Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

ADAPTIVE MONTE CARLO SAMPLING GRADIENT METHOD FOR OPTIMIZATION

Hui Tan

School of Industrial Engineering
Purdue University
315 N Grant St.
West Lafayette, IN 47906, USA

ABSTRACT

We present a stochastic gradient descent algorithm with adaptive sampling for the unconstrained optimization problem where the function or the gradient is not directly accessible. We show that the algorithm exhibits global convergence and discuss the work complexity with different choices of predetermined function in the sampling rule.

1 INTRODUCTION

Consider a set of unconstrained optimization problem where the objective function has no closed form but is at least first order differentiable.

$$\min_{\mathbf{x}\in\mathbb{R}^d} \qquad f(\mathbf{x})$$

For example, the objective function is some expectation whose value at a given point cannot be computed directly. However the gradient at every point can be estimated (unbiased) by simulations or differencing. This is a general problem setting with a wide range of applications in various contexts including finance, machine learning.

2 PREVIOUS WORK

Different SGD (stochastic gradient descent) procedures have been proposed to solve the above optimization problem. Fixed step size algorithm produces a sequence with bounded average gradient norm and there is diminishing step size algorithm that yields convergence of limit inferior of gradient norm (Bottou, Curtis, and Nocedal 2016). In The Adaptive Sampling Gradient Method Optimizing Smooth Functions with an Inexact Oracle (Hashemi, Pasupathy, and Taaffe), an ASGM (adaptive sampling gradient method) achieves convergence of gradient norm and work complexity of $O(\varepsilon^{-2-\frac{1}{\mu(\alpha)-\delta}})$ (where $\mu(\alpha)$ is the error decaying rate of the approximators) when there are deterministic error bounds for approximators, for example, using Quasi Monte Carlo.

3 ALGORITHM AND ITS FEATURES

We present an adaptive sampling SGD method for the Monte Carlo case for solving this problem when f $\in C_L^{1,1}$. Assume that we have some information about the estimator error on how much it varies along every sample path. The Law of Iterated Logarithm applied to a single point, together with certain Lipschitz-like assumptions on the estimator error function path-wise ensure a uniform decay of the error across all $\mathbf{x} \in \mathbb{R}^d$. Meanwhile a sampling rule is imposed so that the error decay on the gradient estimator will satisfy an inequality, which will be essential in the proof of the convergence and work complexity results.

The algorithm is within the general frame of SGD methods. Given an iterate point, construct the gradient estimate and therefore the descent direction through the realizations of the random numbers. The error of the gradient norm decays as the amount of sampling effort increases. We first consider the case where the step size is a constant with some upper bound constraints. Iterations repeat to produce a sequence approaching the solution. The salient features of this procedure are line search and adaptive sampling. The descent direction comes from the gradient estimator. We show that the algorithm, under certain sampling rule, exhibits global convergence. The sampling rules should determine whether the sample size at one iterate point is large enough to construct a gradient estimation. Then for the purpose of reducing work complexity, we can optimize from the set of sampling rules that lead to convergence. There is a trade-of between accuracy and work complexity. So the question is what should be the right amount of effort to spend in constructing the gradient estimator at each iteration. The sampling rule ensures that the procedure spends just the right amount of effort at each step so that difference between the estimators and the true gradients will eventually stay within a predetermined gap function. We also discuss the choice of the predetermined gap function in the sampling rule and compare the corresponding work complexity.

Algorithm 1 Fixed step for f $\in C_L^{1,1}$

Setup:

Fix sample path ω (Common Random Numbers), starting point $\mathbf{x}_0$, function h, sampling size lower bound η_k, step size $\beta \leq L^{-1}$

Inputs:

Current iterate $\mathbf{x}_k$

Initialize:

n =0

repeat $\mathbf{G}(n, \mathbf{x}_k, \omega) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{g}_i(\mathbf{x}_k, \omega)$, n = n + 1

until $n \geq \eta_k$ with $n^{-\frac{1}{2}+\delta}\hat{\sigma}(n, \mathbf{x}, \omega)\Gamma_G(\mathbf{x}_k, \omega) \leq h(||G(n, \mathbf{x}_k, \omega)||)$

$n(\mathbf{X}_k, \omega) = n$

Outputs:

Next iterate $\mathbf{x}_{k+1}(\omega) = \mathbf{x}_k(\omega) - \beta G(n(\mathbf{x}_k, \omega), \mathbf{x}_k, \omega)$

4 SUMMARY OF RESULTS

Our results show that the above algorithm is globally convergent and produces an iterate sequence with true/ estimated gradient norm converging to zero $\lim_{k\to\infty} ||\nabla f(\mathbf{x}_k)|| = 0$. For the efficiency result, we derive work complexity in terms of the predetermined gap function h. We also consider a natural choice of h of the form $h(x) = \theta x^q (q \geq 1)$. When h is chosen to be linear, we have the optimal work complexity $w_k = O(||\nabla f(\mathbf{x}_k)||^{-2-\frac{1}{\frac{1}{2}-\delta}})$. The result is consistent with our intuition in the sense that more effort should be spent in where the gradient estimator gets closed to 0.

REFERENCES

Bottou, L., F. E. Curtis, and J. Nocedal. 2016, June. "Optimization methods for large-scale machine learning". *arXiv:1606.04838*.

Hashemi, F. S., R. Pasupathy, and M. R. Taaffe. "The Adaptive Sampling Gradient Method Optimizing Smooth Functions with an Inexact Oracle".