k-SVD with Gradient Descent

Yassir Jedra
Imperial College London
y.jedra@imperial.ac.uk

Devavrat Shah
MIT
devavrat@mit.edu

Abstract

The emergence of modern compute infrastructure for iterative optimization has led to great interest in developing optimization-based approaches for a scalable computation of k-SVD, i.e., the $k \geq 1$ largest singular values and corresponding vectors of a matrix of rank $d \geq 1$. Despite lots of exciting recent works, all prior works fall short in this pursuit. Specifically, the existing results are either for the *exact-parameterized* (i.e., $k = d$) and *over-parameterized* (i.e., $k > d$) settings; or only establish local convergence guarantees; or use a step-size that requires problem-instance-specific oracle-provided information. In this work, we complete this pursuit by providing a gradient-descent method with a simple, universal rule for step-size selection (akin to pre-conditioning), that provably finds k-SVD for matrix of any rank $d \geq 1$. We establish that the gradient method with random initialization enjoys global linear convergence for any $k, d \geq 1$. Our convergence analysis reveals that the gradient method has an attractive region, and within this attractive region, the method behaves like Heron's method (a.k.a. the Babylonian method). Our analytic results about the said attractive region imply that the gradient method can be enhanced by means of Nesterov's momentum-based acceleration technique. The resulting improved convergence rates match those of rather complicated methods typically relying on Lanczos iterations or variants thereof.

1 Introduction

The task. Consider $M \in \mathbb{R}^{m \times n}$ a matrix of rank $d \leq m \wedge n$. Let SVD of M be given as $M = U\Sigma V^\top$, where $\Sigma = \mathrm{diag}(\sigma_1, \dots, \sigma_d) \in \mathbb{R}^{d \times d}$, with $\sigma_1, \dots, \sigma_d$ being the singular values of M in decreasing order, $U \in \mathbb{R}^{n \times d}$ (resp. $V \in \mathbb{R}^{d \times d}$) be semi-orthogonal matrix containing the left (resp. right) singular vectors.[1] Our objective is to find the the k-SVD of M, i.e. the leading k singular values, $\sigma_i, i \leq k$, corresponding left (resp. right) singular vectors $u_i, i \leq k$ (resp. $v_i, i \leq k$). For ease and clarity of exposition, we will consider $M \in \mathbb{R}^{n \times n}$ that is symmetric, positive semi-definite in which case $u_i = v_i$ for all $i \in [d]$. Indeed, we can always reduce the problem of k-SVD for any matrix M to that of solving the k-SVD of $MM^\top$ and $M^\top M$, which are both symmetric, positive semi-definite.

A bit of history. Singular Value Decomposition (SVD) is an essential tool in modern machine learning with applications spanning numerous fields such as biology, statistics, engineering, natural language processing, econometric and finance, etc (see [12] for a non-exhaustive list of examples). It is a fundamental linear algebraic operation with a very rich history (see [45] for an early history).

[1] We adopt the convention that the ℓ-th element of the diagonal of Σ is σ_ℓ, and that ℓ-th column of U (resp. of V) denoted u_ℓ (resp. denoted v_ℓ) are its corresponding ℓ-th left (resp. left) singular vector. The condition number of M is defined as $\kappa = \sigma_1/\sigma_d$.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MLxOR: Mathematical Foundations and Operational Integration of Machine Learning for Uncertainty-Aware Decision-Making.

Traditional algorithms for performing SVD or related problems like principal component analysis, or eigenvalue and eigenvector computation, have mostly relied on iterative methods such as the Power method [37], Lanczos method [34], the QR algorithm [20], or variations thereof for efficient and better numerical stability [15]. Notably, these methods have also been combined with stochastic approximation schemes to handle streaming and random access to data [42]. The rich history of these traditional algorithms still impacts the solutions of modern machine learning questions [**?** 2, 27, 22, 43, 23, 1, 49].

Why study a gradient-based method? Given this rich history, one wonders why look for another method? Especially a gradient-based one. To start with, the k-SVD problem is typically formulated as a non-convex optimization problem [3] and the ability to understand success (or failure) of gradient-based methods for non-convex settings can expand our understanding for optimization in general. For example, the landscape of loss function associated with PCA has served as a natural nonconvex surrogate to understand landscape of solutions that arise in training for neural networks [6]. It is also worth mentioning some of the recent works have been motivated by this very reason [24, 26, 35, 14]. Moreover, gradient-based methods are known to be robust with respect to noise or missing values [28]. For example, problems like matrix completion [13] or phase retrieval [9] can be formulated as nonconvex matrix factorization problems and are typically solved using (stochastic) gradient-based methods (see [14] and references therein). This is precisely what fueled the recent interest in understanding gradient methods for non-convex matrix factorization [16, 29, 46, 16, 29, 46, 51, 30, 36]. Finally, the recent emergence of scalable compute coupled with software infrastructure for iterative optimization like gradient-based methods can naturally enable computation of k-SVD for large scale matrices in a seamless manner, and potentially overcome existing challenges with the scaling for SVD computation, see [47].

Question of interest: k-SVD with gradient descent. This work aims to develop gradient descent method for k-SVD for any matrix. This is similar in spirit to earlier works devoted to developing gradient-based approaches for solving maximum eigen-pair problems [3, 4, 5, 38, 21, 31, 44]. These works proposed methods that, indeed, leverage gradient information, but their design is different from that of the standard gradient descent methods and their global convergence guarantees remain poorly understood. Recent works on gradient descent for non-convex matrix factorization [16, 29, 46, 16, 29, 46, 51, 30, 36] are also useful for computing k-SVD; however, they fall short in doing so due to various limitations (see Table 1 for a summary and discussion of related works in §A). This has left the question of whether gradient descent can provably compute k-SVD for any matrix unresolved.

Table 1: Here, we contextualize and compare our contributions to prior work. Specifically, on computing k-SVD using gradient-descent for non-convex matrix factorization (i.e., minimizing $\|M - XX^\top\|_F^2$ over $X \in \mathbb{R}^{n\times k}$).

	Parametrization Regime	Linear Convergence	Step-size Selection		Random Initialization	(Local) Acceleration
			Parameter-free	Pre-conditioning		
Prior work	$(k > d)$	[32, 46, 51]	[51]	[51, 36]	[36]	–
	$(k = d)$	[48, 51, 32, 30, 46]	[48, 51, 30]	[30, 36]	[30, 36]	–
	$(k < d)$	[14, 32]	–	[36]	[36]	–
This work	(any k)	✓	✓	✓	✓	✓

Summary of Contributions. The primary contribution of this work is to provide a gradient descent method for k-SVD for any given matrix. The method is parameter-free, enjoys global linear convergence, and works for any setting. In contrast to prior works, as summarized in Table 1, this is the first of such result for the *under-parametrized* setting, i.e. k less than rank of matrix, including $k = 1$. The method can also be immediately enhanced by means of acceleration techniques such as that of Nesterov's. The accelerated method is algorithmically simple, yet enjoys an improved performance. Empirical results corroborate our theoretical findings. The proof of the global linear convergence result is novel. Critical to the analysis is the observation that the gradient descent method for k-SVD is similar to the classical Heron's method [2] (a.k.a. Babylonian method) for root finding. This offers

[2]To find the square root of number a, Heron's method consist in running the iterations $z_{t+1} = \frac{1}{2}(z_t + \frac{a}{z_t})$ starting with some initial point $z_1 > 0$. Heron's method is guaranteed to converge to $\sqrt{a}$ at a quadratic rate.

an interesting explanation to why pre-conditioning works. We believe this insight might shed further light in the study of gradient descent for generic non-convex loss landscapes.

2 Main Results

Gradient descent for k-SVD. Like the power method, the algorithm proceeds by sequentially finding one singular value and its corresponding vector at a time, in a decreasing order until all the $k \geq 1$ leading vectors are found. To find the top singular value and vector of M, we minimize the objective

$$g(x; M) = \frac{1}{4}\|M - xx^\top\|_F^2. \tag{1}$$

Gradient-descent starts by randomly sampling an initial point $x_0 \in \mathbb{R}^n$ as follows:

$$x_0 = Mx, \qquad \text{with} \qquad x \sim \mathcal{N}(0, I_n), \tag{2}$$

then updates for $t \geq 0$,

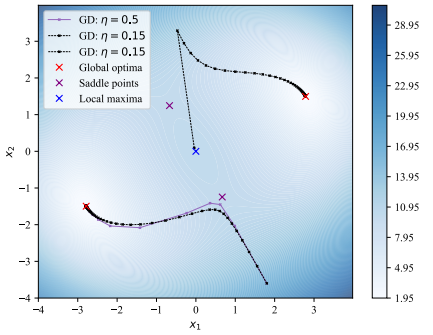
$$x_{t+1} = x_t - \frac{\eta}{\|x_t\|^2}\nabla g(x_t; M), \tag{3}$$

where $\eta \in (0, 1)$. For the above algorithm, we establish the following:

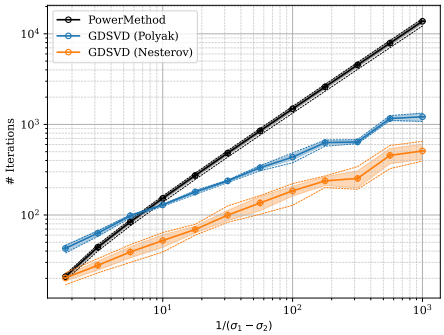
Theorem 2.1. *Let $\epsilon > 0$ and $M \in \mathbb{R}^{n\times n}$ be a symmetric, positive semi-definite with $\sigma_1 - \sigma_2 > 0$. Running gradient descent iterations as described in (3) with the choice $\eta = 1/2$, ensures that for $t \geq 1$, $|\|x_t\|^2 - \sigma_1| \leq \epsilon\sigma_1$, $\|\|x_t\|^{-1}x_t - u_1\| \wedge \|\|x_t\|^{-1}x_t + u_1\| \leq \epsilon$, and $\|x_t + \sqrt{\sigma_1}u_1\| \wedge \|x_t - \sqrt{\sigma_1}u_1\| \leq \epsilon$ [3], for any $\epsilon \in (0, 1)$, so long as*

$$t \geq \underbrace{\frac{c_1\sigma_1}{\sigma_1 - \sigma_2}\log\left(\frac{e}{\epsilon}\frac{\sigma_1}{(\sigma_1 - \sigma_2)}\right)}_{\text{number of iterations to converge within attracting region}} + \underbrace{c_2\log\left(e\left(\frac{1}{\sigma_1} + \sigma_1\right)\right)}_{\text{number of iterations to reach attracting region}}. \tag{4}$$

where c_1, c_2 are constants that only depend on the initial point x_0; with the random initialization (2), the constants c_1, c_2 are almost surely strictly positive.



(a) *The Gradient Descent Trajectory.* We generate a random matrix M of dimension 2×2 and visualize the trajectories of the iterates $(x_t)_{t\geq 0}$ when running the gradient descent iterations (3) with specified initial points along the loss landscape g.



(b) *Gap vs. Number of Iterations.* Comparison between the power method and accelerated variants of gradient descent when minimizing g for different values of the gap $\sigma_1 - \sigma_2$.

In Figure 1a, we illustrate the convergence result of the gradient descent method. We remark that the condition $\sigma_1 - \sigma_2 > 0$ is not necessary for gradient descent (3) to converge. We only require it to simplify the exposition of our results and analysis. Indeed, when $\sigma_1 = \sigma_2 = \cdots = \sigma_i$ and $\sigma_i - \sigma_{i+1} > 0$ for some $i < d$, it can be shown that the iterations (3) still converge to some $x_\star$ where $\|x_\star\| = \sqrt{\sigma_1}$, and $x_\star \in \mathrm{Span}\{u_1, \ldots, u_i\}$ with a numer of iterations that is of order

[3] The notation $\wedge$ is for $\max$ and $\vee$ is for $\min$.

$\widetilde{\Omega}\left(\frac{\sigma_i}{\sigma_i-\sigma_{i+1}}\log\left(\frac{1}{\epsilon}\right)\right)$. Furthermore, the requirement that M must be a symmetric, positive semi-definite matrix can be relaxed to that of M being simply a symmetric semi-definite matrix with $\sigma_1(M) = \max_{i\in[d]}|\sigma_i|$. Indeed, our proofs naturally extend under this more general setting, and this requirement is only made to simplify the analysis.

As a consequence of Theorem 2.1 it immediately follows that by sequential application of (3), we can recover all k singular values and vectors (up to a desired precision).

Theorem 2.2. *Let $\epsilon > 0$ and $M \in \mathbb{R}^{n\times n}$ be a symmetric, positive semi-definite matrix with $\frac{\sigma_i-\sigma_{i+1}}{2\sigma_i} \geq \epsilon$ for all $i \in [k]$. Sequential application of (3) with $\eta = 1/2$ and the random initialization (2), recovers $\hat{\sigma}_1, \ldots, \hat{\sigma}_k$, and $\hat{u}_1, \ldots, \hat{u}_k$ such that: $|\hat{\sigma}_i - \sigma_i| \leq \epsilon\sigma_i$ and $\|\hat{u}_i + u_i\| \wedge \|\hat{u}_i - u_i\| \leq \epsilon$, so long as the number of iterations, denoted t, per every application of (3) satisfies:*

$$t \geq C_1 k \max_{i\in[k]}\left(\frac{\sigma_i}{\sigma_i-\sigma_{i+1}}\right)\log\left(\frac{\sigma_1}{\sigma_k}\max_{i\in[k]}\left(\frac{\sigma_i}{(\sigma_i-\sigma_{i+1})}\right)\frac{k}{\epsilon}\right) + C_2\log\left(e\left(\sigma_1+\frac{1}{\sigma_k}\right)\right), \quad (5)$$

where C_1 and C_2 are constants that are almost surely strictly positive.

Accelerated gradient descent for k-SVD. Gradient methods can achieve better performance when augmented with acceleration schemes [41, 19]. Building upon the known literature, we propose to accelerate (3) as follows:

$$\begin{aligned} y_t &= x_t + \frac{\sqrt{\rho}}{1+\sqrt{\rho}}(v_t - x_t) \\ x_{t+1} &= y_t - \frac{\eta}{\|y_t\|^2}\nabla g(y_t), \\ v_{t+1} &= \left(1-\frac{1}{\sqrt{\rho}}\right)v_t + \frac{1}{\sqrt{\rho}}\left(y_t - \frac{1}{\mu}\nabla g(y_t)\right), \end{aligned} \quad (6)$$

where $\eta, \mu, \rho > 0$. This accelerated gradient descent method is an adaptation of the general scheme of the so-called optimal method proposed by Nesterov [41][Chapter 2.2.1]. We establish in Theorem 2.3 below, that this scheme offers improved bounds then those obtained in Theorem 2.1, at the expense of the convergence being only local.

Theorem 2.3. *Let $M \in \mathbb{R}^{n\times n}$ be a symmetric, positive semi-definite matrix with $\sigma_1 - \sigma_2 > 0$. Assume that $x_0 \in \mathbb{R}^n$, such that $\|x_0 \pm \sqrt{\sigma_1}u_1\| \leq (\sigma_1-\sigma_2)^{3/2}/(90\sqrt{2}\sigma_1)$, $v_0 = x_0$. Running the accelerated gradient-descent iterations (6) with $\eta = 1/6$, $\mu < (\sigma_1-\sigma_2)/4$, $\rho = 9\sigma_1/\mu$, ensures*

$$\frac{1}{\sigma_1^{3/2}}\left|\frac{\|x_{t+1}\|}{\sqrt{\sigma_1}} - 1\right|^2 \vee \sqrt{\sigma_1}\left\|\frac{x_{t+1}}{\|x_{t+1}\|} \pm u_1\right\|^2 \lesssim (1-\sqrt{\rho})^t \wedge \left(\frac{\rho\sqrt{\mu}}{2\sqrt{\rho}+t^2}\right) \quad (7)$$

When one chooses μ of order $\Theta(\sigma_1-\sigma_2)$, and sets $\rho = \Theta(\frac{\sigma_1}{\sigma_1-\sigma_2})$, then in view Theorem 2.3, the gradient descent scheme (6) enjoys a convergence rate of order $\left(1-\Theta(\sqrt{\frac{\sigma_1}{\sigma_1-\sigma_2}})\right)^t$ after t iterations. Thus, to achieve an approximation error of order ϵ, we require $\Omega\left(\sqrt{\frac{\sigma_1}{\sigma_1-\sigma_2}}\log\left(\frac{1}{\epsilon}\right)\right)$ iterations. In contrast, the gradient descent iterations (3), in view of Theorem 2.1, would require $\widetilde{\Omega}\left(\frac{\sigma_1}{\sigma_1-\sigma_2}\log\left(\frac{1}{\epsilon}\right)\right)$ iterations. Finally, we also remark that the presented result shows a gap-independent rate which is similar in spirit to the results of [39, 1].

While indeed the accelerated gradient scheme (6) requires to adequately choose ρ and μ to attain improved convergence rates and properly initialize the method. For empirical purposes, we propose a simpler version (6) where the number of parameters to choose. Start with random initial point x_0 as per (2), then run the iterations:

$$x_{t+1} = x_t + \beta(x_t - x_{t-1}) - \frac{\eta}{\|y_t\|^2}\nabla g(y_t; M), \quad \text{and,} \quad y_t = x_t + \alpha(x_t - x_{t-1}) \quad (8)$$

where parameter $\eta, \beta \in (0,1)$ and $\alpha \in \{0, \beta\}$. We set $\alpha = \beta$ (resp. $\alpha = 0$), to recover a reminiscent version of Nesterov's acceleration [41] (resp. Polyak's heavy ball method [40]) but with an adaptive step-size selection rule. Both methods are appealingly simple. Moreover, as demonstrated in Figure 1b, both acceleration schemes yield a performance improvement from $\frac{\sigma_1}{\sigma_1-\sigma_2}$ to $\sqrt{\frac{\sigma_1}{\sigma_1-\sigma_2}}$ when β is well chosen. This suggests that gradient descent with the acceleration schemes (8) is globally convergent, despite Theorem 2.3 only showing local convergence.

References

[1] Zeyuan Allen-Zhu and Yuanzhi Li. Lazysvd: Even faster svd decomposition yet without agonizing pain. *Advances in neural information processing systems*, 29, 2016.

[2] Raman Arora, Andrew Cotter, Karen Livescu, and Nathan Srebro. Stochastic optimization for pca and pls. In *2012 50th annual allerton conference on communication, control, and computing (allerton)*, pages 861–868. IEEE, 2012.

[3] Giles Auchmuty. Unconstrained variational principles for eigenvalues of real symmetric matrices. *SIAM journal on mathematical analysis*, 20(5):1186–1207, 1989.

[4] Giles Auchmuty. Variational principles for eigenvalues of nonsymmetric matrices. *SIAM Journal on Matrix Analysis and Applications*, 10(1):105–117, 1989.

[5] Giles Auchmuty. Globally and rapidly convergent algorithms for symmetric eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 12(4):690–706, 1991.

[6] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

[7] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *2010 48th Annual allerton conference on communication, control, and computing (Allerton)*, pages 704–711. IEEE, 2010.

[8] Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical programming*, 95(2):329–357, 2003.

[9] Emmanuel J Candes, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, 2015.

[10] Joshua Cape, Minh Tang, and Carey E Priebe. The two-to-infinity norm and singular subspace geometry with applications to high-dimensional statistics. 2019.

[11] Yuxin Chen, Yuejie Chi, Jianqing Fan, and Cong Ma. Gradient descent with random initialization: Fast global convergence for nonconvex phase retrieval. *Mathematical Programming*, 176:5–37, 2019.

[12] Yuxin Chen, Yuejie Chi, Jianqing Fan, Cong Ma, et al. Spectral methods for data science: A statistical perspective. *Foundations and Trends® in Machine Learning*, 14(5):566–806, 2021.

[13] Yuxin Chen, Yuejie Chi, Jianqing Fan, Cong Ma, and Yuling Yan. Noisy matrix completion: Understanding statistical guarantees for convex relaxation via nonconvex optimization. *SIAM journal on optimization*, 30(4):3098–3121, 2020.

[14] Yuejie Chi, Yue M Lu, and Yuxin Chen. Nonconvex optimization meets low-rank matrix factorization: An overview. *IEEE Transactions on Signal Processing*, 67(20):5239–5269, 2019.

[15] Jane K Cullum and Ralph A Willoughby. *Lanczos algorithms for large symmetric eigenvalue computations: Vol. I: Theory*. SIAM, 2002.

[16] Christopher De Sa, Christopher Re, and Kunle Olukotun. Global convergence of stochastic gradient descent for some non-convex matrix problems. In *International conference on machine learning*, pages 2332–2341. PMLR, 2015.

[17] James Demmel and William Kahan. Accurate singular values of bidiagonal matrices. *SIAM Journal on Scientific and Statistical Computing*, 11(5):873–912, 1990.

[18] Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. *Advances in neural information processing systems*, 31, 2018.

[19] Alexandre d'Aspremont, Damien Scieur, Adrien Taylor, et al. Acceleration methods. *Foundations and Trends® in Optimization*, 5(1-2):1–245, 2021.

[20] John GF Francis. The qr transformation a unitary analogue to the lr transformation—part 1. *The Computer Journal*, 4(3):265–271, 1961.

[21] Huan Gao, Yu-Hong Dai, and Xiao-Jiao Tong. Barzilai-borwein-like methods for the extreme eigenvalue problem. *Journal of Industrial & Management Optimization*, 11(3), 2015.

[22] Dan Garber and Elad Hazan. Fast and simple pca via convex optimization. *arXiv preprint arXiv:1509.05647*, 2015.

[23] Dan Garber, Elad Hazan, Chi Jin, Cameron Musco, Praneeth Netrapalli, Aaron Sidford, et al. Faster eigenvector computation via shift-and-invert preconditioning. In *International Conference on Machine Learning*, pages 2626–2634. PMLR, 2016.

[24] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015.

[25] Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242. PMLR, 2017.

[26] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *Advances in neural information processing systems*, 29, 2016.

[27] Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. *Advances in neural information processing systems*, 27, 2014.

[28] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[29] Prateek Jain, Chi Jin, Sham Kakade, and Praneeth Netrapalli. Global convergence of non-convex gradient descent for computing matrix squareroot. In *Artificial Intelligence and Statistics*, pages 479–488. PMLR, 2017.

[30] Xixi Jia, Hailin Wang, Jiangjun Peng, Xiangchu Feng, and Deyu Meng. Preconditioning matters: Fast global convergence of non-convex matrix factorization via scaled gradient descent. *Advances in Neural Information Processing Systems*, 36, 2024.

[31] Bo Jiang, Chunfeng Cui, and Yu-Hong Dai. Unconstrained optimization models for computing several extreme eigenpairs of real symmetric matrices. *Pacific Journal of Optimzation*, 10(1):55–71, 2014.

[32] Liwei Jiang, Yudong Chen, and Lijun Ding. Algorithmic regularization in model-free overparametrized asymmetric matrix factorization. *SIAM Journal on Mathematics of Data Science*, 5(3):723–744, 2023.

[33] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International conference on machine learning*, pages 1724–1732. PMLR, 2017.

[34] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. 1950.

[35] Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent converges to minimizers. *arXiv preprint arXiv:1602.04915*, 2016.

[36] Bingcong Li, Liang Zhang, Aryan Mokhtari, and Niao He. On the crucial role of initialization for matrix factorization. *arXiv preprint arXiv:2410.18965*, 2024.

[37] RV Mises and Hilda Pollaczek-Geiringer. Praktische verfahren der gleichungsauflösung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 9(1):58–77, 1929.

[38] Marcel Mongeau and M Torki. Computing eigenelements of real symmetric matrices via optimization. *Computational Optimization and Applications*, 29:263–287, 2004.

[39] Cameron Musco and Christopher Musco. Randomized block krylov methods for stronger and faster approximate singular value decomposition. *Advances in neural information processing systems*, 28, 2015.

[40] Arkadi S Nemirovskiy and Boris T Polyak. Iterative methods for solving linear ill-posed problems under precise information. *Eng. Cyber.*, (4):50–56, 1984.

[41] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[42] Erkki Oja and Juha Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of mathematical analysis and applications*, 106(1):69–84, 1985.

[43] Ohad Shamir. A stochastic pca and svd algorithm with an exponential convergence rate. In *International conference on machine learning*, pages 144–152. PMLR, 2015.

[44] Zhanwen Shi, Guanyu Yang, and Yunhai Xiao. A limited memory bfgs algorithm for non-convex minimization with applications in matrix largest eigenvalue problem. *Mathematical Methods of Operations Research*, 83:243–264, 2016.

[45] Gilbert W Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, 1993.

[46] Dominik Stöger and Mahdi Soltanolkotabi. Small random initialization is akin to spectral learning: Optimization and generalization guarantees for overparameterized low-rank matrix reconstruction. *Advances in Neural Information Processing Systems*, 34:23831–23843, 2021.

[47] Łukasz Struski, Paweł Morkisz, Przemysław Spurek, Samuel Rodriguez Bernabeu, and Tomasz Trzciński. Efficient gpu implementation of randomized svd and its applications. *Expert Systems with Applications*, 248:123462, 2024.

[48] Tian Tong, Cong Ma, and Yuejie Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *Journal of Machine Learning Research*, 22(150):1–63, 2021.

[49] Peng Xu, Bryan He, Christopher De Sa, Ioannis Mitliagkas, and Chris Re. Accelerated stochastic power iteration. In *International Conference on Artificial Intelligence and Statistics*, pages 58–67. PMLR, 2018.

[50] Xingyu Xu, Yandi Shen, Yuejie Chi, and Cong Ma. The power of preconditioning in overparameterized low-rank matrix sensing. In *International Conference on Machine Learning*, pages 38611–38654. PMLR, 2023.

[51] Gavin Zhang, Salar Fattahi, and Richard Y Zhang. Preconditioned gradient descent for overparameterized nonconvex burer–monteiro factorization with global optimality certification. *Journal of Machine Learning Research*, 24(163):1–55, 2023.

[52] Hongyi Zhang, Sashank J Reddi, and Suvrit Sra. Riemannian svrg: Fast stochastic optimization on riemannian manifolds. *Advances in Neural Information Processing Systems*, 29, 2016.

A Related Work

Our work is primarily concerned with k-SVD computation which is of fundamental importance and has a very long history. Our results broadly relate to three lines of research reviewed below.

Methods from numerical linear algebra. Computation of SVD is typically done via iterative methods that rely on one or more key algorithmic building blocks that includes power method [37], Lanczos' iterations [34], and the QR decomposition [20]. As is, these algorithmic building blocks are not always numerically stable. This has led to a significant body of work to identify stable, efficient algorithms. The stable, efficient variant of algorithms based on these building blocks typically transforms the matrix of interest to a bidiagonal matrix and then uses a variant of the QR algorithm, see [17] and [15] for example, for an extensive literature overview. Such an algorithm, in some form or other, tries to identify singular vectors and values iteratively (either individually or in a block manner). The number of iterations taken by such an algorithm typically depends on the *gap* between singular values and the accuracy desired. In recent years, there have been *gap* independent (but still accuracy dependent) guarantees established when the desired accuracy is far above the *gap* [39, 1]. It is an understatement that such an algorithm is a workhorse of modern scientific computation and, more specifically, machine learning, cf. [2, 22, 43, 23]. The power method, despite not being optimized, is part of this workhorse due to its simplicity [27]. In this work, our objective is not necessarily to develop a method that is *better* than that known in literature. Instead, we seek to develop a fundamental understanding of the performance of a gradient based approach for the k-SVD problem. Subsequently, it will help understand the implications of acceleration methods on performance improvement. In the process, compare it with the rich prior literature and understand relative strengths and limitations.

Gradient descent and nonconvex optimization. The convergence of gradient descent for nonconvex optimization has been studied extensively recently due to interest in empirical risk minimization in the context of large neural network or deep learning [6, 52, 24, 35, 26, 25, 33, 18]. In [24], authors established that the stochastic variant of gradient descent, the stochastic gradient descent (SGD), converges asymptotically to local minima and escapes strict saddle points for the tensor decomposition problem. In [35], it was further established that even the vanilla gradient descent converges to local minima provided that all saddle points are strict and the step-size is relatively small. An efficient procedure for saddle point escaping were also proposed in [33]. For eigenvalue problem, matrix completion, robust PCA, and matrix sensing tasks, [3, 4, 5, 26, 25] have highlighted that optimization landscapes have no spurious local minima, i.e., the objectives possesses only strict saddle points and all their local minima are also global. In [7, 52], authors established convergence of Riemannian or geometric gradient descent method.

While all these works are quite relevant, they primarily provide asymptotic results, require the landscape of objective to satisfy strong conditions, or utilize complicated step-size selection rules where often knowledge of problem specific quantities is needed which are not obvious how to compute apriori. As we shall see, this work overcomes such limitations (see Theorem 2.1).

Matrix factorization as nonconvex optimization. Matrix factorization can be viewed as a minimization of a nonconvex objective. Specifically, given a symmetric matrix M, to obtain rank k approximation, the objective to minimize is $\|M - XX^\top\|_F^2$ for $X \in \mathbb{R}^{n\times k}$. The parameterization $XX^\top$ has been often referred to as the Burrer-Monteiro matrix factorization [8]. Recently, the study of gradient descent for solving this minimization problem has received a lot of interest, see [9, 16, 29, 14, 11, 48, 46, 51, 50, 32, 31, 30, 36] and references therein. Specifically, progress has been made in three different regimes, namely, *over-parameterized setting* when $k > d$, *exact-parameterized setting* when $k = d$, and *under-parameterized setting* when $k < d$. Linear convergence rates for gradient descent were initially established only locally for all regimes (see [18, 14]. In [46], it was shown that small random initialization is enough to ensure global linear convergence provided gradient descent uses a fixed but small enough step-size, which is problem instance dependent. Originally, this was restricted to the regime of $k \geq d$, and subsequently was extended to the regime of $k < d$ in [31]. In [48], authors proposed using gradient descent with preconditioning, $X_{t+1} \leftarrow X_t - \eta\nabla g(X_t; M)(X_t^\top X_t)^{-1}$, with $\eta > 0$ that is constant. They established linear convergence for $k = d$ with spectral initialization. But then it requires already knowing the SVD of the matrix! In [51] extended this result to $k > d$. In [30], authors established that preconditioning with random initialization is in fact sufficient to ensure global linear convergence, when $k = d$. In

[36], authors extend these results to $k > d$ and showed that even quadratic convergence rates can be achieved using the so-called Nyström initialization.

Table 2: Here, we contextualize and compare our contributions to prior work. Specifically, on computing k-SVD using gradient-descent for non-convex matrix factorization (i.e., minimizing $\|M - XX^\top\|_F^2$ over $X \in \mathbb{R}^{n\times k}$).

	Parametrization Regime	Linear Convergence	Step-size Selection		Initialization	(Local) Acceleration
			Parameter-free	Pre-conditioning		
Stoger et al. [46]	$(k > d)$	✓	–	–	`small-random-init`	–
Jia et al. [32]		✓	–	–	`small-random-init`	–
Zhang et al. [51]		✓	–	–	`spectral-init`	–
Li et al. [36]		✓	–	✓	`random-init`	–
Zhang et al. [51]	$(k = d)$	✓	–	–	`spectral-init`	–
Tong et al. [48]		✓	–	✓	`spectral-init`	–
Stoger et al. [46]		✓	–	–	`small-random-init`	–
Jia et al. [32]		✓	–	–	`small-random-init`	–
Jia et al. [30]		✓	✓	✓	`random-init`	–
Li et al. [36]		✓	–	✓	`random-init`	–
Chi et al. [14]	$(k < d)$	✓	–	–	`spectral-init`	–
Jiang et al. [32]		✓	–	–	`small-random-init`	–
Li et al. [36]		–	–	✓	`random-init`	–
This work	(any k)	✓	✓	✓	`random-init`	✓

Despite all this progress, none of the aforementioned works consider the *under-parameterized regimes*, except [14, 31, 36]. Indeed, [14] provided local convergence results for gradient descent with fixed step-size when $k = 1$, see [14, Theorem 1]. In [31], global linear convergence was established but required small random initialization and fixed step-size when $k \leq d$. In [36], authors considered gradient descent with preconditioning when $k < d$. However, they only showed sub-linear convergence, requiring $O((1/\epsilon)\log(1/\epsilon))$ to find an ϵ-optimal solution, see [36, Theorem 2]. In contrast to the prior works, this work establishes global linear convergence, that is gradient descent requires $O(\log(1/\epsilon))$ iterations to find an ϵ-optimal solution, see Theorem 2.1 and Theorem 2.2.

Why study the under-parameterized setting? While the *under parameterized regime*, especially the case $k = 1$ is of fundamental importance as highlighted in [14], note that for $k > 1$, even if one finds an exact solution $X_\star$ that minimizes the objective $\|M - XX^\top\|_F^2$, then for any $k \times k$, orthogonal matrix Q, $X_\star Q$ also minimizes $\|M - XX^\top\|_F^2$. This rotation problem poses a challenge in using the objective $\|M - XX^\top\|_F^2$ for k-SVD while the objective $\|M - xx^\top\|_F^2$ doesn't. More importantly, being able to solve the *under parameterized regime* with $k = 1$, allows us to perform k-SVD for any $k \geq 1$.

B Building Intuition with rank 1 Matrix: Gradient descent is Heron's method

A key insight from our analysis is the observation that gradient descent with adaptive step-size (3) behaves like Heron's method. This is in stark contrast, with the observation of [46] suggesting that gradient descent for low-rank matrix factorization with a fixed step-size and small random initialization behaves like a power-method at an initial phase. To clarify our observation, we present here a convergence analysis in the simple yet instructive case of M being exactly of rank 1.

First, let us note that gradient of the function g at any given point $x \in \mathbb{R}^n$ is given by:

$$\nabla g(x; M) = -Mx + \|x\|^2 x. \tag{9}$$

When M is exactly of rank 1, i.e., $M = \sigma_1 u_1 u_1^\top$, the gradient updates (3) become: for all $t \geq 1$,

$$x_{t+1} = \frac{1}{2}\left(x_t + \frac{\sigma_1 u_1^\top x_t}{\|x_t\|^2} u_1\right) \tag{10}$$

To further simplify things, let us consider that the initial point x_1 is randomly selected as follows:

$$x_1 = Mx \qquad \text{and} \qquad x \sim \mathcal{N}(0, I_n). \tag{11}$$

Thus, we see that $x_1 = \sigma_1(u_1^\top x)u_1$. This, together with the iterations (10), clearly shows that for all $t \geq 1$, $x_t = \|x_t\|u_1$. Hence, for all $t \geq 1$, we have:

$$\|x_{t+1}\|u_1 = \frac{1}{2}\left(\|x_t\| + \frac{\sigma_1}{\|x_t\|}\right) u_1. \tag{12}$$

We see then that $\|x_t\|$ is evolving precisely according to Heron's method (a.k.a. the Babylonian method) for finding the square root of σ_1. Below, we present the convergence rate of the method in this case:

Proposition B.1. *When $M = \sigma_1 u_1 u_1^\top$. Gradient descent as described in (3) with an initial random point as in (11) is guaranteed to converge almost surely, i.e., $\|x_t \pm \sqrt{\sigma_1}u_1\| \longrightarrow 0$ a.s. as $t \to \infty$. More precisely, denoting $\epsilon_t = (\|x_t\|/\sqrt{\sigma_1}) - 1$, we have for all $t \geq 2$, $0 < \epsilon_{t+1} \leq (\epsilon_t^2 \wedge \epsilon_t)/2$*

Proposition of B.1 is immediate and corresponds exactly to the convergence analysis of Heron's method. We provide a proof in Appendix D for completeness. The established convergence guarantee indicates that gradient descent converges at a quadratic rate, meaning, that in order to attain an error accuracy of order ε, the method only requires $O(\log\log(1/\varepsilon))$ iterations.

It is worth mentioning that when the rank of M is exactly 1, then the objective g corresponds to that of an *exact-parameterization* regime. The random initialization scheme (11) we consider is only meant for ease of exposition and has been studied in the concurrent work of [36]. Like us, they also obtain quadratic convergence in this *exact-parameterization* regime. Indeed, if one uses an alternative random initialization, say $x_1 \sim \mathcal{N}(0, I_n)$, then one only obtains a linear convergence rate.

In general, we do not expect the matrix M to be exactly of rank 1. Extending the analysis to the generic setting is more challenging and is exactly the subject of §C.

C Convergence analysis for General Matrix: Establishing Theorem 2.1

In this section, we present the key ingredients for proving Theorem 2.1. The proof strategy is broken into three intermediate steps: *(i)* establishing that gradient descent has a natural region of attraction; *(ii)* showing that once within this region of attraction the iterates $(x_t)_{t\geq 0}$ align with u_1 at a linear rate; *(iii)* showing that the sequence $(\|x_t\|)_{t\geq 0}$ is evolving according to an approximate Heron's method, and is convergent to $\sqrt{\sigma_1}$ at a linear rate.

Without loss of generality, we consider that x_1 is randomly initialized as in (11). This choice of initialization allows us to simplify the analysis as it ensures that x_1 is in the image of M. Most importantly, our analysis only requires that $\mathbb{P}(x_1^\top u_1 \neq 0) = 1$. It will be also convenient to introduce, for all $t \geq 0$, $i \in [d]$, the angle $\theta_{i,t}$ between u_i and x_t, defined through

$$\cos(\theta_{i,t}) = \frac{x_t^\top u_i}{\|x_t\|} \tag{13}$$

whenever $\|x_t\| > 0$.

Step 1. Region of attraction. One hopes that the iterates x_t do not escape to infinity or vanish at 0. It turns out that this is indeed the case and this is what we present in Lemma C.1. We define:

$$a = 2\sqrt{\eta(1-\eta)}\left(|\cos(\theta_{1,0})| \vee \frac{1}{\sqrt{\kappa}}\right), \tag{14}$$

$$b = (1-\eta) + \frac{1}{2}\sqrt{\frac{\eta}{1-\eta}}\left(\frac{1}{|\cos(\theta_{1,0})|} \wedge \sqrt{\kappa}\right), \tag{15}$$

where it is not difficult to verify that $a < 1$ and $b > 1$.

Lemma C.1 (Attracting region of gradient descent)**.** *Gradient descent as described in (3) with the random initialization (11) ensures that: (i) $\forall t > 1, a\sqrt{\sigma_1} \leq \|x_t\|$; (ii) $\forall t > \tau, \|x_t\| \leq b\sqrt{\sigma_1}$ where τ is given by:*

$$\tau = \frac{\eta(b^2-1)}{(1-\eta)b^2+\eta} \log\left(\frac{\|x_1\|}{b\sqrt{\sigma_1}}\right). \tag{16}$$

Moreover, the sequence $(|\cos(\theta_{1,t})|)_{t\geq 0}$ is non-decreasing.

The proof of Lemma C.1 is given in Appendix D.3. Interestingly, the constants a and b do not arbitrarily degrade with the quality of the initial random point. Thus, the number of iterations required to enter the region $[a\sqrt{\sigma_1}, b\sqrt{\sigma_1}]$ can be made constant if for instance one further constrains $\|x_1\| = 1$. Furthermore, the fact that $|\cos(\theta_{1,t})|$ is non-decreasing is remarkable because it means that $x_t/\|x_t\|$ can only get closer to $\{-u_1, u_1\}$. To see that, note that we have $\|(x_t/\|x_t\|) \pm u_1\|^2 = 2 \pm 2\cos(\theta_{1,t}) \geq 2(1 - |\cos(\theta_{1,t})|)$. Indeed, a consequence of Lemma C.1, is the following saddle-point avoidance result.

Lemma C.2 (Saddle-point avoidance)**.** *Let $\delta > 0$. Assume that $\sigma_1 - \sigma_2 > 0$ and that $|\|x_t\| - \sqrt{\sigma_i}| < \delta$ for $i \neq 1$ and some $t > 1$. Then, gradient descent as described in (3) with the random initialization (11) ensures that*

$$\|\nabla g(x_t; M)\| \geq a\sigma_1 |\cos(\theta_{1,0})|\, |\,|\sqrt{\sigma_1} - \sqrt{\sigma_i}| - \delta| \tag{17}$$

We provide a proof of Lemma C.2 in Appendix D.4. We remark that if $\delta \ll \sigma_1 - \sigma_i$, then the gradient cannot vanish. The lower bound depends on the initial point x_0 through $|\cos(\theta_{1,0})|$ which can be very small. In general with the random initialization (11), small values of $|\cos(\theta_{1,0})|$ are improbable.

Step 2. Alignment at linear rate. Another key ingredient in the analysis is Lemma C.3 which we present below:

Lemma C.3. *Assume that $\sigma_1 - \sigma_2 > 0$ and let τ be defined as in Lemma C.1. Gradient descent as described in (3) with random initialization (11) ensures that: $t > \tau$, $i \neq 1$,*

$$\left\|\frac{x_t}{\|x_t\|} - u_1\right\| \wedge \left\|\frac{x_t}{\|x_t\|} + u_1\right\| \leq \left(1 - \frac{\eta(\sigma_1-\sigma_2)}{((1-\eta)b^2+1)\sigma_1}\right)^{t-\tau} \sqrt{2}\,|\tan(\theta_{1,0})|,$$

$$|\cos(\theta_{i,t})| \leq \left(1 - \frac{\eta(\sigma_1-\sigma_i)}{((1-\eta)b^2+1)\sigma_1}\right)^{t-\tau} \left|\frac{\cos(\theta_{i,0})}{\cos(\theta_{1,0})}\right|.$$

The proof of Lemma C.3 is given in Appendix D.5. The result shows that gradient descent is implicitly biased towards aligning its iterates with u_1. This alignment happens at a linear rate with a factor that depends on $\sigma_1 - \sigma_2$, and this is why we require the condition $\sigma_1 - \sigma_2 > 0$. If $\sigma_1 = \sigma_2 > \sigma_3$, then our proof can be easily adjusted to show that x_t will align with the singular subspace spanned by u_1, u_2. Thus, our assumptions are without loss of generality.

Step 3. Convergence with approximate Heron's iterations. The final step of our analysis is to show that $\left| \|x_t\|/\sqrt{\sigma_1} - 1 \right|$ vanishes at a linear rate which is presented in Lemma C.4. To establish this result, our proof strategy builds on the insight that the behavior of the iterates $(\|x_t\|)_{t \geq 0}$ resemble those of an approximate Heron's method. The result is only shown when $\eta = 1/2$ as this gives the cleanest proof.

Lemma C.4. *Assume that $\sigma_1 - \sigma_2 > 0$ and let τ be defined as in Lemma C.1. Gradient descent as described in (3) with $\eta = 1/2$ and random initialization (11) ensures that: for all $t > \tau$,*

$$\left| \frac{\|x_{t+1}\|}{\sqrt{\sigma_1}} - 1 \right| \lesssim C(t-\tau) \left(\left(1 - \frac{\sigma_1 - \sigma_2}{(1+b^2)\sigma_1} \right) \vee \frac{1}{\sqrt{2}} \right)^{2(t-\tau)} \tag{18}$$

where $C = \left(\frac{|\tan(\theta_{1,0})|^2}{4a|\cos(\theta_{1,0})|^2} \left(b + \frac{1}{a} \right)^2 + \frac{|\tan(\theta_{1,0})|}{|\cos(\theta_{1,0})|} \left(b + \frac{1}{a} \right) \right)$.

Proof sketch of Lemma C.4. By taking the scalar product of the two sides of the gradient update equation (3) with u_1, we deduces that

$$\|x_{t+1}\| = \frac{1}{2} \left(\|x_t\| + \frac{\sigma_1}{\|x_t\|} \right) \frac{|\cos(\theta_{1,t})|}{|\cos(\theta_{1,t+1})|}. \tag{19}$$

Next, we can leverage Lemma C.3 to show that the ratio $|\cos(\theta_{1,t})|/|\cos(\theta_{1,t+1})|$ converges to 1 at a linear rate. We then recognize the familiar form of Heron's iterations 12. Starting with this form, we can show convergence of $\|x_t\|$ to $\sqrt{\sigma_1}$. We spare the reader the tedious details of this part and refer them to the complete proof given in Appendix D.6. □

Step 4. Putting everything together. Below, we present a result which is an immediate consequence of Lemma C.1, Lemma C.3, and C.4.

Theorem C.5. *Assume that $\sigma_1 - \sigma_2 > 0$. Gradient descent as described in (3) with $\eta = 1/2$ and random initialization (11) ensures that: for all $t > \tau$,*

$$\|x_{t+\tau} + \sqrt{\sigma_1} u_1\| \wedge \|x_{t+\tau} - \sqrt{\sigma_1} u_1\| \leq c_1 \sqrt{\sigma_1} \left(\left(1 - \frac{(\sigma_1 - \sigma_2)}{(b^2+1)\sigma_1} \right) \vee \frac{1}{\sqrt{2}} \right)^t, \tag{20}$$

so long as $\tau \geq c_2 \left(\frac{\sigma_1 + \sigma_2}{\sigma_1 - \sigma_2} \vee 1 \right) \log \left(\frac{c_3}{(\sigma_1 - \sigma_2)} \right)$ with positive constants c_1, c_2, c_3 that depend only on x_0; with random initialization c_1, c_2, c_3 are strictly positive almost surely.

The proof is given in Appendix D.2. Theorem 2.1 is an immediate consequence of Theorem C.5.

D Global Convergence of Gradient Descent

In this section, we provide the detailed proofs of Theorem B.1, Theorem 2.1, and Theorem C.5.

The proof of Theorem B.1 is self-contained and is provided in §D.1 and serves the purpose of building intuition as of why gradient descent behaves like Heron's method. Theorem 2.1 is an immediate consequence of Theorem C.5, and both their proofs are given in §D.2.

The proof of Theorem C.5 builds on the insight that gradient descent behaves like Heron's method which is captured in the proof of Lemma C.4 given in §D.5. But before that, we need first to ensure that gradient descent has a region of attraction and this made precise in Lemma C.1 and its proof is given in §D.3. And secondly, we require the iterates of the gradient descent method to align with the leading singular vector which is established in Lemma C.3 with its proof given in §D.5.

D.1 Proof of Proposition B.1

Proof of Proposition B.1. First, we immediately see that the event $\{\|x_1\| \neq 0, \quad \text{and} \quad x_1 = \|x_1\|u_1\}$ holds almost surely.

We start with the observation that for all $t \geq 1$, $|x_t^\top u_1| = \|x_t\|$. This is because, by construction, the initial point x_1 is already in the span of M. Next, we have that for all $t \geq 1$,

$$x_{t+1}^\top u_1 = \left((1-\eta) + \eta\frac{\sigma_1}{\|x_t\|^2}\right) x_t^\top u_1$$

which leads to

$$\|x_{t+1}\| = (1-\eta)\|x_t\| + \eta\frac{\sigma_1}{\|x_t\|}.$$

In the above, we recognize the iterations of a Babylonian method (a.k.a. Heron's method) for finding the square root of a real number. For completeness, we provide the proof of convergence. Let us denote

$$\epsilon_t = \frac{\|x_t\|}{\sigma_1} - 1$$

by replacing in the equation above we obtain that

$$\begin{aligned}\epsilon_{t+1} &= (1-\eta)(\epsilon_t + 1) + \frac{\eta}{(\epsilon_t+1)} - 1 \\ &= \frac{(1-\eta)\varepsilon_t^2 + (1-2\eta)\varepsilon_t}{\varepsilon_t + 1}.\end{aligned}$$

With the choice $\eta = 1/2$, we obtain that

$$\epsilon_{t+1} = \frac{\epsilon_t^2}{2(\epsilon_t+1)}$$

from which we first conclude that for all $t \geq 1$, $\varepsilon_{t+1} > 0$ (note that we already have $\varepsilon_t > -1$). Thus we obtain

$$0 < \epsilon_{t+1} \leq \frac{1}{2}\min\left(\epsilon_t^2, \epsilon_t\right).$$

This clearly implies that we have quadratic convergence. □

D.2 Proof of Theorem 2.1 and Theorem C.5

Theorem C.5 is an intermediate step in proving Theorem 2.1 and is therefore included in the proof below.

Proof of Theorem 2.1. First, by applying Lemma C.1 (with $\eta = 1/2$), that after $\tau = \log\left(\frac{\|x_1\|}{b\sqrt{\sigma_1}}\right)$ that for all $t > \tau$, we have

$$a\sqrt{\sigma_1} \leq \|x_t\| \leq b\sqrt{\sigma_1}.$$

and we also have

$$\left|\frac{\|x_t\|^2}{\sigma_1}-1\right| \le \left|\frac{\|x_t\|}{\sqrt{\sigma_1}}-1\right|\left|\frac{\|x_t\|}{\sqrt{\sigma_1}}+1\right| \le (b+1)\left|\frac{\|x_t\|}{\sqrt{\sigma_1}}-1\right| \tag{21}$$

Thus, using the decompositions G.11 and applying Lemma C.3 and C.4, we obtain

$$\|x_{t+\tau} \pm \sqrt{\sigma_1}u_1\| \le C_1\sqrt{\sigma_1}t\left(\left(1+\frac{\sigma_1-\sigma_2}{b^2\sigma_1+\sigma_2}\right)\vee\sqrt{2}\right)^{-2t} + C_2\sqrt{\sigma_1}\left(1+\frac{(\sigma_1-\sigma_2)}{b^2\sigma_1+\sigma_2}\right)^{-t}$$

where we denote

$$C_1 = (b+1)\left(\frac{|\tan(\theta_{1,1})|^2}{4a|\cos(\theta_{1,1})|^2}\left(b+\frac{1}{a}\right)^2 + \frac{|\tan(\theta_{1,1})|}{|\cos(\theta_{1,1})|}\left(b+\frac{1}{a}\right)\right)$$
$$C_2 = \sqrt{2}|\tan(\theta_{1,1})|^2.$$

We can verify that

$$t \ge 2\left(\left(\frac{b^2\sigma_1+\sigma_2}{\sigma_1-\sigma_2}+1\right)\vee\left(\frac{\sqrt{2}}{\sqrt{2}-1}\right)\right)\log\left(2\left(\frac{b^2\sigma_1+\sigma_2}{\sigma_1-\sigma_2}+1\right)\vee\left(\frac{\sqrt{2}}{\sqrt{2}-1}\right)\right) \tag{22}$$

$$\implies t \ge \left(\frac{b^2\sigma_1+\sigma_2}{\sigma_1-\sigma_2}+1\right)\vee\left(\frac{\sqrt{2}}{\sqrt{2}-1}\right)\log(t) \tag{23}$$

$$\implies t\left(\left(1+\frac{\sigma_1-\sigma_2}{b^2\sigma_1+\sigma_2}\right)\vee\sqrt{2}\right)^{-t} \le 1 \tag{24}$$

where we use the elementary fact that if $t \ge 2a\log(2a)$ then $t \ge a\log(t)$ for any $a > 0$ and $\log(1+x) \ge \frac{x}{1+x}$ for all $x > 0$. Thus, under condition (22) we obtain

$$\|x_{t+\tau} \pm \sqrt{\sigma_1}u_1\| \le C_1\sqrt{\sigma_1}\left(\left(1+\frac{\sigma_1-\sigma_2}{b^2\sigma_1+\sigma_2}\right)\vee\sqrt{2}\right)^{-t} + C_2\sqrt{\sigma_1}\left(1+\frac{(\sigma_1-\sigma_2)}{b^2\sigma_1+\sigma_2}\right)^{-t},$$

In view of Lemma G.11, the statement concerning $\|x_t x_t^\top - \sigma_1 u_1 u_1^\top\|$ follows similarly. At this stage we have shown the statement of Theorem C.5.

We also see that

$$t \ge \left(\left(\frac{b^2\sigma_1+\sigma_2}{\sigma_1-\sigma_2}+1\right)\vee\left(\frac{\sqrt{2}}{\sqrt{2}-1}\right)\right)\log\left(\frac{2}{C_1\sqrt{\sigma_1}\epsilon}\right)$$

$$\implies C_1\sqrt{\sigma_1}\left(\left(1+\frac{\sigma_1-\sigma_2}{b^2\sigma_1+\sigma_2}\right)\vee\sqrt{2}\right)^{-t} \le \frac{\epsilon}{2} \tag{25}$$

$$t \ge \left(\frac{b^2\sigma_1+\sigma_2}{\sigma_1-\sigma_2}+1\right)\log\left(\frac{2}{C_2\sqrt{\sigma_1}\epsilon}\right)$$

$$\implies C_1\sqrt{\sigma_1}\left(1+\frac{\sigma_1-\sigma_2}{b^2\sigma_1+\sigma_2}\right)^{-t} \le \frac{\epsilon}{2} \tag{26}$$

where we used again the elementary fact $\log(1+x) \ge \frac{x}{1+x}$. We conclude that if

$$t \ge c_1\left(\frac{b^2\sigma_1+\sigma_2}{\sigma_1-\sigma_2}\vee 1\right)\log\left(\frac{c_2}{\sigma_1\epsilon}\left(\frac{b^2\sigma_1+\sigma_2}{\sigma_1-\sigma_2}\vee 1\right)\right) \qquad \text{and} \qquad \tau = \log\left(\frac{\|x_t\|}{b\sqrt{\sigma_1}}\right)$$

then

$$\|x_t \pm \sqrt{\sigma_1}u_1\| \le \epsilon.$$

□

D.3 Proof of Lemma C.1

Proof of Lemma C.1. Our proof supposes that $x_1^\top u_1 \neq 0$ which guaranteed almost surely by the random initialization (11).

Claim 1. First, we establish the following useful inequalities: for all $t \geq 1$,

(i) $\|x_t\| > 0$.

(ii) $\left((1-\eta) + \eta\frac{\sigma_d}{\|x_t\|^2}\right)\|x_t\| \leq \|x_{t+1}\| \leq \left((1-\eta) + \eta\frac{\sigma_1}{\|x_t\|^2}\right)\|x_t\|$

We start by noting that we can project the iterations of the gradient descent updates (3) onto u_i, for all $i \in [d]$, gives

$$u_i^\top x_{t+1} = \left((1-\eta) + \eta\frac{\sigma_i}{\|x_t\|^2}\right) u_i^\top x_t \tag{27}$$

which follows because of the simple fact that $u_i^\top M = \sigma_i u_i^\top$. Thus, squaring and summing the equations (27) over $i \in [d]$, gives

$$\|x_{t+1}\|^2 = \sum_{i=1}^{d} \left((1-\eta) + \eta\frac{\sigma_i}{\|x_t\|^2}\right)^2 |u_i^\top x_t|^2 \tag{28}$$

Recalling that $\sigma_1 \geq \cdots \geq \sigma_d > 0$, we immediately deduce from (28) the inequality (ii). Inequality (i) because if $\|x_1\| > 0$, than thanks to (ii) it also follows $\|x_t\| > 0$.

Claim 2. Let us denote for all $i \in [d]$, $t \geq 1$, $|\cos(\theta_{i,t})| = |u_i^\top x_t|/\|x_t|$. We establish that the following properties hold:

(i) $(|\cos(\theta_{1,t})|)_{t\geq 1}$ is non-decreasing sequence

(ii) $(|\cos(\theta_{d,t})|)_{t\geq 1}$ is a non-increasing sequence

(iii) for all $t \geq 1$, $\|x_{t+1}\| \geq a\sqrt{\sigma_1}$.

To prove the aforementioned claim we start by noting from (27) that for all $i \in [d]$, we have

$$\|x_{t+1}\||\cos(\theta_{i,t+1})| = \left((1-\eta)\|x_t\| + \eta\frac{\sigma_i}{\|x_t\|}\right)|\cos(\theta_{i,t})| \tag{29}$$

Thus, from the Claim 1 (Eq. (ii)), we immediately see that for all $t \geq 1$, we have

$$1 \geq |\cos(\theta_{1,t+1})| \geq |\cos(\theta_{1,t})| \geq |\cos(\theta_{1,1})| \tag{30}$$

$$|\cos(\theta_{d,t+1})| \leq |\cos(\theta_{d,t})| \leq 1, \tag{31}$$

Now, we see that the l.h.s. inequality of Eq. (ii) in Claim 1, the combination of (30) and (29) leads to the following:

$$\|x_{t+1}\| \geq \max\left\{\left((1-\eta)\|x_t\| + \eta\frac{\sigma_1}{\|x_t\|}\right)|\cos(\theta_{1,1})|, \left((1-\eta)\|x_t\| + \eta\frac{\sigma_d}{\|x_t\|}\right)\right\} \tag{32}$$

$$\overset{(a)}{\geq} 2\sqrt{\eta(1-\eta)}\max\left\{|\cos(\theta_{1,1})|, \sqrt{\frac{\sigma_d}{\sigma_1}}\right\}\sqrt{\sigma_1} \tag{33}$$

$$= a\sqrt{\sigma_1} \tag{34}$$

where we used in (a) the elementary fact that $2\sqrt{\eta(1-\eta)\sigma} = \inf_{x>0}(1-\eta)x + \eta(\sigma/x)$ for all $\sigma > 0$. This concludes Claim 2.

Claim 3. We establish that if $a\sqrt{\sigma_1} \leq \|x_t\| \leq \sqrt{\sigma_1}$ then, $\|x_{t+1}\| \leq b\sqrt{\sigma_1}$.

We can immediately see from Claim 1 that

$$\|x_{t+1}\| \leq (1-\eta)\|x_t\| + \eta\frac{\sigma_1}{\|x_t\|} \tag{35}$$

$$\leq \left((1-\eta) + \frac{\eta}{2\sqrt{\eta(1-\eta)}\max(|\cos(\theta_{1,1})|, \kappa^{-1/2})}\right)\sqrt{\sigma_1} \tag{36}$$

$$\leq \left((1-\eta) + \frac{1}{2}\sqrt{\frac{\eta}{1-\eta}}\min(|\cos(\theta_{1,1})|^{-1}, \kappa^{1/2})\right)\sqrt{\sigma_1} \tag{37}$$

$$= b\sqrt{\sigma_1} \tag{38}$$

where we denote $b = (1-\eta) + \frac{1}{2}\sqrt{\frac{\eta}{1-\eta}}\min(|\cos(\theta_{1,1})|^{-1}, \sqrt{\kappa})$ and note that $b > 1$.

Claim 4: If $\sqrt{\sigma_1} < \|x_t\| \leq b\sqrt{\sigma_1}$, then $\|x_{t+1}\| \leq b\sqrt{\sigma_1}$.

Indeed, start from the inequality in Claim 1, we can immediately verify that

$$\|x_{t+1}\| \leq \left((1-\eta) + \eta\frac{\sigma_1}{\|x_t\|^2}\right)\|x_t\| \leq \|x_t\| \leq b\sqrt{\sigma_1} \tag{39}$$

Indeed we observe that if $\sqrt{\sigma_1} < \|x_t\| \leq b\sqrt{\sigma_1}$, then $\sqrt{\sigma_1}/\|x_t\| \leq 1$.

Claim 5: If $\|x_1\| > b\sqrt{\sigma_1}$, then there exists $k^\star \geq 1$, such that $\|x_{1+k^\star}\| \leq b\sqrt{\sigma_1}$, satisfying

$$k^\star \leq \frac{b^2}{\eta(b^2-1)}\log\left(\frac{\|x_t\|}{b\sqrt{\sigma_1}}\right) \tag{40}$$

Assuming that $\|x_t\| > b\sqrt{\sigma_1}$, let us define $k^\star = \min\{k \geq 1 : \|x_{t+k}\| \leq b\sqrt{\sigma_1}\}$, the number of iterations required for $\|x_{t+k}\| \leq b\sqrt{\sigma_1}$. Using the r.h.s. of the inequality in Claim 1, and by definition of $k^\star$, we know that for $1 \leq k < k^\star$,

$$\|x_{t+k-1}\| \leq \left((1-\eta) + \eta\frac{\sigma_1}{\|x_{t+k}\|^2}\right)\|z_{t+k}\| \leq \left((1-\eta) + \eta b^{-2}\right)\|z_{t+k}\|. \tag{41}$$

Iterating the above inequality, we obtain that

$$\|x_{t+k^\star}\| \leq \left((1-\eta) + \eta b^{-2}\right)^{k^\star}\|x_t\|.$$

Now, let us note that $k^\star$ can not be too large. More specifically, we remark that:

$$k^\star \geq \frac{\eta(b^2-1)}{(1-\eta)b^2+\eta}\log\left(\frac{\|x_t\|}{b\sqrt{\sigma_1}}\right) \geq \frac{\log\left(\frac{\|x_t\|}{b\sigma_1}\right)}{\log\left(\frac{1}{(1-\eta)+\eta b^{-2}}\right)} \implies \left((1-\eta) + \eta b^{-2}\right)^{k^\star}\|x_t\| \leq b\sqrt{\sigma_1}$$

Therefore, it must hold that

$$k^\star \leq \frac{\eta(b^2-1)}{(1-\eta)b^2+\eta}\log\left(\frac{\|x_t\|}{b\sqrt{\sigma_1}}\right)$$

Note that from Claim 3 and 4, as soon as $a\sqrt{\sigma_1} \leq \|x_t\| \leq b\sqrt{\sigma_1}$, then it will never escape this region. Therefore, we only need to use Claim 5 if the first iterate $\|x_1\|$ is outside the region $[a\sqrt{\sigma_1}, b\sqrt{\sigma_1}]$. $\square$

D.4 Proof of Lemma C.2

Proof of Lemma C.2. We prove the saddle-avoidance property of gradient descent. The proof is an immediate application of C.1. Indeed, we have

$$\begin{aligned}
\|\nabla g(x_t; M)\| &= \|Mx_t - \|x_t\|^2 x_t\| \\
&= \left\| \sum_{i=1}^{d} (\sigma_i - \|x_t\|^2)(u_i^\top x_t) u_i \right\| \\
&= \left(\sum_{i=1}^{d} (\sigma_i - \|x_t\|^2)^2 (u_i^\top x_t)^2 \right)^{1/2} \\
&\geq |\sigma_1 - \|x_t\|^2 | |u_1^\top x_t| \\
&\geq |\sqrt{\sigma_1} - \|x_t\|| |\sqrt{\sigma_1} + \|x_t\|| \|x_t\| |\cos(\theta_{1,t})| \\
&\geq ||\sqrt{\sigma_1} - \sqrt{\sigma_i}| - \delta | \sqrt{\sigma_1} a \sqrt{\sigma_1} |\cos(\theta_{1,1})| \\
&\leq a\sigma_1 |\cos(\theta_{1,1})| \, ||\sqrt{\sigma_1} - \sqrt{\sigma_i}| - \delta|
\end{aligned}$$

where we used Lemma C.1 to bound $\|x_t\| \geq a\sqrt{\sigma_1}$ and $|\cos(\theta_{i,t})| \geq |\cos(\theta_{i,1})|$, and reverse triangular inequality □

D.5 Proof of Lemma C.3

Proof of Lemma C.3. We start from the observation that for all $i \in [d]$, we have

$$\|x_{t+1}\| |\cos(\theta_{i,t+1})| = \left((1-\eta)\|x_t\| + \eta \frac{\sigma_i}{\|x_t\|} \right) |\cos(\theta_{i,t})|. \tag{42}$$

Now, note dividing the equations corresponding to 1 and i, we get

$$\frac{|\cos(\theta_{1,t+1})|}{|\cos(\theta_{i,t+1})|} = \frac{\left((1-\eta)\|x_t\| + \eta \frac{\sigma_1}{\|x_t\|} \right)}{\left((1-\eta)\|x_t\| + \eta \frac{\sigma_i}{\|x_t\|} \right)} \frac{|\cos(\theta_{1,t})|}{|\cos(\theta_{i,t})|} \tag{43}$$

$$= \left(1 + \frac{\left(\eta \frac{\sigma_1 - \sigma_i}{\|x_t\|} \right)}{\left((1-\eta)\|x_t\| + \eta \frac{\sigma_i}{\|x_t\|} \right)} \right) \frac{|\cos(\theta_{1,t})|}{|\cos(\theta_{i,t})|} \tag{44}$$

$$= \underbrace{\left(1 + \frac{\eta(\sigma_1 - \sigma_i)}{(1-\eta)\|x_t\|^2 + \eta\sigma_i} \right)}_{>1 \text{ because } \sigma_1 - \sigma_i > 0} \frac{|\cos(\theta_{1,t})|}{|\cos(\theta_{i,t})|} \tag{45}$$

Thus, we have for all $t \geq 1$:

$$\frac{|\cos(\theta_{i,t})|}{|\cos(\theta_{i,t+1})|} = \left(1 + \frac{\eta(\sigma_1 - \sigma_i)}{(1-\eta)\|x_t\|^2 + \eta\sigma_i} \right) \frac{|\cos(\theta_{1,t})|}{|\cos(\theta_{1,t+1})|},$$

which leads to

$$\frac{|\cos(\theta_{i,1})|}{|\cos(\theta_{i,t+1})|} = \left(\prod_{s=1}^{t} \left(1 + \frac{\eta(\sigma_1 - \sigma_i)}{(1-\eta)\|x_s\|^2 + \eta\sigma_i} \right) \right) \frac{|\cos(\theta_{1,1})|}{|\cos(\theta_{1,t+1})|}.$$

We recall that for all $t \geq t_0$,

$$|\cos(\theta_{1,t})| \leq |\cos(\theta_{1,t+1})| \leq 1 \quad \text{and} \quad a\sqrt{\sigma_1} \leq \|x_t\| \leq b\sqrt{\sigma_1}$$

which implies that

$$\begin{aligned}
&\frac{|\cos(\theta_{1,1})|}{|\cos(\theta_{1,t+1})|} \geq |\cos(\theta_{1,1})| \\
&\left(1 + \frac{\eta(\sigma_1 - \sigma_i)}{(1-\eta)\|x_t\|^2 + \eta\sigma_i} \right) \geq 1 + \frac{\eta(\sigma_1 - \sigma_i)}{(1-\eta)b^2\sigma_1 + \eta\sigma_i}
\end{aligned}$$

Hence,

$$|\cos(\theta_{i,t+1})| \leq \left(1 + \frac{\eta(\sigma_1 - \sigma_i)}{(1-\eta)b^2\sigma_1 + \eta\sigma_i}\right)^{-t} \frac{|\cos(\theta_{i,1})|}{|\cos(\theta_{1,1})|} \underset{t\to\infty}{\longrightarrow} 0.$$

Either $u_1^\top x_1 > 0$ or $u_1^\top x_1 < 0$. Without loss of generality, we will present the case when $u_1^\top x_1 > 0$, in which case $\cos(\theta_{1,1}) > 0$ and consequently $\cos(\theta_{1,t}) > 0$. Let us further note that

$$\begin{aligned}
\left\| \frac{1}{\|x_t\|} x_t - u_1 \right\|^2 &= 2(1 - \cos(\theta_{1,t})) \\
&= 2\frac{1 - \cos^2(\theta_{1,t})}{1 + \cos(\theta_{1,t})} \\
&\leq 2\sum_{i=2}^{d} \cos^2(\theta_{i,t}) \\
&\leq 2\sum_{i=2}^{d} \left(1 + \frac{\eta(\sigma_1 - \sigma_i)}{(1-\eta)b^2\sigma_1 + \eta\sigma_i}\right)^{-2t} \frac{|\cos(\theta_{i,1})|^2}{|\cos(\theta_{1,1})|^2} \\
&\leq 2\left(1 + \frac{\eta(\sigma_1 - \sigma_2)}{(1-\eta)b^2\sigma_1 + \eta\sigma_2}\right)^{-2t} \frac{1 - |\cos(\theta_{1,1})|^2}{|\cos(\theta_{1,1})|^2} \\
&\leq 2\left(1 + \frac{\eta(\sigma_1 - \sigma_2)}{(1-\eta)b^2\sigma_1 + \eta\sigma_2}\right)^{-2t} |\tan(\theta_{1,1})|^2
\end{aligned}$$

□

D.6 Proof of Lemma C.4

Proof of Lemma C.4. Let us denote for all $t \geq 1$

$$\rho_t = 1 - \frac{|\cos(\theta_{1,t})|}{|\cos(\theta_{1,t+1})|} \tag{46}$$

$$\epsilon_t = \frac{\|x_t\|}{\sqrt{\sigma_1}} - 1 \tag{47}$$

First, we verify that $\rho_t \underset{t\to\infty}{\longrightarrow} 0$. Indeed, we have

$$\begin{aligned}
\rho_t &= 1 - \frac{|\cos(\theta_{1,t})|}{|\cos(\theta_{1,t+1})|} \\
&\leq \frac{|\cos(\theta_{1,t+1}) - 1| + |1 - \cos(\theta_{1,t})|}{|\cos(\theta_{1,1})|} \\
&\leq \frac{1}{|\cos(\theta_{1,1})|} \max\left(\left\| \frac{1}{\|x_{t+1}\|} x_{t+1} - u_1 \right\|^2, \left\| \frac{1}{\|x_t\|} x_t - u_1 \right\|^2 \right) \\
&\leq \frac{2|\tan(\theta_{1,1})|}{|\cos(\theta_{1,1})|} \left(1 + \frac{\sigma_1 - \sigma_2}{b^2\sigma_1 + \sigma_2}\right)^{-2t}
\end{aligned}$$

where we used the fact that $(\cos(\theta_{1,t}))_{t\geq 1}$ is a non-decreasing sequence, and used Lemma C.3 to obtain the final bound. Thus, we see that $\rho_t \underset{t\to\infty}{\longrightarrow} 0$.

Next, we also recall by assumption that $a\sqrt{\sigma_1} \leq \|x_1\| \leq b\sqrt{\sigma_1}$, and by Lemma C.1 that $a\sqrt{\sigma_1} \leq \|x_t\| \leq b\sqrt{\sigma_1}$ for all $t \geq 1$. Thus, we have:

$$-1 < a - 1 \leq \epsilon_t \leq b - 1 \tag{48}$$

Now, let us show that $\epsilon_t \underset{t\to\infty}{\longrightarrow} 0$. We recall that

$$\|x_{t+1}\| |\cos(\theta_{1,t+1})| = \left((1-\eta)\|x_t\| + \eta\frac{\sigma_1}{\|x_t\|}\right) |\cos(\theta_{1,t})|. \tag{49}$$

Thus, we have

$$\epsilon_{t+1} = \left((1-\eta)(\epsilon_t+1) + \frac{\eta}{\epsilon_t+1}\right)(1-\rho_t) - 1 \tag{50}$$

$$= \left((1-\eta)(\epsilon_t+1) + \frac{\eta}{\epsilon_t+1}\right) - 1 - \rho_t\left((1-\eta)(\epsilon_t+1) + \frac{\eta}{\epsilon_t+1}\right) \tag{51}$$

$$= \frac{(1-\eta)\epsilon_t^2 + (1-2\eta)\epsilon_t}{(\epsilon_t+1)} - \rho_t\left((1-\eta)(\epsilon_t+1) + \frac{\eta}{\epsilon_t+1}\right) \tag{52}$$

$$= \frac{\epsilon_t^2}{2(\epsilon_t+1)} - \frac{\rho_t}{2}\left((\epsilon_t+1) + \frac{1}{\epsilon_t+1}\right) \tag{53}$$

where in the last equality we used $\eta = 1/2$. We conclude from the above that for all $t \geq 1$

$$\varepsilon_{t+1} \geq \max\left(-\frac{\rho_t}{2}\left(b+\frac{1}{a}\right), a-1\right) = -\min\left(\frac{\rho_t}{2}\left(b+\frac{1}{a}\right), 1-a\right)$$

$$\varepsilon_{t+1} \leq \frac{\epsilon_t^2}{2(\epsilon_t+1)} - \rho_t$$

Thus, for all $t \geq 2$, we have

$$|\varepsilon_{t+1}| \leq \frac{\epsilon_t^2}{2(\epsilon_t+1)}\mathbb{1}\{\epsilon_t > 0\} + \frac{\rho_t^2}{8a}\left(b+\frac{1}{a}\right)^2 + \frac{\rho_t}{2}\left(b+\frac{1}{a}\right)$$

$$\leq \frac{\min(|\epsilon_t|, |\epsilon_t|^2)}{2} + \frac{\rho_t^2}{8a}\left(b+\frac{1}{a}\right)^2 + \frac{\rho_t}{2}\left(b+\frac{1}{a}\right)$$

which further gives

$$|\varepsilon_{t+1}| \leq \sum_{k=0}^{t-2}\frac{1}{2^k}\left(\frac{\rho_{t-k}^2}{8a}\left(b+\frac{1}{a}\right)^2 + \frac{\rho_{t-k}}{2}\left(b+\frac{1}{a}\right)\right)$$

$$\leq C\left(\sum_{k=0}^{t-2}\frac{1}{2^k}\left(1+\frac{\sigma_1-\sigma_2}{b^2\sigma_1+\sigma_2}\right)^{-2t+2k}\right)$$

where we define the constant C as follows:

$$C = \left(\frac{|\tan(\theta_{1,1})|^2}{4a|\cos(\theta_{1,1})|^2}\left(b+\frac{1}{a}\right)^2 + \frac{|\tan(\theta_{1,1})|}{|\cos(\theta_{1,1})|}\left(b+\frac{1}{a}\right)\right)$$

To conclude, we will use the following elementary fact that for $\gamma \in (0,1)$, we have

$$\sum_{k=0}^{t-2}\frac{\gamma^{t-k}}{2^k} = \mathbb{1}\{\gamma = 1/2\}(t-1)\gamma^t + \mathbb{1}\{\gamma \neq 1/2\}\gamma^t\frac{1-\frac{1}{(2\gamma)^{t-1}}}{1-\frac{1}{2\gamma}}$$

$$\leq \mathbb{1}\{\gamma = 1/2\}(t-1)\gamma^t + \mathbb{1}\{\gamma \neq 1/2\}\frac{2\gamma^2}{|2\gamma-1|}\left|\gamma^{t-1} - \frac{1}{2^{t-1}}\right|$$

$$\leq \mathbb{1}\{\gamma = 1/2\}\frac{(t-1)}{2^t} + \mathbb{1}\{\gamma \neq 1/2\}\gamma^2(t-1)\left(\frac{1}{2}\vee\gamma\right)^{t-2}$$

$$\leq (t-1)\left(\gamma\vee\frac{1}{2}\right)^t$$

where in our case we have

$$\gamma = \left(1+\frac{\sigma_1-\sigma_2}{b^2\sigma_1+\sigma_2}\right)^{-2} = \left(\frac{b^2\sigma_1+\sigma_2}{(b^2+1)\sigma_1}\right)^2$$

Indeed, we obtain that

$$|\epsilon_{t+1}| \leq C\,(t-1)\left(\left(1+\frac{\sigma_1-\sigma_2}{b^2\sigma_1+\sigma_2}\right)\vee\sqrt{2}\right)^{-2t}$$

□

E Proof of Theorem 2.2

Proof of Theorem 2.2. We recall that $\tilde{M}_{\ell+1} = M - \sum_{i=1}^{\ell} \hat{\sigma}_i \hat{u}_i \hat{u}_i^\top$ and denote $M_{\ell+1} = M - \sum_{i=1}^{\ell} \sigma_i u_i u_i^\top$ with $M_1 = \tilde{M}_1 = M$. We will also denote the leading eigenvalue of $\tilde{M}_\ell$ by $\tilde{\sigma}_\ell$ and its corresponding eigenvector by $\tilde{u}_\ell$. Note that the leading singular value of M_ℓ is σ_ℓ and its corresponding vector is u_ℓ. We also note that

$$\left\| \tilde{M}_{\ell+1} - \left(M - \sum_{i=1}^{\ell} \sigma_i u_i u_i^\top \right) \right\| \leq \sum_{i=1}^{\ell} \left\| \hat{\sigma}_i \hat{u}_i \hat{u}_i^\top - \sigma_i u_i u_i^\top \right\| \tag{54}$$

$$\leq \sum_{i=1}^{\ell} \left\| \hat{\sigma}_i \hat{u}_i \hat{u}_i^\top - \tilde{\sigma}_i \tilde{u}_i \tilde{u}_i^\top \right\| + \left\| \tilde{\sigma}_i \tilde{u}_i \tilde{u}_i^\top - \sigma_i u_i u_i^\top \right\| \tag{55}$$

$$\tag{56}$$

By application of gradient descent (3), we know that if method is run for t large enough then for all for each $i \in [k]$, we have the guarantee:

$$|\tilde{\sigma}_i - \hat{\sigma}_i| \leq \epsilon \tag{57}$$

$$\|\hat{u}_i + \tilde{u}_1\| \wedge \|\hat{u}_i - \tilde{u}_i\| \leq \epsilon \tag{58}$$

$$\|\tilde{\sigma}_i \tilde{u}_i \tilde{u}_1^\top - \hat{\sigma}_i \hat{u}_i \hat{u}_i^\top\| \leq \epsilon \tag{59}$$

Now, we show by induction that things remain well behaved. (for $\ell = 1$), we have

$$|\tilde{\sigma}_1 - \sigma_1| = 0$$
$$|u_1 - \tilde{u}_1| = 0$$
$$|\sigma_1 u_1 u_1^\top - \sigma_1 \tilde{u}_1 \tilde{u}_1 = 0$$

(for $\ell = 2$) We have

$$|\tilde{\sigma}_2 - \sigma_2| \leq \epsilon \tag{60}$$

$$\|\hat{u}_2 + \tilde{u}_2\| \wedge \|\hat{u}_2 - \tilde{u}_2\| \leq \frac{\sqrt{2}\epsilon}{\sigma_2 - \sigma_3} \qquad \text{(Lemma G.10)} \tag{61}$$

$$\|\tilde{\sigma}_2 \tilde{u}_2 \tilde{u}_1^\top - \hat{\sigma}_2 \hat{u}_2 \hat{u}_2^\top\| \leq 3\epsilon + \frac{\sigma_2 \sqrt{2}\epsilon}{\sigma_2 - \sigma_3} \qquad \text{(Lemma G.11)} \tag{62}$$

(for $\ell = 3$) we have

$$|\tilde{\sigma}_3 - \sigma_3| \leq \epsilon + \left(\epsilon + 3\epsilon + \frac{\sigma_2 \sqrt{2}\epsilon}{\sigma_2 - \sigma_3} \right) = \left(5 + \frac{\sigma_2 \sqrt{2}}{\sigma_2 - \sigma_3} \right) \epsilon \tag{63}$$

$$\|\hat{u}_3 + \tilde{u}_3\| \wedge \|\hat{u}_3 - \tilde{u}_3\| \leq \frac{\sqrt{2}}{\sigma_3 - \sigma_4} \left(5 + \frac{\sigma_2 \sqrt{2}}{\sigma_2 - \sigma_3} \right) \epsilon \qquad \text{(Lemma G.10)} \tag{64}$$

$$\|\tilde{\sigma}_3 \tilde{u}_3 \tilde{u}_3^\top - \hat{\sigma}_3 \hat{u}_3 \hat{u}_3^\top\| \leq \left(3 \left(5 + \frac{\sigma_2 \sqrt{2}}{\sigma_2 - \sigma_3} \right) + \frac{\sqrt{2}\sigma_3}{\sigma_3 - \sigma_4} \left(5 + \frac{\sigma_2 \sqrt{2}}{\sigma_2 - \sigma_3} \right) \right) \epsilon \qquad \text{(Lemma G.11)} \tag{65}$$

and so forth, we see that at the end the error is at most

$$|\tilde{\sigma}_\ell - \sigma_\ell| \leq k C_3^k \left(\max_{i \in [k]} \left(\frac{\sigma_i}{\sigma_i - \sigma_{i+1}} \right) \vee 1 \right)^{k-1} \epsilon \tag{66}$$

$$\|\hat{u}_\ell + \tilde{u}_\ell\| \wedge \|\hat{u}_\ell - \tilde{u}_\ell\| \leq k C^k \frac{1}{\sigma_\ell} \left(\max_{i \in [k]} \left(\frac{\sigma_i}{\sigma_i - \sigma_{i+1}} \right) \vee 1 \right)^{k-1} \epsilon \tag{67}$$

$$\|\tilde{\sigma}_\ell \tilde{u}_\ell \tilde{u}_\ell^\top - \hat{\sigma}_\ell \hat{u}_\ell \hat{u}_\ell^\top\| \leq k C^k \left(\max_{i \in [k]} \left(\frac{\sigma_i}{\sigma_i - \sigma_{i+1}} \right) \vee 1 \right)^{k-1} \epsilon \tag{68}$$

Let us denote

$$kC^k \left(\max_{i\in[k]} \left(\frac{\sigma_i}{\sigma_i - \sigma_{i+1}} \right) \vee 1 \right)^{k-1} \epsilon = \frac{\epsilon'}{2}. \tag{69}$$

If ϵ' is such that $\sigma_i - \sigma_{i+1} > 2k\epsilon'$, for $i \in [d]$, and $\sigma_n \geq 2k\epsilon'$, then it will be ensured that the singular values of $\tilde{M}_\ell$ remain close to those of M_ℓ and $\tilde{M}_\ell$ stays positive definite. We may then apply gradient descent and use Theorem 2.1. Now, for a given ϵ', and a precision ϵ satisfying (69), by Theorem 2.1, running gradient descent long enough, namely $t_\circ$, ensures that:

$$\begin{aligned} |\tilde{\sigma}_i - \sigma_i| &\leq \epsilon' \\ \sigma_i(\|u_i + \hat{u}_i\| \wedge \|u_i - \hat{u}_i\|) &\leq \epsilon' \\ \|\sigma_i u_i u_1^\top - \hat{\sigma}_i \hat{u}_i \hat{u}_i\| &\leq \epsilon' \\ \|\sqrt{\sigma_i} u_i - \sqrt{\hat{\sigma}_i} \hat{u}_i \hat{u}_i\| \wedge \|\sqrt{\sigma_i} u_i + \sqrt{\hat{\sigma}_i} \hat{u}_i \hat{u}_i\| &\leq \epsilon' \end{aligned}$$

More precisely, $t_\circ$ is given by

$$t_\circ \geq C_1 \max_{i\in[k]} \left(\frac{\sigma_i}{\sigma_i - \sigma_{i+1}} \vee 1 \right) \left(k \log \left(C_2 \max_{i\in[k]} \left(\frac{\sigma_i}{\sigma_i - \sigma_i} \vee 1 \right) \right) + \log \left(\frac{k}{\epsilon'} \right) \right)$$

In total, we need the total number of iterations to be

$$t \geq C_1 k \max_{i\in[k]} \left(\frac{\sigma_i}{\sigma_i - \sigma_{i+1}} \vee 1 \right) \left(k \log \left(C_2 \max_{i\in[k]} \left(\frac{\sigma_i}{\sigma_i - \sigma_i} \vee 1 \right) \right) + \log \left(\frac{k}{\epsilon'} \right) \right)$$

□

F Local Convergence of Nesterov's Accelerated Gradient Descent

In this section, we prove Theorem 2.3. The key challenge in establishing this results lies in the fact that Nesterov's acceleration method is not a descent method and that the function g is only smooth and strongly-convex on a local neighborhood of the global minima (see Appendix G.2). Thus, we need to ensure that Nesterov's acceleration method remain in this local neighborhood once it enters it. To that end, we adapt the results of Nesterov and establish Theorem F.5 which may be of independent interest. The proof of Theorem F.5 is given in §F.1. The proof of Theorem 2.3 is given in §F.2.

F.1 Local acceleration for Nesterov's gradient method in its general form

Here, we present a general result regarding the local acceleration of Netsterov's method in its general form [41]. Throughout this section, we consider function f that satisfies:

Assumption F.1. f is differentiable, L-smooth and μ-strongly-convex on a local neighborhood $\mathcal{C}_\xi = \{z : \|x - x_\star\| \leq \xi\}$ of a global minima $x_\star$, for some $\mu, L, \xi > 0$.

For convenience, we define also define $\underline{\mathcal{C}}_\xi = \{z : \|x - x_\star\| \leq \xi/2\}$.

Before presenting the algorithm and result we start by introducing certain definitions and notations which are needed for describing the method. We also introduce a Lyapunov function that will be useful in characterizing the convergence properties of the method. This builds on the so-called *estimate sequence* construction (see Definition 2.2.1 in [41]).

To that end, let $(\alpha_k)_{k\geq 0}$ be a sequence taking value in $(0, 1)$, $(y_k)_{k\geq 0}$ be an arbitrary sequence taking values in $\underline{\mathcal{C}}_\xi$, and $\gamma_0 > 0$. For now the choice of this sequences will be arbitrary but will be made precise later. Define a sequence of functions $(\phi_k)_{k\geq 0}$ on $\mathbb{R}^n$ as follows:

$$\begin{aligned} \phi_0(x) &= f(x_0) + \frac{\gamma_0}{2}\|x - x_0\|^2 \\ \phi_{k+1}(x) &= (1 - \alpha_k)\phi_k(x) + \alpha_k \left(f(y_k) + \langle \nabla f(y_k), x - y_k\rangle + \frac{\mu}{2}\|x - y_k\|^2 \right) \end{aligned} \tag{70}$$

In the following Lemma, we see that the functions (ϕ_k) have quadratic form.

Lemma F.2. *The $\phi_k(\cdot)$ defined as per (70) have form, for any $x \in \mathbb{R}^n$,*

$$\phi_k(x) = \phi_k^\star + \frac{\gamma_k}{2}\|x - v_k\|^2,$$

where $\phi_0^\star = f(x_0)$, and for all $k \geq 0$,

$$\gamma_{k+1} = (1 - \alpha_k)\gamma_k + \alpha_k\mu, \qquad v_{k+1} = \frac{1}{\gamma_{k+1}}\left((1 - \alpha_k)\gamma_k v_k + \alpha_k\mu y_k - \alpha_k \nabla f(y_k)\right) \tag{71}$$

$$\begin{aligned} \phi_{k+1}^\star =& (1 - \alpha_k)\phi_k^\star + \alpha_k f(y_k) - \frac{\alpha_k^2}{2\gamma_{k+1}}\|\nabla f(y_k)\|^2 \\ &+ \frac{\alpha_k(1 - \alpha_k)\gamma_k}{\gamma_{k+1}}\left(\frac{\mu}{2}\|y_k - v_k\|^2 + \langle \nabla f(y_k), v_k - y_k\rangle\right). \end{aligned} \tag{72}$$

The proof of Lemma F.2 is analytic and follows from that of Lemma 2.2.3. in [41].

Now, we describe the generic Nesterov's method: initialize $x_0 \in \underline{\mathcal{C}}_\xi$, $v_0 = x_0$, $\gamma_0 > 0$; for $k \geq 0$, define iterates satisfying

$$\alpha_k^2 = \frac{(1 - \alpha_k)\gamma_k + \alpha_k\mu}{\ell} \tag{73}$$

$$y_k = \frac{\alpha_k\gamma_k v_k + \gamma_{k+1}x_k}{\gamma_k + \alpha_k\mu} \tag{74}$$

$$x_{k+1} \quad \text{such that} \quad f(x_{k+1}) \leq f(y_k) - \frac{1}{2\ell}\|\nabla f(y_k)\|^2, \quad \text{and} \quad x_{k+1} \in \mathcal{C}_\xi \tag{75}$$

$$\gamma_{k+1}, v_{k+1} \quad \text{as in (71)},$$

where $\ell = 2L$. In the above, the choices for α_k, y_k and x_{k+1} are motivated by the need to ensure that $\phi_{k+1}^\star \geq f(x_{k+1})$ which in turn will be useful to show decay of a certain Lyapunov function. This will be apparent shortly.

Lemma F.3. *For $k \geq 0$, if $x_k, y_k, v_k \in \underline{\mathcal{C}}_\xi$, and $\phi_k^\star \geq f(x_k)$, then $\phi_{k+1}^\star \geq f(x_{k+1})$.*

Proof. We recall that

$$\begin{aligned}\phi_{k+1}^\star =&(1-\alpha_k)\phi_k^\star + \alpha_k f(y_k) - \frac{\alpha_k^2}{2\gamma_{k+1}}\|\nabla f(y_k)\|^2 \\ &+ \frac{\alpha_k(1-\alpha_k)\gamma_k}{1+\gamma_k}\left(\frac{\mu}{2}\|y_k - v_k\|^2 + \langle \nabla f(y_k), v_k - y_k\rangle\right)\end{aligned}$$

We note that $\phi_k^\star \geq f(x_k)$ and since $x_k, y_k \in \underline{\mathcal{C}}_\xi \subseteq \mathcal{C}_\xi$ and f is strongly convex on $\mathcal{C}_\xi$ we have

$$(1-\alpha_k)\phi_k^\star + \alpha_k f(y_k) \geq f(y_k) + (1-\alpha_k)\langle \nabla f(y_k), x_k - y_k\rangle.$$

We further note that by choice of y_k, that we have

$$\langle \nabla f(y_k), (1-\alpha_k)(x_k - y_k) + \frac{\alpha_k(1-\alpha_k)\gamma_k}{\gamma_{k+1}}(v_k - y_k)\rangle = 0$$

Combinging the above inequalities with the choice of α_k and x_{k+1} yields

$$\phi_{k+1}^\star \geq f(y_k) - \frac{\alpha_k^2}{2\gamma_{k+1}}\|f(y_k)\|^2 = f(y_k) - \frac{1}{2\ell}\|f(y_k)\|^2 \geq f(x_{k+1}).$$

□

Lemma F.4. *For $k \geq 0$, if $x_k, y_k, v_k \in \underline{\mathcal{C}}_\xi$, then for $x \in \mathcal{C}_\xi$,*

$$\phi_{k+1}(x) \leq (1-\alpha_k)\phi_k(x) + \alpha_k f(x).$$

Proof. For all $x \in \mathcal{C}_\xi$,

$$\begin{aligned}\phi_{k+1}(x) &= (1-\alpha_k)\phi_k(x) + \alpha_k(f(y_k) + \langle \nabla f(y_k), x - y_k\rangle + \frac{\mu}{2}\|x - y_k\|^2) \\ &\leq (1-\alpha_k)\phi_k(x) + \alpha_k f(x)\end{aligned}$$

where in the last inequality we use that fact $x, y_k \in \underline{\mathcal{C}}_\xi \subseteq \mathcal{C}_\xi$. □

Next we argue that x_k, y_k, v_k remain in $\underline{\mathcal{C}}_\xi$ provided $v_0 = x_0$ is sufficiently close to $x_\star$.

Theorem F.5. *Assume that $\gamma_0 \in (\mu, L)$, $v_0 = x_0 \in \underline{\mathcal{C}}_\xi$ such that $\|x_0 - x_\star\| \leq (\xi/2)\sqrt{\mu/L}$. For all $k \geq 0$, we have:*

(i) $x_k, y_k, v_k \in \underline{\mathcal{C}}_\xi$,

(ii) $\phi_k^\star \geq f(x_k)$,

(iii) $\phi_{k+1}^\star - f(x_\star) + \frac{\gamma_{k+1}}{2}\|v_{k+1} - x_\star\|^2 \leq \prod_{i=0}^k (1-\alpha_i)\left(f(x_0) - f(x_\star) + \frac{\gamma_0}{2}\|x_0 - x_\star\|^2\right)$.

Proof. It is easy to note, by construction, that for all $k \geq 0$, γ_k is weighted average of μ and γ_0, thus $\mu \leq \gamma_k \leq L$. Now we prove the result by induction.

Base case. $k = 0$., **(1)** we have by definition that $x_0, v_0 \in \underline{\mathcal{C}}_\xi$. Because y_0 is a weighted average of x_0 and v_0 and $\underline{\mathcal{C}}_\xi$ is convex, we also have that $y_0 \in \underline{\mathcal{C}}_\xi$.

(2) By definition we have $\phi_0^\star = f(x_0)$.

(3) Next, since $x_0, y_0, v_0 \in \underline{\mathcal{C}}_\xi$, we have by Lemma F.4, that for all $x \in \mathcal{C}_\xi$

$$\phi_1(x) \leq (1-\alpha_0)\phi_0(x) + \alpha_0 f(x),$$

and recalling that $\phi_k(x) = \phi_k^\star + \frac{\gamma_k}{2}\|v_k - x_\star\|^2$, we have, in particular ($x = x_\star$),

$$\phi_1^\star - f(x_\star) + \frac{\gamma_1}{2}\|v_1 - x_\star\|^2 = \phi_1(x_\star) - f(x_\star) \leq (1-\alpha_0)\left(f(x_0) - f(x_\star) + \frac{\gamma_0}{2}\|x_0 - x_\star\|^2\right).$$

Induction Step. $k \geq 1$. Assume the desired statements are true for $k-1$:

(i) $x_{k-1}, y_{k-1}, v_{k-1} \in \underline{C}_\xi$,

(ii) $\phi_{k-1}^\star \geq f(x_{k-1})$,

(iii) $\phi_k^\star - f(x_\star) + \frac{\gamma_k}{2}\|v_k - x_\star\|^2 \leq \prod_{i=0}^{k-1}(1-\alpha_i)\left(f(x_0) - f(x_\star) + \frac{\gamma_0}{2}\|x_0 - x_\star\|^2\right)$.

We wish to argue that the above remains true for k. We argue that next.

(1) First, since $x_{k-1}, y_{k-1}, v_{k-1} \in \underline{\mathcal{C}}_\xi$, and $\phi_{k-1}^\star \geq f(x_{k-1})$, then by Lemma F.3, we also have $\phi_k^\star \geq f(x_k)$.

(2) Next, combining the fact that $\phi_k^\star \geq f(x_k)$ with the inequality (iii) from the induction hypothesis (for $k-1$), we obtain

$$f(x_k) - f(x_\star) + \frac{\gamma_k}{2}\|v_k - x_\star\|^2 \leq \prod_{i=0}^{k-1}(1-\alpha_i)\left(f(x_0) - f(x_\star) + \frac{\gamma_0}{2}\|x_0 - x_\star\|^2\right). \tag{76}$$

Because by choice of x_k, we have $x_k, x_\star \in \mathcal{C}_\xi$, and because f is L-smooth and μ-strongly convex on $\mathcal{C}_\xi$, it follows that

$$f(x_k) - f(x_\star) \geq \frac{\mu}{2}\|x_1 - x_\star\|^2 \qquad \text{and} \qquad f(x_0) - f(x_\star) \leq \frac{L}{2}\|x_0 - x_\star\|^2,$$

Combining the above inequalities with (76), yields

$$(\|x_\star - x_k\| \vee \|x_\star - v_k\|)^2 \leq \prod_{i=0}^{k-1}(1-\alpha_i)\left(\frac{L+\gamma_0}{\mu+\gamma_1}\right)\|x_\star - x_0\|^2 \leq \left(\frac{L}{\mu}\right)\|x_\star - x_0\|^2 \leq \frac{\xi^2}{4},$$

where we used the fact that $\alpha_i \in (0,1)$ for all $i \geq 0$, and $\|x_\star - x_0\| \leq (\xi/2)\sqrt{\mu/L}$. This means that $x_k, v_k \in \underline{\mathcal{C}}_\xi$. Since y_k is weighted average of x_k, v_k, and $\underline{\mathcal{C}}_\xi$ is convex, we also have $y_k \in \underline{\mathcal{C}}_\xi$.

(3) Since $x_{k-1}, y_{k-1}, v_{k-1} \in \underline{\mathcal{C}}_\xi$, by Lemma F.4, we have

$$\phi_k(x) \leq (1-\alpha_{k-1})\phi_{k-1}(x) + \alpha_{k-1}f(x).$$

In particular, with $x = x_\star \in \mathcal{C}_\xi$, we have

$$\begin{aligned}\phi_k(x_\star) - f(x_\star) &\leq (1-\alpha_{k-1})(\phi_{k-1}(x_\star) - f(x_\star)) \\ &\leq (1-\alpha_{k-1})\left(\phi_{k-1}^\star - f(x_\star) + \frac{\gamma_{k-1}}{2}\|v_{k-1} - x_\star\|^2\right)\end{aligned}$$

Recalling that $\phi_k(x_\star) = \phi_k^\star + \frac{\gamma_k}{2}\|v_k - x_\star\|^2$, and using the above inequality together with the induction hypothesis gives

$$\phi_{k+1}^\star - f(x_\star) + \frac{\gamma_{k+1}}{2}\|v_{k+1} - x_\star\|^2 \leq \prod_{i=0}^{k}(1-\alpha_i)\left(f(x_0) - f(x_\star) + \frac{\gamma_0}{2}\|x_0 - x_\star\|^2\right)$$

□

We will next provide the following result which quantifies the growth of $\prod_{i=0}^k(1-\alpha_i)$.

Lemma F.6 (Lemma 2.2.4 in [41]). *For $\gamma_0 \geq \mu$, we have*

$$\prod_{i=0}^{k}(1-\alpha_i) \leq \min\left\{\left(1 - \sqrt{\frac{\mu}{\ell}}\right)^k, \frac{4\ell}{(2\sqrt{\ell} + k\sqrt{\gamma_0})^2}\right\}.$$

F.2 Proof of Theorem 2.3

To prove Theorem 2.3 we apply Theorem F.5. To that end, let us start by defining $x_\star \in \{-\sqrt{\sigma_1}u_1, +\sqrt{\sigma_1}u_1\}$, $\xi := \frac{\sigma_1 - \sigma_2}{15\sqrt{\sigma_1}}$, and

$$\mathcal{C} := \mathcal{C}_\xi = \{x : \|x - x_\star\| \leq \xi\}, \qquad \underline{\mathcal{C}} := \underline{\mathcal{C}}_\xi = \{x : \|x - x_\star\| \leq \xi/2\}.$$

From Lemma G.9, we see that the set $\mathcal{C}$ constitute a basin of attraction for the function g, in the sense that g is L-smooth and μ-strongly-convex with

$$L = \frac{9\sigma_1}{2}, \qquad \mu = \frac{\sigma_1 - \sigma_2}{4}.$$

Next, we that x_{k+1} is updated as follows:

$$x_{k+1} = y_k - \frac{1}{6\|y_k\|^2}\nabla g(y_k).$$

Before applying Theorem F.5, we need to ensure the following result:

Lemma F.7. *Assume that $y_k \in \underline{\mathcal{C}}_\xi$. Then, $g(x_{k+1}) \leq g(y_k) - \frac{1}{4L}\|\nabla g(y_k)\|^2$.*

Proof. We start by noting that $y_k \in \underline{\mathcal{C}}$. In particular, we have $\|y_k - x_\star\| \leq \frac{\sigma_1-\sigma_2}{15\sqrt{\sigma_1}} \leq \frac{\sigma_1}{(\sqrt{6}+2)\sqrt{\sigma_1}}$ and this implies that

$$\frac{3\sigma_1}{4} \leq \|y_t\|^2 \leq \frac{3\sigma_1}{2}. \tag{77}$$

Next, we have

$$x_{k+1} - x_\star = y_k - x_\star - \frac{1}{6\|y_k\|^2}\nabla g(y_k) \tag{78}$$

Because $y_k \in \underline{\mathcal{C}}$, by L-smoothness of g on $\mathcal{C}$, we have $\|\nabla g(y_k)\| \leq L\|y_k - x_\star\|$. Starting from the above equality and using triangular inequality gives

$$\|x_{k+1} - x_\star\| \leq \left(1 + \frac{L}{6\|y_k\|^2}\right)\|y_k - x_\star\| \leq 2\|y_k - x_\star\| \leq \xi$$

Thus, $x_{k+1} \in \mathcal{C}$. Again, using the fact that g is L-smooth on $\mathcal{C}$, we also have

$$\begin{aligned}
g(x_{k+1}) &\leq g(y_k) + \langle \nabla g(y_k), y_k - x_{k+1}\rangle + \frac{L}{2}\|y_k - x_{k+1}\|^2 && (L-\text{smoothness}) \\
&= g(y_k) - \left(\frac{1}{6\|y_k\|^2} - \frac{L}{72\|y_k\|^4}\right)\|\nabla g(y_k)\|^2 && (\text{using (78)}) \\
&\leq g(y_k) - \frac{1}{18\sigma_1}\|\nabla g(y_k)\|^2.
\end{aligned}$$

The last inequality follows because, using (43), we have

$$\left(\frac{1}{6\|y_k\|^2} - \frac{L}{72\|y_k\|^4}\right) = \frac{1}{6\|y_k\|^2}\left(1 - \frac{9\sigma_1}{24\|y_k\|^2}\right) \geq \frac{1}{9\sigma_1}\left(1 - \frac{1}{2}\right) = \frac{1}{18\sigma_1}.$$

This concludes the proof. □

Specializing further the general scheme of Nesterov's accelerated gradient descent to the case $\gamma_0 = \mu$, and x_{k+1} as per presented before gives the scheme: for all $k \geq 0$

$$\begin{aligned}
\alpha_k &= \alpha := \sqrt{\frac{\mu}{2L}} \\
\gamma_k &= \mu \\
y_k &= \frac{1}{1+\alpha}x_k + \frac{\alpha}{1+\alpha}v_k \\
x_{k+1} &= y_k - \frac{1}{6\|y_k\|^2}\nabla g(y_k) \\
v_{k+1} &= (1-\alpha)v_k + \alpha\left(y_k - \frac{1}{\mu}\nabla g(y_k)\right)
\end{aligned}$$

with $v_0 = x_0$ and $\|x_0 - x_\star\| \leq (\xi/2)\sqrt{\mu/L}$. Immediate application of Theorem F.5 gives

$$g(x_{k+1}) - g(x_\star) + \frac{\mu}{2}\|v_{k+1} - x_\star\|^2 \leq \min\left\{\left(1 - \sqrt{\frac{\mu}{2L}}\right)^k, \frac{8L}{(2\sqrt{2L} + k\sqrt{\mu})^2}\right\} \times \left(g(x_0) - g(x_\star) + \frac{\mu}{2}\|x_0 - x_\star\|^2\right)$$

Since $x_{k+1}, x_\star, x_0, \in \mathcal{C}_\xi$, we have by μ-strong convexity of g on $\mathcal{C}$ we deduce that

$$\frac{\mu}{2}\|x_{k+1} - x_\star\|^2 \leq \min\left\{\left(1 - \sqrt{\frac{\mu}{2L}}\right)^k, \frac{8L}{(2\sqrt{2L} + k\sqrt{\mu})^2}\right\} \frac{L+\mu}{2}\|x_0 - x_\star\|^2$$

With the choice of x_0, using the inequality $(L/\mu)\|x_0 - x_\star\| \leq \xi$, with the above inequality and replacing L and μ by there values gives

$$\|x_{k+1} - x_\star\|^2 \leq \min\left\{\left(1 - \sqrt{\frac{\sigma_1 - \sigma_2}{36\sigma_1}}\right)^k, \frac{144\sigma_1}{(12\sqrt{\sigma_1} + k\sqrt{\sigma_1 - \sigma_2})^2}\right\}\left(\frac{\sigma_1 - \sigma_2}{15\sqrt{\sigma_1}}\right).$$

We conclude by noting that Lemma G.12 ensures that

$$\begin{aligned}
\|x_{k+1} - x_\star\|^2 &= |\|x_{k+1}\| - \sqrt{\sigma_1}|^2 + \|x_{k+1}\|\sqrt{\sigma_1}\left\|\frac{x_{k+1}}{\|x_{k+1}\|} \pm u_1\right\|^2 \\
&\geq |\|x_{k+1}\| - \sqrt{\sigma_1}|^2 + \frac{\sqrt{3}\sigma_1}{2}\left\|\frac{x_{k+1}}{\|x_{k+1}\|} \pm u_1\right\|^2 \\
&\geq \max\left\{|\|x_{k+1}\| - \sqrt{\sigma_1}|^2, \frac{\sqrt{3}\sigma_1}{2}\left\|\frac{x_{k+1}}{\|x_{k+1}\|} \pm u_1\right\|^2\right\}
\end{aligned}$$

where the second to last inequality follows from the fact that $x_{k+1} \in \mathcal{C}$, and thus $\|x_{k+1}\|^2 \geq 3\sigma_1/4$. This concludes the proof.

Remark F.8. Note that in the proof of the result, we can replace $\frac{\sigma_1 - \sigma_2}{4}$ by μ for any value of $\mu \leq \frac{\sigma_1 - \sigma_2}{4}$ and the result will still hold with an error rate that is of order

$$\|x_{k+1} - x_\star\|^2 \leq \min\left\{\left(1 - \sqrt{\frac{\mu}{2L}}\right)^k, \frac{8L}{(2\sqrt{2L} + k\sqrt{\mu})^2}\right\}\left(\frac{\sigma_1 - \sigma_2}{15\sqrt{\sigma_1}}\right)$$

G Miscellaneous tools and lemmas

In this section, we present some tools and concepts that we make use of consistently in our proofs. A review of strong convexity and smoothness are presented in §G.1. In §G.2, we present Lemma G.8 characterizing the critical points of the non-convex function g, and Lemma G.9 which establishes the strong convexity and smoothness of g locally. Finally, in §G.3 we present a few error decomposition lemmas.

G.1 Smoothness and strong convexity

In this subsection we review some concepts and results from convex optimization that are relevant to our analysis. We will not prove these results as these are standard (e.g., see [19]).

The first property corresponds to that of smoothness which is characterized as follows:

Definition G.1 (L-smoothness)**.** Let f be a differentiable function on $\mathcal{C}$. We say that it is L-smooth on $\mathcal{C}$ if for all $x, y \in \mathcal{C}$, we have

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|.$$

An important consequence of a function being L-smooth is the following inequality:

Lemma G.2. *Let f be a differentiable function on $\mathcal{C}$. If it is L-smooth on $\mathcal{C}$, than for all $x, y \in \mathcal{C}$,*

$$f(x) \leq f(y) + \langle \nabla f(y), x - y\rangle + \frac{L}{2}\|x - y\|^2.$$

A consequence of the above result is the following:

Lemma G.3. *Let f be a differentiable function on $\mathcal{C}$. Assume that f is L-smooth on $\mathcal{C}$, and $x_\star \in \mathcal{C}$ is a critical point of f (i.e., $\nabla f(x_\star) = 0$), then for all $x \in \mathcal{X}$,*

$$\|\nabla f(x)\| \leq L\|x - x_\star\| \qquad \text{and} \qquad f(x) - f(x_\star) \leq \frac{L}{2}\|x - x_\star\|^2$$

To verify that a function f that is twice differentiable is L-smooth for some $L > 0$, we often inspect the largest eigenvalue of its Hessian. The following lemma clarifies why.

Lemma G.4. *Let f be a twice differentiable function on $\mathcal{C}$. If $\lambda_{\max}(\nabla^2 f(x)) \leq L$ for all $x \in \mathcal{C}$, then f is L-smooth on $\mathcal{C}$.*

The second property concerns the convexity of a given function, namely that of strong convexity. Specifically, this property is often characterized via an important inequality as we present below:

Definition G.5. Let f be a differentiable function on $\mathcal{C}$. We say that it is μ-strongly convex on $\mathcal{C}$ if for all $x, y \in \mathcal{C}$,

$$f(x) \geq f(y) + \langle \nabla f(x), y - x\rangle + \frac{\mu}{2}\|y - x\|^2.$$

An immediate consequence of strong convexity is the following result

Corollary G.6. *Let f be a differentiable function on $\mathcal{C}$. Assume that f is μ-strongly convex on $\mathcal{C}$, and $x_\star \in \mathcal{C}$ is a critical point of f (i.e., $\nabla f(x_\star) = 0$), then for all $x \in \mathcal{X}$,*

$$f(x) - f(x_\star) \geq \frac{\mu}{2}\|x - x_\star\|^2$$

To verify whether a function f that is twice differentiable is μ-strongly convex for some $\mu > 0$, we often inspect the minimum eigenvalue of its Hessian.

Lemma G.7. *Let f be a twice differentiable function on $\mathcal{C}$. If $\lambda_{\min}(\nabla^2 f(x)) \geq \mu$ for all $x \in \mathcal{C}$, then f is μ-strongly-convex on $\mathcal{C}$.*

G.2 Properties of the non-convex function g

The properties of the function g and its relation to the spectral properties of M have pointed out in early works (e.g., [3, 14]). Here we overview some of these properties which are relevant to our analysis. In the following lemma, we present a characterization of the properties of critical points of g:

Lemma G.8 (1^{st} and 2^{nd} order optimality conditions)**.** *Let $x \in \mathbb{R}^n$, the following properties hold:*

(i) x is a critical point, i.e. $\nabla g(x; M) = 0$, if and only if $x = 0$ or $x = \pm\sqrt{\sigma_i} u_i$, $i \in [k]$.

(ii) if for all $i \neq 1$, $\sigma_1 > \sigma_i$, then all local minima of the loss function g are also global minima[4], 0 is a local maxima, and all other critical points are strict saddle[5]. More specifically, the only local and global minima of g are the points $-\sqrt{\sigma_1} u_1$ and $+\sqrt{\sigma_1} u_1$.

A proof of Lemma G.8 can be found for instance in [14].

The function g is locally smooth and strongly convex. This statement is made precise in the following lemma:

Lemma G.9. *Define $\mathcal{C} = \{x \in \mathbb{R}^n : \|x \pm \sqrt{\sigma_1} u_1\| \leq \frac{\sigma_1 - \sigma_2}{15\sqrt{\sigma_1}}\}$. Then for all $x \in \mathcal{C}$, we have*

$$\frac{(\sigma_1 - \sigma_2)}{4} I_n \preceq \nabla^2 g(x; M) \preceq \frac{9\sigma_1}{2} I_n$$

The proof follows that of [14], and we provide it below for completeness.

Proof of Lemma G.9. Let $x \in \mathcal{C}$, we can easily verify that

$$\sigma_1 - 0.25(\sigma_1 - \sigma_2) \leq \|x\|^2 \leq 1.15\sigma_1 \quad \text{and} \quad \|x - x_\star\|\|x\| \leq (\sigma_1 - \sigma_2)/12 \tag{79}$$

Proving the upper bound. First, we recall that

$$\nabla^2 g(x; M) = \|x\|^2 I_n + 2xx^\top - M,$$

Thus, using the inequalities (79), we obtain

$$\|\nabla^2 g(x; M)\| \leq 3\|x\|^2 + \sigma_1 \leq 4.5\sigma_1$$

Proving the lower bound. We start start by lower bounding $xx^\top$ for $x \in \mathcal{C}$,

$$\begin{aligned} xx^\top &= x_\star x_\star^\top + x_\star(x - x_\star)^\top + (x - x_\star)x_\star^\top + (x - x_\star)(x - x_\star)^\top \\ &\succeq \sigma_1 u_1 u_1^\top - 2\|x - x_\star\|\|x_\star\| I_n \\ &\succeq \sigma_1 u_1 u_1^\top - 0.25(\sigma_1 - \sigma_1) I_n \end{aligned}$$

We know that $I_n = \sum_{i=1}^n u_i u_i^\top$ and $M = \sum_{i=1}^n \sigma_i u_i u_i^\top$. We may therefore write

$$\begin{aligned} \nabla^2 g(x) &\succeq \|x\|^2 \sum_{i=1}^n u_i u_i^\top + 2\left(\sigma_1 u_1 u_1^\top - 0.25(\sigma_1 - \sigma_2)\sum_{i=1}^n u_i u_i^\top\right) - \sum_{i=1}^n \sigma_i u_i u_i^\top \\ &\succeq \|x\|^2 \sum_{i=1}^n u_i u_i^\top + 2\left(\sigma_1 u_1 u_1^\top - 0.25(\sigma_1 - \sigma_2)\sum_{i=1}^n u_i u_i^\top\right) - \sum_{i=1}^n \sigma_i u_i u_i^\top \\ &\succeq (\|x\|^2 + \sigma_1 - 0.5(\sigma_1 - \sigma_2)) u_1 u_1^\top + \sum_{i=2}^n (\|x\|^2 - \sigma_i - 0.5(\sigma_1 - \sigma_2)) u_i u_i^\top \\ &\succeq \left(\|x\|^2 - 0.5(\sigma_1 - \sigma_2) - \sigma_2\right) \sum_{i=1}^n u_i u_i^\top \\ &\succeq 0.25(\sigma_1 - \sigma_2) I_n \end{aligned}$$

where in the last inequality we used (79). This concludes our proof. □

[4] We recall that x is a local minima of g iff $\nabla g(x) = 0$ and $\lambda_{\min}(\nabla^2 g(x)) \geq 0$.

[5] We recall that x is a strict saddle point iff $\nabla g(x) = 0$, and $\lambda_{\min}(\nabla^2 g(x)) < 0$.

Defining the parameters

$$\mu = \frac{\sigma_1 - \sigma_2}{4} \qquad \text{and} \qquad L = \frac{9\sigma_1}{2},$$

we note that Lemma G.9 ensures that the function g enjoys two key properties. First, g is locally L-smooth near its global minima, meaning that for all $x, y \in \mathcal{C}$,

$$g(x) \leq g(y) + \langle \nabla g(y), x - y \rangle + \frac{L}{2}\|x - y\|^2.$$

Second, g is locally μ-strongly convex near its global minima, meaning that for all $x, y \in \mathcal{C}$,

$$g(x) \geq g(y) + \langle \nabla g(y), x - y \rangle + \frac{\mu}{2}\|x - y\|^2.$$

G.3 Error decompositions

Lemma G.10. *Let $M, \tilde{M} \in \mathbb{R}^{n\times n}$ symmetric matrices. Let $\sigma_1, \ldots, \sigma_n$ (resp. $\tilde{\sigma}_1, \ldots, \tilde{\sigma}_n$) be the eigenvalues of M (resp. $\tilde{M}$) in decreasing order, and $u_1, \ldots, u_d$ (resp. $\tilde{u}_1, \ldots, \tilde{u}_k$) be their corresponding eigenvectors. Then, for all $\ell \in [k]$, we have*

$$\min_{W \in \mathcal{O}^{\ell\times\ell}} \|U_{1:\ell} - \tilde{U}_{1:\ell} W\| \leq \sqrt{2}\frac{\|M - \tilde{M}\|}{\sigma_\ell - \sigma_{\ell+1}} \tag{80}$$

$$|\sigma_i - \tilde{\sigma}_i| \leq \|M - \widetilde{M}\| \tag{81}$$

where $\mathcal{O}^{\ell\times\ell}$ denotes the set of $\ell \times \ell$ orthogonal matrices, and using the convention that $\sigma_{d+1} = 0$.

Proof of Lemma G.10. The proof of the result is an immediate consequence of Davis-Kahan and the inequality $\min_{W\in\mathcal{O}^{\ell\times\ell}} \|U_{1:\ell} - \tilde{U}_{1:\ell}W\| \leq \|\sin(U_{1:\ell}, \tilde{U}_{1:\ell})\|$ (e.g., see [10]). The second inequality is simply Weyl's inequality. □

Lemma G.11. *For all $x \in \mathbb{R}^n \backslash \{0\}$, the following inequalities hold*

$$\|x - \sqrt{\sigma_1}u_1\|^2 = \sigma_1 \left(\left(\frac{\|x\|}{\sqrt{\sigma_1}} - 1 \right)^2 + \frac{\|x\|}{\sqrt{\sigma_1}} \left\| \frac{x}{\|x\|} - u_1 \right\|^2 \right)$$

$$\|x + \sqrt{\sigma_1}u_1\|^2 = \sigma_1 \left(\left(\frac{\|x\|}{\sqrt{\sigma_1}} - 1 \right)^2 + \frac{\|x\|}{\sqrt{\sigma_1}} \left\| \frac{x}{\|x\|} + u_1 \right\|^2 \right)$$

$$\|xx^\top - \sigma_1 u_1 u_1^\top\|^2 \leq \sigma_1^2 \left(\left(\frac{\|x\|}{\sqrt{\sigma_1}} - 1 \right)^2 \left(\frac{\|x\|}{\sqrt{\sigma_1}} + 1 \right)^2 + \frac{\|x\|^2}{2\sigma_1} \left\| \frac{x}{\|x\|} - u_1 \right\|^2 \left\| \frac{x}{\|x\|} + u_1 \right\|^2 \right)$$

The proof of Lemma G.11. The first and second equality follow immediately by invoking Lemma G.12 with $y = \sqrt{\sigma_1}u_1$ and $y = -\sqrt{\sigma_1}u_1$. The third inequality follows by first upper bounding $\|xx^\top - \sigma_1 u_1 u_1^\top\| \leq \|xx^\top - \sigma_1 u_1 u_1^\top\|_{\mathrm{F}}$, then using Lemma G.13. □

Lemma G.12. *Let $x, y \in \mathbb{R}^n \backslash \{0\}$. The following equality holds*

$$\|x - y\|^2 = |\|x\| - \|y\||^2 + \|x\|\|y\| \left\| \frac{x}{\|x\|} - \frac{y}{\|y\|} \right\|^2$$

Proof of Lemma G.12. We have through simple algebra the following:

$$\begin{aligned}
\|x - y\|^2 &= \|x\|^2 + \|y\|^2 - 2\langle x, y \rangle \\
&= (\|x\| - \|y\|)^2 + 2\|x\|\|y\| - 2\langle x, y \rangle \\
&= (\|x\| - \|y\|)^2 + \|x\|\|y\| \left(2 - 2\left\langle \frac{x}{\|x\|}, \frac{y}{\|y\|} \right\rangle \right) \\
&= (\|x\| - \|y\|)^2 + \|x\|\|y\| \left\| \frac{x}{\|x\|} - \frac{y}{\|y\|} \right\|^2.
\end{aligned}$$

This concludes the proof. □

Lemma G.13. *Let $x, y \in \mathbb{R}^n \backslash \{0\}$. It also holds that*

$$\|xx^\top - yy^\top\|_{\mathrm{F}}^2 = \left(\|x\|^2 - \|y\|^2\right)^2 + \frac{\|x\|^2\|y\|^2}{2}\left\|\frac{x}{\|x\|} - \frac{y}{\|y\|}\right\|^2\left\|\frac{x}{\|x\|} + \frac{y}{\|y\|}\right\|^2$$

Proof of Lemma G.13. We have through simple algebra

$$\begin{aligned}
\left\|xx^\top - yy^\top\right\|_{\mathrm{F}}^2 &= \left\|xx^\top\right\|_{\mathrm{F}}^2 + \left\|yy^\top\right\|_{\mathrm{F}}^2 - 2\,\mathrm{tr}(xx^\top yy^\top) \\
&= \|x\|^4 + \|y\|^4 - 2|\langle x, y\rangle|^2 \\
&= \left(\|x\|^2 - \|y\|^2\right)^2 + 2\|x\|^2\|y\|^2 - 2|\langle x, y\rangle|^2 \\
&= \left(\|x\|^2 - \|y\|^2\right)^2 + \frac{\|x\|^2\|y\|^2}{2}\left(2 - 2\left\langle\frac{x}{\|x\|}, \frac{y}{\|y\|}\right\rangle\right)\left(2 + 2\left\langle\frac{x}{\|x\|}, \frac{y}{\|y\|}\right\rangle\right) \\
&= \left(\|x\|^2 - \|y\|^2\right)^2 + \frac{\|x\|^2\|y\|^2}{2}\left\|\frac{x}{\|x\|} - \frac{y}{\|y\|}\right\|^2\left\|\frac{x}{\|x\|} + \frac{y}{\|y\|}\right\|^2
\end{aligned}$$

This concludes the proof. □

H Pseudo-codes

Here we presented pseudo-codes detailing k-SVD with gradient descent and with power iterations.

Algorithm 1: k-SVD via gradient descent (`GDSVD`)

Input a symmetric matrix M, approximation rank k, step-size parameter η, tolerance ϵ ;
$M_1 \leftarrow M$;
for $\ell = 1, \ldots, k$ **do**
  $x_0 \leftarrow M_\ell z$ where z is random unit norm vector in $\mathbb{R}^n$; ▷ `initialization`
  $t \leftarrow 0$;
  $\epsilon_0^\sigma, \epsilon_0^u \leftarrow 2\epsilon$;
  while $(\epsilon_t^\sigma > \epsilon)$ *or* $(\epsilon_t^\sigma > \epsilon)$ **do**
    $x_{t+1} \leftarrow x_t - \frac{\eta}{\|x_t\|^2}\nabla g(x_t; M_\ell)$; ▷ `gradient descent step`
    $\epsilon_{t+1}^u \leftarrow \left\|\frac{x_{t+1}}{\|x_{t+1}\|} - \frac{x_t}{\|x_t\|}\right\|$; ▷ `approximation error of` u_ℓ
    $\epsilon_{t+1}^\sigma \leftarrow \left|\|x_{t+1}\|^2 - \|x_t\|^2\right|$; ▷ `approximation error of` σ_ℓ
    $t \leftarrow t + 1$;
  $\hat{\sigma}_\ell \leftarrow \|x_t\|^2$, $\hat{u}_\ell \leftarrow \frac{x_t}{\|x_t\|}$; ▷ `Recovery of` $\hat{\sigma}_\ell$ `and` $\hat{u}_\ell$
  $M_{\ell+1} \leftarrow M_\ell - \hat{\sigma}_\ell \hat{u}_\ell \hat{u}_\ell^\top$;
return $\hat{\sigma}_1, \ldots, \hat{\sigma}_k$ *and* $\hat{u}_1, \ldots, \hat{u}_k$;

Algorithm 2: k-SVD via power iterations (`Power Method`)

Input a symmetric matrix M, approximation rank k, step-size parameter η, tolerance ϵ ;
$M_1 \leftarrow M$;
for $\ell = 1, \ldots, k$ **do**
  $x_0 \leftarrow M_\ell z$ where z is random unit norm vector in $\mathbb{R}^n$; ▷ `initialization`
  $t \leftarrow 0$;
  $\epsilon_0^\sigma, \epsilon_0^u \leftarrow 2\epsilon$;
  while $(\epsilon_t^\sigma > \epsilon)$ *or* $(\epsilon_t^\sigma > \epsilon)$ **do**
    $x_{t+1} \leftarrow \frac{Mx_t}{\|Mx_t\|}$; ▷ `power iteration step`
    $\epsilon_{t+1}^u \leftarrow \|x_{t+1} - x_t\|$; ▷ `approximation error of` u_ℓ
    $\epsilon_{t+1}^\sigma \leftarrow \left|\|Mx_{t+1}\| - \|Mx_t\|\right|$; ▷ `approximation error of` σ_ℓ
    $t \leftarrow t + 1$;
  $\hat{\sigma}_\ell \leftarrow \|Mx_t\|$, $\hat{u}_\ell \leftarrow x_t$; ▷ `Recovery of` $\hat{\sigma}_\ell$ `and` $\hat{u}_\ell$
  $M_{\ell+1} \leftarrow M_\ell - \hat{\sigma}_\ell \hat{u}_\ell \hat{u}_\ell^\top$;
return $\hat{\sigma}_1, \ldots, \hat{\sigma}_k$ *and* $\hat{u}_1, \ldots, \hat{u}_k$;

I Experiments

We present here few experimental results illustrating the performance of k-SVD with gradient descent, denoted `GDSVD`. The method was implemented in both `C` and `Python`. We mainly compare with the `Power Method`.

I.1 Experimental Data

Synthetic data. We utilize synthetic data to evaluate various aspects of the method. This involves generating matrices of the form $M = U\Sigma V^\top$ where U, V are sampled uniformly at random from the space of semi-orthogonal matrices of size $n \times d$, and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_d)$ where $\sigma_1, \ldots, \sigma_d$ being the singular values of M. The choice of number of non-zero singular values, i.e. $d \geq 1$ and their magnitudes are done in a few different ways to capture various types of matrices:

*Rank-*1 *matrices.* Here $d = 1$. These are used to inspect the performance of the methods with respect to the size of the matrix. The matrices were generated for up to size $10^5 \times 10^5$.

*Rank-*2 *matrices.* Here $d = 2$. These are used to inspect the performance of the methods with respect to the gap $\sigma_1 - \sigma_2$. For these matrices, we set $\sigma_1 = 1$ and vary $\sigma_1 - \sigma_2$ in $\{10^{-k/4} : k \in [20]\}$.

Rank-$\lfloor \log(n) \rfloor$ *matrices.* These are used to inspect the robustness of the compared methods with respect to the heterogeneity in the distribution of singular values. Specifically, three types of distribution are chosen.

1. Exponential Decay: Sample $a \sim \mathrm{Unif}(2, \ldots, 10)$ and set $\sigma_i = a^{-i}$, for $i \in [d]$.
2. Polynomial Decay: Set $\sigma_i = i^{-1} + 1$ for $i \in [d]$.
3. Linear Decay: Sample $a \sim \mathrm{Unif}(1, \ldots, 10)$, $b \sim \mathrm{Unif}([0, 1])$, then set $\sigma_i = a - bi$ for $i \in [d]$.

Real-word matrices. To evaluate performance on real-world data, we utilize matrices from datasets `MNIST` and `MovieLens(10K, 1M, 10M)`.

I.2 Implementation Details

`GDSVD`**.** The implementation `GDSVD` is done by sequentially applying gradient descent to find one singular value and corresponding vector at a time, until all $k \geq 1$ are found. For every run of gradient descent, the stopping conditions utilized is: given parameter $\epsilon > 0$, stop at iteration $t \geq 2$ if

$$\begin{aligned} \left\| \|x_t\|^{-1} x_t - \|x_{t-1}\|^{-1} x_{t-1} \right\| &< \epsilon \\ \left| \|x_t\| - \|x_{t-1}\| \right| &< \epsilon. \end{aligned} \tag{82}$$

In all the experiments, we utilize $\epsilon = 10^{-8}$. A detailed pseudo-code for the implementation is provided in Algorithm 1 provided in Appendix **??**.

For acceleration method, we implement two approaches: the Polyak's and Nesterov's. The Polyak's acceleration `GDSVD(Polyak)` uses Polyak's momentum, i.e., the scheme (8) with $\alpha = 0$. The acceleration is implemented beyond iteration $t > 100\beta$ steps before adding momentum in order for the method to converge. The Nesterov's acceleration `GDSVD(Nesterov)` uses Nesterov's momentum, i.e., the scheme (8) with $\beta = 0$. This method is robust in that the acceleration can be implemented starting from $t \geq 1$. For both `GDSVD(Polyak)` and `GDSVD(Nesterov)`, different values of β were tested and the best was reported.

Power iteration. We implement a power method for finding the k-SVD of a symmetric M. As with the gradient based method, we also proceed sequentially finding one singular value at a time and corresponding vector at a time. The method proceeds by iterating the equations $x_{t+1} = \|Mx_t\|^{-1} Mx_t$ until convergence. The stopping conditions utilized: given parameter $\epsilon > 0$, stop at iteration $t \geq 2$ if

$$\begin{aligned} \|x_{t+1} - x_t\| &< \epsilon \\ \left| \|Mx_{t+1}\| - \|Mx_t\| \right| &< \epsilon. \end{aligned} \tag{83}$$

The method is initialized in the same fashion as the one with gradient descent. See Algorithm 2 in Appendix **??** for a detailed pseudo-code.

Dealing with asymmetric matrices. When considering asymmetric matrices M in the experiments, for both `GDSVD` and `Power Method`, we apply k-SVD to $MM^\top$ to identify the leading k singular values of M, $\hat{\sigma}_1, \ldots, \hat{\sigma}_k$ and corresponding left singular vectors $\hat{u}_1, \ldots, \hat{u}_k$, we then simply recover the right singular vectors by setting $\hat{v}_i = {\hat{\sigma}_i}^{-1} M^\top \hat{u}_i$, for $i \in [k]$.

Computation of gradients and matrix-vector multiplications. In all implemented methods in `C`, the computation of gradients or more generally matrix-vector multiplications, were parallelized using CBLAS (`https://www.netlib.org/blas/`) and its default thread parameter settings, or with OpenCilk (`https://www.opencilk.org/`) for efficient memory management and for-loops parallelization.

Machine characteristics. All experiments were carried on machines with a 6-core, 12-thread Intel chip, or a 10-core, 20-thread Intel chip.

I.3 Results

We start by summarizing the key findings of the experiments along with the supporting evidence from experimental results for these findings. They are as follows.

First, `GDSVD` is robust to different singular value decay distribution and scale gracefully with the matrix size n. This is inline with the theoretical results. This can be concluded from the results listed in Table 3 across various datasets. Also see Figure 2. While $\eta = \frac{1}{2}$ is a good choice for `GDSVD`, to understand impact of $\eta \in (0, 1)$ on the performance, as seen from Figure 2, `GDSVD` still works with different values of $\eta \in (0, 1)$ despite our proof only holding for $\eta = 1/2$. The choice of $\eta = 1/2$ in our theoretical statement was made to simplify the exposition of our proof analysis, these experimental results reinforce our claims that the method should converge for all $\eta \in (0, 1)$.

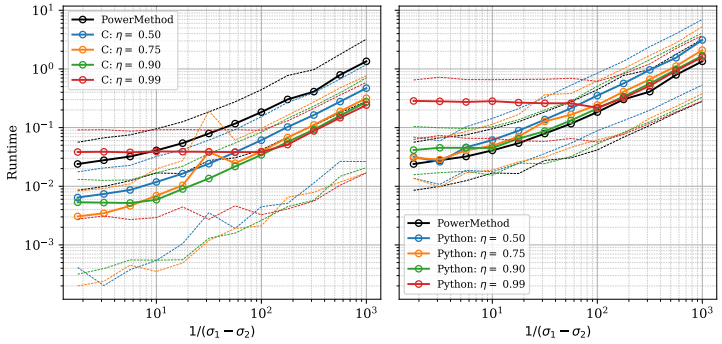
Second, the runtime of the implemented `GDSVD` is competitive if not faster than the `Power Method` which reaffirms the benefits of the gradient-descent approach for solving k-SVD. This can be concluded from the results listed in Table 3 across various datasets.

Third, the scaling of run time for `GDSVD` scales linearly with the inverse of the gap, $\sigma_1 - \sigma_2$ as suggested by theoretical results. The Figure 2 provides evidence for this.

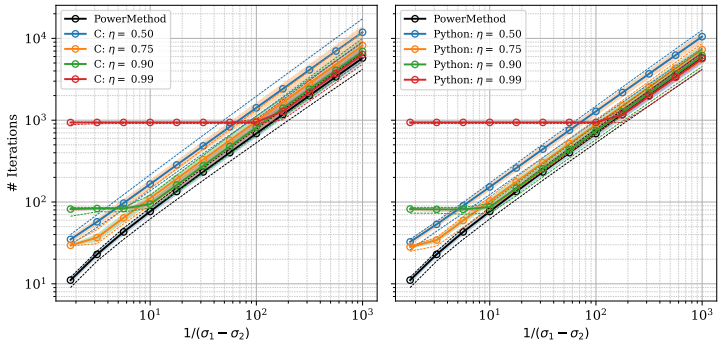
Fourth, the scaling of run time for accelerated version of `GDSVD` scales as square-root of the inverse of the gap, $\sigma_1 - \sigma_2$ as suggested by theoretical results. This confirms the utility of acceleration and ability for optimization based methods to achieve faster scaling inline with other methods. It is worth noting that the `GDSVD(Nesterov)` is relatively better and more robust compared to `GDSVD(Polyak)`. See Figure 1b where this is clearly demonstrated. It is worth noting that while theoretical result in Theorem 2.3 only shows local convergence, the experimental results suggest that `GDSVD(Nesterov)` enjoys global convergence with improved performance.

Table 3: All the considered methods were tested on the rank-$\log(n)$ settings varying $n \in \{50, 75, 100, 200, \ldots, 1000\}$, as well as on the real-word datasets. The reported values correspond to the mean and standard deviation across different values of n (resp. number of datasets) and number of threads used for parallelization of the methods for the rank-$\log(n)$ (resp. real-world setting) settings. Here we report the runtime and the k-SVD recovery errors measured as $\epsilon_\Sigma = \|\Sigma - \hat{\Sigma}\|$ and $\epsilon_{U,V} = \max(\|UU^\top - \hat{U}\hat{U}^\top\|_F, \|VV^\top - \hat{V}\hat{V}^\top\|_F)$. For the rank-$\log(n)$ matrices, $k = \lfloor \log(n) \rfloor$. For the real-world matrices $k = 10$.

Datasets	**Algorithms**	`GDSVD` ($\eta = 0.5$)	`Power Method`
	Evaluations	mean (std)	mean (std)
Exponential	Runtime	1.884 (4.626)	3.598 (6.185)
	ϵ_Σ	1.9x 10^{-13}(5.4x10^{-13})	1.7x 10^{-16}(1.6x10^{-16})
	$\epsilon_{U,V}$	2.8x10^{-6} (9.0x10^{-6})	3.4x10^{-6} (9.2x10^{-6})
Polynomial	Runtime	5.400 (13.42)	6.433 (12.61)
	ϵ_Σ	2.9x10^{-16} (1.1x10^{-16})	2.3x10^{-16} (7.3x10^{-17})
	$\epsilon_{U,V}$	6.1x10^{-08} (9.4x10^{-09})	1.9x10^{-08} (4.5x10^{-09})
Linear	Runtime	10.23 (23.38)	10.56 (21.01)
	ϵ_Σ	1.4x10^{-14} (2.0x10^{-14})	4.5x10^{-15} (5.1x10^{-15})
	$\epsilon_{U,V}$	6.2x10^{-08} (1.1x$10^{-0.8}$)	2.5x10^{-08} (5.6x10^{-09})
Real-world	Runtime	227.2 (509.6)	227.2 (509.6)
	ϵ_Σ	1.8x10^{-05} (3.1x10^{-05})	1.8x10^{-05} (3.1x10^{-05})
	$\epsilon_{U,V}$	2.1x10^{-07} (5.3x10^{-08})	1.0x10^{-07} (4.0x10^{-08})



(a) *Gap vs. Runtime.*



(b) *Gap vs. Number of iterations.*

Figure 2: Here, we illustrate the runtime and convergence performance of `GDSVD` in both `C` and `Python` as we vary the gap $\sigma_1 - \sigma_2$ in the rank-2 setting. The implementations of `GDSVD` with different values of $\eta \in (0, 1)$ were compared with `Power Method`. The curves were averaged over values of $n \in \{50, 100, 200, \ldots, 1000\}$, and the shaded areas correspond to the standard deviations. The doted plots correspond to the lowest and uppermost performance over different values of n.