IE 598: Big Data Optimization **Fall 2016**

Lecture 23: Incremental Gradient Methods – November 10

Lecturer: Niao He *Scribers: Shripad Gade*

Overview: In this lecture we study incremental gradient algorithms for finite sum optimization problems. Incremental gradient algorithms enjoy both the linear convergence (like gradient descent) and the cheap iteration cost (like stochastic gradient descent). We will provide a brief survey on the recently developed incremental gradient algorithms, including SAG, SAGA, SVRG, S2GD, Finito, etc. Specifically, we will analyze the convergence of the Stochastic Variance Reduced Gradient (SVRG) algorithm.

23.1 Finite Sum Problems

Problems where the objective function can be defined as a finite sum of functions, are called finite sum problems, or big-n problem. Formally, a finite sum problem can be written as,

$$\min_{x\in\mathbb{R}^d} \quad f(x) \triangleq \frac{1}{n}\sum_{i=1}^{n} f_i(x). \tag{23.1}$$

Notice that the structure is similar to sample average approximation described in Lecture 21. The number of functions, n, is analogous to the sample size drawn in Monte Carlo sampling (i.i.d. samples). Such type of problems are popular in many applications.

- **Empirical risk minimizatin:** In machine learning problems, the risk associated with a hypothesis (h) is approximated by an empirical risk $R(h)$, defined as the loss over the dataset $(x_1, y_1), \ldots, (x_n, y_n)$. Empirical risk given by $R(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x_i), y_i))$, has the structure of a finite sum problem.
- **Distributed optimization:** Distributed optimization [Ned09] involves a finite sum problem being solved by a group of computational entities (agents). Each agent has access to only a part of the finite sum, however by using an iterative consensus and local gradient based algorithm, one can show the convergence of local state estimate to the optimum. Interested readers are pointed to Prof. Tsitsiklis PhD thesis [Tsi84] which is one of the first works that discussed consensus and distributed optimization ideas.

We are interested in solving convex finite sum optimization problem as posed in (23.1). Suppose $f(x)$ is L-smooth and μ-strongly convex. We have studied two methods for solving convex optimization problem: Gradient Descent method, and Stochastic Gradient Descent. We begin by summarizing convergence results for both these algorithms when applied to the finite sum optimization problem.

Gradient Descent

The gradient descent update equation can be written as,

$$x^{t+1} = x^t - \eta\nabla f(x^t) = x^t - \eta\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x^t). \tag{23.2}$$

We can summarize the convergence properties derived in previous lectures,

- Convergence rate: linear rate, $\mathcal{O}((\frac{\kappa-1}{\kappa+1})^{2t})$, where $\kappa = \frac{L}{\mu}$ is the condition number.
- Iteration cost: $\mathcal{O}(n)$ ... (n gradients are computed for each iteration).
- Overall complexity: $\mathcal{O}(n\,\frac{L}{\mu}\log(\frac{1}{\epsilon}))$.

Stochastic Gradient Descent

The finite sum problem can be rewritten as a stochastic optimization problem,

$$\min_{x\in\mathbb{R}^d} \; f(x) = \mathbb{E}_I[f_I(x)], \tag{23.3}$$

where I is a uniform discrete random variable so that $\mathbb{P}(I=i)=\frac{1}{n}$. The SGD update equation is given by,

$$x^{t+1} = x^t - \eta^t \nabla f_{i_t}(x^t), \text{ where } i_t \sim I \tag{23.4}$$

We can summarize the convergence properties of SGD as derived in previous lectures,

- Convergence rate: sub-linear rate, $\mathcal{O}(\frac{L}{\mu^2 t})$.
- Iteration cost: $\mathcal{O}(1)$ ... (Only 1 gradient is computed for each iteration).
- Overall complexity: $\mathcal{O}(\frac{L}{\mu^2\epsilon})$.

Motivation

To summarize, when solving problems with finite sum structure

- GD enjoys a linear rate but $O(n)$ iteration cost
- SGD enjoys a sublinear rate but with $O(1)$ iteration cost

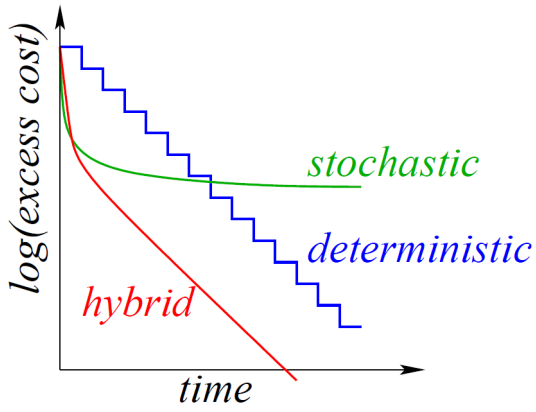

Figure 23.1: GD vs. SGD (Fig from [Bach13])

We would like to develop algorithms that have linear convergence (like GD) and cheap iteration cost (like SGD).

23.2 Brief Survey on Incremental Gradient Algorithms

Incremental gradient descent algorithms were developed to have such characteristics, and hence form an important class of algorithms. A list of few popular incremental algorithms is given below. A detailed summary and comparison among these algorithms is provided in Table 23.1.

- Deterministic Incremental Gradient Algorithms
  - *Incremental Gradient Descent* (IGD) - Bertsekas, 1997 [Ber97]
  - *Incremental Aggregated Gradient* (IAG) - Blatt *et.al*, 2007 [Bla07]
- Stochastic Incremental Gradient Algorithms
  - *Stochastic Average Gradient* (SAG) - Schmidt *et al.*, 2013 [Sch13]
  - *SAGA* - Defazio *et al.*, 2014 [Def14]
  - *Stochastic Variance Reduced Gradient* (SVRG) - Johnson *et al.*, 2013 [Joh13]
  - *Semi-Stochastic Gradient Descent* (S2GD) - Konecny *et al.*, 2014 [Kon13]
  - *FINITO* - Defazio *et al.*, 2014 [Def14b]
  - *Miminization by Incremental Surrogate Optimization* (MISO) - Mairal, 2013 [Mai13]
  - *Randomized Primal-Dual Gradient* [RPDG] - Lan *et al.*, 2015 [Lan15]

23.3 Variance Reduction Techniques

Suppose we want to estimate $\Theta = \mathbb{E}[X]$, the expected value of a random variable X. Suppose we also have access to a random variable Y which is highly correlated with X, and we can compute $\mathbb{E}[Y]$ easily. Let's consider the following point estimator $\hat{\Theta}_\alpha$ with $\alpha \in [0, 1]$:

$$\hat{\Theta}_\alpha = \alpha(X - Y) + \mathbb{E}[Y] \tag{23.5}$$

The expectation and variance are given by,

$$\mathbb{E}[\hat{\Theta}_\alpha] = \alpha\mathbb{E}[X] + (1 - \alpha)\mathbb{E}[Y] \tag{23.6}$$

$$\mathrm{Var}[\hat{\Theta}_\alpha] = \alpha^2 \left(\mathrm{Var}[X] + \mathrm{Var}[Y] - 2\mathrm{Cov}[X, Y]\right) \tag{23.7}$$

Note that

- $\alpha = 1$, this estimator becomes $(X - Y) + \mathbb{E}[Y]$, which is an unbiased estimator.
- $\alpha = 0$, this estimator reduces to a constant $\mathbb{E}[Y]$, which has zero variance but could be heavily biased.
- If $\mathrm{Cov}[X, Y]$ is sufficiently large, then $\mathrm{Var}[\hat{\Theta}_\alpha] < \mathrm{Var}[X]$. The new estimator $\hat{\Theta}_\alpha$ has smaller variance than the direct estimator X.
- As α increases from 0 to 1, the bias decreases and the variance increases.

Recently developed incremental gradient algorithms namely SAG, SAGA, SVRG and S2GD are all special cases of the general variance reduction technique described above. See explanation below Table 23.1.

Algorithm	Key Step	Convergence Result	Notes
SAG	$x^{t+1} = x^t - \gamma \frac{1}{n} \sum_{i=1}^n g_i^t \qquad g_i^t = \begin{cases} \nabla f_{i_t}(x^t) \\ g_i^{t-1} \end{cases}$	$\gamma = \frac{1}{16L}$, Convergence: $\mathcal{O}\left(\left[1 - \min\{\frac{\mu}{16L}, \frac{1}{8n}\}\right]^t\right)$	High Memory Cost: $\mathcal{O}(n)$
SAGA	$x^{t+1} = x^t - \gamma \left[\nabla f_{i_t}(x^t) - g_{i_t}^{t-1} + \frac{1}{n} \sum_{i=1}^n g_i^{t-1}\right]$	$\gamma = \frac{1}{3L}$, Convergence: $\mathcal{O}\left(\left[1 - \min\{\frac{\mu}{3L}, \frac{1}{4n}\}\right]^t\right)$	High Memory Cost: $\mathcal{O}(n)$
SVRG	At epoch "s", do "m" steps: $x^{t+1} = x^t - \gamma \left[\nabla f_{i_t}(x^t) - \nabla f_{i_t}(\tilde{x}) + \frac{1}{n} \sum_{i=1}^n \nabla f_i(\tilde{x})\right]$ where, $\tilde{x}$ is the last iterate of epoch $s-1$	$\gamma < \frac{1}{2L}$, Convergence: $\mathcal{O}\left(\left[\frac{1}{\mu\gamma(1-2L\gamma)m} + \frac{2L\gamma}{1-2L\gamma}\right]^s\right)$	Low Memory Cost Require two loops
S2GD	Same as above, but with random number of inner step See [Kon13] for algorithm.	$\gamma < \frac{1}{2L}$, $\nu \le \mu$, $\beta \triangleq \sum_{t=1}^m (1-\nu\gamma)^{m-t}$ Convergence: $\mathcal{O}\left(\left[\frac{(1-\nu\gamma)^m}{\mu\gamma\beta(1-2L\gamma)} + \frac{2(L-\mu)\gamma}{1-2L\gamma}\right]^s\right)$	Low Memory Cost Require two loops
FINITO	$x^{t+1} = \frac{1}{n} \sum_{i=1}^n \phi_i^t - \frac{1}{\alpha\mu n} \sum_{i=1}^n \nabla f_i(\phi_i^t)$	$n \gg \frac{L}{\mu}$, Convergence: $\mathcal{O}\left(\left[1 - \frac{1}{2n}\right]^t\right)$	Memory cost higher than either SAG or SAGA. State $\phi \to \mathcal{O}(n)$ Gradient $\nabla f_i \to \mathcal{O}(n)$
MISO	See [Mai13] for algorithm.	$\mathcal{O}\left(\left[(1-\delta) + \delta\frac{L}{\rho+u}\right]^{t-1}\right)$	Memory cost higher than either SAG or SAGA. State $\phi \to \mathcal{O}(n)$ Gradient $\nabla f_i \to \mathcal{O}(n)$

Table 23.1: Summary of Incremental Gradient Algorithms and their Convergence properties.

Connection to variance reduction technique: Let $\Theta = \nabla f(x_t)$, $X = \nabla f_{i_t}(x_t)$, the gradient used in the above algorithms can be conceived as special cases $\hat{\Theta}_\alpha$ defined in (23.5):

- SGD: $x^{t+1} = x^t - \gamma^t \nabla f_{i_t}(x^t)$ $(\alpha = 1, Y = 0)$
- SAG: $x^{t+1} = x^t - \gamma \left[\frac{1}{n}(\nabla f_{i_t}(x^t) - g_{i_t}^{t-1}) + \frac{1}{n} \sum_{i=1}^n g_i^{t-1}\right]$, $(\alpha = \frac{1}{n}, Y = g_{i_t}^{t-1})$
- SAGA: $x^{t+1} = x^t - \gamma \left[\nabla f_{i_t}(x^t) - g_{i_t}^{t-1} + \frac{1}{n} \sum_{i=1}^n g_i^{t-1}\right]$, $(\alpha = 1, Y = g_{i_t}^{t-1})$
- SVRG, S2GD: $x^{t+1} = x^t - \gamma \left[\nabla f_{i_t}(x^t) - \nabla f_{i_t}(\tilde{x}) + \frac{1}{n} \sum_{i=1}^n \nabla f_i(\tilde{x})\right]$, $(\alpha = 1, Y = \nabla f_{i_t}(\tilde{x}))$

23.4 Stochastic Variance Reduced Gradient Algorithm

Stochastic Gradient Descent (SGD) is popular among large scale optimization practitioners, however, it has a slower rate of convergence due to inherent variance [Joh13]. Consider a finite-sum optimization problem given in (23.1). The gradient descent update is given by Eq. 23.2,

$$x^{t+1} = x^t - \eta^t \cdot \nabla f(x^t) = x^t - \eta^t \cdot \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x^t).$$

However, since this method requires n gradient computations, Stochastic Gradient Descent algorithm has replaced it as a popular solution. The stochastic gradient descent update is given by Eq. 23.4,

$$x^{t+1} = x^t - \eta^t \nabla f_{i_t}(x^t). \qquad \dots (\mathbb{P}(i_t = i) = \frac{1}{n})$$

Notice that in expectation both gradient descent and stochastic gradient descent are identical, i.e. $\mathbb{E}[x^t|x^{t-1}] = x^{t-1} - \eta^t \left(\frac{1}{n} \sum_{i=1}^{n} \nabla f_i(x^{t-1})\right)$ for both GD as well as SGD. However, SGD has a disadvantage in the sense that the gradient used in every iteration equals $\nabla f(x)$ only in expectation and may have large deviation from the mean in some of the instances. Due to inherent large variance, we observe only a sublinear $\mathcal{O}(1/t)$ rate of convergence. Incremental algorithms have become popular in the past decade due to their linear rate of convergence (improved from sublinear rate, see Section 23.2 and Table 23.1 for list of methods and their convergence rates).

We will study one of the popular incremental algorithms called Stochastic Variance Reduced Gradient (SVRG). The strengths of this work are [Joh13],

- SVRG algorithm does not require storage of gradients as seen in SAG or SAGA.
- Convergence rates for SVRG can be proved easily and a very intuitive explanation can be provided by linking increased speed to reduced variance.

Algorithm 1 Stochastic Variance Reduced Gradient

Parameters update frequency m and learning rate η
Initialize $\tilde{x}_0$
for $s = 1, 2, \dots$ **do**
  $\tilde{x} = \tilde{x}^{s-1}$
  $\tilde{\theta} = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{x})$
  $x_0 = \tilde{x}$
  for $t = 1, 2, \dots, m$ **do**
    Randomly pick $i_t \in \{1, 2, \dots, n\}$ and update weight,
      $x^t = x^{t-1} - \eta \left(\nabla f_{i_t}(x^{t-1}) - \nabla f_{i_t}(\tilde{x}) + \tilde{\theta}\right)$
  end for
  Update $\tilde{x}^s$
    Option I $\tilde{x}^s = x^m$
    Option II $\tilde{x}^s = \frac{1}{m} \sum_{t=1}^{m} x^t$
    Option III $\tilde{x}^s = x^t$ for randomly chosen $t \in \{1, 2, \dots, m\}$
end for

Convergence Analysis

For simplicity we consider Problem 23.1, where $f_i(x)$ is convex and L-smooth for all $i = 1, \ldots, n$, and $f(x)$ is μ-strongly convex. For instance, if each of the function f_i is L_i-smooth, then once can set $L = \max_{i=1,\ldots,n} L_i$. We show linear convergence of SVRG for such a finite-sum optimization problem.

Theorem 23.1 *Assume $f_i(x)$ is is convex and L-smooth and $f(x)$ is μ-strongly convex. Let $x_* = \arg\min f(x)$. Assume m is sufficiently large (and $\eta < \frac{1}{2L}$), so that,*

$$\rho = \frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta} < 1,$$

then we have geometric convergence in expectation for SVRG (under Option II and Option III), i.e.,

$$\mathbb{E}[f(\tilde{x}^s) - f_*] \leq \rho^s \, [f(\tilde{x}^0) - f_*].$$

Remark. Setting $\eta = \theta/L$ with some constant $\theta > 0$, this gives

$$\rho = \frac{L}{\mu\theta(1-2\theta)m} + \frac{2\theta}{1-2\theta} = O\left(\frac{L}{\mu m} + \text{ const.}\right)$$

Hence, if we set $m = O(L/\mu)$, then this will result in a constant rate ρ. The number of epochs needed to achieve an ϵ optimal solution is $O(\log(\frac{1}{\epsilon}))$. Therefore, the overall complexity for SVRG is

$$\mathcal{O}\left((m+n)\log\left(\frac{1}{\epsilon}\right)\right) = \mathcal{O}\left((n+\frac{L}{\mu})\log\left(\frac{1}{\epsilon}\right)\right)$$

Note that the complexity is significantly better that of Gradient Descent, i.e.

$$\mathcal{O}\left(n \cdot \frac{L}{\mu}\log\left(\frac{1}{\epsilon}\right)\right)$$

when the condition number L/μ is large.

Extensions.

1. **Non-uniform sampling:** SVRG algorithm assumes uniform sampling, however, one may choose an adaptive sampling rate,
$$\mathbb{P}(i_t = i) = \frac{L_i}{\sum L_i}$$
where L_i is the smoothness parameter for f_i. This sampling strategy improves the complexity from $\mathcal{O}\left(\left(n + \frac{L_{max}}{\mu}\right)\log(\frac{1}{\epsilon})\right)$ to $\mathcal{O}\left(\left(n + \frac{L_{avg}}{\mu}\right)\log(\frac{1}{\epsilon})\right)$. Intuitively, the function $f_i(x)$ that has a higher Lipschitz constant (which is prone to change relatively rapidly) gets higher probability of getting selected.

2. **Composite convex minimization:** These are problems of the form
$$\min_x \; \frac{1}{n}\sum_i f_i(x) + g(x)$$
where $f_i(x)$ are smooth and convex, but $g(x)$ is convex but possibly nonsmooth. Such problems can be handle by **prox-SVRG** [Xia14] by imposing an additional proximal operator of g at iteration.

3. **Acceleration:** We can accelerate SVRG further to arrive at an optimal complexity of

$$\mathcal{O}\left(\left(n+\sqrt{\frac{nL}{\mu}}\right)\log(\frac{1}{\epsilon})\right).$$

This improvement is significant in problems where $\frac{L}{\mu} \gg n$.

We now prove the main theorem. We start by proving provide the following lemma.

Lemma 23.2 *For any x, we have*

$$\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x)-\nabla f_i(x_*)\|_2^2 \le 2L(f(x)-f(x_*)). \tag{23.8}$$

Proof: For any i, consider a function $g_i(x)$,

$$g_i(x) = f_i(x) - f_i(x_*) - \nabla f_i(x_*)^T(x-x_*).$$

Note that $\nabla g_i(x) = \nabla f_i(x) - \nabla f_i(x_*)$. Clearly, $\nabla g_i(x_*) = 0$, implying that $g_i(x_*) = \min_x g_i(x)$. Therefore, following the definition of minimum and the L-smoothness of function $g_i(x)$, we arrive at

$$0 = g_i(x_*) \le \min_{\eta}\left[g_i(x-\eta\nabla g_i(x))\right] \le g_i(x) - \frac{1}{2L}\|\nabla g_i(x)\|_2^2$$

That is,

$$\begin{aligned}\|\nabla g_i(x)\|_2^2 &\le 2Lg_i(x)\\ \|\nabla f_i(x)-\nabla f_i(x_*)\|_2^2 &\le 2L\left(f_i(x)-f_i(x_*)-\nabla f_i(x_*)^T(x-x_*)\right).\end{aligned}$$

By summing the above inequality for all $i = 1, 2, \ldots, n$ and using the fact $\nabla f(x_*) = 0$ and $f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x)$, we have

$$\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x)-\nabla f_i(x_*)\|_2^2 \le 2L(f(x)-f(x_*)).$$

■

We now proceed to prove the theorem.

Proof of Theorem. Let $v^t = \nabla f_{i_t}(x^{t-1}) - \nabla f_{i_t}(\tilde{x}) + \nabla f(\tilde{x})$. We now take expectation with respect to i_t conditioned on w^{t-1} and obtain,

$$\begin{aligned}
&\mathbb{E}[\|v^t\|^2] = \mathbb{E}\left[\|\left[\nabla f_{i_t}(x^{t-1}) - \nabla f_{i_t}(x_*)\right] + \left[\nabla f_{i_t}(x_*) - \nabla f_{i_t}(\tilde{x}) + \nabla f(\tilde{x})\right]\|_2^2\right] && \\
&\le 2\mathbb{E}\left[\nabla\|f_{i_t}(x^{t-1}) - \nabla f_{i_t}(x_*)\|_2^2\right] + 2\mathbb{E}\left[\|\nabla f_{i_t}(\tilde{x}) - \nabla f_{i_t}(x_*) - \nabla f(\tilde{x})\|_2^2\right] && (\|a+b\|^2 \le 2\|a\|^2 + 2\|b\|^2)\\
&\le 2\mathbb{E}\left[\nabla\|f_{i_t}(x^{t-1}) - \nabla f_{i_t}(x_*)\|_2^2\right] + 2\mathbb{E}\left[\|\nabla f_{i_t}(\tilde{x}) - \nabla f_{i_t}(x_*)\|_2^2\right] && (\nabla f(\tilde{x}) = \mathbb{E}\left[\nabla f_{i_t}(x) - \nabla f_{i_t}(x_*)\right])\\
& && (\mathbb{E}[\|e-\mathbb{E}[e]\|_2^2] \le \mathbb{E}[\|e\|_2^2])\\
&\le 4L\left[f(x^{t-1}) - f(x_*) + f(\tilde{x}) - f(x_*)\right] && (\text{From Eq. 23.8})
\end{aligned}$$

Now notice from the definition of v^t, $\mathbb{E}\left[v^t|x^{t-1}\right] = \nabla f(x^{t-1})$; and this leads to,

$$\begin{aligned}
&\mathbb{E}\left[\|x^t - x_*\|_2^2\right] = \mathbb{E}\left[\|x^{t-1} - \eta v^t - x_*\|_2^2\right] \\
&= \|x^{t-1} - x_*\|_2^2 - 2\eta(x^{t-1} - x_*)^T\mathbb{E}\left[v^t\right] + \eta^2\mathbb{E}\left[\|v^t\|_2^2\right] \\
&\leq \|x^{t-1} - x_*\|_2^2 - 2\eta(x^{t-1} - x_*)^T\nabla f(x^{t-1}) + 4L\eta^2\left[f(x^{t-1}) - f(x_*) + f(\tilde{x}) - f(x_*)\right] \\
&\leq \|x^{t-1} - x_*\|_2^2 - 2\eta(f(x^{t-1}) - f(x_*)) + 4L\eta^2\left[f(x^{t-1}) - f(x_*) + f(\tilde{x}) - f(x_*)\right] \\
&\leq \|x^{t-1} - x_*\|_2^2 - 2\eta(1-2L\eta)(f(x^{t-1}) - f(x_*)) + 4L\eta^2\left[f(\tilde{x}) - f(x_*)\right]
\end{aligned}$$

We consider a fixed stage s, so that $\tilde{w} = \tilde{w}^{s-1}$ and $\tilde{w}^s$ is selected after all the updates have completed. By summing the previous inequality over $t = 1, 2, \ldots, m$, taking expectation with all the history, and using option II (or III) at stage s, we obtain [1],

$$\begin{aligned}
&\mathbb{E}\left[\|x^m - x_*\|^2\right] + 2\eta(1-2L\eta)m\mathbb{E}\left[f(\tilde{x}^s) - f(x_*)\right] \\
&\qquad \leq \mathbb{E}\left[\|x^0 - x_*\|^2\right] + 4Lm\eta^2\mathbb{E}\left[f(\tilde{x}^{s-1}) - f(x_*)\right] \\
&\qquad \leq \mathbb{E}\left[\|\tilde{x} - x_*\|^2\right] + 4Lm\eta^2\mathbb{E}\left[f(\tilde{x}^{s-1}) - f(x_*)\right] \\
&\qquad \leq \frac{2}{\mu}\mathbb{E}\left[f(\tilde{x}) - f(x_*)\right] + 4Lm\eta^2\mathbb{E}\left[f(\tilde{x}) - f(x_*)\right] \qquad (f(x) \text{ is } \mu \text{ strongly convex.})
\end{aligned}$$

Clearly, from the above inequality we get,

$$\mathbb{E}\left[f(\tilde{x}^s) - f(x_*)\right] \leq \left[\frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta}\right]\mathbb{E}\left[f(\tilde{x}^{s-1}) - f(x_*)\right].$$

This gives us the desired geometric convergence rate, $\mathbb{E}[f(\tilde{x}^s) - f_*] \leq \rho^s\,[f(\tilde{x}^0) - f_*]$. ■

References

[Ned09] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, 54.1 (2009): 48-61.

[Tsi84] J. Tsitsiklis, "Problems in Decentralized Decision making and Computation," Ph.D. Thesis, No. LIDS-TH-1424. MIT Cambridge, LIDS, 1984.

[Bach13] F. Bach, "stochastic gradient methods for machine learning," *Slide*, (2013).

[Ber97] D. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM Journal on Optimization* 7.4 (1997): 913-926.

[Bla07] D. Blatt and A. Hero and H. Gauchman "A convergent incremental gradient method with a constant step size," *SIAM Journal on Optimization* 18.1 (2007): 29-51.

[Sch13] M. Schmidt and N. Le Roux and F. Bach "Minimizing finite sums with the stochastic average gradient," *arXiv preprint*, arXiv:1309.2388 (2013).

[1] Note that here we use our choice of update equation (Option II or Option III). If we use Option II, we use the convexity of $f(x)$ to establish the following bound. If we use Option III, we need to use the fact that, since $\tilde{x}$ is chosen randomly from x^t, $\mathbb{E}\left[f(x^{t-1})\right] = \mathbb{E}\left[f(\tilde{x})\right]$.

$$-\sum_{t=1}^{m} 2\eta(1-2L\eta)\mathbb{E}\left[f(x^{t-1}) - f(x_*)\right] \leq -\sum_{t=1}^{m} 2\eta(1-2L\eta)m\mathbb{E}\left[f(\tilde{x}) - f(x_*)\right]$$

[Def14] A. Defazio and F. Bach and S. Lacoste-Julien "Saga: A fast incremental gradient method with support for non-strongly convex composite objectives," *Advances in Neural Information Processing Systems* , 2014.

[Joh13] R. Johnson and T. Zhang "Accelerating stochastic gradient descent using predictive variance reduction," *Advances in Neural Information Processing Systems*, 2013.

[Kon13] J. Konecny and P. Richtarik "Semi-stochastic gradient descent methods," *arXiv preprint*, arXiv:1312.1666 (2013)

[Def14b] A. Defazio and J. Domke and T. Caetano "Finito: A faster, permutable incremental gradient method for big data problems," *ICML*, 2014.

[Mai13] J. Mairal "Optimization with First-Order Surrogate Functions," *ICML*, 2013.

[Lan15] G. Lan and Y. Zhou "An optimal randomized incremental gradient method," *arXiv preprint* arXiv:1507.02000 (2015).

[Xia14] L. Xiao and T. Zhang "A proximal stochastic gradient method with progressive variance reduction," *arXiv preprint SIAM Journal on Optimization* 24(4), 2057-2075, 2014.