Introduction of Stochastic Optimization on Minimizing Finite Sums

Zhong Hui

University of Science and Technology

zhonghui.net@gmail.com

November 2, 2016

Overview

1. Introduction
   - Problem
2. Preliminaries
   - Convergence Rate
   - Proximal Operator
3. Methods
   - FG: Full Gradient Method
   - SG: Stochastic Gradient
   - SAG: Stochastic Average Gradient
   - SVRG: Stochastic Variance Reduced Gradient
   - SAGA
4. Compare
   - Iterations forms
   - Basic summary of method properties

Outline

Examples

- Least-squares regression

$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}(a_i^T x - b_i)^2$$

- Logistic regression

$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-b_i a_i^T x))$$

Minimizing finite average of convex functions

- Minimizing function form

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

- Add an additional regularization function

$$F(x) = g(x) + h(x) \tag{2}$$

- Where
  - $x \in \mathbb{R}^d$ and each f_i is convex and has Lipschitz continuous derivatives with constant L

    $$\|f_i^{'}(x) - f_i^{'}(y)\| \leq L\|x - y\|$$

  - each f_i is strongly convex with constant μ

    $$\nabla^2 f(x) \succeq \mu I$$

  - $h : \mathbb{R}^d \to \mathbb{R}^d$: convex but potentially non-differentiable, and where the proximal operation of h is easy to compute

Outline

Convergence rate[5]

Suppose that the sequence $\{ x_k \}$ converges to the number L. This sequence **converges linearly** to L, if there exists a number $\mu \in (0, 1)$ such that

$$\lim_{k \to \infty} \frac{|x_{k+1} - L|}{|x_k - L|} = \mu.$$

The number μ is called the *rate of convergence*.

If the sequence converges, and

- * $\mu = \mu_k$ varies from step to step with $\mu_k \to 0$ for $k \to \infty$, then the sequence is said to **converge superlinearly**.
- * $\mu = \mu_k$ varies from step to step with $\mu_k \to 1$ for $k \to \infty$, then the sequence is said to **converge sublinearly**

Proximal Operator[4]

the proximal operator of a convex function h is defined as

$$prox_h(x) = arg \min_u \left(h(u) + \frac{1}{2}\|u - x\|^2 \right)$$

Examples

- $h(x) = 0$: $prox_h(x) = x$
- $h(x)$ is indicator function of closed convex set C:$prox_h(x)$is projection on C

$$prox_h(x) = arg \min_{u \in C} \|u - x\|_2^2 = P_C(x)$$

- $h(x) = \|x\|^1$: $prox_h(x)$ is the soft-threshold (shrinkage) operation

$$prox_h(x)_i = \begin{cases} x_i - 1 & x_i \geq 1 \\ 0 & |x_i| \leq 1 \\ x_i + 1 & x_i \leq -1 \end{cases}$$

Proximal gradient method

unconstrained optimization with objective split in two components

$$\text{minimize } f(x) = g(x) + h(x) \tag{3}$$

- g convex, differentiable, $dom\ g = \mathbf{R}^n$
- h convex with inexpensive prox-operator

Proximal gradient algorithm

$$x^{(k)} = prox_{t_k h}\left(x^{(k-1)} - t_k \nabla g(x^{(k-1)})\right) \tag{4}$$

- $t_k > 0$ is step size, constant or determined by line search
- can start at infeasible $x^{(0)}$(however $x^{(k)} \in dom\ f = dom\ h$ for $k \geq 1$)

Outline

FG (full gradient) method

- FG method, which dates back to Cauchy [1847], uses iterations of the form

$$x^{k+1} = x^k - \alpha_k g^{'}(x^k) = x^k - \frac{\alpha^k}{n}\sum_{i=1}^{n} f_i^{'}(x^k) \tag{5}$$

- *linear convergence rate* $O(\rho^k)$ for strongly-convex objectives, $O(1/k)$ for convex objectives.
- can be unappealing when n is large because its iteration cost scales linearly in n

Stochastic Gradient (SG)

- Iterations form
$$x^{k+1} = x^k - \alpha^k f_{i_k}^{'}(x^k) \tag{6}$$
- index i_k is sampled uniformly from the set $\{1, ..., n\}$. The randomly chosen gradient $f_{i_k}^{'}(x_k)$ yields an unbiased estimate of the true gradient $g^{'}(x_k)$.
- for a suitably chosen decreasing step-size sequence $\{\alpha_k\}$, the SG iterations have an expected sub-optimality for convex objectives of
$$\mathbb{E}\left[g(x^k)\right] - g(x^*) = O(1/\sqrt{k})$$
and an expected sub-optimality for strongly-convex objectives of
$$\mathbb{E}\left[g(x^k)\right] - g(x^*) = O(1/k)$$

Stochastic Average Gradient(SAG)[3]

- Iterations form

$$x^{k+1} = x^k - \frac{\alpha^k}{n} \sum_{i=1}^{n} y_i^k \tag{7}$$

$$y_i^k = \begin{cases} f_i'(x^k) & \text{if } i = i_k, \\ y_i^{k-1} & \text{otherwise.} \end{cases} \tag{8}$$

- like the SG method, each iteration only computes the gradient with respect to a single example and the cost of the iterations is independent of n.

Stochastic Average Gradient(SAG)

- with a constant step-size the SAG iterations have an $O(1/k)$ convergence rate for convex objectives and a *linear convergence rate* for strongly-convex objectives, like the FG method.
- by having access to i_k and by keeping a memory of the most recent gradient value computed for each index i, this iteration achieves a faster convergence rate than is possible for standard SG methods.

Stochastic variance reduced gradient (SVRG)[2]

Motivation

- Reduce the variance
- Stochastic gradient descent has slow convergence asymptotically duto the inherent variance.
- SAG needs to store all gradients

Contribution

- No need to store the intermediate gradients
- The same convergence rate as SAG can obtain
- Under mild assumptions, even work on nonconvex cases

SVRG Procedure

Procedure SVRG

Parameters update frequency m and learning rate η
Initialize $\tilde{w}_0$
Iterate: for $s = 1, 2, \ldots$
$\tilde{w} = \tilde{w}_{s-1}$
$\tilde{\mu} = \frac{1}{n} \sum_{i=1}^{n} \nabla \psi_i(\tilde{w})$
$w_0 = \tilde{w}$
Iterate: for $t = 1, 2, \ldots, m$
Randomly pick $i_t \in \{1, \ldots, n\}$ and update weight
$w_t = w_{t-1} - \eta(\nabla \psi_{i_t}(w_{t-1}) - \nabla \psi_{i_t}(\tilde{w}) + \tilde{\mu})$
end
option I: set $\tilde{w}_s = w_m$
option II: set $\tilde{w}_s = w_t$ for randomly chosen $t \in \{0, \ldots, m-1\}$
end

Stochastic Variance Reduced Gradient

SAGA[1]

- SAGA improves on the theory behind SAG and SVRG, with better theoretical convergence rates,
- and has support for composite objectives where a proximal operator is used on the regulariser.
- Unlike SDCA, SAGA supports non-strongly convex problems directly, and is adaptive to any inherent strong convexity of the problem.

SAGA

Iterations form

$$x^{k+1} = x^k - \alpha \left[f_j'(x^k) - f_j'(\phi_j^k) + \frac{1}{n}\sum_{i=1}^{n} f_i'(\phi_i^k) \right] \tag{9}$$

index j is sampled uniformly from the set $\{1, ..., n\}$. $\phi_j^k = x_{k-1}$, and store $f_j'(\phi_j^k)$ in the table of all $\sum f_i'(\phi_i^k)$ sets.

the same convergence rate as FG, *linear convergence rate* $O(\rho^k)$ for strongly-convex objectives, $O(1/k)$ for convex objectives.

Outline

Iterations forms

$$(SAG)\quad x^{k+1} = x^k - \gamma\left[\frac{f_j'(x^k) - f_j'(\phi_j^k)}{n} + \frac{1}{n}\sum_{i=1}^{n} f_i'(\phi_i^k)\right] \tag{10}$$

$$(SAGA)\quad x^{k+1} = x^k - \gamma\left[f_j'(x^k) - f_j'(\phi_j^k) + \frac{1}{n}\sum_{i=1}^{n} f_i'(\phi_i^k)\right] \tag{11}$$

$$(SVRG)\quad x^{k+1} = x^k - \gamma\left[f_j'(x^k) - f_j'(\widetilde{x}) + \frac{1}{n}\sum_{i=1}^{n} f_i'(\widetilde{x})\right] \tag{12}$$

Variance reduction approach

$$\theta_\alpha := \alpha(X - Y) + \mathbb{E}Y, \quad \alpha \in (0,1).$$
$$Var(\theta_\alpha) = \alpha^2 \left[Var(X) + Var(Y) - 2Cov(X,Y)\right]$$

- Here X is the SGD direction sample $f_j^{'}(x_k)$, whereas Y is a past stored gradient $f_j^{'}(\phi_j^k)$, and SVRG using $Y = f_j^{'}(\widetilde{x})$.
- SAG is obtained by using $\alpha = 1/n$, whereas SAGA is the unbiased version with $\alpha = 1$, and SVRG with $\alpha = 1$.
- For the same ϕ's, the variance of the SAG update is $1/n^2$ times the one of SAGA, but at the expense of having a non-zero bias.

Properies

	SAG	SVRG	SAGA
Strong Convex(SC)	✓	✓	✓
Convex,Non-SC^*	✓	?	✓
Prox Reg	?	✓	✓
Non smooth	×	×	×
Low Stroage Cost	×	✓	×
Simple(-ish) Proof	×	✓	✓

Basic summary of method properties. Question marks denote unproven, but not experimentally ruled out cases. (*) Note that any method can be applied to non-strongly convex problems by adding a small amount of $L2$ regularisation, this row describes methods that do not require this trick.

Reference

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien.
Saga: A fast incremental gradient method with support for non-strongly convex composite objectives.
In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

Rie Johnson and Tong Zhang.
Accelerating stochastic gradient descent using predictive variance reduction.
In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

Mark Schmidt, Nicolas Le Roux, and Francis Bach.
Minimizing finite sums with the stochastic average gradient.
arXiv preprint arXiv:1309.2388, 2013.

Prof. L Vandenberghe.
Optimization methods for large-scale systems (spring 2016 ucla), 2016.

Wikipedia.
Rate of convergence — wikipedia, the free encyclopedia, 2016.
[Online; accessed 31-October-2016].

The End