
High Probability Bounds for Stochastic Continuous Submodular Maximization

Evan Becker

Jingdong Gao

Ted Zadouri

Baharan Mirzasoleiman

Department of Computer Science, University of California Los Angeles (UCLA)

Abstract

We consider maximization of stochastic monotone continuous submodular functions (CSF) with a diminishing return property. Existing algorithms only guarantee the performance *in expectation*, and do not bound the probability of getting a bad solution. This implies that for a particular run of the algorithms, the solution may be much worse than the provided guarantee in expectation. In this paper, we first empirically verify that this is indeed the case. Then, we provide the first *high-probability* analysis of the existing methods for stochastic CSF maximization, namely PGA, boosted PGA, SCG, and SCG++. Finally, we provide an improved high-probability bound for SCG, under slightly stronger assumptions, with a better convergence rate than that of the expected solution. Through extensive experiments on non-concave quadratic programming (NQP) and optimal budget allocation, we confirm the validity of our bounds and show that even in the worst-case, PGA converges to $OPT/2$, and boosted PGA, SCG, SCG++ converge to $(1 - 1/e)OPT$, but at a slower rate than that of the expected solution.

1 INTRODUCTION

While in general set functions are hard to optimize over, the subclass of submodular functions have useful properties that allow us to predictably achieve a certain approximation of the true optimal value in polynomial time [27]. Submodular functions exhibit a natural diminishing returns property and appear in a wide variety of applications such as sensor placement [15], graph cuts [12], data summarization [17], marketing [14] and clustering [23]. Thus, theoretical bounds on what optimization methods can achieve have important real-world implications. Continuous sub-

modular functions (CSF) extend the notion of submodularity to continuous domains and provide an interesting class of non-convex functions that are still tractable to optimize over [3]. CSFs have several applications, including non-convex/non-concave quadratic programming [3], robust budget allocation [25, 24], sensor energy management [3], online resource allocation [7], learning assignments [9], and e-commerce and advertising [19]. In addition, they enable solving many discrete submodular problems efficiently through their continuous relaxation such as multi-linear [5] or Lovas extensions [18]. This has motivated a body of work on optimizing CSFs [2, 3, 4].

More recently, constrained maximization of *stochastic* submodular functions has gained a lot of attention [11, 13, 20, 30]. A stochastic CSF can be formulated as the expected value of stochastic functions $\tilde{F} : \mathcal{X} \times \mathcal{Z} \rightarrow \mathbb{R}_+$:

$$\max_{\mathbf{x} \in \mathcal{C}} F(\mathbf{x}) = \max_{\mathbf{x} \in \mathcal{C}} \mathbb{E}_{\mathbf{z} \sim P} [\tilde{F}(\mathbf{x}, \mathbf{z})], \quad (1)$$

where $\mathcal{C} \subseteq \mathbb{R}_+^d$ is a bounded convex set, $\mathbf{x} \in \mathcal{X}$ is the optimization variable, and $\mathbf{z} \in \mathcal{Z}$ is a random variable drawn from a (potentially unknown) distribution P . Note that Problem (1) only assumes that $F(\mathbf{x})$ is DR-submodular, and not necessarily the stochastic functions $\tilde{F}(\mathbf{x}, \mathbf{z})$. The continuous greedy algorithm [3] can produce arbitrarily bad solutions for Problem (1), due to the non-vanishing variance of gradient approximations [11]. To address this, Projected Gradient Ascent (PGA) with diminishing step-sizes is first shown to provide a $[OPT/2 - \epsilon]$ guarantee [11]. Later, Stochastic Continuous Greedy (SCG) suggested to reduce the noise of gradient approximations via a momentum term and provided a tight $[(1 - 1/e)OPT - \epsilon]$ guarantee [20]. This work was followed by Stochastic Continuous Greedy++ (SCG++), which improved the complexity of SCG [13], by leveraging a variance reduction technique [8]. Most recently, boosted PGA algorithm using a non-oblivious function is proposed [30], which also achieves a $[(1 - 1/e)OPT - \epsilon]$ approximation guarantee.

However, the above algorithms only guarantee the performance of the solution *in expectation*. This implies that it is indeed possible that for a particular run of the algorithms, the optimizer gets extremely unlucky with its gradient estimates, and return a solution that is drastically worse than the provided guarantee in expectation. Indeed,

as we confirm by our experiments, all the algorithms for stochastic CSF, namely PGA, boosted PGA, SCG, and SCG++, may have a very high variance in their returned solution, as the noise gets larger. Crucially, the provided expectation bounds do not provide much insight, besides perhaps a basic Markov inequality, into the probability of getting these bad solutions. This is because expected guarantees rely on bounding the variance of the gradient estimation error, and cannot bound the total accumulated error required for deriving high probability bounds.

In this paper, we address the above question by providing the first high probability analysis of the existing methods for stochastic CSF maximization. High-probability bounds have been explored very recently for the most popular optimization methods, namely, SGD [10], and momentum SGD [16]. But, deriving high-probability bounds for submodular optimization has remained unaddressed. Different than the analysis of the expectation bounds for stochastic CSF maximization algorithms [11, 13, 21], our analysis leverages two different strategies to bound the distance between the algorithmic solution and the optimal value. The first strategy is using a martingale process to model functions of the gradient noise, allowing for the use of Azuma-Hoeffding inequality to provide high-probability bounds on the algorithmic solution. The second strategy is to use Chebyshev’s inequality to bound sum of squared errors in gradient estimators, with a high probability. Table 1 summarizes our results.

Our contributions are as follows. We derive the first high-probability analysis for stochastic CSF methods (under the same assumptions used for their expectation bounds), and show that after K queries to the stochastic gradient oracle:

- For Projected Gradient Ascent (PGA) [11] and Boosted PGA [30], the lower-bound on the average function value during a run of the algorithm converges at rate $\mathcal{O}(\frac{1}{K^{1/2}})$.
- For Stochastic Continuous Greedy (SCG) [20], the lower-bound on the final solution converges at rate $\mathcal{O}(\frac{\delta}{K^{1/3}})$, where δ depends on the confidence threshold.
- For Stochastic Continuous Greedy++ (SCG++) [13], the lower-bound on the final solution converges at rate $\mathcal{O}(\frac{\delta}{K^{1/4}})$, where δ depends on the confidence threshold.
- Under the sub-Gaussian assumption on the stochastic gradient oracle, we derive an improved high-probability bound on the final solution of SCG that converges to $(1 - \frac{1}{e})OPT$ at a faster $\mathcal{O}(\frac{1}{K^{1/2}})$ rate. Interestingly, this rate even exceeds the rate of convergence to the *expected* solution provided by [20]. Our analysis involves providing the first high probability bound for adaptive momentum optimization methods, which can be applied to other smooth function classes to provide superior convergence and generalization properties [26]. Hence, it is of independent interest.

- Our extensive experiments on a non-concave quadratic programming example (NQP) and a realistic optimal budget allocation problem confirm the validity of our bounds and show that even in the worst-case PGA still converges to the $OPT/2$, and boosted PGA, SCG, SCG++ still converge to $(1 - 1/e)OPT$, but at a slower rate.

Our results characterize the full distribution of the solutions for stochastic CSF maximization methods. In doing so, they allow an algorithm designer to answer questions about worst and best-case performance and even make modifications to mitigate the risk of getting a bad solution.

2 RELATED WORK

Deterministic continuous submodular maximization.

Maximizing deterministic continuous submodular functions have been first studied by Wolsey [27]. More recently, [4] proposed a multiplicative weight update algorithm that achieves $(1 - 1/e - \epsilon)$ approximation guarantee after $\tilde{O}(n/\epsilon^2)$ oracle calls to gradients of a monotone smooth twice differentiable DR-submodular function, subject to a polytope constraint (n is the ground set size). Later, a conditional gradient method similar to the continuous greedy algorithm is shown to obtain a similar approximation factor after $O(n/\epsilon)$ oracle calls to gradients of monotone DR-submodular functions subject to a down-closed convex body [3]. Such methods, however, require exact computation of the gradient of the function, which is not provided in the stochastic setting.

Stochastic continuous submodular maximization.

For stochastic continuous submodular maximization, conditional gradient methods may lead to arbitrarily poor solutions, due to the high noise variance [11]. While the noise variance can be reduced by averaging the gradient over a (large) mini-batch of samples at each iteration, averaging considerably increases the computational complexity of each iteration and becomes prohibitive in many applications. To address this, stochastic proximal gradient method are first proposed. In particular, [11] showed that when the expected function is monotone and DR-submodular, Projected Gradient Ascent (PGA) provides a $OPT/2 - \epsilon$ guarantee in $\mathcal{O}(1/\epsilon^2)$ iterations. Later, [20] introduced Stochastic Continuous Greedy (SCG), which reduces the noise of gradient approximations via exponential averaging and achieves a $(1 - 1/e)OPT - \epsilon$ guarantee in expectation after $\mathcal{O}(1/\epsilon^3)$ iterations. More recently, Stochastic Continuous Greedy++ (SCG++) improved the complexity of SCG to $\mathcal{O}(1/\epsilon^2)$, by using a stochastic path-integrated differential estimator (SPIDER) [8] to reduce the variance of the stochastic gradient. Most recently, boosted PGA algorithm using a non-oblivious function is proposed [30], which also achieves a $[(1 - 1/e)OPT - \epsilon]$ approximation guarantee in $\mathcal{O}(1/\epsilon^2)$ iterations. Existing works, however, only provide guarantees in expectation and cannot deliver any insight on the distribution of the solutions or worst-case analysis.

Table 1: Comparison of existing expectation and our high-probability bounds for three stochastic monotone DR-submodular maximization algorithms, namely PGA, SCG, SCG++. Here k is the number of queries to the stochastic gradient oracle $\nabla \tilde{F}$. Note that while our original bound is not tight for SCG, by using the slightly stronger condition of a sub-Gaussian gradient noise one can achieve an $\mathcal{O}(1/K^{1/2})$ bound (see Sec. 4.3.1).

Algorithm	Expectation Bound	Original Noise Assumptions	Bound Converges w.h.p.?
PGA [11]	$(\frac{1}{2})OPT - \mathcal{O}(\frac{1}{K^{1/2}})$	$\nabla \tilde{F}$ bounded	Yes, at $\mathcal{O}(1/K^{1/2})$ rate
Boosted PGA [30]	$(1 - \frac{1}{e})OPT - \mathcal{O}(\frac{1}{K^{1/2}})$	$\nabla \tilde{F}$ bounded	Yes, at $\mathcal{O}(1/K^{1/2})$ rate
SCG [20]	$(1 - \frac{1}{e})OPT - \mathcal{O}(\frac{1}{K^{1/3}})$	$Var(\nabla \tilde{F})$ bounded, sub-Gaussian*	Yes*, at $\mathcal{O}(1/K^{1/2})$ rate
SCG++ [13]	$(1 - \frac{1}{e})OPT - \mathcal{O}(\frac{1}{K^{1/2}})$	$\tilde{F}, \nabla \tilde{F}, \nabla^2 \tilde{F}, \log(p(\mathbf{z}))$ bounded	Yes, at $\mathcal{O}(1/K^{1/4})$ rate

High-probability bounds for stochastic submodular minimization. Very recently, [28] studied an extension of the stochastic submodular *minimization* problem, namely, the stochastic L^{\natural} -convex [22] minimization problem. L^{\natural} -convex functions are reduced to submodular functions when the ground set size is 2. Specifically, [28] developed a polynomial time algorithm that returns a near-optimal solution with a high probability. The proposed method relies on the Lovász extension of an L^{\natural} -convex function to transform the original problem to an equivalent continuous convex optimization problem, and applies the stochastic subgradient method to solve the continuous convex problem. To the best of our knowledge, high-probability bounds for continuous submodular *maximization* have not been explored before.

3 BACKGROUND AND PROBLEM FORMULATION

Continuous submodular functions. We start by reviewing definition of submodularity for set functions. A set function $f : 2^V \rightarrow \mathbb{R}_+$, defined on the ground set V , is submodular if for all subsets $A, B \subseteq V$, we have that

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B). \quad (2)$$

The notion of submodularity can be extended to continuous domains. A continuous function $F : \mathcal{X} \rightarrow \mathbb{R}_+$ defined on the set $\mathcal{X} = \prod_{i=1}^n \mathcal{X}_i$, where each \mathcal{X}_i is a compact subset of \mathbb{R}_+ , is continuous submodular if for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ we have

$$F(\mathbf{x}) + F(\mathbf{y}) \geq F(\mathbf{x} \vee \mathbf{y}) + F(\mathbf{x} \wedge \mathbf{y}). \quad (3)$$

Here, $\mathbf{x} \vee \mathbf{y} := \max(\mathbf{x}, \mathbf{y})$ is component-wise maximum and $\mathbf{x} \wedge \mathbf{y} := \min(\mathbf{x}, \mathbf{y})$ is component-wise minimum operations. A submodular function F is monotone on \mathcal{X} , if for every $\mathbf{x}, \mathbf{y} \in \mathcal{X}, \mathbf{x} \leq \mathbf{y}$ we have that $F(\mathbf{x}) \leq F(\mathbf{y})$. A function F defined over \mathcal{X} satisfies the diminishing returns (DR) property, if for every $\mathbf{x}, \mathbf{y} \in \mathcal{X}, \mathbf{x} \leq \mathbf{y}$, and any standard basis vector $e_i \in \mathbb{R}^n$ and any $k \in \mathbb{R}_+$ s.t. $(ke_i + \mathbf{x}) \in \mathcal{X}$ and $(ke_i + \mathbf{y}) \in \mathcal{X}$, it holds that

$$f(ke_i + \mathbf{x}) - f(\mathbf{x}) \geq f(ke_i + \mathbf{y}) - f(\mathbf{y}). \quad (4)$$

When F is twice-differentiable, DR-submodularity implies that all diagonal entries of the Hessian are non-positive [3].

I.e.,

$$\forall i = j, \quad \forall \mathbf{x} \in \mathcal{X} \quad \frac{\partial^2 F(\mathbf{x})}{\partial x_i \partial x_j} \leq 0. \quad (5)$$

Stochastic continuous submodular maximization. In this work, we focus on constrained maximization of stochastic continuous DR-submodular functions. Formally, our goal is to find \mathbf{x}^* that maximizes the expected value $F(\mathbf{x})$ of the stochastic function $\tilde{F}(\mathbf{x}, \mathbf{z})$ over \mathbf{x} , where the expectation is with respect to the random variable \mathbf{Z} :

$$\max_{\mathbf{x} \in \mathcal{C}} F(\mathbf{x}) := \max_{\mathbf{x} \in \mathcal{C}} \mathbb{E}_{\mathbf{z} \sim P} [\tilde{F}(\mathbf{x}, \mathbf{z})], \quad (6)$$

where $\mathcal{C} \subseteq \mathbb{R}_+$ is a convex compact set, and \mathbf{z} is the realization of the random variable \mathbf{Z} drawn from a distribution P . We assume that the expected objective function $F(\mathbf{x})$ is monotone and DR-submodular and the stochastic functions $\tilde{F}(\mathbf{x}, \mathbf{z})$ may not be monotone nor submodular. We denote by $OPT \triangleq \max_{\mathbf{x} \in \mathcal{C}} F(\mathbf{x})$ the optimal value of $F(\mathbf{x})$ over \mathcal{C} .

4 HIGH-PROBABILITY BOUNDS FOR STOCHASTIC CONTINUOUS SUBMODULAR MAXIMIZATION

Next, we discuss our high-probability bounds for stochastic CSF maximization algorithms, namely Projected Gradient Ascent (PGA), boosted PGA, Stochastic Continuous Greedy (SCG), and Continuous Greedy++ (SCG++).

4.1 Projected Gradient Ascent

We start by analyzing the worst-case performance of the PGA method which achieves a $[OPT/2 - \epsilon]$ approximation in expectation, in $\mathcal{O}(1/\epsilon^2)$ iteration [11]. PGA starts from an initial estimate $\mathbf{x}_0 \in \mathcal{C}$. Then at every iteration t , it takes a step in the direction of the noisy gradient $\mathbf{g}_t = \nabla \tilde{F}(\mathbf{x}_t, \mathbf{z}_t)$, and projects the solution onto the convex set \mathcal{C} . The update rule at step t takes the following form:

$$\mathbf{x}_{t+1} = \mathcal{P}_{\mathcal{C}}(\mathbf{x}_t + \mu_t \mathbf{g}_t), \quad (7)$$

where, μ_t is the diminishing learning rate at step t , and $\mathcal{P}_{\mathcal{C}}$ denotes the Euclidean projection onto the set \mathcal{C} . The pseudocode is provided in Appendix B.

Hassani et al. [11] provided a lower bound on the expected function value, $\mathbb{E}[F(\mathbf{x}_\tau)]$, at a time-step τ , sampled uniformly at random from $\{1, \dots, T\}$. Important to note, however, is that the derived expectation is not only over this random variable τ , but *also the noise* coming from the gradient estimates. This implies that it is possible that the optimizer gets extremely unlucky with its gradient estimates, in which case no \mathbf{x}_t satisfies the lower bound (for example, consider the unlikely but still possible scenario where $\forall t, \mathbf{g}_t = 0$). In Theorem 1, we provide an exact answer for how unlikely a failure event like this would be.

To do so, we make similar assumptions to [11]:

Assumption 1. *The diameter of the constraint set \mathcal{C} is bounded by D . I.e., $\forall \mathbf{x}, \mathbf{y} \in \mathcal{C}$, we have*

$$\|\mathbf{x} - \mathbf{y}\| \leq D.$$

Assumption 2. *The function F is Lipschitz smooth with constant L , over \mathcal{X} . I.e., $\forall \mathbf{x}, \mathbf{y} \in \mathcal{C}$, we have*

$$\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|.$$

Assumption 3. *Stochastic gradients $\mathbf{g}_t = \nabla \tilde{F}(\mathbf{x}, \mathbf{z})$ are bounded in distance from their mean $\nabla F(\mathbf{x}_t) = \mathbb{E}[\mathbf{g}_t]$:*

$$\|\nabla F(\mathbf{x}_t) - \mathbf{g}_t\| \leq M.$$

The following theorem shows that for any fixed confidence interval p , the lower bound on $\sum_{t \in [T]} F(\mathbf{x}_t)/T$ will converge to $OPT/2$ at a rate of $\mathcal{O}(\sqrt{\frac{\log(1/(1-p))}{T}})$.

Theorem 1. *Consider running PGA for T iterations with step size of $\eta_t = \frac{2}{\sqrt{t}}$ with Assumptions 1,2,3 satisfied. Then with probability $p \geq 1 - \delta$, where $\delta \in [0, 1]$, the average function value returned by the algorithm is lower bounded by*

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T F(\mathbf{x}_t) &\geq \frac{1}{2} OPT - \frac{C}{\sqrt{T}} - DM \sqrt{\frac{\log(1/\delta)}{2T}} \quad (8) \\ &\geq \frac{1}{2} OPT - \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{T}}\right), \end{aligned}$$

where we denote the constant $C := \left(\frac{8(L+M)^2 + D^2}{8}\right)$.

Unlike the expectation bound provided in [11], Theorem 1 assures that with high probability, *at least one iterate* from a *single* algorithm run will be larger than the lower bound. Therefore, one could modify the default PGA algorithm to return the best iterate, $\max_{t \in [T]} F(\mathbf{x}_t)$, which is guaranteed to also be lower bounded with high probability by $OPT/2$. We note that even in the case where the true function evaluation is hard to compute, one can still find the best iterate with high probability given unbiased stochastic function evaluations $\tilde{F}(\mathbf{x}, \mathbf{z})$. Concretely, given the ordering of iterates from highest to lowest function value $\mathbf{x}_{[1]}, \dots, \mathbf{x}_{[T]}$,

consider the difference between the best two solutions $d := F(\mathbf{x}_{[1]}) - F(\mathbf{x}_{[2]})$. One can use a Hoeffding bound to determine the relatively small number of samples, m , needed to calculate $\bar{F}(\mathbf{x}) := \frac{1}{m} \sum \tilde{F}(\mathbf{x})$ for each iterate, such that $\bar{F}(\mathbf{x}_{[1]}) > F(\mathbf{x}_{[1]}) - d/2$ and $\bar{F}(\mathbf{x}_{[k]}) < F(\mathbf{x}_{[2]}) + d/2$ for all $k > 1$ occurs with very high probability. Alternatively, since $F(\mathbf{x}) \leq OPT \forall \mathbf{x}$, at least $\frac{r}{r+(1/2)OPT}$ fraction of solutions are greater than $\sum_{t \in [T]} F(\mathbf{x}_t)/T - r$ for a slack variable r . That is, with only k true function evaluations, at least one good solution is found with probability $p > 1 - (1 - r)^k$.

Theorem 1 relies on diminishing returns and smoothness of F , along with the bound on \mathcal{C} to first bound the difference between $F(\mathbf{x}_t)$ and $F(\mathbf{x}^*)$ based on the inner product between gradient noise and $\mathbf{x}_t - \mathbf{x}^*$. However, instead of taking the expectation of this inequality, it directly shows that with Assumption 3, these random products satisfy the conditions of a c-lipschitz Martingale difference sequence. This allows using standard high probability bounds (Azuma-Hoeffding). See Appendix A.2 for the full proof.

We note that Assumption 3 is stronger than simply bounded variance and is necessary to apply the Azuma-Hoeffding inequality. However if $\mathbf{g}_t = \nabla F(\mathbf{x}_t) + \mathbf{z}_t$ with each \mathbf{z}_t being zero mean and Sub-Gaussian, a similar version of Theorem 1 can be derived following [10].

Corollary 1. *Consider the case where we set $\delta = \exp(-\sqrt{T})$. Then the averaged function value of PGA is lower bounded with probability $p \geq 1 - 2 \exp(-\sqrt{T})$ by:*

$$\frac{1}{T} \sum_{t=1}^T F(\mathbf{x}_t) \geq \frac{1}{2} OPT - \mathcal{O}\left(\frac{1}{T^{1/4}}\right). \quad (9)$$

We see that as $T \rightarrow \infty$ we have both $p \rightarrow 1$ and $\mathcal{O}(1/T^{1/4}) \rightarrow 0$. Thus, our lower-bound is tight.

4.2 Boosted Projected Gradient Ascent

Very recently, boosted PGA [30] is proposed to provide $[(1 - \frac{1}{e} - \epsilon^2)OPT]$ approximation in expectation, in $\mathcal{O}(1/\epsilon^2)$ iterations. The idea is to find an auxiliary (non-oblivious) function that can provide a better approximation guarantee than the original DR-submodular function. Then the stochastic gradients of the non-oblivious function F' (instead of the stochastic gradient of the original DR-submodular function F) are leveraged by PGA. Specifically, [30] used the following non-oblivious function F' and its gradient $\nabla F'$:

$$F'(\mathbf{x}) := \int_0^1 \frac{e^{(s-1)}}{s} F(s * \mathbf{x}) ds, \quad (10)$$

$$\nabla F'(\mathbf{x}) := \int_0^1 e^{(s-1)} \nabla F(s * \mathbf{x}) ds. \quad (11)$$

This has the nice property that $\langle \mathbf{y} - \mathbf{x}, \nabla F'(\mathbf{x}) \rangle \geq (1 - 1/e)F(\mathbf{y}) - F(\mathbf{x})$, which guarantees $(1 - 1/e)OPT$ approximation. Additionally, when the original function

is Lipschitz smooth (Assumption 2), $F'(\mathbf{x})$ is Lipschitz smooth with constant $L' = L(1 + 1/e)$. To efficiently approximate $\nabla F'$ given a noisy gradient estimate $\nabla \tilde{F}'$, [29] uses the following estimator:

$$\nabla \tilde{F}'(\mathbf{x}_t) := (1 - \frac{1}{e}) \nabla \tilde{F}'(s_t * \mathbf{x}_t). \quad (12)$$

Here, s_t is independently sampled from a distribution $\mathbb{P}(S \leq s) = \int_0^s \frac{1}{1-e^{-1}} \mathbb{1}(u \in [0, 1]) du$, where $\mathbb{1}$ is indicator function.

The following theorem provides a lower bound on $\sum_{t \in [T]} F(\mathbf{x}_t)/T$, with high probability.

Theorem 2. *Consider running boosted PGA for T iterations with step size of $\eta_t = \frac{2}{\sqrt{t}}$ with Assumptions 1,2,3 satisfied. Then with probability $p \geq 1 - \delta$, where $\delta \in [0, 1]$, the average function value returned by the algorithm is lower bounded by*

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T F(\mathbf{x}_t) &\geq (1 - \frac{1}{e}) OPT - \frac{C'}{\sqrt{T}} - DM' \sqrt{\frac{\log(1/\delta)}{2T}} \\ &\geq (1 - \frac{1}{e}) OPT - \mathcal{O}\left(\sqrt{\frac{\log(1/\delta)}{T}}\right), \end{aligned} \quad (13)$$

where we denote the constant $C' := \left(\frac{8(L'D + M')^2 + D^2}{8}\right)$, and constant $M' := (M + 2LD) \left(1 - \frac{1}{e}\right)$.

Note that this bound has the same rate of convergence as Theorem 1 up to a constant factor. This similarity also means a result parallel to Corollary 1 can be derived, demonstrating that this algorithm will also converge with $p \rightarrow 1$ as $T \rightarrow \infty$. The proof of Theorem 2 follows a similar structure to Theorem 1. We bound the sum of differences $F(\mathbf{x}_t) - F(\mathbf{x}^*)$ by a Martingale sequence with a bounded difference property. The key distinction in the non-oblivious case is that we must bound gradients from $\nabla F'(\mathbf{x})$ as well as $\nabla \tilde{F}'(\mathbf{x})$. We defer the full proof to Appendix A.3.

Guarantees for weakly submodular functions. We note that Theorems 1, 2 for PGA and boosted PGA can be extended to γ -weakly DR-submodular functions, where we have $\gamma = \inf_{\mathbf{x} \leq \mathbf{y}} \inf_i (|\nabla F(\mathbf{x})|_i / |\nabla F(\mathbf{y})|_i)$. This setting produces bounds with the same rate of convergence up to a constant, to $(\frac{\gamma^2}{1+\gamma^2}) OPT$ and $(1 - e^{-\gamma}) OPT$ for PGA and Boosted PGA, respectively. Note that $\gamma = 1$ indicates a differentiable and monotone DR-submodular function.

4.3 Stochastic Continuous Greedy

Next, we analyze the worst-case performance of the Stochastic Continuous Greedy (SCG) algorithm. SCG uses a momentum term to reduce the noise of gradient approximations. It starts from $\mathbf{x}_0 = \mathbf{0}$, and at every iteration t , calculates:

$$\bar{\mathbf{g}}_{t+1} = (1 - \rho_t) \bar{\mathbf{g}}_t + \rho_t \nabla \tilde{F}'(\mathbf{x}_t, \mathbf{z}_t), \quad (14)$$

where ρ_t is a stepsize which approaches zero as t approaches infinity, and $\bar{\mathbf{g}}_0 = \mathbf{0}$. The SCG is then ascent in the direction:

$$\mathbf{v}_t \leftarrow \arg \max_{\mathbf{v} \in \mathcal{C}} \{\langle \bar{\mathbf{g}}_t^T, \mathbf{v} \rangle\}, \quad (15)$$

using the following updates rule with step-size $1/T$:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \frac{1}{T} \mathbf{v}_t. \quad (16)$$

The stepsize $\frac{1}{T}$ and the initialization $\mathbf{x}_0 = \mathbf{0}$ ensure that after T iterations the variable \mathbf{x}_T ends up in the convex set \mathcal{C} . The pseudocode is provided in Appendix B.

SCG provides a tight $[(1 - 1/e)OPT - \epsilon]$ guarantee in expectation for the last iterate T , with $\mathcal{O}(1/\epsilon^3)$ stochastic gradient computations [20]. But, similar to PGA [11], the expected guarantee of SCG does not tell us how frequently *bad solutions*, with $F(\mathbf{x}_T) < [(1 - 1/e)OPT - \epsilon]$ are returned.

Here, we answer the above question by providing a high-probability bound on the value of the final solution, $F(\mathbf{x}_T)$, returned by SCG. To do so, instead of assuming bounded gradient error (Assumption 3), we use the weaker assumption from [20] on the variance of the stochastic gradients:

Assumption 4. *Stochastic gradients have mean $\mathbb{E}[\mathbf{g}_t] = \nabla F(\mathbf{x}_t)$ and bounded variance:*

$$\mathbb{E}_{z \sim p} \left[\|\mathbf{g}_t - \nabla F(\mathbf{x})\|^2 \right] \leq \sigma^2.$$

Given Assumption 4, [20] showed that the variance of the momentum error, i.e., $\mathbb{E}[\|F(\mathbf{x}_t) - \bar{\mathbf{g}}_t\|^2]$, converges to zero, as t grows. However, this cannot be directly used to provide a high-probability bound on the value of the final solution (as we require the summation rather than the expectation of error terms). To address this, instead of using the bound on the variance of the noisy gradient at step t , we apply Chebyshev's inequality to bound the probability of the noisy gradient to be far away from its expectation. Then, we use a union bound on iterations $t \in [T - 1]$ to get the next Lemma:

Lemma 1. *Consider the Stochastic Continuous Greedy algorithm, with $\rho_t = \frac{4}{(t+8)^{2/3}}$. Under Assumptions 1, 2, 4, we have the following high probability bound on the total variance of the noisy gradients during $t \in \{0, \dots, T - 1\}$:*

$$\mathbb{P} \left(\sum_{t=0}^{T-1} \|\nabla F(\mathbf{x}_t) - \bar{\mathbf{g}}_t\|^2 \leq \delta^2 \sum_{t=0}^{T-1} \frac{Q}{(t+9)^{2/3}} \right) \geq 1 - \frac{T}{\delta^2}, \quad (17)$$

where $Q := \max\{\|\nabla F(\mathbf{x}_0) - \bar{\mathbf{g}}_0\|^2 9^{2/3}, 16\sigma^2 + 3L^2 D^2\}$, $\delta > 0$.

The proof can be found in Appendix A.4. Note that Lemma 1 does not rely on taking the expectation of the function values or gradient approximations. Equipped with Lemma 1, we derive the following lower-bound on $F(\mathbf{x}_T)$, by recursively bounding the distance of the iterates to the optimum, $F(\mathbf{x}^*)$.

Theorem 3. Consider the Stochastic Continuous Greedy algorithm, with $\rho_t = \frac{4}{(t+8)^{2/3}}$. Under Assumptions 1, 2, 4, we have that with probability greater than $1 - \frac{T}{\delta^2}$:

$$\begin{aligned} F(\mathbf{x}_T) &\geq \left(1 - \frac{1}{e}\right)F(\mathbf{x}^*) - \delta \frac{2Q^{1/2}D}{T^{1/3}} - \frac{LD^2}{2T^2} \\ &= \left(1 - \frac{1}{e}\right)OPT - \mathcal{O}\left(\frac{\delta}{T^{1/3}}\right), \end{aligned} \quad (18)$$

where $Q := \max\left\{\|\nabla F(\mathbf{x}_0) - \bar{\mathbf{g}}_0\|^2 9^{2/3}, 16\sigma^2 + 3L^2D^2\right\}$, $\delta > 0$.

The proof can be found in Appendix A.4. Unlike our Theorem 1 for PGA, we see that for any fixed confidence threshold p implying $\delta = \sqrt{\frac{T}{1-p}}$, our lower bound does not converge. This is a direct result of the weakening of the noise assumption, since the gradient noise may have bounded variance but not be bounded itself. Next, we provide an improved bound when gradient noise is sub-Gaussian.

4.3.1 Improved Bound under Sub-Gaussian Noise

The weak assumptions on the noisy gradients (only bounded variance from Assumption 4) make it difficult to apply typical Martingale or sub-Gaussian inequalities. If instead, we assume that the noisy gradient approximations are sub-Gaussian, we arrive at a surprisingly tight lower bound. First we describe the sub-Gaussian noise assumption as follows:

Assumption 5. Stochastic gradients have mean $\mathbb{E}[\mathbf{g}_t] = \nabla F(\mathbf{x}_t)$ and $\hat{\mathbf{z}}_t := \|\mathbf{g}_t - \nabla F(\mathbf{x})\|$ is sub-Gaussian. I.e. for $\sigma > 0$, we have:

$$\mathbb{E}(e^{\lambda \hat{\mathbf{z}}_t^2}) \leq e^{\sigma^2 \lambda^2 / 2} \quad \forall \lambda \in \mathbb{R} \quad (19)$$

When using the SCG algorithm under this new assumption, we can derive the following high probability bound:

Theorem 4. Consider the Stochastic Continuous Greedy algorithm, with $\rho_t = \frac{1}{t^\alpha}$ where $\alpha \in (0, 1)$. Then under Assumptions 1, 2, 5, with probability greater than $1 - \delta$:

$$\begin{aligned} F(\mathbf{x}_T) &\geq \left(1 - \frac{1}{e}\right)OPT - \frac{2DK\sigma\sqrt{\log(1/\delta)}}{T^{1/2}} \\ &\quad - \left(\frac{4K+1}{2}\right)\frac{LD^2}{T} \\ &= \left(1 - \frac{1}{e}\right)OPT - \mathcal{O}\left(\frac{\sqrt{\log(1/\delta)}}{T^{1/2}}\right), \end{aligned} \quad (20)$$

where $K := \frac{1}{1-\alpha}\Gamma\left(\frac{1}{1-\alpha}\right)$.

At a high level, the proof of Theorem 4 expands the momentum into a weighted summation of gradient approximations, which after some careful manipulations can be treated as a summation of sub-Gaussian variables. This is to our knowledge the first such result for adaptive momentum optimization methods, where the momentum can change over

time. Notably, it is general enough to be used even in the context of other smooth function classes. Adaptive momentum enjoys some superior convergence and generalization properties [26]. See appendix A.5 for the detailed proof.

Notably, the bound in Theorem 4 has a faster convergence rate than the original expectation bound of [20], i.e., $(1 - \frac{1}{e})OPT - \mathcal{O}(1/T^{1/3})$. The new bound suggests that for certain well conditioned problems, SCG can achieve the same convergence rate as SCG++. In our experiments in Sec. 5, we show that empirically the distribution of solutions of SCG do converge when gradient noise is (sub-)Gaussian.

4.4 Stochastic Continuous Greedy++

Finally, we analyze the worst-case performance of the Stochastic Continuous Greedy++ (SCG++), which aims to speed up SCG, using a stochastic path-integrated differential estimator (SPIDER) [8] for the gradient. SCG++ assumes that the probability distribution of the random variable \mathbf{z} depends on the variable \mathbf{x} and may change during the optimization. To obtain an unbiased gradient estimator $\hat{\mathbf{g}}_t$ with a reduced variance, SCG++ uses a mini-batch of samples to first get an unbiased estimate of the Hessian, $\tilde{\nabla}_t^2$:

$$\tilde{\nabla}_t^2 = \frac{1}{|\mathcal{M}|} \sum_{(a, \mathbf{z}(a)) \in \mathcal{M}} \tilde{\nabla}^2 F(\mathbf{x}(a), \mathbf{z}(a)), \quad (22)$$

where a is selected uniformly at random from $[0, 1]$, $\mathbf{z}(a)$ is a random variable with probability distribution $p(\mathbf{z}(a); \mathbf{x}(a))$, $\mathbf{x}(a) := a\mathbf{x}_t + (1-a)\mathbf{x}_{t-1}$, and \mathcal{M} is a mini-batch containing $|\mathcal{M}|$ samples of random tuple $(a, \mathbf{z}(a))$. Then, SCG++ uses the Hessian estimate to recursively calculate unbiased estimates of the gradient, based on the gradient differences $\tilde{\Delta}^t$:

$$\tilde{\Delta}^t := \tilde{\nabla}_t^2(\mathbf{x}_t - \mathbf{x}_{t-1}). \quad (23)$$

A gradient estimate, $\hat{\mathbf{g}}_t$, with a reduced variance is then calculated as the initial noisy gradient estimate plus the sum of all the gradient differences up to time t :

$$\hat{\mathbf{g}}_t = \nabla \tilde{F}(\mathbf{x}_0, \mathcal{M}_0) + \sum_{i=1}^t \tilde{\Delta}^i. \quad (24)$$

With the above gradient estimate $\hat{\mathbf{g}}_t$, SCG++ starts from $\mathbf{x}_0 = \mathbf{0}$, and at each iteration t , performs a standard Frank-Wolfe step with step-size $\frac{1}{T}$. The full update sequence is provided in Appendix B.

SCG++ converges in expectation to the same $[(1 - 1/e)OPT - \epsilon]$ approximation as SCG, but using only $\mathcal{O}(1/\epsilon^2)$ stochastic gradient evaluations and $\mathcal{O}(1/\epsilon)$ calls to the linear optimization oracle [13]. However, similar to PGA and SCG, the expected approximation guarantee of SCG++ does not show the probability of returning a final solution that is much lower than the expected solution.

The analysis of SCG++ requires stronger assumptions than SCG. Besides the monotone DR-submodularity of F and

the bounded diameter on \mathcal{C} (Assumption 1), we use the same assumptions originally used to analyze SCG++ in [13]:

Assumption 6. *The function value at the origin is:*

$$F(\mathbf{0}) \geq 0. \quad (25)$$

Assumption 7. *The stochastic function $\tilde{F}(\mathbf{x}, \mathbf{z})$, its gradient, and its Hessian are bounded:*

$$\tilde{F}(\mathbf{x}, \mathbf{z}) \leq B, \quad (26)$$

$$\left\| \nabla \tilde{F}(\mathbf{x}, \mathbf{z}) \right\| \leq G_{\tilde{F}}, \quad (27)$$

$$\left\| \nabla^2 \tilde{F}(\mathbf{x}, \mathbf{z}) \right\| \leq L_{\tilde{F}}. \quad (28)$$

Assumption 8. *The function $\log p(\mathbf{z})$ has the following bounds on gradient and Hessian:*

$$\mathbb{E} \left(\left\| \nabla \log p(\mathbf{z}) \right\|^4 \right) \leq G_p^4, \quad (29)$$

$$\mathbb{E} \left(\left\| \nabla^2 \log p(\mathbf{z}) \right\|^2 \right) \leq L_p^2. \quad (30)$$

Assumption 9. *The Hessian of the stochastic function $\tilde{F}(\mathbf{x}, \mathbf{z})$ is L_2 Lipschitz continuous with constant $L_{2,\tilde{F}}$. The Hessian of the function $\log p(\mathbf{z})$ is L_2 Lipschitz continuous with constant $L_{2,p}$.*

$$\left\| \nabla^2 \tilde{F}(\mathbf{x}, \mathbf{z}) - \nabla^2 \tilde{F}(\mathbf{y}, \mathbf{z}) \right\| \leq L_{2,\tilde{F}} \|\mathbf{x} - \mathbf{y}\|, \quad (31)$$

$$\left\| \nabla^2 \log p(\mathbf{z}) - \nabla^2 \log p(\mathbf{z}') \right\| \leq L_{2,p} \|\mathbf{z} - \mathbf{z}'\|. \quad (32)$$

Under the above assumptions and with $\mathcal{O}(\epsilon^{-1})$ calls to the stochastic oracle per iteration, the variance of the gradient approximation $\hat{\mathbf{g}}_t$ converges to zero [13]. Instead of directly upper bounding variance as in [13], we use Chebyshev's inequality to prevent any expectations from appearing in our bound. Specifically we use the following Lemma:

Lemma 2. *Given SCG++ under Assumptions 1,5-8, we have the following high probability bound:*

$$\mathbb{P} \left(\sum_{t=0}^{T-1} \left\| \nabla F(\mathbf{x}_t) - \hat{\mathbf{g}}_t \right\|^2 \leq \delta^2 \sum_{t=0}^{T-1} \frac{2L^2 D^2}{t^2} \right) \leq 1 - \frac{T}{\delta^2}. \quad (33)$$

Lemma 2 allows us to directly bound the function value of last iterate of SCG++, $F(\mathbf{x}_T)$, with a high probability.

Theorem 5. *Consider applying SCG++ under Assumptions 1, 5-8. Then with probability $1 - \frac{T}{\delta^2}$:*

$$\begin{aligned} F(\mathbf{x}_T) &\geq \left(1 - \frac{1}{e}\right) F(\mathbf{x}^*) - \delta \frac{LD^2}{T^2} - \frac{LD^2}{2T^2} \\ &= \left(1 - \frac{1}{e}\right) OPT - \mathcal{O}\left(\frac{\delta}{T}\right). \end{aligned} \quad (34)$$

For a fixed probability threshold p we get the next Corollary:

Corollary 2. *For $\delta = \sqrt{\frac{T}{1-p}}$, with probability greater than p , we have:*

$$F(\mathbf{x}_T) \geq \left(1 - \frac{1}{e}\right) F(\mathbf{x}^*) - \frac{LD^2}{\sqrt{1-p}} \frac{1}{\sqrt{T}} - \frac{LD^2}{2T^2} \quad (35)$$

$$= \left(1 - \frac{1}{e}\right) OPT - \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) \quad (36)$$

SCG++ makes $\mathcal{O}(T)$ queries to the stochastic gradient oracle per iteration, and $K = T^2$ queries in total. Hence, with probability greater than p the bound in Corollary 2 becomes:

$$F(\mathbf{x}_T) \geq \left(1 - \frac{1}{e}\right) OPT - \mathcal{O}\left(\frac{1}{K^{1/4}}\right). \quad (37)$$

For any fixed confidence interval p , the lower bound still converges to $(1 - \frac{1}{e})OPT$, albeit at a slower rate. However, we believe that a tighter $1/\sqrt{K}$ high probability bound likely exists, as evident by our experimental results in Sec. 5.

5 NUMERICAL RESULTS

In our experiments, we first show that bad solutions of PGA, boosted PGA SCG, and SCG++ can be much worse than their expected values. Then, we validate our proposed bounds on three simulated and real-world datasets. In practice, due to measurement errors or inexact function calculations, the function and thus the gradient evaluations are often noisy. In such situations, our high probability bounds can effectively quantify the worst-case performance and be utilized to mitigate the risk of getting a bad solution.

5.1 Continuous Submodular Problems

First, we introduce three monotone continuous DR-submodular problems that we use in our experiments.

Non-convex/non-concave quadratic programming (NQP). NQP functions of the form $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{h}^T \mathbf{x}$ arise in many applications, including scheduling, inventory theory, and free boundary problems [2]. When all off-diagonal entries of \mathbf{H} are non-positive, the NQP is submodular.

For our experiment, we randomly generate $n = 100$ monotone DR-submodular NQP functions, where each $\mathbf{H} \in \mathbb{R}^{n \times n}$ is a symmetric matrix sampled uniformly from $[-100, 0]$. We further generated a set of $m = 50$ linear constraints to construct the positive polytope $\mathbb{P} = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{A}\mathbf{x} \leq \mathbf{b}, 0 \leq \mathbf{x} \leq \bar{\mathbf{u}}\}$, where entries in $\mathbf{A} \in \mathbb{R}^{m \times n}$ are uniformly sampled from $[0, 1]$, $\bar{\mathbf{u}} = \mathbf{1}$, and $\mathbf{b} = \mathbf{1}$. To make f monotone, we ensure the gradient of f is non-negative by setting $\mathbf{h} = -\mathbf{H}\bar{\mathbf{u}}$.

Optimal budget allocation with continuous assignments. The budget allocation problem can be modeled as a bipartite graph $(S, T; W)$, where S is a set of advertising channels and T is a set of customers. The edge weight $p_{st} \in W$

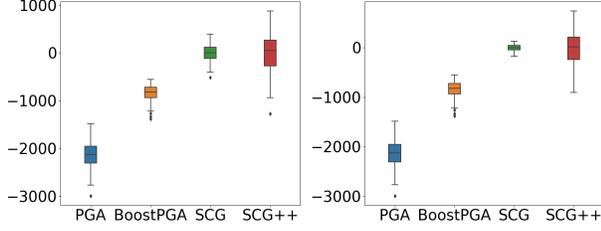


Figure 1: The distribution of $\min_{\tau \in [T]} F(\mathbf{x}_\tau) - \mathbb{E}[F(\mathbf{x}_\tau)]$ for PGA and boosted PGA, and $F(\mathbf{x}_T) - \mathbb{E}[F(\mathbf{x}_T)]$ for SCG and SCG++ on NQP, over 100 runs with $T = 5$ (left), and $T = 100$ (right). Bad solutions can be much worse than the expected values.

represent the influence probability of channel s on customer t . The objective is to maximize the total influence on the customers by allocating the budget to the set of advertising channels. The total influence on customer t from all the channels can be model by a monotone DR-submodular function $I_t(\mathbf{x}) = 1 - \prod_{(s,t) \in W} (1 - p_{st})^{\mathbf{x}_s}$, where $\mathbf{x}_s \in \mathbb{R}_+$ is the budget allocated to channel s . Then for a set of k advertisers, where $\mathbf{x}^i \in \mathbb{R}_+^S$ is the budget allocation of the i^{th} advertiser and $\mathbf{x} = [\mathbf{x}^1, \dots, \mathbf{x}^k]$, the overall objective is

$$g(\mathbf{x}) = \sum_{i=1}^k \alpha_i f(\mathbf{x}^i), \quad \text{with} \quad f(\mathbf{x}^i) = \sum_{t \in T} I_t(\mathbf{x}^i), \quad (38)$$

$$0 \leq \mathbf{x}^i \leq \bar{\mathbf{u}}^i, \quad \forall 1 \leq i \leq k,$$

where α_i is a constant weight coefficient and $\bar{\mathbf{u}}^i$ is the budget limit on each channel for the i^{th} advertiser.

For a real-world instance of the budget allocation problem, we use the Yahoo! Search Marketing Advertiser Bidding Data [1], which includes search keyword phrases and the bids placed on them by online customers. The dataset consists of 1,000 search keywords, 10,475 customers and 52,567 edges, where each edge between a keyword and customer represents the customer has bid on the keyword. A customer may bid on one phrase multiple times, and we use the frequency of a (phrase, customer) pair to measure the influence probability of the phrase on that customer. Additionally, we use the average bidding price across all the bids in the dataset as the limit on the budget of all the advertisers.

5.2 Bad Solutions and High-probability Bounds

Setup. We repeat every experiment 100 times, and use a step-size of $1e-4$, $1e-2$ for PGA methods on NQP and Yahoo! respectively. We set the step-size to $1/T$ for SCG and SCG++, and use batch size of T for SCG++. For SCG, we use $\frac{4}{(t+8)^{2/3}}$ as the momentum coefficient. For PGA methods, we randomly initialize $\mathbf{x}_0 \sim \mathcal{N}(0, 1)$ from a Gaussian distribution, and for SCG and SCG++ we initialize $\mathbf{x}_0 = \mathbf{0}$. Additionally, for PGA, boosted PGA, and SCG experiments, we add a noise sampled from a Gaussian distribution with mean 0 and standard deviation proportional to the gradient norms normalized by its dimensionality to

the queried gradients. SCG++ uses noisy estimates of the Hessian, hence we add a smaller Gaussian noise with mean 0 to the Hessian.

Bad solutions are far from expectation. First, we look at the solution of PGA, SCG, and SCG++ on NQP, to see how far the solution may be from the expected value. Note that when running each algorithm for T iterates, PGA returns the solution for a random iterate $\tau \in [T]$, and SCG, SCG++ return the solution of the final iterate T . Fig. 1 shows the distribution of $\min_{\tau \in [T]} F(\mathbf{x}_\tau)$ for PGA, and $F(\mathbf{x}_T)$ for boosted PGA, SCG and SCG++ for $T = 5$ (left), and $T = 100$ (right), over 100 runs. We see that bad solutions of the algorithms can be much worse than their expected value. While more number of iterations reduces the variance of SCG, we see that PGA and SCG++ have a very high variance even after $T = 100$ rounds (see Appendix C for more details). This shows the insufficiency of the expected guarantees, and confirms the necessity of high-probability analysis.

High-probability bounds. Next, we empirically confirm the validity of our high-probability bounds. To do so, we first apply PGA, Boosted PGA, SCG, and SCG++ to the continuous submodular problems discussed in Sec. 5.1. Then, we compare the empirical results with our bounds in Theorems 1, 2, 4, 5. Specifically, for each iteration t , we report the average utility up to t , i.e. $\frac{1}{t} \sum_{i=1}^t F(\mathbf{x}_i)$, for PGA; and the value of $F(\mathbf{x}_t)$ for Boosted PGA, SCG, and SCG++. To avoid the need for calculating the exact scaling constants in Theorems (e.g. L, D, K , etc.) and the true optimal value OPT (which determines the asymptote), we fit a line in the same form of the lower bound derived in Theorems 1, 2, 4, 5, to the output of the algorithms. For each line, c_1 corresponds to fraction of OPT the bound is converging to, while c_2 scales the rate of convergence depending on the problem-specific constants and desired confidence threshold. Specifically, for each algorithm we fit a line of the form $l(t) = c_1 - \frac{c_2}{\sqrt{t}}$. Importantly, as SCG++ uses batch size of $\mathcal{O}(T)$, an equivalent form of this fitted line is $l(t) = c_1 - \frac{c_2}{k^{1/4}}$, which is slower than the previous two algorithms. For c_1 , we use the same value for min, median, and different percentiles, by taking the average of the c_1 values obtained from the fitted lines for an algorithm on the same dataset. Using the above c_1 , we fit the curves again to get the corresponding c_2 for each line.

Fig. 2(a)-(h) show the median, minimum, and 90% percentile of utility over the course of training of each algorithm, compared to our predicted lower bounds. We see that our bounds closely match these utility statistics for various iterations of PGA, SCG, SCG++, and boosted PGA applied to different problems. Since for each percentile level the bound is of the same order, the differences in percentiles decrease as well (e.g. $\frac{a}{t^{1/2}} - \frac{b}{t^{1/2}} = \frac{c}{t^{1/2}}$). This effect can be seen as the minimum returned value across runs approaches the median and 90% returned values as the number of training iterations increase.

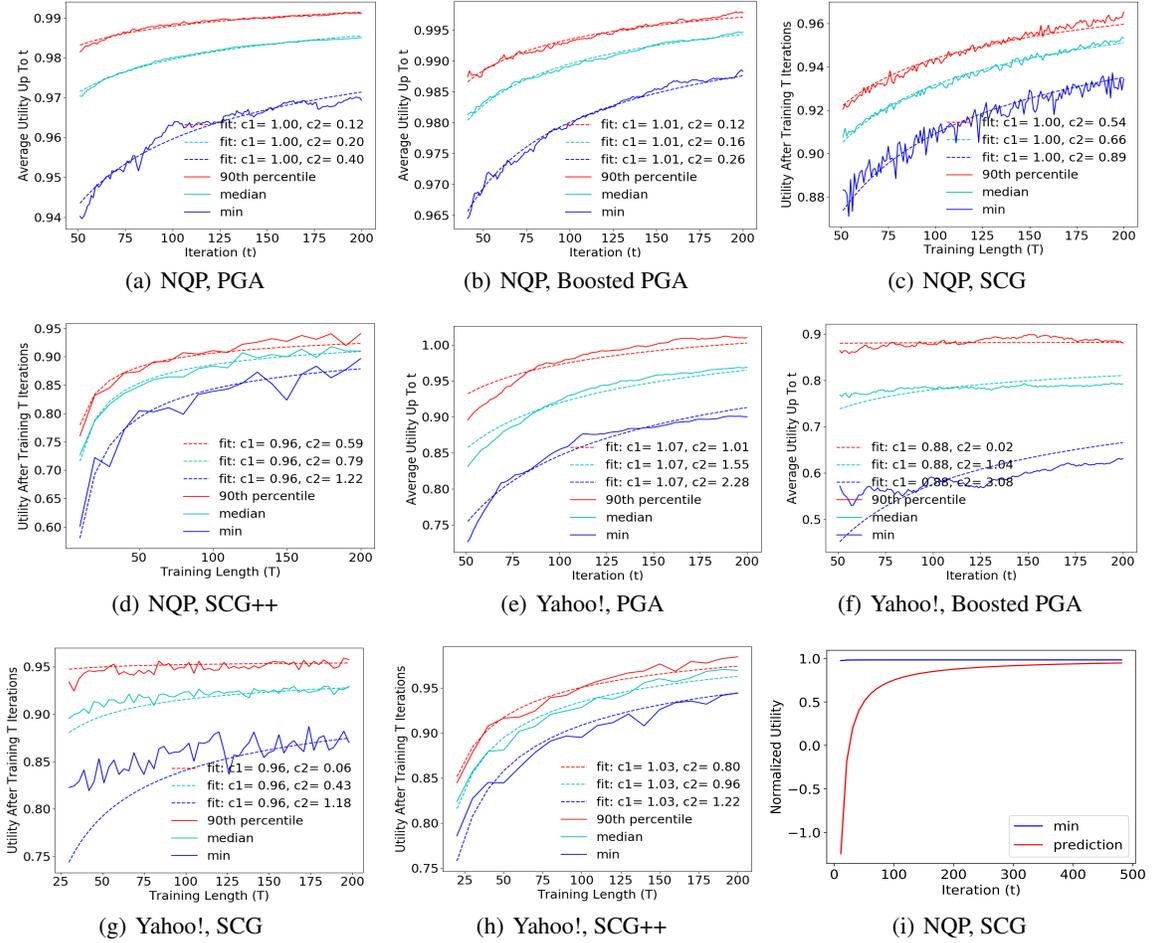


Figure 2: (a)-(h) Median, minimum, and 90% percentile of the normalized solutions obtained by PGA, Boosted PGA, SCG, SCG++ compared to our bounds provided in Theorems 1, 2, 4, 5. The results of PGA, Boosted PGA, SCG, SCG++ are averaged over 100 runs. (i) validating our bounds on a simple NQP example where the constants are known.

We further validate our bounds by adding a simple example where the constants are known and running SCG on the problem. We construct a small NQP where $\mathbf{H} \in \mathbb{R}^{5 \times 5}$ is sampled uniformly from $[-1, 0]$. Hence, the Lipschitz constant $L = \|\mathbf{H}\|_2 = \sqrt{\lambda_{\max}(\mathbf{H}^T \mathbf{H})}$. We use linear constraints as Sec. 5.1 and set $\mathbf{A} = [0.2, 0.2, 0.2, 0.2, 0.2]$. Thus, diameter $D = \|\mathbf{1}\|_2$. For a clipped Gaussian noise $\text{clip}(\mathcal{N}(0, \sigma), -2\sigma, 2\sigma)$ to queried gradients for SCG, $M = 2\sqrt{5}\sigma$. The optimal value for the problem is approximated by taking the maximum value across 100 SCG runs with 5000 iterations. Using above constants in Theorem 1, 2, 4, 5, Fig. 2(i) shows our predicted lower bound with 99% confidence converges quickly to the minimum of the collected utility trajectories.

6 CONCLUSION

We derived the first high probability analysis of the existing methods for stochastic Continuous Submodular Function (CSF) maximization, namely PGA, boosted PGA, SCG, and SCG++. When assumptions on the stochasticity of gradients

are strong enough, we showed that even in the worst case the solutions of the algorithms are lower bounded by a function converging to their expected guarantees. Specifically, with K stochastic gradient computations, we demonstrated that PGA converges with rate of $O(\frac{1}{\sqrt{K}})$ to the expected $OPT/2$ and the boosted version at the same rate to $(1 - 1/e)OPT$. For SCG and SCG++, we showed that both algorithms converge at rates of at least $O(\frac{\delta}{K^{1/3}})$ and $O(\frac{\delta}{K^{1/2}})$ to the expected $(1 - 1/e)OPT$, where δ depends on the confidence threshold. Besides, under the sub-Gaussian assumption on the gradient noise, we provided an improved lower bound of $O(\frac{1}{\sqrt{K}})$ for the convergence of SCG, that is faster than the existing convergence rate to the expected solution. Our results allows characterizing worst and best-case performance of CSF maximization in stochastic settings, and mitigating the risk of getting a bad solution.

Acknowledgment

BM was supported the National Science Foundation CAREER Award 2146492.

References

- [1] Yahoo! advertisement data: <https://webscope.sandbox.yahoo.com/catalog.php?datatype=a>.
- [2] An Bian, Kfir Levy, Andreas Krause, and Joachim M Buhmann. Continuous dr-submodular maximization: Structure and algorithms. *Advances in Neural Information Processing Systems*, 30, 2017.
- [3] Andrew An Bian, Baharan Mirzasoleiman, Joachim Buhmann, and Andreas Krause. Guaranteed non-convex optimization: Submodular maximization over continuous domains. In *Artificial Intelligence and Statistics*, pages 111–120. PMLR, 2017.
- [4] Chandra Chekuri, TS Jayram, and Jan Vondrák. On multiplicative weight updates for concave and submodular function maximization. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 201–210, 2015.
- [5] Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. *SIAM Journal on Computing*, 43(6):1831–1879, 2014.
- [6] Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet mathematics*, 3(1):79–127, 2006.
- [7] Reza Eghbali and Maryam Fazel. Designing smoothing functions for improved worst-case competitive ratio in online optimization. *Advances in Neural Information Processing Systems*, 29, 2016.
- [8] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. *Advances in Neural Information Processing Systems*, 31, 2018.
- [9] Daniel Golovin, Andreas Krause, and Matthew Streeter. Online submodular maximization under a matroid constraint with application to learning assignments. *arXiv preprint arXiv:1407.1082*, 2014.
- [10] Nicholas JA Harvey, Christopher Liaw, and Sikanter Randhawa. Simple and optimal high-probability bounds for strongly-convex stochastic gradient descent. *arXiv preprint arXiv:1909.00843*, 2019.
- [11] S. Hamed Hassani, Mahdi Soltanolkotabi, and Amin Karbasi. Gradient methods for submodular maximization. *CoRR*, abs/1708.03949, 2017.
- [12] Stefanie Jegelka and Jeff A. Bilmes. Submodularity beyond submodular energies: Coupling edges in graph cuts. *CVPR 2011*, pages 1897–1904, 2011.
- [13] Amin Karbasi, Hamed Hassani, Aryan Mokhtari, and Zebang Shen. Stochastic continuous greedy ++: When upper and lower bounds match. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [14] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD ’03*, 2003.
- [15] Andreas Krause, Carlos Guestrin, Anupam Gupta, and Jon Kleinberg. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, page 2–10, 2006.
- [16] Xiaoyu Li and Francesco Orabona. A high probability analysis of adaptive sgd with momentum. In *Workshop on Beyond First Order Methods in ML Systems at ICML’20*, 2020.
- [17] Hui-Ching Lin and Jeff A. Bilmes. A class of submodular functions for document summarization. In *ACL*, 2011.
- [18] László Lovász. Submodular functions and convexity. In *Mathematical programming: The state of the art, Bonn*, pages 235–257, 1982.
- [19] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22–es, 2007.
- [20] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Conditional gradient method for stochastic submodular maximization: Closing the gap. In *International Conference on Artificial Intelligence and Statistics*, pages 1886–1895. PMLR, 2018.
- [21] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *Journal of machine learning research*, 2020.
- [22] Kazuo Murota. Discrete convex analysis. *Mathematical Programming*, 83(1):313–371, 1998.
- [23] Mukund Narasimhan, Nebojsa Jojic, and Jeff A Bilmes. Q-clustering. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2005.
- [24] Tasuku Soma and Yuichi Yoshida. Non-monotone dr-submodular function maximization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

- [25] Matthew Staib and Stefanie Jegelka. Robust budget allocation via continuous submodular functions. In *International Conference on Machine Learning*, pages 3230–3240. PMLR, 2017.
- [26] Tao Sun, Huaming Ling, Zuoqiang Shi, Dongsheng Li, and Bao Wang. Training deep neural networks with adaptive momentum inspired by the quadratic optimization. *arXiv preprint arXiv:2110.09057*, 2021.
- [27] Laurence A Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.
- [28] Haixiang Zhang, Zeyu Zheng, and Javad Lavaei. Stochastic L^1 -convex function minimization. *Advances in Neural Information Processing Systems*, 34:13004–13018, 2021.
- [29] Lijun Zhang, Tianbao Yang, and Rong Jin. Empirical risk minimization for stochastic convex optimization: $\mathcal{O}(1/n)$ and $\mathcal{O}(1/n^2)$ -type of risk bounds. In *Conference on Learning Theory*, pages 1954–1979. PMLR, 2017.
- [30] Qixin Zhang, Zengde Deng, Zaiyi Chen, Haoyuan Hu, and Yu Yang. Stochastic continuous submodular maximization: Boosting via non-oblivious function. In *International Conference on Machine Learning*, pages 26116–26134. PMLR, 2022.