Friday Seminar

Bias in SAG-like Variance Reduced Stochastic Gradient Methods

Martin Morin

October 9, 2020

LUND UNIVERSITY

Example Problems

$$\min_x \frac{1}{n}\sum_{i=1}^n f_i(x) + g(x)$$

- NN Classifiers:
  f_i is the composition of NN and cost
- Least Squares:
  $\frac{1}{n}\sum_{i=1}^n f_i(x) = \frac{1}{n}\sum_{i=1}^n (a_i^T x - b_i)^2 = \frac{1}{n}\|Ax - b\|_2^2$
- SVM: $f_i(x) = \max(0, 1 - y_i(a_i^T x - b_i))$ and $g(x) = \|x\|_2^2$
- Logistic Regression: $f_i(x) = \ln(1 + e^{-y_i(a_i^T x - b_i)})$

In all cases are i associated with a particular data point. The linear predictor/classifier $a_i^T x - b_i$ can be replaced by a nonlinear $h_i(x)$.

Fermat's Rule

$$\min_x \frac{1}{n}\sum_{i=1}^{n} f_i(x) \iff 0 = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x)$$

Note: $\not\Rightarrow \ 0 = \nabla f_i(x)$

Stochastic Gradient Descent

Sample i uniformly from $\{1, ..., n\}$

$$x_{k+1} = x_k - \lambda_k \nabla f_i(x_k)$$

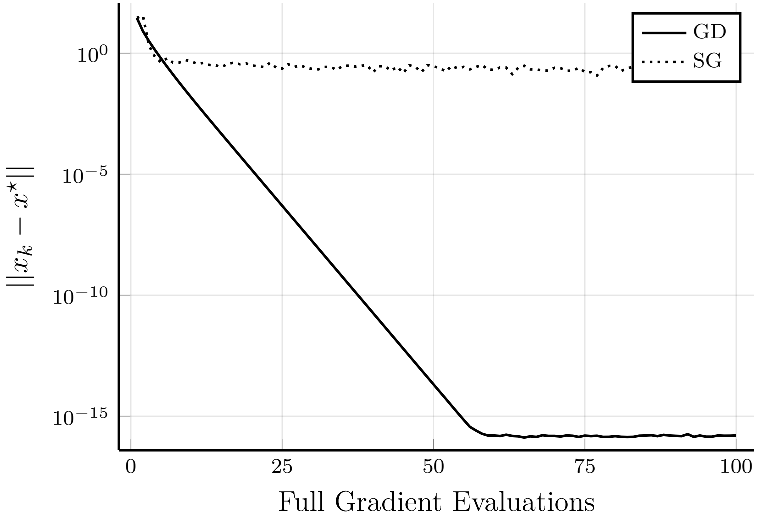
Unbiased: $\mathbb{E}\,\nabla f_i(x) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x)$

However: $x^\star \neq x^\star - \lambda_k \nabla f_i(x^\star)$

Does not converge unless $\lambda_k \to 0$.

Slow convergence, not suitable for high-accuracy solutions.

Stochastic Gradient vs. Gradient Descent



Stochastic Variance Reduced Gradient Methods

SAG:

Sample i uniformly from $\{1, ..., n\}$

$$y_{i,k+1} = \nabla f_i(x_k)$$
$$y_{j,k+1} = y_{j,k}, \quad \forall j \neq i$$
$$x_{k+1} = x_k - \lambda \tfrac{1}{n} \textstyle\sum_{j=1}^{n} y_{j,k+1}$$

SAGA:

Sample i uniformly from $\{1, ..., n\}$

$$x_{k+1} = x_k - \lambda(\nabla f_i(x_k) - y_{i,k} + \tfrac{1}{n} \textstyle\sum_{j=1}^{n} y_{j,k})$$
$$y_{i,k+1} = \nabla f_i(x_k)$$
$$y_{j,k+1} = y_{j,k}, \quad \forall j \neq i$$

SVRG, S2GD,. . .

Stochastic Variance Adjusted Gradient Method (SVAG)

Sample i uniformly from $\{1, ..., n\}$

$$x_{k+1} = x_k - \tfrac{\lambda}{n}\big(\theta(\nabla f_i(x_k) - y_{i,k}) + \textstyle\sum_{i=1}^n y_{i,k}\big)$$

$$y_{i,k+1} = \nabla f_i(x_k)$$

$$y_{j,k+1} = y_{j,k}, \quad \forall j \neq i$$

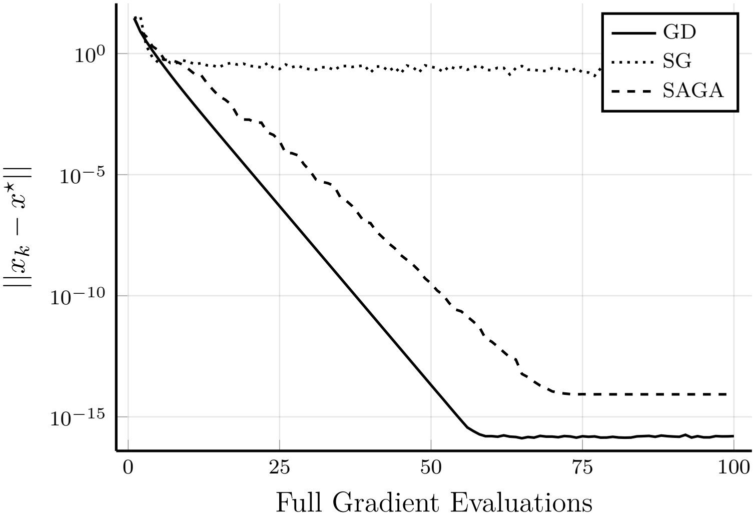
SAG: $\theta = 1$
SAGA: $\theta = n$

At optimum with $y_i^\star = \nabla f_i(x^\star), \forall i$ then

$$x^\star = x^\star - \tfrac{\lambda}{n}\big(\theta \underbrace{(\nabla f_i(x^\star) - y_i^\star)}_{=0} + \underbrace{\textstyle\sum_{i=1}^n y_i^\star}_{=0}\big).$$

Possible to converge with fixed step-size.

SG vs. GD vs. SAGA



Bias/Variance Trade-Off

Gradient Estimate:

$$G_i(x, y) := \tfrac{\theta}{n}\big(\nabla f_i(x) - y_i\big) + \tfrac{1}{n}\sum_{j=1}^{n} y_j$$

Expectation:

$$\mathbb{E}\, G_i(x, y) = \tfrac{\theta}{n^2}\sum_{j=1}^{n} \nabla f_j(x) + \tfrac{n-\theta}{n^2}\sum_{j=1}^{n} y_j$$

Variance:

$$\begin{aligned}&\mathbb{E}\| G_i(x, y) - \mathbb{E}\, G_i(x, y)\|^2 \\ &\quad = \tfrac{\theta^2}{n^2}\,\mathbb{E}\,\|(\nabla f_i(x) - y_i) - \tfrac{1}{n}\textstyle\sum_{j=1}^{n}(\nabla f_j(x) - y_j)\|^2\end{aligned}$$

Unbiased when $\theta = n$. Smaller θ, smaller variance. Zero variance in $(x^\star, y^\star)$.

Main Question

How does bias affect the algorithm?

- What properties affect how the bias should be chosen?
- Can we design ways of selecting the bias?

Current state

- Both SAG and SAGA are well used but neither having no clear advantage.
- Unbiased theory well developed and matching practice.
- Biased theory behind practice.

Main Question

How does bias affect the algorithm?

- What properties affect how the bias should be chosen?
- Can we design ways of selecting the bias?

Current state

- Both SAG and SAGA are well used but neither having no clear advantage.
- Unbiased theory well developed and matching practice.
- Biased theory behind practice.

Main Question

How does bias affect the algorithm?

- What properties affect how the bias should be chosen?
- Can we design ways of selecting the bias?

Current state

- Both SAG and SAGA are well used but neither having no clear advantage.
- Unbiased theory well developed and matching practice.
- Biased theory behind practice.

SVAG - Root Finding Version

Problem:

$$0 = \frac{1}{n}\sum_{i=1}^{n} R_i x$$

where $R_i : \mathcal{H} \to \mathcal{H}$.

Algorithm:

$$\begin{aligned}
&\text{Sample } i \text{ uniformly from } \{1, ..., n\} \\
x_{k+1} &= x_k - \tfrac{\lambda}{n}\big(\theta(R_i x_k - y_{i,k}) + \textstyle\sum_{i=1}^{n} y_{i,k}\big) \\
y_{i,k+1} &= R_i x_k \\
y_{j,k+1} &= y_{j,k}, \quad \forall j \neq i
\end{aligned}$$

$R_i = \nabla f_i$ gives the minimization formulation.

Properties

An operator $R : \mathbb{R}^N \to \mathbb{R}^N$ is β-**cocoercive** if

$$\langle Rx - Ry, x - y \rangle \geq \beta \|Rx - Ry\|^2$$

holds for all $x, y \in \mathbb{R}^N$.

A convex function $f : \mathbb{R}^N \to \mathbb{R}$ is called L-**smooth** if the gradient is L-Lipschitz continuous,

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

The gradient of a L-smooth function is $\frac{1}{L}$-cocoercive.

Cocoercivity vs. Gradients of Smooth Functions

Class of cocoercive operator larger than the class of smooth gradients

However, "gradient descent",

$$x_{k+1} = x_k - \lambda R x_k,$$

behaves the "same", i.e.,

$$R x_k \to 0$$

with same rate for same λ, regardless if R is gradient of smooth function or only cocoercive.

Is the same true for SAGA? SAG? SVAG?

Convergence Theorems

Theorem

Let each R_i be $\frac{1}{L}$-cocoercive. If

$$\frac{1}{L(2+|n-\theta|)} > \lambda > 0,$$

then $x^k \to x^\star$ and $y_i^k \to \nabla f_i(x^\star)$ almost surely.

Theorem

Let each $R_i = \nabla f_i$ where f_i is convex and L-smooth. If $\theta \leq n$ and

$$\frac{1}{L}\frac{1}{2+(1-\frac{\theta}{n})(\theta-1)(\frac{\theta-1}{n}-1+\frac{\theta-1}{|\theta-1|}\sqrt{2})} > \lambda > 0,$$

then $x^k \to x^\star$ and $y_i^k \to \nabla f_i(x^\star)$ almost surely.

Improves or equals the known upper bounds.

For $\theta \neq n$, cocoercivity $\lambda < O(\frac{1}{n})$ while smoothness $\lambda < O(1)$.

Special Cases

SAGA: For both cocoercivity and smoothness assumptions,

$$\frac{1}{2L} > \lambda > 0.$$

SAG: For cocoercivity and smoothness assumptions respectively,

$$\frac{1}{(2+n-1)L} > \lambda > 0, \quad \frac{1}{2L} > \lambda > 0.$$

Only the same when $n = 1$, i.e., ordinary gradient descent.

Tight Convergence Results

Cocoercivity: Empirical.
Smoothness: ???

Example: Each $R_i : \mathbb{R}^2 \to \mathbb{R}^2$ is an averaged rotation,

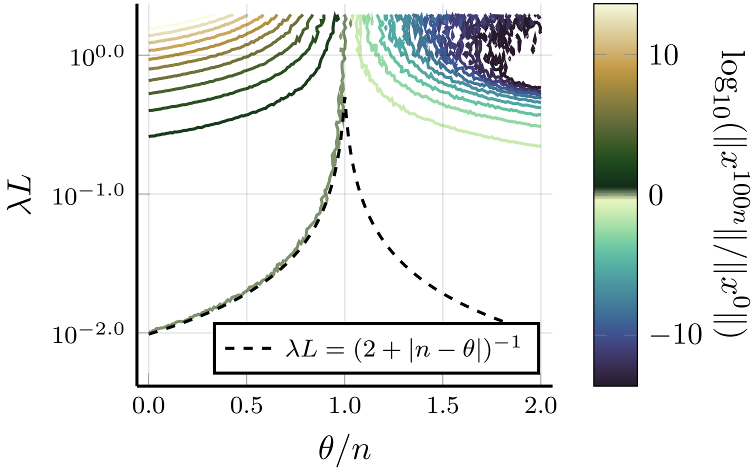
$$R_i = \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \cos\tau & -\sin\tau \\ \sin\tau & \cos\tau \end{bmatrix}$$

for some $\tau \in [0^\circ, 360^\circ)$.

Each R_i is 1-cocoercive and zero is the only solution if $\tau \neq 180\,\mathrm{deg}$.

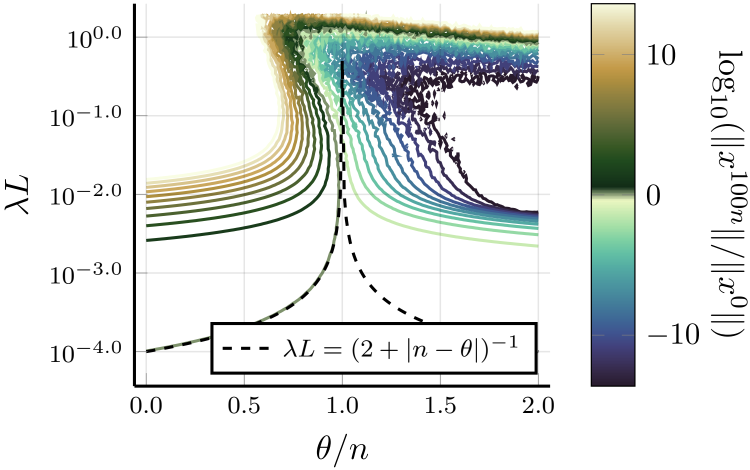
The results appear tight as $\tau \to 180^\circ$.

Tight Example



$\tau = 179^\circ$ and $n = 100$

Tight Example



$\tau = 179^\circ$ and $n = 10000$

Automatic Bias Selection

Goal: Make the approximation,

$$\nabla F(x_k) \approx \tfrac{\theta}{n}\big(\nabla f_i(x_k) - y_{i,k}\big) + \tfrac{1}{n}\sum\nolimits_{j=1}^{n} y_{j,k},$$

as good as possible.

Hence,

$$\min_{\theta} \|\nabla F(x_k) - \big(\tfrac{\theta}{n}(\nabla f_i(x_k) - y_{i,k}) + \tfrac{1}{n}\sum\nolimits_{i=1}^{n} y_{i,k}\big)\|^2.$$

Automatic Bias Selection

Solution

$$\theta = n\frac{\langle \nabla F(x_k) - \frac{1}{n}\sum_{i=1}^{n} y_{i,k}, \nabla f_i(x_k) - y_{i,k}\rangle}{\|\nabla f_i(x_k) - y_{i,k}\|^2}$$

Total innovation

$$\nabla F(x_k) - \tfrac{1}{n}\textstyle\sum_{i=1}^{n} y_{i,k} = \mathbb{E}[\nabla f_i(x_k) - y_{i,k}]$$

Estimate with exponential moving average of $\nabla f_i(x_k) - y_{i,k}$.

Adaptive SVAG

Sample i uniformly from $\{1, ..., n\}$

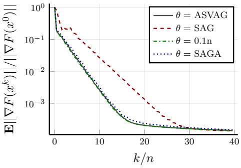
$$
\begin{aligned}
I_{k+1} &= \beta I_k + (1-\beta)(\nabla f_i(x_k) - y_{i,k}) \\
\theta_{k+1} &= \mathrm{saturate}_{-\delta}^{\delta}\left(\tfrac{n}{1-\beta^{k+1}} \tfrac{\langle I_{k+1}, \nabla f_i(x_k) - y_{i,k}\rangle}{\|\nabla f_i(x_k) - y_{i,k}\|^2 + \epsilon}\right) \\
x_{k+1} &= x_k - \tfrac{\lambda}{n}\big(\theta_{k+1}(\nabla f_i(x_k) - y_{i,k}) + \textstyle\sum_{i=1}^n y_{i,k}\big) \\
y_{i,k+1} &= \nabla f_i(x_k) \\
y_{j,k+1} &= y_{j,k}, \quad \forall j \neq i
\end{aligned}
$$

where $\beta \in [0,1]$, $\epsilon > 0$, $\delta \geq 0$ and $I_0 = 0$.

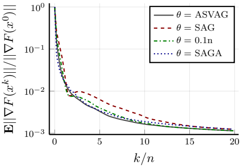
Default choice: $\beta = 0.9$, $\epsilon = 10^{-8}$ and $\delta = n$.

Logistic Regression

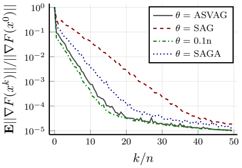
$$\min_x \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{-y_i a_i^T x})$$



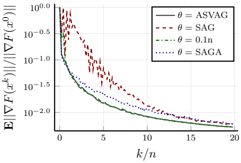
protein



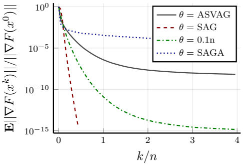
breast-cancer



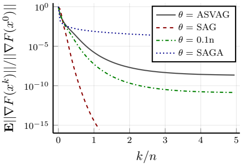
a9a



gisette_scale



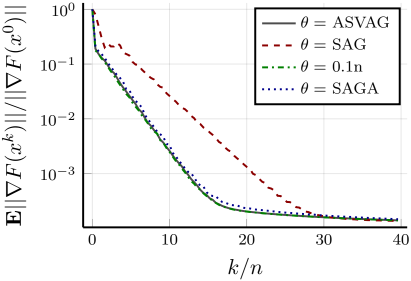
mushrooms



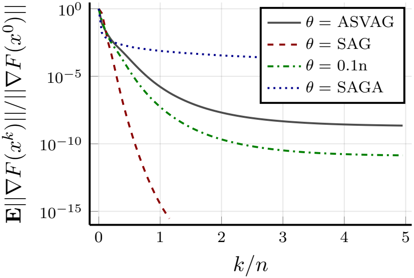
mnist.scale

Logistic Regression

$$\min_x \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{-y_i a_i^T x})$$



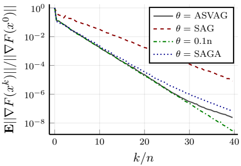
`protein`



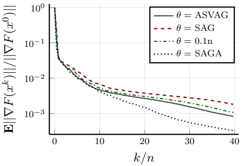
`mnist.scale`

Square Hinge Loss SVM

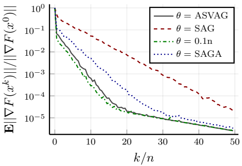
$$\min_x \frac{1}{n}\sum_{i=1}^{n}\left(\max(0, 1 - y_i a_i^T x)^2 + \frac{\gamma}{2}\|x\|^2\right)$$



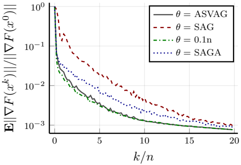
`protein`, $\gamma = 10^{-3}$



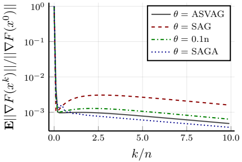
`breast-cancer`, $\gamma = 10^{-3}$



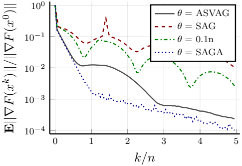
`a9a`, $\gamma = 10^{-4}$



`gisette_scale`, $\gamma = 10^{-1}$



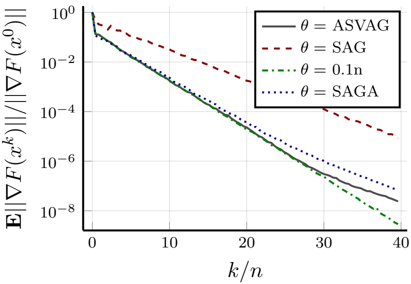
`mushrooms`, $\gamma = 10^{-3}$



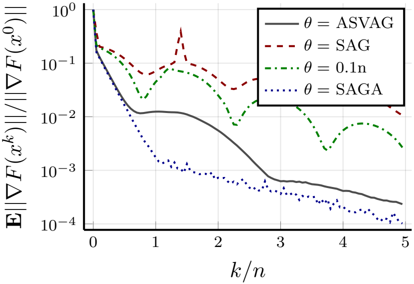
`mnist.scale`, $\gamma = 10^{-1}$

Square Hinge Loss SVM

$$\min_x \frac{1}{n} \sum_{i=1}^{n} \left(\max(0, 1 - y_i a_i^T x)^2 + \frac{\gamma}{2} \|x\|^2 \right)$$



`protein`,
$\gamma = 10^{-3}$



`mnist.scale`,
$\gamma = 10^{-1}$

Conclusion?