A Stochastic Gradient Method with an Exponential Convergence Rate for Strongly-Convex Optimization

Nicolas Le Roux[⋆†] Mark Schmidt[⋆] Francis Bach[⋆]

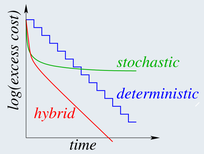
[⋆]INRIA-Sierra Project Team (INRIA/ENS/CNRS UMR 8548), 23, avenue d'Italie, 75214 Paris. France

[†]Now at Criteo, 32, rue Blanche, 75009 Paris, France

Motivation and overview of contribution

- Minimize the strongly convex sum of a finite set of smooth functions:
$$\min_{x\in\mathbb{R}^p} g(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$$
- Stochastic gradient has $O(1)$ iteration cost and $O(\frac{1}{k})$ convergence rate
- Full gradient has $O(n)$ iteration cost and $O(\rho^k)$ convergence rate
- Stochastic methods do not use the finite training set assumption.

- Stochastic methods are efficient early on.
- Batch methods are efficient later on.
- We want the best of both worlds.



- We propose a stochastic gradient method with an $O(1)$ iteration cost and a linear convergence rate.

Related work

- Stochastic version of full gradient methods
  - Schraudolph (1999), Sunehag et al. (2009), Ghadimi and Lan (2010), Martens (2010), Xiao (2010)
  - They do not improve on the $O(1/k)$ rate
- Momentum, gradient/iterate averaging
  - Polyak and Judistky (1992), Tseng (1998), Nesterov (2009), Xiao (2010), Kushner and Yin (2003), Hazan and Kale (2011), Rakhlin et al. (2012)
  - They can improve the constants and robustness but not the $O(1/k)$ rate
- Constant step-size stochastic gradient, accelerated SG
  - Kesten (1958), Delyon and Juditsky (1993), Nedic and Bertsekas (2000)
  - Linear convergence but only up to a fixed tolerance
- Hybrid methods, incremental average gradient
  - Bertsekas (1997), Blatt et al. (2007), Friedlander and Schmidt (2012)
  - Linear rate but iterations make full passes through the data

Assumptions and Stochastic Average Gradient algorithm

- Assumptions
  - Each function f_i is convex with L_i-Lipschitz continuous gradient (and $L = \max_i L_i$).
  - g is strongly convex with constant μ.
- Stochastic Average Gradient Algorithm
  - Start from $y_i^0 = 0$, $i = 1, \dots, n$.
  - Randomly select $i(k) \in \{1, \dots, n\}$ with replacement.
  - $y_i^k = \begin{cases} f_i'(x_k) & \text{if } i = i(k) \\ y_i^{k-1} & \text{if } i \neq i(k) \end{cases}$
  - SAG update: $x_{k+1} = x_k - \frac{\alpha_k}{n}\sum_{i=1}^n y_i^k$
  - Comparison with SG and FG updates:
    - SG update: $x_{k+1} = x_k - \alpha_k y_{i(k)}^k$
    - FG update: $x_{k+1} = x_k - \frac{\alpha_k}{n}\sum_{i=1}^n f_i'(x^k)$
- Intuition
  - $\lim_{k\to+\infty} \|x_k - x_{k-1}\| = 0$ so y_i^k remains close to $f_i'(x_k)$
  - Thus, SAG updates remains close to full gradient updates.

Convergence rates

Proposition 1 - Small step size

With a constant step size of $\alpha_k = \frac{1}{2nL}$, SAG iterations satisfy:

$$\mathbb{E}\left[\|x^k - x^*\|^2\right] \leqslant \left(1 - \frac{\mu}{8Ln}\right)^k \left[3\|x_0 - x^*\|^2 + \frac{9\sigma^2}{4L^2}\right].$$

- Linear rate with cost independent of n.
- Similar to batch gradient and IAG (cyclic version).

Proposition 2 - Large step size

If $n \geqslant \frac{8L}{\mu}$, with a step size of $\alpha_k = \frac{1}{2n\mu}$ SAG iterations satisfy:

$$\mathbb{E}\left[g(x^k) - g(x^*)\right] \leqslant \left(1 - \frac{1}{8n}\right)^k \left[\frac{7}{3}(g(x^0) - g(x^*)) + \frac{7\sigma^2}{6n\mu}\right].$$

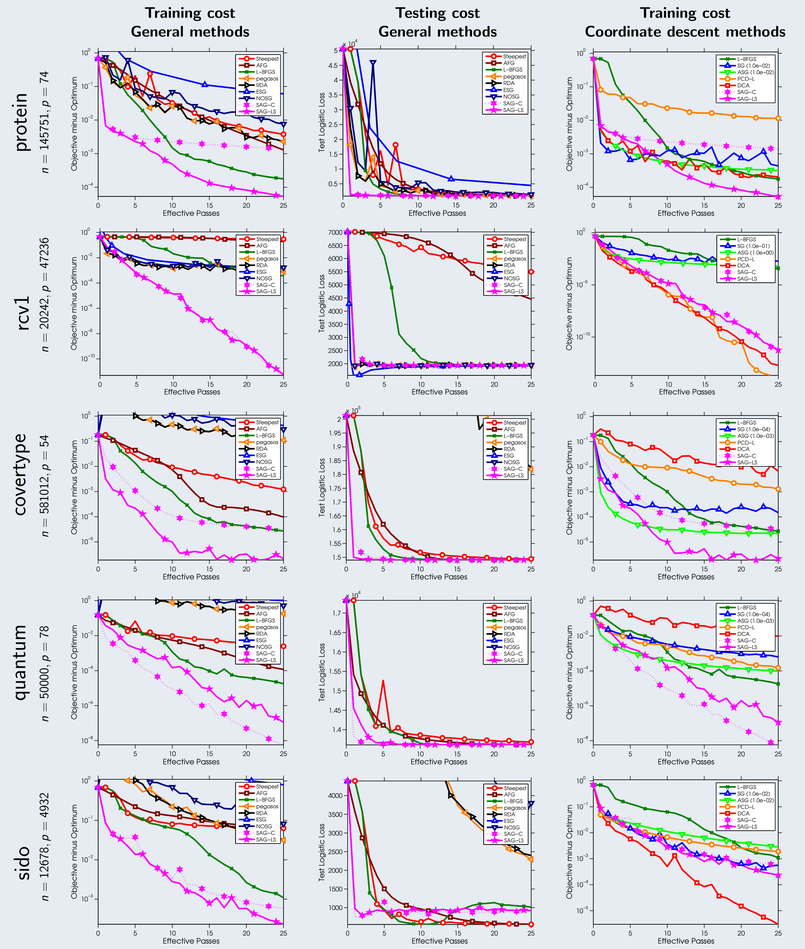
- Linear rate with cost independent of n.
- Linear rate (almost) independent of the condition number.
- The rate works with $\alpha_k = 1/16L$.
- The constant can be improved if SG is used for the first pass.

- Assume $L = 100$, $\mu = 0.01$ and $n = 80000$:
  - Full gradient has rate $\left(1 - \frac{\mu}{L}\right)^2 = 0.9998$
  - Accelerated gradient has rate $\left(1 - \sqrt{\frac{\mu}{L}}\right) = 0.9900$
  - SAG (n iterations) has rate $\left(1 - \frac{1}{8n}\right)^n = 0.8825$
  - Fastest possible first-order method has rate $\left(\frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}\right)^2 = 0.9608$
- For FG methods, L depends on g and not on the f_i's.

Implementation details

- Memory requirements
  - For most problems, $O(np)$ storage can be avoided.
  - For $f_i(x) = f(a_i^T x)$, we can reduce the storage to $O(n)$.
  - Mini-batches reduce storage requirement (one gradient per mini-batch).
  - Proposition 2 offers guidance to find a good mini-batch size.
- Line search
  - Optimal step-size for SAG depends on the Lipschitz constant of the f_i'.
  - We propose a heuristic line-search for approximating each L_i.
  - This allows for automatic selection of the step-size.
- Extensions
  - SAG allows non-uniform sampling, which might improve convergence speed.
  - SAG seems amenable to proximal, coordinate-wise and Newton-like variants.

Experiments (ℓ_2-regularized logistic regression)



Discussion

- We expect a $O(1/k)$ rate on convex problems.
- SAG uses old gradients and thus seems suited to parallelization.
- SAG allows adaptive step-sizes and offers a termination criterion.
- SAG enjoys faster convergence than IAG. What is the influence of random sampling?
- We conjecture that, with a step-size of $\alpha_k = \frac{1}{L}$, SAG converges in all settings with rate $\mathbb{E}\left[g(x^k) - g(x^*)\right] \leqslant C\left[1 - \min\left(\frac{1}{8n}, \frac{\mu}{L}\right)\right]^{2k}$.