一类随机方差缩减算法的分析与改进

刘彦¹, 郭田德^{1,2}, 韩丛英^{1,2*}

1. 中国科学院大学数学科学学院, 北京 100049;

2. 中国科学院大数据挖掘与知识管理重点实验室, 北京 100049

E-mail: liuyan23ucas@outlook.com, tdguo@ucas.ac.cn, hancy@ucas.ac.cn

收稿日期: 2019-12-21; 接受日期: 2020-04-08; 网络出版日期: 2020-09-21; * 通信作者

国家自然科学基金 (批准号: 11731013, 11571014, 11991022 和 U19B2040) 资助项目

摘要 近年来, 随机方差缩减类算法在求解机器学习中的大规模优化问题时得到了广泛应用. 但是如何选择此类算法的合适步长依然是值得研究的问题. 受启发于结合 Barzilai-Borwein 步长的随机方差缩减梯度 (stochastic variance reduced gradient with Barzilai-Borwein step size, SVRG-BB) 算法, 本文针对方差缩减类算法提出基于局部 Lipschitz 常数估计的自适应步长, 并通过构建一个极小极大化问题给出该步长应用于不同算法时的参数选取方法. 然后将该步长与随机递归梯度算法 (stochastic recursive gradient algorithm, SARAH) 和随机方差缩减 (stochastic variance reduced gradient, SVRG) 算法相结合, 分别提出结合自适应步长的随机递归梯度 (SARAH with adaptive step size, SARAH-AS) 方法和结合自适应步长的随机方差缩减梯度 (SVRG with adaptive step size, SVRG-AS) 算法, 并且在强凸假设下证明以上算法点距离序列的线性收敛性质. 此外, 本文还提供一个新颖的视角揭示为什么 SARAH+ 算法是有效的. 在公开数据集上的数值实验结果表明本文提出的自适应步长在方差缩减类算法中表现良好.

关键词 随机方差缩减类算法 自适应步长 线性收敛率

MSC (2020) 主题分类 90C06, 90C25, 68W40

1 引言

在大规模机器学习中, 如下式的优化问题常常出现:

$$\min_{w \in \mathbb{R}^d} P(w) = \frac{1}{n} \sum_{i=1}^n f_i(w), \quad (1.1)$$

其中 n 代表样本数目, d 为变量维度, 本文假设每一个 f_i 都是一阶连续可微的, w_* 表示问题最优解.

英文引用格式: Liu Y, Guo T D, Han C Y. Analysis and improvement for a class of variance reduced methods (in Chinese). Sci Sin Math, 2021, 51: 1433–1450, doi: 10.1360/SCM-2019-0795

对于问题 (1.1), 当 n 较大时, 传统的优化算法会在计算上很难容忍, 甚至有时候是不能执行的. 以梯度下降法为例,

$$w_{k+1} = w_k - \eta_k \cdot \frac{1}{n} \sum_{i=1}^n \nabla f_i(w_k),$$

即要计算 n 个分量函数梯度才能迭代一次. Robbins 和 Monro^[1] 在 1951 年提出了开创性工作: 随机近似 (stochastic approximation, SA). 在此基础上, 研究者提出了随机梯度下降 (stochastic gradient descent, SGD)^[2-5], 该方法每次迭代仅借助一个或者部分样本确定迭代方向. 每次仅选取一个样本计算梯度时的迭代公式如下:

$$w_{k+1} = w_k - \eta_k \nabla f_{i_k}(w_k), \quad (1.2)$$

其中 $\nabla f_{i_k}(w_k)$ 代表第 i_k 个分量函数在 w_k 的梯度, $\eta_k > 0$ 表示步长.

SGD 在机器学习中应用广泛, 有一系列改进 SGD 的工作以提升性能^[4-6]. SGD 算法的收敛性质取决于随机方向和真实梯度的方差, 因此, 缩减随机梯度的方差至关重要. 缩减随机梯度方差的方法整体可以分为 3 类: (1) 梯度聚合方法^[5]; (2) 动态采样; (3) 采样方法.

梯度聚合方法, 又称为随机方差缩减类方法, 通常通过重新使用和/或修改先前计算的梯度信息来缩减随机梯度的方差. 梯度聚合有两种实现方法, 一种需要存储每个分量函数最近计算的梯度 (需内存), 如随机平均梯度 (stochastic average gradient, SAG) 算法^[7,8] 和 SAGA 算法^[9]; 另一种是在迭代的过程中每隔几步计算一次全梯度 (需计算力), 如随机对偶坐标上升 (stochastic dual coordinate ascent, SDCA) 算法^[10,11]、SVRG^[12]、半随机梯度下降 (semi-stochastic gradient descent, S2GD) 算法^[13] 和 SARAH 算法^[14] 等. 还有一些它们的变体算法^[15-21]. 以上这些算法均可以在强凸条件下线性收敛, 并且还与二阶信息相结合求解问题 (1.1), 如 VITE^[22]、随机有限内存 BFGS (Broyden, Fletcher, Goldfarb, Shanno) 算法^[23] 和随机块 BFGS 算法^[24] 等, 均是 SVRG 与拟 Newton 方法相结合.

动态采样通过逐渐增加样本集大小缩减方差, 其主要目标是选择足够多的样本使得当前迭代方向尽可能为下降方向. 具体来说, 假设抽取 $|S_k|$ 个样本使得算法达到线性收敛速度, 必须要求随机梯度的方差指数阶下降至 0, 所以要求 $|S_k|$ 的大小指数倍增加. 研究者提出了范数测试^[25] 和内积测试^[26]. 范数测试设计了一个度量样本集 S_k 大小的不等式, 可以验证当 $d = -\nabla P_{S_k}(w_k)$ 满足下式时是一个下降方向:

$$\mathbb{E}[\|\nabla P(w_k) - \nabla P_{S_k}(w_k)\|^2] \leq \theta^2 \|\nabla P_{S_k}(w_k)\|^2, \quad \theta \in [0, 1). \quad (1.3)$$

内积测试^[26] 需要 $|S_k|$ 足够大以满足下式:

$$\mathbb{E}[(\nabla P_{S_k}(w_k)^T \nabla P(w_k) - \|\nabla P(w_k)\|^2)^2] \leq \theta^2 \|\nabla P(w_k)\|^4, \quad \theta > 0. \quad (1.4)$$

经典的随机梯度下降方法通常采用均匀采样, 这种采样方法的好处是迭代方向为目标函数负梯度方向的无偏估计, 但是可能会包含较高的误差影响收敛过程. 采样方法是指利用数据本身的一些性质采取非均匀采样方法, 一般与聚类算法有关. 经典的有重要采样^[27,28]、分层采样^[29] 和任意采样^[30] 等. 还有一些工作将上述缩减方差的方法互相结合, 如动态采样与梯度聚合方法结合^[31], 采样方法与梯度聚合方法互相结合^[32-34] 等.

对 SGD 而言, 另一个关键因素是步长的选取. 随机梯度类算法, 需要依据近似信息做决策, 造成在传统方法中最经典的线搜索技术无法使用. 因此, 此类方法常根据人工经验选择符合某种规律的递减

步长或者较小的固定步长^[5]. Robbins 和 Monro^[1] 指出步长应该满足 $\sum_{k=1}^{\infty} \eta_k = \infty$ 和 $\sum_{k=1}^{\infty} \eta_k^2 < \infty$. 关于步长的工作也有很多, AdaGrad^[35] 和 Adam^[36] 等算法利用对角修正技术为每个分量自适应的选取步长. SVRG-BB^[37] 结合 BB (Barzilai-Borwein) 算法^[38–40] 与 SVRG, 可以自动地计算步长. 大批量 SGD^[41] 在每次迭代中, 计算二阶近似的最优步长, 也利用了 BB 方法. 随机自适应 BFGS^[42] 算法提出了基于局部曲率信息^[43] 的自适应步长. 本文为方差缩减类算法提出基于局部 Lipschitz 常数的自适应步长.

本文工作主要有以下几方面:

- (1) 结合范数测试分析自适应内循环 SARAH+ 算法有效的原因.
- (2) 针对方差缩减类算法提出了基于局部 Lipschitz 常数估计的自适应步长.
- (3) 给出针对不同算法选取步长参数的框架, 即求解极小极大化问题. 将自适应步长与 SARAH 算法、SVRG 算法相结合, 分别得到 SARAH-AS 和 SVRG-AS 算法, 并在强凸条件下证明线性收敛性质.
- (4) 对 SARAH-AS 和 SVRG-AS 算法进行求解逻辑回归问题的数值实验. 实验结果表明, 对于方差缩减类算法, 自适应步长 (adaptive step size, AS) 要优于 BB 步长.

本文余下内容安排如下. 第 2 节简要介绍随机方差缩减类算法: SARAH 和 SVRG 算法; 第 3 节分析 SARAH+ 算法有效的原因; 第 4 节详细介绍如何构建自适应步长, 并且将步长应用于 SARAH 和 SVRG 算法, 同时分析算法的线性收敛性质; 数值实验结果在第 5 节; 第 6 节给出本文结论.

2 随机方差缩减类算法

以 SARAH 和 SVRG 算法为代表的随机方差缩减类算法通常在外循环计算全梯度 (真实梯度), 在内循环计算方差缩减的随机梯度. SARAH 算法流程如算法 1 所示, SVRG 算法流程如算法 2 所示.

算法 1 SARAH 算法

参数: 内循环次数 m , 步长 $\eta > 0$, 初始点 $\tilde{w}_0$.

```

1: for  $k = 1, 2, \dots$  do
2:    $v_0 = \frac{1}{n} \sum_{i=1}^n \nabla f_i(\tilde{w}_{k-1})$ ;
3:    $w_1 = w_0 - \eta v_0$ ;
4:   for  $t = 1, \dots, m-1$  do
5:     随机选取  $i_t \in \{1, \dots, n\}$ ;
6:      $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$ ;
7:      $w_{t+1} = w_t - \eta v_t$ ;
8:   end for
9:   选取  $\tilde{w}_k$ .
10: end for
```

对于算法 1 和 2 的第 9 行, 有两种方法选取 $\tilde{w}_k$.

选取方法 I: $\tilde{w}_k = w_m$.

选取方法 II: 令 $\tilde{w}_k = w_i$, 其中 i 从 $\{1, 2, \dots, m\}$ 中随机选取.

SARAH 算法最关键的步骤是在内循环对随机梯度近似地递归更新, 即

$$v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}, \quad (2.1)$$

算法 2 SVRG 算法

参数: 内循环次数 m , 步长 $\eta > 0$, 初始点 $\tilde{w}_0$.

```

1: for  $k = 1, 2, \dots$  do
2:    $w_0 = \tilde{w}_{k-1}$ ;
3:    $v_k = \frac{1}{n} \sum_{i=1}^n \nabla f_i(w_0)$ ;
4:   for  $t = 0, \dots, m-1$  do
5:     随机选取  $i_t \in \{1, \dots, n\}$ ;
6:      $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}_k) + v_k$ ;
7:      $w_{t+1} = w_t - \eta v_t$ ;
8:   end for
9:   选取  $\tilde{w}_k$ .
10: end for
    
```

其中 i_t 从 $\{1, 2, \dots, n\}$ 中随机选取. 值得注意的是, SARAH 算法的迭代方向 v_t 是真实梯度的有偏估计, 即 $E[v_t] = \nabla P(w_t) - \nabla P(w_{t-1}) + v_{t-1} \neq \nabla P(w_t)$.

为分析算法收敛性质, 本文做出以下假设:

假设 2.1 假设目标函数 $P(w)$ 是 μ -强凸的, 即

$$P(w) \geq P(w') + \nabla P(w')^T(w - w') + \frac{\mu}{2} \|w - w'\|^2, \quad \forall w, w' \in \mathbb{R}^d.$$

假设每一个分量函数 f_i 是凸函数并且梯度 L -Lipschitz 连续, 即

$$\|\nabla f_i(w) - \nabla f_i(w')\|_2 \leq L \|w - w'\|_2, \quad \forall w, w' \in \mathbb{R}^d.$$

在此假设条件下, 易知 $\nabla P(w)$ 也是 L -Lipschitz 连续的,

$$\|\nabla P(w) - \nabla P(w')\|_2 \leq L \|w - w'\|_2, \quad \forall w, w' \in \mathbb{R}^d.$$

原始 SARAH 算法^[14] 采用选取方法 II, 而本文采用选取方法 I. 非精确 SARAH (inexact SARAH, iSARAH)^[44] 算法为降低计算复杂度, 在外循环计算批次梯度而非真实梯度, 并且在内循环选择小的批次计算随机梯度. 文献 [19] 研究了 SARAH 算法在目标函数非凸情形下的收敛性质. ProxSARAH^[45] 算法考虑分量函数为非凸, 甚至非光滑函数时的机器学习任务, 主要包含两个步骤: 一个近端梯度步和平均步, 这与现有的近端梯度类算法均不同. SARAH-I^[34] 也同样采用了选取方法 I, 并且结合了重要采样. 更进一步, SARAH-I-BB 算法^[34] 利用 BB 方法自动计算步长, 该算法采取 η_1^{BB} 和 η_2^{BB} 的凸组合计算步长, 即 $\eta_1^{\text{BB}} = \frac{\|s_k\|^2}{s_k^T y_k}$ 和 $\eta_2^{\text{BB}} = \frac{s_k^T y_k}{\|y_k\|^2}$, 其中 $s_k = \tilde{w}_k - \tilde{w}_{k-1}$, $y_k = v_k - v_{k-1}$. 并且该文献给出了 SARAH-I 算法当前迭代点至最优点的点距离序列的线性收敛性质如下:

定理 2.1 假设每一个 f_i ($i = 1, 2, \dots, n$) 都是凸函数, 且梯度 L -Lipschitz 连续, 并且目标函数 P 是 μ -强凸的, 那么对于所有的 $k \geq 1$, 有 $E[\|\tilde{w}_{k+1} - w^*\|^2] \leq \sigma \|\tilde{w}_k - w^*\|^2$, 其中

$$\sigma := (1 - 2\mu\eta)^m + \frac{\eta^{1/2} L^{3/2}}{\mu} + \frac{\eta L^3}{\mu^2} + \frac{\eta L^2}{2\mu}. \quad (2.2)$$

相应地, 如果选取 m 和 $\eta \leq 1/L$ 使得 $\sigma < 1$, 那么 SARAH-I 算法在期望意义下线性收敛, 即

$$E[\|\tilde{w}_k - w^*\|^2] \leq \sigma^k \|\tilde{w}_0 - w^*\|^2.$$

不同于 SARAH 算法, SVRG 算法是无偏的. 通常称采用选取方法 I 的 SVRG 方法为 SVRG-I 算法. SVRG-BB 算法^[37] 是第一个将 BB 方法^[39] 结合 SGD 算法的工作, 该算法采用选取方法 I, 在外循环计算步长, 内循环更新规则如下:

$$w_{t+1} = w_t - \frac{1}{m} \cdot \eta_1^{\text{BB}} (\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}_k) + v_k)$$

或者

$$w_{t+1} = w_t - \frac{1}{m} \cdot \eta_2^{\text{BB}} (\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}_k) + v_k).$$

SVRG-I 算法的线性收敛性质如下定理:

定理 2.2 假设每个分量函数 f_i 都是凸函数, 并且梯度均满足 Lipschitz 连续性质. 假设目标函数 P 是 μ -强凸的, 如果选取 m 和 η 使得

$$\alpha := (1 - 2\eta\mu(1 - \eta L))^m + \frac{4\eta L^2}{\mu(1 - \eta L)} < 1,$$

那么 SVRG-I 在期望意义下线性收敛:

$$\mathbb{E} \|\tilde{w}_k - w_*\|_2^2 < \alpha^k \|\tilde{w}_0 - w_*\|_2^2.$$

3 SARAH 算法分析

随机方差缩减类算法的一个缺点在于实际表现对内循环次数 m 的选取较为敏感, 研究发现 m 过大或过小均会造成不良影响, 引起在最优点附近有振荡现象, 导致不必要的计算冗余. 原始 SARAH 算法的作者认为当 $\|v_t\|^2$ 较小时, 更新则毫无意义^[46]. 考虑到强凸条件下对于每个内循环迭代, 当 $t \rightarrow \infty$ 时, $\mathbb{E}[\|v_t\|^2] \rightarrow 0$, 基于此提出了 SARAH+ 算法 (算法 3), 即当满足 $\|v_t\|^2 \leq \gamma \|v_0\|^2$ 时, 内循环停止. SARAH+ 算法基于经验建议选取 $\gamma = \frac{1}{8}$, 但并无理论分析. SARAH++ 算法^[46] 选择了同样的内循环停止准则, 只是该算法选取参数 $\gamma \geq L\eta$, 并且给出了外循环的次数限制.

算法 3 SARAH+ 算法

参数: 最大内循环次数 M , 步长 $\eta > 0$, $0 < \gamma \leq 1$, 初始点 $\tilde{w}_0$.

```

1: for  $k = 1, 2, \dots$  do
2:    $v_0 = \frac{1}{n} \sum_{i=1}^n \nabla f_i(\tilde{w}_{k-1});$ 
3:    $w_1 = w_0 - \eta v_0;$ 
4:    $t = 1;$ 
5:   while  $\|v_{t-1}\|^2 > \gamma \|v_0\|^2$  and  $t < M$  do
6:      $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$ , 其中  $i_t \in \{1, \dots, n\}$  为随机选取;
7:      $w_{t+1} = w_t - \eta v_t;$ 
8:      $t = t + 1;$ 
9:   end while
10:   $\tilde{w}_k = w_t.$ 
11: end for
```

本节提供了一个新颖的视角解释为什么 SARAH+ 算法在强凸条件下是有效的. 首先给出一些引理.

引理 3.1 (参见文献 [14, 定理 1a]) 假设每个分量函数 f_i 是凸函数且梯度 L -Lipschitz 连续, 目标函数 P 是 μ -强凸的. 考虑 SARAH 算法, 当 $\eta < 2/L$ 时, 对于任意的 $t > 1$, 有

$$\mathbb{E}[\|v_t\|^2] \leq \varphi \mathbb{E}[\|v_{t-1}\|^2] \leq \varphi^t \mathbb{E}[\|\nabla P(w_0)\|^2],$$

其中 $\varphi = 1 - (\frac{2}{\eta L} - 1)\mu^2\eta^2$.

引理 3.2 (参见文献 [14, 引理 2]) 假设每个分量函数 f_i 梯度 L -Lipschitz 连续, 考虑 SARAH 算法, 对于任意的 $t > 1$, 有

$$\mathbb{E}[\|\nabla P(w_t) - v_t\|^2] = \sum_{j=1}^t \mathbb{E}[\|v_j - v_{j-1}\|^2] - \sum_{j=1}^t \mathbb{E}[\|\nabla P(w_j) - \nabla P(w_{j-1})\|^2].$$

引理 3.3 (参见文献 [44, 引理 3]) 假设每个分量函数 f_i 梯度 L -Lipschitz 连续, 考虑 SARAH 算法, 对于任意的 $t > 1$, 有

$$\mathbb{E}[\|\nabla P(w_t) - v_t\|^2] \leq L^2\eta^2 \sum_{j=1}^t \mathbb{E}[\|v_{j-1}\|^2].$$

SARAH 算法是真实梯度的有偏估计, 并且当满足强凸条件时, $\mathbb{E}[\|v_t\|^2]$ 在内循环线性收敛, 如引理 3.1 所示. 但是引理 3.2 表明 $\mathbb{E}[\|\nabla P(w_t) - v_t\|^2]$ 随着内循环的进行有递增的趋势, 这意味着向量 v_t 距离真实梯度 $\nabla P(w_t)$ 越来越远, 即 v_t 不是下降方向的可能性随着 $t \rightarrow \infty$ 越来越大. 因此, 算法应该在早期就停止内循环使得 v_t 与 $\nabla P(w_t)$ 在期望意义下足够接近, 而不是一味地让 $\|v_t\|^2$ 逐渐减小.

受启发于范数测试 (1.3), 为使随机方向 v_t 尽可能地为下降方向, 可以令 v_t 满足下述不等式:

$$\mathbb{E}[\|\nabla P(w_t) - v_t\|^2] \leq \theta^2 \|v_t\|^2, \quad \text{其中 } \theta \in (0, 1]. \quad (3.1)$$

从上式易知, 当 v_t 满足 (3.1), 则在期望意义下为下降方向. 下面分析为什么 SARAH+ 算法在实际应用中是有效且合理的. 由引理 3.3 可知,

$$\mathbb{E}[\|\nabla P(w_t) - v_t\|^2] \leq L^2\eta^2 \sum_{j=1}^t \mathbb{E}[\|v_{j-1}\|^2].$$

如果每个分量函数 f_i 的梯度 L -Lipschitz 连续, 则由引理 3.1 可以得到

$$L^2\eta^2 \sum_{j=1}^t \mathbb{E}[\|v_{j-1}\|^2] \leq \frac{L^2\eta^2}{1-\varphi} \|v_0\|^2.$$

令

$$\frac{L^2\eta^2}{1-\varphi} \|v_0\|^2 \leq \theta^2 \|v_t\|^2, \quad \theta \in [0, 1).$$

进一步对上式化简可以得到

$$\hat{\theta}^2 \|v_0\|^2 \leq \|v_t\|^2, \quad (3.2)$$

其中 $\hat{\theta} = \frac{L^2\eta^2}{(1-\varphi)\theta^2}$. 如果令 (3.2) 中 $\gamma = \hat{\theta}$, 那么就得到了 SARAH+ 算法的停止准则. 考虑到 $1-\varphi < 1$ 并且 $\theta < 1$, 令 $\frac{L\eta}{(1-\varphi)\theta^2} \geq 1$, 则 $\gamma \geq L\eta$, 这就是 SARAH++ 算法的内循环停止准则.

然而, 在不增加存储和计算成本的情形下, 很难根据 (3.1) 为 SARAH 类算法构造新的内循环停止准则. 以上分析无法应用于 SVRG 算法, 因为 SVRG 算法在内循环并不具有线性收敛性质.

4 自适应步长

从 SARAH-I 算法和 SVRG-I 算法的收敛性分析 (即定理 2.1 和 2.2), 可以看出, 步长的选取与 Lipschitz 常数 L 密切相关. 然而在一般的优化问题中, L 是无法以先验知识得到的, 因此, 只能采用局部 Lipschitz 估计^[47] 去刻画步长. 结合自适应步长的最速下降 (steepest descent with adaptive step size, SDAS-2)^[48] 算法根据一对相邻的迭代点 w_k 和 w_{k-1} , 可通过下式计算 Lipschitz 常数的局部估计:

$$\Lambda_k = \frac{\|\nabla P(w_k) - \nabla P(w_{k-1})\|}{\|w_k - w_{k-1}\|}. \quad (4.1)$$

则最速下降法结合该步长为

$$w_{k+1} = w_k - \frac{1}{2\Lambda_k} \nabla P(w_k), \quad k = 0, 1, 2, \dots$$

借鉴于 SDAS-2^[48] 算法, 本文提出基于 Lipschitz 常数估计的自适应步长. 考虑到对于 SGD, 真实函数值与梯度值无法获取, 借鉴 SVRG-BB^[37] 算法和 SARAH-I-BB^[34] 算法思想, 在外循环估计局部 Lipschitz 常数, 然后用于计算步长.

自适应步长计算流程如算法 4 所示.

算法 4 计算自适应步长

输入: $\Lambda_i, \beta_i > 0, i \in I_k, c \geq 1, h$.

1: **if** $k = 1$ **then**

2: $L_1 = \Lambda_1$;

3: **else if** $k > 1$ **then**

4: $L_k = \sum_i^k \beta_i \Lambda_i$;

5: **end if**

6: 计算 $\eta_k = \frac{1}{m^h L_k}$;

输出: η_k .

下面详细介绍自适应步长构造方法. 受启发于 Adam^[36] 算法, 对最新的 c 个局部估计, 进行滑动平均, 即

$$L_k = \sum_{i \in I_k} \beta_i \Lambda_i,$$

其中 $I_k := \max\{k - c + 1, 1\}, \dots, k$, 对于所有 $i \in I_k, \beta_i > 0$ 并且 $\sum_{i \in I_k} \beta_i = 1$. 在随机优化中, 研究者经常选取递减步长或者较小的常数步长. SVRG-BB^[37] 算法和 SARAH-I-BB^[34] 算法在应用中将原 BB 步长估计除以内循环次数 m , 因为在内循环从 w_0 到 w_m 需要 m 个无偏梯度估计更新. 结合 BB 步长的小批次半随机梯度下降 (minibatch semi-stochastic gradient descent with Barzilai-Borwein step size, mS2GD-BB)^[49] 算法与此类似, 将原 BB 步长除以 m/b , 其中 b 为计算该步长的批次样本集大小. 与上述两种方法不同的是, 我们构建了一个简单函数 h , 通过下式计算步长:

$$\eta_k := \frac{1}{m^h \cdot L_k}.$$

函数 h 有不同的形式:

$$h = 1 \quad \text{或} \quad h = g(k). \quad (4.2)$$

众所周知, 在早期迭代中, 可以选取较大步长加速收敛, 然后逐渐选择较小步长防止振荡. 因此, 当选取 $h = h(k)$ 时, 可以令 $h(k)$ 是关于外循环数 k 的单调递增函数. 数值实验中, 针对 SARAH-AS 算法和 SVRG-AS 算法, 分别选取函数 $h \in [3/4, 1)$ 和 $h \in [2/3, 1)$, 第 4.3 小节给出做如此选择的原因. 具体形式如下:

$$g(k) = \frac{k}{1/3 + k} \quad \text{和} \quad g(k) = \frac{k}{1/2 + k}. \quad (4.3)$$

实际应用中, 选取方法 I 要优于选取方法 II, 因此, 本文将自适应步长与随机方差缩减类算法相结合均采用选取方法 I, 但同样适用于选取方法 II.

4.1 SARAH-AS 算法与 SVRG-AS 算法

算法 5 SARAH-AS 算法

参数: 内循环次数 m , 初始步长 $\eta_1 > 0$, 初始点 $\tilde{w}_0$.

```

1: for  $k = 1, 2, \dots$  do
2:    $v_0^k = \frac{1}{n} \sum_{i=1}^n \nabla f_i(\tilde{w}_{k-1})$ ;
3:   if  $k > 1$  then
4:     计算并存储  $\Lambda_k = \frac{\|v_0^k - v_0^{k-1}\|}{\|\tilde{w}_k - \tilde{w}_{k-1}\|}$ ;
5:   end if
6:   根据算法 4 计算  $\eta_k$ ;
7:    $w_1 = w_0 - \eta_k v_0^k$ ;
8:   for  $t = 1, \dots, m-1$  do
9:     随机选取  $i_t \in \{1, \dots, n\}$ ;
10:     $v_t^k = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}^k$ ;
11:     $w_{t+1} = w_t - \eta_k v_t^k$ ;
12:   end for
13:    $\tilde{w}_k = w_m$ .
14: end for
```

算法 6 SVRG-AS 算法

参数: 内循环次数 m , 初始步长 $\eta_1 > 0$, 初始点 $\tilde{w}_0$.

```

1: for  $k = 1, 2, \dots$  do
2:    $v_k = \frac{1}{n} \sum_{i=1}^n \nabla f_i(\tilde{w}_{k-1})$ ;
3:   if  $k > 1$  then
4:     计算存储  $\Lambda_k = \frac{\|v_k - v_{k-1}\|}{\|\tilde{w}_k - \tilde{w}_{k-1}\|}$ ;
5:   end if
6:   根据算法 4 计算  $\eta_k$ ;
7:    $w_0 = \tilde{w}_k$ ;
8:   for  $t = 0, \dots, m-1$  do
9:     随机选取  $i_t \in \{1, \dots, n\}$ ;
10:     $w_{t+1} = w_t - \eta_k (\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}_k) + v_k)$ ;
11:   end for
12:    $\tilde{w}_k = w_m$ .
13: end for
```

本节结合自适应步长与 SARAH 和 SVRG 算法分别提出 SARAH-AS 算法 (算法 5) 和 SVRG-AS 算法 (算法 6). 没有结合 SARAH+ 算法的原因在于, 一方面, SARAH+ 算法与 SARAH 算法相比表

现相差不大, 且有时候略差; 另一方面, 自适应步长依赖于内循环次数 m , 但 SARAH+ 算法每一个周期的 m 均不同, 可能会造成不良影响.

4.2 收敛性分析

下面首先分析 SARAH-AS 算法选择不同的函数 h 时的收敛性质. 当 SARAH-AS 算法选择 $h = 1$ 时, 有如下分析:

定理 4.1 给定 $\theta_1 = (1 - e^{-2\mu/L})/4$, 显然, $\theta_1 \in (0, 1/4)$. 如果选取 m 使得

$$m \geq \frac{L^3}{\theta_1^2 \mu^3},$$

那么由 SARAH-AS 算法产生的点距离序列在期望意义下线性收敛:

$$\mathbb{E}[\|\tilde{w}_k - w_*\|^2] \leq (1 - \theta_1)^k \|\tilde{w}_0 - w_*\|^2.$$

证明 根据目标函数 $P(w)$ 的强凸性质可知,

$$L_k = \sum_i \beta_i \frac{\|\nabla P(w_i) - \nabla P(w_{i-1})\|}{\|w_i - w_{i-1}\|} \geq \sum_i \beta_i \mu = \mu, \quad \eta_k = \frac{1}{m \cdot L_k} \leq \frac{1}{m\mu}.$$

类似地, 由 $\nabla P(w)$ 的 Lipschitz 连续性质, 可以得到 $\eta_k \geq 1/(mL)$, 因此有 $\frac{1}{mL} \leq \eta_k \leq \frac{1}{m\mu}$, 则根据定理 2.1 可知, (2.2) 中参数 σ 有上界,

$$\begin{aligned} \sigma &= (1 - 2\mu\eta)^m + \frac{\eta^{1/2}L^{3/2}}{\mu} + \frac{\eta L^3}{\mu^2} + \frac{\eta L^2}{2\mu} \\ &\leq \exp\left\{-\frac{2\mu}{mL} \cdot m\right\} + \frac{L^{3/2}}{m^{1/2}\mu^{3/2}} + \frac{L^3}{m\mu^3} + \frac{L^2}{2m\mu^2} \\ &\leq \exp\left\{-\frac{2\mu}{L}\right\} + \frac{L^{3/2}}{m^{1/2}\mu^{3/2}} + \frac{L^3}{m\mu^3} + \frac{L^2}{2m\mu^2} \\ &< 1 - 4\theta_1 + \theta_1 + \theta_1 + \theta_1 \\ &= 1 - \theta_1. \end{aligned}$$

结论得证. □

接下来分析当 SARAH-AS 选择 $h = g(k)$ 时产生的点距离序列的收敛性质.

定理 4.2 给定 $0 < \theta_2 < 1/4$, 对于 SARAH-AS 算法, 当选择 $h = g(k)$ 时, 如果有关 m 的表达式分别满足下式:

$$m^h \geq \frac{L^3}{\theta_2^2 \mu^3} \quad \text{和} \quad m^{1-h} \geq \frac{-\log(1 - 4\theta_2)L}{2\mu}, \quad (4.4)$$

那么 SARAH-AS 算法产生的点距离序列在期望意义下线性收敛:

$$\mathbb{E}[\|\tilde{w}_k - w_*\|_2^2] < (1 - \theta_2)^k \|\tilde{w}_0 - w_*\|_2^2.$$

证明 易知 $\frac{1}{m^h L} \leq \eta \leq \frac{1}{m^h \mu}$, 根据定理 2.1, 有

$$\sigma = (1 - 2\mu\eta)^m + \frac{\eta^{1/2}L^{3/2}}{\mu} + \frac{\eta L^3}{\mu^2} + \frac{\eta L^2}{2\mu}$$

$$\begin{aligned}
 &\leq \exp \left\{ -\frac{2\mu}{m^h L} \cdot m \right\} + \frac{L^{3/2}}{m^{h/2} \mu^{3/2}} + \frac{L^3}{m^h \mu^3} + \frac{L^2}{2m^h \mu^2} \\
 &\leq \exp \left\{ -\frac{2m^{1-h} \mu}{L} \right\} + \frac{L^{3/2}}{m^{h/2} \mu^{3/2}} + \frac{L^3}{m^h \mu^3} + \frac{L^2}{2m^h \mu^2} \\
 &< 1 - 4\theta_2 + \theta_2 + \theta_2 + \theta_2 \\
 &= 1 - \theta_2.
 \end{aligned}$$

结论得证. $\square$

接着给出 SVRG-AS 算法选取不同的 h 时的收敛性分析, 同样根据定理 2.2 分析由算法产生的点距离序列的线性收敛性质.

定理 4.3 给定 $\theta_3 = (1 - e^{-\mu/L})/2$. 当 SVRG-AS 算法选择 $h = 1$ 时, 如果选取 m 满足

$$m \geq \frac{4L^2}{\theta_3 \mu^2} + \frac{L}{\mu},$$

那么 SVRG-AS 算法在期望意义下线性收敛:

$$\mathbb{E} \|\tilde{w}_k - w_*\|_2^2 < (1 - \theta_3)^k \|\tilde{w}_0 - w_*\|_2^2.$$

证明 易知 $\frac{1}{mL} \leq \eta \leq \frac{1}{m\mu}$, 则根据定理 2.2, 有

$$\begin{aligned}
 \alpha &\leq \left[1 - \frac{2\mu}{mL} \left(1 - \frac{L}{m\mu} \right) \right]^m + \frac{4L^2}{m\mu^2[1 - L/(m\mu)]} \\
 &\leq \exp \left\{ -\frac{2\mu}{L} \left(1 - \frac{L}{m\mu} \right) \right\} + \frac{4L^2}{m\mu^2[1 - L/(m\mu)]} \\
 &\leq \exp \left\{ -\frac{2\mu}{L} \left(1 - \frac{L}{m\mu} \right) \right\} + \frac{4L^2}{m\mu^2 - L\mu} \\
 &< \exp \left\{ -\frac{\mu}{L} \right\} + \frac{4L^2}{m\mu^2 - L\mu} \\
 &\leq 1 - 2\theta_3 + \theta_3 = 1 - \theta_3.
 \end{aligned}$$

证毕. $\square$

定理 4.4 给定 $\theta_4 \in (0, 1/2)$. 当 SVRG-AS 算法选择 $h = g(k)$, 如果选取 m 满足

$$m^h \geq \frac{4L^2}{\theta_4 \mu^2} + \frac{L}{\mu} \quad \text{和} \quad m^{1-h} \geq \frac{-\log(1 - 2\theta_4)L}{\mu}, \quad (4.5)$$

那么 SVRG-AS 算法在期望意义下线性收敛:

$$\mathbb{E} \|\tilde{w}_k - w_*\|_2^2 < (1 - \theta_4)^k \|\tilde{w}_0 - w_*\|_2^2.$$

证明 易知 $\frac{1}{mL} \leq \eta \leq \frac{1}{m\mu}$, 根据定理 2.2, 有

$$\begin{aligned}
 \alpha &\leq \left[1 - \frac{2\mu}{m^h L} \left(1 - \frac{L}{m^h \mu} \right) \right]^m + \frac{4L^2}{m^h \mu^2[1 - L/(m^h \mu)]} \\
 &\leq \exp \left\{ -\frac{2m^{1-h} \mu}{L} \left(1 - \frac{L}{m^h \mu} \right) \right\} + \frac{4L^2}{m^h \mu^2[1 - L/(m^h \mu)]}
 \end{aligned}$$

$$\begin{aligned}
&\leq \exp \left\{ -\frac{2m^{1-h}\mu}{L} \left(1 - \frac{L}{m^h\mu} \right) \right\} + \frac{4L^2}{m^h\mu^2 - L\mu} \\
&< \exp \left\{ -\frac{m^{1-h}\mu}{L} \right\} + \frac{4L^2}{m^h\mu^2 - L\mu} \\
&\leq 1 - 2\theta_4 + \theta_4 = 1 - \theta_4.
\end{aligned}$$

定理得证. $\square$

4.3 参数选取的极小极大问题

第 4.2 小节分别分析了 SARAH-AS 和 SVRG-AS 算法收敛时内循环次数 m 需要满足的条件, 那么当 SARAH-AS 和 SVRG-AS 算法选择 $h = g(k)$ 时, 如何获取 m 的下界呢? 根据定理 4.2 和 4.4, 有

$$m^h \geq F(\theta) \quad \text{和} \quad m^{1-h} \geq G(\theta),$$

则上述问题可以转化为一个极小极大问题:

$$\arg \min_h \max \{ F(\theta)^{\frac{1}{h}}, G(\theta)^{\frac{1}{1-h}} \}. \quad (4.6)$$

对于 SARAH-AS 算法, 根据定理 4.2, 有

$$F(\theta) = \frac{L^3}{\theta^2\mu^3} \quad \text{和} \quad G(\theta) = \frac{-\log(1-4\theta)L}{2\mu}.$$

很容易看出, 在上述定理中 $F(\theta) > 1$ 均成立, 因此, 求解极小极大问题的最优解时可以根据 $G(\theta)$ 的值是否大于 1 分为两种情形.

从图 1 很可以容易分析出极小极大化问题 (4.6) 的解为

$$h = \begin{cases} 1, & G(\theta) \leq 1, \\ \frac{\ln F(\theta)}{\ln F(\theta) + \ln G(\theta)}, & G(\theta) > 1. \end{cases} \quad (4.7)$$

图 1 (网络版彩图) 两种情形. (a) $G(\theta) \leq 1$; (b) $G(\theta) > 1$

下面给出 SARAH-AS 算法点距离序列线性收敛时的推论.

推论 4.1 对于 SARAH-AS 算法, 当 $0 < \theta \leq (1 - e^{-2\mu/L})/4$ 时, 应该选择 $h = 1$, 为了加快收敛, 即当 $(1 - e^{-2\mu/L})/4 < \theta < 1/4$ 时, h 应该选择为单调递增函数并且 $h \in [3/4, 1)$.

证明 定义 $\theta_a = (1 - e^{-2a\mu/L})/4$, 根据定理 4.2, 有

$$F(\theta) = \frac{L^3}{\theta_a^2 \mu^3}, \quad G(\theta) = a.$$

当 $a \in (0, 1]$ 时, $0 < \theta_a \leq (1 - e^{-2\mu/L})/4$, $G(\theta) \leq 1$. 在这种情形下, 极小极大化问题 (4.6) 的解为 $h = 1$, 如 (4.7) 所示. 固定 $h = 1$, 容易看出 m 的下界随着 θ 增加而递减, 因此, 当 $h = 1$ 时, 最佳选择为 $\theta = (1 - e^{-2\mu/L})/4$, 即正如定理 4.1 所示, 内循环次数 m 需要满足下式:

$$m \geq \frac{L^3}{\theta^2 \mu^3}.$$

在这种情形下, $m \sim \mathcal{O}(\kappa^3/\theta^2)$.

当 $a \in (1, \kappa]$ 时, $(1 - e^{-2\mu/L})/4 < \theta_a < 1/4$, 并且 $G(\theta) > 1$, 解为

$$h = \frac{\ln F(\theta)}{\ln F(\theta) + \ln G(\theta)}.$$

从图 1(b) 可以看出 $g(\theta)$ 越小越好. 当 $a \rightarrow 1$ 时, $h \approx 1$, $m \sim \mathcal{O}(\kappa^3/\theta_a^2)$. 但是当 a 接近于 κ 时, $F(\theta) \sim \mathcal{O}(\kappa^3)$, $G(\theta) \sim \mathcal{O}(\kappa)$, 意味着 $h \approx 3/4$ 并且 $m \sim \mathcal{O}(\kappa^4)$.

考虑到在迭代初期, 较大步长可以加速收敛, 并且步长通常是递减的; 另一方面想要保持 m 在可接受范围内的同时使得 θ 尽可能大. 因此, 选取 $h \in [3/4, 1)$ 并且关于 k 单调递增, 其中 k 代表第 k 个外循环. $\square$

当根据函数值至最优值序列收敛性质分析时, $h \in [2/3, 1)$ 为单调递增函数, 证明过程类似, 故省略.

与 SARAH-AS 算法类似, 当 SVRG-AS 算法选择 $h = G(k)$ 时仍旧需要求解极小极大化问题 (4.6). 根据定理 4.4, 此时

$$F(\theta) = \frac{4L^2}{\theta\mu^2} + \frac{L}{\mu} \quad \text{和} \quad G(\theta) = \frac{-\log(1 - 2\theta)L}{\mu}.$$

推论 4.2 对于 SVRG-AS 算法, 当 $0 < \theta \leq (1 - e^{-\mu/L})/2$ 时, 应选择 $h = 1$. 为加速收敛, 即当 $(1 - e^{-\mu/L})/2 < \theta < 1/2$ 时, 给定 $h \in [2/3, 1)$ 且为单调递增函数.

证明 定义 $\theta_b = (1 - e^{-b\mu/L})/2$, 那么

$$F(\theta) = \frac{4L^2}{\theta_b\mu^2} + \frac{L}{\mu}, \quad G(\theta) = b.$$

当 $b \in (0, 1]$ 时, $0 < \theta \leq (1 - e^{-\mu/L})/2$, $G(\theta) \leq 1$. 在这种情形下, 极小极大化问题 (4.6) 的解为 $h = 1$, 如 (4.7) 所示. 很明显如定理 4.3, 当 $\theta = (1 - e^{-\mu/L})/2$ 时, $h = 1$ 为最佳选择. 此时,

$$m \geq \frac{4L^2}{\theta\mu^2} + \frac{L}{\mu}.$$

这种情形下, $m \sim \mathcal{O}(\kappa^2/\theta^2)$.

当 $b \in (1, \kappa]$ 时, $(1 - e^{-\mu/L})/2 < \theta < 1/2$, $g(\theta) > 1$, 问题解为

$$h = \frac{\ln F(\theta)}{\ln F(\theta) + \ln G(\theta)}.$$

图 1(b) 表明 $G(\theta)$ 越小越好. 当 $b \rightarrow 1$ 时, $h \approx 1$, $m \sim \mathcal{O}(\kappa^2/\theta_b^2)$. 但是当 b 接近于 κ 时, $F(\theta) \sim \mathcal{O}(\kappa^2)$, $G(\theta) \sim \mathcal{O}(\kappa)$, 这时 $h \approx 2/3$ 并且 $m \sim \mathcal{O}(\kappa^3)$.

很容易看出 h 随着 θ 的增加而下降. 为了加速收敛, 对于 SVRG-AS 算法, 应使得 $h \in [2/3, 1)$ 并且关于 k 单调递增. $\square$

5 数值实验结果

本节将 SARAH-AS 算法与 SVRG-AS 算法应用于求解二分类问题的 ℓ_2 正则逻辑回归模型给出实验结果, 模型分量函数如下:

$$f_i(w) = \log(1 + \exp(-y_i x_i^T w)) + \frac{\lambda}{2} \|w\|^2,$$

其中 $x_i \in \mathbb{R}^d$, $y_i \in \{-1, 1\}$ 为给定数据. 所有的实验实现环境均为 MATLAB 9.1, i7-6700K CPU. 本文选择公开数据集 Libsvmdata¹⁾ 中以下数据集 ijcnn1、real-sim 和 rcv1, 如表 1 所示.

接下来的图表中, x 坐标轴均代表周期数目, 即外循环数, 而 y 轴代表最优间隔 (optimality gap) 的值, 即 $P(\tilde{w}_k) - P(w_*)$, 其中 w_* 由运行 SVRG-BB 算法获得. 在所有数据集上设置参数为 $\lambda = 1/n$, $m = n$, 对于 SARAH-AS 算法和 SVRG-AS 算法, 分别选取函数 h 具体形式如 (4.3), 并且选择简单移动平均 (simple moving average, SMA) 和指数移动平均 (exponential moving average, EMA) 处理最新的 c 个局部 Lipschitz 常数估计, 即

$$\text{SMA: } L_k = \frac{1}{c} \sum_{i=k-c+1}^k \Lambda_i,$$

$$\text{EMA: } L_k = \rho[\Lambda_k + (1 - \rho)\Lambda_{k-1} + \cdots + (1 - \rho)^{c-1}\Lambda_{k-c+1}] + (1 - \rho)^c \Lambda_k.$$

在所有图中, 实线代表自适应步长 (包括 BB 步长) 的方差缩减类算法, 虚线对应着固定步长.

图 2 和 3 测试了自适应步长的有效性. 图 2 中, 橙色、红色和紫色实线分别对应着 SARAH-I-BB、SARAH-AS ($h = 1$) 和 SARAH-AS ($h = g(k)$) 算法. 在图 2(a) 和 2(d) 中, 淡蓝、深蓝和绿色虚线分别对应 $\eta = 0.01$ 、 $\eta = 0.1$ 和 $\eta = 1$; 在图 2(b) 和 2(e) 中, 淡蓝、深蓝和绿色虚线分别对应 $\eta = 0.3$ 、 $\eta = 1.2$ 和 $\eta = 4.8$; 在图 2(c) 和 2(f) 中, 淡蓝、深蓝和绿色虚线分别对应 $\eta = 0.88$ 、 $\eta = 2.2$ 和 $\eta = 5.5$, 其中深蓝色虚线一直代表最优固定步长的 SARAH 算法. 选择 $\eta_1 = 0.1$ 、 $\eta_1 = 1.2$ 和 $\eta_1 = 2.2$

表 1 数据集信息

数据集	样本数目 n	维度 d
ijcnn1	49,990	22
real-sim	72,309	20,958
rcv1	20,242	47,236

1) 数据集网址: <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

图 2 (网络版彩图) SARAH-AS 算法、SARAH-I-BB 算法和 SARAH-I 算法固定步长的比较

图 3 (网络版彩图) SVRG-AS、SVRG-BB 算法和带有固定步长的 SVRG 算法在不同数据集上的表现比较

分别为数据集 `ijcnn1`、`real-sim` 和 `rcv1` 的初始步长. 事实上, 不同的初始步长对于我们提出的自适应步长影响较小.

从图 2(a)–2(c) 中可以看出, $\text{SARAH-AS } (h = g(k))$ 要比 $\text{SARAH-AS } (h = 1)$ 和 SARAH-I-BB 表现良好, 并且可以媲美最优固定步长的 SARAH 算法. 从图 2(d)–2(f) 中可以看出, 我们的自适应步长逐渐稳定, 并且 $\text{SARAH-AS } (h = g(k))$ 的步长在初期要大于 SARAH-I-BB 的步长, 以此加速收敛. SARAH-I-BB 和 $\text{SARAH-AS } (h = 1)$ 算法计算的步长收敛于最优固定步长邻域, 而 $\text{SARAH-AS } (h = g(k))$ 的步长可以收敛于最优步长. 总之, SARAH-AS 算法在实际应用中有令人满意的表现.

图 3 中, 橙色、红色和紫色实线分别对应着 SVRG-BB 、 $\text{SVRG-AS } (h = 1)$ 和 $\text{SVRG-AS } (h = g(k))$ 算法. 在图 3(a) 和 3(d) 中, 淡蓝色、深蓝和绿色虚线分别对应 $\eta = 0.01$ 、 $\eta = 0.1$ 和 $\eta = 1$; 在图 3(b) 和 3(e) 中, 淡蓝色、深蓝和绿色虚线分别对应 $\eta = 0.3$ 、 $\eta = 1.2$ 和 $\eta = 4.8$; 在图 3(c) 和 3(f) 中, 淡蓝色、深蓝和绿色虚线分别对应 $\eta = 0.8$ 、 $\eta = 2$ 和 $\eta = 5$. 选择 $\eta_1 = 0.1$ 、 $\eta_1 = 1.2$ 和 $\eta_1 = 2$ 分别作为数据集 `ijcnn1`、`real-sim` 和 `rcv1` 的初始步长. 深蓝色虚线一直代表最优固定步长的 SVRG 算法.

从图 3(a)–3(c) 中可以看出, $\text{SVRG-AS } (h = g(k))$ 要比 $\text{SVRG-AS } (h = 1)$ 和 SVRG-BB 表现良好, 并且可以媲美最优固定步长的 SVRG 算法. 从图 3(d)–3(f) 中可以看出, 我们的自适应步长逐渐稳定, 并且 $\text{SVRG-AS } (h = g(k))$ 的步长在初期要大于 SVRG-BB 的步长, 以此加速收敛. SVRG-BB 和 $\text{SVRG-AS } (h = 1)$ 算法计算的步长收敛于最优固定步长邻域, 而 $\text{SVRG-AS } (h = g(k))$ 的步长可以收敛于最优步长. 总之, SVRG-AS 算法表现良好, 并且性能可以媲美最优固定步长的 SVRG 算法.

从图 2 和 3 可以看出自适应步长选择 $h = g(k)$ 时的表现优于 $h = 1$. 为方便比较不同的移动平均技术对于自适应步长的影响, 图 4 中的步长参数统一选择 $h = g(k)$. 图 4 展示了实验结果. 选取参

图 4 (网络版彩图) 不同移动平均策略 SMA 和 EMA 分别在 SARAH-AS 和 SVRG-AS 算法中的比较

数 $c = 2$, EMA 的衰减率为 $\rho = 0.9$. 所有的子图中 y 轴代表最优间隔, 上面图中展示了 SMA 和 EMA 对 SARAH-AS 算法的影响; 而下面图展示 SMA 和 EMA 在 SVRG-AS 算法中的效果. 图 4(a)–4(c) 中的橙色虚线表示带有固定步长的 SARAH 算法, 步长分别为 $\eta = 0.1$ 、 $\eta = 1.2$ 和 $\eta = 2.2$, 蓝色、红色、紫色实线分别对应 SARAH ($h = g(k)$)、SARAH ($h = g(k)$, SMA) 和 SARAH ($h = g(k)$, EMA) 算法; 图 4(d)–4(f) 中的橙色虚线表示带有固定步长的 SVRG 算法, 步长分别为 $\eta = 0.1$ 、 $\eta = 1.2$ 和 $\eta = 2$, 蓝色、红色、紫色实线分别对应 SVRG ($h = g(k)$)、SVRG ($h = g(k)$, SMA) 和 SVRG ($h = g(k)$, EMA) 算法. 图 4 结果表示滑动平均技术对于自适应步长性能提升效果有限. EMA 要比 SMA 更加适合, 但也作用有限.

6 结论

本文揭示了 SARAH+ 算法有效的原因, 但是很难据此给出新的内循环停止准则. 然后提出适用于方差缩减类算法的自适应步长, 该步长基于局部 Lipschitz 常数估计, 并结合了移动平均技术. 然后通过构建极小极大化问题分析步长参数选取, 不同的算法或者分析方法可以得到不同的参数. 结合 SARAH 和 SVRG 算法分别得到了 SARAH-AS 和 SVRG-AS 算法, 并从理论方面分析了两个算法的线性收敛性质. 从实验结果来看, 自适应步长策略在实际应用中表现良好, 但我们并没有分析步长参数的最优选择, 如 $g(k)$ 的最佳表达式和移动平均的最优形式, 这是接下来的工作. 我们还将考虑如何为方差缩减类算法构建一个通用的内循环停止准则.

致谢 感谢三位匿名审稿专家提出的宝贵意见, 使我们的论文更加完善.

参考文献

- 1 Robbins H, Monro S. A stochastic approximation method. *Ann Math Statist*, 1951, 22: 400–407
- 2 Bottou L, Bousquet O. The tradeoffs of large scale learning. In: *Proceedings of the 20th International Conference on Neural Information Processing*. Vancouver: Curran Associates, 2007, 161–168
- 3 Bottou L. Stochastic gradient learning in neural networks. In: *Proceedings of Neuro-Nimes*. Nimes: EC2, 1991, 12
- 4 Ding F, Yang H Z, Liu F. Performance analysis of stochastic gradient algorithms under weak conditions. *Sci China Ser F-Inf Sci*, 2008, 51: 1269–1280
- 5 Bottou L, Curtis F E, Nocedal J. Optimization methods for large-scale machine learning. *SIAM Rev*, 2018, 60: 223–311
- 6 Gu J, Xiao X T. A framework of convergence analysis of mini-batch stochastic projected gradient methods. *J Oper Res Soc China*, 2019, 22: 1–23
- 7 Roux N L, Schmidt M, Bach F R. A stochastic gradient method with an exponential convergence rate for finite training sets. In: *Proceedings of Neural Information Processing Systems*. Nevada: Curran Associates, 2012, 2663–2671
- 8 Schmidt M, Le Roux N, Bach F. Minimizing finite sums with the stochastic average gradient. *Math Program*, 2017, 162: 83–112
- 9 Defazio A, Bach F, Lacoste-Julien S. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In: *Proceedings of Neural Information Processing Systems*. Montreal: Curran Associates, 2014, 1646–1654
- 10 Shalev-Shwartz S, Zhang T. Stochastic dual coordinate ascent methods for regularized loss minimization. *J Mach Learn Res*, 2013, 14: 567–599
- 11 Richtárik P, Takáč M. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Math Program*, 2014, 144: 1–38
- 12 Johnson R, Zhang T. Accelerating stochastic gradient descent using predictive variance reduction. In: *Proceedings of Neural Information Processing Systems*. Nevada: Curran Associates, 2013, 315–323
- 13 Konečný J, Richtárik P. Semi-stochastic gradient descent methods. *Front Appl Math Stat*, 2017, 3: 9
- 14 Nguyen L M, Liu J, Scheinberg K, et al. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In: *Proceedings of the 34th International Conference on Machine Learning*. Sydney: PMLR, 2017, 2613–2621

- 15 Xiao L, Zhang T. A proximal stochastic gradient method with progressive variance reduction. *SIAM J Optim*, 2014, 24: 2057–2075
- 16 Allen-Zhu Z, Yuan Y. Improved SVRG for non-strongly-convex or sum-of-non-convex objectives. In: *Proceedings of the 33rd International Conference on Machine Learning*. New York: PMLR, 2016, 1080–1089
- 17 Reddi S J, Hefny A, Sra S, et al. Stochastic variance reduction for nonconvex optimization. In: *Proceedings of the 33rd International Conference on Machine Learning*. New York: PMLR, 2016, 314–323
- 18 Lei L, Jordan M. Less than a single pass: Stochastically controlled stochastic gradient. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Fort Lauderdale: PMLR, 2017, 148–156
- 19 Nguyen L M, Liu J, Scheinberg K, et al. Stochastic recursive gradient algorithm for nonconvex optimization. *arXiv:1705.07261*, 2017
- 20 Zhou D, Xu P, Gu Q. Stochastic nested variance reduction for nonconvex optimization. In: *Proceedings of the 32nd International Conference on Neural Information Processing*. Montreal: Curran Associates, 2018, 3925–3936
- 21 Fang C, Li C J, Lin Z, et al. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In: *Proceedings of Neural Information Processing Systems*. Montreal: Curran Associates, 2018, 689–699
- 22 Lucchi A, McWilliams B, Hofmann T. A variance reduced stochastic Newton method. *arXiv:1503.08316*, 2015
- 23 Moritz P, Nishihara R, Jordan M. A linearly-convergent stochastic L-BFGS algorithm. In: *Proceedings of Artificial Intelligence and Statistics*. Cadiz: PMLR, 2016, 249–258
- 24 Gower R, Goldfarb D, Richtárik P. Stochastic block BFGS: Squeezing more curvature out of data. In: *Proceedings of International Conference on Machine Learning*. New York: PMLR, 2016, 1869–1878
- 25 Byrd R H, Chin G M, Nocedal J, et al. Sample size selection in optimization methods for machine learning. *Math Program*, 2012, 134: 127–155
- 26 Bollapragada R, Byrd R, Nocedal J. Adaptive sampling strategies for stochastic optimization. *SIAM J Optim*, 2018, 28: 3312–3343
- 27 Needell D, Ward R, Srebro N. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. In: *Proceedings of Neural Information Processing Systems*. Montreal: Curran Associates, 2014, 1017–1025
- 28 Zhao P, Zhang T. Stochastic optimization with importance sampling for regularized loss minimization. In: *Proceedings of International Conference on Machine Learning*. Lille: PMLR, 2015, 1–9
- 29 Zhao P, Zhang T. Accelerating minibatch stochastic gradient descent using stratified sampling. *arXiv:1405.3080*, 2014
- 30 Qu Z, Richtárik P, Zhang T. Randomized dual coordinate ascent with arbitrary sampling. *arXiv:1411.5873*, 2014
- 31 Jalilzadeh A, Shanbhag U V, Blanchet J H, et al. Optimal smoothed variable sample-size accelerated proximal methods for structured nonsmooth stochastic convex programs. *arXiv:1803.00718*, 2018
- 32 Horváth S, Richtárik P. Nonconvex variance reduced optimization with arbitrary sampling. *arXiv:1809.04146*, 2018
- 33 Qian X, Qu Z, Richtárik P. SAGA with arbitrary sampling. *arXiv:1901.08669*, 2019
- 34 Liu Y, Wang X, Guo T. A linearly convergent stochastic recursive gradient method for convex optimization. *Optim Lett*, 2020, 14: 2265–2283
- 35 Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. *J Mach Learn Res*, 2011, 12: 2121–2159
- 36 Kingma D P, Ba J. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014
- 37 Tan C, Ma S, Dai Y H, et al. Barzilai-Borwein step size for stochastic gradient descent. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Barcelona: Curran Associates, 2016, 685–693
- 38 Dai Y H, Liao L Z. R -linear convergence of the Barzilai and Borwein gradient method. *IMA J Numer Anal*, 2002, 22: 1–10
- 39 Barzilai J, Borwein J M. Two-point step size gradient methods. *IMA J Numer Anal*, 1988, 8: 141–148
- 40 Raydan M. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J Optim*, 1997, 7: 26–33
- 41 De S, Yadav A, Jacobs D, et al. Automated inference with adaptive batches. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. Fort Lauderdale: PMLR, 2017, 1504–1513
- 42 Zhou C, Gao W, Goldfarb D. Stochastic adaptive quasi-Newton methods for minimizing expected values. In: *Proceedings of the 34th International Conference on Machine Learning*. Sydney: PMLR, 2017, 4150–4159
- 43 Gao W, Goldfarb D. Quasi-Newton methods: Superlinear convergence without line searches for self-concordant functions. *Optim Methods Softw*, 2019, 34: 194–217
- 44 Nguyen L M, Scheinberg K, Takáč M. Inexact SARAH algorithm for stochastic optimization. *arXiv:1811.10105*, 2018
- 45 Pham N H, Nguyen L M, Phan D T, et al. ProxSARAH: An efficient algorithmic framework for stochastic composite nonconvex optimization. *arXiv:1902.05679*, 2019
- 46 Nguyen L M, Van Dijk M, Phan D T, et al. Finite-sum smooth optimization with SARAH. *arXiv:1901.07648*, 2019

- 47 Armijo L. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific J Math*, 1966, 16: 1–3
- 48 Vrahatis M N, Androulakis G S, Lambrinos J N, et al. A class of gradient unconstrained minimization algorithms with adaptive stepsize. *J Comput Appl Math*, 2000, 114: 367–386
- 49 Yang Z, Wang C, Zang Y, et al. Mini-batch algorithms with Barzilai-Borwein update step. *Neurocomputing*, 2018, 314: 177–185

Analysis and improvement for a class of variance reduced methods

Yan Liu, Tiande Guo & Congying Han

Abstract Stochastic variance reduced methods have recently surged into prominence for solving large scale optimization problems in machine learning. However, how to choose the step sizes is still a problem to work out. Inspired by SVRG-BB (stochastic variance reduced gradient with Barzilai-Borwein step size), we propose an adaptive step size which is based on local estimation of Lipschitz constant for variance reduced methods. A framework was given of how to select the crucial parameter for different algorithms in our step size by solving a minimax problem. Then we adapt this step size to SARAH (stochastic recursive gradient algorithm) and SVRG (stochastic variance reduced gradient), which leads to two algorithms SARAH-AS (SARAH with adaptive step size) and SVRG-AS (SVRG with adaptive step size), respectively. Both of them converge linearly in the strongly convex case. Furthermore, we provide a novel perspective to explore why SARAH+ performs well in practice. Numerical experiments on standard datasets demonstrate the efficiency of our adaptive step size for stochastic variance reduced methods.

Keywords stochastic variance reduced methods, adaptive step size, linear convergence rate

MSC(2020) 90C06, 90C25, 68W40

doi: 10.1360/SCM-2019-0795