Theoretical and Experimental Analysis of the SAGA Algorithm

A Comprehensive Study on Variance Reduction in Optimization

Kra Gérard

Master's in Mathematical Engineering
Université Côte d'Azur

February 26, 2025

Presentation Outline

1. Introduction
2. SAGA on strongly convex loss functions
3. SAGA on merely convex loss functions
4. A Comparison of SAGA with SGD, SAG, and SVRG
5. Numerical Experiments

Optimization Problem

Problem Formulation

$$\min_{w \in \mathbb{R}^d} \left\{ F(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w) + R(w) \right\}$$

- $f_i(w)$: Loss functions (smooth and convex).
- $R(w)$: Regularization term (possibly non-smooth).
- w : The parameter vector being optimized.

Optimization Problem

SAGA Algorithm: Pseudocode

1. Initialize $w^{(0)}$ and gradient table $\{\phi_i^{(0)}\}$.
2. For each iteration t:
   - Sample i_t uniformly.
   - Compute $g_{i_t} = \nabla f_{i_t}(w^{(t)})$.
   - Update gradient table: $\phi_{i_t}^{(t+1)} = g_{i_t}$.
   - Update $w^{(t+1)}$ using:

   $$w^{(t+1)} = \mathrm{prox}_{\eta R}\left(w^{(t)} - \eta\left(g_{i_t} - \phi_{i_t}^{(t)} + \frac{1}{n}\sum_{j=1}^{n}\phi_j^{(t)}\right)\right),$$

   for a given stepsize η.

SAGA Algorithm on L-smooth and μ-strongly convex functions

Assume each f_i is an L-smooth and μ-strongly convex function ($\mu > 0$), mapping from $\mathbb{R}^d$ to $\mathbb{R}$ and where R is given by a proximal operator. Then, we have the following theorem:

Theorem

For $\eta = 2(\mu n + L)$, SAGA achieves linear (geometric) convergence:

$$\mathbb{E}[\|w^{(k)} - w^*\|^2] \leq \left(1 - \frac{1}{\kappa}\right)^k \mathbb{E}[\|w^{(0)} - w^*\|^2 + C_0].$$

Remarks:

- $\eta = 3L$ is also viable. But in this case, the geometric constant adjusts to $\left(1 - \min\left\{\frac{1}{2n}, \frac{\mu}{3L}\right\}\right)$.
- $\kappa = \frac{\eta}{\mu}$: is the condition number of the problem. Greater κ is, the slower the algorithm converges.
- w^*: is optimal solution of the problem.
- C_0: Depends only on the initial conditions.

SAGA Algorithm on L-smooth and merely convex functions

Assume each f_i is an L-smooth and merely convex function mapping from $\mathbb{R}^d$ to $\mathbb{R}$. By considering the averaged iterate

$$\bar{w}^{(k)} = \frac{1}{k} \sum_{t=1}^{k} w^{(t)},$$

and with an appropriate stepsize η, one can prove that:

Theorem

For $\eta = 3L$, SAGA achieves a sub-linear convergence:

$$\mathbb{E}\left[F(\bar{w}^{(k)}) - F(w^*)\right] = \mathcal{O}\left(\frac{1}{k}\right).$$

Remarks:

- This $\mathcal{O}(1/k)$ convergence rate is optimal for first-order methods in the convex setting.
- The functional gap $F(w^{(k)}) - F(w^*)$ is widely employed as a convergence metric in this context.

A Comparison of SAGA with SGD, SAG, and SVRG

In their article [1], the authors compare the SAGA algorithm with three key incremental gradient methods: SGD (Stochastic Gradient Descent), SAG (Stochastic Average Gradient), and SVRG (Stochastic Variance Reduced Gradient).

SAGA vs. SAG:

Both SAGA and SAG maintain a table of past gradient estimates to reduce variance. However, SAGA updates stored gradients in an unbiased manner, leading to a more theoretically sound and stable convergence guarantee.
Unlike SAG, SAGA also supports composite objectives with proximal operators, broadening its applicability.

A Comparison of SAGA with SGD, SAG, and SVRG

SAGA vs. SVRG:

SVRG computes a full gradient snapshot at periodic intervals, while SAGA continuously updates stored gradients, eliminating the need for an outer loop and additional tuning of iteration parameters.
Although SVRG has a lower memory footprint, it requires $2\times$ to $3\times$ more gradient evaluations per epoch compared to SAGA. This makes SAGA more efficient for problems where storing past gradients is not a significant limitation.

A Comparison of SAGA with SGD, SAG, and SVRG

SAGA vs. SGD:

Unlike standard Stochastic Gradient Descent (SGD), which suffers from high variance in gradient updates, SAGA leverages variance reduction techniques to achieve significantly faster convergence.
While SGD requires careful step-size tuning, SAGA adapts naturally to strong convexity and supports non-strongly convex problems without modifications. This makes SAGA a more robust choice for large-scale optimization.

Experimental Setup

- **Datasets:** Diabetes dataset (scikit-learn).
- **Problems:**
  - Ridge Regression (L_2 regularization).
  - Lasso (L_1 regularization).
- **Algorithms:** SGD, SAG, SVRG, SAGA.

Ridge Regression

Optimization Problem

$$\min_{w \in \mathbb{R}^d} \left\{ F_{\text{ridge}}(w) = \frac{1}{2n} \|Xw - y\|^2 + \frac{\lambda}{2} \|w\|^2 \right\}$$

- $X \in \mathbb{R}^{n \times d}$: Data matrix.
- $y \in \mathbb{R}^n$: Target vector.
- $\lambda > 0$: Regularization parameter. For our experiments, we set $\lambda = 1 \times 10^{-5}$.
- $\|w\|^2$: L_2 norm of w, encouraging small weights.

Objective: Illustrate the SAGA convergence rate in μ-strongly convex case.

Lasso

Optimization Problem

$$\min_{w \in \mathbb{R}^d} \left\{ F_{\text{lasso}}(w) = \frac{1}{2n} \|Xw - y\|^2 + \lambda \|w\|_1 \right\}$$

- $\|w\|_1$: L_1 norm of w, defined as $\sum_{j=1}^{d} |w_j|$.
- $\lambda > 0$: Regularization parameter. Here, we set $\lambda = 1.0$.

Objective: Illustrate the SAGA convergence rate in convex case.

Results for Ridge Regression

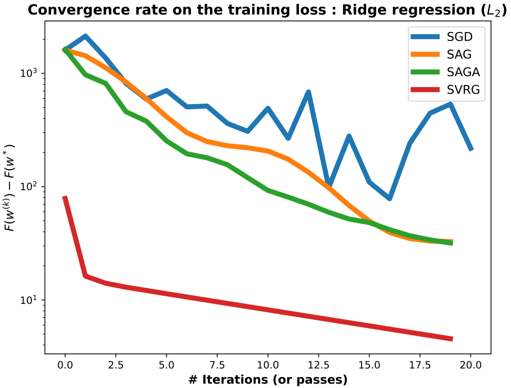

Figure: Convergence on Ridge Regression (Strongly Convex).

Results for Ridge Regression

- SAGA achieves linear convergence, as predicted by theory.
- Outperforms SGD in terms of convergence speed.

Results for Lasso

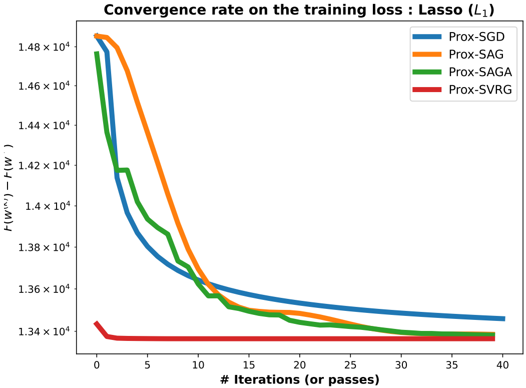

Figure: Convergence on Lasso (Convex).

Results for Lasso

- SAGA achieves $\mathcal{O}(1/k)$ convergence, matching theoretical expectations.
- Proximal-SAGA handles the non-smooth L_1 term effectively.

Thank You!

References

[1] Aaron Defazio, Francis Bach, Simon Lacoste-Julien. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly Convex Composite Objectives. 2014. hal-01016843v2