Stochastic Quasi-Gradient Methods: Variance Reduction via Jacobian Sketching

Peter Richtárik

Randomized Numerical Linear Algebra and Applications
(Program: Foundations of Data Science)
Simons Institute for the Theory of Computing, UC Berkeley
September 24-27, 2018

Stochastic Quasi-Gradient Methods: Variance Reduction via Jacobian Sketching

Robert M. Gower* Peter Richtárik† Francis Bach‡

April 25, 2018

Abstract

We develop a new family of variance reduced stochastic gradient descent methods for minimizing the average of a very large number of smooth functions. Our method—JacSketch—is motivated by novel developments in randomized numerical linear algebra, and operates by maintaining a stochastic estimate of a Jacobian matrix composed of the gradients of individual functions. In each iteration, JacSketch efficiently updates the Jacobian matrix by first obtaining a random linear measurement of the true Jacobian through (cheap) sketching, and then projecting the previous estimate onto the solution space of a linear matrix equation whose solutions are consistent with the measurement. The Jacobian estimate is then used to compute a variance-reduced unbiased estimator of the gradient, followed by a stochastic gradient descent step. Our strategy is analogous to the way quasi-Newton methods maintain an estimate of the Hessian, and hence our method can be seen as a *stochastic quasi-gradient method*. Indeed, quasi-Newton methods project the current Hessian estimate onto a solution space of a linear equation consistent with a certain linear (but non-random) measurement of the true Hessian. Our method can also be seen as stochastic gradient descent applied to a *controlled stochastic optimization reformulation* of the original problem, where the control comes from the Jacobian estimates.

We prove that for smooth and strongly convex functions, JacSketch converges linearly with a meaningful rate dictated by a single convergence theorem which applies to general sketches. We also provide a refined convergence theorem which applies to a smaller class of sketches, featuring a novel proof technique based on a *stochastic Lyapunov function*. This enables us to obtain sharper complexity results for variants of JacSketch with importance sampling. By specializing our general approach to specific sketching strategies, JacSketch reduces to the celebrated stochastic average gradient (SAGA) method, and its several existing and many new minibatch, reduced memory, and importance sampling variants. Our rate for SAGA with importance sampling is the current best-known rate for this method, resolving a conjecture by Schmidt et al (2015). The rates we obtain for minibatch SAGA are also superior to existing rates. Moreover, we obtain the first minibatch SAGA method with importance sampling.

*Télécom ParisTech, France.

†King Abdullah University of Science and Technology (KAUST), Saudi Arabia — University of Edinburgh, United Kingdom — Moscow Institute of Physics and Technology (MIPT), Russia.

‡INRIA - ENS - PSL Research University, France.

Contents

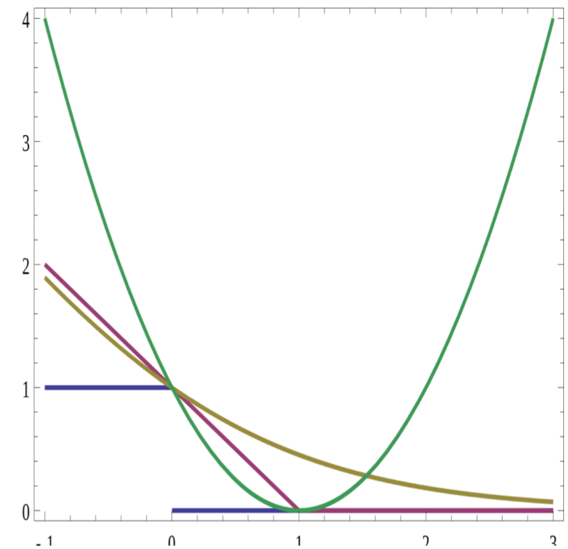
Outline

1. Introduction
2. Jacobian Sketching
3. Controlled Stochastic Reformulations
4. JacSketch and SAGA
5. Iteration Complexity of JacSketch
6. Experiments

1. Introduction

Finite Sum Minimization Problem

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$



Data vector

Label

L2 regularizer

L2 regularized least squares
(ridge regression)

$$f_i(x) = \frac{1}{2}(a_i^\top x - y_i)^2 + \frac{\lambda}{2}\|x\|^2$$

L2 regularized logistic regression

$$f_i(x) = \frac{1}{2} \log\left(1 + e^{-y_i a_i^\top x}\right) + \frac{\lambda}{2}\|x\|^2$$

Stochastic Gradient Methods

Current iterate

Stepsize

$$x^{k+1} = x^k - \alpha g^k$$

Next iterate

Unbiased estimator of the gradient:

$$\mathbb{E}\left[g^k\right] = \nabla f(x^k)$$

Variance Matters

$$\mathbb{V}\left[g^k\right] := \mathbb{E}\left[\|g^k - \underbrace{\nabla f(x^k)}_{\mathbb{E}\left[g^k\right]}\|^2\right]$$

Gradient Descent (GD)

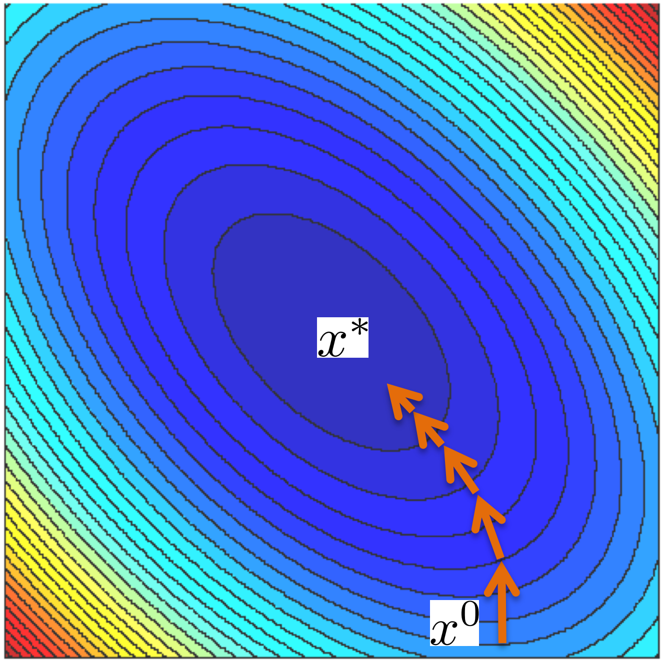
$$g^k \leftarrow \nabla f(x^k) \quad \Rightarrow \quad \mathbb{V}\left[g^k\right] = 0$$

Stochastic Gradient Descent (SGD)

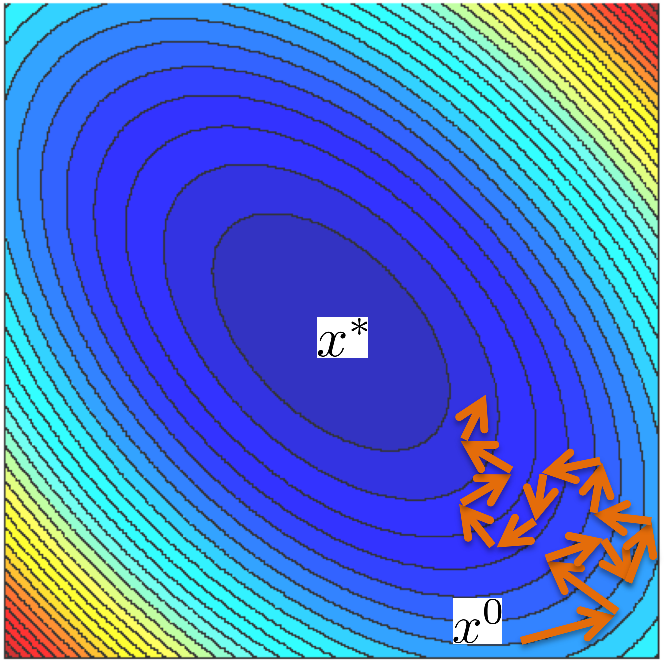
$$g^k \leftarrow \nabla f_{\color{red}{i}}(x^k) \quad \Rightarrow \quad \mathbb{V}\left[g^k\right] = \mathrm{BIG}$$

GD vs SGD

Gradient Descent (GD)



Stochastic Gradient Descent (SGD)



Variance Reduction

	Decreasing stepsizes	Mini-batching	Importance sampling	Adjusting the direction
How does it work?	Scaling down the noise	More samples, less variance	Sample more important data (or parameters) more often	Duality (SDCA) or Control Variate (SVRG, S2GD, SAGA)
CONS:	Slow down; Hard to tune the stepsize	More work per iteration	Might overfit probabilities to outliers	A bit (SVRG, S2GD) or a lot (SDCA, SAGA) more memory needed
PROS:	Still converges Widely known	Parallelizable	Improved condition number	Improved dependence on epsilon

All tricks can be combined!

2. Jacobian Sketching

(JacSketch as a Stochastic Quasi-Gradient Method)

Robert M Gower, Peter Richtárik and Francis Bach
Stochastic Quasi-Gradient Methods: Variance Reduction via Jacobian Sketching
arXiv:1805.02632, 2018

Lift and Sketch

Lift and Sketch

1 LIFT

$$F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix} \in \mathbb{R}^n$$

Jacobian of *F*

$$\nabla \mathbf{F}(x) = [\nabla f_1(x), \nabla f_2(x), \ldots, \nabla f_n(x)] \in \mathbb{R}^{d \times n}$$

2 SKETCH

*i*th unit basis vector

Vector of all ones

$$\nabla \mathbf{F}(x) e_i = \nabla f_i(x)$$

Leads to Stochastic Gradient Descent

$$\frac{1}{n} \nabla \mathbf{F}(x) e = \nabla f(x)$$

Leads to Gradient Descent

Introducing General Sketches

We would like to solve the linear matrix equation:

$$d\left\{\underbrace{\mathbf{J}}_{n}\right. = \nabla \mathbf{F}(x^k)$$

Too expensive to solve!

Solve a random linear matrix equation instead:

$$\underbrace{\mathbf{J}\mathbf{S}_k}_{q} = \underbrace{\nabla \mathbf{F}(x^k)\mathbf{S}_k}_{\text{Jacobian sketch}}$$

Random matrix

$\mathbf{S}_k \sim \mathcal{D}$

Has many solutions: which solution to pick?

Sketch and Project

Sketch and Project

New Jacobian estimate

Current Jacobian estimate

Frobenius norm

$$\mathbf{J}^{k+1} := \arg\min_{\mathbf{J}\in\mathbb{R}^{d\times n}} \|\mathbf{J}-\mathbf{J}^k\|$$

$$\text{subject to} \quad \mathbf{J}\mathbf{S}_k = \nabla\mathbf{F}(x^k)\mathbf{S}_k$$

Random LME ensuring consistency with Jacobian sketch

Solution:

$$\mathbf{J}^{k+1} = \mathbf{J}^k + (\nabla\mathbf{F}(x^k) - \mathbf{J}^k)\mathbf{\Pi}_{\mathbf{S}_k}$$

$$\mathbf{\Pi}_{\mathbf{S}_k} \overset{\text{def}}{=} \mathbf{S}_k\left(\mathbf{S}_k^\top \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top$$

Sketch and Project I

Original sketch and project

- 2017 IMA Fox Prize (2nd Prize) in Numerical Analysis
- Most downloaded SIMAX paper (2017)

Robert Mansel Gower and P.R.
Randomized Iterative Methods for Linear Systems
SIAM J. Matrix Analysis and Applications 36(4):1660-1690, 2015

Removal of full rank assumption + duality

Robert Mansel Gower and P.R.
Stochastic Dual Ascent for Solving Linear Systems
arXiv:1512.06890, 2015

Inverting matrices & connection to quasi-Newton updates

Robert Mansel Gower and P.R.
Randomized Quasi-Newton Methods are Linearly Convergent Matrix Inversion Algorithms
SIAM J. on Matrix Analysis and Applications 38(4), 1380-1409, 2017

Computing the pseudoinverse

Robert Mansel Gower and P.R.
Linearly Convergent Randomized Iterative Methods for Computing the Pseudoinverse
arXiv:1612.06255, 2016

Application to machine learning

Robert Mansel Gower, Donald Goldfarb and P.R.
Stochastic Block BFGS: Squeezing More Curvature out of Data
ICML 2016

Sketch and project revisited: stochastic reformulations of linear systems

P.R. and Martin Takáč
Stochastic Reformulations of Linear Systems: Algorithms and Convergence Theory
arXiv:1706.01108, 2017

Sketch and Project II

Linear convergence of the stochastic heavy ball method

Nicolas Loizou and P.R.
Momentum and stochastic momentum for stochastic gradient, Newton, proximal point and subspace descent methods
arXiv:1712.09677, 2017

Stochastic projection methods for convex feasibility

Ion Necoara, Andrei Patrascu and P.R.
Randomized projection methods for convex feasibility problems: conditioning and convergence rates
arXiv:1801.04873, 2018

Stochastic spectral & conjugate descent

Dmitry Kovalev, Eduard Gorbunov, Elnur Gasanov and P.R.
Stochastic Spectral and Conjugate Descent Methods
NIPS 2018

Accelerated stochastic matrix inversion

Robert M. Gower, Filip Hanzely, P.R. and Sebastian Stich
Accelerated Stochastic Matrix Inversion: General Theory and Speeding up BFGS Rules for Faster Second-Order Optimization
NIPS 2018

SAGD: a “strange” special case of JacSketch

Adel Bibi, Alibek Sailanbayev, Bernard Ghanem, Robert Mansel Gower and P.R.
Improving SAGA via a Probabilistic Interpolation with Gradient Descent
arXiv:1806.05633, 2018

Gradient sketching

Filip Hanzely, Konstantin Mishchenko and P.R.
SEGA: Variance Reduction via Gradient Sketching
NIPS 2018

Constructing an Unbiased Gradient Estimate

Gradient Estimate

Bias-correcting random variable:

$\mathbb{E}_{\mathbf{S}_k \sim \mathcal{D}}\left[\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} e\right] = e$

Average of the columns of $\mathbf{J}^k$

Average of the columns of $\mathbf{J}^{k+1}$

$$
\begin{aligned}
g^k &:= (1-\theta_{\mathbf{S}_k})\boxed{\frac{1}{n}\mathbf{J}^k e} + \theta_{\mathbf{S}_k}\boxed{\frac{1}{n}\mathbf{J}^{k+1} e} \\
&= \frac{1}{n}\mathbf{J}^k e + \frac{1}{n}(\nabla \mathbf{F}(x^k) - \mathbf{J}^k)\theta_{\mathbf{S}_k}\mathbf{\Pi}_{\mathbf{S}_k} e
\end{aligned}
$$

Unbiased estimator of the gradient

$\mathbb{E}_{\mathbf{S}_k \sim \mathcal{D}}\left[g^k\right] = \nabla f(x^k)$

$\theta_{\mathbf{S}_k} \equiv 1 \Rightarrow g^k = \frac{1}{n}\mathbf{J}^{k+1} e$

SAG: Stochastic Average Gradient

Le Roux, Schmidt and Bach

A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets

NIPS 2012

3. Stochastic Reformulation

(JackSketch as SGD Applied to Controlled Stochastic Reformulation)

Simple Stochastic Reformulation

$$F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix} \in \mathbb{R}^n$$

Reformulation

Bias-correcting random variable:

$$\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[\theta_{\mathbf{S}}\mathbf{\Pi}_{\mathbf{S}}e\right] = e$$

$$f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x) = \frac{1}{n}\langle F(x), e\rangle \quad = \quad \frac{1}{n}\langle F(x), \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[\theta_{\mathbf{S}}\mathbf{\Pi}_{\mathbf{S}}e\right]\rangle$$

Linearity of expectation

$$= \quad \underbrace{\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[\frac{1}{n}\langle F(x), \theta_{\mathbf{S}}\mathbf{\Pi}_{\mathbf{S}}e\rangle\right]}_{=:f_{\mathbf{S}}(x)}$$

$$f_{\mathbf{S}}(x) = \sum_{i=1}^n \left(\tfrac{1}{n}\theta_{\mathbf{S}}\mathbf{\Pi}_{\mathbf{S}}e\right)_i f_i(x)$$

Original problem

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{n}\sum_{i=1}^n f_i(x)$$

Simple stochastic reformulation

$$\min_{x\in\mathbb{R}^d} f(x) = \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[f_{\mathbf{S}}(x)\right]$$

We are minimizing the expectation over **random linear combinations** of the original functions

SGD Applied to Simple Stochastic Reformulation

$\mathbf{S}_k \sim \mathcal{D}$

$$x^{k+1} = x^k - \alpha \nabla f_{\mathbf{S}_k}(x^k)$$

$\mathbf{S} \equiv \mathbf{I}$ $\theta_{\mathbf{S}} \equiv 1$ ➡ Gradient descent

$$x^{k+1} = x^k - \alpha \nabla f(x^k)$$

$\mathbb{P}(\mathbf{S} = e_i) = p_i$ $\theta_{e_i} \equiv \frac{1}{p_i}$ ➡ Non-uniform SGD

$$x^{k+1} = x^k - \frac{\alpha}{n p_i} \nabla f_i(x^k)$$

$\mathbb{P}\left(\mathbf{S} = e_S := \sum_{i \in S} e_i\right) = p_S$ $\theta_{e_S} \equiv \frac{1}{c_1 p_S}$ ➡ Non-uniform minibatch SGD

$$x^{k+1} = x^k - \frac{\alpha}{n c_1 p_{S_k}} \sum_{i \in S_k} \nabla f_i(x^k)$$

Controlled Stochastic Reformulation

Adding Control Variate to Reduce Variance

$$\min_{x\in\mathbb{R}^d} f(x) = \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[f_{\mathbf{S},\mathbf{J}}(x)\right]$$

$$f_{\mathbf{S},\mathbf{J}}(x) \overset{\text{def}}{=} f_{\mathbf{S}}(x) - z_{\mathbf{S},\mathbf{J}}(x) + \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[z_{\mathbf{S},\mathbf{J}}(x)\right]$$

Recall:

$$f_{\mathbf{S}}(x) = \frac{1}{n}\langle F(x), \theta_{\mathbf{S}}\mathbf{\Pi}_{\mathbf{S}}e\rangle$$

$$z_{\mathbf{S},\mathbf{J}}(x) = \frac{1}{n}\langle \mathbf{J}^\top x, \theta_{\mathbf{S}}\mathbf{\Pi}_{\mathbf{S}}e\rangle$$

JacSketch = SGD Applied Controlled Stochastic Reformulation

$\mathbf{S}_k \sim \mathcal{D}$

$$x^{k+1} = x^k - \alpha \nabla f_{\mathbf{S}_k, \mathbf{J}^k}(x^k)$$

Sketch and project

$\mathbf{J}^{k+1} = \mathbf{J}^k + (\nabla \mathbf{F}(x^k) - \mathbf{J}^k)\mathbf{\Pi}_{\mathbf{S}_k}$

Variance of the Stochastic Gradient

Weighted Frobenius norm

$$\|\mathbf{A}\|_{\mathbf{B}} \stackrel{\text{def}}{=} \sqrt{\mathrm{Tr}\left(\mathbf{A}\mathbf{B}\mathbf{A}^\top\right)}$$

Theorem $$\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[\|\nabla f_{\mathbf{S},\mathbf{J}}(x) - \nabla f(x)\|^2\right] = \frac{1}{n^2}\|\mathbf{J} - \nabla \mathbf{F}(x)\|^2_{\mathbf{B}}$$

$$\mathbf{B} = \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[v_{\mathbf{S}} v_{\mathbf{S}}^\top\right]$$

$$v_{\mathbf{S}} \stackrel{\text{def}}{=} (\mathbf{I} - \theta_{\mathbf{S}}\mathbf{\Pi}_{\mathbf{S}})e$$

$\theta_{\mathbf{S}}$ is bias correcting:

$$\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[v_{\mathbf{S}}\right] = 0$$

$$\begin{aligned}
\lambda_{\max}(\mathbf{B}) &= \lambda_{\max}\left(\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[v_{\mathbf{S}} v_{\mathbf{S}}^\top\right]\right) \\
&\leq \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[\lambda_{\max}\left(v_{\mathbf{S}} v_{\mathbf{S}}^\top\right)\right] \\
&= \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[\|v_{\mathbf{S}}\|^2\right].
\end{aligned}$$

$$\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[\|\nabla f_{\mathbf{S},\mathbf{J}}(x) - \nabla f(x)\|^2\right] \leq \frac{\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[\|v_{\mathbf{S}}\|^2\right]}{n^2}\|\mathbf{J} - \nabla \mathbf{F}(x)\|^2$$

Variance of $v_{\mathbf{S}}$ as an estimator of 0

4. JacSketch and SAGA

Algorithm: JacSketch

Algorithm 1 JacSketch: Variance Reduced Gradient Method via Jacobian Sketching

Input: $(\mathcal{D}, \mathbf{W}, \theta_{\mathbf{S}})$
Initialize: $x^0 \in \mathbb{R}^d$, Jacobian estimate $\mathbf{J}^0 \in \mathbb{R}^{d \times n}$, stepsize $\alpha > 0$
for $k = 0, 1, 2, \dots$ **do**
    Sample a fresh copy $\mathbf{S}_k \sim \mathcal{D}$
    Calculate $\nabla \mathbf{F}(x^k)\mathbf{S}_k$ ▷ Sketch the Jacobian
    $\mathbf{J}^{k+1} = \mathbf{J}^k + (\nabla \mathbf{F}(x^k) - \mathbf{J}^k)\mathbf{\Pi}_{\mathbf{S}_k} = \mathbf{J}^k(\mathbf{I} - \mathbf{\Pi}_{\mathbf{S}_k}) + \nabla \mathbf{F}(x^k)\mathbf{\Pi}_{\mathbf{S}_k}$ ▷ Update Jacobian estimate
    $g^k = \frac{1}{n}\mathbf{J}^k e + \frac{\theta_{\mathbf{S}_k}}{n}(\nabla \mathbf{F}(x^k) - \mathbf{J}^k)\mathbf{\Pi}_{\mathbf{S}_k} e = \frac{1-\theta_{\mathbf{S}_k}}{n}\mathbf{J}^k e + \frac{\theta_{\mathbf{S}_k}}{n}\mathbf{J}^{k+1} e$ ▷ Update gradient estimate
    $x^{k+1} = x^k - \alpha g^k$ ▷ Take a step

Initialize: $x^0 \in \mathbb{R}^d$, $\mathbf{J}^0 \in \mathbb{R}^{d \times n}$

Iterate:

Draw $\mathbf{S}_k \sim \mathcal{D}$

$$\mathbf{\Pi}_{\mathbf{S}_k} \overset{\text{def}}{=} \mathbf{S}_k \left(\mathbf{S}_k^\top \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top$$

Update the Jacobian estimate:

$$\mathbf{J}^{k+1} = \mathbf{J}^k + (\nabla \mathbf{F}(x^k) - \mathbf{J}^k)\mathbf{\Pi}_{\mathbf{S}_k}$$

Update the gradient estimate:

$$g^k = \frac{1}{n}\mathbf{J}^k e + \frac{1}{n}(\nabla \mathbf{F}(x^k) - \mathbf{J}^k)\theta_{\mathbf{S}_k}\mathbf{\Pi}_{\mathbf{S}_k} e$$

Take a gradient step:

$$x^{k+1} = x^k - \alpha g^k$$

$$\mathbb{E}_{\mathbf{S}_k \sim \mathcal{D}}\left[\theta_{\mathbf{S}_k}\mathbf{\Pi}_{\mathbf{S}_k} e\right] = e$$

SAGA as JacSketch

A. Defazio, F. Bach and S. Lacoste-Julien
SAGA: A Fast Incremental Gradient Method with Support for Non-strongly Convex Composite Objectives
NIPS 2014

Minibatch SAGA

$n = 5$

$S_k = \{1, 3, 4\}$

$$\mathbf{S}_k = \mathbf{I}_{:,S_k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{J}^{k+1}_{:i} = \begin{cases} \mathbf{J}^k_{:i} & i \notin S_k \\ \nabla f_i(x^k) & i \in S_k \end{cases}$$

$$g^k = \frac{1}{n}\mathbf{J}^k e + \frac{\theta_{\mathbf{S}_k}}{n} \sum_{i \in S_k} \left(\nabla f_i(x^k) - \mathbf{J}^k_{:i}\right)$$

$$x^{k+1} = x^k - \alpha g^k$$

5. Iteration Complexity of JacSketch

General Theorem

First Main Result (Theorem 3.6)

Sketch residual

$\rho := \lambda_{\max}\left(\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[(\mathbf{I}-\theta_{\mathbf{S}}\mathbf{\Pi}_{\mathbf{S}})ee^\top(\mathbf{I}-\theta_{\mathbf{S}}\mathbf{\Pi}_{\mathbf{S}})^\top\right]\right)$

Expected smoothness constants

$$k \geq \max\left\{\frac{1}{\kappa} + \frac{\frac{\rho}{n^2}4\mathcal{L}_2}{\kappa\mu}, \frac{4\mathcal{L}_1}{\mu}\right\}\log\left(\frac{1}{\epsilon}\right)$$

Stochastic condition number

$\kappa := \lambda_{\min}\left(\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}\left[\mathbf{\Pi}_{\mathbf{S}}\right]\right)$

always: $0 \leq \kappa \leq 1$

Relative error

$$\mathbb{E}\left[\Psi^k\right] \leq \epsilon\Psi^0$$

Strong convexity parameter of *f*

Lyapunov function

$\Psi^k := \|x^k - x^*\|^2 + \frac{\alpha}{2\mathcal{L}_2}\|\mathbf{J}^k - \nabla\mathbf{F}(x^*)\|^2$

Special Cases

Smoothness constant of f

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x-y\|$$

$$f(x) \leq f(y) + \langle \nabla f(y), x-y \rangle + \tfrac{L}{2}\|x-y\|^2$$

1. Gradient Descent

$$\frac{4L}{\mu}\log\left(\frac{1}{\epsilon}\right)$$

Strong convexity parameter of f

2. SAGA with uniform sampling

Worst smoothness constant of f_i

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i\|x-y\|$$

$$L_{\max} := \max_i L_i$$

$$\left(n + \frac{4L_{\max}}{\mu}\right)\log\left(\frac{1}{\epsilon}\right)$$

Special Cases

3. Minibatch SAGA with uniform sampling

$L \leq \mathcal{L}_1 \leq L_{\max}$

$$\max\left\{\frac{n}{\tau} + \frac{n-\tau}{(n-1)\tau}\frac{4L_{\max}}{\mu},\ \frac{4\mathcal{L}_1}{\mu}\right\}\log\left(\frac{1}{\epsilon}\right)$$

Minibatch size

$S =$ random subset of $\{1,2,\ldots,n\}$ of size τ chosen uniformly of random

In this version of JacSketch we sample gradients $\nabla_i f(x)$ for $i \in S$

This is better than the best known bound for minibatch SAGA due to Hofmann, Lucchi, Lacoste-Julien and McWilliams (*NIPS* 2015)

Specialized Theorem

Minibatch Partition Sketch

Partition

$|C_j| = \tau$ for all j

$m = \frac{n}{\tau}$

$$\{1, 2, \ldots, n\} = C_1 \cup C_2 \cup \cdots \cup C_m$$

$$S = C_j \text{ with probability } p_{C_j} > 0$$

Sketch matrix

$$\mathbf{S} = \mathbf{I}_{:,S}$$

Bias-correcting random variable

$$\theta_{\mathbf{S}} = \frac{1}{p_S}$$

Second Main Result (Theorem 5.2)

Smoothness constant of *C-subsampled* function $f_C(x) := \frac{1}{|C|}\sum_{i\in C} f_i(x)$

$$\|\nabla f_C(x) - \nabla f_C(y)\| \leq L_C \|x - y\|$$

Minibatch size

$$k \geq \max_{j=1,2,\ldots,m} \left\{ \frac{1}{p_{C_j}} + \frac{\tau}{n p_{C_j}} \frac{4 L_{C_j}}{\mu} \right\} \log\left(\frac{1}{\epsilon}\right)$$

$p_{C_j} := \mathbb{P}(S = C_j)$

Strong convexity parameter of *f*

$$\mathbb{E}\left[\Psi^k_S\right] \leq \epsilon \mathbb{E}\left[\Psi^0_S\right]$$

Stochastic Lyapunov function

$$\Psi^k_S := \|x^k - x^*\|^2 + \frac{n\alpha}{2\tau L_S}\|\tfrac{1}{n}\mathbf{J}^k e - \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^*)\|^2$$

Special Cases

4. SAGA with importance sampling

$$\left(n + \frac{4\frac{1}{n}\sum_i L_i}{\mu}\right)\log\left(\frac{1}{\epsilon}\right)$$

This resolves a conjecture of
Schmidt, Babanezhad, Ahmed, Defazio, Clifton and Sarkar (*AISTATS* 2015)

5. Minibatch SAGA with importance sampling

$$\left(\frac{n}{\tau} + \frac{4\frac{1}{m}\sum_j L_{C_j}}{\mu}\right)\log\left(\frac{1}{\epsilon}\right)$$

First result on minibatch SAGA with importance sampling

Summary of Complexity Results

ID	Method	Sketch $\mathbf{S} \in \mathbb{R}^{n\times\tau}$ $\mathbf{W} \succ 0$	Iteration complexity ($\times \log\frac{1}{\epsilon}$)	Reference
1	JacSketch	any unbiased any	$\max\left\{\frac{4\mathcal{L}_1}{\mu}, \frac{1}{\kappa} + \frac{4\rho\mathcal{L}_2}{\kappa\mu n^2}\right\}$	Thm 3.6
2	JacSketch (with any probabilities for τ–partition)	$\mathbf{I}_S$ $\mathbf{I}$	$\max_{C\in\mathrm{supp}(S)}\left(\frac{1}{p_C} + \frac{\tau}{np_C}\frac{4L_C}{\mu}\right)$	Thm 5.2
3	Gradient descent	$\mathbf{I}$ $\mathbf{I}$	$\frac{4L}{\mu}$	Thm 3.6 (101)
4	Gradient descent	$\mathbf{I}$ $\mathbf{I}$	$\frac{4L}{\mu}$	Thm 5.2 (130)
5	SAGA (with uniform sampling)	$\mathbf{I}_S$?	$n + \frac{4L_{\max}}{\mu}$	Thm 3.6 (102)
6	SAGA (with uniform sampling)	$\mathbf{I}_S$ $\mathbf{I}$	$n + \frac{4L_{\max}}{\mu}$	Thm 5.2 (131)
7	SAGA (with importance sampling)	$\mathbf{I}_S$ —	no improvement on uniform sampling	Thm 3.6
8	SAGA (with importance sampling)	$\mathbf{I}_S$ $\mathbf{I}$	$n + \frac{4\bar{L}}{\mu}$	Thm 5.2 (133)
9	Minibatch SAGA (τ–uniform sampling)	$\mathbf{I}_S$ $\mathrm{Diag}(w_i)$	$\max\left\{\frac{4L^{\mathcal{G}}_{\max}}{\mu}, \frac{n}{\tau} + \frac{4\rho}{\mu n}\max_i\left(\frac{L_i}{w_i}\right)\right\}$	Thm 3.6 (100)
10	Minibatch SAGA (τ–nice sampling)	$\mathbf{I}_S$ $\mathbf{I}$	$\max\left\{\frac{4L^{\mathcal{G}}_{\max}}{\mu}, \frac{n}{\tau} + \frac{n-\tau}{(n-1)\tau}\frac{4L_{\max}}{\mu}\right\}$	Thm 3.6 (103)
11	Minibatch SAGA (τ–nice sampling)	$\mathbf{I}_S$ $\mathrm{Diag}(L_i)$	$\max\left\{\frac{4L^{\mathcal{G}}_{\max}}{\mu}, \frac{n}{\tau} + \frac{n-\tau}{n\tau}\frac{4(\bar{L}+L_{\max})}{\mu}\right\}$	Thm 3.6 (104)
12	Minibatch SAGA (τ–partition sampling)	$\mathbf{I}_S$ $\mathbf{I}$	$\frac{n}{\tau} + \frac{4L_{\max}}{\mu}$	Thm 3.6 (105)
13	Minibatch SAGA (τ–partition sampling)	$\mathbf{I}_S$ $\mathrm{Diag}(L_i)$	$\frac{n}{\tau} + \frac{4\max_{C\in\mathrm{supp}(S)}\frac{1}{\tau}\sum_{i\in C}L_i}{\mu}$	Thm 3.6 (106)
14	Minibatch SAGA (importance τ–partition sampling)	$\mathbf{I}_S$ $\mathbf{I}$	$\frac{n}{\tau} + \frac{4\frac{1}{\lvert\mathrm{supp}(S)\rvert}\sum_{C\in\mathrm{supp}(S)}L_C}{\mu}$	Thm 5.2 (135)

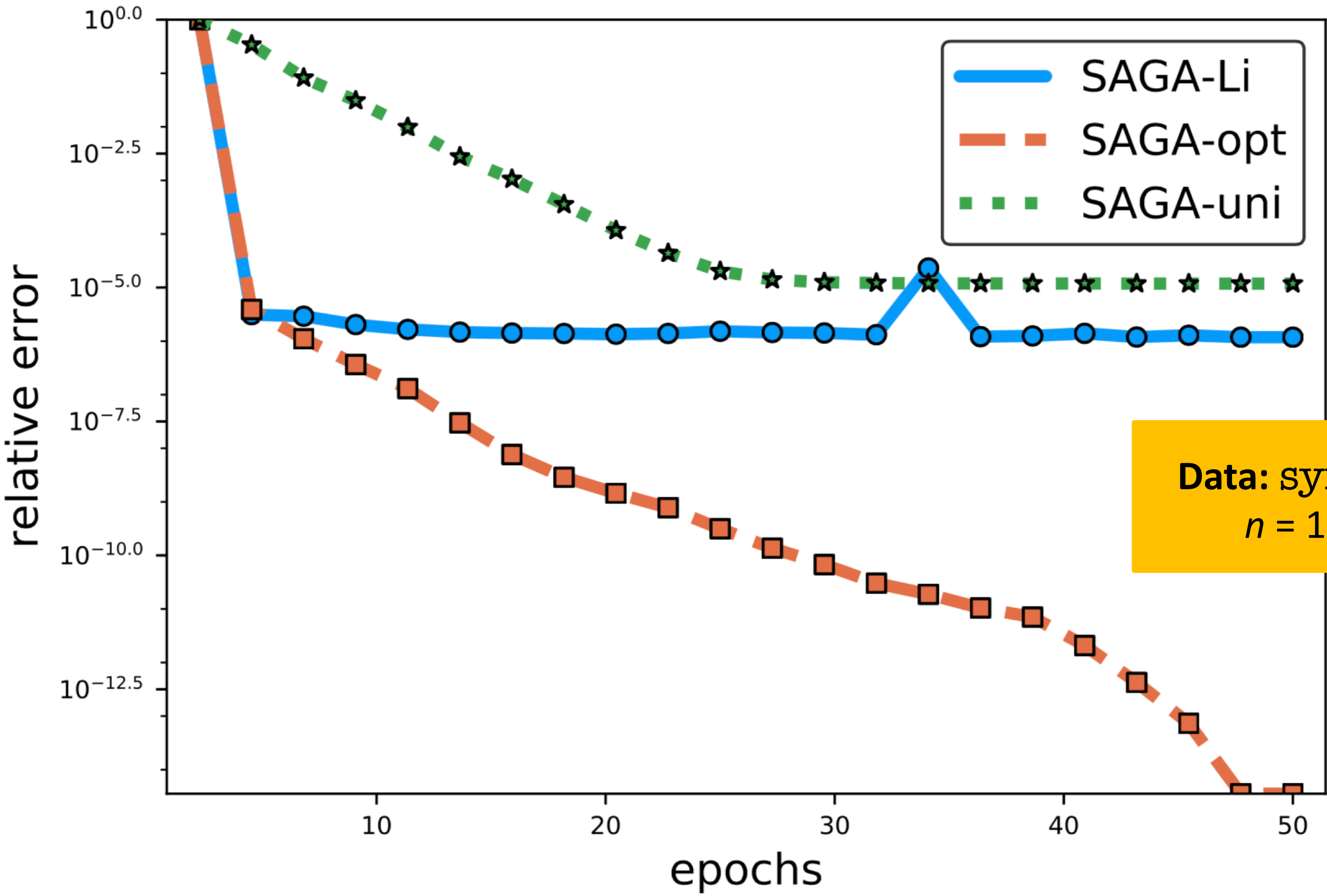
6. Experiments

Ridge Regression

$$f_i(x) = \frac{1}{2}(a_i^\top x - y_i)^2 + \frac{\lambda}{2}\|x\|^2$$

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

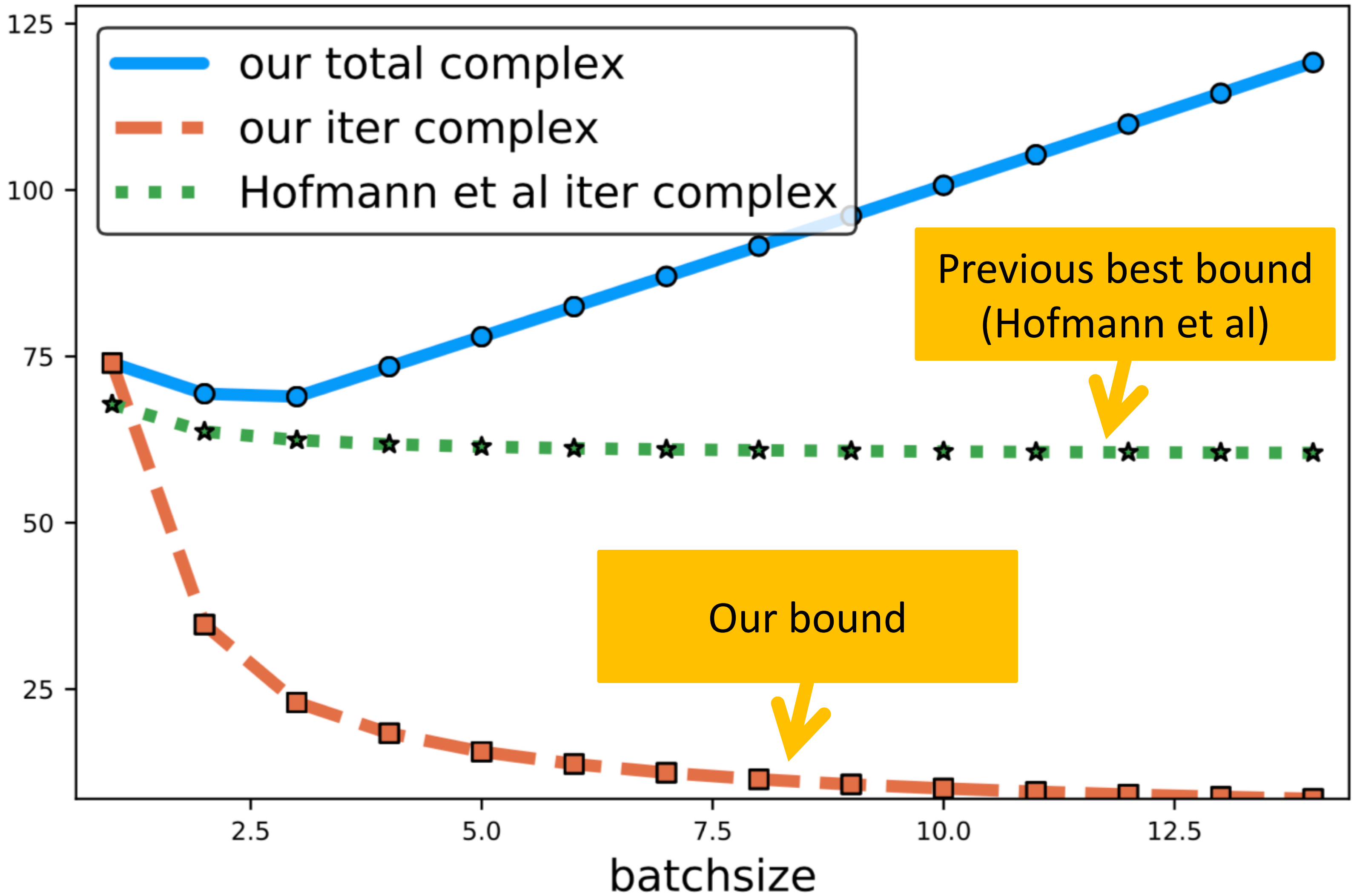
Uniform vs Optimal Probabilities



Data: synthetic
n = 1,000

Minibatch SAGA

Data: australian
LIB-SVM

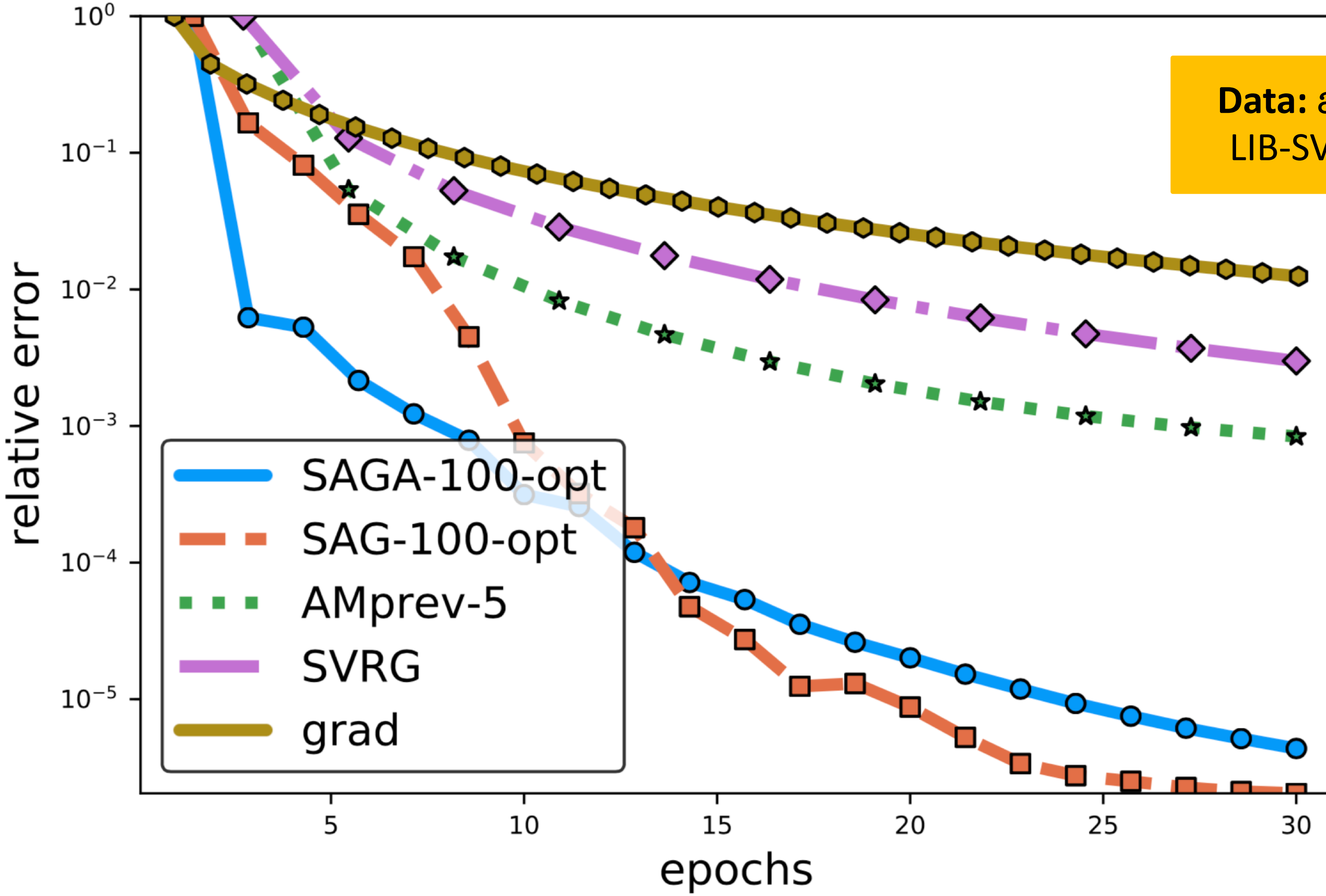


Logistic Regression

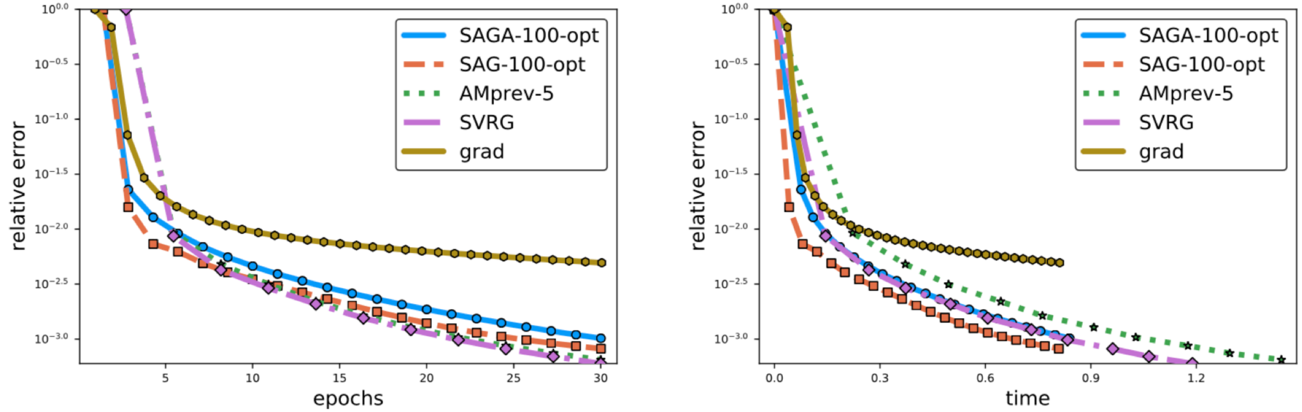
$$f_i(x) = \frac{1}{2}\log\left(1 + e^{-y_i a_i^\top x}\right) + \frac{\lambda}{2}\|x\|^2$$

$$\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x)$$

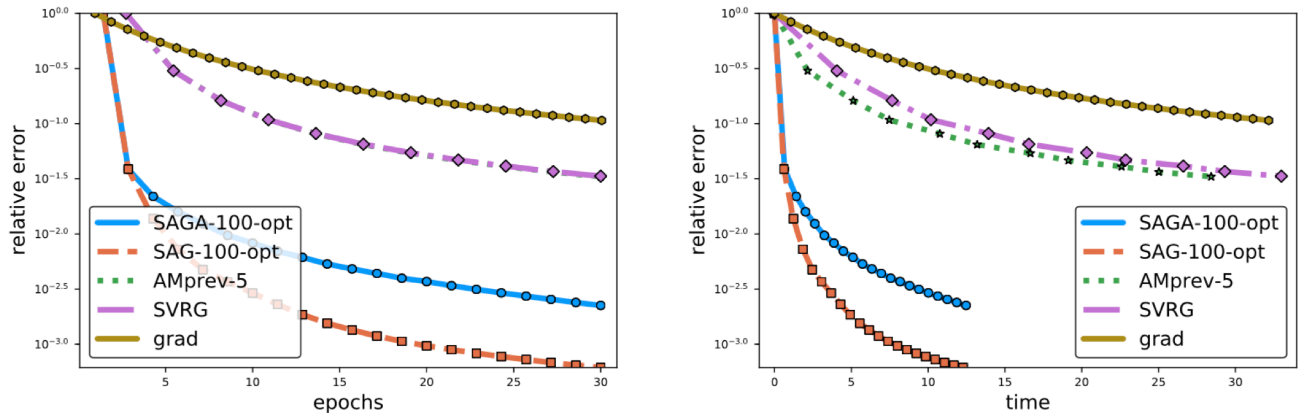
JacSketch vs Other Methods



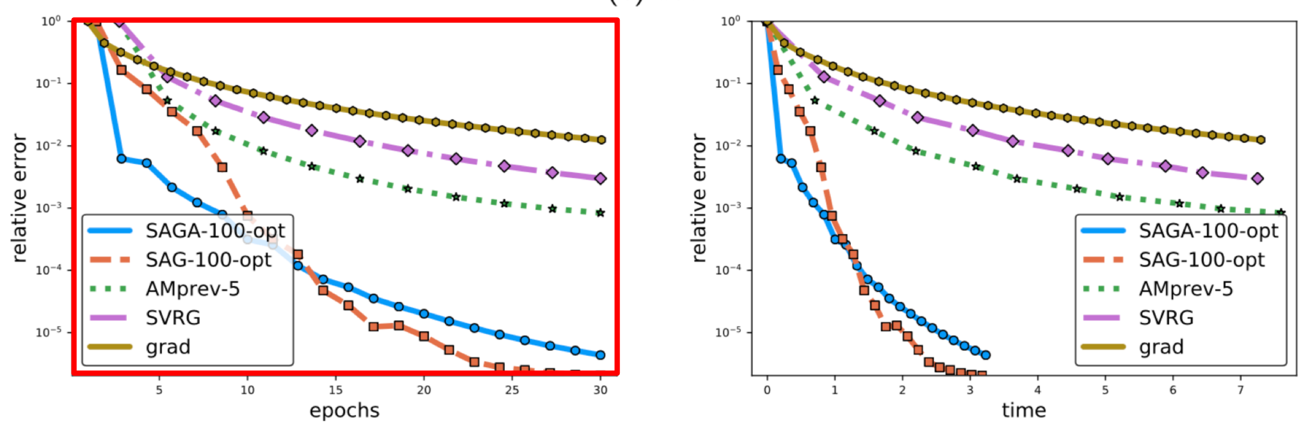
Data: a9a
LIB-SVM



(a) mushrooms



(b) w8a



(c) a9a

The End