

---

# STOCHASTIC NEWTON WITH ARBITRARY SAMPLING

---

Denis Shveykin

Rustem Islamov

## Abstract

The problem of minimizing the average of a large number of sufficiently smooth and strongly convex functions is ubiquitous in machine learning. Stochastic first-order methods for this problem of Stochastic Gradient Descent type are well studied. In turn, second-order methods, such as Newton, have certain advances since they can adapt to the curvature of the problem. They are also known for their fast convergence rates. But stochastic variants of Newton-type methods are not studied as good as SGD-type ones and have limitations on the batch size. [9] proposed a method which requires no limitations on batch sizes. Our goal is to explore this method with different sampling strategies that lead to practical improvements.

**Keywords** Stochastic Newton, sampling strategy

## 1 Introduction

The problem is to minimize the empirical risk which has finite-sum structure [9]:

$$\min_{x \in \mathbb{R}^d} \left[ f(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n f_i(x) \right] \quad (1)$$

where each  $f_i$  is assumed to have Lipschitz Hessian.

Typically,  $n$  is very large for modern real-world problems. Thus, the stochastic approach is used because it is computationally difficult to evaluate the gradient of all  $f_i$  at each step. The Stochastic Gradient Descent (SGD) method [14] calculates the gradients of some randomly chosen  $f_i$  which leads to cheaper computation per-iteration cost compared to the vanilla Gradient Descent (GD). The analysis of SGD and its modifications is rich and well explored. Firstly, the theory of SGD-type methods does not restrict the batch size. Therefore, these algorithms can

be applied even with small batches. It is known that simple SGD converges only to a neighbourhood of the solution only [12, 4] whose size is proportional to the variance of the stochastic gradient. However, there are techniques (aka variance reduction) to solve this issue. These techniques [15, 6, 2, 5] modify the update rule of vanilla SGD which allows to mitigate the aforementioned effect without changing per-iteration cost. However, the main disadvantage of all gradient-type methods is that a computation complexity depends on the curvature of the problem, which is called the condition number and is defined as the ratio of Lipschitzness and strong convexity parameters.

This is a place where second-order methods such as the Newton method [11, 7, 3] come to play. Taking into account second-order derivatives it is possible to adjust the algorithm's step sizes to the curvature of the problem [11]. Unfortunately, much less work has been done in the direction of stochastic Newton-type methods. Many algorithms [8, 1, 20, 17, 16, 19] require large batch sizes. In particular, the required batch size is commonly quadratically proportional to the inverse of the desired accuracy. That means that one needs to evaluate a large number of  $f_i$  Hessians which sometimes can be much larger than  $n$ . [9] proposes a simple Stochastic Newton algorithm, which can work with batches of any size. Their algorithms achieve local linear and super-linear convergence.

In practice, various sampling strategies are used for SGD-type algorithms to improve further the performance. One of the most famous sampling mechanisms is so-called Importance Sampling [18, 10]. The idea is to compute gradients of the functions that have more impact on the problem. [13] studies many other sampling mechanisms. We aim to analyse such strategies, but for the Algorithms 1 of [9] to improve the theoretical and practical applications of algorithms. We investigate various sampling strategies supporting them with rigorously constructed experiments.

## 2 Problem statement

We consider classical empirical risk minimization (ERM) which typically arises in many machine learning problems to train a model. The objective function  $f$  (1) is an average of a big number of functions  $f_i$ , where  $f_i$  represents a loss on the  $i$ -th train data point. For example, this notation can be attributed to linear regression. In this case the goal is to find optimal model parameters  $x$  that minimize the mean squared error (MSE) on the train data.

### 2.1 Assumptions

We make standard assumptions on functions  $f_i$ ; the same that were originally taken into account in [9].

**Assumption 1** (Strong convexity). A differentiable function  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$  is  $\mu$ -strongly convex, where  $\mu > 0$ , if  $\forall x, y \in \mathbb{R}^d$

$$\phi(x) \geq \phi(y) + \langle \nabla \phi, x - y \rangle + \frac{\mu}{2} \|x - y\|^2, \quad (2)$$

where the norm  $\|\cdot\|$  is Euclidean. For twice differentiable functions this assumption is equivalent to the Hessian having each eigenvalue  $\geq \mu$ .

**Assumption 2** (Lipschitz Hessian). A function  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$  has  $H$ -Lipschitz Hessian if  $\forall x, y \in \mathbb{R}^d$

$$\|\nabla^2 \phi(x) - \nabla^2 \phi(y)\| \leq H \|x - y\| \quad (3)$$

## 2.2 Sampling

**Definition 1** (Sampling). A random set-valued mapping  $\hat{S} : [n] \rightarrow 2^{[n]}$  is called sampling.

That means that each  $S_k \subseteq [n]$  is a realization of  $\hat{S}$ . In this case we can call any particular probability distribution on  $2^{[n]}$  a sampling strategy.

## 2.3 Algorithm

We apply different sampling strategies on top of the Algorithm 1 from [9]:

---

### Algorithm 1 Stochastic Newton (SN)

---

**Initialize:** Choose starting iterates  $w_1^0, w_2^0, \dots, w_n^0 \in \mathbb{R}^d$

**for**  $k = 0, 1, 2, \dots$  **do**

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

Choose a subset  $S^k \subseteq \{1, 2, \dots, n\}$  with one of the sampling strategies

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

**end for**

---

## 2.4 Goal of the project

We aim to explore different sampling strategies for the Algorithm 1 proving convergence guarantees and showing practical improvements over the baseline strategy.

## 3 Theory

### 3.1 Proofs for the Algorithm

The convergence of the Algorithm 1 was clearly described and proven in [9]. Firstly, we look at the statements of three lemmas from the aforementioned article.

**Lemma 1.** *Let  $f_i$  be  $\mu$ -strongly convex and have  $H$ -Lipschitz Hessian for all  $i = 1, \dots, n$  and consider the following Lyapunov function*

$$\mathcal{W}^k \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n \|w_i^k - x^*\|^2, \quad (4)$$

where  $x^*$  is the minimizer of the objective function  $f$ . Then the iterates of the Algorithm 1 satisfy

$$\|x^{k+1} - x^*\| \leq \frac{H}{2\mu} \mathcal{W}^k. \quad (5)$$

**Lemma 2.** *Assume that every  $f_i$  is  $\mu$ -strongly convex and has  $H$ -Lipschitz Hessian. If  $\|w_i^0 - x^*\| \leq \frac{\mu}{H}$  for  $i = 1, \dots, n$ , then for all  $k$*

$$\mathcal{W}^k \leq \frac{\mu^2}{H^2}. \quad (6)$$

The last lemma is the one that incorporates the nature of the sampling strategy. The authors of the Algorithm 1 used so called  $\tau$ -nice sampling, which chooses a subset  $S^k \subseteq \{1, 2, \dots, n\}$  precisely of size  $\tau$  uniformly at random. Therefore it is possible to determine the expectation  $\mathbb{E}_k[\mathcal{W}^{k+1}] \stackrel{\text{def}}{=} \mathbb{E}[\mathcal{W}^{k+1} | x^k, w_1^k, \dots, w_n^k]$ .

**Lemma 3.** *The random iterates of the Algorithm 1 with  $\tau$ -nice sampling satisfy the identity*

$$\mathbb{E}_k[\mathcal{W}^{k+1}] = \frac{\tau}{n} \|x^{k+1} - x^*\|^2 + \left(1 - \frac{\tau}{n}\right) \mathcal{W}^k \quad (7)$$

*Доказательство.* For each  $i$ ,  $w_i^{k+1}$  is equal to  $x^{k+1}$  with probability  $\frac{\tau}{n}$  and to  $w_i^k$  with probability  $1 - \frac{\tau}{n}$ , therefore considering the linearity of expectation, the result is obtained.  $\square$

Then follows the theorem, which estimates the convergence rate of the Algorithm 1 with  $\tau$ -nice sampling.

**Theorem 1.** *Assume that every  $f_i$  is  $\mu$ -strongly convex and has  $H$ -Lipschitz Hessians. Then for the random iterates of the Algorithm 1 we have the recursion*

$$\mathbb{E}_k[\mathcal{W}^{k+1}] \leq \left(1 - \frac{\tau}{n} + \frac{\tau}{n} \left(\frac{H}{2\mu}\right)^2\right) \mathcal{W}^k. \quad (8)$$

Moreover, if  $\forall i \ \|w_i^0 - x^*\| \leq \frac{\mu}{H}$  then

$$\mathbb{E}_k[\mathcal{W}^{k+1}] \leq \left(1 - \frac{3\tau}{4n}\right) \mathcal{W}^k. \quad (9)$$

## 3.2 Sampling Strategies

### 3.2.1 Doubly Uniform Strategies

To begin with, we discuss the sampling strategies that were presented in [13] in relation to the Parallel Coordinate Descend method. These strategies are *Doubly Uniform*.

**Definition 2** (Doubly Uniform (DU) sampling). This is a sampling strategy that generates all sets of equal cardinality with equal probability. That is, if  $|S_1| = |S_2|$  then  $\mathbb{P}(S_1) = \mathbb{P}(S_2)$ .

The name comes from the fact this property is stronger than *uniformity*, which means that all  $i \in [n]$  have equal probability to be included in the chosen set  $\hat{S}$ . That can be seen if we denote the probability  $q_j = \mathbb{P}(|\hat{S}| = j)$  for  $j = 0, 1, \dots, n$ . Then for every  $S \subseteq [n]$  we have  $\mathbb{P}(S) = q_j / \binom{n}{j}$ . Hence

$$p_i = \sum_{j=1}^n \sum_{|S|=j, i \in S} \mathbb{P}(S) = \sum_{j=1}^n \sum_{|S|=j, i \in S} \frac{q_j}{\binom{n}{j}} = \sum_{j=1}^n \frac{\binom{n-1}{j-1} q_j}{\binom{n}{j}} = \frac{1}{n} \sum_{j=1}^n j q_j = \frac{\mathbb{E}[|\hat{S}|]}{n} \quad (10)$$

It is clear that a Doubly Uniform sampling strategy is determined by the vector  $q$  of probability distribution on the possible sizes  $j = 0, 1, \dots, n$  of the set  $\hat{S}$ . There are some interesting DU strategies.

1.  **$\tau$ -Nice sampling.** Fix  $1 \leq \tau \leq n$ . A sampling is called  $\tau$ -nice if it is DU with  $q_\tau = 1$ . This is the strategy used by the authors of the Algorithm 1. This strategy can be interpreted in terms of multi-process computation. So, there are  $\tau$  processors available. On the moment of sampling we choose a batch of size  $\tau$  and assign each train object  $i \in \hat{S}$  to a dedicated processor. Therefore, each processor will update the values of the objective function, gradients and Hessians with respect to new  $w_i^k$ .
2.  **$\tau$ -Independent sampling.** Fix  $1 \leq \tau \leq n$ . A  $\tau$ -independent sampling is a DU sampling with

$$q_k = \begin{cases} \binom{n}{k} c_k, & k = 1, 2, \dots, \tau \\ 0, & k = \tau + 1, \dots, n \end{cases} \quad (11)$$

where  $c_1 = (1/n)^\tau$  and  $c_k = (k/n)^\tau - \sum_{i=1}^{k-1} \binom{k}{i} c_i$  for  $k \geq 2$ .

The interpretation is that each of  $\tau$  processors chooses one of the train objects  $i \in [n]$ . If several processors have chosen the same train object, then only one of them gets access to it. This strategy is very easy to implement in terms of parallel computation. Besides, when batches are small, we have  $\tau \ll n$  and then  $\tau$ -independent sampling is a good approximation of  $\tau$ -nice variant.

3.  **$(\tau, p_b)$ -Binomial sampling.** Fix  $1 \leq \tau \leq n$  and  $0 \leq p_b \leq 1$ . The strategy is a DU sampling with

$$q_k = \binom{\tau}{k} p_b^k (1 - p_b)^{\tau-k}, \quad k \leq \tau \quad (12)$$

That is a model of the situation when each of  $\tau$  processors is available at the moment of sampling with probability  $p_b$ , hence  $q_k$  is the probability that  $k$  processors are available.

### 3.2.2 Independent strategies

Apart from Doubly Uniform strategies we consider strategies, which make a decision about each train object  $i \in [n]$  independently with its own probability  $p_i$ . Notably, in case  $p_1 = p_2 = \dots = p_n$  the strategy remains DU. But in general case some changes appear in the convergence rate estimation. There is the a strategy, which assigns probabilities proportionally to the Lipschitz parameters of the gradients  $\nabla f_i(x)$ .

**Definition 3** (Lipschitz gradients). A function  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$  has  $L$ -Lipschitz gradient and is called  $L$ -smooth, if  $\forall x, y \in \mathbb{R}^d$

$$\|\nabla \phi(x) - \nabla \phi(y)\| \leq L \|x - y\| \quad (13)$$

**Importance sampling.** The strategy is to assign

$$p_i = \frac{L_i}{\sum_{i=1}^n L_i} \quad (14)$$

## 3.3 Convergence rate

### 3.3.1 Estimations for Doubly Uniform sampling

To determine the convergence rate of the Algorithm 1 with Doubly Uniform sampling (2) we turn to the Lemma 3, since it is the first place in the Proofs for the Algorithm

1 which depends on the sampling strategy. It is generalized to an arbitrary DU sampling.

**Lemma 4.** *The random iterates of the Algorithm 1 with Doubly Uniform sampling satisfy*

$$\mathbb{E}_k[\mathcal{W}^{k+1}] = p\mathbb{E}_k[\|x^{k+1} - x^*\|^2] + (1 - p)\mathcal{W}^k, \quad (15)$$

where  $p = \frac{\mathbb{E}[|\hat{S}|]}{n}$ .

*Доказательство.* The sampling strategy being Doubly Uniform implies that all train objects have equal probability  $p = \frac{\mathbb{E}[|\hat{S}|]}{n}$  to be included in the chosen set  $\hat{S}$ . Therefore, by the linearity of expectation, we obtain the result.  $\square$

Thereby the final estimation has the same form as the one presented in [9]:

$$\mathbb{E}_k[\mathcal{W}^{k+1}] \leq \left(1 - p + p \left(\frac{H}{2\mu}\right)^2 \mathcal{W}^k\right) \mathcal{W}^k \quad (16)$$

in general, and in case  $\forall i \ \|w_i^0 - x^*\| \leq \frac{\mu}{H}$ :

$$\mathbb{E}_k[\mathcal{W}^{k+1}] \leq \left(1 - \frac{3}{4}p\right) \mathcal{W}^k. \quad (17)$$

That means, if we fix the expected batch size  $\mathbb{E}[|\hat{S}|] = \tau$  then the convergence rate is the same as in  $\tau$ -nice sampling.

### 3.3.2 Independent sampling

Now we consider general the case of choosing each train object independently with probability  $p_i$ .

**Proposition 1.** *In a particular iteration of Algorithm 1 given a fixed expectation of the batch size  $\mathbb{E}_k[|\hat{S}^k|] = \tau$  the minimum value in the estimation of  $\mathbb{E}_k[\mathcal{W}^{k+1}]$  is obtained by setting  $p_i = 1$  for several  $i \in [n]$ , possibly  $p_i \in (0, 1)$  for a single object  $i$ , and  $p_i = 0$  for others.*

*Доказательство.* We can generalize the estimation from Lemma 4 to our case.

$$n \cdot \mathbb{E}_k[\mathcal{W}^{k+1}] = \sum_{i=1}^n p_i \|x^{k+1} - x^*\|^2 + \sum_{i=1}^n (1 - p_i) \|w_i^k - x^*\|^2. \quad (18)$$

Considering  $\tau = \mathbb{E}_k[|\hat{S}^k|] = \sum_{i=1}^n p_i$ , we have

$$n \cdot \mathbb{E}_k[\mathcal{W}^{k+1}] = (\tau \|x^{k+1} - x^*\|^2 + n \cdot \mathcal{W}^k) - \sum_{i=1}^n p_i \|w_i^k - x^*\|^2. \quad (19)$$

So the goal is to maximize  $\sum_{i=1}^n p_i \|w_i^k - x^*\|^2$  satisfying the following:

$$\begin{cases} \sum_{i=1}^n p_i = \tau \\ 0 \leq p_i \leq 1 \quad \forall i \end{cases} \quad (20)$$

The solution of this Linear Programming problem is the vertex of the polyhedron defined by the constraints 20. The optimal vertex has the maximum projection of its coordinate vector  $(p_1, \dots, p_n)^\top$  on the vector  $(\|w_1 - x^*\|^2, \dots, \|w_n - x^*\|^2)$ . That is the vertex, which coefficients corresponding to the biggest  $\|w_i - x^*\|^2$  are set to 1, corresponding to the smallest  $\|w_i - x^*\|^2$  are set to 0, and one coordinate is intermediate if  $\tau$  is not an integer.  $\square$

### 3.3.3 Estimations for Importance sampling

...

## 4 Experiments

The computational experiments can be seen in the code folder.

...

## References

- [1] Raghu Bollapragada, Richard H Byrd, and Jorge Nocedal. Exact and inexact subsampled Newton methods for optimization. *IMA Journal of Numerical Analysis*, 39(2):545–578, 04 2018.
- [2] Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent, 2019.
- [3] Robert Gower, Dmitry Kovalev, Felix Lieder, and Peter Richtarik. Rsn: Randomized subspace newton. 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.
- [4] Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtarik. Sgd: General analysis and improved rates. 2019.
- [5] Filip Hanzely and Peter Richtárik. One method to rule them all: Variance reduction for data, parameters and many new methods, 2019.
- [6] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C.J. Burges, L. Bottou, M. Welling,



- Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
- [7] Sai Praneeth Karimireddy, Sebastian U. Stich, and Martin Jaggi. Global linear convergence of newton’s method without strong-convexity or lipschitz gradients, 2018.
  - [8] Jonas Moritz Kohler and Aurelien Lucchi. Sub-sampled cubic regularization for non-convex optimization. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1895–1904. PMLR, 06–11 Aug 2017.
  - [9] Dmitry Kovalev, Konstantin Mishchenko, and Peter Richtárik. Stochastic newton and cubic newton methods with simple local linear-quadratic rates. *arXiv preprint arXiv:1912.01597*, 2019.
  - [10] Huikang Liu, Xiaolu Wang, Jiajin Li, and Anthony Man-Cho So. Low-cost lipschitz-independent adaptive importance sampling of stochastic gradients. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2150–2157, 2021.
  - [11] Yurii Nesterov. *Introductory lectures on convex optimization: a basic course/ by Yurii Nesterov*. Applied optimization, 87. Kluwer Academic Publishers, Boston, 2004.
  - [12] Lam Nguyen, PHUONG HA NGUYEN, Marten van Dijk, Peter Richtarik, Katya Scheinberg, and Martin Takac. SGD and hogwild! Convergence without the bounded gradients assumption. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3750–3758. PMLR, 10–15 Jul 2018.
  - [13] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156:433–484, 2016.
  - [14] Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400 – 407, 1951.
  - [15] Nicolas Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence \_rate for finite training sets. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
  - [16] Nilesh Tripuraneni, Mitchell Stern, Chi Jin, Jeffrey Regier, and Michael I Jordan. Stochastic cubic regularization for fast nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

- [17] Junyu Zhang, Lin Xiao, and Shuzhong Zhang. Adaptive stochastic variance reduction for subsampled newton method with cubic regularization. *INFORMS Journal on Optimization*, 4(1):45–64, 2022.
- [18] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling, 2014.
- [19] Dongruo Zhou and Quanquan Gu. Stochastic recursive variance-reduced cubic regularization methods. In *International Conference on Artificial Intelligence and Statistics*, pages 3980–3990. PMLR, 2020.
- [20] Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic variance-reduced cubic regularized newton methods. In *International Conference on Machine Learning*, pages 5990–5999. PMLR, 2018.