SAN: Stochastic Average Newton Algorithm for Minimizing Finite Sums

Jiabin Chen [1,6] Rui Yuan [2,5,6] Guillaume Garrigos [3] Robert M. Gower [4,5,6]

[1]Baidu Inc. [2]Meta AI. [3]Université de Paris, Sorbonne Université, CNRS, LPSM [4]CCM, Flatiron Institute [5]LTCI, Télécom Paris [6]Institut Polytechnique de Paris

Overview

Consider the optimization problem

$$w^* \;=\; \arg\min_{w\in\mathbb{R}^d} \;\; \frac{1}{n}\sum_{i=1}^n f_i(w) \;\; \overset{\text{def}}{=} \;\; f(w), \qquad (1)$$

where f_i is strictly convex and twice differentiable, and $n, d \gg 1$.

- First-order methods: SVRG [2], SAG [5], etc. **Issue:** require parameter tuning, and/or knowledge about the problem
- Second-order methods: SQN [1], IQN [4], SNM [3]
  Issues: not incremental, or too expensive even for GLMs ($\mathcal{O}(d^2)$)

Derivation of SAN : split the functions, sample, linearize & project

The optimality condition of (1) is $\nabla f(w) = \frac{1}{n}\sum_{i=1}^n \nabla f_i(w) = 0$.

Split the optimality conditions with slack variables :

$$\frac{1}{n}\sum_{i=1}^n \alpha_i = 0, \qquad (2)$$

$$\alpha_i = \nabla f_i(w), \quad \forall i \in \{1,\ldots,n\}. \qquad (3)$$

The advantage of (2–3) is that each gradient lies on a separate equation. This motivates us to sample one equation per iteration, and project our current iterate on the linearization of this equation.

More concretely, given a current iterate $w^k, \alpha_1^k, \ldots, \alpha_n^k \in \mathbb{R}^d$:

▶ with probability $\frac{1}{n+1}$, we sample equation (2) (which is linear) and project onto its set of solutions:

$$\alpha_1^{k+1},\ldots,\alpha_n^{k+1} = \underset{\alpha_1,\ldots,\alpha_n}{\operatorname{argmin}} \quad \sum_{i=1}^n \|\alpha_i - \alpha_i^k\|^2 \;\text{ s.t. }\; \frac{1}{n}\sum_{i=1}^n \alpha_i = 0. \qquad (4)$$

▶ with probability $\frac{1}{n+1}$, we sample the j-th equation of (3), and project onto the set of solutions of its linearization at w^k:

$$\alpha_j^{k+1}, w^{k+1} = \underset{\alpha_j, w\in\mathbb{R}^d}{\operatorname{argmin}} \quad \|\alpha_j - \alpha_j^k\|^2 + \|w - w^k\|^2_{\nabla^2 f_j(w^k)} \qquad (5)$$
$$\text{s.t. } \nabla f_j(w^k) + \nabla^2 f_j(w^k)(w - w^k) = \alpha_j.$$

In this step the main iterate w^k is updated, and we chose to project it w.r.t. the metric induced by the Hessian.

Closed form expression for SAN

The projection steps in (4-5) can be made explicit:

Algorithm 1 SAN: Stochastic Average Newton

Input: $\{f_i\}_{i=1}^n$, max iteration T

Initialize $\alpha_1^0, \cdots, \alpha_n^0, w^0 \in \mathbb{R}^d$

For $k = 1, \ldots, T$ do:

Either with probability $\frac{1}{n+1}$, update:

$$\alpha_i^{k+1} = \alpha_i^k - \frac{1}{n}\sum_{j=1}^n \alpha_j^k, \quad \text{for all } i \in \{1,\cdots,n\}$$

Or with probability $\frac{1}{n+1}$, sample $j \in \{1,\cdots,n\}$ and update:

$$d^k = \left(\mathbf{I}_d + \nabla^2 f_j(w^k)\right)^{-1}\left(\nabla f_j(w^k) - \alpha_j^k\right)$$

$$w^{k+1} = w^k - d^k$$

$$\alpha_j^{k+1} = \alpha_j^k + d^k$$

Output: last iterate w^{T+1}

SAN perform Newton-like steps on w^k w.r.t sampled functions, while averaging the slack variables α_i^k from time to time.

Convergence analysis

Assumption: the f_i are μ_f-strongly convex and have a L_f-Lipschitz continuous gradient.

Theorem: Let $(w^k, \alpha_1^k, \ldots, \alpha_n^k)_{k\in\mathbb{N}}$ be a sequence generated by SAN, and $w^* = \operatorname{argmin} f$. Under **technical assumptions**, we have

$$\mathbb{E}\left[\|w^k - w^*\|^2\right] + \sum_{i=1}^n \mathbb{E}\left[\|\alpha_i^k - \nabla f_i(w^*)\|^2\right] \le C(1-\rho)^k \quad \text{a.s.}$$

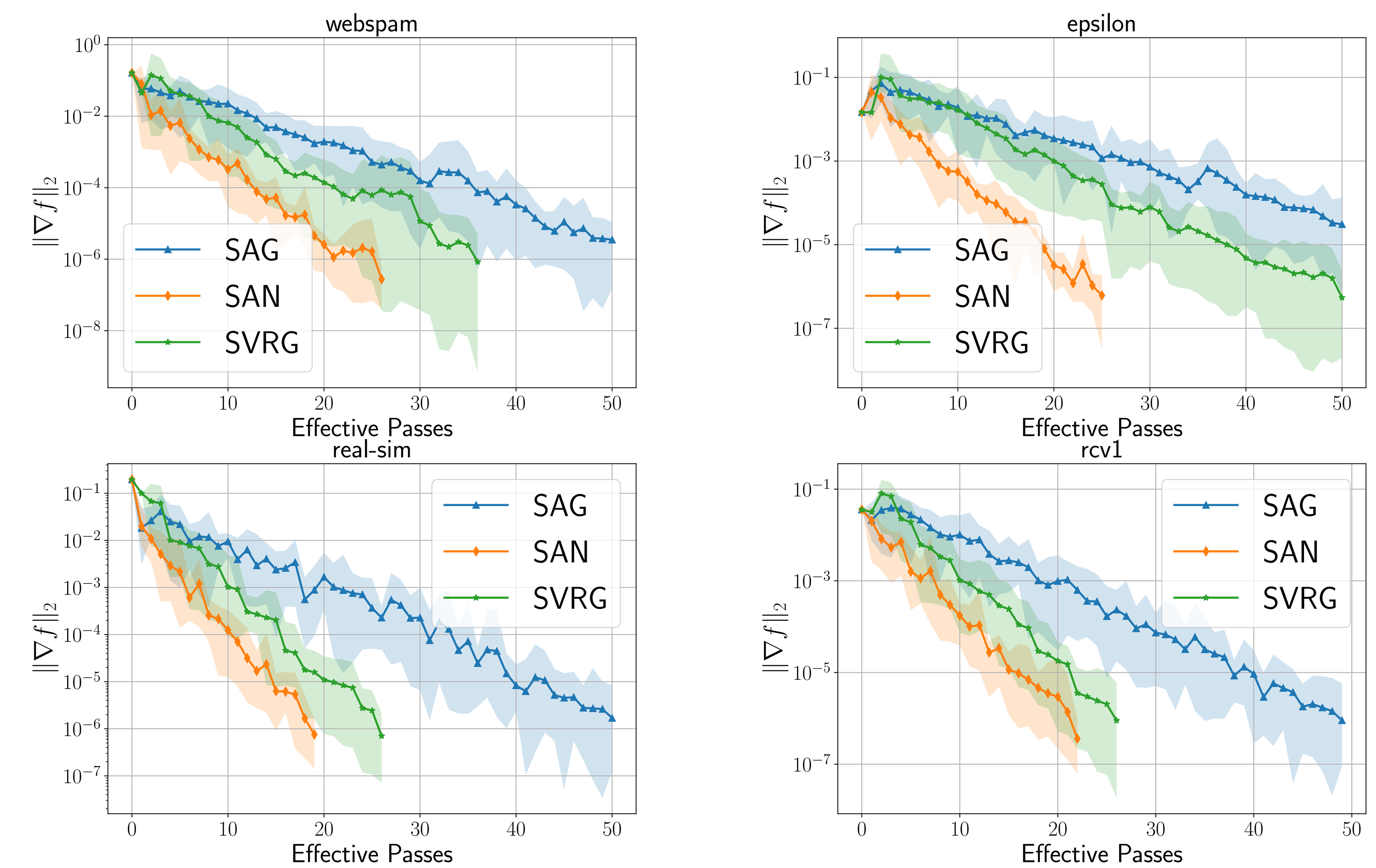
where C and ρ depend on μ_f, L_f, x^0.

Take-away

We develop a second order method that is 1) incremental, 2) efficient, 3) scales well with the dimension d, 4) requires no knowledge from the problem, 5) neither parameter tuning.

Numerical experiments

We evaluate SAN on a L2-regularized logistic regression problem:



dataset	dimension (d)	samples (n)	sparsity	condition number
webspam	$254+1$	350000	0.6648	6.9973×10^{255}
epsilon	$2000+1$	400000	0.0	3.2110×10^{10}
rcv1	$47236+1$	20242	0.9984	5.3915×10^{25}
real-sim	$20958+1$	72309	0.9976	1.3987×10^{20}

Table 1. Details of the binary data sets used in the logistic regression experiments

	memory	memory access	data access	computational cost
SAN	$\mathcal{O}(nd)$	$\boldsymbol{\mathcal{O}(d)}$	$\boldsymbol{\mathcal{O}(1)}$	$\boldsymbol{\mathcal{O}(d)}$
SAG	$\mathcal{O}(nd)$	$\boldsymbol{\mathcal{O}(d)}$	$\boldsymbol{\mathcal{O}(1)}$	$\boldsymbol{\mathcal{O}(d)}$
SVRG	$\boldsymbol{\mathcal{O}(d)}$	$\boldsymbol{\mathcal{O}(d)}$	$\boldsymbol{\mathcal{O}(1)}$	$\boldsymbol{\mathcal{O}(d)}$

Table 2. Average cost per iteration of various stochastic methods applied to GLM.

References

[1] Robert M. Gower, Donald Goldfarb, and Peter Richtárik. Stochastic block BFGS: Squeezing more curvature out of data. *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[2] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323. Curran Associates, Inc., 2013.

[3] Dmitry Kovalev, Konstantin Mishchenko, and Peter Richtarik. Stochastic Newton and cubic Newton methods with simple local linear-quadratic rates. *arxiv:1912.01597*, 2019.

[4] Aryan Mokhtari, Mark Eisen, and Alejandro Ribeiro. Iqn: An incremental quasi-newton method with local superlinear convergence rate. *SIAM Journal on Optimization*, 28(2):1670–1698, 2018.

[5] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, Mar 2017.