Chapter 9

In this chapter, we will consider the case of low-rank recovery/learning. This case is similar to that of Chapter 8, where we talked about sparsity constraints: pure low-rank constraints are non-convex in nature. The problem is very similar, where the sparsity of the vector is analogous to the low-rankness of the matrix. Despite the similarities between the two cases, the low-rank case will reveal a different approach to handle such constraints: that of matrix factorization. Overall, in this chapter, we will provide motivation, background and solutions of a low rank optimization problem, that of matrix sensing, which is simple enough to allow us to derive rigorous guarantees and obtain intuition when and why such methods work in practice.

Low rankness, Matrix sensing

Motivation: Quantum state Tomography. Quantum tomography is one of the main procedures to identify the nature of imperfections and deviations in quantum processing unit (QPU) implementation [66,67]. Generally, quantum tomography is composed of two main parts: $i)$ measuring the quantum system, and $ii)$ analyzing the measurement data to obtain an estimation of the density matrix (in the case of state tomography [66]), or of the quantum process (in the case of process tomography [68]). In this chapter, we focus on the case of state tomography.

As the number of free parameters that define quantum states and processes scale exponentially with the number of subsystems, generally quantum tomography is a non-scalable protocol [69]. In particular, quantum state tomography (QST) suffers from two bottlenecks related to its two main parts. The first concerns with the large data one needs to collect to perform tomography; the second concerns with numerically searching in an exponentially large space for a density matrix that is consistent with the data.

Put simply, in QST, the goal is to test whether the output of a quantum circuit (which implements a quantum algorithm, a quantum simulation, etc) in quantum computer is what we expect. While this argument reads weird (i.e., if we know what to expect, why do we run the quantum algorithm in the first place?), QST is a verification tool: it is used in cases where we know the answer to the problem, and we measure the system to see how far we are from that answer (due to inconsistencies, errors in the quantum implementation, etc). Overall, a quantum computer is a non-deterministic machine, where we do not know the final state exactly, unless we measure it (this is where Schröedinger's cat come into the picture!). In QST, we only have measurements of final state (we will define it shortly), and we cannot see any intermediate state of the procedure without ruining the whole process: taking observation in quantum information sciences means that we "destroy" any quantum process followed up to this point (i.e., we cannot "take a look" and then ask the system to continue its process). Thus, in QST, the procedure is to prepare the system to output a state that we expect: if we perform the steps "correctly", w.h.p. we measure parts of the anticipated state with some added noise; if we can repeat the measurement many times, we keep the data, and we try to inverse the procedure to get the value of the state we expect as an output. This way we can measure how errors add and propagate in this implementation of a quantum system, thus leading to a verification tool.

Setup of QST. The setup we consider here is that of an q-qubit state, under the prior assumption that the state is close to a pure state, and thus its density matrix is of low-rank. This assumption is justified by state-of-the-art experiments, where our aim is to manipulate the pure states by unitary maps. From a theoretical perspective, the low-rank assumption means that we can use compressed sensing techniques, which allow the recovery of the density matrix from relatively few measurement data [70]. As we show below, this is similar to a least-square problem, where we want to measure how close the output of the quantum machine is to the ground-truth matrix.

A quantum state can be described by a density matrix (i.e., the ground-truth matrix) $X^\star \in \mathbb{R}^{2^q \times 2^q}$; this matrix represents the state that the quantum system is in, also called q-qubit state. The measurements of the state are the expected values of q-qubit Pauli's observables, which are represented as matrices $A_i \in \mathbb{C}^{2^q \times 2^q}$; *pay attention that we are working on the complex plane.* Then, based on the above, we obtain a measurement vector $y_i \in \mathbb{R}^m$ which follows the next rule:[15]

$$y_i = \langle A_i, X^\star \rangle + e_i = \text{tr}(A_i \cdot X^\star) + e_i, \qquad , i = 1, ...m,$$

for some error noise term, $e_i \in \mathbb{R}$. Here, for this particular problem case, A_i's are Kronecker products of Pauli operators.[16] In particular, they take the form:

$$A_i = \sigma_{i_1} \otimes \sigma_{i_2} \otimes \dots \otimes \sigma_{i_q},$$

where $\sigma_{i_j} \in \sigma_{x,y,z,I}$ are selected randomly and the matrices $\sigma_{x,y,z,I}$ are:

$$\sigma_I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$
$$\sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Classical quantum state tomography is like solving linear equations; with no prior knowledge of $X^\star$ about the low-rankness, if we have a $O(2^q \cdot 2^q) = O(4^q)$ observations y_i measurements, it is possible to reverse the procedure and recover an approximation of $X^\star$ from y_i's and knowing the used A_i Pauli matrices. When these notes where written, the current biggest quantum computer in the world was using $q = 53$; currently, we are even in the $q = 127$ case. This makes the size of $X^\star$, very very large! Further, asking for $O(4^q)$ measurements is simply impossible.

Thus, if we do not assume anything about the state $X^\star$, the number of measurements needed is so large ($2^{53} \times 2^{53}$ - do the math!) Similar to measurements in the sparsity problem, if we know the state is low-rank, one can hope for less than $O(4^q)$ measurements. For instance, if we have a rank-1 matrix for $X^\star$, we only need to know the vector of length 2^q and take the outer vector with itself to get the rank-1 matrix $X^\star \in \mathbb{R}^{2^q \times 2^q}$, instead of knowing the whole matrix that have $2^q \cdot 2^q$ measurements. Further, we might know additional information about $X^\star$ (e.g., that is a positive semi-definite matrix). Overall, quantum states that can be well-approximated with low-rank density matrices $X^\star$ are called *pure quantum states*; these are states that might not be the most interesting ones in the quantum community, but they are considered as a first step before going into more mixed states. In practice, even if we assume $X^\star$ is rank-1, it will be heavily contaminated

[15] A lot of details are "glossed out" at this stage, and this formulation satisfies the purpose of this chapter.

[16] For two matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$, the Kronecker product $A \otimes B$ is a $pm \times qn$ block matrix such that:

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}$$

.

with noise + other phenomena appear that increase the rank in practice. However, for the purposes of this chapter, we will assume that $X^\star$ is low-rank (of rank r, for known r). And, similar to the sparsity case, we will see that for rank-r matrices of size $p \times p$ matrix we need $O(pr)$ measurements, instead of $O(p^2)$.

Related work on QST. There have been various approaches over the years to improve the scalability of QST, as compared to full QST [71–73]. Focusing on the data collection bottleneck, to reduce the resources required, prior information about the unknown quantum state is often assumed. For example, in compressed sensing QST [69, 74], it is assumed that the density matrix of the system is low-rank. In neural network QST [75–77], one assumes real and positive wave-functions, which occupy a restricted place in the landscape of quantum states. Extensions of neural networks to complex wave-functions, or the ability to represent density matrices of mixed states, have been further considered in the literature, after proper reparameterization of the Restricted Boltzmann machines [75]. The prior information considered in these cases is that they are characterized by structured quantum states, which is the reason for the very high performances of neural network QST [75].[17] Similarly, in matrix-product-state tomography [78, 79], one assumes that the state-to-be-estimated can be represented with low bond-dimension matrix-product state.

Focusing on the computational bottleneck, several works introduce sophisticated numerical methods to improve the efficiency of QST. Particularly, variations of gradient descent convex solvers—e.g., [80–83]—are time-efficient in idealized (synthetic) scenarios [83], and only after a proper distributed system design [84]. The problem is that achieving such results seems to require utilizing special-purpose hardware (like GPUs). Thus, going beyond current capabilities requires novel methods that efficiently search in the space of density matrices under more realistic scenarios. Importantly, such numerical methods should come with guarantees on their performance and convergence.

Indeed, by now, compressed sensing QST is widely used for estimating highly-pure quantum states, e.g., [69, 85–87]. However, compressed sensing QST usually relies on convex optimization for the estimation part [74]; this limits the applicability to relatively small system sizes [69]. On the other hand, non-convex optimization can preform much faster than its convex counterpart [88]. Although non-convex optimization typically lacks convergence guarantees, it was recently shown that one can formulate compressed sensing QST as a non-convex problem and solve it with rigorous convergence guarantees (under certain but generic conditions), allowing state estimation of larger system sizes [88].

Matrix sensing. QST is an instance of what is called *matrix sensing.* Formally, the matrix sensing problem is as follows: Given a measurement mechanism $\mathcal{A} : \mathbb{R}^{p\times p} \rightarrow \mathbb{R}^m$, matrix sensing is seeking a solution to the following optimization problem:[18]

$$\begin{array}{ll} \underset{X\in\mathbb{R}^{p\times p}}{\text{minimize}} & \frac{1}{2}\cdot\sum_{i=0}^{m-1}\left(y_i - \left(\mathcal{A}(X)\right)_i\right)^2 \\ \text{subject to} & \text{rank}(X) \leq r. \end{array}$$

Here, the linear measurements are y_i, $i \in [0, m-1]$, which is assumed to be generated by the model $y_i = \left(\mathcal{A}(X^\star)\right)_i := \langle A_i, X^\star\rangle$, where $X^\star \in \mathbb{R}^{p\times p}$. I.e., $\left(\mathcal{A}(\cdot)\right)_i = \langle A_i, \cdot\rangle$. Without constraints or a structural assumption on $\mathbf{X}^\star$, the problem is under-determined with infinite solutions. However, given a rank r matrix and a sufficiently large number of measurements, it is possible that a unique solution exists that may be found via optimization.

Restricted Isometry Property. Similar to the sparsity case, a pivotal assumption is that the linear map $\mathcal{A}$ satisfies the *restricted isometry property* for low rank matrices:

Definition 32. (Restricted Isometry Property (RIP) [89]) *A linear operator $\mathcal{A} : \mathbb{C}^{d\times d} \rightarrow \mathbb{R}^m$ satisfies the RIP on rank-r matrices, with parameter $\delta_r \in (0, 1)$, if the following holds for any rank-r matrix $X \in \mathbb{C}^{d\times d}$, with high probability:*

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

Such maps (almost) preserve the Frobenius norm of low-rank matrices, and, as an extension, of low-rank Hermitian matrices. The intuition behind RIP is that $\mathcal{A}(\cdot)$ behaves as almost a bijection between the subspaces $\mathbb{C}^{d\times d}$ and $\mathbb{R}^m$, when we focus on low rank matrices.

Algorithmic solutions for matrix sensing. Similar to the sparse case, to solve the matrix sensing problem (for now consider QST formulation without the PSD and trace constraint), we have several approaches, split into the convex and nonconvex camps:

i) Through convexification: Nuclear Norm Minimization. Similar to the ℓ_0-pseudonorm case where ℓ_1-norm is the tightest convex relaxation, the question is what is the tightest convex relaxation of the set for matrices $A \in \mathbb{R}^{p\times p}$:

$$\{A : \text{rank}(A) = 1, \|A\|_F = 1\}?$$

The answer to this question is that of *nuclear norm*:

$$\|A\|_* = \sum_i \sigma_i(A).$$

I.e., by bounding the nuclear norm of the solution, we implicitly enforce a "sparsity" constraint on the set of singular values of the matrix A. More strict nuclear norm bounds lead to "sparser" set of singular values, forcing some of them to be zero (*remember, the singular values cannot be negative, so the lowest point they can get is that of zero*), which means that the matrix starts becoming more and more rank-deficient (*the more singular values of a matrix are zero, the more the rank of that matrix decreases*).

Given this intuition, one can throw away the rank constraint in the matrix sensing scenario and substitute that with the nuclear norm constraint, as follows:

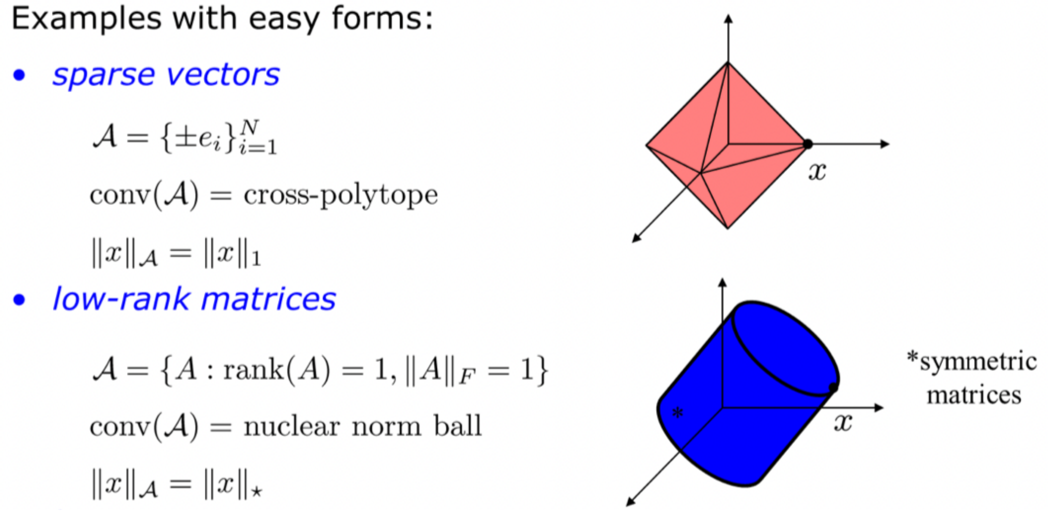
$$\begin{array}{ll} \underset{X\in\mathbb{R}^{p\times p}}{\text{minimize}} & \frac{1}{2}\cdot\sum_{i=0}^{m-1}\left(y_i - \left(\mathcal{A}(X)\right)_i\right)^2 \\ \text{subject to} & \|X\|_* \leq \lambda, \end{array}$$

for some $\lambda > 0$ as a regularizer parameter.

[17] [75] considers also the case of a completely unstructured case and test the limitation of this technique, which does not perform as expected due to lack of structure.

[18] To be precise, in QST, we have the PSD version of the matrix sensing problem with additional trace constraints:

$$\begin{array}{ll} \underset{X\in\mathbb{R}^{p\times p}}{\text{minimize}} & \frac{1}{2}\cdot\sum_{i=0}^{m-1}\left(y_i - \left(\mathcal{A}(X)\right)_i\right)^2 \\ \text{subject to} & X \succeq 0, \text{rank}(X) \leq r,\ \text{tr}(X) \leq 1. \end{array}$$



Examples with easy forms:

- *sparse vectors*

$\mathcal{A} = \{\pm e_i\}_{i=1}^N$

$\text{conv}(\mathcal{A}) = \text{cross-polytope}$

$\|x\|_{\mathcal{A}} = \|x\|_1$

- *low-rank matrices*

$\mathcal{A} = \{A : \text{rank}(A) = 1, \|A\|_F = 1\}$

$\text{conv}(\mathcal{A}) = \text{nuclear norm ball}$

$\|x\|_{\mathcal{A}} = \|x\|_\star$

*symmetric matrices

Fig. 48. Illustration of some convex relaxations of known non-convex sets. *The notation $\mathcal{A}$ should not be confused with the linear map in this chapter.* At the top, the set of unit-norm vectors that "live" on the coordinate axes can be "convexified" into a convex hull that matches the ℓ_1-norm with unit norm. At the bottom, the set of rank-1 matrices A with unit Frobenius norm $\|A\|_F = 1$ can be "convexified" into a convex hull that matches the nuclear-norm of matrices of the same dimensions with unit norm.

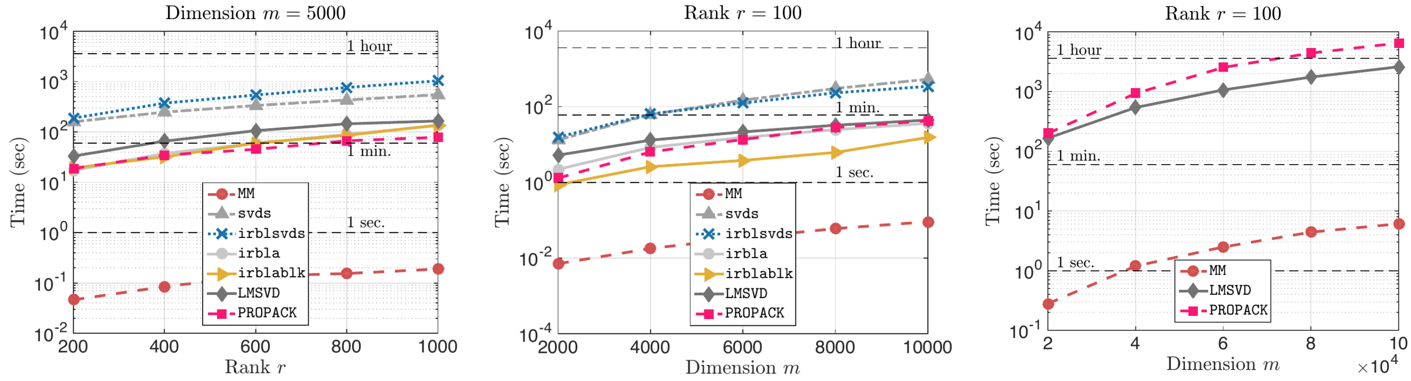


Fig. 49. Comparison between matrix multiplication (MM), i.e. $X.U$ where $X \in \mathbb{R}^{m \times m}$ and $U \in \mathbb{R}^{m \times r}$ and SVD (all methods of truncated SVD) of X. In both the scenarios (varying rank r and varying dimension m), there is a big gap in SVD and matrix multiplication calculation. Theoretically the complexity is same for both operation but there are many matrix multiplications involved in SVD.

Given this problem formulation, a natural way to solve the problem is via convex projected gradient descent. In each iteration, the gradient descent step is followed by projection onto the bounded nuclear norm constraint set. In math terms, the update looks like:

$$X_{t+1} = \Pi_{\|\cdot\|_* \leq \lambda} \left(X_t - \eta \nabla f\left(X_t\right) \right),$$

where $f(X) := \frac{1}{2}\|y - \mathcal{A}(X)\|_2^2$, and $\Pi_{\|\cdot\|_* \leq \lambda}(\cdot)$ is the result of the optimization problem:

$$\begin{aligned} \Pi_{\|\cdot\|_* \leq \lambda}(Z) = & \underset{X \in \mathbb{R}^{p \times p}}{\text{argmin}} \quad \tfrac{1}{2} \cdot \|X - Z\|_F^2 \\ & \text{subject to} \quad \|X\|_* \leq \lambda, \end{aligned}$$

As we have already discussed in previous chapters, the projection onto the set of bounded nuclear norm is calculated in closed form via the singular value decomposition (or recursively using the power iteration method). I.e., in order to compute the projection, one needs to first compute the $O(p^3)$ SVD in order to find the singular values; then, these are "projected" and clipped so that their summation is bounded by λ; finally, the remaining updated singular values (along with the corresponding singular vectors) gives us back the answer to this projection step. Key note here is that the nuclear norm projection *does not guarantee low-rankness:* we hope that by successively projecting the singular values over many iterations, several will be supressed and stay zero, over the course of the algorithm.

Overall, the above algorithm is convex, and comes with nice theoretical guarantees. However, by looking at Figure 49, it is clear that the complexity of the nuclear norm is an expensive operation, that scales cubicly with the size of the problem p. As p increases in modern machine learning and optimization applications, this is not a viable solution for efficient solutions. This leads to the other alternative below.

ii) By keeping the rank-constraint: Iterative Hard Thresholding. By keeping the rank constraint in the optimization description, we end up with a non-convex problem, similar to the sparse problem in the previous chapter. The projection is now on the rank-constraint instead of the nuclear norm: Using the same notation as in the previous chapter, the rank-r

hard-thresholding projection is defined as:

$$\begin{aligned} H_r(Z) = &\operatorname*{argmin}_{X\in\mathbb{R}^{p\times p}} \quad \tfrac{1}{2}\cdot\|X-Z\|_F^2 \\ &\text{subject to} \quad \text{rank}(X)\le r. \end{aligned}$$

In fact, the above problem has a name: it is the well-known Eckart-Young-Minsky theorem that proves that the best rank-r approximation of a given matrix Z (with respect to its Frobenius norm distance) is provided by the rank-r SVD approximation of the matrix Z (also known as the *truncated SVD*). But, what is the computational complexity of rank-r truncated SVD? Based on arguments on power iteration, one can argue that the complexity is of the order $O(rp^2)$, where r is usually independent of p. This favorably compares to the nuclear norm minimization formulation, where the projection there has $O(p^3)$ complexity.

Given the above, the definition of the IHT matrix for low-rank matrix sensing problems is straightforward:

$$X_{t+1} = H_r\left(X_t - \eta\nabla f\left(X_t\right)\right),$$

where $f(X) := \frac{1}{2}\|y-\mathcal{A}(X)\|_2^2$. Theorems on matrix version of IHT, step size selections and adaptive schedules in previous lectures still hold here.

Figure 49 highlights the different price we pay, even if we do truncated SVD. It is obvious that even for this case, alternatives should be devised to get a faster algorithm, if possible.

Low-rank matrices are matrices that are factorized. From now on, we generalize our discussion to include more generic cases. In particular, we study matrix problems of the form:

$$\operatorname*{minimize}_{X\in\mathbb{R}^{p\times p}} \quad f(X),$$

where the minimizer $X^\star \in \mathbb{R}^{p\times p}$ is rank-$r^\star$ ($r^\star \le p$), or *nearly* low rank; *i.e.*, $\|X^\star - X^\star_{r^\star}\|_F$ is sufficiently small, for $X^\star_{r^\star}$ being the best rank-$r^\star$ approximation of $X^\star$. In our discussions, f is a differentiable convex function. Further assumptions on f will be described later in the text.

Specific instances of the above problem appear in several applications in diverse research fields. A non-exhaustive list includes factorization-based recommender systems [42, 90–95], multi-label classification tasks [96–101], dimensionality reduction techniques [102–107], density matrix estimation of quantum systems [69, 74, 108], phase retrieval applications [109, 110], sensor localization [111, 112] and protein clustering [113] tasks, image processing problems [114], as well as applications in system theory [115]. Thus, it is critical to devise user-friendly, efficient and provable algorithms, taking into consideration the (near) low-rank structure of $X^\star$.

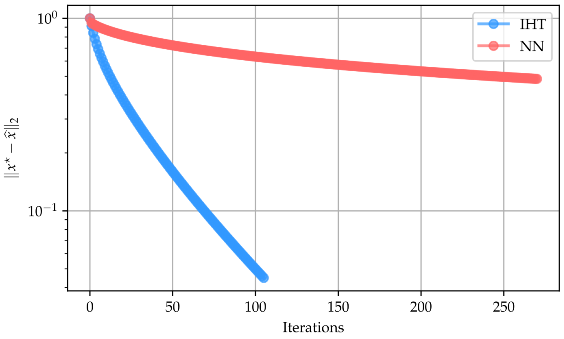


Fig. 50. Demo on Iterative Hard Thresholding and Nuclear norm minimization. $p = 128$, $r = 2$. Refer to pyhton notebook

In general, imposing a low-rank constraint could result in an NP-hard problem. However, the above minimization with a rank-constraint can be solved in polynomial-time for applications where f has specific structure. A prime example is the matrix sensing problem [89, 94, 116], There, $X^\star$ can be recovered in polynomial time, by solving the above problem with a rank-constraint [58, 117–121], or by solving its convex nuclear-norm relaxation, as in [54, 122–126].

Although algorithms operating on X space have attractive convergence rates, they simultaneously manipulate $p \times p$ variables in X. This is computationally expensive in the high-dimensional regime: typically, each iteration requires computing at least the top-r singular value/vectors of matrices. As p scale, the computational demands per iteration are prohibitive.

Optimizing over factors. In this paper, we follow a different path: a rank-r matrix $X \in \mathbb{R}^{p\times p}$ can be written as a product of two matrices $UV^\top$, where $U \in \mathbb{R}^{p\times r}$ and $V \in \mathbb{R}^{p\times r}$. Based on this, we are interested in solving our problem at hand via the $UV^\top$ parametrization:

$$\operatorname*{minimize}_{U\in\mathbb{R}^{p\times r}, V\in\mathbb{R}^{p\times r}} \quad f(UV^\top) \qquad \text{where } r \le \text{rank}(X^\star) \le p.$$

Note that characterizations of the above and of the original problem are equivalent in the case $\text{rank}(X^\star) = r$.[19] Observe that such parameterization leads to a very specific kind of non-convexity in f. Proving convergence for these settings becomes a harder task, due to the bi-linearity of the variable space.

Motivation. When r is much smaller than p, $U \in \mathbb{R}^{p\times r}$ and $V \in \mathbb{R}^{p\times r}$ contain far fewer variables than $X = UV^\top$. Thus, by construction, such parametrization makes it easier to update and store the iterates U, V.

Key is that $UV^\top$ reformulation automatically encodes the rank constraint. Approaches working on X require computing a truncated SVD[20] per iteration, which can get cumbersome in large-scale settings. In stark contrast, working with $f(UV^\top)$ replaces singular value computations with matrix-matrix multiplication operations. This turns out to be a more practical and realistic option, when the dimension of the problem is large. *E.g.*, matrix-matrix multiplications could be parallelized much easier than SVD computations.

Rank-1 Matrix Approximation Through Rank-1 PCA. To understand the above, we will consider a simpler rank-1 case. The PCA problem is not an algorithm, because we need SVD to solve it. To solve the SVD problem, we need an algorithm such as the power iteration. So the PCA can be recast to the following case.

Consider a simpler objective of matrix factorization where we are minimizing the following objective.

$$\min_{X\in\mathbb{R}^m, W\in\mathbb{R}^n} \left\|M - XW^\top\right\|_F^2, \quad M \in \mathbb{R}^{m\times n}$$

[19] By equivalent, we mean that the set of global minima in one contains that of the other. It remains an open question though whether the reformulation introduces spurious local minima in the factored space for *the majority of f cases*.

[20] This holds in the best scenario; in the convex case, where the rank constraint is "relaxed" by the nuclear norm, the projection onto the nuclear-norm ball often requires a full SVD calculation.

Where M is rank-1 matrix, $\|X\|_2 = \|W\|_2 = 1$.
This is equivalent to

$$\min_{Y\in\mathbb{R}^{m\times n}} \|M-Y\|_F^2, \quad rank(Y)=1$$

When we connect with matrix sensing, the objective can also be vectorized:

$$\left\|M - XW^\top\right\|_F^2 = \left\|\text{vec}(M) - \text{vec}\left(XW^\top\right)\right\|_2^2$$

$$= |Y - \mathcal{A}(X)\|_2^2$$

Where, $Y = \text{vec}(M)$ and $\mathcal{A}$ is like an identity mapping that takes a matrix and makes it to a vector to compute the difference. Hence it is a matrix sensing formulation with rank-1 approximation.
Then consider the SVD decomposition,

$$M = \sum_{i=1}^{\min\{m,n\}} \sigma_i u_i v_i^\top$$

Where $\|u_i\| = \|v_i\| = 1, \sigma_1 \geq \sigma_2 ... \geq \sigma_{\min\{m,n\}} \geq 0$, which we assume the matrix is full-rank. Also the singular vectors are orthogonal unless they have the same index, so $u_i^\top u_j = 0, v_i^\top v_j = 0$ for $i \neq j$.
To simplify this problem

$$\begin{aligned}\left\|M - XW^\top\right\|_F^2 &= \|M\|_F^2 - 2X^\top MW + \left\|XW^\top\right\|_F^2 \\ &= \|M\|_F^2 - 2X^\top MW + \|X\|_2^2 \cdot \|W\|_2^2\end{aligned}$$

Hence our objective is:

$$f(x) = \min_{X\in\mathbb{R}^m, W\in\mathbb{R}^n} -2X^\top MW + \|X\|_2^2 \cdot \|W\|_2^2$$

Is a minimization over X and W. So we can do alternate minimization (fix one and minimize over other).
Assuming we know X optimal and we only minimize over W. Let

$$f(x) = \min_{W\in\mathbb{R}^n} -2X^\top MW + \|X\|_2^2 \cdot \|W\|_2^2,$$

Where $f(x)$ is convex, $-2X^\top MW$ is linear term and $\|X\|_2^2 \cdot \|W\|_2^2$ is quadratic, thus by convexity

$$\nabla_w = 0 \mathbb{R}ightarrow - 2M^T X + 2\|X\|_2^2 \cdot W = 0$$

$$\mathbb{R}ightarrow W = \frac{M^T X}{\|X\|_2^2}$$

Substituting W in original problem.

$$f(X,W) = -\frac{2X^\top MM^\top X}{\|X\|_2^2} + \|X\|_2^2 \cdot \frac{X^\top MM^\top X}{\|X\|_2^L}$$

$$f(X,W) = f(x) = -2X^\top MW + \|X\|_2^2 \|W\|_2^2$$

$$f(X,W) = -\frac{X^\top MM^\top X}{\|X\|_2^2}$$

Thus the original problem is equivalent to:

$$\min_{X,W\in\mathbb{R}^m} f(X) := -\frac{X^\top MM^\top X}{\|X\|_2^2}$$

Where we have a close form solution for W.

Length of X does not matter, only its direction matters. To see this, define temporarily $Y = \frac{X}{\|X\|_2^2}$. Then

$$\min_{x\in\mathbb{R}^m} f(x) \equiv \min_{y\in\mathbb{R}^m, \|y\|=1} -y^\top MM^\top y$$

Consider the PCA problem for rank-1, which is usually defined as

$$\max X^\top \Sigma X, \|X\|_2 = 1$$

Given the co-variance matrix Σ of the data, we want to find the direction of the maximum variance(i.e.to find the normalized vector that correlates with the direction that best approximate the data). So putting things together, we have

$$\min(-y^\top MM^\top y) = \max y^\top MM^\top y = y^\top \Sigma y$$

This problem is non-convex now, where the objective can be perceived as finding the max eigenvalue of $MM^\top$.

How the objective looks like?: To maximize on a bowl (it not a perfect bowl because of different eigenvalues) with a constraint of a ring on it. The solution is unique because we assume no two eigenvalue are same.

We then want to apply gradient descent on x instead of using power iteration.
Via the inner product expression:

$$\langle x, u_1\rangle = \cos(\theta) \cdot \|u_1\| \cdot \|x\|_2$$

Since θ depends on x, and $\|u_1\|_2 = 1$, we have:

$$\theta(x) = \cos^{-1}\left(\frac{1}{\|x\|_2} < x, u_1 >\right)$$

To solve this optimisation problem we will do gradient descent (can't do SVD/power-iteration as we are trying to avoid it).

$$X_{t+1} = X_t - \eta \cdot \nabla f\left(X_t\right)$$

Applying quotient rule

$$\begin{aligned}\nabla f\left(X_t\right) = &\frac{1}{\|X\|_2^4} \cdot \Big[\nabla_X\left(-X^\top MM^\top X\right)\|X\|_2^2 \\ &+ X^\top MM^\top X \cdot \nabla_X \|X\|_2^2\Big] \\ &= \frac{1}{\|X\|_2^4}\left[-2\|X\|_2^2 \cdot MM^\top X + 2\left(X^\top MM^\top X\right) \cdot X\right] \\ &= \frac{2}{\|X\|_2^4}\left[\left(X^\top MM^\top X\right) X - \|X\|_2^2 \cdot MM^\top X\right]\end{aligned}$$

We can write M as

$$M = \sum_{i=1}^{\min\{m,n\}} \sigma_i \cdot U_i V_i^T$$

$$\sigma_1 > \sigma_2 \geqslant \ldots \geqslant 0$$

The solution corresponds to the biggest/first singular value.

Key observation for gradient descent on PCA is that if $\langle X_t, u_1\rangle = 0$ (e.g. $x_t = u_2$), then $\langle X_{t+1}, u_1\rangle = 0$, implies we are going orthogonal to the eigen vector u_1. To prove it:

$$\begin{aligned}&\langle X_{t+1}, u_1\rangle = \left\langle X_t - \eta \nabla f\left(X_t\right), u_1\right\rangle \\ &= \langle x_t, u_1\rangle - \eta \langle \nabla f\left(X_t\right)\rangle = -\eta \left\langle \nabla f\left(X_t\right), u_1\right\rangle\end{aligned}$$

$$
\begin{aligned}
\langle \nabla f\left(X_t\right), u_1 \rangle &= \frac{2}{\|X_t\|_2^4}\Big[\Big(X_t^\top MM^\top X_t\Big)X_t^\top u_1 - \\
&\quad \|X_t\|_2^2 X_t^\top MM^\top u_1\Big] \\
&= -\frac{2}{\|X_t\|_2^4}\cdot\|X_t\|_2^2 \quad X_t^\top MM^\top u_1 \\
&= -\frac{2}{\|X_t\|_2^2}\cdot X_t^\top \cdot \left(\sum_i \sigma_i^2 u_i u_i^\top\right) u_1 \\
&= -\frac{2}{\|X_t\|_2^2}\cdot X_t^\top \cdot (\sigma_1^2 u_1 u_1^\top) u_1 \\
&= -\frac{2}{\|X_t\|_2^2}\sigma_1^2 X_t^\top u_1 = 0
\end{aligned}
$$

i.e.,

$$\langle X_{t+1}, u_1\rangle = -\frac{2}{\|X_t\|_2^2}\sigma_1^2 X_t^\top u_1 = 0$$

Remark:

i) If X_t is orthogonal to u_1 then X_{t+1} is also orthogonal to u_1. This is a no improvement state.
ii) This further means that if we start from a point such that $\langle X_0, u_1\rangle = 0$, we fail to recover u_1. We will be trapped in saddle point here.
iii) However, maybe there is a hope, if we start from any point not orthogonal to U_1. This is like selecting a point not from the vector span of $u_i, i \neq 1$. A randomly selected point $X_0 \in \mathbb{R}^m$ almost surely has non-zero component on the span of u_1.

We want to study the behavior of the potential function. Define a potential function

$$\psi_{t+1} = 1 - \frac{\langle X_{t+1}, u_1\rangle^2}{\|X_{t+1}\|_2^2}$$

Intuition: if $\psi_{t+1} \to 0$, x_{t+1} aligns with u_1 and

$$\frac{\langle X_{t+1}, u_1\rangle^2}{\|X_{t+1}\|_2^2} \to 1,$$

which is the optimal thing to achieve for normalized vectors.

We have the following:

$$\|x_{t+1}\|_2^2 = \|x_t - y\nabla f\left(x_t\right)\|_2^2$$

$$= \|X_t\|_2^2 - 2\eta X_i^\top \nabla f\left(X_t\right) + \eta^2\cdot\|\nabla f\left(X_0\right)\|_2^2$$

Observe that:

$$
\begin{aligned}
X_t^\top \nabla f\left(X_t\right) &= \frac{2}{\|X_t\|_2^4}\Big(\Big(X_t^\top MM^\top X_t\Big)\cdot\|X_t\|_2^2 \\
&\quad - \|X_t\|_2^2\Big(X_t^\top MM^\top X_t\Big)\Big) \\
&= 0
\end{aligned}
$$

Hence

$$
\begin{aligned}
\|X_{t+1}\|_2^2 &= \|X_t\|_2^2 - 2\eta X_i^\top \nabla f\left(X_t\right) + \eta^2\cdot\|\nabla f\left(X_0\right)\|_2^2 \\
&= \|X_t\|_2^2 + \eta^2\cdot\|\nabla f\left(X_t\right)\|_2^2
\end{aligned}
$$

Then

$$\Psi_{t+1} = 1 - \frac{\langle X_{t+1}, u_1\rangle^2}{\|X_{t+1}\|_2^2}$$

$$= 1 - \frac{\langle X_t, u_1\rangle^2 - 2\eta\cdot\langle X_t, u_1\rangle\langle\nabla f\left(X_t\right), u_1\rangle + \eta^2.\langle\nabla f\left(X_1\right), u_1\rangle}{\|X_t\|_2^2 + \eta^2\cdot\|\nabla f\left(X_t\right)\|_2^2}$$

We know that

$$\frac{\langle X_t, u_1\rangle^2}{\|X_{t+1}\|_2^2} = \left\langle \frac{X_{t+1}}{\|X_{t+1}\|_2}, u_1\right\rangle^2 = \cos^2\left(\theta\left(X_{t+1}\right)\right)$$

And $1 - \cos^2(\theta(X+1)) = \sin^2\left(\theta\left(X_t+1\right)\right)$.

Using these facts, it turns out that

$$
\begin{aligned}
\sin^2\left(\theta\left(X_{t+1}\right)\right) &\leq \sin^2\theta\left(X_t\right) + \frac{\eta^2}{\|X_t\|_2^2}\cdot\|\nabla f\left(X_t\right)\|_2^2 \\
&\quad + \frac{2\eta}{\|X_t\|_2^2}\cdot\langle X_t, u_1\rangle\langle\nabla f\left(X_t\right), u_1\rangle.
\end{aligned}
$$

1) For the inner product $\langle\nabla f\left(x_t\right), u_1\rangle$ we have:

$$\langle\nabla f\left(X_t\right), u_1\rangle \leqslant -\frac{2}{\|X_t\|_2}\left(\sigma_1^2 - \sigma_2^2\right)\cdot\sin^2\theta\left(X_t\right)\cdot\cos\theta\left(X_t\right) \leq 0$$

2) For $\|\nabla f\left(X_t\right)\|_2^2$ we have:

$$\|\nabla f\left(X_t\right)\|_2^2 \leq \frac{4}{\|X_t\|_2^2}\left(\sigma_1^4 + \sigma_2^4\right)\cdot\sin^2\theta\left(X_t\right)$$

Then we will get:

$$
\begin{aligned}
\sin^2\left(\theta\left(X_{t+1}\right)\right) \leq \sin^2\left(\theta\left(X_t\right)\right)\Big(1 &+ \frac{4\eta^2}{\|X_1\|_2^4}\left(\sigma_1^4 + \sigma_2^4\right) \\
&- \frac{4\eta}{\|X_t\|_2^2}\cdot\left(\sigma_1^2 - \sigma_2^2\right)\cdot\frac{\langle X_t, u_1\rangle^2}{\|X_t\|_2^2}\Big)
\end{aligned}
$$

We will provide local convergence guarantees if given a proper initialization, we get convergence to global minimum.

1) If $\frac{\langle X_t, u_1\rangle^2}{\|X_t\|_2^2} \geq c$, such that $0 \leq c < 1$,
we obtain:

$$
\begin{aligned}
\sin^2\theta(X_{t+1}) \leq \sin^2\theta(X_t)(1 + \frac{4\eta^2}{\|X_t\|_2^4}(\sigma_1^4 + \sigma_2^4) - \\
\frac{4\eta}{\|X_t\|_2^2}(\sigma_2^2 - \sigma_2^2)\cdot c)
\end{aligned}
$$

2) Select $\eta = \frac{c}{2}\cdot\frac{\sigma_1^2 - \sigma_2^2}{\sigma_1^4 + \sigma_2^4}\cdot\|x_t\|_2^2$

Then

$$p = 1 + c^2\cdot\frac{\sigma_1^2 - \sigma_2^2}{\sigma_1^2 + \sigma_2^2} - 2\cdot c^2\cdot\frac{\sigma_1^2 - \sigma_2^2}{\sigma_1^2 + \sigma_2^2} = 1 - c^2\cdot\frac{\sigma_1^2 - \sigma_2^2}{\sigma_1^2 + \sigma_2^2} < 1$$

Thus

$$\sin^2\left(\theta\left(X_{t+1}\right)\right) \leq \rho\sin^2\left(\theta\left(X_t\right)\right)$$

Where $\rho < 1$. We achieve linear convergence $O(\log\frac{1}{\epsilon})$, without using SVD at any step.

Some properties of the proof:
1) Initialization does matter: e.g., for PCA there are initializations that do not lead to convergence.
2) After proper initialization, one can prove convergence to global minimum. Despite this, such convergence results are called local convergence guarantees.
3) Often the theory dictates how to set the step size, in order to obtain convergence. For some cases it is a range of values, in other cases we just rely on a specific step size.
It gives a good motivation that matrix factorization is useful.

Alternate minimization. Back to original problem, we have matrix sensing objective

$$\min_{X \in \mathbb{R}^{p \times p}} \frac{1}{2} \sum_{i=1}^{n} (y_i - \langle A_i, X \rangle)^2$$

With low rank constraint $rank(X) < r$. Instead of that we can represent X as

$$X = UV^T$$

The objective now is constraint free

$$X = \arg \min_{U \in \mathbb{R}^{n \times r}, V \in \mathbb{R}^{p \times r}} \frac{1}{2} \sum_{i=0}^{m-1} (y_i - \langle \mathbf{A}_i, \mathbf{UV^T} \rangle)^2$$

Key differences with PCA: 1) Number of observations are less than number of parameters, 2) Mapping $\mathcal{A}$ is not identity, but satisfies a restricted isometry property.
Now if we not restrict the objective to least squares:

$$X = \arg \min_{U \in \mathbb{R}^{m \times r_r}, V \in \mathbb{R}^{n \times r}} f\left(UV^\top\right)$$

Here Restricted isometry can be substituted by Restricted Strong Convexity.

To solve this perform, we perform alternate minimization (not in true sense, as we are not using updated U_{i+1} from first step in the second step). The method is also called **Factored Gradient descent**.

$$U_{i+1} = U_i - \eta \nabla f\left(U_i V_i^\top\right) \cdot V_i$$

$$V_{i+1} = V_i - \eta \nabla f\left(\left(U_i V_i^\top\right)^\top\right) \cdot U_i$$

Although we have constraint less optimization problem now, factorization brings another problem. Objective is not convex. There are new saddle points, global and local minimas introduced.

$$X^\star = U^\star V^{\star T} = U^\star R \cdot R^T V^{\star T} = \widehat{U}^\star \widehat{V}^{\star T}$$

For all R such that $RR^T = I$
For example, if:

$$f(X) = \frac{1}{2} \cdot \|y - \mathrm{vec}(A \cdot X)\|_2^2$$

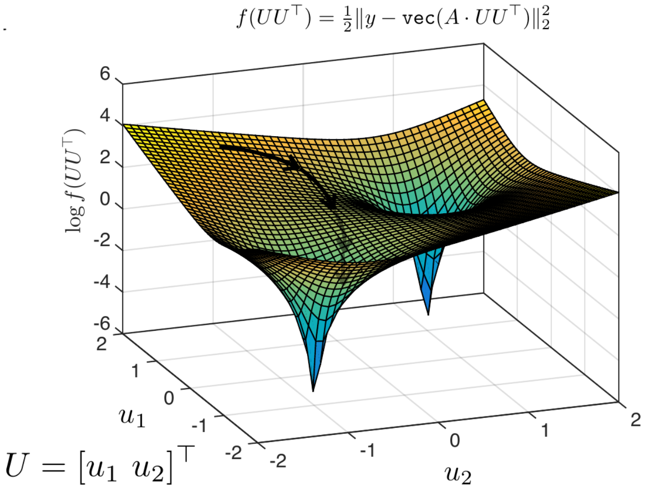
Where

$$X^\star = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

is a unique solution with $r = 1$

$$U^\star = \begin{bmatrix} 1 & 1 \end{bmatrix}^\top \text{ or } [-1 - 1]^\top$$

Multiple factorizations are possible. Hence it ruins convexity.

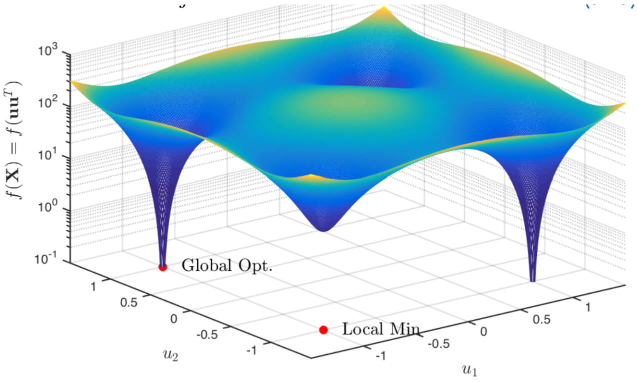


Another example:
Weighted low-rank approximation

$$f\left(uu^\top\right) = \sum_{ij} W_{ij} \cdot \left(X^\star_{ij} - u_i u_j\right)^2$$

where

$$X^\star = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{and} \quad W = \begin{bmatrix} 100 & 1 \\ 1 & 100 \end{bmatrix}$$



As there is non-convexity introduced, proper initialization is the key.

To find some guarantees on convergence: We will start with general recipe of proving convergence.

$$\begin{aligned} \|x_{t+1} - x^\star\|_\#^2 &= \|x_t - \eta \nabla f(x_t) - x^\star\|_\#^2 \\ &= \|x_t - x^\star\|_\#^2 - 2\eta \langle \nabla f(x_t), x_t - x^\star \rangle + \\ &\eta^2 \|\nabla f(x_t)\|_\#^2 \end{aligned}$$

Where # is norm, indicates a general class of distance functions. The geometric intuition of $\langle \nabla f(x_t), x_t - x^\star \rangle$:

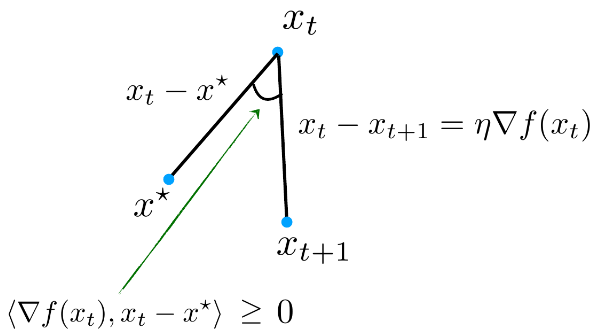

x_t
$x_t - x^\star$
$x_t - x_{t+1} = \eta\nabla f(x_t)$
$x^\star$
x_{t+1}
$\langle\nabla f(x_t), x_t - x^\star\rangle \geq 0$

Fig. 51. Angle between the direction of gradient and correct direction should be less than $\pi/2$

We need the following to hold true to **bound** $\|x_{t+1} - x^\star\|_\#^2$

$$\langle\nabla f(x_t), x_t - x^\star\rangle \geq \alpha\|x_t - x^\star\|_\#^2 + \beta\|\nabla f(x_t)\|_\#^2$$

for sufficient $\alpha, \beta \geq 0$ such that

$$\|x_t - x^\star\|_\#^2 - 2\eta\langle\nabla f(x_t), x_t - x^\star\rangle + \eta^2\|\nabla f(x_t)\|_\#^2$$
$$\leq \|x_t - x^\star\|_\#^2 - c\alpha\eta\|x_t - x^\star\|_\#^2 - \left(c\eta\beta - \eta^2\right)\|\nabla f(x_t)\|_\#^2$$

Now this has some connections with the convex optimization problem we have seen so far.

$$\langle\nabla f(x) - \nabla f(y), x - y\rangle \geq \frac{\mu L}{\mu + L}\|x - y\|_2^2 + \frac{1}{\mu + L}\|\nabla f(x) - \nabla f(y)\|_2^2$$

If $y = X^\star$ and since $\nabla f(X^\star) = 0$

$$\langle\nabla f(x), x - X^\star\rangle \geq \frac{\mu L}{\mu + L}\|x - X^\star\|_2^2 + \frac{1}{\mu + L}\|\nabla f(x)\|_2^2$$

It encourages that our approach for get a bound is correct.

For simplicity now consider X to be positive semi definite. Hence $X = UU^\top$.
Define a **distance metric**, where the distance between any arbitrary matrix U and U^*, DIST is defined as:

$$\mathrm{DIST}(U, U^*) = \min_{R:R\in O_r}\|U - U^*R\|_F$$

O is the set of $r \times r$ orthonormal matrices R, such that $R^TR = I$. R is also called a rotational matrix since $UU^\top = URR^\top U^\top$. There can be infinite U^* but we need an optimal U where distance is one with the closest U^* upto one rotation R. The DIST will help us getting a good initial point.
We also know

$$U_{t+1} = U_t - \eta\nabla f\left(U_tU_t^\top\right)\cdot U_t = U_t - \eta\nabla f(X_t)\cdot U_t$$

Then we have

$$\begin{aligned}
\mathrm{DIST}(U_{t+1}, U^*)^2 &= \min_{R\in O_r}\|U_{t+1} - U^*R\|_F^2 \\
&\leqslant \|U_{t+1} - U^*R_t\|_F^2 \\
&= \|U_{t+1} - U_t + U_t - U^*R_t\|_F^2 \\
&= \|U_{t+1} - U_t\|_F^2 + \|U_t - U^*R_t\|_F^2 \\
&+ 2\langle U_{t+1} - U_t, U_t - U^*R_t\rangle \\
&= \|U_{t+1} - U_t\|_F^2 + \mathrm{DIST}(U_t, U^*)^2 \\
&+ 2\langle U_{t+1} - U_t, U_t - U^*R_t\rangle \\
&= \eta^2\|\nabla f(X_t)U_t\|_F^2 + \|U_t - U^*R_t\|_F^2 \\
&+ 2\eta\langle\nabla f(X_t)U_t, U_t - U^*R_t\rangle
\end{aligned}$$

Key result is the fact that we can prove a regulatory condition:

$$\left\langle\nabla f(X_t)U_t, U - U^kR\right\rangle \geqslant \frac{2}{3}\eta\cdot\|\nabla f(X_t)U\|_F^2 + \frac{3\mu}{20}\sigma_r(X^*)\cdot\mathrm{DIST}(U_t, U^*)^2$$

Using the last two equations, we have:

$$\begin{aligned}
DIST(U_{t+1}, U^*)^2 &\leqslant DIST(U_t - U^*R_t)^2 + \eta^2\cdot\|\nabla f(X_t)U_t\|_F^2 \\
&- \frac{4}{3}\eta^2\|\nabla f(X_t)U_t\|_F^2 - \frac{6\mu\eta}{20}\sigma_r(X^*)\cdot DIST(U_t, U^*)^2 \\
&\leqslant \left(1 - \frac{3\mu\eta}{10}\sigma_r(X^*)\right)\cdot DIST(U_t, U^*)^2
\end{aligned}$$

This defines the step size η.
In practice, the paper ”Dropping Convexity for Faster Semidefinite Optimization” has more sophisticated but more practical η.
However, in order to prove the regulatory condition,
we require

$$DIST(U_t, U^*) \leq \rho\cdot\sigma_r(X^\star)^{\frac{1}{2}}$$

for all t,
which means

$$DIST(U_t, U^*) \leq \rho\cdot\sigma_r(X^\star)^{\frac{1}{2}}$$

leads to good initialization.
As we have gone from convex to non-convex regime, now we have created a dependence over the singular values of $X^\star$, which we do not know.
At the end it gives the following convergence guarantee:

THEOREM: LOCAL CONVERGENCE

Theorem 8. *If f is a “nice” function and (U_i, V_i) are sufficiently close to $U^\star, V^\star)$, then non-convex alternating gradient descent i) converges to $(U^\star, V^\star)$, and ii) achieves the same convergence guarantees with convex optimization:*

Theorem 9. *Global convergence with better initialization: If the function f is “well-conditioned” , then non-convex alternating gradient descent converges to the global optimum / optima.*

i.e in $O(\frac{1}{\epsilon})$ or in $O(\log\frac{1}{\epsilon})$ we will have

$$f\left(\widehat{U}\widehat{V}^\top\right) - f\left(U^\star V^{\star\top}\right) \leq \varepsilon$$

Goal: Initialize such that (U_0, V_0) is sufficiently close to (U^*, V^*)

Proposed initialization
1) Compute $X_0 \propto \nabla f(0_{n\times p})$
2) Perform one SVD calculation:

$$X_0 = U_0V_0^T$$

If the function f is “well-conditioned” , then non-convex alternating gradient descent converges to the global optimum / optima.
The impact here will be: instead of SVD at each step we calculate SVD for first step. The guarantees are weak, but often it works in practice!

Appendix

1. J. Nocedal and S. Wright. Numerical optimization. Springer Science & Business Media, 2006.
2. Y. Nesterov. Introductory lectures on convex optimization: A basic course, volume 87. Springer Science & Business Media, 2013.
3. S. Boyd and L. Vandenberghe. Convex optimization. Cambridge university press, 2004.
4. D. Bertsekas. Convex optimization algorithms. Athena Scientific Belmont, 2015.
5. Sébastien Bubeck. Convex optimization: Algorithms and complexity. Foundations and Trends® in Machine Learning, 8(3-4):231–357, 2015.
6. S. Weisberg. Applied linear regression, volume 528. John Wiley & Sons, 2005.
7. T. Hastie, R. Tibshirani, and M. Wainwright. Statistical learning with sparsity: the lasso and generalizations. CRC press, 2015.
8. J. Friedman, T. Hastie, and R. Tibshirani. The elements of statistical learning, volume 1. Springer series in statistics New York, 2001.
9. M. Paris and J. Rehacek. Quantum state estimation, volume 649. Springer Science & Business Media, 2004.
10. M. Daskin. A maximum expected covering location model: formulation, properties and heuristic solution. Transportation science, 17(1):48–70, 1983.
11. I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. MIT press, 2016.
12. L. Trefethen and D. Bau III. Numerical linear algebra, volume 50. Siam, 1997.
13. G. Strang. Introduction to linear algebra, volume 3. Wellesley-Cambridge Press Wellesley, MA, 1993.
14. G. Golub. Cmatrix computations. The Johns Hopkins, 1996.
15. A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.
16. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.
17. S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015.
18. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119, 2013.
19. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
20. Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 1243–1252. JMLR. org, 2017.
21. Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Fifteenth annual conference of the international speech communication association, 2014.
22. Tom Sercu, Christian Puhrsch, Brian Kingsbury, and Yann LeCun. Very deep multilingual convolutional neural networks for LVCSR. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4955–4959. IEEE, 2016.
23. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. page arXiv:1706.03762, 2017.
24. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. page arXiv:1810.04805, 2018.
25. Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and VQA. In AAAI, pages 13041–13049, 2020.
26. Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. arXiv preprint arXiv:2005.14165, 2020.
27. Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using gpu model parallelism. arXiv preprint arXiv:1909.08053, 2019.
28. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683, 2019.
29. Gary Marcus, Ernest Davis, and Scott Aaronson. A very preliminary analysis of DALL-E 2. arXiv preprint arXiv:2204.13807, 2022.
30. John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. Nature, 596(7873):583–589, 2021.
31. Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.
32. Or Sharir, Barak Peleg, and Yoav Shoham. The cost of training nlp models: A concise overview. arXiv preprint arXiv:2004.08900, 2020.
33. H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 795–811. Springer, 2016.
34. Philip Wolfe. Convergence conditions for ascent methods. SIAM review, 11(2):226–235, 1969.
35. Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. Pacific Journal of mathematics, 16(1):1–3, 1966.
36. Stephen Wright and Jorge Nocedal. Numerical optimization. Springer Science, 35(67-68):7, 1999.
37. B. Polyak. Introduction to optimization. Inc., Publications Division, New York, 1, 1987.
38. Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. lecture notes of EE392o, Stanford University, Autumn Quarter, 2004:2004–2005, 2003.
39. Marguerite Frank, Philip Wolfe, et al. An algorithm for quadratic programming. Naval research logistics quarterly, 3(1-2):95–110, 1956.
40. M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Proceedings of the 30th international conference on machine learning, number CONF, pages 427–435, 2013.
41. J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ_1-ball for learning in high dimensions. In Proceedings of the 25th international conference on Machine learning, pages 272–279, 2008.
42. Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. Computer, (8):30–37, 2009.
43. A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In Advances in neural information processing systems, pages 1257–1264, 2008.
44. T. Booth and J. Gubernatis. Improved criticality convergence via a modified Monte Carlo power iteration method. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
45. S. Zavriev and F. Kostyuk. Heavy-ball method in nonconvex optimization problems. Computational Mathematics and Modeling, 4(4):336–341, 1993.
46. E. Ghadimi, H. Feyzmahdavian, and M. Johansson. Global convergence of the heavy-ball method for convex optimization. In 2015 European control conference (ECC), pages 310–315. IEEE, 2015.
47. Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(\frac{1}{k^2})$. In Soviet Mathematics Doklady, volume 27, pages 372–376, 1983.
48. B. O'Donoghue and E. Candes. Adaptive restart for accelerated gradient schemes. Foundations of computational mathematics, 15(3):715–732, 2015.
49. O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. Mathematical Programming, 146(1-2):37–75, 2014.
50. L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. Siam Review, 60(2):223–311, 2018.
51. S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. SIAM review, 43(1):129–159, 2001.
52. R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996.
53. P. Hoff. Lasso, fractional norm and structured sparse estimation using a Hadamard product parametrization. Computational Statistics & Data Analysis, 115:186–198, 2017.
54. S. Becker, J. Bobin, and E. Candès. NESTA: A fast and accurate first-order method for sparse recovery. SIAM Journal on Imaging Sciences, 4(1):1–39, 2011.
55. T. Blumensath and M. Davies. Iterative hard thresholding for compressed sensing. Applied and computational harmonic analysis, 27(3):265–274, 2009.
56. D. Needell and J. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Applied and computational harmonic analysis, 26(3):301–321, 2009.
57. S. Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. SIAM Journal on Numerical Analysis, 49(6):2543–2563, 2011.
58. J. Tanner and K. Wei. Normalized iterative hard thresholding for matrix completion. SIAM Journal on Scientific Computing, 35(5):S104–S125, 2013.
59. K. Wei. Fast iterative hard thresholding for compressed sensing. IEEE Signal processing letters, 22(5):593–597, 2014.
60. Rajiv Khanna and Anastasios Kyrillidis. Iht dies hard: Provable accelerated iterative hard thresholding. In International Conference on Artificial Intelligence and Statistics, pages 188–198. PMLR, 2018.
61. Jeffrey D Blanchard and Jared Tanner. GPU accelerated greedy algorithms for compressed sensing. Mathematical Programming Computation, 5(3):267–304, 2013.
62. A. Kyrillidis, G. Puy, and V. Cevher. Hard thresholding with norm constraints. In 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3645–3648. Ieee, 2012.
63. A. Kyrillidis and V. Cevher. Recipes on hard thresholding methods. In Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2011 4th IEEE International Workshop on, pages 353–356. IEEE, 2011.
64. X. Zhang, Y. Yu, L. Wang, and Q. Gu. Learning one-hidden-layer ReLU networks via gradient descent. In The 22nd International Conference on Artificial Intelligence and Statistics, pages 1524–1534, 2019.

65. Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on information theory, 52(2):489–509, 2006.
66. Joseph B Altepeter, Daniel FV James, and Paul G Kwiat. 4 qubit quantum state tomography. In Quantum state estimation, pages 113–145. Springer, 2004.
67. Jens Eisert, Dominik Hangleiter, Nathan Walk, Ingo Roth, Damian Markham, Rhea Parekh, Ulysse Chabaud, and Elham Kashefi. Quantum certification and benchmarking. arXiv preprint arXiv:1910.06343, 2019.
68. Masoud Mohseni, AT Rezakhani, and DA Lidar. Quantum-process tomography: Resource analysis of different strategies. Physical Review A, 77(3):032322, 2008.
69. D. Gross, Y.-K. Liu, S. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. Physical review letters, 105(15):150401, 2010.
70. Y.-K. Liu. Universal low-rank matrix recovery from Pauli measurements. In Advances in Neural Information Processing Systems, pages 1638–1646, 2011.
71. K Vogel and H Risken. Determination of quasiprobability distributions in terms of probability distributions for the rotated quadrature phase. Physical Review A, 40(5):2847, 1989.
72. Miroslav Ježek, Jaromír Fiurášek, and Zdeněk Hradil. Quantum inference of states and processes. Physical Review A, 68(1):012305, 2003.
73. Konrad Banaszek, Marcus Cramer, and David Gross. Focus on quantum tomography. New Journal of Physics, 15(12):125020, 2013.
74. A. Kalev, R. Kosut, and I. Deutsch. Quantum tomography protocols with positivity are compressed sensing protocols. Nature partner journals (npj) Quantum Information, 1:15018, 2015.
75. Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. Nat. Phys., 14:447–450, May 2018.
76. Matthew JS Beach, Isaac De Vlugt, Anna Golubeva, Patrick Huembeli, Bohdan Kulchytskyy, Xiuzhe Luo, Roger G Melko, Ejaaz Merali, and Giacomo Torlai. Qucumber: wavefunction reconstruction with neural networks. SciPost Physics, 7(1):009, 2019.
77. Giacomo Torlai and Roger Melko. Machine-learning quantum states in the NISQ era. Annual Review of Condensed Matter Physics, 11, 2019.
78. M. Cramer, M. B. Plenio, S. T. Flammia, R. Somma, D. Gross, S. D. Bartlett, O. Landon-Cardinal, D. Poulin, and Y.-K. Liu. Efficient quantum state tomography. Nat. Comm., 1:149, 2010.
79. BP Lanyon, C Maier, Milan Holzäpfel, Tillmann Baumgratz, C Hempel, P Jurcevic, Ish Dhand, AS Buyskikh, AJ Daley, Marcus Cramer, et al. Efficient tomography of a quantum many-body system. Nature Physics, 13(12):1158–1162, 2017.
80. D. Gonçalves, M. Gomes-Ruggiero, and C. Lavor. A projected gradient method for optimization over density matrices. Optimization Methods and Software, 31(2):328–341, 2016.
81. E. Bolduc, G. Knee, E. Gauger, and J. Leach. Projected gradient descent algorithms for quantum state tomography. npj Quantum Information, 3(1):44, 2017.
82. Jiangwei Shang, Zhengyun Zhang, and Hui Khoon Ng. Superfast maximum-likelihood reconstruction for quantum tomography. Phys. Rev. A, 95:062336, Jun 2017.
83. Zhilin Hu, Kezhi Li, Shuang Cong, and Yaru Tang. Reconstructing pure 14-qubit quantum states in three hours using compressive sensing. IFAC-PapersOnLine, 52(11):188 – 193, 2019. 5th IFAC Conference on Intelligent Control and Automation Sciences ICONS 2019.
84. Zhibo Hou, Han-Sen Zhong, Ye Tian, Daoyi Dong, Bo Qi, Li Li, Yuanlong Wang, Franco Nori, Guo-Yong Xiang, Chuan-Feng Li, et al. Full reconstruction of a 14-qubit state within four hours. New Journal of Physics, 18(8):083036, 2016.
85. C. Riofrío, D. Gross, S.T. Flammia, T. Monz, D. Nigg, R. Blatt, and J. Eisert. Experimental quantum compressed sensing for a seven-qubit system. Nature Communications, 8, 2017.
86. Martin Kliesch, Richard Kueng, Jens Eisert, and David Gross. Guaranteed recovery of quantum processes from few measurements. Quantum, 3:171, 2019.
87. S. Flammia, D. Gross, Y.-K. Liu, and J. Eisert. Quantum tomography via compressed sensing: Error bounds, sample complexity and efficient estimators. New Journal of Physics, 14(9):095022, 2012.
88. A. Kyrillidis, A. Kalev, D. Park, S. Bhojanapalli, C. Caramanis, and S. Sanghavi. Provable quantum state tomography via non-convex methods. npj Quantum Information, 4(36), 2018.
89. B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM review, 52(3):471–501, 2010.
90. N. Srebro, J. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In Advances in neural information processing systems, pages 1329–1336, 2004.
91. J. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In Proceedings of the 22nd international conference on Machine learning, pages 713–719. ACM, 2005.
92. D. DeCoste. Collaborative prediction using ensembles of maximum margin matrix factorizations. In Proceedings of the 23rd international conference on Machine learning, pages 249–256. ACM, 2006.
93. J. Bennett and S. Lanning. The Netflix prize. In Proceedings of KDD cup and workshop, volume 2007, page 35, 2007.
94. M. Jaggi and M. Sulovsk. A simple algorithm for nuclear norm regularized problems. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pages 471–478, 2010.
95. R. Keshavan. Efficient algorithms for collaborative filtering. PhD thesis, Stanford University, 2012.
96. R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In Proceedings of the 22nd international conference on World Wide Web, pages 13–24. International World Wide Web Conferences Steering Committee, 2013.
97. K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain. Sparse local embeddings for extreme multi-label classification. In Advances in Neural Information Processing Systems, pages 730–738, 2015.
98. G. Carneiro, A. Chan, P. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 29(3):394–410, 2007.
99. A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In Computer Vision–ECCV 2008, pages 316–329. Springer, 2008.
100. C. Wang, S. Yan, L. Zhang, and H.-J. Zhang. Multi-label sparse coding for automatic image annotation. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, pages 1643–1650. IEEE, 2009.
101. J. Weston, S. Bengio, and N. Usunier. WSABIE: Scaling up to large vocabulary image annotation. In IJCAI, volume 11, pages 2764–2770, 2011.
102. Andrew I. Schein, Lawrence K. Saul, and Lyle H. Ungar. A generalized linear model for principal component analysis of binary data. In AISTATS, 2003.
103. K.-Y. Chiang, C.-J. Hsieh, N. Natarajan, I. Dhillon, and A. Tewari. Prediction and clustering in signed networks: A local to global perspective. The Journal of Machine Learning Research, 15(1):1177–1213, 2014.
104. C. Johnson. Logistic matrix factorization for implicit feedback data. Advances in Neural Information Processing Systems, 27, 2014.
105. Koen Verstrepen. Collaborative Filtering with Binary, Positive-only Data. PhD thesis, University of Antwerpen, 2015.
106. N. Gupta and S. Singh. Collectively embedding multi-relational data for predicting user preferences. arXiv preprint arXiv:1504.06165, 2015.
107. Y. Liu, M. Wu, C. Miao, P. Zhao, and X.-L. Li. Neighborhood regularized logistic matrix factorization for drug-target interaction prediction. PLoS Computational Biology, 12(2):e1004760, 2016.
108. S. Aaronson. The learnability of quantum states. In Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, volume 463, pages 3089–3114. The Royal Society, 2007.
109. E. Candes, Y. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. SIAM Review, 57(2):225–251, 2015.
110. I. Waldspurger, A. d'Aspremont, and S. Mallat. Phase recovery, MaxCut and complex semidefinite programming. Mathematical Programming, 149(1-2):47–81, 2015.
111. P. Biswas, T.-C. Liang, K.-C. Toh, Y. Ye, and T.-C. Wang. Semidefinite programming approaches for sensor network localization with noisy distance measurements. IEEE transactions on automation science and engineering, 3(4):360, 2006.
112. K. Weinberger, F. Sha, Q. Zhu, and L. Saul. Graph Laplacian regularization for large-scale semidefinite programming. In Advances in Neural Information Processing Systems, pages 1489–1496, 2007.
113. F. Lu, S. Keles, S. Wright, and G. Wahba. Framework for kernel regularization with application to protein clustering. Proceedings of the National Academy of Sciences of the United States of America, 102(35):12332–12337, 2005.
114. H. Andrews and C. Patterson III. Singular value decomposition (SVD) image coding. Communications, IEEE Transactions on, 24(4):425–432, 1976.
115. M. Fazel, H. Hindi, and S. Boyd. Rank minimization and applications in system theory. In American Control Conference, 2004. Proceedings of the 2004, volume 4, pages 3273–3278. IEEE, 2004.
116. E. Candès and B. Recht. Exact matrix completion via convex optimization. Foundations of Computational mathematics, 9(6):717–772, 2009.
117. P. Jain, R. Meka, and I. Dhillon. Guaranteed rank minimization via singular value projection. In Advances in Neural Information Processing Systems, pages 937–945, 2010.
118. S. Becker, V. Cevher, and A. Kyrillidis. Randomized low-memory singular value projection. In 10th International Conference on Sampling Theory and Applications (Sampta), 2013.
119. L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on, pages 704–711. IEEE, 2010.
120. K. Lee and Y. Bresler. ADMiRA: Atomic decomposition for minimum rank approximation. Information Theory, IEEE Transactions on, 56(9):4402–4416, 2010.
121. A. Kyrillidis and V. Cevher. Matrix recipes for hard thresholding methods. Journal of mathematical imaging and vision, 48(2):235–265, 2014.
122. Z. Lin, M. Chen, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint arXiv:1009.5055, 2010.
123. S. Becker, E. Candès, and M. Grant. Templates for convex cone problems with applications to sparse signal recovery. Mathematical Programming Computation, 3(3):165–218, 2011.
124. J. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 20(4):1956–1982, 2010.
125. Y. Chen, S. Bhojanapalli, S. Sanghavi, and R. Ward. Coherent matrix completion. In Proceedings of The 31st International Conference on Machine Learning, pages 674–682, 2014.
126. A. Yurtsever, Q. Tran-Dinh, and V. Cevher. A universal primal-dual convex optimization framework. In Advances in Neural Information Processing Systems 28, pages 3132–3140. 2015.