Pure Mathematics 理论数学, 2023, 13(11), 3165-3175
Published Online November 2023 in Hans. https://www.hanspub.org/journal/pm
https://doi.org/10.12677/pm.2023.1311328

Hans汉斯

一种赋有新的BB类步长的随机递归梯度算法

陈炫睿

贵州大学数学与统计学院，贵州 贵阳

收稿日期：2023年10月2日；录用日期：2023年11月2日；发布日期：2023年11月13日

摘　要

随机递归梯度算法(SARAH)最近引起了人们的广泛关注。它允许一个简单的递归框架来更新随机梯度估计。SARAH与重要性抽样策略相结合得到了SARAH-I算法。基于此，本文提出了一种新的随机递归梯度方法。该算法将SARAH-I算法与具有二维二次终止性的BB类步长相结合，使SARAH-I算法的步长能够自适应计算，具有较好的数值性能。最后通过数值实验我们观察到，新算法对初始步长的选取不敏感，并且具有自动生成最优步长的能力。

关键词

随机递归梯度算法，BB步长，自适应计算，随机优化

A New Stochastic Recursive Gradient Algorithm with a BB-Like Stepsize

Xuanrui Chen

School of Mathematics and Statistics, Guizhou University, Guiyang Guizhou

Received: Oct. 2nd, 2023; accepted: Nov. 2nd, 2023; published: Nov. 13th, 2023

Abstract

The stochastic recursive gradient algorithm (SARAH) attracts much interest recently. It admits a simple recursive framework for updating stochastic gradient estimates. SARAH-I algorithm is obtained by combining SARAH with importance sampling strategy. Based on this, a new stochastic recursive gradient method is proposed in this paper. This algorithm combines SARAH-I algorithm with a BB-like stepsize with two dimensional quadratic termination property, which makes the SARAH-I algorithm automatically compute stepsizes and has good numerical performance. Finally, through numerical experiments, we observe that the new algorithm is insensitive to the selection of the initial stepsize, and has the ability to automatically generate the optimal stepsize.

文章引用: 陈炫睿. 一种赋有新的 BB 类步长的随机递归梯度算法[J]. 理论数学, 2023, 13(11): 3165-3175.
DOI: 10.12677/pm.2023.1311328

Keywords

Stochastic Recursive Gradient Algorithm, BB Stepsize, Adaptive Computing, Stochastic Optimization

1. 引言

数据科学中的许多问题(例如，机器学习，优化和统计)可以被描述为这种形式的损失最小化问题：

$$\min_{x\in R^d} f(x), \tag{1}$$

其中：

$$f(x)=\frac{1}{n}\sum_{i=1}^{n} f_i(x). \tag{2}$$

这里 n 为样本量，每个 $f_i: R^d \to R$ 为第 i 个样本数据对应的损失函数。在本文中，我们假设每个分量函数 f_i 都是凸函数且可微，并且函数 $f(x)$ 是强凸的。

问题(1)当 n 非常大时是极具挑战性的。由于精确的全梯度信息不易获得，因此基于精确梯度的方法不切实际且被禁止。然而，随机梯度下降法(SGD)可以追溯到[1]的开创性工作，已经成为解决问题(1)的主要方法。在第 n 次迭代中，SGD 随机选取索引 $i\in[n]$，然后更新迭代 x_t 通过：

$$x_{t+1}=x_t-a_t\nabla f_{i_t}(x_t),\ \ t=0,1,2,\cdots \tag{3}$$

其中 $a_t>0$ 为步长(也称为学习率)，$\nabla f_{i_t}(x_t)$ 表示 x_t 处的样本梯度。在(3)中，通常假设 ∇f_{i_t} 是对 $\nabla f(x)$ 的无偏估计，即：

$$\mathbb{E}\left[\nabla f_{i_t}(x_t)\middle| x_t\right]=\nabla f(x_t). \tag{4}$$

然而，已知 SGD 的梯度评估的总次数取决于随机梯度的方差，并且对于强凸光滑问题(1)具有次线性收敛速率。为了提升 SGD 方法的性能，人们后续提出了许多改进方法。其中包括梯度聚合算法[2]，如 SAG [3] [4]和 SAGA [5]。他们将随机梯度计算为在之前迭代中评估的随机梯度的平均值。然后它们以牺牲内存为代价来存储之前的随机梯度。[3]表明 SAG 对于强凸问题是线性收敛的。Defazio 等[5]提出的 SAGA 方法是 SAG 的改进版本，它不需要强凸性假设。SVRG [6]有两个循环，外层循环计算一个完整的梯度，内层循环计算方差较小的随机梯度。S2GD [7]在每个历元中运行随机数的随机梯度，遵循几何规律。Batching SVRG [8]选择一个大的批样集来近似每个外环的全梯度。上述算法中的梯度估计量是无偏的。随机梯度递归算法(SARAH) [9]采用一种简单的递归框架来更新随机梯度估计。SARAH 算法结合了现有算法的一些优点，如 SAGA 和 SVRG，同时旨在改进这两种方法。特别是，SARAH 算法没有沿着随机梯度方向更新，而是沿着使用过去的随机梯度信息(如 SAGA)和偶尔的精确梯度信息(如 SVRG)的累积方向更新。SARAH 算法和 SVRG 算法基本相似，其不同的地方就在于在内循环。两者更新方式的差别如下所示：

SARAH 算法的关键步骤是随机梯度估计的递归更新(SARAH 更新)。

$$v_t = \nabla f_{i_t}\left(x_t\right) - \nabla f_{i_t}\left(x_{t-1}\right) + v_{t-1}, \tag{5}$$

其迭代更新为：

$$x_{t+1} = x_t - av_t. \tag{6}$$

SVRG 更新则是：

$$v_t = \nabla f_{i_t}\left(x_t\right) - \nabla f_{i_t}\left(x_0\right) + v_0. \tag{7}$$

近年来，Barzilai-Borwein (BB)方法越来越受到学者的关注。许多与 BB 算法相关的算法已经被提出。Tan 等[10]提出了 SGD-BB 和 SVRG-BB 方法。Liu 等[11]把重要抽样策略引入到 SARAH 方法中得到了 SARAH-I 算法，计算每个内循环最后一次迭代的全梯度。并且将 BB 算法与 SARAH-I 算法结合，得到了 SARAH-I-BB 算法。

BB 方法在求解非线性优化问题方面非常成功。BB 方法背后的关键思想是由拟牛顿方法激发的。假设我们要解决无约束最小化问题为：

$$\min_x f\left(x\right), \tag{8}$$

其中 f 是可微的。求解(8)的拟牛顿方法的典型迭代形式如下：

$$x_{t+1} = x_t - B_t^{-1}\nabla f\left(x_t\right), \tag{9}$$

其中 B_t 是 f 在当前迭代 x_t 下的 Hessian 矩阵的近似值。 B_t 最重要的特征是它必须满足割线方程：

$$B_t s_t = y_t, \tag{10}$$

BB 步长满足最小二乘意义上的某些正割方程，引入了以下长、短选择：

$$a_k^{BB1} = \arg\min_{a\in R}\left\|a^{-1}s_{k-1} - y_{k-1}\right\|_2 = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}} \tag{11}$$

以及

$$a_k^{BB2} = \arg\min_{a\in R}\left\|s_{k-1} - ay_{k-1}\right\|_2 = \frac{s_{k-1}^T s_{k-1}}{y_{k-1}^T y_{k-1}}, \tag{12}$$

其中 $g_k = \nabla f\left(x_k\right)$、 $s_{k-1} = x_k - x_{k-1}$ 和 $y_{k-1} = g_k - g_{k-1}$。BB 步长在二维严格凸二次型中带来了惊人的 R 超线性收敛。Raydan [12]通过配合 Grippo 等人[13]的非单调直线搜索，首先将 BB 方法扩展到无约束优化，得到了一种非常高效的梯度方法 GBB。Birgin 等[14]进一步扩展了 BB 法求解约束优化问题。Yuan [15] [16]建议计算步长，使前一步和后一步使用 SD 步长，从而在三次迭代中实现二维严格凸二次函数的最小化。Dai 和 Yuan [17]通过修改 Yuan 的步长，适当地与 SD 步长交替，提出了所谓的 Dai-Yuan (单调)梯度法，其性能甚至优于(非单调) BB 法。

Huang 等人[18]通过自适应地采用长 BB 步和与新步长相关的短步长结合，开发了一种有效的梯度方法，用于二次优化，使梯度法实现二维二次终止。本文将该步长与 SARAH-I 算法相结合，使其数值效果得到不错的提升。

本文的主要贡献如下：

在本文中，我们提出了一种新的算法，在 SARAH 算法中引入了能够自适应计算的步长。采用固定步长的 SARAH 算法对于初始步长的选取非常敏感，而新算法对于初始步长的选取并不敏感。此外，我们给出了新提出算法在强凸条件下的线性收敛性证明。数值结果证明了新算法的有效性，并表明该算法与在最佳步长调整情况下的 SARAH-I 算法数值性能相当。

本文的其余部分组织如下。在第 2 节中，我们提出了新的随机递归梯度算法。在第 3 节中，我们针对提出的算法，证明了它对强凸函数的线性收敛性。第 4 节中我们针对提出的算法进行了数值实验。最后，我们在第 5 节中得出了一些结论。

2. 新算法的提出

原始 SARAH [8]方法通过内环中与之前的 v_{t-1} 的分量梯度加减，递归地更新随机梯度步长 v_t。SARAH-I 算法考虑一般分布 $Q\sim\{q_1,q_2,\cdots,q_n\}$ 的随机抽样，这比 SARAH 中原来的均匀抽样方案更灵活。SARAH-I 算法伪代码如下表算法 1 所示。

算法 1 SARAH-I 算法

初始参数更新频率 m，初始步长 a_0，初始点 $\tilde{x}_0$

1: for $k=0,1,\cdots$ **do**

2: $x_0=\tilde{x}_{k-1}$

3: $v_0^k=\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(x_0)$

4: $x_1=x_0-av_0$

5: **for** $t=1,\cdots,m-1$ **do**

6: 根据 Q 随机选取 $i_t\in\{1,\cdots,n\}$

7: $v_t=\left(\nabla f_{i_t}(\omega_k)-\nabla f_{i_t}(\omega_{k-1})\right)/\left(nq_{i_t}\right)+v_{t-1}$

8: $x_{t+1}=x_t-a_kv_t$

9: **end for**

10: $\tilde{x}_k=x_m$

11: end for

步骤 6 表示的是随机抽样的方案。在 x_{t-1} 和 x_t 的条件下，我们得到了 v_t 关于 i_t 的期望：

$$\mathbb{E}\left[v_t\right]=\sum_{i=1}^{n}\frac{q_{i_t}}{nq_{i_t}}\left(\nabla f_{i_t}(x_t)-\nabla f_{i_t}(x_{t-1})\right)+v_{t-1}=\nabla f(x_t)-\nabla f(x_{t-1})+v_{t-1}. \tag{13}$$

因为 v_t 是 $\nabla f(x_t)$ 的无偏估计。在步骤 10 中，将 $\tilde{x}$ 设置为内部迭代的最后一次迭代，这是 SARAH-I 与原始 SARAH [8]最重要的区别。这似乎是更合理的选择，因为使用了每个内循环中的最新信息。

Huang 等人[18]提出的步长由下面给出。

$$\tilde{a}_k=\frac{2}{\frac{\phi_2}{\phi_3}+\sqrt{\left(\frac{\phi_2}{\phi_3}\right)^2-4\frac{\phi_1}{\phi_3}}} \tag{14}$$

其中：

$$\frac{\phi_1}{\phi_3}=\frac{a_{k-1}^{BB2}-a_k^{BB2}}{a_{k-1}^{BB2}a_k^{BB2}\left(a_{k-1}^{BB1}-a_k^{BB1}\right)}\text{ 以及 }\frac{\phi_2}{\phi_3}=\frac{a_{k-1}^{BB1}a_{k-1}^{BB2}-a_k^{BB1}a_k^{BB2}}{a_{k-1}^{BB2}a_k^{BB2}\left(a_{k-1}^{BB1}-a_k^{BB1}\right)}$$

下面的定理分别给出了步长(14)在 $\phi_1/\phi_3\geq 0$，$\phi_2\neq 0$ 时与 $\phi_1/\phi_3<0$，$\phi_2\neq 0$ 两种情况下的界，具体证明过程参见[18]。

定理 1 [18]：由(14)式给出的步长是良定的，并且当 $\phi_1/\phi_3\geq 0$ 与 $\phi_2\neq 0$ 时有：

$$\phi_3/\phi_2 \le \tilde{a}_k \le \min\left\{a_k^{BB2}, a_{k-1}^{BB2}\right\}; \tag{15}$$

当 $\phi_1/\phi_3 < 0$ 及 $\phi_2 \ne 0$，有

$$\max\left\{a_k^{BB2}, a_{k-1}^{BB2}\right\} \le \tilde{a}_k \le \left|\phi_3/\phi_2\right| \tag{16}$$

大量研究表明，采用长 BB 步长 a_k^{BB1} (因为 $a_k^{BB1} \ge a_k^{BB2}$)和一些短步长交替或自适应的方法在数值上优于原始 BB 方法。如果 $\phi_1/\phi_3 \ge 0$ 和 $\phi_2 \ne 0$，由定理 1 可以知道，步长 $\tilde{a}_k$ 小于 a_k^{BB2} 和 a_{k-1}^{BB2} 两个短步长。另一方面，当 $\phi_1/\phi_3 < 0$ 和 $\phi_2 \ne 0$ 时，步长 $\tilde{a}_k$ 都大于 a_k^{BB2} 和 a_{k-1}^{BB2} 两个步长。结合这两种情况，将 a_k 进行替换，替换为 $\max\left\{a_k^{BB2}, a_{k-1}^{BB2}, \tilde{a}_k\right\}$，替换后的步长为 a_k^{BB2}、a_{k-1}^{BB2}、及 $\tilde{a}_k$ 中最大的一个。

由于自适应方案的成功，Huang 等人[18]在其基础上提出步长的截断形式如下：

$$a_k = \begin{cases} \max\left\{a_{k-1}^{BB2}, a_k^{BB2}, \tilde{a}_k\right\}, & \text{if } a_k^{BB2}/a_k^{BB1} < \tau_k; \\ a_k^{BB1}, & \text{otherwise,} \end{cases} \tag{17}$$

其中 $\tau_k > 0$。更新(17)中的 τ_k 的最简单方法是将对所有 k 都设置为某个常数 $\tau \in (0,1)$。对于这种固定格式，步长(17)的性能可能在很大程度上取决于 τ 的值。另一个策略是动态地更新 τ_k，更新方式如下：

$$\tau_{k+1} = \begin{cases} \tau_k/\gamma, & \text{if } a_k^{BB2}/a_k^{BB1} < \tau_k; \\ \tau_k/\gamma, & \text{otherwise,} \end{cases} \tag{18}$$

显然，固定格式的 τ 是(18)中 $\gamma = 1$ 的一种特殊情况。

本文基于上述思想的提出，将 SARAH-I 算法和如(17)形式的 BB 类步长相结合。使原本采用固定步长的 SARAH-I 算法装备上能够自适应计算的步长(17)。

SARAH-I 算法引入了重要抽样原则，计算每个内循环最后一次迭代的全梯度。在求解强凸问题(1)时，Liu 等人[11]证明了非强凸优化 SARAH-I 的线性收敛性。在严格割线不等式下，证明了迭代到最优集的距离期望是线性收敛的。

Huang 等人[18]引入的步长 $\tilde{a}_k$ 如(17)所示。使得 BB 方法在加入 $\tilde{a}_k$ 后具有二维二次终止性。这种新颖的步长只利用了之前迭代的 BB 步长，因此可以很容易地用于一般的无约束和有约束优化。将步长(17)应用在 SARAH-I 算法中是非常具有潜力的。在后续的数值实验中我们可以发现新算法具有能够自动生成 SARAH-I 算法中最佳步长的能力，并且对于初始步长的选取很不敏感。

下面，我们提出新的算法方法，采用(17)式自适应计算步长。我们现在在下面的算法 2 中描述它的框架。

算法 2

初始参数 更新频率 m，初始步长 a_0，初始点 $\tilde{x}_0$

1: for $k = 0,1,\cdots$ **do**

2: $x_0 = \tilde{x}_{k-1}$

3: $v_0^k = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x_0)$

4: **if** $k > 1$ **then**

5: $a_k = \frac{1}{m}\tilde{a}_k$,

6: 其中 $\tilde{a}_k$ 由(17)给出。

7: **end if**

Continued

8: $x_1 = x_0 - a_k v_0^k$

9: **for** $t = 1, \cdots, m-1$ **do**

10: 随机选取 $i_t \in \{1, \cdots, n\}$.

11: $v_t = \left(\nabla f_{i_t}\left(x_k\right) - \nabla f_{i_t}\left(x_{k-1}\right)\right) \Big/ \left(nq_{i_t}\right) + v_{t-1}$

12: $x_{t+1} = x_t - a_k v_t$

13: end for

14: $\tilde{x}_k = x_m$

15: end for

3. 收敛性分析

为了验证新算法的收敛性。我们先引入 SARAH-I 算法在强凸条件下的收敛性结果。在此之前先引入如下的假设。

假设 1：每个分量函数 $f_i, i = 1, \cdots n$，是凸的且一阶连续可微的。以及每个分量函数 f_i 的梯度是 L-Lipschitz 连续的，即：对任意的 $x, x' \in R^d$ 有：

$$\left\|\nabla f_i\left(x\right) - \nabla f_i\left(x'\right)\right\|_2 \le L\left\|x - x'\right\|_2.$$

在此假设下，不难看出 $\nabla f\left(x\right)$ 也是 L-Lipschitz 连续的。为简单起见，我们将 L_Q 表示为：

$$L_Q = \max_i \frac{L}{nq_i}.$$

引理 1 是假设 1 成立下的结果。

引理 1：假设 f 是凸函数以及 ∇f 是 L-Lipschitz 连续的。则对任意的 $x, x' \in R^d$ 有：

$$\left(\nabla f\left(x\right) - \nabla f\left(x'\right)\right)^T \left(x - x'\right) \ge \frac{1}{L}\left\|\nabla f\left(x\right) - \nabla f\left(x'\right)\right\|^2.$$

现在给出 SARAH-I 算法的收敛性结果。

定理 2 [11]：假定假设 1 成立以及 $a \le 1/L_Q$。如果 f 是 μ 强凸的，则对于任意的 $k \ge 1$，当 $\sigma = \left(1 - 2\mu a\right)^m + a^{1/2} L_Q^{3/2} / \mu + a L_Q^3 \Big/ \left(\mu\right)^2 + a L_Q^2 / \left(2\mu\right)$. 有：

$$\mathbb{E}\left[\left\|\tilde{x}_{k+1} - x^*\right\|^2\right] \le \sigma\left\|\tilde{x}_k - x^*\right\|^2$$

因此，如果 m 和 a 的选择使得 $\sigma < 1$，则那么 SARAH-I 算法以期望线性收敛，即：

$$\mathbb{E}\left[\left\|\tilde{x}_k - x^*\right\|^2\right] \le \sigma^k\left\|\tilde{x}_0 - x^*\right\|^2$$

下面的定理表明，只要内部迭代次数 m 足够大，算法 2 具有线性收敛速率。假定 $\{\tilde{x}_k\}$ 是由算法 2 产生的序列。

定理 3：假定假设 1 成立以及 f 是 μ 强凸的。给定 $\theta = \left(1 - \mathrm{e}^{-2\mu/L_Q}\right)\Big/4$，如果 m 的选择使得 $m \ge L_Q^3 \Big/ \left(\theta^2 \mu^3\right)$，则对于任意的 $k \ge 1$ 有：

$$\mathbb{E}\left[\left\|\tilde{x}_k - x^*\right\|^2\right] \le \left(1 - \theta\right)^k\left\|\tilde{x}_0 - x^*\right\|^2.$$

证明：由步长(17)的形式可以很容易得出 $a_k \ge a_k^{BB2}$。由 $\nabla f\left(x\right)$ 的 Lipschitz 连续性可以得出：

$$a_k \geq a_k^{BB2} \geq \frac{\left\|v_0^k - v_0^{k-1}\right\|^2}{L\left\|v_0^k - v_0^{k-1}\right\|} = \frac{1}{L}.$$

同时由f的强凸性可以得到$a_k \leq 1/(m\mu)$因此，定理 3 中的系数σ有下界：

$$\begin{aligned}\sigma &= (1-2\mu a)^m + \frac{a^{1/2}L_Q^{3/2}}{\mu} + \frac{aL_Q^3}{\mu^2} + \frac{aL_Q^2}{2\mu} \\ &\leq \exp\left\{-\frac{2\mu}{L}\right\} + \frac{L_Q^{\frac{3}{2}}}{m^{\frac{1}{2}}\mu^{\frac{2}{3}}} + \frac{L_Q^3}{m\mu^3} + \frac{L_Q^2}{2m\mu^2} \\ &< 1-4\theta+\theta+\theta+\theta = 1-\theta.\end{aligned}$$

这就完成了我们的证明。

4. 数值实验

在本节中，我们进行了大量的实验来证明我们提出的算法的优势。我们分别在 LIBSVM[1] 网站上公开提供的 3 个不同的训练数据集上评估了我们的算法。这些数据集的详细信息如表 1 所示。注意，这五列分别表示数据集的名称、数据集的大小、特征的维度、应用的模型、正则化方法的系数。其中 LR 表示 logistic 回归，SVM 表示带有l_2范数正则化的平方铰链损失支持向量机。为了公平起见所有的数值实验都是在一台 CPU 为 i5-7300HQ 的笔记本电脑上，在 MATLAB 8.4 中实现的。

Table 1. Data and model information of the experiments
表 1. 实验数据和模型信息

数据集	n	d	模型	λ
w8a	49,749	300	LR	10^{-4}
a9a	37,561	123	LR	10^{-3}
ijcnn1	49,990	22	SVM	10^{-4}

新算法解决的两个问题如下：

带有l_2范数正则化的逻辑回归(LR)：

$$\min_x F(x) = \frac{1}{n}\sum_{i=1}^n \log\left[1+e^{\left(-b_i a_i^T x\right)}\right] + \frac{\lambda}{2}\|x\|_2^2, \tag{19}$$

和带有l_2范数正则化的平方铰链损失支持向量机(SVM)：

$$\min_x F(x) = \frac{1}{n}\sum_{i=1}^n \left(\left[1-b_i a_i^T x\right]_+\right)^2 + \frac{\lambda}{2}\|x\|_2^2, \tag{20}$$

下面我们给出算法 2 和 SARAH-I 算法在不同步长(对于算法 SARAH-I)和不同初始步长(对于算法 2)下的次优性$\left(f(\tilde{x}_k) - f(x^*)\right)$比较。这里我们设置内循环大小$m = n$。

在图 1~3 中，横坐标代表的是迭代次数，纵坐标代表的是次优性：$f(\tilde{x}_k) - f(x^*)$。

红色的虚线对应于每个图中具有最佳调优固定步长的 SARAH-I，黄色虚线代表较最调优步长稍小的步长，蓝色虚线代表较最调优步长稍大的步长。带*号的实线就是我们算法 2 的数值效果。观察图 1~3，我们可以看到 SARAH-I 算法对于三种不同大小步长的选取，其数值效果差别很大。但是算法 2 对于三种

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

不同大小初始步长的选取，其数值效果非常接近。所以算法 2 对初始步长的选择不敏感。

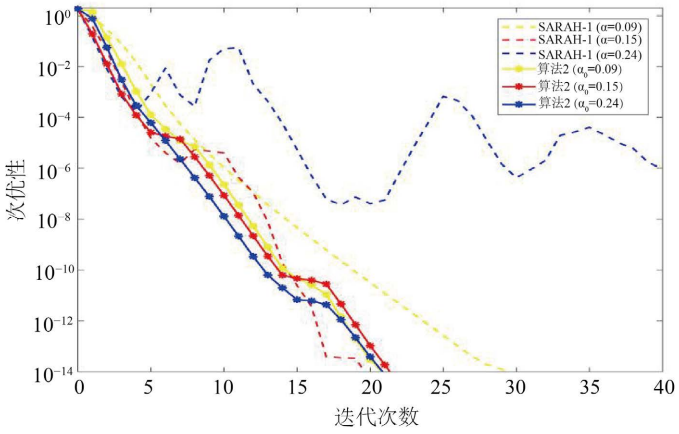

Figure 1. Sub-optimality comparison on w8a
图 1. w8a 上的次优性比较

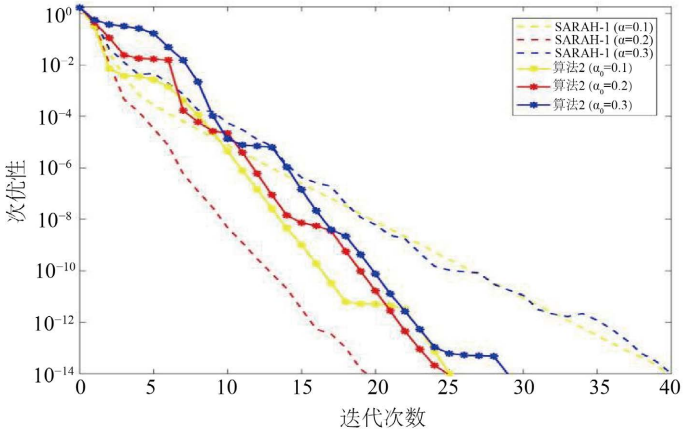

Figure 2. Sub-optimality comparison on a9a
图 2. a9a 上的次优性比较

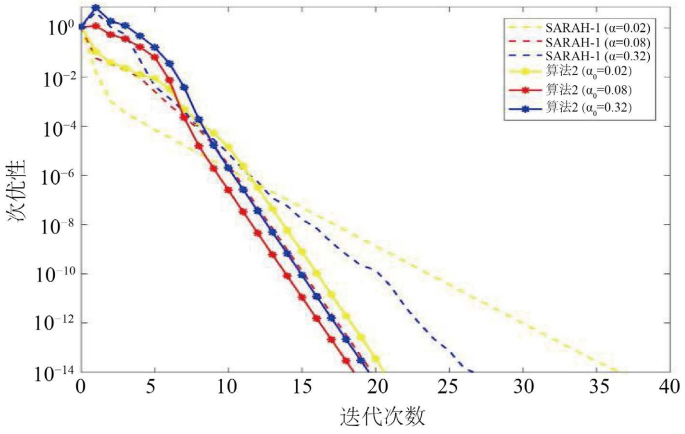

Figure 3. Sub-optimality comparison on ijcnn1
图 3. ijcnn1 上的次优性比较

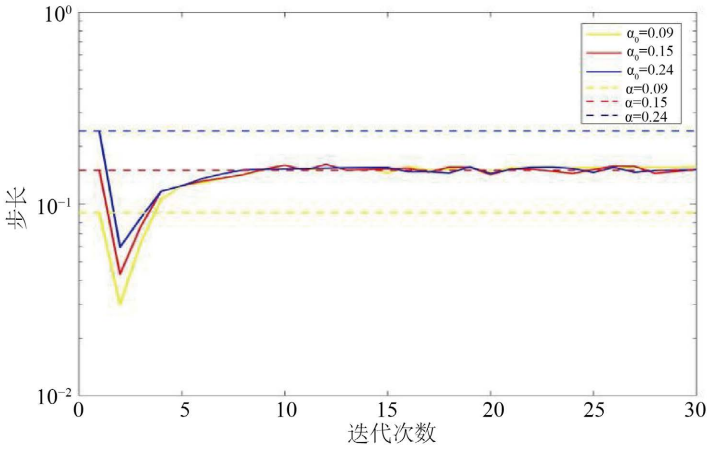

Figure 4. Stepsizes variation on w8a
图 4. w8a 上的步长变化

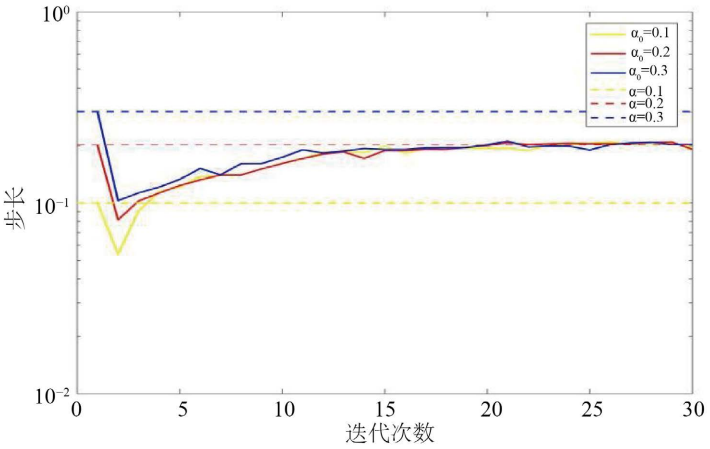

Figure 5. Stepsizes variation on a9a
图 5. a9a 上的步长变化

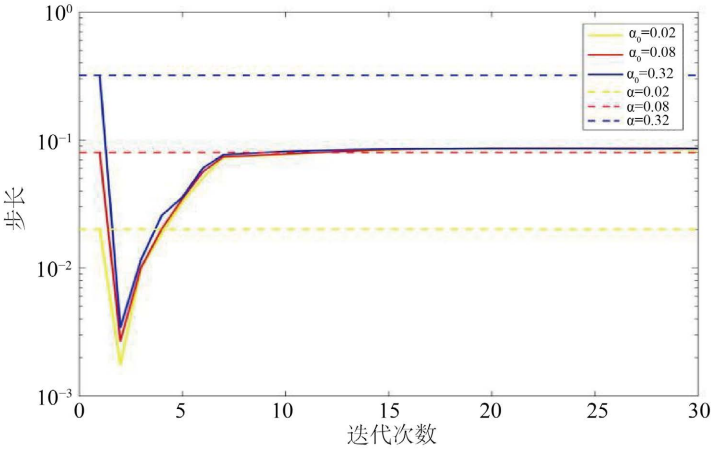

Figure 6. Stepsizes variation on ijcnn1
图 6. ijcnn1 上的步长变化

在图 4~6 中，横坐标代表的是迭代次数，纵坐标代表的是步长变化。其中三条虚线表示的是取不同固定步长下的 SARAH-I 算法。红色虚线表示最调优步长，黄色虚线是比最调优步长小的步长，蓝色虚线是比最调优步长大的步长。三条实线则表示不同初始步长下的算法 2 中的步长变化。从图 4~6 可以看出算法 2 中产生的步长最终将接近于 SARAH-I 算法中的最调优步长。因此新算法具有能自动生成最优步长的能力。

5. 结论

本文通过对 SARAH-I 算法采取自适应学习率的策略，引入了一种具有二维二次终止性的 BB 类步长。该步长自适应地采用长 BB 步和与该步长相关的短步长结合，具有较好的数值性能。并且我们在强凸条件下证明了算法 2 的线性收敛性。对于提出的新算法，我们在 3 个不同的训练数据集上进行了数值实验。实验结果发现，新算法能够达到与最具调优步长的 SARAH-I 算法相同的次优性程度，并且新算法对于初始步长的选取是非常不敏感的，而采用固定步长的 SARAH-I 算法对于不同步长的选取所得出的数值性能差异太大。最后通过对步长变化的观察，我们发现在算法 2 中产生的步长最终将接近于 SARAH-I 算法中的最调优步长。因此算法 2 具有能自动生成最优步长的能力，我们有理由相信新算法是鲁棒的。

参考文献

[1] Robbins, H. and Monro, S. (1951) A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, **22**, 400-407. https://doi.org/10.1214/aoms/1177729586

[2] Bottou, L., Curtis, F.E. and Nocedal, J. (2018) Optimization Methods for Large-Scale Machine Learning. *SIAM Review*, **60**, 223-311. https://doi.org/10.1137/16M1080173

[3] Roux, N.L., Schmidt, M. and Bach, F.R. (2013) A Stochastic Gradient Method with an Exponential Convergence Rate for Finite Training Sets. *Neural Information Processing Systems*, Lake Tahoe, 5 December 2013, 2663-2671.

[4] Schmidt, M.W., Roux, N.L. and Bach, F. (2017) Minimizing Finite Sums with the Stochastic Average Gradient. *Mathematical Programming*, **162**, 83-112. https://doi.org/10.1007/s10107-016-1030-6

[5] Defazio, A., Bach, F. and Lacoste-Julien, S. (2014) SAGA: A Fast Incremental Gradient Method with Support for Non-Strongly Convex Composite Objectives. *Neural Information Processing Systems*, Montreal, 8 December 2014, 1646-1654.

[6] Johnson, R. and Zhang, T. (2013) Accelerating Stochastic Gradient Descent Using Predictive Variance Reduction. *Neural Information Processing Systems*, Lake Tahoe, 5 December 2013, 315-323.

[7] Konecný, J. and Richtarik, P. (2017) Semi-Stochastic Gradient Descent Methods. *Frontiers in Applied Mathematics & Statistics*, **3**. https://doi.org/10.3389/fams.2017.00009

[8] Babanezhad, R., Ahmed, M.O., Virani, A., Schmidt, M.W., Konecný, J. and Sallinen, S. (2015) Stop Wasting My Gradients: Practical SVRG. Neural Information Processing Systems, Montreal, 7 December 2015, 2251-2259.

[9] Nguyen, L.M., Liu, J., Scheinberg, K. and Takac, M. (2017) SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient. *Neural Information Processing Systems*, Long Beach, 4 December 2017, 2613-2621.

[10] Tan, C., Ma, S., Dai, Y.H. and Qian, Y. (2016) Barzilai-Borwein Step Size for Stochastic Gradient Descent. *Neural Information Processing Systems*, Barcelona, 5 December 2016, 685-693.

[11] Liu, Y., Wang, X. and Guo, T. (2020) A Linearly Convergent Stochastic Recursive Gradient Method for Convex Optimization. *Optimization Letters*, **14**, 2265-2283. https://doi.org/10.1007/s11590-020-01550-x

[12] Raydan, M. (1997) The Barzilai and Borwein Gradient Method for the Large Scale Unconstrained Minimization Problem. *SIAM Journal on Optimization*, **7**, 26-33. https://doi.org/10.1137/S1052623494266365

[13] Grippo, L. and Lucidi, F.L. (1986) A Nonmonotone Line Search Technique for Newton's Method. *SIAM Journal on Numerical Analysis*, **23**, 707-716. https://doi.org/10.1137/0723046

[14] Birgin, E.G., Martínez, J.M. and Raydan, M. (2000) Nonmonotone Spectral Projected Gradient Methods on Convex Sets. *SIAM Journal on Optimization*, **10**, 1196-1211. https://doi.org/10.1137/S1052623497330963

[15] Yuan, Y.X. (2006) A New Stepsize for the Steepest Descent Method. *Journal of Computational Mathematics*, **24**, 149-156.

[16] Yuan, Y. (2008) Step-Sizes for the Gradient Method. AMS/IP Studies in Advanced Mathematics, 785-796. https://doi.org/10.1090/amsip/042.2/23

[17] Dai, Y. and Yuan, Y.X. (2017) Analysis of Monotone Gradient Methods. *Journal of Industrial & Management Optimization*, **1**, 181-192. https://doi.org/10.3934/jimo.2005.1.181

[18] Huang, Y., Dai, Y.H. and Liu, X.W. (2021) Equipping Barzilai-Borwein Method with Two Dimensional Quadratic Termination Property. *SIAM Journal on Optimization*, **31**, 3068-3069. https://doi.org/10.1137/21M1390785