

Solving Separable Nonsmooth Problems Using Frank-Wolfe with Uniform Affine Approximations

Edward Cheung and Yuying Li

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

Abstract

Frank-Wolfe methods (FW) have gained significant interest in the machine learning community due to their ability to efficiently solve large problems that admit a sparse structure (e.g. sparse vectors and low-rank matrices). However the performance of the existing FW method hinges on the quality of the linear approximation. This typically restricts FW to smooth functions for which the approximation quality, indicated by a global curvature measure, is reasonably good.

In this paper, we propose a modified FW algorithm amenable to nonsmooth functions, subject to a separability assumption, by optimizing for approximation quality over all affine functions, given a neighborhood of interest. We analyze theoretical properties of the proposed algorithm and demonstrate that it overcomes many issues associated with existing methods in the context of nonsmooth low-rank matrix estimation.

1 Introduction

We are interested in solving problems of the form,

$$\min_{X \in \mathcal{D}} f(X) \text{ s.t. } \|X\|_{\text{tr}} \leq \delta$$

where the trace norm, $\|X\|_{\text{tr}} = \sum \sigma_i(X)$, is the sum of the singular values of X . This problem is well studied in the case where f is a smooth convex function. For example, in matrix completion, many efficient algorithms have been proposed, including Frank-Wolfe [Jaggi, 2011], active set methods [Hsieh and Olsen, 2014], and proximal methods [Parikh *et al.*, 2014].

Recently, there has also been interest in solving the trace norm constrained problem where the objective function is not differentiable, e.g.,

$$\min_{X: \|X\|_{\text{tr}} \leq \delta} f(X) \equiv L(X) + \lambda_1 \|X\|_1. \quad (1)$$

where $L(X)$ is an empirical loss function. Problem (1) has been found useful [Richard *et al.*, 2012] for sparse covariance estimation and graph link prediction, for which solutions are expected to exhibit simultaneously sparse and low-rank structure. For problem (1), proximal methods will likely fail to

scale due to the full SVD required at each iteration. In addition, the active set method in [Hsieh and Olsen, 2014] utilizes second order information, making it unclear how to develop a scalable algorithm when the function is not differentiable, let alone twice differentiable.

FW appears to be able to address these issues, only requiring first order information and one singular vector pair calculation per iteration. However, the linear minimization problem is no longer straightforward if the ℓ_1 norm is added to the constraint set. Thus, we propose a variant of the FW algorithm to address nonsmooth objectives, focusing especially on low-rank matrix estimation problems. Nondifferentiability in the objective function often leads to an unbounded curvature constant, and standard convergence analysis can no longer be applied. Moreover, it becomes unclear how to define the linear approximation appropriately since choosing an arbitrary subgradient often leads to inadequate local approximations, leading to poor empirical results.

To address these issues, we replace the traditional linear minimization problem by a Chebyshev uniform affine approximation. This modification allows for a well-defined linear optimization problem, even when the objective is nonsmooth or has unbounded curvature. To find the Chebyshev uniform affine approximation, we require the following *separability assumption* for the remainder of the paper.

Assumption 1.1. Assume that $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ can be *separated* into a sum of component functions, i.e., $f(X) = \sum_{i,j} f_{ij}(X_{ij})$, where each $f_{ij} : \mathbb{R} \rightarrow \mathbb{R}$.

We demonstrate experimentally that this carefully selected linear minimization leads to significant improvement over a variety of matrix estimation problems, such as sparse covariance estimation, graph link prediction, and ℓ_1 -loss matrix completion.

2 Background

2.1 Frank-Wolfe for Nonsmooth Functions

The FW algorithm is a first-order method for solving $\min_{x \in \mathcal{D}} f(x)$, where $f(x)$ is a convex function and \mathcal{D} is a convex and compact set [Frank and Wolfe, 1956]. The algorithm is motivated by replacing the objective function $f(x)$ with its first-order Taylor expansion and solving the first-order surrogate on the domain \mathcal{D} . Formally, the Frank-Wolfe

algorithm solves the following linear minimization problem at the iteration k ,

$$s^{(k)} := \arg \min_{s \in \mathcal{D}} f(x^{(k)}) + \langle s - x^{(k)}, f(x^{(k)}) \rangle.$$

The next iterate is then given by a convex combination of $s^{(k)}$ and $x^{(k)}$, which guarantees that the resulting iterate remains feasible, assuming the initial $x_0 \in \mathcal{D}$.

For smooth convex functions, the Frank-Wolfe algorithm is known to converge at a rate of $O(1/k)$. The convergence analysis relies on the concept of *curvature constant* [Clarkson, 2010; Jaggi, 2011], which measures the quality of the linear approximation.

Let f be a convex and differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, and let \mathcal{D} be a convex and compact subset of \mathbb{R}^n . Then, the curvature constant C_f is defined as

$$C_f := \sup_{\substack{x, s \in \mathcal{D} \\ \alpha \in [0, 1] \\ y = x + \alpha(s - x)}} \frac{1}{\alpha^2} (f(y) - f(x) - \langle y - x, \nabla f(x) \rangle).$$

When the value of C_f is large, it suggests that there are regions in \mathcal{D} where the local linear approximation is poor. It can be seen in [Clarkson, 2010; Jaggi, 2011] that the curvature constant is closely related to the Lipschitz constant of the gradient. However, if f is not differentiable, even for simple functions such as $f(x) = \lambda \|x\|_1$, it is easy to verify that C_f is unbounded regardless of the choice of subgradient. The convergence properties for nonsmooth functions becomes difficult to analyze, and in practice we see that using an arbitrary subgradient for FW typically performs poorly.

2.2 Existing Work

To motivate better local approximations, [White, 1993] employs an approximate subdifferential, $T(x, \epsilon)$, which considers all the subgradients for any y in an ϵ -neighborhood of x . This idea attempts to find an appropriate linearization of the objective that is good within a specified neighborhood of interest. However, arbitrary approximate subgradients can still perform poorly in practice if they are not chosen carefully. In [Ravi *et al.*, 2017], the idea is extended and the authors proposed a generalized curvature constant to best determine which approximate subgradient to choose. However, this leads to an optimization problem which is not generally easy to solve and it is not obvious how to extend these ideas to (1) efficiently.

In [Pierucci *et al.*, 2014] and [Argyriou *et al.*, 2014], the objective in (1) is replaced by a smoothed objective. The gradients of the smoothed objective are given by,

$$[G^{(k)}(X^{(k)})]_{ij} = \nabla L(X^{(k)}) + \begin{cases} \lambda \operatorname{sgn}(X_{ij}^{(k)}), & \text{if } |X_{ij}^{(k)}| \geq \mu \\ \frac{\lambda}{\mu} X_{ij}^{(k)}, & \text{if } |X_{ij}^{(k)}| < \mu. \end{cases}$$

for the Smoothed Composite Conditional Gradient (SCCG) algorithm in [Pierucci *et al.*, 2014], and

$$G^{(k)}(X^{(k)}) = \nabla L(X^{(k)}) - \frac{1}{\beta^{(k)}} X_k + \frac{1}{\beta^{(k)}} S(X^{(k)}, \lambda \beta^{(k)})$$

for the Hybrid Conditional Gradient with Smoothing (HCGS) algorithm in [Argyriou *et al.*, 2014], where

$$S(X^{(k)}, \lambda \beta^{(k)}) = \operatorname{sgn}(X) \odot \max\{|X| - \lambda \beta^{(k)}, 0\}$$

is the soft-thresholding operator. In both cases, the objective is smoothed by parameters μ and $\beta^{(k)}$ respectively, where the smooth approximation given is the best approximation across all $(1/\mu)$ -smooth (or resp. $1/\beta^{(k)}$ -smooth) functions. However, determining how smooth the approximation should be is not easy a priori and it often varies depending on the iterate. Empirically, we also observe that there is a nontrivial dependency between the smoothing parameters and convergence rate. At any given iteration, if the smoothing parameters are not set appropriately, the algorithm often makes no progress for many iterations.

In [Yao and Kwok, 2016], a nonsmooth generalization to the rank-one matrix pursuit is proposed which utilizes subgradients in the linearized subproblem. To ensure convergence, the rank-one update at each iteration is replaced by a rank- k variant where k is computed by taking however many leading singular vectors are required to ensure the solution to the rank- k subproblem is not too far from the subgradient in norm. Since the subgradients are typically not low rank, the number of singular vectors required can often be very large (possibly requiring a full SVD), and this approach can still fail to scale in similar ways to proximal methods.

The work in [Odor *et al.*, 2016] also considers Frank-Wolfe methods when the curvature constant is unbounded. However, the algorithm is specific to the phase retrieval problem for which the objective is still differentiable, simplifying the analysis.

Lastly, the Generalized Forwards-Backwards (GenFB) algorithm was introduced to solve (1) by alternating between proximal steps using the trace and ℓ_1 norm [Richard *et al.*, 2012]. As alluded to earlier, these algorithms tend to scale poorly due to the full SVD required at each iteration.

2.3 Achieving a better linear approximation

The previous work on FW for nonsmooth minimization (1) shares a common idea, i.e., finding a meaningful way to define an appropriate linear optimization subproblem in a scalable manner. We consider directly minimizing the approximation error over all possible affine functions over a neighborhood specified carefully for the Frank-Wolfe steps. We will show that under modest assumptions, the linear subproblems we propose will be simple to solve and do not rely on specifying the desired level of smoothness, as required by methods discussed in the previous section.

Definition 2.1. Given some $r > 0$, a **uniform affine approximation** to a function $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ is defined as $\ell(Y) = b + \langle Y - X, \xi \rangle$ where,

$$(\xi, b) \in \arg \min_{(\xi, b)} \max_{Y \in \bar{\mathcal{B}}_\infty(X, r)} |f(Y) - b - \langle Y - X, \xi \rangle|$$

and $\bar{\mathcal{B}}_\infty(X, r)$ is the closed element-wise infinity norm ball of radius r around X .

The above uniform affine approximation can be viewed as choosing an optimal affine approximation in a given neighborhood of X . The optimality is defined in the uniform

sense, by minimizing the maximum absolute deviation from the original objective f .

This motivates a natural variant of FW where, at each iteration, the linear subproblem using a subgradient is replaced with the uniform affine approximation. In particular, we can view the FW iterates as,

$$X^{(k+1)} = X^{(k)} - \alpha^{(k)}(X^{(k)} - S)$$

for some $S \in \mathcal{D}$. Thus, $X^{(k+1)} \in \bar{B}_\infty(X^{(k)}, \alpha^{(k)} \mathbf{diam}(\mathcal{D}))$ where $\mathbf{diam}(\mathcal{D}) = \max_{S \in \mathcal{D}} \|X^{(k)} - S\|_\infty$. For FW, this implies that we can restrict our attention to a neighborhood around the current iterate $X^{(k)}$ where the neighborhood has radius $\alpha^{(k)} \mathbf{diam}(\mathcal{D})$.

Specifically, we observe that in FW there exist step size schedules, e.g., $\alpha^{(k)} = 2/(k+2)$, which are independent of the current iterate and guarantee convergence. Our proposed approach is to assume that such a step size schedule is specified *a priori* and to use the uniform affine approximation for the FW subproblems. This allows the linear optimization subproblems to be defined a meaningful way that is related to the FW steps. Moreover, the subproblems no longer require f to have bounded curvature or even to be differentiable.

3 Frank-Wolfe with Uniform Approximations

For the proposed uniform approximation approach to be viable, it is important that the uniform affine approximation can be calculated efficiently. We begin by considering real-valued functions and Chebyshev approximations.

3.1 Chebyshev Approximations

Given a real-valued function f and an interval $[a, b] \subseteq \mathbf{dom}(f)$, the Chebyshev polynomial, denoted as $p_d(x)$, is a polynomial of degree not exceeding d that best approximates f on the interval $[a, b]$ in the uniform sense,

$$p_d(x) = \arg \min_{p \in \Pi_d} \max_{a \leq z \leq b} |f(z) - p(z)|$$

where Π_d is the set of polynomials of degree at most d .

The Chebyshev Equioscillation Theorem gives a simple characterization of the Chebyshev polynomial [Mayans, 2006]. Although the equioscillation theorem only applies to a function of one variable, below, we show it can also be applied when a function satisfies the separability Assumption 1.1, and we can construct the best uniform affine approximation by determining the best affine approximation on an interval for each component function.

Theorem 3.1. *Suppose $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ is a continuous function that satisfies Assumption 1.1. For a given $X^{(k)} \in \mathbb{R}^{m \times n}$ and $\tau > 0$, if $\ell_{ij}(Y_{ij})$ is the Chebyshev polynomial of degree 1 for f_{ij} over the interval $[X_{ij} - \tau, X_{ij} + \tau]$, then the function $\ell(Y) = \sum_{i=1}^m \sum_{j=1}^n \ell_{ij}(Y_{ij})$ is the uniform affine approximation to f .¹*

Thus, under Assumption 1.1, it suffices to find the Chebyshev polynomials for the component functions. Furthermore, if each f_{ij} is convex, then there exists a closed form solution for a linear Chebyshev polynomial [Davis, 1975].

¹Due to space limitations, the proofs will be omitted but will be available here: <https://arxiv.org/abs/1710.05776>.

4 FWUA and Convergence

Using the uniform affine approximation, we propose a FW variant with Uniform Approximations (FWUA), which is described in Algorithm 1. The function `update_tau` will be described in full in Section 4.1, where a specific update rule for τ will be required to guarantee convergence.

Algorithm 1 Frank-Wolfe with Uniform Approximations (FWUA)

Input: f : A function satisfying Assumptions 1.1 and 4.1
 \mathcal{D} : A convex and compact subset of $\mathbb{R}^{m \times n}$
 ϵ : Approximation threshold
 K : Max iteration count

- 1: Let $X^{(0)} \in \mathcal{D}$.
- 2: Let $\tau^0 \leftarrow \mathbf{diam}(\mathcal{D})$.
- 3: **for** $k = 0..K$ **do**
- 4: $\alpha^{(k)} \leftarrow \frac{2}{k+2}$
- 5: $\tau^{(k)} \leftarrow \text{update_tau}(k, \tau^{(k-1)}, \epsilon, f)$
- 6: $(\xi^{(k)}, b^{(k)}) \leftarrow \arg \min_{\xi, b} \max_{Y \in \bar{B}_\infty(X^{(k)}, \tau^{(k)})} |f(Y) - b - \langle Y - X, \xi \rangle|$
- 7: $S^{(k)} \leftarrow \arg \min_S b^{(k)} + \langle S - X^{(k)}, \xi_k \rangle$
- 8: $X^{(k+1)} \leftarrow X^{(k)} + \alpha^{(k)}(S^{(k)} - X^{(k)})$.
- 9: **end for**

To establish convergence, subsequently we make the following assumptions.

Assumption 4.1. *Assume that f satisfies Assumption 1.1. In addition, each component function f_{ij} has the following properties:*

- (a) f_{ij} is a convex, L_{ij} -Lipschitz continuous function.
- (b) f_{ij} is not differentiable on at most a finite set.
- (c) If f_{ij} is differentiable at a , then it is also twice differentiable at a .

While the above set of assumptions appears restrictive, our main goal in this work is to efficiently solve (1) in the context of the trace-norm constrained matrix estimation problem which has a combination of ℓ_1 and ℓ_2 loss/regularization, for which these assumptions are satisfied.

Definition 4.2. *Let f be satisfy Assumption 1.1 and 4.1. For a given $\tau > 0$, we define the **uniform slope function**, $m_{ij}(X_{ij}, \tau) : \mathbb{R} \times \mathbb{R}^+ \rightarrow \mathbb{R}$, which is the slope of the uniform affine approximation for f_{ij} on the interval $[X_{ij} - \tau, X_{ij} + \tau]$.*

The uniform slope function can be viewed as a surrogate for the gradient which does not rely on differentiability of f .

Theorem 4.3. *Let $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ be a function that satisfies Assumption 1.1 and 4.1 and \mathcal{D} be a convex and compact set. Given $\tau > 0$, let $m_{ij}(X_{ij}, \tau)$ be the uniform slope function for f_{ij} at X_{ij} , and let $X_{\min} = \min\{X_{ij} | X = (X_{1,1}, \dots, X_{ij}, \dots, X_{mn})^\top \in \mathcal{D}\}$. Let*

$$\hat{f}(X, \tau) := \sum_{i=1}^m \sum_{j=1}^n \int_{X_{\min}}^{X_{ij}} m_{ij}(x, \tau) dx + f_{ij}(X_{\min})$$

Then the following statements hold:

- (a) $\hat{f}(X, \tau)$ is convex in X ,
- (b) $\nabla_X \hat{f}(X, \tau)$ is L/τ -Lipschitz continuous w.r.t. the ℓ_∞ -norm, where L is the max. Lipschitz constant of all f_{ij} ,
- (c) $\max_{X \in \mathcal{D}} \left| \hat{f}(X, \tau) - f(X) \right| \leq mn(M+1)D\Delta_f\tau$,
where,

$M :=$ the maximum number of points

any f_{ij} is not differentiable at

$D := \text{diam}(\mathcal{D})$,

$$\Delta_f := \max \left\{ \begin{array}{l} \max_{\substack{i,j \\ X,Y \in \mathcal{D} \\ f''_{ij} \text{ exists at } X_{ij} \text{ and } Y_{ij}}} \left| \frac{f''_{ij}(X_{ij}) - f''_{ij}(Y_{ij})}{2} \right|, \\ \max_{\substack{i,j \\ X_{ij}, Y_{ij} \in [X_{\min} - \tau, X_{ij} + \tau] \\ g_{ij} \in \partial f_{ij}(X_{ij}) \\ h_{ij} \in \partial f_{ij}(Y_{ij})}} 2|g_{ij} - h_{ij}| \end{array} \right\} \quad (2)$$

For completeness, we provide a brief proof sketch for part (c). To prove the uniform error bound, we show that $m(x, \tau)$ can be expressed as a central-difference. Thus, when f is differentiable in a τ neighborhood, Lagrange Remainder Theorem can be used to guarantee that the error bound is $O(\tau)$. When f is not differentiable in a τ -neighborhood, we note that this only occurs at finitely many neighbourhoods of size 2τ . Thus, the total error contributed from these points is $O(\tau)$ as well.

Suppose the neighborhood size at iteration k is $\tau^{(k)}$. Theorem 4.3 states that as $\tau^{(k)} \rightarrow 0$, the sequence of uniform affine approximations generated by the FWUA algorithm uniformly converges to the original objective. In particular, given a sequence of neighborhood sizes $\{\tau^{(k)}\}$, we consider the sequence of smooth approximations given by

$$\hat{f}^{(k)}(X) = \sum_{i=1}^m \sum_{j=1}^n \int_{X_{\min}}^{X_{ij}} m_{ij}(x, \tau^{(k)}) dx + f_{ij}(X_{\min}). \quad (3)$$

Since $\hat{f}^{(k)}$ is differentiable with a $\frac{L}{\tau^{(k)}}$ -Lipschitz gradient, it can be shown that the curvature constant for $\hat{f}^{(k)}$ is bounded,

$$C_{\hat{f}^{(k)}} \leq \frac{L}{\tau^{(k)}} \text{diam}(\mathcal{D})^2$$

see, e.g., [Jaggi, 2011]. Thus, we can leverage standard Frank-Wolfe convergence argument while maintaining an upper bound on the approximation quality of the solution.

To make these concepts concrete, consider $f(X) = \|X\|_1$ and \mathcal{D} is the trace norm ball of radius δ . Then,

$$\hat{f}(X, \tau) = \sum_{ij} \left(\frac{X_{ij}^2}{2\tau} + \frac{\tau}{2} \right) \cdot \mathbf{1}_{|X_{ij}| < \tau} + |X_{ij}| \cdot \mathbf{1}_{|X_{ij}| \geq \tau}$$

where $\mathbf{1}$ is the indicator function. Figure 1 illustrates the component functions.

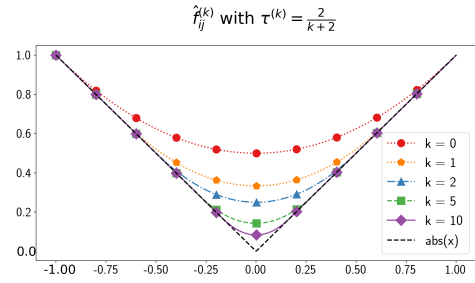


Figure 1: The component functions $\hat{f}_{ij}^{(k)}$ with $\tau^{(k)} = 2/(k+2)$ and varying k for $f(X) = \|X\|_1$.

When $f(X) = \|X\|_1$, its uniform affine approximation has an attractive property that $\hat{f} \geq f$. In general, this property may not hold. However from the uniform error bound, there always exists some value $N^{(k)} \in [0, n(M+1)D\Delta_f\tau^{(k)}]$ such that $\hat{f}^{(k)} + N^{(k)} \geq f$, and the function $\hat{f}^{(k)} + N^{(k)}$ has all the properties listed in Theorem 4.3, except the error bound in (2) becomes $\|\hat{f}^{(k)} - f\|_\infty \leq 2mn(M+1)D\Delta_f\tau^{(k)}$. Thus, we redefine the sequence of approximations as follows.

Definition 4.4. Let $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ be a function that satisfies Assumptions 1.1 and 4.1. The sequence of **FWUA smooth approximations** are given by

$$\tilde{f}^{(k)}(X) = \sum_{i,j} \int_{X_{\min}}^{X_{ij}} m_{ij}(x, \tau^{(k)}) dx + f_{ij}(X_{\min}) + N^{(k)} \quad (4)$$

where $N^{(k)} \geq 0$ is the smallest number such that $\tilde{f}^{(k)} \geq f$.

4.1 Update τ

The role of $\tau^{(k)}$ in the algorithm will be to reflect the maximum deviation in any component of $X^{(k)}$ after a FW step. That is, we are interested in bounding the quantity $\|X^{(k+1)} - X^{(k)}\|_\infty$. Since the FW steps take the form,

$$X^{(k+1)} = X^{(k)} + \alpha^{(k)}(S^{(k)} - X^{(k)})$$

for some $S^{(k)} \in \mathcal{D}$, this leads to a straightforward bound of

$$\|X^{(k+1)} - X^{(k)}\|_\infty \leq \alpha^{(k)} \text{diam}(\mathcal{D}).$$

However, setting $\tau^{(k)} = \alpha^{(k)} \text{diam}(\mathcal{D})$ can be overly conservative and lead to poor local approximations. Ideally, we would set $\tau^{(k)} = \|S^{(k)} - X^{(k)}\|_\infty$ to have the smallest value for $\tau^{(k)}$ which ensures that $S^{(k)}$ is in the neighborhood of interest. This is not possible since $S^{(k)}$ depends on the choice of $\tau^{(k)}$. In implementation, we estimate $\tau^{(k)}$ by the previous FW steps and consider the update,

$$\tau^{(k+1)} \leftarrow \alpha^{(k)} \max_{j \in \{0, \dots, 4\}} \|X^{(k-j)} - S^{(k-j)}\|_\infty. \quad (5)$$

Another concern with updating $\tau^{(k)}$ is that we cannot allow the function $\tilde{f}^{(k)}$ to become arbitrarily close to f , since the Lipschitz constant for $\tilde{f}^{(k)}$ can also grow arbitrarily large

given a nonsmooth f . However, if only an ϵ -accurate solution is desired with ϵ specified a priori, one can stop refining the approximation \hat{f} at some iteration k' since there exists an explicit upper bound on the approximation error. The FWUA algorithm can then proceed as a standard FW algorithm on the smoothed function defined by iteration k' .

Specifically, the uniform error bound from Theorem 4.3 and a step size of $\alpha^{(k)} = 2/(k+2)$ guarantees that

$$\frac{4mn(M+1)D^2\Delta_f}{k+2} \leq \frac{\epsilon}{2} \text{ when } k \geq \frac{8mn(M+1)D^2\Delta_f}{\epsilon} - 2.$$

At iteration $k' = \frac{8mn(M+1)D^2\Delta_f}{\epsilon} - 1$, we stop refining the neighborhoods and $\tau^{(k)} = \tau^{(k')}$ for all $k \geq k'$. The `update_tau` function is formalized in Algorithm 2.

Algorithm 2 `update_tau`

Input: k : Iteration number
 $\tau^{(k-1)}$: Previous neighborhood size
 ϵ : accuracy tolerance
 f : Original function to optimize
 $\{X^{(i)}\}_{i=0}^{k-1}$: Sequence of previous Frank-Wolfe iterates
 $\{S^{(i)}\}_{i=0}^{k-1}$: Solutions to the Frank-Wolfe linear subproblems of previous iterates
 $[m, n, M, D, \Delta_f] \leftarrow$ parameters of f as described in (2).
 1: $k' \leftarrow \frac{8mn(M+1)D^2\Delta_f}{\epsilon} - 1$.
 2: **if** $k > k'$ **then**
 3: **return** $\tau^{(k-1)}$
 4: **else**
 5: **return** $\tau^{(k)} \leftarrow \frac{2 \max_{j \in \{0, \dots, 4\}} \|X^{(k-j)} - S^{(k-j)}\|_\infty}{k+2}$
 6: **end if**

The rationale is that once the approximation is sufficiently accurate, i.e., the approximation error is no bigger than $\epsilon/2$, then standard Frank-Wolfe analysis can be applied to bound the suboptimality of the smoothed problem defined by iteration k' by $\epsilon/2$. In Theorem 4.5, we show that given an $\epsilon > 0$ a priori, Algorithm 1 requires at most $O(1/\epsilon)$ iterations to achieve a global error of ϵ , where $\tau^{(k)}$ is updated using Algorithm 2.

Theorem 4.5. *Let $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ satisfy Assumptions 1.1 and 4.1, $X^* \in \arg \min_{X \in \mathcal{D}} f(X)$, and let M, D, Δ_f be the constants defined in (2). Then for any $\epsilon > 0$, if*

$$k' := \left\lceil \frac{8mn(M+1)D^2\Delta_f}{\epsilon} - 1 \right\rceil, \text{ and}$$

$$K := \left(\max_{i \in \{0, \dots, k'\}} C_{\hat{f}^{(i)}} \right) + 8mn(M+1)D^2\Delta_f.$$

then

$$f(X^{(k)}) - f(X^*) < \epsilon$$

whenever $k \geq \frac{8K}{\epsilon}$.

We provide a proof sketch. The global error can be decomposed into the ‘‘approximation error’’ introduced by using a smooth approximation, $\|\hat{f}^{(k)} - f\|_\infty$, and the ‘‘suboptimality of the smooth problem’’, $\hat{f}^{(k)}(X^{(k)}) - \min_{X \in \mathcal{D}} \hat{f}^{(k)}(X)$,

which is the suboptimality of minimizing \hat{f} over \mathcal{D} at X^k . The uniform error bound in Theorem 4.3 identifies the iteration k' required to guarantee $\epsilon/2$ error, and a slightly modified FW convergence proof generalized to sequences of functions, can be used to determine when the suboptimality of the smooth problem is small.

5 Experimental Results

5.1 Sparse and Low-Rank Structure

To highlight benefits of the proposed FWUA, we first compare it against other state-of-the-art solvers for the problem,

$$\min_{X: \|X\|_{tr} \leq \delta} \|P_\Omega(X - Y)\|_F^2 + \lambda_1 \|X\|_1.$$

where Y is the given data, $\Omega = \{(i, j)\}$ is the set of observed indices, and $P_\Omega(\cdot)$ projects the loss onto Ω . For all experiments, the FW based methods terminate after 1000 iterations, and all other methods use default stopping criteria suggested by the authors.

We compare FWUA with GenFB [Richard *et al.*, 2012], HCGS [Argyriou *et al.*, 2014], and SCCG [Pierucci *et al.*, 2014]. For each problem instance, the same λ_1 value is used by all methods and this value is tuned, by searching over a grid of parameter values, to yield the best test performance for GenFB. The bound δ for the trace norm, is then set to the trace norm of the solution given by GenFB. For SCCG, the smoothing parameter μ is additionally tuned to yield the smallest average objective value. HCGS sets $\beta^{(k)} = 1/\sqrt{k+1}$ as suggested by the authors. We also compare the limiting behavior for SCCG when $\mu = 0$. This corresponds to a specific subgradient, denoted as SCCG (SG).²

Sparse Covariance Estimation

We follow the synthetic experiments described in [Richard *et al.*, 2012], where the goal is to recover a block diagonal matrix. We consider square matrices where $n = 750 : 250 : 2000$ (here we use MATLAB notation). The true underlying matrix is generated with 5 blocks, where the entries are i.i.d. and uniformly sampled from $[-1, 1]$. Gaussian noise, $\mathcal{N}(0, \sigma^2)$ is then added with $\sigma^2 = 0.2$. For this experiment, all entries in Ω are observed.

In Figure 2, we only present the convergence results for $n = 2000$ due to space limitations, but the patterns are similar throughout. We remark that since the GenFB algorithm is a regularized algorithm, the intermediate iterates are not feasible for the constrained problem used for FW. Only the performance of the solution at convergence is compared.

Graph Link Prediction

Next we consider predicting links in a noisy social graph. The input data is a matrix corresponding to an undirected graph, where the entry $A_{ij} = 1$ indicates that user i and j are friends and $A_{ij} = 0$ otherwise. We consider the Facebook dataset from [Leskovec and Krevl, 2014] which consists of a graph with 4,039 nodes and 88,234 edges, and assume 50% of the entries are observed. The goal is to recover the remaining

²The final parameters used are omitted due to space, but will appear in the appendix in <https://arxiv.org/abs/1710.05776>.

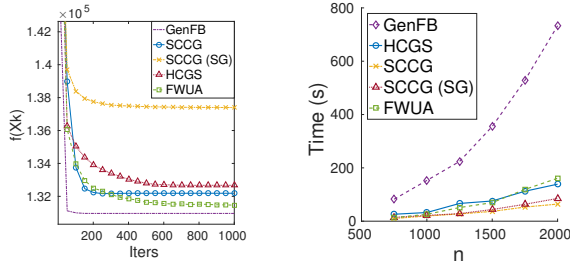
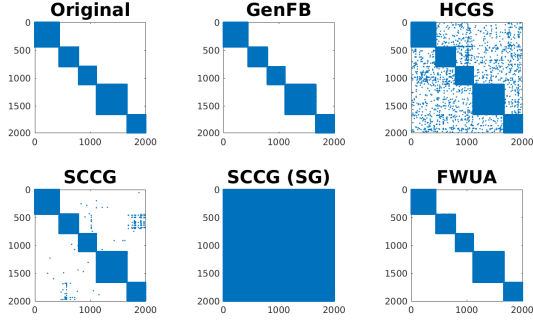


Figure 2: Sparse covariance estimation.


 Figure 3: An example sparsity pattern at convergence, values thresholded at $0.01\|X\|_\infty$.

edges in the graph. Additionally, each entry A_{ij} is flipped with probability $\sigma \in \{0, 0.05, 0.1\}$, potentially removing or adding labels to the graph. We report the AUC performance measure of the link prediction on the remaining entries of the graph as well as the average CPU time over 5 random initializations summarized in Table 1 across all levels of σ .

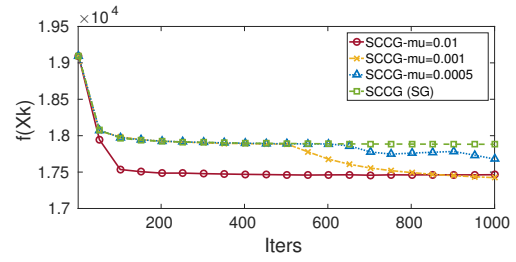
Discussion

For both applications, the results agree with our initial intuition that FWUA can improve the performance of the FW variants while scaling much better than the GenFB algorithm. We observe in the covariance plots in Figure 3, the sparsity patterns for HCGS and SCCG are much noisier than FWUA and in Table 1, the AUC for SCCG and HCGS methods are lower than GenFB and FWUA.

For SCCG, tuning the smoothing parameter μ yields theoretical tradeoffs between accuracy and convergence rates. Initially, we expected this tradeoff to be smooth, where gradually decreasing μ gradually worsened the rate of improvement per iteration. In return we expected small values of μ

σ		GenFB	SCCG	SCCG (SG)	HCGS	FWUA
0	AUC	0.968	0.949	0.873	0.868	0.972
	Time (s)	3746.30	1127.91	288.46	263.27	449.89
0.05	AUC	0.829	0.820	0.799	0.806	0.829
	Time (s)	4024.96	332.23	378.68	407.87	408.37
0.10	AUC	0.715	0.708	0.707	0.708	0.732
	Time (s)	4006.92	363.18	403.27	420.22	481.82

Table 1: Graph link prediction averaged over 5 random initializations for the Facebook dataset. Top performers are bolded.


 Figure 4: One trial with varying μ for sparse covariance estimation with $n = 750$. Here we see large delays as μ becomes smaller, almost making no progress for 700 iterations when $\mu = 0.0005$.

to yield better final solutions. Instead, we observed that the algorithm will make no progress for many iterations if μ is too small, as highlighted in Figure 4. We see that the delay observed increases as μ decreases. This seems to give support to the hypothesis for FWUA where the approximation quality must be closely related to the step size since it appears that SCCG can only make progress once the step sizes, following the step-size schedule $2/(k+2)$, becomes sufficiently small. Since HCGS and SCCG do not factor in step size into the smoothing schedule, we observe that empirically, the solutions returned from these methods are not as competitive as the FWUA or GenFB algorithm.

5.2 ℓ_1 Loss Matrix Completion

The last experiment we consider is matrix completion with an ℓ_1 loss function on the MovieLens datasets³. Here, we consider the objective function below

$$f(X) = \|P_\Omega(X - Y)\|_1 + \lambda \|P_{\Omega^c}(X)\|_F^2 \quad (6)$$

which is proposed in [Cambier and Absil, 2016] for robustness to outliers. Here the regularization penalizes entries in the complement of Ω , potentially preventing overfitting.

We compare with the Robust Low-Rank Matrix Completion (RLRMC) algorithm proposed in [Cambier and Absil, 2016], which solves a nonconvex fixed rank problem by the smoothing ℓ_1 term. We additionally compare to the Greedy Low-Rank Learning (GLRL) algorithm proposed in [Yao and Kwok, 2016], which greedily updates the solution with low-rank solutions found by computing the truncated SVD of a subgradient. For scalability, we utilize the Rank-Drop variant (RDFW) proposed in [Cheung and Li, 2017] for the FWUA algorithm, which empirically reduces the rank and computation time of the FW algorithm.

³<https://grouplens.org/datasets/movielens/>

Dataset		RLRMC	GLRL	FWUA
ML-100k	RMSE	0.892	0.935	0.876
	Time (s)	10.62	43.56	65.39
ML-1M	RMSE	0.817	0.917	0.812
	Time (s)	108.97	1,079.11	862.15
ML-10M	RMSE	0.810	0.901	0.801
	Time (s)	2,197.20	26,473.89	6,830.31

Table 2: Low-rank Matrix Completion averaged over 5 random initializations. Best performers are bolded.

Discussion

We observe that FWUA performs better than both RLRMC and GLRL in terms of out of sample RMSE, but RLRMC is much faster. This is not surprising since RLRMC is a non-convex fixed rank model. We observe that the performance of RLRMC is very sensitive to both the rank and λ parameters, requiring extensive parameter tuning to find reasonable results. Thus, FWUA can be an attractive alternative when a good estimate of the true rank is not known a priori. We also note that for the large scale example, the number of singular values required in the truncated SVD used by the GLRL updates became very high, leading to scalability issues.

6 Conclusion

We propose a variant of the Frank-Wolfe algorithm for a nonsmooth objective, by replacing the linear FW subproblem with one defined by the Chebyshev uniform affine approximation. We show that for nonsmooth matrix estimation problems, this uniform approximation is easy to compute and allows for convergence analysis without assuming a bounded curvature constant. Experimentally we demonstrate that the FWUA algorithm can improve both speed and classification performance in a variety of sparse and low-rank learning tasks, while providing a viable convex alternative for ℓ_1 loss matrix completion when little is known about the underlying data.

Acknowledgements

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

References

[Argyriou *et al.*, 2014] Andreas Argyriou, Marco Signoretto, and Johan AK Suykens. Hybrid conditional gradient-smoothing algorithms with applications to sparse and low rank regularization. In *Regularization, Optimization, Kernels, and Support Vector Machines*, pages 53–82. Chapman and Hall/CRC, 2014.

[Cambier and Absil, 2016] Léopold Cambier and P-A Absil. Robust low-rank matrix completion by riemannian optimization. *SIAM Journal on Scientific Computing*, 38(5):S440–S460, 2016.

[Cheung and Li, 2017] Edward Cheung and Yuying Li. Projection free rank-drop steps. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1539–1545, 2017.

[Clarkson, 2010] Kenneth L Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):63, 2010.

[Davis, 1975] Philip J Davis. *Interpolation and approximation*. Courier Corporation, 1975.

[Frank and Wolfe, 1956] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics (NRL)*, 3(1-2):95–110, 1956.

[Hsieh and Olsen, 2014] Cho-Jui Hsieh and Peder Olsen. Nuclear norm minimization via active subspace selection. In *International Conference on Machine Learning*, pages 575–583, 2014.

[Jaggi, 2011] Martin Jaggi. *Sparse Convex Optimization Methods for Machine Learning*. PhD thesis, ETH Zurich, October 2011.

[Leskovec and Krevl, 2014] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014.

[Mayans, 2006] Robert Mayans. The chebyshev equioscillation theorem. *Journal of Online Mathematics and Its Applications*, 6, 2006.

[Odor *et al.*, 2016] Gergely Odor, Yen-Huan Li, Alp Yurtsever, Ya-Ping Hsieh, Quoc Tran-Dinh, Marwa El Halabi, and Volkan Cevher. Frank-wolfe works for non-lipschitz continuous gradient objectives: Scalable poisson phase retrieval. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pages 6230–6234. Ieee, 2016.

[Parikh *et al.*, 2014] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[Pierucci *et al.*, 2014] Federico Pierucci, Zaid Harchaoui, and Jérôme Malick. A smoothing approach for composite conditional gradient with nonsmooth loss. 2014.

[Ravi *et al.*, 2017] Sathya N Ravi, Maxwell D Collins, and Vikas Singh. A deterministic nonsmooth frank wolfe algorithm with coresets guarantees. *arXiv preprint arXiv:1708.06714*, 2017.

[Richard *et al.*, 2012] Emile Richard, Pierre-André Savalle, and Nicolas Vayatis. Estimation of simultaneously sparse and low rank matrices. In *Proceedings of the 29th International Conference on International Conference on Machine Learning, ICML’12*, pages 51–58, USA, 2012. Omnipress.

[White, 1993] DJ White. Extension of the frank-wolfe algorithm to concave nondifferentiable objective functions. *Journal of optimization theory and applications*, 78(2):283–301, 1993.

[Yao and Kwok, 2016] Quanming Yao and James T Kwok. Greedy learning of generalized low-rank models. In *IJCAI*, pages 2294–2300, 2016.