STOCHASTIC BILEVEL PROJECTION-FREE OPTIMIZATION

Authors

Address - Line 1
Address - Line 2
Address - Line 3

ABSTRACT

Bi-level optimization is a powerful framework to solve a rich class of problems such as hyper-parameter optimization, model-agnostic meta-learning, data distillation, and matrix completion. The existing first-order solutions to bi-level problems exhibit scalability limitations (for example, in matrix completion) because of the requirement of projecting solutions onto the feasible set. In this work, we propose a novel **S**tochastic **B**i-level **F**rank-**W**olfe (SBFW) algorithm to solve the stochastic bi-level optimization problems in a projection-free manner. We utilize a momentum-based gradient tracker that results in a sample complexity of $\mathcal{O}(\epsilon^{-3})$ for convex outer objectives with strongly convex inner objectives. We formulate the matrix completion problem with denoising as a stochastic bilevel problem and show that SBFW outperforms the state-of-the-art methods for the problem of matrix completion with denoising and achieves improvements of up to 82% in terms of the wall-clock time required to achieve the same level of accuracy.

***Index Terms*—** Stochastic gradient, bilevel optimization.

1. INTRODUCTION

We consider the two-level hierarchical optimization problem

$$\min_{\mathbf{x}\in\mathcal{X}\subset\mathbb{R}^m} \mathbb{E}_\theta[f(\mathbf{x},\mathbf{y}^\star(\mathbf{x});\theta)], \tag{1a}$$

$$\text{s.t.} \quad \mathbf{y}^\star(\mathbf{x}) \in \arg\min_{\mathbf{y}\in\mathbb{R}^n} \mathbb{E}_\xi[g(\mathbf{y},\mathbf{x};\xi)]. \tag{1b}$$

Here, the outer problem involves minimizing the objective function $F(\mathbf{x},\mathbf{y}^\star(\mathbf{x})) := \mathbb{E}_\theta[f(\mathbf{y}^\star(\mathbf{x});\theta)]$ with respect to $\mathbf{x}$ over the convex compact constraint set $\mathcal{X}\subset\mathbb{R}^m$. Here $\mathbf{y}^\star(\mathbf{x})$ is a unique solution of the inner optimization problem, which for a given $\mathbf{x}$, entails minimizing the strongly convex function $G(\mathbf{y},\mathbf{x}) := \mathbb{E}_\xi[g(\mathbf{y},\mathbf{x};\xi)]$ with respect to optimization variable $\mathbf{y}$. The function $F(\cdot)$ and $G(\cdot)$ are the expected values of continuous and proper closed functions $f:\mathbb{R}^m\to\mathbb{R}$ and $g:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ with respect to independent random variables $\theta\in\Theta$ and $\xi\in\Xi$, respectively. Observe that for bilevel problems of type (1), the inner and outer problems are interdependent and cannot be solved in isolation. Yet, these problems arise in a number of areas, such as meta-learning [1], continual learning [2], reinforcement learning [3], and hyper-parameter optimization [4].

To solve such problems, the first-order stochastic approximation algorithms have been recently proposed [5, 6]. In some works, such as [7], the constraint set $\mathcal{X}$ in the outer optimization problem is taken to be $\mathcal{X}=\mathbb{R}^m$, resulting in a simpler unconstrained outer optimization problem. However, in applications such as meta-learning, personalized federated learning, and corsets [2], the constraint set $\mathcal{X}$ is a strict subset of $\mathcal{X}\subset\mathbb{R}^m$. The standard approach to deal with such constraint sets is to project the updates of the outer optimization problem onto $\mathcal{X}$ at every iteration. Though popular and widely used, the projected gradient approaches may not necessarily be practical, for instance, in cases where the projection sub-problem is too expensive to be solved at every iteration. The difficulties surrounding projection-based methods have motivated the development of projection-free algorithms that use the Frank-Wolfe (FW) updates. These FW-based algorithms only require solving a linear program over $\mathcal{X}$, which could be significantly cheaper than solving a non-linear projection problem, as in the case of ℓ_1-norm or nuclear norm ball constraints. Projection-free algorithms for single-level stochastic optimization algorithms are well-known, and state-of-the-art algorithms achieve a sample complexity of $\mathcal{O}(\epsilon^{-2})$ [8, 9]. These algorithms rely on a recursive gradient tracking approach that allows the samples to be processed sequentially and achieves variance reduction without the use of checkpoints or large batches. Motivated by these developments, we ask the following question:

"*Is it possible to develop efficient projection-free algorithms for bi-level stochastic optimization problems?*"

This work puts forth the **S**tochastic **B**i-level **F**rank-**W**olfe (SBFW) algorithm, which is the first projection-free algorithm for bi-level problems. Our main contributions are:

- We propose a novel projection-free SBFW algorithm, utilizing the idea of momentum-based gradient update [16] to track the gradient of the outer objective function. The proposed algorithm is able to achieve a sample complexity of $\mathcal{O}(\epsilon^{-3})$.
- We test the proposed algorithm on matrix completion and establish the efficacy of the proposed techniques

Reference	Projection Free	Problem Type	SFO Complexity (Outer)	SFO Complexity (Inner)
SFW [10]	yes	Single-Level	$\mathcal{O}(\epsilon^{-3})$	-
ORGFW [8]	yes	Single-Level	$\mathcal{O}(\epsilon^{-2})$	-
SFW^{++} [11]	yes	Single-Level	$\mathcal{O}(\epsilon^{-3})$	-
BSA [12]	no	Bi-Level	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-3})$
stocBiO [13], STABLE [6], MSTSA [14]	no	Bi-Level	$\mathcal{O}(\epsilon^{-2})$	$\mathcal{O}(\epsilon^{-2})$
TTSA [15]	no	Bi-Level	$\mathcal{O}(\epsilon^{-2.5})$	$\mathcal{O}(\epsilon^{-2.5})$
SUSTAIN [7]	no	Bi-Level	$\mathcal{O}(\epsilon^{-1.5})$	$\mathcal{O}(\epsilon^{-1.5})$
SBFW (proposed)	yes	Bi-Level	$\mathcal{O}(\epsilon^{-3})$	$\mathcal{O}(\epsilon^{-1.5})$

Table 1: Comparision of the stochastic first-order complexity (SFO) for the outer and inner optimization problems.

compared to state-of-the-art algorithms (cf. Sec. 4). We achieve an improvement of up to 82% in the computation time for the proposed algorithm as compared to state-of-the-art methods.

1.1. Related Works

A series of works proposed to solve the problem of the form (1) has appeared recently [5–7, 17]. The seminal works in [5, 12] proposed a class of double-loop approximation algorithms to iteratively approximate the stochastic gradient of the outer objective and incurred a sample complexity of $\mathcal{O}(\epsilon^{-2})$ in order to achieve the ϵ-stationary point. The double loop structure of these approaches made them impractical for large-scale problems; [12] required solving an inner optimization problem to a predefined accuracy, while [5] required a large batch size of $\mathcal{O}(\epsilon^{-1})$ at each iteration. To address this issue, various single-loop methods involving simultaneous updates of inner and outer optimization variables have been developed [5–7]. A single-loop two-time scale stochastic algorithm proposed in [15] incurred a sub-optimal sample complexity of $\mathcal{O}(\epsilon^{-2.5})$. This is further improved recently in [5–7], in which the authors have utilized the momentum-based variance reduction technique from [16] to obtain optimal convergence rates. While all of the above-mentioned works seek to solve (1), they all require a projection onto $\mathcal{X}$ at every iteration. In this work, we are interested in developing projection-free stochastic optimization algorithms for bi-level problems, which is still an open problem and the subject of the work in this paper. A comprehensive list of all existing related works is provided in Table 1.

1.2. Motivating Example

In general, for noise-free data, the data matrix in the matrix completion problem can be modeled as a low-rank matrix motivating the use of the nuclear norm constraint. Low-rank matrix completion problem arises in various applications such as image processing, multi-task learning, and collaborative filtering. However, under noisy observations, directly solving the matrix completion problem with just the nuclear norm constraints can result in sub-optimal performance [18]. Further, noise is present in many vision applications, and using only low-rank priors is insufficient to recover the underlying matrix. A common approach to tackle the noise is to apply a denoising algorithm as a pre-processing step. In general, however, it is necessary to apply some heuristics since denoising algorithms require access to the full matrix, which is not available in the pre-processing stage. Denoising is also impractical in online settings, where a random subset of the matrix entries is observed at every iteration. The bilevel optimization framework provides a way out, allowing the incorporation of denoising steps within the inner-level sub-problem. Mathematically, the bi-level matrix completion with denoising problem can be written as

$$\begin{aligned} \min_{\|\mathbf{X}\|_* \leq \alpha} \; & \frac{1}{|\Omega_1|} \sum_{(i,j)\in\Omega_1} (\mathbf{X}_{i,j} - \mathbf{Y}_{i,j})^2, \\ \text{s. t.} \quad & \mathbf{Y} \in \arg\min_{\mathbf{V}} \Big\{ \frac{1}{|\Omega_2|} \sum_{(i,j)\in\Omega_2} (\mathbf{V}_{i,j} - \mathbf{M}_{i,j})^2 \\ & \qquad + \lambda_1 \|\mathbf{V}\|_1 + \lambda_2 \|\mathbf{X} - \mathbf{V}\|_F^2 \Big\}, \end{aligned} \tag{2}$$

where $\mathbf{M} \in \mathbb{R}^{n\times m}$ is the given incomplete noisy matrix, $\|\mathbf{V}\|_1 := \sum_{i,j} |\mathbf{V}_{i,j}|$ is the sum-absolute-value (ℓ_1) norm, λ_1 and λ_2 are regularization parameters, and Ω_1 and Ω_2 represents the set of available entries at outer and inner level respectively. Note that the regularization over the discrepancy between $\mathbf{X}$ and denoised matrix $\mathbf{Y}$ results in bilevel formulation (2). A similar technique in deterministic settings is utilized in various other applications in machine learning and signal processing problems [19]. The problem in (2) is a special case of general formulation in (1) with $f(\mathbf{x}, \mathbf{y}^\star) := \sum_{i,j} (\mathbf{X}_{i,j} - \mathbf{Y}_{i,j})^2$ with $\mathbf{x} := \mathbf{X}$, $\mathbf{y}^\star := \mathbf{Y}$ and $g := \|\mathbf{X} - \mathbf{V}\|_F^2$ with $\mathbf{y} := \mathbf{V}$. However, when the entries are revealed in the form of randomly selected subsets $\Omega_1^t \subset \Omega_1$ and $\Omega_2^t \subset \Omega_2$ at every iteration, it becomes stochastic in nature. The main challenge here is due to the nuclear norm constraint, which makes it quite computationally expensive (sometimes even impractical) to solve (2) using projection-based bilevel algorithms. In Sec. 4, we will show experimentally that the proposed algorithm SBFW is best suited to address such challenges.

2. ALGORITHM DEVELOPMENT

We note that solving the bi-level optimization problem in (1) is NP-hard in general, but we restrict our focus to problems where the inner objective is continuously twice differentiable in $(\mathbf{x}, \mathbf{y})$ and also strongly convex w.r.t $\mathbf{y}$ with parameter

$\mu_g > 0$. Such an assumption is common in the related works [5, 6, 12] and ensures that $\mathbf{y}^\star(\mathbf{x})$ is unique for any $\mathbf{x} \in \mathcal{X}$. A stochastic projected gradient descent update to solve (1) can be written as

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{X}}\left[\mathbf{x}_t - \alpha h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t)\right], \tag{3}$$

where we can write the biased estimate $h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t)$ as

$$\begin{aligned} &h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) \\ &\quad = \nabla_{\mathbf{x}} f(\mathbf{x}_t, \mathbf{y}_t; \theta_t) - M(\mathbf{x}_t, \mathbf{y}_t; \tilde{\xi}_t) \cdot \nabla_{\mathbf{y}} f(\mathbf{x}_t, \mathbf{y}_t; \theta_t), \end{aligned} \tag{4}$$

where $\nabla_{\mathbf{x}} f(\mathbf{x}_t, \mathbf{y}_t; \theta_t)$ is an unbiased estimate of $\nabla_{\mathbf{x}} F(\mathbf{x}_t, \mathbf{y}_t)$, $\nabla_{\mathbf{y}} f(\mathbf{x}_t, \mathbf{y}_t; \theta_t)$ is an unbiased estimate of $\nabla_{\mathbf{y}} F(\mathbf{x}_t, \mathbf{y}_t)$, and $M(\mathbf{x}_t, \mathbf{y}_t; \tilde{\xi}_t)$ is a biased estimate of product $\nabla^2_{\mathbf{xy}} G(\mathbf{y}_t, \mathbf{x}_t) \cdot [\nabla^2_{\mathbf{yy}} G(\mathbf{y}_t, \mathbf{x}_t)]^{-1}$. Here, we have used ξ_t in LHS to highlight the fact that the hessian of the function $g(\cdot)$ is also random in nature. The term $M(\mathbf{x}_t, \mathbf{y}_t; \tilde{\xi}_t)$ is a biased estimation of $[\nabla^2_{\mathbf{yy}} G(\mathbf{y}_t, \mathbf{x}_t)]^{-1}$ with bounded variance. The explicit form of $M(\mathbf{x}_t, \mathbf{y}_t; \tilde{\xi}_t)$ is

$$\begin{aligned} M(\mathbf{x}_t, \mathbf{y}_t; \tilde{\xi}_t) =& \nabla^2_{\mathbf{xy}} g(\mathbf{y}_t, \mathbf{x}_t; \xi_{t,0}) \times \\ & \times \left[\frac{k}{L_g} \Pi_{i=1}^{l} \left(I - \frac{1}{L_g} \nabla^2_{\mathbf{yy}} g(\mathbf{y}_t, \mathbf{x}_t; \xi_{t,i})\right)\right], \end{aligned} \tag{5}$$

here, $\tilde{\xi}_t$ is a collection of $k+1$ i.i.d. samples i.e. $\tilde{\xi}_t := \{\xi_{t,i} : i \in \{0, 1, \cdots, k\}\}$, with $\xi_{t,0}$ being the sample of $\nabla^2_{\mathbf{xy}} g(\mathbf{y}_t, \mathbf{x}_t)$ and $\xi_{t,1}, \cdots, \xi_{t,k}$ being the i.i.d. samples of $\nabla^2_{\mathbf{yy}} g(\mathbf{y}_t, \mathbf{x}_t)$. Further, the parameter l is selected uniformly from $\{1, \cdots, k\}$ and for l=0, we use the convention $\Pi_{i=1}^{l}\left(I - \frac{1}{L_g}\nabla^2_{\mathbf{yy}} g(\mathbf{x}_t, \mathbf{y}_t; \xi_{t,i})\right) = I$. Here, I is the identity matrix, and L_g is the Lipschitz parameter for gradient $\nabla_{\mathbf{y}} g(\mathbf{x}, \mathbf{y})$.

Similar to the update in (3), a significant challenge that remains unaddressed to date in the literature for the bi-level problems is associated with the projection operator in (3). The projection is easy to evaluate when the constraint set is a simple convex set (onto which projection operation is computationally cheap such as probability simplex) or has a closed-form solution (set of unit-ball). However, the projection step is often computationally costly (e.g., nuclear norm constraint), and its complexity could be comparable to the problem at hand [20]. In this work, we alleviate this issue by proposing projection-free algorithms for bi-level optimization problems, which is the key novel aspect of our work.

2.1. Stochastic Projection-Free Bi-level Optimization

Before proceeding, we discuss the particular choice of $\mathbf{y}_t$ in (3). A popular choice (see [6, 13, 15]) for $\mathbf{y}_t$ is the stochastic gradient descent update for the inner optimization problem given by $\mathbf{y}_t = \mathbf{y}_{t-1} - \delta_t \nabla_{\mathbf{y}} g(\mathbf{y}_{t-1}, \mathbf{x}_{t-1}; \xi_t)$, where $\nabla_{\mathbf{y}} g(\mathbf{y}_{t-1}, \mathbf{x}_{t-1}; \xi_t)$ is the unbiased estimate of the gradient $\mathbb{E}_\xi\left[\nabla_{\mathbf{y}} g(\mathbf{y}_{t-1}, \mathbf{x}_{t-1}; \xi)\right]$, and $\delta_t > 0$ denotes the step size. Now we are ready to propose the first projection-free algorithm for bi-level stochastic optimization problems. We propose to use a conditional gradient method (CGM) based approach [20, 21] instead of calculating the projection in (3). That is, we solve a linear minimization problem to find a feasible direction $\mathbf{s}_t \in \mathcal{X}$ for a given stochastic gradient direction $h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t)$, given by, $\mathbf{s}_t := \arg\min_{\mathbf{s} \in \mathcal{X}} \mathbf{s}, h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t)$. This reduces the optimization problem of evaluating the projection operator in (3) to solving a linear program which is easier to solve in practice. Hence, the iterate in (3) gets modified to

Algorithm 1: **S**tochastic **B**i-level **F**rank **W**olfe

Input: $\mathbf{x}_1 \in \mathcal{X}, \mathbf{y}_1 \in \mathbb{R}^m, \eta_t, \delta_t, \rho_t, \beta_t$, and $\mathbf{d}_1 = h_1(\theta_1; \xi_1)$ using (4)

for $t = 2$ ***to*** T **do**

  Update approximate inner optimization solution

$$\mathbf{y}_t = \mathbf{y}_{t-1} - \delta_t \nabla_{\mathbf{y}} g(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}, \xi_t)$$

  Gradient tracking evaluate $h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t)$ and $h(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}; \theta_t, \xi_t)$ using (4) and compute

$$\begin{aligned} \mathbf{d}_t = (1-\rho_t)(&\mathbf{d}_{t-1} - h(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}; \theta_t, \xi_t) \\ &+ h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) \end{aligned}$$

  Evaluate feasible direction $\mathbf{s}_t = \arg\min_{\mathbf{s} \in \mathcal{X}} \mathbf{s}, \mathbf{d}_t$;

  Update solution $\mathbf{x}_{t+1} = (1-\eta_t)\mathbf{x}_t + \eta_t \mathbf{s}_t$

Output: $\mathbf{x}_{T+1}$ or $\hat{\mathbf{x}}$ selected uniformly from $\{\mathbf{x}_i\}_{i=1}^T$

$$\mathbf{s}_t := \arg\min_{\mathbf{x} \in \mathcal{X}} \mathbf{x}, h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t) \tag{6}$$

$$\mathbf{x}_{t+1} = (1-\eta_{t+1})\mathbf{x}_t + \eta_{t+1}\mathbf{s}_t, \tag{7}$$

where $\eta_t > 0$ is the step size. To this end, we would like to emphasize that naive use of $h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t)$ in (6) for the evaluation of $\mathbf{s}_t$ which is then used in (7) can result in the iterate divergence due to the non-vanishing variance of the gradient estimate [10]. The standard approach to deal with this issue is to use a biased gradient estimate with low variance instead of an unbiased one. For example, a mini-batch approximation is proposed in [21, 22] with linearly increasing batch size with iteration index. Such an approach runs into memory issues when utilized in practice. To address the issue of memory and iterate divergence, we took motivation from the momentum-based approach in [16] and propose to use the following gradient tracking scheme given by

$$\begin{aligned} \mathbf{d}_t =& (1-\rho_t)(\mathbf{d}_{t-1} - h(\mathbf{x}_{t-1}, \mathbf{y}_{t-1}; \theta_t, \xi_t)) \\ &+ h(\mathbf{x}_t, \mathbf{y}_t; \theta_t, \xi_t). \end{aligned} \tag{8}$$

We remark that such a tracking technique is recently utilized in [7] for projection-based bi-level optimization problems. However, in this work, our focus lies in developing

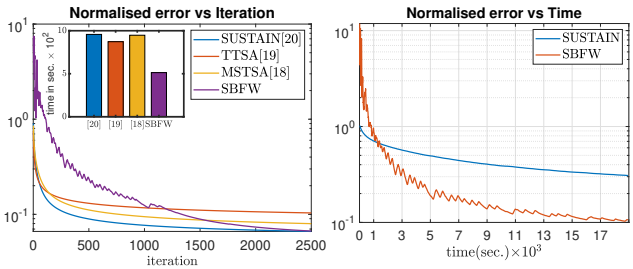

Fig. 1: Left: Comparision between the proposed SBFW algorithm for matrix completion problem on MovieLens 100k dataset with TTSA [15], MSTSA [14], and SUSTAIN [7]. Right: This figure compares the normalized error concerning computation time required for SBFW and SUSTAIN [14] on MovieLens 1M dataset.

projection-free algorithms, and hence analysis is significantly different from [7]. Our proposed algorithm is summarised in Algorithm 1.

3. CONVERGENCE ANALYSIS: SBFW

Assumption 1 *For some* $\sigma_{\mathbf{x}}^2 > 0$, $\sigma_{\mathbf{y}}^2 > 0$, $\sigma_{\mathbf{xy}}^2 > 0$, *and* $\sigma_g^2 > 0$ *we have* $\mathbb{E}[\{\nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{y},\theta)\}^2] \leq \sigma_{\mathbf{x}}^2$, $\mathbb{E}[\{\nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y},\theta)\}^2] \leq \sigma_{\mathbf{y}}^2$, $\mathbb{E}[\{\nabla_{\mathbf{xy}}^2 g(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{xy}}^2 g(\mathbf{x},\mathbf{y},\theta)\}^2] \leq \sigma_{\mathbf{xy}}^2$, $\mathbb{E}[\{\nabla_{\mathbf{y}} g(\mathbf{x},\mathbf{y}) - \nabla_{\mathbf{y}} g(\mathbf{x},\mathbf{y},\xi)\}^2] \leq \sigma_g^2$.

Assumption 2 *For any given* $\mathbf{x} \in \mathcal{X}$, *the terms* $\nabla_{\mathbf{x}} f(\mathbf{x},\mathbf{y})$, $\nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y})$, $\nabla_{\mathbf{y}} g(\mathbf{x},\mathbf{y})$, $\nabla_{\mathbf{xy}}^2 g(\mathbf{x},\mathbf{y})$ *and* $\nabla_{\mathbf{yy}}^2 g(\mathbf{x},\mathbf{y})$ *are Lipschitz continuous with respect to* $\mathbf{y}$ *with Lipschitz parameter* $L_{f_{\mathbf{x}}}$, $L_{f_{\mathbf{y}}}$, L_g, $L_{g_{\mathbf{xy}}}$ *and* $L_{g_{\mathbf{yy}}}$, *respectively. Similarly, for any given* $\mathbf{y} \in \mathbb{R}^n$, *the terms* $\nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y})$, $\nabla_{\mathbf{xy}}^2 g(\mathbf{x},\mathbf{y})$ *and* $\nabla_{\mathbf{yy}}^2 g(\mathbf{x},\mathbf{y})$ *are Lipschitz continuous with respect to* $\mathbf{x}$ *with positive constants* $L_{f_{\mathbf{y}}}$, $L_{g_{\mathbf{xy}}}$ *and* $L_{g_{\mathbf{yy}}}$, *respectively. Note that for the sake of simplicity, here we slightly abused the notation and used the same constants.*

Assumption 3 *For all* $\mathbf{x} \in \mathcal{X}$ *and* $\mathbf{y} \in \mathbb{R}^n$, *it holds that* $\mathbb{E}[\nabla_{\mathbf{y}} f(\mathbf{x},\mathbf{y})] \leq C_{\mathbf{y}}$ *and* $\mathbb{E}[\nabla_{\mathbf{xy}}^2 g(\mathbf{x},\mathbf{y})] \leq C_{\mathbf{xy}}$ *for some for constants* $C_{\mathbf{y}} > 0$ *and* $C_{\mathbf{xy}} > 0$.

Assumption 4 *The inner function* $g(\mathbf{x},\mathbf{y})$ *is* μ_g*-strongly convex in* $\mathbf{y}$ *for any* $\mathbf{x} \in \mathcal{X}$.

Theorem 1 *Consider the proposed Algorithm 1 with* $\delta_t = \frac{a_0}{t^{2/3}}$, *where* $a_0 = \min\{\frac{2}{3\mu_g}, \frac{\mu_g}{2(1+\sigma_g^2)L_g^2}\}$, $\rho_t = \frac{2}{t^{2/3}}$, $\eta_t = \frac{2}{t+1}$ *and* $k = \frac{2L_g}{3\mu_g}(\log(1+t))$, *then the output is feasible* $\mathbf{x}_{T+1} \in \mathcal{X}$ *and satisfies*

$$\mathbb{E}[Q(\mathbf{x}_{T+1}) - Q(\mathbf{x}^\star)] \leq \frac{12D\sqrt{C_1}}{5(T+1)^{\frac{1}{3}}} + \frac{2L_Q D^2}{(T+1)}. \quad (9)$$

here $L_Q = \frac{(L_{f_y}+L)C_{\mathbf{xy}}}{\mu_g} + L_{f_x} + C_y\left[\frac{L_{g_{\mathbf{xy}}}C_y}{\mu_g} + \frac{L_{g_{yy}}C_x y}{\mu_g^2}\right]$ *and* $C_1 = 3(\max\{2\mathbf{y}_1 - \mathbf{y}^\star(\mathbf{x}_1)^2, (2(C_{\mathbf{xy}}/\mu_g)^2 D^2 + 16a_0^2\sigma_g^2)/(2a_0 - 1)\} + \frac{C_{\mathbf{xy}}C_{\mathbf{y}}}{\mu_g} + 8(2L_k L_g^2 + L_k D^2 + \sigma_f^2))$.

Dataset	#users	#movies	#ratings	Time		
				SUSTAIN	SBFW	%imp.
Movielens 100k	1000	1700	10^5	959 sec.	433 sec.	55%
Movielens latest	600	9000	10^5	66.6 mins.	12.9 mins.	81%
Movielens 1M	6000	4000	10^6	10.16 hrs.	1.82 hrs.	82%

Table 2: Comparison of computation time of the proposed algorithm SBFW and the state-of-the-art projection-based algorithm SUSTAIN over the different sizes of real data sets.

The proof of Theorem 1 is available in an extended paper of this work at [23] as an extended version of this work. Theorem 1 shows that the optimality gap for SBFW decays as $\mathcal{O}(T^{-1/3})$ for general convex objectives, where T is the total number of iterations. We note that for at each iteration, SBFW requires $2k+1$ gradient samples to obtain gradient estimate: $2k$ samples for outer gradient estimate (8) and one sample for the inner variable update. Further, we have set $k \approx \mathcal{O}(\log(t))$. Hence, the SFO complexity of SBFW for the outer objective is $\mathcal{O}(\log(\epsilon^{-1})\epsilon^{-3}) \approx \mathcal{O}(\epsilon^{-3})$. Similarly, observe that $\mathbb{E}[\mathbf{y}_t - \mathbf{y}^\star(\mathbf{x}_t)] \leq \mathcal{O}((t+1)^{-q})$, where $q = 2/3$ (as $\delta_t = \mathcal{O}(t^{-2/3})$). Hence, the SFO complexity of the inner objective for SBFW is $\mathcal{O}(\epsilon^{-1.5})$. It can be seen that complexity for the inner level objective of the proposed algorithm SBFW is comparable to the projection-based state-of-the-art methods [6, 7, 13]; however, it shows slightly worse performance in terms of the outer level complexity. This is not surprising as we are tackling the outer level in a projection-free manner.

4. NUMERICAL EXPERIMENTS

In this section, first, we consider the problem of low-rank matrix completion formulated in (2) to illustrate the performance of our proposed SBFW algorithm. All the experiments are performed in MATLAB R2018a with Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz. To test the scalability of our proposed projection-free algorithm, we run an experiment over large-size matrices of MovieLens [1] datasets, which contain user ratings of movies ranging from 0 to 5. We start with Movielens 100k dataset of 10^5 ratings from 1000 users for 1700 movies. This dataset is denoted by observation matrix $\mathbf{M}$ of size 1000×1700. For the simulations, we define the set of observed entries Ω by sampling the matrix uniformly at random from $\mathbf{M}$ with a batch size of $b = 500$. We present the experimental results in Fig. 1. From Fig. 1(left), note that SBFW takes the least amount of time (shown in the inner bar plot) to attain the same level of accuracy. Also note that (Fig. 1(right)) even though SUSTAIN has better theoretical convergence guarantees, it performs poorly in terms of computation time required to achieve a particular level of normalized error.

In Table 2, observe that for large data sets, SBFW is approximately $10\times$ faster than the SUSTAIN and exhibits an improvement up to 82% in the computation time.

[1] https://grouplens.org/datasets/movielens/

5. REFERENCES

[1] Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine, "Meta-learning with implicit gradients," *arXiv preprint arXiv:1909.04630*, 2019.

[2] Zalán Borsos, Mojmir Mutny, and Andreas Krause, "Coresets via bilevel optimization for continual learning and streaming," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14879–14890, 2020.

[3] Haifeng Zhang, Weizhe Chen, Zeren Huang, Minne Li, Yaodong Yang, Weinan Zhang, and Jun Wang, "Bi-level actor-critic for multi-agent coordination," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 7325–7332.

[4] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil, "Bilevel programming for hyperparameter optimization and meta-learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1568–1577.

[5] Junjie Yang, Kaiyi Ji, and Yingbin Liang, "Provably faster algorithms for bilevel optimization," *arXiv preprint arXiv:2106.04692*, 2021.

[6] Tianyi Chen, Yuejiao Sun, and Wotao Yin, "A single-timescale stochastic bilevel optimization method," *arXiv preprint arXiv:2102.04671*, 2021.

[7] Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang, "A near-optimal algorithm for stochastic bilevel optimization via double-momentum," *arXiv preprint arXiv:2102.07367*, 2021.

[8] Jiahao Xie, Zebang Shen, Chao Zhang, Boyu Wang, and Hui Qian, "Efficient projection-free online methods with stochastic recursive gradient.," in *AAAI*, 2020, pp. 6446–6453.

[9] Zeeshan Akhtar and Ketan Rajawat, "Zeroth and first order stochastic frank-wolfe algorithms for constrained optimization," *IEEE Transactions on Signal Processing*, 2022.

[10] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi, "Stochastic conditional gradient methods: From convex minimization to submodular maximization," *Journal of Machine Learning Research*, vol. 21, no. 105, pp. 1–49, 2020.

[11] Mingrui Zhang, Zebang Shen, Aryan Mokhtari, Hamed Hassani, and Amin Karbasi, "One sample stochastic frank-wolfe," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 4012–4023.

[12] Saeed Ghadimi and Mengdi Wang, "Approximation methods for bilevel programming," *arXiv preprint arXiv:1802.02246*, 2018.

[13] Kaiyi Ji, Junjie Yang, and Yingbin Liang, "Bilevel optimization: Nonasymptotic analysis and faster algorithms," *arXiv preprint arXiv:2010.07962*, 2020.

[14] Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang, "A momentum-assisted single-timescale stochastic approximation algorithm for bilevel optimization," *arXiv e-prints*, pp. arXiv–2102, 2021.

[15] Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang, "A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic," *arXiv preprint arXiv:2007.05170*, 2020.

[16] Ashok Cutkosky and Francesco Orabona, "Momentum-based variance reduction in non-convex sgd," *arXiv preprint arXiv:1905.10018*, 2019.

[17] Feihu Huang and Heng Huang, "Enhanced bilevel optimization via bregman distance," *arXiv preprint arXiv:2107.12301*, 2021.

[18] Andrew D McRae and Mark A Davenport, "Low-rank matrix completion and denoising under poisson noise," *Information and Inference: A Journal of the IMA*, vol. 10, no. 2, pp. 697–720, 2021.

[19] Caroline Crockett and Jeffrey A Fessler, "Motivating bilevel approaches to filter learning: A case study," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 2803–2807.

[20] Martin Jaggi, "Revisiting frank-wolfe: Projection-free sparse convex optimization," in *International Conference on Machine Learning*. PMLR, 2013, pp. 427–435.

[21] Elad Hazan and Haipeng Luo, "Variance-reduced and projection-free stochastic optimization," in *International Conference on Machine Learning*. PMLR, 2016, pp. 1263–1271.

[22] Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola, "Stochastic frank-wolfe methods for nonconvex optimization," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2016, pp. 1244–1251.

[23] Zeeshan Akhtar, Amrit Singh Bedi, Srujan Teja Thomdapu, and Ketan Rajawat, "Projection-free algorithm for stochastic bi-level optimization," *arXiv preprint arXiv:2110.11721*, 2021.