Chapter 4

We will continue our journey within the convex world and discuss alternatives to (projected) gradient descent. In this chapter, we will introduce *conditional gradient*, also known as the Frank-Wolfe algorithm, for a more efficient convex-constrained optimization. The Frank-Wolfe algorithm got much attention because of the unique structure of some essential convex constraints, such as the ℓ_1-norm constraint—surrogate function for sparsity—and the nuclear norm constraint—surrogate function for low-rankness.

Conditional gradient (Frank-Wolfe) | ℓ_1-norm constraint | nuclear norm constraint

In this chapter, we will focus on the constrained case:

$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & f(x) \\ \text{subject to} \quad & x \in \mathcal{C}. \end{aligned}$$

The discussion thus far focuses on (projected) gradient descent, which can be easily motivated by the following set of motions:

- Under the assumption that f is a L-smooth function (*this a rather mild assumption, that does not even imply convexity*), we know that we can upper bound f at a point x_t as follows:
$$f(x) \leq f(x_t) + \langle \nabla f(x_t),\, x - x_t\rangle + \tfrac{L}{2}\|x - x_t\|_2^2.$$
- This indicates that *locally* and for any given x_t, we can approximate our optimization problem by optimizing the surrogate objective:
$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & f(x_t) + \langle \nabla f(x_t),\, x - x_t\rangle + \tfrac{L}{2}\|x - x_t\|_2^2 \\ \text{subject to} \quad & x \in \mathcal{C}. \end{aligned}$$
- Observe that the surrogate objective itself can be reformulated as:
$$\begin{aligned} &\min_x \left\{ f(x_t) + \langle \nabla f(x_t),\, x - x_t\rangle + \tfrac{L}{2}\|x - x_t\|_2^2 \right\} \\ \propto &\min_x \left\{ \langle \nabla f(x_t),\, x - x_t\rangle + \tfrac{L}{2}\|x - x_t\|_2^2 \right\} \\ \propto &\min_x \left\{ \tfrac{L}{2}\|x - x_t\|_2^2 + \langle \nabla f(x_t),\, x - x_t\rangle + \tfrac{1}{L}\|\nabla f(x_t)\|_2^2 \right\} \\ \propto &\min_x \left\{ \tfrac{L}{2} \cdot \left\| x - \left(x_t - \tfrac{1}{L}\nabla f(x_t)\right)\right\|_2^2 \right\} \\ \propto &\min_x \left\{ \left\| x - \left(x_t - \tfrac{1}{L}\nabla f(x_t)\right)\right\|_2^2 \right\} \end{aligned}$$
The symbol $\propto$ denotes that we can remove/add terms in the objective without affecting the course of the optimization since the removed/added terms are considered constants.
- Thus, our problem becomes:
$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & \left\| x - \left(x_t - \tfrac{1}{L}\nabla f(x_t)\right)\right\|_2^2 \\ \text{subject to} \quad & x \in \mathcal{C}. \end{aligned}$$
- If we denote $y := x_t - \frac{1}{L}\nabla f(x_t)$, then the above problem is just a projection onto $\mathcal{C}$:
$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & \|x - y\|_2^2 \\ \text{subject to} \quad & x \in \mathcal{C}, \end{aligned}$$
which by itself motivates the two-step iterative procedure:
$$x_{t+1} = \Pi_{\mathcal{C}}\left(x_t - \tfrac{1}{L}\nabla f(x_t)\right).$$

What the above motions dictate is that by L-smoothness, we exploit the local quadratic approximations iteratively to minimize f, which by itself motivates the projected gradient descent motions (*the same arguments also hold for the non-projected gradient descent*). A key observation of the above reasoning is that, because of the quadratic form approximation and the $\|x_t - x\|_2^2$ term, we can complete the squares and generate the Euclidean projection operation in the objective.

But is this the only way we can perform/define the projection? Also, what if we take a different order in the Taylor approximation of the objective?

Conditional gradient. The conditional gradient method, also known as the Frank-Wolfe algorithm (see [42]), is based on two approximations compared to the discussion above:

- Instead of a local quadratic approximation, we approximate f locally with a linear function:
$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & f(x_t) + \langle \nabla f(x_t),\, x - x_t\rangle \\ \text{subject to} \quad & x \in \mathcal{C}. \end{aligned}$$
Let s_t be the solution to this problem. To see how s_t is derived, it is the minimizer of the following optimization problem:
$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & \langle \nabla f(x_t),\, x\rangle \\ \text{subject to} \quad & x \in \mathcal{C}. \end{aligned}$$
Given s_t, the direction to move to is $d_t = s_t - x_t$ (*remember that the direction is provided by the term $x - x_t$ in the Taylor approximation $f(x_t) + \langle \nabla f(x_t),\, x - x_t\rangle$*). Thus, using a descent iteration, we have:
$$\begin{aligned} x_{t+1} &= x_t + \eta_t d_t = x_t + \eta_t(s_t - x_t) \\ &= (1 - \eta_t)x_t + \eta_t s_t. \end{aligned}$$
- Observe that the step for finding s_t is a type of projection using the inner product rather than the Euclidean norm. See the figure above for an illustration.
- Of course, we have yet to describe how the projection for s_t is computed. This depends on the set $\mathcal{C}$. More in the text below.

The above summarizes the conditional gradient algorithm. A standard way to set up the step size is as follows:

$$\eta_t = \frac{2}{t+2};$$

i.e., using a decreasing step size [43]. We will see below that this step size selection leads to convergence.

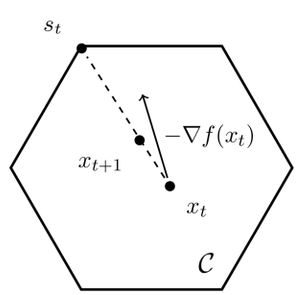

Fig. 32. Borrowed from appendix [5] - Illustration of conditional gradient descent

Conditional gradient convergence analysis. The Frank-Wolfe algorithm has similar convergence rate guarantees with the projected gradient descent algorithm. In the following theorem, we will make the L-smoothness assumption:

Theorem 3. *Let function $f : \mathcal{C} \to \mathbb{R}$ be convex, L-smooth, and assume it attains its global minimum at a point $x^\star \in \mathcal{C}$. Then, Frank-Wolfe iterates achieve:*

$$f(x_{t+1}) - f(x^\star) \leq \frac{2LD^2}{t+2},$$

with step size

$$\eta_t = \frac{2}{t+2}.$$

Here, D is the diameter of $\mathcal{C}$: $D = \max_{x,y\in\mathcal{C}} \|x - y\|_2$.

Proof: By smoothness:

$$f(x_{t+1}) \leq f(x_t) + \langle \nabla f(x_t), x_{t+1} - x_t \rangle + \tfrac{L}{2} \|x_{t+1} - x_t\|^2 .$$

Sequentially substituting the definition x_{t+1} in the above recursion:

$$\begin{aligned} f(x_{t+1}) &\leq f(x_t) + \langle \nabla f(x_t), (1-\eta_t)x_t + \eta_t s_t - x_t \rangle \\ &\quad + \tfrac{L}{2} \|x_{t+1} - x_t\|^2 \\ &= f(x_t) + \eta_t \langle \nabla f(x_t), s_t - x_t \rangle + \tfrac{L}{2} \|x_{t+1} - x_t\|^2 \\ &= f(x_t) + \eta_t \langle \nabla f(x_t), s_t - x_t \rangle \\ &\quad + \tfrac{L}{2} \|(1-\eta_t)x_t + \eta_t s_t - x_t\|^2 \\ &= f(x_t) + \eta_t \langle \nabla f(x_t), s_t - x_t \rangle + \tfrac{\eta_t^2 L}{2} \|s_t - x_t\|^2 \end{aligned}$$

By the definition of D, observe that $\|s_t - x_t\|^2 \leq D^2$. The above inequality then becomes:

$$f(x_{t+1}) \leq f(x_t) + \eta_t \langle \nabla f(x_t), s_t - x_t \rangle + \tfrac{\eta_t^2 LD^2}{2}.$$

Using convexity, we know that

$$\begin{aligned} f(x^\star) \geq f(x_t) + \langle \nabla f(x_t), x^\star - x_t \rangle \Rightarrow \\ \langle \nabla f(x_t), x^\star - x_t \rangle \leq f(x^\star) - f(x_t). \end{aligned}$$

But we also know that s_t is the solution to the problem

$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & f(x_t) + \langle \nabla f(x_t),\ x - x_t \rangle \\ \text{subject to} \quad & x \in \mathcal{C}. \end{aligned}$$

which is equivalent to the following problem:

$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & \langle \nabla f(x_t), x \rangle \\ \text{subject to} \quad & x \in \mathcal{C}. \end{aligned}$$

This implies that:

$$\langle \nabla f(x_t), s_t \rangle \leq \langle \nabla f(x_t), x^\star \rangle$$

and thus:

$$\langle \nabla f(x_t), s_t - x_t \rangle \leq f(x^\star) - f(x_t).$$

Combining all the above in the main recursion (after subtracting $f(x^\star)$ on both sides):

$$f(x_{t+1}) - f(x^*) \leq (1-\eta_t)\left(f(x_t) - f(x^\star)\right) + \tfrac{\eta_t^2 LD^2}{2}$$

We use induction in order to prove $f(x_t) - f(x^\star) \leq \frac{2LD^2}{t+2}$ based on above.

`Base case` $t = 0$. Observe that the above holds for all t. For $t = 0$, we have $\eta_t = \frac{2}{0+2} = 1$. Hence:

$$\begin{aligned} f(x_1) - f(x^\star) &\leq (1-\eta_0)(f(x_0) - f(x^\star)) + \tfrac{\eta_0^2 LD^2}{2} \\ &= (1-1)(f(x_0) - f(x^\star)) + \tfrac{LD^2}{2} \\ &= \tfrac{LD^2}{2} \\ &\leq \tfrac{2LD^2}{2} \end{aligned}$$

Thus, the induction hypothesis holds for our base case.

`Inductive step: from` t `to` $t+1$. Let us assume that for iteration count up to t the following holds: $f(x_t) - f(x^\star) \leq \frac{2LD^2}{t+2}$. We need to show that, under this assumption, it also holds for $t+1$.

By the main recursion, we have:

$$\begin{aligned} f(x_{t+1}) - f(x^\star) &\leq (1-\eta_t)\left(f(x_t) - f(x^\star)\right) + \tfrac{\eta_t^2 LD^2}{2} \\ &= \left(1 - \tfrac{2}{t+2}\right)\left(f(x_t) - f(x^\star)\right) + \tfrac{4}{2(t+2)^2} LD^2 \\ &\leq \left(1 - \tfrac{2}{t+2}\right) \cdot \tfrac{2LD^2}{t+2} + \tfrac{4}{2(t+2)^2} LD^2 \\ &= LD^2 \left(\tfrac{2t}{(t+2)^2} + \tfrac{2}{(t+2)^2}\right) \\ &= 2LD^2 \cdot \tfrac{t+1}{(t+2)^2} \\ &= 2LD^2 \cdot \tfrac{t+1}{t+2} \cdot \tfrac{1}{t+2} \\ &\leq 2LD^2 \cdot \tfrac{t+2}{t+3} \cdot \tfrac{1}{t+2} \\ &= 2LD^2 \tfrac{1}{t+3} \end{aligned}$$

Thus, the inequality also holds for the $t+1$ case.

What if f is also strongly convex. Intuition suggests that we can achieve a linear convergence rate when f is simultaneously L-smooth and μ-strongly convex. Unfortunately, this does not necessarily hold for the Frank-Wolfe/conditional gradient method when we make no assumptions about $\mathcal{C}$, other than convexity and other than the minimum of f over $\mathcal{C}$ exists.

There is a negative result about this statement:

Claim 6. *Consider the following problem:*

$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & \tfrac{1}{2} \langle Qx,\ x \rangle + \langle b, x \rangle \\ \textit{subject to} \quad & x = Av,\ v_i \geq 0,\ \sum_{i=1}^{p} v_i = 1, \end{aligned}$$

for some Q, A matrices with appropriate dimensions, and for a vector b. Observe that the constraint set represents the convex hull of the columns of A.

Let $\{x_t\}$ denote the putative solutions obtained by running conditional gradient with exact line search. Then, there is an initial point x_0 such that, for every $\varepsilon > 0$, we have:

$$f(x_t) - f(x^\star) \geq \tfrac{1}{t^{1+\varepsilon}}$$

In other words, this claim states that *there are problem instances* for which we cannot improve upon the $O(1/t)$ convergence rate. Note though, that this does not exclude the possibility that there is a specific function instance f and a specific constraint $\mathcal{C}$ for which we can provably attain linear convergence.

Where is conditional gradient useful. A quick comparison with projected gradient descent can be summarized as follows:

$$\overset{\text{(Proj. Gradient)}}{O\left(\tfrac{1}{T}\right)} \quad \text{vs} \quad \overset{\text{(Cond. Gradient)}}{O\left(\tfrac{1}{T}\right)}$$

Why then do we care about conditional gradient, given that the iteration complexity is similar to convex projected gradient descent? The answer lies in the projection step and what is the computational complexity needed to complete that step. To depict this clearly, we will consider specific but widely used convex constraints $\mathcal{C}$.

(*Applications and motivation are provided in the corresponding notebook.*)

ℓ_1-norm constraint: The convex set $\mathcal{C}$ in this case is:

$$\mathcal{C} = \{x \in \mathbb{R}^p \;\mid\; \|x\|_1 \leq 1\}.$$

(*The discussion easily extends to the general case $\|x\|_1 \leq \alpha$ for some $\alpha > 0$, but we keep unit ball for simplicity.*) Then, our generic problem looks like:

$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & f(x) \\ \text{subject to} \quad & \|x\|_1 \leq 1. \end{aligned}$$

(Why we use the ℓ_1-norm will be apparent later on in Chapter 7)

To provide some examples, consider the sparse linear regression problem (Lasso problem), as in:

$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & \tfrac{1}{2}\|Ax - b\|_2^2 \\ \text{subject to} \quad & \|x\|_1 \leq 1. \end{aligned}$$

In this ℓ_1-norm case, if we were to perform the Euclidean norm projection:

$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & \|x - y\|_2^2 \\ \text{subject to} \quad & \|x\|_1 \leq 1, \end{aligned}$$

this can be completed through the soft-thresholding operator, where[10]:

$$\widehat{x} = \max\{y - \theta,\, 0\},$$

where

$$\theta = \tfrac{1}{\rho} \cdot \left(\sum_{i=1}^{\rho} y_{\sigma(i)} - 1\right),$$

and

$$\rho = \max\left\{j \in [p] \mid y_{\sigma(j)} - \tfrac{1}{j} \cdot \left(\sum_{q=1}^{j} y_{\sigma(q)} - 1\right) > 0\right\}.$$

Here, $\sigma(\cdot)$ is a descending sorting index. In words, to compute the projection, we need to *i*) sort the input vector y in (usually) $O(p \log p)$ time, *ii*) compute quantities ρ and θ, and *iii*) apply the entrywise rule: $\widehat{x} = \max\{y - \theta,\, 0\}$, *per iteration.*

On the other hand, the conditional gradient "projection" step has the form:

$$\begin{aligned} \min_{x\in\mathbb{R}^p} \quad & \langle \nabla f(x_t), x\rangle \\ \text{subject to} \quad & \|x\|_1 \leq 1. \end{aligned}$$

It is easy to see that if there were no constraints on x, the solution to this minimization is unbounded and leads to $-\infty$ objective value. Under the $\mathcal{C}$, we restrict our search space within a bounded ℓ_1-norm with radius 1; e.g., see the constraint in the figure on the first page of the chapter. We can prove that (one of) the solution(s) to this problem is to put all the "energy" on the component of the gradient $\nabla f(x_t)$ according to:

$$i^\star \in \underset{i\text{-th component}}{\operatorname{argmax}} \quad |\langle \nabla f(x_t), e_i\rangle|$$

where e_i are the basis/coordinate vectors (e.g., e_1 is the p-th dimensional zero vector, except for the first coordinate being 1). Then, s_t is given by:

$$s_t = -1 \cdot \texttt{sgn}\left(\langle \nabla f(x_t), e_{i^\star}\rangle\right) \cdot e_{i^\star}.$$

The first term -1 was chosen on purpose; if we had radius α, then we substitute -1 with $-\alpha$. In other words, to compute the Frank-Wolfe "projection", we need to *i*) find the maximum component (in magnitude) of $\nabla f(x_t)$ in $O(p)$ time, and *ii*) compute s_t in constant time, *per iteration.*

Comparing the two approaches and assuming that (under hidden constants in Big-Oh notation) the number of iterations these algorithms require is the same, the per-iteration complexity of conditional gradient is lower than that of projected gradient descent. In conjunction with the fact that the asymptotic iteration complexity is the same, $O(\frac{1}{T})$, this means that we might prefer a conditional gradient in practice.

Nuclear norm constraint: While the per iteration improvement in the example above seems negligible (we gain $O(\log p)$ per iteration), things get much more interesting in the matrix case. Problems such as matrix completion and matrix sensing take the general form (*we restrict our attention to square matrices just for clarity*):

$$\min_{X\in\mathbb{R}^{p\times p}} \quad f(X) \quad \text{subject to} \quad \|X\|_* \leq 1.$$

Here, the variable is a matrix in $p \times p$, $f(\cdot)$ is a matrix-valued function, and $\|X\|_*$ is the nuclear norm that has the closed-form expression:

$$\|X\|_* = \sum_{i=1}^{p} \sigma_i(X), \quad \text{where } \sigma_i(X) \text{ is the } i\text{-th singular value.}$$

To provide an example, consider the least-squares version of matrix variables:

$$\min_{X\in\mathbb{R}^{p\times p}} \quad \tfrac{1}{2}\|\mathcal{A}(X) - b\|_2^2 \quad \text{subject to} \quad \|X\|_* \leq 1,$$

for some linear transformation mapping $\mathcal{A} : \mathbb{R}^{p\times p} \to \mathbb{R}^m$.

(Why we use the nuclear norm will be apparent later on in Chapter 8)

If we were to perform the Euclidean norm projection:

$$\begin{aligned} \min_{X\in\mathbb{R}^{p\times p}} \quad & \|X - Y\|_F^2 \\ \text{subject to} \quad & \|X\|_* \leq 1, \end{aligned}$$

this can be completed, again, through the soft-thresholding operator over matrices, where instead of "soft-thresholding" elements of a vector, we now soft-threshold the vector of singular values:

$$y \equiv \left[\sigma_1(X),\, \sigma_2(X),\, \ldots,\, \sigma_p(X)\right]$$

[10] For those interested in understanding this part, the instructor suggests you read the paper "Efficient projections onto the ℓ_1-ball for learning in high dimensions" [44]; it was included in the list of papers to review in previous homework.

and

$$\widehat{y} = \max\{y - \theta,\ 0\}.$$

(*The details how to compute the corresponding θ and ρ are left to the reader. Further, details about projections onto nuclear norm and low-rank recovery problems will be the task of Chapter 8.*)

The gist of this description is that to compute the projection of X onto the nuclear ball, we need to compute the *full singular value decomposition* of the input matrix X. This further implies that, for the Euclidean projection, we need to i) compute an SVD in $O(p^3)$ time, ii) perform soft-thresholding and apply the entrywise rule: $\widehat{x} = \max\{y - \theta,\ 0\}$ in $O(p)$, *per iteration*. In real applications (see Netflix recommendation system [45] and [46]), the size of these matrices could be in several hundred thousand, if not millions; thus affording a cubic computational complexity per iteration, is often infeasible.

On the other hand, the conditional gradient "projection" step has the form:

$$\begin{aligned} \min_{x \in \mathbb{R}^p} \quad & \langle \nabla f(X_t), X \rangle \equiv \operatorname{Tr}\left(\nabla f(X_t)^\top X\right) \\ \text{subject to} \quad & \|X\|_* \leq 1. \end{aligned}$$

Let the SVD of the matrix $\nabla f(X_t)$ be:

$$\nabla f(X_t) = U \Sigma V^\top,$$

where, without loss of generality, we have sorted Σ (and the corresponding U, V components) to contain the singular values in descending order. Denote $\sigma_1(X) \equiv \Sigma_{1,1},\ u_1 \equiv U_{:,1},\ v_1 \equiv V_{:,1}$. Then, the Frank-Wolfe projection is equivalent to the solution:

$$S_t = -1 \cdot u_1 v_1^\top.$$

Similarly to the vector case, we substitute the first term -1 with $-\alpha$ if we had radius α. In words, to compute the Frank-Wolfe "projection", we need to i) find the maximum singular value-vector pair of $\nabla f(X_t)$, and ii) compute S_t in $O(p^2)$, *per iteration. The critical difference between this step and the Euclidean projection is that in this case, we care only about the top singular value-vector pair, while in the Euclidean projection step, we need all the singular value-vector pairs.*

A simple way to put it is that assuming that in practice, the SVD (either partial or full) is computed in approximation through iterative methods—such as the Power Iteration method or the Lanczos algorithm—the Frank-Wolfe projection is $\times O(p)$ faster, as it can be roughly be computed in $O(p^2)$ complexity. Thus, the central intuition behind using conditional gradient—as opposed to standard projected gradient descent algorithms— is that, per iteration, we require the best 1-sparse or rank-1 approximation of the gradient to proceed, as opposed to the full ℓ_1-norm constrained or nuclear norm constrained approximation in the former case. For sparsity, a quick analysis shows that this saves a logarithmic factor per iteration. In contrast, in the case of low-rankness, we can save up to $\times O(p)$ per iteration, as we are interested in the rank-1 approximation of the gradient instead of a full rank soft-thresholding projection.

———— ∞ ————

As an interlude, we will study the *power iteration method*. For simplicity, we will consider here $A \in \mathbb{R}^{p \times p}$ is a diagonalizable matrix (e.g., a real symmetric matrix); thus, our focus here is on the eigenvalue decomposition of A, where:

$$A = U \Lambda U^\top, \quad U \in \mathbb{R}^{p \times p}, \Lambda \in \mathbb{R}^{p \times p},$$

where the columns of U are orthonormal and represent the eigenvectors, and Λ is a diagonal matrix, with the eigenvalues, λ_i, on its diagonal. We will make the assumption the eigenvalues are sorted such that

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_p|.$$

Please pay attention that we assume that there is a gap between $|\lambda_1|$ and $|\lambda_2|$. In that case, λ_1 is considered the dominant eigenvalue of the matrix.

In other words, the power iteration method approximates the extremal eigenvalues of the matrix, that is, the eigenvalues having the largest and smallest modules and their associated eigenvectors. Power iteration is simple and can be described by the following two-step procedure:

$$\begin{aligned} \widetilde{q}_{t+1} &= A q_t \\ q_{t+1} &= \frac{\widetilde{q}_{t+1}}{\|\widetilde{q}_{t+1}\|_2} \end{aligned}$$

Observe that:

- The algorithm requires an initial vector $q_0 \in \mathbb{R}^p$.
- The algorithm has no step sizes.
- The algorithm solves a non-convex problem due to the second step (a projection step on $\|\cdot\|_2 = 1$).
- The algorithm requires mostly matrix-vector multiplications, which makes it efficient in practice.

Let us analyze the convergence properties of the power iteration method. Unfolding the recursion, we observe that:

$$q_{t+1} = \frac{A^t q_0}{\|A^t q_0\|_2}.$$

This relation explains the role played by the powers of the input matrix A. For A in $\mathbb{R}^{p \times p}$, its eigenvectors $u_1,\ u_2,\ \ldots,\ u_p$ form a basis in $\mathbb{R}^p$. This means that the initial vector q_0 can be represented as:

$$q_0 = \sum_{i=1}^{p} \alpha_i u_i, \quad \alpha_i \in \mathbb{R}.$$

Moreover, by definition of the eigenvectors, we have:

$$A u_i = \lambda_i u_i, \quad \forall i.$$

The above leads to an equivalent representation of $A^t q_0$ as:

$$\begin{aligned} A^t q_0 = A^t \cdot \sum_{i=1}^{p} \alpha_i u_i &= \sum_{i=1}^{p} \alpha_i A^t u_i \\ &= \sum_{i=1}^{p} \alpha_i \lambda_i^t u_i \\ &= \alpha_1 \lambda_1 \cdot \left(u_1 + \sum_{j=2}^{p} \tfrac{\alpha_j}{\alpha_1} \cdot \left(\tfrac{\lambda_j}{\lambda_1}\right)^t u_j \right). \end{aligned}$$

Using this representation, we can show the following lemma.

Lemma 6. *Assume $A \in \mathbb{R}^{p \times p}$ is a diagonalizable matrix, with eigenvalues $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_p|$. Suppose that $\alpha_1 \neq 0$; then, there exists a constant $c > 0$ such that:*

$$\|z_t - u_1\|_2 \leq c \cdot \left|\tfrac{\lambda_2}{\lambda_1}\right|^t, \quad t \geq 1,$$

where z_t is a scaled version of q_t:

$$z_t = \frac{q_t \cdot \|A^t q_0\|_2}{\alpha_1 \lambda_1^t} = u_1 + \sum_{j=2}^{p} \tfrac{\alpha_j}{\alpha_1} \cdot \left(\tfrac{\lambda_j}{\lambda_1}\right)^t u_j.$$

Proof: It is easy to see that:

$$\begin{aligned}
\|z_t - u_1\|_2 &= \left\| u_1 + \sum_{j=2}^{p} \frac{\alpha_j}{\alpha_1} \cdot \left(\frac{\lambda_j}{\lambda_1}\right)^t u_j - u_1 \right\|_2 \\
&= \left\| \sum_{j=2}^{p} \frac{\alpha_j}{\alpha_1} \cdot \left(\frac{\lambda_j}{\lambda_1}\right)^t u_j \right\|_2 \\
&\leq \left(\sum_{j=2}^{p} \left(\frac{\alpha_j}{\alpha_1}\right)^2 \cdot \left(\frac{\lambda_j}{\lambda_1}\right)^{2t} \right)^{1/2} \\
&\leq \left|\frac{\lambda_2}{\lambda_1}\right|^t \cdot \left(\sum_{j=2}^{p} \left(\frac{\alpha_j}{\alpha_1}\right)^2 \right)^{1/2} \\
&= c \cdot \left|\frac{\lambda_2}{\lambda_1}\right|^t
\end{aligned}$$

for $c := \left(\sum_{j=2}^{p} \left(\frac{\alpha_j}{\alpha_1}\right)^2 \right)^{1/2}$. □

See also [47].

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. Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In Neural networks: Tricks of the trade, pages 9–50. Springer, 2002.
16. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision, pages 1026–1034, 2015.
17. Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. Advances in neural information processing systems, 31, 2018.
18. 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.
19. 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.
20. 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.
21. 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.
22. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
23. 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.
24. 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.
25. 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.
26. 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.
27. 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.
28. 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.
29. 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.
30. 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.
31. 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.
32. Gary Marcus, Ernest Davis, and Scott Aaronson. A very preliminary analysis of DALL-E 2. arXiv preprint arXiv:2204.13807, 2022.
33. 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.
34. 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.
35. Or Sharir, Barak Peleg, and Yoav Shoham. The cost of training nlp models: A concise overview. arXiv preprint arXiv:2004.08900, 2020.
36. 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.
37. Philip Wolfe. Convergence conditions for ascent methods. SIAM review, 11(2):226–235, 1969.
38. Larry Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. Pacific Journal of mathematics, 16(1):1–3, 1966.
39. Stephen Wright and Jorge Nocedal. Numerical optimization. Springer Science, 35(67-68):7, 1999.
40. B. Polyak. Introduction to optimization. Inc., Publications Division, New York, 1, 1987.
41. Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. lecture notes of EE392o, Stanford University, Autumn Quarter, 2004:2004–2005, 2003.
42. Marguerite Frank, Philip Wolfe, et al. An algorithm for quadratic programming. Naval research logistics quarterly, 3(1-2):95–110, 1956.
43. 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.
44. 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.
45. Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. Computer, (8):30–37, 2009.
46. A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In Advances in neural information processing systems, pages 1257–1264, 2008.
47. 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.
48. S. Zavriev and F. Kostyuk. Heavy-ball method in nonconvex optimization problems. Computational Mathematics and Modeling, 4(4):336–341, 1993.
49. 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.
50. 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.
51. B. O'Donoghue and E. Candes. Adaptive restart for accelerated gradient schemes. Foundations of computational mathematics, 15(3):715–732, 2015.
52. 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.
53. L. Bottou, F. Curtis, and J. Nocedal. Optimization methods for large-scale machine learning. Siam Review, 60(2):223–311, 2018.
54. S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. SIAM review, 43(1):129–159, 2001.
55. R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996.
56. P. Hoff. Lasso, fractional norm and structured sparse estimation using a Hadamard product parametrization. Computational Statistics & Data Analysis, 115:186–198, 2017.
57. 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.
58. T. Blumensath and M. Davies. Iterative hard thresholding for compressed sensing. Applied and computational harmonic analysis, 27(3):265–274, 2009.
59. D. Needell and J. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Applied and computational harmonic analysis, 26(3):301–321, 2009.
60. S. Foucart. Hard thresholding pursuit: an algorithm for compressive sensing. SIAM Journal on Numerical Analysis, 49(6):2543–2563, 2011.
61. J. Tanner and K. Wei. Normalized iterative hard thresholding for matrix completion. SIAM Journal on Scientific Computing, 35(5):S104–S125, 2013.
62. K. Wei. Fast iterative hard thresholding for compressed sensing. IEEE Signal processing letters, 22(5):593–597, 2014.
63. 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.
64. Jeffrey D Blanchard and Jared Tanner. GPU accelerated greedy algorithms for compressed sensing. Mathematical Programming Computation, 5(3):267–304, 2013.
65. 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.
66. 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.

67. 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.
68. 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.
69. Joachim Dahl, Lieven Vandenberghe, and Vwani Roychowdhury. Covariance selection for nonchordal graphs via chordal embedding. Optimization Methods & Software, 23(4):501–520, 2008.
70. Joseph B Altepeter, Daniel FV James, and Paul G Kwiat. 4 qubit quantum state tomography. In Quantum state estimation, pages 113–145. Springer, 2004.
71. 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.
72. Masoud Mohseni, AT Rezakhani, and DA Lidar. Quantum-process tomography: Resource analysis of different strategies. Physical Review A, 77(3):032322, 2008.
73. 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.
74. Y.-K. Liu. Universal low-rank matrix recovery from Pauli measurements. In Advances in Neural Information Processing Systems, pages 1638–1646, 2011.
75. 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.
76. Miroslav Ježek, Jaromír Fiurášek, and Zdeněk Hradil. Quantum inference of states and processes. Physical Review A, 68(1):012305, 2003.
77. Konrad Banaszek, Marcus Cramer, and David Gross. Focus on quantum tomography. New Journal of Physics, 15(12):125020, 2013.
78. 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.
79. 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.
80. 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.
81. Giacomo Torlai and Roger Melko. Machine-learning quantum states in the NISQ era. Annual Review of Condensed Matter Physics, 11, 2019.
82. 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.
83. 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.
84. 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.
85. E. Bolduc, G. Knee, E. Gauger, and J. Leach. Projected gradient descent algorithms for quantum state tomography. npj Quantum Information, 3(1):44, 2017.
86. Jiangwei Shang, Zhengyun Zhang, and Hui Khoon Ng. Superfast maximum-likelihood reconstruction for quantum tomography. Phys. Rev. A, 95:062336, Jun 2017.
87. 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.
88. 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.
89. 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.
90. Martin Kliesch, Richard Kueng, Jens Eisert, and David Gross. Guaranteed recovery of quantum processes from few measurements. Quantum, 3:171, 2019.
91. 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.
92. 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.
93. 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.
94. N. Srebro, J. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In Advances in neural information processing systems, pages 1329–1336, 2004.
95. 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.
96. 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.
97. J. Bennett and S. Lanning. The Netflix prize. In Proceedings of KDD cup and workshop, volume 2007, page 35, 2007.
98. 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.
99. R. Keshavan. Efficient algorithms for collaborative filtering. PhD thesis, Stanford University, 2012.
100. 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.
101. 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.
102. 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.
103. A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In Computer Vision–ECCV 2008, pages 316–329. Springer, 2008.
104. 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.
105. J. Weston, S. Bengio, and N. Usunier. WSABIE: Scaling up to large vocabulary image annotation. In IJCAI, volume 11, pages 2764–2770, 2011.
106. Andrew I. Schein, Lawrence K. Saul, and Lyle H. Ungar. A generalized linear model for principal component analysis of binary data. In AISTATS, 2003.
107. 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.
108. C. Johnson. Logistic matrix factorization for implicit feedback data. Advances in Neural Information Processing Systems, 27, 2014.
109. Koen Verstrepen. Collaborative Filtering with Binary, Positive-only Data. PhD thesis, University of Antwerpen, 2015.
110. N. Gupta and S. Singh. Collectively embedding multi-relational data for predicting user preferences. arXiv preprint arXiv:1504.06165, 2015.
111. 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.
112. 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.
113. E. Candes, Y. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. SIAM Review, 57(2):225–251, 2015.
114. I. Waldspurger, A. d'Aspremont, and S. Mallat. Phase recovery, MaxCut and complex semidefinite programming. Mathematical Programming, 149(1-2):47–81, 2015.
115. 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.
116. 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.
117. 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.
118. H. Andrews and C. Patterson III. Singular value decomposition (SVD) image coding. Communications, IEEE Transactions on, 24(4):425–432, 1976.
119. 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.
120. E. Candès and B. Recht. Exact matrix completion via convex optimization. Foundations of Computational mathematics, 9(6):717–772, 2009.
121. 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.
122. S. Becker, V. Cevher, and A. Kyrillidis. Randomized low-memory singular value projection. In 10th International Conference on Sampling Theory and Applications (Sampta), 2013.
123. 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.
124. K. Lee and Y. Bresler. ADMiRA: Atomic decomposition for minimum rank approximation. Information Theory, IEEE Transactions on, 56(9):4402–4416, 2010.
125. A. Kyrillidis and V. Cevher. Matrix recipes for hard thresholding methods. Journal of mathematical imaging and vision, 48(2):235–265, 2014.
126. 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.
127. 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.
128. J. Cai, E. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 20(4):1956–1982, 2010.
129. 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.

130. 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.
131. F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6):386–408, 1958.
132. Robin M. Schmidt, Frank Schneider, and Philipp Hennig. Descending through a crowded valley - benchmarking deep learning optimizers. CoRR, abs/2007.01547, 2020.
133. John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res., 12(null):2121–2159, jul 2011.
134. Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc' aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc Le, and Andrew Ng. Large scale distributed deep networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems, volume 25. Curran Associates, Inc., 2012.
135. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.