Faster One-Sample Stochastic Conditional Gradient Method for Composite Convex Minimization

Gideon Dresdner
ETH Zürich, Switzerland

Maria-Luiza Vladarean
EPFL, Switzerland

Gunnar Rätsch
ETH Zürich, Switzerland

Francesco Locatello
Amazon Web Services

Volkan Cevher
EPFL, Switzerland

Alp Yurtsever
Umeå University, Sweden

Abstract

We propose a stochastic conditional gradient method (CGM) for minimizing convex finite-sum objectives formed as a sum of smooth and non-smooth terms. Existing CGM variants for this template either suffer from slow convergence rates, or require carefully increasing the batch size over the course of the algorithm's execution, which leads to computing full gradients. In contrast, the proposed method, equipped with a stochastic average gradient (SAG) estimator, requires only one sample per iteration. Nevertheless, it guarantees fast convergence rates on par with more sophisticated variance reduction techniques. In applications we put special emphasis on problems with a large number of separable constraints. Such problems are prevalent among semidefinite programming (SDP) formulations arising in machine learning and theoretical computer science. We provide numerical experiments on matrix completion, unsupervised clustering, and sparsest-cut SDPs.

1 INTRODUCTION

Consider the following composite finite-sum template:

$$\min_{w\in\mathcal{W}} \left\{ F(w) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(x_i^T w) + g(Aw) \right\}. \quad (1)$$

$\mathcal{W} \subset \mathbb{R}^d$ is a compact and convex set, each $f_i : \mathbb{R} \to \mathbb{R}$ is convex and L_f-smooth (i.e., its derivative is Lipschitz continuous with constant L_f), A is an $m \times d$ matrix, and $g : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is convex but possibly non-smooth. The function $g(Aw)$ can capture constraints of the form $Aw = b$ (or $Aw \in \mathcal{K}$, for closed, convex sets $\mathcal{K} \subseteq \mathbb{R}^m$) via indicator functions $\delta_{\{b\}}$ (resp., $\delta_{\mathcal{K}}$ which takes 0 for all points in $\mathcal{K}$ and $+\infty$ everywhere else). Throughout, we assume that g is either Lipschitz continuous or an indicator function.

We study conditional gradient methods (CGM, also known as the Frank-Wolfe Algorithm) tailored for Problem (1). For computational efficiency, we suppose linear minimization over $\mathcal{W}$ is easy. We separately focus on two specific settings of g:

- (S1) g admits an efficient prox-operator,
- (S2) g is a finite-sum of the form $g \triangleq \frac{1}{m}\sum_{i=1}^{m} g_i(a_i^T w)$, where each $g_i : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is convex and a_i^T is the i-th row of A. This *separable* finite-sum structure allows us to tackle g stochastically and therefore more efficiently when m is large.

Our problem template covers a variety of applications in machine learning, statistics and signal processing, including the finite-sum formulations that arise in M-estimation and empirical risk minimization problems.

Application Focus: Strongly Constrained SDPs

A particular example of our model problem is the standard semidefinite programming (SDP) template:

$$\begin{aligned} \min_{W\in\mathbb{S}_+^{d\times d}} \quad & \langle X, W\rangle \\ \text{subj. to} \quad & \langle A_i, W\rangle \triangleleft b_i, \quad i = 1,\dots,m, \end{aligned} \quad (2)$$

where $\mathbb{S}_+^{d\times d}$ denotes the set of symmetric positive semidefinite matrices, $X \in \mathbb{S}^{d\times d}$ is the symmetric cost matrix, $(A_i, b_i) \in \mathbb{S}^{d\times d} \times \mathbb{R}$ characterize the constraints, and '$\triangleleft$' represents either equality '$=$' or inequality '$\leq$' operations.

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

SDPs are ubiquitous in theoretical computer science. Examples include relaxations of combinatorial optimization problems such as maximum cut (Goemans and Williamson, 1995), quadratic assignment (Zhao et al., 1998), and sparsest cut (Arora et al., 2009). SDPs are also found in machine learning problems such as matrix completion (Alfakih et al., 1999), unsupervised clustering (Kulis et al., 2007), certifying robustness of neural networks (Raghunathan et al., 2018) and estimating their Lipschitz constants (Latorre et al., 2020).

The remarkable flexibility of SDPs comes at the cost of severe computational challenges. The cone constraint itself poses a major challenge for a majority of the first-order methods because projection onto positive semidefinite cone requires expensive eigen-decompositions. CGM is popular in this setting (see Hazan (2008); Jaggi and Sulovskỳ (2010); Garber (2016); Yurtsever et al. (2018)) since it avoids projection by leveraging the so-called linear minimization oracle (lmo) which computes only the top eigenvectors rather than the full spectrum. Additionally, CGM is also used to reduce storage cost (Yurtsever et al., 2015; Freund et al., 2017; Yurtsever et al., 2021), which is often a critical bottleneck for solving SDPs in large scale.

However, scalable approaches to solving SDPs with a large number of constraints, which we term as strongly-constrained SDPs, remain largely unexplored. This gap can be bridged by developing CGM variants which handle linear constraints in a randomized fashion.

Contributions. We propose a new CGM variant for convex finite-sum problems. The proposed method extends the recent work on stochastic Frank-Wolfe (Négiar et al., 2020) to the composite template in Problem (1). In particular:

- ▷ In (S1), our algorithm finds an ε-suboptimal solution after $\mathcal{O}(\varepsilon^{-2})$ iterations (see Optimality Conditions, Sec. 3 for the definition of ε-suboptimal).
- ▷ In (S2), our algorithm finds an ε-suboptimal solution after $\mathcal{O}(\varepsilon^{-2})$ iterations, matching the iteration complexity in Vladarean et al. (2020). However, we achieve this rate without using an increasing batch-size strategy. Thus, our algorithm enjoys a total cost of $\mathcal{O}(\varepsilon^{-2}d)$ which is independent of m. In contrast, the cost in Vladarean et al. (2020) is $\mathcal{O}(\varepsilon^{-2}dm)$.

Finally, we present numerical experiments on matrix completion, k-means clustering, and sparsest cut problems. In these experiments, the proposed algorithm performs on par with the state-of-the-art variance reduced CGM variants. Importantly, however, our algorithm does not require computing full gradients or increasing the batch size.

2 RELATED WORK

CGM for Smooth Objectives. CGM is introduced by Frank and Wolfe (1956) for minimizing a convex quadratic function over a polytope. Later, the analysis is extended to general convex smooth functions and arbitrary convex and compact sets by Levitin and Polyak (1966). Clarke (1990) and Hazan (2008) propose CGM as an effective method to tackle simplex and spectrahedron constraints respectively. We refer to Jaggi (2013) for an excellent survey on the efficiency of CGM for machine learning applications.

The last decade has witnessed a surge of interest in the CGM framework for machine learning applications which has prompted researchers to study stochastic extensions of CGM. Unlike gradient descent, CGM does not immediately work when the gradient in the algorithm is replaced with an unbiased stochastic gradient estimator with bounded variance. To address this problem, several stochastic CGM variants have been proposed by combining CGM with existing variance reduction techniques (Reddi et al., 2016; Hazan and Luo, 2017; Mokhtari et al., 2018; Yurtsever et al., 2019b; Shen et al., 2019; Zhang et al., 2020) and more recently in Négiar et al. (2020).

In general, the convergence rate of an algorithm is determined by the stochastic gradient estimator. Hazan and Luo (2017) develop an estimator with small variance, resulting in a fast $\mathcal{O}(\varepsilon^{-3/2})$ iteration complexity but at the cost of exponentially increasing batch sizes. Mokhtari et al. (2018) and Zhang et al. (2020) maintain a constant batch size but have slower convergence rates of $\mathcal{O}(\epsilon^{-3})$ and $\mathcal{O}(\epsilon^{-2})$, respectively. We refer to Yurtsever et al. (2019b) for a detailed comparison of the existing stochastic CGM variants.

Our work draws from (Négiar et al., 2020) where the authors propose a stochastic CGM with an iteration complexity of $\mathcal{O}(\varepsilon^{-1})$ which is on par with deterministic CGM. This is achieved by assuming a separable finite-sum model and using the Stochastic Average Gradient (SAG) estimation technique (Schmidt et al., 2017).

CGM for Composite Objectives. CGM is not directly applicable to problems with a non-smooth objective (see Section 2 in (Nesterov, 2018) for a counter-example). Lan (2013) tackle this problem in the case of Lipschitz continuous non-smooth functions by combining CGM with Nesterov smoothing (Nesterov, 2005). Yurtsever et al. (2018) further extend it for indicator functions through a quadratic penalty technique, which they call Homotopy CGM.

Locatello et al. (2019) extend Homotopy CGM to stochastic objectives but only for the case in which

Algorithm	Reference	Iteration Complexity	Total Cost	Fixed Batch Size
HCGM, CGAL	Yurtsever et al. (2018; 2019a)	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2} d \max\{n, m\})$	N/A
SHCGM	Locatello et al. (2019)	$\mathcal{O}(\varepsilon^{-3})$	$\mathcal{O}(\varepsilon^{-3} d\, m)$	N/A
MOST-FW	Akhtar and Rajawat (2021)	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2} d\, m)$	N/A
H-SAG-CGM v1	*This Paper*	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2} d\, m)$	N/A
H-1SFW	Vladarean et al. (2020)	$\mathcal{O}(\varepsilon^{-6})$	$\mathcal{O}(\varepsilon^{-6} d)$	✓
H-SPIDER-FW	Vladarean et al. (2020)	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2} d\, m)$	✗
MOST-FW$^+$	Akhtar and Rajawat (2021)	$\mathcal{O}(\varepsilon^{-4})$	$\mathcal{O}(\varepsilon^{-4} d)$	✓
H-SAG-CGM v2	*This Paper*	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2} d)$	✓

Table 1: This table presents asymptotic costs of finding an ε-suboptimal solution to a given problem, *i.e.*, we treat problem parameters d, n and m as constants and characterize the behavior as $\varepsilon \to 0$. $\mathcal{O}$ notation hides the parameters L_f, $\|A\|$, $D_{\mathcal{W}}$, and the absolute constants. We tailor the cost of existing methods for Problem (1), their cost for other problems can be different. The last column indicates whether the algorithm has increasing or fixed batch size.

the non-smooth part g is deterministic. More recently, Vladarean et al. (2020) proposed new variants that can handle stochastic constraints. They provide algorithms for an arbitrary number of constraints under minimal assumptions. However, for the common practical setting of a finite number of constraints, their algorithm requires full passes over the constraints.

This paper works in the same vein by proposing a randomized algorithm for the finite-sum template in Problem (1). Our algorithm for deterministic g in (S1) outperforms the method of Locatello et al. (2019) both in theory and in practice. Our algorithm for separable g in (S2) performs on par with the methods described in (Vladarean et al., 2020). However, in contrast to the previous work, it maintains a constant batch size.

After submitting this paper, we became aware of the recent work of Akhtar and Rajawat (2021). They study a similar problem and also propose an algorithm with two variants to address the cases of deterministic and stochastic g. In the case of deterministic g, the cost of their algorithm is $\mathcal{O}(\varepsilon^{-2}dm)$; the same as our method. However, in the case of stochastic g, their method's cost is $\mathcal{O}(\varepsilon^{-4}d)$. In contrast, our algorithm achieves $\mathcal{O}(\varepsilon^{-2}d)$ by taking advantage of the separable finite-sum structure.

Proximal Methods. A growing body of work aims to address strongly constrained problems through proximal methods in various settings (Patrascu and Necoara, 2017; Fercoq et al., 2019; Mishchenko and Richtárik, 2019; Xu, 2020). These algorithms process a random subset of constraints at each iteration and converge to a feasible point asymptotically, similar to (Vladarean et al., 2020) and the algorithm that we propose in this paper. However, when applied to SDPs, proximal methods require a costly eigenvalue decomposition at each iteration. Hence, these methods are not practical for solving SDPs in large scale.

Primal vs. Dual Problem. When there are many constraints, solving the dual problem can be more plausible from a computational perspective. However, converting a dual solution to a primal solution is a non-trivial problem itself, especially in large-scale setting where we are restricted from using projection or proximal operators. Moreover, since our problem is stochastic, we can expect finding only a rough estimate of the dual solution. In this work, we assume that we are interested in the primal variable and that it is large. To this end, we focus on solving the primal formulation.

3 PRELIMINARIES

Notation. The operator norm of a matrix A is written $\|A\|$ and the Euclidean inner-product is denoted $\langle \cdot, \cdot \rangle$. We define the diameter of $\mathcal{W}$ as

$$D_{\mathcal{W}} = \max_{x,y \in \mathcal{W}} \|x - y\|_2 \tag{3}$$

and the ℓ_1 and ℓ_∞ diameters with respect to the column space of a matrix M as

$$D_1(M) \triangleq \max_{u,v \in \mathcal{W}} \|M(u - v)\|_1 \tag{4}$$

$$D_\infty(M) \triangleq \max_{u,v \in \mathcal{W}} \|M(u - v)\|_\infty. \tag{5}$$

The linear minimization oracle of set $\mathcal{W}$ is given by

$$\text{LMO}_{\mathcal{W}}(v) \triangleq \arg\min_{u \in \mathcal{W}} \langle u, v \rangle. \tag{6}$$

The proximal operator of $g : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is

$$\text{prox}_g(z) \triangleq \arg\min_{y \in \mathbb{R}^m} \; g(y) + \frac{1}{2}\|y - z\|_2^2. \tag{7}$$

When g is the indicator function of a convex set $\mathcal{K}$, its proximal operator is equal to the Euclidean projection, $\text{prox}_{\delta_{\mathcal{K}}}(z) = \text{proj}_{\mathcal{K}}(z)$.

Assumption. When g is an indicator function we assume that strong duality holds. Slater's condition is a well-known sufficient condition for strong duality.

Optimality Conditions. We denote a solution of Problem (1) by $w^\star$:

$$F^\star \triangleq F(w^\star) \leq F(w), \quad \forall w \in \mathcal{W}. \tag{8}$$

If g is continuous valued on $\mathcal{W}$, we say that $w_k \in \mathcal{W}$ is an ε-suboptimal solution when it satisfies

$$\mathbb{E}F(w_k) - F^\star \leq \varepsilon. \tag{9}$$

If $g = \delta_{\mathcal{K}}$ is an indicator function, the $F(w_k) - F^\star$ can be $+\infty$ even when w_k is arbitrarily close to a solution. To this end, we relax the definition of an ε-suboptimal solution in this case and say that $w_k \in \mathcal{W}$ is an ε-suboptimal solution of Problem (1) if it satisfies

$$|\mathbb{E}f(w_k) - F^\star| \leq \varepsilon \quad \text{and} \quad \mathbb{E}[\operatorname{dist}(Aw_k; \mathcal{K})] \leq \varepsilon. \tag{10}$$

Our algorithm guarantees at every iteration that w_k is in $\mathcal{W}$ and asymptotically that $Aw_k \in \mathcal{K}$.

3.1 Smoothing

Building on the existing Homotopy CGM framework (Yurtsever et al., 2018; Locatello et al., 2019; Vladarean et al., 2020), we use the smoothing technique of Nesterov (2005) and its extension to indicator functions as studied in Tran-Dinh et al. (2018). Specifically, given a convex (possibly non-smooth) function g, its approximation is defined as

$$g_\beta(z) \triangleq \sup_y \langle y, z\rangle - g^*(y) - \frac{\beta}{2}\|y\|^2, \tag{11}$$

where $g^*(y) \triangleq \sup_x \langle x, y\rangle - g(x)$ is the Fenchel conjugate of g. Importantly, g_β is $\frac{1}{\beta}$-smooth (Nesterov, 2005). When $g = \delta_{\mathcal{K}}$ for some closed and convex set $\mathcal{K}$, its approximation becomes $g_\beta(z) = \frac{1}{2\beta}\operatorname{dist}(z, \mathcal{K})^2$. If g allows for an efficient prox operator, we can compute the gradient of g_β as

$$\nabla g_\beta(Aw) = \beta^{-1}\left(Aw - \operatorname{prox}_{\beta g}(Aw)\right). \tag{12}$$

4 ALGORITHM & CONVERGENCE

4.1 Stochastic Homotopy-Based CGM for Separable Problems

First, we transform the objective in Problem (1) using the smoothing technique summarized in Section 3.1 to obtain the following smooth surrogate objective:

$$F_\beta(w) \triangleq \frac{1}{n}\sum_{i=1}^n f_i(x_i^T w) + g_\beta(Aw). \tag{13}$$

Algorithm 1 H-SAG-CGM

1: **Input:** $\beta_0 > 0$, $w_0 \in \mathcal{W}$, $\alpha_0 \in \mathbb{R}^n$, $\gamma_0 \in \mathbb{R}^m$, $v_0^f \in \mathbb{R}^d$, $v_0^g \in \mathbb{R}^d$
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad \eta_k = \frac{2}{k+1}$
4: $\quad \beta_k = \beta_0/\sqrt{k+1}$
5: $\quad$ Sample $j \sim \text{Uniform}[1, 2, \ldots, n]$
6: $\quad \alpha_{k,i} = \begin{cases} \frac{1}{n} f_j'(x_j^T w_k) & i = j \\ \alpha_{k-1,i} & i \neq j \end{cases}$
7: $\quad v_k^f = v_{k-1}^f + (\alpha_{k,j} - \alpha_{k-1,j}) x_i$
8: $\quad v_k^g \leftarrow$ **use Variant 1 or Variant 2**
9: $\quad v_k = v_k^f + v_k^g$
10: $\quad s_k = \text{LMO}_{\mathcal{W}}(v_k)$
11: $\quad w_{k+1} = w_k + \eta_k(s_k - w_k)$
12: **end for**

Variant 1 Non-separable Constraints

1: **return** $\frac{1}{\beta_k} A^T(Aw_k - \operatorname{prox}_{\beta_k g}(Aw_k))$

Variant 2 Randomized Constraints

1: Sample $l \sim \text{Uniform}[1, 2, \ldots, m]$
2: $\gamma_{k,q} = \begin{cases} \frac{1}{m} g'_{\beta_k, l}(a_l^T w_k) & q = l \\ \gamma_{k-1,q} & q \neq l \end{cases}$
3: **return** $v_{k-1}^g + (\gamma_{k,l} - \gamma_{k-1,l}) a_l$

In particular, if we consider (S2) in which g is separable, then the smooth approximation g_β is also separable:

$$g_\beta(Aw) = \frac{1}{m}\sum_{j=1}^m g_{\beta,j}(a_j^T w). \tag{14}$$

This will allow for a fully randomized algorithm (H-SAG-CGM/v2) which can tackle strongly constrained SDPs with a non-increasing batch size.

The fundamental mechanism of homotopy CGM is to enforce a theoretically-determined schedule for β_k such that $F_{\beta_k} \to F$ asymptotically. Broadly speaking, stochastic homotopy CGMs perform these three steps at each iteration:

(1) Compute a gradient estimator v_k of the smooth surrogate function F_{β_k} (lines 5-8 of Alg. 1, implemented in Variant 1 and Variant 2).

(2) Perform a conditional gradient update by solving $\text{LMO}_{\mathcal{W}}(v_k)$ (Alg. 1, line 10) and moving the current estimate towards this solution (Alg. 1, line 11).

(3) Decrease β_k to enforce feasibility (line 4) and go to Step (1).

The main contribution of our algorithm is Step (1) where we use a SAG estimator for f and either the full gradient of g_{β_k} (Variant 1) or another SAG estimator

for g_{β_k} (Variant 2). This key innovation over previous work in Vladarean et al. (2020) yields comparable, state-of-the-art complexity bounds without requiring full passes over the set of constraints. Then, Step (2) comes from the classical CGM and Step (3) is the homotopy smoothing step from Yurtsever et al. (2018).

In the following section, we give an overview of the theoretical analysis.

4.2 Analysis of Stochastic homotopy CGMs

The analysis is composed of two main parts. First, we establish the convergence rate for the smoothed-gap

$$S_{\beta_k}(w_{k+1}) \triangleq \mathbb{E}[F_{\beta_k}(w_{k+1}) - F^\star]. \tag{15}$$

Then, in the second part that we present in Section 4.4, we translate convergence of the smoothed-gap S_{β_k} into guarantees for the original problem based on the techniques described in (Tran-Dinh et al., 2018).

For the first part, we rely on a recursive inequality involving $S_{\beta_k}(w_{k+1})$ which appears with slight variations in (Locatello et al., 2019; Vladarean et al., 2020). A generic version of this lemma is presented below.

Lemma 4.1. *For both variants of H-SAG-CGM, and for all $k \geq 1$ it holds that*

$$\begin{aligned} S_{\beta_k}(w_{k+1}) \leq (1-\eta_k) S_{\beta_{k-1}}(w_k) \\ + \eta_k D_{\mathcal{W}} \mathbb{E}\|\nabla F_{\beta_k}(w_k) - v_k\| \\ + \frac{\eta_k^2 D_{\mathcal{W}}^2 L_{F_{\beta_k}}}{2}, \end{aligned}$$

where $L_{F_{\beta_k}}$ represents the smoothness constant of the surrogate objective F_{β_k}. If we consider the setting (S1) and Variant 1 of the algorithm, then $L_{F_{\beta_k}} = \frac{\|X\| L_f}{n} + \frac{\|A\|}{\beta_k}$. Otherwise, if g is separable as in (S2) and we use Variant 2, then $L_{F_{\beta_k}} = \frac{\|X\| L_f}{n} + \frac{\|A\|}{\beta_k m}$.

See Appendix B for the proof.

Discussion. Lemma 4.1 shows how the convergence rate depends on the variance of the stochastic gradient estimator and the design parameters η_k and β_k. Since we can choose η_k and β_k to get the best possible rates in the analysis, this leaves the variance of the stochastic gradient estimator as the decisive term. By combining this lemma with two gradient estimators, corresponding to Variants 1 and 2 of Algorithm 1, we get convergence rates on S_{β_k} which we present in Theorem 4.1.

Existing stochastic homotopy CGMs (Locatello et al., 2019; Vladarean et al., 2020) rely on variance-reduced gradient estimators devised to handle arbitrary stochastic objectives, thus failing to exploit the separable finite-sum structure often encountered in practice.

Recently, Négiar et al. (2020) showed that optimal convergence guarantees can be obtained for separable objectives by considering a SAG-like gradient estimator (Schmidt et al., 2017). By combining this idea with the homotopy framework, we are able to provide an improved randomized algorithm (in two variants) for composite objectives. We now proceed by defining the SAG estimators and presenting their useful properties.

4.3 Stochastic Average Gradient (SAG) Error Bounds

The following two SAG estimators approximate the two parts of the gradient of F_β.

At each iteration of Algorithm 1, the j-th coordinate of the gradient of f is updated using a SAG estimator (lines 5-7):

$$\alpha_{k,i} = \begin{cases} \frac{1}{n} f'(x_i^T w_k) & i = j, \\ \alpha_{k-1,i} & i \neq j, \end{cases} \tag{16}$$

In particular, if we consider setting (S2) with a separable g, then we can use Variant 2 of the algorithm which employs another SAG estimator and updates the l-th coordinate of the gradient of g_{β_k}:

$$\gamma_{k,q} = \begin{cases} \frac{1}{m} g'_{\beta_k,l}(a_l^T w_k) & q = l, \\ \gamma_{k-1,q} & q \neq l. \end{cases} \tag{17}$$

Otherwise, in setting (S1) with a non-separable non-smooth g, we use full gradients of g_{β_k} as in Variant 1.

In summary, Variant 1 assumes stochastic ∇f approximated by α_k and a non-separable g whose gradient is fully computed. Thus, the stochastic gradient is a sum of a stochastic and deterministic terms: $v_k = X^T \alpha_k + A^T \nabla g_{\beta_k}(A w_k)$.

On the other hand, Variant 2 assumes that g separable in addition to f, hence ∇g_{β_k} can be approximated by γ_k. Thus, the overall gradient term v_k in this case is the sum of two stochastic terms given by $v_k = X^T \alpha_k + A^T \gamma_k$.

We now present two lemmas characterizing the errors of α_k and γ_k in ℓ_1-norm.

Lemma 4.2. *[Lemma 3 in (Négiar et al., 2020)] Consider H-SAG-CGM, with the SAG estimator α_k defined in (16). Then, for all $k \geq 2$,*

$$\begin{aligned} \mathbb{E}\left[\|\nabla f(X w_k) - \alpha_k\|_1\right] \leq \left(1 - \tfrac{1}{n}\right)^k \|\nabla f(X w_0) - \alpha_0\|_1 \\ + C_1 \left(1 - \tfrac{1}{n}\right)^{k/2} \log k + \frac{C_2}{k}, \end{aligned}$$

where $C_1 = 2n^{-1} L_f D_1(X)$, $C_2 = 4n^{-1}(n-1) L_f D_1(X)$ and the expectation is taken over all previous steps in the algorithm.

Lemma 4.3. *Consider Variant 2 of H-SAG-CGM with the SAG estimators defined in (16) and (17). Then, for all $k \geq 2$,*

$$\mathbb{E}[\|\nabla g_{\beta_k}(Aw_k) - \gamma_k\|_1] \leq \left(1 - \tfrac{1}{m}\right)^k \|\nabla g_{\beta_0}(Aw_0) - \gamma_0\|_1 + \frac{C}{\sqrt{k}}$$

where $C = 10\beta_0^{-1} D_1(A)$ and the expectation is taken over all previous steps of the algorithm.

We refer to (Négiar et al., 2020) for the proof of Lemma 4.2. We present the proof of Lemma 4.3 in Appendix D under the assumption that g is an indicator function or a Lipschitz continuous function.

Discussion. Lemma 4.2 shows that the SAG-like estimator provides an error bound in ℓ_1-norm that decays as $\mathcal{O}(1/k)$ in expectation. This decay does not carry over to the separable case in Variant 2, as demonstrated by Lemma 4.3, due to the $\frac{1}{\beta_k}$-factor associated with the smoothed approximation g_{β_k}.

4.4 Convergence Rates

Combining Lemmas 4.2 and 4.3 with Lemma 4.1 gives the convergence rates for the two variants of H-SAG-CGM which we now present.

Theorem 4.1. *The sequence generated by H-SAG-CGM (Algorithm 1) satisfies, for all $k \geq 2$,*

$$S_{\beta_k}(w_{k+1}) \leq \frac{C_1}{\sqrt{k}} + \frac{C_2}{k} + \frac{C_3}{k^2}.$$

The constants are defined for H-SAG-CGM/v1 as follows:

- $C_1 = 2D_{\mathcal{W}}^2 \|A\| \beta_0^{-1}$
- $C_2 = 8L_f D_1(X) D_\infty(X) + 2n^{-1} L_f \|X\| D_{\mathcal{W}}^2$
- $C_3 = 2n^2 D_\infty(X)\big(\|\nabla f(Xw_1) - \alpha_0\|_1 + 32 L_f D_1(X)\big)$

and for Variant 2 as follows:

- $C_1 = \beta_0^{-1}(2D_{\mathcal{W}}^2\|A\| + 10 D_1(A))$.
- $C_2 = 8L_f D_1(X) D_\infty(X) + 2n^{-1} L_f \|X\| D_{\mathcal{W}}^2$
- $C_3 = 2n^2 D_\infty(X)\big(\|\nabla f(Xw_1) - \alpha_0\|_1 + 32 L_f D_1(X)\big) + 2m^2 D_\infty(A)\|\nabla g_{\beta_0}(Aw_1) - \gamma_0\|_1$

Using the techniques described in (Tran-Dinh et al., 2018), we translate this bound to convergence guarantees on the original problem in the following corollaries.

Corollary 4.1. *Suppose $g : \mathbb{R}^m \to \mathbb{R}$ is L_g-Lipschitz continuous. Then, the estimates generated by H-SAG-CGM (Algorithm 1) satisfy*

$$\mathbb{E}[F(w_{k+1}) - F^\star] \leq \frac{C_1}{\sqrt{k}} + \frac{C_2}{k} + \frac{C_3}{k^2} + \frac{\beta_0 L_g^2}{2\sqrt{k}}$$

where the constants C_1, C_2 and C_2 are defined in Theorem 4.1.

Corollary 4.2. *Suppose g is the indicator function of a closed and convex set $\mathcal{K}$. Then, for H-SAG-CGM (Algorithm 1), we have a lower bound on the suboptimality as $\mathbb{E}\left[f(Xw_{k+1}) - f(Xw^*)\right] \geq -\|y^*\| \mathbb{E}\left[\operatorname{dist}(Aw_{k+1}, \mathcal{K})\right]$ and the following upper bounds on the suboptimality and feasibility:*

$$\mathbb{E}\left[f(Xw_{k+1}) - f(Xw^*)\right] \leq \frac{C_1 + \beta_0}{\sqrt{k}} + \frac{C_2}{k} + \frac{C_3}{k^2}, \quad \text{and}$$

$$\mathbb{E}\left[\operatorname{dist}(Aw_{k+1}, \mathcal{K})\right] \leq \frac{C_4}{\sqrt{k}} + \frac{\sqrt{2C_2}}{k^{3/4}} + \frac{\sqrt{2C_3}}{k^{5/4}}$$

where the constants C_1, C_2 and C_3 are defined in Theorem 4.1 and $C_4 = (\frac{3\beta_0\|y^*\|}{2} + \sqrt{2C_1})$.

Discussion. Even in the deterministic setting studied in (Yurtsever et al., 2018), the convergence rates of Homotopy CGM is bounded below by $\Omega(1/\sqrt{k})$, as demonstrated theoretically in (Lan, 2013) and practically in (Kerdreux et al., 2021). Corollaries 4.1 and 4.2 show that both variants of our algorithm achieves this lower bound.

H-SAG-CGM/v1 provides an order of magnitude improvement (from $\mathcal{O}(\varepsilon^{-3})$ to $\mathcal{O}(\varepsilon^{-2})$) over the previous state-of-the-art in deterministic constraints, Locatello et al. (2019).

While H-SAG-CGM/v2 and H-SPIDER-FW Vladarean et al. (2020) enjoy a similar overall rate, the latter requires an exponentially increasing batch size. Combined with occasional full passes, this quickly becomes impractical for strongly constrained problems. As an alternative, Vladarean et al. (2020) propose H-1SFW which does use a fixed batch size but at the cost of an impractical $\mathcal{O}(\varepsilon^{-6})$ rate. In stark contrast, our algorithm enjoys the optimal rate without resorting to increasing the batch size.

5 NUMERICAL EXPERIMENTS

This section demonstrates the empirical performance of the proposed method across a number of different problems: matrix completion, k-means clustering and uniform sparsest cut. We performed these experiments in MATLAB R2019b and the codes are publicly available at `https://github.com/ratschlab/faster-hcgm-composite`.

Baselines. We compare the proposed method against the following methods:

- SHCGM (Locatello et al., 2019)
- H-SPIDER-FW (Vladarean et al., 2020)
- H-1SFW (Vladarean et al., 2020)

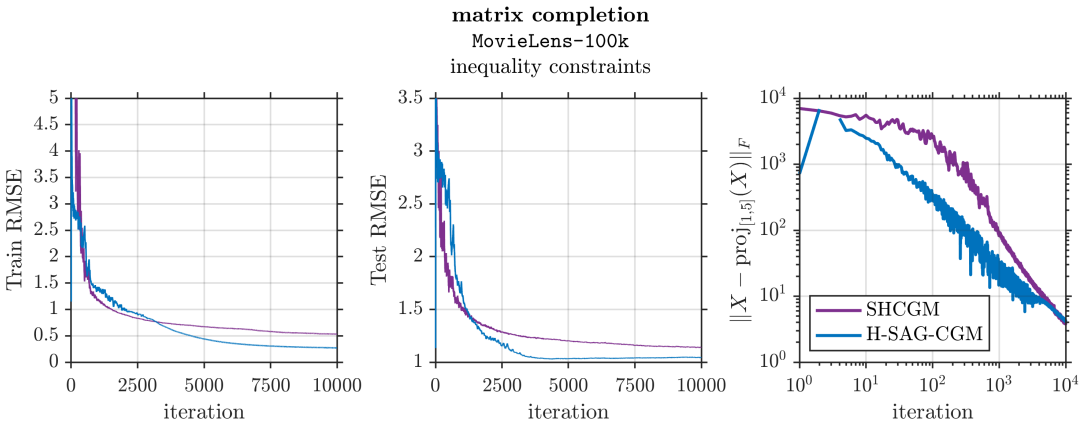

Figure 1: Empirical comparison of H-SAG-CGM/v1 with SHCGM on matrix completion with inequality constraints (18) with the `MovieLens-100k` dataset.

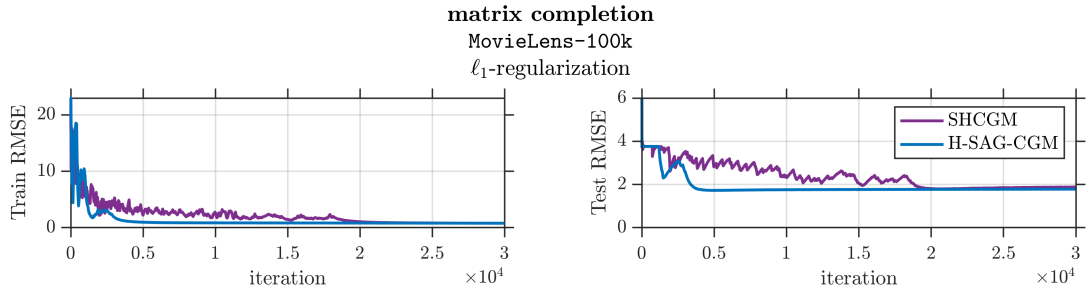

Figure 2: Empirical comparison of H-SAG-CGM/v1 with SHCGM on matrix completion with ℓ_1-regularization (19) with the `MovieLens-100k` dataset.

Note that SHCGM only works in the case of deterministic g and is hence a natural baseline comparison for H-SAG-CGM/v1. H-SPIDER-FW can handle stochastic g so it is used to compare to H-SAG-CBM/v2 but importantly in this case, H-SPIDER-FW requires an increasing batch size.

Challenges. The parameter β determines a trade-off between convergence in the objective residual and the infeasibility error. However, since we do not know the optimal value a priori, β_0 is not always easy to interpret given a particular task. This leaves practitioners to develop the intuition on how to tune this parameter. This challenge is not unique to H-SAG-CGM but is shared among homotopy CGM approaches (Yurtsever et al., 2018; Locatello et al., 2019; Vladarean et al., 2020). Automating β_0-tuning is an important direction for future research.

5.1 Matrix Completion

We consider two different formulations of the matrix completion problem. First, we focus on matrix completion with hard inequality constraints studied in Locatello et al. (2019):

$$\min_{\|w\|_\star \leq \zeta} \sum_{(i,j)\in\Omega} (w_{ij} - X_{ij})^2 \text{ subject to } 1 \leq w \leq 5 \quad (18)$$

where Ω is the observed entries of the input data X, and $\|X\|_\star$ denotes the nuclear norm. The inequality constraints $1 \leq w \leq 5$ are hard thresholds which specify that all the entries of w must lie between 1 and 5.

For X, we used the MovieLens-100k dataset* containing approximately 100,000 integer valued movie ratings between 1 and 5, assigned by 1682 users to 943 movies. We used the `ub.train` and `ub.test` partitions provided with the original data for the train/test split.

This numerical setup was studied also in Locatello et al. (2019). We used the parameter setting that they reported without any further tuning. We set $\zeta =$ `7e3` for the nuclear norm bound, $\beta_0 =$ `10` for the initial smoothing parameter, and we compute gradient estimators with `1000` *iid* samples at each iteration.

Figure 1 compares the performance of H-SAG-CGM/v1 against SHCGM (Locatello et al., 2019) in terms of

*F.M. Harper, J.A. Konstan. — Available at https://grouplens.org/datasets/movielens

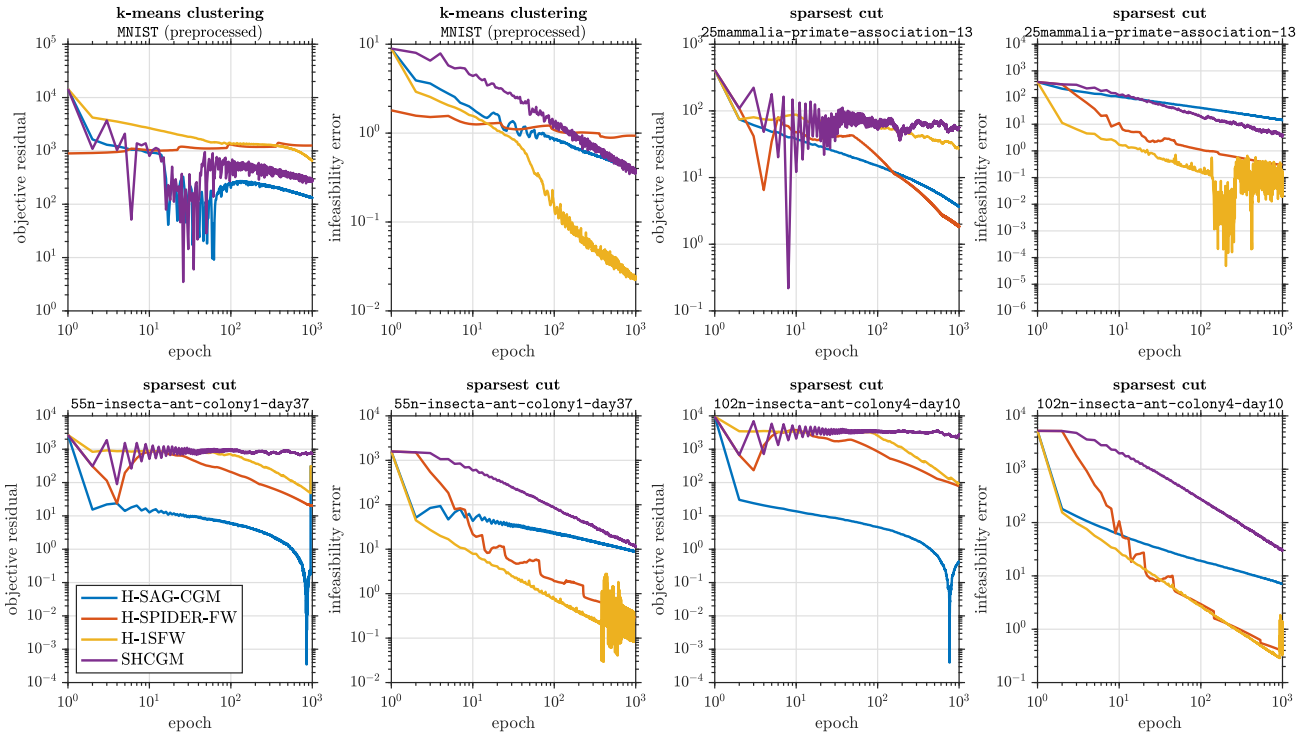

Figure 3: Comparing H-SAG-CGM/v2 to state-of-the-art baselines on two distinct SDP-relaxation tasks, k-means (20) and sparsest cut (22). The x-scale is in terms of the constraint epochs. One constraint epoch corresponds to a full pass over all the constraints. Note that for the k-means clustering experiment, we deliberately restricted H-SPIDER-FW to not perform full passes over all of the constraints resulting in noticeable degradation in performance.

train and test root mean squared error (RMSE) and infeasibility error. The comparison is based on the iteration counter, which is an arguably fair representation of the time cost of the algorithms since both methods use the same number of samples per iteration.

Next, we test our algorithm for a setting in which g is Lipschitz continuous by performing experiments on matrix completion with ℓ_1-regularization:

$$\min_{\|w\|_\star \le \zeta} \sum_{(i,j)\in\Omega} (w_{ij} - X_{ij})^2 + \lambda \|w\|_1. \tag{19}$$

We use the same dataset and parameter settings as in (18) with the regularization parameter set to $\lambda = 0.1$. Figure 2 presents the train and test RMSE obtained in this experiment. Note that the estimates remain feasible in this experiment since g is not an indicator function.

5.2 k-Means Clustering

In this experiment, we test H-SAG-CGM/v2. The goal in k-means is to assign n data points to k clusters. We consider the following SDP relaxation of this problem (Peng and Wei, 2007):

$$\min_{w\in\mathcal{X}} \quad \langle w, C\rangle \quad \text{subject to } w\vec{1} = \vec{1}, \text{ and } w \ge 0 \tag{20}$$

where $\mathcal{X} = \{w \in \mathbb{S}_+^{n\times n} \mid \mathrm{Tr}(w) \le \frac{1}{n}\}$, $\vec{1} = [1, 1, \ldots, 1] \in \mathbb{R}^n$, and $w \ge 0$ denotes entry-wise non-negativity. The problem is strongly constrained with a total of $n^2 + n$ constraints — n equality and n^2 inequality constraints.

This problem is also studied in the related works on homotopy CGM in (Yurtsever et al., 2018; Locatello et al., 2019; Vladarean et al., 2020). We use the same test setup: For the input data C, we use MNIST dataset[†] with the preprocessing considered in Mixon et al. (2016). We set $\beta_0 = 7$.

We compare the methods based on the number of epochs (an epoch corresponds to a full pass over the constraints) since different methods use different batch sizes in this experiment. The first two plots in Figure 3 present the outcomes of this experiment. We measure the objective residual as relative suboptimality $|f(w_k) - f^\star|/|f^\star|$ and the infeasibility error as the Euclidean distance to the feasible set $\mathrm{dist}(\mathrm{Aw_k}, \mathcal{K})$.

[†] http://yann.lecun.com/exdb/mnist/

5.3 Uniform Sparsest Cut

In this experiment, we test H-SAG-CGM/v2 on the uniform sparsest cut SDP. This problem is particularly interesting because of the $\mathcal{O}(n^3)$ number of constraints.

Let $G = (V, E)$ be a graph with n nodes $|V| = n$ and a set of edges E. The goal in uniform sparsest cut is to split vertices into two partitions $(S, \bar{S})$ that minimize

$$\frac{|E(S, \bar{S})|}{|S||\bar{S}|} \tag{21}$$

where $E(S, \bar{S}) \subseteq E$ is the set of edges between the nodes in S and $\bar{S}$.

This canonical problem has applications across many fields including VLSI circuit layout design, the topological design of communication networks, image segmentation, and many others. In machine learning, it is a sub-problem of hierarchical clustering (Dasgupta, 2016; Chatziafratis et al., 2018).

Arora et al. (2009) propose a $\mathcal{O}(\sqrt{\log n})$-approximation algorithm for this problem based on an SDP relaxation with $\mathcal{O}(n^3)$ triangle inequality constraints. We adapt their formulation to our SDP model (2):

$$\begin{aligned} \min_{\substack{w \in \mathbb{S}_+^{n\times n} \\ \operatorname{Tr}(w) \leq n}} \quad & \langle L, w \rangle \\ \text{subj. to} \quad & n \operatorname{Tr}(w) - \operatorname{Tr}(\mathbb{1}_{n\times n} w) = \frac{n^2}{2} \\ & w_{ij} + w_{jk} - w_{ik} - w_{jj} \leq 0 \quad \forall i, j, k \in V \end{aligned} \tag{22}$$

where L is the graph Laplacian of G.

We used three datasets from the Network Repository (Rossi and Ahmed, 2015):[‡] 25MAMMALIA-PRIMATE-ASSOCIATE-13, 55N-INSECTA-ANT-COLONY1-DAY37, and 102N-INSECTA-ANT-COLONY4-DAY10. These three datasets differ in size by a factor of ten. See Table S1 in the Appendix for more details. We use $\beta_0 = 100$ for all three network datasets.

Figure 3 presents the results of this experiment. As in the k-means experiment, the objective residual infeasibility error represent $|f(w_k) - f^\star|/|f^\star|$ and $\operatorname{dist}(\mathrm{A}w_k, \mathcal{K})$ respectively. H-SPIDER-FW is affected by the growing number of constraints because of its increasing batch size strategy. Other methods, with constant batch size, are less affected. H-SAG-CGM/v2 performs competitively against H-SPIDER-FW without requiring an increasing batch size.

[‡] https://networkrepository.com

6 CONCLUSION

We developed a fast randomized conditional gradient method for solving convex composite finite-sum problems. The proposed method is particularly suitable for solving SDPs with a large number of affine constraints. Theoretically, the proposed method has favorable scaling properties compared to the previous state-of-the-art. Empirically, it performs on par with more sophisticated variance reduction techniques.

The proposed method takes advantage of a structural assumption on the separability of the objective by applying randomization. For the non-smooth term, the proposed method tackles the two subcases of deterministic and stochastic separately. If the non-smooth term is deterministic, the proposed method obtains an ε-suboptimal solution after $\mathcal{O}(\varepsilon^{-2}dm)$ arithmetic operations (where d is the dimensionality of the decision variable and m is the number of constraints comprising g). This improves the previous complexity of $\mathcal{O}(\varepsilon^{-3}dm)$ found in Locatello et al. (2019).

If we further assume that the non-smooth part is also separable, then we can employ a fully randomized scheme to find an ε-suboptimal solution after $\mathcal{O}(\varepsilon^{-2}d)$ arithmetic operations. This total cost complexity is independent of m and thus represents a significant improvement compared over previous work (Vladarean et al., 2020) which has a total cost of $\mathcal{O}(\varepsilon^{-2}dm)$.

Acknowledgments

The authors thank Vincent Fortuin for his helpful feedback on an initial draft of this work and the anonymous reviewers for their detailed comments. This work started while F.L. was at ETH Zürich and is based on the research done outside of Amazon.

This work was supported by ETH core funding to G.R. (funding G.D.). V.C. has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 725594 – time-data). M.L.V. was supported by the Swiss National Science Foundation (SNSF) for the project "Theory and Methods for Storage-Optimal Optimization" grant number 200021_178865. A.Y. received support from the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

References

Zeeshan Akhtar and Ketan Rajawat. "Zeroth and First Order Stochastic Frank-Wolfe Algorithms for Constrained Optimization". arXiv preprint arXiv:2107.06534, 2021.

Abdo Y Alfakih, Amir Khandani, and Henry Wolkowicz. Solving Euclidean distance matrix completion problems via semidefinite programming. *Computational optimization and applications*, 1999.

Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 2009.

Vaggos Chatziafratis, Rad Niazadeh, and Moses Charikar. Hierarchical clustering with structural constraints. In *International Conference on Machine Learning*. PMLR, 2018.

Frank H Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.

Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, 2016.

Olivier Fercoq, Ahmet Alacaoglu, Ion Necoara, and Volkan Cevher. Almost surely constrained convex optimization. *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 1956.

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. *SIAM Journal on optimization*, 27(1): 319–346, 2017.

Dan Garber. Faster Projection-free Convex Optimization over the Spectrahedron. *Advances in Neural Information Processing Systems*, 29:874–882, 2016.

Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 1995.

Elad Hazan. Sparse approximate solutions to semidefinite programs. In *Latin American symposium on theoretical informatics*. Springer, 2008.

Elad Hazan and Haipeng Luo. Variance-Reduced and Projection-Free Stochastic Optimization. *arXiv preprint arXiv:1602.02101*, 2017.

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *International Conference on Machine Learning*. PMLR, 2013.

Martin Jaggi and Marek Sulovskỳ. A simple algorithm for nuclear norm regularized problems. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010.

Thomas Kerdreux, Alexandre d'Aspremont, and Sebastian Pokutta. Local and Global Uniform Convexity Conditions. *ArXiv*, abs/2102.05134, 2021.

Brian Kulis, Arun C Surendran, and John C Platt. Fast low-rank semidefinite programming for embedding and clustering. In *Artificial Intelligence and Statistics*. PMLR, 2007.

Guanghui Lan. The complexity of large-scale convex programming under a linear optimization oracle. *arXiv preprint arXiv:1309.5550*, 2013.

Fabian Latorre, Paul Rolland, and Volkan Cevher. Lipschitz constant estimation of neural networks via sparse polynomial optimization. *arXiv preprint arXiv:2004.08688*, 2020.

E.S. Levitin and B.T. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 1966. ISSN 0041-5553.

Francesco Locatello, Alp Yurtsever, Olivier Fercoq, and Volkan Cevher. Stochastic Frank-Wolfe for Composite Convex Minimization. *Advances in Neural Information Processing Systems*, 2019.

Konstantin Mishchenko and Peter Richtárik. A stochastic decoupling method for minimizing the sum of smooth and non-smooth functions. *arXiv:1905.11535*, 2019.

Dustin G Mixon, Soledad Villar, and Rachel Ward. Clustering subgaussian mixtures by semidefinite programming. *arXiv preprint arXiv:1602.06612*, 2016.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic Conditional Gradient Methods: From Convex Minimization to Submodular Maximization. *The Journal of Machine Learning Research (JMLR)*, 2018.

Geoffrey Négiar, Gideon Dresdner, Alicia Yi-Ting Tsai, Laurent El Ghaoui, Francesco Locatello, and Fabian Pedregosa. Stochastic Frank-Wolfe for Constrained Finite-Sum Minimization. *The International Conference on Machine Learning (ICML)*, 2020.

Yu. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 2005.

Yu. Nesterov. Complexity bounds for primal-dual methods minimizing the model of objective function. *Mathematical Programming*, 2018.

Andrei Patrascu and Ion Necoara. Nonasymptotic convergence of stochastic proximal point methods for constrained convex optimization. *The Journal of Machine Learning Research*, 2017.

Jiming Peng and Yu Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM journal on optimization*, 2007.

Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi,

and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Stochastic Frank-Wolfe methods for nonconvex optimization . In 54*th Annual Allerton Conf. Communication, Control, and Computing*, 2016.

Ryan Rossi and Nesreen Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing Finite Sums with the Stochastic Average Gradient. *Mathematical Programming*, 2017.

Zebang Shen, Cong Fang, Peilin Zhao, Junzhou Huang, and Hui Qian. Complexities in Projection-Free Stochastic Non-Convex Minimization . In *Proc. 22nd Int. Conf. Artificial Intelligence and Statistics*, 2019.

Quoc Tran-Dinh, Olivier Fercoq, and Volkan Cevher. A Smooth Primal-Dual Optimization Framework for Nonsmooth Composite Convex Minimization. *SIAM Journal on Optimization*, 2018. doi:10.1137/16M1093094.

Maria-Luiza Vladarean, Ahmet Alacaoglu, Ya-Ping Hsieh, and Volkan Cevher. Conditional gradient methods for stochastically constrained convex minimization. *arXiv preprint arXiv:2007.03795*, 2020.

Yangyang Xu. Primal-dual stochastic gradient method for convex programs with many functional constraints. *SIAM Journal on Optimization*, 2020.

Alp Yurtsever, Ya-Ping Hsieh, and Volkan Cevher. Scalable convex methods for phase retrieval. In *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2015.

Alp Yurtsever, Olivier Fercoq, Francesco Locatello, and Volkan Cevher. A conditional gradient framework for composite convex minimization with applications to semidefinite programming. *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Alp Yurtsever, Olivier Fercoq, and Volkan Cevher. A Conditional-Gradient-Based Augmented Lagrangian Framework. *Proceedings of the 36th International Conference on Machine Learning*, 97:7272–7281, 09–15 Jun 2019a.

Alp Yurtsever, Suvrit Sra, and Volkan Cevher. Conditional gradient methods via stochastic path-integrated differential stimator. In *Proceedings of the 36th International Conference on Machine Learning*, 2019b.

Alp Yurtsever, Joel A Tropp, Olivier Fercoq, Madeleine Udell, and Volkan Cevher. Scalable semidefinite programming. *SIAM Journal on Mathematics of Data Science*, 3(1):171–200, 2021.

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.

Qing Zhao, Stefan E Karisch, Franz Rendl, and Henry Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, 1998.

Supplementary Material: Faster One-Sample Stochastic Conditional Gradient Method for Composite Convex Minimization

A Background on Smoothing

This section recalls some useful properties about the smoothing technique (Nesterov, 2005). We present these known properties in this section for completeness, since we use them in our analysis.

Let $g : \mathbb{R}^m \to \mathbb{R} \cap \{+\infty\}$ be a proper, closed and convex function. The smooth approximation of g is defined by

$$g_\beta(z) = \max_{y \in \mathbb{R}^d} \left\{ \langle z, y \rangle - g^*(y) - \frac{\beta}{2}\|y\|^2 \right\} \tag{23}$$

where g^* denotes the Fenchel conjugate and $\beta > 0$ is the smoothing parameter. Then, g_β is convex and $\frac{1}{\beta}$-smooth. Let $y^*_\beta(z)$ denote the solution of the maximization sub-problem in (23), i.e.,

$$\begin{aligned}
y^*_\beta(z) &= \arg\max_{y \in \mathbb{R}^d} \left\{ \langle z, y \rangle - g^*(y) - \frac{\beta}{2}\|y\|^2 \right\} && (24)\\
&= \arg\min_{y \in \mathbb{R}^d} \left\{ \frac{1}{\beta} g^*(y) - \frac{1}{\beta}\langle z, y \rangle + \frac{1}{2}\|y\|^2 + \frac{1}{2}\|\frac{1}{\beta} z\|^2 \right\} && (25)\\
&= \arg\min_{y \in \mathbb{R}^d} \left\{ \frac{1}{\beta} g^*(y) + \frac{1}{2}\|y - \frac{1}{\beta} z\|^2 \right\} && (26)\\
&= \mathrm{prox}_{\beta^{-1} g^*}(\beta^{-1} z) && (27)\\
&= \frac{1}{\beta}\big(z - \mathrm{prox}_{\beta g}(z)\big) && (28)
\end{aligned}$$

where the last line is the Moreau decomposition. Then, the followings hold $\forall z_1, z_2 \in \mathbb{R}^m$ and $\forall \beta, \gamma > 0$

$$\begin{aligned}
g_\beta(z_1) &\geq g_\beta(z_2) + \langle \nabla g_\beta(z_2), z_1 - z_2 \rangle + \frac{\beta}{2}\|y^*_\beta(z_2) - y^*_\beta(z_1)\|^2 && (29)\\
g(z_1) &\geq g_\beta(z_2) + \langle \nabla g_\beta(z_2), z_1 - z_2 \rangle + \frac{\beta}{2}\|y^*_\beta(z_2)\|^2 && (30)\\
g_\beta(z_1) &\leq g_\gamma(z_1) + \frac{\gamma - \beta}{2}\|y^*_\beta(z_1)\|^2 && (31)
\end{aligned}$$

We refer to Lemma 10 in (Tran-Dinh et al., 2018) for the proofs.

Suppose that g is L_g-Lipschitz continuous. Then, for $\forall \beta > 0$ and $\forall z \in \mathbb{R}^m$,

$$g_\beta(z) \leq g(z) \leq g_\beta(z) + \frac{\beta}{2} L_g^2, \tag{32}$$

The proof follows immediately from Equation (2.7) in (Nesterov, 2005) with a remark on the duality between bounded domain and Lipschitz continuity.

B Proof of Lemma 4.1

We follow the steps laid out in Theorem 4.1 in (Vladarean et al., 2020), which in turn builds upon Theorem 9 in (Locatello et al., 2019).

We use the quadratic upper bound ensured by the fact that F_{β_k} is $L_{F_{\beta_k}}$-smooth:

$$F_{\beta_k}(w_{k+1}) \leq F_{\beta_k}(w_k) + \langle \nabla F_{\beta_k}(w_k), w_{k+1} - w_k \rangle + \frac{L_{F_{\beta_k}}}{2}\|w_{k+1} - w_k\|^2 \tag{33}$$

$$\leq F_{\beta_k}(w_k) + \eta_k \langle \nabla F_{\beta_k}(w_k), s_k - w_k \rangle + \frac{\eta_k^2 L_{F_{\beta_k}} D_{\mathcal{W}}^2}{2} \tag{34}$$

where the second line follows from the boundedness of $\mathcal{W}$.

Next, we use the rule for change of β in smoothing (see (31)), which gives

$$F_{\beta_k}(w_{k+1}) \leq F_{\beta_{k-1}}(w_k) + \frac{\beta_{k-1} - \beta_k}{2}\|y_{\beta_k}^\star(Aw_k)\|^2 + \eta_k \langle \nabla F_{\beta_k}(w_k), s_k - w_k \rangle + \frac{\eta_k^2 L_{F_{\beta_k}} D_{\mathcal{W}}^2}{2}, \tag{35}$$

where $y_{\beta_k}^\star$ is defined as in (24).

Then, we bound the term $\langle \nabla F_{\beta_k}(w_k), s_k - w_k \rangle$ as follows:

$$\langle \nabla F_{\beta_k}(w_k), s_k - w_k \rangle = \langle \nabla F_{\beta_k}(w_k) - v_k, s_k - w_k \rangle + \langle v_k, s_k - w_k \rangle \tag{36}$$

$$= \langle \nabla F_{\beta_k}(w_k) - v_k, s_k - w^\star \rangle + \langle \nabla F_{\beta_k}(w_k) - v_k, w^\star - w_k \rangle + \langle v_k, s_k - w_k \rangle \tag{37}$$

$$\leq \langle \nabla F_{\beta_k}(w_k) - v_k, s_k - w^\star \rangle + \langle \nabla F_{\beta_k}(w_k) - v_k, w^\star - w_k \rangle + \langle v_k, w^\star - w_k \rangle \tag{38}$$

$$= \langle \nabla F_{\beta_k}(w_k) - v_k, s_k - w^\star \rangle + \langle \nabla F_{\beta_k}(w_k), w^\star - w_k \rangle \tag{39}$$

where the inequality follows by the definition of s_k.

Now, we focus on the term $\langle \nabla F_{\beta_k}(w_k), w^\star - w_k \rangle$ and bound it as follows:

$$\langle \nabla F_{\beta_k}(w_k), w^\star - w_k \rangle = \langle X^T \nabla f(Xw_k) + A^T \nabla g_{\beta_k}(Aw_k), w^\star - w_k \rangle \tag{40}$$

$$= \langle \nabla f(Xw_k), X(w^\star - w_k) \rangle + \langle \nabla g_{\beta_k}(Aw_k), A(w^\star - w_k) \rangle \tag{41}$$

$$\leq f(Xw^\star) - f(Xw_k) + g(Aw^\star) - g_{\beta_k}(Aw_k) - \frac{\beta_k}{2}\|y_{\beta_k}^\star(Aw_k)\|^2 \tag{42}$$

$$= F^\star - F_{\beta_k}(w_k) - \frac{\beta_k}{2}\|y_{\beta_k}^\star(Aw_k)\|^2, \tag{43}$$

where the inequality holds due to the convexity of f and g and the smoothing property in (30).

Combining all these bounds and subtracting $F^\star$ from both sides, we get

$$F_{\beta_k}(w_{k+1}) - F^\star \leq (1-\eta_k)\left(F_{\beta_{k-1}}(w_k) - F^\star\right) + \eta_k \langle \nabla F_{\beta_k}(w_k) - v_k, s_k - w^\star \rangle + \frac{1}{2}((1-\eta_k)(\beta_{k-1} - \beta_k) - \eta_k \beta_k)\|y_{\beta_k}^\star(Aw_k)\|_2^2 + \frac{\eta_k^2 L_{F_{\beta_k}} D_{\mathcal{W}}^2}{2} \tag{44}$$

We cannot bound $\|y_{\beta_k}^\star(Aw_k)\|_2^2$ in general, so we choose η_k and β_k carefully to vanish this term. Let $\eta_k = \frac{2}{k+1}$ and $\beta_k = \frac{\beta_0}{\sqrt{k+1}}$ for an arbitrary $\beta_0 > 0$. Then,

$$(1-\eta_k)(\beta_{k-1} - \beta_k) - \eta_k \beta_k = \frac{\beta_0}{\sqrt{k}}\left(\frac{k-1}{k+1} - \frac{\sqrt{k}}{\sqrt{k+1}}\right) < 0, \qquad \text{for all } k \geq 1. \tag{45}$$

Finally, taking expectation on both sides and applying the definition of $S_\beta(w) \triangleq \mathbb{E}\left[F_\beta(w) - F^\star\right]$ we arrive at our stated result:

$$S_{\beta_k}(w_{k+1}) \leq (1-\eta_k) S_{\beta_{k-1}}(w_k) + \eta_k \mathbb{E}[\langle \nabla F_{\beta_k}(w_k) - v_k, s_k - w^\star \rangle] + \frac{\eta_k^2 L_{F_{\beta_k}} D_{\mathcal{W}}^2}{2}. \tag{46}$$

C Proof of Theorem 4.1

Our aim is to get a rate on the smoothed gap $S_{\beta_k}(w_{k+1})$. We start from Lemma 4.1:

$$S_{\beta_k}(w_{k+1}) \leq (1-\eta_k) S_{\beta_{k-1}}(w_k) + \eta_k \mathbb{E}[\langle \nabla F_{\beta_k}(w_k) - v_k, s_k - w^\star \rangle] + \frac{\eta_k^2}{2}\left(\frac{\|X\| L_f}{n} + \frac{\|A\|}{\beta_k}\right) D_{\mathcal{W}}^2. \tag{47}$$

Multiply both sides by $k(k+1)$ and unroll the recurrence to get

$$\begin{aligned} k(k+1) S_{\beta_k}(w_{k+1}) &\leq (k-1)k S_{\beta_{k-1}}(w_k) + 2k\mathbb{E}[\langle \nabla F_{\beta_k}(w_k) - v_k, s_k - w^\star \rangle] + \frac{2k}{k+1}\left(\frac{\|X\| L_f}{n} + \frac{\|A\|}{\beta_k}\right) D_{\mathcal{W}}^2 \\ &\leq \underbrace{\sum_{i=1}^k 2i \mathbb{E}[\langle \nabla F_{\beta_i}(w_i) - v_i, s_i - w^\star \rangle]}_{\text{(A)}} + \underbrace{\sum_{i=1}^k \frac{2i}{i+1}\left(\frac{\|X\| L_f}{n} + \frac{\|A\|}{\beta_i}\right) D_{\mathcal{W}}^2}_{\text{(B)}}. \end{aligned} \tag{48}$$

First, we get an upper-bound on the variance term (A) as follows:

$$\begin{aligned} \mathbb{E}[\langle \nabla F_{\beta_k}(w_k) - v_k, s_k - w^\star \rangle] &= \mathbb{E}[\langle X^T(\nabla f(Xw_k) - \alpha_k) + A^T(\nabla g_{\beta_k}(Aw_k) - \gamma_k), s_k - w^\star \rangle] && (49) \\ &= \mathbb{E}[\langle \nabla f(Xw_k) - \alpha_k, X(s_k - w^\star) \rangle] && (50) \\ &\leq \mathbb{E}[\|\nabla f(Xw_k) - \alpha_k\|_1 \, \|X(s_k - w^\star)\|_\infty] && (51) \\ &\leq \mathbb{E}[\|\nabla f(Xw_k) - \alpha_k\|_1] \, D_\infty(X) && (52) \end{aligned}$$

where, the first inequality is the Hölder's inequality, and the second one is based on the boundedness of $\mathcal{W}$.

Then, by Lemma 4.2, we have

$$\mathbb{E}[\|\nabla f(Xw_k) - \alpha_k\|_1] \leq \left(1 - \tfrac{1}{n}\right)^k \|\nabla f(Xw_1) - \alpha_0\|_1 + \frac{2L_f D_1(X)}{n}\left(\left(1-\tfrac{1}{n}\right)^{k/2} \log k + \frac{2(n-1)}{k}\right). \tag{53}$$

Finally, we combine (52) and (53) to get

$$\begin{aligned} \text{(A)} &\leq 2D_\infty(X)\left[\|\nabla f(Xw_1) - \alpha_0\|_1 \sum_{i=1}^k i\left(1-\tfrac{1}{n}\right)^i + \frac{2L_f D_1(X)}{n}\sum_{i=1}^k \left(i\left(1-\tfrac{1}{n}\right)^{i/2}\log i + 2(n-1)\right)\right] && (54) \\ &\leq 2D_\infty(X)\left[\|\nabla f(Xw_1) - \alpha_0\|_1 \, n^2 + \frac{2L_f D_1(X)}{n}\left(16n^3 + 2(n-1)k\right)\right] && (55) \\ &\leq 2D_\infty(X)\left[\|\nabla f(Xw_1) - \alpha_0\|_1 \, n^2 + 4L_f D_1(X)\left(8n^2 + k\right)\right] && (56) \end{aligned}$$

where we use Lemma F.1 for the second line.

Next, we focus on the term (B), and we use once again Lemma F.1 and obtain

$$\text{(B)} = 2D_{\mathcal{W}}^2\left(\frac{\|X\| L_f}{n}\sum_{i=1}^k \frac{i}{i+1} + \frac{\|A\|}{\beta_0}\sum_{i=1}^k \frac{i}{\sqrt{i+1}}\right) \leq 2D_{\mathcal{W}}^2\left(\frac{\|X\| L_f}{n}k + \frac{\|A\|}{\beta_0}k\sqrt{k+1}\right). \tag{57}$$

To finalize, we substitute the bounds on (A) and (B) into (48) and divide both sides by $k(k+1)$ to get the desired bound on $S_{\beta_k}(w_{k+1})$:

$$\begin{aligned} S_{\beta_k}(w_{k+1}) &\leq \frac{2D_\infty(X)}{k(k+1)}\left[\|\nabla f(Xw_1) - \alpha_0\|_1 \, n^2 + 4L_f D_1(X)\left(8n^2 + k\right)\right] + \frac{2D_{\mathcal{W}}^2}{k(k+1)}\left(\frac{\|X\| L_f}{n}k + \frac{\|A\|}{\beta_0}k\sqrt{k+1}\right) \\ &\leq \frac{C_3}{k(k+1)} + \frac{C_2}{k+1} + \frac{C_1}{\sqrt{k+1}}, \quad \text{where} \quad \begin{aligned} C_3 &= 2n^2 D_\infty(X)\left(\|\nabla f(Xw_1) - \alpha_0\|_1 + 32L_f D_1(X)\right) \\ C_2 &= 8L_f D_1(X) D_\infty(X) + 2n^{-1} L_f \|X\| D_{\mathcal{W}}^2 \\ C_1 &= 2D_{\mathcal{W}}^2 \|A\| \beta_0^{-1}. \end{aligned} \end{aligned}$$

C.1 Proof of Corollary 4.1

Suppose g is L_g-Lipschitz continuous. Then, from (32) we get

$$\mathbb{E}F(x_{k+1}) - F^\star = \mathbb{E}[f(Xw_{k+1}) + g(Aw_{k+1})] - F^\star \tag{58}$$

$$\leq \mathbb{E}[f(Xw_{k+1}) + g_{\beta_k}(Aw_{k+1})] - F^\star + \frac{\beta_k L_g^2}{2} \tag{59}$$

$$= S_{\beta_k}(w_{k+1}) + \frac{\beta_0 L_g^2}{2\sqrt{k+1}}. \tag{60}$$

C.2 Proof of Corollary 4.2

Suppose $g(z) = \delta_{\mathcal{K}}(z)$, the indicator function of a closed and convex set. We can write the Lagrangian as

$$\mathcal{L}(w, r, y) \triangleq f(Xw) + \langle Aw - r, y\rangle, \qquad w \in \mathcal{W},\ r \in \mathcal{K}. \tag{61}$$

From the Lagrange saddle point theory, we have

$$f(Xw^\star) \leq \mathcal{L}(w, r, y^\star) \leq f(Xw) + \|Aw - r\|\|y^\star\|, \qquad \forall w \in \mathcal{W} \text{ and } \forall r \in \mathcal{K}. \tag{62}$$

Letting $w = w_{k+1} \in \mathcal{W}$ and $r = \mathrm{proj}_{\mathcal{K}}(Aw_{k+1}) \in \mathcal{K}$, taking expectation on both sides and rearranging, we get

$$\mathbb{E}\left[f(Xw_{k+1}) - f(Xw^*)\right] \geq -\|y^\star\|\,\mathbb{E}\left[\mathrm{dist}(Aw_{k+1}, \mathcal{K})\right] \tag{63}$$

This is the desired lower-bound on objective residual.

Next, we derive an upper bound on objective residual. By definition of g_β (see (23)) for $\delta_{\mathcal{K}}$,

$$g_\beta(Aw) = \frac{1}{2\beta}\,\mathrm{dist}(Aw, \mathcal{K})^2. \tag{64}$$

Note that $f(Xw^\star) = F(w^\star)$ since $g(Aw^\star) = 0$. Then,

$$\mathbb{E}[f(Xw_{k+1}) - f(Xw^\star)] = \mathbb{E}[F_{\beta_k}(w_{k+1}) - F^\star - g_{\beta_k}(Aw_{k+1})] \tag{65}$$

$$\leq S_{\beta_k}(w_{k+1}) - \frac{1}{2\beta_k}\mathbb{E}[\mathrm{dist}(Aw_{k+1}, \mathcal{K})^2] \tag{66}$$

$$\leq S_{\beta_k}(w_{k+1}). \tag{67}$$

Finally, we derive convergence rate of the infeasibility error. To this end, we combine (63) and (66):

$$-\|y^\star\|\,\mathbb{E}\left[\mathrm{dist}(Aw_{k+1}, \mathcal{K})\right] \leq S_{\beta_k}(w_{k+1}) - \frac{1}{2\beta_k}\mathbb{E}[\mathrm{dist}(Aw_{k+1}, \mathcal{K})^2] \tag{68}$$

We rearrange and apply Jensen's inequality to $\mathbb{E}[\mathrm{dist}(Aw_{k+1}, \mathcal{K})^2]$, and we get a second order inequality with respect to $\mathbb{E}[\mathrm{dist}(Aw_{k+1}, \mathcal{K})]$:

$$\frac{1}{2\beta_k}\underbrace{\mathbb{E}[\mathrm{dist}(Aw_{k+1}, \mathcal{K})]^2}_{t^2} - \|y^\star\|\underbrace{\mathbb{E}[\mathrm{dist}(Aw_{k+1}, \mathcal{K})]}_{t} - S_{\beta_k}(w_{k+1}) \leq 0. \tag{69}$$

By solving this inequality for t, we achieve the desired bound:

$$\mathbb{E}[\mathrm{dist}(Aw_{k+1}, \mathcal{K})] \leq \beta_k\left(\|y^\star\| + \sqrt{\|y^\star\|^2 + \frac{2S_{\beta_k}(w_{k+1})}{\beta_k}}\right) \leq 2\beta_k\|y^\star\| + \sqrt{2\beta_k S_{\beta_k}(w_{k+1})}, \tag{70}$$

where we used $\sqrt{a^2 + b^2} \leq a + b$ for $a, b \geq 0$ in the last inequality to simplify the terms.

D Proof of Lemma 4.3

The following Lemma will be needed in the subsequent characterization of the estimator variance.

Lemma D.1. *Let $\rho \in (0,1)$, $C \in \mathbb{R}$ and $\{u_k\}_{k\in\mathbb{N}}$ be a sequence such that*

$$u_k \leq \rho(u_{k-1} + \frac{1}{\sqrt{k}}C). \tag{71}$$

Then, it holds that

$$u_k \leq \rho^k u_1 + \frac{2C\rho}{\sqrt{k}(1-\rho)}. \tag{72}$$

Proof. Unrolling the recurrence yields

$$u_k \leq \rho^{k-1}u_1 + C\sum_{i=2}^{k}\frac{\rho^{k-i+1}}{\sqrt{i}} \tag{73}$$

Observe that ρ^{k+1-i} is a monotonically increasing with i because $\rho \in (0,1)$. Therefore,

$$\frac{1}{\sum_{i=1}^{k}\frac{1}{\sqrt{i}}}\sum_{i=1}^{k}\frac{\rho^{k-i+1}}{\sqrt{i}} \leq \frac{1}{k}\sum_{i=1}^{k}\rho^{k-i+1} = \frac{1}{k}\sum_{i=1}^{k}\rho^i \tag{74}$$

since the left side of the inequality is a weighted average of ρ^{k-i+1} with decreasing weights and the right side is the simple average with uniform weights. The equality holds simply by change of indices. Now, we rearrange as

$$\sum_{i=1}^{k}\frac{\rho^{k-i+1}}{\sqrt{i}} \leq \frac{1}{k}\left(\sum_{i=1}^{k}\frac{1}{\sqrt{i}}\right)\left(\sum_{i=1}^{k}\rho^i\right) \leq \frac{2\rho}{\sqrt{k}(1-\rho)} \tag{75}$$

We complete the proof by combining (73) and (75). □

D.1 Proof of Lemma 4.3 for indicator functions

First, we prove Lemma 4.3 for the case in which g is an indicator function. Observe that

$$\mathbb{E}_k[|\nabla g_{\beta_k}(Aw_k)_j - \gamma_{k,j}|] = \frac{1}{m}0 + \frac{m-1}{m}|\nabla g_{\beta_k}(Aw_k)_j - \gamma_{k-1,j}|. \tag{76}$$

Summing over all coordinates gives

$$\begin{aligned}
\mathbb{E}[\|\nabla g_{\beta_k}(Aw_k) - \gamma_k\|_1] &= \frac{m-1}{m}\mathbb{E}[\|\nabla g_{\beta_k}(Aw_k) - \gamma_{k-1}\|_1] && (77)\\
&= \frac{m-1}{m}\mathbb{E}[\|\nabla g_{\beta_k}(Aw_k) - \nabla g_{\beta_{k-1}}(Aw_{k-1}) + \nabla g_{\beta_{k-1}}(Aw_{k-1}) - \gamma_{k-1}\|_1] && (78)\\
&\leq \frac{m-1}{m}\Big(\mathbb{E}[\|\nabla g_{\beta_{k-1}}(Aw_{k-1}) - \gamma_{k-1}\|_1] + \mathbb{E}[\|\nabla g_{\beta_k}(Aw_k) - \nabla g_{\beta_{k-1}}(Aw_{k-1})\|_1]\Big). && (79)
\end{aligned}$$

Now, we focus on the last term and bound it as follows:

$$\begin{aligned}
\|\nabla g_{\beta_k}(Aw_k) - \nabla g_{\beta_{k-1}}(Aw_{k-1})\|_1 &= \|\nabla g_{\beta_k}(Aw_k) \pm \nabla g_{\beta_k}(Aw_{k-1}) - \nabla g_{\beta_{k-1}}(Aw_{k-1})\|_1 && (80)\\
&\leq \|\nabla g_{\beta_k}(Aw_k) - \nabla g_{\beta_k}(Aw_{k-1})\|_1 + \|\nabla g_{\beta_k}(Aw_{k-1}) - \nabla g_{\beta_{k-1}}(Aw_{k-1})\|_1 && (81)\\
&\leq \frac{1}{m\beta_k}\|A(w_{k-1} - w_k)\|_1 + \frac{1}{m}\Big(\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\Big)\|Aw_{k-1} - \mathrm{proj}_K(Aw_{k-1})\|_1 && (82)\\
&\leq \frac{\eta_{k-1}}{m\beta_k}D_1(A) + \frac{1}{m}\Big(\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\Big)\|Aw_{k-1} - Aw^\star\|_1 && (83)\\
&\leq \frac{D_1(A)}{m}\Big(\frac{\eta_{k-1}}{\beta_k} + \frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\Big) && (84)
\end{aligned}$$

where the third inequality is due to the fact that $\mathcal{K} = \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_m$. Simplifying further: $\frac{\eta_{k-1}}{\beta_k} + \frac{1}{\beta_k} - \frac{1}{\beta_{k-1}} = \frac{2}{k}\frac{\sqrt{k+1}}{\beta_0} + \frac{\sqrt{k+1}}{\beta_0} - \frac{\sqrt{k}}{\beta_0} < \frac{2}{k}\frac{\sqrt{k}+1}{\beta_0} + \frac{\sqrt{k}\sqrt{k+1}}{\beta_0\sqrt{k}} - \frac{k}{\beta_0\sqrt{k}} < \frac{2}{\beta_0\sqrt{k}} + \frac{2}{\beta_0 k} + \frac{k+1}{\beta_0\sqrt{k}} - \frac{k}{\beta_0\sqrt{k}} < \frac{5}{\beta_0\sqrt{k}}$, gives

$$\|\nabla g_{\beta_k}(Aw_k) - \nabla g_{\beta_{k-1}}(Aw_{k-1})\|_1 \leq \frac{5D_1(A)}{m\beta_0\sqrt{k}}. \tag{85}$$

Substituting this back into (79), we get

$$\mathbb{E}[\|\nabla g_{\beta_k}(Aw_k) - \gamma_k\|_1] \leq \frac{m-1}{m}\Big(\mathbb{E}[\|\nabla g_{\beta_{k-1}}(Aw_{k-1}) - \gamma_{k-1}\|_1] + \frac{5D_2(A)\sqrt{m}}{\beta_0\sqrt{k}}\Big). \tag{86}$$

This is in the form of (71). We conclude the proof by applying Lemma D.1:

$$\mathbb{E}[\|\nabla g_{\beta_k}(Aw_k) - \gamma_k\|_1] \leq \left(\frac{m-1}{m}\right)^k \mathbb{E}[\|\nabla g_{\beta_0}(Aw_0) - \gamma_0\|_1] + \frac{10D_2(A)\sqrt{m}(m-1)}{\beta_0\sqrt{k}}. \tag{87}$$

D.2 Proof of Lemma 4.3 for Lipschitz continuous functions

Suppose g is Lipschitz continuous with parameter L_g. Then, from (32), we get

$$\underbrace{f(Xw_{k+1}) + g(Aw_{k+1})}_{F(w_{k+1})} \leq \underbrace{f(Xw_{k+1}) + g_{\beta_k}(Aw_{k+1})}_{F_{\beta_k}(w_{k+1})} + \frac{\beta_k}{2}L_g^2 = F_{\beta_k}(w_{k+1}) + \frac{\beta_0 L_g^2}{2\sqrt{k+1}}. \tag{88}$$

We achieve the desired bound by subtracting $F^\star$ and taking expectation on both sides:

$$\mathbb{E}[F(w_{k+1}) - F^\star] \leq S_{\beta_k}(w_{k+1}) + \frac{\beta_0 L_g^2}{2\sqrt{k+1}}. \tag{89}$$

To bound S_{β_k}, we can follow the proof of Lemma 4.3 up to (81), which we repeat here for convenience:

$$\|\nabla g_{\beta_k}(Aw_k) - \nabla g_{\beta_k}(Aw_{k-1})\|_1 + \|\nabla g_{\beta_k}(Aw_{k-1}) - \nabla g_{\beta_{k-1}}(Aw_{k-1})\|_1$$

Recall that $\nabla g_\beta(z) = \beta^{-1}(z - \operatorname{prox}_{\beta g}(z))$. The first term can be bounded using the $1/\beta$-smoothness of g_β. For the second term, recall the well-established fact that $\operatorname{prox}_g(z) = \lambda \operatorname{prox}_{g/\lambda}(x/\lambda)$ for any $\lambda > 0$. Thus,

$$\begin{aligned}
\nabla g_{\beta_k}(Aw_{k-1}) &= {\beta_k}^{-1}(Aw_{k-1} - \operatorname{prox}_{\beta_k g}(Aw_{k-1})) && (90)\\
&= {\beta_k}^{-1}(Aw_{k-1} - \frac{\beta_k}{\beta_{k-1}}\operatorname{prox}_{\beta_{k-1} g}(\frac{\beta_{k-1}}{\beta_k}Aw_{k-1})) && (91)\\
&= \nabla g_{\beta_{k-1}}(\frac{\beta_{k-1}}{\beta_k}Aw_{k-1}) && (92)
\end{aligned}$$

Thus,

$$\begin{aligned}
&\|\nabla g_{\beta_k}(Aw_k) - \nabla g_{\beta_k}(Aw_{k-1})\|_1 + \|\nabla g_{\beta_k}(Aw_{k-1}) - \nabla g_{\beta_{k-1}}(Aw_{k-1})\|_1 && (93)\\
&\leq \frac{1}{m\beta_k}\|A(w_k - w_{k-1})\|_1 + \frac{1}{m\beta_{k-1}}(\frac{\beta_{k-1}}{\beta_k} - 1)\|Aw_{k-1}\|_1 && (94)\\
&\leq \frac{\eta_{k-1}}{m\beta_k}D_1(A) + \frac{1}{m}(\frac{1}{\beta_k} - \frac{1}{\beta_{k-1}})\|Aw_{k-1}\|_1 && (95)\\
&\leq \frac{D_1(A)}{m}\left(\frac{\eta_{k-1}}{\beta_k} + \frac{1}{\beta_k} - \frac{1}{\beta_{k-1}}\right) && (96)
\end{aligned}$$

Note that this is identical to (84) in Lemma D.1. Thus, the rest of Lemma D.1 can be applied to arrive at the same bound.

E Proof of Theorem 4.1 for H-SAG-CGM/v2

The proof is same until (48). Then, get an upper-bound on the variance term (A) as follows:

$$\begin{aligned}
\mathbb{E}[\langle \nabla F_{\beta_k}(w_k) - v_k, s_k - w^\star\rangle] &= \mathbb{E}[\langle X^T(\nabla f(Xw_k) - \alpha_k) + A^T(\nabla g_{\beta_k}(Aw_k) - \gamma_k), s_k - w^\star\rangle] && (97)\\
&= \mathbb{E}[\langle \nabla f(Xw_k) - \alpha_k, X(s_k - w^\star)\rangle + \langle \nabla g_{\beta_k}(Aw_k) - \gamma_k, A(s_k - w^\star)\rangle] && (98)\\
&\le \mathbb{E}[\|\nabla f(Xw_k) - \alpha_k\|_1 \, \|X(s_k - w^\star)\|_\infty + \|\nabla g_{\beta_k}(Aw_k) - \gamma_k\|_1 \, \|A(s_k - w^\star)\|_\infty] && (99)\\
&\le \mathbb{E}[\|\nabla f(Xw_k) - \alpha_k\|_1] \, D_\infty(X) + \mathbb{E}[\|\nabla g_{\beta_k}(Aw_k) - \gamma_k\|_1] \, D_\infty(A) && (100)
\end{aligned}$$

where, the first inequality is the Hölder's inequality, and the second one is based on the boundedness of $\mathcal{W}$.

Then, by Lemma 4.2, we have

$$\mathbb{E}[\|\nabla f(Xw_k) - \alpha_k\|_1] \le \left(1 - \tfrac{1}{n}\right)^k \|\nabla f(Xw_1) - \alpha_0\|_1 + \frac{2L_f D_1(X)}{n}\left(\left(1 - \tfrac{1}{n}\right)^{k/2} \log k + \frac{2(n-1)}{k}\right) \tag{101}$$

And by Lemma 4.3, we have

$$\mathbb{E}[\|\nabla g_{\beta_k}(Aw_k) - \gamma_k\|_1] \le \left(1 - \tfrac{1}{m}\right)^k \mathbb{E}[\|\nabla g_{\beta_0}(Aw_1) - \gamma_0\|_1] + \frac{10 D_2(A)\sqrt{m}(m-1)}{\beta_0 \sqrt{k}}. \tag{102}$$

Finally, we substitute (101) and (102) back into (100) to get

$$\begin{aligned}
\text{(A)} &\le 2D_\infty(X)\left[\|\nabla f(Xw_1) - \alpha_0\|_1 \sum_{i=1}^{k} i\left(1 - \tfrac{1}{n}\right)^i + \frac{2L_f D_1(X)}{n}\sum_{i=1}^{k}\left(i\left(1 - \tfrac{1}{n}\right)^{i/2}\log i + 2(n-1)\right)\right]\\
&\quad + 2D_\infty(A)\left[\|\nabla g_{\beta_0}(Aw_1) - \gamma_0\|_1 \sum_{i=1}^{k} i\left(1 - \tfrac{1}{m}\right)^i + \frac{10 D_2(A)\sqrt{m}(m-1)}{\beta_0}\sum_{i=1}^{k}\sqrt{i}\right] && (103)\\
&\le 2D_\infty(X)\left[\|\nabla f(Xw_1) - \alpha_0\|_1 \, n^2 + \frac{2L_f D_1(X)}{n}\left(16n^3 + 2(n-1)k\right)\right]\\
&\quad + 2D_\infty(A)\left[\|\nabla g_{\beta_0}(Aw_1) - \gamma_0\|_1 \, m^2 + \frac{10 D_2(A)\sqrt{m}(m-1)}{\beta_0} k^{3/2}\right] && (104)\\
&\le 2D_\infty(X)\left[\|\nabla f(Xw_1) - \alpha_0\|_1 \, n^2 + 4L_f D_1(X)\left(8n^2 + k\right)\right]\\
&\quad + 2D_\infty(A)\left[\|\nabla g_{\beta_0}(Aw_1) - \gamma_0\|_1 \, m^2 + \frac{10 D_2(A)\, m^{3/2}}{\beta_0} k^{3/2}\right] && (105)
\end{aligned}$$

where we use Lemma F.1 for the second inequality.

Combining this with the bound on the smoothness term (B) from (48) gives the desired result:

$$\begin{aligned}
S_{\beta_k}(w_{k+1}) &\le \frac{2D_\infty(X)}{k(k+1)}\Bigg\{\|\nabla f(Xw_1) - \alpha_0\|_1 \, n^2 + 4L_f D_1(X)\left(8n^2 + k\right)\\
&\qquad + 2D_\infty(A)\left[\|\nabla g_{\beta_0}(Aw_1) - \gamma_0\|_1 \, m^2 + \frac{10 D_2(A)\, m^{3/2}}{\beta_0} k^{3/2}\right]\Bigg\}\\
&\quad + \frac{2D_{\mathcal{W}}^2}{k(k+1)}\left(\frac{\|X\| L_f}{n} k + \frac{\|A\|}{\beta_0} k\sqrt{k+1}\right)\\
&\le \frac{C_3}{k(k+1)} + \frac{C_2}{k+1} + \frac{C_1}{\sqrt{k+1}}, \quad \text{where} \quad
\begin{aligned}
C_3 &= 2n^2 D_\infty(X)\left(\|\nabla f(Xw_1) - \alpha_0\|_1 + 32 L_f D_1(X)\right)\\
&\quad + 2m^2 D_\infty(A)\|\nabla g_{\beta_0}(Aw_1) - \gamma_0\|_1\\
C_2 &= 8L_f D_1(X) D_\infty(X) + 2n^{-1} L_f \|X\| D_{\mathcal{W}}^2\\
C_1 &= \beta_0^{-1}(2D_{\mathcal{W}}^2\|A\| + 10 D_1(A)).
\end{aligned}
\end{aligned}$$

F Supporting Lemmas

Lemma F.1. *Let $\rho_n = 1 - \frac{1}{n}$ and $\rho_m = 1 - \frac{1}{m}$, $m, n \geq 1$. We present the following bounds:*

a) $\sum_{i=1}^{k} i\rho_n^i < n^2 \quad$ *and* $\quad \sum_{i=1}^{k} i\rho_m^i < m^2$

b) $\sum_{i=1}^{k} i\rho_n^{i/2} \log i < 16n^3$

Proof. **a)** Note that since $\rho_n \in [0, 1)$, $\sum_{i=1}^{k} i\rho_n^i \leq \sum_{i=1}^{k} i\rho_n^{i-1}$. Furthermore,

$$\sum_{i=1}^{k} i\rho_n^{i-1} \leq \sum_{i=1}^{\infty} i\rho_n^{i-1} = \sum_{i=1}^{\infty} \frac{\partial \rho_n^i}{\partial \rho_n} = \frac{\partial \sum_{i=1}^{\infty} \rho_n^i}{\partial \rho_n} = \frac{\partial \left[\frac{1}{1-\rho_n} - 1\right]}{\partial \rho_n} = \frac{1}{(1-\rho_n)^2} = n^2, \tag{106}$$

where the inequality comes from all terms being non-negative, and the second equality comes from the fact that the infinite sum exists for any $\rho_n \in (-1, 1)$ and is the Taylor series expansion of $\frac{1}{1-\rho_n}$.

b) Use the loose bound $\log i < i + 1$ and the fact that $\sqrt{\rho_n} \in [0, 1)$:

$$\sum_{i=1}^{k} i\rho_n^{i/2} \log i \leq \sum_{i=1}^{\infty} i\rho_n^{i/2} \log i \leq \sum_{i=1}^{\infty} i(i+1)\sqrt{\rho_n}^{i-1} \tag{107}$$

$$= \frac{\partial^2 \sum_{i=2}^{\infty} \sqrt{\rho_n}^i}{\partial(\sqrt{p_n})^2} = \frac{\partial^2 \frac{1}{1-\sqrt{\rho_n}} - \sqrt{\rho_n} - 1}{\partial(\sqrt{p_n})^2} = \frac{2}{(1-\sqrt{\rho_n})^3} \tag{108}$$

where the inequalities and equalities follow the same reasoning as in point a). Further noting that

$$\frac{2}{(1-\sqrt{\rho_n})^3} = \frac{2(1+\sqrt{\rho_n})^3}{(1-\rho_n)^3} = 2n^3 \underbrace{(1+\sqrt{\rho_n})}_{\leq 2}{}^3 \leq 16n^3 \tag{109}$$

□

G Uniform Sparsest Cut Datasets

Table S1: Datasets used for Uniform Sparsest Cut experiments. "Deg." stands for "Degree," "# Constraints" refers to the number of constraints in the SDP relaxation, and the dimension of the decision variable w is $n \times n$.

	$\lvert V\rvert = n$	$\lvert E\rvert$	Avg. Node Deg.	Max. Node Deg.	# Constraints
`mammalia-primate-association-13`	25	181	14	19	$\approx$ 6.90e3
`insecta-ant-colony1-day37`	55	1e3	42	53	$\approx$ 7.87e4
`insecta-ant-colony4-day10`	102	4e3	79	99	$\approx$ 5.15e5