Taylor & Francis
Taylor & Francis Group

Journal of the American Statistical Association

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uasa20

Online Covariance Matrix Estimation in Stochastic Gradient Descent

Wanrong Zhu, Xi Chen & Wei Biao Wu

To cite this article: Wanrong Zhu, Xi Chen & Wei Biao Wu (2021): Online Covariance Matrix Estimation in Stochastic Gradient Descent, Journal of the American Statistical Association, DOI: 10.1080/01621459.2021.1933498

To link to this article: https://doi.org/10.1080/01621459.2021.1933498

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION
2021, VOL. 00, NO. 0, 1–12: Theory and Methods
https://doi.org/10.1080/01621459.2021.1933498

Taylor & Francis
Taylor & Francis Group

Online Covariance Matrix Estimation in Stochastic Gradient Descent

Wanrong Zhu[a], Xi Chen[b], and Wei Biao Wu[c]

[a]Department of Statistics, University of Chicago, Chicago,IL ; [b]Leonard N. Stern School of Business, New York University, New York, NY; [c]Department of Statistics, University of Chicago, Chicago, IL

ABSTRACT

The stochastic gradient descent (SGD) algorithm is widely used for parameter estimation, especially for huge datasets and online learning. While this recursive algorithm is popular for computation and memory efficiency, quantifying variability and randomness of the solutions has been rarely studied. This article aims at conducting statistical inference of SGD-based estimates in an online setting. In particular, we propose a fully online estimator for the covariance matrix of averaged SGD (ASGD) iterates only using the iterates from SGD. We formally establish our online estimator's consistency and show that the convergence rate is comparable to offline counterparts. Based on the classic asymptotic normality results of ASGD, we construct asymptotically valid confidence intervals for model parameters. Upon receiving new observations, we can quickly update the covariance matrix estimate and the confidence intervals. This approach fits in an online setting and takes full advantage of SGD: efficiency in computation and memory.

ARTICLE HISTORY
Received June 2020
Accepted May 2021

KEYWORDS
Asymptotic normality; Averaging stochastic gradient descent; Recursive; Statistical inference;

1. Introduction

Model parameter estimation through optimization of an objective function is a fundamental problem in statistics and machine learning. Here, we consider the classic setting where the true model parameter $x^* \in \mathbb{R}^d$ can be characterized as the minimizer of a convex objective function $F : \mathbb{R}^d \to \mathbb{R}$, that is,

$$x^* = \arg\min_{x\in\mathbb{R}^d} F(x). \tag{1}$$

The objective function $F(x)$ is defined as $F(x) = \mathbb{E}_{\xi\sim\Pi} f(x,\xi)$, where $f(x,\xi)$ is a noisy measurement of $F(x)$ and ξ is a random variable following the distribution Π.

In the recent years, huge datasets and streaming data arise frequently. Classic deterministic optimization methods that require storing all the data are not appealing due to expensive memory cost and computational inefficiency. To resolve these issues, one can apply the Robbins-Monro algorithm (Robbins and Monro 1951; Kiefer and Wolfowitz 1952), also known as stochastic gradient descent (SGD), especially for online learning (Bottou 1998; Mairal et al. 2010; Hoffman, Bach, and Blei 2010). Setting x_0 as the initial point, the ith iteration of the SGD algorithm takes the following form:

$$x_i = x_{i-1} - \eta_i \nabla f(x_{i-1}, \xi_i),\ i \geq 1, \tag{2}$$

where $\{\xi_i\}_{i\geq 1}$ is a sequence of iid sample from the distribution Π, ∇f is the gradient of $f(x,\xi)$ with respect to the first argument x, and η_i is the step size at the ith step. This recursive adaptive algorithm performs one update at a time and does not need to remember outcomes in previous iterations. Therefore, it is computationally efficient, memory friendly, and able to process data on the fly.

Despite these advantages, SGD performs frequent updates with high variability, and the outcomes can fluctuate heavily. The crucial problem is to understand the variability and randomness of the solutions. In this article, we address the uncertainty quantification problem in the *online setting* where data can arrive sequentially. In particular, we propose a fully online approach to estimate the covariance matrix of SGD-based estimates only using the iterates from SGD. The efficient algorithm we propose is recursive. It performs an immediate update of the covariance estimate as new data arrives, which follows the spirit of SGD. We can then conduct statistical inference with the estimated covariance matrix and construct confidence intervals for model parameters in a fully online fashion.

Before discussing our method, we provide a brief review of the literature on SGD. The asymptotic convergence of SGD iterates has been studied extensively in the early years (Blum 1954; Dvoretzky 1956; Robbins and Siegmund 1971; Ljung 1977; Sacks 1958; Fabian 1968; Lai 2003). To further investigate the asymptotic distribution of SGD, Polyak and Juditsky (1992) and Ruppert (1988) introduced the averaged SGD (ASGD), a simple modification where iterates are averaged, and established the asymptotic normality of the obtained estimate. Moreover, it is known that ASGD estimates achieve the optimal central limit theorem rate $\mathcal{O}_P(1/\sqrt{n})$ by running SGD for n iterations under certain regularity conditions. For linear stochastic approximation, Mou et al. (2020) modified the Polyak-Ruppert covariance with an additional correction term concerning the constant step size. Differently from the SGD algorithm, Toulis and Airoldi

CONTACT Xi Chen xchen3@stern.nyu.edu Leonard N. Stern School of Business, New York University, New York, NY 10012-1126.

(2017) introduced implicit SGD procedures and analyzed the asymptotic distribution of the averaged implicit SGD iterates. Convergence in nonasymptotic fashion has also been studied recently for SGD and its variants with different objective functions (Rakhlin, Shamir, and Sridharan 2012; Moulines and Bach 2011; Hazan and Kale 2014; Bach and Moulines 2013; Duchi, Hazan, and Singer 2011; Kingma and Ba 2015; Shamir and Zhang 2013). Our method and analysis rely on the averaged SGD and its asymptotic normality in later discussions.

In addition to convergence and error bounds of SGD-based estimators, statistical inference problems based on SGD have recently started to gain more attention. Instead of focusing on point estimators, one is interested in assessing the uncertainty of the estimates through their confidence intervals/regions. Chen et al. (2020) introduced the inference problem and proposed a batch-means method to construct asymptotically valid confidence intervals based on asymptotic normality of ASGD. Fang, Xu, and Yang (2018) and Fang (2019) proposed bootstrap procedures for constructing confidence intervals through the perturbed-SGD. Meanwhile, variants of the SGD algorithm and corresponding inference in nonasymptotic fashion are studied in Su and Zhu (2018) and Liang and Su (2019). For online l_1 penalized problems, Chao and Cheng (2019) proposed a class of generalized regularized dual averaging and made uncertainty quantification possible for online sparse algorithms.

1.1. Problem Formulation

Our work in this article is applicable to vanilla SGD, which is most widely used in practice. We use the ASGD iterate

$$\bar{x}_n = n^{-1}\sum_{i=1}^{n} x_i$$

as the estimate for the model parameter at the nth step. We set step size $\eta_i = \eta i^{-\alpha}(i \geq 1)$ with $\eta > 0$ and $\alpha \in (0.5, 1)$ as suggested by Polyak and Juditsky (1992). Define

$$A = \nabla^2 F(x^*),\ S = \mathbb{E}\left([\nabla f(x^*, \xi)][\nabla f(x^*, \xi)]^T\right). \tag{3}$$

From Polyak and Juditsky (1992), under suitable conditions, $\bar{x}_n$ has the asymptotic normality:

$$\sqrt{n}(\bar{x}_n - x^*) \Rightarrow N(0, \Sigma), \tag{4}$$

where $\Sigma = A^{-1}SA^{-1}$, which is known as the "sandwich" form of the covariance matrix. To leverage the asymptotic normality result for inference, it is critical to estimate the limiting covariance matrix Σ. Intuitively, one can estimate S with a simple sample average $\widehat{S}_n = n^{-1}\sum_{i=1}^{n}[\nabla f(x_{i-1}, \xi_i)][\nabla f(x_{i-1}, \xi_i)]^T$, and similarly estimate A with $\widehat{A}_n = n^{-1}\sum_{i=1}^{n}\nabla^2 f(x_{i-1}, \xi_i)$. Then the limiting covariance matrix Σ can be estimated by the consistent plug-in estimator $\widehat{A}_n^{-1}\widehat{S}_n\widehat{A}_n^{-1}$ (see Chen et al. 2020). However, computation of the Hessian matrix of the loss function is not always available, for example, certain computations are not available in many existing codebases that only adopt SGD for optimization and in cases such as quantile regression, the Hessian matrix does not even exist. Also, the plug-in estimator may be computationally costly when d is large since it involves matrix inversion with $O(d^3)$ time complexity in general.

Our goal is to obtain an online estimate of the covariance matrix of $\sqrt{n}\bar{x}_n$, only through the SGD iterates $\{x_1, x_2, \ldots, x_n\}$. Our approach is attractive in situations where the computation for A^{-1} and S are difficult, which is quite typical in practice. Also, the approach is efficient in both computation and memory due to its recursive property, that is, the estimate at the nth step $\widehat{\Sigma}_n$ can be updated from $\widehat{\Sigma}_{n-1}$ within $O(d^2)$ computation. With the estimate, we can perform uncertainty quantification and statistical inference with desirable computation and memory efficiency. The approach is useful for online learning, where the data is constantly arriving over time, such as streaming data.

For the time-homogeneous Markov chain, $\{x_i\}_{i\in\mathbb{Z}}$ is a stationary process. Under certain short-range dependence conditions, we have

$$\sqrt{n}\,(\bar{x}_n - \mathbb{E}x_i) \Rightarrow N(0, \sigma^2),$$

where

$$\sigma^2 = \lim_{n\to\infty}\mathrm{var}(\sqrt{n}\bar{x}_n) = \sum_{i=-\infty}^{\infty}\mathrm{cov}(x_0, x_i)$$

is the long-run variance, and it plays a fundamental role in the statistical inference of stationary processes. To estimate the long-run variance, one can apply the batch-means method (Glynn and Whitt 1991; Kitamura et al. 1997; Politis, Romano, and Wolf 1999; Lahiri 2003; Flegal and Jones 2010). Given $x_1, \ldots, x_n$, let $1 \leq l_n \leq n$ be the batch size. Based on batch-means $\sum_{k=i}^{i+l_n} x_k/l_n - \bar{x}_n$ for $1 \leq i \leq n - l_n + 1$, one can estimate σ^2 by

$$\sigma_n^2 = \frac{l_n}{n - l_n + 1}\sum_{i=1}^{n-l_n+1}\left(\sum_{k=i}^{i+l_n-1} x_k/l_n - \bar{x}_n\right)^2.$$

As an alternative, one can use the nonoverlapping batch-means $\sum_{k=i}^{i+l_n} x_k/l_n - \bar{x}_n$ for $i = 1, 1 + l_n, 1 + 2l_n, \ldots$, to construct a similar estimate. Properties of overlapping and nonoverlapping batch-means estimators are discussed in Politis, Romano, and Wolf (1999) and Lahiri (2003). In our problem, estimation of Σ in (4) becomes more complicated since SGD iterates form a nonstationary Markov Chain.

To apply to SGD, Chen et al. (2020) modified the classic nonoverlapping batch-means by allowing increasing batch sizes and showed that the modified batch-means estimator is consistent. However, their approach is not in line along with the spirit of SGD, the fully online fashion. Their construction of covariance estimator $\widehat{\Sigma}_n$ requires the information on the total number of iterations n a priori. There is no simple algebraic relation between $\widehat{\Sigma}_n$ and $\widehat{\Sigma}_{n+1}$. In other words, when a new data point x_{n+1} arrives later, their algorithm needs to recompute their estimate from the beginning and cannot perform efficient sequential updating. So the approach is computationally expensive for online learning, where the dynamic training data are arriving over time, and the goal is to make sequential predictions; see Remark 2.1 for a detailed discussion of Chen et al. (2020).

To address the above problems, we develop in this article a fully *online approach* for asymptotic covariance matrix estimation, which we refer to as online batch-means method. The construction does not require prior knowledge of the total sample

size. Immediate updates from $\widehat{\Sigma}_n$ to $\widehat{\Sigma}_{n+1}$ can be performed recursively as new data are coming in, which fits our online setting. To achieve this goal, we design a novel construction of batches with time-varying size, which substantially extends the one in Chen et al. (2020). Similar to the recursive nature of SGD, our algorithm is also recursive and it updates the covariance matrix estimate once at a time only through the stochastic gradient within $O(d^2)$ computation. Note that since we are learning a $d \times d$ covariance matrix, it requires at least $O(d^2)$ computation to update the covariance matrix estimates. In the important special case of marginal inference of each coordinate of the parameter vector, our online batch-means estimator only needs to compute and store diagonals of the covariance matrix estimate, which only require $O(d)$ computation and $O(d)$ memory. The idea of online estimation is motivated by Wu (2009), who studied the estimation of long-run variances of stationary and ergodic processes. As mentioned above, the SGD iterates in Equation (2) form a nonhomogeneous (nonstationary) Markov Chain since the step size η_k decays as k increases, for example $\eta_k = \eta k^{-\alpha}$ for $\alpha \in (1/2, 1)$ as suggested by Polyak and Juditsky (1992). Hence, the asymptotic behaviors of SGD and stationary processes are fundamentally different. The construction, which is associated with batch sizes, is novel and different for SGD iterates and stationary sequences. This nonstationarity also brings substantial difficulties in technical analysis. The convergence of our estimator is far from being trivial. We formally establish the consistency result and obtain the convergence rate of our online estimator in Section 3.

We summarize our contributions as follows. We propose a fully online approach to estimate the asymptotic covariance matrix of the ASGD solution and conduct statistical inference. The fully online fashion allows efficient sequentially updating. It is important for online learning, where data come in a stream and real-time update of predictions is needed before seeing future data. It has potential applications such as online advertisement placement and online web ranking (Richardson, Dominowska, and Ragno 2007; Zhang et al. 2016). Our method is efficient in both computation and memory. In particular, the computational and memory complexity at the update step is $O(d^2)$, and the total computational cost only scales linearly in n. In terms of theoretical merits, the proposed estimator is the first fully online fashion estimator with rigorous convergence property for asymptotic covariance of ASGD. We show that the convergence rate of our online estimator is comparable to the offline counterparts.

1.2. Organization and Notation

The remainder of this article is organized as follows. In Section 2, we propose the online estimator (two versions) for the asymptotic covariance matrix of ASGD iterates and corresponding algorithms. In Section 3, we show that the online estimator is consistent and obtains the desired convergence rate. Also, confidence intervals/regions based on our online estimator are constructed for statistical inference. Section 4 provides a simulation study to demonstrate the convergence rate of the online estimator and the asymptotically valid coverage of the confidence intervals. Further discussion and future work are presented in Section 5.

Throughout the article, for a vector $\mathbf{a} = (a_1, \ldots, a_d) \in \mathbb{R}^d$, $\|\mathbf{a}\|_2$ is defined as the vector l_2 norm $\|\mathbf{a}\|_2 = \left(\sum_{i=1}^d a_i^2\right)^{1/2}$. For a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{d\times d}$, we use $\|\mathbf{A}\|_F$ to denote its Frobenius norm $\|\mathbf{A}\|_F = \left(\sum_{i=1}^d \sum_{j=1}^d a_{ij}^2\right)^{1/2}$, and $\|\mathbf{A}\|_2$ to denote its operator norm $\|\mathbf{A}\|_2 = \max_{\|x\|_2\le 1} \|\mathbf{A}x\|_2$. When $\mathbf{A}$ is positive semidefinite, λ_A denotes the largest eigenvalue of $\mathbf{A}$ and $\mathrm{tr}(\mathbf{A})$ denotes its trace. We use $\mathbf{I}_d$ to denote a $d \times d$ identity matrix. For positive sequences $\{a_n\}_{n\in\mathbb{N}}$ and $\{b_n\}_{n\in\mathbb{N}}$, $a_n \lesssim b_n$ means there exists some constant C such that $a_n \le Cb_n$ for all large n. And $a_n \asymp b_n$ if both $a_n \lesssim b_n$ and $b_n \lesssim a_n$ hold. For $t \in \mathbb{R}$, $\lfloor t \rfloor$ is the largest integer less than or equal to t. For notational simplicity, we use notation C for constants which can take different values in different equations. We define conditional expectation $\mathbb{E}_n(\cdot) = \mathbb{E}(\cdot|\mathcal{F}_n)$, where $\mathcal{F}_n$ is σ-algebra generated by $\{\xi_i\}_{i\le n}$. Moreover, we use $\Rightarrow$ to denote convergence in distribution.

2. Online Approach

We first introduce a time varying batch scheme used in our online approach. Consider infinite sequentially arriving SGD iterates $\{x_i\}_{i=1,2,\ldots}$ in (2). Let $\{a_m\}_{m\in\mathbb{N}}$ be a strictly increasing integer-valued sequence with $a_1 = 1$. For the ith iterate x_i, we consider a data block B_i including iterates from past iterations t_i to i, that is,

$$B_i = \{x_{t_i}, \ldots, x_i\},$$

where t_i is the index of iterate we trace back to at the ith step. The value of t_i is determined by the sequence $\{a_m\}_{m\in\mathbb{N}}$ through $t_i = a_m$ when $i \in [a_m, a_{m+1})$. For example, $t_i = \left\lfloor \sqrt{i} \right\rfloor^2$ if $a_m = m^2$. In this case, we have
$B_1 = \{x_1\}$, $B_2 = \{x_1, x_2\}$, $B_3 = \{x_1, x_2, x_3\}$,
$B_4 = \{x_4\}$, $B_5 = \{x_4, x_5\}$, $B_6 = \{x_4, x_5, x_6\}$, $B_7 = \{x_4, x_5, x_6, x_7\}$,
$B_8 = \{x_4, x_5, x_6, x_7, x_8\}$,
$B_9 = \{x_9\}$, $B_{10} = \{x_9, x_{10}\}$, $B_{11} = \{x_9, x_{10}, x_{11}\}, \ldots$.
We can see that the batch sizes are time-varying. The blocks $\{B_i : a_m \le i < a_{m+1}\}$ can also be viewed as the so-called forward scans in block subsampling (McElroy et al. 2007; Nordman, Bunzel, and Lahiri 2013). That is, given nonoverlapping blocks $\{x_{a_m}, \ldots, x_{a_{m+1}-1}\}$, the forward scans are overlapping blocks of sequentially increasing length starting from x_{a_m}.

2.1. Online Covariance Matrix Estimator Based on Batch Means

Based on blocks $\{B_i\}_{i\in\mathbb{N}}$, the covariance matrix estimator is defined as the sum of squared block sums (centered) divided by the sum of block lengths, that is, at the nth step

$$\widehat{\Sigma}_n = \frac{\sum_{i=1}^n \left(\sum_{k=t_i}^i x_k - l_i\bar{x}_n\right)\left(\sum_{k=t_i}^i x_k - l_i\bar{x}_n\right)^T}{\sum_{i=1}^n l_i}, \quad (5)$$

where $l_i = |B_i| = i - t_i + 1$ denotes the length of B_i.

The novel idea of constructing data block B_i, which only includes past iterates, is the key to make the algorithm fully

online. Next, we will show that the estimate $\widehat{\Sigma}_n$ can be computed recursively. Let W_i denote the sum of the block $B_i = \{x_{t_i}, \ldots, x_i\}$, that is,

$$W_i = \sum_{k=t_i}^{i} x_k. \tag{6}$$

When $t_{i+1} = t_i = a_m$ for some m, $B_{i+1} = B_i \cup \{x_{i+1}\}$ and

$$W_{i+1} = W_i + x_{i+1},\ l_{i+1} = l_i + 1.$$

When $t_{i+1} = a_{m+1}$ for some m, we start a new block $B_{i+1} = \{x_{i+1}\}$ and

$$W_{i+1} = x_{i+1},\ l_{i+1} = 1.$$

We can see that both the batch sum W_i and the batch length l_i can be updated recursively. With the notation of W_i, the estimator in (5) can be expressed as

$$\widehat{\Sigma}_n = \frac{\sum_{i=1}^n W_i W_i^T + \sum_{i=1}^n l_i^2 \bar{x}_n \bar{x}_n^T - \left(\sum_{i=1}^n l_i W_i\right) \bar{x}_n^T - \bar{x}_n \left(\sum_{i=1}^n l_i W_i\right)_n^T}{\sum_{i=1}^n l_i}. \tag{7}$$

To further simplify the form, we introduce

$$\begin{aligned} V_n &= \sum_{i=1}^{n} W_i W_i^T,\ P_n = \sum_{i=1}^{n} l_i W_i. \\ v_n &= \sum_{i=1}^{n} l_i, \quad \text{and} \quad q_n = \sum_{i=1}^{n} l_i^2. \end{aligned} \tag{8}$$

They can be computed recursively since both W_i and l_i can be updated recursively. Now, $\widehat{\Sigma}_n$ in Equation (5) can be finally rewritten as

$$\widehat{\Sigma}_n = \frac{V_n + q_n \bar{x}_n \bar{x}_n^T - P_n \bar{x}_n^T - \bar{x}_n P_n^T}{v_n}. \tag{9}$$

All five components in Equation (9): $V_n, q_n, P_n, v_n, \bar{x}_n$ can be updated recursively. Thus, $\widehat{\Sigma}_n$ can be updated through results in the $(n-1)$th step and the new iterate x_n within $O(d^2)$ computation.

To summarize, we propose Algorithm 1. As shown in Algorithm 1, the five components of $\widehat{\Sigma}_{n+1}$ can be easily updated from their values in the nth step. There is no need to store all the outcomes in the previous steps. The memory complexity is $O(d^2)$, independent of the sample size n. In the update step, the computational complexity is also $O(d^2)$. The total computational cost scales linearly in n. The algorithm is much more efficient compared to non-recursive methods and naturally fits online learning scenarios.

2.1.1. An Alternative Version

The estimate $\widehat{\Sigma}_n$ in Equation (5) includes squared block sums from all n blocks $\{B_i\}_{i=1,2,\ldots,n}$. Block B_i and B_j are overlapped when $a_m \le i < j < a_{m+1}$ for some m. So $\widehat{\Sigma}_n$ in Equation (5) is a full overlapping version of the online batch-means estimator. We also introduce an alternative nonoverlapping version with a slightly simpler form which has a comparable performance. As data arriving sequentially, we follow the same batch scheme above to construct $\{B_i\}_{i=1,2,\ldots,}$ while only include a few squared block sums. At the nth step, define set $S_n = \{n\} \bigcup \{a_i - 1 : i > 1, a_i \le n\}$. Consider a set of nonoverlapping blocks $\{B_i\}_{i \in S_n}$, that is,

$$\underbrace{\{x_{a_1}, \ldots, x_{a_2-1}\}}_{B_{a_2-1}}, \ldots, \underbrace{\{x_{a_{m-1}}, \ldots, x_{a_m-1}\}}_{B_{a_m-1}}, \underbrace{\{x_{a_m}, \ldots, x_n\}}_{B_n}\}.$$

The alternative nonoverlapping estimate at the nth step includes squared block sums of $\{B_i\}_{i \in S_n}$. It is then defined as

$$\widehat{\Sigma}_{n,\mathrm{NOL}} = \frac{1}{n} \sum_{i \in S_n} \left(\sum_{k=t_i}^{i} x_k - l_i \bar{x}_n\right) \left(\sum_{k=t_i}^{i} x_k - l_i \bar{x}_n\right)^T. \tag{10}$$

The nonoverlapping version estimator is also recursive and can perform a real-time update. The algorithm is almost the same as the overlapping one with same computational and memory complexity. One can follow the derivation of Algorithm 1 to get Algorithm 2.

In the stationary process case, Lahiri (2003, 1999) showed that the mean squared error of the classic (non-recursive) nonoverlapping batch-means estimate is 33% larger than that of its overlapping version, while the convergence rates are the same. The comparison between the full overlapping version and the nonoverlapping version of our online estimators is more complicated in the nonstationary case. In Section 3.3, we provide upper bounds for estimation errors for both overlapping and nonoverlapping estimators. The two upper bounds are of the same order. The nonoverlapping version is easier to analyze theoretically, given its simpler structure. In the mean estimation model, we can obtain the precise order of the mean squared error for the nonoverlapping one; see Section 3.1. We also compare the empirical performance of the two versions

Algorithm 1: Update ASGD iterate and covariance matrix estimate recursively

Input: function $f(\cdot)$, parameter (α, η), step size $\eta_i = \eta i^{-\alpha}$ for $i \ge 1$, predefined sequence $\{a_m\}_{m \in N}$.
Initialize:
$m_0 = l_0 = 0, v_0 = P_0 = q_0 = V_0 = W_0 = \bar{x}_0 = 0, x_0$;
For $n = 0, 1, 2, 3, \ldots$
Receive: new data ξ_{n+1}
Do the following update:
1. $x_{n+1} = x_n - \eta_{n+1} \nabla f(x_n, \xi_{n+1})$;
2. $\bar{x}_{n+1} = (n\bar{x}_n + x_{n+1})/(n+1)$;
3. **if** $n+1 = a_{m_n+1}$, **then**:
$m_{n+1} = m_n + 1; l_{n+1} = 1; W_{n+1} = x_{n+1}$;
else:
$m_{n+1} = m_n; l_{n+1} = l_n + 1$;
$W_{n+1} = W_n + x_{n+1}$;
4. $q_{n+1} = q_n + l_{n+1}^2$;
5. $v_{n+1} = v_n + l_{n+1}$;
6. $V_{n+1} = V_n + W_{n+1} W_{n+1}^T$;
7. $P_{n+1} = P_n + l_{n+1} W_{n+1}$;
8.
$S = V_{n+1} + q_{n+1} \bar{x}_{n+1} \bar{x}_{n+1}^T - P_{n+1} \bar{x}_{n+1}^T - \bar{x}_{n+1} P_{n+1}^T$;
Output: ASGD estimator $\bar{x}_{n+1}$, estimated covariance $\widehat{\Sigma}_{n+1} = S/v_{n+1}$

Algorithm 2: Update ASGD estimator and covariance matrix estimate (nonoverlapping version) recursively

Input: function $f(\cdot)$, parameter (α, η), step size $\eta_i = \eta i^{\alpha}$ for $i \geq 1$, predefined sequence $\{a_m\}_{m\in\mathbb{N}}$.
Initialize:
$m_0 = l_0 = 0, v_0 = P_0 = q_0 = V_0 = W_0 = \bar{x}_0 = 0, x_0$;
For $n = 0, 1, 2, 3, \ldots$
Receive: new data ξ_{n+1}
Do the following update:
1. $x_{n+1} = x_n - \eta_{n+1}\nabla f(x_n, \xi_{n+1})$;
2. $\bar{x}_{n+1} = (n\bar{x}_n + x_{n+1})/(n+1)$;
4. **if** $n + 1 = a_{m_n+1}$, **then**:
$m_{n+1} = m_n + 1; l_{n+1} = 1; W_{n+1} = x_{n+1}$;
$q_{n+1} = q_n + l_n^2; V_{n+1} = V_n + W_n W_n^T$;
$P_{n+1} = P_n + l_n W_n$
else:
$m_{n+1} = m_n; l_{n+1} = l_n + 1$;
$W_{n+1} = W_n + x_{n+1}$;
$q_{n+1} = q_n; V_{n+1} = V_n; P_{n+1} = P_n$
5. $S' = W_{n+1}W_{n+1}^T + l_{n+1}^2 \bar{x}_{n+1}\bar{x}_{n+1}^T - l_{n+1}W_{n+1}\bar{x}_{n+1}^T - l_{n+1}\bar{x}_{n+1}W_{n+1}^T$;
6.
$S = V_{n+1} + q_{n+1}\bar{x}_{n+1}\bar{x}_{n+1}^T - P_{n+1}\bar{x}_{n+1}^T - \bar{x}_{n+1}P_{n+1}^T + S'$;
Output: ASGD estimator $\bar{x}_{n+1}$, estimated covariance $\widehat{\Sigma}_{n+1,NOL} = S/(n+1)$

in Section 4.1. However, it is hard to tell which one is more efficient based on the simulation results. We leave the rigorous comparison as a future research problem by extending Lahiri (2003) to nonstationary processes.

Remark 2.1 (Comparison with the non-recursive batch-means covariance matrix estimator). The nonoverlapping version (10) appears similar to the batch-means estimator (Chen et al. 2020). However, the batch schemes of the two methods are fundamentally different. Chen et al. (2020) split n iterates of SGD into $M+1$ nonoverlapping blocks, where M and batch sizes $b_{m,n}$ $(m = 0, \ldots, M)$ are chosen based on n for desired convergence. With $e_{m,n}$ denoting the ending index of the kth block, the covariance matrix estimator at nth iteration in Chen et al. (2020) is defined as

$$\widehat{\Sigma}_{n,\mathrm{BM}} = \frac{1}{M}\sum_{m=1}^{M} b_{m,n}\left(\sum_{k=e_{m-1,n}+1}^{e_{m,n}} x_k/b_{m,n} - \bar{x}_n\right)\left(\sum_{k=e_{m-1,n}+1}^{e_{m,n}} x_k/b_{m,n} - \bar{x}_n\right)^T, \tag{11}$$

where $e_{M,n} = n$. The optimal batch size setting as suggested in Chen et al. (2020) is $e_{m,n} = ((m+1)/(M+1))^{1/(1-\alpha)}\, n$ with the number of batches $M = n^{(1-\alpha)/2}$. Since $e_{m,n}$ must depend on n to ensure the desired convergence rate at the nth iteration, there is no simple algebraic relation between $\widehat{\Sigma}_{n,\mathrm{BM}}$ and $\widehat{\Sigma}_{n+1,\mathrm{BM}}$. So the batch-means estimator (Chen et al. 2020) is only suitable for offline tasks requiring final prediction/inference given the *prespecified* total sample size n. In contrast, our fully online estimator can sequentially improve over each iteration. Also, n does not need to be specified beforehand.

Remark 2.2 (Choice of batch-sizes when n is unknown). Chen et al. (2020) also proposed an approach based on a target error tolerance to apply the batch-means estimator when n is unknown. In particular, given the prespecified error ϵ, Chen et al. (2020) propose to set the ending index of the kth batch by $e_k = \left((k+1)C\epsilon^{-2}\right)^{1/(1-\alpha)}$, where C is a constant. The approach indeed enables an online updating, thus achieve the goal of recursive processing. However, choosing the constant C can be difficult or arbitrary in online settings. Moreover, there is a fundamental difference. The approach in Chen et al. (2020) only ensures that the expected spectrum norm loss of the covariance matrix is smaller than ϵ (up to a constant) for large n, rather than goes to 0. In other words, the covariance matrix estimator is not necessarily *consistent*. While our online method constantly improves the covariance matrix estimate as $n \to \infty$, and the estimation error goes to 0.

2.1.2. Choice of Batch Sizes

The remaining question is to specify the sequence $\{a_m\}_{m\in\mathbb{N}}$. This predefined sequence does not depend on n. This ensures that we can construct batches even if the total number of data is unknown (which is a typical situation), and the incoming data will not affect the recursive estimation process. In Section 3.3, we show that a_m is required to take a polynomial form so that the estimator is consistent. Next, we shall give some intuitive explanation and one example of choice.

The formula in Equation (5) bears a certain similarity to the sample covariance matrix $S_n = n^{-1}\sum_{i=1}^{n}(x_i - \bar{x}_n)(x_i - \bar{x}_n)^T$. On the other hand, in contrast to the standard sample covariance matrix where $\{x_i\}_{i\geq 1}$ are independent, our SGD iterates in Equation (5) are highly correlated. In other words, we cannot ignore the covariance between data as in the construction of the sample covariance matrix. According to Equation (2), the correlation between x_i and x_j diminishes as the distance $|j - i|$ becomes larger, while the correlation between x_i and x_{i+1} becomes stronger as i goes to infinity. The idea of online estimation is to choose sequence $(a_m)_{m\in\mathbb{N}}$ and form nonoverlapping blocks $\{B_{a_m-1}\}_{m>1}$ as mentioned above such that the correlation between x_i and x_j is sufficiently small when they are in different nonoverlapping blocks. So when considering the effect of x_i, we trace back to the starting point of the nonoverlapping block x_i belongs to, that is, construct data block $B_i = \{x_{t_i}, \ldots, x_i\}$. Recall that the ith iterate x_i through SGD takes the form

$$x_i = x_{i-1} - \eta_i \nabla f(x_{i-1}, \xi_i).$$

Let $\delta_i = x_i - x^*$ be the error sequence, where x^* is the minimizer in Equation (1). Then

$$\delta_i = \delta_{i-1} - \eta_i \nabla F(x_{i-1}) + \eta_i \epsilon_i, \tag{12}$$

where $\epsilon_i = \nabla F(x_{i-1}) - \nabla f(x_{i-1}, \xi_i)$. Note that $\nabla F(x^*) = 0$ since x^* is the minimizer of $F(x)$. By Taylor's expansion of $\nabla F(x_{i-1})$ around x^*, we have $\nabla F(x_{i-1}) \approx \nabla A\delta_{i-1}$, where $A = \nabla^2 F(x^*)$. Thus, by modifying Equation (12) with $\nabla F(x_{i-1})$ approximated by $A\delta_{i-1}$, for large i

$$\delta_i \approx (I - \eta_i A)\delta_{i-1} + \eta_i \epsilon_i. \tag{13}$$

Then for the ith iterate x_i and the jth iterate x_j (assume $j < i$), the strength of correlation between them is roughly

$$\Pi_{k=j+1}^{i} \|I_d - \eta_k A\|_2 \leq (1 - \eta\lambda_A i^{-\alpha})^{i-j}, \tag{14}$$

when $\eta_k = \eta k^{-\alpha}$. To make the correlation small, one can choose $i - j \approx Ki^{(\alpha+1)/2}$, where K is a constant. Then the correlation is less than $(1 - \eta\lambda_A i^{-\alpha})^{Ki^{\alpha} i^{(1-\alpha)/2}}$, which goes to zero as i goes to infinity. Combining the correlation between x_i, x_j and the form of $i-j$, a reasonable setting is that the sequence $\{a_m\}_{m\in\mathbb{N}}$ satisfies

$$a_m - a_{m-1} = Ka_m^{(\alpha+1)/2}. \tag{15}$$

Let a_m increase polynomially, that is, $a_m = Cm^{\beta}$ for some constant C. We obtain $\beta = 2/(1-\alpha)$ by solving Equation (15). Thus, a natural choice of a_m is

$$a_m = \left\lfloor Cm^{2/(1-\alpha)} \right\rfloor. \tag{16}$$

This is also the best choice in the general setting, as discussed in Section 3.3. However, the best choice of β may change considering specific objective functions.

2.2. Statistical Inference

Now the limiting covariance matrix Σ can be approximated through the online estimation proposed above. Let $0 < q < 1$. Based on the asymptotic normality of ASGD in Equation (4), the $(1-q)100\%$ confidence interval for x_i^*, the ith coordinate of x^*, can be constructed as

$$\left[\bar{x}_{n,i} - z_{1-q/2}\sqrt{\widehat{\sigma}_{ii}/n},\ \bar{x}_{n,i} + z_{1-q/2}\sqrt{\widehat{\sigma}_{ii}/n}\right], \tag{17}$$

where $\bar{x}_{n,i}$ is the ith coordinate of $\bar{x}_n$, $z_{1-q/2}$ is the $(1-q/2)$th percentile of the standard Gaussian distribution and $\widehat{\sigma}_{ii}$ is the ith diagonal of the covariance matrix estimate. The confidence interval is constructed in a fully online fashion since both $\bar{x}_{n,i}$ and $\widehat{\sigma}_{ii}$ can be computed recursively. Joint confidence regions and general form of confidence intervals are discussed in Section 3.4.

2.2.1. Relation to Empirical Likelihood

As pointed out by a reviewer, the construction of the nonoverlapping version estimator shares a similar spirit with the blocking scheme and covariance estimator by Kim, Lahiri, and Nordman (2013), who developed a progressive block empirical likelihood (PBEL) method. They considered a stationary, weakly dependent sequence $(X_1, \ldots, X_n)$ with mean μ such that the CLT $\sqrt{n}(\bar{X}_n - \mu) \Rightarrow N(0, \sigma^2)$ holds. The variance estimator $\widehat{\sigma}^2_{n,\mathrm{NOL}}$ in Kim, Lahiri, and Nordman (2013) matched our scheme in Section 2.1.1 with $a_m = (m-1)m/2 + 1$ (or the ith block has length i) and is shown to be a consistent variance estimator. The chi-squared limit of the log-likelihood ratio based on PBEL is established following the consistency of $\widehat{\sigma}^2_{n,\mathrm{NOL}}$. It would be interesting to see if one can obtain similar results as the PBEL ratio and establish a limiting distribution that can be used to calibrate confidence regions in the SGD case here.

3. Theoretical Results

3.1. Preamble: Mean Estimation Model

Before investigating the convergence property of the online batch-means estimators in the general setting, we shall look at the simple mean estimation example. Taking advantage of the simpler structure of the nonoverlapping version, we can obtain the exact order of convergence. Consider the mean estimation model

$$y = x^* + e,$$

where $x^* \in \mathbb{R}$ is the mean we want to estimate, e is the random error with mean 0. Let $\{y_i\}_{i\in\mathbb{N}}$ be a sequence of iid sample from the model. Consider the squared loss function at x, $F(x) = (y - x)^2/2$. The ith SGD iterate takes the form

$$x_i = x_{i-1} + \eta_i(y_i - x_{i-1}), i \geq 1, \tag{18}$$

where we choose the step size $\eta_i = \eta i^{-\alpha}, \alpha \in (1/2, 1)$. Then the error $\delta_i = x_i - x^*$ takes the form

$$\delta_i = (1 - \eta_i)\delta_{i-1} + \eta_i e_i.$$

In this case, one can have an explicit form of $\mathrm{var}(\sqrt{n}\bar{x}_n)$ and $\widehat{\Sigma}_{n,\mathrm{NOL}}$. Additionally, we can have an explicit form for the order of magnitude of the mean squared error of $\widehat{\Sigma}_{n,\mathrm{NOL}}$. Let the variance $\mathrm{var}(\sqrt{n}\bar{x}_n) = \sigma_n^2$. We have the following proposition.

Proposition 3.1. For $m \geq 2$, let $a_m = \lfloor cm^{\beta} \rfloor$, where $\beta > 1$ and $c > 0$ are constants. Given the SGD iterates defined in (18), we have

$$\mathbb{E}(\widehat{\Sigma}_{n,\mathrm{NOL}} - \sigma_n^2)^2 \asymp n^{-1/\beta} + n^{2\alpha+2/\beta-2}. \tag{19}$$

Choose $\beta = 3/(2(1-\alpha))$. In the mean estimation model, the above proposition asserts that the convergence rate of the mean squared error of our recursive nonoverlapping variance estimate is $n^{-2(1-\alpha)/3}$. For α close to $1/2$, the latter rate approaches $n^{-1/3}$. This rate is faster than that of the batch-means estimator in Chen et al. (2020), which approaches $n^{-1/4}$. So, besides the advantage of the recursive property, our estimator may improve the convergence rate.

In the general setting, the analysis is much more complicated due to the nonlinearity. Upper bounds for the convergence rates of online estimators for both overlapping and nonoverlapping versions are given in Section 3.3.

3.2. Assumptions and Existing Convergence Results

In the work of Polyak and Juditsky (1992), assumptions on the objective function $F(x)$ and the gradient difference are proposed to prove the asymptotic normality of ASGD estimate. Those assumptions are necessary for our problem since we adopt the ASGD as the point estimator and require the asymptotic normality for statistical inference. Those assumptions, as well as some error bounds, are also proposed in other literature. We impose similar assumptions and review some existing results in this section.

Assumption 1. Assume that the objective function $F(x)$ is continuously differentiable and strongly convex with parameter $\mu > 0$. That is, for any x_1 and x_2,

$$F(x_2) \geq F(x_1) + \langle \nabla F(x_1), x_2 - x_1 \rangle + \frac{\mu}{2}\|x_1 - x_2\|_2^2.$$

Furthermore, assume that $\nabla^2 F(x^*)$ exists and $\nabla F(x)$ is Lipschitz continuous in the sense that there exist $L > 0$ such that,

$$\|\nabla F(x_1) - \nabla F(x_2)\|_2 \leq L\|x_1 - x_2\|_2.$$

Assumption 2. For the nth iteration, define error $\delta_n = x_n - x^*$ and gradient difference $\epsilon_n = \nabla F(x_{n-1}) - \nabla f(x_{n-1}, \xi_n)$. Recall that $\mathbb{E}_n(\cdot) = \mathbb{E}(\cdot|\xi_n, \xi_{n-1}, \ldots)$. The following hold:

1. The function $f(x, \xi)$ is continuously differentiable with respect to x for any ξ and $\|\nabla f(x, \xi)\|_2$ is uniformly integrable for any x. So $\mathbb{E}_{n-1}\left[\nabla f(x_{n-1}, \xi_n)\right] = \nabla F(x_{n-1})$, which implies that $\mathbb{E}_{n-1}(\epsilon_n) = 0$.
2. The conditional covariance of ϵ_n has an expansion around S which satisfies

$$\left\|\mathbb{E}_{n-1}\left(\epsilon_n \epsilon_n^T\right) - S\right\|_2 \leq C\left(\|\delta_{n-1}\|_2 + \|\delta_{n-1}\|_2^2\right), \quad (20)$$

   where $C > 0$ is some constant. Here S is defined in Equation (3).
3. There exists a constant $C > 0$ such that the fourth conditional moment of ϵ_n is bounded by

$$\mathbb{E}_{n-1}\left(\|\epsilon_n\|_2^4\right) \leq C\left(1 + \|\delta_{n-1}\|_2^4\right).$$

Assumption 1 imposes strong convexity of the objective function $F(x)$ and Lipschitz continuity of its gradient. Assumption 2 asserts the regularity and the bound of the noisy gradient. These assumptions are widely used in SGD literature (Ruppert 1988; Polyak and Juditsky 1992; Moulines and Bach 2011; Rakhlin, Shamir, and Sridharan 2012). With these assumptions, we have the asymptotic normality for averaged SGD iterates by Polyak and Juditsky (1992) and Ruppert (1988). We also review the error bound for SGD iterates in Lemma 3.1.

Lemma 3.1. Under Assumptions 1 and 2, for some constant $C > 0$ and $n_0 \in \mathbb{N}$, we have for any $n > n_0$, the sequence of error $\delta_n = x_n - x^*$ satisfies

$$\begin{aligned}\mathbb{E}(\|\delta_n\|_2) &\leq Cn^{-\alpha/2}(1 + \|\delta_0\|_2),\\ \mathbb{E}(\|\delta_n\|_2^2) &\leq Cn^{-\alpha}(1 + \|\delta_0\|_2^2),\\ \mathbb{E}(\|\delta_n\|_2^4) &\leq Cn^{-2\alpha}(1 + \|\delta_0\|_2^4),\end{aligned}$$

when the step size is chosen to be $\eta_n = \eta n^{-\alpha}$ with $1/2 < \alpha < 1$.

3.3. Convergence Properties for the Online Estimator

Theorem 3.1. Under Assumptions 1 and 2, let $a_m = \lfloor Cm^\beta \rfloor$, where $C > 0$ is a constant, $\beta > (1-\alpha)^{-1}$. Set step size at the ith iteration as $\eta_i = \eta i^{-\alpha}$ with $1/2 < \alpha < 1$. Then for $\widehat{\Sigma}_n$ defined in Equation (5)

$$\mathbb{E}\left\|\widehat{\Sigma}_n - \Sigma\right\|_2 \lesssim n^{-1/(2\beta)} + n^{(\alpha-1)/2+1/(2\beta)}. \quad (21)$$

Theorem 3.1 shows that as $n \to \infty$, the estimator $\widehat{\Sigma}_n$ converges to the limiting covariance matrix of the averaged SGD iterates in terms of operator norm loss. The convergence rate is associated with the parameters α and β. We state the following Corollary 3.1 to suggest the best choice of β.

Corollary 3.1. Under conditions in Theorem 3.1 and let $\beta = 2/(1-\alpha)$, we have

$$\mathbb{E}\left\|\widehat{\Sigma}_n - \Sigma\right\|_2 \lesssim n^{-(1-\alpha)/4}. \quad (22)$$

Remark 3.1. This convergence rate is the same as that of the nonrecursive batch-means estimator in Chen et al. (2020). According to Chen et al. (2020, corol. 4.5), the upper bound of the batch-means estimator is also $O(n^{-(1-\alpha)/4})$ with the prior knowledge of the sample size n. So we make it possible that online estimation of covariance matrix achieves the same efficiency as offline methods. The plug-in approach in Chen et al. (2020) achieved the rate of $O(n^{-\alpha/2})$ when the ith step size is chosen to be $i^{-\alpha}$. As a tradeoff, the online estimator enjoys efficient computation without the necessity of accessing Hessian information but pays the price in terms of the slower convergence rate.

Next, we will show in Theorem 3.2 that the alternative version $\widehat{\Sigma}_{n,\mathrm{NOL}}$ shares the same upper bound.

Theorem 3.2. Under conditions in Theorem 3.1, the alternative version $\widehat{\Sigma}_{n,\mathrm{NOL}}$ defined in Equation (10) satisfies

$$\mathbb{E}\left\|\widehat{\Sigma}_{n,\mathrm{NOL}} - \Sigma\right\|_2 \lesssim n^{-1/(2\beta)} + n^{(\alpha-1)/2+1/(2\beta)}. \quad (23)$$

3.4. Asymptotically Accurate Confidence Intervals/Regions

The next corollary shows that the confidence interval/region based on the online estimator achieves asymptotically correct coverage level $1 - q$ for a prespecified q with $0 < q < 1$.

Corollary 3.2. Under conditions in Theorem 3.1, as n goes to infinity

$$\mathbb{P}(x_i^* \in \mathrm{CI}_{q,n,i}) \to 1 - q, \quad (24)$$

where

$$\mathrm{CI}_{q,n,i} = \left[\bar{x}_{n,i} - z_{1-q/2}\sqrt{\widehat{\sigma}_{ii}/n},\ \bar{x}_{n,i} + z_{1-q/2}\sqrt{\widehat{\sigma}_{ii}/n}\right]$$

and $\widehat{\sigma}_{ii}$ is the ith diagonal of the online batch-means estimator $\widehat{\Sigma}_n$ (or $\widehat{\Sigma}_{n,\mathrm{NOL}}$). We can also construct joint confidence regions as follows:

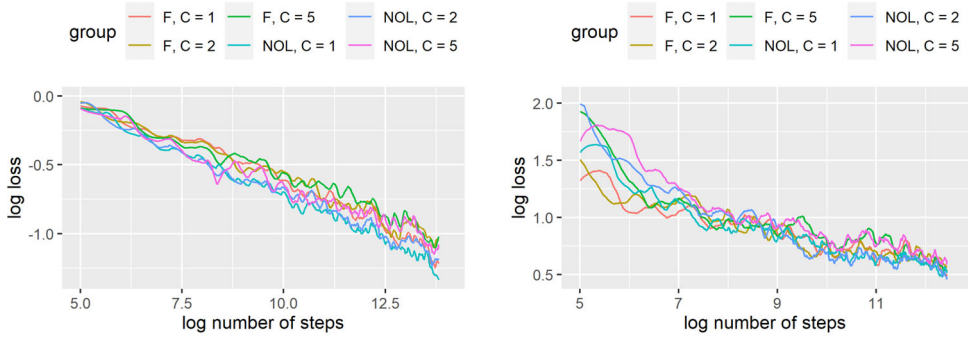
$$\mathbb{P}\left(x^* \in C_{q,n}\right) \to 1 - q, \quad (25)$$

where

$$C_{q,n} = \left\{x \in \mathbb{R}^d : n\left(\bar{x}_n - x\right)^T \widehat{\Sigma}_n^{-1}\left(\bar{x}_n - x\right) \leq \chi^2_{d,1-2/q}\right\}.$$

Corollary 3.2 constructs asymptotic valid confidence intervals for each coordinate of x^* and joint confidence regions for $x^* \in \mathbb{R}^d$. More generally, for any unit length vector $w \in \mathbb{R}^d$ (i.e., $\|w\|_2 = 1$), we have by Theorem 3.1 and Slutsky's theorem,

$$\frac{\sqrt{n}w^T(\bar{x}_n - x^*)}{\sqrt{w^T\widehat{\Sigma}_n w}} \Rightarrow N(0, 1). \quad (26)$$



(a) d=1 (b) d=5

Figure 1. Linear regression: Log loss (operator norm) of the estimated covariance matrix against the log of total number of steps. Here F denotes the full overlapping version (5), NOL denotes the nonoverlapping version (10), and C denotes the constant in $a_m = \lfloor Cm^{2/(1-\alpha)} \rfloor$.

Therefore, the $(1-q)100\%$ confidence interval for $w^T x^*$ can be constructed as

$$\left[w^T \bar{x}_n - z_{1-q/2}\sqrt{w^T \widehat{\Sigma}_n w/n}, w^T \bar{x}_n + z_{1-q/2}\sqrt{w^T \widehat{\Sigma}_n w/n} \right]. \tag{27}$$

3.4.1. Stopping Rule

In principle, SGD constantly improves the quality of $\bar{x}_n$, and our method constantly improves the covariance estimate $\widehat{\Sigma}_n$ as n grows. A natural questions is when can we stop updating $\bar{x}_n$ and $\widehat{\Sigma}_n$? There are several heuristics of stopping rules widely used in machine learning. For example, an online algorithm can stop when the neighboring estimates become sufficiently close. Or a more widely used approach in stopping SGD is to evaluate the error on a separate validation dataset and stops the SGD when the error becomes stable.

We can better answer this question and assess the SGD error based on the inference results, inspired by stopping rules for Markov chain Monte Carlo (MCMC) that rely on a Markov chain central limit theorem. Especially, one can apply the fixed-width sequential stopping rule in Jones et al. (2006), where the updating is terminated the first time when the width of the confidence interval for each component is small enough. More formally, for a desired tolerance of ϵ_i for the ith coordinate, the rule terminates updating the first time after the nth iteration when the following condition is satisfied for all the coordinates $i = 1, \ldots, d$,

$$t_* \frac{\widehat{\sigma}_{n,i}}{\sqrt{n}} + n^{-1} \le \epsilon_i,$$

where $\widehat{\sigma}_{n,i}$ is the ith diagonal of the online estimator $\widehat{\Sigma}_n$ (or $\widehat{\Sigma}_{n,\mathrm{NOL}}$), and t_* is an appropriate t-distribution quantile. For the joint inference, one may consider simplifying the relative standard deviation fixed-volume sequential stopping rule in Vats, Flegal, and Jones (2019), where updating is terminated the first time when the volume of the confidence region C_n (25) is small enough. For a desired tolerance of ϵ, the rule terminates updating the first time after the nth iteration when

$$\mathrm{Vol}(C_n)^{1/d} + n^{-1} \le \epsilon,$$

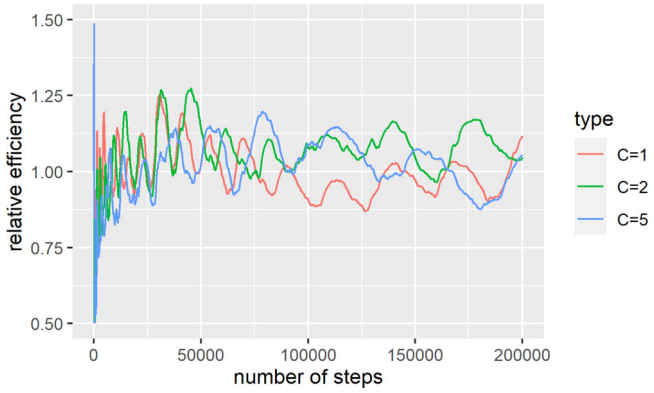


Figure 2. Relative efficiency (ratio of MSE) of the full overlapping version (5) and nonoverlapping version (10). We set $d = 5$ in linear regression. Here C denotes the constant in $a_m = \lfloor Cm^{2/(1-\alpha)} \rfloor$.

where $\mathrm{Vol}(C_n) = 2\left(\pi \chi_*^2/n\right)^{d/2} |\widehat{\Sigma}_n|^{1/2}/(d\Gamma(d/2))$, $|\cdot|$ denotes determinant, χ_*^2 is an appropriate chi-squared distribution quantile, and $\widehat{\Sigma}_n$ is our online estimator. We also include a simple simulation study of the stopping rule in the last section of the supplementary material.

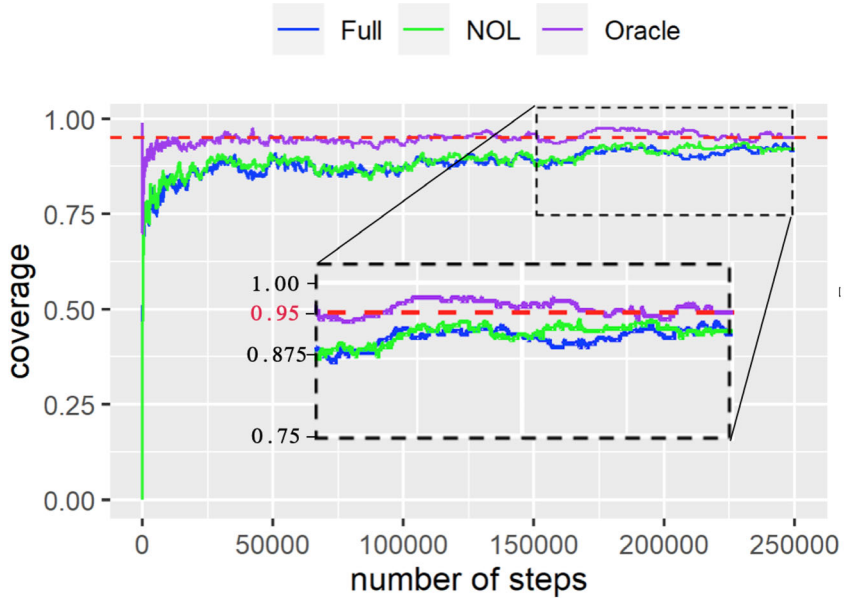
Remark 3.2. The original stopping rule in Vats, Flegal, and Jones (2019) avoided the practical issue of choosing ϵ with the idea of effective sample size (ESS). They consider an F-invariant Harris recurrent Markov chain and define a multivariate approach to ESS. The stopping rule in Vats, Flegal, and Jones (2019) terminated the MCMC simulation the first time the estimated ESS is larger than a prespecified lower bound. However, we need to redefine ESS in the nonstationary case, which requires more careful considerations. We will leave it as a future research direction.

4. Simulation Studies

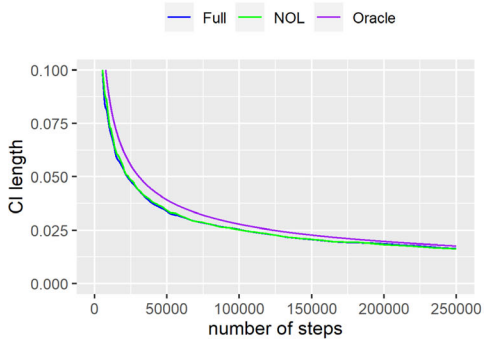
In this section, we evaluate the empirical performance of the proposed online approach. We focus on two classes of examples: linear regression and logistic regression. Let $\{\xi_i \equiv (a_i, b_i)\}_{i=1,2,\ldots}$ denotes an iid sequence of pairs, and x^* denote the true parameter in the models. In both linear regression

and logistic regression cases, $a_i \in \mathbb{R}^d$ is generated from $N(0, \mathbf{I}_d)$. In the former case, $b_i = a_i^T x^* + \epsilon_i$, where ϵ_i is independently generated from $N(0,1)$. In the latter case, $b_i|a_i \sim \text{Bernoulli}((1+\exp(-a_i^T x^*))^{-1})$. The loss function $f(\cdot)$ is defined as the negative log-likelihood function, so we have

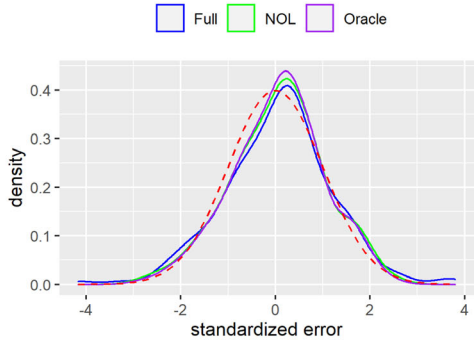
$$f(x, a_i, b_i) = \begin{cases} \frac{1}{2}(a_i^T x - b_i)^2 & \text{linear regression} \\ (1-b_i)a_i^T x & \\ \quad + \log(1+\exp(-a_i^T x)) & \text{logistic regression.} \end{cases}$$



(a) Empirical cover rate



(b) CI length



(c) Normality

Figure 3. Linear regression with $d = 5$: (a): Empirical coverage rate against the number of steps. Red dashed line denotes the nominal coverage rate of 0.95. (b): Length of confidence intervals. (c): Density plot for the standardized error. Red curve denotes the standard normal density.

Table 1. Empirical coverage rates: the average coverage rate for the nominal coverage probability 95%.

Linear model				
($d = 5$)	$n = 50{,}000$	$n = 80{,}000$	$n = 100{,}000$	n= 125,000
online-BM	0.894 (0.02177)	0.901 (0.02114)	0.917 (0.01951)	0.935 (0.01746)
BM	0.894 (0.02177)	0.904 (0.02085)	0.910 (0.02022)	0.928 (0.01831)
($d = 20$)	$n = 50{,}000$	$n = 100{,}000$	$n = 150{,}000$	$n = 200{,}000$
online-BM	0.904 (0.02078)	0.907 (0.02050)	0.910 (0.02022)	0.914 (0.01986)
BM	0.878 (0.02312)	0.901 (0.02121)	0.908 (0.02043)	0.910 (0.02029)
		Logistic model		
($d = 5$)	$n = 100{,}000$	$n = 200{,}000$	$n = 300{,}000$	$n = 400{,}000$
online-BM	0.828 (0.01011)	0.844 (0.00933)	0.875 (0.00770)	0.889 (0.00700)
BM	0.822 (0.01032)	0.847 (0.00919)	0.875 (0.00771)	0.885 (0.00721)
($d = 20$)	$n = 100{,}000$	$n = 300{,}000$	$n = 500{,}000$	$n = 700{,}000$
online-BM	0.791 (0.01167)	0.829 (0.01004)	0.845 (0.00926)	0.864 (0.00834)
BM	0.787 (0.01188)	0.827 (0.01011)	0.839 (0.00955)	0.859 (0.00856)

NOTE: Standard errors are reported in the brackets.

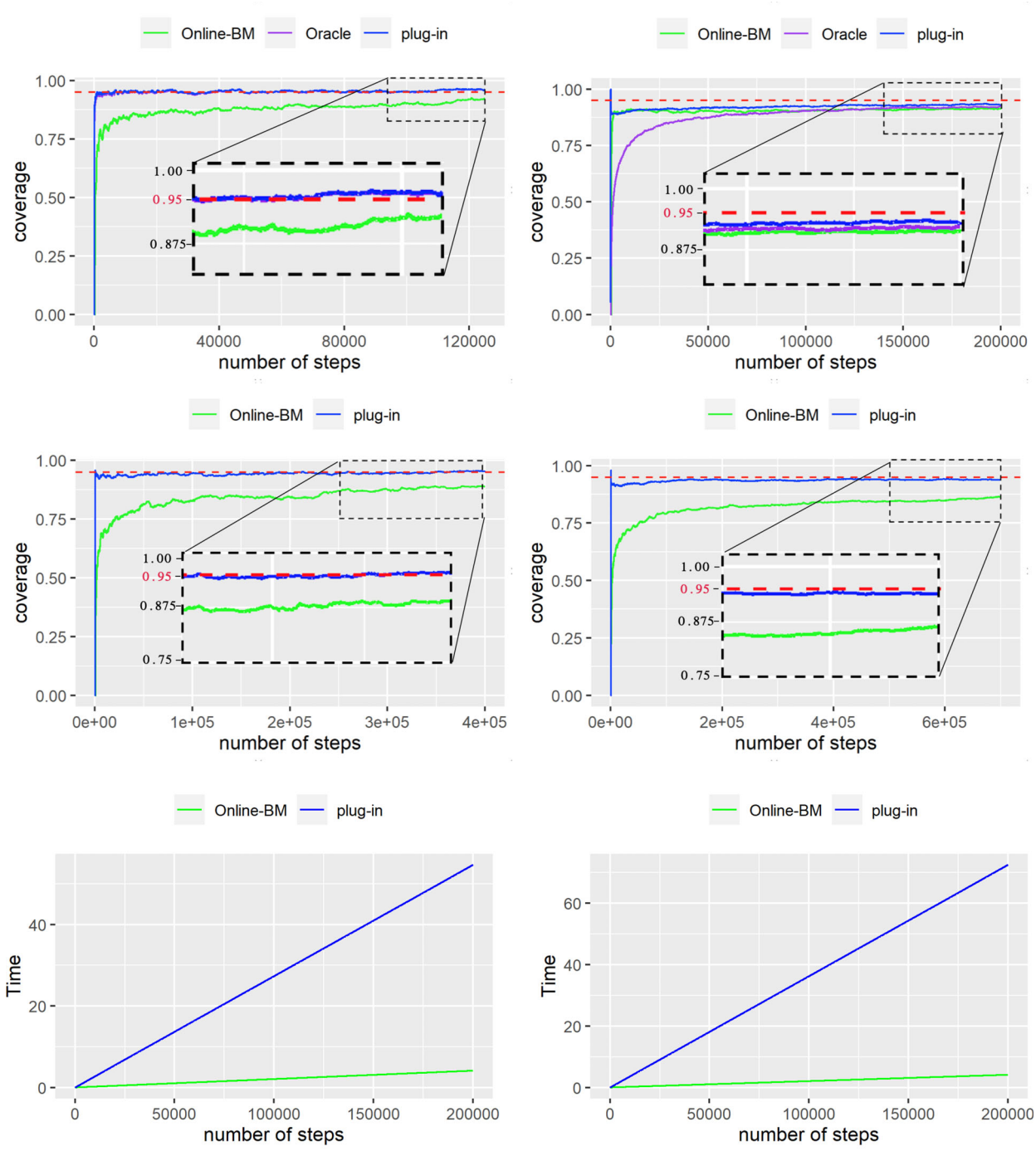


Figure 4. Comparison of online-BM and Plug-in estimators. First/Middle row: Empirical coverage rate against the number of steps in linear/logistic model. Red dashed line denotes the nominal coverage rate of 0.95. Third row: total computation time for updating covariance estimate and confidence intervals in SGD.

The true coefficient x^* is a d-dimensional vector linearly spaced between 0 and 1. In the SGD procedure, the step size η_j is set to be $0.5j^{-\alpha}$ and the parameter α is chosen to be 0.505. The sequence $\{a_k\}_{k\geq 1}$ in our online approach is chosen in the form of $a_m = \lfloor Cm^{2/(1-\alpha)} \rfloor$, for some constant C. All the measurements in the following discussions are averaged over 200 independent runs.

4.1. Empirical Performance of the Proposed Online Approach

4.1.1. Convergence of the Recursive Estimator

We focus on linear regression here since the true limiting covariance matrix is easy to compute. In the linear regression model described above

$$A = \mathbb{E}\left[\nabla^2 f(x^*)\right] = \mathbb{E}\left(aa^T\right) = \mathbf{I}_d,$$

$$S = \mathbb{E}\left([\nabla f(x^*, \xi)][\nabla f(x^*, \xi)]^T\right) = \mathbb{E}(\epsilon^2)\mathbb{E}\left(aa^T\right) = \mathbf{I}_d.$$

Then the limiting covariance matrix

$$\Sigma = A^{-1}SA^{-1} = \mathbf{I}_d.$$

We check the convergence of our proposed online estimators, both the full overlapping and the nonoverlapping versions, by computing the operator norm loss of the covariance matrix estimate, that is, $\|\widehat{\Sigma}_n - \Sigma\|_2$. Figure 1 shows that the log loss of the online estimators are approximately linear with the log number of steps and the slopes are about $-1/8$ for the large total number of steps. It suggests that both the full overlapping and the nonoverlapping versions converge to the limiting covariance matrix with the same convergence rate, about $O(n^{-1/8})$. We also compute the relative efficiency (MSE of the full overlapping version (5) divided by MSE of the nonoverlapping version (10)); see Figure 2. Their performances are comparable. Also, the performance is relatively insensitive to the choice of C in $a_m = \lfloor Cm^{2/(1-\alpha)} \rfloor$. Therefore, we will implement the nonoverlapping version and set $C = 1$ in the subsequent simulations without any specification.

4.1.2. Asymptotic normality and CI coverage

With the covariance matrix estimates, we construct 95% confidence intervals for the averaged coefficient $\mu = 1^T x^*$ according to Equation (27), that is,

$$\left[1^T\bar{x}_n - z_{1-q/2}\sqrt{1^T\widehat{\Sigma}_n 1/n}, 1^T\bar{x}_n + z_{1-q/2}\sqrt{1^T\widehat{\Sigma}_n 1/n}\right].$$

We also compute the oracle 95% confidence intervals based on the true limiting covariance matrix. Figure 3 shows that for both overlapping and nonoverlapping versions, the empirical coverage rate converges to 95%, and the standardized error $\sqrt{n}1^T(\bar{x}-x^*)/\sqrt{1^T\widehat{\Sigma}_n 1}$ is approximately standard normal. Also, the estimated CI length converges to the oracle length.

4.2. Comparison With Other Methods

In this section, we compare the performance of the proposed online estimator, which we refer to as online-BM in the subsequent numerical experiments, with other estimators for marginal inference of each individual regression coefficient. We consider both linear and logistic regression examples. The nominal coverage probability is set to 95%.

We first compare the empirical coverage rates of the proposed estimator with the plug-in estimator in Chen et al. (2020). As we mentioned in the introduction, the plug-in estimator requires the computation of the Hessian matrix (of the loss function) and its inverse. Figure 4 shows that our online estimator (online-BM) has a comparable performance as the plug-in estimator when the number of iterations is large enough. Although the online-BM has a slower convergence rate, it has an advantage in computational efficiency since it only uses the iterates from SGD. The online-BM is more desirable for practitioners when the computation is limited or only stochastic gradient information is available.

Next, we compare the finite sample coverage rate of the proposed online-BM estimator and the batch-means covariance matrix estimator from Chen et al. (2020), which we refer to as BM. Table 1 shows that the finite sample coverage rates of the two estimators are close to each other in all cases, and the finite sample performance of our method slightly outperforms Chen et al. (2020) when n is large. In fact, this is not a totally fair comparison for us since we implement the method in Chen et al. (2020) based on the prior knowledge of the exact sample size.

5. Conclusion and Future Work

In this article, we propose a fully online approach to estimate the asymptotic covariance matrix in SGD. The recursive algorithm to compute the covariance matrix estimate is computationally efficient. We demonstrate that the online batch-means covariance matrix estimator (both full overlapping version and nonoverlapping version) is consistent with the upper bound of convergence rate $O(n^{-(1-\alpha)/4})$ in the general case. Based on the estimated covariance matrix, we construct confidence intervals/regions with asymptotically correct coverage probabilities for the model parameters. As for future directions, it would be of interest to develop a lower bound result on the online estimation of limiting covariance matrices. With such a result, we will be able to tell whether the proposed estimator is rate-optimal. Also, as mentioned in Section 2.2.1, it would be interesting to see if one can obtain statistics similar to the PBEL ratio based on the nonoverlapping version online covariance estimator and establish a limiting distribution that can be used to calibrate confidence regions for SGD solutions without using the asymptotic normality results.

Acknowledgments

We thank to the anonymous reviewers and editors for the constructive feedbacks that significantly improved our article. Wanrong Zhu and Wei Biao Wu thank to the support from NSF via NSF-DMS-1916351 and NSF-DMS-2027723. Xi Chen thank to the support from NSF via IIS-1845444.

References

Bach, F. R., and Moulines, E. (2013), "Non-Strongly-Convex Smooth Stochastic Approximation With Convergence Rate $\mathcal{O}(1/n)$," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pp. 773–781. [2]

Blum, J. R. (1954), "Approximation Methods Which Converge With Probability One," *Annals of Mathematical Statistics*, 25, 382–386. [1]

Bottou, L. (1998), "Online Learning and Stochastic Approximations," in *Line Learning in Neural Networks*, ed. David Saad, Cambridge: Cambridge University Press, pp. 9–42. [1]

Chao, S.-K., and Cheng, G. (2019), "A Generalization of Regularized Dual Averaging and Its Dynamics," arXiv no. 1909.10072. [2]

Chen, X., Lee, J. D., Tong, X. T., and Zhang, Y. (2020), "Statistical Inference for Model Parameters in Stochastic Gradient Descent," *Annals of Statistics*, 48, 251–273. [2,3,5,6,7,11]

Duchi, J., Hazan, E., and Singer, Y. (2011), "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, 12, 2121–2159. [2]

Dvoretzky, A. (1956), "On Stochastic Approximation," in *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*. [1]

Fabian, V. (1968), "On Asymptotic Normality in Stochastic Approximation," *Annals of Mathematical Statistics*, 39, 1327–1332. [1]

Fang, Y. (2019), "Scalable Statistical Inference for Averaged Implicit Stochastic Gradient Descent," *Scandinavian Journal of Statistics*, 46, 987–1002. [2]

Fang, Y., Xu, J., and Yang, L. (2018), "Online Bootstrap Confidence Intervals for the Stochastic Gradient Descent Estimator," *Journal of Machine Learning Research*, 19, 3053–3073. [2]

Flegal, J. M., and Jones, G. L. (2010), "Batch Means and Spectral Variance Estimators in Markov Chain Monte Carlo," *Annals of Statistics*, 38, 1034–1070. [2]

Glynn, P. W., and Whitt, W. (1991), " Estimating the Asymptotic Variance With Batch Means," *Operation Research Letters*, 10, 431–435. [2]

Hazan, E., and Kale, S. (2014), "Beyond the Regret Minimization Barrier: Optimal Algorithms for Stochastic Strongly-Convex Optimization," *Journal of Machine Learning Research*, 15, 2489–2512. [2]

Hoffman, M., Bach, F. R., and Blei, D. M. (2010), "Online Learning for Latent Dirichlet Allocation," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pp. 451–459. [1]

Jones, G. L., Haran, M., Caffo, B. S., and Neath, R. (2006), "Fixed-Width Output Analysis for Markov Chain Monte Carlo," *Journal of American Statistical Association*, 101, 1537–1547. [8]

Kiefer, J., and Wolfowitz, J. (1952), "Stochastic Estimation of the Maximum of a Regression Function," *Annals Mathematical Statistics*, 23, 462–466. [1]

Kim, Y. M., Lahiri, S. N., and Nordman, D. J. (2013), "A Progressive Block Empirical Likelihood Method for Time Series," *Journal of American Statistical Association*, 108, 1506–1516. [6]

Kingma, D. P., and Ba, J. (2015), "Adam: A Method for Stochastic Optimization," in International Conference on Learning Representations. [2]

Kitamura, Y. et al. (1997), "Empirical Likelihood Methods With Weakly Dependent Processes," *Annals of Statistics*, 25, 2084–2102. [2]

Lahiri, S. N. (1999), "Theoretical Comparisons of Block Bootstrap Methods," *Annals of Statistics*, 27, 386–404. [4]

Lahiri, S. N. (2003), *Resampling Methods for Dependent Data*, Springer Series in Statistics, New York: Springer-Verlag. [2,4,5]

Lai, T. L. (2003), "Stochastic Approximation," *Annals of Statistics*, 31, 391–406. [1]

Liang, T., and Su, W. (2019), "Statistical Inference for the Population Landscape Via Moment-Adjusted Stochastic Gradients," *Journal of Royal Statistical Society*, Series B, 81, 431–456. [2]

Ljung, L. (1977), "Analysis of Recursive Stochastic Algorithms," *IEEE Transactions on Automatic Control*, 22, 551–575. [1]

Mairal, J., Bach, F. R., Ponce, J., and Sapiro, G. (2010), "Online Learning for Matrix Factorization and Sparse Coding," *Journal of Machine Learning Research*, 11, 19–60. [1]

McElroy, T., and Politis, D. N. (2007), "Computer-Intensive Rate Estimation, Diverging Statistics and Scanning," *Annals of Statistics*, 35, 1827–1848. [3]

Mou, W., Li, C. J., Wainwright, M. J., Bartlett, P. L., and Jordan, M. I. (2020), "On Linear Stochastic Approximation: Fine-Grained Polyak-Ruppert and Non-Asymptotic Concentration," in *Proceedings of Thirty Third Conference on Learning Theory*, pp. 2947–2997. [1]

Moulines, E., and Bach, F. R. (2011), "Non-Asymptotic Analysis of Stochastic Approximation Algorithms for Machine Learning," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, pp. 856–864. [2,7]

Nordman, D. J., Bunzel, H., and Lahiri, S. N. (2013), "A Nonstandard Empirical Likelihood for Time Series," *Annals of Statistics*, 41, 3050–3073. [3]

Politis, D. N., Romano, J. P., and Wolf, M. (1999), *Subsampling*, Springer Series in Statistics, New York: Springer-Verlag. [2]

Polyak, B. T., and Juditsky, A. B. (1992), "Acceleration of Stochastic Approximation by Averaging," *SIAM Journal of Control Optimization*, 30, 838–855. [1,2,3,6,7]

Rakhlin, A., Shamir, O., and Sridharan, K. (2012), "Making Gradient Descent Optimal for Strongly Convex Stochastic Optimization," in *Proceedings of the 29th International Conference on Machine Learning*, pp. 1571–1578. [2,7]

Richardson, M., Dominowska, E., and Ragno, R. (2007), "Predicting Clicks: Estimating the Click-Through Rate for New Ads," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 665–674. [3]

Robbins, H., and Monro, S. (1951), "A Stochastic Approximation Method," *Annals of Mathematical Statistics*, 22, 400–407. [1]

Robbins, H., and Siegmund, D. (1971), " A Convergence Theorem for Non Negative Almost Supermartingales and Some Applications," in *Optimizing Methods in Statistics*, eds. Jagdish S. Rustagi, Academic Press, pp. 233–257. [1]

Ruppert, D. (1988), "Efficient Estimations From a Slowly Convergent Robbins-Monro Process," Technical report, Cornell University Operations Research and Industrial Engineering. [1,7]

Sacks, J. (1958), "Asymptotic Distribution of Stochastic Approximation Procedures," *Annals of Mathematical Statistics*, 29, 373–405. [1]

Shamir, O., and Zhang, T. (2013), "Stochastic Gradient Descent for Non-Smooth Optimization: Convergence Results and Optimal Averaging Schemes," in *Proceedings of the 30th International Conference on Machine Learning*, pp. 71–79. [2]

Su, W., and Zhu, Y. (2018), "Uncertainty Quantification for Online Learning and Stochastic Approximation Via Hierarchical Incremental Gradient Descent," arXiv:1802.04876. [2]

Toulis, P., and Airoldi, E. M. (2017), "Asymptotic and Finite-Sample Properties of Estimators Based on Stochastic Gradients," *Annals of Statistics*, 45, 1694–1727. [2]

Vats, D., Flegal, J. M., and Jones, G. L. (2019), "Multivariate Output Analysis for Markov Chain Monte Carlo," *Biometrika*, 106, 321–337. [8]

Wu, W. B. (2009), "Recursive Estimation of Time-Average Variance Constants," *Annals of Applied Probability*, 19, 1529–1552. [3]

Zhang, W., Zhou, T., Wang, J., and Xu, J. (2016), "Bid-Aware Gradient Descent for Unbiased Learning With Censored Data in Display Advertising," in *Proceedings of the 16th International Conference on World Wide Web*, pp. 521–530. [3]