Engineering Applications of Artificial Intelligence 72 (2018) 124–135

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Random Barzilai–Borwein step size for mini-batch algorithms

Zhuang Yang [a], Cheng Wang [a,*], Zhemin Zhang [a], Jonathan Li [a,b]

[a] Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Information Science and Engineering, Xiamen University, Xiamen, FJ 361005, China

[b] Department of Geography and Environmental Management, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

ARTICLE INFO

Keywords:
Stochastic gradient descent
Mini batches
Barzilai–Borwein method
Variance reduction
Convex optimization

ABSTRACT

Mini-batch algorithms, a well-studied, highly popular approach in stochastic optimization methods, are used by practitioners because of their ability to accelerate training through better use of parallel processing power and reduction of stochastic variance. However, mini-batch algorithms often employ either a diminishing step size or a tuning step size by hand, which, in practice, can be time consuming. In this paper, we propose using the improved Barzilai–Borwein (BB) method to automatically compute step sizes for the state of the art mini-batch algorithm (mini-batch semi-stochastic gradient descent (mS2GD) method), which leads to a new algorithm: mS2GD-RBB. We theoretically prove that mS2GD-RBB converges with a linear convergence rate for strongly convex objective functions. To further validate the efficacy and scalability of the improved BB method, we introduce it into another modern mini-batch algorithm, Accelerated Mini-Batch Prox SVRG (Acc-Prox-SVRG) method. In a machine learning context, numerical experiments on three benchmark data sets indicate that the proposed methods outperform some advanced stochastic optimization methods.

1. Introduction

In recent years, the variety and volume of data have grown rapidly. Masses of data have led to increased interest in scalable optimization. One of the most popular and practical methods, dating back to the 1951 seminal work of Robbins and Monro (1951), is the stochastic gradient descent (SGD) method. The SGD method has significant theoretical and empirical advantages in machine learning (Bekkerman et al., 2011; Wang and Han, 2015), as well as in compressed sensing (Carpentier and Munos, 2012; Xu and Minin, 2015), wireless sensor networks (Lavanya and Udgata, 2011; Manjarres et al., 2013), matrix factorization (Gemulla et al., 2011; Luo et al., 2012), and large scale natural language processing (Gimpel et al., 2010).

In machine learning, the traditional SGD method (Zhang, 2004; Shamir and Zhang, 2013) uses a single random example in each iteration. The information obtained by computing the gradient of the empirical risk function associated with this example is used to update the predictor. This leads to a more fine-grained iterative process with low computational cost per iteration, but concurrently introduces considerable stochastic noise. The most obvious manifestation is that the stochastic estimate of the gradient has a non-vanishing variance.

Typically, there have been two approaches to deal with the issue of stochastic noise. (1) Use a decreasing step size (a.k.a learning rate) (Luo, 1991; Solodov, 1998; Zhang, 2004; Nemirovski et al., 2008; Shamir and Zhang, 2013). However, a diminishing step size, often leading to slow convergence near the eventual limit, demands exhaustive experimentation to determine how rapidly the step size must decrease in order to prevent scenarios where the step size becomes too small when the iterations are far from the eventual limit. (2) Use a mini-batching technique (Shalev-Shwartz et al., 2007; Dekel et al., 2012; Cotter et al., 2011; Konečnỳ et al., 2016). However, this technique leads to the unwelcome side-effect of requiring more computations. As these two cases indicate, traditional methods manage to decrease the variance in the stochastic estimate, but that decrease comes at a cost.

Does a mini-batching strategy allow the stochastic optimization methods to use a non-decreasing step size? Actually, mini-batch algorithms often employ either a diminishing step size, or a tuning step size by hand, which, in practice, can be time consuming. For instance, under certain assumptions, some researchers (Duchi and Singer, 2009; Nesterov, 2009; Xiao, 2010; Dekel et al., 2012; Lan, 2012; Byrd et al., 2016) employ a diminishing step size in their proposed mini-batch methods. Berahas et al. (2016) show that the Multi-Batch L-BFGS method, with a constant step size, converges to within a neighborhood of the optimal solution. They also point out that, according to the schedule proposed by Robbins and Monro (1951), by using a step size sequence, $\{\eta_k\}$ to zero, the Multi-Batch L-BFGS method converges to the optimal solution. Li et al. (2014) introduce a technique based on an approximate optimization of a conservatively regularized objective function within

[*] Corresponding author.
E-mail address: cwang@xmu.edu.cn (C. Wang).

https://doi.org/10.1016/j.engappai.2018.03.017
Received 2 November 2017; Received in revised form 27 January 2018; Accepted 22 March 2018

each mini-batch and establish convergence on a decreasing step size for the proposed method. In addition, under certain assumptions, they argue that the step size can develop into a constant step size. Ghadimi et al. (2016) propose a randomized stochastic projected gradient (RSPG) algorithm and analyze its convergence when it employs a non-increasing step size or a non-decreasing step size. Recently, Konečný et al. (2016) proposed the mini-batch semi-stochastic gradient descent (mS2GD) method, which uses a tuning constant step size.

As Roux et al. (2012) indicate, one vital issue regarding stochastic algorithms, that has not been fully addressed in the literature, is how to choose an appropriate step size while running the algorithms. In the classical deterministic method, step size is often obtained by employing line search techniques. However, line search is computationally prohibitive in stochastic gradient methods, because it uses randomly chosen gradient samples and does not allow for a strict sequence of decisions that collapse the search space. Hence, a decreasing or best-tuned fixed step size is often employed in stochastic optimization methods.

Inspired by recent works (Sopyła and Drozda, 2015; Tan et al., 2016; Bordes et al., 2009; Byrd et al., 2016), instead of using a diminishing step size or a tuning step size by hand in the mini-batch algorithms, we equip the state of the art mini-batch algorithm, mS2GD , with the ability to automatically compute step size by using the improved Barzilai–Borwein (BB) method. Sopyła and Drozda (2015) incorporated the BB method into the classic SGD algorithm for training the linear SVM in its primal form. In Sopyła and Drozda (2015), the proposed methods use a random sample to compute step size. However, such methods perform worse than the existing methods. Moreover, in Sopyła and Drozda (2015), theoretical justifications are not established. Tan et al. (2016) proposed using the BB method to compute step size for SGD and its variants: the stochastic variance reduced gradient (SVRG) method, which leads to two algorithms: SGD-BB and SVRG-BB. Each step size in SGD-BB and SVRG-BB is computed using the full gradient of objective functions after a succession of stochastic iterations. SVRG-BB and SGD-BB show promise because, while running, they automatically generate the best step sizes. Indeed, the key idea behind the BB method is motivated by the quasi-Newton property in deterministic optimization. Bordes et al. (2009) and Byrd et al. (2016) used batch samples to approximate quasi-Newton property in stochastic optimization and indicated that their proposed methods show great promise for solving the problems that arise in machine learning.

In our proposed method, to compute step size, the improved BB method uses partial samples, randomly chosen from full samples. Compared with SGD-BB and SVRG-BB, which update each step size after a large number of stochastic steps, our proposed method updates the step size in each stochastic iteration faster and performs well in practice.

The following are some recent works that discuss the choice of step size in stochastic optimization methods: Cotter et al. (2011) specify a novel, accelerated gradient strategy for mini-batch algorithms, where step size, η_k, is scaled polynomially in iteration, k. Schmidt et al. (2015) incorporate the standard backtracking line search into SAG to obtain step size. Mahsereci and Hennig (2015) suggest performing line search to obtain step size for a univariate optimization objective in the Gaussian process.

The primary contributions of this paper are as follows:

- We equip the state of the art mini-batch algorithm, mS2GD, which already has a fast rate, with the ability to automatically compute step size by using the improved BB method, thereby, obtaining a new method: mS2GD-RBB. We prove that our mS2GD-RBB method converges linearly for strongly convex objective functions.
- To further validate the efficacy and scalability of the improved BB method, we introduce it into another modern mini-batch algorithm, the Accelerated Mini-Batch Prox SVRG (Acc-Prox-SVRG) method, which leads to another new mini-batch algorithm: Acc-Prox-SVRG-RBB.
- We conduct experiments, using the proposed methods, to solve logistic regression in three benchmark data sets. Experimental results show that our proposed method obtains a rapidly updated step size sequence in each stochastic stage and achieves better performance than the variants of some advanced SGD and batch algorithms.

The remainder of this paper is organized as follows: Section 2 gives the problem statement and background. Section 3 introduces the details of our proposed method. Section 4 analyzes the convergence of our proposed method. Section 5 presents our numerical results. Section 6 further discusses the efficacy and scalability of the improved BB method. Section 7 concludes the paper.

2. Problem statement and background

Many problems of interest are often formulated as the following optimization problem:

$$\min_{w\in\mathbb{R}^d} F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w). \tag{1}$$

Throughout this paper, we focus on such problems where both each f_i and $F(w)$ have Lipschitz continuous derivatives, and, also, are strongly convex. The canonical example is least squares, and in that case, $f_i(w) = \frac{1}{2}(x_i^T w - y_i)^2 + \frac{\lambda}{2}\|w\|_2^2$, where λ is a regularization parameter. Another widespread example is logistic regression, described by the choice $f_i(w) = \log(1 + \exp[-y_i x_i^T w]) + \frac{\lambda}{2}\|w\|_2^2$.

To solve the above optimization, the standard mini-batch SGD (Byrd et al., 2012; Dekel et al., 2012) uses the following stochastic update rule: at each iteration k, mini-batch $S_1 \subset \{1,\dots,n\}$ of size b_1 is picked at random and let

$$w_{k+1} = w_k - \eta_k \nabla F_{S_1}(w_k), \tag{2}$$

where $\eta_k > 0$ is the step size in the kth iteration, and

$$\nabla F_{S_1}(w_k) = \frac{1}{b_1}\sum_{i\in S_1} \nabla f_i(w_k), \tag{3}$$

where ∇f_i is the gradient of the ith component function at w_k. If we set mini-batch size $b_1 = 1$, the iteration scheme Eq. (2) degrades into the common SGD method (Bottou, 2010) that employs a single sample per iteration, i.e., $w_{k+1} = w_k - \eta_k \nabla f_i(w_k)$.

3. The algorithm

In this section, we introduce the random BB step size, followed by the introduction of mS2GD, and then describe our mS2GD-RBB method, which equips mS2GD with the random BB step size.

3.1. Random Barzilai–Borwein step size

The BB method, proposed by Barzilai and Borwein in Barzilai and Borwein (1988), has been proven to be an efficient gradient method for solving nonlinear optimization problems. In the BB method, some quasi-Newton properties are used (Zheng and Zheng, 2016). Suppose we want to solve the unconstrained minimization problem

$$\min_{w\in\mathbb{R}^n} f(w), \tag{4}$$

where f is differentiable. A typical iteration of the quasi-Newton methods (Dennis and More, 1974) for solving Eq. (4) is:

$$w_{k+1} = w_k - H_k^{-1}\nabla f(w_k), \tag{5}$$

where H_k is an approximation of the Hessian matrix of f at the current iteration, w_k. The most important feature of H_k is that it must satisfy the so-called secant equation (Biglari and Solimanpur, 2013; Dai, 2013): $H_k s_k = y_k$, where $s_k = w_k - w_{k-1}$ and $y_k = \nabla f(w_k) - \nabla f(w_{k-1})$. Now approximate Hessian matrix H_k by $H_k = (1/\eta_k)I$ with $\eta_k > 0$ and