On Momentum Methods and Acceleration in Stochastic Optimization

Praneeth Netrapalli

MSR India

Based on joint work with

Prateek Jain, Sham M. Kakade, Rahul Kidambi and Aaron Sidford

MSR India | Univ. of Washington, Seattle | Stanford University

Overview

- Optimization is a big part of large scale machine learning
- Stochastic gradient descent (SGD) is the workhorse

- Three key aspects
  - Minibatching (parallelization)
  - Model averaging (distributed over multiple machines)
  - Acceleration (speeding up convergence)

- Empirically widely used but little understanding

Overview

- Optimization is a big part of large scale machine learning
- Stochastic gradient descent (SGD) is the workhorse

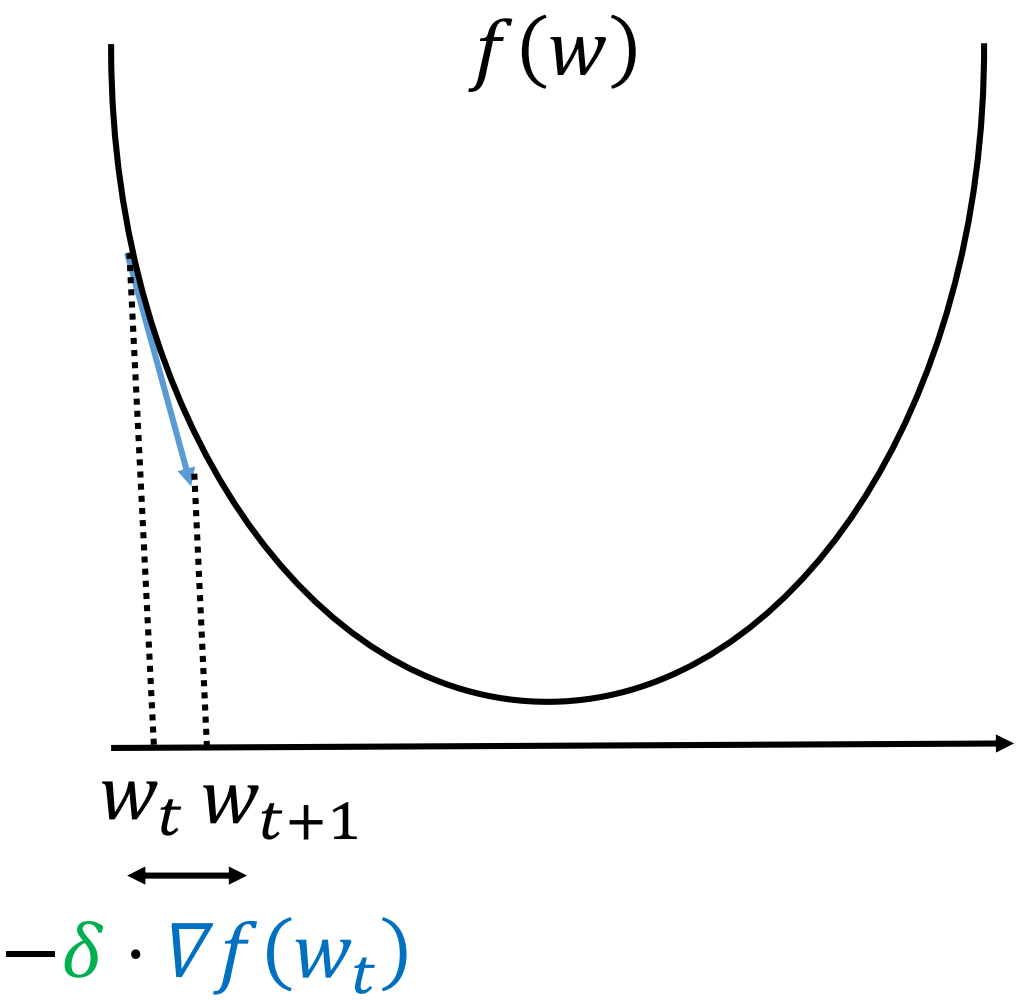
Our work

Thorough investigation of these three aspects for linear regression

- Three key aspects
  - Minibatching (parallelization) [accepted to JMLR]
  - Model averaging (distributed over multiple machines) [accepted to JMLR]
  - Acceleration (speeding up convergence) [COLT 2018 & ICLR 2018]

- Empirically widely used but little understanding

Gradient descent (GD) (Cauchy 1847)



Problem:

$$\min_w f(w)$$

Gradient descent:

$$w_{t+1} = w_t - \delta \cdot \nabla f(w_t)$$

Stepsize

Gradient

Linear regression

$$f(w) = \left\| \boxed{X^\top} \boxed{w} - \boxed{Y} \right\|_2^2$$

- Basic problem and arises pervasively in applications
- Deeply studied in literature

Gradient descent for linear regression

$$w_{t+1} = w_t - \delta \cdot \underbrace{X(X^\top w_t - b)}_{\text{Gradient}}$$

- $f^* = \min_w f(w)$; Condition number: $\kappa \stackrel{\text{def}}{=} \frac{\sigma_{max}(XX^\top)}{\sigma_{min}(XX^\top)}$
- Find w such that $f(w) - f^* \leq \epsilon$
- Convergence rate: $\Theta\left(\kappa \log \frac{f(w_0) - f^*}{\epsilon}\right)$ iterations

Question: Is it possible to do better?

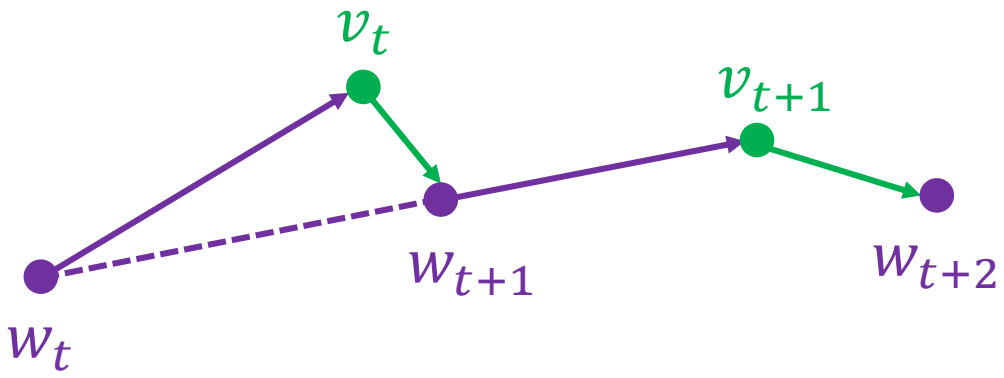
- Hope: GD does not reuse past gradients
- Answer: Yes!

Gradient descent:

$$w_{t+1} = w_t - \delta \cdot \nabla f(w_t)$$

- Conjugate gradient (Hestenes and Stiefel 1952)
- Heavy ball method (Polyak 1964)
- Accelerated gradient (Nemirovsky and Yudin 1977, Nesterov 1983)
- Also known as *momentum* methods

Nesterov's accelerated gradient (NAG)



Gradient steps
Momentum steps

- $f^* = \min_w f(w)$; Condition number: $\kappa \stackrel{\text{def}}{=} \frac{\sigma_{max}(XX^\top)}{\sigma_{min}(XX^\top)}$
- Find w such that $f(w) - f^* \leq \epsilon$
- Convergence rate: $\Theta\left(\sqrt{\kappa}\log\frac{f(w_0)-f^*}{\epsilon}\right)$ iterations

Nesterov's accelerated gradient (NAG)

$$w_{t+1} = v_t - \delta \nabla f(v_t)$$
$$v_{t+1} = w_{t+1} + \gamma(w_{t+1} - w_t)$$

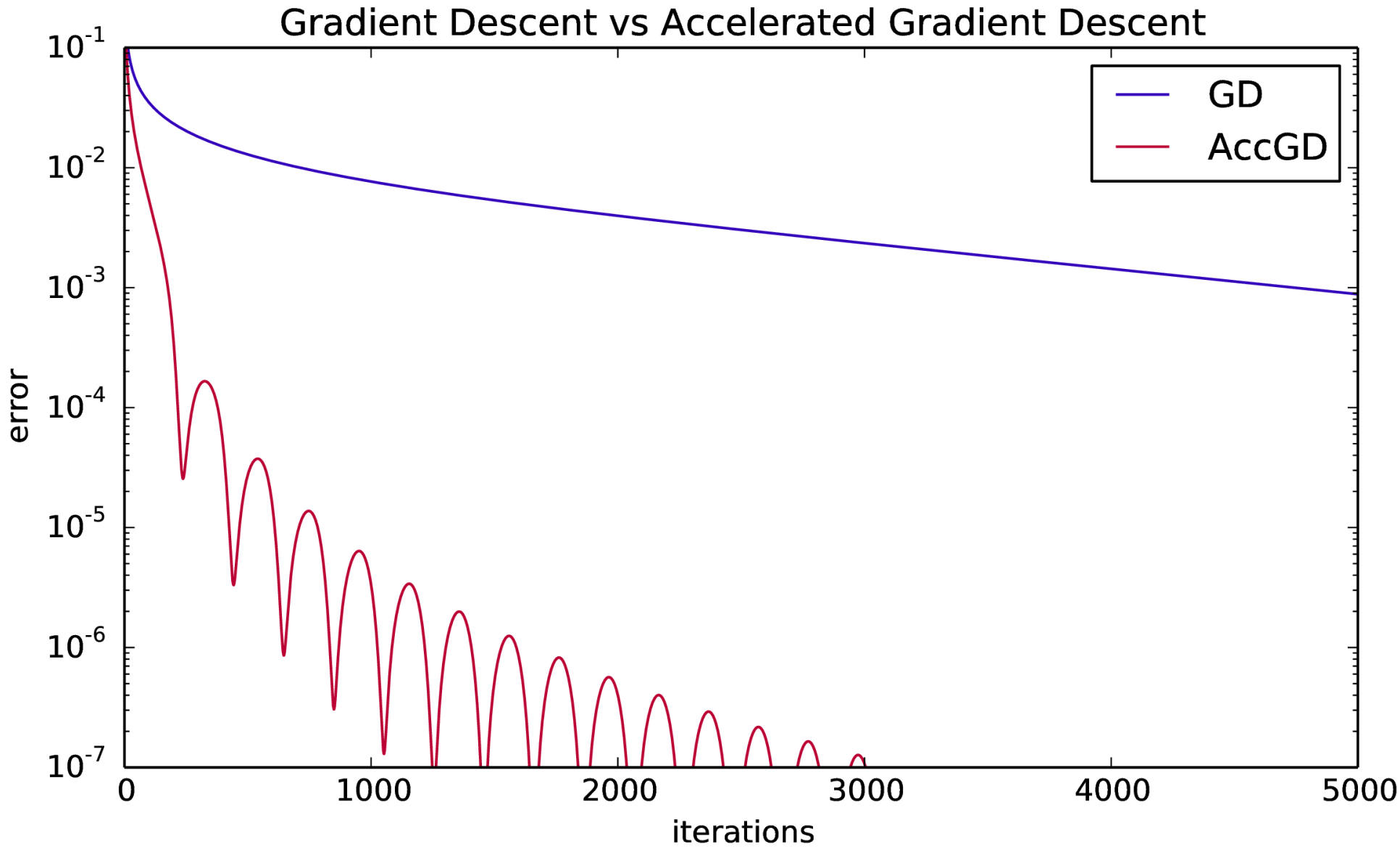
Gradient steps
Momentum steps

- $f^* = \min_w f(w)$; Condition number: $\kappa \stackrel{\text{def}}{=} \frac{\sigma_{max}(XX^\top)}{\sigma_{min}(XX^\top)}$
- Find w such that $f(w) - f^* \leq \epsilon$
- Convergence rate: $\Theta\left(\sqrt{\kappa} \log \frac{f(w_0) - f^*}{\epsilon}\right)$ iterations

Nesterov's accelerated gradient (NAG)

Compared to: $O\left(\kappa \log \frac{f(w_0)-f^*}{\epsilon}\right)$ for GD

- $f^* = \min_w f(w)$; Condition number: $\kappa \stackrel{\text{def}}{=} \frac{\sigma_{max}(XX^\top)}{\sigma_{min}(XX^\top)}$
- Find w such that $f(w) - f^* \leq \epsilon$
- Convergence rate: $\Theta\left(\sqrt{\kappa} \log \frac{f(w_0)-f^*}{\epsilon}\right)$ iterations



Source: http://blog.mrtz.org/2014/08/18/robustness-versus-acceleration.html

Optimization in machine learning

- Observe n samples $(x_1, y_1), \cdots, (x_n, y_n) \sim \mathcal{D}(\mathbb{R}^d \times \mathbb{R})$

$$\min_w \hat{f}(w) \triangleq \frac{1}{n} \sum_i \left(x_i^\top w - y_i\right)^2$$

- Ultimate goal: $\min_w f(w) \triangleq \mathbb{E}_{(x,y)\sim\mathcal{D}}[(x^\top w - y)^2]$
- GD not applicable: cannot compute exact gradients
- Well studied: “stochastic approximation”

Stochastic algorithms (Robbins & Monro 1951)

- $\nabla f(w_t) \to \hat{\nabla} f(w_t)$; $\hat{\nabla} f(w_t)$ computed just using (x_t, y_t)
- $\mathbb{E}\left[\hat{\nabla} f(w_t)\right] = \nabla f(w_t)$
- Return $\frac{1}{n}\sum_i w_i$ (Polyak and Juditsky 1992)

Examples

Gradient descent	⟶	Stochastic GD
Heavy ball	⟶	Stochastic HB
Nesterov	⟶	Stochastic NAG

- For linear regression, SGD: $w_{t+1} = w_t - \delta \cdot \left(x_t^\top w_t - y_t\right) x_t$
- **Streaming algorithm**: extremely efficient and widely used in practice

Convergence rate of SGD

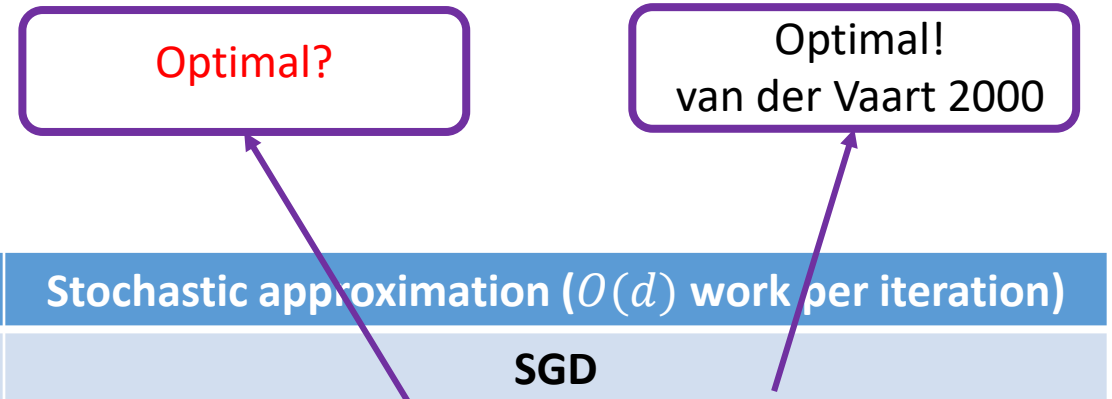
- Consider special case: $y = x^\top w^*$ i.e., no noise
- Convergence rate: $\tilde{\Theta}\left(\kappa \log \frac{f(w_0)-f^*}{\epsilon}\right)$ iterations
- $f^* = \min_w f(w)$; $\epsilon =$ Target suboptimality
- Condition number: $\kappa \triangleq \frac{\max\|x\|_2^2}{\sigma_{min}(\mathbb{E}[xx^\top])}$
- Noisy case: $y = x^\top w^* + \mathrm{n}$. Additive term based on $\sigma^2 \stackrel{\text{def}}{=} \mathbb{E}[\mathrm{n}^2]$

State of the art (# iterations)

Deterministic case ($O(nd)$ work per iteration)	Stochastic approximation ($O(d)$ work per iteration)
GD $\Theta\left(\kappa \log \frac{f(w_0) - f^*}{\epsilon}\right)$	**SGD** $\widetilde{\Theta}\left(\kappa \log \frac{f(w_0) - f^*}{\epsilon} + \frac{\sigma^2 d}{\epsilon}\right)$
NAG $\Theta\left(\sqrt{\kappa} \log \frac{f(w_0) - f^*}{\epsilon}\right)$	**Accelerated SGD?** Unknown

Main question: Is accelerating SGD possible?

State of the art (# iterations)



Deterministic case ($O(nd)$ work per iteration)	Stochastic approximation ($O(d)$ work per iteration)
GD $\Theta\left(\kappa \log \frac{f(w_0) - f^*}{\epsilon}\right)$	**SGD** $\widetilde{\Theta}\left(\kappa \log \frac{f(w_0) - f^*}{\epsilon} + \frac{\sigma^2 d}{\epsilon}\right)$
NAG $\Theta\left(\sqrt{\kappa} \log \frac{f(w_0) - f^*}{\epsilon}\right)$	**Accelerated SGD?** Unknown

Main question: Is accelerating SGD possible?

Is this really important?



- Extremely important in practice
  - ➢As we saw, acceleration can give orders of magnitude improvement
  - ➢Almost all deep learning packages use SGD+momentum
  - ➢Our earlier work shows acceleration leads to more parallelizability

- No understanding of stochastic HB/NAG in ML setting
  - Studied non-ML settings by [d'Aspremont 2008, Ghadimi & Lan 2010]

Outline of our results

1. Is it always possible to improve κ to $\sqrt{\kappa}$?
   - ➢ No
2. Is improvement ever possible?
   - ➢ Perhaps, depends on other problem parameters
3. Do existing algorithms (stochastic HB/NAG) achieve this improvement?
   - ➢ No, in fact they are no better than SGD
4. Can we design an algorithm improving over SGD?
   - ➢ Yes, we design such an algorithm (ASGD)

Question 1: Is acceleration always possible?

- $y = x^\top w^*$ (Noiseless)
- SGD convergence rate: $\widetilde{\Theta}\left(\kappa \log \frac{f(w_0)-f^*}{\epsilon}\right)$
- Accelerated rate: $\tilde{O}\left(\sqrt{\kappa} \log \frac{f(w_0)-f^*}{\epsilon}\right)$?

Example I: Discrete distribution

$$x = \begin{cases} \binom{1}{0} \text{ w.p. } 0.9999, \\ \binom{0}{1} \text{ w.p. } 0.0001 \end{cases}$$

- In this case, $\kappa \triangleq \frac{\max\|x\|_2^2}{\sigma_{min}(\mathbb{E}[xx^\top])} = \frac{1}{0.0001} = 10^4$
- Is $\tilde{O}\left(\sqrt{\kappa}\log\frac{f(w_0)-f^*}{\epsilon}\right)$ possible?
- Or even, halve the error using $\sim \sqrt{\kappa} = 100$ samples?

Example I: Discrete distribution

$$x = \begin{cases} \binom{1}{0} \text{ w.p. } 0.9999 \\ \binom{0}{1} \text{ w.p. } 0.0001; \end{cases} \qquad \kappa = 10^4$$

- Fewer than κ samples $\Rightarrow$ do not observe $\binom{0}{1}$ direction
  $\Rightarrow$ cannot estimate w_1^* at all

- Cannot do better than $O(\kappa)$

- Acceleration not possible for this distribution

Example I: Discrete distribution

$$x = \begin{cases} \binom{1}{0} \text{ w.p. } 0.9999 \\ \binom{0}{1} \text{ w.p. } 0.0001; \end{cases} \qquad \kappa = 10^4$$

- Fewer than κ samples $\Rightarrow$ do not observe $\binom{0}{1}$ direction
  $\Rightarrow$ cannot estimate w_1^* at all
- Cannot do better than $O(\kappa)$
- Acceleration not possible for this distribution

Question 1

Is it always possible to improve κ to $\sqrt{\kappa}$?

Answer: No

Example II: Gaussian

- $x \sim \mathcal{N}(0, H)$, $H = \begin{bmatrix} 0.9999 & 0 \\ 0 & 0.0001 \end{bmatrix}$
- In this case, $\kappa \sim \frac{\mathrm{Tr}(H)}{\sigma_{\min}(H)} = 10^4 \gg 2$

- However, after 2 samples: $\frac{1}{n}\sum_i x_i x_i^\top$ invertible
  - That is, possible to find $w^* = \left(\sum_i x_i x_i^T\right)^{-1} \sum_i x_i y_i$

- Possible to find w^* after 2 samples!

- Acceleration *might be* possible for this distribution

Example II: Gaussian

- $x \sim \mathcal{N}(0, H)$, $H = \begin{bmatrix} 0.9999 & 0 \\ 0 & 0.0001 \end{bmatrix}$
- In this case, $\kappa \sim \frac{\mathrm{Tr}(H)}{\sigma_{\min}(H)} = 10^4 \gg 2$

Question 2

Is acceleration ever possible?

Answer: Perhaps, but depends on other problem parameters

- However, after 2 samples: $\frac{1}{n}\sum_i x_i x_i^\top$ invertible
  - That is, possible to find $w^* = \left(\sum_i x_i x_i^T\right)^{-1} \sum_i x_i y_i$
- Possible to find w^* after 2 samples!
- Acceleration *might be* possible for this distribution

Matrix spectral concentration

Recall: $x_i \sim \mathcal{D}$. Let $H \stackrel{\text{def}}{=} \mathbb{E}\left[x_i x_i^\top\right]$

Need $\left\| \frac{1}{n} \sum_{i=1}^{n} x_i x_i^\top - H \right\|$ small

How many samples (n) are required?

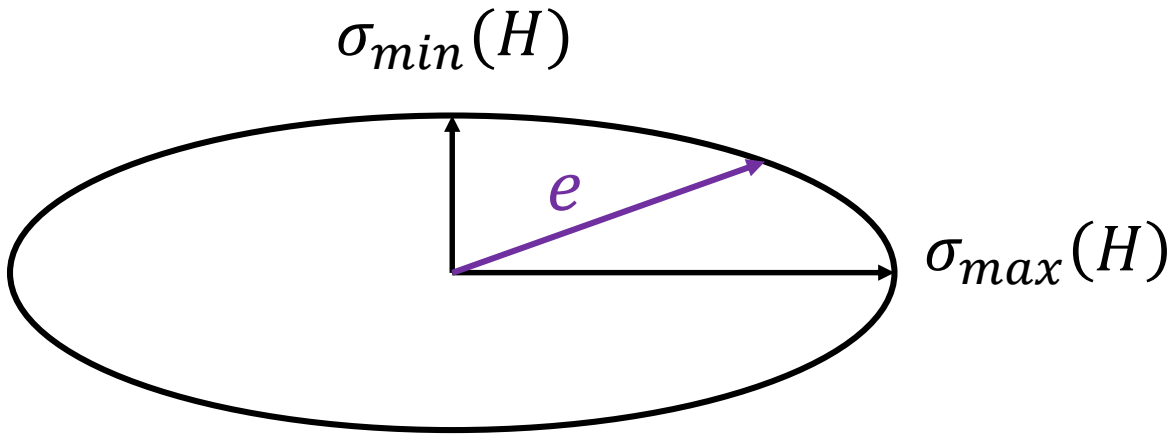
For scalars: variance, $\mathbb{E}\left[\left(x^2 - \mathbb{E}[x^2]\right)^2\right]$

For matrices: matrix variance (Tropp 2015), $\tilde{\kappa}$

statistical condition number

Statistical vs computational condition number

$$H \overset{\text{def}}{=} \mathbb{E}\left[x_i x_i^\top\right]$$



Define random variable: $(e^\top x_i)^2$

$$\mathbb{E}[(e^\top x_i)^2] = e^\top H e$$

Statistical: $\tilde{\kappa}$

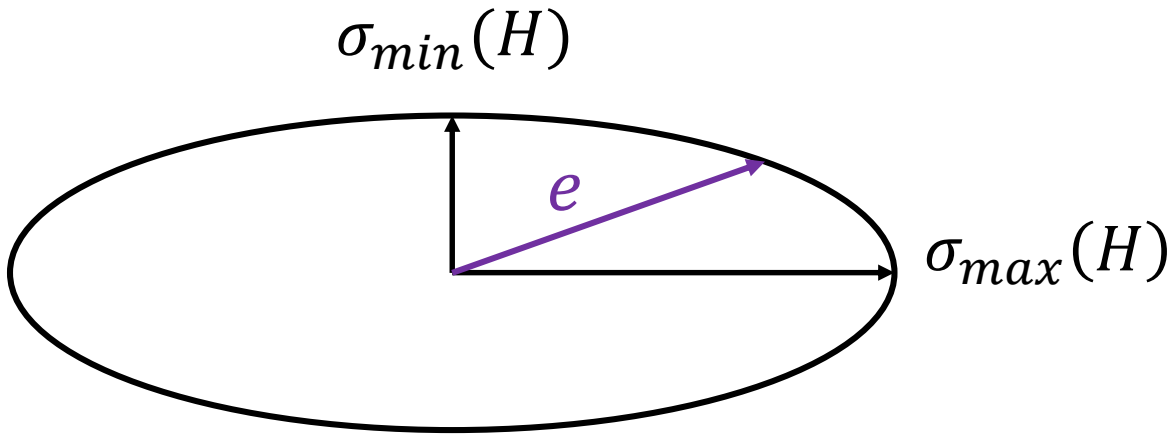
$$\sum_e \frac{\|e^\top x_i\|^2}{\mathbb{E}[(e^\top x_i)^2]}$$

Computational: κ

$$\frac{\|x_i\|^2}{\min_e \mathbb{E}[(e^\top x_i)^2]}$$

Statistical vs computational condition number

$$H \stackrel{\text{def}}{=} \mathbb{E}\left[x_i x_i^\top\right]$$



Define random variable: $(e^\top x_i)^2$

$$\mathbb{E}[(e^\top x_i)^2] = e^\top H e$$

Statistical: $\tilde{\kappa}$

$$\sum_e \frac{\|e^\top x_i\|^2}{\mathbb{E}[(e^\top x_i)^2]}$$

$\leq$

Computational: κ

$$\frac{\|x_i\|^2}{\min_e \mathbb{E}[(e^\top x_i)^2]}$$

Acceleration might be possible if $\tilde{\kappa} \ll \kappa$

Discrete vs Gaussian

Statistical: $\tilde{\kappa}$

$$\sum_e \frac{\|e^\top x_i\|^2}{\mathbb{E}[(e^\top x_i)^2]}$$

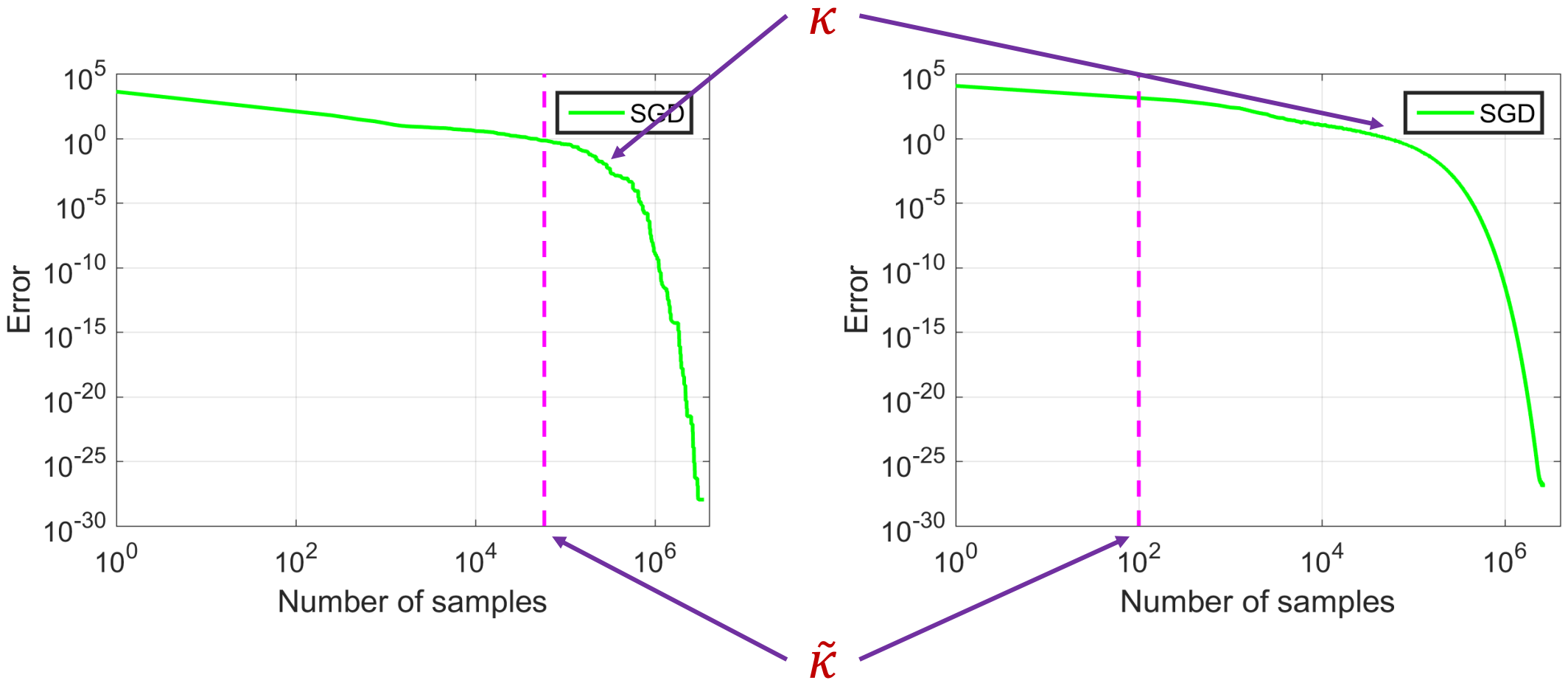
$\leq$

Computational: κ

$$\frac{\|x_i\|^2}{\min_e \mathbb{E}[(e^\top x_i)^2]}$$

Discrete	10^4	$\sim$	10^4
Gaussian	100	$\ll$	10^4

Discrete vs Gaussian



Discrete distribution

Gaussian distribution

Question 3: Do existing algorithms stochastic HB/NAG achieve this improvement?

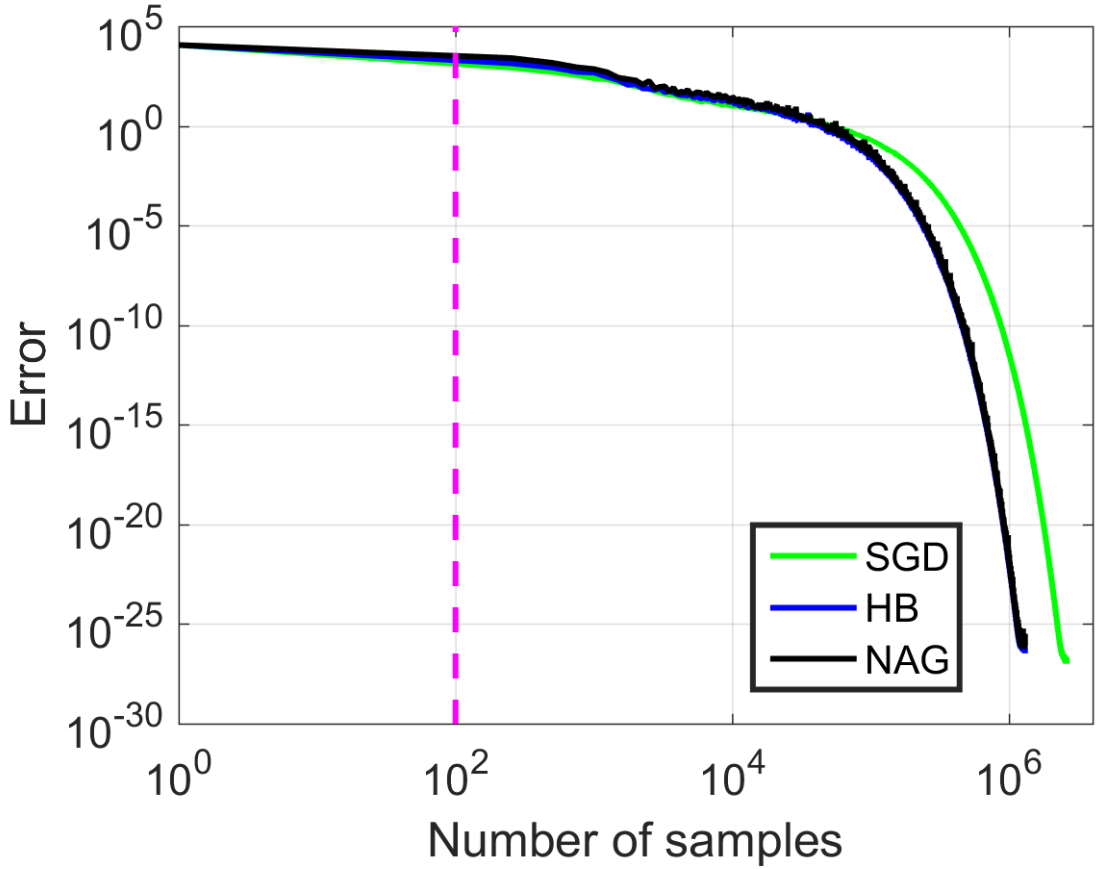
- Answer: No, there exist distributions where $\tilde{\kappa} \ll \kappa$ but rate of HB is $\widetilde{\Omega}\left(\kappa \log \frac{f(w_0)-f^*}{\epsilon}\right)$

- No better than SGD

- Same statement seems true empirically for NAG as well

- Fairly natural distributions

Empirical behavior of stochastic HB/NAG



Gaussian distribution

Question 4: Can we design an algorithm improving over SGD?

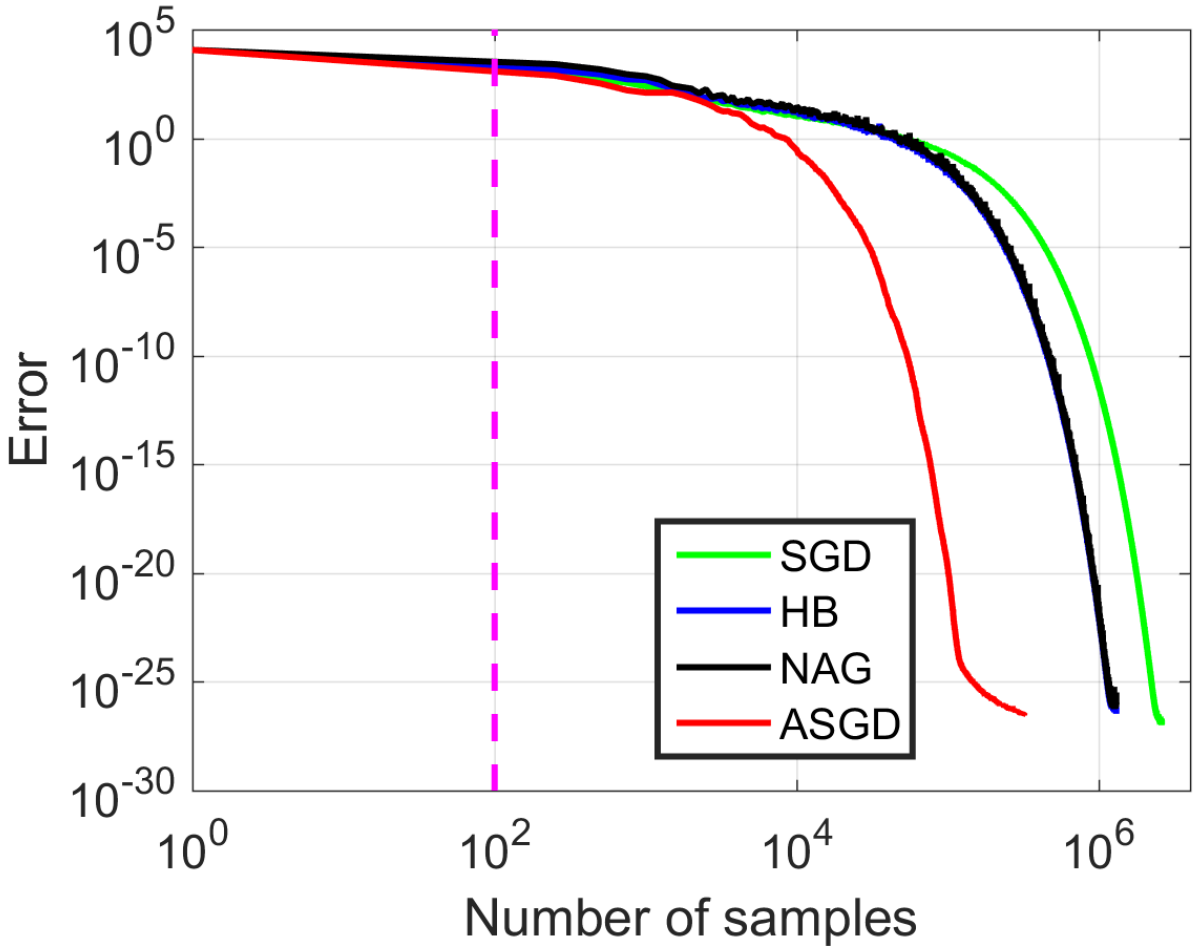
- Yes!

- Convergence rate of ASGD: $\tilde{O}\left(\sqrt{\kappa\tilde{\kappa}}\log\frac{f(w_0)-f^*}{\epsilon}+\frac{\sigma^2 d}{\epsilon}\right)$

- Compared to SGD: $\widetilde{\Theta}\left(\kappa\log\frac{f(w_0)-f^*}{\epsilon}+\frac{\sigma^2 d}{\epsilon}\right)$

- Improvement since $\tilde{\kappa}\leq\kappa$

Simulations



Does not achieve lower bound $\tilde{\kappa} \sim 100$. We believe not possible.

Algorithm

- Several equivalent versions of NAG in deterministic setting
  - 2 iterate version – most popular (gradient and momentum interpretation)
  - 4 iterate version – less popular (upper and lower bound interpretation)
  - Not equivalent in the stochastic setting

Parameters: $\alpha, \beta, \gamma, \delta$

1. $u_0 = w_0$
2. $v_{t-1} = \alpha w_{t-1} + (1-\alpha)u_{t-1}$
3. $w_t = v_{t-1} - \delta \hat{\nabla}_t f(v_{t-1})$
4. $z_{t-1} = \beta v_{t-1} + (1-\beta)u_{t-1}$
5. $u_t = z_{t-1} - \gamma \hat{\nabla}_t f(v_{t-1})$

Our algorithm

Proof overview

- Recall our guarantee: $\tilde{O}\left(\sqrt{\kappa\tilde{\kappa}}\log\frac{f(w_0)-f^*}{\epsilon}+\frac{\sigma^2 d}{\epsilon}\right)$
- First term depends on initial error; second is statistical error
- Different analyses for the two terms
- For the first term, analyze assuming $\sigma = 0$
- For the second term, analyze assuming $w_0 = w^*$

Part I: Potential function

- Iterates w_t, v_t of ASGD. $H \triangleq \mathbb{E}[xx^\top]$.

- Existing analyses use potential function

$$\|w_t - w^*\|_H^2 + \sigma_{\min}(H) \cdot \|u_t - w^*\|_2^2$$

- We use $\|w_t - w^*\|_2^2 + \sigma_{\min}(H) \cdot \|u_t - w^*\|_{H^{-1}}^2$

- We show $\|w_t - w^*\|_2^2 + \sigma_{\min}(H) \cdot \|u_t - w^*\|_{H^{-1}}^2$

$$\leq \left(1 - \frac{1}{\sqrt{\kappa\widetilde{\kappa}}}\right) \cdot \left(\|w_{t-1} - w^*\|_2^2 + \sigma_{\min}(H) \cdot \|u_{t-1} - w^*\|_{H^{-1}}^2\right)$$

Part II: Stochastic process analysis

$$\begin{bmatrix} w_{t+1} - w^* \\ v_{t+1} - w^* \end{bmatrix} = C \begin{bmatrix} w_t - w^* \\ w_t - w^* \end{bmatrix} + \text{noise}$$

Let $\theta_t \triangleq \mathbb{E} \begin{bmatrix} w_t - w^* \\ u_t - w^* \end{bmatrix} \begin{bmatrix} w_t - w^* \\ u_t - w^* \end{bmatrix}^\top$

$$\theta_{t+1} = \mathfrak{B}\theta_t + \text{noise} \cdot \text{noise}^\top$$

$$\theta_n \to \sum_i \mathfrak{B}^i (\text{noise} \cdot \text{noise}^\top)$$
$$= (\mathbb{I} - \mathfrak{B})^{-1} (\text{noise} \cdot \text{noise}^\top)$$

Parameters: $\alpha, \beta, \gamma, \delta$

1. $u_0 = w_0$
2. $v_{t-1} = \alpha w_{t-1} + (1 - \alpha) u_{t-1}$
3. $w_t = v_{t-1} - \delta \hat{\nabla} f(v_{t-1})$
4. $z_{t-1} = \beta v_{t-1} + (1 - \beta) u_{t-1}$
5. $u_t = z_{t-1} - \gamma \hat{\nabla} f(v_{t-1})$

Our algorithm

Part II: Stochastic process analysis

- Need to understand $(\mathbb{I} - \mathfrak{B})^{-1}(\text{noise} \cdot \text{noise}^\top)$
- $\mathfrak{B}$ has singular values > 1, but fortunately eigenvalues < 1
- Solve the 1-dim version of $(\mathbb{I} - \mathfrak{B})^{-1}(\text{noise} \cdot \text{noise}^\top)$ via explicit computations
- Combine the 1-dim bounds with (statistical) condition number bounds
- $\mathbb{E}[ww^\top] \preccurlyeq \tilde{\kappa} H^{-1} + \delta \cdot I$

Recap so far

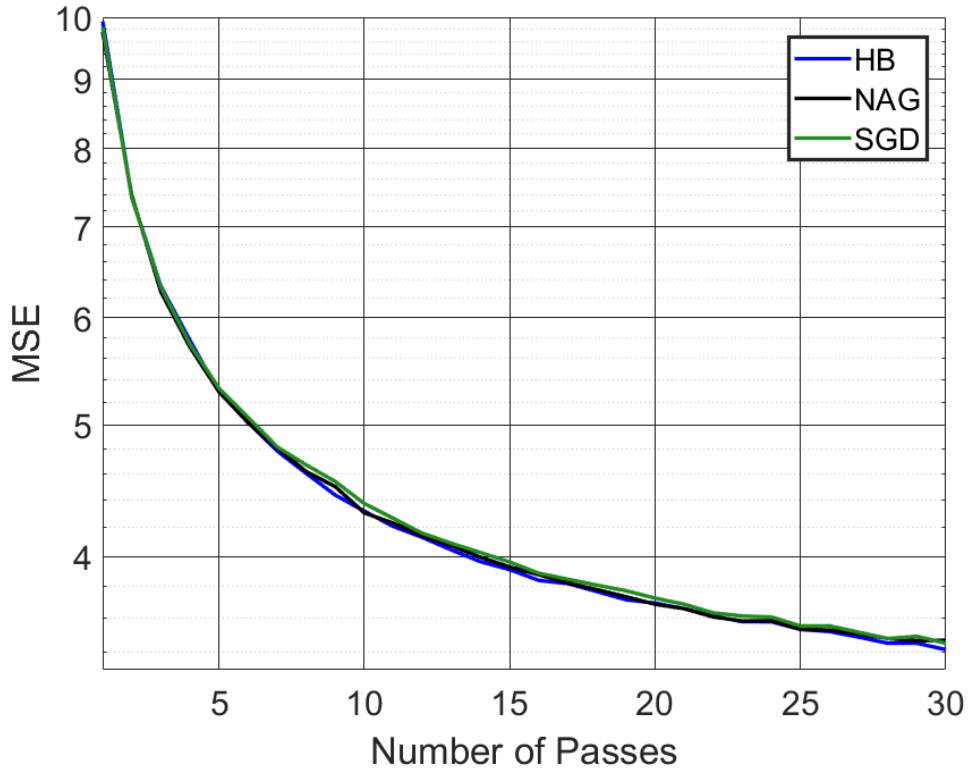
- For linear regression
  - Completely (theoretically) understand the behavior of various algorithms
  - Stochastic HB and NAG do not provide any improvement
  - Our algorithm (ASGD) improves over SGD, HB and NAG

- In practice
  - What does this mean for other problems e.g., training neural nets?
  - No proofs but do these intuitions help?

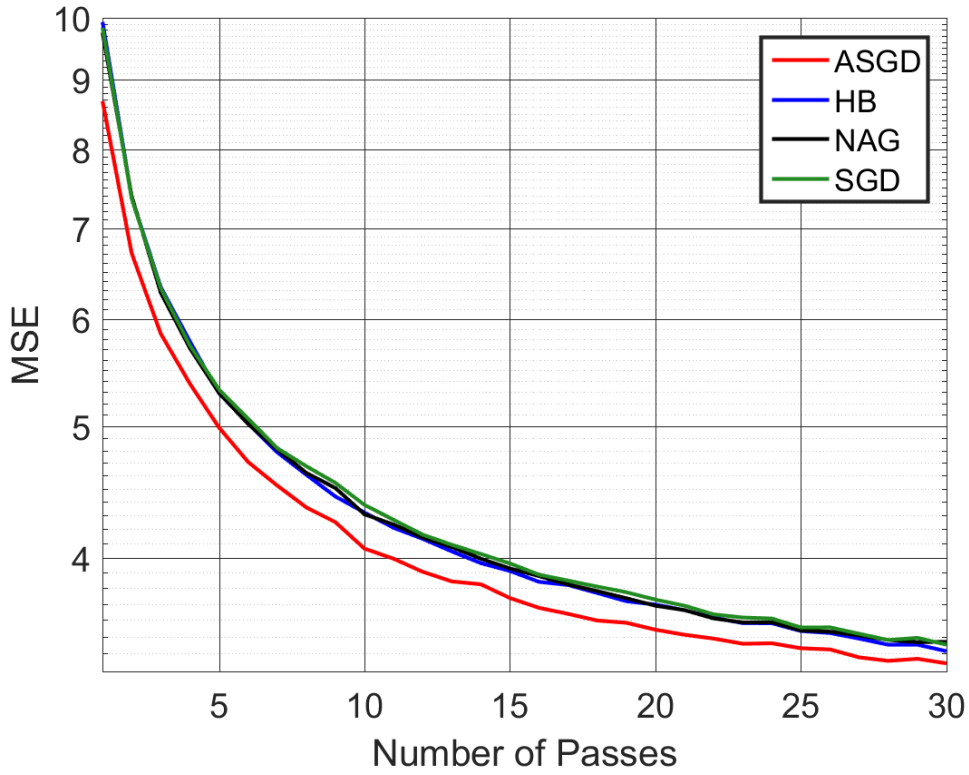
Stochastic HB and NAG in practice

- Why do stochastic HB and NAG still work better than SGD in practice?
- Conjecture: Minibatching; Large minibatch → Deterministic setting
- Precise quantification of small/large depends on dataset
- Our algorithm ASGD improves over SGD even for small minibatches

Deep autoencoder for mnist, small batch size (1)

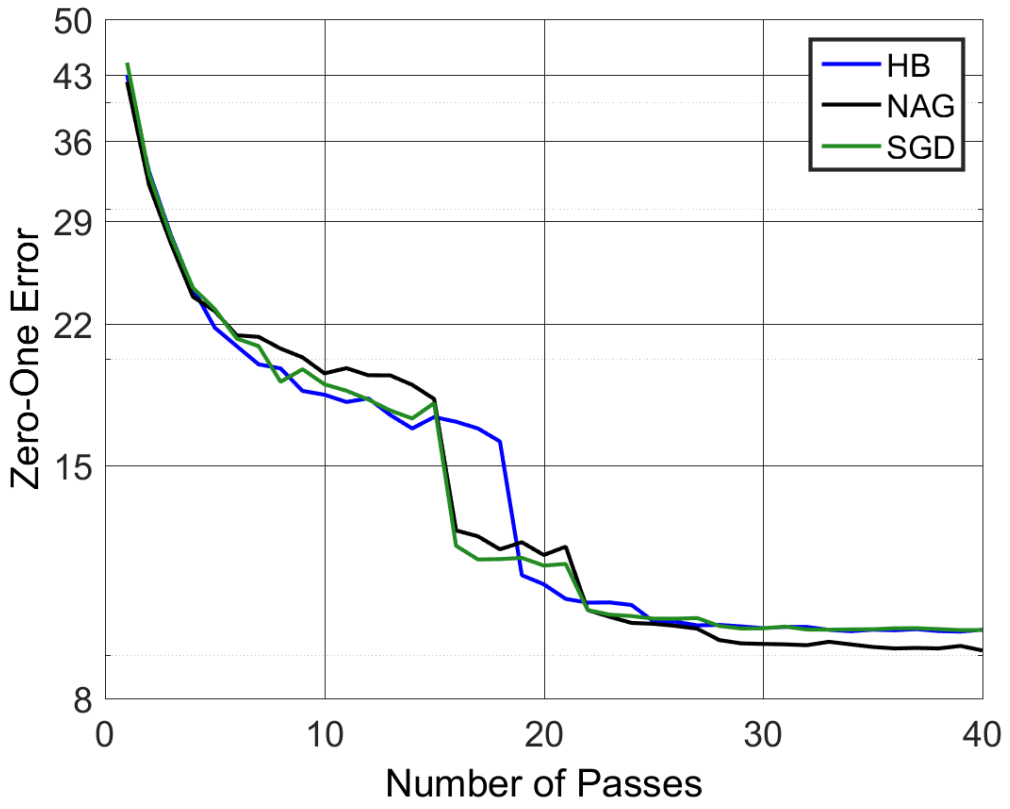


Deep autoencoder for mnist, small batch size (1)

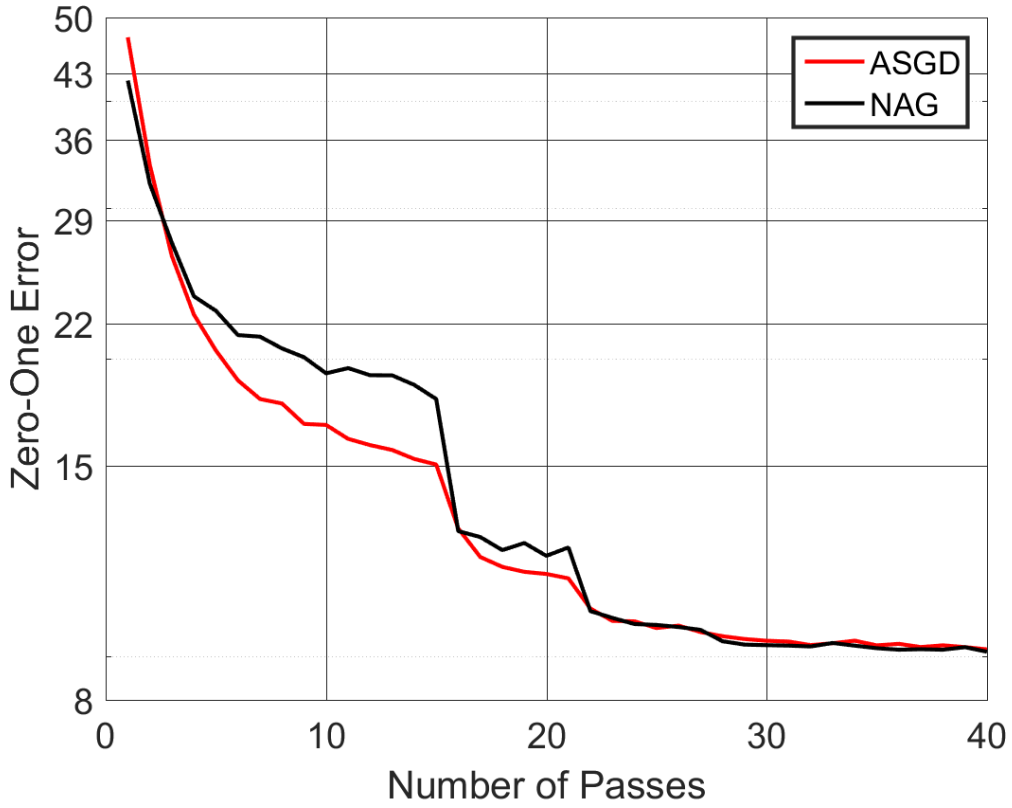


ASGD (our algorithm) vs others

Resnet for cifar-10 for small batch size (8)

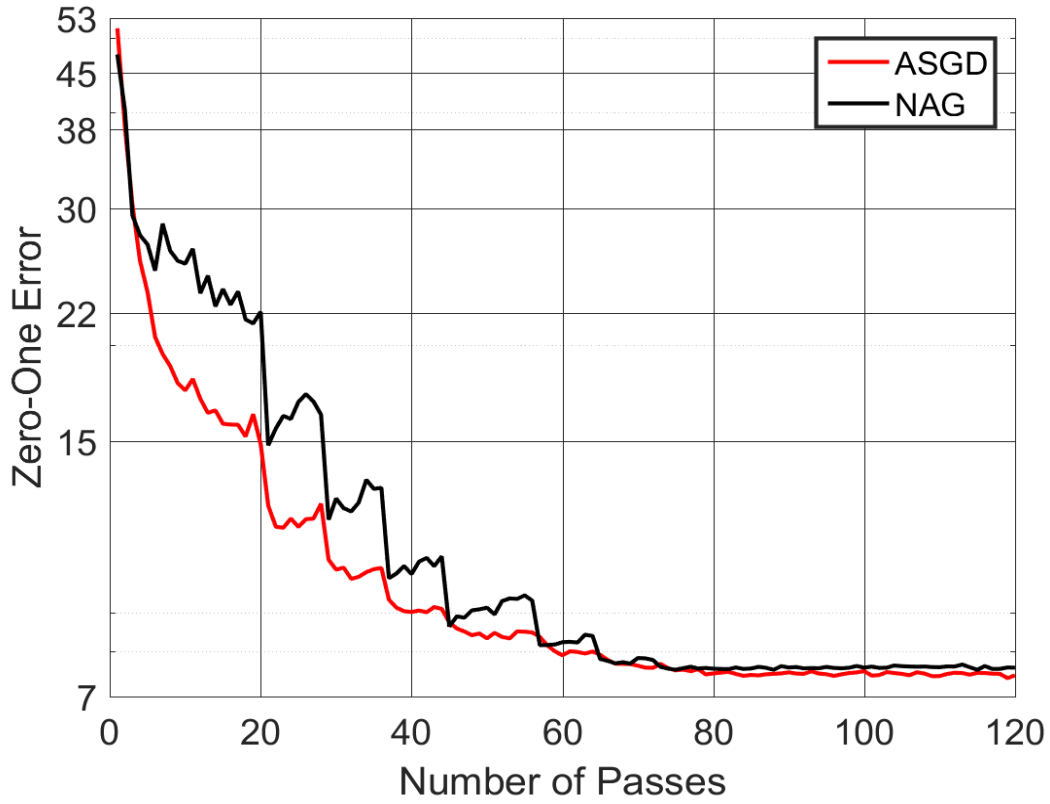


Resnet for cifar-10 for small batch size (8)



ASGD (our algorithm) vs NAG

Resnet for cifar-10 for usual batch size (128)



ASGD vs NAG

- Fast initial convergence – e.g., 1.5X faster than others to get to 90% accuracy
- Important for speeding up exploratory search for hyperparameters

Recap

Deterministic case	Stochastic approximation
GD $\Theta\left(\kappa \log \frac{f(x_0) - f^*}{\epsilon}\right)$	**SGD** $\widetilde{\Theta}\left(\kappa \log \frac{f(x_0) - f^*}{\epsilon} + \frac{\sigma^2 d}{\epsilon}\right)$
AGD $\Theta\left(\sqrt{\kappa} \log \frac{f(x_0) - f^*}{\epsilon}\right)$	**ASGD** $\tilde{O}\left(\sqrt{\kappa\tilde{\kappa}} \log \frac{f(x_0) - f^*}{\epsilon} + \frac{\sigma^2 d}{\epsilon}\right)$

- Stochastic HB/NAG do not accelerate in stochastic setting
- ASGD – acceleration possible in the stochastic setting
- Theoretical results for linear regression
- Works well for training neural nets (even for small batchsizes)
- Code in PyTorch: https://github.com/rahulkidambi/AccSGD

Optimization in neural networks

- Optimization methods used in training large scale neural networks not well understood, even for convex problems
- E.g., Adam, RMSProp etc. Interest in other methods as well
- Stochastic approximation provides a good framework
- Strong non-asymptotic results are of interest

Thank you!