非光滑强凸情形 Adam 型算法的最优收敛速率

陇 盛[1,3],陶 蔚[2],张泽东[3],陶 卿[3]
(1. 国防科技大学信息系统工程重点实验室,湖南长沙 410073; 2. 军事科学院战略评估咨询中心,北京 100091;
3. 陆军炮兵防空兵学院信息工程系,安徽合肥 230031)

摘 要: 对于非光滑强凸问题,在线梯度下降(Online Gradient Decent,OGD)取适当步长参数可以得到对数阶后悔界.然而,这并不能使一阶随机优化算法达到最优收敛速率. 为解决这一问题,研究者通常采取两种方案:其一是改进算法本身,另一种是修改算法输出方式. 典型的 Adam(Adaptive moment estimation)型算法 SAdam(Strongly convex Adaptive moment estimation)采用了改进算法的方式,并添加了自适应步长策略和动量技巧,虽然得到更好的数据依赖的后悔界,但在随机情形仍然达不到最优. 针对这个问题,本文改用加权平均的算法输出方式,并且重新设计与以往算法同阶的步长超参数,提出了一种名为 WSAdam(Weighted average Strongly convex Adaptive moment estimation)的 Adam 型算法. 证明了 WSAdam 达到了非光滑强凸问题的最优收敛速率. 经过 Reddi 问题的测试和在非光滑强凸函数优化中的实验,验证了所提方法的有效性.

关键词: 非光滑; 强凸优化; 自适应步长; 动量方法; Adam 型算法; 加权平均; 收敛速率
中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112(2022)09-2049-11
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20210589

The Optimal Convergence Rate of Adam-Type Algorithms for Non-Smooth Strongly Convex Cases

LONG Sheng[1,3],TAO Wei[2],ZHANG Ze-dong[3],TAO Qing[3]
(1. *Science and Technology on Information Systems Engineering Laboratory*, *National University of Defense Technology*, *Changsha*, *Hunan* 410073, *China*;
2. *Center for Strategic Assessment and Consulting*, *Academy of Military Science*, *Beijing* 100091, *China*;
3. *Department of Information Engineering*, *Army Academy of Artillery and Air Defense*, *Hefei*, *Anhui* 230031, *China*)

Abstract: For non-smooth strongly convex problems, the logarithmic regret bound can be obtained by online gradient descent with the appropriate step size parameter. However, it cannot make the first order stochastic optimization algorithm achieve the optimal convergence rate. To solve this problem, researchers usually adopt two schemes: one is to modify the iterate of the algorithm, the other is to change the output mode of the algorithm. SAdam, a typical Adam-type algorithm, modify the algorithm by adding adaptive step size strategy and momentum technique. Although it obtains a tighter data dependent regret bound, it still cannot reach the optimal bound in the stochastic case. To solve this problem, this paper redesigns the step size scheme which is the same order as the previous algorithm, uses the weighted average algorithm output mode, and proposes an Adam-type algorithm named WSAdam. It is proved that WSAdam achieves the optimal convergence rate for non-smooth strongly convex problems. The validity of the proposed method is verified by the test of Reddi´s problem and the experiment in the optimization of non-smooth strongly convex functions.

Key words: non-smooth; strongly convex; adaptive step size; momentum methods; Adam-type algorithms; weighted average; convergence rate

1 引言

在线学习(online learning)是用来分析迭代算法的流行框架,后悔界(regret bound)则是衡量在线优化算法性能的重要指标[1]. 针对一般凸优化问题,Zinkevich 提出的 OGD(Online Gradient Decent)[2]方法达到了最坏情况下 $O(\sqrt{T})$的后悔界,其中 T 是总迭代次数. 而在非光滑强凸

收稿日期:2021-05-09;修回日期:2021-11-02;责任编辑:梅志强
基金项目:国家自然科学基金(No.62076252, No.62106281)

情形中，Hazan等人在OGD基础上调整步长阶为$O(1/t)$，得到了更好的$O(\log T)$对数阶后悔界[3]，其中$t=1,2,\cdots,T$.

虽然在线学习理论和应用方面都取得成功，但是实验中模拟在线流程较为复杂，算法往往需要更简单的随机实验环境[4]．因此，本文关注OGD经过标准的online-to-batch技巧转换后，得到的随机算法SGD（Stochastic Gradient Decent）．两者本质上是同一算法，区别在于应用场景不同，OGD用后悔界度量其在线学习性能，SGD靠收敛速率评价在随机优化中的表现．在强凸情形中，SGD得到了$O(\log T/T)$的收敛速率．与之相比，Agarwal等人证明了在最好情况下，一阶随机优化算法解非光滑强凸问题的收敛速率是$\Omega(1/T)$[5]．为达到与之匹配的最坏情况下的最优收敛速率$O(1/T)$，许多算法在分析中引入了光滑条件（例如梯度Lipschitz连续、高阶可微等）．但是这些假设往往是不平凡的，并且无法应用于非光滑目标函数（例如hinge损失）．文献[6]提出一种结合了COMID（Composite Objective Mirror Descent）的非光滑随机坐标下降方法，不仅保持了正则化结构，而且计算代价极低，遗憾的是在强凸情形中未能达到最优．因此长期以来，SGD都无法跨过对数阶的鸿沟，达到非光滑强凸情形的最优收敛速率．

为了解决这个问题，研究者通常采取两种方案：其一是改进SGD算法本身，结合各种加速技巧提升算法收敛速率．2011年，Hazan等人提出著名的Epoch-GD（Epoch Gradient Descent）[7]，该算法其实是在SGD基础上引入了“多阶段循环”这个新的概念．虽然Epoch-GD达到了最优收敛速率$O(1/T)$，但Rakhlin等人认为，大幅修改算法不足以证明SGD彻底突破了强凸优化中对数因子的阻碍，因此提出了第二种方案——修改算法输出方式．在以往收敛性分析中，SGD输出全部T次迭代平均结果，Rakhlin提出在不改变算法的前提下，用α-suffix[8]方式（输出后半部分迭代平均）进行替换，最终达到了$O(1/T)$收敛速率．然而，α-suffix技巧也存在问题，首先它给收敛性分析增加了难度，其次不能以on-the-fly的模式存储历史迭代结果，从而增加了计算开销．幸运的是，文献[9~11]中采用的加权平均输出方式克服了这个缺点．该方法对理论分析十分友好，且只需对SGD每次迭代结果赋予权重值最后进行平均输出，就可以在支持on-the-fly计算方式的同时，保证最优的收敛速率．

近年来，在SGD基础上使用自适应梯度调整步长，并且用动量搜索方向的算法称为Adam型算法，例如Adam[12]、NAdam（Nesterov-accelerated Adaptive moment estimation）[13]、PAdam（Partially Adaptive moment estimation）[14]、Adaptive HB（Adaptive Polyak's Heavy-Ball）[15]等．这类算法在非光滑凸情形中保证$O(1/\sqrt{T})$的收敛速率，并且具有适合稀疏优化、体现不同维度差异等优点．然而文献[16]指出，在某些简单的凸环境中，所有基于指数移动平均（Exponential Moving Average，EMA）的Adam型算法都不收敛，这就是著名的Reddi问题．针对该问题，Reddi等人提出了改进算法AMSGrad[16]和AdamNC[16]．另一方面，Adam型算法在强凸优化中的应用也逐渐发展起来．2017年Mukkamala等人提出了SC-Adagrad（Strongly Convex Adagrad）[17]和SC-RMSProp（Strongly Convex RMSProp）[17]算法，应对在线学习问题得到了数据依赖（处理稀疏数据时表现更好）的对数阶后悔界．2018年，Chen等人在Epoch-GD基础上结合AdaGrad[18]提出了SadaGrad[19]，虽然在随机情形下得到了$O(1/T)$的最优收敛速率，但是只适用于弱强凸环境．2019年，Wang等人提出SAdam[20]，尽管在处理稀疏数据时得到比OGD更好的后悔界，体现出自适应步长方法的优势，但是转换为随机算法时只能得到$O(\log T/T)$的收敛速率，因此没有体现动量的加速作用，与最优收敛速率依然存在对数阶的间隙．

面对非光滑强凸优化问题，SGD能够得到最优收敛速率$O(1/T)$，但是到目前为止，SGD改良产生的Adam型算法反而无法达到上述目标．因此，如何使Adam型算法达到最优收敛亟待解决．正如文献[20]中所说，寄希望于SAdam与Epoch-GD技巧结合是不平凡的．综上所述，本文旨在基于动量法和自适应步长，结合修改输出方式这一技巧提出新的Adam型算法，保证其在非光滑强凸情形中达到最优收敛速率$O(1/T)$.

本文的主要贡献如下：

（1）提出了一种名为WSAdam的Adam型算法，该算法在SAdam基础上进行改进，采用加权平均的输出方式，设置了与以往强凸算法同阶的步长超参数．既保持了Adam型算法体现不同维度差异的优点，又通过on-the-fly计算降低了运行成本；

（2）针对约束的非光滑强凸优化问题，证明了本文所提的WSAdam随机情形下具有$O(1/T)$的最优收敛速率（见定理1）．据我们所知，这一结果消去了强凸优化中常见的对数阶因子，填补了Adam型算法强凸最优收敛性方面的缺失；

（3）证明了在导致Adam发散的优化问题[16]上，WSAdam仍能保持收敛，表明WSAdam可以解决Reddi问题．另外，选择了典型的l_2范数约束下的hinge损失函数强凸优化问题，通过与几种常见强凸算法进行比较实验，验证了理论分析的正确性，也表明所提算法优于现有的强凸Adam型算法．

2 相关工作

本文主要考虑求解如下非光滑约束优化问题：

$$\min f(\boldsymbol{w}),\ \text{s. t. } \boldsymbol{w}\in Q \tag{1}$$

其中$Q\in\mathbf{R}^d$是闭凸集,$\boldsymbol{w}^*=\arg\min\limits_{\boldsymbol{w}\in Q} f(\boldsymbol{w})$为式(1)的一个最优解,$f$是$Q$上的非光滑强凸函数,定义如下:

定义 1　如果$\exists\boldsymbol{\lambda}\in\mathbf{R}^d$对$\forall i=1,\cdots,d$有$\lambda_i>0$,且对$\forall\boldsymbol{u},\boldsymbol{w}\in Q$有下式成立:

$$f(\boldsymbol{u})\geq f(\boldsymbol{w})+\left\langle\nabla f(\boldsymbol{w}),\boldsymbol{u}-\boldsymbol{w}\right\rangle+\left\|\boldsymbol{u}-\boldsymbol{w}\right\|^2_{\mathrm{diag}(\lambda)}$$

那么称函数f为$\boldsymbol{\lambda}$-强凸.

在线学习的目标是最小化后悔界(Regret bound),定义如下:

$$\text{Regret bound}=\sum_{t=1}^{T}f_t(\boldsymbol{w}_t)-\min_{\boldsymbol{w}\in Q}\sum_{t=1}^{T}f_t(\boldsymbol{w})\tag{2}$$

其中$f_t(t=1,2,\cdots,T)$均为强凸函数,$f_t(\boldsymbol{w}_t)$表示f_t在$\boldsymbol{w}_t$处的损失. 常用优化器是OGD,见算法1.

算法 1　在线梯度下降(OGD)算法

输入:$\boldsymbol{w}_1=\boldsymbol{0}$

For $t=1$ to T:

Submit $\boldsymbol{w}_t$ and then receive f_t

Suffer loss $f_t(\boldsymbol{w}_t)$, and observe $\boldsymbol{g}_t$

Update $\boldsymbol{w}_{t+1}=P_Q\left[\boldsymbol{w}_t-\alpha_t\boldsymbol{g}_t\right]$

End for

输出:$\bar{\boldsymbol{w}}_T=\left(\boldsymbol{w}_1+\ldots+\boldsymbol{w}_T\right)/T$

算法1中α_t代表步长,$\boldsymbol{g}_t$表示$f_t(\boldsymbol{w}_t)$的次梯度,P_Q表示在Q上投影算子.

然而在线设置中,不可预见整体目标函数,需要学习环境响应上一轮迭代结果后提供损失f_t,然后才能观察到当前迭代的次梯度$\boldsymbol{g}_t$,因此不适用于算法的实验验证.

为方便实验,本文考虑随机情形. 假设全体样本集$\xi=\left\{\left(\boldsymbol{x}_i,y_i\right)\right\}_{i=1}^{n}$,目的是最小化期望形式的误差界(即上文所述收敛速率),表示如下:

$$\mathrm{E}\left[f\left(\bar{\boldsymbol{w}}_T\right)-f\left(\boldsymbol{w}^*\right)\right]\tag{3}$$

通常用SGD解得上述随机情形中的收敛速率,具体形式见算法2.

算法 2　随机梯度下降(SGD)算法

输入:$\boldsymbol{w}_1=\boldsymbol{0}$

For $t=1$ to T

Compute $\hat{\boldsymbol{g}}_t=\nabla f\left(\boldsymbol{w}_t,\xi_t\right)$

Update $\boldsymbol{w}_{t+1}=P_Q\left[\boldsymbol{w}_t-\alpha_t\hat{\boldsymbol{g}}_t\right]$

End for

输出:$\bar{\boldsymbol{w}}_T=\left(\boldsymbol{w}_1+\ldots+\boldsymbol{w}_T\right)/T$

算法2中α_t代表步长,$\xi_t\subseteq\xi$表示第t次迭代时随机抽取的样本,$\hat{\boldsymbol{g}}_t$表示$f\left(\boldsymbol{w}_t,\xi_t\right)$的次梯度.

SGD计算次梯度只与每轮随机抽取的样本相关,当假设全体样本独立同分布时,在第t次迭代时刻,关于部分样本的目标函数$f\left(\boldsymbol{w}_t,\xi_t\right)$的次梯度$\hat{\boldsymbol{g}}_t$是整个目标函数$f\left(\boldsymbol{w}_t,\xi\right)$次梯度的无偏估计,也就是$\mathrm{E}\left(\hat{\boldsymbol{g}}_t\right)=\nabla f\left(\boldsymbol{w}_t,\xi\right)$.

从算法2中看出,SGD输出所有迭代平均后的结果$\bar{\boldsymbol{w}}_T=\left(\boldsymbol{w}_1+\cdots+\boldsymbol{w}_T\right)/T$,这样只能得到次优的对数阶收敛速率. 为了使其达到最优收敛,提出了α-suffix技巧,即将原来的平均输出方式改为:

$$\bar{\boldsymbol{w}}_T^{\alpha}=\frac{\left(\boldsymbol{w}_{(1-\alpha)T+1}+\cdots+\boldsymbol{w}_T\right)}{\alpha T}\tag{4}$$

其中$\alpha\in(0,1)$,令αT为整数. 但是,这种方式需要将所有的迭代结果存入内存或者提前知道总迭代次数T,这极大增加了计算开销.

针对这个问题,一种能够on-the-fly计算的加权平均输出方式被提出:

$$\bar{\boldsymbol{w}}_T^{w}=\frac{2}{T(T+1)}\sum_{t=1}^{T}t\boldsymbol{w}_t=\left(1-\rho_T\right)\bar{\boldsymbol{w}}_{T-1}^{w}+\rho_T\boldsymbol{w}_T\tag{5}$$

其中$\rho_T=2/(T+1)$,从上式可以看出加权平均输出方式不需要存储所有的迭代,只要保存$\bar{\boldsymbol{w}}_{T-1}^{w}$就可以对$\bar{\boldsymbol{w}}_T^{w}$进行更新,这种方式极大提高了算法运行速度.

除了改进输出方式,升级为Adam型算法也是提高SGD性能的主要途径之一,其具体描述见算法3.

算法 3　Adam型算法

输入:$\boldsymbol{w}_1=\boldsymbol{0}$

For $t=1$ to T

Compute $\hat{\boldsymbol{g}}_t=\nabla f\left(\boldsymbol{w}_t,\xi_t\right)$

Update $\boldsymbol{m}_t$ and $\boldsymbol{V}_t$

Update $\boldsymbol{w}_{t+1}=P_Q^{\sqrt{\boldsymbol{V}_t}}\left[\boldsymbol{w}_t-\alpha_t\boldsymbol{m}_t/\sqrt{\boldsymbol{V}_t}\right]$

End for

输出:$\bar{\boldsymbol{w}}_T=\left(\boldsymbol{w}_1+\ldots+\boldsymbol{w}_T\right)/T$

算法3中$P_Q^{\sqrt{\boldsymbol{V}_t}}(\boldsymbol{u})=\arg\min\limits_{\boldsymbol{w}\in Q}\left\|\boldsymbol{w}-\boldsymbol{u}\right\|^2_{\sqrt{\boldsymbol{V}_t}}$表示$Q$上加权投影算子.

在算法3中,动量由历史梯度缓冲器$\boldsymbol{m}_t$承载,自适应步长由$\alpha_t/\boldsymbol{V}_t$构成. Adam型算法的自动调整步长机制,关键技术是平方梯度的指数移动平均(Exponential Moving Average,EMA):

$$\boldsymbol{V}_t=\beta\boldsymbol{V}_{t-1}+\left(1-\beta\right)\boldsymbol{g}_t^2\tag{6}$$

虽然该策略可以摒弃过早的梯度,并且避免训练提前终止,但是不能保证$\alpha_t/\boldsymbol{V}_t$是单调非增的. 迭代后期过大的步长可能导致算法不收敛,从而陷入Reddi问题(详细例子在第4节实验中描述).

解决方案是AMSGrad和AdamNC两种算法,SAdam在AdamNC基础上改进而来也有效避免了不收敛问题. 不同的$\alpha_t,\boldsymbol{m}_t,\boldsymbol{V}_t$设定方案对应不同Adam型算法,我们将常见的几种列举出来,后悔界和随机情形下收

敛速率对比如表1所示. 其中前三种算法针对一般凸函数,后三种针对强凸函数.

表1 常见Adam型算法对比

算法	$\alpha_t, \boldsymbol{m}_t, \boldsymbol{V}_t$	后悔界	收敛速率
Adam	$\alpha_t=\alpha/\sqrt{t}$ $\boldsymbol{m}_t=\beta_1\boldsymbol{m}_{t-1}+(1-\beta_1)\hat{\boldsymbol{g}}_t$ $\boldsymbol{V}_t=\mathrm{diag}(\boldsymbol{v}_t)+\delta\boldsymbol{I}_d/t$ $\boldsymbol{v}_t=\beta_2\boldsymbol{v}_{t-1}+(1-\beta_2)\hat{\boldsymbol{g}}_t^2$	$O(\sqrt{T})$	$O(1/\sqrt{T})$
AMSGrad	$\alpha_t=\alpha/\sqrt{t}$ $\boldsymbol{m}_t=\beta_1\boldsymbol{m}_{t-1}+(1-\beta_1)\hat{\boldsymbol{g}}_t$ $\boldsymbol{V}_t=\max(\boldsymbol{V}_{t-1},\hat{\boldsymbol{V}}_t)$ $\hat{\boldsymbol{V}}_t=\mathrm{diag}(\boldsymbol{v}_t)+\delta\boldsymbol{I}_d/t$ $\boldsymbol{v}_t=\beta_2\boldsymbol{v}_{t-1}+(1-\beta_2)\hat{\boldsymbol{g}}_t^2$	$O(\sqrt{T})$	$O(1/\sqrt{T})$
AdamNC	$\alpha_t=\alpha/\sqrt{t}$ $\boldsymbol{m}_t=\beta_1\boldsymbol{m}_{t-1}+(1-\beta_1)\hat{\boldsymbol{g}}_t$ $\boldsymbol{V}_t=\mathrm{diag}(\boldsymbol{v}_t)+\delta\boldsymbol{I}_d/t$ $\boldsymbol{v}_t=\beta_{2,t}\boldsymbol{v}_{t-1}+(1-\beta_{2,t})\hat{\boldsymbol{g}}_t^2$ $1-1/t\le\beta_{2,t}\le1-\gamma/t$ $\gamma\in(0,1)$	$O(\sqrt{T})$	$O(1/\sqrt{T})$
SC-Adagrad	$\alpha_t=\alpha$ $\boldsymbol{m}_t=\hat{\boldsymbol{g}}_t$ $\boldsymbol{V}_t=\mathrm{diag}(\boldsymbol{v}_t^2)+\delta\boldsymbol{I}_d$ $\boldsymbol{v}_t=\boldsymbol{v}_{t-1}+\hat{\boldsymbol{g}}_t^2$	$O(\log T)$	$O(\log T/T)$
SC-RMSProp	$\alpha_t=\alpha/t$ $\boldsymbol{m}_t=\hat{\boldsymbol{g}}_t$ $\boldsymbol{V}_t=\mathrm{diag}(\boldsymbol{v}_t^2)+\delta\boldsymbol{I}_d/t^2$ $\boldsymbol{v}_t=\beta_{2,t}\boldsymbol{v}_{t-1}+(1-\beta_{2,t})\hat{\boldsymbol{g}}_t^2$ $1-1/t\le\beta_t\le1-\gamma/t$ $\gamma\in(0,1)$	$O(\log T)$	$O(\log T/T)$
SAdam	$\alpha_t=\alpha/t$ $\boldsymbol{m}_t=\beta_1\boldsymbol{m}_{t-1}+(1-\beta_1)\hat{\boldsymbol{g}}_t$ $\boldsymbol{V}_t=\mathrm{diag}(\boldsymbol{v}_t^2)+\delta\boldsymbol{I}_d/t^2$ $\boldsymbol{v}_t=\beta_{2,t}\boldsymbol{v}_{t-1}+(1-\beta_{2,t})\hat{\boldsymbol{g}}_t^2$ $1-1/t\le\beta_t\le1-\gamma/t$ $\gamma\in(0,1)$	$O(\log T)$	$O(\log T/T)$

表1中α为某一固定参数,向量或矩阵间运算$(\cdot)^2$、$\sqrt{(\cdot)}$、$\max(\cdot)$都是基于元素的,$\mathrm{diag}(\cdot)$是取对角矩阵运算,$\boldsymbol{I}_d$是d维单位矩阵,δ是平滑系数.

3 WSAdam算法

对于非光滑强凸优化问题,为了构造达到最优收敛速率$O(1/T)$的新算法,我们的思路是在SAdam基础上,重新设计与以往强凸算法同阶的步长超参数(即最终步长满足$O(1/t)$),摒弃以往的标准平均输出方式,用加权平均输出方式取代之. 本文提出的WSAdam算法见算法4.

算法4 WSAdam算法

输入: $\boldsymbol{w}_1=\boldsymbol{0}$

For $t=1$ to T

Compute $\hat{\boldsymbol{g}}_t=\nabla f(\boldsymbol{w}_t,\xi_t)$

Update $\boldsymbol{m}_t=\beta_{1,t}\boldsymbol{m}_{t-1}+(1-\beta_{1,t})\hat{\boldsymbol{g}}_t$

Update $\boldsymbol{V}_t=\beta_{2,t}\boldsymbol{V}_{t-1}+(1-\beta_{2,t})\mathrm{diag}(\hat{\boldsymbol{g}}_t^2)$

Update $\hat{\boldsymbol{V}}_t=\boldsymbol{V}_t+\delta\boldsymbol{I}_d$

Update $\boldsymbol{w}_{t+1}=P_Q^{\hat{V}_t}\left[\boldsymbol{w}_t-\alpha_t\boldsymbol{m}_t\hat{\boldsymbol{V}}_t^{-1}\right]$

End for

输出: $\bar{\boldsymbol{w}}_T^w=\dfrac{2}{T(T+1)}\sum_{t=1}^{T}t\boldsymbol{w}_t$

算法4中,$\beta_{1,t}=\beta_1v^{t-1}$,$\beta_1=0.9$,$v\in(0,1)$,$1-1/t\le\beta_{2,t}\le1-\gamma/t$,$\gamma\in(0,1)$,$\delta$是平滑系数取值1e−8,$\boldsymbol{I}_d$是$d$维单位矩阵,$\alpha_t=\alpha/\left[(1-\beta_{1,t})(t+1)\right]$,$\alpha$为某一固定参数,$\hat{\boldsymbol{V}}_t^{-1}$为$\hat{\boldsymbol{V}}_t$的逆.

从算法4可以看出,WSAdam的步长为对角矩阵$\alpha_t\hat{\boldsymbol{V}}_t^{-1}$,展开其第$i$维如下:

$$\frac{\alpha}{(t+1)(1-\beta_{1,t})\left[V_{t,i}+\delta\right]}\le\frac{\alpha}{(t+1)(1-\beta_1)\delta}\tag{7}$$

其中$V_{t,i}=\beta_{2,t}V_{t-1,i}+(1-\beta_{2,t})\hat{\boldsymbol{g}}_{t,i}^2$恒大于0.

由式(7)可知,WSAdam的有效步长为$O(1/t)$,与以往强凸算法步长同阶. 由于$V_{t,i}+\delta$积累矩阵第i维度数值,算法步长因此在不同维度上得到加权区分,从而在不同待训参数之间体现出差异性.

另一方面,WSAdam采用加权平均的输出方式,保持了on-the-fly计算的优点,更为重要的一点是,所加权重消去了导致以往算法产生对数阶的结构,因此能够达到最优收敛,这将在下一节中展开说明.

4 WSAdam算法收敛速率分析

为了达到非光滑强凸情形的最优收敛速率,我们首先寻找SGD产生对数阶的原因,然后介绍加权平均输出技巧解决此问题的原理.

首先,我们需要给出一些假设条件,这些假设在以往收敛性分析中普遍存在.

假设1 存在常数$G>0$和$G_\infty>0$使得:

$$\max_{t\ge1}\|\hat{\boldsymbol{g}}_t\|_2\le G,\ \max_{t\ge1}\|\hat{\boldsymbol{g}}_t\|_\infty\le G_\infty$$

假设2 存在常数$D>0$和$D_\infty>0$使得:

$$\max_{\forall\boldsymbol{w},\boldsymbol{z}\in Q}\|\boldsymbol{w}-\boldsymbol{z}\|_2\le D,\ \max_{\forall\boldsymbol{w},\boldsymbol{z}\in Q}\|\boldsymbol{w}-\boldsymbol{z}\|_\infty\le D_\infty$$

然后,根据文献[9]中对强凸SGD的分析得下式:

$$f(\boldsymbol{w}_t)-f(\boldsymbol{w}^*)\le\frac{(\alpha_t^{-1}-\lambda_{\min})\|\boldsymbol{w}_t-\boldsymbol{w}^*\|^2}{2}-\frac{\alpha_t^{-1}\|\boldsymbol{w}_{t+1}-\boldsymbol{w}^*\|^2}{2}+\frac{\alpha_tG^2}{2}$$

其中,α_t是步长,$\lambda_{\min}$是强凸系数$\boldsymbol{\lambda}$中的最小元素值. 令$\alpha_t=1/(\lambda_{\min}t)$上式得:

$$f(\boldsymbol{w}_t)-f(\boldsymbol{w}^*)\le\frac{\lambda_{\min}(t-1)\|\boldsymbol{w}_t-\boldsymbol{w}^*\|^2}{2}-\frac{\lambda_{\min}t\|\boldsymbol{w}_{t+1}-\boldsymbol{w}^*\|^2}{2}+\frac{G^2}{2\lambda_{\min}t}$$

对上式从$t=1$到$t=T$求和得:

$$\sum_{t=1}^{T}f(\boldsymbol{w}_{t-1})-\sum_{t=1}^{T}f(\boldsymbol{w}^*)\le\frac{\lambda_{\min}}{2}\left(0-T\|\boldsymbol{w}_{T+1}-\boldsymbol{w}^*\|^2\right)+\frac{G^2}{2\lambda_{\min}}\sum_{t=1}^{T}\frac{1}{t}\le\frac{G^2}{2\lambda_{\min}}(1+\log T)$$

从上式第二行可以观察到,前一项为负数可以放缩消去,第二项导致了对数因子的产生.

因此我们着重处理后一项,采用权重为t的加权平均输出方式,令$\alpha_t=2/(\lambda_{\min}(t+1))$,上式不等号两边同时乘$t$得到:

$$t\left[f(\boldsymbol{w}_t)-f(\boldsymbol{w}^*)\right]\le\frac{\lambda_{\min}(t-1)t\|\boldsymbol{w}_t-\boldsymbol{w}^*\|^2}{4}-\frac{\lambda_{\min}t(t+1)\|\boldsymbol{w}_{t+1}-\boldsymbol{w}^*\|^2}{4}+\frac{tG^2}{\lambda_{\min}(t+1)}$$

$$\le\frac{\lambda_{\min}}{4}\left[(t-1)t\|\boldsymbol{w}_{t-1}-\boldsymbol{w}^*\|^2-t(t+1)\|\boldsymbol{w}_t-\boldsymbol{w}^*\|^2\right]+\frac{G^2}{\lambda_{\min}}$$

观察上式最后一行,发现后一项上的$1/t$已被消去,此时从$t=1$到$t=T$求和不会再产生对数因子,做加权平均可得如下最优收敛速率:

$$f\left(\frac{2}{T(T+1)}\sum_{t=1}^{T}t\boldsymbol{w}_{t-1}\right)-f(\boldsymbol{w}^*)\le\frac{2}{T(T+1)}\sum_{t=1}^{T}t\left[f(\boldsymbol{w}_{t-1})-f(\boldsymbol{w}^*)\right]\le\frac{2G^2}{(T+1)\lambda_{\min}}$$

本文将上述原理迁移到WSAdam算法的收敛性分析中,此外,还需要如下引理.

引理1　假设$1\le t\le T,0<v<1$,$f(v)$表示关于v的函数,$(f(v))'$表示$f(v)$的导函数,则有下式成立:

$$\sum_{t=1}^{T}tv^{t-1}\le\frac{1}{(1-v)^2}$$

证明

$$\sum_{t=1}^{T}tv^{t-1}=\sum_{t=0}^{T-1}(t+1)v^t=\left(\sum_{t=0}^{T-1}v^{t+1}\right)'=\left(\frac{v(1-v^T)}{1-v}\right)'\le\frac{1}{(1-v)^2}$$

引理1证毕.

定理1　令假设1和假设2成立,$\{\boldsymbol{w}_t\}_{t=1}^{\infty}$由算法4产生,$\alpha\ge\dfrac{(3-\gamma)G_\infty^2+2\delta}{2\min_i(\lambda_i)}$,$f$满足定义1中的$\boldsymbol{\lambda}$-强凸性质,$\boldsymbol{w}^*\in Q$为问题式(1)的一个最优解,结合引理1,随机WSAdam能够保证如下收敛速率:

$$\mathrm{E}\left[f\left(\frac{2}{T(T+1)}\sum_{t=1}^{T}t\boldsymbol{w}_t\right)-f(\boldsymbol{w}^*)\right]\le\frac{2(1-\gamma)dD_\infty^2G_\infty^2}{(T+1)}+\frac{2adD_\infty^2}{(T+1)(1-\beta_1)^4\delta}+\frac{dD_\infty^2\beta_1(G_\infty^2+\delta)}{a(1-v^2)T}$$

注意,上式表明WSAdam具有$O(1/T)$的最优收敛速率. 与SAdam达到$O(\log T/T)$次优收敛速率相比,WSAdam体现出了动量方法的加速性,填补了Adam型算法在非光滑强凸情形最优收敛性方面的缺失.

证明　根据算法5中步骤7,由投影非扩张性可得:

$$\begin{aligned}\|\boldsymbol{w}_{t+1}-\boldsymbol{w}^*\|_{\hat{V}_t}^2&=\left\|\Pi_D^{\hat{V}_t}\left(\boldsymbol{w}_t-a_t\hat{\boldsymbol{V}}_t^{-1}[\beta_{1,t}\boldsymbol{m}_{t-1}+(1-\beta_{1,t})\hat{\boldsymbol{g}}_t]\right)-\boldsymbol{w}^*\right\|_{\hat{V}_t}^2\\&\le\left\|\boldsymbol{w}_t-a_t\hat{\boldsymbol{V}}_t^{-1}[\beta_{1,t}\boldsymbol{m}_{t-1}+(1-\beta_{1,t})\hat{\boldsymbol{g}}_t]-\boldsymbol{w}^*\right\|_{\hat{V}_t}^2\\&=\|\boldsymbol{w}_t-\boldsymbol{w}^*\|_{\hat{V}_t}^2-2a_t\beta_{1,t}(\boldsymbol{w}_t-\boldsymbol{w}^*)^{\mathrm{T}}\boldsymbol{m}_{t-1}\\&\quad-2a_t(1-\beta_{1,t})(\boldsymbol{w}_t-\boldsymbol{w}^*)^{\mathrm{T}}\hat{\boldsymbol{g}}_t+a_t^2\|\boldsymbol{m}_{t-1}\|_{\hat{V}_t^{-1}}^2\end{aligned}$$

上式移项,两边除以$2a_t(1-\beta_{1,t})$得:

$$(\boldsymbol{w}_t-\boldsymbol{w}^*)^{\mathrm{T}}\hat{\boldsymbol{g}}_t\le\frac{1}{2a_t(1-\beta_{1,t})}\left[\|\boldsymbol{w}_t-\boldsymbol{w}^*\|_{\hat{V}_t}^2-\|\boldsymbol{w}_{t+1}-\boldsymbol{w}^*\|_{\hat{V}_t}^2\right]+\frac{\beta_{1,t}}{(1-\beta_{1,t})}(\boldsymbol{w}^*-\boldsymbol{w}_t)^{\mathrm{T}}\boldsymbol{m}_{t-1}+\frac{a_t\|\boldsymbol{m}_{t-1}\|_{\hat{V}_t^{-1}}^2}{2(1-\beta_{1,t})}$$

因为$f(\boldsymbol{w}_t,\xi_t)$满足$\boldsymbol{\lambda}$-强凸,联立上式得:

$$\begin{aligned}&f(\boldsymbol{w}_t,\xi_t)-f(\boldsymbol{w}^*,\xi_t)\le(\boldsymbol{w}_t-\boldsymbol{w}^*)^{\mathrm{T}}\hat{\boldsymbol{g}}_t-\|\boldsymbol{w}_t-\boldsymbol{w}^*\|_{\mathrm{diag}(\boldsymbol{\lambda})}^2\\&=\frac{1}{2a_t(1-\beta_{1,t})}\left[\|\boldsymbol{w}_t-\boldsymbol{w}^*\|_{\hat{V}_t}^2-\|\boldsymbol{w}_{t+1}-\boldsymbol{w}^*\|_{\hat{V}_t}^2\right]+\frac{\beta_{1,t}}{(1-\beta_{1,t})}(\boldsymbol{w}^*-\boldsymbol{w}_t)^{\mathrm{T}}\boldsymbol{m}_{t-1}+\frac{a_t\|\boldsymbol{m}_{t-1}\|_{\hat{V}_t^{-1}}^2}{2(1-\beta_{1,t})}-\|\boldsymbol{w}_t-\boldsymbol{w}^*\|_{\mathrm{diag}(\boldsymbol{\lambda})}^2\\&\le\frac{1}{2a_t(1-\beta_{1,t})}\left[\|\boldsymbol{w}_t-\boldsymbol{w}^*\|_{\hat{V}_t}^2-\|\boldsymbol{w}_{t+1}-\boldsymbol{w}^*\|_{\hat{V}_t}^2\right]+\frac{\beta_{1,t}\|\boldsymbol{w}_t-\boldsymbol{w}^*\|_{\hat{V}_t}^2}{2a_t(1-\beta_{1,t})}\\&\quad+\frac{\beta_{1,t}a_t\|\boldsymbol{m}_{t-1}\|_{\hat{V}_t^{-1}}^2}{2(1-\beta_{1,t})}+\frac{a_t\|\boldsymbol{m}_{t-1}\|_{\hat{V}_t^{-1}}^2}{2(1-\beta_{1,t})}-\|\boldsymbol{w}_t-\boldsymbol{w}^*\|_{\mathrm{diag}(\boldsymbol{\lambda})}^2\end{aligned}$$

上式不等号两边同时取期望得:

$$\begin{aligned}&\mathrm{E}\left[f(\boldsymbol{w}_t)-f(\boldsymbol{w}^*)\right]\\&\le\mathrm{E}\left[\frac{1}{2a_t(1-\beta_{1,t})}\left[\|\boldsymbol{w}_t-\boldsymbol{w}^*\|_{\hat{V}_t}^2-\|\boldsymbol{w}_{t+1}-\boldsymbol{w}^*\|_{\hat{V}_t}^2\right]-\|\boldsymbol{w}_t-\boldsymbol{w}^*\|_{\mathrm{diag}(\boldsymbol{\lambda})}^2\right]\\&+\mathrm{E}\left[\frac{\beta_{1,t}\|\boldsymbol{w}_t-\boldsymbol{w}^*\|_{\hat{V}_t}^2}{2a_t(1-\beta_{1,t})}+\frac{\beta_{1,t}a_t\|\boldsymbol{m}_{t-1}\|_{\hat{V}_t^{-1}}^2}{2(1-\beta_{1,t})}+\frac{a_t\|\boldsymbol{m}_{t-1}\|_{\hat{V}_t^{-1}}^2}{2(1-\beta_{1,t})}\right]\end{aligned}$$

上式不等号两边同乘以t,并从$t=1$到$t=T$求和得:

$$\sum_{t=1}^{T}t\mathrm{E}\left[f\left(\boldsymbol{w}_t\right)-f\left(\boldsymbol{w}^*\right)\right]\le\underbrace{\sum_{t=1}^{T}\mathrm{E}\left[\frac{t}{2a_t\left(1-\beta_{1,t}\right)}\left(\left\|\boldsymbol{w}_t-\boldsymbol{w}^*\right\|_{\hat{\boldsymbol{V}}_t}^2-\left\|\boldsymbol{w}_{t+1}-\boldsymbol{w}^*\right\|_{\hat{\boldsymbol{V}}_t}^2\right)-t\left\|\boldsymbol{w}_t-\boldsymbol{w}^*\right\|_{\mathrm{diag}(\boldsymbol{\lambda})}^2\right]}_{P_1}$$
$$+\underbrace{\sum_{t=1}^{T}\mathrm{E}\left[\frac{t\beta_{1,t}a_t\left\|\boldsymbol{m}_{t-1}\right\|_{\hat{\boldsymbol{V}}_t^{-1}}^2}{2\left(1-\beta_{1,t}\right)}+\frac{ta_t\left\|\boldsymbol{m}_{t-1}\right\|_{\hat{\boldsymbol{V}}_t^{-1}}^2}{2\left(1-\beta_{1,t}\right)}\right]}_{P_2}+\underbrace{\sum_{t=1}^{T}\mathrm{E}\left[\frac{t\beta_{1,t}\left\|\boldsymbol{w}_t-\boldsymbol{w}^*\right\|_{\hat{\boldsymbol{V}}_t}^2}{2a_t\left(1-\beta_{1,t}\right)}\right]}_{P_3}$$

首先处理P_1：

$$\begin{aligned}P_1=&\sum_{t=2}^{T}\mathrm{E}\left[\frac{t\left\|\boldsymbol{w}_t-\boldsymbol{w}^*\right\|_{\hat{\boldsymbol{V}}_t}^2}{2a_t\left(1-\beta_{1,t}\right)}-\frac{(t-1)\left\|\boldsymbol{w}_t-\boldsymbol{w}^*\right\|_{\hat{\boldsymbol{V}}_{t-1}}^2}{2a_{t-1}\left(1-\beta_{1,t-1}\right)}-t\left\|\boldsymbol{w}_t-\boldsymbol{w}^*\right\|_{\mathrm{diag}(\boldsymbol{\lambda})}^2\right]\\&+\frac{\left\|\boldsymbol{w}_1-\boldsymbol{w}^*\right\|_{\hat{\boldsymbol{V}}_1}^2}{2a_1\left(1-\beta_{1,1}\right)}-\frac{T\left\|\boldsymbol{w}_{T+1}-\boldsymbol{w}^*\right\|_{\hat{\boldsymbol{V}}_T}^2}{2a_T\left(1-\beta_{1,T}\right)}-\left\|\boldsymbol{w}_1-\boldsymbol{w}^*\right\|_{\mathrm{diag}(\boldsymbol{\lambda})}^2\\\le&\sum_{t=2}^{T}\mathrm{E}\left[\frac{\left\|\boldsymbol{w}_t-\boldsymbol{w}^*\right\|_{t(t+1)\hat{\boldsymbol{V}}_t-t(t-1)\hat{\boldsymbol{V}}_{t-1}-2at\mathrm{diag}(\boldsymbol{\lambda})}^2}{2a}\right]+\frac{\left\|\boldsymbol{w}_1-\boldsymbol{w}^*\right\|_{\hat{\boldsymbol{V}}_1}^2}{a}\\&-\left\|\boldsymbol{w}_1-\boldsymbol{w}^*\right\|_{\mathrm{diag}(\boldsymbol{\lambda})}^2\end{aligned}$$

考虑$t(t+1)\hat{\boldsymbol{V}}_t-t(t-1)\hat{\boldsymbol{V}}_{t-1}-2at\mathrm{diag}(\boldsymbol{\lambda})$的第$i$维元素并代入$a\ge\dfrac{(3-\gamma)G_\infty^2+2\delta}{2\min_i(\lambda_i)}$得：

$$\begin{aligned}&t(t+1)\hat{\boldsymbol{V}}_{t,i}-t(t-1)\hat{\boldsymbol{V}}_{t-1,i}-2at\lambda_i\\=&t(t+1)\beta_2\boldsymbol{V}_{t-1,i}+t(t+1)\left(1-\beta_2\right)\hat{\boldsymbol{g}}_{t,i}^2-t(t-1)\boldsymbol{V}_{t-1,i}\\&+2t\delta-2at\lambda_i\\\le&(t+1)\left(t-\gamma\right)\boldsymbol{V}_{t-1,i}+(t+1)\hat{\boldsymbol{g}}_{t,i}^2-t(t-1)\boldsymbol{V}_{t-1,i}+2t\delta-2at\lambda_i\\=&\left(2t-t\gamma-\gamma\right)\boldsymbol{V}_{t-1,i}+(t+1)\hat{\boldsymbol{g}}_{t,i}^2+2t\delta-2at\lambda_i\\\le&\left(3t-t\gamma\right)G_\infty^2+\left(1-\gamma\right)G_\infty^2+2t\delta-2at\lambda_i\\=&t\left(\left(3-\gamma\right)G_\infty^2+2\delta\right)+\left(1-\gamma\right)G_\infty^2-2at\lambda_i\le\left(1-\gamma\right)G_\infty^2\end{aligned}$$

即：

$$\left\|\boldsymbol{w}_t-\boldsymbol{w}^*\right\|_{t(t+1)\hat{\boldsymbol{V}}_t-t(t-1)\hat{\boldsymbol{V}}_{t-1}-2at\mathrm{diag}(\boldsymbol{\lambda})}^2\le\sum_{i=1}^{d}\left(\boldsymbol{w}_{t,i}-\boldsymbol{w}_i^*\right)\left(1-\gamma\right)G_\infty^2$$

$$\begin{aligned}P_1\le&\sum_{i=1}^{d}\frac{\left(\boldsymbol{w}_{1,i}-\boldsymbol{w}_i^*\right)^2\left(\hat{\boldsymbol{V}}_{1,i}-a\lambda_i\right)}{a}\\&+\sum_{t=2}^{T}\mathrm{E}\left[\sum_{i=1}^{d}\left(\boldsymbol{w}_{t,i}-\boldsymbol{w}_i^*\right)\left(1-\gamma\right)G_\infty^2\right]\\\le&\sum_{i=1}^{d}\frac{\left(\boldsymbol{w}_{1,i}-\boldsymbol{w}_i^*\right)^2\left(G_\infty^2+\delta-\dfrac{\left(3-\gamma\right)G_\infty^2+2\delta}{2}\right)}{a}\\&+\sum_{t=2}^{T}\mathrm{E}\left[\sum_{i=1}^{d}\left(\boldsymbol{w}_{t,i}-\boldsymbol{w}_i^*\right)\left(1-\gamma\right)G_\infty^2\right]\\\le&\sum_{i=1}^{d}\frac{\left(\boldsymbol{w}_{1,i}-\boldsymbol{w}_i^*\right)^2\left(\gamma-1\right)G_\infty^2}{2a}\\&+\sum_{t=2}^{T}\mathrm{E}\left[\sum_{i=1}^{d}\left(\boldsymbol{w}_{t,i}-\boldsymbol{w}_i^*\right)\left(1-\gamma\right)G_\infty^2\right]\\\le&\sum_{t=2}^{T}\mathrm{E}\left[\sum_{i=1}^{d}\left(\boldsymbol{w}_{t,i}-\boldsymbol{w}_i^*\right)\left(1-\gamma\right)G_\infty^2\right]\le\left(1-\gamma\right)dD_\infty^2G_\infty^2T\end{aligned}$$

然后处理P_2，由$\boldsymbol{m}_0=\boldsymbol{0},\beta_{1,t}\le1,\beta_{1,t}\le\beta_{1,t-1}$得：

$$\begin{aligned}P_2=&\sum_{t=1}^{T}\mathrm{E}\left[\frac{t\beta_{1,t}a_t\left\|\boldsymbol{m}_{t-1}\right\|_{\hat{\boldsymbol{V}}_t^{-1}}^2}{2\left(1-\beta_{1,t}\right)}+\frac{ta_t\left\|\boldsymbol{m}_{t-1}\right\|_{\hat{\boldsymbol{V}}_t^{-1}}^2}{2\left(1-\beta_{1,t}\right)}\right]\\\le&\sum_{t=1}^{T}\mathrm{E}\left[\frac{ta_t\left\|\boldsymbol{m}_{t-1}\right\|_{\hat{\boldsymbol{V}}_t^{-1}}^2}{2\left(1-\beta_{1,t}\right)}\right]+\sum_{t=1}^{T}\mathrm{E}\left[\frac{ta_t\left\|\boldsymbol{m}_{t-1}\right\|_{\hat{\boldsymbol{V}}_t^{-1}}^2}{2\left(1-\beta_{1,t}\right)}\right]\\=&\sum_{t=1}^{T-1}\mathrm{E}\left[\frac{(t+1)a_{t+1}\left\|\boldsymbol{m}_t\right\|_{\hat{\boldsymbol{V}}_{t+1}^{-1}}^2}{\left(1-\beta_{1,t+1}\right)}\right]\le\sum_{t=1}^{T}\mathrm{E}\left[\frac{(t+1)a\left\|\boldsymbol{m}_t\right\|_{\hat{\boldsymbol{V}}_t^{-1}}^2}{(t+1)\left(1-\beta_{1,t}\right)^2}\right]\end{aligned}$$

将$\boldsymbol{m}_t$和$\hat{\boldsymbol{V}}_t^{-1}$展开得：

$$\begin{aligned}P_2\le&\sum_{t=1}^{T}\mathrm{E}\left[\sum_{i=1}^{d}\frac{(t+1)a\left[\beta_{1,t}\boldsymbol{m}_{t-1,i}+\left(1-\beta_{1,t}\right)\hat{\boldsymbol{g}}_{t,i}\right]^2}{(t+1)\left(1-\beta_{1,t}\right)^2\left[\boldsymbol{V}_{t,i}+\delta\right]}\right]\\\le&\sum_{t=1}^{T}\mathrm{E}\left[\sum_{i=1}^{d}\frac{a\left[\sum_{j=1}^{t}\left(1-\beta_{1,j}\right)\prod_{k=1}^{t-j}\beta_{1,(t-k+1)}\hat{\boldsymbol{g}}_{j,i}\right]^2}{\left(1-\beta_{1,t}\right)^2\delta}\right]\\\le&\sum_{t=1}^{T}\mathrm{E}\left[\sum_{i=1}^{d}\frac{a\left[\sum_{j=1}^{t}\prod_{k=1}^{t-j}\beta_{1,(t-k+1)}\right]\left[\sum_{j=1}^{t}\prod_{k=1}^{t-j}\beta_{1,(t-k+1)}\hat{\boldsymbol{g}}_{j,i}^2\right]}{\left(1-\beta_{1,t}\right)^2\delta}\right]\\\le&\sum_{t=1}^{T}\mathrm{E}\left[\sum_{i=1}^{d}\frac{a\left[\sum_{j=1}^{t}\prod_{k=1}^{t-j}\beta_{1,(t-k+1)}\hat{\boldsymbol{g}}_{j,i}^2\right]}{\left(1-\beta_1\right)^3\delta}\right]\end{aligned}$$

$$\le \sum_{t=1}^{T}\mathrm{E}\left[\sum_{i=1}^{d}\frac{a\left[\sum_{j=1}^{t}\beta_1^{t-j}\hat{\boldsymbol{g}}_{j,i}^2\right]}{(1-\beta_1)^3\delta}\right]$$

$$\le \sum_{t=1}^{T}\frac{adG_\infty^2}{(1-\beta_1)^4\delta} \le \frac{adTG_\infty^2}{(1-\beta_1)^4\delta}$$

最后处理P_3:

$$P_3=\sum_{t=1}^{T}\mathrm{E}\left[\frac{t(t+1)\beta_{1,t}\left\|\boldsymbol{w}_t-\boldsymbol{w}^*\right\|_{\hat{V}_t}^2}{2a}\right]$$

$$\le \sum_{t=1}^{T}\mathrm{E}\left[\sum_{i=1}^{d}\frac{t(t+1)\beta_{1,t}\left(\boldsymbol{w}_{t,i}-\boldsymbol{w}_i^*\right)^2\left(G_\infty^2+\delta\right)}{2a}\right]$$

$$\le \frac{dD_\infty^2\left(G_\infty^2+\delta\right)}{2a}\sum_{t=1}^{T}t(t+1)\beta_{1,t}$$

$$\le \frac{dD_\infty^2\beta_1\left(G_\infty^2+\delta\right)(T+1)}{2a}\sum_{t=1}^{T}tv^{t-1}$$

上式结合引理1得:

$$P_3\le \frac{dD_\infty^2\beta_1\left(G_\infty^2+\delta\right)(T+1)}{2a\left(1-v^2\right)}$$

联立P_1,P_2,P_3得:

$$\sum_{t=1}^{T}t\mathrm{E}\left[f\left(\boldsymbol{w}_t\right)-f\left(\boldsymbol{w}^*\right)\right]$$

$$\le (1-\gamma)dD_\infty^2G_\infty^2T+\frac{adTG_\infty^2}{(1-\beta_1)^4\delta}+\frac{dD_\infty^2\beta_1\left(G_\infty^2+\delta\right)(T+1)}{2a\left(1-v^2\right)}$$

上式两边同时乘以$\frac{2}{T(T+1)}$得:

$$\frac{2}{T(T+1)}\sum_{t=1}^{T}t\mathrm{E}\left[f\left(\boldsymbol{w}_t\right)-f\left(\boldsymbol{w}^*\right)\right]$$

$$\le \frac{2(1-\gamma)dD_\infty^2G_\infty^2}{(T+1)}+\frac{2adG_\infty^2}{(T+1)(1-\beta_1)^4\delta}+\frac{dD_\infty^2\beta_1\left(G_\infty^2+\delta\right)}{a\left(1-v^2\right)T}$$

由凸函数基本性质得最终加权平均收敛速率:

$$\mathrm{E}\left[f\left(\frac{2}{T(T+1)}\sum_{t=1}^{T}t\boldsymbol{w}_t\right)-f\left(\boldsymbol{w}^*\right)\right]$$

$$\le \frac{2}{T(T+1)}\sum_{t=1}^{T}t\mathrm{E}\left[f\left(\boldsymbol{w}_t\right)-f\left(\boldsymbol{w}^*\right)\right]$$

$$=\frac{2(1-\gamma)dD_\infty^2G_\infty^2}{(T+1)}+\frac{2adG_\infty^2}{(T+1)(1-\beta_1)^4\delta}+\frac{dD_\infty^2\beta_1\left(G_\infty^2+\delta\right)}{a\left(1-v^2\right)T}$$

定理1证毕.

5　实验

本节分两部分对上一节中最优收敛速率的理论分析进行实验验证. 第一部分验证WSAdam算法能够解决Reddi问题;第二部分验证WSAdam在非光滑强凸情形优于现有算法.

5.1　Reddi问题的实验结果与分析

2018年,Reddi等人证明了Adam算法在优化一个经过特殊构造的一般凸函数时发散. 事实上,所有基于EMA技巧的Adam型算法都有可能存在这个问题,也被称为Reddi问题:

考虑如下定义域为$[-1,+1]$的线性函数序列:

$$f_t(w)=\begin{cases}Cw, & \text{for } t \bmod 3=1\\ -w, & \text{otherwise}\end{cases}$$

其中$C=3$. 在这个函数序列中,可以明显看出当$w=-1$时得到最小的后悔界. 然而,Adam错误地将参数指向+1方向进行更新,导致不收敛.

本文实验设置初始$w=1$,$t=[1,5000]$,将WSAdam与其他4种经典Adam型算进行比较,观察它们解上述在线优化问题的表现. 为公平起见,所有算法统一设置参数$\alpha=0.5$,$\beta_1=0$,$\delta=1\mathrm{e}-8$,Adam和AMSGrad均设置$\beta_2=0.1$,AdamNC、SAdam和WSAdam均设置$\beta_{2,t}=1-0.1/t$. 实验结果如图1所示,其中图1(a)的横坐标代表迭代次数(t),纵坐标代表平均后悔界(Regret bound/t);图1(b)的横坐标代表迭代次数(t),纵坐标代表参数(w). 如图1所示,在迭代5000次后,Adam参数值$w=+1$是次优解,平均后悔界无法收敛到0,从而证实了Reddi问题. AMSGrad、AdamNC、SAdam和WSAdam的参数值$w=-1$达到了最优解,且平均后悔界均收敛到0,证实了这些算法改进Adam是有效的,成功解决了Reddi问题.

另外,从图1(a)中还可以看出,强凸算法SAdam、WSAdam比一般凸算法Adam、AMSGrad、AdamNC收敛更快,说明SAdam、WSAdam对一般凸函数同样适用,并且本文所提WSAdam收敛最快,优于现有的Adam型算法.

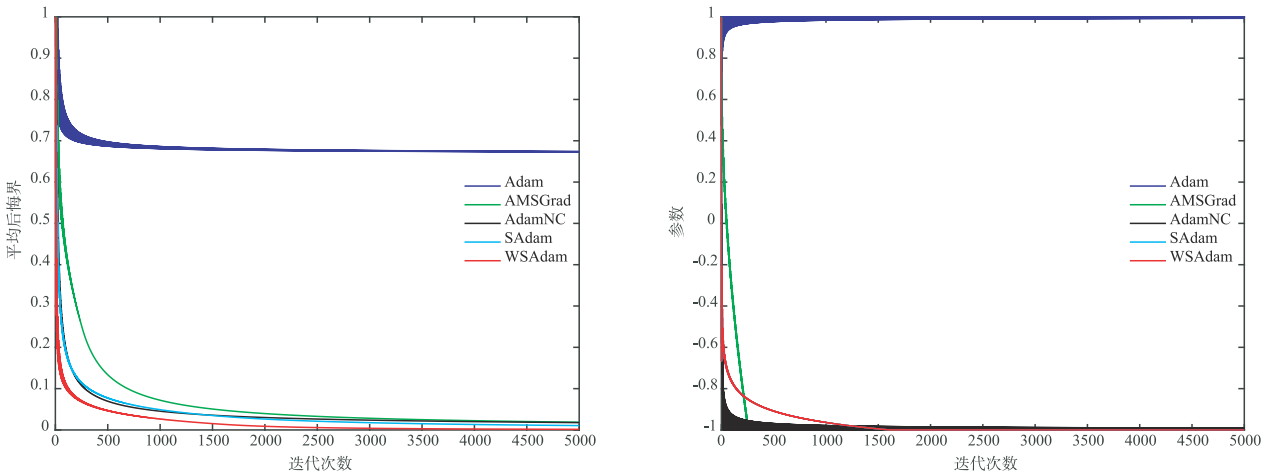
5.2　非光滑强凸情形标准数据集的实验结果与分析

本文第二个实验继承文献[21]中随机设置环境,考虑典型的二分类强凸支持向量机(SVM)问题,假设全体样本集$\xi=\left\{\left(\boldsymbol{x}_i,y_i\right)\right\}_{i=1}^{n}$,$\boldsymbol{x}_i\in\mathbf{R}^d$,$y_i\in\{+1,-1\}$,目标函数$f\left(\boldsymbol{w},\xi\right)$由$l_2$范数结构项和非光滑hinge损失组成,描述如下:

$$\min_{\boldsymbol{w}}\frac{\boldsymbol{\lambda}}{2}\|\boldsymbol{w}\|^2+\frac{1}{n}\sum_{(x,y)\in\xi}\ell\left(\boldsymbol{w},\left(\boldsymbol{x},y\right)\right)\tag{8}$$

其中$\ell\left(\boldsymbol{w},\left(\boldsymbol{x},y\right)\right)=\max\left\{0,1-y\langle\boldsymbol{w},\boldsymbol{x}\rangle\right\}$.

第t次迭代时,抽取样本子集ξ_t参与计算的次梯度$\nabla f\left(\boldsymbol{w}_t,\xi_t\right)$可以写成如下形式:



(*a*) 平均后悔界随迭代变化曲线 (*b*) w 参数随迭代变化曲线

图1 Reddi问题实验结果

$$\nabla f\left(\boldsymbol{w}_t,\xi_t\right)=\lambda\boldsymbol{w}_t-\frac{1}{k}\sum_{(\boldsymbol{x},y)\in\xi_t^+}y\boldsymbol{x} \tag{9}$$

其中 $k=\left|\xi_t\right|$, $\xi_t^+=\left\{\left(\boldsymbol{x},y\right)\in\xi_t: y\left\langle\boldsymbol{w}_t,\boldsymbol{x}\right\rangle<1\right\}$, λ 为强凸系数 $\boldsymbol{\lambda}$ 中的最小值. 本小节所有算法均采用上述次梯度计算方式,实验中 k 取值为10,也就是每次迭代更新参数时只用10个样本计算次梯度,并设置 $\lambda=0.01$.

采用6个标准数据集,分别是cod-rna、ijcnn1、gisette、madelon、a9a和live-disorders. 这些数据集均来自于LIBSVM网站(https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/),具体描述可见表2.

表2 标准数据库描述

Datasets	Training Samples	Test Samples	Dimensions
cod-rna	59535	271617	8
ijcnn1	49990	91701	22
gisette	6000	1000	5000
madelon	2000	600	500
a9a	32561	16281	123
liver-disorders	145	200	5

实验选择了著名的解SVM问题的pegasos[21]算法,以及几种典型的强凸Adam型算法作为比较对象. 为公平起见,所有算法设置参数 $\alpha=1$. 另外,pegasos根据文献[21]所述无其他预设参数;SAdam根据文献[20]设置 $\beta_1=0.9$, $\beta_{2,t}=1-0.9/t$, $\delta=1\mathrm{e}-2$;WSAdam根据算法4设置 $\beta_1=0.9$, $v=0.999$, $\beta_{2,t}=1-0.9/t$, $\delta=1\mathrm{e}-8$;SC-Adagrad根据文献[17]设置 $\varepsilon_1=0.1$, $\varepsilon_2=1$;SC-RMSProp根据文献[17]设置 $\beta_{2,t}=1-0.9/t$, $\varepsilon_1=0.1$, $\varepsilon_2=1$.

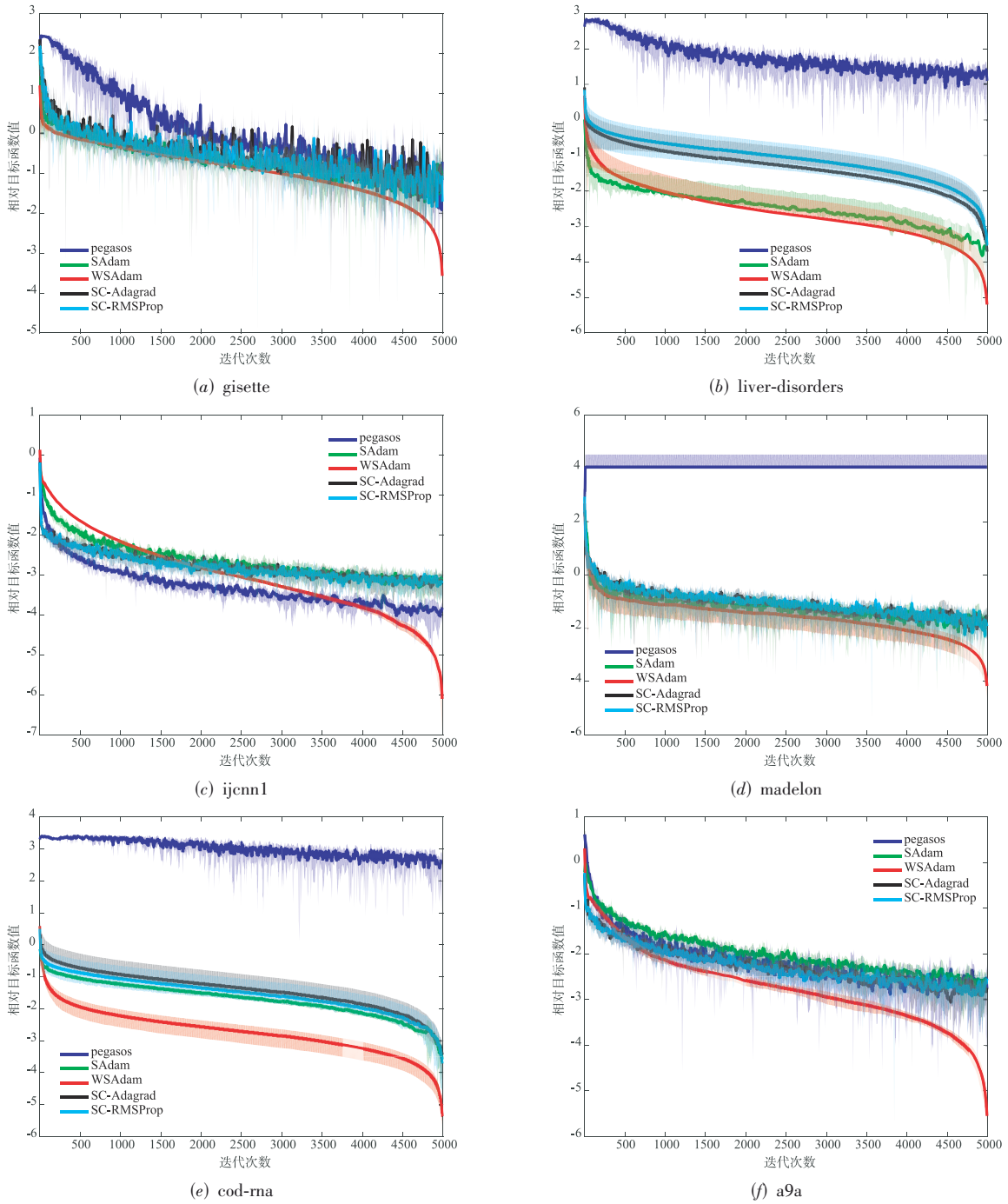
所有算法在每个数据集上运行10次并取平均值绘制收敛曲线errorbar对比图. 如图2所示,横坐标表示迭代次数 $t=[1,5000]$,纵坐标为相对目标函数值,即当前迭代目标函数值与目标函数最优值(最优值取所有迭代结果中的最小值)之差的对数值,4种比较算法的相对目标函数值形式为 $\log\left(f\left(\bar{\boldsymbol{w}}_t\right)-f\left(\boldsymbol{w}^*\right)\right)$,WSAdam的相对目标函数值形式为 $\log\left(f\left(\bar{\boldsymbol{w}}_t^w\right)-f\left(\boldsymbol{w}^*\right)\right)$. 蓝色实线代表pegasos算法的收敛趋势;绿色实线代表SAdam算法的收敛趋;红色实现代表WSAdam算法的收敛趋;黑色实线代表SC-Adagrad算法的收敛趋;青绿色实现代表SC-RMSProp算法的收敛趋.

从图2可以看出,没有使用自适应步长和动量技巧的pegasos十次平均曲线波动较大、方差大也大,收敛速率平缓,总体性能要差于其他4种Adam型算法. 而本文所提出的WSAdam十次平均曲线非常平滑(这是更改为加权平均输出所导致的),方差也较小. 在6个标准数据集上,WSAdam与现有流行的强凸Adam型算法均表现出基本相同的收敛趋势,甚至在一些训练集上(cod-rna)性能远超现有算法. 并且在同一精度要求下,WSAdam的收敛速度总体上是最快的. 这与理论分析中,WSAdam能达到优于其他算法的 $O(1/T)$ 收敛速率结果相吻合.

表3和表4分别给出了算法在6个数据集上训练所得模型的训练准确率(以及十次结果方差)、测试准确率(以及十次结果方差). 容易看出:WSAdam在所有训练数据集上的准确率均为最高,方差较其他算法处于较低的层次. WSAdam在绝大部分测试数据集上准确率最高(在cod-rna上SAdam算法准确率最高)). 一定程度上说明了WSAdam比其他几种算法训练的模型泛化性能更好,并且在训练和测试集上都保持较小实验方差,反映出其出色的稳定性.

表 3　训练准确率和方差比较

数据库/算法	Pegasos	SAdam	WSAdam	SC-Adagrad	SC-RMSProp
cod-rna	0.5994±0.1737	0.8575±0.0165	**0.8846±0.0011**	0.8099±0.1468	0.8260±0.0881
ijcnn1	0.9024±0.0000	0.9024±0.0000	**0.9024±0.0000**	0.9024±0.0000	0.9024±0.0000
gisette	0.9795±0.0367	0.9792±0.0148	**0.9861±0.0118**	0.9800±0.0236	0.9850±0.0071
madelon	0.5000±0.0000	0.5681±0.0507	**0.5760±0.0569**	0.5545±0.0573	0.5340±0.0436
a9a	0.8425±0.0016	0.8423±0.0008	**0.8428±0.0005**	0.8420±0.0006	0.8423±0.0009
liver-disorders	0.6228±0.1216	0.6897±0.0065	**0.6979±0.0121**	0.6648±0.0732	0.6531±0.0758



(*a*) gisette　(*b*) liver-disorders

(*c*) ijcnn1　(*d*) madelon

(*e*) cod-rna　(*f*) a9a

图 2　目标函数收敛速率比较图

表4 测试准确率和方差比较

数据库/算法	Pegasos	SAdam	WSAdam	SC-Adagrad	SC-RMSProp
cod-rna	0.6374±0.3886	**0.9495±0.0017**	0.9460±0.0015	0.9142±0.0965	0.9387±0.0178
ijcnn1	0.9050±0.0000	0.9050±0.0000	**0.9050±0.0000**	0.9050±0.0000	0.9050±0.0000
gisette	0.9626±0.0320	0.9568±0.0152	**0.9642±0.0125**	0.9578±0.0238	0.9617±0.0097
madelon	0.5000±0.0000	0.5612±0.0459	**0.5620±0.0569**	0.5450±0.0657	0.5220±0.0395
a9a	0.8462±0.0023	0.8454±0.0009	**0.8460±0.0013**	0.8452±0.0013	0.8458±0.0016
liver-disorders	0.5335±0.0231	0.5360±0.0084	**0.5410±0.0097**	0.5280±0.0289	0.5190±0.0281

6 结论

本文提出了一种名为WSAdam的Adam型算法,证明了在非光滑强凸情形,WSAdam能达到$O(1/T)$的最优收敛速率,体现了动量方法的加速性.据我们所知这是第一个被证明具有最优收敛速率的自适应步长策略与动量方法结合的算法.与SAdam算法相比,WSAdam改用了加权平均的输出方式,使算法在保持on-the-fly计算特点的同时,直接去掉了理论收敛速率上的对数阶因子.实验验证了所提算法成功避免Reddi提出的不收敛问题,并在解决非光滑强凸优化问题时比现有算法性能更优.

另一方面,自适应步长算法利用对角矩阵中记录的历史数据几何知识,缓和了对超参数的依赖性,因此非常适合训练深度神经网络.将WSAdam与动量方法[22]结合,探索其瞬时收敛速率[23]并推广到深度学习[24,25]中,将是我们下一步研究的方向.

参考文献

[1] CUTKOSKY A. Parameter-free, dynamic, and strongly-adaptive online learning[C]//Proceedings of the 37th International Conference on Machine Learning. Virtual Event: ICML, 2020: 2250-2259.

[2] ZINKEVICH M. Online convex programming and generalized infinitesimal gradient ascent[C]//Proceedings of the Twentieth International Conference on International Conference on Machine Learning. Washington DC: ICML, 2003: 928-935.

[3] HAZAN E, KALAI A, KALE S, et al. Logarithmic regret algorithms for online convex optimization[C]//Proceedings of 19th Annual Conference on Learning Theory(COLT). Berlin, Heidelberg: Springer, 2006: 499-513.

[4] SHAMIR O, ZHANG T. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes[C]//Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28. Atlanta: ICML, 2013: I-71-I-79.

[5] AGARWAL A, BARTLETT P L, RAVIKUMAR P, et al. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization[J]. IEEE Transactions on Information Theory, 2012, 58(5): 3235-3249.

[6] 陶卿, 朱烨雷, 罗强, 等. 一种基于Comid的非光滑损失随机坐标下降方法[J]. 电子学报, 2013, 41(4): 768-775.
TAO Q, ZHU Y L, LUO Q, et al. A new comid-based stochastic coordinate descent method for non-smooth losses[J]. Acta Electronica Sinica, 2013, 41(4): 768-775. (in Chinese)

[7] HAZAN E, KALE S. Beyond the regret minimization barrier: Optimal algorithms for stochastic strongly-convex optimization[J]. Journal of Machine Learning Research, 2011: 421-436.

[8] RAKHLIN A, SHAMIR O, SRIDHARAN K. Making gradient descent optimal for strongly convex stochastic optimization[C]//Proceedings of the 29th International Conference on Machine Learning(ICML). Edinburgh: ICML, 2012: 1571-1578.

[9] LACOSTE-JULIEN S, SCHMIDT M, BATCH F. A Simpler Approach to Obtaining an convergence rate for projected stochastic subgradient descent[EB/OL]. (2012-12-10). http://arxiv.org/abs/1212.2002.

[10] 邵言剑, 陶卿, 姜纪远, 等. 一种求解强凸优化问题的最优随机算法[J]. 软件学报, 2014, 25(9): 2160-2171.
SHAO Y J, TAO Q, JIANG J Y, et al. Stochastic algorithm with optimal convergence rate for strongly convex optimization problems[J]. Journal of Software, 2014, 25 (9): 2160-2171. (in Chinese)

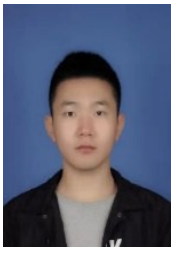
[11] CUTKOSKY A. Anytime online-to-batch, optimism and acceleration[C]//Proceedings of the 36th International Conference on Machine Learning(ICML). Long Beach: ICML, 2019: 1446-1454.

[12] KINGMA D P, BA J. Adam: A method for stochastic optimization[C]//Proceedings of the 3th International Conference on Learning Representations(ICLR). San Diego: ICLR, 2015: 1-13.

[13] TIMOTHY D. Incorporating Nesterov momentum into Adam[EB/OL]. (2015-12-12). http://cs229. stanford. edu/

proj2015/054_report.pdf.
[14] CHEN J H, ZHOU D R, TANG Y Q, et al. Closing the generalization gap of adaptive gradient methods in training deep neural networks[C]//Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. California: International Joint Conferences on Artificial Intelligence Organization, 2020: 3267-3275.
[15] TAO Wei, LONG Sheng, WU Gao-wei, et al. The role of momentum parameters in the optimal convergence of adaptive Polyak's heavy-ball methods[EB/OL].(2021-02-15). http://arxiv.org/abs/2102.07314.
[16] REDDI S J, KALE S, KUMAR S. On the convergence of Adam and Beyond[EB/OL].(2019-04-19). http://arxiv.org/abs/1904.09237.
[17] MUKKAMALA M C, HEIN M. Variants of RMSProp and Adagrad with logarithmic regret bounds[C]//Proceedings of the 34th International Conference on Machine Learning. Sydney: ICML, 2017: 2545-2553.
[18] DUCHI J, HAZAN E, SINGER Y. Adaptive subgradient methods for online learning and stochastic optimization [J]. Journal of Machine Learning Research, 2011, 12(7): 257-269.
[19] CHEN Zai-yi, XU Yi, CHEN En-hong, et al. SADAGRAD: Strongly adaptive stochastic gradient methods[C]// Proceedings of the 35th International Conference on Machine Learning(ICML). Stockholm: ICML, 2018: 912-920.
[20] WANG Guang-hui, LU Shi-yin, et al. SAdam: A variant of Adam for strongly convex functions[EB/OL].(2019-05-08). http://arxiv.org/abs/1905.02957.
[21] SHALEV-SHWARTZ S, SINGER Y, SREBRO N, et al. Pegasos: primal estimated sub-gradient solver for SVM [J]. Mathematical Programming, 2011, 127(1): 3-30.
[22] 沈卉卉, 李宏伟. 基于动量方法的受限玻尔兹曼机的一种有效算法[J]. 电子学报, 2019, 47(1): 176-182.
SHEN H H, LI H W. An effective algorithm of restricted boltzmann machine based on momentum method[J]. Acta Electronica Sinica, 2019, 47(1): 176-182. (in Chinese)
[23] 姜纪远, 陶卿, 邵言剑, 等. 随机COMID的瞬时收敛速率分析[J]. 电子学报, 2015, 43(9): 1850-1858.
JIANG J Y, TAO Q, SHAO Y J, et al. The analysis of convergence rate of individual COMID iterates[J]. Acta Electronica Sinica, 2015, 43(9): 1850-1858. (in Chinese)
[24] 邹军华, 段晔鑫, 任传伦, 等. 基于噪声初始化、Adam-Nesterov方法和准双曲动量方法的对抗样本生成方法[J]. 电子学报, 2022, 50(1): 207-216.
ZOU J H, DUAN Y X, REN C L, et al. Perturbation initialization, Adam-Nesterov and quasi-hyperbolic momentum for adversarial examples[J]. Acta Electronica Sinica, 2022, 50(1): 207-216. (in Chinese)
[25] 罗会兰, 袁璞, 童康. 基于深度学习的显著性目标检测方法综述[J]. 电子学报, 2021, 49(7): 1417-1427.
LUO H L, YUAN P, TONG K. Review of the methods for salient object detection based on deep learning[J]. Acta Electronica Sinica, 2021, 49(7): 1417-1427. (in Chinese)

作者简介



陇　盛　男,1998年1月出生于贵州省盘州市. 硕士研究生. 主要研究领域为机器学习,模式识别.
E-mail: ls15186322349@163.com

陶　蔚(通讯作者)　男,1991年出生于安徽省合肥市. 博士,助理研究员,主要研究领域为机器学习.
E-mail: wtao_plaust@163.com

张泽东　男,1994年出生于山东省临清市. 硕士研究生,主要研究领域为机器学习,模式识别.
E-mail: l632783823@qq.com

陶　卿　男,1965年出生于安徽省合肥市. 博士, 教授,博士生导师,CCF高级会员,主要研究领域为机器学习,模式识别, 应用数学.
E-mail: qing.tao@ia.ac.cn