SAGA

Jiayu Qin | 50477971 | 09/13/2022

Contents

What is SAGA

- An optimization method
- A kind of incremental gradient algorithm
- Fast linear convergence rate
- Support composite objectives
- Support non-strongly convex problems

Contents

Some background knowledge

- For a finite sum function
- $f(x) = \sum_{i=1}^{n} f_i(x) + h(x)$
- Where x$\in$ R^d and $f_i(x)$ are strongly convex and h is convex regularization term that is convex but potentially non-differentiable
- Few incremental method are applicable in this setting

Contents

SAGA algorithm

- Starting from some known initial vector $x_0 \in R^d$ and of known derivatives $f_i'(\varphi_i^k)$ with $\varphi_i^0 = x_0$ for each i
- Here φ_i^k is initialized as n copies of x, and during algorithm processing, they are gradually updated
- These derivatives are stored in a table structure of length n, or a n*d matrix
- With a step size of d and initial k=0,do:

SAGA Algorithm: Given the value of x^k and of each $f_i'(\phi_i^k)$ at the end of iteration k, the updates for iteration $k+1$ is as follows:

1. Pick a j uniformly at random.
2. Take $\phi_j^{k+1} = x^k$, and store $f_j'(\phi_j^{k+1})$ in the table. All other entries in the table remain unchanged. The quantity ϕ_j^{k+1} is not explicitly stored.
3. Update x using $f_j'(\phi_j^{k+1})$, $f_j'(\phi_j^k)$ and the table average:

$$w^{k+1} = x^k - \gamma \left[f_j'(\phi_j^{k+1}) - f_j'(\phi_j^k) + \frac{1}{n} \sum_{i=1}^{n} f_i'(\phi_i^k) \right], \tag{1}$$

$$x^{k+1} = \text{prox}_\gamma^h \left(w^{k+1} \right). \tag{2}$$

SAGA algorithm

SAGA Algorithm: Given the value of x^k and of each $f_i'(\phi_i^k)$ at the end of iteration k, the updates for iteration $k+1$ is as follows:

1. Pick a j uniformly at random.
2. Take $\phi_j^{k+1} = x^k$, and store $f_j'(\phi_j^{k+1})$ in the table. All other entries in the table remain unchanged. The quantity ϕ_j^{k+1} is not explicitly stored.
3. Update x using $f_j'(\phi_j^{k+1})$, $f_j'(\phi_j^k)$ and the table average:

$$w^{k+1} = x^k - \gamma \left[f_j'(\phi_j^{k+1}) - f_j'(\phi_j^k) + \frac{1}{n} \sum_{i=1}^{n} f_i'(\phi_i^k) \right], \tag{1}$$

$$x^{k+1} = \operatorname{prox}_\gamma^h \left(w^{k+1} \right). \tag{2}$$

- Here, the proximal operator is defined as:

$$\operatorname{prox}_\gamma^h (y) := \operatorname*{argmin}_{x \in \mathbb{R}^d} \left\{ h(x) + \frac{1}{2\gamma} \|x - y\|^2 \right\}. \tag{3}$$

Contents

Related work

- Comparison between SAGA and similar works

	SAGA	SAG	SDCA	SVRG	FINITO
Strongly Convex (SC)	✓	✓	✓	✓	✓
Convex, Non-SC*	✓	✓	✗	?	?
Prox Reg.	✓	?	✓[6]	✓	✗
Non-smooth	✗	✗	✓	✗	✗
Low Storage Cost	✗	✗	✗	✓	✗
Simple(-ish) Proof	✓	✗	✓	✓	✓
Adaptive to SC	✓	✓	✗	?	?

Figure 1: Basic summary of method properties. Question marks denote unproven, but not experimentally ruled out cases. (*) Note that any method can be applied to non-strongly convex problems by adding a small amount of L2 regularisation, this row describes methods that do not require this trick.

What's wrong with SGD?

- Variance of SGD's update direction can only go to zero when decreasing step sizes are used
- Thus preventing a linear convergence rate as in GD
- Use variance reduction process instead to get constant step size
- Some possible ways:

$$\text{(SAG)} \qquad x^{k+1} = x^k - \gamma \left[\frac{f_j'(x^k) - f_j'(\phi_j^k)}{n} + \frac{1}{n} \sum_{i=1}^{n} f_i'(\phi_i^k) \right], \qquad (4)$$

$$\text{(SAGA)} \qquad x^{k+1} = x^k - \gamma \left[f_j'(x^k) - f_j'(\phi_j^k) + \frac{1}{n} \sum_{i=1}^{n} f_i'(\phi_i^k) \right], \qquad (5)$$

$$\text{(SVRG)} \qquad x^{k+1} = x^k - \gamma \left[f_j'(x^k) - f_j'(\tilde{x}) + \frac{1}{n} \sum_{i=1}^{n} f_i'(\tilde{x}) \right]. \qquad (6)$$

A more generalized situation

- Suppose we want to estimate E(x), but hard to compute
- We have Y highly correlated to x, while easier to compute E(Y)
- $\theta_\alpha = \alpha(X - Y) + (1 - \alpha)E(Y)$ to estimate E(X)
- E(θ_α) is a convex combination of E(X) and E(Y)
- When $\alpha = 1$, it's a unbiased estimation
- Var(θ_α)=$\alpha^2(\mathrm{Var}(X) + Var(Y) - 2Cov(X,Y))$
- By varying α from 0 to 1, we increase variance but decrease bias

SAGA , SAG and SVRG

$$\text{(SAG)} \qquad x^{k+1} = x^k - \gamma \left[\frac{f_j'(x^k) - f_j'(\phi_j^k)}{n} + \frac{1}{n} \sum_{i=1}^{n} f_i'(\phi_i^k) \right], \qquad (4)$$

$$\text{(SAGA)} \qquad x^{k+1} = x^k - \gamma \left[f_j'(x^k) - f_j'(\phi_j^k) + \frac{1}{n} \sum_{i=1}^{n} f_i'(\phi_i^k) \right], \qquad (5)$$

$$\text{(SVRG)} \qquad x^{k+1} = x^k - \gamma \left[f_j'(x^k) - f_j'(\tilde{x}) + \frac{1}{n} \sum_{i=1}^{n} f_i'(\tilde{x}) \right]. \qquad (6)$$

- Both SAG and SAGA can be viewed in such framework:
- X is SGD direction sample $f_j'(x_k)$, and Y is past gradient $f_j'(\varphi_k^j)$
- SAG: $\alpha = 1/n$ biased, low variance
- SAGA: $\alpha = 1$ unbiased, high variance
- SVRG: $\alpha = 1$ while Y=$f_j'(\tilde{x})$,in which $\tilde{x}$ is updated in outer loop

SAGA vs. SVRG

- SAGA updates φ_j value each time j is picked while SVRG updates all φ as a batch
- The usage of SAGA vs SVRG is problem dependent
- Compared to SAGA, SVRG saves memory, but consume more computing power
- SVRG has additional parameter to be tuned
- In NNs, however, memory is more expensive.

Contents

Implementation tricks

- In many problems, $f_i{}'$ is just a simple weighing of vector x, such as logistic regression and least square
- So we need only store weighing constants instead of whole derivative matrix

- When data points are introduced one by one in the first round, we can compute average based on known ones' gradient
- This also works for SAG

Implementation tricks

- When derivatives are sparse, you may want to update x in time so that sparse updates are done at each iteration

- SAGA assumes that each f_i is strongly convex, but in most cases we have only convex f_i.
- We introduce quadratic regularization term $\frac{\mu}{2}||x^2||$ to maintain strong convexity, and SAGA formula should be changed:

$$x^{k+1} = (1 - \gamma\mu)\, x^k - \gamma \left[f_j'(x^k) - f_j'(\phi_j^k) + \frac{1}{n}\sum_i f_i'(\phi_i^k) \right]$$

- As if the regularization term is separated into each f_i

Contents

Proof on convergence

- In this section, all expectations are taken with respect to the choice of j at iteration k+1 and conditioned on x_k and each $f_i'(\varphi_i^k)$.
- Lemma 1: Let $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$,Suppose each f_i is µ-strongly convex and has Lipschitz continuous gradients with constant L. Then for all x and x^* :

$$\langle f'(x), x^* - x\rangle \leq \frac{L-\mu}{L}\left[f(x^*) - f(x)\right] - \frac{\mu}{2}\left\|x^* - x\right\|^2$$

$$- \frac{1}{2Ln}\sum_i \left\|f_i'(x^*) - f_i'(x)\right\|^2 - \frac{\mu}{L}\left\langle f'(x^*), x - x^*\right\rangle .$$

- Property of convex functions

- Lemma 2. We have that for all φ_i and x^* :

$$\frac{1}{n}\sum_i \left\|f_i'(\phi_i) - f_i'(x^*)\right\|^2 \leq 2L\left[\frac{1}{n}\sum_i f_i(\phi_i) - f(x^*) - \frac{1}{n}\sum_i \langle f_i'(x^*), \phi_i - x^*\rangle\right].$$

- Lemma 3. It holds that for any φ_i^k,x^*,x_k and β> 0, with w^{k+1}:

$$\mathbb{E}\left\|w^{k+1} - x^k - \gamma f'(x^*)\right\|^2 \leq \gamma^2(1+\beta^{-1})\mathbb{E}\left\|f_j'(\phi_j^k) - f_j'(x^*)\right\|^2 + \gamma^2(1+\beta)\mathbb{E}\left\|f_j'(x^k) - f_j'(x^*)\right\|^2 - \gamma^2\beta\left\|f'(x^k) - f'(x^*)\right\|^2.$$

- Lemma 4. Let f be μ-strongly convex and have Lipschitz continuous gradients with constant L. Then we have for all x and y:

$$f(x) \geq f(y) + \langle f'(y), x - y\rangle + \frac{1}{2(L-\mu)}\left\|f'(x) - f'(y)\right\|^2 + \frac{\mu L}{2(L-\mu)}\|y - x\|^2 + \frac{\mu}{(L-\mu)}\langle f'(x) - f'(y), y - x\rangle.$$

- Lemma 5. Let $f(x) = \sum_{i=1}^{n} f_i(x)$. Suppose each fi is μ-strongly convex and has Lipschitz continuous gradients with constant L. Then for all x and x^* :

$$\langle f'(x), x^* - x \rangle \leq \frac{L-\mu}{L}[f(x^*) - f(x)] - \frac{\mu}{2}\|x^* - x\|^2 - \frac{1}{2Ln}\sum_i \|f_i'(x^*) - f_i'(x)\|^2 - \frac{\mu}{L}\langle f'(x^*), x - x^* \rangle .$$

- Lemma 6. We have that for all φi and x^* :

$$\frac{1}{n}\sum_i \|f_i'(\phi_i) - f_i'(x^*)\|^2 \leq 2L\left[\frac{1}{n}\sum_i f_i(\phi_i) - f(x^*) - \frac{1}{n}\sum_i \langle f_i'(x^*), \phi_i - x^* \rangle\right].$$

- Theorem 1. With x^* the optimal solution, define the Lyapunov function T as:

$$T^k := T(x^k, \{\phi_i^k\}_{i=1}^n) := \frac{1}{n}\sum_i f_i(\phi_i^k) - f(x^*) - \frac{1}{n}\sum_i \langle f_i'(x^*), \phi_i^k - x^* \rangle + c\left\|x^k - x^*\right\|^2.$$

- Then with $\gamma = \frac{1}{2(\mu N+L)}$, $c = \frac{1}{2\gamma(1-\gamma\mu)n}$, and $\kappa = \frac{1}{\gamma\mu}$, we have the following expected change in the Lyapunov function between steps of the SAGA algorithm (conditional on T^k):

$$\mathbb{E}[T^{k+1}] \le (1 - \frac{1}{\kappa})T^k.$$

- Proof. The first three terms in $T^{k+1}$are straight-forward to simplify:

$$\mathbb{E}\left[\frac{1}{n}\sum_i f_i(\phi_i^{k+1})\right] = \frac{1}{n}f(x^k) + \left(1-\frac{1}{n}\right)\frac{1}{n}\sum_i f_i(\phi_i^k).$$

$$\mathbb{E}\left[-\frac{1}{n}\sum_i \left\langle f_i'(x^*), \phi_i^{k+1} - x^*\right\rangle\right] = -\frac{1}{n}\left\langle f'(x^*), x^k - x^*\right\rangle - \left(1-\frac{1}{n}\right)\frac{1}{n}\sum_i \left\langle f_i'(x^*), \phi_i^k - x^*\right\rangle.$$

- For the change in the last term in T^{k+1}, we apply the non-expansiveness of the proximal operator :

$$\begin{aligned} c\left\|x^{k+1} - x^*\right\|^2 &= c\left\|\mathrm{prox}_\gamma(w^{k+1}) - \mathrm{prox}_\gamma(x^* - \gamma f'(x^*))\right\|^2 \\ &\leq c\left\|w^{k+1} - x^* + \gamma f'(x^*)\right\|^2. \end{aligned}$$

- We expand the quadratic and apply $E[w^{k+1}] = x^{\mathrm{k}} - \gamma f'(x^k)$ to simplify the inner product term:

$$
\begin{aligned}
&c\mathbb{E}\left\|w^{k+1} - x^* + \gamma f'(x^*)\right\|^2 = c\mathbb{E}\left\|x^k - x^* + w^{k+1} - x^k + \gamma f'(x^*)\right\|^2 \\
&= c\left\|x^k - x^*\right\|^2 + 2c\mathbb{E}\left[\left\langle w^{k+1} - x^k + \gamma f'(x^*), x^k - x^*\right\rangle\right] + c\mathbb{E}\left\|w^{k+1} - x^k + \gamma f'(x^*)\right\|^2 \\
&= c\left\|x^k - x^*\right\|^2 - 2c\gamma\left\langle f'(x^k) - f'(x^*), x^k - x^*\right\rangle + c\mathbb{E}\left\|w^{k+1} - x^k + \gamma f'(x^*)\right\|^2 \\
&\leq c\left\|x^k - x^*\right\|^2 - 2c\gamma\left\langle f'(x^k), x^k - x^*\right\rangle + 2c\gamma\left\langle f'(x^*), x^k - x^*\right\rangle - c\gamma^2\beta\left\|f'(x^k) - f'(x^*)\right\|^2 \\
&\quad + \left(1 + \beta^{-1}\right)c\gamma^2\mathbb{E}\left\|f_j'(\phi_j^k) - f_j'(x^*)\right\|^2 + (1 + \beta)\,c\gamma^2\mathbb{E}\left\|f_j'(x^k) - f_j'(x^*)\right\|^2. \quad \text{(Lemma 7)}
\end{aligned}
$$

- The value of β shall be fixed later. Now we apply Lemma 1 to bound $-2c\gamma < f'(x^k), x^k - x^* >$ and Lemma 6 to bound $E||f_j'(\varphi_j^k) - f_j'(x^*)||^2$:

$$
\begin{aligned}
c\mathbb{E}\left\|x^{k+1}-x^*\right\|^2 \leq (c - c\gamma\mu)\left\|x^k - x^*\right\|^2 + \left((1+\beta)c\gamma^2 - \frac{c\gamma}{L}\right)\mathbb{E}\left\|f_j'(x^k) - f_j'(x^*)\right\|^2 \\
- \frac{2c\gamma(L-\mu)}{L}\left[f(x^k) - f(x^*) - \langle f'(x^*), x^k - x^*\rangle\right] - c\gamma^2\beta\left\|f'(x^k) - f'(x^*)\right\|^2 \\
+ 2\left(1+\beta^{-1}\right)c\gamma^2 L\left[\frac{1}{n}\sum_i f_i(\phi_i^k) - f(x^*) - \frac{1}{n}\sum_i \langle f_i'(x^*), \phi_i^k - x^*\rangle\right].
\end{aligned}
$$

- We can now combine the bounds that we have derived for each term in T, and pull out a fraction $1/k$ of T^k (for any κ at this point). Together with the inequality $-||f'(x_k) - f'(x^*)||^2 \leq -2\mu[f(x_k) - f(x^*) - <f'(x^*), x_k - x^*>]$, that yields:

$$\begin{aligned}\mathbb{E}[T^{k+1}] - T^k \leq -\frac{1}{\kappa}T^k + \left(\frac{1}{n} - \frac{2c\gamma(L-\mu)}{L} - 2c\gamma^2\mu\beta\right)\left[f(x^k) - f(x^*) - \left\langle f'(x^*), x^k - x^*\right\rangle\right] \\ + \left(\frac{1}{\kappa} + 2(1+\beta^{-1})c\gamma^2 L - \frac{1}{n}\right)\left[\frac{1}{n}\sum_i f_i(\phi_i^k) - f(x^*) - \frac{1}{n}\sum_i \left\langle f_i'(x^*), \phi_i^k - x^*\right\rangle\right] \\ + \left(\frac{1}{\kappa} - \gamma\mu\right)c\left\|x^k - x^*\right\|^2 + \left((1+\beta)\gamma - \frac{1}{L}\right)c\gamma\mathbb{E}\left\|f_j'(x^k) - f_j'(x^*)\right\|^2. \qquad (10)\end{aligned}$$

- Adaptivity to strong convexity result:
- Note that when using the γ = 1/3L step size, the same c as above can be used with β = 2 and 1/κ = min(1/4n , μ/3L) to ensure non-positive terms.

- Corollary .
- Note that $c||x_k - x^*||^2 \leq T^k$, and therefore by chaining the expectations, plugging in the constants explicitly and using μ(n − 0.5) ≤ μn to simplify the expression, we get:

$$\mathbb{E}\left[\left\|x^k - x^*\right\|^2\right] \leq \left(1 - \frac{\mu}{2(\mu n + L)}\right)^k \left[\left\|x^0 - x^*\right\|^2 + \frac{n}{\mu n + L}\left[f(x^0) - \langle f'(x^*), x^0 - x^*\rangle - f(x^*)\right]\right].$$

- Here the expectation is over all choices of index j^k up to step k.

Contents

Experiment results

- Finito (perm) performs the best on a per epoch equivalent basis, but it can be the most expensive method per step.
- SVRG is similarly fast on a per epoch basis, but when considering the number of gradient evaluations per epoch is double that of the other methods for this problem, it is middle of the pack.
- SAGA can be seen to perform similar to the non-permuted Finito case, and to SDCA.

- SAG is slower than the other methods with constant step size.
- In general, these tests confirm that the choice of methods should be done based on their properties, rather than their convergence rate

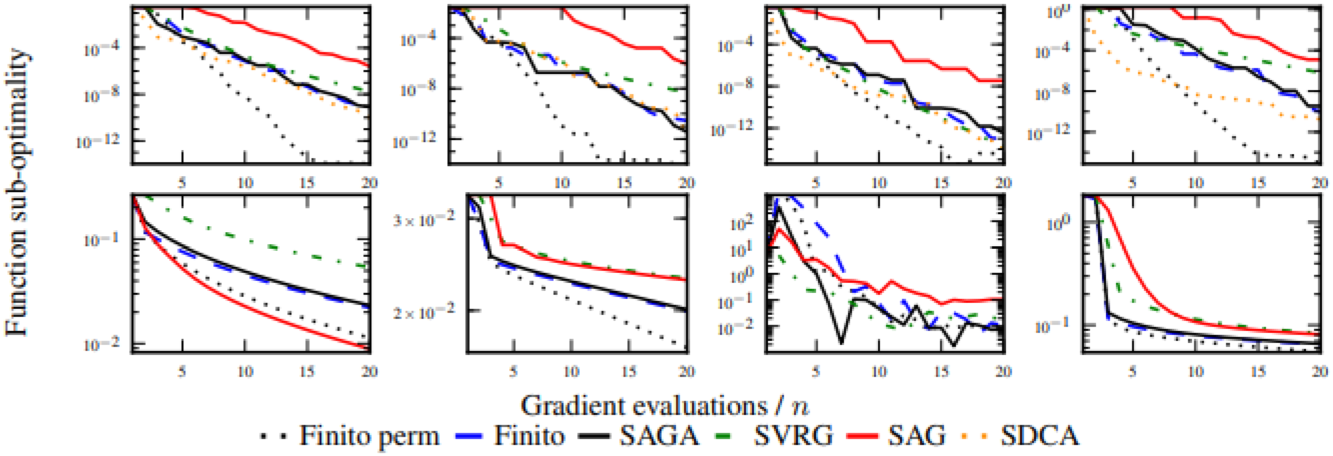

Figure 2: From left to right we have the MNIST, COVTYPE, IJCNN1 and MILLIONSONG datasets. Top row is the L2 regularised case, bottom row the L1 regularised case.

THANKS