Fast Algorithms for Estimating Covariance Matrices of Stochastic Gradient Descent Solutions

Wei Biao Wu, joint with Zeqi Mao, Wanrong Zhu and Xi Chen

The University of Chicago

February 11, 2024, IPH

Overview

Model-parameter estimation

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 $F(x)$ from $\mathbb{R}^d$ to $\mathbb{R}$, i.e

$$x^* = \text{argmin}_{x \in \mathbb{R}^d} F(x) = \text{argmin}_{x \in \mathbb{R}^d} \mathbb{E}_{\xi \sim \Pi} f(x, \xi), \qquad (1)$$

where $f(x, \xi)$ is a loss function and ξ is a random variable following the distribution Π.

Example 1. Let $d = 1$ and $f(x, \xi) = |x - \xi|$ (resp. $|x - \xi|^2$). Then x^* is the median (resp. mean) of ξ.

Example 2. $\xi = (Z, Y)$, where Z is a d-dim vector and Y is a scalar and $f(x, \xi) = |Z^T x - Y|^2$.

Model-parameter estimation

- If the target function F is known, we can apply the Gradient descent algorithm

$$x_{n+1} = x_n - \gamma_n \nabla F(x_n), \tag{2}$$

where step size $\gamma_n \to 0$ and $\nabla F(x)$ is the gradient of F at x

- When F is not known, we can use the estimate

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} f(x, \xi_i) \tag{3}$$

based on the data $\xi_1, \xi_2, \ldots$. For example, empirical risk minimization, maximum-likelihood estimation, M-estimation, least squares estimation etc

Model-parameter estimation

- The standard (or "batch") gradient descent method

$$x_{n+1} = x_n - \gamma \nabla F_n(x_n) = x_n - \gamma \sum_{i=1}^{n} \nabla f(x_n, \xi_i)/n, \quad (4)$$

  where γ is the step size.
- doable if the function f has a simple structure
- can be very expensive to compute the sum-gradient when the training set is huge (stream/online/sequential data) and f has a complicated form
- One way out: Stochastic gradient descent by Robbins–Monro algorithm (1951), Siegmund, Lai, Yin, Kushner.....

How to deal with extremely large datasets?
How to process data on the fly?

Stochastic Gradient Decent (SGD)

Let $\{\xi_i\}_{i\geq 1}$ be a sequence of i.i.d sample from the distribution Π. Set x_0 as the initial point. The k-th iteration through SGD algorithm takes the following form

$$x_k = x_{k-1} - \eta_k \nabla f(x_{k-1}, \xi_k), \tag{5}$$

where η_k is the learning rate, the step size at k-th step.

SGD

- **Advantage**: Excellent computation and memory efficiency
- Very popular algorithm for model training in machine learning
- Coupled with backpropagation algorithm: standard algorithm for training artificial neural networks
- **Statistical Inference Problem**: How to address the Uncertainty?
  (SGD performs frequent updates with a high variability that causes the outcome fluctuate heavily.)

Averaged SGD: Acceleration by Averaging

- The Robbins-Monro algorithm can perform poorly in practice since it is sensitive to the choice of the learning rate sequence.
- Following Ruppert (1988), Polyak (1990), Polyak, Juditsky (1992), setting

$$\eta_k = \eta k^{-\alpha}, \ \eta > 0, \ \alpha \in (0.5, 1),$$

let the average

$$\bar{x}_n = n^{-1} \sum_{i=1}^{n} x_i \tag{6}$$

be the final estimator for x^*. The Averaged SGD (ASGD) is more robust to the choice of step sizes.

Averaged SGD: Acceleration by Averaging

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

$$\sqrt{n}(\bar{x}_n - x^*) \Rightarrow N(0, A^{-1}SA^{-1}), \tag{7}$$

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

Existing work

- **Convergence properties for** x_n: Well studied.
- **Statistical Inference/Uncertainty quantification?**
  There are few works:
  Chen et al. (2019); Fang et al. (2018); Su and Zhu (2023).
  (**Not on-line approach!)**

Motivation: efficient computation

In modern neural networks applications, the dimension d can be in millions. Want to reduce the dimensionality.
With confidence intervals of the estimated parameters, we can

- prune the unimportant connections
- learning only important connections
- simplify the network structure
- reduce the computation

Motivation: efficient computation

Han et al. (2015) *NIPS, Learning both Weights and Connections for Efficient Neural Network*):

- After an initial training phase, remove all connections whose weight is lower than a threshold. This pruning converts a dense, fully-connected layer to a sparse layer.
- reduce the storage and computation required by neural networks by an order of magnitude without affecting their accuracy by learning only the important connections.
- Network pruning has been used both to reduce network complexity and to reduce over-fitting.

Question:

How to obtain the **confidence intervals/regions** of the true model parameter

- in a fully online fashion?
- only through SGD iterates?

Our goal is to obtain an online estimate of the covariance matrix $A^{-1}SA^{-1}$ based only on the SGD iterates $x_1, \ldots, x_n, \ldots$.

With the above estimate, we can perform uncertainty quantification and statistical inference with excellent computation and memory efficiency.

Averaged SGD: Acceleration by Averaging

Following Ruppert (1988), Polyak (1990), Polyak, Juditsky (1992), consider the ASGD $\bar{x}_n = n^{-1} \sum_{i=1}^n x_i$.

Theorem. Polyak and Juditsky (1992). Let $\eta_k = \eta k^{-\alpha}$ with $\eta > 0$ and $\alpha \in (1/2, 1)$ and $\bar{x}_n = n^{-1} \sum_{i=1}^n x_i$. Then under suitable conditions we have the asymptotic normality of $\bar{x}_n$:

$$\sqrt{n}(\bar{x}_n - x^*) \Rightarrow N(0, A^{-1} S A^{-1}), \tag{8}$$

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

To leverage the CLT for inference, it is critical to estimate the asymptotic covariance matrix $\Sigma = A^{-1} S A^{-1}$!

Plug-in Estimate for $\Sigma = A^{-1}SA^{-1}$

- $\widehat{S}_n = n^{-1}\sum_{i=1}^{n}[\nabla f(x_{i-1}, \xi_i)][\nabla f(x_{i-1}, \xi_i)]^\top$
- $\widehat{A}_n = n^{-1}\sum_{i=1}^{n}\nabla^2 f(x_{i-1}, \xi_i)$
- The sandwich estimate $\widehat{\Sigma}_n = \widehat{A}_n^{-1}\widehat{S}_n\widehat{A}_n^{-1}$
- Potential problems: computation of the Hessian matrix of the loss function is not always available
- For quantile regression, the Hessian matrix does not even exist
- For legacy codes, only the SGD iterates are computed

Covariance matrix estimation: overview

- The sandwich estimate $\widehat{\Sigma}_n = \widehat{A}_n^{-1} \widehat{S}_n \widehat{A}_n^{-1}$
- Manipulations of $d \times d$ matrix: naive algorithm $O(d^3)$, Strassen $O(d^{2.8074})$, Coppersmith–Winograd $O(d^{2.3755})$.
- Our online algorithm, which is based only on the SGD iterates $x_1, x_2, \ldots$, only requires $O(d^2)$ updates, achieving desirable computation and memory efficiency.
- What happens if d is in millions?
- In the important special case of marginal inference of coordinates/entries of x^*, $O(d)$ computation suffices.

Covariance matrix estimation: overview

- In the important special case of marginal inference of coordinates/entries of x^*, $O(d)$ computation suffices.
- Mao, Zhu and Wu. Music Recognition using Mel Spectrogram: one input layer with dimension of 128, one hidden layer with dimension 128 and one output layer of dimension 1 with $d = 16384$. Need marginal inference.
- handwritten digital classification: 1M
- deepface: 120M

Stationary processes and Non-stationary Markov Chains

- Note that by (5), since ξ_k are i.i.d.,

$$x_k = x_{k-1} - \eta_k \nabla f(x_{k-1}, \xi_k) = m_k(x_{k-1}, \xi_k) \tag{9}$$

defines a non-homogeneous (non-stationary) Markov chain, since $\eta_k = \eta k^{-\alpha}$. Iterations of (9) lead to

$$x_k = g_k(\xi_k, \xi_{k-1}, \ldots, \xi_1, x_0). \tag{10}$$

- For a mean 0 stationary process

$$z_k = g(\xi_k, \xi_{k-1}, \ldots), \tag{11}$$

under suitable weak dependence conditions, we have the CLT

$$\sqrt{n}\bar{z}_n \Rightarrow N(0, \sigma_\infty^2), \text{ where } \sigma_\infty^2 = \sum_{k=-\infty}^{\infty} \operatorname{cov}(z_0, z_k) \tag{12}$$

where is the long run variance.

Long-run Variance Estimation for Stationary Processes

The batched mean estimate for the long-run variance σ_∞^2 is

$$\widehat{\sigma}_{n,l_n}^2 = \frac{1}{n-l_n}\sum_{i=1}^{n-l_n}\frac{(z_i + z_{i+1} + \ldots + z_{i+l_n-1} - l_n\bar{z}_n)^2}{l_n},$$

where l_n is the batch size.

Theorem (Liu and Wu). Assume that $l_n \to \infty$ and $l_n/n \to 0$.

- We have the consistency $\|\widehat{\sigma}_{n,l_n}^2 - \sigma_\infty^2\|_{q/2} \to 0$ if

$$\sum_{j=1}^{\infty}\delta_q(j) < \infty \text{ holds for some } q > 2.$$

- We have the CLT $\sqrt{n/l_n}(\widehat{\sigma}_{n,l_n}^2 - E\widehat{\sigma}_{n,l_n}^2) \Rightarrow N(0,\pi)$ if

$$\sum_{j=1}^{\infty}\delta_4(j) < \infty.$$

Long-run Variance Estimation

- The sample mean $\bar{z}_n = \sum_{i=1}^n z_i/n$ can be recursively updated:

$$\bar{z}_{n+1} = (n\bar{z}_n + z_{n+1})/(n+1)$$

Memory complexity is $O(1)$ and the computational complexity scales linearly in n.

- For the long-run variance estimate, assume $\mu = 0$:

$$\widehat{\sigma}^2_{n,l_n} = \frac{1}{l_n(n-l_n)} \sum_{i=1}^{n-l_n} (z_i + z_{i+1} + \ldots + z_{i+l_n-1})^2.$$

If $l_n \neq l_{n+1}$, one then has to update all the sums $z_i + \ldots + z_{i+l_n-1}$, $1 \leq i \leq n - l_n$. The memory complexity is $O(n)$ and the computational complexity $>> O(n)$

Online Long-run Variance Estimation

- In Markov Chain Monte Carlo, it is argued that $\bar{X}_n \pm 1.96 \times \widehat{\sigma}_{n,l_n}/\sqrt{n}$ can be used for convergence diagnostics for MCMC. The problem is:

  asymptotically 100% of one's computer time will be expended on computing the estimate of the σ^2_{n,l_n} (as opposed to simulating the trajectory of the process). This is clearly (very!) undesirable.

- Wu (2009) designed an online algorithm for computing estimates of σ^2_∞ for stationary processes
- Chan, K.W. and Yau, C.Y. (2016, 2017) made important improvements for the online algorithm.
- The same algorithm of Wu (2009) can be applied to estimate $\Sigma = A^{-1}SA^{-1}$ for outcomes of SGD, which form a non-stationary Markov Chain!

Online Long-run Variance Estimation

- In the batched mean variance estimate

$$\widehat{\sigma}^2_{n,l_n} = \frac{1}{n-l_n}\sum_{i=1}^{n-l_n}\frac{(z_i + z_{i+1} + \ldots + z_{i+l_n-1} - l_n\bar{z}_n)^2}{l_n},$$

  where l_n is the batch size which is same for all blocks $\{z_i, z_{i+1}, \ldots, z_{i+l_n-1}\}$.
- To account for dependence, $l_n \to \infty$, making $\widehat{\sigma}^2_{n,l_n}$ non recursive
- Wu's (2009) algorithm allows varying batch sizes

Online Long-run Variance Estimation

Assume at the outset that $\mu = 0$.

- $a_k = k^2, k \in \mathbb{N}$; $t_i = \lfloor \sqrt{i} \rfloor^2$. In general $a_k = \lfloor ck^p \rfloor, p > 1$.
- $V_n = \sum_{i=1}^n W_i^2$, where $W_i = X_{t_i} + X_{t_i+1} + \ldots + X_i$
- $v_n = \sum_{i=1}^n l_i$, where $l_i = i - t_i + 1$.
- Overlap batched estimate: $\widetilde{\sigma}_n^2 := V_n / v_n$

$$\begin{aligned} V_{17} &= X_1^2 + (X_1 + X_2)^2 + (X_1 + X_2 + X_3)^2 \\ &+ X_4^2 + (X_4 + X_5)^2 + \ldots + (X_4 + \ldots + X_8)^2 + \\ &+ X_9^2 + (X_9 + X_{10})^2 + \ldots + (X_9 + \ldots + X_{15})^2 \\ &+ X_{16}^2 + (X_{16} + X_{17})^2 = V_{16} + W_{17}^2; \\ v_{17} &= 1 + 2 + 3 \\ &+ 1 + 2 + 3 + 4 + 5 \\ &+ 1 + 2 + \ldots + 7 \\ &+ 1 + 2 = v_{16} + l_{17} \end{aligned}$$

- Key idea: $W_i = X_i$ if i is a square and $W_i = W_{i-1} + X_i$ if not

Non-overlap Online Long-run Variance Estimation

Assume at the outset that $\mu = 0$.

- $a_k = k^2, k \in \mathbb{N}$; $t_i = \lfloor \sqrt{i} \rfloor^2$, $i \in \mathbb{N}$
- $V_n = \sum_{i=1}^{n} W_i^2$, where $W_i = X_{t_i} + X_{t_i+1} + \ldots + X_i$
- $v_n = \sum_{i=1}^{n} l_i$, where $l_i = i - t_i + 1$.
- Non-Overlap batched estimate: $\widehat{\sigma}_n^2 := V_n^\sharp / v_n^\sharp$

$$\begin{aligned} V_{17}^\sharp &= (X_1 + X_2 + X_3)^2 + (X_4 + \ldots + X_8)^2 \\ &+ (X_9 + \ldots + X_{15})^2 + (X_{16} + X_{17})^2; \\ v_{17}^\sharp &= (2^2 - 1) + (3^2 - 2^2) + (4^2 - 3^2) + (17 - 4^2 + 1) \end{aligned}$$

Online Long-run Variance Estimation

Key observations:

- Both V_n and v_n can be recursively updated
- Length of block sums W_i are time-varying
- Convergence properties of the recursive estimate $\widetilde{\sigma}_n^2 := V_n/v_n$ can be developed by using the functional dependence measure (Wu, 2005):
  - (ε_i'): iid copy of (ε_i)
  - $\mathcal{F}_n = (\ldots, \varepsilon_{n-1}, \varepsilon_n)$
  - Coupling: $\mathcal{F}_n^* = (\mathcal{F}_{-1}, \varepsilon_0', \varepsilon_1, \ldots, \varepsilon_n)$
  - $\|X\|_p = [E(|X|^p)]^{1/p}$, $p \geq 1$
  - $\delta_p(n) = \|g(\mathcal{F}_n) - g(\mathcal{F}_n^*)\|_p$

Long-run Variance Estimation

Convergence of $\widetilde{\sigma}_n^2 = V_n/v_n$? Far from being trivial!

Theorem (Wu, 2009). Assume $X_i \in \mathcal{L}^q$, $q > 2$, $EX_i = 0$, and

$$\sum_{j=0}^{\infty} \delta_q(j) < \infty. \tag{13}$$

Then $\|\widetilde{\sigma}_n^2 - \sigma^2\|_{q/2} = [E|\widetilde{\sigma}_n^2 - \sigma^2|^{q/2}]^{2/q} = o(1)$.

Long-run Variance Estimation

Convergence of $\widetilde{\sigma}_n^2 = V_n/v_n$? Far from being trivial!

Theorem (Wu, 2009). Assume $X_i \in \mathcal{L}^q$, $q > 2$, $EX_i = 0$, and

$$\sum_{j=0}^{\infty} \delta_q(j) < \infty. \tag{13}$$

Then $\|\widetilde{\sigma}_n^2 - \sigma^2\|_{q/2} = [E|\widetilde{\sigma}_n^2 - \sigma^2|^{q/2}]^{2/q} = o(1)$.

Theorem (Wu (2009)). Assume that $X_i \in \mathcal{L}^4$, $EX_i = 0$, and

$$\sum_{j=0}^{\infty} j\delta_4(j) < \infty. \tag{14}$$

Let $a_k = \lfloor ck^p \rfloor$ with $p = 3/2$. Then the Mean Squares Error

$$\mathrm{MSE}(\widetilde{\sigma}_n^2) = \|\widetilde{\sigma}_n^2 - \sigma^2\|_2 := [E(\widetilde{\sigma}_n^2 - \sigma^2)^2]^{1/2} = O(n^{-1/3}).$$

Long-run Variance Estimation

Theorem. Assume (14). For the batched mean estimate

$$\widehat{\sigma}^2_{n,l_n} = \frac{1}{l_n(n-l_n)} \sum_{i=1}^{n-l_n} (X_i + X_{i+1} + \ldots + X_{i+l_n-1})^2,$$

let $\theta = 2\sum_{k=1}^{\infty} k\gamma(k)$. We have

$$\|\widehat{\sigma}^2_{n,l_n} - \sigma^2\|_2 = O(n^{-1/3}), \text{ where } l_n = \lfloor (\lambda_* n)^{1/3} \rfloor, \ \lambda_* = \frac{3\theta^2}{2\sigma^4}.$$

Chan, K.W. and Yau, C.Y. (2016, 2017) made important improvements for the online algorithm on high order correction and optimal batch size selection; see Chan and Yau (2017).

Long-run Variance Estimation

Theorem (Wu, 2009). Assume (14). Let $a_k = \lfloor ck^{3/2} \rfloor$ and choose c as $c = (4/3)^{3/2}\lambda_*^{1/2}$. Then

$$\frac{\|\widetilde{\sigma}_n^2 - \sigma^2\|_2}{\|\widehat{\sigma}_{n,l_n}^2 - \sigma^2\|_2} \to \frac{4}{3}$$

Under (14), the optimal p in $a_k = \lfloor ck^p \rfloor$ is $p = 3/2$.

Estimation of Asymptotic Covariance Matrix

Let $\{a_k\}_{k\in\mathbb{N}}$ be a strictly increasing integer-valued sequence with $a_1 = 1$.

We split SGD iterates $\{x_1, ..., x_n, ...\}$ into big batches based on $(a_k)_{k\in\mathbb{N}}$ as follows:

$$\{x_{a_1}, ..., x_{a_2-1}\}, \{x_{a_2}, ..., x_{a_3-1}\}, ..., \{x_{a_M}, ..., x_n, ...\}, ...$$

where M satisfies $a_M \leq n < a_{M+1}$.

Note: the introduction of (big) batches $\{x_{a_m}, ..., x_{a_{m+1}-1}, ...\}$ is only used for motivating our overlapping construction of small batches in following analysis.

Review: Batch means estimator

Batch-means estimator in Chen et al. (2019) is defined as

$$\sum_{m=1}^{M} \frac{n_m}{M} \left(\sum_{k=a_m}^{a_{m+1}-1} x_k/n_m - \bar{x}_n \right) \left(\sum_{k=a_m}^{a_{m+1}-1} x_k/n_m - \bar{x}_n \right)^T, \qquad (15)$$

based on the batch-means

$$\sum_{k=a_m}^{a_{m+1}-1} x_k/n_m - \bar{x}_n, \text{ for } 1 \le m \le M,$$

where $n_m = a_{m+1} - a_m$.

- Construction of batch-means estimator is based on big batch and can only be updated batch by batch.
- To ensure convergence, choice of $\{a_k\}_{k\in\mathbb{N}}$ in batch-means estimator depends on total number of steps n.

So the estimator is not recursive!

Modified overlapping batch means

To update the covariance estimate **step by step**, upon receiving a new data point x_i, we construct a new batch including previous data points from iterations t_i to i, i.e.,

$$\{x_{t_i}, ..., x_i\}.$$

Based on the small batch, we compute a new batch mean

$$\sum_{k=t_i}^{i} x_k/l_i - \bar{x}_n, \text{ for } 1 \leq i \leq n.$$

where $t_i = a_m$ when $i \in [a_m, a_{m+1})$ and $l_i = i - t_i + 1$.

The recursive estimator $\widehat{\Sigma}_n$ is then defined as

$$\widehat{\Sigma}_n = \sum_{i=1}^{n} \frac{l_i^2}{\sum_{i=1}^{n} l_i} \left(\sum_{k=t_i}^{i} x_k / l_i - \bar{x}_n \right) \left(\sum_{k=t_i}^{i} x_k / l_i - \bar{x}_n \right)^T . \quad (16)$$

Here, $\{a_k\}_{k \in \mathbb{N}}$is pre-defined, which means the construction **does not** depend on total number of steps!

Construction intuition: A new local variance estimation term, which can be viewed as the effect of the new data point x_i on the final variance estimator, is added to the final covariance estimate with a novel re-weighting step.

How to choose $\{a_k\}_{k\in\mathbb{N}}$?

Let $\delta_n = x_n - x^*$ and $\epsilon_n = \nabla F(x_{n-1}) - \nabla f(x_{n-1}, \xi_n)$. Then

$$\delta_n = \delta_{n-1} - \eta_n \nabla F(x_{n-1}) + \eta_n \epsilon_n.$$

With $\nabla F(x_{n-1})$ approximated by $A\delta_{n-1}$, for large n

$$\delta_n \approx (I - \eta_n A)\delta_{n-1} + \eta_n \epsilon_n. \tag{17}$$

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

$$\Pi_{k=j+1}^{i} \|I_d - \eta_k A\|_2 \leq (1 - \eta\lambda_A i^{-\alpha})^{i-j}, \tag{18}$$

when $\eta_k = \eta k^{-\alpha}$.

One can choose $i - j = Ki^{(\alpha+1)/2}$, where K is a large 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 infinite. Then a reasonable setting is that the sequence $\{a_k\}_{k\in\mathbb{N}}$ satisfies

$$a_k - a_{k-1} = Ka_k^{(\alpha+1)/2}. \tag{19}$$

Let a_k increase polynomially, i.e., $a_k = Ck^{\beta}$ for some constant C. Solve equation (19), we obtain that $\beta = 2/(1-\alpha)$. Thus a natural choice of a_k is

$$a_k = \lfloor Ck^{2/(1-\alpha)} \rfloor. \tag{20}$$

Recall in the stationary case the optimal $a_k = \lfloor ck^{3/2} \rfloor$, smaller than the one above.

Recursive Algorithm

Given sequentially arriving SGD iterates $x_1, ..., x_n, \ldots$, define

$$W_i = \sum_{k=t_i}^{i} x_k. \tag{21}$$

W_{i+1} can be updated recursively (e.g x_i in the m-th batch):

- When x_{i+1} is in the same batch as x_i, i.e $t_{i+1} = a_m$, then $W_{i+1} = W_i + x_{i+1}$.
- When x_{i+1} belongs to a new batch, i.e $t_{i+1} = a_{m+1}$, then $W_{i+1} = x_{i+1}$.
- the batch size $i - t_i + 1$ is time-varying

Then equation (16) can be expanded as

$$\begin{aligned}\widehat{\Sigma}_n = \left(\sum_{i=1}^{n} l_i\right)^{-1} \Bigg\{ & \sum_{i=1}^{n} W_i W_i^T + \sum_{i=1}^{n} l_i^2 \bar{x}_n \bar{x}_n^T \\ & - \left(\sum_{i=1}^{n} l_i W_i\right) \bar{x}_n^T - \bar{x}_n \left(\sum_{i=1}^{n} l_i W_i\right)^T \Bigg\}. \end{aligned} \tag{22}$$

To further simplify (16), we introduce

$$V_n = \sum_{i=1}^{n} W_i W_i^T, \; P_n = \sum_{i=1}^{n} l_i W_i, \; v_n = \sum_{i=1}^{n} l_i \text{ and } q_n = \sum_{i=1}^{n} l_i^2.$$

They can be updated recursively since both W_i and l_i can be updated recursively. Now, $\widehat{\Sigma}_n$ can be finally rewritten as

$$\widehat{\Sigma}_n = \frac{1}{v_n}(V_n + q_n \bar{x}_n \bar{x}_n^T - \bar{x}_n P_n^T - P_n \bar{x}_n^T). \tag{23}$$

All five terms in (23): $V_n, q_n, P_n, v_n, \bar{x}_n$ can be updated recursively. Thus we can update $\widehat{\Sigma}_n$ only through the results in the $(n-1)$-th step and the new iterate x_n at the n-th step.

Advantages:

- The estimate can be updated step by step (online fashion)
- Memory complexity is $O(d^2)$, which is independent of the sample size n.
- In the update step, the computational complexity is also $O(d^2)$. Then the total computational cost scales linearly in n.
- In the important special case of marginal inference of coordinates/entries of x^*, $O(d)$ computation suffices.

Convergence of the recursive estimator

Assumption 1: Strong convexity of the objective function $F(x)$ and Lipschitz continuity of its gradient.

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: Regularity and bound of the noisy gradient

Let error sequence $\delta_n = x_n - x^*$ and gradient difference sequence

$$\epsilon_n = \nabla F(x_{n-1}) - \nabla f(x_{n-1}, \xi_n).$$

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 $\mathbb{E}_{n-1} \nabla f(x_{n-1}, \xi_n) = \nabla F(x_{n-1})$, which implies that $\mathbb{E}_{n-1} \epsilon_n = 0$.

Assumption 2 (Continued)

2. The conditional covariance of ϵ_n has an expansion around S which satisfies the following:

$$\|\mathbb{E}_{n-1}(\epsilon_n \epsilon_n^T) - S\|_2 \leq C \left(\|\delta_{n-1}\|_2 + \|\delta_{n-1}\|_2^2 \right), \tag{24}$$

where C is some constant. Here S is the asymptotic covariance matrix for ASGD estimator.

3. There exists a constant C such that the fourth conditional moment of ϵ_n is bounded by

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

Convergence of $\widehat{\Sigma}_n$

Theorem. (Zhu et al. (2023)) Under Assumptions 1 and 2, let

$$a_k = \lfloor ck^{2/(1-\alpha)} \rfloor, \tag{25}$$

where c is a constant. Set step size at the i-th iteration as $\eta_i = \eta i^{-\alpha}$ with $\frac{1}{2} < \alpha < 1$. Then for $\widehat{\Sigma}_n$ defined in (16)

$$\mathbb{E}\|\widehat{\Sigma}_n - \Sigma\|_2 \lesssim M^{\frac{-\alpha}{2(1-\alpha)}} + M^{-\frac{1}{2}}, \tag{26}$$

where M is the number of batches such that $a_M \leq n < a_{M+1}$. Same bound holds for Non-overlap batch estimator.

Remark: Using the relationship between number of batches M and the total sample size n, we translate the above Theorem into the following results:

$$\mathbb{E}\|\widehat{\Sigma}_n - \Sigma\|_2 \lesssim n^{-\alpha/4} + n^{-(1-\alpha)/4} \asymp n^{-(1-\alpha)/4}. \tag{27}$$

We achieve the fastest possible rate $n^{-1/8}$ when α is close to $1/2$.

Remark: Using the relationship between number of batches M and the total sample size n, we translate the above Theorem into the following results:

$$\mathbb{E}\|\widehat{\Sigma}_n - \Sigma\|_2 \lesssim n^{-\alpha/4} + n^{-(1-\alpha)/4} \asymp n^{(\alpha-1)/4}. \tag{28}$$

We achieve the fastest possible rate $n^{-1/8}$ when α is close to $1/2$.

Convergence of $\widehat{\Sigma}_n$

Recent development. Wanrong Zhu and Wu (August, 2023) are making a substantial improvement by proposing a bias-corrected covariance matrix estimator $\widetilde{\Sigma}_n$ such that

$$\mathbb{E}\|\widetilde{\Sigma}_n - \Sigma\|_2^2 \lesssim n^{\alpha-1} \log n \tag{29}$$

Statistical inference

Statistical inference:

As n goes to infinity, for i-th coordinate of x^*

$$Pr(x_i^* \in \mathrm{CI}_{n,i}) \to 1 - q, \tag{30}$$

where

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

and $\widehat{\sigma}_{ii}$ is the i-th diagonal of $\widehat{\Sigma}_n$ defined in (16). We can also construct joint confidence region as follows:

$$Pr\left((\bar{x}_n - x^*)^T \widehat{\Sigma}_n^{-1} (\bar{x}_n - x^*) \leq \chi^2_{d,1-2/q}\right) \to 1 - q. \tag{31}$$

More generally, for any unit length vector $w \in \mathbb{R}^d$ (i.e., $\|w\|_2 = 1$), the following convergence result holds:

$$\frac{\sqrt{n} w^T (\bar{x}_n - x^*)}{\sqrt{w^T \widehat{\Sigma}_n w}} \Rightarrow N(0,1). \tag{32}$$

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

$$w^T \bar{x}_n \pm z_{1-q/2} \sqrt{w^T \widehat{\Sigma}_n w / n} \tag{33}$$

Simulation study

Linear and logistic regression

- Let $b_i = a_i^T x^* + \epsilon_i$, where $a_i \in \mathbb{R}^d \sim N(0, I_d)$, $\epsilon_i \sim N(0, 1)$. The loss function $f(\cdot)$ is defined as the negative log likelihood function, i.e.

$$f(x, a_i, b_i) = \frac{1}{2}(a_i^T x - b_i)^2.$$

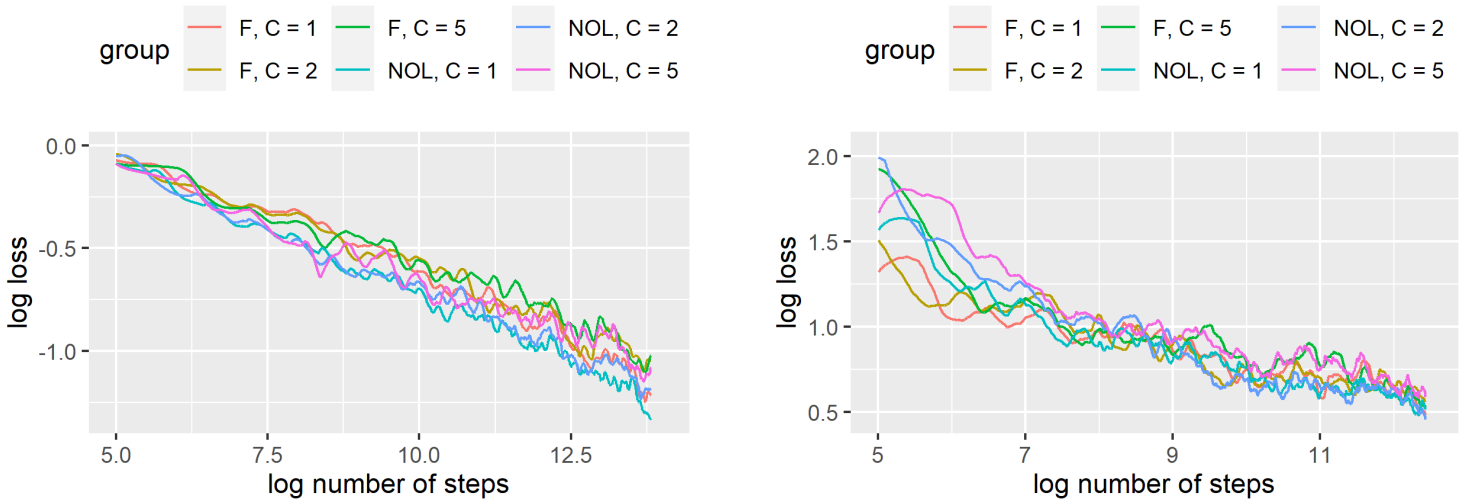
- Logistic regression: $b_i | a_i \sim Bernoulli((1 + \exp(-a_i^T x^*))^{-1})$:

$$f(x, a_i, b_i) = (1 - b_i) a_i^T x + \log(1 + \exp(-a_i^T x))^{-1}$$

Check:

- Convergence of recursive estimator
- CI coverage

Convergence of recursive estimator



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

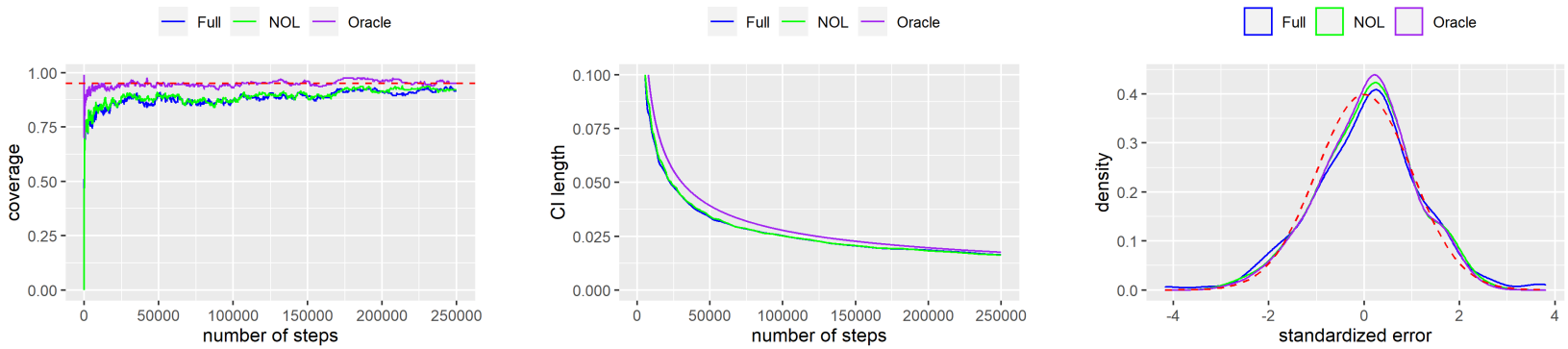
Figure 1: Linear regression: Log loss (operator norm) of estimated covariance matrix against the log of total number of steps. F denotes the full overlapping version (16), and NOL denotes the non-overlapping version. C denotes the constant in $a_m = \lfloor Cm^{2/(1-\alpha)} \rfloor$.

CI coverage

We construct 95% confidence interval for mean predictor $\mu = 1^T x^*$ based on (33) i.e.,

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

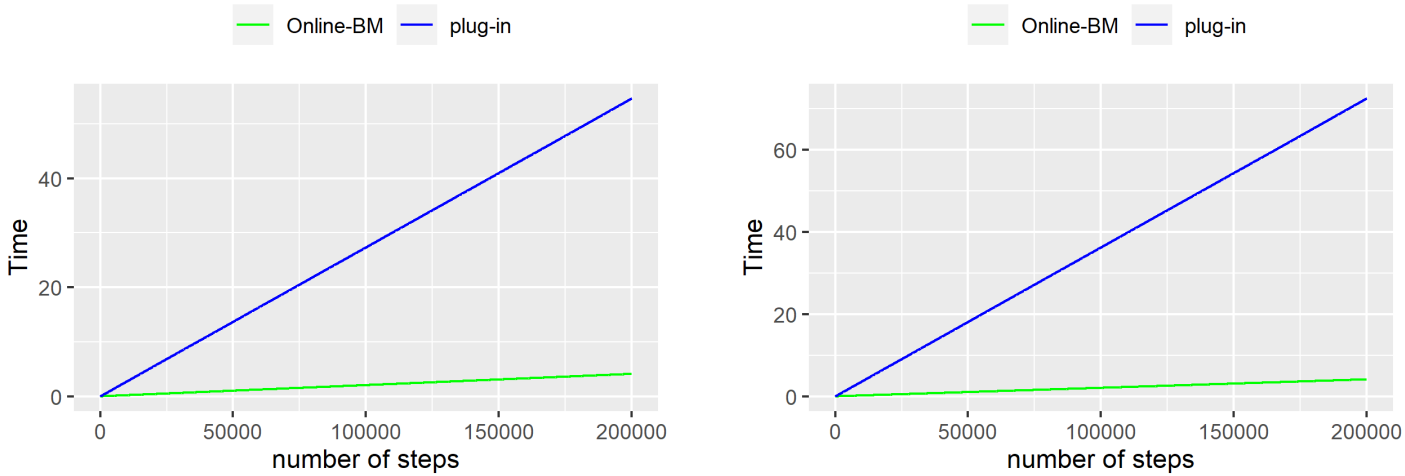
From Figure 2, the empirical coverage rate converges to 95% and the standardized error $\sqrt{n}1^T(\widehat{x} - x^*)/\sqrt{1^T \widehat{\Sigma}_n 1}$ is approximately standard normal.



(a) Empirical cover rate (b) CI length (c) Normality

Figure 2: Linear regression with $d = 5$: (a): Empirical coverage rate vs the number of steps. Red dashed line denotes the nominal coverage rate of 0.95. (b): Length of confidence intervals. (c): Density plot for standardized error. Red curve denotes density plot of $N(0, 1)$.

Computational Time

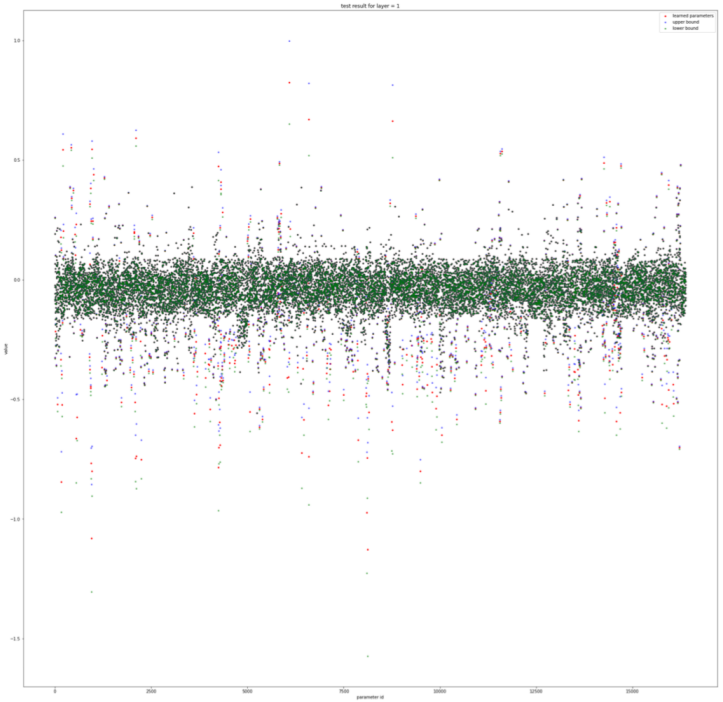


(a) $d = 5$ (b) $d = 20$

Figure 3: Comparison of online-BM and Plug-in estimators. Total computation time for updating covariance estimate and confidence intervals in SGD (same for both models).

Music recognition example

Mao, Zhu and Wu: Plot of $x_i/\widehat{\sigma}_i$ for the music recognition example with $d = 16384$:



Thank you!

Chan, K. W. and Yau, C. Y. (2017). Automatic optimal batch size selection for recursive estimators of time-average covariance matrix. *Journal of the American Statistical Association*, 112(519):1076–1089.

Chen, X., Lee, J. D., Tong, X. T., and Zhang, Y. (2019). Statistical inference for model parameters in stochastic gradient descent. *The Annals of Statistics, To appear*.

Fang, Y., Xu, J., and Yang, L. (2018). Online bootstrap confidence intervals for the stochastic gradient descent estimator. *The Journal of Machine Learning Research*, 19(1):3053–3073.

Han, S., Pool, J., Tran, J., and Dally, W. (2015). Learning both weights and connections for efficient neural network. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of

stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855.

Su, W. and Zhu, Y. (2023). Uncertainty quantification for online learning and stochastic approximation via hierarchical incremental gradient descent. *JMLR*.

Wu, W. B. (2009). Recursive estimation of time-average variance constants. *The Annals of Applied Probability*, 19(4):1529–1552.

Zhu, W., Chen, X., and Wu, W. B. (2023). Online covariance matrix estimation in stochastic gradient descent. *Journal of the American Statistical Association*, 0(ja):1–30.