Stochastic Gradient Methods for Large-Scale Machine Learning

Leon Bottou
Facebook AI Research

Frank E. Curtis
Lehigh University

Jorge Nocedal
Northwestern University

LEHIGH 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

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

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

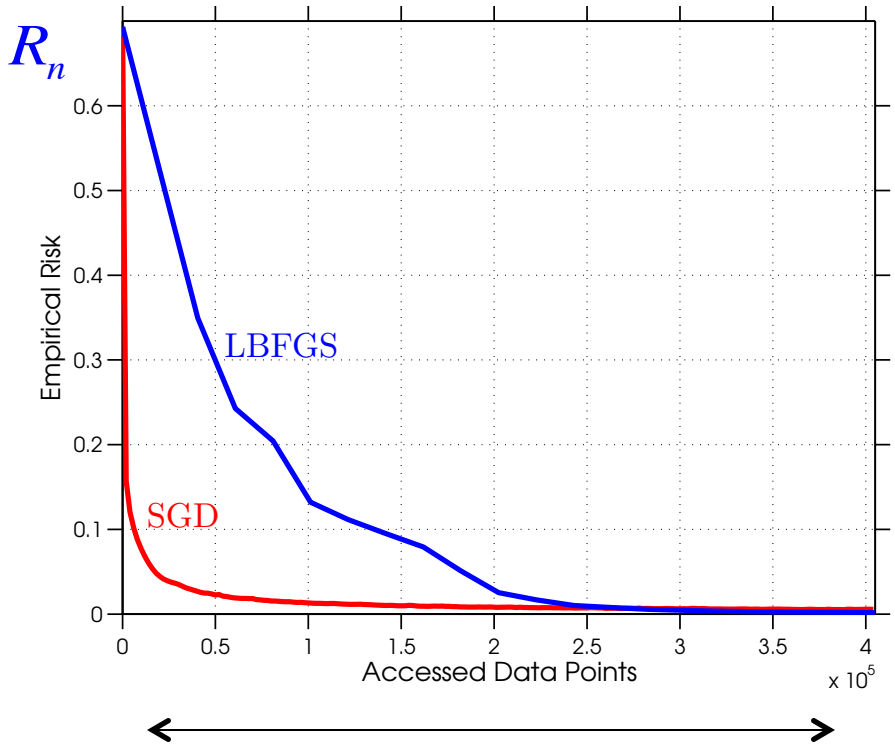
Argument 1:

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

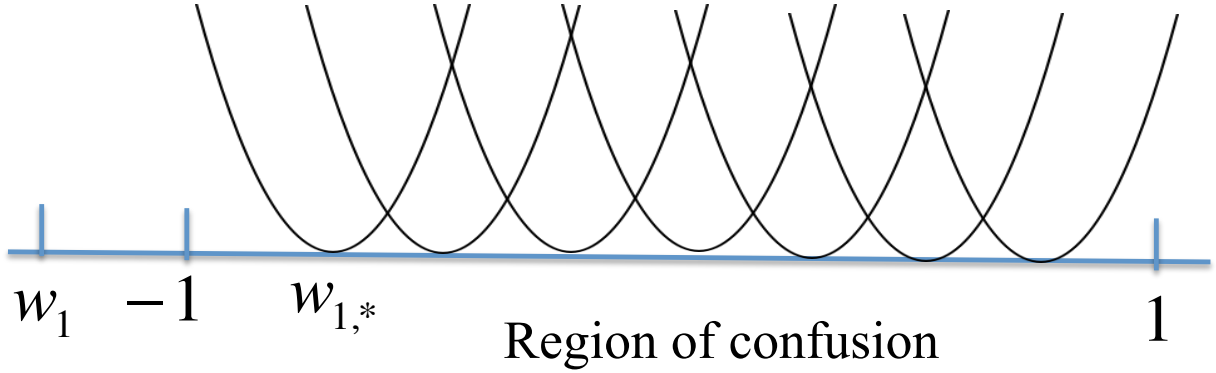


Fast initial progress of SG followed by drastic slowdown

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 $\alpha_k \to 0$ 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