HiGrad: Statistical Inference for Online Learning and Stochastic Approximation

Weijie J. Su
University of Pennsylvania

Princeton University, May 14, 2018
Collaborator

- Yuancheng Zhu (Renaissance Technologies)
Learning by optimization

Sample $Z_1, \ldots, Z_N$, and $f(\theta, z)$ is cost function

Learning model by minimizing

$$\arg\min_{\theta} \frac{1}{N} \sum_{n=1}^{N} f(\theta, Z_n)$$
Learning by optimization

Sample $Z_1, \ldots, Z_N$, and $f(\theta, z)$ is cost function

Learning model by minimizing

$$\argmin_{\theta} \frac{1}{N} \sum_{n=1}^{N} f(\theta, Z_n)$$

- Maximum likelihood estimation (MLE). More generally, $M$-estimation
- Often no closed-form solution
- Need optimization
Gradient descent

- Start at some $\theta_0$
- Iterate

$$
\theta_j = \theta_{j-1} - \gamma_j \sum_{n=1}^{N} \nabla f(\theta_{j-1}, Z_n) \frac{1}{N},
$$

where $\gamma_j$ are step sizes.
Gradient descent

- Start at some $\theta_0$
- Iterate

\[ \theta_j = \theta_{j-1} - \gamma_j \frac{\sum_{n=1}^{N} \nabla f(\theta_{j-1}, Z_n)}{N}, \]

where $\gamma_j$ are step sizes

However

- Offline algorithm
- Computational cost is high
Stochastic gradient descent (SGD)

Aka incremental gradient descent

- Start at some $\theta_0$
- Iterate

$$\theta_j = \theta_{j-1} - \gamma_j \nabla f(\theta_{j-1}, Z_j)$$
Stochastic gradient descent (SGD)

Aka incremental gradient descent
- Start at some $\theta_0$
- Iterate

$$\theta_j = \theta_{j-1} - \gamma_j \nabla f(\theta_{j-1}, Z_j)$$

SGD resolved these challenges
- Online in nature
Stochastic gradient descent (SGD)

Aka incremental gradient descent

- Start at some $\theta_0$
- Iterate

$$
\theta_j = \theta_{j-1} - \gamma_j \nabla f(\theta_{j-1}, Z_j)
$$

SGD resolved these challenges

- Online in nature
- One pass over data
Stochastic gradient descent (SGD)

Aka incremental gradient descent

- Start at some $\theta_0$
- Iterate

\[ \theta_j = \theta_{j-1} - \gamma_j \nabla f(\theta_{j-1}, Z_j) \]

SGD resolved these challenges

- Online in nature
- One pass over data
SGD in one line
Using SGD for prediction

**Averaged SGD**

An estimator of $\theta^* := \arg\min \mathbb{E} f(\theta, Z)$ is given by averaging

$$\bar{\theta} = \frac{1}{N} \sum_{j=1}^{N} \theta_j$$

Recall that $\theta_j = \theta_{j-1} - \gamma_j \nabla f(\theta_{j-1}, Z_j)$ for $j = 1, \ldots, N$. 

**Generalized linear models:**

- **Linear regression:** $\mu_x(\theta) = x'\theta$
- **Logistic regression:** $\mu_x(\theta) = \frac{e^{x'\theta}}{1+e^{x'\theta}}$
- **Generalized linear models:** $\mu_x(\theta) = \mathbb{E}_{\theta}(Y | X = x)$
Using SGD for prediction

Averaged SGD

An estimator of $\theta^* := \arg\min_{\theta} \mathbb{E} f(\theta, Z)$ is given by averaging

$$\bar{\theta} = \frac{1}{N} \sum_{j=1}^{N} \theta_j$$

Recall that $\theta_j = \theta_{j-1} - \gamma_j \nabla f(\theta_{j-1}, Z_j)$ for $j = 1, \ldots, N$.

Given a new instance $z = (x, y)$ with $y$ unknown

Interested in $\mu_x(\bar{\theta})$

- Linear regression: $\mu_x(\bar{\theta}) = x' \bar{\theta}$
- Logistic regression: $\mu_x(\bar{\theta}) = \frac{e^{x' \bar{\theta}}}{1 + e^{x' \bar{\theta}}}$
- Generalized linear models: $\mu_x(\bar{\theta}) = \mathbb{E}_{\bar{\theta}}(Y | X = x)$
How much can we trust SGD predictions?

We would observe a different $\mu_x(\bar{\theta})$ if

- Re-sample $Z'_1, \ldots, Z'_N$
- Sample with replacement $N$ times from a finite population
A real data example

*Adult* dataset on UCI repository\(^1\)

- 123 features
- \(Y = 1\) if an individual’s annual income exceeds $50,000
- 32,561 instances

Randomly pick 1,000 as a test set. Run SGD 500 times independently, each with 20 epochs and step sizes \(\gamma_j = 0.5j^{-0.55}\). Construct empirical confidence intervals with \(\alpha = 10\%\)

\(^1\)https://archive.ics.uci.edu/ml/datasets/Adult
High variability of SGD predictions
What is desired

Can we construct a confidence interval for $\mu_x^* := \mu_x(\theta^*)$?
What is desired

Can we construct a confidence interval for $\mu^*_x := \mu_x(\theta^*)$?

Remarks

• Bootstrap is computationally infeasible
• Most existing works concern bounding generalization errors or minimizing regrets (Shalev-Shwartz et al, 2011; Rakhlin et al, 2012)
• Chen et al (2016) proposed a batch-mean estimator of SGD covariance, and Fang et al (2017) proposed a perturbation-based resampling procedure
This talk: HiGrad

A new method: Hierarchical Incremental GRAdient Descent
This talk: HiGrad

A new method: Hierarchical Incremental GRAdient Descent

Properties of HiGrad

- Online in nature with same computational cost as vanilla SGD
This talk: HiGrad

A new method: Hierarchical Incremental GRAdient Descent

Properties of HiGrad
- Online in nature with same computational cost as vanilla SGD
- A confidence interval for $\mu_x^*$ in addition to an estimator
This talk: HiGrad

A new method: Hierarchical Incremental GRAdient Descent

Properties of HiGrad

- Online in nature with same computational cost as vanilla SGD
- A confidence interval for $\mu_x^*$ in addition to an estimator
- Estimator (almost) as accurate as vanilla SGD
The 90% HiGrad confidence interval for $\mu^*_{ji}$ is
\[
\left[ \mu_{ji} - t_{1-0.95} \cdot \sqrt{0.375}, \mu_{ji} + t_{1-0.95} \cdot \sqrt{0.375} \right],
\]
where $t_{1-0.95}$ is the critical value from the t-distribution with 1 degree of freedom.
• $\bar{\theta}_1 = \frac{1}{3} \bar{\theta}^0 + \frac{2}{3} \bar{\theta}^1$, $\bar{\theta}_2 = \frac{1}{3} \bar{\theta}^0 + \frac{2}{3} \bar{\theta}^2$
• $\bar{\theta}_1 = \frac{1}{3} \bar{\theta}^0 + \frac{2}{3} \bar{\theta}^1$,  
  $\bar{\theta}_2 = \frac{1}{3} \bar{\theta}^0 + \frac{2}{3} \bar{\theta}^2$
• $\overline{\theta}_1 = \frac{1}{3} \overline{\theta}^0 + \frac{2}{3} \overline{\theta}^1$, $\overline{\theta}_2 = \frac{1}{3} \overline{\theta}^0 + \frac{2}{3} \overline{\theta}^2$

• $\mu_x^1 := \mu_x(\overline{\theta}_1) = 0.15$, $\mu_x^2 := \mu_x(\overline{\theta}_2) = 0.11$
• $\bar{\theta}_1 = \frac{1}{3} \theta^0 + \frac{2}{3} \theta^1$, $\bar{\theta}_2 = \frac{1}{3} \theta^0 + \frac{2}{3} \theta^2$

• $\mu^1_x := \mu_x(\bar{\theta}_1) = 0.15$, $\mu^2_x := \mu_x(\bar{\theta}_2) = 0.11$

• HiGrad estimator is $\bar{\mu}_x = \frac{\mu^1_x + \mu^2_x}{2} = 0.13$
• $\bar{\theta}_1 = \frac{1}{3} \bar{\theta}^0 + \frac{2}{3} \bar{\theta}^1$, $\bar{\theta}_2 = \frac{1}{3} \bar{\theta}^0 + \frac{2}{3} \bar{\theta}^2$

• $\mu_x^1 := \mu_x(\bar{\theta}_1) = 0.15$, $\mu_x^2 := \mu_x(\bar{\theta}_2) = 0.11$

• HiGrad estimator is $\overline{\mu}_x = \frac{\mu_x^1 + \mu_x^2}{2} = 0.13$

• The 90% HiGrad confidence interval for $\mu_x$ is

\[
\overline{\mu}_x - t_{1,0.95} \sqrt{0.375} |\mu_x^1 - \mu_x^2|, \quad \overline{\mu}_x + t_{1,0.95} \sqrt{0.375} |\mu_x^1 - \mu_x^2|
\]

\[
= [-0.025, 0.285]
\]
Outline

1. Deriving HiGrad

2. Constructing Confidence Intervals

3. Empirical Performance
Problem statement

Minimizing convex $f$

$$\theta^* = \arg\min_{\theta} f(\theta) \equiv \mathbb{E} f(\theta, Z)$$

Observe i.i.d. $Z_1, \ldots, Z_N$ and can evaluate unbiased noisy gradient $g(\theta; Z)$

$$\mathbb{E} g(\theta, Z) = \nabla f(\theta) \text{ for all } \theta$$

To be fulfilled

- Online in nature with same computational cost as vanilla SGD
- A confidence interval for $\mu_x^*$ in addition to an estimator
- Estimator (almost) as accurate as vanilla SGD
The idea of contrasting and sharing

- Need more than one value $\mu_x$ to quantify variability: **contrasting**
The idea of contrasting and sharing

- Need more than one value $\mu_x$ to quantify variability: **contrasting**
- Need to share gradient information to elongate threads: **sharing**
The HiGrad tree

- $K + 1$ levels
- each $k$-level segment is of length $n_k$ and is split into $B_{k+1}$ segments

\[ n_0 + B_1n_1 + B_1B_2n_2 + B_1B_2B_3n_3 + \cdots + B_1B_2 \cdots B_Kn_K = N \]
The HiGrad tree

- $K + 1$ levels
- each $k$-level segment is of length $n_k$ and is split into $B_{k+1}$ segments

\[ n_0 + B_1n_1 + B_1B_2n_2 + B_1B_2B_3n_3 + \cdots + B_1B_2\cdots B_Kn_K = N \]

An example of HiGrad tree: $B_1 = 2, B_2 = 3, K = 2$
The HiGrad tree

- $K + 1$ levels
- each $k$-level segment is of length $n_k$ and is split into $B_{k+1}$ segments

\[ n_0 + B_1 n_1 + B_1 B_2 n_2 + B_1 B_2 B_3 n_3 + \cdots + B_1 B_2 \cdots B_K n_K = N \]

An example of HiGrad tree: $B_1 = 2$, $B_2 = 3$, $K = 2$
The HiGrad tree

- $K + 1$ levels
- each $k$-level segment is of length $n_k$ and is split into $B_{k+1}$ segments

$$n_0 + B_1n_1 + B_1B_2n_2 + B_1B_2B_3n_3 + \cdots + B_1B_2\cdots B_Kn_K = N$$

An example of HiGrad tree: $B_1 = 2, B_2 = 3, K = 2$
Iterate along HiGrad tree

Recall: noisy gradient $g(\theta, Z)$ unbiased for $\nabla f(\theta)$; partition $\{Z^s\}$ of $\{Z_1, \ldots, Z_N\}$; and $L_k := n_0 + \cdots + n_k$
Iterate along HiGrad tree

Recall: noisy gradient $g(\theta, Z)$ unbiased for $\nabla f(\theta)$; partition $\{Z^s\}$ of $\{Z_1, \ldots, Z_N\}$; and $L_k := n_0 + \cdots + n_k$

- Iterate along level 0 segment: $\theta_j = \theta_{j-1} - \gamma_j g(\theta_{j-1}, Z_j)$ for $j = 1, \ldots, n_0$, starting from some $\theta_0$
Iterate along HiGrad tree

Recall: noisy gradient $g(\theta, Z)$ unbiased for $\nabla f(\theta)$; partition $\{Z^s\}$ of $\{Z_1, \ldots, Z_N\}$; and $L_k := n_0 + \cdots + n_k$

- Iterate along level 0 segment: $\theta_j = \theta_{j-1} - \gamma_j g(\theta_{j-1}, Z_j)$ for $j = 1, \ldots, n_0$, starting from some $\theta_0$

- Iterate along each level 1 segment $s = (b_1)$ for $1 \leq b_1 \leq B_1$

$$\theta^s_j = \theta^s_{j-1} - \gamma_j L_0 g(\theta^s_{j-1}, Z^s_j)$$

for $j = 1, \ldots, n_1$, starting from $\theta_{n_0}$
Iterate along HiGrad tree

Recall: noisy gradient $g(\theta, Z)$ unbiased for $\nabla f(\theta)$; partition $\{Z^s\}$ of $\{Z_1, \ldots, Z_N\}$; and $L_k := n_0 + \cdots + n_k$

- Iterate along level 0 segment: $\theta_j = \theta_{j-1} - \gamma_j g(\theta_{j-1}, Z_j)$ for $j = 1, \ldots, n_0$, starting from some $\theta_0$

- Iterate along each level 1 segment $s = (b_1)$ for $1 \leq b_1 \leq B_1$

\[
\theta^s_j = \theta^s_{j-1} - \gamma_j + L_0 g(\theta^s_{j-1}, Z^s_j)
\]

for $j = 1, \ldots, n_1$, starting from $\theta_{n_0}$

- Generally, for the segment $s = (b_1 \cdots b_k)$, iterate

\[
\theta^s_j = \theta^s_{j-1} - \gamma_j + L_{k-1} g(\theta^s_{j-1}, Z^s_j)
\]

for $j = 1, \ldots, n_k$, starting from $\theta^{(b_1 \cdots b_{k-1})}_{n_{k-1}}$
A second look at the HiGrad tree

An example of HiGrad tree: $B_1 = 2, B_2 = 3, K = 2$
A second look at the HiGrad tree

An example of HiGrad tree: $B_1 = 2$, $B_2 = 3$, $K = 2$

Fulfilled

- Online in nature with same computational cost as vanilla SGD
A second look at the HiGrad tree

An example of HiGrad tree: $B_1 = 2$, $B_2 = 3$, $K = 2$

**Fulfilled**
- Online in nature with same computational cost as vanilla SGD

**Bonus**
Easier to parallelize than vanilla SGD!
1. Deriving HiGrad

2. Constructing Confidence Intervals

3. Empirical Performance
Estimate $\mu^*_x$ through each thread

Average over each segment $s = (b_1, \ldots, b_k)$

$$\bar{\theta}^s = \frac{1}{n_k} \sum_{j=1}^{n_k} \theta_j^s$$

Given weights $w_0, w_1, \ldots, w_K$ that sum up to 1, weighted average along thread $t = (b_1, \ldots, b_K)$ is

$$\bar{\theta}_t = \sum_{k=0}^{K} w_k \bar{\theta}^{(b_1, \ldots, b_k)}$$
Estimate $\mu^*_x$ through each thread

Average over each segment $s = (b_1, \ldots, b_k)$

$$\bar{\theta}^s = \frac{1}{n_k} \sum_{j=1}^{n_k} \theta_j^s$$

Given weights $w_0, w_1, \ldots, w_K$ that sum up to 1, weighted average along thread $t = (b_1, \ldots, b_K)$ is

$$\bar{\theta}_t = \sum_{k=0}^{K} w_k \bar{\theta}^{(b_1, \ldots, b_k)}$$

Estimator yielded by thread $t$

$$\mu^t_x := \mu_x(\bar{\theta}_t)$$
How to construct a confidence interval based on 
$T := B_1 B_2 \cdots B_K$ many such $\mu^t_x$ estimates?
Assume normality

Denote by \( \mu_x \) the \( T \)-dimensional vector consisting of all \( \mu_x^t \)

**Normality of \( \mu_x \) (to be shown later)**

\[
\sqrt{N}(\mu_x - \mu_x^*1) \text{ converges weakly to normal distribution } \mathcal{N}(0, \Sigma) \text{ as } N \to \infty
\]
Convert to simple linear regression

From $\mu_x \sim \mathcal{N}(\mu_x^* 1, \Sigma/N)$ we get

$$\Sigma^{-\frac{1}{2}} \mu_x \approx (\Sigma^{-\frac{1}{2}} 1) \mu_x^* + \tilde{z}, \quad \tilde{z} \sim \mathcal{N}(0, I/N)$$
From $\mu_x \sim N(\mu_x^* 1, \Sigma/N)$ we get

$$\Sigma^{-\frac{1}{2}} \mu_x \approx (\Sigma^{-\frac{1}{2}} 1) \mu_x^* + \tilde{z}, \quad \tilde{z} \sim N(0, I/N)$$

Simple linear regression! Least-squares estimator of $\mu_x^*$ given as

$$\begin{align*}
(1' \Sigma^{-\frac{1}{2}} \Sigma^{-\frac{1}{2}} 1)^{-1} 1' \Sigma^{-\frac{1}{2}} \Sigma^{-\frac{1}{2}} \mu_x \\
= (1' \Sigma^{-1} 1)^{-1} 1' \Sigma^{-1} \mu_x \\
= \frac{1}{T} \sum_{t \in \mathcal{T}} \mu_x^t \equiv \bar{\mu}_x
\end{align*}$$

HiGrad estimator

Just the sample mean $\bar{\mu}_x$
A $t$-based confidence interval

A pivot for $\mu^*_x$

$$\frac{\mu_x - \mu^*_x}{SE_x} \sim t_{T-1},$$

where the standard error is given as

$$SE_x = \sqrt{\frac{(\mu'_x - \mu_x 1') \Sigma^{-1} (\mu_x - \mu_x 1)}{T - 1}} \cdot \frac{\sqrt{1' \Sigma 1}}{T}$$
A $t$-based confidence interval

A pivot for $\mu_x^*$

$$\frac{\bar{\mu}_x - \mu_x^*}{SE_x} \sim t_{T-1},$$

where the standard error is given as

$$SE_x = \sqrt{\frac{(\mu'_x - \mu_{x1}')\Sigma^{-1}(\mu_x - \mu_{x1})}{T - 1}} \cdot \frac{\sqrt{1'\Sigma1}}{T}$$

HiGrad confidence interval of coverage $1 - \alpha$

$$[\bar{\mu}_x - t_{T-1,1-\frac{\alpha}{2}} SE_x, \quad \bar{\mu}_x + t_{T-1,1-\frac{\alpha}{2}} SE_x]$$
Do we know the covariance $\Sigma$?
An extension of Ruppert–Polyak normality

Given a thread \( t = (b_1, \ldots, b_K) \), denote by segments \( s_k = (b_1, b_2, \ldots, b_k) \)

Fact (informal)

\[
\sqrt{n_0}(\bar{\theta}^{s_0} - \theta^*) , \sqrt{n_1}(\bar{\theta}^{s_1} - \theta^*) , \ldots , \sqrt{n_K}(\bar{\theta}^{s_K} - \theta^*)
\]

converge to i.i.d. centered normal distributions.
An extension of Ruppert–Polyak normality

Given a thread $t = (b_1, \ldots, b_K)$, denote by segments $s_k = (b_1, b_2, \ldots, b_k)$

Fact (informal)

$\sqrt{n_0}(\bar{\theta}^{s_0} - \theta^*)$, $\sqrt{n_1}(\bar{\theta}^{s_1} - \theta^*)$, $\ldots$, $\sqrt{n_K}(\bar{\theta}^{s_K} - \theta^*)$ converge to i.i.d. centered normal distributions

- Hessian $H = \nabla^2 f(\theta^*)$ and $V = \mathbb{E} [g(\theta^*, Z)g(\theta^*, Z)']$. Ruppert (1988), Polyak (1990), and Polyak and Juditsky (1992) prove

\[
\sqrt{N}(\bar{\theta}_N - \theta^*) \Rightarrow \mathcal{N}(0, H^{-1}V H^{-1})
\]
An extension of Ruppert–Polyak normality

Given a thread $t = (b_1, \ldots, b_K)$, denote by segments $s_k = (b_1, b_2, \ldots, b_k)$

**Fact (informal)**

$\sqrt{n_0}(\theta^{s_0} - \theta^*)$, $\sqrt{n_1}(\theta^{s_1} - \theta^*)$, $\ldots$, $\sqrt{n_K}(\theta^{s_K} - \theta^*)$ converge to i.i.d. centered normal distributions

- Hessian $H = \nabla^2 f(\theta^*)$ and $V = \mathbb{E} [g(\theta^*, Z)g(\theta^*, Z)']$. Ruppert (1988), Polyak (1990), and Polyak and Juditsky (1992) prove

  $$\sqrt{N}(\theta_N - \theta^*) \Rightarrow N(0, H^{-1}VH^{-1})$$

- Difficult to estimate sandwich covariance $H^{-1}VH^{-1}$ (Chen et al, 2016)
An extension of Ruppert–Polyak normality

Given a thread $t = (b_1, \ldots, b_K)$, denote by segments $s_k = (b_1, b_2, \ldots, b_k)$

Fact (informal)

$\sqrt{n_0}(\bar{\theta}^{s_0} - \theta^*)$, $\sqrt{n_1}(\bar{\theta}^{s_1} - \theta^*)$, \ldots, $\sqrt{n_K}(\bar{\theta}^{s_K} - \theta^*)$ converge to i.i.d. centered normal distributions

- Hessian $H = \nabla^2 f(\theta^*)$ and $V = \mathbb{E}[g(\theta^*, Z)g(\theta^*, Z)']$. Ruppert (1988), Polyak (1990), and Polyak and Juditsky (1992) prove

  $$\sqrt{N}(\bar{\theta}_N - \theta^*) \Rightarrow \mathcal{N}(0, H^{-1}VH^{-1})$$

- Difficult to estimate sandwich covariance $H^{-1}VH^{-1}$ (Chen et al, 2016)

- To know covariance of $\{\mu_x(\bar{\theta}_t)\}$, really need to know $H^{-1}VH^{-1}$?
Lemma

For any two threads $t$ and $t'$ that agree at the first $k$ segments and differ henceforth, we have

$$\text{Cov} \left( \mu_x^t, \mu_x^{t'} \right) = (1 + o(1))\sigma^2 \sum_{i=0}^{k} \frac{w_i^2}{n_i}$$
Specify $\Sigma$ up to a multiplicative factor

$$\Sigma_{t,t'} = (1 + o(1))C \sum_{i=0}^{k} \frac{\omega_i^2 N}{n_i}$$
Specify $\Sigma$ up to a multiplicative factor

$$\Sigma_{t,t'} = (1 + o(1))C \sum_{i=0}^{k} \frac{\omega_i^2 \nu_i}{n_i}$$

- Do we need to know $C$ as well?
Specify $\Sigma$ up to a multiplicative factor

$$
\Sigma_{t,t'} = (1 + o(1))C \sum_{i=0}^{k} \frac{\omega_i^2 N}{n_i}
$$

- Do we need to know $C$ as well?
- No! Standard error of $\mu_x$ invariant under multiplying $\Sigma$ by a scalar

$$
SE_x = \sqrt{\frac{(\mu'_x - \mu_x 1')\Sigma^{-1}(\mu_x - \mu_x 1)}{T - 1} \cdot \frac{\sqrt{1'\Sigma 1}}{T}}
$$
Formal statement of theoretical results
Assumptions

1. **Local strong convexity.** $f(\theta) \equiv \mathbb{E} f(\theta, Z)$ convex, differentiable, with Lipschitz gradients. Hessian $\nabla^2 f(\theta)$ locally Lipschitz and positive-definite at $\theta^*$

2. **Noise regularity.** $V(\theta) = \mathbb{E} [g(\theta, Z)g(\theta, Z)']$ Lipschitz and does not grow too fast. Noisy gradient $g(\theta, Z)$ has $2 + o(1)$ moment locally at $\theta^*$
Examples satisfying assumptions

- **Linear regression**: \( f(\theta, z) = \frac{1}{2} (y - x^\top \theta)^2. \)
- **Logistic regression**: \( f(\theta, z) = -yx^\top \theta + \log \left( 1 + e^{x^\top \theta} \right). \)
- **Penalized regression**: Add a ridge penalty \( \lambda \| \theta \|^2. \)
- **Huber regression**: \( f(\theta, z) = \rho_\lambda(y - x^\top \theta), \) where \( \rho_\lambda(a) = a^2/2 \) for \( |a| \leq \lambda \) and \( \rho_\lambda(a) = \lambda|a| - \lambda^2/2 \) otherwise.

**Sufficient conditions**

\( X \) in *generic* position, and \( \mathbb{E}\|X\|^{4+o(1)} < \infty \) and \( \mathbb{E}|Y|^{2+o(1)}\|X\|^{2+o(1)} < \infty \)
Main theoretical results

Theorem (S. and Zhu)

Assume $K$ and $B_1, \ldots, B_K$ are fixed, $n_k \propto N$ as $N \to \infty$, and $\mu_x$ has a nonzero derivative at $\theta^*$. Taking $\gamma_j \asymp j^{-\alpha}$ for $\alpha \in (0.5, 1)$ gives

$$\frac{\mu_x - \mu^*_x}{SE_x} \Rightarrow t_{T-1}$$
Main theoretical results

Theorem (S. and Zhu)

Assume $K$ and $B_1, \ldots, B_K$ are fixed, $n_k \propto N$ as $N \to \infty$, and $\mu_x$ has a nonzero derivative at $\theta^*$. Taking $\gamma_j \asymp j^{-\alpha}$ for $\alpha \in (0.5,1)$ gives

$$\frac{\bar{\mu}_x - \mu^*_x}{\text{SE}_x} \xrightarrow{\text{dist}} t_{T-1}$$

Confidence intervals

$$\lim_{N \to \infty} \mathbb{P} \left( \mu^*_x \in \left[ \bar{\mu}_x - t_{T-1,1-\frac{\alpha}{2}} \text{SE}_x, \quad \bar{\mu}_x + t_{T-1,1-\frac{\alpha}{2}} \text{SE}_x \right] \right) = 1 - \alpha$$
Main theoretical results

Theorem (S. and Zhu)

Assume $K$ and $B_1, \ldots, B_K$ are fixed, $n_k \propto N$ as $N \to \infty$, and $\mu_x$ has a nonzero derivative at $\theta^*$. Taking $\gamma_j \approx j^{-\alpha}$ for $\alpha \in (0.5, 1)$ gives

$$\frac{\mu_{x} - \mu^*_x}{SE_x} \imp t_{T-1}$$

Confidence intervals

$$\lim_{N \to \infty} P \left( \mu^*_x \in \left[ \bar{\mu}_x - t_{T-1,1-\frac{\alpha}{2}} SE_x, \quad \bar{\mu}_x + t_{T-1,1-\frac{\alpha}{2}} SE_x \right] \right) = 1 - \alpha$$

Fulfilled

- Online in nature with same computational cost as vanilla SGD
- A confidence interval for $\mu^*_x$ in addition to an estimator
How accurate is the HiGrad estimator?
Optimal variance with optimal weights

By Cauchy–Schwarz

\[
N \text{Var}(\bar{\mu}_x) = (1 + o(1))\sigma^2 \left[ \sum_{k=0}^{K} n_k \prod_{i=1}^{k} B_i \right] \left[ \sum_{k=0}^{K} \frac{w_k^2}{n_k \prod_{i=1}^{k} B_i} \right]
\geq (1 + o(1))\sigma^2 \left[ \sum_{k=0}^{K} \sqrt{w_k^2} \right]^2 = (1 + o(1))\sigma^2,
\]

with equality if

\[
w_k^* = \frac{n_k \prod_{i=1}^{k} B_i}{N}
\]
Optimal variance with optimal weights

By Cauchy–Schwarz

\[ N \text{Var}(\bar{\mu}_x) = (1 + o(1))\sigma^2 \left[ \sum_{k=0}^{K} n_k \prod_{i=1}^{k} B_i \right] \left[ \sum_{k=0}^{K} \frac{w_k^2}{n_k \prod_{i=1}^{k} B_i} \right] \]

\[ \geq (1 + o(1))\sigma^2 \left[ \sum_{k=0}^{K} \sqrt{w_k^2} \right]^2 = (1 + o(1))\sigma^2, \]

with equality if

\[ w_k^* = \frac{n_k \prod_{i=1}^{k} B_i}{N} \]

- Segments at an early level are weighted less
Optimal variance with optimal weights

By Cauchy–Schwarz

\[ N \text{Var}(\bar{\mu}_x) = (1 + o(1))\sigma^2 \left[ \sum_{k=0}^{K} n_k \prod_{i=1}^{k} B_i \right] \left[ \sum_{k=0}^{K} \frac{w_k^2}{n_k \prod_{i=1}^{k} B_i} \right] \]

\[ \geq (1 + o(1))\sigma^2 \left[ \sum_{k=0}^{K} \sqrt{w_k^2} \right]^2 = (1 + o(1))\sigma^2, \]

with equality if

\[ w_k^* = \frac{n_k \prod_{i=1}^{k} B_i}{N} \]

- Segments at an early level are weighted less
- HiGrad estimator has the same asymptotic variance as vanilla SGD
Optimal variance with optimal weights

By Cauchy–Schwarz

\[ N \, \text{Var}(\bar{\mu}_x) = (1 + o(1))\sigma^2 \left[ \sum_{k=0}^{K} n_k \prod_{i=1}^{k} B_i \right] \left[ \sum_{k=0}^{K} \frac{w_k^2 n_k \prod_{i=1}^{k} B_i}{n_k \prod_{i=1}^{k} B_i} \right] \]

\[ \geq (1 + o(1))\sigma^2 \left[ \sum_{k=0}^{K} \sqrt{w_k^2} \right]^2 = (1 + o(1))\sigma^2, \]

with equality if

\[ w_k^* = \frac{n_k \prod_{i=1}^{k} B_i}{N} \]

- Segments at an early level are weighted less
- HiGrad estimator has the same asymptotic variance as vanilla SGD
- Achieves Cramér–Rao lower bound when model specified
Prediction intervals for vanilla SGD

Theorem (S. and Zhu)

Run vanilla SGD on a fresh dataset of the same size, producing $\mu^{\text{SGD}}_x$. Then, with optimal weights,

$$\lim_{N \to \infty} \mathbb{P} \left( \mu^{\text{SGD}}_x \in \left[ \bar{\mu}_x - \sqrt{2t_{T-1,1-\frac{\alpha}{2}}} \text{SE}_x, \quad \bar{\mu}_x + \sqrt{2t_{T-1,1-\frac{\alpha}{2}}} \text{SE}_x \right] \right) = 1 - \alpha.$$ 

- $\mu^{\text{SGD}}_x$ can be replaced by the HiGrad estimator with the same structure
- Interpretable even under model misspecification
Three properties

Under certain assumptions, for example, $f$ being locally strongly convex

Fulfilled

- Online in nature with same computational cost as vanilla SGD
- A confidence interval for $\mu_x^*$ in addition to an estimator
- Estimator (almost) as accurate as vanilla SGD
Outline

1. Deriving HiGrad

2. Constructing Confidence Intervals

3. Empirical Performance
General simulation setup

$X$ generated as i.i.d. $\mathcal{N}(0, 1)$ and $Z = (X, Y) \in \mathbb{R}^d \times \mathbb{R}$. Set $N = 10^6$ and use $\gamma_j = 0.5j^{-0.55}$

- Linear regression $Y \sim \mathcal{N}(\mu_X(\theta^*), 1)$, where $\mu_x(\theta) = x'\theta$
- Logistic regression $Y \sim \text{Bernoulli}(\mu_X(\theta^*))$, where

$$
\mu_x(\theta) = \frac{e^{x'\theta}}{1 + e^{x'\theta}}
$$

Criteria

- Accuracy: $\|\overline{\theta} - \theta^*\|^2$, where $\overline{\theta}$ averaged over $T$ threads
- Coverage probability and length of confidence interval
Accuracy

Dimension $d = 50$. MSE $\|\bar{\theta} - \theta^*\|^2$ normalized by that of vanilla SGD

- **null case** where $\theta_1 = \cdots = \theta_{50} = 0$
- **dense case** where $\theta_1 = \cdots = \theta_{50} = \frac{1}{\sqrt{50}}$
- **sparse case** where $\theta_1 = \cdots = \theta_{5} = \frac{1}{\sqrt{5}}, \theta_{6} = \cdots = \theta_{50} = 0$
Accuracy

Linear regression, null

Linear regression, sparse

Linear regression, dense

Logistic regression, null

Logistic regression, sparse

Logistic regression, dense
Coverage and CI length

HiGrad configurations

- $K = 1$, then $n_1/n_0 = r = 1$;
- $K = 2$, then $n_1/n_0 = n_2/n_1 = r \in \{0.75, 1, 1.25, 1.5\}$

Set $\theta_i^* = (i - 1)/d$ for $i = 1, \ldots, d$ and $\alpha = 5\%$. Use measure

$$\frac{1}{20} \sum_{i=1}^{20} 1(\mu_{x_i}(\theta^*) \in \text{Cl}_{x_i})$$
### Linear regression: $d = 20$

<table>
<thead>
<tr>
<th>$d$</th>
<th>0.956</th>
<th>0.938</th>
<th>0.9185</th>
<th>0.887</th>
<th>0.8488</th>
<th>0.9425</th>
<th>0.9472</th>
<th>0.9452</th>
<th>0.9448</th>
<th>0.924</th>
<th>0.9318</th>
<th>0.935</th>
<th>0.9378</th>
<th>0.925</th>
<th>0.9185</th>
<th>0.9245</th>
<th>0.9348</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y$</td>
<td>1, 4, 1</td>
<td>1, 8, 1</td>
<td>1, 12, 1</td>
<td>1, 16, 1</td>
<td>1, 20, 1</td>
<td>2, 2, 1</td>
<td>2, 2, 1.25</td>
<td>2, 2, 1.5</td>
<td>2, 2, 2</td>
<td>3, 2, 1</td>
<td>3, 2, 1.25</td>
<td>3, 2, 1.5</td>
<td>3, 2, 2</td>
<td>2, 3, 1</td>
<td>2, 3, 1.25</td>
<td>2, 3, 1.5</td>
<td>2, 3, 2</td>
</tr>
<tr>
<td>$w$</td>
<td>0.0621</td>
<td>0.0618</td>
<td>0.0606</td>
<td>0.0605</td>
<td>0.0633</td>
<td>0.062</td>
<td>0.0614</td>
<td>0.0612</td>
<td>0.0615</td>
<td>0.061</td>
<td>0.0614</td>
<td>0.062</td>
<td>0.0633</td>
<td>0.0605</td>
<td>0.0606</td>
<td>0.0618</td>
<td>0.0621</td>
</tr>
</tbody>
</table>
## Linear regression: $d = 100$

<table>
<thead>
<tr>
<th>Value</th>
<th>1, 4, 1</th>
<th>1, 8, 1</th>
<th>1, 12, 1</th>
<th>1, 16, 1</th>
<th>1, 20, 1</th>
<th>2, 2, 1</th>
<th>2, 2, 1.25</th>
<th>2, 2, 1.5</th>
<th>2, 2, 2</th>
<th>3, 2, 1</th>
<th>3, 2, 1.25</th>
<th>3, 2, 1.5</th>
<th>3, 2, 2</th>
<th>2, 3, 1</th>
<th>2, 3, 1.25</th>
<th>2, 3, 1.5</th>
<th>2, 3, 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.15</td>
<td>0.2403</td>
<td>0.2197</td>
<td>0.2312</td>
<td>0.2495</td>
<td>0.2649</td>
<td>0.1917</td>
<td>0.1927</td>
<td>0.1946</td>
<td>0.1972</td>
<td>0.1412</td>
<td>0.1428</td>
<td>0.1453</td>
<td>0.1489</td>
<td>0.1457</td>
<td>0.1466</td>
<td>0.1491</td>
<td>0.15</td>
</tr>
<tr>
<td>0.894</td>
<td>0.8992</td>
<td>0.897</td>
<td>0.9115</td>
<td>0.9125</td>
<td>0.9108</td>
<td>0.9302</td>
<td>0.9312</td>
<td>0.9318</td>
<td>0.9324</td>
<td>0.934</td>
<td>0.9345</td>
<td>0.9351</td>
<td>0.9362</td>
<td>0.9368</td>
<td>0.9374</td>
<td>0.9379</td>
<td>0.9383</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9338</td>
<td>0.9358</td>
<td>0.9302</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td>0.9232</td>
<td>0.9248</td>
<td>0.9253</td>
<td>0.9261</td>
</tr>
<tr>
<td>0.92</td>
<td>0.9308</td>
<td>0.92</td>
<td>0.9125</td>
<td>0.9312</td>
<td>0.9338</td>
<td>0.9358</td>
<td>0.9302</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
</tr>
<tr>
<td>0.9472</td>
<td>0.9478</td>
<td>0.9308</td>
<td>0.92</td>
<td>0.9125</td>
<td>0.9312</td>
<td>0.9338</td>
<td>0.9358</td>
<td>0.9302</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
</tr>
<tr>
<td>0.9478</td>
<td>0.9308</td>
<td>0.92</td>
<td>0.9125</td>
<td>0.9312</td>
<td>0.9338</td>
<td>0.9358</td>
<td>0.9302</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
</tr>
<tr>
<td>0.9308</td>
<td>0.92</td>
<td>0.9125</td>
<td>0.9312</td>
<td>0.9338</td>
<td>0.9358</td>
<td>0.9302</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td></td>
</tr>
<tr>
<td>0.92</td>
<td>0.9125</td>
<td>0.9312</td>
<td>0.9338</td>
<td>0.9358</td>
<td>0.9302</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9125</td>
<td>0.9312</td>
<td>0.9338</td>
<td>0.9358</td>
<td>0.9302</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9312</td>
<td>0.9338</td>
<td>0.9358</td>
<td>0.9302</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9338</td>
<td>0.9358</td>
<td>0.9302</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9358</td>
<td>0.9302</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9302</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8992</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.897</td>
<td>0.9115</td>
<td>0.9125</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9115</td>
<td>0.9125</td>
<td>0.917</td>
<td>0.894</td>
<td>0.9065</td>
<td>0.9148</td>
<td>0.9162</td>
<td>0.9175</td>
<td>0.919</td>
<td>0.9205</td>
<td>0.9211</td>
<td>0.9222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The table shows various values and their corresponding indices, possibly indicating a pattern or relationship in the context of linear regression with $d = 100$.
A real data example: setup

From the 1994 census data based on UCI repository. $Y$ indicates if an individual’s annual income exceeds $50,000$

- 123 features
- 32,561 instances
- Randomly pick 1,000 as a test set

Use $N = 10^6$, $\alpha = 10\%$, and $\gamma_j = 0.5j^{-0.55}$. Run HiGrad for $L = 500$ times. Use measure

$$\text{coverage}_i = \frac{1}{L(L-1)} \sum_{\ell_1} \sum_{\ell_2 \neq \ell_1} 1(\hat{p}_{i\ell_1} \in \text{PI}_{i\ell_2})$$
A real data example: histogram

![Histogram of Coverage Probability]
## Comparisons of HiGrad configurations

<table>
<thead>
<tr>
<th>Configurations</th>
<th>Accuracy</th>
<th>Coverage</th>
<th>CI length</th>
</tr>
</thead>
<tbody>
<tr>
<td>________________</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>________________</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>________________</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>________________</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>________________</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>________________</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>________________</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
<tr>
<td>________________</td>
<td>★★★★★</td>
<td>★★★★★</td>
<td>★★★★★</td>
</tr>
</tbody>
</table>

Note: The stars represent varying levels of comparison metrics.
Default HiGrad parameters

HiGrad R package default values

\[ K = 2, B_1 = 2, B_2 = 2, n_0 = n_1 = n_2 = \frac{N}{7} \]
Concluding Remarks
Possible extensions

Improving statistical properties
  ▶ Finite-sample guarantee
    ◆ Better coverage probability

Extend Ruppert-Polyak to high dimensions

A new template for online learning
  ▶ Non-convex problems
    Online PCA, stochastic EM, etc

A criterion for early stopping
  Detect overfitting through contrasting
  Need to incorporate selective inference

Any ideas? Happy to talk offline
Possible extensions

Improving statistical properties

- Finite-sample guarantee
  - Better coverage probability

- Extend Ruppert-Polyak to high dimensions
  - Number of unknown variables growing
Possible extensions

Improving statistical properties
  ▶ Finite-sample guarantee
    ● Better coverage probability
  ▶ Extend Ruppert-Polyak to high dimensions
    ● Number of unknown variables growing

A new template for online learning
  ▶ Non-convex problems
    ● Online PCA, stochastic EM, etc
Possible extensions

Improving statistical properties
► Finite-sample guarantee
  ◦ Better coverage probability
► Extend Ruppert-Polyak to high dimensions
  ◦ Number of unknown variables growing

A new template for online learning
► Non-convex problems
  ◦ Online PCA, stochastic EM, etc
► A criterion for early stopping
  ◦ Detect overfitting through contrasting
  ◦ Need to incorporate selective inference
► Any ideas? Happy to talk offline
Take-home messages

Idea
Contrasting and sharing through hierarchical splitting
Take-home messages

Idea
Contrasting and sharing through hierarchical splitting

Properties (local strong convexity)
- Online in nature with same computational cost as vanilla SGD
- A confidence interval for $\mu_x^*$ in addition to an estimator
- Estimator (almost) as accurate as vanilla SGD
Take-home messages

Idea
Contrasting and sharing through hierarchical splitting

Properties (local strong convexity)
- Online in nature with same computational cost as vanilla SGD
- A confidence interval for $\mu_x^*$ in addition to an estimator
- Estimator (almost) as accurate as vanilla SGD

Bonus
Easier to parallelize than vanilla SGD!
Thanks!


- **Software.** R package higrad, available on CRAN

- **Webpage.** [http://stat.wharton.upenn.edu/~suw/higrad](http://stat.wharton.upenn.edu/~suw/higrad)

- **Acknowledgement.** NSF via grant CCF-1763314