Optimization Methods for Large-Scale Machine Learning

V. Estellers

WS 2017

What this lectures are about

The optimization problems that result from training a large-scale machine learning model have characteristics that make the stochastic gradient (SG) method more effective than conventional gradient-based nonlinear optimization techniques.

1. Characteristic of optimization of large-scale machine learning models
2. Stochastic gradient algorithm
3. Analysys of SG algorithm
4. Improved SG convergence with noise-reduction techniques
5. Improved SG convergence with second-order derivatives

References

The lectures are organized following:
Bottou, L., Curtis, F. E., and Nocedal, J. *Optimization Methods for Large-Scale Machine Learning.* 2016.http://arxiv.org/abs/1606.04838
We will also cover some material from:

1. Gower, R. M., Roux, N. Le, and Bach, F. *Tracking the gradients using the Hessian: A new look at variance reducing stochastic methods.*, 2017.
2. Roux, N. Le, Schmidt, M., and Bach, F. *A Stochastic Gradient Method with an Exponential Convergence Rate for Strongly-Convex Optimization with Finite Training Sets.* 2012.

Optimization Problems in Machine Learning

We illustrate how optimization problems arise in machine learning and what makes them challenging with two case studies:

1. linear regressor with bag-of-words features for text classification
2. open-ended deep neural network for speech and image recognition.

Both problems have some common characteristics:

- Large-scale: models described by a large number of parameters.
- Stochastic: models designed to make decisions on unseen data..

They differ in the optimization problem: (1) convex, (2) nonconvex.

Text Classification via Convex Optimization

Text classification: assigning a predefined class to a natural language text based on its contents. For example, determine if a text discusses politics.

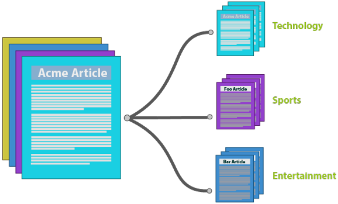


Fig.: http://blog.thedigitalgroup.com/rajendras/2015

Given a set of examples $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where

- feature vector x_i of a text document (e.g., the words it includes).
- scalar label y_i indicating if the document belongs $(y_i = 1)$ or not $(y_i = 1)$ to a particular class.

Construct a classifier that predicts the class of an unseen text.

First Solution: Minimizing Empirical Risk

Design a prediction function h s.t. $h(x)$ predicts the text document.

- Performance measure: how often $h(x_i)$ differs from the prediction y_i.
- Search h that minimizes the frequency of observed misclassifications:

$$R_n(h) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}[h(x_i) \neq y_i], \quad \text{where} \quad \mathbb{1}[A] = \begin{cases} 1 & \text{if } A \text{ is true,} \\ 0 & \text{otherwise.} \end{cases} \tag{1.1}$$

R_n is the empirical risk of misclassification.

Minimizing Empirical Risk is Not Enough

Rote memorization with

$$h^{\mathrm{rote}}(x) = \begin{cases} y_i & \text{if } x = x_i \text{ for some } i \in \{1, \dots, n\}, \\ \pm 1 & \text{(arbitrarily) otherwise.} \end{cases} \tag{1.2}$$

minimizes the empirical risk but has no guarantees on unseen documents.

The prediction function should generalizes the concepts learned from the examples. To this goal, we choose

- parametric functions satisfying certain smoothness conditions
- use cross-validation to choosing between classes of prediction functions

Minimizing Expected Risk with Cross-validation

Cross-validation minimizes the expected risk by splitting examples into:

- training set to optimize the parameters of h by minimizing R_n. The selects a candidate for each class of parametric functions $h_1, \ldots, h_k$
- validation set to estimate the performance of $h_1, \ldots, h_k$. This selects the best candidate h^*
- testing set to estimate the performance of h^*

Cross-validation has shown the success of **bag-of-words** approach for text classification.

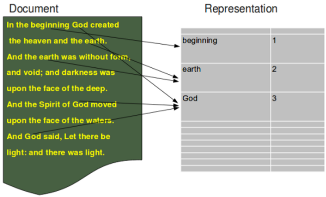
Linear Regression with Bag-of-Words Features

Bag-of-Words features:

- represents a text document by a feature vector $x \in \mathbb{R}^d$, where each component measures the appearance of a specific word.
- very sparse vectors of high-dimensionality.

Affine prediction function classifies the documents:

$$h(x; w, \tau) = w^T x - \tau \tag{1.3}$$



In the beginning God created
the heaven and the earth.
And the earth was without form,
and void; and darkness was
upon the face of the deep.
And the Spirit of God moved
upon the face of the waters.
And God said, Let there be
light: and there was light.

beginning	1
earth	2
God	3

Fig.: https://www.python-kurs.eu/text_klassifikation_python.php

Optimization of the Model

Finding w, τ that minimize the empirical risk of misclassification

$$R_n(h) = \frac{1}{n} \sum_{i=1}^{n} \operatorname{sign}(-h(x_i; w, \tau) \cdot y_i) \tag{1.4}$$

is difficult because the sign is discontinuous, takes discrete values, and results in a combinatorial problem. For this reason, we approximate it by a continuous loss function that we can minimize effective like

$$\ell(h, y) = \log(1 + \exp(-h(x_i)y)). \tag{1.5}$$

Classes of prediction functions h_λ are determined by a regularization term

$$\min_{(w,\tau) \in \mathbb{R}^{d+1}} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-h(x_i)y_i)) + \frac{\lambda}{2} \|w\|^2. \tag{1.6}$$

Optimizing the model parameters with various $\lambda_1, \ldots, \lambda_k$ on the training set gives the candidate solution $h_{\lambda_1}, \ldots, h_{\lambda_k}$. The final solution is the candidate with best performance on the validation set.

Perceptual Tasks via Deep Networks

Deep/Convolutional neural networks have recently achieved spectacular success on perceptual problems such as speech and image recognition.

They are essentially the same types of networks from the 90s, but their successes is now possible due to the availability of larger datasets and computational resources.

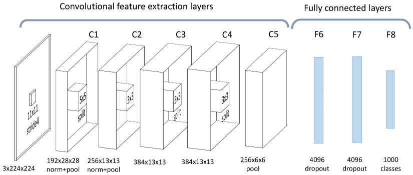


Fig.: Architecture for image recognition. The 2012 ILSVRC winner consists of eight layers: each layer performs a linear transformation followed by nonlinear transformations.

Neural Networks

DNN/CNNs construct a prediction function h whose value is computed by applying successive transformations to a given input vector $x_i \in \mathbb{R}^{d_0}$. These transformations are made in layers.

$$x_i^{(j)} = s(W_j \, x_i^{(j-1)} + b_j) \in \mathbb{R}^{d_j}, \tag{1.7}$$

where $x_i^{(0)} = x_i$ and

- $x_i^{(j)}$ is the input vector to layer j
- j-th layer parameters: matrix $W_j \in \mathbb{R}^{d_j \times d_{j-1}}$ and vector $b_j \in \mathbb{R}^{d_j}$
- s componentwise nonlinear activation/pooling function.

Deep Neural Networks

Neural Networks use simple activation functions, like the sigmoid or the rectified linear unit (ReLU)

$$s(x) = 1/(1 + \exp(-x)) \qquad\qquad s(x) = \max\{0, x\}.$$

CNNs are networks where layers have

- circulant matrices W_j, s.t. $W_j x_i^{(j1)}$ is an image convolution.
- activation functions rectify, normalize, or subsample images.

The output vector $x_i^{(J)}$ is the prediction function value $h(x_i; w)$, where $w = \{(W_1, b_1), \ldots, (W_J, b_J)\}$ collects the parameters of all the layers.

Optimization of Deep Neural Networks

The optimization of DNN/CNN over the training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ with a loss function ℓ define the problem

$$\min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i; w), y_i). \tag{1.8}$$

This problem is nonconvex. Finding a global optimum is intractable and we look for approximate solutions with gradient-based methods.

The gradient of the objective function of (1.8) can be computed efficiently by the chain rule (back propagation).

Convolutional neural networks

The training process of DNNs and CNNs requires extreme care to overcome the difficulties of large, nonlinear and nonconvex problems:

1. initialize the optimization process with a good starting point
2. monitor its progress to correct conditioning issues as they appear (vanishing gradients).

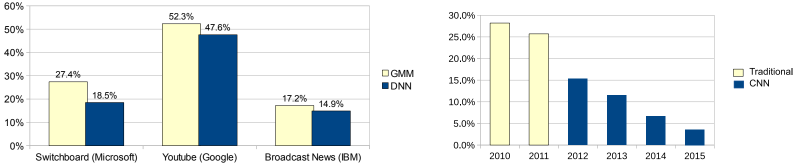


Fig.: Left:Word error rates reported by three different research groups on three standard speech recognition benchmarks. Right: Historical *top5* error rate of the annual winner of the ImageNet image classification challenge.

Formulation of a Supervised Classification Problem

Classification: choose a prediction function from an input space $\mathscr{X}$ to an output space $\mathscr{Y}$

$$h : \mathscr{X} \to \mathscr{Y}$$

s.t., given $x \in \mathscr{X}$, $h(x)$ offers an accurate prediction about the output y.

Supervised: h that generalizes the properties meaningful to determine y from x that can be learned from input-output examples $\{(x_i, y_i)\}_{i=1}^{n}$.

Problem: avoid rote memorization by choosing a prediction function h that minimizes a risk measure over a family of prediction functions $\mathscr{H}$.

Expected Risk instead of Empirical Risk

Let $\{(x_i, y_i)\}_{i=1}^{n}$ be samples from a joint probability distribution function $P(x, y)$. Rather than finding h that minimizes the empirical risk

$$R_n(h) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[h(x_i) = y_1] \qquad \mathbb{1}[A] = \begin{cases} 1 & \text{if A is true} \\ 0 & \text{otherwise} \end{cases}, \tag{1.9}$$

we find h that minimizes the expected misclassification risk over all possible inputs, i.e., an h that minimizes

$$R(h) = \mathbb{P}[h(x) \neq y] = \mathbb{E}[\mathbb{1}[h(x) \neq y]], \tag{1.10}$$

Stochastic problem (objective is an expectation) is substituted by a surrogate problem constructed from $\{(x_i, y_i)\}_{i=1}^{n}$ as we do not know P.

Choice of Prediction Function Family

We choose the family of functions $\mathscr{H}$ with three goals in mind:

1. $\mathscr{H}$ should contain functions that achieve a low empirical risk to avoid underfitting the data. $\Rightarrow$ select a rich family of functions
2. $\mathscr{H}$ should be selected to make the optimization problem solvable
3. $R(h) - R_n(h)$ should be small over all $h \in \mathscr{H}$. This gap might increase when $\mathscr{H}$ becomes too rich and overfits the training data.

Gap Between Expected and Empirical Risk

When the expected risk represents a misclassification probability, with probability at least $1 - \eta$,

$$\sup_{h \in \mathscr{H}} |R(h) - R_n(h)| \leq \mathscr{O}\left(\sqrt{\frac{1}{2n}\log\left(\frac{2}{\eta}\right) + \frac{d_{\mathscr{H}}}{n}\log\left(\frac{n}{d_{\mathscr{H}}}\right)}\right). \quad (1.11)$$

- $d_{\mathscr{H}}$: Vapnik-Chervonenkis dimension measures the capacity of $\mathscr{H}$
- fixed $d_{\mathscr{H}}$, the gap decreases by increasing number of examples (n).
- fixed n, the gap can widen for larger $d_{\mathscr{H}}$ (richer function families).

Bound (1.11) is not used in practice because it is easier to estimate the gap with cross-validation than calculate the VC dimension of $\mathscr{H}$.

Structural Risk Minimization

Structural risk minimization: technique for choosing a prediction function. Consider a nested families of function parametrized by function Ω

$$H_C = \{h \in \mathscr{H} \ : \ \Omega(h) \leq C\} \Rightarrow H_C \subset H_D \text{ for } C < D$$

1. Increasing C reduces the R_n because it enlarges the family of functions we can optimize over.
2. For large C, the $R_n - R$ increases because the prediction function overfits the training data.

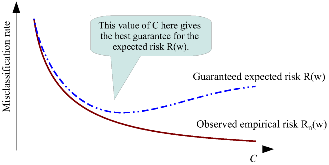


Fig.: The optimal empirical risk decreases when C increases. $|R(w) - R_n(w)|$ is bounded above by a quantity that increases with C. The value of C that offers the best guarantee on the expected risk increases with n.

In the following, we consider the problem...

Assume that the prediction function h is parameterized by a real vector $w \in \mathbb{R}^d$. This vector defines our optimization variable and the family of prediction functions

$$\mathscr{H} = \{h(\cdot; w) \colon\ \mathbb{R}^d \times \mapsto \mathbb{R}^{d_y} \mid w \in \mathbb{R}^d\}.$$

Given a loss function $\ell \colon \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \mapsto \mathbb{R}$ that measures the loss associated with the prediction $h(x; w)$ when the true label is y with $\ell(h(x; w), y)$, we define:

- ξ: random variable that represents a sample or a set of samples $\{(x_i, y_i)\}_{i \in S}$.
- $f(w; \xi) = \ell(h(w; \xi), y)$: the loss incurred for a given (w, ξ)

Expected and Empirical Risk

Let $P(x, y)$ be the probability distribution between inputs and outputs, we define **expected risk** by

$$R(w) = \int_{\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}} \ell(h(x; w), y) \mathrm{d}P(x, y) = \mathbb{E}[\ell(h(x; w), y)] = \mathbb{E}[f(w; \xi)]$$

To minimize the expected risk, we need complete information about P. As this is not possible, we minimize the **empirical risk**

$$R_n(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i; w), y_i) = \frac{1}{n} \sum_{i=1}^{n} f(w; \xi[i])$$

that estimates the expected risk (in supervised classification) from n independently drawn input-output samples $\{\xi[i]\}_{i=1}^{n} = \{(x_i, y_i)\}_{i=1}^{n}$.

Stochastic Optimization for Empirical Risk Minimization

The stochastic gradient method (SG) minimizes the empirical risk R_n with the sequence:

$$w_{k+1} = w_k - \alpha_k \nabla f_{i_k}(w_k) \quad \forall k \in \mathbb{N}, \tag{2.12}$$

where w_1 is given, α_k is a positive stepsize, and i_k is chosen randomly. Characteristics:

1. Cheap iterations that only computate one gradient $\nabla f_{i_k}(w_k)$
2. The sequence is not determined uniquely by R_n, w_1, and stepsizes, but depends also on the random sequence $\{i_k\}$.
3. $-\nabla f_{i_k}(w_k)$ might not be a descent direction from w_k.

Batch Optimization for Empirical Risk Minimization

A batch approach minimizes the Empirical Risk directly. The simplest steepest descent or gradient method defines the sequence:

$$w_{k+1} = w_k - \alpha_k \nabla R_n(w_k) = w_k - \alpha_k \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(w_k) \quad \forall k \in \mathbb{N}, \quad (2.13)$$

Characteristics:

1. Computing $\alpha_k \nabla R_n(w_k)$ is more expensive than $\alpha_k \nabla f_{i_k}(w_k)$ in SG.
2. By iterating over all samples, batch methods compute better steps.
3. It can use (quasi) Newton methods to speed up optimization of R_n.
4. The sum structure of R_n allows parallel or distributed updates.

Typical Performance of Stochastic and Batch Methods

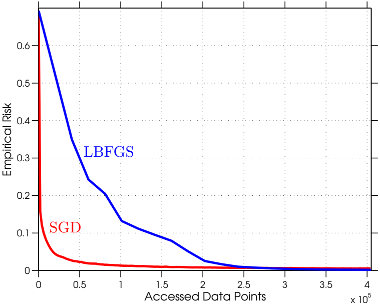


Fig.: Empirical risk R_n as a function of the number of accessed data points (ADP) for a batch L-BFGS method and the SG method on a binary classification problem with a logistic loss objective and the RCV1 dataset.

Stochastic vs Batch Methods

SG is used in machine learning when cannot afford to iterate over all samples to compute the next iterate.

SG uses the samples more efficient than a batch method. Intuitively:

- Consider a training set S with ten copies of a set S_s.
- In a batch approach, the iterations that use S as training set are ten times more expensive than iteration that only use one copy of S_s.
- In the SG method, the iterations using S and S_s as training sets cost the same.
- In reality, a training set does not consist of exact duplicates of sample data, but it has enough redundancy to make using all of the samples in every iteration inefficient.

Next week: Stochastic vs Batch Methods

Let R_n^* be the minimal value of R_n, then if R_n is strongly convex

- the error of a batch gradient method satisfies

$$|R_n(w_k) - R_n^*| \leq \mathcal{O}(\rho^k), \quad \rho \in (0,1).$$

  The number of iterations where the training error is above ϵ is proportional to $\log(\frac{1}{\epsilon})$, and the cost of ϵ-optimality is $\mathcal{O}(n \log(\frac{1}{\epsilon}))$.

- the SG error for i_k is drawn uniformly from $\{1, \ldots, n\}$ is

$$\mathbb{E}[|Rn(w_k) - R_n^*|] = \mathcal{O}(\frac{1}{k}) \tag{2.14}$$

  As it does not depend on n, the cost of ϵ-optimality is $\mathcal{O}(\frac{1}{\epsilon})$.

The SG cost $\mathcal{O}(\frac{1}{\epsilon})$ is smaller than the batch cost $\mathcal{O}(n \log(\frac{1}{\epsilon}))$ if n is large.

Next week: Stochastic vs Batch Methods

SG avoids overfitting in the sense that the minimizer of the empirical risk found by SG has some minimization guarantees on the expected risk.

By applying the SG iteration with $\nabla f(w_k; x_{i_k})$ replaced by $\nabla f(w_k; \xi_k)$ with each ξ_k drawn independently according to the distribution P,

$$\mathbb{E}[|R(w_k) - R^*|] = \mathcal{O}(\frac{1}{k}). \tag{2.15}$$

This is again a sublinear rate, but on the expected risk.

Other Methods

A mini-batch approach chooses a subset of samples $S_k \subset \{1, \dots, n\}$ randomly in each iteration to improve the gradient estimate as follows:

$$w_{k+1} = w_k - \frac{\alpha_k}{|S_k|} \sum_{i \in S_k} \nabla f_i(w_k) \quad \forall k \in \mathbb{N}, \tag{2.16}$$

This allows some degree of parallelization and reduces the variance of the stochastic gradient estimates.

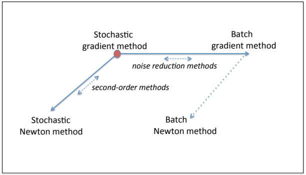


Fig.: Schematic of a two-dimensional spectrum of optimization methods for machine learning. The horizontal axis represents methods designed to control stochastic noise; the second axis, methods that deal with ill conditioning.

Optimization for Supervised Learning

Given a set of examples $(x_1, y_1), \dots, (x_n, y_n)$

- each example $\xi = (x, y)$ is a pair of an input x and a scalar output y.
- loss $\ell(\hat{y}, y)$ measures the cost of predicting $\hat{y}$ when the answer is y
- family $\mathscr{H}$ of functions $h(\cdot; w)$ parametrized by a weight vector w.

We seek $h \in \mathscr{H}$ that minimizes the loss $f(\xi; w) = \ell(h(x; w), y)$.
Although we would like to average over the unknown distribution $P(x, y)$

$$f(w) = R(w) = \mathbb{E}[\ell(h(x; w), y)] = \int \ell(h(x; w), y) \mathrm{d}P(x, y)$$

we must settle for computing the average over the samples

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

Statistical learning theory (Vapnik and Chervonenkis, 1971) justifies minimizing R_n instead of R when $\mathscr{H}$ is sufficiently restrictive.

Stochastic Gradient Method

The objective function $F\colon \mathbb{R}^d \mapsto \mathbb{R}$ can be the expected or empirical risk:

$$F(w) = \mathbb{E}[f(w,\xi)] \qquad \text{or} \qquad F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w).$$

The analysis applies to both objectives. The only difference is the way that the stochastic gradient estimates are chosen.

1. $F = R_n$: pick samples uniformly from a finite training set with replacement (sample discrete distribution with equal weights for every sample).
2. $F = R$: pick samples in each iteration according to distribution P (online or large-scale setting).

Stochastic Gradient Method

```
Choose an initial iterate $w_1$
for k=1,2,... 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 $w_{k+1} = w_k - \alpha_k g(w_k, \xi_k)$
end for
```

The algorithm requires three computational tools:

1. mechanism for generating a realization of a random variable ξ_k, where $\{\xi_k\}$ is a sequence of jointly independent random variables.
2. mechanism for computing stochastic vector $g(w_k, \xi_k) \in \mathbb{R}^d$
3. mechanism for computing a scalar stepsize $\alpha_k > 0$

General version of Stochastic Gradient Method

This SG algorithm generalizes many stochastic gradient-based algorithms:

$$g(w_k, \xi_k) = \begin{cases} \nabla f(w_k, \xi_k) & \text{simple or base SG} \\ \frac{1}{n_k} \sum_{i=1}^{n_k} \nabla f(w_k, \xi_{k,i}) & \text{mini-batch SG} \\ H_k \frac{1}{n_k} \sum_{i=1}^{n_k} \nabla f(w_k, \xi_{k,i}) & \text{2nd-order SG} \end{cases}$$

flexibility choosing mini-batch size n_k and symmetric positive definite H_k.

Prove convergence of SG with two assumptions:

1. smoothness of the objective function
2. bounded 1st and 2nd moments of stochastic vectors $\{g(w_k, \xi_k)\}$

If the objective is strongly convex SG converges to the minimum, otherwise to a stationary point.

Assumption 1: L-smooth function

The objective function $F\colon \mathbb{R}^d \mapsto \mathbb{R}$ is continuously differentiable and the gradient function of F, $\nabla F\colon \mathbb{R}^d \mapsto \mathbb{R}^d$, is Lipschitz continuous with Lipschitz constant L, that is

$$\|\nabla F(w) - \nabla F(\bar{w})\| \leq L\|w - \bar{w}\| \ \ \forall w, \bar{w} \in \mathbb{R}^d.$$

This assumption ensures that the gradient of F does not change arbitrarily quickly with respect to the parameter vector and can be used to estimate how far to move (SG stepsize) to decrease F.

An important consequence of F being L-smooth is that

$$F(w) \leq F(\bar{w}) + \nabla F(\bar{w})^T(w - \bar{w}) + \frac{1}{2}L\|w - \bar{w}\|^2 \ \ \forall w, \bar{w} \in \mathbb{R}^d.$$

First Lemma

Lemma

If F is an L-smooth function and $\mathbb{E}_{\xi_k}[\cdot]$ denotes the expected value taken w.r.t the distribution of the random variable ξ_k, the iterates of SG satisfy:

$$\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\alpha_k \underbrace{\nabla F(w_k)^T \mathbb{E}_{\xi_k}[g(w_k, \xi_k)]}_{\substack{\textit{expected directional derivative} \\ \textit{of } F \textit{ along direction } g(w_k,\xi_k)}} + \frac{\alpha_k^2 L}{2} \underbrace{\mathbb{E}_{\xi_k}[\|g(w_k, \xi_k)\|^2]}_{\textit{second moment } g(w_k,\xi_k)}$$

If $g(w_k, \xi_k)$ is an unbiased estimate of $\nabla F(w_k)$, we have

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

Proof

As F is L-smooth and the SG iterates $w_{k+1} = w_k - \alpha_k g(w_k, \xi_k)$, we have

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

Taking expectations on both sides w.r.t the distribution of ξ_k, and noting that only w_{k+1} and $g(w_k, \xi_k)$ depend on ξ_k, we obtain the desired bound

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

Assumption 2: Bounds on First and Second Moments

The objective function and the SG Algorithm satisfy:

1. $\{w_k\}$ are in an open set where F is bounded below by scalar F_{inf}.
2. In expectation, $-g(w_k, \xi_k)$ is a direction of sufficient descent with a norm comparable to the norm of the gradient. There is $\mu_G \geq \mu > 0$

$$\frac{1}{\mu} \nabla F(w_k)^T \mathbb{E}[g(w_k, \xi_k)] \geq \|\nabla F(w_k)\|^2$$

$$\|\nabla F(w_k)\| \geq \frac{1}{\mu_G} \|\mathbb{E}[g(w_k, \xi_k)]\|$$

3. There exist scalars $M, M_V \geq 0$ such that, for all $k \in \mathbb{N}$

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

Assumption 2: In practice

Point 1, just means that there is no *trivial* solution $\min_w \ F(w) = -\infty$.

Point 2 holds if $g(w_k, \xi_k)$ is an unbiased estimate of $\nabla F(w_k)$ multiplied by positive definite H_k with eigenvalues in a fixed interval.

Points 2 and 3 are combined into a single inequality with $M_G \geq \mu^2 \geq 0$

$$\begin{aligned}\mathbb{E}_{\xi_k}[\|g(w_k, \xi_k)\|^2] &= \mathbb{V}_{\xi_k}[g(w_k, \xi_k)] + \|\mathbb{E}_{\xi_k}[g(w_k, \xi_k)]\|^2 \\ &\leq M + M_V\|\nabla F(w_k)\|^2 + \mu_G^2\|\nabla F(w_k)\|^2 \\ &\leq M + M_G\|\nabla F(w_k)\|^2\end{aligned}$$

Lemma 2

Lemma

If F is L-smooth and Assumption 2 holds, the SG iterates satisfy

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

$$\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\underbrace{(\mu - \frac{1}{2}\alpha_k L M_G)\alpha_k \|\nabla F(w_k)\|^2 + \frac{1}{2}\alpha_k^2 L M}_{\textit{deterministic}}.$$

In English, regardless of how the method arrived at w_k, the optimization continues in a Markovian manner: w_{k+1} that depends only on the iterate w_k, the seed ξ_k, and the stepsize α_k and not on any past iterates.

Lemma 2, Proof

Let us prove the first inequality. As F is L-smooth, Lemma 1 states

$$\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) \leq \underbrace{-\alpha_k \nabla F(w_k)^T \mathbb{E}_{\xi_k}[g(w_k, \xi_k)]}_{\substack{\text{Assumption 2} \\ \nabla F(w_k)^T \mathbb{E}[g(w_k,\xi_k)] \geq \mu \|\nabla F(w_k)\|^2}} + \frac{1}{2}\alpha_k^2 L \mathbb{E}_{\xi_k}[\|g(w_k, \xi_k)\|^2]$$

$$\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\mu\alpha_k \|\nabla F(w_k)\|^2 + \frac{1}{2}\alpha_k^2 L \underbrace{\mathbb{E}_{\xi_k}[\|g(w_k, \xi_k)\|^2]}_{\substack{\text{combined assumption 2} \\ \leq M + M_G \|\nabla F(w_k)\|^2}}$$

$$\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\mu\alpha_k \|\nabla F(w_k)\|^2 + \frac{1}{2}\alpha_k^2 L(M + M_G \|\nabla F(w_k)),$$

$$\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) \leq (-\mu\alpha_k + \frac{1}{2}\alpha_k^2 L M_G) \|\nabla F(w_k)\|^2 + \frac{1}{2}\alpha_k^2 L M.$$

Intuitive Convergence of SG with fixed stepsize

Consider the inequality of the second lemma

$$\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) \leq \underbrace{-(\mu - \frac{1}{2}\alpha_k L M_G)\alpha_k \|\nabla F(w_k)\|^2}_{\text{tends to } 0 \text{ as } \nabla F(w_k) \to 0} + \frac{1}{2}\alpha_k^2 L M.$$

For a fixes stepsize, the last term remains constant and, after some point, we cannot expect to reduce the distance between the objective iterates beyond $\frac{1}{2}\alpha^2 LM$. That is, we converge to a neighborhood of the optimal.

$\Rightarrow$ SG needs diminishing stepsizes $\alpha_k \to 0$ to converge.

Convergence of SG with fixed stepsize

Theorem

If F is an L-smooth and c-strongly convex function that satisfies Assumption 2, with $F_{\mathit{inf}} = F^*$, and the SG method is run with a positive stepsize $\alpha \leq \frac{\mu}{LM_G}$, then the expected optimality gap satisfies for all k

$$\mathbb{E}[F(w_k) - F^*] \leq \frac{\alpha LM}{2c\mu} + (1 - \alpha c\mu)^{k-1} \left(F(w_1) - F^* - \frac{\alpha LM}{2c\mu} \right),$$

where $\mathbb{E}$ is the expectation w.r.t joint distribution of all random variables.

A direct result of this theorem states that the SG iterates converge to the $\frac{\alpha LM}{2c\mu}$ neighborhood of the optimal value as $k \to \infty$.

Strongly Convex Functions

Definition

The objective function $F\colon \mathbb{R}^d \mapsto \mathbb{R}$ is **strongly convex** in that there exists a constant $c > 0$ such that

$$F(w) \geq F(\bar{w}) + \nabla F(\bar{w})^T(w - \bar{w}) + \frac{1}{2}c\|w - \bar{w}\|^2 \ \ \forall w,\ \bar{w} \in \mathbb{R}^d.$$

Moreover, F has a unique minimizer $w^* \in \mathbb{R}^d$ with $F^* = F(w^*)$ and satisfies

$$2c(F(w) - F^*) \leq \|\nabla F(w)\|^2 \quad \forall w \in \mathbb{R}^d.$$

If F is L-smooth and c-strongly convex, then $c \leq L$.

Proof

Let us use the bound on the stepsize $\alpha \leq \frac{\mu}{LM_G}$ in Lemma 2

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

As F is c-strongly convex, $2c(F(w_k) - F^*) \leq \|\nabla F(w_k)\|^2$ and we get

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

Subtracting F^* from both sides, taking total expectations, and rearranging

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

Subtracting the constant $\frac{\alpha LM}{2c\mu}$ from both sides

$$\begin{aligned}\mathbb{E}[F(w_{k+1}) - F^*] - \frac{\alpha LM}{2c\mu} &\leq (1 - \mu\alpha c)\mathbb{E}[F(w_k) - F^*] + \frac{1}{2}\alpha^2 LM - \frac{\alpha LM}{2c\mu} \\ &\leq (1 - \mu\alpha c)\mathbb{E}[F(w_k) - F^*] + \frac{\alpha LM}{2c\mu}(\alpha c\mu - 1) \\ &\leq (1 - \mu\alpha c)\left(\mathbb{E}[F(w_k) - F^*] - \frac{\alpha LM}{2c\mu}\right).\end{aligned}$$

Continuation Proof

This inequality

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

is a contraction because

$$0 < \alpha c\mu \underbrace{\leq}_{\alpha \leq \frac{\mu}{LM_G}} \frac{c\mu^2}{LM_G} \underbrace{\leq}_{M_G \geq \mu^2} \frac{c\mu^2}{L\mu^2} = \frac{c}{L} \underbrace{\leq}_{L \geq c} 1$$

Applying the contraction inequality $k - 1$ times, we obtain the desired result

$$\mathbb{E}[F(w_k) - F^*] - \frac{\alpha LM}{2c\mu} \leq (1 - \alpha c\mu)^{k-1}\left(F(w_1) - F^* - \frac{\alpha LM}{2c\mu}\right)$$

$$\mathbb{E}[F(w_k) - F^*] \leq \frac{\alpha LM}{2c\mu} + (1 - \alpha c\mu)^{k-1}\left(F(w_1) - F^* - \frac{\alpha LM}{2c\mu}\right).$$

Choice of Stepsize (Learning Rate)

From the inequality

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

we see that selecting a smaller stepsize worsens the contraction constant, but ensures convergence to a smaller neighborhood of the optimal value.

We have two cases of interest:

1. If the noise in the gradient decays with $\|\nabla F(w_k)\|^2$ $(M = 0)$, SG converges linearly to the optimal value.
2. If the gradient computation is noisy $(M > 0)$, we only have linear convergence to a neighborhood of the optimal value. After some point, the noise in the gradient prevents further progress.

Intuitive Approach to Decreasing Stepsizes

Run SG with a fixed stepsize and when progress stalls halve the stepsize. For instance:

- Run SG until iteration k_2 where the expected suboptimality gap is twice the asymptotic value

$$\mathbb{E}[F(w_{K_2}) - F^*] \leq 2\frac{\alpha_1 LM}{2c\mu} = 2F_{\alpha_1}.$$

- - Halve the stepsize $\{\alpha_{r+1}\} = \{\alpha_1 2^{-r}\}$ and repeat the process.

The sequence of optimality gaps converges to 0 and SG to a minimum.

$$\mathbb{E}[F(w_{k_{r+1}}) - F^*] \leq \alpha_1 2^{-r} \leq 2F_{\alpha_r} \quad \mathbb{E}[F(w_{k_r}) - F^*] \approx 2F_{\alpha_{r-1}} = 4F_{\alpha_r}$$

The speed of convergence depends on how many iterations it takes to reach each bound.

First Approach to Decreasing Stepsizes

We can show that each time the stepsize is cut in half, we need twice as many iterations to reach the next bound. As doubling the number of iterations, halves the suboptimality gap, the convergence rate is $\mathcal{O}(1/k)$.

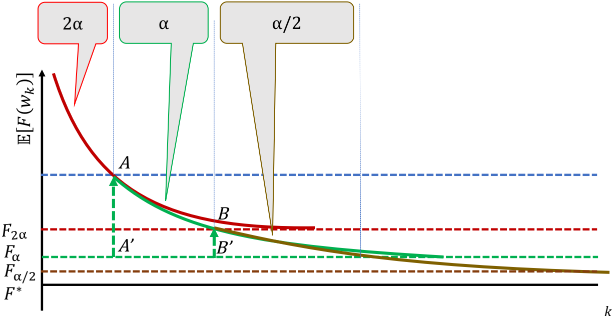


Fig.: Depiction of the strategy of halving the stepsize α when the expected suboptimality gap is smaller than twice the asymptotic limit F_α.

Convergence of SG with decaying stepsizes

Theorem

If F is an L-smooth and c-strongly convex function that satisfies Assumption 2, with $F_{\textit{inf}} = F^*$, and the SG method is run with a fixed stepsize sequence satisfying

$$\alpha_k = \frac{\beta}{\gamma + k} \quad \textit{for some} \quad \beta > \frac{1}{c\mu},\ \gamma > 0 \quad \textit{such that} \quad \alpha_1 \leq \frac{\mu}{LM_G}.$$

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

$$\mathbb{E}[F(w_k) - F^*] \leq \frac{\eta}{\gamma + k} \qquad \eta = \max\left\{\frac{\beta^2 LM}{2(\beta c\mu - 1)}, (\gamma + 1)(F(w_1) - F^*)\right\}.$$

Proof

As the step size decays $\alpha_k LM \leq \alpha_1 LM \leq \mu$, which in Lemma 2 gives

$$\begin{aligned}\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) &\leq -(\mu - \frac{1}{2}\alpha_k LM_G)\alpha_k \|\nabla F(w_k)\|^2 + \frac{1}{2}\alpha_k^2 LM \\ &\leq -\frac{\mu}{2}\alpha_k \|\nabla F(w_k)\|^2 + \frac{1}{2}\alpha_k^2 LM.\end{aligned}$$

As F is strongly convex, $2c(F(w_k) - F^*) \leq \|\nabla F(w_k)\|^2$ and

$$\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\mu\alpha_k c(F(w_k) - F^*) + \frac{1}{2}\alpha_k^2 LM$$

Subtracting F^* from both sides, taking total expectations, and rearranging,

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

Proof

We prove

$$\mathbb{E}[F(w_k) - F^*] \le \frac{\eta}{\gamma + k} \qquad \eta = \max\left\{\frac{\beta^2 LM}{2(\beta c\mu - 1)}, (\gamma+1)(F(w_1) - F^*)\right\}.$$

by induction. The definition of η ensures that it is true for $k = 1$. We assume that the inequality holds for some $k \ge 1$ and use it in the inequality from the previous slide

$$\mathbb{E}[F(w_{k+1}) - F^*] \le (1 - \mu\alpha_k c)\mathbb{E}[F(w_k) - F^*] + \frac{1}{2}\alpha_k^2 LM \tag{3.17}$$

$$\le (1 - \mu\alpha_k c)\frac{\eta}{\gamma + k} + \frac{1}{2}\alpha_k^2 LM. \tag{3.18}$$

Let $\hat{k} = \gamma + k$ and write $\alpha_k = \frac{\beta}{\hat{k}}$, the previous expression becomes

$$\begin{aligned}\mathbb{E}[F(w_{k+1}) - F^*] &\le (1 - \frac{\beta\mu c}{\hat{k}})\frac{\eta}{\hat{k}} + \frac{1}{2}\frac{\beta^2 LM}{\hat{k}^2} = \frac{\hat{k} - \beta\mu c}{\hat{k}^2}\eta + \frac{1}{2}\frac{\beta^2 LM}{\hat{k}^2} \\ &= \frac{\hat{k} - 1}{\hat{k}^2}\eta - \frac{\beta\mu c - 1}{\hat{k}^2}\eta + \frac{1}{2}\frac{\beta^2 LM}{\hat{k}^2} \\ &\le \frac{\hat{k} - 1}{\hat{k}^2}\eta + \underbrace{\frac{-2(\beta\mu c - 1)\eta + \beta^2 LM}{2\hat{k}^2}}_{\text{non positive, by definition of } \eta} \le \frac{\hat{k} - 1}{\hat{k}^2 - 1}\eta \le \frac{\eta}{\hat{k} + 1},\end{aligned}$$

As we promised last week

Let the objective function $F\colon \mathbb{R}^d \mapsto \mathbb{R}$ be the expected or empirical risk:

$$F(w) = \mathbb{E}[f(w,\xi)] \qquad \text{or} \qquad F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w).$$

1. If we pick samples uniformly from a finite training set with replacement, the SG iterates satisfy

$$\mathbb{E}[|R_n(w_k) - R_n^*|] = \mathcal{O}(\frac{1}{k})$$

2. If we pick samples in each iteration according to distribution P, the SG iterates satisfy

$$\mathbb{E}[|R(w_k) - R^*|] = \mathcal{O}(\frac{1}{k}).$$

Trade-Offs of mini-batch SG method

Compare simple SG to mini-batch SG with mini-batches of size $n_{mb} \ll n$

$$g(w_k, \xi_k) = \nabla f_{i_k}(w_k) \qquad g(w_k, \xi_k) = \frac{1}{|S_k|} \sum_{i \in S_k} \nabla f_i(w_k)$$

Observe that

- minibatch iterations are n_{mb} times more expensive than SG
- $\mathbb{V}[g(w_k, \xi_k)]$, M and M_V (assumption 2) are n_{mb} times smaller

Does the variance reduction pay for higher iteration cost?

Trade-Offs of mini-batch SG method

Assume we run minibatch SG with stepsize α and SG with α/n_{mb} and compare the number of iterations to reach the same optimality gap

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

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

SG needs n_{mb} times more iterations to obtain the optimality gap of minibatch SG, but each SG iteration is n_{mb} times cheaper.

⇒ The cost of SG and minibatch are the same if we can run minibatch SG with a stepsize n_{mb} times larger than the SG stepsize, which might not be possible because of the bound $\alpha < \frac{\mu}{LMG} \underbrace{<}_{M_G \geq \mu^2} \frac{1}{\mu L}$

SG for General Objectives

This is not part of the convex class, it is here to relate SG to the algorithms that some of you use in the deep learning lecture.

- Many important machine learning models lead to nonconvex optimization problems.
- Analyzing the SG method when minimizing nonconvex objectives is more challenging because functions may possess multiple local minima and other stationary points.
- Two results: one for employing a fixed positive stepsize and one for diminishing stepsizes.

Fixed Stepsize SG for General Objectives

Theorem

If F is an L-smooth function that satisfies Assumption 2, with $F_{\textit{inf}}$ the lower bound on the sequence of function values $\{F(w_k)\}$, and the SG method is run with positive stepsize $\alpha \leq \frac{\mu}{LM_G}$, then for all $K \in \mathbb{N}$:

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(w_k)\|^2\right] \leq \frac{\alpha L M}{\mu} + 2\frac{F(w_1) - F_{\textit{inf}}}{K\mu\alpha} \xrightarrow{K\to\infty} \frac{\alpha L M}{\mu}.$$

Observe

1. the asymptotic behavior illustrates that noise in the gradients inhibits further progress, as happens with the convex case.
2. The average norm of the gradients can be made arbitrarily small by selecting a small stepsize, but doing so reduces the speed at which the norm of the gradient approaches its limiting distribution.

Proof

Taking the total expectation in Lemma 2 and using the bound $0 < \alpha \leq \frac{\mu}{LM_G}$

$$\begin{aligned}\mathbb{E}[F(w_{k+1})] - \mathbb{E}[F(w_k)] &\leq -(\mu - \frac{1}{2}\alpha L M_G)\alpha \mathbb{E}[\|\nabla F(w_k)\|^2] + \frac{1}{2}\alpha^2 LM \\ &\leq -\frac{1}{2}\mu\alpha\mathbb{E}[\|\nabla F(w_k)\|^2] + \frac{1}{2}\alpha^2 LM,\end{aligned}$$

Summing the inequality for $k \in \{1, \dots, K\}$ and recalling $F_{\mathsf{inf}} \leq F(w_k)$,

$$F_{\mathsf{inf}} - F(w_1) \leq \mathbb{E}[F(w_{K+1})] - F(w_1) \leq -\frac{1}{2}\mu\alpha \sum_{k=1}^{K} \mathbb{E}[\|\nabla F(w_k)\|^2] + \frac{1}{2}K\alpha^2 LM.$$

Re-arranging terms we obtain the desired result

$$\sum_{k=1}^{K} \mathbb{E}[\|\nabla F(w_k)\|^2] \leq \frac{K\alpha LM}{\mu} + 2\frac{F(w_1) - F_{\mathsf{inf}}}{\mu\alpha}.$$

Interesting Cases

1. if the noise reduces proportionally to $\|\nabla F(w_k)\|^2$ $(M = 0)$

$$\sum_{k=1}^{K} \mathbb{E}[\|\nabla F(w_k)\|^2] \leq \underbrace{\frac{K\alpha LM}{\mu}}_{0} + 2\frac{F(w_1) - F_{\text{inf}}}{\mu\alpha}.$$

   the sum of squared gradients remains finite and $\{\|\nabla F(w_k)\|^2\} \to 0$.

2. in the presence of noise $(M > 0)$ the rhs of

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} \|\nabla F(w_k)\|^2\right] \leq \frac{\alpha LM}{\mu} + 2\frac{F(w_1) - F_{\text{inf}}}{K\mu\alpha} \xrightarrow{K\to\infty} \frac{\alpha LM}{\mu}.$$

   gets smaller as K increases and the SG method spends increasingly more time in regions where the objective has a small gradient.

Decreasing Stepsize SG for General Objectives

Theorem

If F is an L-smooth function that satisfies Assumption 2 and the SG method is run with a fixed stepsize sequence satisfying

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

Then, with $A_K = \sum_{k=1}^{K} \alpha_k$

$$\mathbb{E}\left[\sum_{k=1}^{K} \alpha_k \|\nabla F(w_k)\|^2\right] < \infty \Rightarrow \quad \mathbb{E}\left[\frac{1}{A_K} \sum_{k=1}^{K} \alpha_k \|\nabla F(w_k)\|^2\right] \xrightarrow{K\to\infty} 0.$$

Proof.

We proof the implication. The condition $\sum_{k=1}^{\infty} \alpha_k = \infty$ ensures that $A_K \to \infty$ as $K \to \infty$ and $\mathbb{E}\left[\frac{1}{A_K} \sum_{k=1}^{K} \alpha_k \|\nabla F(w_k)\|^2\right] \xrightarrow{K\to\infty} 0.$ □

Proof

The condition $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$ ensures that $\{\alpha_k\} \to 0$, and we can assume without loss of generality that $\alpha_k L M_G \leq \mu$ for all $k \in \mathbb{N}$. Taking the total expectation in Lemma 2 we have

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

Summing the inequality for $k \in \{1, \dots, K\}$ and recalling $F_{\text{inf}} \leq F(w_k)$

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

Dividing by $\frac{\mu}{2}$ and rearranging the terms, we obtain

$$\sum_{k=1}^{K} \alpha_k \mathbb{E}[\|\nabla F(w_k)\|^2] \leq 2\frac{F(w_1) - F_{\text{inf}}}{\mu} + \frac{LM}{\mu} \sum_{k=1}^{K} \alpha_k^2. \qquad (4.19)$$

As $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$, the right-hand side converges to a finite limit when $K \to \infty$, which proves $\mathbb{E}\left[\sum_{k=1}^{K} \alpha_k \|\nabla F(w_k)\|^2\right] < \infty$.

Consequences of Decreasing Stepsize SG for General Objectives

Under the theorem's assumptions, the weighted average norm of the squared gradients converges to zero even if the gradients are noisy because

$$\mathbb{E}\left[\frac{1}{A_K}\sum_{k=1}^{K}\alpha_k\|\nabla F(w_k)\|^2\right] \xrightarrow{K\to\infty} 0$$

Theorem

If F is an L-smooth function that satisfies Assumption 2 and the SG method is run with a fixed stepsize sequence satisfying

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

Then, the expected optimality gap satisfies the following inequality

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

Work Complexity[1]

We have discussed the convergence of SG when minimizing an objective function, but we have not discussed its computational cost. To this goal, define

$$h^* = \arg\min_h \mathbb{E}[\ell(h(x), y)] \qquad \text{optimal (w.r.t } R\text{) predictor function}$$

$$h^*_w = \arg\min_w \mathbb{E}[\ell(h(x;w), y)] \qquad \text{optimal (w.r.t } R\text{) parametric function}$$

$$h^*_n = \arg\min_w \ \frac{1}{n}\sum_{i=1}^{n} \ell(h(x_i;w), y_i) \quad \text{optimal (w.r.t } R_n\text{) parametric function}$$

We will stop the optimization when it reaches a solution $\tilde{h}_n$ that minimizes the objective function with ρ accuracy, that is,

$$R_n(\tilde{h}_n) \leq R_n(h^*_n) + \rho$$

[1] This Section is based on L. Bottou. *Stochastic Gradient Descent Tricks*. 2012.

Evaluation of Complexity for a Fixed Excess Error

The excess error $\varepsilon = \mathbb{E}[R_n(\tilde{h}_n) - R(h^*)]$ can then be decomposed in three terms

$$\varepsilon = \underbrace{\mathbb{E}[R_n(\tilde{h}_n) - R_n(h_n^*)]}_{\varepsilon_{\text{opt}}} + \underbrace{\mathbb{E}[R_n(h_n^*) - R(h_n^*)]}_{\varepsilon_{\text{est}}} + \underbrace{\mathbb{E}[R(h_w^*) - R(h^*)]}_{\varepsilon_{\text{approx}}}$$

1. The approximation error $\varepsilon_{\text{approx}}$ measures how closely functions in $\mathscr{H}$ can approximate the optimal solution h^*.
2. The estimation error ε_{est} measures the effect of minimizing the empirical risk R_n instead of the expected risk R.
3. The optimization error ε_{opt} measures the impact of the approximate optimization on the expected risk.

Small or Large scale Problems

Given constraints on the maximal computation time $T_{\max}$ and training set size $n_{\max}$, this decomposition outlines a trade-off

$$\min_{\mathscr{H},\rho,n} \varepsilon_{\mathsf{approx}} + \varepsilon_{\mathsf{est}} + \varepsilon_{\mathsf{opt}} \quad \text{s.t.} \quad n \leq n_{\max}, \quad T(\mathscr{H},\rho,n) \leq T_{\max}$$

Two cases should be distinguished:

1. small-scale problems are constrained by the maximal number of examples ($n = n_{\max}$) because computing time is not an issue ($T(\mathscr{H},\rho,n) \ll T_{\max}$) and $\varepsilon_{\mathsf{opt}} = \rho$ can be made arbitrarily small.
2. Large-scale learning problems are constrained by the maximal computing time ($T(\mathscr{H},\rho,n) = T_{\max}$) because the supply of training examples is very large ($n \ll n_{\max}$).

Asymptotic Analysis

In the asymptotic regime, the solution of

$$\min_{\mathscr{H},\rho,n} \varepsilon_{\textsf{approx}} + \underbrace{\varepsilon_{\textsf{opt}}}_{\rho} + \varepsilon_{\textsf{est}} \quad \text{s.t.} \quad n \leq n_{\max}, \quad T(\mathscr{H}, \epsilon, n) \leq T_{\max}$$

ensures that all the terms decrease at similar rates because the convergence of the sum is governed by its slowest term. That is:

$$\varepsilon_{\textsf{approx}} \approx \rho \approx \left(\frac{\log(n)}{n}\right)^{\beta} \quad \text{where } \varepsilon_{\textsf{est}} \sim \left(\frac{\log(n)}{n}\right)^{\beta} \beta \in [0.5, 1].$$

Simple vs Batch Gradient Method

Recall that gradient descent on an L-smooth and strongly convex function has a converges rate $\mathcal{O}(a^k)$, $0 < a < 1$.

A batch gradient method achieves ρ-optimality with a computing cost of the order $n\log(\frac{1}{\rho})$. Within the time budget $T_{\max}$, it can achieve ρ-optimality by processing $n \sim \frac{T_{\max}}{\log(\frac{1}{\rho})}$ examples.

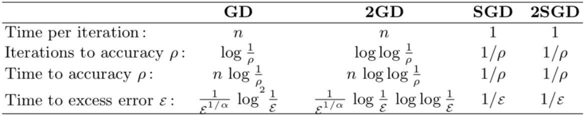
Assuming we operate at the optimum of the approximation, estimation, optimization trade-off, we can compute the computational cost necessary to reach a predefined value of the excess error by applying the equivalences

$$\varepsilon_{\mathsf{approx}} \approx \rho \approx \left(\frac{\log(n)}{n}\right)^{\beta}.$$

to eliminate the variables n and ρ from the cost.

Simple vs Batch Gradient Method

As SG with decreasing stepsize converges with $\mathcal{O}(\frac{1}{k})$ method can achieve ρ-optimality with a computing cost of the order $\frac{1}{\rho}$.



	GD	**2GD**	**SGD**	**2SGD**
Time per iteration :	n	n	1	1
Iterations to accuracy ρ :	$\log \frac{1}{\rho}$	$\log\log \frac{1}{\rho}$	$1/\rho$	$1/\rho$
Time to accuracy ρ :	$n \log \frac{1}{\rho}$	$n \log\log \frac{1}{\rho}$	$1/\rho$	$1/\rho$
Time to excess error ε :	$\frac{1}{\varepsilon^{1/\alpha}} \log^2 \frac{1}{\varepsilon}$	$\frac{1}{\varepsilon^{1/\alpha}} \log \frac{1}{\varepsilon} \log\log \frac{1}{\varepsilon}$	$1/\varepsilon$	$1/\varepsilon$

Fig.: Asymptotic equivalents for batch gradient descent (GD), second order batch gradient descent (2GD), simple stochastic gradient descent (SGD), and second order simple stochastic gradient descent (2SGD).

SGD and 2SGD are the worst optimization algorithms but achieve the fastest convergence speed on the expected risk.

Noise-Reduction Methods

SG suffers from the adverse effect of noisy gradient estimates.

- fixed α: convergence to solution neighborhood of size $\sim$ noise
- diminishing α: sublinear convergence to the solution

Noise reduction methods reduce noise in the gradient to achieve a linear rate of convergence

1. dynamic sampling methods: increase the mini-batch size.
2. gradient aggregation: average gradient estimates from past iterates.

Intuition

Recall the fundamental inequality

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

Which is the rate of decrease in noise that allows a SG method to converge at a linear rate?

- if $-g(w_k, \xi_k)$ is a descent direction in expectation $\Rightarrow$ first term < 0
- if $\mathbb{E}_{\xi_k}[\|g(w_k, \xi_k)\|^2]$ decreases fast enough (geometrically)

Noise Reduction and Convergence Rate

Theorem

Strongly Convex Objective, Noise Reduction Let F be an L-smooth and c-stronlgy convex function satisfying a modified Assumption 2, with constants $M \geq 0$ and $\zeta \in (0,1)$ such that, for all $k \in \mathbb{N}$

$$\mathbb{V}_{\xi_k}[g(w_k,\xi_k)] \cancel{\leq M + M_V \|\nabla F(w_k)\|^2} \leq M\zeta^{k-1}$$

Then the expected optimality gap of SG method with a fixed positive stepsize $\alpha \leq \min\{\frac{\mu}{L\mu_G^2}, \frac{1}{c\mu}\}$ satisfies

$$\mathbb{E}[F(w_k) - F^*] \leq \omega\rho^{k-1} \quad \textit{where} \quad \begin{cases} \omega & = \max\{\frac{\alpha L M}{c\mu}, F(w_1) - F^*\} \\ \rho & = \max\{1 - \frac{\alpha c\mu}{2}, \zeta\} < 1 \end{cases}$$

Proof

Recall Lemma 2

$$\begin{aligned}
\mathbb{E}_{\xi_k}[F(w_{k+1})]-F(w_k) &\leq -\mu\alpha\|\nabla F(w_k)\|^2 + \frac{\alpha^2}{2}L\mathbb{E}_{\xi_k}[\|g(w_k,\xi_k)\|^2] \\
&\leq -\mu\alpha\|\nabla F(w_k)\|^2 + \frac{\alpha^2}{2}L\left(\mathbb{V}_{\xi_k}[g(w_k,\xi_k)] + \|\mathbb{E}_{\xi_k}[g(w_k,\xi_k)]\|^2\right. \\
&\leq -\mu\alpha\|\nabla F(w_k)\|^2 + \frac{\alpha^2}{2}L\left(M\zeta^{k-1} + \mu_G^2\|\nabla F(w_k)\|^2\right) \\
&\leq -(\mu - \frac{1}{2}\alpha L\mu_G^2)\alpha\|\nabla F(w_k)\|^2 + \frac{1}{2}\alpha^2 LM\zeta^{k-1} \\
&\leq -\frac{1}{2}\mu\alpha\|\nabla F(w_k)\|^2 + \frac{1}{2}\alpha^2 LM\zeta^{k-1} \\
&\leq -\mu\alpha c(F(w) - F^*) + \frac{1}{2}\alpha^2 LM\zeta^{k-1}.
\end{aligned}$$

where in

- line 3, we have used Assumption 2
- line 5, $\alpha \leq \min\{\frac{\mu}{LM_G^2}, \frac{1}{c\mu}\}$
- line 6, F is c-strongly convex $2c(F(w) - F^*) \leq \|\nabla F(w_k)\|^2 \quad \forall w \in \mathbb{R}^d$

.

Proof'

From

$$\mathbb{E}_{\xi_k}[F(w_{k+1})] - F(w_k) \leq -\mu\alpha c(F(w) - F^*) + \frac{1}{2}\alpha^2 LM\zeta^{k-1},$$

we add and substract F^* and take total expectations to obtain

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

We now use induction to prove the bound on the gap

$$\mathbb{E}[F(w_k) - F^*] \leq \omega\rho^{k-1} \quad \text{where} \quad \begin{cases} \omega & = \max\{\frac{\alpha LM}{c\mu}, F(w_1) - F^*\} \\ \rho & = \max\{1 - \frac{\alpha c\mu}{2}, \zeta\} < 1 \end{cases}.$$

By definition of ω, it holds for $k = 1$.

Proof”

Assume that it holds for $k \geq 1$ and use $\mathbb{E}[F(w_k) - F^*] \leq \omega \rho^{k-1}$ in

$$
\begin{aligned}
\mathbb{E}[F(w_{k+1}) - F^* &\leq (1 - \alpha c \mu)\mathbb{E}[F(w_k) - F^*] + \frac{1}{2}\alpha^2 L M \zeta^{k-1} \\
&\leq (1 - \alpha c \mu)\omega \rho^{k-1} + \frac{1}{2}\alpha^2 L M \zeta^{k-1} \\
&\leq \omega \rho^{k-1}\left(1 - \alpha c \mu + \frac{\alpha^2 L M}{2\omega}\left(\frac{\zeta}{\rho}\right)^{k-1}\right) \\
&\leq \omega \rho^{k-1}\left(1 - \alpha c \mu + \frac{\alpha^2 L M}{2\omega}\right) \qquad \text{as } \rho > \zeta \\
&\leq \omega \rho^{k-1}\left(1 - \alpha c \mu + \frac{\alpha c \mu}{2}\right) \qquad \text{as } \omega > \frac{\alpha L M}{c\mu} \\
&\leq \omega \rho^{k-1}\left(1 - \frac{\alpha c \mu}{2}\right) \\
&\leq \omega \rho^{k-1} \qquad \text{as } 0 < \alpha < \frac{1}{c\mu}.
\end{aligned}
$$

Dynamic Sample Size Methods

Mini-batch SG where minibatch size grows geometrically $|S_k| = \lceil \tau^{k-1} \rceil$

$$\begin{cases} g(w_k, \xi_k) & = & \frac{1}{|S_k|} \sum_{i \in S_k} \nabla f(w_k; \xi_{k,i}) \\ w_{k+1} & = & w_k - \alpha g(w_k, \xi_k) \end{cases}. \tag{5.20}$$

Theorem

Let $\{w_k\}$ be the iterates generated by the dynamic sample size method (5.20) *with unbiased gradient estimates, then, the variance condition $\mathbb{V}_{\xi_k}[g(w_k, \psi_k)] \leq M\zeta^{k-1}$ is satisfied and, if the other assumptions of the previous Theorem hold, the expected optimality gap vanishes linearly.*

Dynamic Sample Size Methods

For instance, if the $\{\xi_{k,i}\}_{i \in S_k}$ are drawn independently according to P and we each stochastic gradient $\nabla f(w_k; \xi_{k,i})$ has an expectation equal to the true gradient $\nabla F(w_k)$ with a variance bounded by $M \geq 0$, then

$$\mathbb{V}_{\xi_k}[g(w_k, \xi_k)] \leq \frac{\mathbb{V}_{\xi_k}[\nabla f(w_k; \xi_{k,i})]}{n_k} \leq \frac{M}{n_k} \leq \frac{M}{\lceil \tau^{k-1} \rceil} \leq M \zeta^{k-1}.$$

Question: Is the method really linearly convergent if the per-iteration cost increases without bound with the minibatch size?

Theorem

Let F be an L-smooth and c-strongly convex function satisfying Assumption 2, with $F_{\inf} = F^*$, and run the dynamic sampling SG method with a positive stepsize $\alpha \leq \min\{\frac{\mu}{L\mu_G^2}, \frac{1}{c\mu}\}$ and $\tau \in (1, (1 - \frac{\alpha c \mu}{2})^{-1})$. Then, the total number of evaluations of a stochastic gradient of the form $\nabla f(w_k; \xi_{k,i})$ required to obtain $\mathbb{E}[F(w_k) - F^*] \leq \epsilon$ is $\mathcal{O}(\frac{1}{\epsilon})$.

Gradient Aggregation Methods

Rather than reducing the variance of the stochastic gradients by using more samples in each iteration, gradient aggregation methods achieve a lower variance by reusing previously computed information.

Estimate the bias of the SG gradient estimates and correct it

1. SVGR: each iteration is as costly as batch SG
2. SAGA: each iteration as cheap as simple SG, but large memory requirements

SVGR: Stochastic Variance Reduced Gradient

SVGR operates in cycles.

1. at the beginning of each cycle, an iterate w_k is available at which the algorithm computes a batch gradient

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

2. compute w_{k+1} in m inner iterations with $i_j \in \{1, \ldots, n\}$ random.

$$\tilde{g}_j = \nabla f_{i_j}(\tilde{w}_j) - \underbrace{(\nabla f_{i_j}(w_k) - R_n(w_k))}_{\substack{\text{bias in gradient estimate } \nabla f_{i_j}(w_k) \\ \text{because } \mathbb{E}_{i_j}[\nabla f_{i_j}(w_k)] = R_n(w_k)}} \tag{5.21}$$

$$\tilde{w}_{j+1} = \tilde{w}_j - \alpha \tilde{g}_j. \tag{5.22}$$

In every iteration, SVGR randomly draws a stochastic gradient $\nabla f_{i_j}(\tilde{w}_k)$ and corrects it based on a perceived bias, i.e., $\tilde{g}_j$ is an unbiased estimate of $\nabla R_n(\tilde{w}_j)$ with smaller variance than the SG estimate $\tilde{g}_j = \nabla f_{i_j}(\tilde{w}_j)$.

SVGR: Stochastic Variance Reduced Gradient

```
Choose an initial iterate $w_1 \in \mathbb{R}^d$, stepsize $\alpha > 0$, and integer $m$.
for $k = 1, 2, \ldots$ do
    Compute the batch gradient $\nabla R_n(w_k)$.
    Initialize $\tilde{w}_1 \leftarrow w_k$.
    for $j = 1, \ldots, m$ do
        Chose $i_j$ uniformly from $\{1, \ldots, n\}$.
        Set $\tilde{g}_j \leftarrow \nabla f_{i_j}(\tilde{w}_j) - (\nabla f_{i_j}(w_k) - \nabla R_n(w_k))$.
        Set $\tilde{w}_{j+1} \leftarrow \tilde{w}_j - \alpha \tilde{g}_j$.
    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
```

SAGA

The SAGA only computes batch gradients at the initial point but needs to store n stochastic gradient vectors $\nabla f_i(w_{[1]}), \nabla f_i(w_{[2]}), \ldots, \nabla f_i(w_{[n]})$, where $w_{[i]}$ is the latest iterate at which ∇f_i was evaluated.

In each iteration, SAGA computes a stochastic vector g_k as the average of stochastic gradients evaluated at previous iterates. Let $j \in \{1, \ldots, n\}$ be chosen at random,

$$g_k = \nabla f_j(wk) - \nabla f_j(w_{[j]}) + \frac{1}{n} \sum_{i=1}^{n} f_i(w_{[i]})$$

As $\mathbb{E}_{j \in \{1,\ldots,n\}}[g_k] = \nabla R_n(w_k)$, SAGA uses unbiased gradient estimates with variance expected to be less than the variance of simple SG.

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**
```