Stochastic Newton with Arbitrary Sampling

Igor Melnikov
Moscow Institute of Physics and Technology
Dolgoprudny, Russia
melnikov.ia@phystech.edu

Rustem Islamov
Institut Polytechnique de Paris
Palaiseau, France
rustem.islamov@ip-paris.fr

March 17, 2022

Abstract

We analyse stochastic Newton-type methods for solving Empirical Risk Minimization problem. We prove fast local convergence rates independent of the condition number. Unlike most other stochastic variants of second order methods, which require the evaluation of a large number of gradients and/or Hessians in each iteration to guarantee convergence, the method do not have this shortcoming. We investigate the performance of the method by applying existing sampling strategies.

1 Introduction

$$\min_{x\in\mathbb{R}^d}\left[f\left(x\right):=\frac{1}{n}\sum_{i=1}^{n}f_i\left(x\right)\right]. \tag{1}$$

Here n is the number of data points that is typically extremely large in real problems; d is the number of model parameters. Ususally, f_i denotes the value of a loss function on i-th data point (a_i, b_i). One of the examples of the problem that has the form of (1) is Logistic Regression problem where

$$f_i(x) = \log\left(1+\exp(-b_i a_i^\top x)\right), \tag{2}$$

where $a_i \in \mathbb{R}^d$ and $b_i \in \{-1, 1\}$.

As n is large the problem (1) is typically solved by First-order methods that uses only one data point per iteration. These methods are extensively studied [3] and there are a wide variety of variations of such techniques. In

particular, the Stochastic Gradient Descent(SGD) is often used, the distinguishing feature of which is cheap iterations independent of n. Nevertheless, SGD with constant-stepsize has a number of disadvantages, the main of which is that it converges only up to the neighbourhood of the solution, not the exact solution. This problem arises since stochastic gradient estimator has non-zero variance. Radius of this convergence area is proportional to the variance of the stochastic gradient. The so-called variance-reduced methods [2, 8] are used to solve this problem. They have the same iteration cost as SGD, but now the algorithm converges to the exact solution. However, all first-order methods known to us are characterized by the dependence of the required number of iterations on the condition number[1]. This makes impossible using SGD and its variants for ill-conditioned problems.

In classic optimization one of the solutions is to use second-order information about the objective. Classic Newton's method adapts to the curvature of the problem and thereby decrease the dependence on the condition number. The step of Newton's method has the following form

$$x^{k+1} = x^k - \left(\nabla^2 f\left(x^k\right)\right)^{-1} \nabla f\left(x^k\right) \tag{3}$$

In the case of ERM we need to compute n Hessians per iteration which is extremely costly in practice. Our desire is to use only a few Hessians in each iteration. One of the most popular directions is so called Subsampled Stochastic Newton's methods [1]. Despite first-order methods, these methods are poorly understood, and the theory usually requires a large batch sizes. To the best of our knowledge there are just a few works that provable work with arbitrary batch sizes [5, 6, 7].

In this work we focus on the Algorith 1 of [4]. They presentes the following algorithm:

Algorithm 1 Stochastic Newton (SN)

Initialize: Choose starting iterates $w_1^0, w_2 a^0, \ldots, w_n^0 \in \mathbb{R}^d$ and minibatch size $\tau \in \{1, 2, \ldots, n\}$

for $k = 1, \ldots$ **do**

$$x^{k+1} = \left(\sum_{i=1}^{n} \nabla^2 f_i\left(w_i^k\right)\right)^{-1} \sum_{i=1}^{n} \left(\nabla^2 f_i\left(x^k\right) w_i^k - \nabla f_i\left(w_i^k\right)\right)$$

Choose a subset $k \subseteq \{1, \ldots, n\}$ of size τ uniformly at random

$$w_i^{t+1} = \begin{cases} w_i^t & i \notin S^t \\ x^{t+1} & i \in S^t \end{cases}$$

end for

We investigate how the sampling strategies affect the performance of

[1]For a continuously Differentiable function f condition number is defined as $\lim\limits_{\varepsilon \to 0} \sup\limits_{\|\partial x\| \leq \varepsilon} \frac{\|\partial f(x)\|}{\|\partial x\|}$,

for L-smooth and μ-convex function the condition number is $\frac{L}{\mu}$

Algorithm (1). In practice, the uniform sampling is not the best choice, and we need to use another strategies how to choose a set S^t.

2 Problem Statement

Assume we have n training points (a_i, b_i) for $i \in \overline{1,n}$. We also assume n to be large. Let $f_i(x)$ be a loss function on i-th training point . We analyze second order methods solving Empirical Risk Minimization problem of the form.

One of the examples of the problem that has the form of (1) is Logistic Regression problem where

$$f_i(x) = \log\left(1 + \exp(-b_i a_i^\top x)\right), \tag{4}$$

where $a_i \in \mathbb{R}^d$ and $b_i \in \{-1, 1\}$.

$$\min_{x \in R}\left[f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)\right]. \tag{5}$$

3 Experiment Plan

The experiment applies Algorithm 1 to minimize the logistic regression risk function with l_2 regularization.

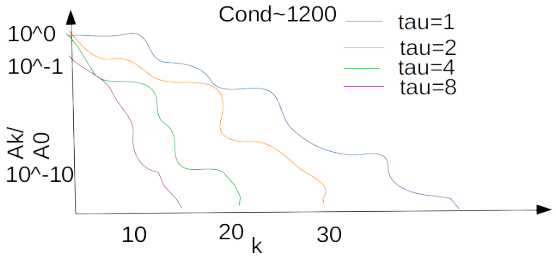
$$f_i(x) = \log\left(1 + \exp(-b_i a_i^\top x)\right) + \frac{\lambda}{2}\|x\|, \tag{6}$$

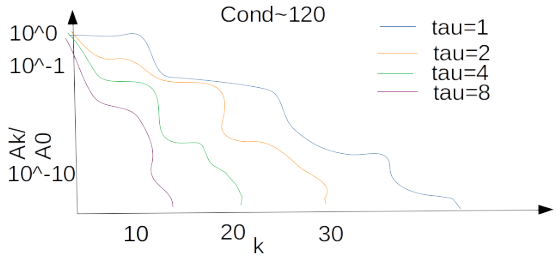
The main experiment is to establish the dependence of convergence accuracy on the size of the batch. The goal is to establish that conditional number does not significantly affect convergence. To do this, it is sufficient to establish that the curves are not very different.

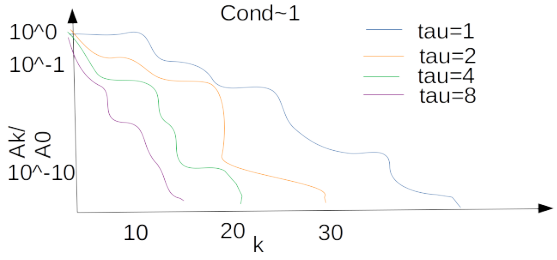
We generate the synthetic data by the make classification function of the scikit-learn library, where n = 500 and d = 5. Each row of the data matrix is normalized such that $\|a_i\| = 1$. We run experiments for $\lambda = \{10^{-1}, 10^{-3}, 10^{-4}\}$, which leads to conditional number differ more than in 10 times. Initial point x^0 is taken as the result of GD in 30 iteration.

We expect table in presented as pandas.DataFrame with columns=['k(iter num)', 'Ak/A0', 'tau', 'lambda'].

Expected plots.



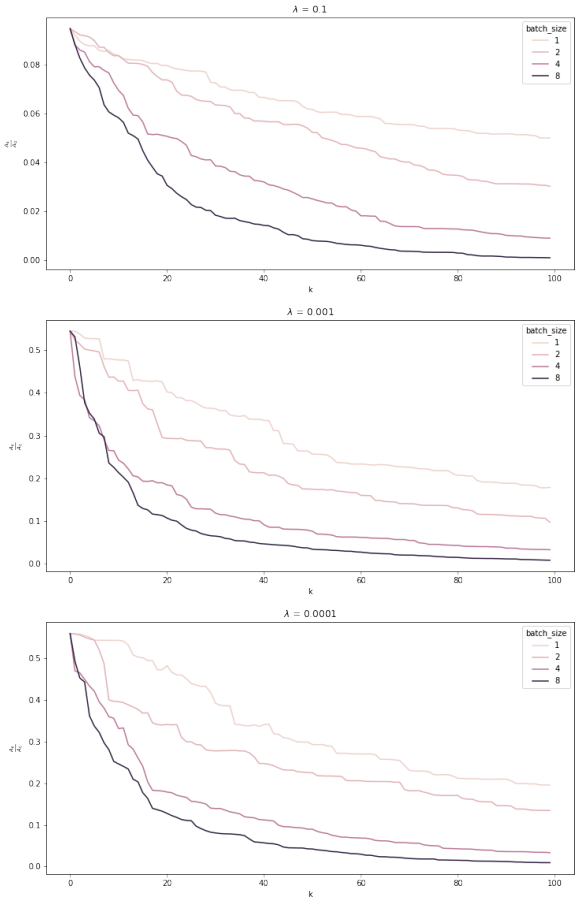




x-axis is number of iterations of the algorithm. y-axis is ratio $\frac{A_0}{A_k}$,where $A_0 = |x^* - x^0|, A_k = |x^* - x^k|$.

4 Preliminary report

Convergence has been established. For lambdas 0.001 and 0.0001 the difference in accuracy at 100th iteration is less than in 1.2 times for all butch sizes. However for lambda 0.1 accuracy is much higher - that is a fact to investigate in more detail.



5 Run Basic Code

We have implemented algorithm 1.

Competitive models for our algorithm were described in Introduction section with links on description. These models are Gradient Descent, Stochastic Gradient Descent, variance-reduced methods, classical Newton's method.

A description of the algorithm in the form of pseudocode is given in the

introduction.

References

[1] Raghu Bollapragada, Richard Byrd, and Jorge Nocedal. *Exact and Inexact Subsampled Newton Methods for Optimization*. 2016. arXiv: 1609.08502 [math.OC].

[2] Rie Johnson and Tong Zhang. "Accelerating stochastic gradient descent using predictive variance reduction, Advances in Neural Information Processing Systems". In: (2013).

[3] Ahmed Khaled et al. "Unified Analysis of Stochastic Gradient Methods for Composite Convex and Smooth Optimization". In: (2020). arXiv: 2006.11573 [cs.LG].

[4] Dmitry Kovalev, Konstantin Mishchenko, and Peter Richtárik. "Stochastic Newton and Cubic Newton Methods with Simple Local Linear-Quadratic Rates". In: *CoRR* abs/1912.01597 (2019). arXiv: 1912.01597. URL: http://arxiv.org/abs/1912.01597.

[5] Peter Richtárik and Martin Takáč. *On Optimal Probabilities in Stochastic Coordinate Descent Methods*. 2013. arXiv: 1310.3438 [stat.ML].

[6] Peter Richtárik and Martin Takáč. *Parallel Coordinate Descent Methods for Big Data Optimization*. 2013. arXiv: 1212.0873 [math.OC].

[7] Anton Rodomanov and Dmitry Kropotov. "A Superlinearly-Convergent Proximal Newton-type Method for the Optimization of Finite Sums". In: (2016).

[8] M. Schmidt, N. Le Roux, and F. Bach. "Minimizing finite sums with the stochastic average gradient". In: *Math. Program.* 162.1-2 (2017), pp. 83–112.