

Projection Free Rank-Drop Steps

Edward Cheung Yuying Li

Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada
 {echeung, yuying}@uwaterloo.ca

Abstract

The Frank-Wolfe (FW) algorithm has been widely used in solving nuclear norm constrained problems, since it does not require projections. However, FW often yields high rank intermediate iterates, which can be very expensive in time and space costs for large problems. To address this issue, we propose a rank-drop method for nuclear norm constrained problems. The goal is to generate descent steps that lead to rank decreases, maintaining low-rank solutions throughout the algorithm. Moreover, the optimization problems are constrained to ensure that the rank-drop step is also feasible and can be readily incorporated into a projection-free minimization method, e.g., FW. We demonstrate that by incorporating rank-drop steps into the FW algorithm, the rank of the solution is greatly reduced compared to the original FW or its common variants.

1 Introduction

The Frank-Wolfe algorithm has been widely used for many machine learning applications due to its projection-free property, particularly when linear minimization on the domain is easy but projection is difficult. A particularly interesting problem in machine learning is the nuclear norm constrained problem,

$$\min_{X \in \mathbb{R}^{m \times n}} f(X) \text{ s.t. } \|X\|_{NN} \leq \delta \quad (1)$$

where $\|\cdot\|_{NN}$ is the nuclear norm, which is a typical convex relaxation for rank constrained optimization problems [Fazel *et al.*, 2001]. Common applications of nuclear-norm constrained problems are matrix completions, multivariate regression, multi-task learning, and clustering with missing information. For (1), a projection operation onto the nuclear norm ball will require a full singular value decomposition (SVD), which can be too expensive to perform at each iteration, but is required by methods such as the projected gradient descent.

Without any projection, the FW iterate is guaranteed to be feasible. For (1), the linear subproblem used by FW only requires computing the singular vector pair corresponding to the largest singular value of the gradient at each iteration [Jaggi, 2013]. This is significantly cheaper than computing

the full SVD when the dimension of the matrix is large. A challenge when using FW for (1) is that the nuclear norm ball is a convex hull of an infinite number of rank-one matrices, referred to as the *atoms* of the set of feasible points. Since an atom is added at each iteration, the solution is expressed as a convex combination of an arbitrarily large atomic set, which lacks the crucial low-rank property we desire from nuclear norm constrained problems.

While other methods exist for solving instances of (1), e.g., Active Subspace Selection [Hsieh and Olsen, 2014] or Redistributing Nonconvex Regularizers [Yao *et al.*, 2016], implementations for these methods utilize specific properties of $f(\cdot)$. In [Yao *et al.*, 2016], it appears that for each choice of $f(\cdot)$, a specialized solver must be written, and in [Hsieh and Olsen, 2014], knowledge of special structures using $\nabla f(\cdot)$ and $\nabla^2 f(\cdot)$ is necessary to use the approach efficiently. In contrast, the FW can be readily used for a general $f(\cdot)$ without special modifications.

One of our main goals is to devise a new way to efficiently generate descent feasible steps which also decrease the rank of the current iterate when solving (1) without assuming a specific form for $f(\cdot)$. We propose a new nonconvex optimization formulation to determine the steepest descent rank-drop steps. By further considering the interior rank-drop step and exterior rank-drop step cases separately, we obtain formulations which lead to efficient feasible rank-drop step computation.

We establish theoretical properties of the proposed formulations. In addition we demonstrate computationally that the proposed rank-drop FW method can obtain much lower rank solutions than previous FW alternatives, greatly improving computational efficiency.

1.1 Notation

In this paper, $\sigma_i(A)$ denotes the i^{th} singular value of A , with $\sigma_1 \geq \dots \geq \sigma_{\min\{m,n\}}$. For a matrix $A = [a_{ij}] \in \mathbb{R}^{m \times n}$, $\|A\|_{NN} := \sum_i \sigma_i(A)$ denotes the nuclear norm and $\|A\|_{sp} := \sigma_1(A)$ denotes the spectral norm. Let $\mathcal{B}_{\|\cdot\|}(A, \epsilon) := \{X \in \mathbb{R}^{m \times n} : \|A - X\| \leq \epsilon\}$ be the *closed* norm-ball with a specified norm. We use the term *thin SVD* to mean the SVD with strictly positive singular values, i.e., if $A = U\Sigma V^T$ is a thin SVD and $\text{rank}(A) = r$, then $U \in \mathbb{R}^{m \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$, and $V \in \mathbb{R}^{n \times r}$. Finally, $\langle A, B \rangle := \text{tr}(A^T B)$ denotes the usual trace inner product.

2 The Frank-Wolfe Algorithm and Away Step

We briefly review FW along with its common variants. Here FW is described for a general problem,

$$\min_{x \in \mathcal{S}} f(x) \quad (2)$$

where \mathcal{S} is a closed and bounded convex set.

The FW algorithm can be summarized in Algorithm 1. We denote the FW direction as $d_{fw} := s_k - x_k$.

Algorithm 1 Frank-Wolfe (FW)

```

Let  $x_0 \in \mathcal{S}$ 
for  $k = 0 \dots$  do
     $s_k \leftarrow \arg \min_{s \in \mathcal{S}} \langle s, \nabla f(x_k) \rangle$ 
     $d_{fw} \leftarrow s_k - x_k$ 
     $x_{k+1} \leftarrow x_k + \tau d_{fw}$  for  $\tau \in [0, 1]$ 
end for
    
```

There are many choices for the stepsize τ which ensure convergence. We will assume in this paper that the optimal step-size is used since the stepsize choice is not the focus of this work.

Away steps has been introduced in [Wolfe, 1970] and analyzed in [Guélat and Marcotte, 1986] to move away from “bad atoms” to accelerate convergence. The away step method maintains a set of active atoms, \mathcal{A}_k , such that the current iterate x_k is decomposed into an *atomic decomposition*, i.e., $x_k = \sum_{a_i \in \mathcal{A}_k} \alpha_{a_i} a_i$, where the weight satisfy $\alpha_{a_i} > 0$ and $\sum_{a_i \in \mathcal{A}_k} \alpha_{a_i} \leq 1$.

The away step provides a direction that moves away from one of the active atoms which yields a better descent direction than the current FW direction. This direction solves the following optimization problem,

$$v_k := \arg \min_{v \in \mathcal{A}_k} \langle \nabla f(x_k), x_k - v_k \rangle \quad (3)$$

To ensure convergence, the away direction $d_{away} := x_k - v_k$ is only taken if $\langle \nabla f(x_k), d_{away} \rangle \leq \langle \nabla f(x_k), d_{fw} \rangle$. Algorithm 2 summarizes the away step FW method (AFW) in [Lacoste-Julien and Jaggi, 2015].

2.1 In-Face Steps

In [Freund *et al.*, 2015], *In-Face Steps* are proposed to generalize away-steps. Instead of generating descent directions by moving away from bad atoms, the in-face step selects the best descent direction along the minimal face containing the current iterate. By aiming to move along the minimal face to the boundary of the feasible region to reach a lower dimensional face, the algorithm explicitly maintains a low-rank structure without compromising convergence. For (1), as in [Freund *et al.*, 2015], the minimal face $\mathcal{F}(X_k)$ of $\mathcal{B}_{NN}(0, \delta)$ containing a point X_k is given by the set,

$$\mathcal{F}(X_k) = \begin{cases} \mathcal{B}_{NN}(0, \delta), & \text{when } \|X_k\| < \delta \\ U M V^\top, & \text{otherwise,} \end{cases} \quad (4)$$

where X_k has a thin r rank SVD, $U \Sigma V^\top$, M is a real positive semidefinite matrix with $\text{tr}(M) = \delta$.

Algorithm 2 (Atomic) Away Steps Frank-Wolfe (AFW)

```

Let  $x_0 \in \mathcal{S}$  and  $\mathcal{A}_0 \leftarrow \{x_0\}$ .
Initialize  $\alpha_s \leftarrow 0, \forall s \in \mathcal{S}$ .
for  $k = 0 \dots$  do
     $s_k \leftarrow \arg \min_{s \in \mathcal{S}} \langle s, \nabla f(x_k) \rangle, d_{fw} \leftarrow s_k - x_k$ 
     $v_k \leftarrow \arg \max_{v \in \mathcal{A}_k} \langle v, \nabla f(x_k) \rangle, d_{away} \leftarrow x_k - v_k$ 
    if  $\langle -\nabla f(x_k), d_{fw} \rangle \geq \langle -\nabla f(x_k), d_{away} \rangle$  then
         $d_k \leftarrow d_{fw}$ 
         $\tau^* \leftarrow \arg \min_{\tau \in [0, 1]} f(x_k + \tau d_k)$ 
         $\alpha_{a_i} \leftarrow (1 - \tau^*) \alpha_{a_i}, \forall a_i \in \mathcal{A}_k$ 
         $\alpha_{s_k} \leftarrow \alpha_{s_k} + \tau^*, \mathcal{A}_k \leftarrow \mathcal{A}_k \cup \{s_k\}$ 
    else
         $d_k \leftarrow d_{away}$ 
         $\tau^* \leftarrow \arg \min_{\tau \in [0, \frac{\alpha_{v_k}}{1 - \alpha_{v_k}}]} f(x_k + \tau d_k)$ 
         $\alpha_{v_i} \leftarrow (1 + \tau^*) \alpha_{v_i}, \forall v_i \in \mathcal{A}_k$ 
         $\alpha_{v_k} \leftarrow \alpha_{v_k} - \tau^*$ 
    end if
     $x_{k+1} \leftarrow x_k + \tau^* d_k$ 
     $\mathcal{A}_{k+1} \leftarrow \{a \in \mathcal{A}_k : \alpha_a > 0\}$ 
end for
    
```

In particular, for comparison, we consider the In-Face step from the *Away-step Strategy* described in [Freund *et al.*, 2015], which uses the following direction,

$$Z_k := \arg \max_{Z \in \mathcal{F}(X_k)} \langle \nabla f(X_k), Z \rangle. \quad (5)$$

3 Rank-Drop Steps

One significant issue with the FW method for nuclear norm constrained problems is that the rank of the intermediate solution often steadily increases [Freund *et al.*, 2015], and frequently yielding a high rank solution upon termination. In Figure 1, we highlight this phenomenon. For very large problems, we cannot store the matrix X_k , and instead, only maintain the rank-one factors. Thus, computing the gradient at each iteration can be computationally prohibitive. For matrix completion, as an example, if Ω is the set of observed entries, then forming the gradient requires at least $|\Omega| \cdot r$ inner product calculations. Since the number of observed entries, $|\Omega|$, is typically very large, any increase in the rank of the solution greatly increases the computational time in each iteration.

To address this issue, we propose to search over the set of rank-one matrices which decrease the rank of the iterate by one. The proposed optimization formulation for determining such a rank-one matrix is motivated by the following theorem.

Theorem 3.1 ([Egerváry, 1960]). *Let $u \in \mathbb{R}^m, v \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}$, and $B = A - \sigma^{-1} u v^\top$. Then $\text{rank}(B) = \text{rank}(A) - 1$ if and only if there are vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ such that $u = Ax, v = A^\top y$ and $\sigma = y^\top Ax \neq 0$.*

3.1 The Rank-Drop Optimization Framework

Assume that $\text{rank}(X_k) > 1$. Our goal is to determine a rank-drop step which reduces the rank of X_k but has a comparable objective value. We restrict our attention to the rank-one matrices that can decrease the rank of a matrix; let $\mathcal{R}(A)$ denote,

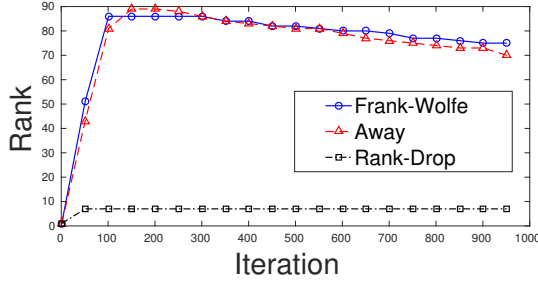


Figure 1: The Frank-Wolfe and Away-Step methods reach very high rank solutions at convergence for the MovieLens 100k dataset with $\delta = 1$. The rank of the iterate becomes very large before the rank begins to decrease very slowly. In contrast, the proposed rank-drop FW method maintains very low-rank intermediate solutions throughout the entire computation. The objective values at termination are 1905.4, 1905.4, 1905.6 for FW, Away Step, Rank-Drop respectively.

for a matrix A , the set of *rank-drop steps*, which are these rank-one matrices,

$$\mathcal{R}(A) := \{\sigma^{-1}uv^\top : \exists x \in \mathbb{R}^n, y \in \mathbb{R}^m \text{ s.t. } u = Ax, v = A^\top y, \sigma = y^\top Ax \neq 0\}. \quad (6)$$

Since u and v must be in the column and row spans of A respectively, Lemma 3.2 shows that the set of rank-drop steps can be expressed in a more concise form.

Lemma 3.2.¹ Let $A \in \mathbb{R}^{m \times n}$ with a thin SVD $A = U\Sigma V^\top$. Then we can rewrite the set of rank-drop steps as,

$$\mathcal{R}(A) = \left\{ \frac{Ust^\top V^\top}{s^\top \Sigma^{-1}t}, s^\top \Sigma^{-1}t > 0 \right\}. \quad (7)$$

Let $Z_k \in \mathcal{R}(X_k)$ and $\hat{Z}_k = Z_k / \|Z_k\|_{NN}$. Then, similar to away steps, we consider iterates in the following form,

$$X_{k+1} := X_k + \tau D_{rd}, \quad \text{where } D_{rd} := X_k - \delta \hat{Z}_k. \quad (8)$$

Our goal is to find a rank-drop step that results in a feasible iterate which best minimizes the objective. Since not all rank-drop steps lead to feasible iterates and testing for feasibility at each iteration is computationally expensive, we want to establish a verifiable sufficient condition for rank-drop steps that ensure feasibility. In Lemma 3.3, we first establish a lower bound on the nuclear norm of the rank-drop step.

Lemma 3.3. Let $A \in \mathbb{R}^{m \times n}$ with rank r and let $Z \in \mathcal{R}(A)$ be an arbitrary rank-drop step. Then $\|Z\|_{NN} \geq \sigma_r(A)$.

To ensure feasibility of the iterate after a rank-drop step, we consider two distinct cases. Let $\kappa(X_k)$ be half of the distance between X_k and the boundary of the nuclear norm ball,

$$\kappa(X_k) := \frac{\delta - \|X_k\|_{NN}}{2} \quad (9)$$

We consider $\kappa(X_k) \geq \sigma_r(X_k)$, the *interior case*, and $\kappa(X_k) < \sigma_r(X_k)$, the *exterior case*. For each case, we will motivate a rank-drop step formulation to guarantee feasibility of the next iterate and provide a method for its computation.

¹Due to space limitations, proofs will be omitted. The proofs will be added in an appendix at <https://arxiv.org/abs/1704.04285>.

3.2 The Interior Rank-Drop Problem

Assume that $\kappa(X_k) \geq \sigma_r(X_k)$, where $\kappa(X_k)$ is defined in (9). Theorem 3.4 below establishes an easily verifiable condition guaranteeing that X_{k+1} , after taking the rank-drop step, remains in the interior of the nuclear ball.

Theorem 3.4. Let $X_k \in \mathbb{R}^{m \times n}$ with $\|X_k\|_{NN} < \delta$, $Z_k \in \mathcal{R}(A)$, and $\hat{Z}_k = Z_k / \|Z_k\|_{NN}$. If $\|Z_k\| \leq (\delta - \|X_k\|_{NN})/2$, then $X_{k+1} = X_k + \tau(X_k - \delta \hat{Z}_k) \in \mathcal{B}_{NN}(0, \delta)$ for every $\tau \in [0, \tau^*]$, where $\tau^* = \|Z_k\|_{NN} / (\delta - \|Z_k\|_{NN})$. Moreover, if $\tau = \tau^*$, then $\text{rank}(X_{k+1}) = \text{rank}(X_k) - 1$.

Theorem 3.4 motivates the following formulation,

$$\begin{aligned} \min_Z \quad & \langle -Z, \nabla f(X_k) \rangle \\ \text{s.t.} \quad & Z \in \mathcal{B}_{NN}(0, \kappa(X_k)) \cap \mathcal{R}(X_k). \end{aligned} \quad (10)$$

Problem (10) determines, in the interior case, the best rank-drop step for solving (1) based on the first order approximation to the objective. The constraint on Z guarantees that the next iterate is feasible if a rank-drop step is taken.

Theorem 3.5. If $\text{rank}(X_k) \geq 1$ with $\sigma_r(X_k) \leq \kappa(X_k)$, then the feasible region for (10) is non-empty.

Assume that a thin SVD for X_k , $X_k = U\Sigma V^\top$, is given. Using Lemma 3.2, the constraint in (10) can be made explicit,

$$\begin{aligned} \min_{s,t \in \mathbb{R}^r} \quad & \left\langle X_k - \frac{Ust^\top V^\top}{s^\top \Sigma^{-1}t}, \nabla f(X_k) \right\rangle \\ \text{s.t.} \quad & \frac{Ust^\top V^\top}{s^\top \Sigma^{-1}t} \in \mathcal{B}_{NN}(0, \kappa(X_k)) \\ & s^\top \Sigma^{-1}t > 0. \end{aligned} \quad (11)$$

To make (11) more amenable to computation, we remove the fraction using the normalization constraint $s^\top \Sigma^{-1}t = \kappa(X_k)^{-1}$ and formulate the problem equivalently as follows,

$$\begin{aligned} \min_{s,t \in \mathbb{R}^r} \quad & q(s,t) := \langle \nabla f(X_k), -\kappa(X_k)Ust^\top V^\top \rangle \\ \text{s.t.} \quad & s^\top \Sigma^{-1}t = \kappa(X_k)^{-1} \\ & \|s\|_2 = 1, \|t\|_2 \leq 1 \end{aligned} \quad (12)$$

Note that the constraints in (12) also ensure that s and t cannot be rescaled to obtain a different solution yielding an identical rank-drop step. The equivalence of (10) and (12) is formally established in Theorem 3.6.

Theorem 3.6. If X_k is a feasible point of (1) and $\kappa(X_k) \geq \sigma_r(X_k)$, then an optimal solution to (12) is an optimal solution to (11). Moreover, an optimal to (11) can always be rescaled into an optimal solution to (12).

3.3 The Exterior Rank-Drop Problem

Now consider the case $\kappa(X_k) < \sigma_r(X_k)$. If $\kappa(X_k)$ is relatively large and X_k is not nearly rank deficient, rank-drop is less important. If $\kappa(X_k)$ is small, however, then X_k is close to the boundary of the nuclear ball. In this case, we are interested in rank-drop descent directions that move inside the nuclear norm ball. We establish the following theorems to facilitate formulating appropriate optimization problems for this case. Subsequently $\partial g(X)$ denotes the subdifferential of $g(X)$ at X .

Theorem 3.7. Let $X_k \in \mathcal{B}_{NN}(0, \delta)$ have the thin SVD $X_k = U\Sigma V^\top$. Define $D_k = X_k - \delta \hat{Z}_k$ with $Z_k \in \mathcal{R}(X_k)$ and $\hat{Z}_k = Z_k / \|Z_k\|_{NN} = Us^\top V^\top$, for some s and t with $\|s\| = \|t\| = 1$. Then,

$$\left(\max_{G \in \partial \|X_k\|_{NN}} \langle D_k, G \rangle \right) \leq 0, \quad (13)$$

if and only if $\delta s^\top t \geq \|X_k\|_{NN}$.

Following Theorem 3.7, D_k will be a descent direction for the nuclear norm at X_k if $\delta s^\top t \geq \|X_k\|_{NN}$.

When X_k is on the boundary of the nuclear norm ball, Corollary 3.7.1 below states a simpler and useful characterization.

Corollary 3.7.1. Let X_k have the thin SVD $X_k = U\Sigma V^\top$ with $\|X_k\|_{NN} = \delta$. Define $D_k = X_k - \delta \hat{Z}_k$ with $Z_k \in \mathcal{R}(X_k)$ and $\hat{Z}_k = Z_k / \|Z_k\|_{NN} = Us^\top V^\top$, for some s and t with $\|s\|_2 = \|t\|_2 = 1$. Then,

$$\left(\max_{G \in \partial \|X_k\|_{NN}} \langle D_k, G \rangle \right) \leq 0 \quad (14)$$

if and only if $s = t$.

When X_k is on the boundary of the nuclear ball, $\kappa(X_k) = 0$. Lemma 3.8 below shows that X_k is close to the boundary when either r is large or when $\sigma_r(X_k)$ is small, assuming $\kappa(X_k) < \sigma_r(X_k)$.

Lemma 3.8. If $\kappa(X_k) < \sigma_r(X_k)$, then $\|X_k\|_{NN} > \frac{r}{r+2}\delta$.

For the exterior case, $\kappa(X_k) < \sigma_r(X_k)$, we are mostly interested in the situation when X_k is close to the boundary of the nuclear norm ball. Consequently, we motivate the next formulation assuming $\|X_k\|_{NN} = \delta$ as a good approximation.

When X_k is on the boundary of the nuclear norm ball, we consider all directions which move into the nuclear norm ball, hence feasible directions, as candidate solutions. From Corollary 3.7.1, this can be reduced to searching over directions $D_k = X_k - \delta Us^\top V^\top$. Let $W = U^\top \nabla f(X_k) V$. Note that $\frac{1}{2}s^\top (W^\top + W)s = \langle \nabla f(X_k), Us^\top V^\top \rangle$. Hence, in the exterior case, we solve the following optimization problem,

$$\max_{s \in \mathbb{R}^r} \frac{1}{2} \frac{s^\top (W^\top + W)s}{s^\top \Sigma^{-1} s} \quad s.t. \quad \|s\|_2 = 1. \quad (15)$$

We note that (15) is a generalized eigenvalue problem, which can be further reduced to a standard eigenvalue problem since Σ^{-1} is a nonsingular diagonal matrix.

Theorem 3.9 below establishes that, using (15), the required step-size to decrease the rank of the current iterate also guarantees that the next iterate is feasible.

Theorem 3.9. Let s be an optimal solution to (15) and let $D_k = X_k - \delta Us^\top V^\top$, where $X_k = U\Sigma V^\top$ is a thin SVD with $\|X_k\|_{NN} \leq \delta$. If $\tau^* = (\delta s^\top \Sigma^{-1} s - 1)^{-1}$ and $X_{k+1} = X_k + \tau^* D_k$, then $\text{rank}(X_{k+1}) = \text{rank}(X_k) - 1$ and $\|X_{k+1}\|_{NN} \leq \delta$.

We observe that τ^* matches the upper bound on the step-size for in-face steps when an away-step method is used and when $\|X_k\| = \delta$.

Comparison with In-Face Steps

Lemma 3.10 and 3.11 below show that the rank-drop steps derived from (12) and (15) lie on the minimal face of $\mathcal{B}_{NN}(0, \delta)$ containing X_k , denoted as $\mathcal{F}(X_k)$.

Lemma 3.10. Suppose $\|X_k\|_{NN} < \delta$ and $\kappa(X_k) \geq \sigma_r(X_k)$, with thin SVD $X_k = U\Sigma V^\top$. Then $Z = \kappa(X_k) Us^\top V^\top \in \mathcal{F}(X_k)$, where (s, t) is the solution to (12).

Lemma 3.11. Suppose $\kappa(X_k) < \sigma_r(X_k)$, with thin SVD $X_k = U\Sigma V^\top$. Then $Z = \delta Us^\top V^\top \in \mathcal{F}(X_k)$, where s is the solution to (15).

We emphasize, however, that the rank-drop steps are constructed explicitly to decrease the rank of the current iterate. Specifically, we highlight the following differences between rank-drop steps and the in-face direction (5) suggested in [Freund *et al.*, 2015]. Firstly, when the iterate is in the interior of the nuclear norm ball, the in-face steps often do not lead to rank-drop steps. Moreover, at each iteration when the iterate is in the interior, a binary search is required to determine the maximum feasible step length once the direction is computed. This requires several SVD updates, leading to very expensive intermediate iterates. Thus, it is critical for the iterates of the in-face method to reach the boundary of the nuclear norm ball quickly. Secondly, the parametrization suggested in [Freund *et al.*, 2015] (i.e. $\gamma_1 = 0$ and $\gamma_2 = \infty$) for large datasets corresponds to only taking the maximum step length. On the boundary of the nuclear norm ball, this is equivalent to only accepting iterates with rank decreased. Since our proposed rank-drop is determined optimally to decrease the objective (up to the first order) among all rank-drop steps explicitly, we believe that this is a better optimization formulation when the iterate approaches the boundary.

Solving the Optimization Problem in the Interior Case

Now we discuss how to solve the Rank-Drop optimization problem (12) in the interior case. Let $W := U^\top \nabla f(X_k) V$.

The Lagrangian for (12) is,

$$\begin{aligned} \mathcal{L}(s, t, \lambda, \alpha, \beta) := & s^\top W t + \lambda (s^\top \Sigma^{-1} t - \kappa(X_k)^{-1}) \\ & + \alpha (s^\top s - 1) + \beta (t^\top t - 1). \end{aligned}$$

Theorem 3.12. Suppose $(s, t, \lambda, \alpha, \beta)$ satisfies the KKT conditions of (12) with $\|t\|_2 < 1$. Then $\alpha = \beta = 0$, λ is an eigenvalue of $-\Sigma W$, and $M_\lambda := -\frac{1}{2}(W + \lambda \Sigma^{-1})$ has zero as a singular value. Conversely, assume that (\hat{s}, \hat{t}) forms a singular vector pair of M_λ associated with the singular value zero and $\kappa(X_k) \hat{s}^\top \Sigma^{-1} \hat{t} > 1$, then the KKT conditions of (12) are satisfied at $(\hat{s}, \hat{t} / (\kappa(X_k) \hat{s}^\top \Sigma^{-1} \hat{t}), \lambda, 0, 0)$.

Since problem (12) is nonconvex and generally difficult to solve, we only consider candidate KKT points characterized in Theorem 3.12. For each eigenvalue λ of $-\Sigma W x$, we fix $\alpha = \beta = 0$ and choose (\hat{s}, \hat{t}) that corresponds to the singular vector pair of $-0.5(W + \lambda \Sigma^{-1})$ with the singular value zero. If $(s, t) = (\hat{s}, \hat{t} / (\kappa(X_k) \hat{s}^\top \Sigma^{-1} \hat{t}))$ is feasible with respect to (12) and $\|t\|_2 < 1$, then by Theorem 3.12, $(s, t, \lambda, 0, 0)$ satisfies the KKT conditions.

In the event that no feasible candidate is found, we solve (15) instead and obtain a feasible rank-drop step, guaranteed

by Theorem 3.9. We remark that, although it is always possible to find feasible rank-drop steps using (15), in the interior case, it is still preferable to solve problem (12) since the assumption that $s = t$ can be restrictive and unnecessary when far away from the boundary. Algorithm 3 presents the details of the rank-drop direction computation.

3.4 Step Selection Criteria

The criteria in [Freund *et al.*, 2015] for away-steps require solving the regular FW linear subproblem at each iteration even when the FW step is not taken. In the rank-drop framework, we accept any rank-drop step that does not increase the objective. This allows the algorithm to maintain a lower rank SVD as well as allows the algorithm to skip computing a FW step unnecessarily. Additionally, for the rank-drop step, we observe that it is very rare that the algorithm chooses a rank-drop step two iterations in a row. Thus, we only generate the rank-drop direction when the previous iterate is a regular FW step, avoiding unnecessary rank-drop step computations.

4 Convergence Analysis

Following the proof for Theorem 4 in [Guélat and Marcotte, 1986], we show that the iterates, from Rank Drop FW in Algorithm 4, converge to the global optimum of (1).

Theorem 4.1. *Let $\{X_k\}$ be a sequence generated by Algorithm 4 and let f^* be the optimal value for problem (1). Assume $\nabla f(X)$ is Lipschitz continuous in the feasible region. Then $f(X_k) - f^* \leq \frac{8\delta^2 L}{4 + N_{fw}^k}$, where N_{fw}^k be the number of FW steps taken up to the iteration k .*

5 Complexity Per Iteration

When computing the rank-drop steps, we note that the dimension of the subproblems is r , the rank of the current iterate. First, forming the matrix $W := U^\top \nabla f(X_k) V$ requires $O(r^2 h \min\{m, n\})$ operations, where h is the maximum number of nonzero elements in any row or column. In the interior case, we must compute an eigen-decomposition of an $r \times r$ matrix which takes $O(r^3)$ time. Then, each eigenvalue λ is used to form the matrix $-0.5(W + \lambda \Sigma^{-1})$ where the singular vector pair corresponding to the zero singular value is computed. The total time required for the interior case is $O(r^3 + r^2 h \min\{m, n\})$. In the exterior case, $O(r^2)$ flops are required to compute the largest eigenvalue. Thus, the total complexity per iteration is $O(r^3 + r^2 h \min\{m, n\})$.

6 Experimental Results

We validate the rank-drop steps on a matrix completion task using various datasets from MovieLens². We first center and scale each data set to have mean 0 and standard deviation 1. We compare the proposed Rank-Drop Frank Wolfe (RDFW) against the aforementioned FW variants, FW [Frank and Wolfe, 1956], AFW [Lacoste-Julien and Jaggi, 2015], and IF(0, ∞), [Freund *et al.*, 2015], as well as a state-of-the-art nuclear norm regularized solver in ActiveALT [Hsieh and Olsen, 2014].

²<http://grouplens.org/datasets/movielens/>

Algorithm 3 Compute Rank-Drop Direction (rankDrop)

```

Input: thin SVD  $X_k := U \Sigma V^\top$  and  $\nabla f(X_k)$ .
 $\kappa(X_k) \leftarrow \frac{\delta - \|X_k\|_{NN}}{2}$ 
 $W \leftarrow U^\top \nabla f(X_k) V$ 
if  $\kappa(X_k) \geq \sigma_r(X_k)$  then
    (Interior Case)
     $\Lambda \leftarrow \text{eigs}(-\Sigma W)$ 
     $b \leftarrow -\infty$ 
    for  $\lambda_i \in \Lambda$  do
         $M_\lambda := -\frac{1}{2}(W + \lambda_i \Sigma^{-1})$ 
         $(s, t, \sigma) \leftarrow \text{SVD}(M_\lambda)$  (return the zero SV)
        if  $\kappa(X_k) s^\top \Sigma^{-1} t \geq 1$  and  $q\left(s, \frac{t}{\kappa(X_k) s^\top \Sigma^{-1} t}\right) > b$ 
            then
                 $(s^*, t^*) \leftarrow (s, t)$ 
                 $b \leftarrow q\left(s, \frac{t}{\kappa(X_k) s^\top \Sigma^{-1} t}\right)$ 
            end if
        end for
     $\alpha \leftarrow (s^\top \Sigma^{-1} t)^{-1}$ 
     $\tau^* \leftarrow \frac{\alpha}{\delta - \alpha}$ 
end if
if  $\kappa(X_k) < \sigma_r(X_k)$  or  $b = -\infty$  then
    (Exterior case or no candidates from the Interior Case)
     $s^* \leftarrow \text{genEig}(0.5(W + W^\top), \Sigma^{-1})$ 
     $t^* \leftarrow s^*$ 
     $\tau^* \leftarrow (\delta s^\top \Sigma^{-1} s - 1)^{-1}$ 
end if
return  $(s^*, t^*, \tau^*)$ 
    
```

Algorithm 4 Rank-Drop Frank-Wolfe (RDFW)

```

Let  $X_0 \in \mathcal{S}$ , with initial SVD  $X_0 = U \Sigma V^\top$ , and maximum iteration count  $T$ 
prevStep  $\leftarrow$  "RD" (Start with Frank-Wolfe step)
for  $k = 0 \dots T$  do
    Compute  $\nabla f(X_k)$ 
    if prevStep == "RD" then
        GO TO (Frank-Wolfe)
    end if
    Compute Rank-Drop direction (see Algorithm 3):
     $(s_k, t_k, \tau^*) \leftarrow \text{rankDrop}(U, \Sigma, V, \nabla f(X_k))$ 
     $\tilde{X} := X_k + \tau^*(X_k - \delta U s_k t_k^\top V^\top)$ 
    if  $f(\tilde{X}) \leq f(X_k)$  then
         $X_{k+1} \leftarrow \tilde{X}$ 
         $Z_k \leftarrow -\delta U s_k t_k^\top V^\top$ 
        prevStep  $\leftarrow$  "RD"
    else
        (Frank-Wolfe)
         $Z_k \leftarrow \arg \min_{Z \in \mathcal{B}_{NN}(0, \delta)} \langle Z, \nabla f(X_k) \rangle$ 
         $\tau^* \leftarrow \arg \min_{\tau \in [0, 1]} f(X_k + \tau(Z_k - X_k))$ 
         $X_{k+1} := X_k + \tau^*(Z_k - X_k)$ 
        prevStep  $\leftarrow$  "Frank-Wolfe"
    end if
     $(U, \Sigma, V) \leftarrow \text{updateSVD}(U, \Sigma, V, \tau^* Z_k)$ 
end for
    
```

Dataset		Frank-Wolfe	Away-Step FW	In-Face(0, ∞)	ActiveALT	Rank-Drop FW
ML-100k	RMSE	0.878	0.878	0.878	0.877	0.879
	Rank (Max)	501.4 (504)	466.8 (477)	54.2 (64)	43.4 (50)	41.6 (44)
	Time (s)	146.09	152.49	37.85	30.28	35.45
ML-1M	RMSE	0.820	0.820	0.822	0.818	0.820
	Rank (Max)	467.4 (478)	425.2 (430)	77.6 (112)	65 (68)	67 (69)
	Time (s)	1,127.70	1,197.67	356.42	503.39	284.85
ML-10M	RMSE	0.806	0.806	0.807	0.806	0.808
	Rank (Max)	1000 (1000)	889.8 (893)	210 (297)	141.8 (144)	140 (140)
	Time (s)	41,467.60	44,327.57	29,630.43	13,281.95	12,499.43
ML-20M	RMSE	-	-	0.800	0.800	0.801
	Rank (Max)	-	-	274.6 (471)	206.2 (214)	202 (203)
	Time (s)	-	-	117,535.82	38,497.16	29,102.62

Table 1: Computational results on matrix completion problems averaged over 5 random initializations. The max rank is the maximum rank observed over all 5 trials. For ML-20M, the FW and AFW algorithms took too long to successfully terminate.

Dataset	# Users	# Movies	# Ratings
MovieLens 100k	943	1,682	100,000
MovieLens 1M	6,040	3,900	1,000,209
MovieLens 10M	82,248	10,681	10,000,054
MovieLens 20M	138,493	27,278	20,000,263

Table 2: MovieLens Data

Following [Yao *et al.*, 2016], we randomly partition each dataset into 50% training, 25% validation, and 25% testing. The δ value in (1) is tuned with $\delta = \mu_j \cdot \|Y\|_F$, where $\|Y\|_F$ is the Frobenius norm of the training data matrix, and $\mu_j = 2 + 0.2j$, $j \in \mathbb{N}$. We increase j until the mean RMSE on the validation set does not improve by more than 10^{-3} . We terminate the algorithm when an upper bound on the relative optimality gap ensures $(f(X_k) - f^*)/f^* < 10^{-2}$ or a maximum iteration count of 1000 is reached.

For ActiveALT, a regularized nuclear norm problem (not constrained) is solved where the regularization parameter λ is chosen by approximately solving for the Lagrange multiplier from the solution to the constrained problem. From the optimality conditions, we have $U^\top \nabla f(X^*)V + \lambda I = 0$. Thus, λ is approximated by the mean of the diagonal values of $U^\top f(X_k)V$, where X_k is the converged solution of RDFW. ActiveALT terminates when $f(X_{k-1}) - f(X_k) < 10^{-4}$, to match with the criterion suggested in [Yao *et al.*, 2016] with a maximum iteration count of 150.

Dataset	Frank-Wolfe δ	ActiveALT λ
MovieLens 100k	3	10.94
MovieLens 1M	3.4	22.7
MovieLens 10M	5.2	49.4
MovieLens 20M	6.6	59.04

Table 3: Parameters used for each dataset.

6.1 Computational Details

All simulations were run in MATLAB. For all FW variants, we maintain a thin SVD for the current iterate, where the SVD is updated at each iteration using a rank-one update as

described in [Brand, 2006]. The rank is calculated by counting all singular values larger than 10^{-6} , in agreement with the definition of rank in [Freund *et al.*, 2015].

7 Discussion

We observe that the proposed method, RDFW, shows large improvements over FW and its common variants in terms of the rank and computing time. While ActiveALT, the leading nuclear norm based matrix completion solver, is very competitive with RDFW, RDFW has the additional attractive property that it does not require knowledge of the structure of the Hessian, or even require $f(\cdot)$ to be twice differentiable, allowing for greater generality. Moreover, since the matrix completion objective is quadratic, this is the ideal situation for ActiveALT since the underlying solver in ActiveALT only requires one Newton step to converge, whereas with a non-quadratic objective, the computational burden can increase.

8 Conclusions

We have proposed a rank-drop optimization formulation to determine optimally descent rank-drop steps for the nuclear-norm constrained minimization. By considering the interior and exterior cases separately to ensure feasibility, we also devise subproblems that can be efficiently solved. The proposed formulation can be deployed in a projection free minimization method, e.g., FW method, to efficiently compute a low rank solution by maintaining low rank intermediate iterates, without compromising the strong convergence guarantees. While classic FW methods tend to have very high rank solutions for nuclear-norm constrained problems, we have shown that the addition of rank-drop steps can drastically reduce the rank of the iterates, allowing for much faster algorithms to reach low rank solutions with less required space.

Acknowledgements

Authors acknowledge funding from the National Sciences and Engineering Research Council of Canada.

References

- [Brand, 2006] Matthew Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear algebra and its applications*, 415(1):20–30, 2006.
- [Egerváry, 1960] Eugen Egerváry. On rank-diminishing operations and their applications to the solution of linear equations. *Zeitschrift für angewandte Mathematik und Physik ZAMP*, 11(5):376–386, 1960.
- [Fazel *et al.*, 2001] Maryam Fazel, Haitham Hindi, and Stephen P Boyd. A rank minimization heuristic with application to minimum order system approximation. In *American Control Conference, 2001. Proceedings of the 2001*, volume 6, pages 4734–4739. IEEE, 2001.
- [Frank and Wolfe, 1956] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.
- [Freund *et al.*, 2015] Robert M Freund, Paul Grigas, and Rahul Mazumder. An extended frank-wolfe method with “in-face” directions, and its application to low-rank matrix completion. *arXiv preprint arXiv:1511.02204*, 2015.
- [Guélat and Marcotte, 1986] Jacques Guélat and Patrice Marcotte. Some comments on wolfe’s away step. *Mathematical Programming*, 35(1):110–119, 1986.
- [Hsieh and Olsen, 2014] Cho-Jui Hsieh and Peder A Olsen. Nuclear norm minimization via active subspace selection. In *ICML*, pages 575–583, 2014.
- [Jaggi, 2013] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435, 2013.
- [Lacoste-Julien and Jaggi, 2015] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504, 2015.
- [Wolfe, 1970] Philip Wolfe. Convergence theory in nonlinear programming. *Integer and nonlinear programming*, pages 1–36, 1970.
- [Yao *et al.*, 2016] Quanming Yao, James Kwok, et al. Efficient learning with a family of nonconvex regularizers by redistributing nonconvexity. *arXiv preprint arXiv:1606.03841*, 2016.