Stochastic Optimization Methods

Lecturer: Pradeep Ravikumar
Co-instructor: Aarti Singh

Convex Optimization 10-725/36-725

Adapted from slides from Ryan Tibshirani

Stochastic gradient descent

Consider sum of functions

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

Gradient descent applied to this problem would repeat

$$x^{(k)} = x^{(k-1)} - t_k \cdot \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^{(k-1)}), \quad k = 1, 2, 3, \ldots$$

In comparison, stochastic gradient descent (or incremental gradient descent) repeats

$$x^{(k)} = x^{(k-1)} - t_k \cdot \nabla f_{i_k}(x^{(k-1)}), \quad k = 1, 2, 3, \ldots$$

where $i_k \in \{1, \ldots n\}$ is some chosen index at iteration k

Notes:

- Typically we make a (uniform) random choice $i_k \in \{1, \ldots n\}$
- Also common: mini-batch stochastic gradient descent, where we choose a random subset $I_k \subset \{1, \ldots n\}$, of size $b \ll n$, and update according to

$$x^{(k)} = x^{(k-1)} - t_k \cdot \frac{1}{b} \sum_{i \in I_k} \nabla f_i(x^{(k-1)}), \quad k = 1, 2, 3, \ldots$$

- In both cases, we are approximating the full graident by a noisy estimate, and our noisy estimate is unbiased

$$\mathbb{E}[\nabla f_{i_k}(x)] = \nabla f(x)$$

$$\mathbb{E}\left[\frac{1}{b} \sum_{i \in I_k} \nabla f_i(x)\right] = \nabla f(x)$$

The mini-batch reduces the variance by a factor $1/b$, but is also b times more expensive!

Example: regularized logistic regression

Given labels $y_i \in \{0, 1\}$, features $x_i \in \mathbb{R}^p$, $i = 1, \ldots n$. Consider logistic regression with ridge regularization:

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \Big(- y_i x_i^T \beta + \log(1 + e^{x_i^T \beta}) \Big) + \frac{\lambda}{2} \|\beta\|_2^2$$

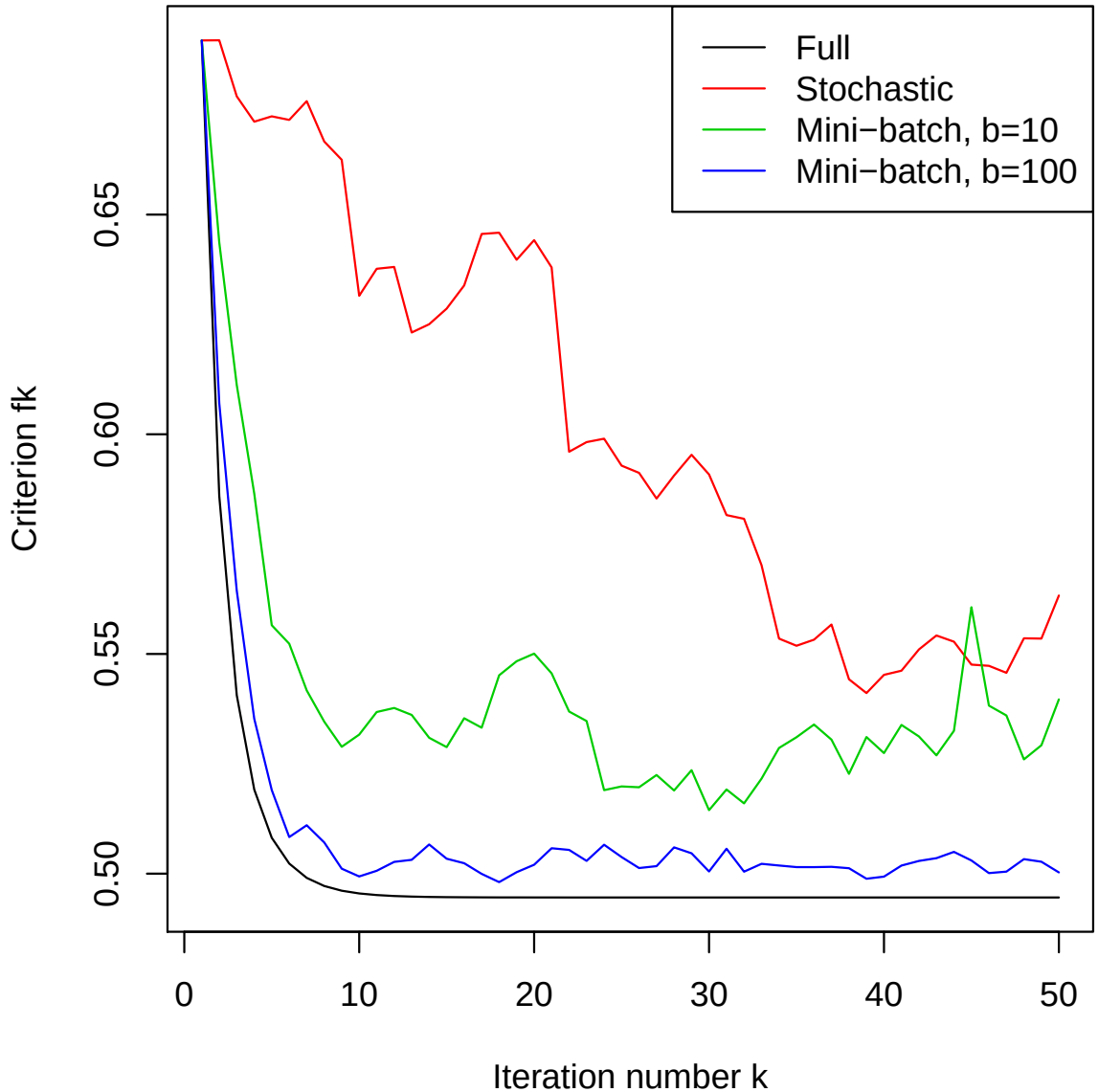
Write the criterion as

$$f(\beta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\beta), \quad f_i(\beta) = -y_i x_i^T \beta + \log(1 + e^{x_i^T \beta}) + \frac{\lambda}{2} \|\beta\|_2^2$$

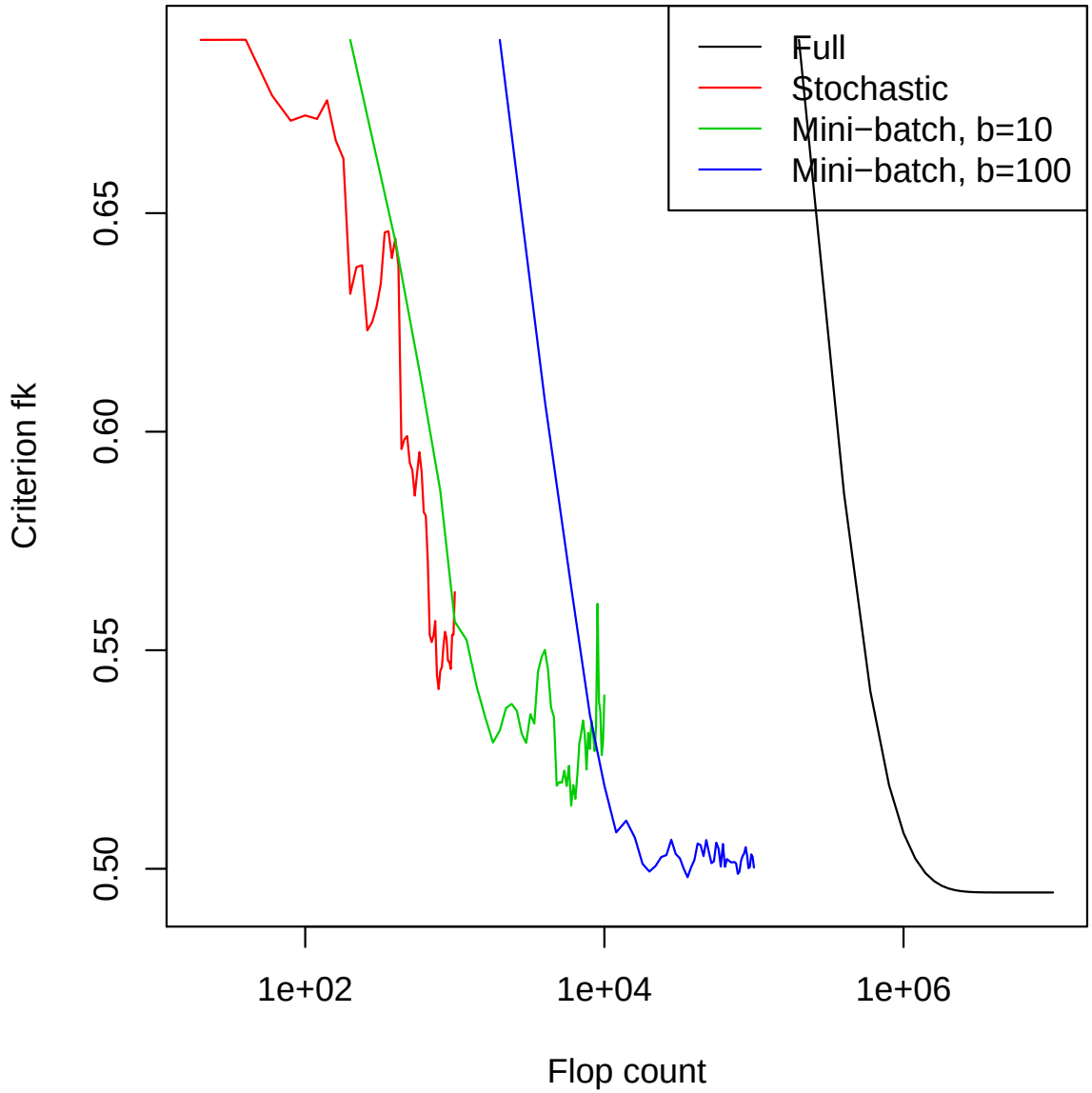
The gradient computation $\nabla f(\beta) = \sum_{i=1}^{n} \big(y_i - p_i(\beta)\big) x_i + \lambda \beta$ is doable when n is moderate, but not when n is huge. Note that:

- One batch update costs $O(np)$
- One stochastic update costs $O(p)$
- One mini-batch update costs $O(bp)$

Example with $n = 10,000$, $p = 20$, all methods employ fixed step sizes (diminishing step sizes give roughly similar results):



What's happening? Iterations make better progress as mini-batch size b gets bigger. But now let's parametrize by flops:



Convergence rates

Recall that, under suitable step sizes, when f is convex and has a Lipschitz gradient, full gradient (FG) descent satisfies

$$f(x^{(k)}) - f^\star = O(1/k)$$

What about stochastic gradient (SG) descent? Under diminishing step sizes, when f is convex (plus other conditions)

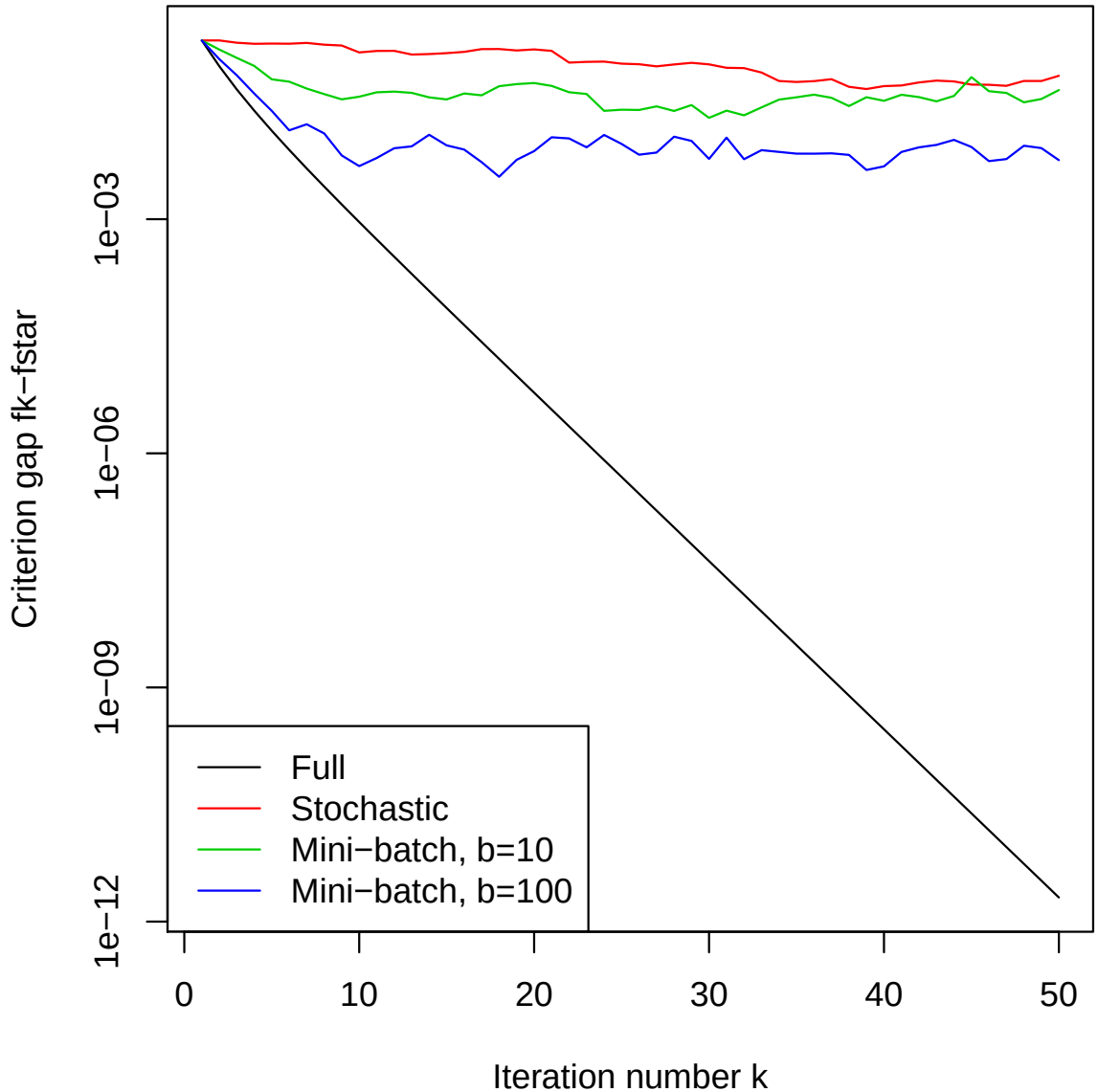
$$\mathbb{E}[f(x^{(k)})] - f^\star = O(1/\sqrt{k})$$

Finally, what about mini-batch stochastic gradient? Again, under diminishing step sizes, for f convex (plus other conditions)

$$\mathbb{E}[f(x^{(k)})] - f^\star = O(1/\sqrt{bk} + 1/k)$$

But each iteration here b times more expensive ... and (for small b), in terms of flops, this is the same rate

Back to our ridge logistic regression example, we gain important insight by looking at suboptimality gap (on log scale):



Recall that, under suitable step sizes, when f is strongly convex with a Lipschitz gradient, gradient descent satisfies

$$f(x^{(k)}) - f^\star = O(\rho^k)$$

where $\rho < 1$. But, under diminishing step sizes, when f is strongly convex (plus other conditions), stochastic gradient descent gives

$$\mathbb{E}[f(x^{(k)})] - f^\star = O(1/k)$$

So stochastic methods do not enjoy the linear convergence rate of gradient descent under strong convexity

For a while, this was believed to be inevitable, as Nemirovski and others had established matching lower bounds ... but these applied to stochastic minimization of criterions, $f(x) = \int F(x, \xi)\, d\xi$. *Can we do better for finite sums?*

Stochastic Gradient: Bias

For solving $\min_x \ f(x)$, stochastic gradient is actually a class of algorithms that use the iterates:

$$x^{(k)} = x^{(k-1)} - \eta_k \, g(x^{(k-1)}; \xi_k),$$

where $g(x^{(k-1)}, \xi_k)$ is a *stochastic gradient* of the objective $f(x)$ evaluated at $x^{(k-1)}$.

Bias: The bias of the stochastic gradient is defined as:

$$\text{bias}(g(x^{(k-1)}; \xi_k)) := \mathbb{E}_{\xi_k}(g(x^{(k-1)}; \xi_k)) - \nabla f(x^{(k-1)}).$$

Unbiased: When $\mathbb{E}_{\xi_k}(g(x^{(k-1)}; \xi_k)) = \nabla f(x^{(k-1)})$, the stochastic gradient is said to be unbiased. (e.g. the stochastic gradient scheme discussed so far)
Biased: We might also be interested in biased estimators, but where the bias is small, so that $\mathbb{E}_{\xi_k}(g(x^{(k-1)}; \xi_k)) \approx \nabla f(x^{(k-1)})$.

Stochastic Gradient: Variance

Variance. In addition to small (or zero) bias, we also want the variance of the estimator to be small:

$$\begin{aligned}\text{variance}(g(x^{(k-1)}, \xi_k)) &:= \mathbb{E}_{\xi_k}(g(x^{(k-1)}, \xi_k) - \mathbb{E}_{\xi_k}(g(x^{(k-1)}, \xi_k)))^2 \\ &\leq \mathbb{E}_{\xi_k}(g(x^{(k-1)}, \xi_k))^2.\end{aligned}$$

The caveat with the stochastic gradient scheme we have seen so far is that its variance is large, and in particular doesn't decay to zero with the iteration index.

Loosely: because of above, we have to decay the step size η_k to zero, which in turn means we can't take "large" steps, and hence the convergence rate is slow.

Can we get the variance to be small, and decay to zero with iteration index?

Variance Reduction

Consider an estimator X for a parameter θ.
Note that for an unbiased estimator, $\mathbb{E}(X) = \theta$.

Now consider the following modified estimator: $Z := X - Y$, such that $\mathbb{E}(Y) \approx 0$. Then the bias of Z is also close to zero, since

$$\mathbb{E}(Z) = \mathbb{E}(X) - \mathbb{E}(Y) \approx \theta.$$

If $\mathbb{E}(Y) = 0$, then Z is unbiased iff X is unbiased.

What about the **variance** of estimator X?
$\mathrm{Var}(X - Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) - 2\,\mathrm{Cov}(X, Y)$. This can be seen to much less than $\mathrm{Var}(X)$ if Y is highly correlated with X.
Thus, given any estimator X, we can reduce its variance, if we can construct a Y that (a) has expectation (close to) zero, and (b) is highly correlated with X. This is the abstract template followed by SAG, SAGA, SVRG, SDCA, ...

Outline

Rest of today:

- Stochastic average gradient (SAG)
- SAGA (does this stand for something?)
- Stochastic Variance Reduced Gradient (SVRG)
- Many, many others

Stochastic average gradient

Stochastic average gradient or SAG (Schmidt, Le Roux, Bach 2013) is a breakthrough method in stochastic optimization. Idea is fairly simple:

- Maintain table, containing gradient g_i of f_i, $i = 1, \ldots n$
- Initialize $x^{(0)}$, and $g_i^{(0)} = x^{(0)}$, $i = 1, \ldots n$
- At steps $k = 1, 2, 3, \ldots$, pick a random $i_k \in \{1, \ldots n\}$ and then let

$$g_{i_k}^{(k)} = \nabla f_i(x^{(k-1)}) \quad \text{(most recent gradient of } f_i\text{)}$$

  Set all other $g_i^{(k)} = g_i^{(k-1)}$, $i \neq i_k$, i.e., these stay the same
- Update

$$x^{(k)} = x^{(k-1)} - t_k \cdot \frac{1}{n} \sum_{i=1}^{n} g_i^{(k)}$$

Notes:

- Key of SAG is to allow each f_i, $i = 1, \ldots n$ to communicate a part of the gradient estimate at each step
- This basic idea can be traced back to incremental aggregated gradient (Blatt, Hero, Gauchman, 2006)
- SAG gradient estimates are no longer unbiased, but they have greatly reduced variance
- Isn't it expensive to average all these gradients? (Especially if n is huge?) This is basically just as efficient as stochastic gradient descent, as long we're clever:

$$x^{(k)} = x^{(k-1)} - t_k \cdot \underbrace{\left(\frac{g_{i_k}^{(k)}}{n} - \frac{g_{i_k}^{(k-1)}}{n} + \underbrace{\frac{1}{n} \sum_{i=1}^{n} g_i^{(k-1)}}_{\text{old table average}} \right)}_{\text{new table average}}$$

SAG Variance Reduction

Stochastic gradient in SAG:

$$\underbrace{g_{i_k}^{(k)}}_{X} - \underbrace{(g_{i_k}^{(k-1)} - \sum_{i=1}^{n} g_i^{(k-1)})}_{Y}.$$

It can be seen that $\mathbb{E}(X) = \nabla f(x^{(k)})$.
But that $\mathbb{E}(Y) \neq 0$, so that we have a *biased* estimator.

But we do have that Y seems correlated with X (in line with variance reduction template). In particular, we have that $X - Y \to 0$, as $k \to \infty$, since $x^{(k-1)}$ and $x^{(k)}$ converge to $\bar{x}$, the difference between first two terms converges to zero, and the last term converges to gradient at optimum, i.e. also to zero.

Thus, the overall estimator ℓ_2 norm (and accordingly its variance) decays to zero.

SAG convergence analysis

Assume that $f(x) = \frac{1}{n}\sum_{i=1}^n f_i(x)$, where each f_i is differentiable, and ∇f_i is Lipschitz with constant L

Denote $\bar{x}^{(k)} = \frac{1}{k}\sum_{\ell=0}^{k-1} x^{(\ell)}$, the average iterate after $k-1$ steps

Theorem (Schmidt, Le Roux, Bach): SAG, with a fixed step size $t = 1/(16L)$, and the initialization

$$g_i^{(0)} = \nabla f_i(x^{(0)}) - \nabla f(x^{(0)}), \quad i = 1, \ldots n$$

satisfies

$$\mathbb{E}[f(\bar{x}^{(k)})] - f^\star \leq \frac{48n}{k}\big(f(x^{(0)}) - f^\star\big) + \frac{128L}{k}\|x^{(0)} - x^\star\|_2^2$$

where the expectation is taken over the random choice of index at each iteration

Notes:

- Result stated in terms of the average iterate $\bar{x}^{(k)}$, but also can be shown to hold for best iterate $x_{\text{best}}^{(k)}$ seen so far
- This is $O(1/k)$ convergence rate for SAG. Compare to $O(1/k)$ rate for FG, and $O(1/\sqrt{k})$ rate for SG
- But, the constants are different! Bounds after k steps:

$$\text{SAG}: \quad \frac{48n}{k}\big(f(x^{(0)}) - f^\star\big) + \frac{128L}{k}\|x^{(0)} - x^\star\|_2^2$$

$$\text{FG}: \quad \frac{L}{2k}\|x^{(0)} - x^\star\|_2^2$$

$$\text{SG}^*: \quad \frac{L\sqrt{5}}{\sqrt{2k}}\|x^{(0)} - x^\star\|_2 \quad \text{(*not a real bound, loose translation)}$$

- So first term in SAG bound suffers from factor of n; authors suggest smarter initialization to make $f(x^{(0)}) - f^\star$ small (e.g., they suggest using result of n SG steps)

Convergence analysis under strong convexity

Assume further that each f_i is strongly convex with parameter m

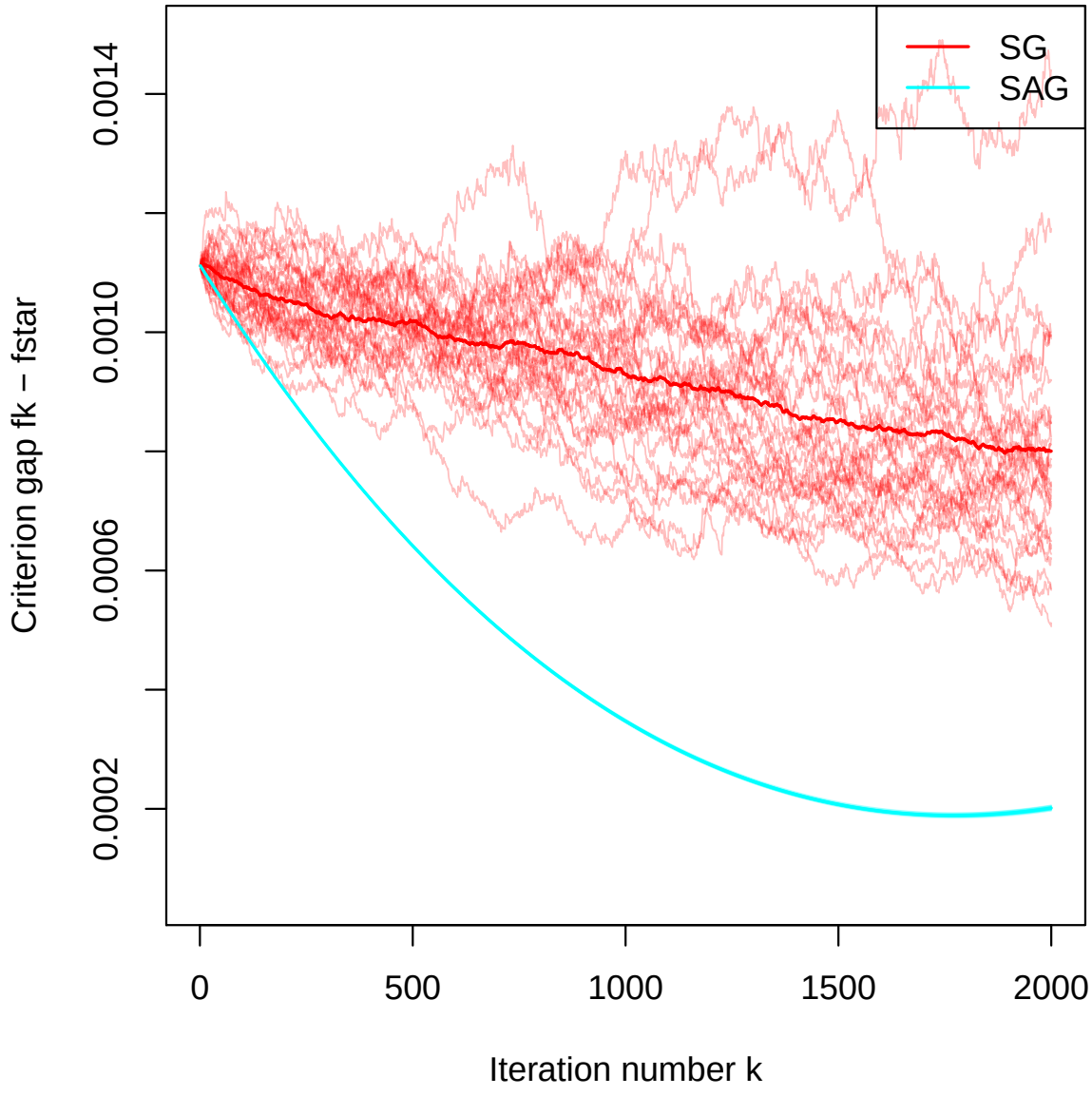
Theorem (Schmidt, Le Roux, Bach): SAG, with a step size $t = 1/(16L)$ and the same initialization as before, satisfies

$$\mathbb{E}[f(x^{(k)})] - f^\star \leq \left(1 - \min\left\{\frac{m}{16L}, \frac{1}{8n}\right\}\right)^k \cdot \left(\frac{3}{2}(f(x^{(0)}) - f^\star) + \frac{4L}{n}\|x^{(0)} - x^\star\|_2^2\right)$$

More notes:

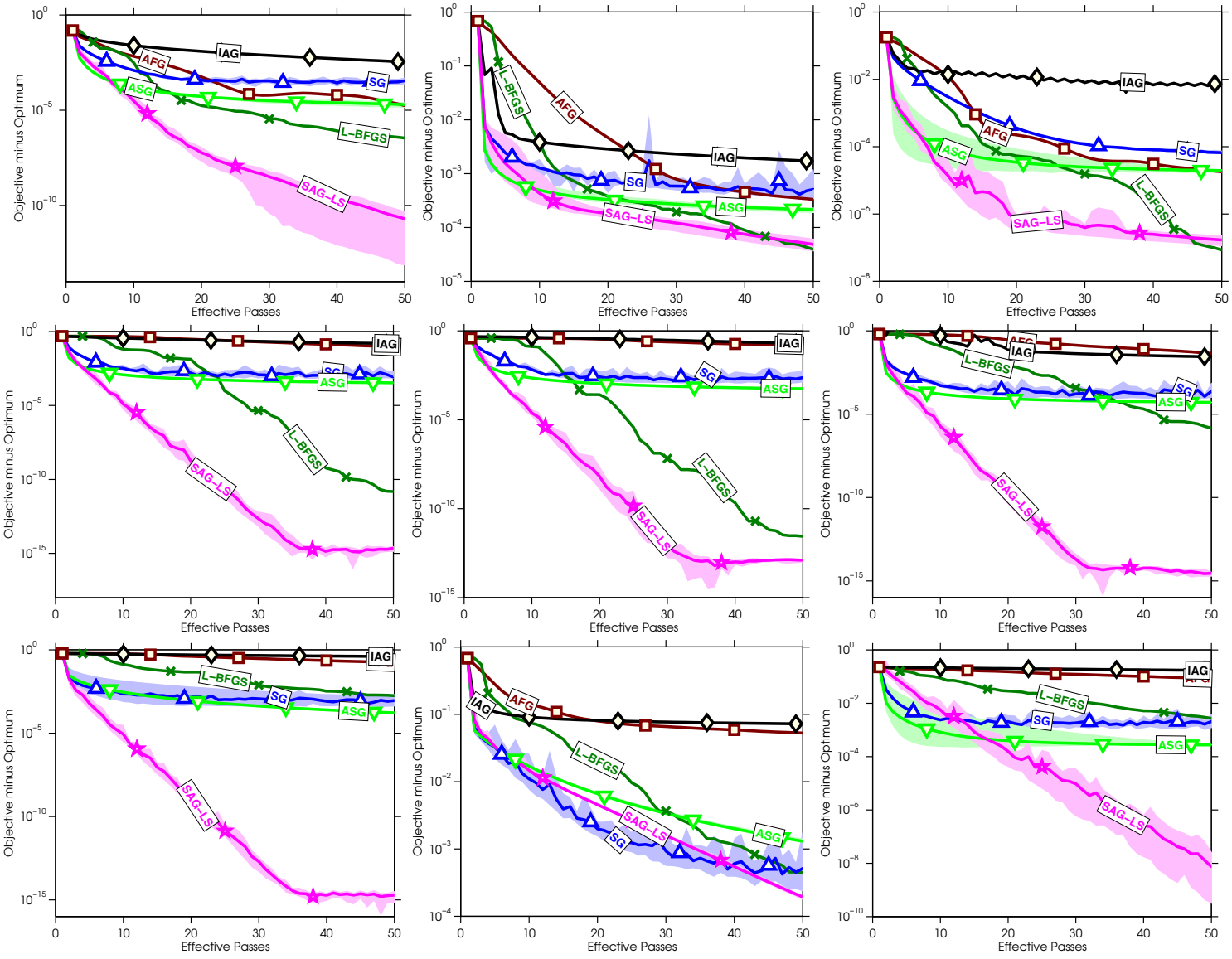
- This is linear convergence rate $O(\rho^k)$ for SAG. Compare this to $O(\rho^k)$ for FG, and only $O(1/k)$ for SG
- Like FG, we say SAG is adaptive to strong convexity (achieves better rate with same settings)
- Proofs of these results not easy: 15 pages, computed-aided!

Back to our ridge logistic regression example, SG versus SAG, over 30 reruns of these randomized algorithms:



- SAG does well, but did not work out of the box; required a specific setup
- Took one full cycle of SG (one pass over the data) to get $\beta^{(0)}$, and then started SG and SAG both from $\beta^{(0)}$. This warm start helped a lot
- SAG initialized at $g_i^{(0)} = \nabla f_i(\beta^{(0)})$, $i = 1, \ldots n$, computed during initial SG cycle. Centering these gradients was much worse (and so was initializing them at 0)
- Tuning the fixed step sizes for SAG was very finicky; here now hand-tuned to be about as large as possible before it diverges

Experiments from Schmidt, Le Roux, Bach (each plot is a different problem setting):



SAGA

SAGA (Defazio, Bach, Lacoste-Julien, 2014) is another recent stochastic method, similar in spirit to SAG. Idea is again simple:

- Maintain table, containing gradient g_i of f_i, $i = 1, \ldots n$
- Initialize $x^{(0)}$, and $g_i^{(0)} = x^{(0)}$, $i = 1, \ldots n$
- At steps $k = 1, 2, 3, \ldots$, pick a random $i_k \in \{1, \ldots n\}$ and then let

$$g_{i_k}^{(k)} = \nabla f_i(x^{(k-1)}) \quad \text{(most recent gradient of } f_i\text{)}$$

  Set all other $g_i^{(k)} = g_i^{(k-1)}$, $i \neq i_k$, i.e., these stay the same
- Update

$$x^{(k)} = x^{(k-1)} - t_k \cdot \left(g_{i_k}^{(k)} - g_{i_k}^{(k-1)} + \frac{1}{n} \sum_{i=1}^{n} g_i^{(k-1)} \right)$$

Notes:

- SAGA gradient estimate $g_{i_k}^{(k)} - g_{i_k}^{(k-1)} + \frac{1}{n}\sum_{i=1}^n g_i^{(k-1)}$, versus SAG gradient estimate $\frac{1}{n} g_{i_k}^{(k)} - \frac{1}{n} g_{i_k}^{(k-1)} + \frac{1}{n}\sum_{i=1}^n g_i^{(k-1)}$
- Recall, SAG estimate is biased; remarkably, SAGA estimate is unbiased!

SAGA Variance Reduction

Stochastic gradient in SAGA:

$$\underbrace{g_{i_k}^{(k)}}_{X} - \underbrace{(g_{i_k}^{(k-1)} - \frac{1}{n}\sum_{i=1}^{n} g_i^{(k-1)})}_{Y}.$$

It can be seen that $\mathbb{E}(X) = \nabla f(x^{(k)})$.
And that $\mathbb{E}(Y) \neq 0$, so that we have an *unbiased* estimator.

Moreover, we have that Y seems correlated with X (in line with variance reduction template). In particular, we have that $X - Y \to 0$, as $k \to \infty$, since $x^{(k-1)}$ and $x^{(k)}$ converge to $\bar{x}$, the difference between first two terms converges to zero, and the last term converges to gradient at optimum, i.e. also to zero.

Thus, the overall estimator ℓ_2 norm (and accordingly its variance) decays to zero.

- SAGA basically matches strong convergence rates of SAG (for both Lipschitz gradients, and strongly convex cases), but the proofs here much simpler
- Another strength of SAGA is that it can extend to composite problems of the form

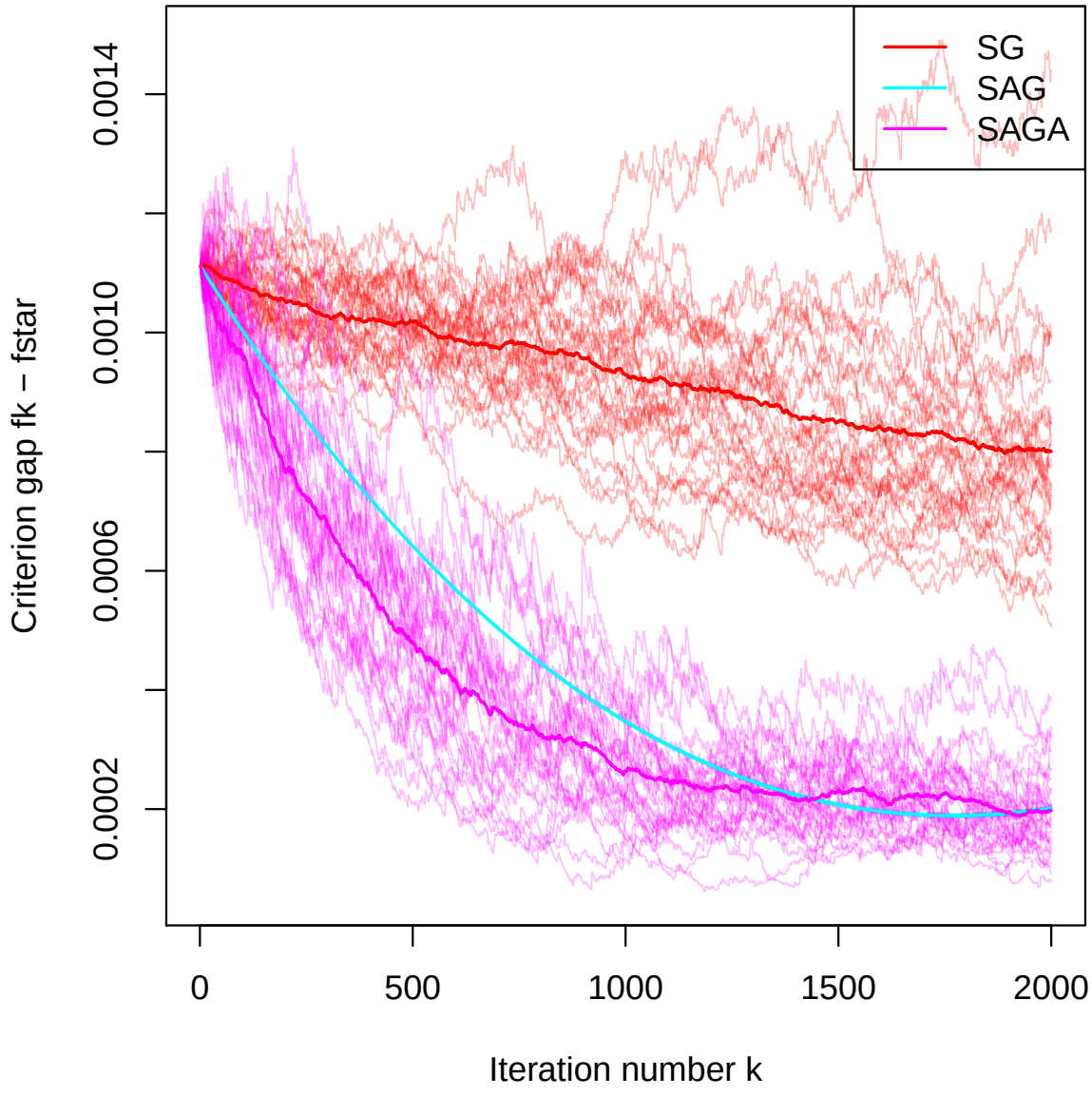
$$\min_x \frac{1}{n}\sum_{i=1}^m f_i(x) + h(x)$$

where each f_i is smooth and convex, and h is convex and nonsmooth but has a known prox. The updates are now

$$x^{(k)} = \text{prox}_{h,t_k}\left(x^{(k-1)} - t_k \cdot \left(g_i^{(k)} - g_i^{(k-1)} + \frac{1}{n}\sum_{i=1}^n g_i^{(k-1)}\right)\right)$$

- It is not known whether SAG is generally convergent under such a scheme

Back to our ridge logistic regression example, now adding SAGA to the mix:



- SAGA does well, but again it required somewhat specific setup
- As before, took one full cycle of SG (one pass over the data) to get $\beta^{(0)}$, and then started SG, SAG, SAGA all from $\beta^{(0)}$. This warm start helped a lot
- SAGA initialized at $g_i^{(0)} = \nabla f_i(\beta^{(0)})$, $i = 1, \ldots n$, computed during initial SG cycle. Centering these gradients was much worse (and so was initializing them at 0)
- Tuning the fixed step sizes for SAGA was fine; seemingly on par with tuning for SG, and more robust than tuning for SAG
- Interestingly, the SAGA criterion curves look like SG curves (realizations being jagged and highly variable); SAG looks very different, and this really emphasizes the fact that its updates have much lower variance

Stochastic Variance Reduced Gradient (SVRG)

The Stochastic Variance Reduced Gradient (SVRG) algorithm (Johnson, Zhang, 2013) runs in epochs:

- Initialize $\widetilde{x}^{(0)}$.
- For $k = 1, \ldots$:
  - Set $\widetilde{x} = \widetilde{x}^{(k-1)}$.
  - Compute $\widetilde{\mu} := \nabla f(\tilde{x})$.
  - Set $x^{(0)} = \widetilde{x}$. For $\ell = 1, \ldots, m$:
    - Pick coordinate i_ℓ at random from $\{1, \ldots, n\}$.
    - Set:

$$x^{(\ell)} = x^{(\ell-1)} - \eta \left(\nabla f_{i_\ell}(x^{(\ell-1)}) - \nabla f_{i_\ell}(\widetilde{x}) + \tilde{\mu} \right).$$

  - Set $\widetilde{x}^{(k)} = x^{(m)}$.

Stochastic Variance Reduced Gradient (SVRG)

Just like SAG/SAGA, but does not store a full table of gradients, just an average, and updates this occasionally.

SAGA: $\nabla f_{i_k}^{(k)} - \nabla f_{i_k}^{(k-1)} + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i^{(k-1)}$,
SVRG: $\nabla f_{i_\ell}(x^{(\ell-1)}) - \nabla f_{i_\ell}(\widetilde{x}) + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\widetilde{x})$.

Can be shown to achieve variance reduction similar to SAGA.

Convergence rates similar to SAGA, but formal analysis much simpler.

Many, many others

A lot of recent work revisiting stochastic optimization:

- SDCA (Shalev-Schwartz, Zhang, 2013): applies randomized coordinate ascent to the dual of ridge regularized problems. Effective primal updates are similar to SAG/SAGA.
- There's also S2GD (Konecny, Richtarik, 2014), MISO (Mairal, 2013), Finito (Defazio, Caetano, Domke, 2014), etc.

	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	✓	✓	✗	?	?

(From Defazio, Bach, Lacoste-Julien, 2014)

- Are we approaching optimality with these methods? Agarwal and Bottou (2014) recently proved nonmatching lower bounds for minimizing finite sums
- Leaves three possibilities: (i) algorithms we currently have are not optimal; (ii) lower bounds can be tightened; or (iii) upper bounds can be tightened
- Very active area of research, this will likely be sorted out soon

References and further reading

- D. Bertsekas (2010), "Incremental gradient, subgradient, and proximal methods for convex optimization: a survey"
- A. Defasio and F. Bach and S. Lacoste-Julien (2014), "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives"
- R. Johnson and T. Zhang (2013), "Accelerating stochastic gradient descent using predictive variance reduction"
- A. Nemirosvki and A. Juditsky and G. Lan and A. Shapiro (2009), "Robust stochastic optimization approach to stochastic programming"
- M. Schmidt and N. Le Roux and F. Bach (2013), "Minimizing finite sums with the stochastic average gradient"
- S. Shalev-Shwartz and T. Zhang (2013), "Stochastic dual coordinate ascent methods for regularized loss minimization"