Stochastic Gradient Methods for Large-Scale Machine Learning

Leon Bottou

Facebook AI Research

Frank E. Curtis

Lehigh University

Jorge Nocedal

Northwestern University

Our Goal

1. This is a tutorial about the stochastic gradient (SG) method
2. Why has it risen to such prominence?
3. What is the main mechanism that drives it?
4. What can we say about its behavior in convex and non-convex cases?
5. What ideas have been proposed to improve upon SG?

Organization

I. Motivation for the stochastic gradient (SG) method:
Jorge Nocedal

II. Analysis of SG: Leon Bottou

III. Beyond SG: noise reduction and 2nd -order methods:
Frank E. Curtis

Reference

This tutorial is a summary of the paper

"Optimization Methods for Large-Scale Machine Learning"

L. Bottou, F.E. Curtis, J. Nocedal

http://arxiv.org/abs/1606.04838

Prepared for SIAM Review

Problem statement

Given training set $\{(x_1, y_1), \ldots (x_n, y_n)\}$

Given a loss function $\ell(h, y)$ (hinge loss, logistic,...)

Find a prediction function $h(x; w)$ (linear, DNN,...)

$$\min_w \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i; w), y_i)$$

Notation: random variable $\xi = (x_i, y_i)$

$$R_n(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w)$$

empirical risk

The real objective

$$R(w) = \mathbb{E}[f(w; \xi)]$$

expected risk

Stochastic Gradient Method

First present algorithms for empirical risk minimization

$$R_n(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$

$$w_{k+1} = w_k - \alpha_k \nabla f_i(w_k) \qquad i \in \{1,\ldots,n\} \text{ choose at random}$$

- Very cheap iteration; gradient w.r.t. just 1 data point
- Stochastic process dependent on the choice of i
- Not a gradient descent method
- Robbins-Monro 1951
- Descent in expectation

Batch Optimization Methods

$$\boxed{w_{k+1} = w_k - \alpha_k \nabla R_n(w_k)}$$

batch gradient method

$$w_{k+1} = w_k - \frac{\alpha_k}{n} \sum_{i=1}^{n} \nabla f_i(w_k)$$

- More expensive step
- Can choose among a wide range of optimization algorithms
- Opportunities for parallelism

Why has SG emerged at the preeminent method?

Understanding: study computational trade-offs between stochastic and batch methods, and their ability to minimize R

Intuition

SG employs information more efficiently than batch method

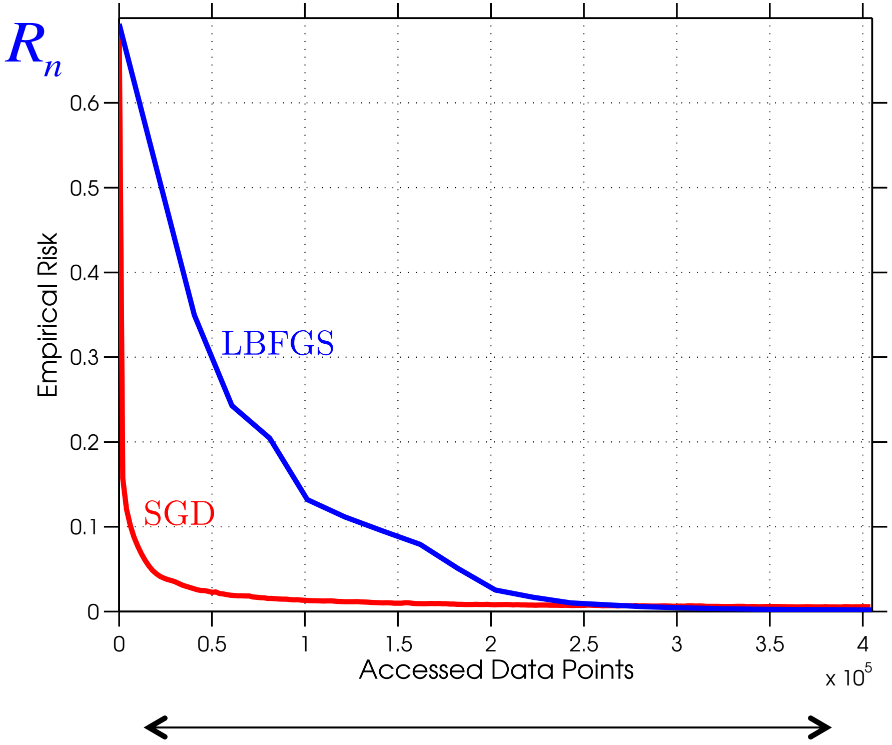
Argument 1:

Suppose data is 10 copies of a set S
Iteration of batch method 10 times more expensive
SG performs same computations

Argument 2:

Training set (40%), test set (30%), validation set (30%).
Why not 20%, 10%, 1%..?

Practical Experience



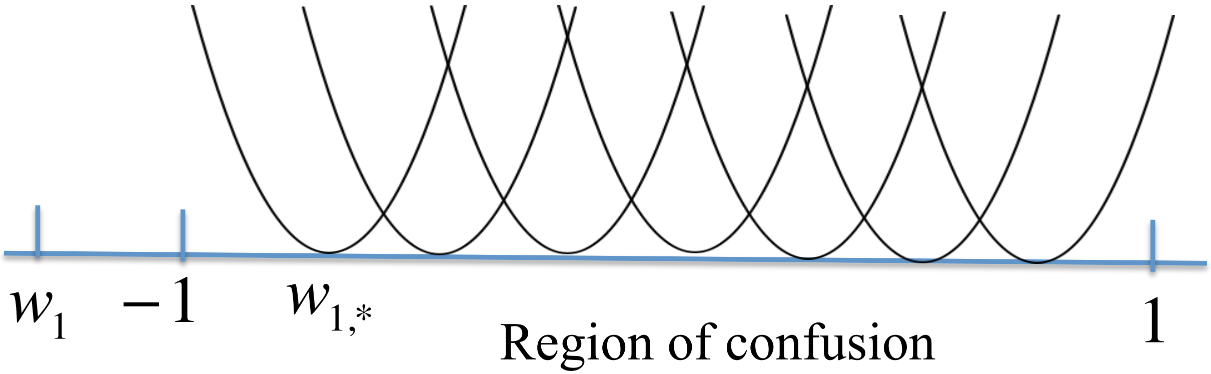
10 epochs

Fast initial progress of SG followed by drastic slowdown

Can we explain this?

Example by Bertsekas

$$R_n(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w)$$



Note that this is a geographical argument

Analysis: given w_k what is the expected decrease in the objective function R_n as we choose one of the quadratics randomly?

A fundamental observation

$$\mathbb{E}[R_n(w_{k+1}) - R_n(w_k)] \leq -\alpha_k \|\nabla R_n(w_k)\|_2^2 + \alpha_k^2 \mathbb{E}\|\nabla f(w_k, \xi_k)\|^2$$

Initially, gradient decrease dominates; then variance in gradient hinders progress (area of confusion)

To ensure convergence $\alpha_k \rightarrow 0$ in SG method to control variance.
What can we say when $\alpha_k = \alpha$ is constant?

Noise reduction methods in Part 3 directly control the noise given in the last term

Theoretical Motivation - strongly convex case

⊙ Batch gradient: linear convergence

$$R_n(w_k) - R_n(w^*) \leq O(\rho^k) \qquad \rho < 1$$

Per iteration cost proportional to n

⊙ SG has sublinear rate of convergence

$$\mathbb{E}[R_n(w_k) - R_n(w^*)] = O(1/k)$$

Per iteration cost and convergence constant independent of n

Same convergence rate for generalization error

$$\mathbb{E}[R(w_k) - R(w^*)] = O(1/k)$$

Computational complexity

Total work to obtain $R_n(w_k) \leq R_n(w^*) + \epsilon$

Batch gradient method: $n \log(1/\epsilon)$

Stochastic gradient method: $1/\epsilon$

Think of $\epsilon = 10^{-3}$

Which one is better?

A discussion of these tradeoffs *is next!*

Disclaimer: although much is understood about the SG method There are still some great mysteries, e.g.: why is it so much better than batch methods on DNNs?

End of Part I

Optimization Methods for Machine Learning
Part II – The theory of SG

Leon Bottou
Facebook AI Research

Frank E. Curtis
Lehigh University

Jorge Nocedal
Northwestern University

Summary

1. Setup
2. Fundamental Lemmas
3. SG for Strongly Convex Objectives
4. SG for General Objectives
5. Work complexity for Large-Scale Learning
6. Comments

1- Setup

The generic SG algorithm

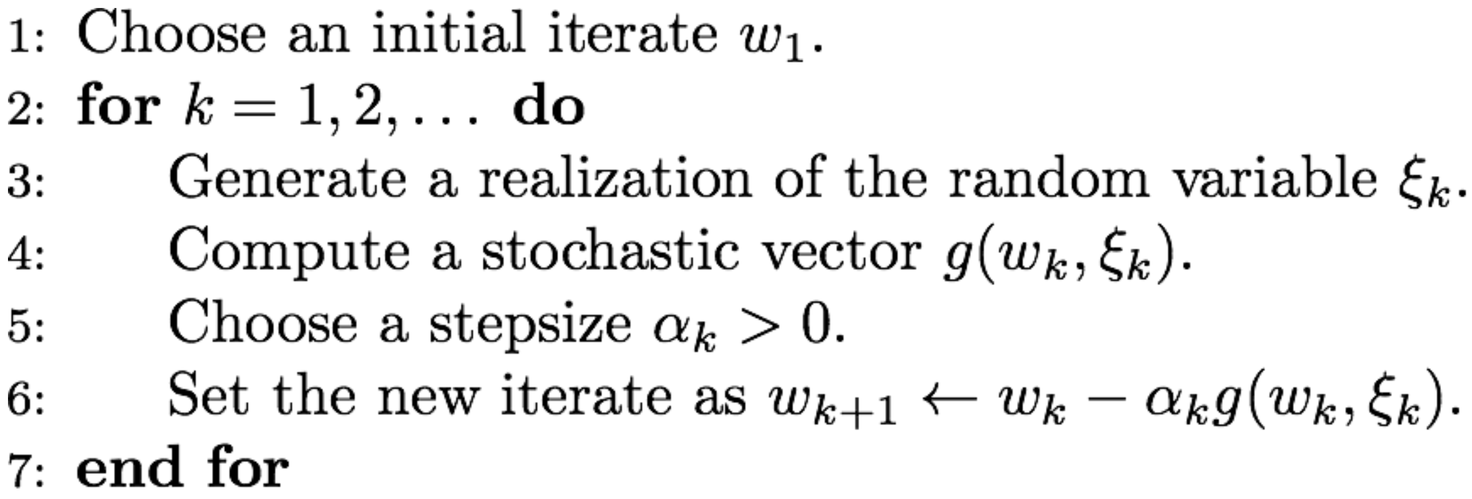
The SG algorithm produces successive iterates $w_k \in \mathbb{R}^d$ with the goal to minimize a certain function $F : \mathbb{R}^d \to \mathbb{R}$.

We assume that we have access to three mechanisms

1. Given an iteration number k ,
   a mechanism to generate a realization of a random variable ξ_k.
   The $\{\xi_k\}$ form a sequence of jointly independent random variables
2. Given an iterate w_k and a realization ξ_k,
   a mechanism to compute a stochastic vector $g(w_k, \xi_k) \in \mathbb{R}^d$
3. Given an iteration number,
   a mechanism to compute a scalar stepsize $\alpha_k > 0$

The generic SG algorithm

Algorithm 4.1 (Stochastic Gradient (SG) Method)



```
Choose an initial iterate $w_1$.
for $k = 1, 2, \ldots$ do
    Generate a realization of the random variable $\xi_k$.
    Compute a stochastic vector $g(w_k, \xi_k)$.
    Choose a stepsize $\alpha_k > 0$.
    Set the new iterate as $w_{k+1} \leftarrow w_k - \alpha_k g(w_k, \xi_k)$.
end for
```

The generic SG algorithm

The function $F : \mathbb{R}^d \to \mathbb{R}$ could be

$$F(w) = \begin{cases} R(w) = \mathbb{E}[f(w;\xi)] & \text{the expected risk,} \\ R_n(w) = \frac{1}{n}\sum_{\xi=1}^{n} f(w;\xi) & \text{the empirical risk.} \end{cases}$$

The stochastic vector could be

$$g(w_k, \xi_k) = \begin{cases} \nabla f(w_k;\xi_k) & \text{the gradient for one example,} \\ \frac{1}{n_k}\sum_{i=1}^{n_k} \nabla f(w_k;\xi_{k,i}) & \text{the gradient for a minibatch,} \\ H_k \frac{1}{n_k}\sum_{i=1}^{n_k} \nabla f(w_k;\xi_{k,i}), & \text{possibly rescaled} \end{cases}$$

The generic SG algorithm

Stochastic processes

- We assume that the $\{\xi_k\}$ are jointly independent to avoid the full machinery of stochastic processes. But everything still holds if the $\{\xi_k\}$ form an adapted stochastic process, where each ξ_k can depend on the previous ones.

Active learning

- We can handle more complex setups by view ξ_k as a "random seed". For instance, in active learning, $g(w_k, \xi_k)$ firsts construct a multinomial distribution on the training examples in a manner that depends on w_k, then uses the random seed ξ_k to pick one according to that distribution.

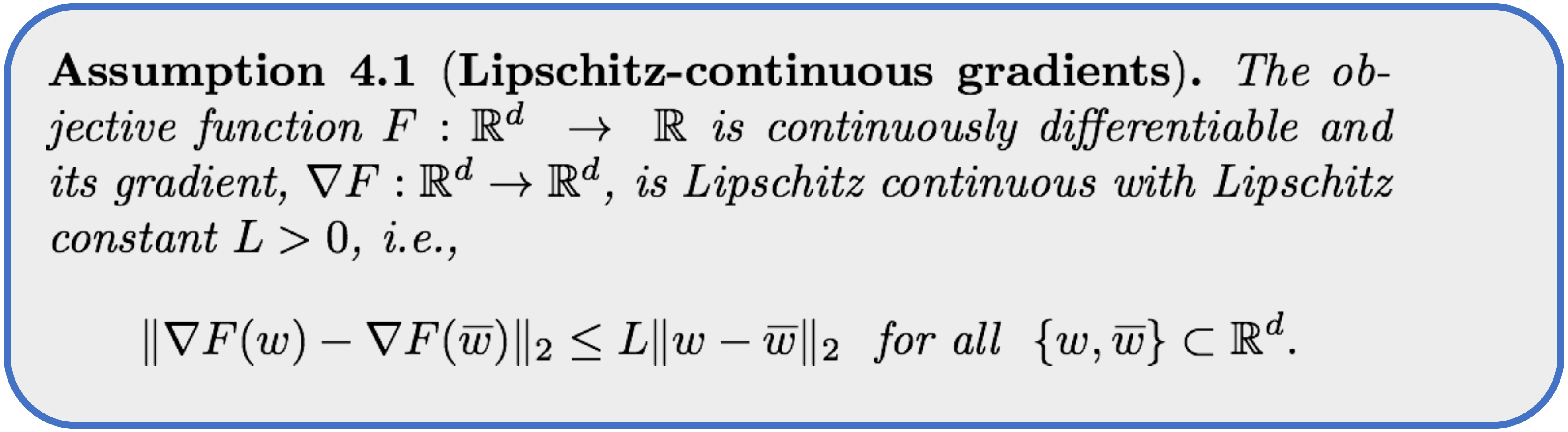
The same mathematics cover all these cases.

2- Fundamental lemmas

Smoothness

Smoothness

- Our analysis relies on a smoothness assumption.
  We chose this path because it also gives results for the nonconvex case.
  We'll discuss other paths in the commentary section.



Assumption 4.1 (Lipschitz-continuous gradients). *The objective function $F : \mathbb{R}^d \to \mathbb{R}$ is continuously differentiable and its gradient, $\nabla F : \mathbb{R}^d \to \mathbb{R}^d$, is Lipschitz continuous with Lipschitz constant $L > 0$, i.e.,*

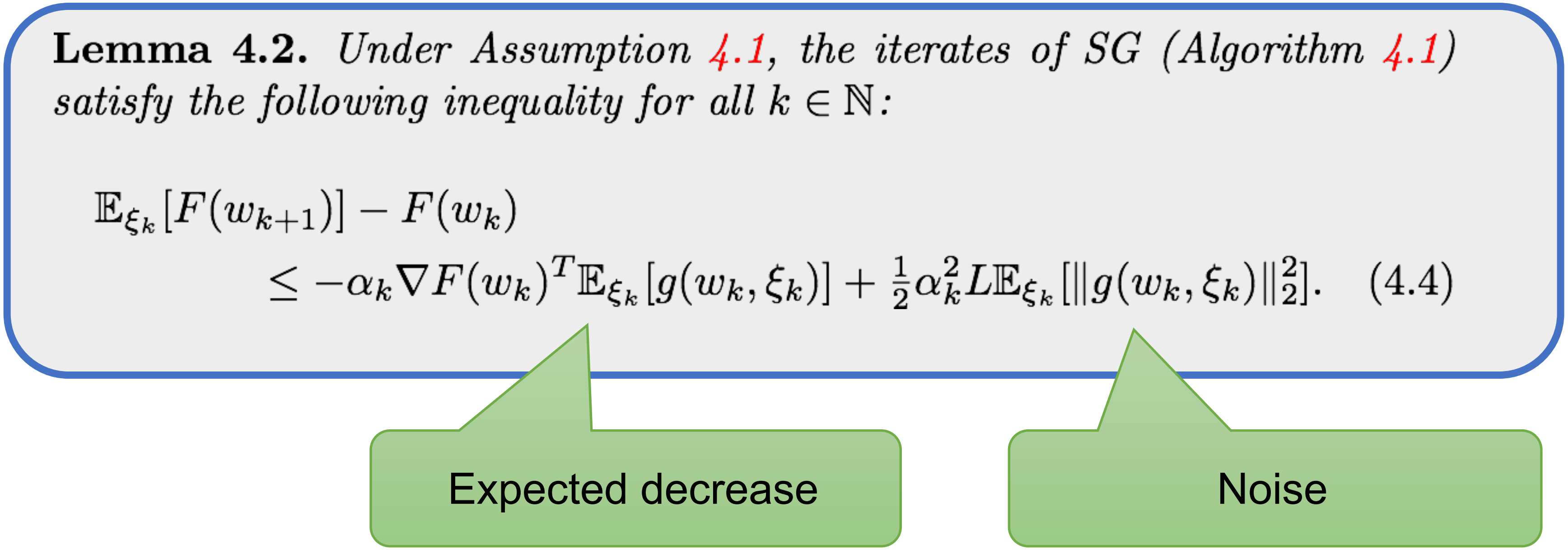
$$\|\nabla F(w) - \nabla F(\overline{w})\|_2 \leq L\|w - \overline{w}\|_2 \quad \textit{for all} \quad \{w, \overline{w}\} \subset \mathbb{R}^d.$$

Well known consequence

$$F(w) \leq F(\overline{w}) + \nabla F(\overline{w})^T(w - \overline{w}) + \tfrac{1}{2}L\|w - \overline{w}\|_2^2 \quad \text{for all} \quad \{w, \overline{w}\} \subset \mathbb{R}^d. \qquad (4.3)$$

Smoothness

- $\mathbb{E}_{\xi_k}[\quad]$ is the expectation with respect to the distribution of ξ_k only.
- $\mathbb{E}_{\xi_k}[F(w_{k+1})]$ is meaningful because w_{k+1} depends on ξ_k (step 6 of SG)



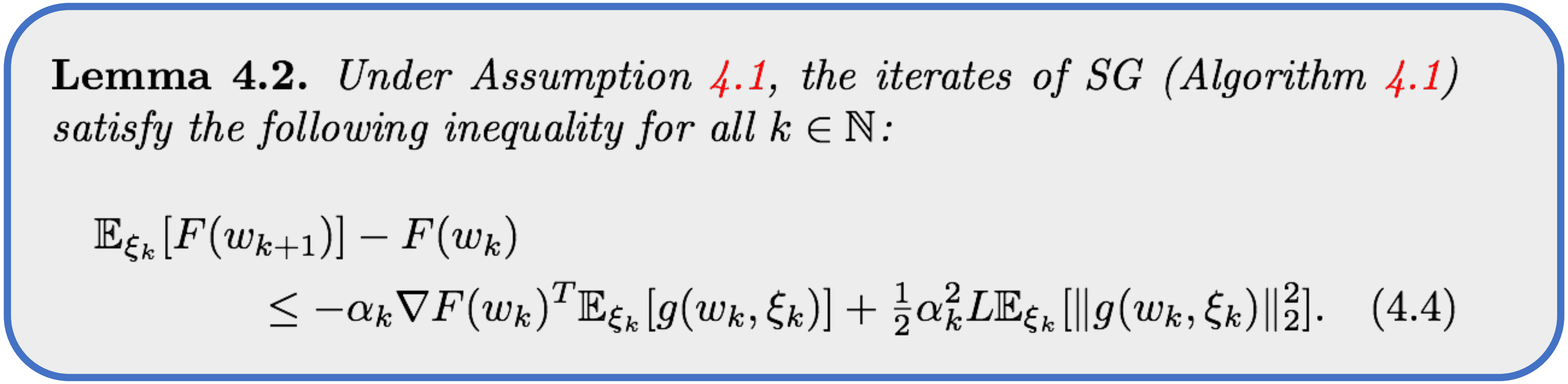
Lemma 4.2. *Under Assumption 4.1, the iterates of SG (Algorithm 4.1) satisfy the following inequality for all $k \in \mathbb{N}$:*

$$\begin{aligned}\mathbb{E}_{\xi_k}[F(w_{k+1})] &- F(w_k) \\ &\leq -\alpha_k \nabla F(w_k)^T \mathbb{E}_{\xi_k}[g(w_k,\xi_k)] + \tfrac{1}{2}\alpha_k^2 L \mathbb{E}_{\xi_k}[\|g(w_k,\xi_k)\|_2^2]. \quad (4.4)\end{aligned}$$

Expected decrease

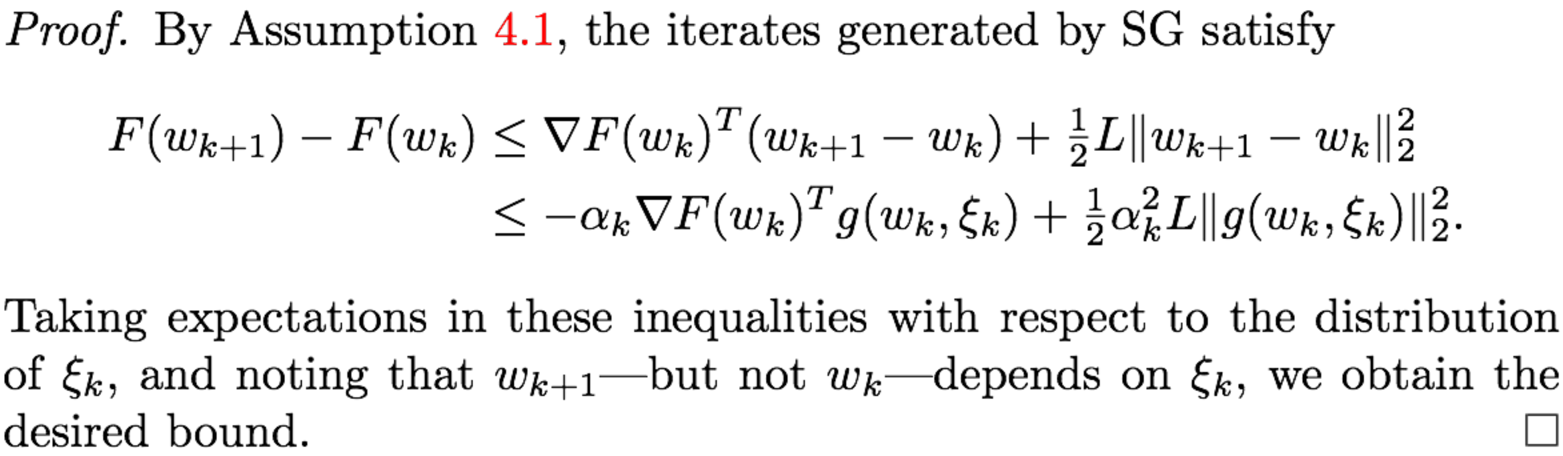
Noise

Smoothness



Lemma 4.2. *Under Assumption 4.1, the iterates of SG (Algorithm 4.1) satisfy the following inequality for all $k \in \mathbb{N}$:*

$$\begin{aligned} &\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) \\ &\qquad \leq -\alpha_k \nabla F(w_k)^T \mathbb{E}_{\xi_k}[g(w_k,\xi_k)] + \tfrac{1}{2}\alpha_k^2 L \mathbb{E}_{\xi_k}[\|g(w_k,\xi_k)\|_2^2]. \qquad (4.4) \end{aligned}$$

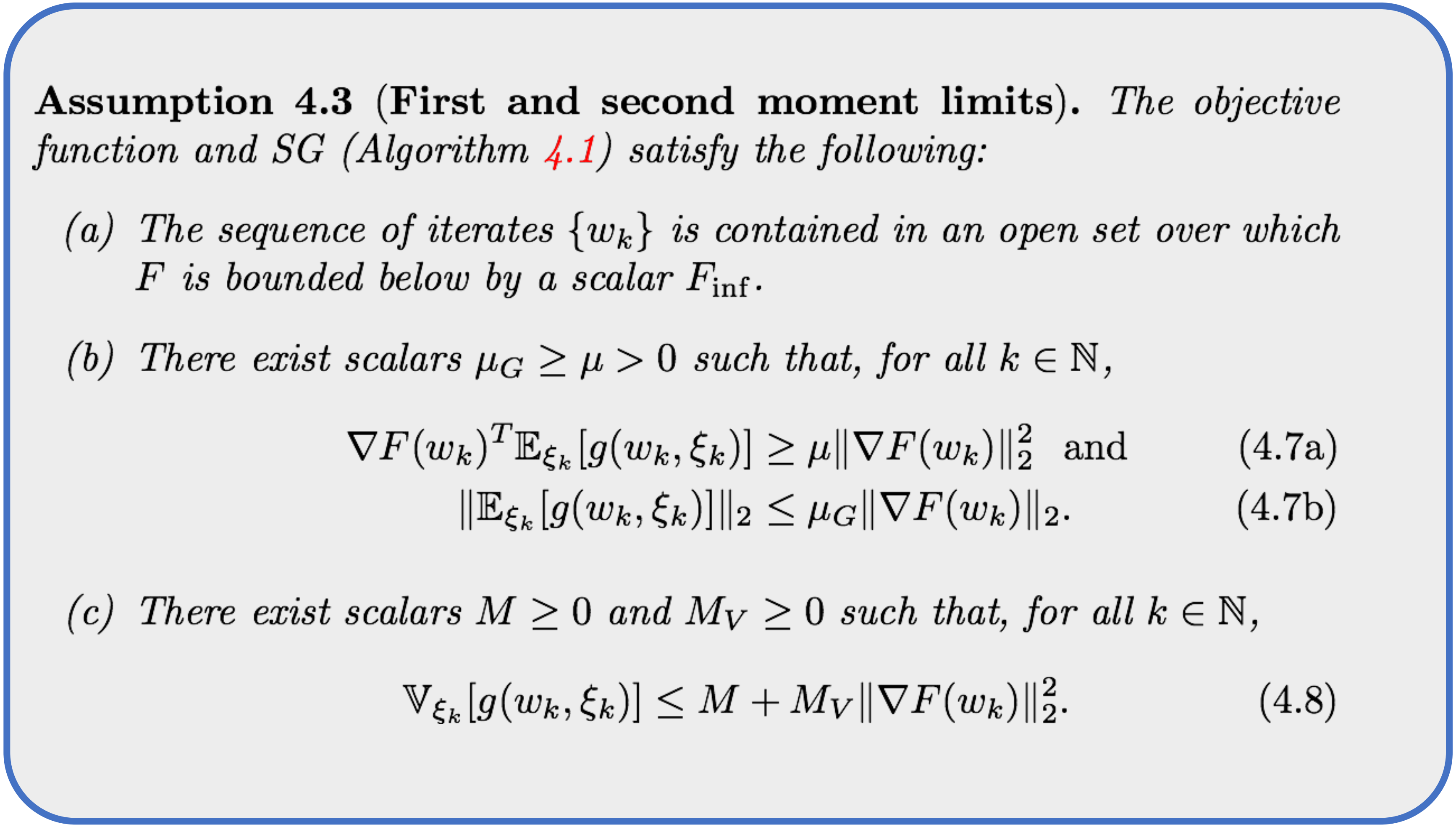


Proof. By Assumption 4.1, the iterates generated by SG satisfy

$$\begin{aligned} F(w_{k+1}) - F(w_k) &\leq \nabla F(w_k)^T (w_{k+1} - w_k) + \tfrac{1}{2} L \|w_{k+1} - w_k\|_2^2 \\ &\leq -\alpha_k \nabla F(w_k)^T g(w_k,\xi_k) + \tfrac{1}{2}\alpha_k^2 L \|g(w_k,\xi_k)\|_2^2. \end{aligned}$$

Taking expectations in these inequalities with respect to the distribution of ξ_k, and noting that w_{k+1}—but not w_k—depends on ξ_k, we obtain the desired bound. □

Moments



Assumption 4.3 (First and second moment limits). *The objective function and SG (Algorithm 4.1) satisfy the following:*

(a) The sequence of iterates $\{w_k\}$ is contained in an open set over which F is bounded below by a scalar F_{inf}.

(b) There exist scalars $\mu_G \geq \mu > 0$ such that, for all $k \in \mathbb{N}$,

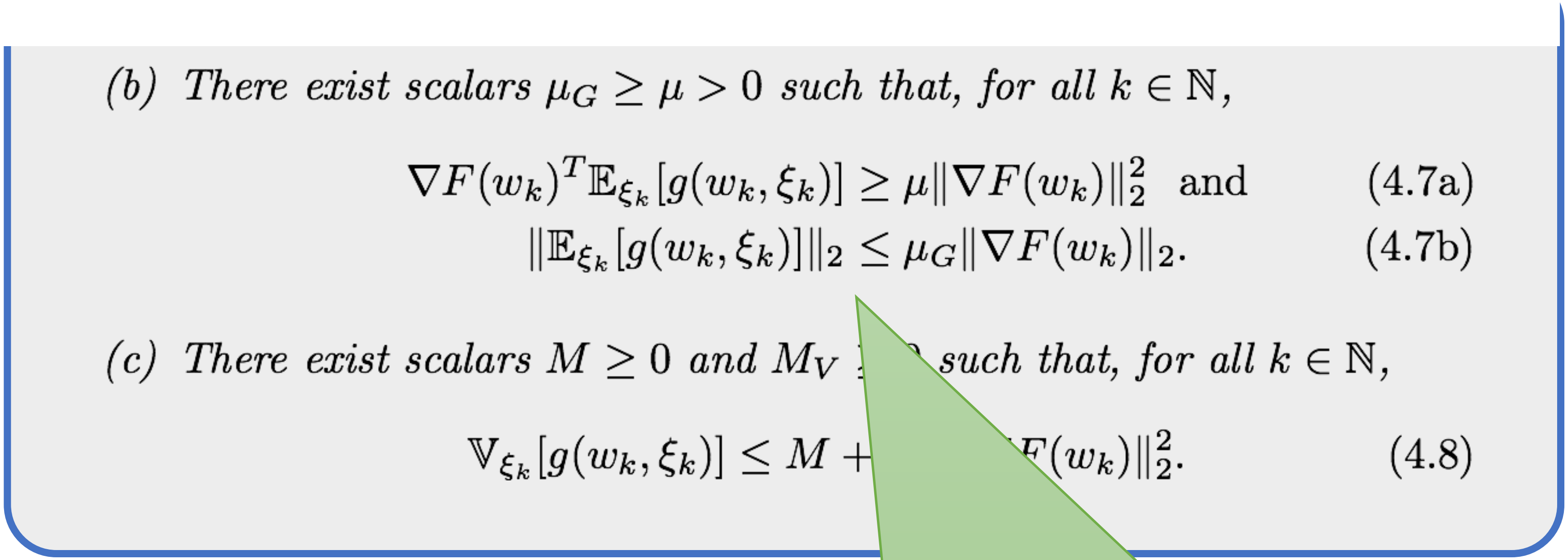
$$\nabla F(w_k)^T \mathbb{E}_{\xi_k}[g(w_k, \xi_k)] \geq \mu \|\nabla F(w_k)\|_2^2 \quad \text{and} \tag{4.7a}$$

$$\|\mathbb{E}_{\xi_k}[g(w_k, \xi_k)]\|_2 \leq \mu_G \|\nabla F(w_k)\|_2. \tag{4.7b}$$

(c) There exist scalars $M \geq 0$ and $M_V \geq 0$ such that, for all $k \in \mathbb{N}$,

$$\mathbb{V}_{\xi_k}[g(w_k, \xi_k)] \leq M + M_V \|\nabla F(w_k)\|_2^2. \tag{4.8}$$

Moments



(b) There exist scalars $\mu_G \geq \mu > 0$ *such that, for all* $k \in \mathbb{N}$,

$$\nabla F(w_k)^T \mathbb{E}_{\xi_k}[g(w_k,\xi_k)] \geq \mu \|\nabla F(w_k)\|_2^2 \quad \text{and} \tag{4.7a}$$

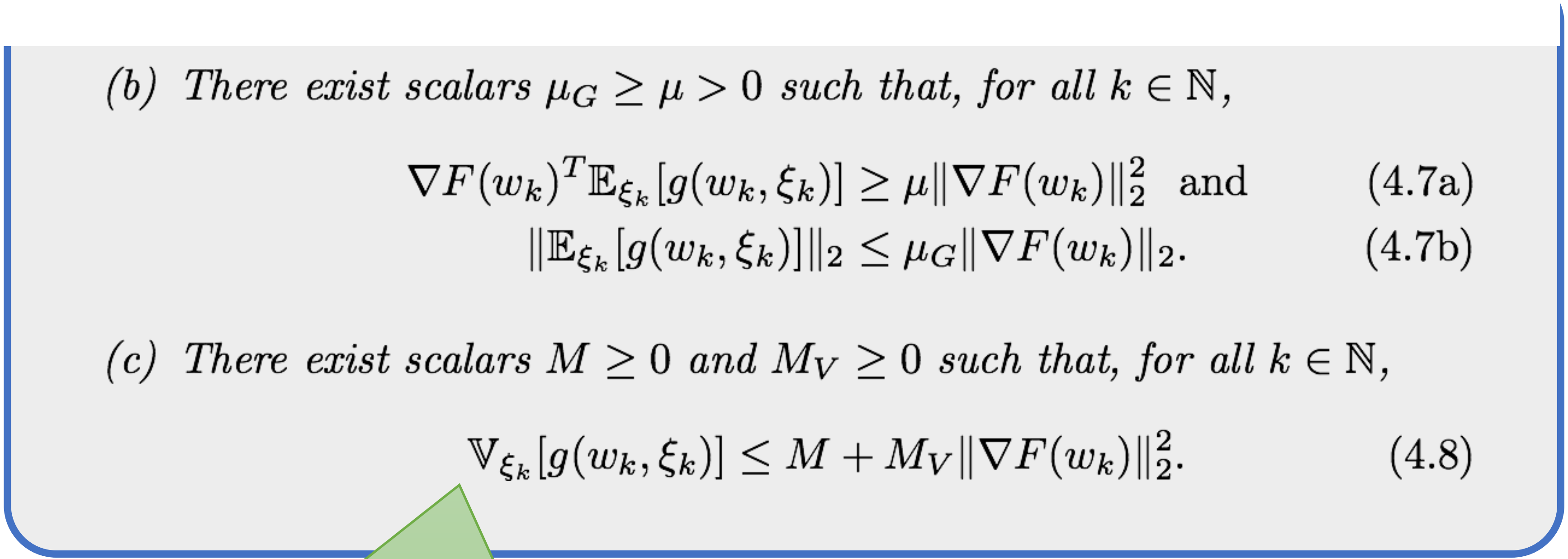
$$\|\mathbb{E}_{\xi_k}[g(w_k,\xi_k)]\|_2 \leq \mu_G \|\nabla F(w_k)\|_2. \tag{4.7b}$$

(c) There exist scalars $M \geq 0$ *and* $M_V \geq 0$ *such that, for all* $k \in \mathbb{N}$,

$$\mathbb{V}_{\xi_k}[g(w_k,\xi_k)] \leq M + \quad F(w_k)\|_2^2. \tag{4.8}$$

- In expectation $g(w_k,\xi_k)$ is a sufficient descent direction.
- True if $\mathbb{E}_{\xi_k}[g(w_k,\xi_k)] = \nabla F(w_k)$ with $\mu = \mu_G = 1$.
- True if $\mathbb{E}_{\xi_k}[g(w_k,\xi_k)] = H_k \nabla F(w_k)$ with bounded spectrum.

Moments



(b) There exist scalars $\mu_G \geq \mu > 0$ *such that, for all* $k \in \mathbb{N}$,

$$\nabla F(w_k)^T \mathbb{E}_{\xi_k}[g(w_k, \xi_k)] \geq \mu \|\nabla F(w_k)\|_2^2 \quad \text{and} \tag{4.7a}$$

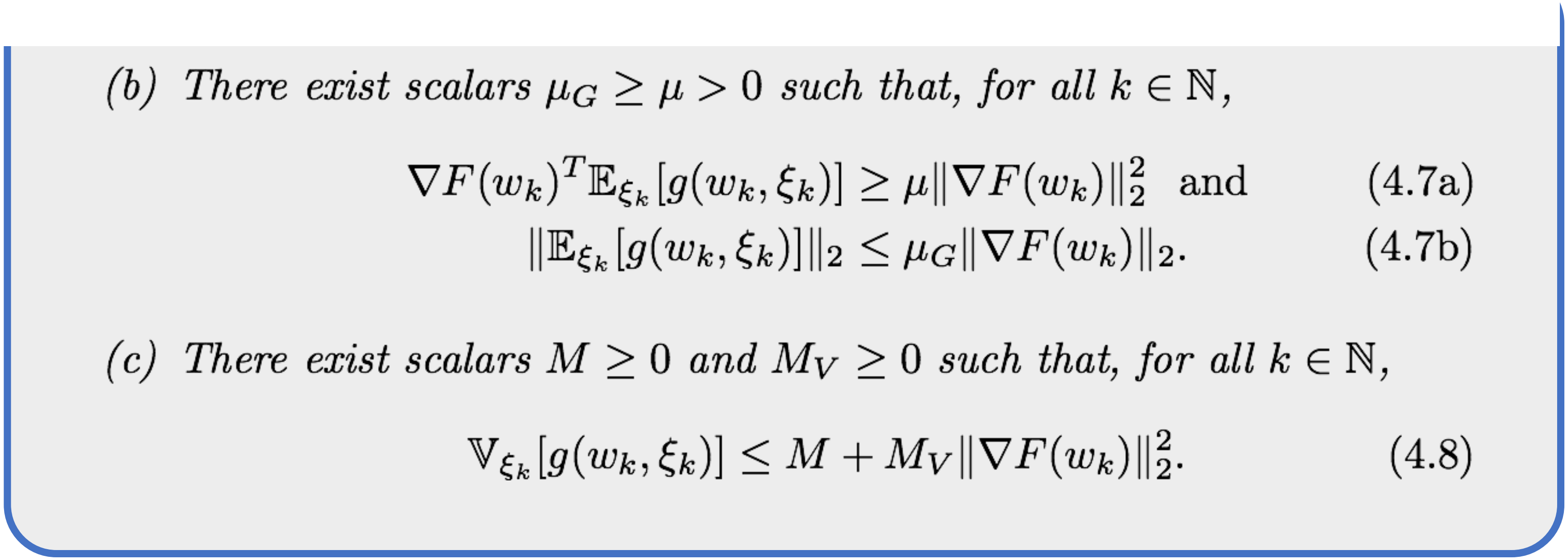
$$\|\mathbb{E}_{\xi_k}[g(w_k, \xi_k)]\|_2 \leq \mu_G \|\nabla F(w_k)\|_2. \tag{4.7b}$$

(c) There exist scalars $M \geq 0$ *and* $M_V \geq 0$ *such that, for all* $k \in \mathbb{N}$,

$$\mathbb{V}_{\xi_k}[g(w_k, \xi_k)] \leq M + M_V \|\nabla F(w_k)\|_2^2. \tag{4.8}$$

- $\mathbb{V}_{\xi_k}[\quad]$ denotes the variance w.r.t. ξ_k
- Variance of the noise must be bounded in a mild manner.

Moments



(b) There exist scalars $\mu_G \geq \mu > 0$ *such that, for all* $k \in \mathbb{N}$,

$$\nabla F(w_k)^T \mathbb{E}_{\xi_k}[g(w_k, \xi_k)] \geq \mu \|\nabla F(w_k)\|_2^2 \quad \text{and} \tag{4.7a}$$

$$\|\mathbb{E}_{\xi_k}[g(w_k, \xi_k)]\|_2 \leq \mu_G \|\nabla F(w_k)\|_2. \tag{4.7b}$$

(c) There exist scalars $M \geq 0$ *and* $M_V \geq 0$ *such that, for all* $k \in \mathbb{N}$,

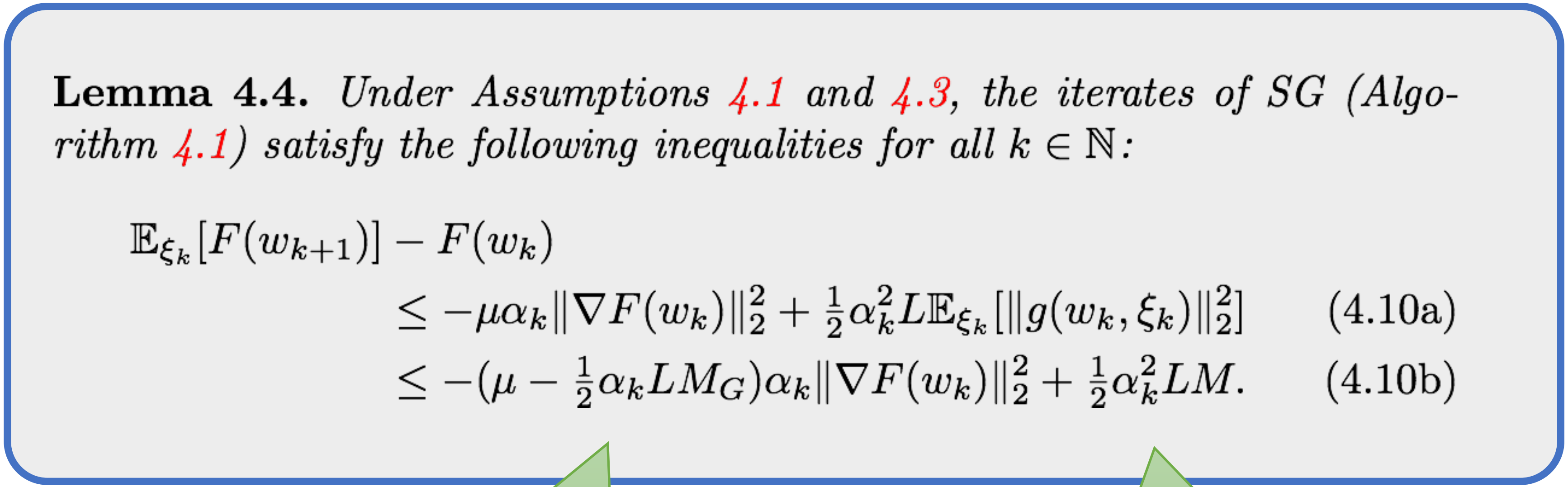
$$\mathbb{V}_{\xi_k}[g(w_k, \xi_k)] \leq M + M_V \|\nabla F(w_k)\|_2^2. \tag{4.8}$$

- Combining (4.7b) and (4.8) gives

$$\mathbb{E}_{\xi_k}[\|g(w_k, \xi_k)\|_2^2] \leq M + M_G \|\nabla F(w_k)\|_2^2 \tag{4.9}$$

with $M_G := M_V + \mu_G^2 \geq \mu^2 > 0.$

Moments



Lemma 4.4. *Under Assumptions 4.1 and 4.3, the iterates of SG (Algorithm 4.1) satisfy the following inequalities for all $k \in \mathbb{N}$:*

$$\begin{aligned}\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) & \\ &\leq -\mu\alpha_k\|\nabla F(w_k)\|_2^2 + \tfrac{1}{2}\alpha_k^2 L\mathbb{E}_{\xi_k}[\|g(w_k,\xi_k)\|_2^2] && (4.10a) \\ &\leq -(\mu - \tfrac{1}{2}\alpha_k L M_G)\alpha_k\|\nabla F(w_k)\|_2^2 + \tfrac{1}{2}\alpha_k^2 LM. && (4.10b)\end{aligned}$$

Expected decrease

Noise

- The convergence of SG depends on the balance between these two terms.

Moments

Lemma 4.4. *Under Assumptions 4.1 and 4.3, the iterates of SG (Algorithm 4.1) satisfy the following inequalities for all $k \in \mathbb{N}$:*

$$\begin{aligned} \mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) & \\ &\leq -\mu\alpha_k \|\nabla F(w_k)\|_2^2 + \tfrac{1}{2}\alpha_k^2 L \mathbb{E}_{\xi_k}[\|g(w_k,\xi_k)\|_2^2] \quad (4.10a) \\ &\leq -(\mu - \tfrac{1}{2}\alpha_k L M_G)\alpha_k \|\nabla F(w_k)\|_2^2 + \tfrac{1}{2}\alpha_k^2 L M. \quad (4.10b) \end{aligned}$$

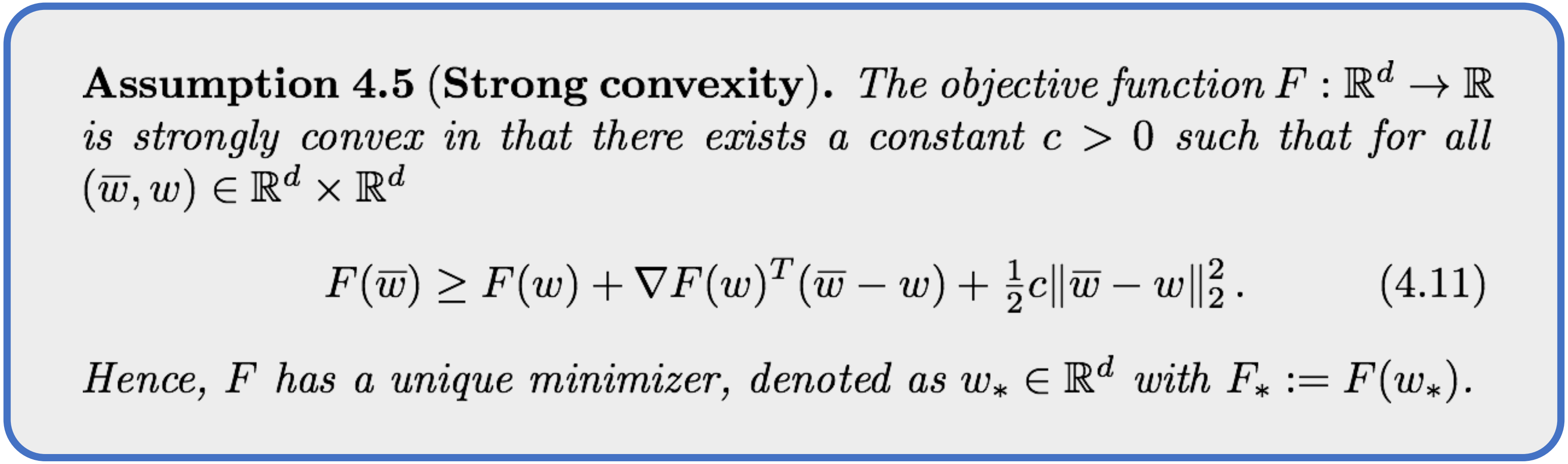
Proof. By Lemma 4.2 and (4.7a), it follows that

$$\begin{aligned} \mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) &\leq -\alpha_k \nabla F(w_k)^T \mathbb{E}_{\xi_k}[g(w_k,\xi_k)] + \tfrac{1}{2}\alpha_k^2 L \mathbb{E}_{\xi_k}[\|g(w_k,\xi_k)\|_2^2] \\ &\leq -\mu\alpha_k \|\nabla F(w_k)\|_2^2 + \tfrac{1}{2}\alpha_k^2 L \mathbb{E}_{\xi_k}[\|g(w_k,\xi_k)\|_2^2], \end{aligned}$$

which is (4.10a). Assumption 4.3, giving (4.9), then yields (4.10b). □

3- SG for Strongly Convex Objectives

Strong convexity



Assumption 4.5 (Strong convexity). *The objective function $F : \mathbb{R}^d \to \mathbb{R}$ is strongly convex in that there exists a constant $c > 0$ such that for all $(\overline{w}, w) \in \mathbb{R}^d \times \mathbb{R}^d$*

$$F(\overline{w}) \geq F(w) + \nabla F(w)^T(\overline{w} - w) + \tfrac{1}{2}c\|\overline{w} - w\|_2^2 . \qquad (4.11)$$

Hence, F has a unique minimizer, denoted as $w_* \in \mathbb{R}^d$ with $F_* := F(w_*)$.

Known consequence

$$2c(F(w) - F_*) \leq \|\nabla F(w)\|_2^2 \quad \text{for all} \quad w \in \mathbb{R}^d. \qquad (4.12)$$

Why does strong convexity matter?

- It gives the strongest results.
- It often happens in practice (one regularizes to facilitate optimization!)
- It describes any smooth function near a strong local minimum.

Total expectation

Different expectations

- $\mathbb{E}_{\xi_k}[\quad]$ is the expectation with respect to the distribution of ξ_k only.
- $\mathbb{E}[\quad]$ is the total expectation w.r.t. the joint distribution of all ξ_k.

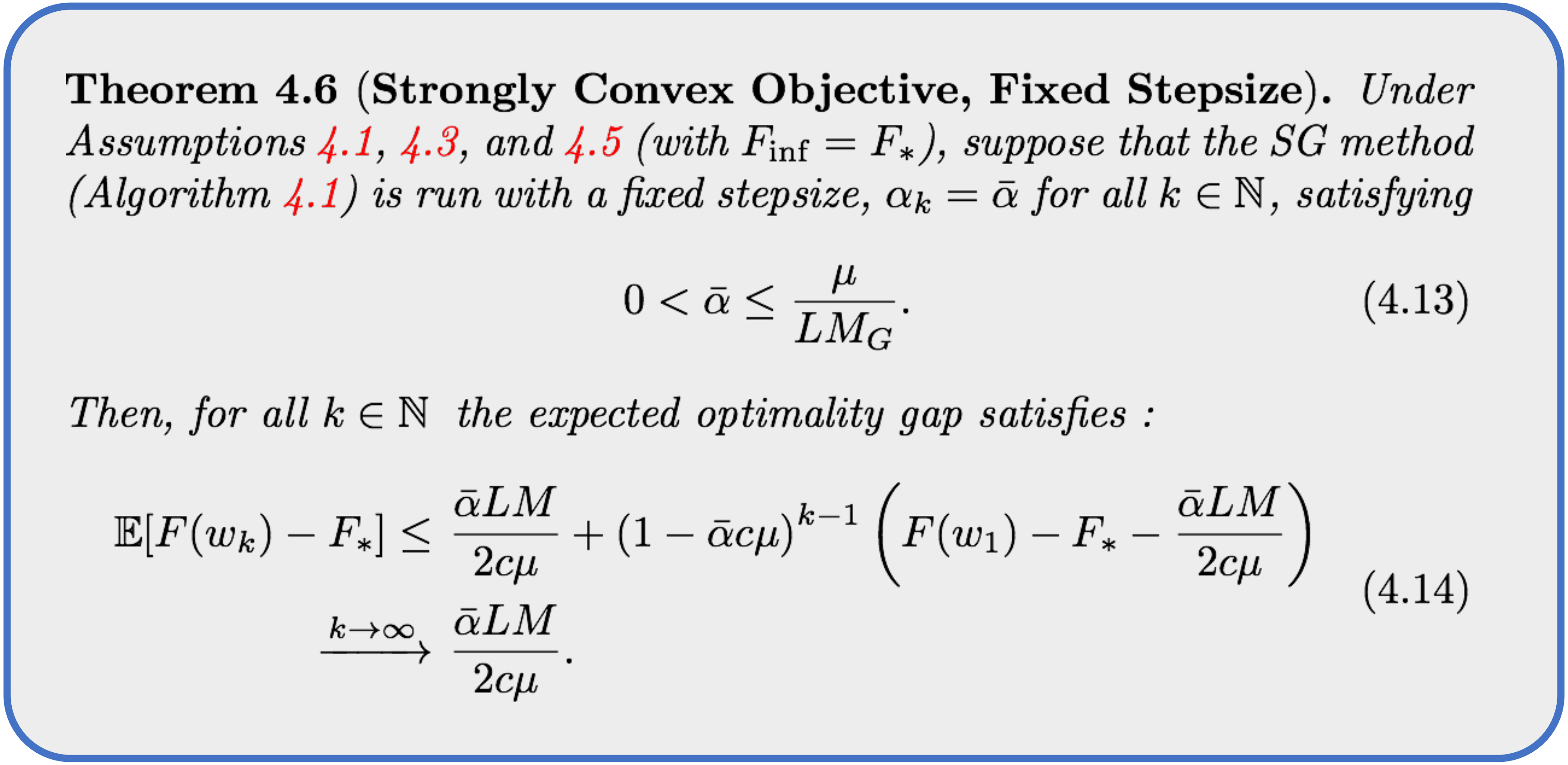
For instance, since w_k depends only on $\xi_1, \xi_2, \ldots, \xi_{k-1}$,

$$\mathbb{E}[F(w_k)] = \mathbb{E}_{\xi_1}\mathbb{E}_{\xi_2} \ldots \mathbb{E}_{\xi_{k-1}}[F(w_k)]$$

Results in expectation

- We focus on results that characterize the properties of SG in expectation.
- The stochastic approximation literature usually relies on rather complex martingale techniques to establish almost sure convergence results. We avoid them because they do not give much additional insight.

SG with fixed stepsize



Theorem 4.6 (Strongly Convex Objective, Fixed Stepsize). *Under Assumptions 4.1, 4.3, and 4.5 (with $F_{\text{inf}} = F_*$), suppose that the SG method (Algorithm 4.1) is run with a fixed stepsize, $\alpha_k = \bar{\alpha}$ for all $k \in \mathbb{N}$, satisfying*

$$0 < \bar{\alpha} \leq \frac{\mu}{L M_G}. \tag{4.13}$$

Then, for all $k \in \mathbb{N}$ the expected optimality gap satisfies :

$$\begin{aligned} \mathbb{E}[F(w_k) - F_*] &\leq \frac{\bar{\alpha} L M}{2c\mu} + (1 - \bar{\alpha} c\mu)^{k-1} \left(F(w_1) - F_* - \frac{\bar{\alpha} L M}{2c\mu} \right) \\ &\xrightarrow{k\to\infty} \frac{\bar{\alpha} L M}{2c\mu}. \end{aligned} \tag{4.14}$$

- Only converges to a neighborhood of the optimal value.
- Both (4.13) and (4.14) describe well the actual behavior.

SG with fixed stepsize (proof)

Proof. Using Lemma 4.4 with (4.13) and (4.12), we have for all $k \in \mathbb{N}$ that

$$\begin{aligned}\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k)] &\leq -(\mu - \tfrac{1}{2}\bar{\alpha}LM_G)\bar{\alpha}\|\nabla F(w_k)\|_2^2 + \tfrac{1}{2}\bar{\alpha}^2 LM \\ &\leq -\tfrac{1}{2}\bar{\alpha}\mu\|\nabla F(w_k)\|_2^2 + \tfrac{1}{2}\bar{\alpha}^2 LM \\ &\leq -\bar{\alpha}c\mu(F(w_k) - F_*) + \tfrac{1}{2}\bar{\alpha}^2 LM.\end{aligned}$$

Subtracting F_* from both sides and taking total expectations,

$$\mathbb{E}[F(w_{k+1}) - F_*] \leq (1 - \bar{\alpha}c\mu)\mathbb{E}[F(w_k) - F_*] + \tfrac{1}{2}\bar{\alpha}^2 LM.$$

Subtracting the constant $\bar{\alpha}LM/(2c\mu)$ from both sides, one obtains

$$\mathbb{E}[F(w_{k+1}) - F_*] - \frac{\bar{\alpha}LM}{2c\mu} \leq (1 - \bar{\alpha}c\mu)\left(\mathbb{E}[F(w_k) - F_*] - \frac{\bar{\alpha}LM}{2c\mu}\right). \qquad (4.15)$$

Observe that (4.15) is a contraction inequality since, by (4.13) and (4.9),

$$0 < \bar{\alpha}c\mu \leq \frac{c\mu^2}{LM_G} \leq \frac{c\mu^2}{L\mu^2} = \frac{c}{L} \leq 1. \qquad (4.16)$$

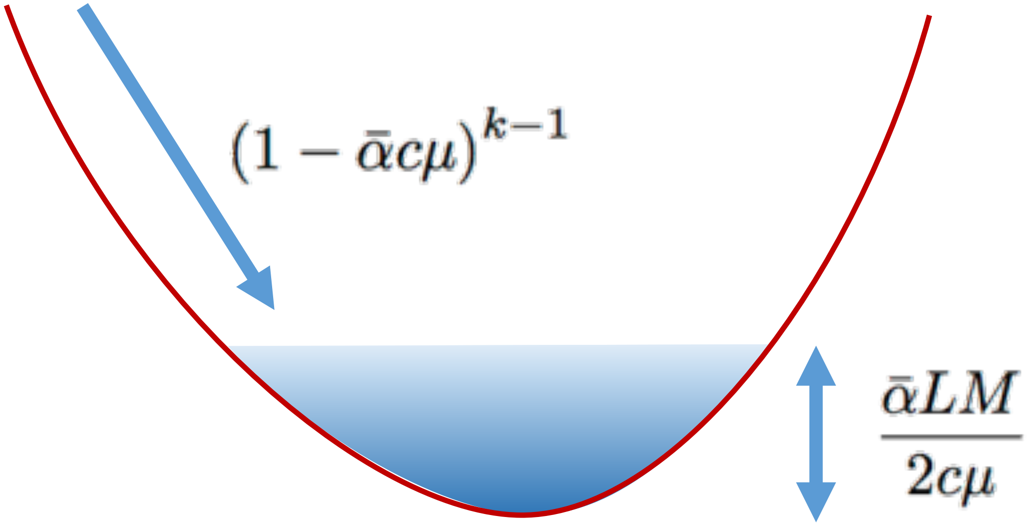
The result thus follows by applying (4.15) repeatedly. □

SG with fixed stepsize

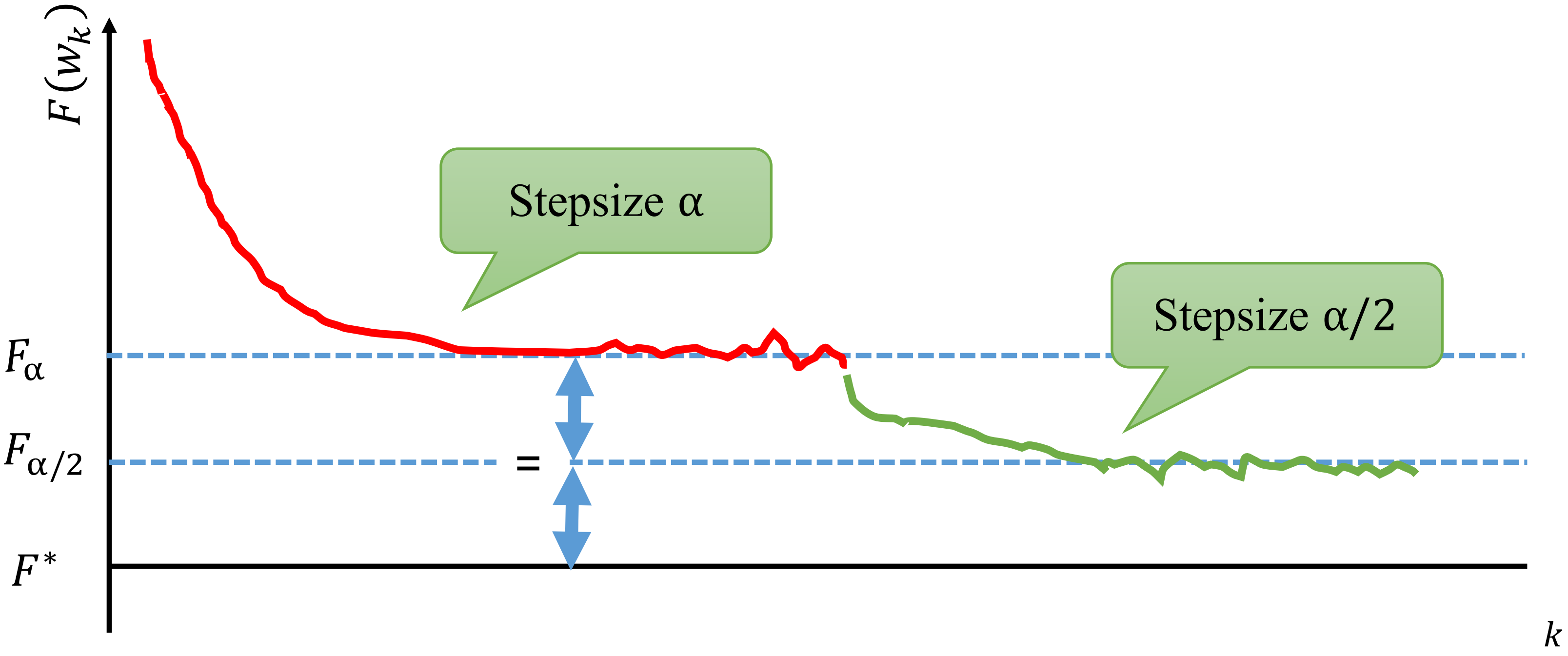
$$\mathbb{E}[F(w_k) - F_*] \leq \frac{\bar{\alpha} L M}{2c\mu} + (1 - \bar{\alpha} c\mu)^{k-1} \left(F(w_1) - F_* - \frac{\bar{\alpha} L M}{2c\mu} \right) \quad (4.14)$$

Note the interplay between the stepsize $\bar{\alpha}$ and the variance bound M.

- If $M = 0$, one recovers the linear convergence of batch gradient descent.
- If $M > 0$, one reaches a point where the noise prevents further progress.

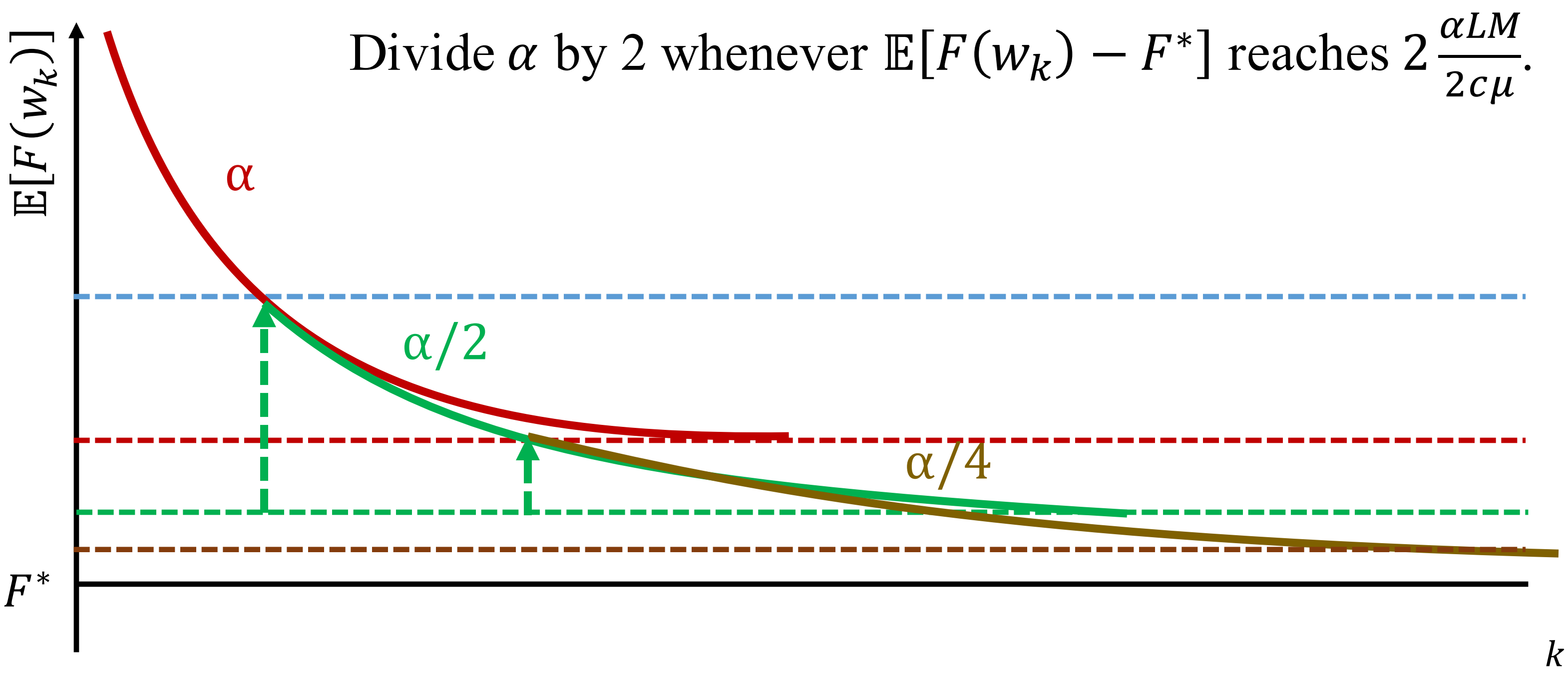

$(1 - \bar{\alpha} c\mu)^{k-1}$
$\frac{\bar{\alpha} L M}{2c\mu}$

Diminishing the stepsizes



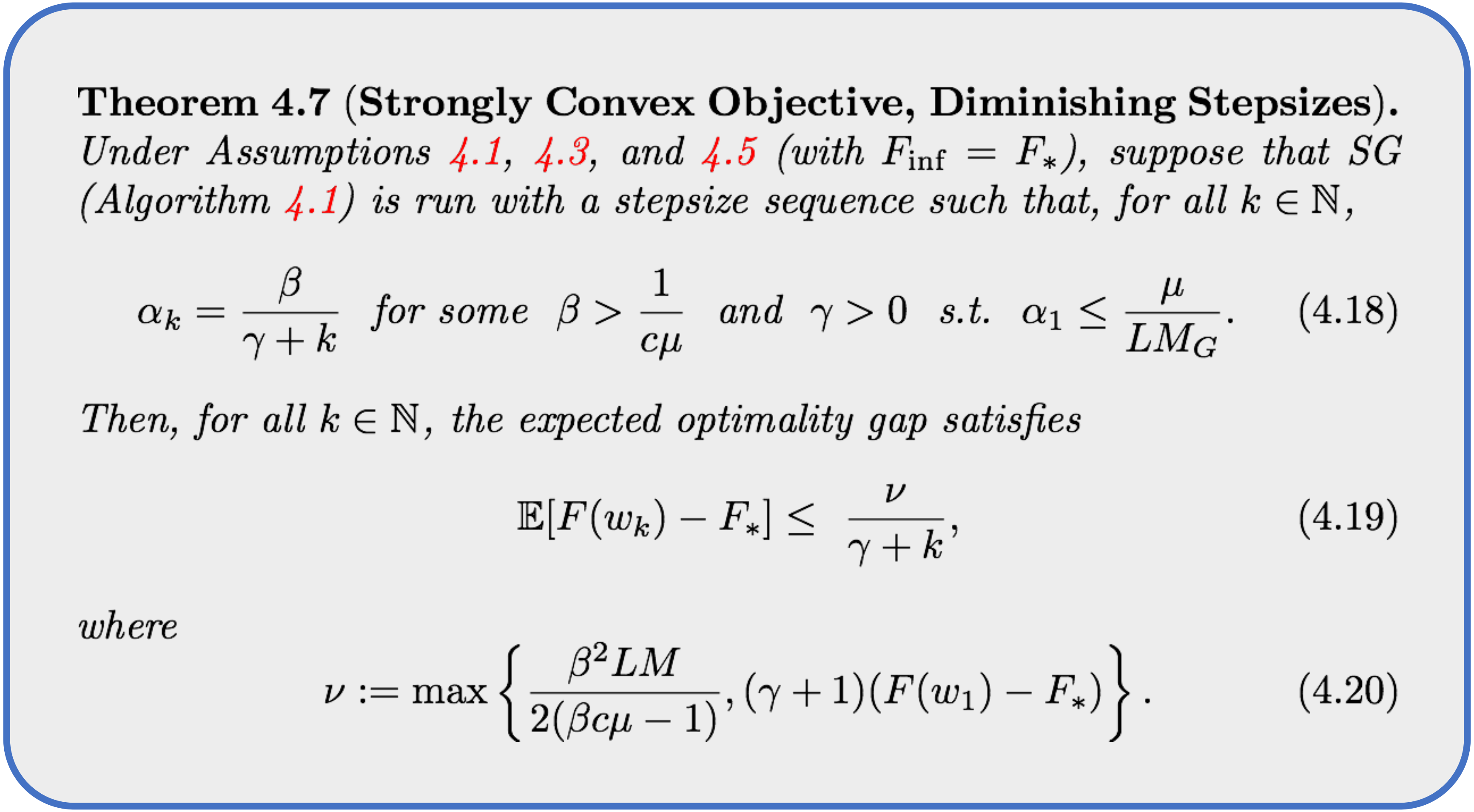
- If we wait long enough, halving the stepsize α eventually halves $F(w_k) - F^*$.
- We can even estimate $F^* \approx 2F_{\alpha/2} - F_\alpha$

Diminishing the stepsizes faster



- Divide α by 2 whenever $\mathbb{E}[F(w_k)]$ reaches $\alpha LM/c\mu$.
- Time τ_α between changes : $(1-\alpha c\mu)^{\tau_\alpha} = 1/3$ means $\tau_\alpha \propto 1/\alpha$.
- Whenever we halve α we must wait twice as long to halve $F(w) - F^*$.
- Overall convergence rate in $\mathcal{O}(1/k)$.

SG with diminishing stepsizes



Theorem 4.7 (Strongly Convex Objective, Diminishing Stepsizes). *Under Assumptions 4.1, 4.3, and 4.5 (with $F_{\inf} = F_*$), suppose that SG (Algorithm 4.1) is run with a stepsize sequence such that, for all $k \in \mathbb{N}$,*

$$\alpha_k = \frac{\beta}{\gamma + k} \quad \textit{for some} \quad \beta > \frac{1}{c\mu} \quad \textit{and} \quad \gamma > 0 \quad \textit{s.t.} \quad \alpha_1 \leq \frac{\mu}{LM_G}. \tag{4.18}$$

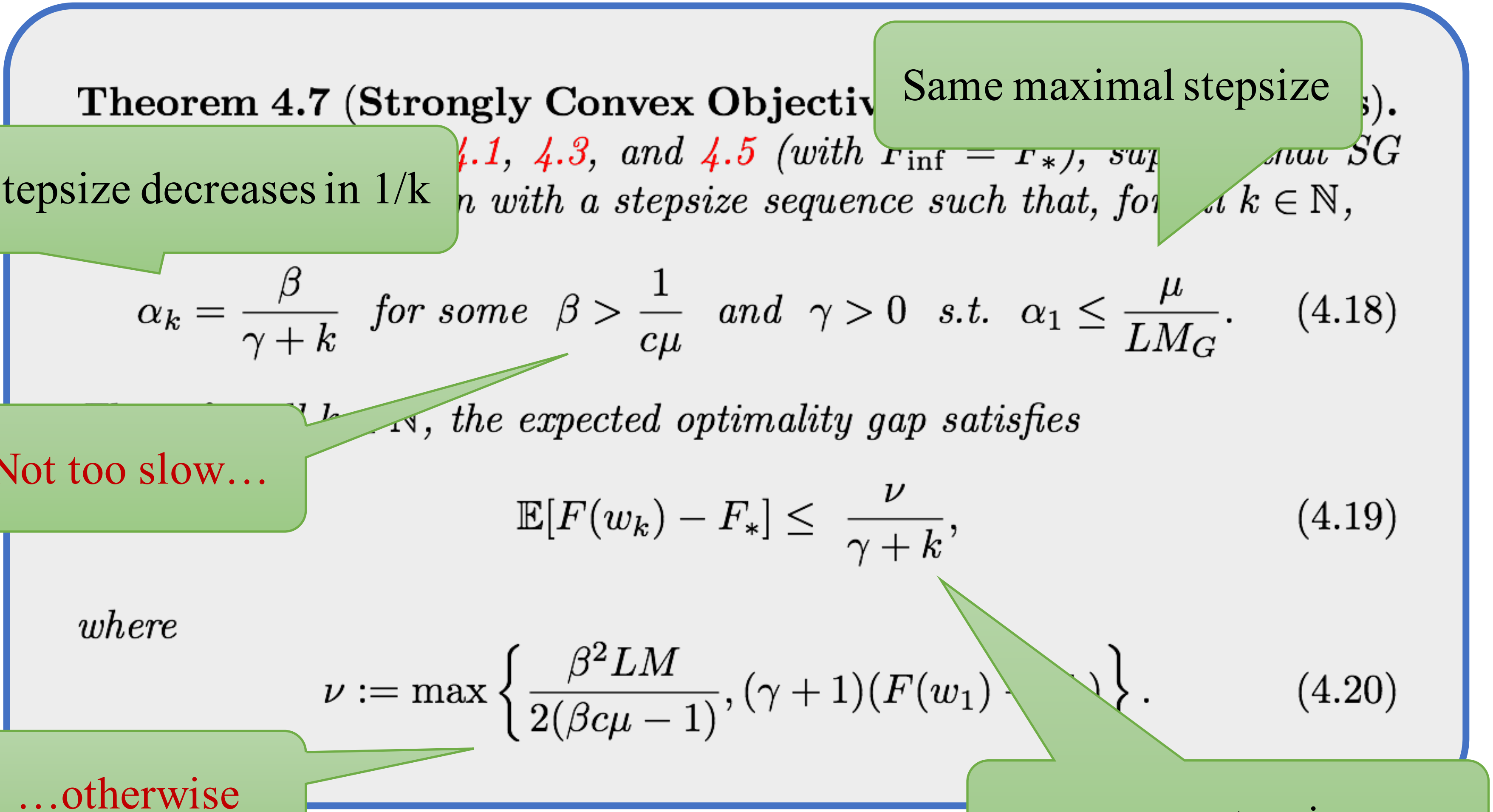
Then, for all $k \in \mathbb{N}$, the expected optimality gap satisfies

$$\mathbb{E}[F(w_k) - F_*] \leq \frac{\nu}{\gamma + k}, \tag{4.19}$$

where

$$\nu := \max\left\{\frac{\beta^2 LM}{2(\beta c\mu - 1)}, (\gamma + 1)(F(w_1) - F_*)\right\}. \tag{4.20}$$

SG with diminishing stepsizes



Theorem 4.7 (Strongly Convex Objectiv **s).**

4.1, 4.3, and 4.5 (with $F_{\text{inf}} = F_*$*), su* *hat SG*

n with a stepsize sequence such that, fo *ll* $k \in \mathbb{N}$,

$$\alpha_k = \frac{\beta}{\gamma + k} \ \text{ for some } \ \beta > \frac{1}{c\mu} \ \text{ and } \ \gamma > 0 \ \text{ s.t. } \ \alpha_1 \leq \frac{\mu}{LM_G}. \qquad (4.18)$$

$\mathbb{N}$*, the expected optimality gap satisfies*

$$\mathbb{E}[F(w_k) - F_*] \leq \frac{\nu}{\gamma + k}, \qquad (4.19)$$

where

$$\nu := \max\left\{\frac{\beta^2 LM}{2(\beta c\mu - 1)}, (\gamma + 1)(F(w_1) - \right\}. \qquad (4.20)$$

Same maximal stepsize

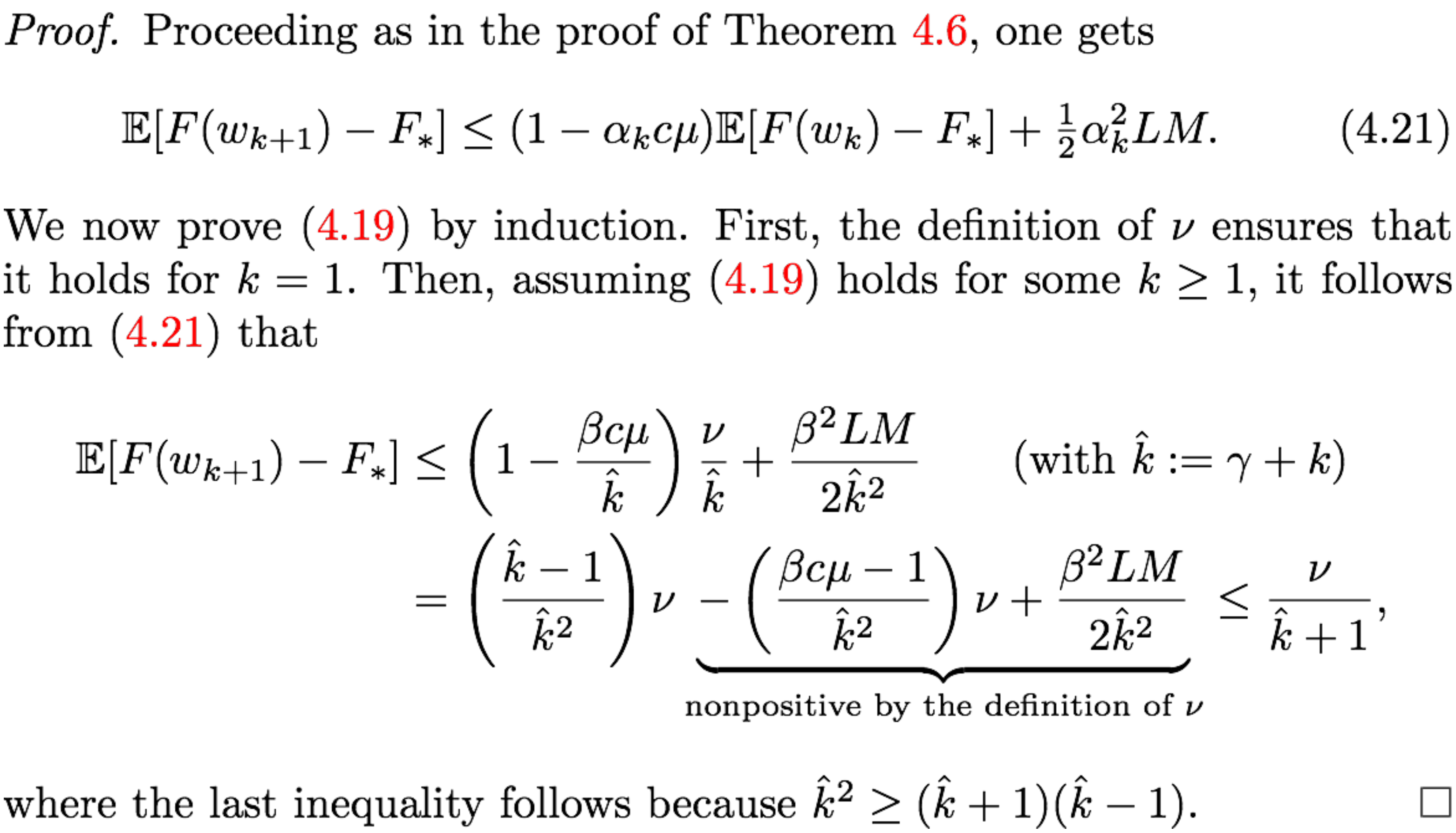
Stepsize decreases in 1/k

Not too slow…

…otherwise

gap ∝ stepsize

SG with diminishing stepsizes (proof)



Proof. Proceeding as in the proof of Theorem 4.6, one gets

$$\mathbb{E}[F(w_{k+1}) - F_*] \leq (1 - \alpha_k c\mu)\mathbb{E}[F(w_k) - F_*] + \tfrac{1}{2}\alpha_k^2 LM. \qquad (4.21)$$

We now prove (4.19) by induction. First, the definition of ν ensures that it holds for $k = 1$. Then, assuming (4.19) holds for some $k \geq 1$, it follows from (4.21) that

$$\begin{aligned}\mathbb{E}[F(w_{k+1}) - F_*] &\leq \left(1 - \frac{\beta c\mu}{\hat{k}}\right)\frac{\nu}{\hat{k}} + \frac{\beta^2 LM}{2\hat{k}^2} \qquad (\text{with } \hat{k} := \gamma + k) \\ &= \left(\frac{\hat{k}-1}{\hat{k}^2}\right)\nu \underbrace{- \left(\frac{\beta c\mu - 1}{\hat{k}^2}\right)\nu + \frac{\beta^2 LM}{2\hat{k}^2}}_{\text{nonpositive by the definition of } \nu} \leq \frac{\nu}{\hat{k}+1},\end{aligned}$$

where the last inequality follows because $\hat{k}^2 \geq (\hat{k}+1)(\hat{k}-1)$. □

Mini batching

		Computation	Noise
$g(w_k, \xi_k) =$	$\nabla f(w_k; \xi_k)$	1	M
	$\frac{1}{n_{\mathrm{mb}}} \sum_{i=1}^{n_{\mathrm{mb}}} \nabla f(w_k; \xi_{k,i})$	n_{mb}	M/n_{mb}

Using minibatches with stepsize $\bar{\alpha}$:

$$\mathbb{E}[F(w_k) - F_*] \leq \frac{\bar{\alpha} L M}{2c\mu\, n_{\mathrm{mb}}} + [1 - \bar{\alpha} c \mu]^{k-1} \left(F(w_1) - F_* - \frac{\bar{\alpha} L M}{2c\mu\, n_{\mathrm{mb}}} \right).$$

Using single example with stepsize $\bar{\alpha}$ / n_{mb} :

$$\mathbb{E}[F(w_k) - F_*] \leq \frac{\bar{\alpha} L M}{2c\mu\, n_{\mathrm{mb}}} + \left[1 - \frac{\bar{\alpha} c \mu}{n_{\mathrm{mb}}}\right]^{k-1} \left(F(w_1) - F_* - \frac{\bar{\alpha} L M}{2c\mu\, n_{\mathrm{mb}}} \right).$$

n_{mb} times more iterations that are n_{mb} times cheaper.

same

Minibatching

Ignoring implementation issues

- We can match minibatch SG with stepsize $\bar{\alpha}$
  using single example SG with stepsize $\bar{\alpha}\ /\ n_{\mathrm{mb}}$.
- We can match single example SG with stepsize $\bar{\alpha}$
  using minibatch SG with stepsize $\bar{\alpha} \times n_{\mathrm{mb}}$
  provided $\bar{\alpha} \times n_{\mathrm{mb}}$ is smaller than the max stepsize.

With implementation issues

- Minibatch implementations use the hardware better.
- Especially on GPU.

4- SG for General Objectives

Nonconvex objectives

Nonconvex training objectives are pervasive in deep learning.

Nonconvex landscape in high dimension can be very complex.

- Critical points can be local minima or saddle points.
- Critical points can be first order of high order.
- Critical points can be part of critical manifolds.
- A critical manifold can contain both local minima and saddle points.

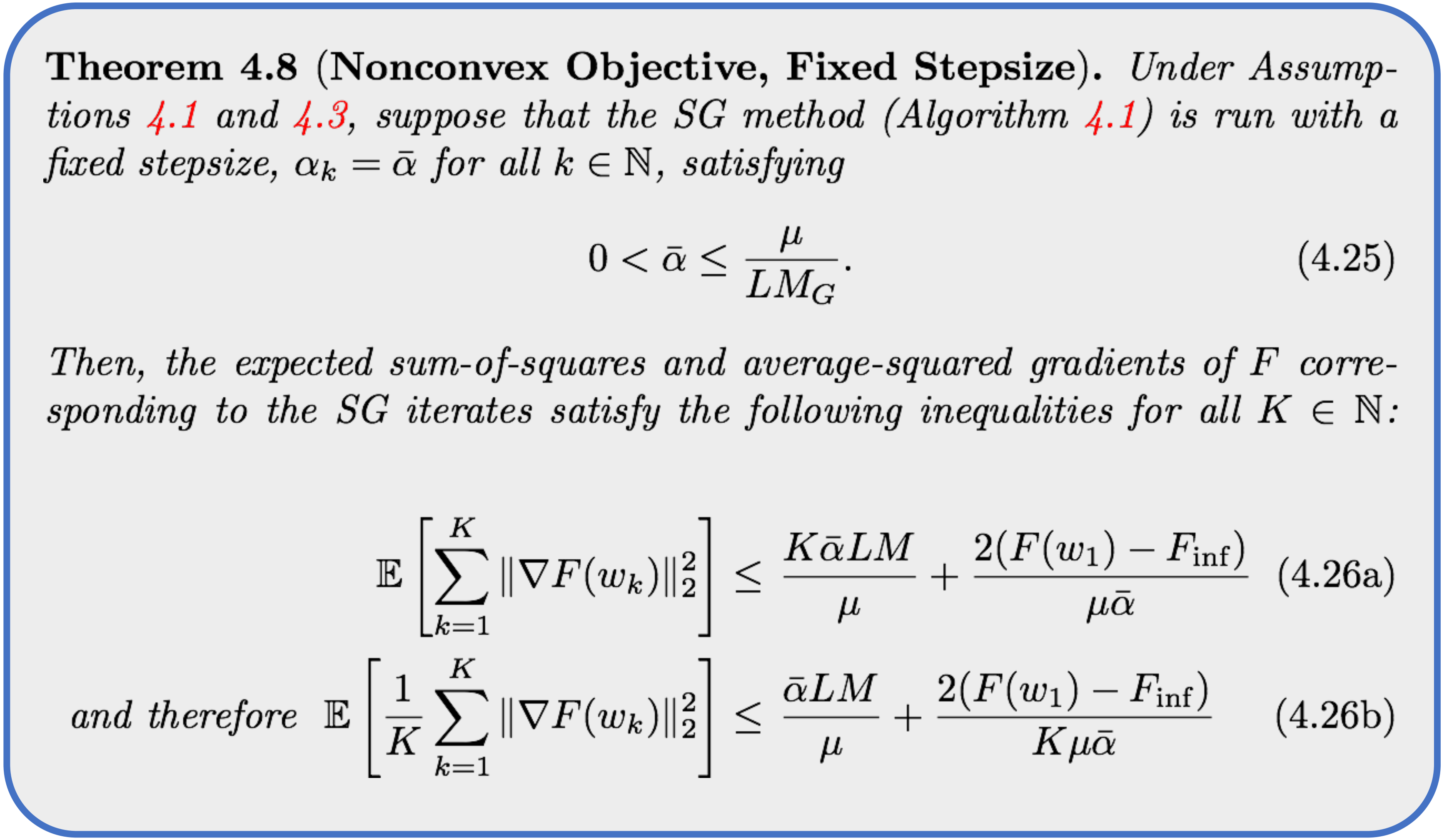
We describe meaningful (but weak) guarantees

- Essentially, SG goes to critical points.

The SG noise plays an important role in practice

- It seems to help navigating local minima and saddle points.
- More noise has been found to sometimes help optimization.
- But the theoretical understanding of these facts is weak.

Nonconvex SG with fixed stepsize



Theorem 4.8 (Nonconvex Objective, Fixed Stepsize). *Under Assumptions 4.1 and 4.3, suppose that the SG method (Algorithm 4.1) is run with a fixed stepsize,* $\alpha_k = \bar{\alpha}$ *for all* $k \in \mathbb{N}$, *satisfying*

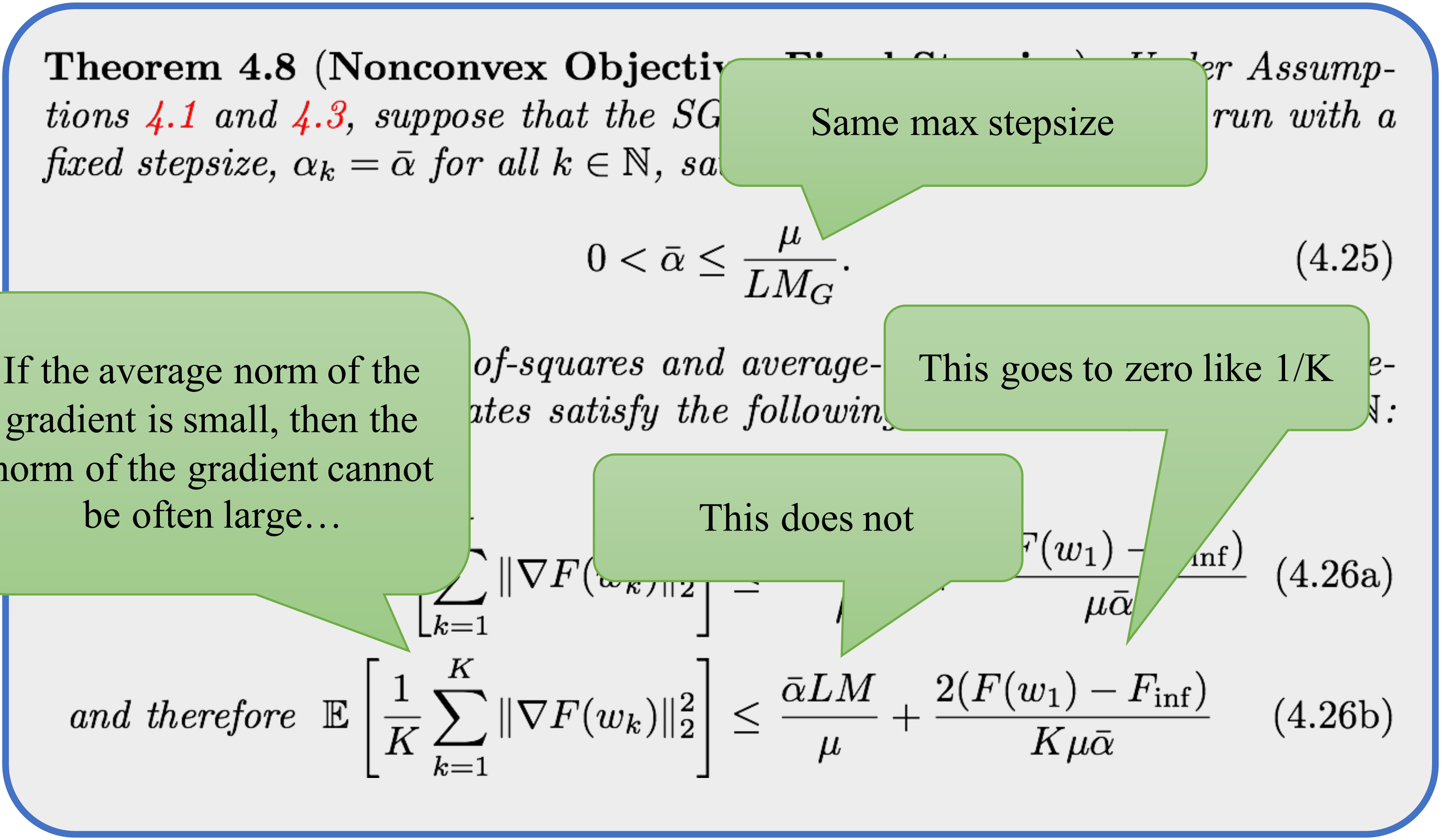
$$0 < \bar{\alpha} \leq \frac{\mu}{LM_G}. \tag{4.25}$$

Then, the expected sum-of-squares and average-squared gradients of F *corresponding to the SG iterates satisfy the following inequalities for all* $K \in \mathbb{N}$:

$$\mathbb{E}\left[\sum_{k=1}^{K} \|\nabla F(w_k)\|_2^2\right] \leq \frac{K\bar{\alpha}LM}{\mu} + \frac{2(F(w_1) - F_{\text{inf}})}{\mu\bar{\alpha}} \tag{4.26a}$$

and therefore $$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} \|\nabla F(w_k)\|_2^2\right] \leq \frac{\bar{\alpha}LM}{\mu} + \frac{2(F(w_1) - F_{\text{inf}})}{K\mu\bar{\alpha}} \tag{4.26b}$$

Nonconvex SG with fixed stepsize



Theorem 4.8 (Nonconvex Objectiv *ler Assumptions 4.1 and 4.3, suppose that the SG* *run with a fixed stepsize, $\alpha_k = \bar{\alpha}$ for all $k \in \mathbb{N}$, sa*

$$0 < \bar{\alpha} \le \frac{\mu}{L M_G}. \qquad (4.25)$$

of-squares and average- *e-* *ates satisfy the followin* *$\mathbb{N}$:*

$$\left[\sum_{k=1} \|\nabla F(\qquad \frac{F(w_1) - \quad \text{inf})}{\mu\bar{\alpha}} \qquad (4.26a)$$

$$\textit{and therefore} \quad \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(w_k)\|_2^2\right] \le \frac{\bar{\alpha} L M}{\mu} + \frac{2(F(w_1) - F_{\inf})}{K\mu\bar{\alpha}} \qquad (4.26b)$$

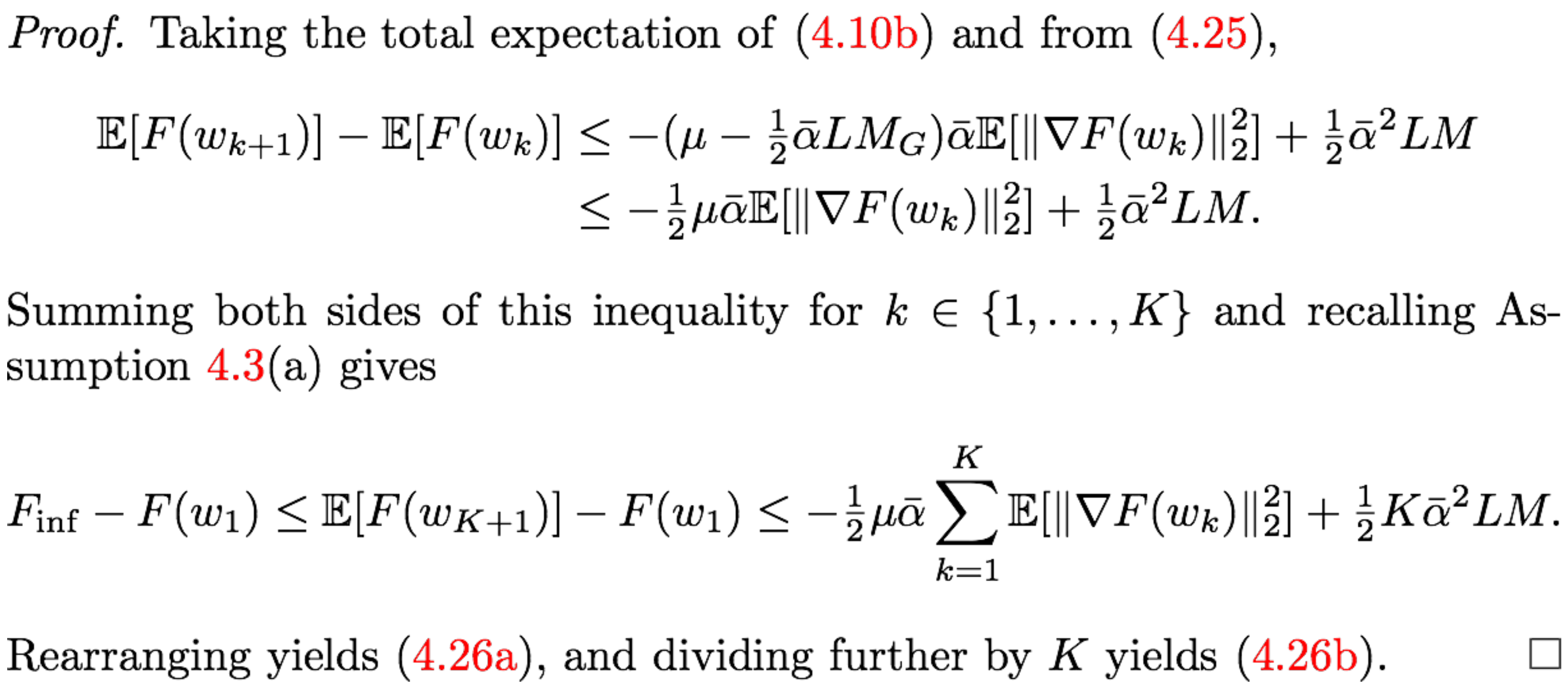
Same max stepsize

If the average norm of the gradient is small, then the gradient cannot be often large…

This goes to zero like 1/K

This does not

Nonconvex SG with fixed stepsize (proof)



Proof. Taking the total expectation of (4.10b) and from (4.25),

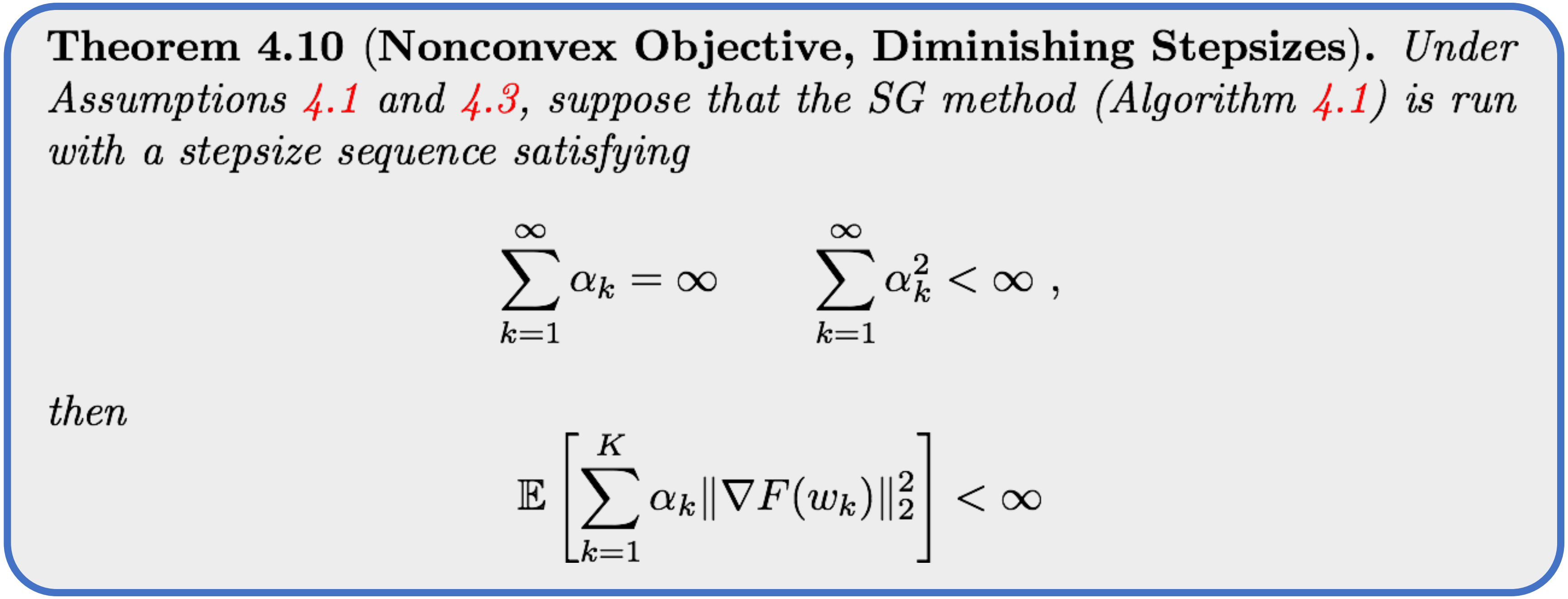
$$\begin{aligned}\mathbb{E}[F(w_{k+1})] - \mathbb{E}[F(w_k)] &\leq -(\mu - \tfrac{1}{2}\bar{\alpha} L M_G)\bar{\alpha}\mathbb{E}[\|\nabla F(w_k)\|_2^2] + \tfrac{1}{2}\bar{\alpha}^2 L M \\ &\leq -\tfrac{1}{2}\mu\bar{\alpha}\mathbb{E}[\|\nabla F(w_k)\|_2^2] + \tfrac{1}{2}\bar{\alpha}^2 L M.\end{aligned}$$

Summing both sides of this inequality for $k \in \{1, \ldots, K\}$ and recalling Assumption 4.3(a) gives

$$F_{\text{inf}} - F(w_1) \leq \mathbb{E}[F(w_{K+1})] - F(w_1) \leq -\tfrac{1}{2}\mu\bar{\alpha}\sum_{k=1}^{K}\mathbb{E}[\|\nabla F(w_k)\|_2^2] + \tfrac{1}{2}K\bar{\alpha}^2 L M.$$

Rearranging yields (4.26a), and dividing further by K yields (4.26b). □

Nonconvex SG with diminishing step sizes

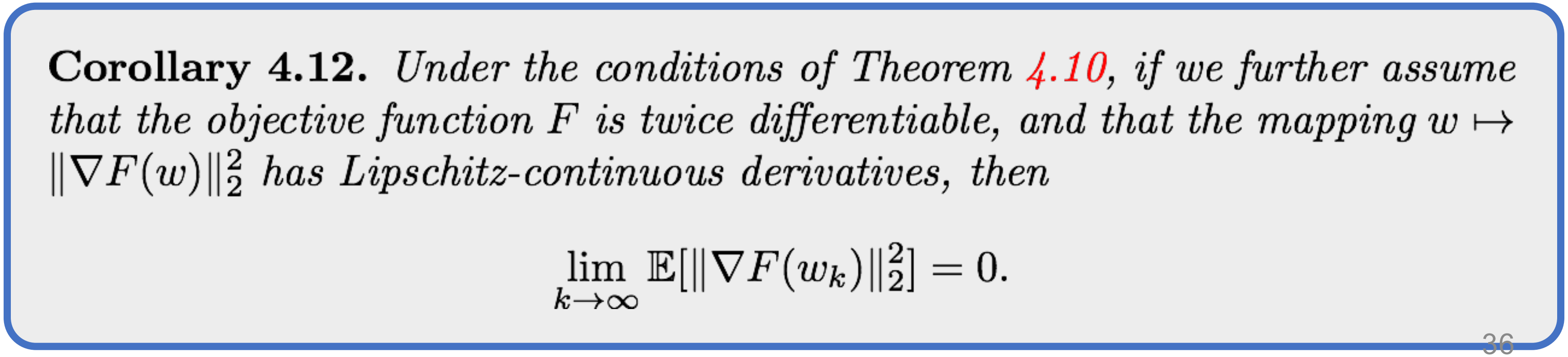


Theorem 4.10 (Nonconvex Objective, Diminishing Stepsizes). *Under Assumptions 4.1 and 4.3, suppose that the SG method (Algorithm 4.1) is run with a stepsize sequence satisfying*

$$\sum_{k=1}^{\infty} \alpha_k = \infty \qquad \sum_{k=1}^{\infty} \alpha_k^2 < \infty ,$$

then

$$\mathbb{E}\left[\sum_{k=1}^{K} \alpha_k \|\nabla F(w_k)\|_2^2\right] < \infty$$



Corollary 4.12. *Under the conditions of Theorem 4.10, if we further assume that the objective function F is twice differentiable, and that the mapping $w \mapsto \|\nabla F(w)\|_2^2$ has Lipschitz-continuous derivatives, then*

$$\lim_{k\to\infty} \mathbb{E}[\|\nabla F(w_k)\|_2^2] = 0.$$

5- Work complexity for Large-Scale Learning

Large-Scale Learning

Assume that we are in the large data regime

- Training data is essentially unlimited.
- Computation time is limited.

The good

- More training data ⇒ less overfitting
- Less overfitting ⇒ richer models.

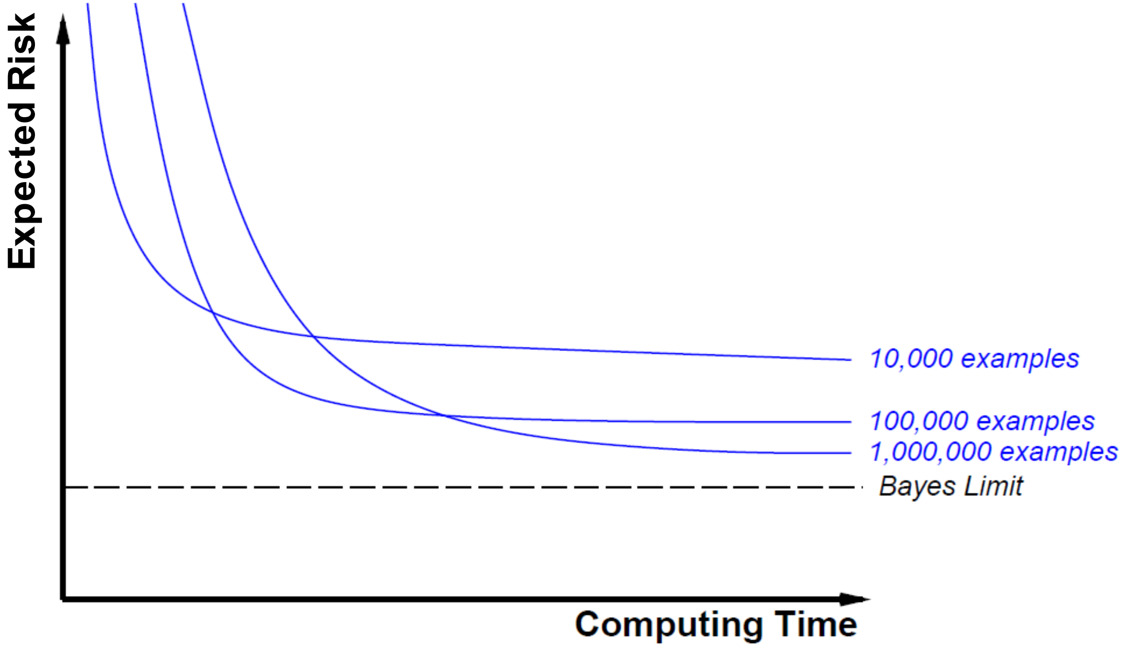
The bad

- Using more training data or rich models quickly exhausts the time budget.

The hope

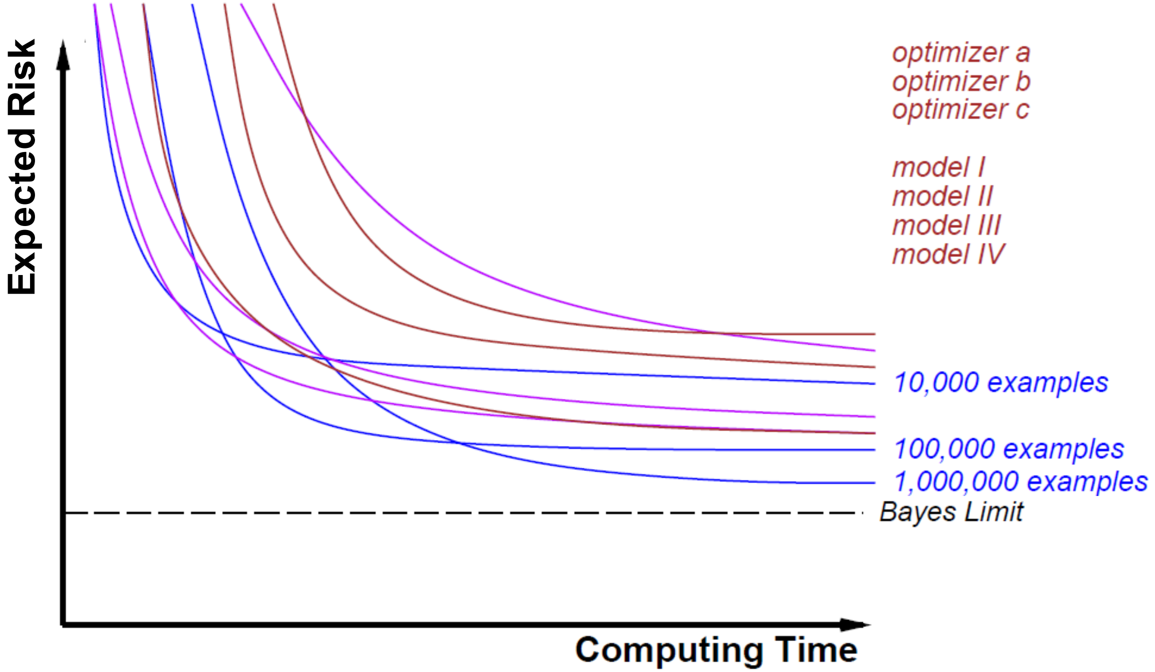
- How thoroughly do we need to optimize $R_n(w)$ when we actually want another function $R(w)$ to be small ?

Expected risk versus training time



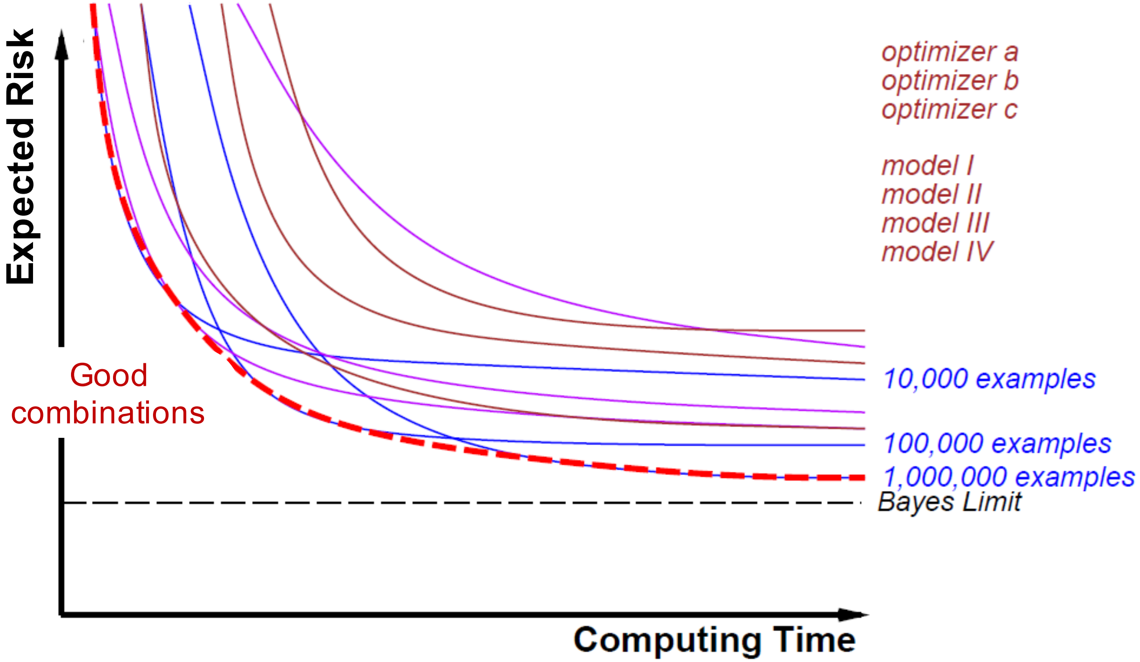
- When we vary the number of examples

Expected risk versus training time



- When we vary the number of examples, the model, and the optimizer…

Expected risk versus training time



- The optimal combination depends on the computing time budget

Formalization

The components of the expected risk

$$\mathbb{E}[R(\tilde{w}_n)] = \underbrace{R(w_*)}_{\mathcal{E}_{app}(\mathcal{H})} + \underbrace{\mathbb{E}[R(w_n) - R(w_*)]}_{\mathcal{E}_{est}(\mathcal{H},n)} + \underbrace{\mathbb{E}[R(\tilde{w}_n) - R(w_n)]}_{\mathcal{E}_{opt}(\mathcal{H},n,\epsilon)} \qquad (4.29)$$

Question

- Given a fixed model $\mathcal{H}$ and a time budget $\mathcal{T}_{\max}$, choose $n, \epsilon \ldots$

$$\min_{n,\epsilon} \ \mathcal{E}(n,\epsilon) = \mathbb{E}[R(\tilde{w}_n) - R(w_*)] \ \text{s.t.} \ \mathcal{T}(n,\epsilon) \leq \mathcal{T}_{\max}. \qquad (4.30)$$

Approach

- Statistics tell us $\mathcal{E}_{est}(n)$ decreases with a rate in range $1/\sqrt{n} \ldots 1/n$.
- For now, let's work with the fastest rate compatible with statistics

$$\mathcal{E}(n,\epsilon) \sim \frac{1}{n} + \epsilon \qquad (4.32)$$

Batch versus Stochastic

Typical convergence rates

- Batch algorithm: $\mathcal{T}(n,\epsilon) \sim n \log(1/\epsilon)$
- Stochastic algorithm: $\mathcal{T}(n,\epsilon) \sim 1/n$

Rate analysis

		Batch	Stochastic
$\mathcal{T}(n,\epsilon)$	$\sim$	$n \log\left(\frac{1}{\epsilon}\right)$	$\frac{1}{\epsilon}$
n^*	$\sim$	$\frac{\mathcal{T}_{\max}}{\log(\mathcal{T}_{\max})}$	$\mathcal{T}_{\max}$
$\mathcal{E}^*$	$\sim$	$\frac{\log(\mathcal{T}_{\max})}{\mathcal{T}_{\max}} + \frac{1}{\mathcal{T}_{\max}}$	$\frac{1}{\mathcal{T}_{\max}}$

Processing more training examples beats optimizing more thoroughly.

This effect only grows if $\mathcal{E}_{est}(n)$ decreases slower than $1/n$.

6- Comments

Asymptotic performance of SG is fragile

Diminishing stepsizes are tricky

- Theorem 4.7 (strongly convex function) suggests

$$\alpha_k = \frac{\beta}{\gamma + k}$$

SG converges very slowly if $\beta < \frac{1}{c\mu}$

SG usually diverges when α is above $\frac{2\mu}{LM_G}$

Constant stepsizes are often used in practice

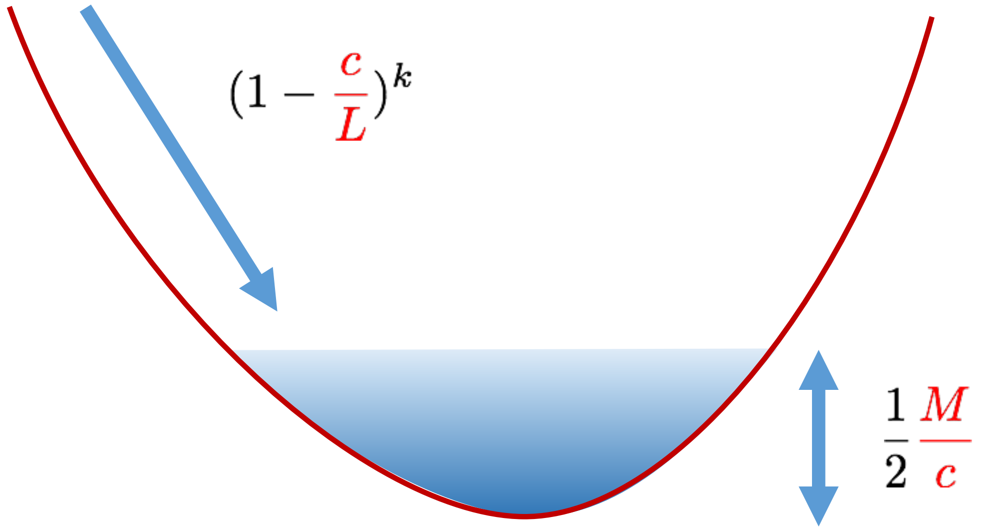
- Sometimes with a simple halving protocol.

Spoiler – Certain SG variants are more robust.

Condition numbers

The ratios $\frac{L}{c}$ and $\frac{M}{c}$ appear in critical places

- Theorem 4.6. With $\mu = 1, M_V = 0$, the optimal stepsize is $\bar{\alpha} = \frac{1}{L}$



Distributed computing

SG is notoriously hard to parallelize

- Because it updates the parameters w with high frequency
- Because it slows down with delayed updates.

SG still works with relaxed synchronization

- Because this is just a little bit more noise.

Communication overhead give room for new opportunities

- There is ample time to compute things while communication takes place.
- Opportunity for optimization algorithms with higher per-iteration costs

➔ SG may not be the best answer for distributed training.

Smoothness versus Convexity

Analyses of SG that only rely on convexity

- Bounding $\|w_k - w^*\|^2$ instead of $F(w_k) - F^*$ and assuming $\mathbb{E}_{\xi_k}[g(w_k, \xi_k)] = \hat{g}(w_k) \in \partial F(w_k)$ gives a result similar to Lemma 4.4.

$$\begin{aligned}\mathbb{E}_{\xi_k}[\|w_{k+1} - w_*\|_2^2] - \|w_k - w_*\|_2^2 \\ = \; \underbrace{-\,2\alpha_k \hat{g}(w_k)^T (w_k - w_*)}_{\text{Expected decrease}} + \alpha_k^2 \underbrace{\mathbb{E}_{\xi_k}[\|g(w_k, \xi_k)\|_2^2]}_{\text{Noise}}, \qquad \text{(A.2)}\end{aligned}$$

- Ways to bound the expected decrease

$$\begin{aligned}&\text{General convexity :} \quad \hat{g}(w_k)^T (w_k - w_*) \geq F(w_k) - F(w_*) \geq 0 \\ &\text{Strong convexity :} \quad \hat{g}(w_k)^T (w_k - w_*) \geq c\|w_k - w_*\|^2 \geq 0\end{aligned}$$

- Proof does not easily support second order methods.

Beyond SG: Noise Reduction and Second-Order Methods

https://arxiv.org/abs/1606.04838

Frank E. Curtis, Lehigh University

joint work with

Léon Bottou, Facebook AI Research
Jorge Nocedal, Northwestern University

International Conference on Machine Learning (ICML)
New York, NY, USA

19 June 2016

Outline

SG

Noise Reduction Methods

Second-Order Methods

Other Methods

What have we learned about SG?

Assumption $\langle L/c \rangle$

The objective function $F : \mathbb{R}^d \to \mathbb{R}$ *is*

- c*-strongly convex (*$\Rightarrow$ *unique minimizer) and*
- L*-smooth (i.e.,* ∇F *is Lipschitz continuous with constant* L*).*

Theorem SG (sublinear convergence)

Under Assumption $\langle L/c \rangle$ *and* $\mathbb{E}_{\xi_k}[\|g(w_k,\xi_k)\|_2^2] \leq M + \mathcal{O}(\|\nabla F(w_k)\|_2^2)$,

$$w_{k+1} \leftarrow w_k - \alpha_k g(w_k,\xi_k)$$

yields

$$\alpha_k = \frac{1}{L} \qquad \Longrightarrow \mathbb{E}[F(w_k) - F_*] \to \frac{M}{2c};$$

$$\alpha_k = \mathcal{O}\left(\frac{1}{k}\right) \qquad \Longrightarrow \mathbb{E}[F(w_k) - F_*] = \mathcal{O}\left(\frac{(L/c)(M/c)}{k}\right).$$

(*Let's assume unbiased gradient estimates; see paper for more generality.)

Illustration

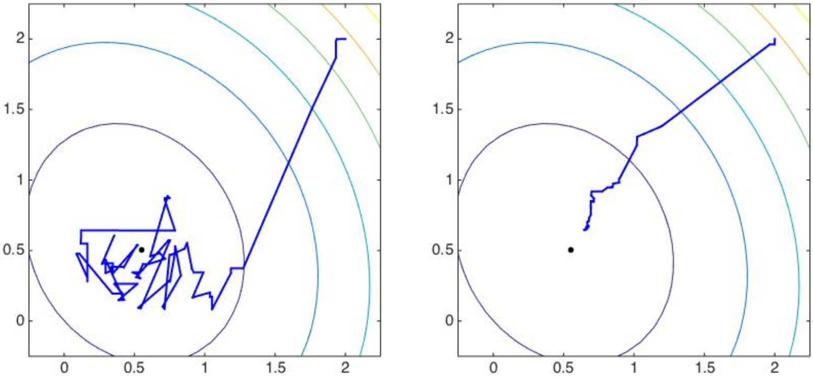
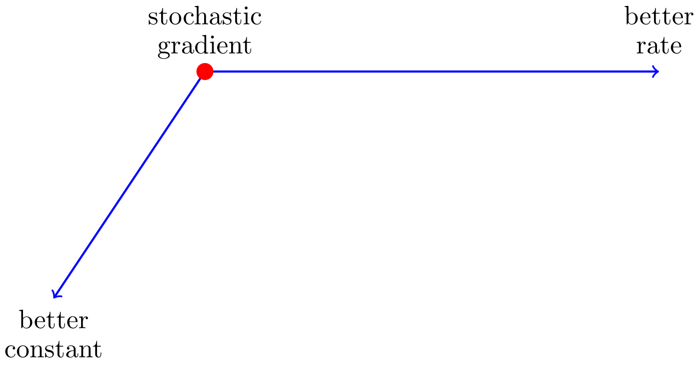
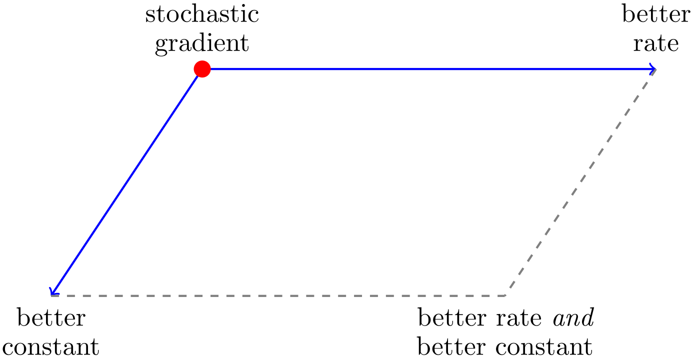


Figure: SG run with a fixed stepsize (left) vs. diminishing stepsizes (right)

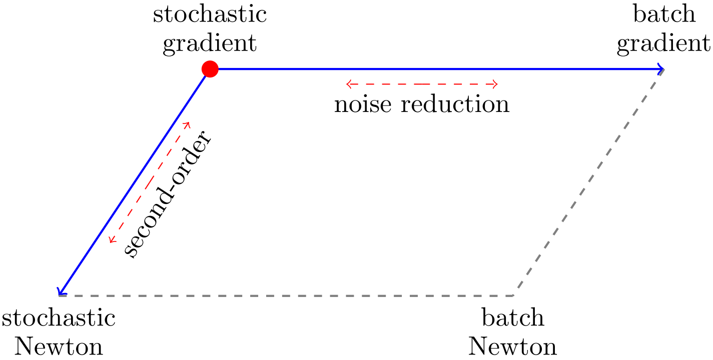
What can be improved?



What can be improved?



Two-dimensional schematic of methods



Nonconvex objectives

Despite loss of convergence rate, motivation for nonconvex problems as well:

- Convex results describe behavior near strong local minimizer
- Batch gradient methods are unlikely to get trapped near saddle points
- Second-order information can
  - avoid negative effects of nonlinearity and ill-conditioning
  - *require* mini-batching (noise reduction) to be efficient

Conclusion: explore entire plane, not just one axis

Outline

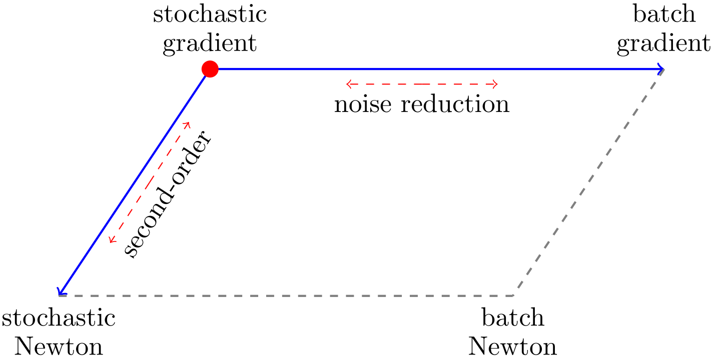
SG

Noise Reduction Methods

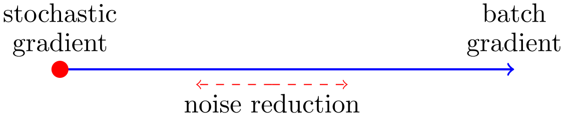
Second-Order Methods

Other Methods

Two-dimensional schematic of methods

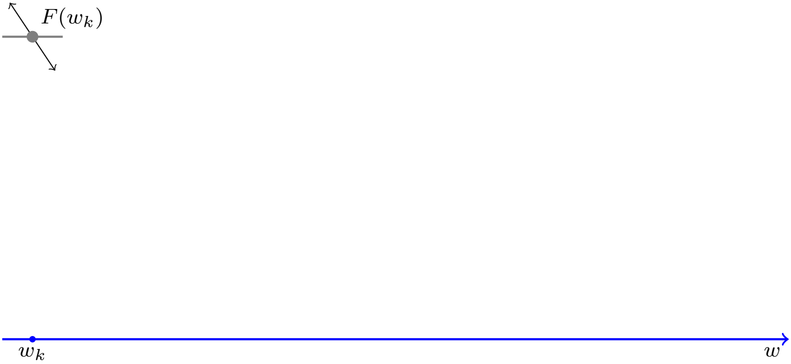


2D schematic: Noise reduction methods

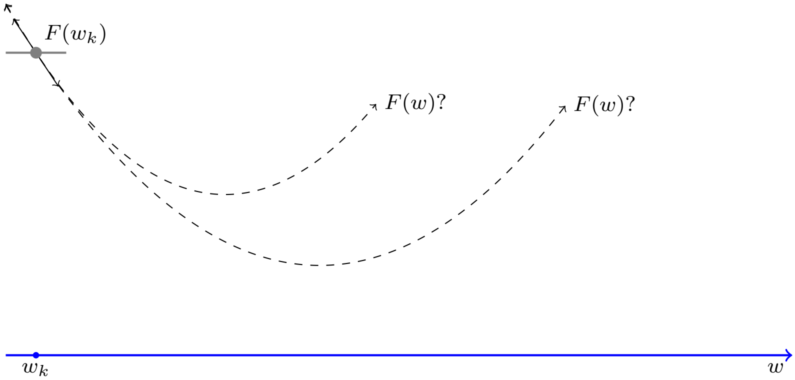


- dynamic sampling
- gradient aggregation
- iterate averaging

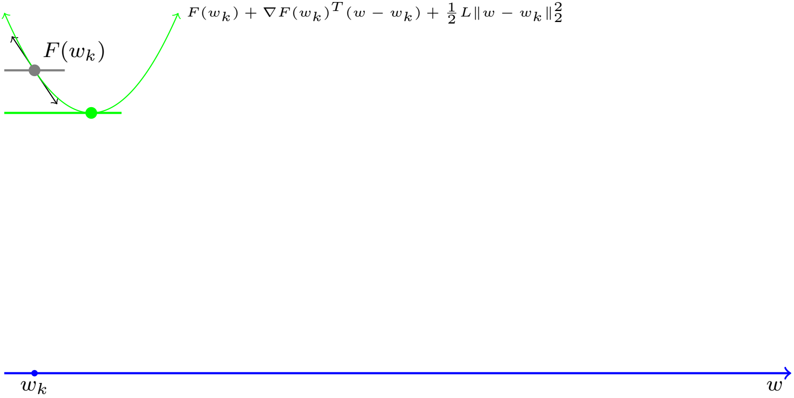
Ideal: Linear convergence of a batch gradient method



Ideal: Linear convergence of a batch gradient method

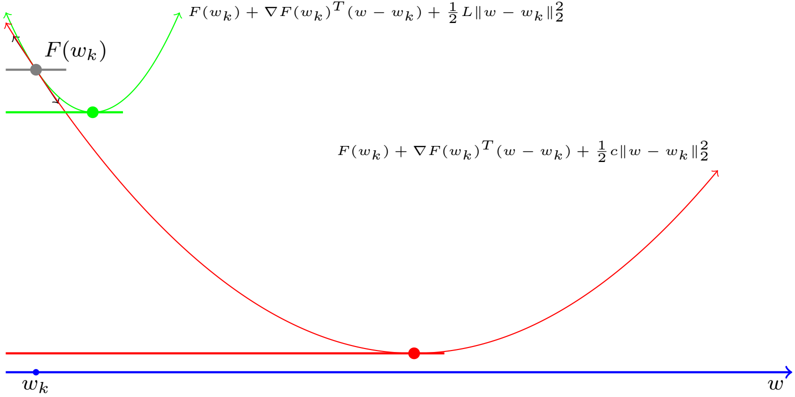


Ideal: Linear convergence of a batch gradient method

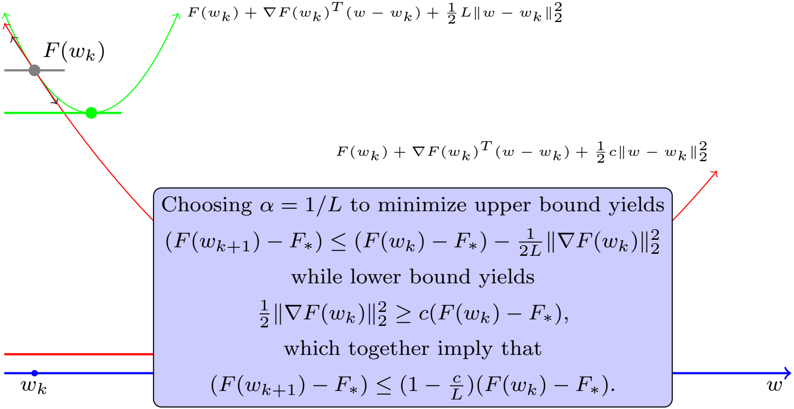


$$F(w_k) + \nabla F(w_k)^T(w - w_k) + \frac{1}{2}L\|w - w_k\|_2^2$$

Ideal: Linear convergence of a batch gradient method


$F(w_k) + \nabla F(w_k)^T(w - w_k) + \frac{1}{2}L\|w - w_k\|_2^2$
$F(w_k)$
$F(w_k) + \nabla F(w_k)^T(w - w_k) + \frac{1}{2}c\|w - w_k\|_2^2$
w_k
w

Ideal: Linear convergence of a batch gradient method


$F(w_k) + \nabla F(w_k)^T(w - w_k) + \frac{1}{2}L\|w - w_k\|_2^2$
$F(w_k)$
$F(w_k) + \nabla F(w_k)^T(w - w_k) + \frac{1}{2}c\|w - w_k\|_2^2$
w_k
w

Choosing $\alpha = 1/L$ to minimize upper bound yields

$$(F(w_{k+1}) - F_*) \leq (F(w_k) - F_*) - \tfrac{1}{2L}\|\nabla F(w_k)\|_2^2$$

while lower bound yields

$$\tfrac{1}{2}\|\nabla F(w_k)\|_2^2 \geq c(F(w_k) - F_*),$$

which together imply that

$$(F(w_{k+1}) - F_*) \leq (1 - \tfrac{c}{L})(F(w_k) - F_*).$$

Illustration

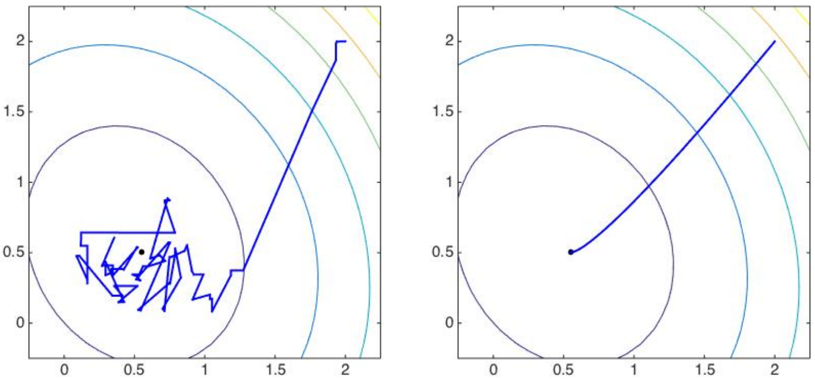


Figure: SG run with a fixed stepsize (left) vs. batch gradient with fixed stepsize (right)

Idea #1: Dynamic sampling

We have seen

- fast initial improvement by SG
- long-term linear rate achieved by batch gradient

$\Longrightarrow$ accumulate increasingly accurate gradient information during optimization.

Idea #1: Dynamic sampling

We have seen

- fast initial improvement by SG
- long-term linear rate achieved by batch gradient

$\Longrightarrow$ accumulate increasingly accurate gradient information during optimization.

But at what rate?

- too slow: won't achieve linear convergence
- too fast: loss of optimal work complexity

Geometric decrease

Correct balance achieved by decreasing noise at a geometric rate.

Theorem 3

Suppose Assumption $\langle L/c \rangle$ holds and that

$$\mathbb{V}_{\xi_k}[g(w_k,\xi_k)] \leq M\zeta^{k-1} \quad \textit{for some} \quad M \geq 0 \quad \textit{and} \quad \zeta \in (0,1).$$

Then, the SG method with a fixed stepsize $\alpha = 1/L$ *yields*

$$\mathbb{E}[F(w_k) - F_*] \leq \omega\rho^{k-1},$$

where

$$\omega := \max\left\{\frac{M}{c}, F(w_1) - F_*\right\}$$

$$\textit{and} \quad \rho := \max\left\{1 - \frac{c}{2L}, \zeta\right\} < 1.$$

Effectively ties rate of noise reduction with convergence rate of optimization.

Geometric decrease

Proof.

The now familiar inequality

$$\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\alpha \|\nabla F(w_k)\|_2^2 + \tfrac{1}{2}\alpha^2 L \mathbb{E}_{\xi_k}[\|g(w_k, \xi_k)\|_2^2],$$

strong convexity, and the stepsize choice lead to

$$\mathbb{E}[F(w_{k+1}) - F_*] \leq \left(1 - \frac{c}{L}\right) \mathbb{E}[F(w_k) - F_*] + \frac{M}{2L}\zeta^{k-1}.$$

- Exactly as for batch gradient (in expectation) except for the last term.
- An inductive argument completes the proof.

Practical geometric decrease (unlimited samples)

How can geometric decrease of the variance be achieved in practice?

$$g_k := \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} \nabla f(w_k; \xi_{k,i}) \quad \text{with} \quad |\mathcal{S}_k| = \lceil \tau^{k-1} \rceil \quad \text{for} \quad \tau > 1,$$

since, for all $i \in \mathcal{S}_k$,

$$\mathbb{V}_{\xi_k}[g_k] \leq \frac{\mathbb{V}_{\xi_k}[\nabla f(w_k; \xi_{k,i})]}{|\mathcal{S}_k|} \leq M(\lceil \tau \rceil)^{k-1}.$$

Practical geometric decrease (unlimited samples)

How can geometric decrease of the variance be achieved in practice?

$$g_k := \frac{1}{|\mathcal{S}_k|} \sum_{i \in \mathcal{S}_k} \nabla f(w_k; \xi_{k,i}) \ \text{ with } \ |\mathcal{S}_k| = \lceil \tau^{k-1} \rceil \ \text{ for } \ \tau > 1,$$

since, for all $i \in \mathcal{S}_k$,

$$\mathbb{V}_{\xi_k}[g_k] \leq \frac{\mathbb{V}_{\xi_k}[\nabla f(w_k; \xi_{k,i})]}{|\mathcal{S}_k|} \leq M(\lceil \tau \rceil)^{k-1}.$$

But is it too fast? What about work complexity?

$$\text{same as SG as long as } \ \tau \in \left(1, (1 - \frac{c}{2L})^{-1}\right].$$

Illustration

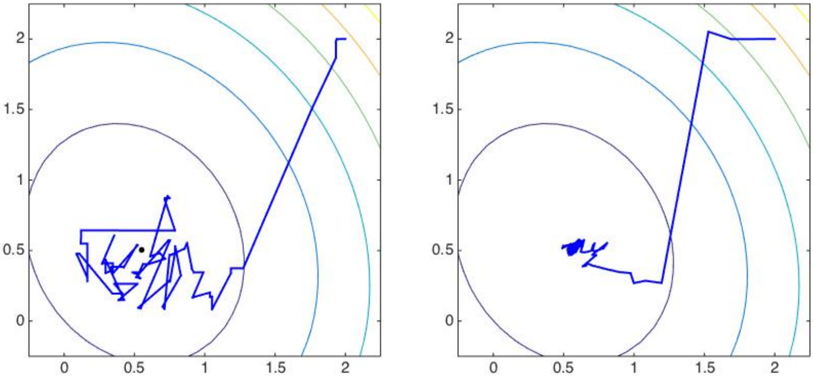


Figure: SG run with a fixed stepsize (left) vs. dynamic SG with fixed stepsize (right)

Additional considerations

In practice, choosing τ is a challenge.

- What about an adaptive technique?
- Guarantee descent in expectation
- Methods exist, but need geometric sample size increase as backup

Idea #2: Gradient aggregation

"I'm minimizing a finite sum and am willing to store previous gradient(s)."

$$F(w) = R_n(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w).$$

Idea: reuse and/or revise previous gradient information in storage.

- SVRG: store full gradient, correct sequence of steps based on perceived bias
- SAGA: store *elements* of full gradient, revise as optimization proceeds

Stochastic variance reduced gradient (SVRG) method

At $w_k =: w_{k,1}$, compute a batch gradient:

$$\underbrace{\begin{array}{|c|c|c|c|c|}\hline \nabla f_1(w_k) & \nabla f_2(w_k) & \nabla f_3(w_k) & \nabla f_4(w_k) & \nabla f_5(w_k) \\ \hline \end{array}}_{g_{k,1} \leftarrow \nabla F(w_k)}$$

then step

$$w_{k,2} \leftarrow w_{k,1} - \alpha g_{k,1}$$

Stochastic variance reduced gradient (SVRG) method

Now, iteratively, choose an index *randomly* and correct bias:

$$\underbrace{\begin{array}{|c|c|c|c|c|}\hline \nabla f_1(w_k) & \nabla f_2(w_k) & \nabla f_3(w_k) & \nabla f_4(w_{k,2}) & \nabla f_5(w_k) \\ \hline\end{array}}_{g_{k,2} \leftarrow \nabla F(w_k) - \nabla f_4(w_k) + \nabla f_4(w_{k,2})}$$

then step

$$w_{k,3} \leftarrow w_{k,2} - \alpha g_{k,2}$$

Stochastic variance reduced gradient (SVRG) method

Now, iteratively, choose an index *randomly* and correct bias:

$$\underbrace{\begin{array}{|c|c|c|c|c|}\hline \nabla f_1(w_k) & \nabla f_2(w_{k,3}) & \nabla f_3(w_k) & \nabla f_4(w_k) & \nabla f_5(w_k) \\ \hline \end{array}}_{g_{k,3} \leftarrow \nabla F(w_k) - \nabla f_2(w_k) + \nabla f_2(w_{k,3})}$$

then step

$$w_{k,4} \leftarrow w_{k,3} - \alpha g_{k,3}$$

Stochastic variance reduced gradient (SVRG) method

Each $g_{k,j}$ is an unbiased estimate of $\nabla F(w_{k,j})$!

Algorithm SVRG

```
Choose an initial iterate $w_1 \in \mathbb{R}^d$, stepsize $\alpha > 0$, and positive integer $m$.
for $k = 1, 2, \ldots$ do
    Compute the batch gradient $\nabla F(w_k)$.
    Initialize $w_{k,1} \leftarrow w_k$.
    for $j = 1, \ldots, m$ do
        Chose $i$ uniformly from $\{1, \ldots, n\}$.
        Set $g_{k,j} \leftarrow \nabla f_i(w_{k,j}) - (\nabla f_i(w_k) - \nabla F(w_k))$.
        Set $w_{k,j+1} \leftarrow w_{k,j} - \alpha g_{k,j}$.
    end for
    Option $(a)$: Set $w_{k+1} = \tilde{w}_{m+1}$
    Option $(b)$: Set $w_{k+1} = \frac{1}{m} \sum_{j=1}^{m} \tilde{w}_{j+1}$
    Option $(c)$: Choose $j$ uniformly from $\{1, \ldots, m\}$ and set $w_{k+1} = \tilde{w}_{j+1}$.
end for
```

Under Assumption $\langle L/c \rangle$, options (b) and (c) linearly convergent for certain (α, m)

Stochastic average gradient (SAGA) method

At w_1, compute a batch gradient:

$$\underbrace{\begin{array}{|c|c|c|c|c|}\hline \nabla f_1(w_1) & \nabla f_2(w_1) & \nabla f_3(w_1) & \nabla f_4(w_1) & \nabla f_5(w_1) \\ \hline \end{array}}_{g_1 \leftarrow \nabla F(w_1)}$$

then step

$$w_2 \leftarrow w_1 - \alpha g_1$$

Stochastic average gradient (SAGA) method

Now, iteratively, choose an index *randomly* and revise table entry:

$$\underbrace{\begin{array}{|c|c|c|c|c|}\hline \nabla f_1(w_1) & \nabla f_2(w_1) & \nabla f_3(w_1) & \nabla f_4(w_2) & \nabla f_5(w_1) \\ \hline \end{array}}_{g_2 \leftarrow \text{new entry} - \text{old entry} + \text{average of entries (before replacement)}}$$

then step

$$w_3 \leftarrow w_2 - \alpha g_2$$

Stochastic average gradient (SAGA) method

Now, iteratively, choose an index *randomly* and revise table entry:

$\nabla f_1(w_1)$	$\nabla f_2(w_3)$	$\nabla f_3(w_1)$	$\nabla f_4(w_2)$	$\nabla f_5(w_1)$

$$g_3 \leftarrow \text{new entry} - \text{old entry} + \text{average of entries (before replacement)}$$

then step

$$w_4 \leftarrow w_3 - \alpha g_3$$

Stochastic average gradient (SAGA) method

Each g_k is an unbiased estimate of $\nabla F(w_k)$!

Algorithm SAGA

```
Choose an initial iterate $w_1 \in \mathbb{R}^d$ and stepsize $\alpha > 0$.
for $i = 1, \ldots, n$ do
    Compute $\nabla f_i(w_1)$.
    Store $\nabla f_i(w_{[i]}) \leftarrow \nabla f_i(w_1)$.
end for
for $k = 1, 2, \ldots$ do
    Choose $j$ uniformly in $\{1, \ldots, n\}$.
    Compute $\nabla f_j(w_k)$.
    Set $g_k \leftarrow \nabla f_j(w_k) - \nabla f_j(w_{[j]}) + \frac{1}{n}\sum_{i=1}^n \nabla f_i(w_{[i]})$.
    Store $\nabla f_j(w_{[j]}) \leftarrow \nabla f_j(w_k)$.
    Set $w_{k+1} \leftarrow w_k - \alpha g_k$.
end for
```

Under Assumption $\langle L/c \rangle$, linearly convergent for certain α

- storage of gradient vectors reasonable in some applications
- with access to feature vectors, need only store n scalars

Idea #3: Iterative averaging

Averages of SG iterates are less noisy:

$$w_{k+1} \leftarrow w_k - \alpha_k g(w_k, \xi_k)$$

$$\tilde{w}_{k+1} \leftarrow \frac{1}{k+1} \sum_{j=1}^{k+1} w_j \quad \text{(in practice: running average)}$$

Unfortunately, no better theoretically when $\alpha_k = \mathcal{O}(1/k)$, but

- long steps (say, $\alpha_k = \mathcal{O}(1/\sqrt{k})$) *and* averaging
- lead to a better sublinear rate (like a second-order method?)

See also

- mirror descent
- primal-dual averaging

Idea #3: Iterative averaging

Averages of SG iterates are less noisy:

$$w_{k+1} \leftarrow w_k - \alpha_k g(w_k, \xi_k)$$

$$\tilde{w}_{k+1} \leftarrow \frac{1}{k+1} \sum_{j=1}^{k+1} w_j \quad \text{(in practice: running average)}$$

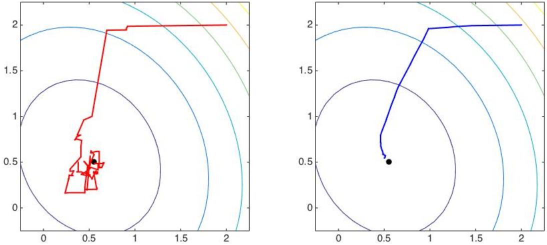


Figure: SG run with $\mathcal{O}(1/\sqrt{k})$ stepsizes (left) vs. sequence of averages (right)

Outline

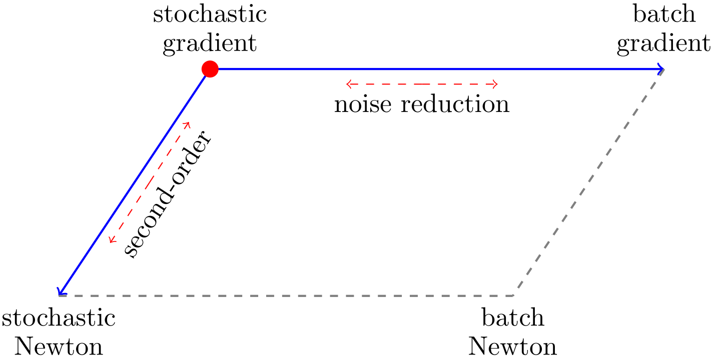
SG

Noise Reduction Methods

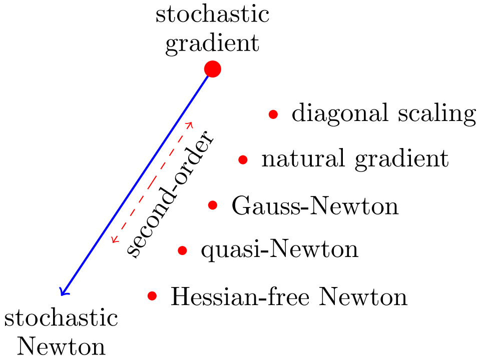
Second-Order Methods

Other Methods

Two-dimensional schematic of methods



2D schematic: Second-order methods



Ideal: Scale invariance

Neither SG nor batch gradient are invariant to linear transformations!

$$\min_{w\in\mathbb{R}^d} F(w) \quad\Longrightarrow\quad w_{k+1} \leftarrow w_k - \alpha_k \nabla F(w_k)$$

$$\min_{\tilde{w}\in\mathbb{R}^d} F(B\tilde{w}) \quad\Longrightarrow\quad \tilde{w}_{k+1} \leftarrow \tilde{w}_k - \alpha_k B\nabla F(B\tilde{w}_k) \quad \text{(for given } B \succ 0\text{)}$$

Ideal: Scale invariance

Neither SG nor batch gradient are invariant to linear transformations!

$$\min_{w\in\mathbb{R}^d} F(w) \quad \Longrightarrow \quad w_{k+1} \leftarrow w_k - \alpha_k \nabla F(w_k)$$

$$\min_{\tilde{w}\in\mathbb{R}^d} F(B\tilde{w}) \quad \Longrightarrow \quad \tilde{w}_{k+1} \leftarrow \tilde{w}_k - \alpha_k B\nabla F(B\tilde{w}_k) \quad \text{(for given } B \succ 0\text{)}$$

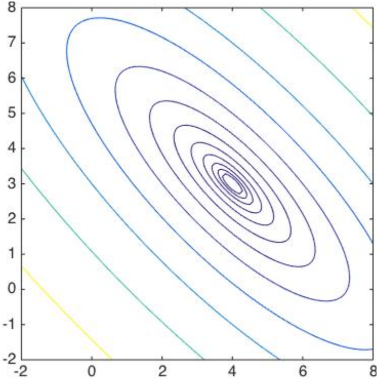
Scaling latter by B and defining $\{w_k\} = \{B\tilde{w}_k\}$ yields

$$w_{k+1} \leftarrow w_k - \alpha_k B^2 \nabla F(w_k)$$

- Algorithm is clearly affected by choice of B
- Surely, some choices may be better than others (in general?)

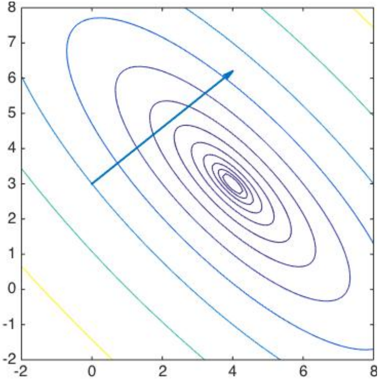
Newton scaling

Consider the function below and suppose that $w_k = (0, 3)$:



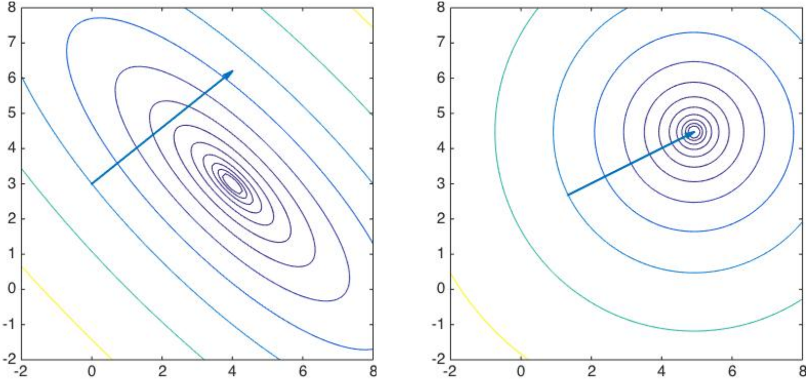
Newton scaling

Batch gradient step $-\alpha_k \nabla F(w_k)$ ignores curvature of the function:



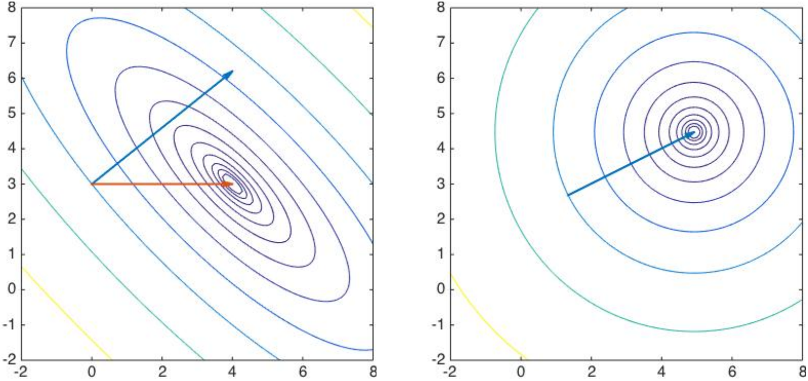
Newton scaling

Newton scaling $(B = (\nabla F(w_k))^{-1/2})$: gradient step moves to the minimizer:



Newton scaling

. . . corresponds to minimizing a quadratic model of F in the original space:



$$w_{k+1} \leftarrow w_k + \alpha_k s_k \quad \text{where} \quad \nabla^2 F(w_k) s_k = -\nabla F(w_k)$$

Deterministic case

What is known about Newton's method for deterministic optimization?

- local rescaling based on inverse Hessian information
- locally quadratically convergent near a strong minimizer
- global convergence rate better than gradient method (*when regularized*)

Deterministic case to stochastic case

What is known about Newton's method for deterministic optimization?

- local rescaling based on inverse Hessian information
- locally quadratically convergent near a strong minimizer
- global convergence rate better than gradient method (*when regularized*)

However, it is way too expensive in our case.

- But all is not lost: scaling is viable.
- Wide variety of scaling techniques improve performance.
- Our convergence theory for SG still holds with B-scaling.
- . . . could hope to remove condition number (L/c) from convergence rate!
- Added costs can be minimial when coupled with noise reduction.

Idea #1: Inexact Hessian-free Newton

Compute Newton-like step

$$\nabla^2 f_{\mathcal{S}_k^H}(w_k) s_k = -\nabla f_{\mathcal{S}_k^g}(w_k)$$

- mini-batch size for Hessian $=: |\mathcal{S}_k^H| < |\mathcal{S}_k^g| :=$ mini-batch size for gradient
- cost for mini-batch gradient: g_{cost}
- use CG and terminate early: max_{cg} iterations
- in CG, cost for each Hessian-vector product: $factor \times g_{cost}$
- choose $max_{cg} \times factor \approx$ small constant so total per-iteration cost:

$$max_{cg} \times factor \times g_{cost} = \mathcal{O}(g_{cost})$$

- convergence guarantees for $|\mathcal{S}_k^H| = |\mathcal{S}_k^g| = n$ are well-known

Idea #2: (Generalized) Gauss-Newton

Classical approach for nonlinear least squares, linearize inside of loss/cost:

$$\begin{aligned} f(w;\xi) &= \tfrac{1}{2}\|h(x_\xi;w) - y_\xi\|_2^2 \\ &\approx \tfrac{1}{2}\|h(x_\xi;w_k) + J_h(w_k;\xi)(w - w_k) - y_\xi\|_2^2 \end{aligned}$$

Leads to Gauss-Newton approximation for second-order terms:

$$G_{\mathcal{S}_k^H}(w_k;\xi_k^H) = \frac{1}{|\mathcal{S}_k^H|} J_h(w_k;\xi_{k,i})^T J_h(w_k;\xi_{k,i})$$

Idea #2: (Generalized) Gauss-Newton

Classical approach for nonlinear least squares, linearize inside of loss/cost:

$$\begin{aligned} f(w;\xi) &= \tfrac{1}{2}\|h(x_\xi;w) - y_\xi\|_2^2 \\ &\approx \tfrac{1}{2}\|h(x_\xi;w_k) + J_h(w_k;\xi)(w-w_k) - y_\xi\|_2^2 \end{aligned}$$

Leads to Gauss-Newton approximation for second-order terms:

$$G_{\mathcal{S}_k^H}(w_k;\xi_k^H) = \frac{1}{|\mathcal{S}_k^H|} J_h(w_k;\xi_{k,i})^T J_h(w_k;\xi_{k,i})$$

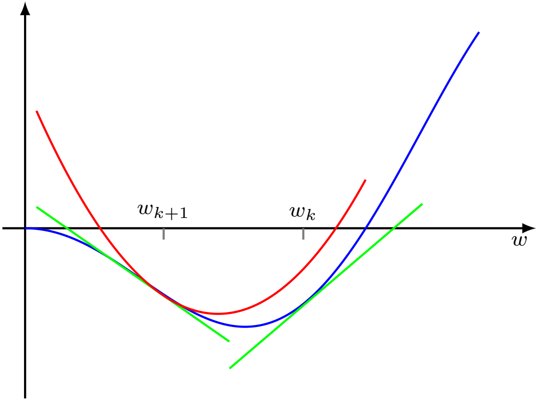
Can be generalized for other (convex) losses:

$$\widetilde{G}_{\mathcal{S}_k^H}(w_k;\xi_k^H) = \frac{1}{|\mathcal{S}_k^H|} J_h(w_k;\xi_{k,i})^T \underbrace{H_\ell(w_k;\xi_{k,i})}_{=\frac{\partial^2 \ell}{\partial h^2}} J_h(w_k;\xi_{k,i})$$

- costs similar as for inexact Newton
- . . . but scaling matrices are always positive (semi)definite
- see also *natural gradient*, invariant to more than just linear transformations

Idea #3: (Limited memory) quasi-Newton

Only *approximate* second-order information with gradient displacements:



Secant equation $H_k v_k = s_k$ to match gradient of F at w_k, where

$$s_k := w_{k+1} - w_k \quad \text{and} \quad v_k := \nabla F(w_{k+1}) - \nabla F(w_k)$$

Deterministic case

Standard update for inverse Hessian ($w_{k+1} \leftarrow w_k - \alpha_k H_k g_k$) is BFGS:

$$H_{k+1} \leftarrow \left(I - \frac{v_k s_k^T}{s_k^T v_k} \right)^T H_k \left(I - \frac{v_k s_k^T}{s_k^T v_k} \right) + \frac{s_k s_k^T}{s_k^T v_k}$$

What is known about quasi-Newton methods for deterministic optimization?

- local rescaling based on iterate/gradient displacements
- strongly convex function $\implies$ positive definite (p.d.) matrices
- only first-order derivatives, no linear system solves
- locally superlinearly convergent near a strong minimizer

Deterministic case to stochastic case

Standard update for inverse Hessian ($w_{k+1} \leftarrow w_k - \alpha_k H_k g_k$) is BFGS:

$$H_{k+1} \leftarrow \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right)^T H_k \left(I - \frac{v_k s_k^T}{s_k^T v_k}\right) + \frac{s_k s_k^T}{s_k^T v_k}$$

What is known about quasi-Newton methods for deterministic optimization?

- local rescaling based on iterate/gradient displacements
- strongly convex function $\Longrightarrow$ positive definite (p.d.) matrices
- only first-order derivatives, no linear system solves
- locally superlinearly convergent near a strong minimizer

Extended to stochastic case? How?

- Noisy gradient estimates $\Longrightarrow$ challenge to maintain p.d.
- Correlation between gradient and Hessian estimates
- Overwriting updates $\Longrightarrow$ poor scaling that plagues!

Proposed methods

- gradient displacements using same sample:

$$v_k := \nabla f_{\mathcal{S}_k}(w_{k+1}) - \nabla f_{\mathcal{S}_k}(w_k)$$

  (requires two stochastic gradients per iteration)

- gradient displacement replaced by action on subsampled Hessian:

$$v_k := \nabla^2 f_{\mathcal{S}_k^H}(w_k)(w_{k+1} - w_k)$$

- decouple iteration and Hessian update to amortize added cost
- limited memory approximations (e.g., L-BFGS) with per-iteration cost $4md$

Idea #4: Diagonal scaling

Restrict added costs through only diagonal scaling:

$$w_{k+1} \leftarrow w_k - \alpha_k D_k g_k$$

Ideas:

- $D_k^{-1} \approx \text{diag}(\text{Hessian (approximation)})$
- $D_k^{-1} \approx \text{diag}(\text{Gauss-Newton approximation})$
- $D_k^{-1} \approx$ running average/sum of gradient components

Last approach can be motivated by minimizing regret.

Outline

SG

Noise Reduction Methods

Second-Order Methods

Other Methods

Plenty of ideas not covered here!

- gradient methods with momentum
- gradient methods with acceleration
- coordinate descent/ascent in the primal/dual
- proximal gradient/Newton for regularized problems
- alternating direction methods
- expectation-maximization
- . . .