运筹学与控制论

DOI:10.11721/cqnuj20240511

一种赋有 BB 类步长的新随机方差缩减梯度算法*

陈炫睿，刘泽显，倪　艳
(贵州大学 数学与统计学院，贵阳 550025)

摘要:为随机方差缩减梯度(stochastic variance reduced gradient,SVRG)算法引入自适应步长,并在此基础上进一步提高算法数值性能。首先利用具有二维二次终止性的 BB 类步长自适应计算 SVRG 算法的步长。然后在 SVRG 算法的内循环中引入停止准则和负动量来加速算法的收敛速度。利用 Matlab 对提出的新算法进行数值实验,观察算法的数值性能。通过分析算法的数值实验结果,得出算法性能与在最佳步长调整下的 SVRG 算法相当,此外新算法对于初始步长的选取不敏感,且具有自动生成最优步长的能力。

关键词:随机方差缩减梯度算法;BB 类步长;自适应计算;负动量框架

中图分类号:O224　　**文献标志码**:A　　**文章编号**:1672-6693(2024)05-0007-11

优化问题

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

在优化领域中经常出现,其中:$f_i(\cdot)$代表第 i 个损失函数,n 为样本量,每个分量函数 f_i,$i\in\{1,2,\cdots,n\}$是一阶连续可微的。这类问题常应用于监督学习的研究中。

近些年来,针对问题(1)已经提出了许多先进的优化方法。其中全梯度下降(full gradient descent,FGD)算法是解决此问题的典型方法,它的参数更新形式为:

$$\boldsymbol{x}_{t+1}=\boldsymbol{x}_t-\eta_t\nabla F(\boldsymbol{x}_t)=\boldsymbol{x}_t-\frac{\eta_t}{n}\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}_t), \tag{2}$$

其中:t 代表第 t 步迭代,η_t 表示第 t 步迭代的步长。FGD 算法具有线性收敛速度,但当 n 较大时,计算 $F(\boldsymbol{x})$与$\nabla F(\boldsymbol{x})$非常昂贵。这时,随机梯度下降(stochastic variance reduced gradient,SGD)算法[1]作为一种替代算法,成为求解式(1)的首选方法。在 SGD 算法求解的每一步,均匀随机选取 1 个分量梯度进行计算,参数更新为:

$$\boldsymbol{x}_{t+1}=\boldsymbol{x}_t-\eta_t\nabla f_{i_t}(\boldsymbol{x}_t), \tag{3}$$

其中:$\nabla f_{i_t}(\boldsymbol{x}_t)$表示第 i 个分量函数在 $\boldsymbol{x}_t$ 点处的梯度。在式(3)中通常假设∇f_{i_t} 是对∇F 的无偏估计,即:

$$E[\nabla f_{i_t}(\boldsymbol{x}_t)\mid\boldsymbol{x}_t]=\nabla F(\boldsymbol{x}_t)。 \tag{4}$$

SGD 算法大大降低了算法的复杂度,启发了许多新的优化方法。但在实际中,SGD 算法往往比较慢,并且它的性能对样本梯度的方差太敏感。于是,为了提高 SGD 算法的性能,人们又开发了大量的算法,出现了一类更复杂的算法,例如:随机平均梯度下降(stochastic average gradient descent,SAG)法[2]、随机对偶坐标上升(stochastic dual coordinate ascent,SDCA)法[3]、随机方差缩减梯度(stochastic variance reduced gradient,SVRG)法[4]和随机递归梯度算法(stochastic recursive gradient algorithm,SARAH)[5-6],以及它们的近端变体 Prox-SVRG[7]、Prox-SAGA[8]和 Prox-SARAH[9],用于解决复合有限和及随机优化问题。这些算法使用式(1)的特定有限和形式,并结合一些确定性和随机方面来减少步骤的方差。

* **收稿日期**:2023-10-14　**修回日期**:2024-04-01　**网络出版时间**:2024-11-13T15:52
资助项目:国家自然科学基金面上项目(No. 12261019);贵州省自然科学基金一般项目(No. 黔科合基础-ZK[2022]一般 084)
第一作者简介:陈炫睿,男,研究方向为最优化理论及应用,Email:cxr1661375576@163.com;**通信作者**:刘泽显,男,副教授,博士,Email:liuzexian2008@163.com
网络出版地址:https://link.cnki.net/urlid/50.1165.n.20241113.1435.002

SVRG算法与SGD算法不同,SVRG算法的学习速率 η_t 不必衰减,这会产生更快的收敛速度,因为可以使用相对较大的学习速率。SVRG算法每 m 次更新后计算一次全梯度,它的参数更新形式为:

$$\boldsymbol{x}_{t+1}=\boldsymbol{x}_t-\eta_k(\nabla f_{i_t}(\boldsymbol{x}_t)-\nabla f_{i_t}(\tilde{\boldsymbol{x}}_k)+\tilde{\boldsymbol{\mu}}),$$

其中:$\tilde{\boldsymbol{\mu}}=\nabla F(\tilde{\boldsymbol{x}})=\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\tilde{\boldsymbol{x}})$。

SVRG算法构造了方差缩减方法的一般框架,产生的方差也随迭代次数不断衰减。然而就传统SVRG算法而言,一方面需要采用固定步长,但过大的步长和过小的步长对收敛性能都会产生很大的影响:采用较小的步长,算法则可能很难达到收敛;采用较大的步长,又可能会带来学习过程震荡的结果。另一方面需要用户指定内循环 m 的大小,这对算法的收敛性能至关重要。因此,在SVRG算法的基础上,学者们又提出了许多变体来进一步提高性能。

近些年来,Barzilai-Borwein(BB)[10]方法受到越来越多学者的关注。钱玉香等人[11]将改进的BB步长与近端随机递归动量算法结合,提出了一种带BB步长的随机方差缩减算法。Tan等人[12]将BB步长与SGD和SVRG算法相结合,得到了SGD-BB和SVRG-BB方法,并证明了SVRG-BB对于强凸目标函数是线性收敛的。受SVRG-BB算法提出的影响,刘彦等人[13]提出了基于局部Lipschitz常数估计的自适应步长,并将该步长与SVRG算法结合得到了SVRG-AS算法。Min等人[14]提出了一种新的随机梯度下降算法AESVRG,该算法能自适应的调解内循环 m 的大小。Liu等人[15-17]提出了近似最优步长的概念,从新的角度观察BB步长。秦传东等人[18]融合BB方法、小批量算法与SVRG算法的优势,提出了一种带有随机改进BB步长的小批量稀疏随机方差缩减梯度法。王福胜等人[19]将SVRG算法与一种谱梯度BB步长方法有机结合,再利用重要抽样策略,提出了Im-SVRG-BB算法。Huang等人[20]通过自适应地采用长BB步长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。

同时,求解各类优化问题的加速技术也受到了广泛的关注。代表包括重球法[21]和Nesterov的加速技术[22],它们最初是为加速确定性优化问题而开发的。理论结果表明,当目标函数为强凸光滑时,重球法比全梯度下降法收敛速度更快。然而,在求解一般凸问题时,重球法和全梯度下降法的收敛速度是相同的。相比之下,Nesterov的加速技术不需要更强的条件,即对于一般凸问题,它可以获得比重球法和全梯度下降法更快的收敛速度[23]。更详细的理论分析请参阅文献[24]。受确定性优化应用取得巨大成功的启发,近些年来一些学者将Nesterov加速技术引入随机算法中,以提高算法的收敛性能[25-28]。但是在SGD算法中单纯地加入Nesterov外推加速步长不一定能保证更快的收敛速度,反而可能增加随机抽样过程中的累积误差。为了减少这种影响,Allen-Zhu[29-30]借助凸组合的思想提出了另一种形式的加速度框架,即Katyusha法。与Nesterov的加速框架不同的是,Katyusha算法使用前一个迭代点和2个辅助变量的凸组合来构造当前加速步长,然后与SVRG算法框架结合,可以潜在地减少累积误差,实现相对于SVRG算法更快的收敛速度。读者可以参考文献[31-33]了解Katyusha加速框架的进一步应用。吉梦等人[34]将自适应策略及动量加速思想引入到批量SVRG算法中,提出了AdaSVRG算法。He等人[35]设计了一个类似Katyusha的动量阶跃,即负动量框架,并将它纳入经典的方差缩减随机梯度算法中。在构建的基于负动量的框架基础上,提出了一种求解一类凸有限和问题的加速随机算法,即基于负动量的随机方差缩减梯度(negative momentum stochastic average gradient,NMSVRG)算法。

受文献[20]中梯度方法、AESVRG方法以及NMSVRG算法成功解决问题(1)的启发,本文将它们的思想应用于随机方差缩减梯度算法中,主要贡献在于以下几个方面:

1) 在SVRG算法中引入了能自动计算步长的BB类方法,这对于仅采用固定步长的SVRG算法来说,有着巨大的提升;

2) 本文在SVRG算法的内循环中添加了停止准则。它可以动态地将内循环大小调整到一个合适的值。在损失函数值开始波动之前停止迭代。新算法优于采用恒定内循环大小的SVRG算法;

3) 为加速算法的收敛速度,本文还在内循环中引入了负动量,并证明了算法在强凸条件下具有线性收敛性。

1　预备知识

1.1　自适应步长的选取

BB 方法在求解非线性优化问题方面非常成功，该方法的关键思想是由拟牛顿方法激发的。假设要解决的无约束最小化问题为：

$$\min_{x} f(\boldsymbol{x}), \tag{5}$$

其中：f 是可微的。求解式(5)的拟牛顿方法的典型迭代形式如下：

$$\boldsymbol{x}_{t+1}=\boldsymbol{x}_t-\boldsymbol{B}_t^{-1}\nabla f(\boldsymbol{x}_t), \tag{6}$$

其中：B_t 是 f 在当前迭代 $\boldsymbol{x}_t$ 下的 Hessian 矩阵的近似值。$\boldsymbol{B}_t$ 最重要的特征是它必须满足割线方程：

$$\boldsymbol{B}_t\boldsymbol{s}_t=\boldsymbol{y}_t, \tag{7}$$

其中：$\boldsymbol{s}_t=\boldsymbol{x}_t-\boldsymbol{x}_{t-1}$，$\boldsymbol{y}_t=\nabla f(\boldsymbol{x}_t)-\nabla f(\boldsymbol{x}_{t-1})$。

BB 步长满足最小二乘意义上的某些正割方程，引入以下长、短选择：

$$\eta_k^{\mathrm{BB1}}=\underset{a\in\mathbf{R}}{\operatorname{argmin}}\left\|\eta^{-1}\boldsymbol{s}_{k-1}-\boldsymbol{y}_{k-1}\right\|_2=\frac{\boldsymbol{s}_{k-1}^{\mathrm{T}}\boldsymbol{s}_{k-1}}{\boldsymbol{s}_{k-1}^{\mathrm{T}}\boldsymbol{y}_{k-1}},\ \eta_k^{\mathrm{BB2}}=\underset{a\in\mathbf{R}}{\operatorname{argmin}}\left\|\boldsymbol{s}_{k-1}-\eta\boldsymbol{y}_{k-1}\right\|_2=\frac{\boldsymbol{s}_{k-1}^{\mathrm{T}}\boldsymbol{s}_{k-1}}{\boldsymbol{y}_{k-1}^{\mathrm{T}}\boldsymbol{y}_{k-1}}。$$

Huang 等人[11]提供了一种梯度法实现二维二次终止的机制。然后利用该机制推导出一种新的步长，从而使 BB 方法具有二维二次终止，提出的步长如下：

$$\tilde{\eta}_k=\frac{2}{\dfrac{\varphi_2}{\varphi_3}+\sqrt{\left(\dfrac{\varphi_2}{\varphi_3}\right)^2-4\dfrac{\varphi_1}{\varphi_3}}}, \tag{8}$$

其中：$\dfrac{\varphi_1}{\varphi_3}=\dfrac{\eta_{k-1}^{\mathrm{BB2}}-\eta_k^{\mathrm{BB2}}}{\eta_{k-1}^{\mathrm{BB2}}\eta_k^{\mathrm{BB2}}(\eta_{k-1}^{\mathrm{BB1}}-\eta_k^{\mathrm{BB1}})}$，$\dfrac{\varphi_2}{\varphi_3}=\dfrac{\eta_{k-1}^{\mathrm{BB1}}\eta_{k-1}^{\mathrm{BB2}}-\eta_k^{\mathrm{BB1}}\eta_k^{\mathrm{BB2}}}{\eta_{k-1}^{\mathrm{BB2}}\eta_k^{\mathrm{BB2}}(\eta_{k-1}^{\mathrm{BB1}}-\eta_k^{\mathrm{BB1}})}$。

下面的引理分别给出了步长式(8)在 $\dfrac{\varphi_1}{\varphi_3}\geqslant 0$，$\varphi_2\neq 0$ 时与 $\dfrac{\varphi_1}{\varphi_3}<0$，$\varphi_2\neq 0$ 这 2 种情况下的界，具体证明过程参见文献[20]。

引理 1[20]　由式(8)给出的步长是良定的，并且：

1) 当 $\dfrac{\varphi_1}{\varphi_3}\geqslant 0$ 与 $\varphi_2\neq 0$ 时，有 $\dfrac{\varphi_3}{\varphi_2}\leqslant\tilde{\eta}_k\leqslant\min\{\eta_k^{\mathrm{BB2}},\eta_{k-1}^{\mathrm{BB2}}\}$；

2) 当 $\dfrac{\varphi_1}{\varphi_3}<0$ 及 $\varphi_2\neq 0$ 时，有 $\max\{\eta_k^{\mathrm{BB2}},\eta_{k-1}^{\mathrm{BB2}}\}\leqslant\tilde{\eta}_k\leqslant\left|\dfrac{\varphi_3}{\varphi_2}\right|$。

大量研究表明，采用长 BB 步长 η_k^{BB1}(因为 $\eta_k^{\mathrm{BB1}}\geqslant\eta_k^{\mathrm{BB2}}$)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 $\dfrac{\varphi_1}{\varphi_3}\geqslant 0$ 和 $\varphi_2\neq 0$，由定理 1 可以知道，步长 $\tilde{\eta}_k$ 小于 η_k^{BB2} 和 $\eta_{k-1}^{\mathrm{BB2}}$2 个短步长。另一方面，当 $\dfrac{\varphi_1}{\varphi_3}<0$ 和 $\varphi_2\neq 0$ 时，步长 $\tilde{\eta}_k$ 都大于 η_k^{BB2} 和 $\eta_{k-1}^{\mathrm{BB2}}$ 这 2 个步长。结合这 2 种情况，将 η_k 进行替换，替换为 $\min\{\eta_k^{\mathrm{BB2}},\eta_{k-1}^{\mathrm{BB2}},\tilde{\eta}_k\}$，替换后的步长为 η_k^{BB2}、$\eta_{k-1}^{\mathrm{BB2}}$ 及 $\tilde{\eta}_k$ 中最小的一个。

由于自适应方案的成功，Huang 等人[20]在此基础上提出步长的截断形式如下：

$$\eta_k=\begin{cases}\min\{\eta_{k-1}^{\mathrm{BB2}},\eta_k^{\mathrm{BB2}},\tilde{\eta}_k\}，\text{如果 } \eta_k^{\mathrm{BB2}}/\eta_k^{\mathrm{BB1}}<\tau_k；\\ \eta_k^{\mathrm{BB1}}，\text{否则。}\end{cases} \tag{9}$$

其中：$\tau_k>0$。更新式(9)中 τ_k 最简单的方法是将所有 k 都设置为某个常数 $\tau\in(0,1)$。对于这种固定格式，步长式(9)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：

$$\tau_{k+1}=\begin{cases}\dfrac{\tau_k}{\gamma}，\text{如果 } a_k^{\mathrm{BB2}}/a_k^{\mathrm{BB1}}<\tau_k；\\ \dfrac{\tau_k}{\gamma}，\text{否则。}\end{cases} \tag{10}$$

显然,固定格式的 τ 是式(10)中 $\gamma=1$ 的一种特殊情况。

1.2 内循环大小 m 的自适应选取

内循环大小的选取与步长是密切相关的。一方面,当步长较大时,训练损失在少量迭代后开始波动。另一方面,较小的步长会承受远超过 n 次的迭代。AESVRG 算法设置了一个停止准则,当内循环开始出现波动的时候,算法会跳出内循环,进入外循环。这样 AESVRG 算法就可以动态地将内循环大小调整为合适的值。具体思想是将梯度下降法应用于凸问题时,当 $\boldsymbol{x}_t$ 逐渐接近最优值 $\boldsymbol{x}^*$ 时,梯度 $\nabla F(\boldsymbol{x}_t)$ 不断减小。因为梯度法的迭代是:

$$\boldsymbol{x}_{t+1}-\boldsymbol{x}_t=-\eta\nabla F(\boldsymbol{x}_t),$$

于是有 $\|\boldsymbol{x}_{t+1}-\boldsymbol{x}_t\|<\|\boldsymbol{x}_t-\boldsymbol{x}_{t-1}\|$。直观地看,如果 $\boldsymbol{x}_t$ 不断接近最优值 $\boldsymbol{x}^*$,则 $\nabla F(\boldsymbol{x}_t)$ 一般衰减,并有方差引起的轻微波动。此时如果考虑一个窗口,则当 $\|\boldsymbol{x}_{t+m_0}-\boldsymbol{x}_t\|>\|\boldsymbol{x}_t-\boldsymbol{x}_{t-m_0}\|$ 时,认为算法产生了波动,此时停止准则成立,算法跳出内循环进入外循环。

1.3 负动量框架

目前,一些随机算法在经典的 Nesterov 加速框架中加入了外推步骤,迭代如下:

$$\boldsymbol{y}_k^s=\boldsymbol{x}_k^s+\beta_k(\boldsymbol{x}_k^s-\boldsymbol{x}_{k-1}^s),$$

其中:s 与 k 分别表示外层循环和内层循环的迭代次数。$\beta_k\in(0,1)$ 代表加速参数,序列 $\{\boldsymbol{x}_k^s\}$ 是基于一些方差缩减算法产生的。从迭代中可以看出,这一步是向前的一步。然而,负随机梯度估计并不总是下降方向。如果在点 $\boldsymbol{x}_k^s$ 处的负随机梯度估计不是下降方向,则加速点处的梯度估计方差可能大于点 $\boldsymbol{x}_k^s$ 处的方差。因此,为了减少动量步的影响,负动量框架被提出,即 Katyusha 式加速步:

$$\boldsymbol{y}_k^s=\theta\boldsymbol{x}_k^s+(1-\theta)\tilde{\boldsymbol{x}}^{s-1},$$

其中:$\theta\in(0,1)$ 以及 $\tilde{\boldsymbol{x}}^{s-1}$ 是 1 个快照点。

一方面,与 Nesterov 的外推步骤相比,负动量框架使用凸组合步骤,即在某种意义上的插值步骤。另一方面,不同于原来 Katyusha 动量框架涉及的 2 个不同辅助变量,这里的算法中只增加了 1 个额外的辅助变量。因此,提出的算法对参数的选择更加友好。

2 新算法的提出

为了求解问题(1),Johnson 等人[4]提出了 SVRG 算法,如算法 1 所示。

算法 1(SVRG 算法) 步骤 0,设置初始参数:内循环迭代次数 m,外循环迭代次数 T,固定步长 $\eta>0$,初始点 $\tilde{x}_0$,$k=0$。

步骤 1,计算全梯度 $\boldsymbol{g}_k=\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\tilde{\boldsymbol{x}}_k)$,令 $\boldsymbol{x}_0=\tilde{\boldsymbol{x}}_k$,$t=0$。

步骤 2,随机选取 $i_t\in\{1,\cdots,n\}$,计算 $\boldsymbol{x}_{t+1}=\boldsymbol{x}_t-\eta(\nabla f_{i_t}(\boldsymbol{x}_t)-\nabla f_{i_t}(\tilde{\boldsymbol{x}}_k)+\boldsymbol{g}_k)$,令 $t=t+1$,转步骤 3。

步骤 3,如果 $t>m$,则转步骤 4;否则转步骤 2。

步骤 4,更新 $\tilde{\boldsymbol{x}}_k$。

步骤 5,令 $k=k+1$,如果 $k<T$,转步骤 1;否则停止。

对于 $\tilde{\boldsymbol{x}}_{k+1}$ 的更新有 2 种选择:要么为 $\tilde{\boldsymbol{x}}_{k+1}=\boldsymbol{x}_m$,要么为 $\tilde{\boldsymbol{x}}_{k+1}=\boldsymbol{x}_t$。其中,$t$ 从 $\{1,\cdots,m\}$ 中随机选取。

在 SVRG 算法中有 2 个循环,在外部循环中计算一个完整的梯度 $\boldsymbol{g}_k$,在内部循环中使用该梯度生成方差较小的随机梯度。然后根据内循环的输出选择 $\tilde{\boldsymbol{x}}$,用于下一个外部循环。

根据前面的分析,为了使 SVRG 算法能够自适应的计算步长和调节内循环的大小,本文提出了一种新的随机梯度下降算法(算法 2)。

算法 2 步骤 0,设置初始参数:窗口大小 m_0,内循环次数 m,外循环迭代次数 T,初始步长 $\eta_k>0$,初始点 $\tilde{\boldsymbol{x}}_0$,负动量系数 $\theta\in(0,1)$,$k=0$。

步骤 1,计算全梯度 $\boldsymbol{g}_k=\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\tilde{\boldsymbol{x}}_k)$。

步骤 2,如果 $k>0$,则计算步长 $\eta_k=\eta_k/m$,其中 η_k 由式(9)给出。令 $\boldsymbol{x}_0=\tilde{\boldsymbol{x}}_k$,$t=0$。

步骤 3,随机选取 $i_t\in\{1,\cdots,n\}$,计算 $\boldsymbol{y}_t=\theta\boldsymbol{x}_t+(1-\theta)\tilde{\boldsymbol{x}}_k$,$\boldsymbol{x}_{t+1}=\boldsymbol{x}_t-\eta_k(\nabla f_{i_t}(\boldsymbol{y}_t)-\nabla f_{i_t}(\tilde{\boldsymbol{x}}_k)+\boldsymbol{g}_k)$。令 $t=t+1$,转步骤 4。

步骤 4,如果 $t\%m_0=0$,$t\geqslant 2m_0$ 且 $\|\boldsymbol{x}_t-\boldsymbol{x}_{t-m_0}\|>\|\boldsymbol{x}_{t-m_0}-\boldsymbol{x}_{t-2m_0}\|$ 转步骤 5;否则转步骤 3。

步骤 5,更新 $\tilde{\boldsymbol{x}}_{k+1}=\boldsymbol{x}_{t+1}$,令 $m=t+1$,转步骤 6。

步骤 6,令 $k=k+1$,如果 $k<T$,转步骤 1;否则停止。

现对算法 2 进行一些解释。窗口大小 m_0 需要用户指定,相关研究表明 m_0 大小应该设置在 $0.1n$ 与 $0.2n$ 之间,其中 n 代表数据集维度。算法 2 中步骤 5 对于 $\tilde{x}_{k+1}$ 的更新采用的是 SVRG 算法中的选择 1。步骤 4 是满足停止准则的条件:第一个条件意味着算法每 m_0 次迭代检查不等式 $\|\boldsymbol{x}_t-\boldsymbol{x}_{t-m_0}\|>\|\boldsymbol{x}_{t-m_0}-\boldsymbol{x}_{t-2m_0}\|$,$t\geqslant 2m_0$ 是因为算法至少需要 2 个窗口来检查停止条件。如果满足停止准则就跳出内循环,进入步骤 5,否则就回到步骤 3。

算法 2 的优点主要体现在以下几个方面:1) 与采用固定步长的 SVRG 算法相比,本算法采用长 BB 步长和形如式(8)相关的短步长结合来自适应计算步长;2) 本算法不需要用户去指定内循环大小,采用带有窗口的停止准则来自适应的调节内循环大小;3) 在内循环中引入了负动量框架进一步提升了 SVRG 算法的收敛速度。

3　收敛性分析

本节用强凸目标函数 $F(\boldsymbol{x})$来分析算法 2 的线性收敛性。下面的假设贯穿本节。

假设 1　假设式(4)对任何 x_t 都成立。假设目标函数 $F(\boldsymbol{x})$是 μ 强凸的,即:

$$F(\boldsymbol{y})\geqslant F(\boldsymbol{x})+\nabla F(\boldsymbol{x})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{x})+\frac{\mu}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2,\ \forall\,\boldsymbol{x},\boldsymbol{y}\in\mathbf{R}^d。$$

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 $\boldsymbol{x},\boldsymbol{x}'\in\mathbf{R}^d$ 有:

$$\|\nabla f_i(\boldsymbol{x})-\nabla f_i(\boldsymbol{y})\|_2\leqslant L\|\boldsymbol{x}-\boldsymbol{y}\|_2,\ \forall\,\boldsymbol{x},\boldsymbol{y}\in\mathbf{R}^d。$$

在此假设下,不难看出$\nabla F(x)$也是 L-Lipschitz 连续的,即:

$$\|\nabla F(\boldsymbol{x})-\nabla F(\boldsymbol{y})\|_2\leqslant L\|\boldsymbol{x}-\boldsymbol{y}\|_2,\ \forall\,\boldsymbol{x},\boldsymbol{y}\in\mathbf{R}^d。$$

下面来自文献[36]的引理,有助于本节的收敛性分析。

引理 2　如果 $f(\boldsymbol{x}):\mathbf{R}^d\to\mathbf{R}$ 是凸的,且梯度是 L-Lipschitz 连续的,则:

$$\|\nabla f(\boldsymbol{x})-\nabla f(\boldsymbol{y})\|_2^2\leqslant L(\boldsymbol{x}-\boldsymbol{y})^{\mathrm{T}}(\nabla f(\boldsymbol{x})-\nabla f(\boldsymbol{y})),\ \forall\,\boldsymbol{x},\boldsymbol{y}\in\mathbf{R}^d。$$

接下来的引理则揭示了 2 个连续迭代到最优点的距离之间的关系。

引理 3　定义 $\alpha_k:=(1-2\eta_k\mu(1-\eta_kL))^m+\dfrac{4\eta_kL^2}{\mu(1-\eta_kL)}$,对于算法 2,则第 k 次迭代有不等式:

$$E\|\tilde{\boldsymbol{x}}_{k+1}-\boldsymbol{x}^*\|_2^2<\alpha_k\|\tilde{\boldsymbol{x}}_k-\boldsymbol{x}^*\|_2^2,$$

其中:$\boldsymbol{x}^*$ 是问题(1)的最优解。

证明　令 $v_{i_t}^t=\nabla f_{i_t}(\boldsymbol{y}_t)-\nabla f_{i_t}(\tilde{\boldsymbol{x}}_k)+\nabla F(\tilde{\boldsymbol{x}}_k)$是算法 2 中的第 k 次迭代。则:

$$\begin{aligned}
&E\|v_{i_t}^t\|_2^2=E\|\theta\,\nabla f_{i_t}(\boldsymbol{x}_t)+(1-\theta)\nabla f_{i_t}(\tilde{\boldsymbol{x}}_k)-\nabla f_{i_t}(\tilde{\boldsymbol{x}}_k)+\nabla F(\tilde{\boldsymbol{x}}_k)\|_2^2=\\
&E\|(\theta\,\nabla f_{i_t}(\boldsymbol{x}_t)-\theta\,\nabla f_{i_t}(\boldsymbol{x}^*))-(\theta\,\nabla f_{i_t}(\tilde{\boldsymbol{x}}_k)-\theta\,\nabla f_{i_t}(\boldsymbol{x}^*))+\nabla F(\tilde{\boldsymbol{x}}_k)\|_2^2\leqslant\\
&2E\|\theta\,\nabla f_{i_t}(\boldsymbol{x}_t)-\theta\,\nabla f_{i_t}(\boldsymbol{x}^*)\|_2^2+4E\|\theta\,\nabla f_{i_t}(\tilde{\boldsymbol{x}}_k)-\theta\,\nabla f_{i_t}(\boldsymbol{x}^*)\|_2^2+4\|\nabla F(\tilde{\boldsymbol{x}}_k)\|_2^2=\\
&2\theta^2E\|\nabla f_{i_t}(\boldsymbol{x}_t)-\nabla f_{i_t}(\boldsymbol{x}^*)\|_2^2+4\theta^2E\|\nabla f_{i_t}(\tilde{\boldsymbol{x}}_k)-\nabla f_{i_t}(\boldsymbol{x}^*)\|_2^2+4\|\nabla F(\tilde{\boldsymbol{x}}_k)\|_2^2\leqslant\\
&2E\|\nabla f_{i_t}(\boldsymbol{x}_t)-\nabla f_{i_t}(\boldsymbol{x}^*)\|_2^2+4E\|\nabla f_{i_t}(\tilde{\boldsymbol{x}}_k)-\nabla f_{i_t}(\boldsymbol{x}^*)\|_2^2+4\|\nabla F(\tilde{\boldsymbol{x}}_k)\|_2^2\leqslant\\
&2LE[(\boldsymbol{x}_t-\boldsymbol{x}^*)^{\mathrm{T}}(\nabla f_i(x_t)-\nabla f_i(\boldsymbol{x}^*))]+4L^2\|\tilde{\boldsymbol{x}}_k-\boldsymbol{x}^*\|_2^2+4L^2\|\boldsymbol{x}_k-\boldsymbol{x}^*\|_2^2=\\
&2L(\tilde{\boldsymbol{x}}_t-\boldsymbol{x}^*)^{\mathrm{T}}\,\nabla F(\boldsymbol{x}_t)+8L^2\|\tilde{\boldsymbol{x}}_k-\boldsymbol{x}^*\|_2^2。
\end{aligned}$$

在第 1 个不等式中使用了 2 次不等式$(a-b)^2\leqslant 2a^2+2b^2$,在第 2 个不等式中利用了 $\theta\in(0,1)$的事实。在第 3

个不等式中,在 $f_{i_t}(\boldsymbol{x})$上应用引理2以及利用了∇f_{i_t}和∇F的L-Lipschitz连续性。在最后1个等式中,使用了$E[\nabla f_{i_t}(\boldsymbol{x})]=\nabla F(\boldsymbol{x})$和$\nabla F(\boldsymbol{x}^*)=0$的事实。

有了上述结论后,根据文献[12]的结果,直接给出在$\boldsymbol{x}_t$和$\tilde{\boldsymbol{x}}_k$条件下,$\boldsymbol{x}_{t+1}$到$\boldsymbol{x}^*$距离的界:

$$E\|\boldsymbol{x}_{t+1}-\boldsymbol{x}^*\|_2^2\leqslant[1-2\eta_k\mu(1-\eta_kL)]\|\boldsymbol{x}_t-\boldsymbol{x}^*\|_2^2+8\eta_k^2L^2\|\tilde{\boldsymbol{x}}_k-\boldsymbol{x}^*\|_2^2。$$

通过对t递归应用上面的不等式,并且注意$\tilde{\boldsymbol{x}}_k=\boldsymbol{x}_0$以及$\tilde{\boldsymbol{x}}_{k+1}=\boldsymbol{x}_m$,可以得到:

$$E|\tilde{\boldsymbol{x}}_{k+1}-\boldsymbol{x}^*|_2^2\leqslant[1-2\eta_k\mu(1-\eta_kL)]^m\|\tilde{\boldsymbol{x}}_k-\boldsymbol{x}^*\|_2^2+8\eta_k^2L^2\sum_{j=0}^{m-1}[1-2\eta_k\mu(1-\eta L)]^j\|\tilde{\boldsymbol{x}}_k-\boldsymbol{x}^*\|_2^2<$$

$$\left[(1-2\eta_k\mu(1-\eta_kL))^m+\frac{4\eta_kL^2}{\mu(1-\eta_kL)}\right]\|\tilde{\boldsymbol{x}}_k-\boldsymbol{x}^*\|_2^2=\alpha_k\|\tilde{\boldsymbol{x}}_k-\boldsymbol{x}^*\|_2^2。$$

故结论得证。 证毕

有了上面的引理后,现在给出算法2的线性收敛性证明。

定理1 定义$\beta=\dfrac{1-\mathrm{e}^{-2\mu/L}}{2}$,在算法2中,如果选择$m$使得:

$$m>\max\left\{\frac{2}{\log(1-2\beta)+2\mu/L},\frac{4L^2}{\beta\mu^2}+\frac{L}{\mu}\right\},\tag{11}$$

则算法2以期望收敛,即$E\|\tilde{\boldsymbol{x}}_k-\boldsymbol{x}^*\|_2^2\leqslant(1-\beta)^k\|\tilde{\boldsymbol{x}}_0-\boldsymbol{x}^*\|_2^2$。

证明 由式(8)步长的形式可以知道,所定义的步长η_k不大于长BB步长η_k^{BB1},结合$F(\boldsymbol{x})$的强凸性,不难得到算法2计算的步长的上界:

$$\frac{1}{m}\cdot\eta_k\leqslant\frac{1}{m}\cdot\eta_k^{\mathrm{BB1}}=\frac{1}{m}\cdot\frac{\|\tilde{\boldsymbol{x}}_k-\tilde{\boldsymbol{x}}_{k-1}\|_2^2}{(\tilde{\boldsymbol{x}}_k-\tilde{\boldsymbol{x}}_{k-1})^{\mathrm{T}}(\boldsymbol{g}_k-\boldsymbol{g}_{k-1})}\leqslant\frac{1}{m}\cdot\frac{\|\tilde{\boldsymbol{x}}_k-\tilde{\boldsymbol{x}}_{k-1}\|_2^2}{\mu\|\tilde{\boldsymbol{x}}_k-\tilde{\boldsymbol{x}}_{k-1}\|_2^2}=\frac{1}{m\mu}。$$

根据$\nabla F(\boldsymbol{x})$的L-Lipschitz连续性,不难得到η_k的下界是$\dfrac{1}{mL}$。因此有:

$$\alpha_k\leqslant\left[1-\frac{2\mu}{mL}\left(1-\frac{L}{m\mu}\right)\right]^m+\frac{4L^2}{m\mu^2\left[1-\dfrac{L}{m\mu}\right]}\leqslant$$

$$\exp\left\{-\frac{2\mu}{mL}\left(1-\frac{L}{m\mu}\right)\cdot m\right\}+\frac{4L^2}{m\mu^2\left[1-\dfrac{L}{m\mu}\right]}=\exp\left\{-\frac{2\mu}{L}+\frac{2}{m}\right\}+\frac{4L^2}{m\mu^2-L\mu}。$$

把式(11)代入上面的不等式则有:

$$\alpha_k<\exp\left\{-\frac{2\mu}{L}+\log(1-2\beta)+\frac{2\mu}{L}\right\}+\frac{4L^2}{\dfrac{4L^2}{\beta}+L\mu-L\mu}=(1-2\beta)+\beta=1-\beta。$$

最后应用引理3就可以得到定理1的结果。 证毕

4 数值实验

本节进行了大量的实验来验证新提出算法的优点。为了公平起见,所有的数值实验均在CPU为i5-10500的同一台电脑上通过Matlab 9.0进行。

分别在LIBSVM网站(http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/)上公开提供的4个不同的训练数据集上评估了本文提出的新算法。特别地,应用新算法来解决机器学习中的2个标准测试问题,即:

1) 带有l_2范数正则化的逻辑回归:

$$\min_x F(\boldsymbol{x})=\frac{1}{n}\sum_{i=1}^n\log[1+\mathrm{e}^{-b_i\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}}]+\frac{\lambda}{2}\|\boldsymbol{x}\|_2^2;$$

2) 带有l_2范数正则化的平方铰链损失支持向量机:

$$\min_x F(\boldsymbol{x})=\frac{1}{n}\sum_{i=1}^n([1-b_i\boldsymbol{a}_i^{\mathrm{T}}\boldsymbol{x}]_+)^2+\frac{\lambda}{2}\|\boldsymbol{x}\|_2^2。$$

其中：$\boldsymbol{a}_i \in \mathbf{R}^d$ 以及 $b_i \in \{\pm 1\}$ 分别为第 i 个样本的特征向量和标签。

训练数据集的具体信息见表 1。

表 1　实验数据和模型信息

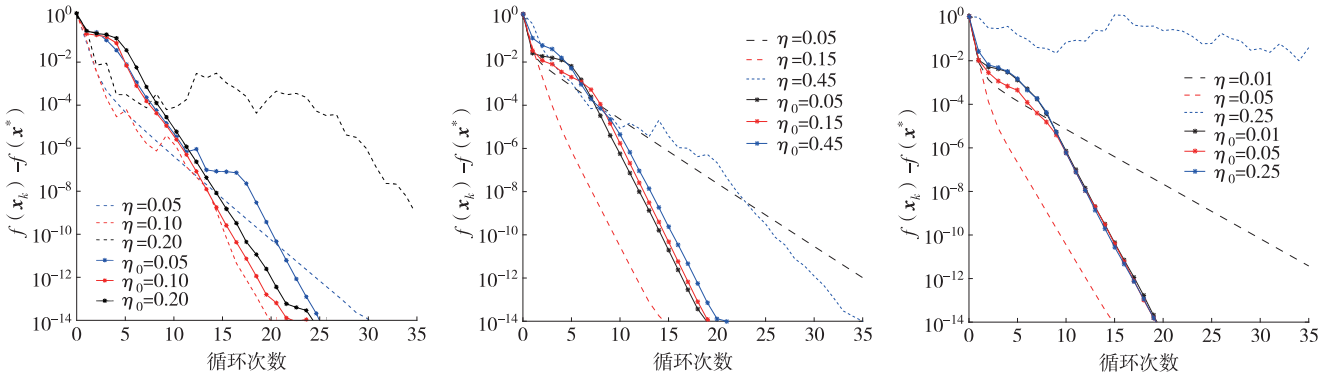
Tab. 1　Data and model information of the experiments

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}

在下面图 1 的几个子图中，给出算法 2 和 SVRG 算法在不同步长（对于算法 SVRG）和不同初始步长（对于算法 2）下的次优性（$f(\tilde{\boldsymbol{x}}_k)-f(\boldsymbol{x}^*)$）比较。这里设置 SVRG 算法的内循环大小为 $m=2n$。

在图 2 的 3 个子图中，观察算法 2 中步长的变化趋势。

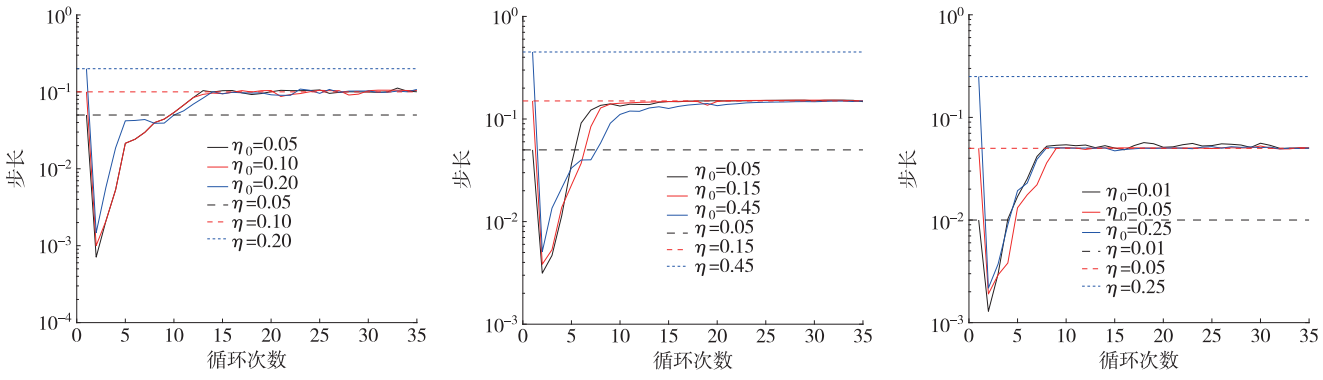
算法 2 和 SVRG 的对比结果如图 1 和图 2 所示。在 6 个子图中，x 轴均表示循环次数 k，即算法 2 中的外循环的次数。在图 1 中 y 轴表示次优性：$f(\tilde{\boldsymbol{x}}_k)-f(\boldsymbol{x}^*)$，在图 2 中 y 轴表示步长 η_k 大小。在 6 个子图中，虚线对应固定步长的 SVRG。此外，红色虚线总是代表最佳调优的固定步长 SVRG，黑色虚线与最佳调优的固定步长相比，使用较小的固定步长，蓝色虚线使用较大的固定步长。实线对应不同初始步长 η_0 的算法 2。图 1b 和图 2b 中蓝色、红色和黑色实线分别对应初始步长 $\eta_0=0.45,0.15,0.05$。图 1c 和图 2c 中蓝色、红色和黑色实线分别对应初始步长 $\eta_0=0.25,0.05,0.01$。



a　w8a 上的次优性比较　　b　a9a 上的次优性比较　　c　ijcnn1 上的次优性比较

图 1　3 个数据集上的次优性比较

Fig. 1　Sub-optimality comparison on three datasets



a　w8a 上的步长变化　　b　a9a 上的步长变化　　c　ijcnn1 上的步长变化

图 2　3 个数据集上的步长变化

Fig. 2　Step size variation on three datasets

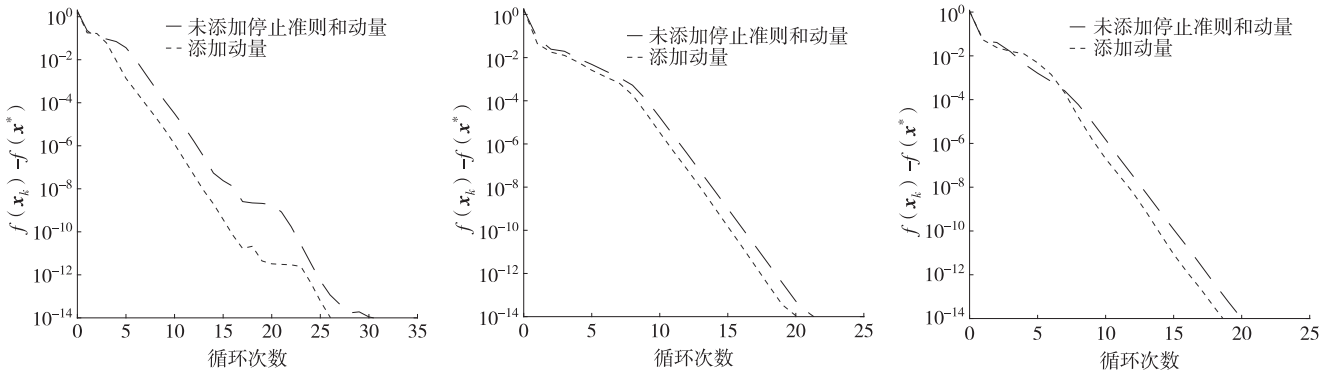
从图1a、1b、1c可以看出,算法2始终可以达到与最调优步长SVRG算法相同的次优性水平。尽管算法2有时比具有最佳调优步长的SVRG需要更多的循环次数,但它明显优于具有其他2种步长选择的SVRG。此外,从图2a、2b、2c中可以看到,算法2计算的步长会收敛到SVRG算法的最佳调优步长。从图1中还可以看到,算法2对于初始步长的选择并不敏感。对于不同大小的初始步长,数值效果差别很小,所以有理由相信算法2在实际中具有很大的潜力。

算法2在内循环中引入了动量以及停止准则,它们都对算法的数值效果有较大的提升。图3是算法2的内循环中没引入动量、停止准则与只引入动量次优性的对比。3个子图中x轴表示循环次数k,y轴表示次优性:$f(\tilde{\boldsymbol{x}}_k)-f(\boldsymbol{x}^*)$。其中长虚线代表内循环中未添加停止准则和动量的数值实验结果,短虚线代表添加动量后算法的数值实验结果。

图4是算法2的内循环中未引入动量及停止准则与仅引入停止准则次优性的结果对比。3个子图中x轴表示循环次数k,y轴表示次优性:$f(\tilde{\boldsymbol{x}}_k)-f(\boldsymbol{x}^*)$。其中长虚线代表内循环中未添加停止准则和动量的数值实验结果,短虚线代表添加停止准则后算法的数值实验结果。

图5是算法2的内循环中引入动量以及停止准则与两者都未引入的次优性的结果比较。3个子图中x轴表示循环次数k,y轴表示次优性:$f(\tilde{\boldsymbol{x}}_k)-f(\boldsymbol{x}^*)$。其中长虚线代表内循环中未添加停止准则和动量的数值实验结果,短虚线代表添加动量和停止准则后算法的数值实验结果。

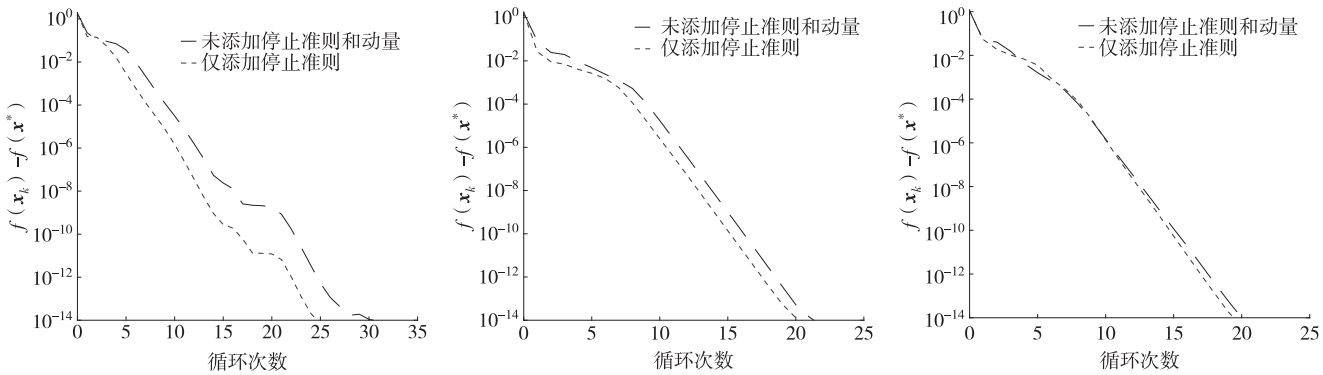
从图3~图5的所有子图中,可以观察到无论是添加动量还是添加停止准则都对算法的数值效果有所提升。特别是当两者都添加时,算法的提升效果最大。



a w8a数据集 b a9a数据集 c ijcnn1数据集

图3 算法2只添加动量和两者都不添加的次优性对比

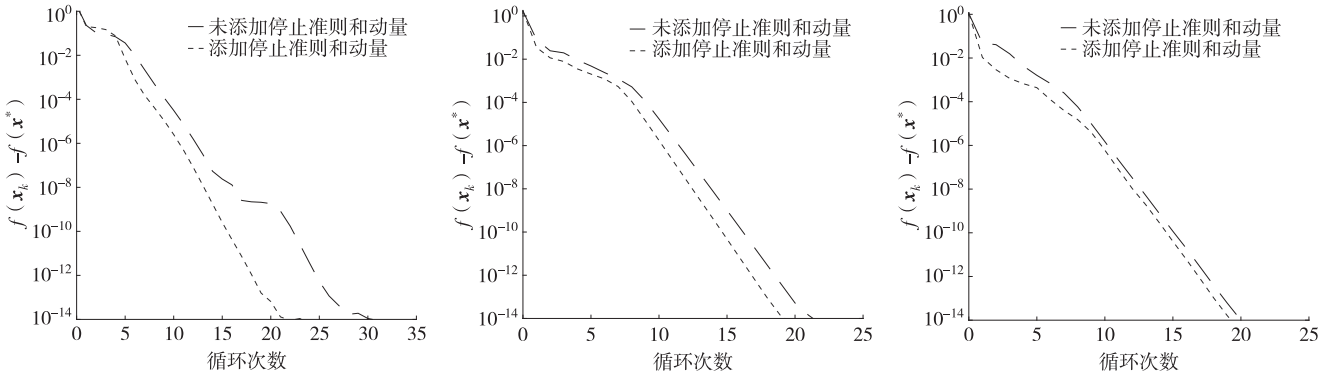
Fig. 3 Algorithm 2 adds momentum only compared to the sub-optimality of adding neither



a w8a数据集 b a9a数据集 c ijcnn1数据集

图4 算法2只添加停止准则和两者都不添加的次优性对比

Fig. 4 Algorithm 2 adds the stopping criterion only compared to the sub-optimality of adding neither



a　w8a 数据集　　b　a9a 数据集　　c　ijcnn1 数据集

图 5　算法 2 添加动量和停止准则与两者都不添加的次优性对比

Fig. 5　Algorithm 2 adds momentum and stop criteria compared with the sub-optimality of adding neither

5　结论

本文提出利用具有二维二次终止性的 BB 类方法自动计算 SVRG 算法的步长，并在 SVRG 算法中添加了负动量以及停止准则去加速算法的收敛速度，从而得到了一类新的随机梯度方法，且证明了新算法在强凸条件下的线性收敛性。通过在真实数据集上进行数值实验，比较了新提出的算法和与传统 SVRG 算法的性能。数值结果表明，新提出的算法性能与在最佳步长调整下的原始 SVRG 方法相当。此外新算法对于初始步长的选取不敏感，而传统的 SVRG 方法在选取不同步长时，数值效果差异较大。通过新算法步长的变化，可以观察到所产生的步长最终将接近于 SVGR 算法中的最调优步长。因此新算法具有自动生成最优步长的能力。最后通过是否对算法添加动量或停止准则的数值实验，得到在 SVRG 算法的内循环中添加动量和停止准则能加速算法的收敛速度。

参考文献：

[1] ROBBINS H，MONRO S. A stochastic approximation method[J]. Annals of Mathematical Statistics，1951，22(3)：400-407.

[2] ROUX N L，SCHMIDT M，BACH F. A stochastic gradient method with an exponential convergence rate for finite training sets [C]//PEREIRA F，BURGES J C，BOTTOU L，et al. NIPS'12：Proceedings of the 25th International Conference on Neural Information Processing Systems. New York：Curran Associates Inc，2012，2：2663-2671.

[3] SHALEV-SHWARTZ S，ZHANG T. Stochastic dual coordinate ascent methods for regularized loss[J]. The Journal of Machine Learning Research，2013，14(1)：567-599.

[4] JOHNSON R，ZHANG T. Accelerating stochastic gradient descent using predictive variance reduction[C]//BURGES C J C，BOTTOU L，WELLING M，et al. NIPS'13：Proceedings of the 26th International Conference on Neural Information Processing Systems. New York：Curran Associates Inc，2013，1：315-323.

[5] NGUYEN L M，LIU J，SCHEINBERG K，et al. SARAH：a novel method for machine learning problems using stochastic recursive gradient[C]//PRECUP D，THE Y W. ICML'17：Proceedings of the 34th International Conference on Machine Learning. Cambridge：JMLR. org，2017：2613-2621.

[6] NGUYEN L M，SCHEINBERG K，MARTIN TAKÁ. Inexact SARAH algorithm for stochastic optimization[J]. Informa UK Limited，2021，36(1)：237-258.

[7] ZHANG T，XIAO L. A proximal stochastic gradient method with progressive variance reduction[J]. SIAM Journal on Optimization A Publication of the Society for Industrial & Applied Mathematics，2014，24(4)：2057-2075.

[8] DEFAZIO A，BACH F，LACOSTE-JULIEN S. SAGA：a fast incremental gradient method with support for non-strongly convex composite objectives[J]. Advances in neural information processing systems，2014，27(1)：1-9.

[9] PHAM N H，NGUYEN L M，PHAN D T，et al. Prox-SARAH：an efficient algorithmic framework for stochastic composite

nonconvex optimization[J]. Journal of Machine Learning Research,2020,21(1):1-48.

[10] BARZILAI J,BORWEIN J M. Two-point step size gradient methods[J]. IMA journal of numerical analysis,1988,8(1):141-148.

[11] 钱玉香,赵勇,杨帆. 基于BB步长的近端随机递归动量算法[J]. 北华大学学报(自然科学版),2024,25(1):8-16.
QIAN Y X,ZHAO Y,YANG F. A proximal stochstic recursive momentum algorithm using Barzilai-Borwein stepsize[J]. Journal of Beihua University(Natural Science),2024,25(1):8-16.

[12] TAN C H,MA S Q,DAI Y H,et al. Barzilai-Borwein step size for stochastic gradient descent[C]//LEE D D,von LUXBURG U,GARNETT R,et al. NIPS'16:Proceedings of the 30th International Conference on Neural Information Processing Systems. New York:Curran Associates Inc,2016:685-693.

[13] 刘彦,郭田德,韩丛英. 一类随机方差缩减算法的分析与改进[J]. 中国科学:数学,2021,51(9):1433-1450.
LIU Y,GUO T D,HAN C Y. Analysis and improvement for a class of variance reduced methods[J]. Scientia Sinica(Mathematica),2021,51(9):1433-1450.

[14] MIN E X,ZHAO Y W,LONG J,et al. SVRG with adaptive epoch size[C]//2017 International Joint Conference on Neural Networks,May 14-19,2017,Anchorage,AK,USA. Piscataway:IEEE,2017:2935-2942.

[15] LIU Z X,LIU H W,DONG X L. An efficient gradient method with approximate optimal stepsize for the strictly convex quadratic minimization problem[J]. Optimization,2017,67(3):427-440.

[16] LIU Z X,LIU H W. Several efficient gradient methods with approximate optimal stepsizes for large scale unconstrained optimization[J]. Journal of Computational and Applied Mathematics,2018,328:400-413.

[17] LIU Z X,LIU H W. An efficient gradient method with approximate optimal stepsize for large-scale unconstrained optimization [J]. Numerical Algorithms,2018,78(4):21-39.

[18] 秦传东,杨旭. 带有随机改进Barzilai-Borwein步长的小批量稀疏随机方差缩减梯度法[J]. 计算机应用研究,2023,40(12):3655-3659.
QIN C D,YANG X. Mini-batch sparse stochastic variance reduced gradient method with randomly improved Barzilai-Borwein steps[J]. Application Research of Computers,2023,40(12):3655-3659.

[19] 王福胜,甄娜,李晓桐. R-线性收敛的重要样本抽样随机梯度下降算法[J]. 工程数学学报,2023,40(5):833-842.
WANG F S,ZHEN N,LI X T. R-linear convergence of importance sampling stochastic gradient decent algorithm[J]. Chinese Journal of Engineering Mathematics,2023,40(5):833-842.

[20] HUANG Y K,DAI Y H,LIU X W. Equipping the Barzilai-Borwein method with the two dimensional quadratic termination property[J]. SIAM Journal on Optimization,2021,31(4):3068-3096.

[21] ZAVRIEV S K,KOSTYUK F V. Heavy-ball method in nonconvex optimization problems[J]. Computational Mathematics and Modeling,1993,4(4):336-341.

[22] NESTEROV Y E. A method of solving a convex programming problem with convergence rate $O(1/k^2)$[J]. Proceedings of the USSR Academy of Sciences,1983,269(3):543-547.

[23] LIN Z C,LI H,FANG C. Accelerated optimization for machine learning:first-order algorithms[M]. Singapore:Springer,2020.

[24] NESTEROV Y. Introductory lectures on convex optimization:a basic course[M]. Boston:Springer,2004.

[25] GHADIMI S,LAN G. Accelerated gradient methods for nonconvex nonlinear and stochastic programming[J]. Mathematical Programming,2016,156(1/2):59-99.

[26] YANG T B,LIN Q H,LI Z. Unified convergence analysis of stochastic momentum methods for convex and non-convex optimization[EB/OL]. (2016-04-12)[2023-10-14]. http://export.arxiv.org/abs/1604.03257v1.

[27] LAN G. An optimal method for stochastic composite optimization[J]. Mathematical Programming,2012,133(1/2):365-397.

[28] NITANDA A. Stochastic proximal gradient descent with acceleration techniques[C]//GHAHRAMANI Z,WELLING M,CORTES C,et al. NIPS'14:Proceedings of the 27th International Conference on Neural Information Processing Systems. Cambridge:MIT Press,2014,1:1574-1582.

[29] ALLEN-ZHU Z. Katyusha:the first direct acceleration of stochastic gradient methods[J]. The Journal of Machine Learning Research,2017,18(1):8194-8244.

[30] ALLEN-ZHU Z. Katyusha X:practical momentum method for stochastic sum-of-nonconvex optimization[EB/OL]. (2018-02-

12)[2023-10-14]. https://arxiv.org/pdf/1802.03866.

[31] LUO Z J, CHEN S Y, QIAN Y T, et al. Multi-stage stochastic gradient method with momentum acceleration[J]. Signal Processing, 2021, 188(C): 108201.

[32] SHANG F H, JIAO L C, ZHOU K W, et al. ASVRG: accelerated proximal SVRG[EB/OL]. (2018-11-17)[2023-10-14]. https://arxiv.org/pdf/1810.03105.

[33] ZHOU K W, SHANG F H, CHENG J. A simple stochastic variance reduced algorithm with fast convergence rates[EB/OL]. (2018-06-28)[2023-10-14]. https://arxiv.org/pdf/1806.11027.

[34] 吉梦,何清龙. AdaSVRG:自适应学习率加速 SVRG[J]. 计算机工程与应用,2022,58(9):83-90.
JI M, HE Q L. AdaSVRG: accelerating SVRG by adaptive learning rate[J]. Computer Engineering and Applications, 2022, 58(9): 83-90.

[35] HE L L, YE J M, JIANWEI E. Accelerated stochastic variance reduction for a class of convex optimization problems[J]. Journal of Optimization Theory and Applications, 2023, 196(3): 810-828.

[36] NESTEROV Y. Introductory lectures on convex optimization[M]. New York: Springer Science & Business Media, 2004.

Operations Research and Cybernetics

A New Stochastic Variance Reduced Gradient Algorithm with BB-Like Step Size

CHEN Xuanrui, LIU Zexian, NI Yan

(School of Mathematics and Statistics, Guizhou University, Guiyang 550025, China)

Abstract: Adaptive step size is introduced into the Stochastic Variance Reduction Gradient(SVRG) algorithm, and the numerical performance of the algorithm is further improved on this basis. Firstly, the step size of the SVRG algorithm is calculated by BB step size with two-dimensional quadratic termination. Then the stopping criterion and negative momentum are introduced into the inner loop of the SVRG algorithm to accelerate the convergence rate. The proposed algorithm's numerical experiment was carried out using Matlab, and the numerical performance of the algorithm was observed. By analyzing the numerical experimental results of the algorithm, it is concluded that the algorithm's performance is comparable to that of the SVRG method with the optimal step size adjustment. In addition, the new algorithm is insensitive to the selection of the initial step size and can automatically generate the optimal step size.

Keywords: stochastic variance reduced gradient algorithm; BB-like step size; adaptive computing; negative momentum framework

(责任编辑 黄 颖)