Stochastic Quasi-Gradient Methods: Variance Reduction via Jacobian Sketching

Peter Richtárik

Summer School: “Control, Information and Optimization”
Voronovo - June 11, 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 formulation of the original problem, where the control comes from the Jacobian estimate.

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.
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

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

**L2 regularized least squares**
(ridge regression)

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

**L2 regularized logistic regression**

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

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

Current iterate

Stepsize

Next iterate

Unbiased estimator of the gradient:

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

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

Gradient Descent (GD)

\[ g^k \leftarrow \nabla f(x^k) \quad \Rightarrow \quad \nabla \left[ g^k \right] = 0 \]

Stochastic Gradient Descent (SGD)

\[ g^k \leftarrow \nabla f_i(x^k) \quad \Rightarrow \quad \nabla \left[ g^k \right] = \text{BIG} \]
GD vs SGD

Gradient Descent (GD)

Stochastic Gradient Descent (SGD)
## Variance Reduction

| How does it work? | Decreasing stepsizes | Mini-batching | Importance sampling | Adjusting the direction |
|-------------------|----------------------|---------------|---------------------|-------------------------|
| **CONS:**         | Scaling down the noise | More samples, less variance | Sample more important data (or parameters) more often | Duality (SDCA) or Control Variate (SVRG, S2GD, SAGA) |
| **PROS:**         | 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 |

|          | 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 \( F \)

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

2. **SKETCH**

   The \( i \)th unit basis vector

   \[
   \nabla F(x)e_i = \nabla f_i(x)
   \]

   Leads to Stochastic Gradient Descent

   \[
   \frac{1}{n} \nabla F(x)e = \nabla f(x)
   \]

   Leads to Gradient Descent

   Vector of all ones
Introducing General Sketches

We would like to solve the linear matrix equation:

\[ d \begin{bmatrix} \mathbf{J} \\ \nabla \mathbf{F}(x^k) \end{bmatrix} \]

Solve a random linear matrix equation instead:

\[ \mathbf{J} \mathbf{S}_k = \nabla \mathbf{F}(x^k) \mathbf{S}_k \]

Random matrix \( \mathbf{S}_k \sim \mathcal{D} \)

Has many solutions: which solution to pick?
Sketch and Project
Sketch and Project

**New Jacobian estimate**

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

subject to \(JS_k = \nabla F(x^k)S_k\)

**Current Jacobian estimate**

**Frobenius norm**

**Solution:**

\[
J^{k+1} = J^k + (\nabla F(x^k) - J^k) \Pi S_k
\]

**Random LME ensuring consistency with Jacobian sketch**

\[
\Pi S_k \overset{\text{def}}{=} S_k (S_k^T S_k)^{\dagger} S_k^T
\]
Sketch and Project

Original sketch and project

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

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

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

P.R. and Martin Takáč
Stochastic Reformulations of Linear Systems: Algorithms and Convergence Theory
*arXiv*:1706.01108, 2017
Constructing an Unbiased Gradient Estimate
Gradient Estimate

Bias-correcting random variable:
\[ \mathbb{E}_{\theta_k \sim D}[\theta_k \Pi \theta_k e] = e \]

Average of the columns of
\[ \mathbf{J}^k \]

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

Unbiased estimator of the gradient
\[ \mathbb{E}_{\theta_k \sim D}[g^k] = \nabla f(x^k) \]
3. Stochastic Reformulation

(JackSketch as SGD Applied to Controlled Stochastic Reformulation)
Simple Stochastic Reformulation
Reformulation

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

\[ f(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) = \frac{1}{n} \langle F(x), e \rangle = \frac{1}{n} \langle F(x), \mathbb{E}_{S \sim \mathcal{D}}[\theta_S \Pi_S e] \rangle \]

Linearity of expectation

\[ f_S(x) = \sum_{i=1}^{n} \left( \frac{1}{n} \theta_S \Pi_S e \right) f_i(x) \]

Bias-correcting random variable:
\[ \mathbb{E}_{S \sim \mathcal{D}}[\theta_S \Pi_S e] = e \]

Original problem

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

Simple stochastic reformulation

\[ \min_{x \in \mathbb{R}^d} f(x) = \mathbb{E}_{S \sim \mathcal{D}}[f_S(x)] \]

We are minimizing the expectation over random linear combinations of the original functions.
SGD Applied to Simple Stochastic Reformulation

\[ x^{k+1} = x^k - \alpha \nabla f_{S_k}(x^k) \]

**Gradient descent**

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

**Non-uniform SGD**

\[ x^{k+1} = x^k - \frac{\alpha}{np_i} \nabla f_i(x^k) \]

**Non-uniform minibatch SGD**

\[ x^{k+1} = x^k - \frac{\alpha}{nc_1ps_k} \sum_{i \in S_k} \nabla f_i(x^k) \]

\( S_k \sim D \)

- \( S \equiv I \)
- \( \theta_S \equiv 1 \)
- \( \mathbb{P}(S = e_i) = p_i \)  \( \theta_{e_i} \equiv \frac{1}{p_i} \)
- \( \mathbb{P}\left(S = e_S := \sum_{i \in S} e_i \right) = p_s \)  \( \theta_{e_S} \equiv \frac{1}{c_1p_S} \)
Controlled Stochastic Reformulation
Adding Control Variate to Reduce Variance

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

Recall:

\[
f_{S}(x) = \frac{1}{n} \langle F(x), \theta_S \Pi_S e \rangle
\]

\[
z_{S,J}(x) = \frac{1}{n} \langle J^\top x, \theta_S \Pi_S e \rangle
\]

\[
f_{S,J}(x) \overset{\text{def}}{=} f_{S}(x) - z_{S,J}(x) + \mathbb{E}_{S \sim \mathcal{D}} \left[ z_{S,J}(x) \right]
\]
JacSketch = SGD Applied Controlled Stochastic Reformulation

\[ x^{k+1} = x^{k} - \alpha \nabla f_{S_k, J^k}(x^k) \]

Sketch and project

\[ J^{k+1} = J^k + (\nabla F(x^k) - J^k) \Pi_{S_k} \]
**Theorem**

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

\[
\lambda_{\text{max}}(\mathbf{B}) = \lambda_{\text{max}} \left( \mathbb{E}_{S \sim \mathcal{D}} \left[ \| \mathbf{v}_S \|^2 \right] \right) \leq \mathbb{E}_{S \sim \mathcal{D}} \left[ \lambda_{\text{max}} \left( \| \mathbf{v}_S \|^2 \right) \right] = \mathbb{E}_{S \sim \mathcal{D}} \left[ \| \mathbf{v}_S \|^2 \right].
\]

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

\[
\mathbf{v}_S \overset{\text{def}}{=} (\mathbf{I} - \theta_S \Pi_S) \mathbf{e}
\]

\[\theta_S \text{ is bias correcting: } \mathbb{E}_{S \sim \mathcal{D}} [\mathbf{v}_S] = 0\]

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

\[\text{Variance of } \mathbf{v}_S \text{ as an estimator of 0}\]
4. JacSketch and SAGA
Algorithm: JacSketch

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

Iterate:

Draw $S_k \sim \mathcal{D}$

Update the Jacobian estimate:

$$J^{k+1} = J^k + (\nabla F(x^k) - J^k)\Pi_{S_k}$$

Update the gradient estimate:

$$g^k = \frac{1}{n}J^k e + \frac{1}{n}(\nabla F(x^k) - J^k)\Pi_{S_k} e$$

Take a gradient step:

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

\[ \mathbb{E}_{S_k \sim \mathcal{D}}[\theta_{S_k} \Pi_{S_k} e] = e \]
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

\[ n = 5 \]
\[ S_k = \{1, 3, 4\} \]
\[ S_k = \mathbf{I}, S_k = \begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0
\end{bmatrix} \]

\[
J_{k+1}^{i} = \begin{cases} 
J_k^{i} & \text{if } i \notin S_k \\
\nabla f_i(x^k) & \text{if } i \in S_k
\end{cases}
\]

\[
g^k = \frac{1}{n} J^k e + \frac{\theta s_k}{n} \sum_{i \in S_k} (\nabla f_i(x^k) - J^k_i)
\]

\[
x^{k+1} = x^k - \alpha g^k
\]
5. Iteration Complexity of JacSketch
General Theorem
First Main Result (Theorem 3.6)

\[ \kappa \geq \max \left\{ \frac{1}{\kappa} + \frac{\rho}{n^2} \frac{4L_2}{\kappa \mu}, \frac{4L_1}{\mu} \right\} \log \left( \frac{1}{\epsilon} \right) \]

Sketch residual
\[ \rho \:= \lambda_{\text{max}} \left( W^{1/2} E_{S \sim D} \left[ (I - \theta_S \Pi_S) e e^\top (I - \theta_S \Pi_S^\top) \right] W^{1/2} \right) \]

Lyapunov function
\[ \Psi^k := \|x^k - x^*\|^2 + \frac{\alpha}{2L_2} \|J^k - \nabla F(x^*)\|^2 \]

Expected smoothness constants

Stochastic condition number
\[ \kappa := \lambda_{\text{min}} (E_{S \sim D} [\Pi_S]) \]
always: \( 0 \leq \kappa \leq 1 \)

Relative error
\[ \mathbb{E} \left[ \Psi^k \right] \leq \epsilon \Psi_0 \]

Expected smoothness constants

Strong convexity parameter of \( f \)
Special Cases

1. Gradient Descent

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

Strong convexity parameter of \( f \)

2. SAGA with uniform sampling

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

Worst smoothness constant of \( f_i \)

\[ \| \nabla f_i(x) - \nabla f_i(y) \| \leq L_i \| x - y \| \]

\[ L_{\text{max}} := \max_i L_i \]
Special Cases

3. Minibatch SAGA with uniform sampling

\[
\max \left\{ \frac{n}{\tau} + \frac{n - \tau}{(n - 1)\tau} \frac{4L_{\text{max}}}{\mu}, \frac{4L_1}{\mu} \right\} \log \left( \frac{1}{\epsilon} \right)
\]

Minibatch size

\[ S = \text{random subset of } \{1, 2, \ldots, n\} \text{ of size } \tau \text{ chosen uniformly of random} \]

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

\[ \{1, 2, \ldots, n\} = \bigcup_{j=1}^{m} C_j \]

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

Partition

\[ |C_j| = \tau \text{ for all } j \]

\[ m = \frac{n}{\tau} \]

Sketch matrix \quad Bias-correcting random variable

\[ S = I_{\cdot,S} \quad \theta_S = \frac{1}{p_S} \]
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

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

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

Strong convexity parameter of \( f \)

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

Stochastic Lyapunov function

\[ \Psi_S^k := \|x^k - x^*\|^2 + \frac{n\alpha}{2\tau L_S} \left\| \frac{1}{n} \mathbf{J}^k e - \nabla f_{I_S}(x^*) \right\|^2 \]
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 $S \in \mathbb{R}^{n \times \tau}$ | Iteration complexity ($\times \log \frac{1}{\epsilon}$) | Reference |
|----|--------------------------------------------|-----------------------------------------|------------------------------------------------------|-----------|
| 1  | JacSketch                                  | any unbiased                            | $\max \left\{ 4C_1 \mu, \frac{1}{\kappa}, 4\rho C^2 \kappa \mu n^2 \right\}$ | Thm 3.6   |
| 2  | JacSketch (with any probabilities for $\tau$–partition) | $I_S$                                   | $\max_{C \in \text{supp}(S)} \left( \frac{1}{PC} + \frac{\tau}{nPC} \frac{4LC}{\mu} \right)$ | Thm 5.2   |
| 3  | Gradient descent                           | $I$                                     | $\frac{4L}{\mu}$                                     | Thm 3.6   (101) |
| 4  | Gradient descent                           | $I$                                     | $\frac{4L}{\mu}$                                     | Thm 5.2   (130) |
| 5  | SAGA (with uniform sampling)               | $I_S$                                   | $n + \frac{4L_{\max}}{\mu}$                          | Thm 3.6   (102) |
| 6  | SAGA (with uniform sampling)               | $I_S$                                   | $n + \frac{4L_{\max}}{\mu}$                          | Thm 5.2   (131) |
| 7  | SAGA (with importance sampling)            | $I_S$                                   | no improvement on uniform sampling                    | Thm 3.6   |
| 8  | SAGA (with importance sampling)            | $I_S$                                   | $n + \frac{4L}{\mu}$                                 | Thm 5.2   (133) |
| 9  | Minibatch SAGA ($\tau$–uniform sampling)   | $I_S$ $\text{Diag}(w_i)$               | $\max \left\{ \frac{4\mathcal{G}_{\max}}{\mu}, \frac{n}{\tau} + \frac{4\rho}{\mu n} \max_i \left( \frac{L_i}{w_i} \right) \right\}$ | Thm 3.6   (100) |
| 10 | Minibatch SAGA ($\tau$–nice sampling)      | $I_S$ $\text{Diag}(w_i)$               | $\max \left\{ \frac{4\mathcal{G}_{\max}}{\mu}, \frac{n}{\tau} + \frac{n-\tau}{(n-1)\tau} \frac{4L_{\max}}{\mu} \right\}$ | Thm 3.6   (103) |
| 11 | Minibatch SAGA ($\tau$–nice sampling)      | $I_S$ $\text{Diag}(L_i)$               | $\max \left\{ \frac{4\mathcal{G}_{\max}}{\mu}, \frac{n}{\tau} + \frac{n-\tau}{n\tau} \frac{4(L+L_{\max})}{\mu} \right\}$ | Thm 3.6   (104) |
| 12 | Minibatch SAGA ($\tau$–partition sampling) | $I_S$ $\text{Diag}(w_i)$               | $\frac{n}{\tau} + \frac{4L_{\max}}{\mu}$            | Thm 3.6   (105) |
| 13 | Minibatch SAGA ($\tau$–partition sampling) | $I_S$ $\text{Diag}(L_i)$               | $\frac{n}{\tau} + \frac{4\max_{C \in \text{supp}(S)} \frac{1}{C} \sum_{i \in C} L_i}{\mu}$ | Thm 3.6   (106) |
| 14 | Minibatch SAGA (importance $\tau$–partition sampling) | $I_S$ $\text{Diag}(L_i)$               | $\frac{n}{\tau} + \frac{4\max_{C \in \text{supp}(S)} \frac{1}{C} \sum_{i \in C} L_i}{\mu}$ | Thm 5.2   (135) |
6. Experiments
Ridge Regression

\[ f_i(x) = \frac{1}{2} (a_i^T x - y_i)^2 + \frac{\lambda}{2} \|x\|^2 \]

\[
\min_{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x)
\]
Uniform vs Optimal Probabilities

Data: synthetic
\( n = 1,000 \)
Minibatch SAGA

Data: australian LIB-SVM

- Our total complex
- Our iter complex
- Hofmann et al iter complex

Previous best bound (Hofmann et al)

Our bound
Logistic Regression

\[
\min _{x \in \mathbb{R}^d} f(x) := \frac{1}{n} \sum _{i=1} ^n f_i(x)
\]

\[
f_i(x) = \frac{1}{2} \log \left( 1 + e^{-y_i a_i ^T x} \right) + \frac{\lambda}{2} \|x\|^2
\]
JacSketch vs Other Methods

Data: a9a
LIB-SVM
