Definition 2.2. *f is $(\widehat{m}, r)$-restricted strongly convex if for any rank-r matrices $X, Y \in \mathbb{S}_+^n$:*

$$f(Y) \geq f(X) + \langle \nabla f(X), Y - X \rangle + \tfrac{\widehat{m}}{2} \|Y - X\|_F^2. \tag{4}$$

This definition has previously appeared in [20, 1]. Given the above definitions, we define $\kappa = \frac{M}{m}$ as the condition number of function f.

3 Factored gradient descent

We are interested in solving (2) via gradient descent. For step size η, the update rule is

$$U^+ = U - \eta \nabla f(UU^\top) \cdot U.$$

Factored gradient descent does this, but with two key innovations: initialization and a special step size η. We next provide some intuition behind the η choice. Initialization is discussed in Section 6.

4 Step size

Even though f is restricted strongly convex over $X \succeq 0$, the fact that we operate with the non-convex $UU^\top$ parametrization means that we need to be careful about the step size η; *e.g.*, our *constant* η selection should be such that, when we are close to $X^\star$, we do not "overshoot" the optimum $X^\star$.

To this end, let us consider a simple setting where $U \in \mathbb{R}^{n \times r}$ with $r = 1$; *i.e.*, U is a vector. For clarity, denote it as u. Let f be a separable function with $f(X) = \sum_{ij} f_{ij}(X_{ij})$. Furthermore, for $f : \mathbb{R}^{n \times n} \to \mathbb{R}$, define the function $g : \mathbb{R}^n \to \mathbb{R}$ such that $f(uu^\top) \equiv g(u)$. It is easy to compute:

$$\nabla g(u) = \nabla f(uu^\top) \cdot u \in \mathbb{R}^n \text{ and } \nabla^2 g(u) = \texttt{mat}\left(\texttt{diag}(\nabla^2 f(uu^\top)) \cdot \texttt{vec}\left(uu^\top\right)\right) + \nabla f(uu^\top) \in \mathbb{R}^{n \times n},$$

where $\texttt{mat} : \mathbb{R}^{n^2} \to \mathbb{R}^{n \times n}$, $\texttt{vec} : \mathbb{R}^{n \times n} \to \mathbb{R}^{n^2}$ and, $\texttt{diag} : \mathbb{R}^{n^2 \times n^2} \to \mathbb{R}^{n^2 \times n^2}$ are the matricization, vectorization and diagonalization operations, respectively; for the last case, diag generates a diagonal matrix from the input, discarding its off-diagonal elements. We remind that $\nabla f(uu^\top) \in \mathbb{R}^{n \times n}$ and $\nabla^2 f(uu^\top) \in \mathbb{R}^{n^2 \times n^2}$.[2]

Assume that the current putative estimate u is close to the optimum. Standard convex optimization suggests that η should be chosen $\eta < 1/\|\nabla^2 g(\cdot)\|_2$, in the case when we are close to the optimum. Let us interpret the hessian of g, as described in the expression above. We know that, due to smoothness of f, $\|\nabla^2 f(uu^\top)\|_2 \leq M$ and, by assumption, $uu^\top$ is close to $X^\star$. Similarly, the second term is the gradient at a point close to $X^\star$; our surrogate in this case will be the gradient $\nabla f(X^0)$, where X^0 is the initialization point. This suggests:

$$\eta < \frac{1}{\|\nabla^2 g(\cdot)\|_2} \propto \frac{1}{M\|X^0\|_2 + \|\nabla f(X^0)\|_2}.$$

[2]Note that Hessian is diagonal for a separable function $f(X) = \sum_{ij} f_{ij}(X_{ij})$.

5 Convergence

The following theorem characterizes the convergence rate of our scheme for f that satisfy (m, r)-restricted strong convexity.

Theorem 5.1 (Convergence rate for rank-r estimate of $X^\star$)**.** *Let $f : \mathbb{S}_+^n \to \mathbb{R}$ be a M-smooth and (m, r)-restricted strongly convex function, with restricted condition number $\kappa = \frac{M}{m}$. Let $X^\star$ be its minimum over the set of PSD matrices, such that $\|X^\star - X_r^\star\|_F \leq \frac{\sigma_r(X^\star)}{200\kappa^{1.5}} \frac{\sigma_r(X^\star)}{\sigma_1(X^\star)}$. Let $X^0 = U^0(U^0)^\top$ be a rank-r PSD matrix such that $\text{Dist}(U^0, U_r^\star) \leq \rho\sigma_r(U_r^\star)$, for $\rho = \frac{1}{100\kappa} \frac{\sigma_r(X^\star)}{\sigma_1(X^\star)}$. Let current iterate be U and $X = UU^\top$. Let $\text{Dist}(U, U_r^\star) \leq \rho\sigma_r(U_r^\star)$ and set the step size as $\eta = \frac{1}{16\,(M\|X^0\|_2 + \|\nabla f(X^0)\|_2)}$. Then, the new estimate $U^+ = U - \eta\nabla f(X) \cdot U$ satisfies*

$$\text{Dist}(U^+, U_r^\star)^2 \leq \alpha \cdot \text{Dist}(U, U_r^\star)^2 + \beta \cdot ||X^\star - X_r^\star||_F^2, \tag{5}$$

where $\alpha = 1 - \frac{m\sigma_r(X^\star)}{64(M\|X^\star\|_2 + \|\nabla f(X^\star)\|_2)}$ and $\beta = \frac{M}{28(M\|X^\star\|_2 + \|\nabla f(X^\star)\|_2)}$. Further, U^+ satisfies $\text{Dist}(U^+, U_r^\star) \leq \rho\sigma_r(U_r^\star)$.

The theorem states that provided we (i) choose the step size based on a point that is constant relative distance to $U_r^\star$, and (ii) we start from such a point, gradient descent on U will converge linearly to a neighborhood of $U_r^\star$. The above theorem immediately implies linear convergence rate for the setting where f satisfies standard strong convexity with parameter m. This follows from observing that standard strong convexity implies restricted strong convexity for all values of rank r.

Corollary 5.2 (Exact recovery of $X^\star$)**.** *Let $X^\star$ be the optimal point of f, over the set of PSD matrices, such that $rank(X^\star) = r$. Consider X as in Theorem 5.1. Then, under the same assumptions and with the same convergence factor α as in Theorem 5.1, we have*

$$\text{Dist}(U^+, U^\star)^2 \leq \alpha \cdot \text{Dist}(U, U^\star)^2.$$

Further for $r = n$ we recover the exact case of semi-definite optimization.

6 Initialization

In the previous section we have seen that gradient descent over U achieves linear convergence once the iterates are closer to the optimum $U_r^\star$. Since the overall problem is non-convex, intuition suggests that we need to start from a "decent" initial point, in order to get provable convergence to the global optimum. One way to satisfy this condition is to use one of the standard convex algorithms to obtain U within constant error to $U^\star$ and switch to factored gradient descent to get the high precision solution. In this section we present a new way to compute initialization for general smooth and strong convex f. The results extend to the case where the optimum $X^\star$ is of rank-r.

Theorem 6.1 (Initialization)**.** *Let f be a M-smooth and m-strongly convex function, with condition number $\kappa = \frac{M}{m}$, and let $X^\star$ be its minimum over PSD matrices. Let X^0 be defined as:*

$$X^0 := \frac{1}{\|\nabla f(0) - \nabla f(e_1 e_1{}^\top)\|_F} \mathcal{P}_+ \left(-\nabla f(0)\right), \tag{6}$$

and X_r^0 is its rank-r approximation. Let $\|X^\star - X_r^\star\|_F \leq \tilde{\rho}\|X_r^\star\|_2$ for some $\tilde{\rho}$. Then, $\text{Dist}(U_r^0, U_r^\star) \leq \gamma\sigma_r(U_r^\star)$, where $\gamma = 4\tau(X_r^\star)\sqrt{2r} \cdot \left(\sqrt{\kappa^2 - 2/\kappa + 1}\left(srank^{1/2}(X_r^\star) + \tilde{\rho}\right) + \tilde{\rho}\right)$.

While the above result guarantees a good initialization for only small values of κ, in many applications [15, 21, 10], this is indeed the case and X^0 has constant relative error to the optimum.

7 Conclusion

In this paper, we focus on how to efficiently minimize a convex function f over the positive semi-definite cone. Inspired by the seminal work [4, 5], we drop convexity by factorizing the optimization variable $X = UU^\top$ and show that *factored gradient descent* with a non-trivial step size selection results in linear convergence, even though the problem is now non-convex. In addition, we present a new initialization scheme that uses only first order information and guarantees to find a starting point with small relative distance from optimum.

References

[1] A. Agarwal, S. Negahban, and M. J. Wainwright. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In *Advances in Neural Information Processing Systems*, pages 37–45, 2010.

[2] S. Becker, V. Cevher, and A. Kyrillidis. Randomized low-memory singular value projection. In *10th International Conference on Sampling Theory and Applications (Sampta)*, 2013.

[3] S. Bhojanapalli, A. Kyrillidis, and S. Sanghavi. Dropping convexity for faster semi-definite optimization. *arXiv preprint arXiv:1509.03917*, 2015.

[4] S. Burer and R. D. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[5] S. Burer and R. D. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.

[6] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. *SIAM Review*, 57(2):225–251, 2015.

[7] E. J. Candes and X. Li. Solving quadratic equations via phaselift when there are about as many equations as unknowns. *Foundations of Computational Mathematics*, 14(5):1017–1026, 2014.

[8] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[9] Y. Chen, S. Bhojanapalli, S. Sanghavi, and R. Ward. Coherent matrix completion. In *Proceedings of The 31st International Conference on Machine Learning*, pages 674–682, 2014.

[10] Y. Chen and M. J. Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*, 2015.

[11] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet. A direct formulation for sparse pca using semidefinite programming. *SIAM review*, 49(3):434–448, 2007.

[12] Q. T. Dinh, A. Kyrillidis, and V. Cevher. Composite self-concordant minimization. *Journal of Machine Learning Research*, 16:371–416, 2015.

[13] C.-J. Hsieh, I. S. Dhillon, P. K. Ravikumar, and M. A. Sustik. Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing Systems*, pages 2330–2338, 2011.

[14] P. Jain, R. Meka, and I. S. Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems*, pages 937–945, 2010.

[15] P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 665–674. ACM, 2013.

[16] A. Kyrillidis and V. Cevher. Matrix recipes for hard thresholding methods. *Journal of mathematical imaging and vision*, 48(2):235–265, 2014.

[17] A. Kyrillidis, R. Karimi Mahabadi, Q. Tran Dinh, and V. Cevher. Scalable sparse covariance estimation via self-concordance. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[18] S. Laue. A hybrid algorithm for convex semidefinite optimization. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 177–184, 2012.

[19] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[20] S. Negahban and M. J. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *The Journal of Machine Learning Research*, 13(1):1665–1697, 2012.

[21] P. Netrapalli, P. Jain, and S. Sanghavi. Phase retrieval using alternating minimization. In *Advances in Neural Information Processing Systems*, pages 2796–2804, 2013.

[22] S. Shalev-shwartz, A. Gonen, and O. Shamir. Large-scale convex minimization with a low-rank constraint. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 329–336, 2011.

[23] Q. Tran Dinh, A. Kyrillidis, and V. Cevher. A proximal newton framework for composite minimization: Graph learning without cholesky decompositions and matrix inversions. In *Proceedings of the 30th International Conference on Machine Learning*, number EPFL-CONF-183012, 2013.

[24] S. Tu, R. Boczar, M. Soltanolkotabi, and B. Recht. Low-rank solutions of linear matrix equations via procrustes flow. *arXiv preprint arXiv:1507.03566*, 2015.

[25] Q. Zheng and J. Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. *arXiv preprint arXiv:1506.06081*, 2015.