University of Colorado **Boulder**

High-Probability convergence and algorithmic stability for stochastic gradient descent

Stephen Becker

University of Colorado Boulder (CU)

Machine Learning in Montpellier, Theory & Practice (ML-MTP) seminar

Sept 22 2022

Joint work with:

Emiliano Dall'Anese (CU)

Liam Madden (formerly CU, now UBC Vancouver)

High probability convergence for stochastic gradient descent assuming the Polyak-Lojasiewicz inequality, https://arxiv.org/abs/2006.05610 2021

The opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the National Science Foundation

Problem setting: Stochastic Gradient Descent (SGD)

$$\min_{x\in\mathbb{R}^n} f(x) \stackrel{\text{def}}{=} \mathbb{E}_\xi\left[F(x,\xi)\right] \text{ or } \mathbb{E}_{s\sim\mathbb{D}}[\ell(x,s)] \quad s=\{\text{features, label}\}$$

At every iteration, independently sample ξ_t and form our stochastic gradient $g=\nabla F(x_t,\xi_t)$ then iterate

$$x_{t+1}=x_t-\eta_t g$$

After T iterations, pick y such that $\|\nabla f(y)\|^2$ is small

Technical assumptions

- $F(\cdot,\xi)$ is differentiable a.s.
- minimizers exist
- ∇f is Lipschitz continuous
- $f\in C^1$
- $\mathbb{E}\left[\|\nabla f(x)-g\|^2\right]\le\sigma^2$

Machine learning example: **empirical risk minimization (ERM)**

Example: consider a data set $S:=(s_i)_{i=1}^n\sim\mathbb{D}^n$. The empirical risk is

$$f(x)=\frac{1}{n}\sum_{i=1}^n \ell(x,s_i)=\mathbb{E}_{i\sim U([n])}[\ell(x,s_i)].$$

Problem setting: learning

Empirical risk $f_n(x) \stackrel{\text{def}}{=} \dfrac{1}{n}\sum_{i=1}^{n} \ell(x, s_i)$ $\quad$ S or $S_n = (s_i)_{i=1}^n \stackrel{\text{iid}}{\sim} \mathbb{D}^n$

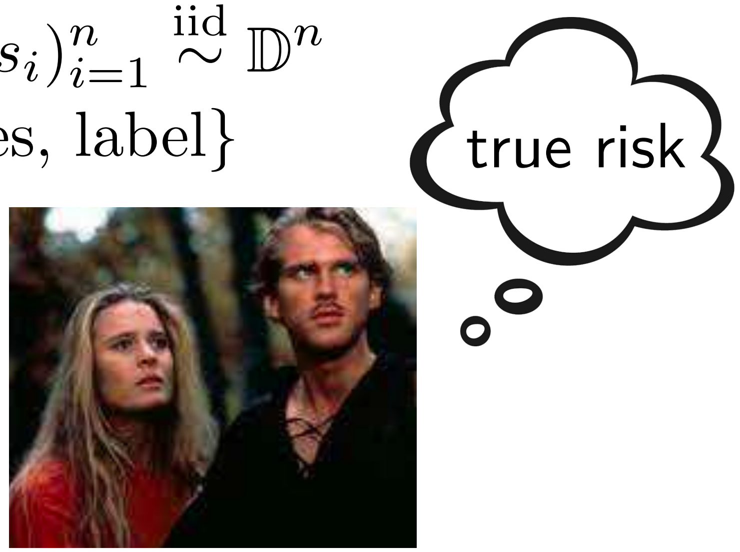
$s = \{\text{features, label}\}$

Problem setting: learning

Empirical risk
(training error)

$$f_n(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \ell(x, s_i) \qquad S \text{ or } S_n = (s_i)_{i=1}^n \stackrel{\text{iid}}{\sim} \mathbb{D}^n$$

$$s = \{\text{features, label}\}$$



True risk
(testing error)

$$f_\infty(x) \stackrel{\text{def}}{=} \mathbb{E}_{s\sim\mathbb{D}}[\ell(x, s)]$$

Problem setting: learning

Empirical risk $f_n(x) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \ell(x, s_i)$ $\quad S$ or $S_n = (s_i)_{i=1}^n \overset{\text{iid}}{\sim} \mathbb{D}^n$

$s = \{\text{features, label}\}$

True risk $f_\infty(x) \overset{\text{def}}{=} \mathbb{E}_{s\sim\mathbb{D}}[\ell(x, s)]$

No big deal? Strong law of large numbers says:

$$\forall x \text{ (a.s.)}, \lim_{n\to\infty} f_n(x) = f_\infty(x)$$

... but **fails** if x depends on n, e.g., $x = x(S_n), \text{e.g.}, x \in \operatorname{argmin} f_n(x)$

Problem setting: learning

Empirical risk $f_n(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \ell(x, s_i)$ $\quad S$ or $S_n = (s_i)_{i=1}^n \stackrel{\text{iid}}{\sim} \mathbb{D}^n$

$s = \{\text{features, label}\}$

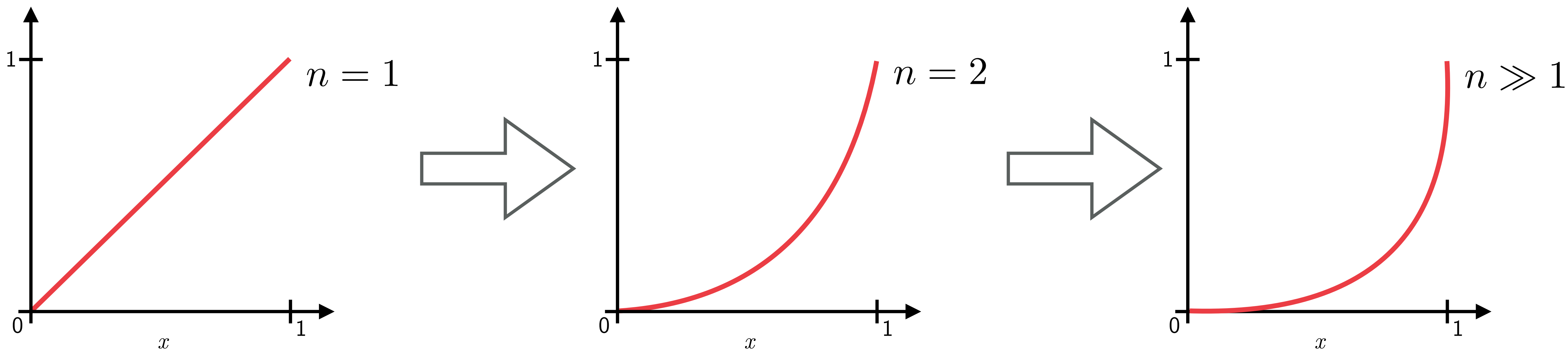
True risk $f_\infty(x) \stackrel{\text{def}}{=} \mathbb{E}_{s\sim\mathbb{D}}[\ell(x,s)]$

No big deal? Strong law of large numbers says:

$$\forall x \text{ (a.s.)}, \lim_{n\to\infty} f_n(x) = f_\infty(x)$$

... but **fails** if x depends on n, e.g., $x = x(S_n)$, e.g., $x \in \operatorname{argmin} f_n(x)$

Toy example: $f_n : [0,1] \to \mathbb{R}$, $\quad f_n(x) = x^n$, $f_n \stackrel{a.e.}{\to} 0$



Problem setting: learning

Empirical risk $\quad f_n(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \ell(x, s_i) \qquad S \text{ or } S_n = (s_i)_{i=1}^n \stackrel{\text{iid}}{\sim} \mathbb{D}^n$

$s = \{\text{features, label}\}$

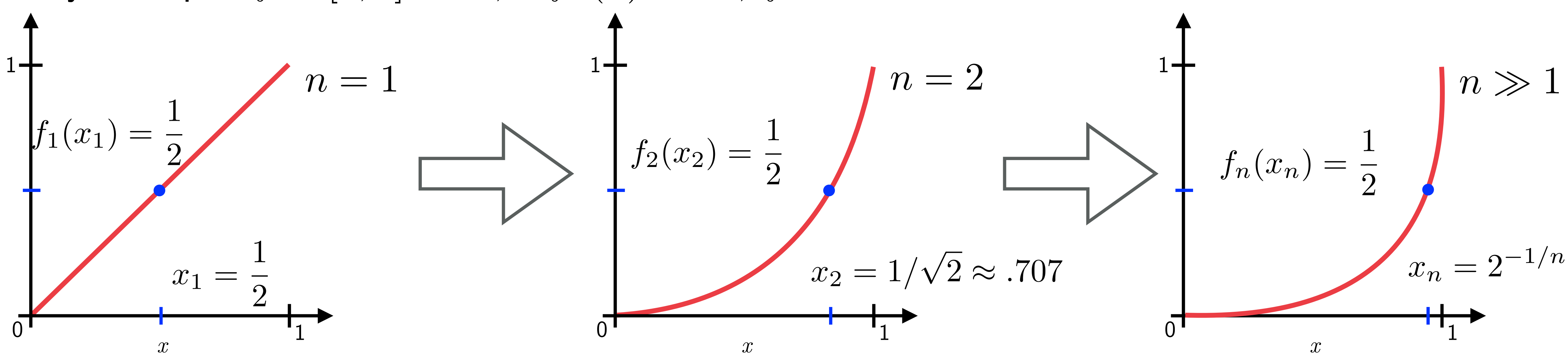
True risk $\quad f_\infty(x) \stackrel{\text{def}}{=} \mathbb{E}_{s\sim\mathbb{D}}[\ell(x,s)]$

No big deal? Strong law of large numbers says:

$$\forall x \text{ (a.s.)}, \lim_{n\to\infty} f_n(x) = f_\infty(x)$$

... but **fails** if x depends on n, e.g., $x = x(S_n), \text{e.g.}, x \in \operatorname{argmin} f_n(x)$

Toy example: $f_n : [0,1] \to \mathbb{R}, \quad f_n(x) = x^n, \; f_n \stackrel{a.e.}{\to} 0$



SGD has been analyzed since the 50's. What's new?

We're *not* assuming convexity, so just looking for a stationary point

Example of typical **theorem**. Assuming:

- ∇f is β Lipschitz continuous, f is bounded below (wlog, nonnegative)
- $\mathbb{E}\left[\|g_t\|^2\right] \le M + M'\|\nabla f(x)\|^2$ (e.g., iterates are bounded, or f is Lipschitz)

then

- Fixed stepsize: $0 < \eta \le \frac{1}{\beta M'}$ then $\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla f(x_t)\|^2\right] \le \eta\beta M + 2\frac{f(x_1)}{\eta T}$
- Decaying stepsize: $\sum_{t=1}^{\infty}\eta_t = \infty, \sum_{t=1}^{\infty}\eta_t^2 < \infty,$ e.g., $\eta_t = 1/t$

then $\liminf_{T\to\infty} \mathbb{E}\left[\|\nabla f(x_T)\|^2\right] = 0$

(and limit exists under additional smoothness assumptions)

Bottou, Curtis, Nocedal, *SIAM Review* 2018

More existing results

- Bertsekas and Tsitsiklis 2000:

$$P\left(x_t \to x \text{ such that } \nabla f(x) = 0\right) = 1.$$

"almost sure" convergence

- Ghadimi and Lan 2013:

$$\mathbb{E}\left[\|\nabla f(y)\|^2\right] \le O(\log(T)/\sqrt{T}).$$

convergence in mean
aka L^1 convergence

- Sebbouh *et al* 2021:

$$P\left(\min_{t\in[T]} \|\nabla f(x_t)\|^2 = o(1/T^{0.5-\epsilon})\right) = 1.$$

"almost sure" w/ rate

... and many other results and with different assumptions.

What about something **concrete** like $P\left(\|\nabla f(y)\|^2 < \epsilon\right) > 1 - \delta$?

Outline

Analyze SGD.

1. Robustness: allow heavier tailed noise, derive **high probability** bounds

   Assumptions: gradient Lipschitz, function Lipschitz, allow sub-Weibull noise

2. Learning/generalization

   Assumptions: gradient Lipschitz, PL inequality, only sub-Gaussian noise

New assumptions

Allow noise to be heavier tailed [beyond sub-Gaussian]

Example: consider a data set $S := (s_i)_{i=1}^n \sim \mathbb{D}^n$. The empirical risk is

$$f(x) = \frac{1}{n}\sum_{i=1}^n \ell(x, s_i) = \mathbb{E}_{i\sim U([n])}[\ell(x, s_i)].$$

$$f(x) = \frac{1}{n}\sum_{i=1}^n \ell(x, s_i) \approx \underbrace{\frac{1}{b}\sum_{j=1}^b \ell(x, s_{i_{(j)}})}_{\text{minibatch approximation}}$$

By CLT, the error in minibatch approximation converges in distribution to a Gaussian as $b \to \infty$

But empirically, noise is *not Gaussian* for small b

Panigrahi *et al.* (2019) looked at Resnet18 with CIFAR10 and MNIST data sets and found:

- Noise is Gaussian for $b = 4096$
- Noise is not Gaussian for $b = 32$
- Noise starts Gaussian then (after some epochs) becomes non-Gaussian for $b = 256$

Also, purposefully having heavier tail noise may help SGD find models that generalize well

Heavier-tailed noise: sub-Weibull distribution

punchline: like sub-Gaussian but heavier tailed

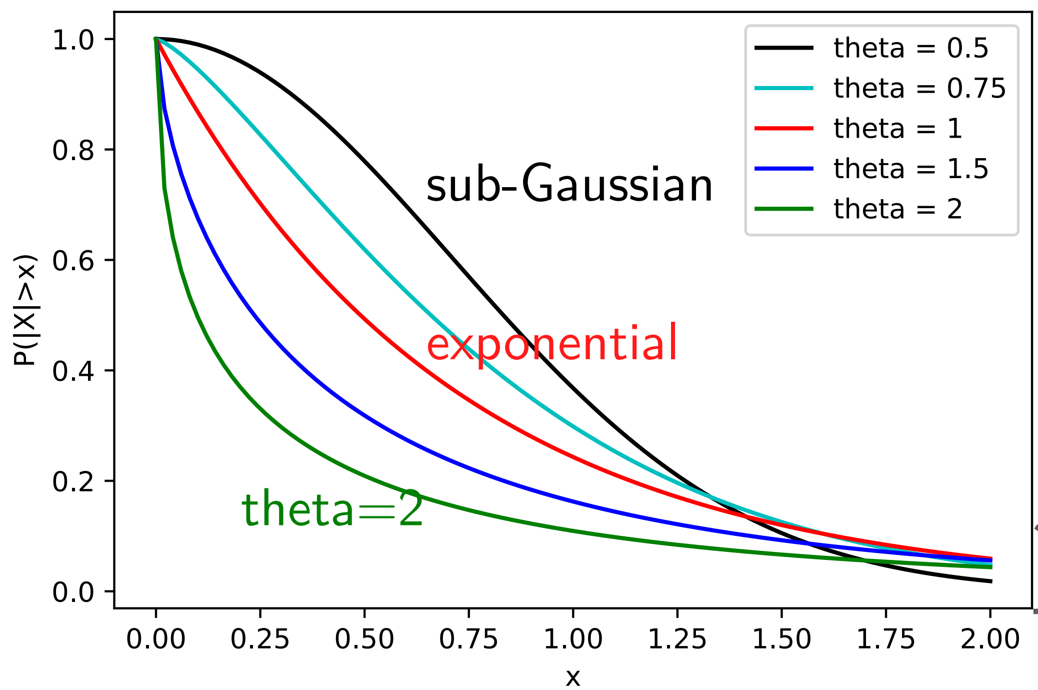
X is σ-sub-Gaussian if

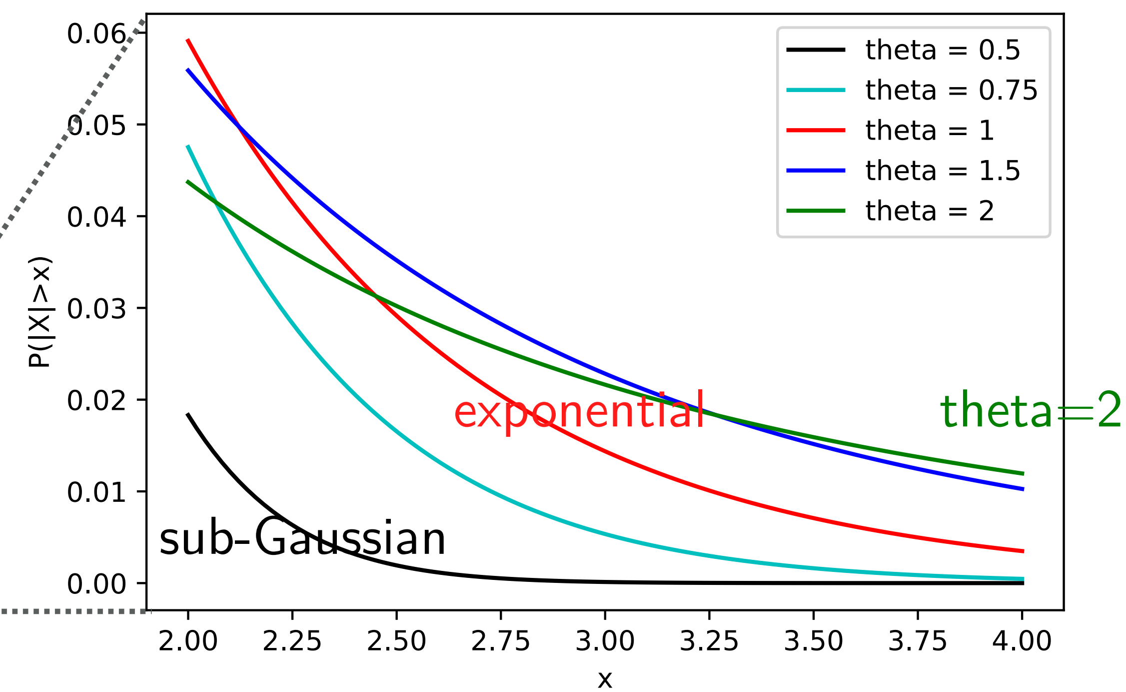
$$P(|X| \geq x) \leq 2\exp\left(-x^2/\sigma^2\right).$$

X is σ-sub-Weibull(θ) if (Vladimirova *et al* 2020)

$$P(|X| \geq x) \leq 2\exp\left(-(x/\sigma)^{1/\theta}\right).$$

Sub-Gaussian is $\theta = 1/2$. Sub-exponential is $\theta = 1$.





Mariia Vladimirova, Stéphane Girard, Hien Nguyen, and Julyan Arbel, *Sub-Weibull distributions: Generalizing sub-Gaussian and sub-exponential properties to heavier tailed distributions*, Stat **9** (2020), no. 1, e318.

High probability

What about something **concrete** like $P\left(\|\nabla f(y)\|^2 < \epsilon\right) > 1 - \delta$?

aka $P\left(\|\nabla f(y)\|^2 \ge \epsilon\right) \le \delta$

Using a result like

$$\mathbb{E}\left[\|\nabla f(y)\|^2\right] \le O\left(\log(T)/\sqrt{T}\right)$$

(y chosen after T iterations of SGD)

we can use Markov's inequality

$$\boxed{P\left(X \ge a\right) \le \frac{\mathbb{E}[X]}{a}}$$

(X is a non-negative r.v.)

to derive:

$$P\left(\|\nabla f(y)\|^2 \ge \frac{1}{\delta}\log(T)/\sqrt{T}\right) \le \delta$$

"low probability" result

Usual workaround is "probability amplification": run K independent algorithms, and pick the best output [sometimes tricky]

Via concentration inequalities, can get effectively $P\left(\|\nabla f(y)\|^2 \ge \log\left(\frac{1}{\delta}\right)\log(T)/\sqrt{T}\right) \le \delta$

"high probability" result

High probability

What about something **concrete** like $P\left(\|\nabla f(y)\|^2 < \epsilon\right) > 1 - \delta$?

Using

we ca

to de

$P\Big($

Usual

and p

Via concentration inequalities, can get effectively $P\left(\|\nabla f(y)\|^2 \ge \log\left(\frac{1}{\delta}\right)\log(T)/\sqrt{T}\right) \le \delta$

"high probability" result

Probability Amplification

Suppose we have a r.v. such that $\mathbb{P}[X > \epsilon(\delta)] \le \delta$

$$\epsilon(\delta) = \frac{1}{\delta} \cdot \frac{1}{T}$$

High probability

What about something **concrete** like $P\left(\|\nabla f(y)\|^2 < \epsilon\right) > 1-\delta$?

Using

we ca

to de

$P\Big(\|$

Usual

and p

Probability Amplification

Suppose we have a r.v. such that $\mathbb{P}[X > \epsilon(\delta)] \le \delta$ $\qquad \epsilon(\delta) = \frac{1}{\delta}\cdot\frac{1}{T}$

If we can make **independent** copies of it with $(\forall i = 1,\ldots,K) \quad \mathbb{P}[X_i > \epsilon(\delta)] \le \delta$

and if we have a way to select the **best** (smallest), then:

$$\mathbb{P}[\text{all } X_i > \epsilon(\delta_0)] = \prod_{i=1}^{K}\mathbb{P}[X_i > \epsilon(\delta_0)] \text{ by independence}$$

$$\le {\delta_0}^K \equiv \delta. \quad \text{Ex.: } \delta_0 = \frac{1}{2},\ 2^{-K} = \delta,\ \text{i.e., } K = \log_2(\delta^{-1})$$

Via concentration inequalities, can get effectively $P\left(\|\nabla f(y)\|^2 \ge \log\left(\frac{1}{\delta}\right)\log(T)/\sqrt{T}\right) \le \delta$

"high probability" result

High probability

What about something **concrete** like $P\left(\|\nabla f(y)\|^2 < \epsilon\right) > 1 - \delta$?

Using

we ca

to de

$P\Big(\|$

Usual

and p

Probability Amplification

Suppose we have a r.v. such that $\mathbb{P}[X > \epsilon(\delta)] \le \delta$

$$\epsilon(\delta) = \frac{1}{\delta} \cdot \frac{1}{T}$$

If we can make **independent** copies of it with $(\forall i = 1, \ldots, K) \quad \mathbb{P}[X_i > \epsilon(\delta)] \le \delta$

and if we have a way to select the **best** (smallest), then:

$$\mathbb{P}[\text{all } X_i > \epsilon(\delta_0)] = \prod_{i=1}^{K} \mathbb{P}[X_i > \epsilon(\delta_0)] \text{ by independence}$$

$$\le {\delta_0}^K \equiv \delta. \quad \text{Ex.: } \delta_0 = \frac{1}{2},\ 2^{-K} = \delta, \text{ i.e., } K = \log_2(\delta^{-1})$$

The effect is:

$$\epsilon_{\text{amplified}}(\delta) = \frac{1}{\frac{1}{2}} \cdot \frac{1}{\frac{T}{K}} = 2\log_2(\delta^{-1}) \cdot \frac{1}{T}$$

Via concentration inequalities, can get effectively $P\left(\|\nabla f(y)\|^2 \ge \log\left(\frac{1}{\delta}\right) \log(T)/\sqrt{T}\right) \le \delta$

"high probability" result

High probability

What about something **concrete** like $P\left(\|\nabla f(y)\|^2 < \epsilon\right) > 1 - \delta$?

aka $P\left(\|\nabla f(y)\|^2 \ge \epsilon\right) \le \delta$

Using a result like

$$\mathbb{E}\left[\|\nabla f(y)\|^2\right] \le O\left(\log(T)/\sqrt{T}\right)$$ (y chosen after T iterations of SGD)

we can use Markov's inequality $\boxed{P(X \ge a) \le \frac{\mathbb{E}[X]}{a}}$ (X is a non-negative r.v.)

to derive:

$$P\left(\|\nabla f(y)\|^2 \ge \frac{1}{\delta}\log(T)/\sqrt{T}\right) \le \delta$$ **"low probability"** result

Usual workaround is "probability amplification": run K independent algorithms, and pick the best output [sometimes tricky]

Via concentration inequalities, can get effectively $P\left(\|\nabla f(y)\|^2 \ge \log\left(\frac{1}{\delta}\right)\log(T)/\sqrt{T}\right) \le \delta$

"high probability" result

Research goal

Allowing for **heavier-tailed noise** (e.g., sub-Weibull with $\theta = 1$ or even $\theta > 1$)

can we derive a (single-run) **high-probability** convergence result?

Yes!

Main result

Recall: θ is the sub-Weibull parameter

Theorem [Thm. 12 in Madden, Dall'Anese, B. '21]

If $P\left(\|\nabla f(x) - \nabla F(x,\xi)\| \geq r) \leq 2\exp(-(r/\sigma)^{1/\theta}\right)$ $\forall r > 0,$ then for T iterations of SGD with step-size $\eta_t = \Theta(1/\sqrt{t}),$ we have, w.p. $\geq 1 - \delta$,

$$\min_{t\in[T]} \|\nabla f(x_t)\|^2 \leq \frac{1}{\sqrt{T}} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \|\nabla f(x_t)\|^2$$

$$\leq \mathcal{O}\left(\frac{\log(T)\log(1/\delta)^{2\theta} + \log(T/\delta)^{\max\{0,\theta-1\}})\log(1/\delta)}{\sqrt{T}} \right)$$

*Do need to assume function is Lipschitz if beyond sub-Gaussian noise unfortunately

Main result

Recall: θ is the sub-Weibull parameter

Theorem [Thm. 12 in Madden, Dall'Anese, B. '21]

If $P\left(\|\nabla f(x) - \nabla F(x,\xi)\| \geq r\right) \leq 2\exp(-(r/\sigma)^{1/\theta})\right)$ $\forall r > 0$, then for T iterations of SGD with step-size $\eta_t = \Theta(1/\sqrt{t})$, we have, w.p. $\geq 1-\delta$,

$$\min_{t\in[T]} \|\nabla f(x_t)\|^2 \leq \frac{1}{\sqrt{T}} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \|\nabla f(x_t)\|^2$$

$$\leq \mathcal{O}\left(\frac{\log(T)\log(1/\delta)^{2\theta} + \log(T/\delta)^{\max\{0,\theta-1\}})\log(1/\delta)}{\sqrt{T}}\right)$$

Special case of sub-Gaussian ($\theta = \frac{1}{2}$) already had results by Li and Orabona '20:

$$P\left(\min_{t\in[T]} \|\nabla f(x_t)\|^2 \geq \Omega\left(\log(T/\delta)\log(T)/\sqrt{T}\right)\right) \leq O(\delta)$$

In addition to generalizing this, we slightly improve it:

$$P\left(\min_{t\in[T]} \|\nabla f(x_t)\|^2 \geq \Omega\left(\log(1/\delta)\log(T)/\sqrt{T}\right)\right) \leq O(\delta)$$

*Do need to assume function is Lipschitz if beyond sub-Gaussian noise unfortunately

Detail: post-processing

For a stochastic problem, it can be expensive or impossible to compute $\min_{t\in[T]} \|\nabla f(x_t)\|^2$

(note: for convex problems, this is not an issue since we can use Jensen's inequality)

issue 1

issue 2

Detail: post-processing, 1

For a stochastic problem, it can be expensive or impossible to compute $\min_{t \in [T]} \|\nabla f(x_t)\|^2$

(note: for convex problems, this is not an issue since we can use Jensen's inequality)

issue 1

issue 2

Solution: **sampling**. Use standard concentration inequalities (Hoeffding, etc.) under various assumptions; all samples are iid, so classical analysis.

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} \ell(x, s_i) \approx \frac{1}{b} \sum_{j=1}^{b} \ell(x, s_{i_{(j)}})$$

Detail: post-processing, 2

For a stochastic problem, it can be expensive or impossible to compute $\underset{t\in[T]}{\min}\|\nabla f(x_t)\|^2$

(note: for convex problems, this is not an issue since we can use Jensen's inequality)

issue 1

issue 2

Saeed Ghadimi and Guanghui Lan, *Stochastic first-and zeroth-order methods for nonconvex stochastic programming*, SIAM Journal on Optimization **23** (2013), no. 4, 2341–2368.

Solution: **sampling again**! We extend a variant of a trick used by Ghadimi and Lan '13

Proposition [Corollary of Lemma 33 in Madden, Dall'Anese, B. '21]

If we sample a set $\mathcal{S}$ of n_{ind} indices in $[T]$ choosing t w.p. $\propto 1/\sqrt{t}$ independently with replacement, then $(\forall \epsilon > 0)$

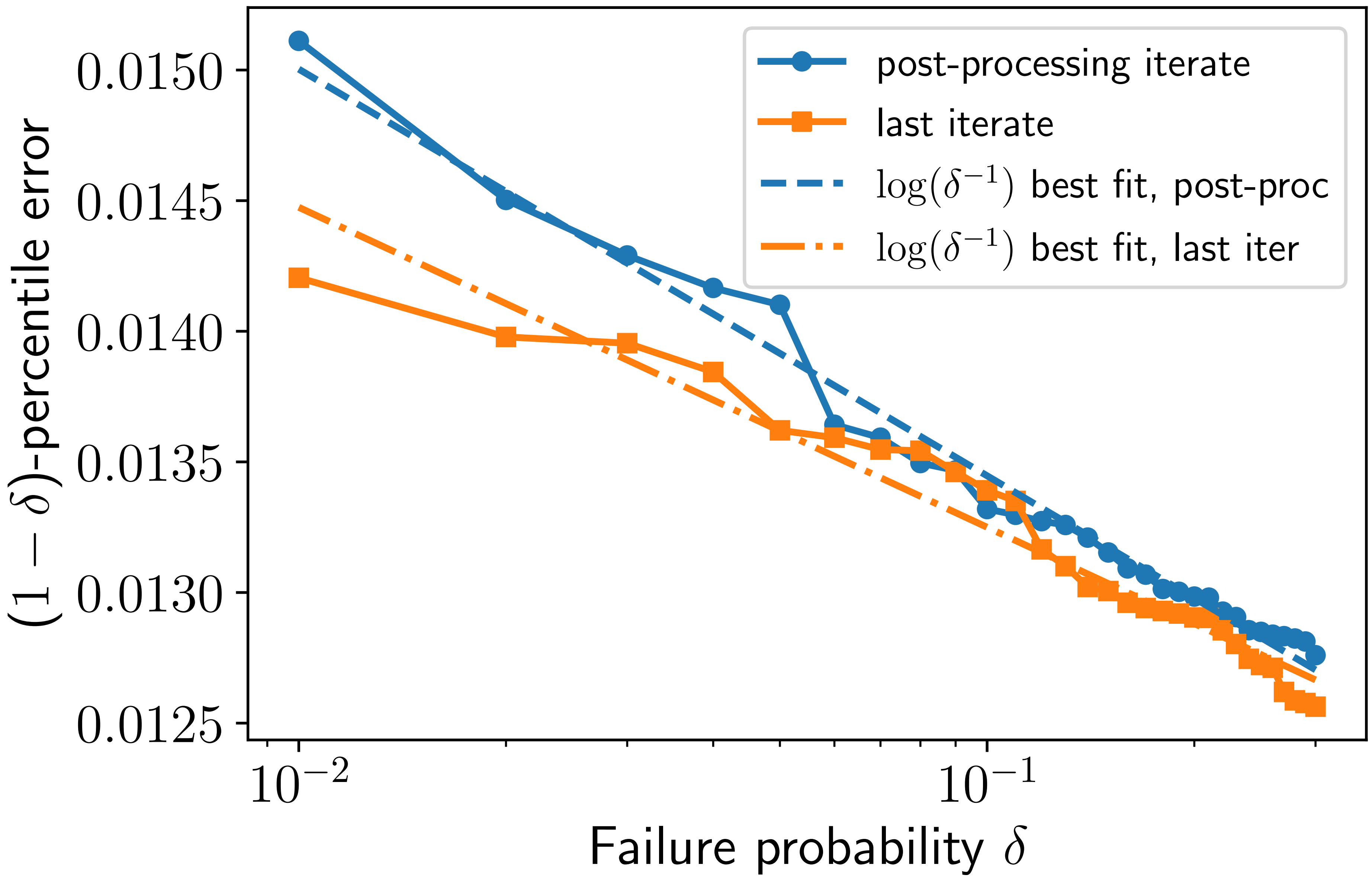
$$P\left(\min_{t\in\mathcal{S}}\|\nabla f(x_t)\|^2 > \exp(1)\epsilon\right) \le \exp(-n_{\text{ind}}) + \underbrace{P\left(\frac{1}{\sqrt{T}}\sum_{t=1}^{T}\frac{1}{\sqrt{t}}\|\nabla f(x_t)\|^2 > \epsilon\right)}$$

(this is the core quantity bounded in the easier theorem)

Numerics

Neural net (2 hidden layers) example

Is the error actually dependent on $\log(\delta)$?



Technique

Step 1: standard optimization analysis

- We can derive

$$O\left(\sum \eta_t \|\nabla f(x_t)\|^2\right) \leq O(1) + O\left(\sum \eta_t \langle \nabla f(x_t), e_t\rangle\right) + O\left(\sum \eta_t^2 \|e_t\|^2\right)$$

where $e_t = \nabla f(x_t) - \nabla F(x_t, \xi_t)$.

- Define $\mathcal{F}_t = \sigma(\xi_0, \ldots, \xi_t)$. Then $(\eta_t \underbrace{\langle \nabla f(x_t), e_t\rangle}_{\xi_t})$ is adapted to $(\mathcal{F}_t)$ and $\mathbb{E}\left[\eta_t \langle \nabla f(x_t), e_t\rangle \mid \mathcal{F}_{t-1}\right] = 0$.

Need to condition here since x_t is a random variable

Technique

Existing bounds: sub-exponential Martingale Difference Sequence concentration

Theorem (Freedman)
(ξ_i) is a martingale difference sequence (MDS) if it is adapted to a filtration $(\mathcal{F}_i)$ and $\mathbb{E}[\xi_i \mid \mathcal{F}_{i-1}] = 0$. Let (V_i) be adapted to $(\mathcal{F}_i)$. Assume $V_i \geq 0\ \forall i \in [n]$ and, for some $\lambda \geq 0$ and $f \geq 0$,

$$\mathbb{E}[\exp(\lambda \xi_i) \mid \mathcal{F}_{i-1}] \leq \exp(f(\lambda) V_{i-1})\ \forall i \in [n].$$

?

Interpretation: if $\mathbb{E}[\xi] = 0$ then

$$\mathbb{E}[\exp(\lambda \xi)] \leq \exp(\lambda^2 V)$$

$$\forall \lambda \in \mathbb{R} \quad \text{or} \quad \forall |\lambda| \leq V^{-1/2}$$

sub-Gaussian sub-exponential

Technique

Existing bounds: sub-exponential Martingale Difference Sequence concentration

Theorem (Freedman)
(ξ_i) is a martingale difference sequence (MDS) if it is adapted to a filtration $(\mathcal{F}_i)$ and $\mathbb{E}\left[\xi_i \mid \mathcal{F}_{i-1}\right] = 0$. Let (V_i) be adapted to $(\mathcal{F}_i)$. Assume $V_i \geq 0\ \forall i \in [n]$ and, for some $\lambda \geq 0$ and $f \geq 0$,

$$\mathbb{E}\left[\exp(\lambda \xi_i) \mid \mathcal{F}_{i-1}\right] \leq \exp(f(\lambda) V_{i-1})\ \forall i \in [n].$$

Then, for all $x, v \geq 0$,

$$P\left(\bigcup_{k \in [n]} \left\{ \sum_{i=1}^{k} \xi_i \geq x \text{ and } \sum_{i=1}^{k} V_{i-1} \leq v \right\}\right) \leq \exp(-\lambda x + f(\lambda) v).$$

This comes from Fan *et al* 2015 but goes back to Freedman 1975.

Xiequan Fan, Ion Grama, Quansheng Liu, et al., *Exponential inequalities for martingales with applications*, Electronic Journal of Probability **20** (2015).

David A Freedman, *On tail probabilities for martingales*, The Annals of Probability (1975), 100–118.

Technique

Freedman's inequality from the last slide:

$$\mathbb{E}\left[\exp(\lambda\xi_i) \mid \mathcal{F}_{i-1}\right] \leq \exp(f(\lambda)V_{i-1}) \ \forall i \in [n].$$

Then, for all $x, v \geq 0$,

$$P\left(\bigcup_{k\in[n]} \left\{ \sum_{i=1}^{k} \xi_i \geq x \text{ and } \sum_{i=1}^{k} V_{i-1} \leq v \right\}\right) \leq \exp(-\lambda x + f(\lambda)v).$$

Special case:

$$f(\lambda) = \frac{\lambda^2}{2}, \quad V_{i-1} = \sigma_i^2, \quad v = \sum_{i=1}^{n} \sigma_i^2, \quad \lambda = x/v$$

$$\implies \quad P\left(\bigcup_{k\in[n]} \left\{ \sum_{i=1}^{k} \xi_i \geq x \right\}\right) \leq \exp\left(-\frac{x^2}{2v}\right)$$ (maximal) Azuma-Hoeffding inequality

e.g. $v = n\sigma^2$

(just a fancy version of Hoeffding... Hoeffding is for bounded independent random variables, we're generalizing to sub-exponential Martingales)

$$\underbrace{\mathbb{E}[\xi_i] = 0, \ \xi_i \in [-\sigma/2, \sigma/2]}_{\text{independent}} \overset{\text{Hoeffding}}{\implies} P\left(\sum_{i=1}^{n} \xi_i \geq x\right) \leq \exp\left(-2\frac{x^2}{n\sigma^2}\right)$$

Technique

Recall: θ is the sub-Weibull parameter

Step 2: generalized generalized Freedman

Nicholas J. A. Harvey, Christopher Liaw, Yaniv Plan, and Sikander Randhawa, *Tight analyses for non-smooth stochastic gradient descent*, Conference on learning theory (COLT), 2019, pp. 1579–1613.

Harvey et al. ('19) generalize to "self-normalized" Freedman for MDS [assuming sub-Gaussian]. We generalize to all sub-Weibull (below, showing just $\theta > 0$ case).

Theorem [Prop. 11 in Madden, Dall'Anese, B. '21]

Assume (ξ_i) is a MDS, and let (V_i) be adapted to $(\mathcal{F}_i)$. Assume $0 \leq V_{i-1} \leq a_i$ $(\forall i \in [n])$ and, for some $\theta > 1$,

$$\mathbb{E}\left[\exp\left((|\xi|_i / V_{i-1})^{1/\theta}\right) |\mathcal{F}_{i-1}\right] \leq 2 \quad (\forall i \in [n]).$$

Then, for all $x, \beta \geq 0$, $\delta \in (0,1)$, $\alpha \geq 2\log(n/\delta)^{\theta-1} \max_{i\in[n]} a_i$, and $\lambda \in \left[0, \frac{1}{2\alpha}\right]$,

$$P\left(\bigcup_{k\in[n]} \left\{\sum_{i=1}^{k} \xi_i \geq x \text{ and } c_\theta \sum_{i=1}^{k} V_{i-1}^2 \leq \alpha \sum_{i=1}^{k} \xi_i + \beta\right\}\right) \leq \exp(-\lambda x + 2\beta\lambda^2) + 2\delta$$

where $c_\theta = (2^{2\theta+1}+2)\Gamma(2\theta+1) + 2^{3\theta}\Gamma(3\theta+1)/3$.

Proof used MGF truncation techniques of Bakhshizadeh at al. 2020

Outline

Analyze SGD.

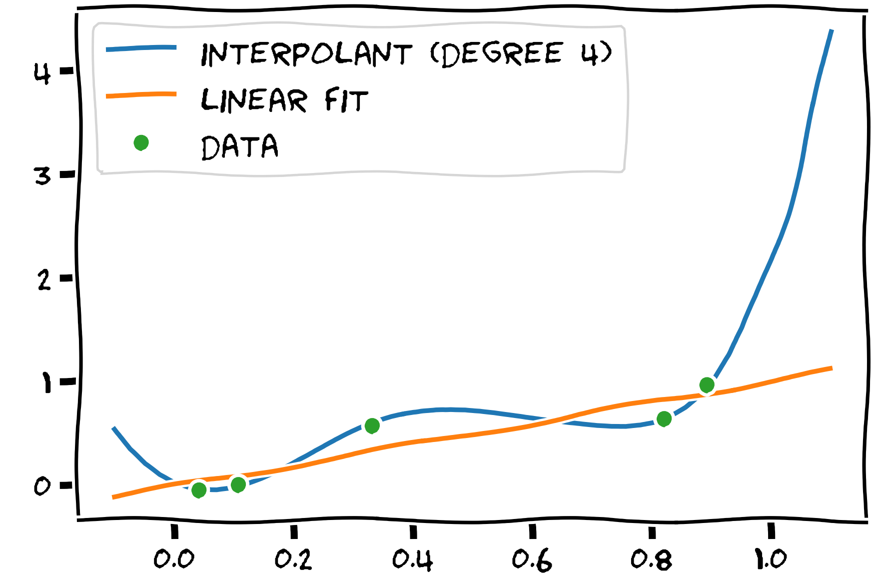
1. Robustness: allow heavier tailed noise, derive **high probability** bounds ✓

   Assumptions: gradient Lipschitz, function Lipschitz, allow sub-Weibull noise

2. Learning/generalization ✓

   Assumptions: gradient Lipschitz, PL inequality, only sub-Gaussian noise

Stability

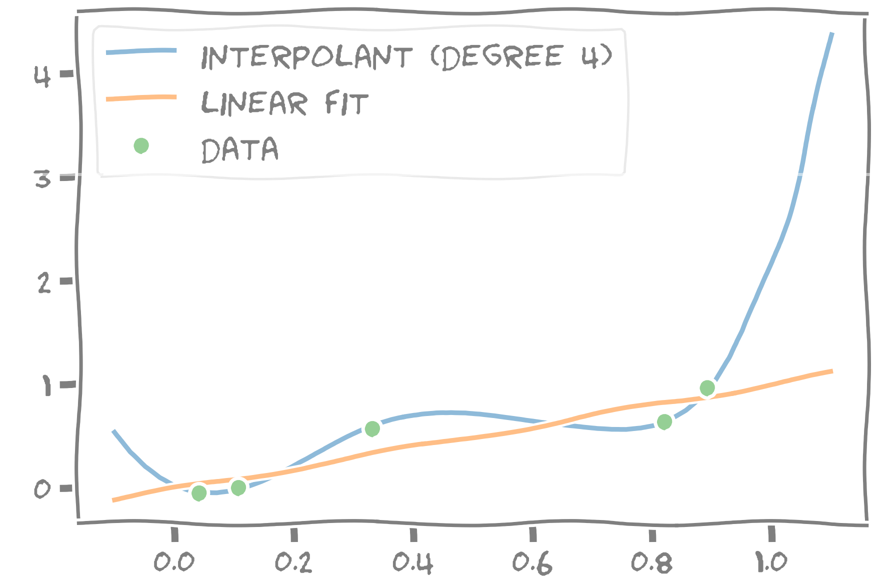


Old thinking:

An algorithm ALGO might be any global minimizer to the ERM problem

Tools: regularization or restrict the complexity of the hypothesis class (VC dimensions)

Stability



Old thinking:

An algorithm ALGO might be any global minimizer to the ERM problem

Tools: regularization or restrict the complexity of the hypothesis class (VC dimensions)

New thinking:

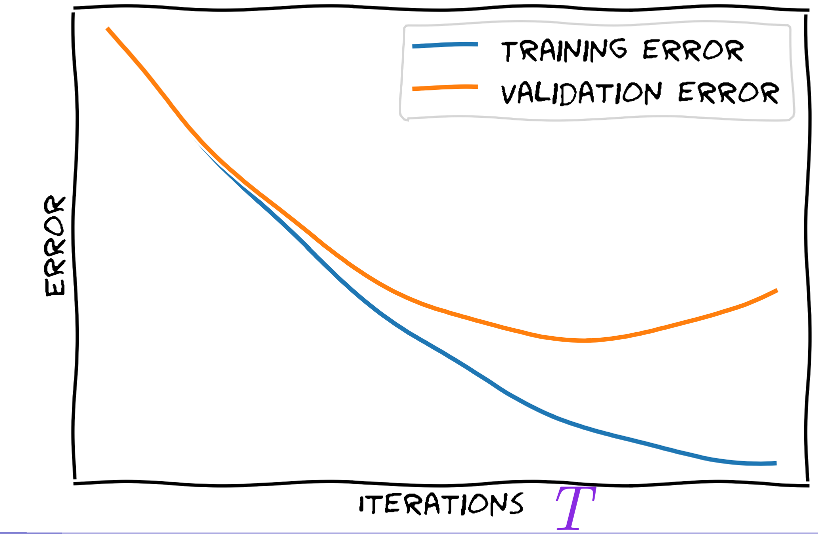
An algorithm ALGO can be the output of SGD after T iterations trying to solve the ERM problem. No longer need to assume we found **global** minimizer

e.g., take 0 iterations, then clearly it's insensitive to input...

... but we'll see a tradeoff with optimization error.

Change the stepsize? Change to using ADAM? etc. Then need new analysis (either a pro or con, depending on the situation)

Tools: stability. We'll use "stability" in the way you think we would: a **stable** algorithm is not overly sensitive to changes in the input.



Stability (technical definition)

Bousquet and Elisseef, *JMLR*, 2002
Elisseeff, Evgeniou and Pontil, *JMLR*, 2005

Definition Uniformly Stable in Expectation*

A randomized algorithm `ALGO` is $\varepsilon_{\text{stab}}$-uniformly stable if for all datasets S and S' (both of size n) that differ in at most one example,

$$\sup_{s \in \mathcal{S}} \underbrace{\mathbb{E}_{\texttt{ALGO}}\left[\ell(x,s) - \ell(x',s)\right]} \le \varepsilon_{\text{stab}}, \qquad x = \texttt{ALGO}(S),\ x' = \texttt{ALGO}(S')$$

*There are many variants, eg. "pointwise" variants

$$\overbrace{\mathbb{E}_{\texttt{ALGO}}\left[\ell(\texttt{ALGO}(S),s) - \ell(\texttt{ALGO}(S'),s)\right]} \overset{\text{def}}{=} \mathbb{E}_{\xi}\left[\ell(\texttt{ALGO}(S,\xi),s) - \ell(\texttt{ALGO}(S',\xi),s)\right]$$

ξ represents all the randomness in the algorithm
like a **seed** for a pseudo-random number generator
ex: initialization, and/or minibatch samples

It is the **same** for both runs (as in Hardt et al., different than in Elisseef)

Recall

$$f_n(x) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \ell(x, s_i)$$

$$f_\infty(x) \overset{\text{def}}{=} \mathbb{E}_{s\sim\mathbb{D}}[\ell(x,s)]$$

Stability: usefulness

Bousquet and Elisseef, *JMLR*, 2002
Elisseeff, Evgeniou and Pontil, *JMLR*, 2005

Definition Uniformly Stable in Expectation

A randomized algorithm $\texttt{ALGO}$ is $\varepsilon_{\text{stab}}$-uniformly stable if for all datasets S and S' (both of size n) that differ in at most one example,

$$\sup_{s \in \mathcal{S}} \mathbb{E}_{\texttt{ALGO}} \left[\ell(x, s) - \ell(x', s) \right] \le \varepsilon_{\text{stab}}, \quad x = \texttt{ALGO}(S),\ x' = \texttt{ALGO}(S')$$

Theorem Stable algorithms generalize in expectation

Assume $\ell(\cdot,\cdot) \in [0, M]$ then if $\texttt{ALGO}$ is $\varepsilon_{\text{stab}}$-uniformly stable, then with probability at least $1-\delta$ (over the data and the algorithm's randomness)

$$f_\infty(x) \le f_n(x) + \underbrace{\sqrt{\frac{6Mn\varepsilon_{\text{stab}} + M^2}{2n\delta}}}_{\varepsilon_{\text{gen}}}, \quad x = \texttt{ALGO}(S) \qquad |S| = n$$

Reasonable for classification

Recall

$$f_n(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \ell(x, s_i)$$

$$f_\infty(x) \stackrel{\text{def}}{=} \mathbb{E}_{s \sim \mathbb{D}}[\ell(x, s)]$$

Informally, call an algorithm "stable" if $\varepsilon_{\text{stab}} = \mathcal{O}(1/n)$

SGD (w/ early stopping) is stable

Hardt, Recht and Singer, *ICML*, 2016

Definition Uniformly Stable in Expectation

A randomized algorithm ALGO is $\varepsilon_{\text{stab}}$-uniformly stable if for all datasets S and S' (both of size n) that differ in at most one example,

$$\sup_{s \in \mathcal{S}} \mathbb{E}_{\texttt{ALGO}} \left[\ell(x, s) - \ell(x', s) \right] \le \varepsilon_{\text{stab}}, \quad x = \texttt{ALGO}(S),\ x' = \texttt{ALGO}(S')$$

Theorem Stable algorithms generalize in expectation

Assume $\ell(\cdot,\cdot) \in [0, M]$ then if ALGO is $\varepsilon_{\text{stab}}$-uniformly stable, then with probability at least $1-\delta$ (over the data and the algorithm's randomness)

$$f_\infty(x) \le f_n(x) + \underbrace{\sqrt{\frac{6Mn\varepsilon_{\text{stab}} + M^2}{2n\delta}}}_{\varepsilon_{\text{gen}}}, \ x = \texttt{ALGO}(S) \qquad |S| = n$$

Theorem SGD is stable (... hence generalizes)

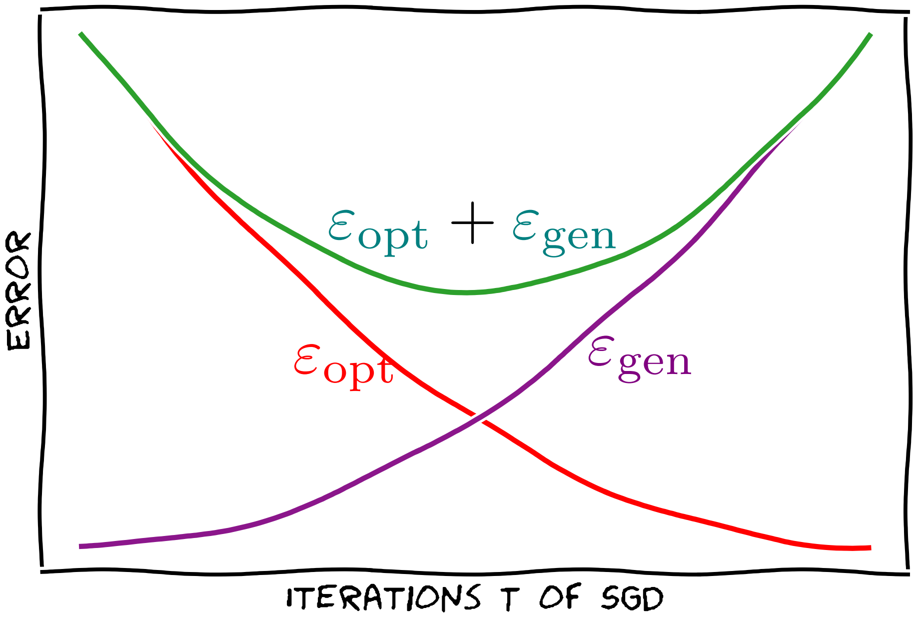
Assume $(\forall s)\, x \mapsto \ell(x, s) \in [0, 1]$ and is ρ-Lipschitz and its gradient is β-Lipschitz.

Then SGD for T iterations with stepsize $\eta_t = c/t$ is uniformly stable in expectation, with

$$\varepsilon_{\text{stab}} \le \frac{1 + 1/\beta c}{n-1} \left(2c\rho^2\right)^{\frac{1}{\beta c + 1}} T^{\frac{\beta c}{\beta c + 1}}$$

*Note: if $T < n$, then this is not new, since it essentially falls under stochastic approximation (SA) theory (no duplicate samples)

Putting it altogether



$$f_\infty(x) \le f_n(x) + \underbrace{\sqrt{\frac{6Mn\varepsilon_{\text{stab}} + M^2}{2n\delta}}}_{\varepsilon_{\text{gen}}}, \quad x = \texttt{ALGO}(S)$$

$$f_\infty(x_T) = \underbrace{f_\infty(x_T) - f_n(x_T)}_{\varepsilon_{\text{gen}}} + \underbrace{f_n(x_T)}_{\varepsilon_{\text{opt}}}$$

Theorem Combined SGD bound [Madden, Dall'Anese, B. 2021]

Theorem 10. *Assume $\ell(x,s) \in [0,M]$ for all x and s. Assume $\ell(\cdot,s)$ is ρ-Lipschitz and L-smooth for all s. Assume f is μ-PL. Let $\kappa = L/\mu$. Assume $\nabla f(x) - g(x,1)$ is centered and $\sigma/\sqrt{d}$-sub-Gaussian for all x. Let $b_t = b$, $c = 1/(\mu + L)$, and $T = \Theta(n/b)$. Then, T iterations of SGD with $\eta_t = c/(t+1)$ satisfies, w.p. $\ge 1-\delta$ over S and (I_t) for all $\delta \in (0, 1/e)$,*

$$f_\infty(x_T) - \min_{x'} f_n(x') = \mathcal{O}\left(\frac{b^{1/(2\kappa+2)}}{n^{1/(2\kappa+2)}\sqrt{\delta}} + \frac{\log(1/\delta)}{b^{1-1/(\kappa+1)} n^{1/(\kappa+1)}}\right)$$

bT is the number of epochs

$1/\sqrt{\delta}$ isn't "high-probability" but we can boost, and beats usual $1/\delta$ bound

Polyak-Łojasiewicz inequality

Definition f is μ-PL if $(\forall x)\ \dfrac{1}{2}\|\nabla f(x)\|^2 \geq \mu\left(f(x) - \min_{x'} f(x')\right)$

popularized by
Karimi, Nutini, Schmidt '16

Strongly convex implies PL...
✓ but there are also non-strongly-convex PL functions
and even **non-convex** PL functions

Ex.: $f(x) = \dfrac{1}{2}\|Ax - b\|^2$

even if A isn't injective

PL implies stationary points are global minimizers,
and gradient descent converges at a linear rate,
but does not prove uniqueness of minimizers

✗ PL is **not** closed under nonnegative sums, unlike (strong) convexity
✗ PL does **not** play nicely with constraints

For sufficiently wide neural nets, f_n is locally μ-PL with constant $\mu = \Omega(1/n^2)$

Allen-Zhu, Li, and Song, 1811.03962 '18 and *NeurIPS* '19

What's wrong with early-stopping?

2016

Train faster, generalize better:
Stability of stochastic gradient descent

Moritz Hardt* Benjamin Recht† Yoram Singer‡

February 9, 2016

"In a nutshell, our results establish that:

Any model trained with stochastic gradient method in a reasonable amount of time attains small generalization error."

What's wrong with early-stopping?

2016

Train faster, generalize better:
Stability of stochastic gradient descent

Moritz Hardt* Benjamin Recht† Yoram Singer‡

February 9, 2016

"In a nutshell, our results establish that:

Any model trained with stochastic gradient method in a reasonable amount of time attains small generalization error."

Math wasn't wrong... but perhaps not that useful:

2017

UNDERSTANDING DEEP LEARNING REQUIRES RE-THINKING GENERALIZATION

Chiyuan Zhang*
Massachusetts Institute of Technology
chiyuan@mit.edu

Samy Bengio
Google Brain
bengio@google.com

Moritz Hardt
Google Brain
mrtz@google.com

Benjamin Recht†
University of California, Berkeley
brecht@berkeley.edu

Oriol Vinyals
Google DeepMind
vinyals@google.com

"Even optimization on random labels remains easy. In fact, training time increases only by a small constant factor compared with training on the true labels"

2021

DOI:10.1145/3446776

Understanding Deep Learning (Still) Requires Rethinking Generalization

By Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals

MARCH 2021 | VOL. 64 | NO. 3 | COMMUNICATIONS OF THE ACM

Their experiment

Take the CIFAR10 dataset

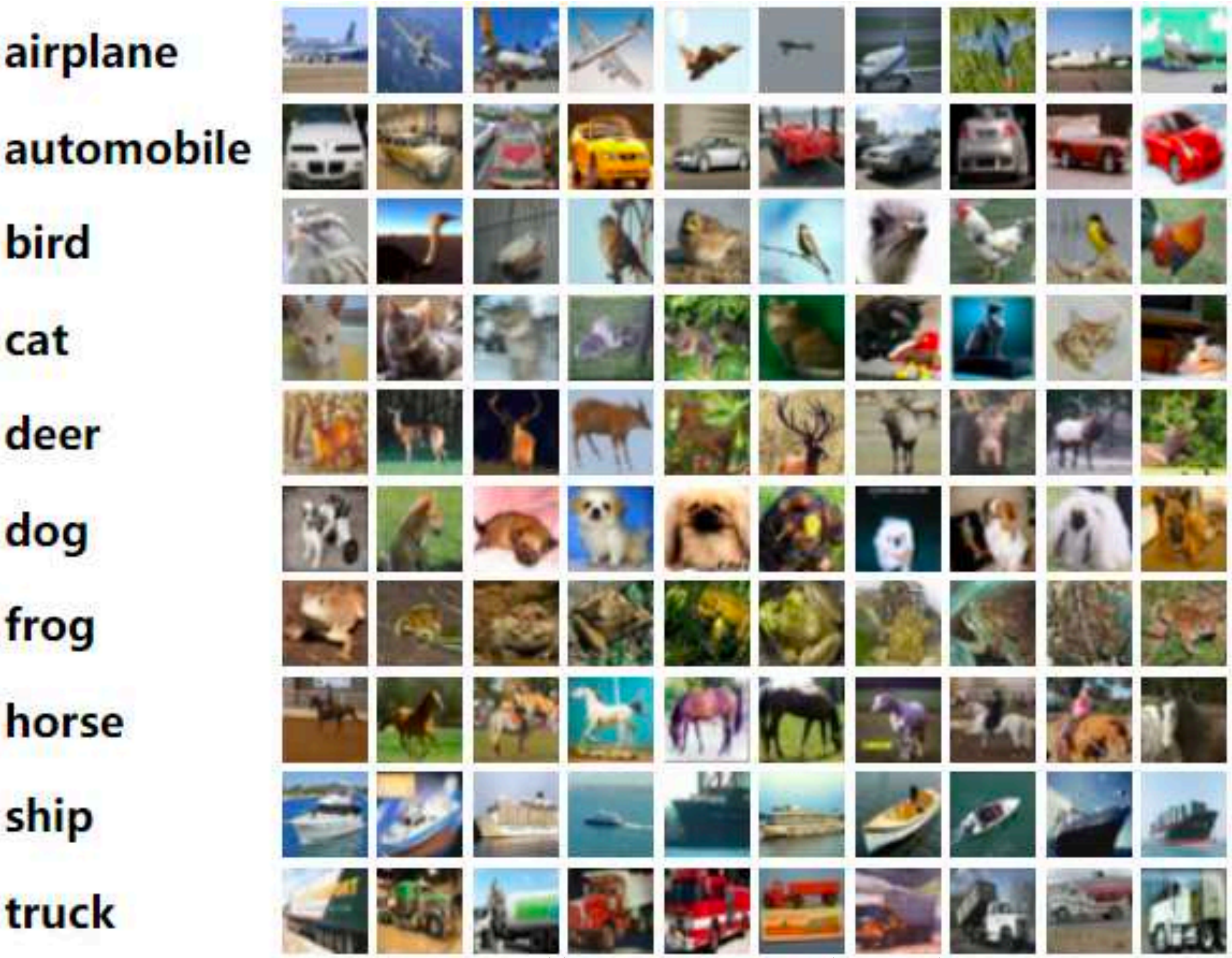


image credit: https://paperswithcode.com/dataset/cifar-10

Learning Multiple Layers of Features from Tiny Images, Alex Krizhevsky, '09

10 possible labels, n=60000, 32x32 images

Now **corrupt** the data:

For each datapoint, give it a random label

 **automobile** → **frog**

It's still possible to have 0 **training** error

... but cannot beat 10% **testing** error

Results: test accuracy

Train on CIFAR10 with *Inception* or *AlexNet*:

- **normal labels**
  - 75-90% test accuracy
- **random labels**
  - 10% test accuracy
  - *No learning is possible: test accuracy is no better than random guessing*

So far, this is not surprising

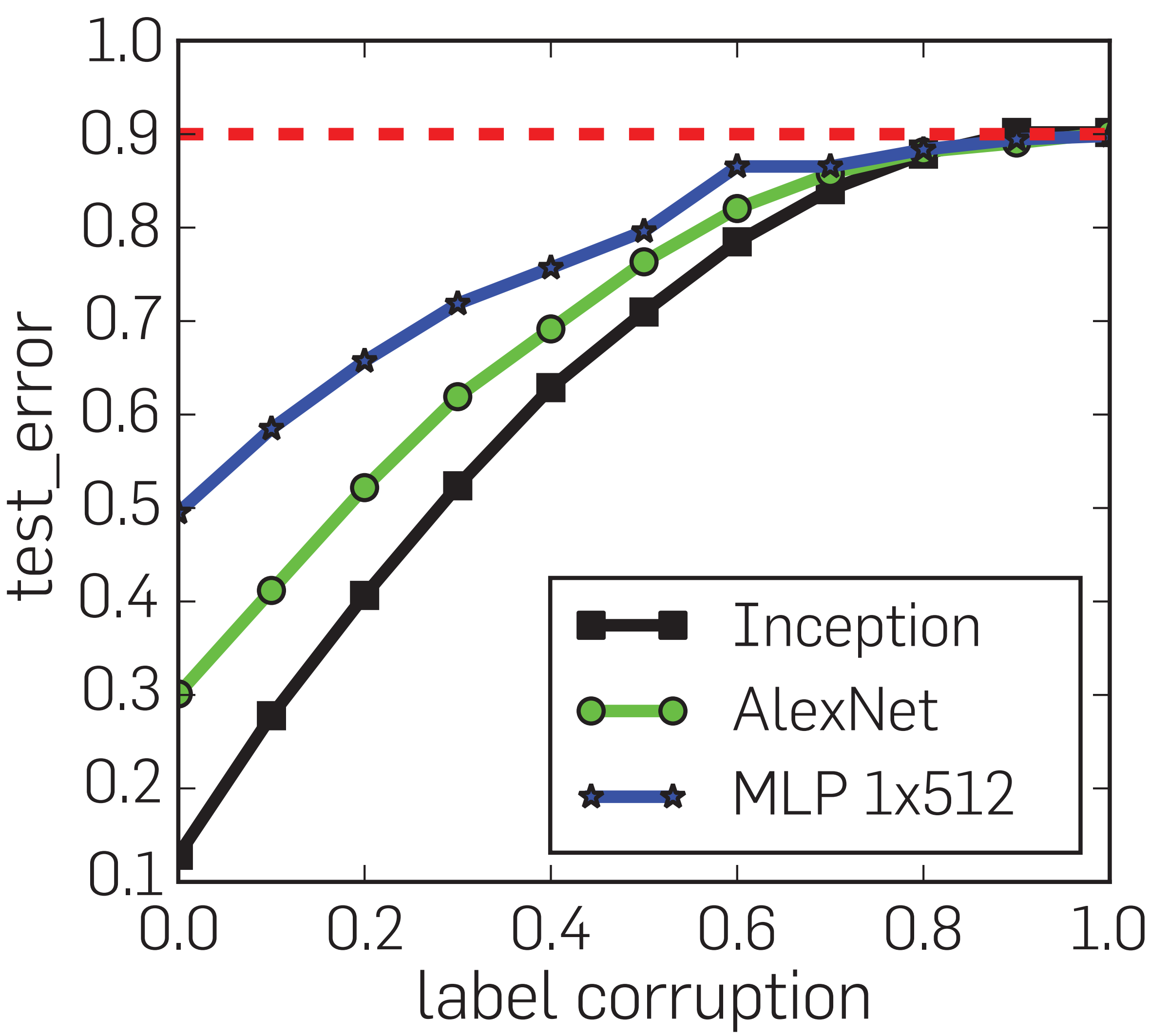


figure credit: Zhang et al. 2017

Results: training accuracy

Both **normal** labels and **random** labels have 100% **training** accuracy

...and for the **random** labels, convergence is still pretty quick (maybe 3x slower)

No useful stability bound for SGD at 15k steps is possible, since we know learning isn't possible for random labels

... but this means no useful bounds for true labels either.

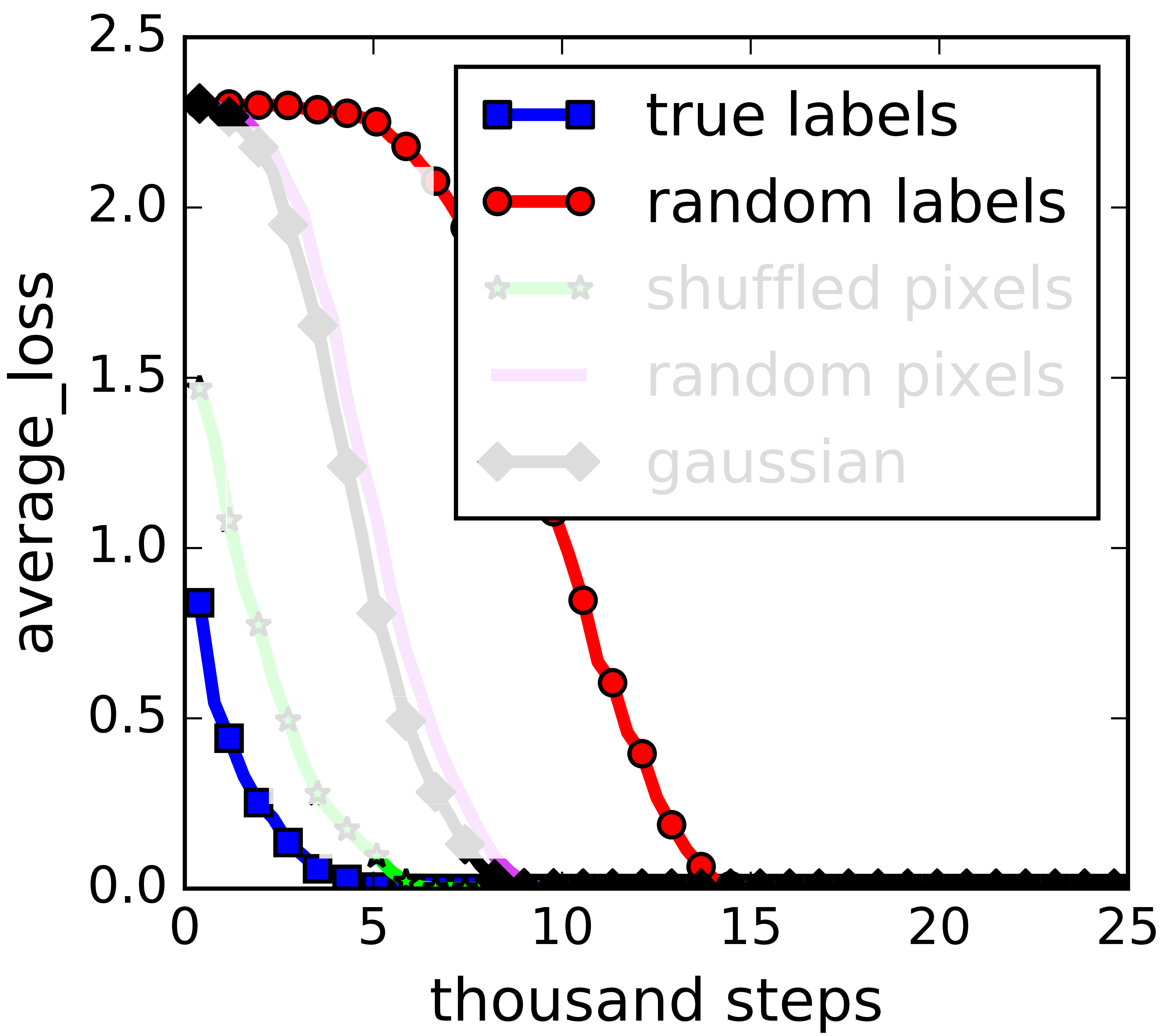


figure credit: Zhang et al. 2017

So then what?

Definition Uniformly Stable in Expectation*

A randomized algorithm ALGO is $\varepsilon_{\text{stab}}$-uniformly stable if for all datasets S and S' (both of size n) that differ in at most one example,

$$\sup_{s\in\mathcal{S}} \mathbb{E}_{\texttt{ALGO}}\left[\ell(x,s)-\ell(x',s)\right] \le \varepsilon_{\text{stab}}, \quad x=\texttt{ALGO}(S),\ x'=\texttt{ALGO}(S')$$

It wasn't just that their early-stopping analysis wasn't tight

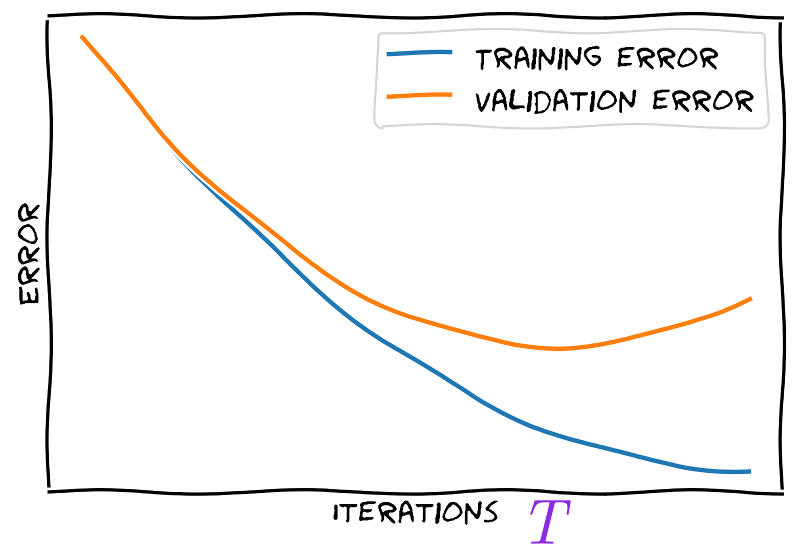
Possible fix #1: use a relaxed (non-uniform) notion of stability

Possible fix #2: take another approach (not using stability)

Possible fix #3: change what we mean by "algorithm" and "early stopping"

A proposed approach

preliminary work, joint with Aurelien Lucchi (U. Basel)



What's the **stopping condition** for SGD in the real-world?

People do not used a fixed number of epochs/iterations.

Instead, they look at when **validation error** starts to go up

New perspective: think of the **validation data** as part of the training data.

T becomes a random variable

<u>Fixes a few problems</u>: with **random** labels, the algorithm would stop (almost) immediately

... **before** it has time to memorize the training data.

It still "generalizes" (testing error = training error), but training error is high

Conclusion

- High-probability results are nice to have
- SGD *naturally* has high-probability results, no need to do probability amplification
- Assumptions are tricky but important (need to avoid vacuous results!)

- SGD with early stopping will allow you to **generalize**
- ... but current theory is not sharp enough to be useful
- Improved analysis is ongoing

Thanks for listening