第 40 卷第 9 期
2023 年 9 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 40 No. 9
Sep. 2023

有向网络分布式优化的Barzilai-Borwein梯度跟踪方法

高 娟[1], 刘新为[2†]

(1. 河北工业大学 人工智能与数据科学学院, 天津 300401; 2. 河北工业大学 数学研究院, 天津 300401)

摘要: 本文研究有向网络上的分布式优化问题, 其全局目标函数是网络上所有光滑强凸局部目标函数的平均值. 受Barzilai-Borwein步长改善梯度方法表现的启发, 本文提出了一种分布式Barzilai-Borwein梯度跟踪方法. 与文献中使用固定步长的分布式梯度算法不同, 所提出的方法中每个智能体利用其局部梯度信息自动地计算其步长. 通过同时使用行随机和列随机权重矩阵, 该方法避免了由特征向量估计引起的计算和通信. 当目标函数是光滑和强凸函数时, 本文证明了该算法产生的迭代序列可以线性地收敛到最优解. 对分布式逻辑回归问题的仿真结果验证了所提出的算法比使用固定步长的分布式梯度算法表现更好.

关键词: 分布式优化; 多智能体系统; 有向图; Barzilai-Borwein方法; 优化算法; 收敛速度

引用格式: 高娟, 刘新为. 有向网络分布式优化的Barzilai-Borwein梯度跟踪方法. 控制理论与应用, 2023, 40(9): 1637 – 1645

DOI: 10.7641/CTA.2022.20125

Barzilai-Borwein gradient tracking method for distributed optimization over directed networks

GAO Juan[1], LIU Xin-wei[2†]

(1. School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China;
2. Institute of Mathematics, Hebei University of Technology, Tianjin 300401, China)

Abstract: This paper studies the distributed optimization problem over directed networks. The global objective function of this problem is the average of all smooth and strongly convex local objective functions on the networks. Motivated by the capability of Barzilai-Borwein step sizes in improving the performance of gradient methods, a distributed Barzilai-Borwein gradient tracking method is proposed. Different from the distributed gradient algorithms using fixed step sizes in the literature, the proposed method allows each agent to calculate its step size automatically using its local gradient information. By using row- and column-stochastic weights simultaneously, the method can avoid the computation and communication on eigenvector estimation. It is proved that the iterative sequence generated by the proposed method converges linearly to the optimal solution for smooth and strongly convex functions. Simulation results on the distributed logistic regression problem show that the proposed method performs better than some advanced distributed gradient algorithms with fixed step sizes.

Key words: distributed optimization; multi-agent systems; directed graphs; Barzilai-Borwein method; optimization algorithm; convergence rate

Citation: GAO Juan, LIU Xinwei. Barzilai-Borwein gradient tracking method for distributed optimization over directed networks. *Control Theory & Applications*, 2023, 40(9): 1637 – 1645

1 引言

随着大数据时代的到来和多智能体系统协调技术的发展, 分布式优化受到了学术界和工业界越来越多的关注, 现已被广泛应用于分布式机器学习[1]、编队控制[2]、多智能体目标搜索[3]、无线网络[4]和协调控制[5]等领域. 这些实际问题通常被建模为最小化网络上多个局部损失函数的平均值. 网络中智能体之间的通信通常可以抽象为一个图. 由于实际中许多网络的拓扑结构是复杂的且不是双向的, 如基于广播的通信协议[6], 所以基于有向图的分布式优化问题受到了众多学者的关注和青睐[7–8]. 本文主要关注基于有向图的分布式优化问题以及求解这类问题的优化算法.

收稿日期: 2022−02−21; 录用日期: 2022−08−09.
†通信作者. E-mail: mathlxw@hebut.edu.cn.
本文责任编委: 柯良军.
国家自然科学基金项目(12071108, 11671116, 91630202)资助.
Supported by the National Natural Science Foundation of China (12071108, 11671116, 91630202).

分布式梯度下降(distributed gradient descent, DGD)算法[9–10]已经成为求解多智能体网络优化问题最流行的方法之一. 当采用衰减步长时, DGD方法可以次线性收敛到最优解, 但是收敛速度较慢; 当使用固定步长时, DGD方法可以达到线性收敛速度, 但是只能收敛到最优解的邻域内. 近年来, 该算法得到了显著的改进并发展了许多变形, 可参考文献[11–18]. 例如, Shi等[11]通过利用两次连续的DGD迭代的差构造了精确的一阶算法(exact first-order algorithm, EXTRA). 当使用固定步长时, 该算法可以线性收敛到精确解. 然而, 该算法要求权重矩阵是对称双随机矩阵. 进一步, Xu 等[12]首次将基于动态平均一致性[19]的梯度跟踪技巧与DGD方法相结合提出了分布式梯度跟踪方法. 该算法中每个智能体使用不一致固定步长, 并且权重矩阵可以是非对称双随机矩阵. 当目标函数是光滑和强凸函数时, Nedić等[13]和Qu等[14]证明了该算法线性地收敛到最优解. 另外, Berahas等[15]结合了多次一致步技巧和DGD算法, 提出了改进的DGD方法. 当多次一致步数随着迭代次数增加时, 该方法在采用固定步长的情况下可以精确线性地收敛到最优解. 最近, Li和Lin[16]进一步研究了EXTRA方法并对其进行了复杂性分析. 同时, Li和Lin[16]也将Catalyst加速技巧应用到EXTRA中提出了加速的EXTRA算法, 并分析了该算法的通信和计算复杂性. Qu和Li[17]提出了加速的分布式Nesterov梯度跟踪算法, 并且分析了该算法的线性收敛性质. Li等[18]提出了两种基于惩罚方法的分布式加速梯度算法.

上述文献中的算法仅适用于基于无向图的分布式优化问题, 这些算法要求构造双随机权重矩阵. 然而对于基于有向图的分布式优化问题, 网络拓扑结构可能是非平衡的, 因此构造双随机权重矩阵是件不切实际的事情. 于是, Nedić和Olshevsky[20–21]将Push-sum技巧[22]与DGD方法结合提出了求解有向网络优化问题的分布式梯度推力(gradient-push, GP)算法. 该方法采用了列随机权重矩阵并且利用Push-sum方法学习该矩阵的特征向量来消除有向图的非平衡性. 然而在GP方法中, 由局部梯度引起的误差, 学习特征向量所需要的迭代以及衰减步长都可能恶化算法的性能表现.

为了消除由局部梯度引起的误差, 许多学者提出了改进的方法, 如参考文献[23–28]. 结合EXTRA和GP算法, Zeng等[23]和Xi等[24]提出了快速的分布式算法, 并证明了其线性收敛性质. 然而该分布式算法的步长范围比较严格, 即步长的最大下界严格大于0. 文献[25–28]提出了基于梯度跟踪的方法. 对于光滑和强凸函数来说, 与GP算法的次线性收敛不同, 这些方法[25–28]被证明可以线性地收敛到最优解. 例如, Xi等[25]提出了加速的基于有向图的分布式优化方法(accelerated distributed directed optimization, ADD-OPT). ADD-OPT算法结合了列随机权重矩阵, Push-sum技巧和梯度跟踪策略, 并放宽了对步长的限制. 另一些算法(参见文献[27–28])则使用了行随机权重矩阵, Push-sum技巧和梯度跟踪策略. 其中Xin等[28]提出基于不一致固定步长的快速行随机优化算法(fast row-stochastic optimization with uncoordinated step sizes, FROST). 上述所有的基于有向图的方法都需要专门的迭代来学习行或列随机矩阵的特征向量, 这导致此类算法需要额外的计算和通信. 最近, 通过同时地使用行随机和列随机权重矩阵, Xin和Khan[29]提出了$\mathcal{AB}$算法(文献[30]也研究了该算法). 该算法的一个显著特点是它避免了由特征向量估计引起的计算与通信. 当目标函数是光滑和强凸函数时, Xin和Khan[29]证明了$\mathcal{AB}$算法的线性收敛性质. 进一步, Xin和Khan[31]将$\mathcal{AB}$和Heavy-ball动量加速技巧相结合, 提出了分布式Heavy-ball动量加速算法, 并证明了该算法对于光滑强凸函数可达到线性收敛速度. Gao等[32]提出了带有参数的分布式动量方法, 并证明了其线性收敛性质. 该方法是分布式Heavy-ball动量加速算法的一种推广. Li等[33]提出了Nesterov和Heavy-ball双动量加速的分布式异步优化算法, 并分析了其收敛性质.

众所周知, 梯度类算法中步长的选取对算法的表现起着重要作用. 上面提到的算法[9–18, 20–21, 23–33]涉及两类步长: 一类是一致或者不一致固定步长, 在实验中它们被手动调试使得算法达到最佳表现; 另一类是衰减步长, 当迭代点靠近最优解时, 它使得算法收敛速度很慢. 这些分布式梯度类方法[9–18, 20–21, 23–33]没有考虑如何充分利用局部信息为每个智能体选择合适的步长, 以此改善算法的数值表现. 最近, Gao等[34]将Barzilai-Borwein(BB)方法[35]引入基于无向图的分布式优化中, 提出了带有BB步长的分布式梯度方法并证明了该方法具有几何收敛速度. 进一步, 结合行随机权重矩阵和Push-sum技巧, Hu等[36]将该方法推广到有向图上, 提出了ADBB(accelerated distributed BB)算法. 此外, 这两篇文献中提出的分布式BB方法都需要结合多次一致步内循环技术, 这大大地增加了智能体之间的通信成本.

受以自适应截断循环方式生成的BB步长[37]改善梯度法表现的启发, 本文给出该BB步长的分布式形式, 然后将其引入到$\mathcal{AB}$算法中, 从而提出了一种求解基于强连通有向图的优化问题的分布式 Barzilai-Borwein梯度跟踪方法, 简称$\mathcal{AB}$-BB. 该方法中每个智能体利用其局部梯度信息自动地计算步长. 与现有的分布式BB方法[34, 36]相比, $\mathcal{AB}$-BB方法不需要使用多次一致步内循环技巧. 所以, 该方法在保证不增加每次迭代的计算和通信成本的基础上通过采用分布式的BB步长策略提高了$\mathcal{AB}$算法的性能. 与只采用行

随机或列随机权重矩阵的分布式梯度算法[25–28,36]相比, $\mathcal{AB}$-BB算法避免了估计特征向量, 从而降低了计算和通信成本. 本文在与$\mathcal{AB}$算法相同的假设下, 证明了$\mathcal{AB}$-BB算法可以线性地收敛到最优解. 最后, 本文对所提出的$\mathcal{AB}$-BB算法进行了数值仿真. 仿真结果表明了本文所提出的算法的有效性.

本文其余部分结构安排为: 第2节描述了所求解的优化问题; 第3节提出了新算法; 第4节给出了新算法的收敛性结果; 第5节为仿真实验; 第6节为本文结论.

记智能体i在第k次迭代时的变量为$x_k^i \in \mathbb{R}^p$. 用$\mathbf{1}_n$代表元素全为1的n维列向量, I_n表示$n \times n$阶的单位矩阵, $X \otimes Y$定义为矩阵X和Y的克罗内克积, $\rho(X)$表示方阵X的谱半径, $\mathrm{diag}\{x\}$代表以向量x的元素生成的对角矩阵. 给定矩阵$S \in \mathbb{R}^{n\times n}$, S_∞表示为它的无限次幂(如果它存在), 即$S_\infty = \lim\limits_{k\to\infty} S^k$. 对于一个本原行随机矩阵$A \in \mathbb{R}^{n\times n}$, 它的特征值为1的左特征向量和右特征向量分别记为π_r和$\mathbf{1}_n$, 且满足$\pi_r^{\mathrm{T}}\mathbf{1}_n = 1$. 相应地, 对于一个本原列随机矩阵$B \in \mathbb{R}^{n\times n}$, 它的特征值为1的左特征向量和右特征向量分别记为$\mathbf{1}_n$和π_c, 且满足$\mathbf{1}_n^{\mathrm{T}}\pi_c = 1$. 由Perron-Frobenius定理[38]有$A_\infty = \mathbf{1}_n\pi_r^{\mathrm{T}}$和$B_\infty = \pi_c\mathbf{1}_n^{\mathrm{T}}$. 符号$\|\cdot\|$表示为向量或矩阵的欧氏范数.

2　问题描述

考虑由n个智能体组成的有向网络, 网络中智能体之间的信息通信通过有向图$\mathcal{G} = (\mathcal{V}, \mathcal{E})$来刻画. 其中$\mathcal{V} = \{1, 2, \cdots, n\}$是所有智能体(节点)构成的集合, $\mathcal{E}$是有向边集且代表智能体之间的通信链路. 记$\mathcal{N}_i^{\mathrm{in}} = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\}$为智能体$i$的入邻居集合, 即该集合中的每个智能体可以将信息发送给智能体i. 相应地, 记$\mathcal{N}_i^{\mathrm{out}} = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$为智能体$i$的出邻居集合, 即该集合中的每个智能体可以接收智能体i发送的信息. 注意, 在这里$\mathcal{N}_i^{\mathrm{in}}$和$\mathcal{N}_i^{\mathrm{out}}$包含智能体$i$本身. 所有智能体协同地求解下列分布式优化问题:

$$\min_{x\in\mathbb{R}^p} f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1}$$

其中:$x \in \mathbb{R}^p$是全局决策变量; 局部损失函数$f_i : \mathbb{R}^p \to \mathbb{R}$是连续可微的, 且仅被智能体$i$所知. 本文对目标函数和通信图作出以下标准假设.

假设 1　对于每个$i \in \mathcal{V}$, 它的目标函数f_i是L_i光滑的, 即对$\forall x, y \in \mathbb{R}^p$, 存在常数$L_i > 0$, 使得

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leqslant L_i\|x - y\|.$$

假设 2　对于每个$i \in \mathcal{V}$, 它的目标函数f_i是μ_i强凸的, 即对$\forall x, y \in \mathbb{R}^p$, 存在常数$\mu_i > 0$, 使得

$$f_i(x) \geqslant f_i(y) + \nabla f_i(y)^{\mathrm{T}}(x - y) + \frac{\mu_i}{2}\|x - y\|^2.$$

假设 3　通信图$\mathcal{G}$是强连通的.

假设2表明问题(1)存在唯一最优解$x^* \in \mathbb{R}^p$. 在之后的分析中, 本文记$L_f = \frac{1}{n}\sum_{i=1}^{n} L_i$, $\mu_f = \frac{1}{n}\sum_{i=1}^{n}\mu_i$, $L = \max\limits_i\{L_i\}$和$\mu = \min\limits_i\{\mu_i\}$.

3　算法

本节首先介绍了分布式的Barzilai-Borwein步长, 然后将其引入到$\mathcal{AB}$算法中并提出了$\mathcal{AB}$-BB算法.

3.1　分布式的Barzilai-Borwein步长

本小节首先回顾原始的Barzilai-Borwein方法. 求解问题(1)的原始的BB方法[39]的迭代格式为

$$x_{k+1} = x_k - \alpha_k\nabla f(x_k),$$

其中步长α_k由$\alpha_k^{\mathrm{BB1}} = \dfrac{s_k^{\mathrm{T}}s_k}{s_k^{\mathrm{T}}y_k}$或者$\alpha_k^{\mathrm{BB2}} = \dfrac{s_k^{\mathrm{T}}y_k}{y_k^{\mathrm{T}}y_k}$计算所得, $s_k = x_k - x_{k-1}, y_k = \nabla f(x_k) - \nabla f(x_{k-1})$. 如果$s_k^{\mathrm{T}}y_k > 0$, 则有$\alpha_k^{\mathrm{BB1}} \geqslant \alpha_k^{\mathrm{BB2}}$, 即$\alpha_k^{\mathrm{BB1}}$是一个长BB步长而$\alpha_k^{\mathrm{BB2}}$是一个短BB步长.

文献[37,40–41]中的数值仿真表明, BB方法以交替或自适应的方式使用长BB步长和一些短BB步长, 比原始的BB方法[39]的性能要好得多. 所以, 本文考虑以自适应截断循环方式生成的BB步长[37]. 显然, 直接将该BB步长应用到分布式优化中是一件不切实际的事情. 因为它需要计算全局梯度$\nabla f(x_k)$. 因此, 本文给出该BB步长的分布式形式.

对于每个智能体i, 定义

$$(\alpha_k^i)^{\mathrm{BB1}} = \frac{1}{c}\cdot\frac{(s_k^i)^{\mathrm{T}}s_k^i}{(s_k^i)^{\mathrm{T}}y_k^i} \tag{2}$$

和

$$(\alpha_k^i)^{\mathrm{BB2}} = \frac{1}{c}\cdot\frac{(s_k^i)^{\mathrm{T}}y_k^i}{(y_k^i)^{\mathrm{T}}y_k^i}, \tag{3}$$

其中: $c > 0$是保障参数, $s_k^i = x_k^i - x_{k-1}^i$和$y_k^i = \nabla f_i(x_k^i) - \nabla f_i(x_{k-1}^i), k \geqslant 1$. 由假设2, 有$(s_k^i)^{\mathrm{T}}y_k^i > 0$. 因此, 很容易地得到$(\alpha_k^i)^{\mathrm{BB1}} \geqslant (\alpha_k^i)^{\mathrm{BB2}}$. 这表明$(\alpha_k^i)^{\mathrm{BB1}}$是一个长BB步长而$(\alpha_k^i)^{BB2}$是一个短BB步长. 当$c = 1$时, 式(2)和式(3)退化为原始的BB步长的分布式形式. 所以参数$c = 1$表明保障参数不起作用.

本文给出的分布式BB步长有下列形式:

$$\alpha_k^i = \begin{cases} (\alpha_k^i)^{\mathrm{BB1}}, & \mathrm{mod}(k, h) = 0, \\ \widetilde{\alpha}_k^i, & \text{其他}, \end{cases} \tag{4}$$

其中$h > 0$是间隔参数, $k \geqslant 1$是迭代次数,

$$\widetilde{\alpha}_k^i = \begin{cases} (\alpha_k^i)^{\mathrm{BB2}}, & \alpha_{k-1}^i \leqslant (\alpha_k^i)^{\mathrm{BB2}}, \\ (\alpha_k^i)^{\mathrm{BB1}}, & \alpha_{k-1}^i \geqslant (\alpha_k^i)^{\mathrm{BB1}}, \\ \alpha_{k-1}^i, & \text{其他}. \end{cases} \tag{5}$$

由式(5)可知, $\widetilde{\alpha}_k^i$由前一次迭代的步长信息自适应地得到. 具体来说, 当上一次迭代的步长太小时, 步长应该增大; 当上一次迭代的步长太大时, 步长应该减小;

当上一次迭代的步长属于区间$[(\alpha_k^i)^{\mathrm{BB2}},(\alpha_k^i)^{\mathrm{BB1}}]$时，算法继续使用该步长. 从式(4)可得，步长$\alpha_k^i$在每间隔$h$次迭代时使用一次长BB步长$(\alpha_k^i)^{\mathrm{BB1}}$，这避免了算法在多次迭代时采用同一种步长. 间隔参数h的最优取值就是使得算法表现最佳的值. 显然，由式(4)计算所得的BB步长属于区间$[(\alpha_k^i)^{\mathrm{BB2}},(\alpha_k^i)^{\mathrm{BB1}}]$. 对于分布式BB步长的初始步长$\alpha_0^i, i\in\mathcal{V}$，本文由用户确定它的取值. 从仿真实验可以观察到，本文所提出的算法的性能对初始步长的选取并不敏感.

3.2 Barzilai-Borwein梯度跟踪方法

对于分布式优化问题(1)，Xin和Khan[29]提出了同时使用行随机和列随机权重矩阵的$\mathcal{AB}$算法. 与先前提出的只采用行随机或列随机权重矩阵的算法[25–28]相比，该算法降低了计算和通信成本. 本文将分布式的BB步长与$\mathcal{AB}$算法相结合，提出了一种分布式的Barzilai-Borwein梯度跟踪方法，简称$\mathcal{AB}$-BB. 在第k次迭代时，每个智能体i更新3个变量，即x_k^i, $z_k^i\in\mathbb{R}^p$和$\alpha_k^i\in\mathbb{R}$. 其中x_k^i为每个智能体对全局极小点x^*的局部估计，z_k^i是每个智能体对全局梯度$\frac{1}{n}\sum\limits_{i=1}^n\nabla f_i(x_k^i)$的局部估计，$\alpha_k^i$代表每个智能体$i$的局部BB步长. 对于$i\in\mathcal{V}$，给定任意的初始迭代点$x_0^i\in\mathbb{R}^p$，令$z_0^i=\nabla f_i(x_0^i)$, $\alpha_0^i>0$. $\mathcal{AB}$-BB算法的迭代格式为

$$x_{k+1}^i=\sum_{j=1}^n a_{ij}(x_k^j-\alpha_k^j z_k^j),$$

$$z_{k+1}^i=\sum_{j=1}^n b_{ij}(z_k^j+\nabla f_j(x_{k+1}^j)-\nabla f_j(x_k^j)),$$

其中α_k^i由式(4)计算所得，a_{ij}和b_{ij}满足下列性质:

$$a_{ij}\begin{cases}>0,\ j\in\mathcal{N}_i^{\mathrm{in}},\\ =0,\ 其他,\end{cases}\quad \sum_{j=1}^n a_{ij}=1,\forall i,$$

$$b_{ij}\begin{cases}>0,\ i\in\mathcal{N}_j^{\mathrm{out}},\\ =0,\ 其他,\end{cases}\quad \sum_{i=1}^n b_{ij}=1,\forall j,$$

其中: $\mathcal{N}_i^{\mathrm{in}}=\{j\in\mathcal{V}|(j,i)\in\mathcal{E}\}$和$\mathcal{N}_j^{\mathrm{out}}=\{i\in\mathcal{V}|(j,i)\in\mathcal{E}\}$. 显然，权重矩阵$A=[a_{ij}]\in\mathbb{R}^{n\times n}$是行随机的和$B=[b_{ij}]\in\mathbb{R}^{n\times n}$是列随机的.

当$\alpha_k^i=\alpha(\alpha>0)$时，$\mathcal{AB}$-BB算法退化为$\mathcal{AB}$[29].

为方便起见，令$\boldsymbol{x}_k$, $\boldsymbol{z}_k$和$\nabla\boldsymbol{f}(\boldsymbol{x}_k)\in\mathbb{R}^{np}$分别是由$x_k^i$, z_k^i和$\nabla f_i(x_k^i)$集成的向量. 那么，$\mathcal{AB}$-BB算法可以等价地写为

$$\boldsymbol{x}_{k+1}=\mathcal{A}(\boldsymbol{x}_k-D_k\boldsymbol{z}_k),\tag{6a}$$

$$\boldsymbol{z}_{k+1}=\mathcal{B}(\boldsymbol{z}_k+\nabla\boldsymbol{f}(\boldsymbol{x}_{k+1})-\nabla\boldsymbol{f}(\boldsymbol{x}_k)),\tag{6b}$$

其中: $\mathcal{A}=A\otimes I_p$, $\mathcal{B}=B\otimes I_p$和$D_k=\mathrm{diag}\{\alpha_k\}\otimes I_p$, $\alpha_k=[\alpha_k^1\ \cdots\ \alpha_k^n]^{\mathrm{T}}$.

4 收敛性分析

本节证明了$\mathcal{AB}$-BB算法具有R-线性收敛速度.

以下引理给出了分布式BB步长的取值范围.

引理 1 设假设1–2成立. 对所有的$k\geqslant 1$和$i\in\mathcal{V}$, $\mathcal{AB}$-BB算法中的BB步长α_k^i满足

$$\frac{1}{cL_i}\leqslant\alpha_k^i\leqslant\frac{1}{c\mu_i}.\tag{7}$$

证 回顾由式(4)定义的BB步长α_k^i，可知该步长α_k^i位于区间$[(\alpha_k^i)^{BB2},(\alpha_k^i)^{\mathrm{BB1}}]$上. 为了证明，本文分别考虑$(\alpha_k^i)^{\mathrm{BB1}}$的上界和$(\alpha_k^i)^{\mathrm{BB2}}$的下界. 利用局部目标函数$f_i(x)$强凸性和局部梯度$\nabla f_i(x)$的$L_i$-Lipschitz连续性，本文很容易地得到$(\alpha_k^i)^{\mathrm{BB1}}\leqslant\dfrac{1}{c\mu_i}$和$(\alpha_k^i)^{\mathrm{BB2}}\geqslant\dfrac{1}{cL_i}$. 证毕.

定义最大步长$\bar\alpha=\max\limits_{k\geqslant1}\{\alpha_k^i\}$. 由引理1得

$$\frac{1}{cL}\leqslant\bar\alpha\leqslant\frac{1}{c\mu}.\tag{8}$$

下面两个引理对收敛性证明有着重要作用，可参考文献[29].

引理 2 考虑增广的权重矩阵$\mathcal{A}$和$\mathcal{B}$. 对于$0<\sigma_{\mathcal{A}}<1$和$0<\sigma_{\mathcal{B}}<1$，存在向量范数$\|\cdot\|_{\mathcal{A}}$和$\|\cdot\|_{\mathcal{B}}$，对$\forall\boldsymbol{a}\in\mathbb{R}^{np}$，有

$$\|\mathcal{A}\boldsymbol{a}-\mathcal{A}_\infty\boldsymbol{a}\|_{\mathcal{A}}\leqslant\sigma_{\mathcal{A}}\|\boldsymbol{a}-\mathcal{A}_\infty\boldsymbol{a}\|_{\mathcal{A}},$$

$$\|\mathcal{B}\boldsymbol{a}-\mathcal{B}_\infty\boldsymbol{a}\|_{\mathcal{B}}\leqslant\sigma_{\mathcal{B}}\|\boldsymbol{a}-\mathcal{B}_\infty\boldsymbol{a}\|_{\mathcal{B}},$$

其中$\sigma_{\mathcal{A}}=|||\mathcal{A}-\mathcal{A}_\infty|||_{\mathcal{A}}$, $\sigma_{\mathcal{B}}=|||\mathcal{B}-\mathcal{B}_\infty|||_{\mathcal{B}}$，以及$|||\cdot|||_{\mathcal{A}}$和$|||\cdot|||_{\mathcal{B}}$分别是与上述两个向量范数相容的矩阵范数.

根据有限维线性空间中任意两种向量的等价性，存在正数r,l,t,u使得

$$\|\cdot\|\leqslant r\|\cdot\|_{\mathcal{A}},\ \ \|\cdot\|\leqslant t\|\cdot\|_{\mathcal{B}},$$

$$\|\cdot\|_{\mathcal{A}}\leqslant l\|\cdot\|,\ \ \|\cdot\|_{\mathcal{B}}\leqslant u\|\cdot\|.$$

引理 3 $(\mathbf{1}_n^{\mathrm{T}}\otimes I_p)\boldsymbol{z}_k=(\mathbf{1}_n^{\mathrm{T}}\otimes I_p)\nabla\boldsymbol{f}(\boldsymbol{x}_k)$, $\forall k\geqslant 0$.

类似于$\mathcal{AB}$算法的收敛性证明思路，本文也考虑下列3个量:

1) 一致误差，$\|\boldsymbol{x}_{k+1}-\mathcal{A}_\infty\boldsymbol{x}_{k+1}\|_{\mathcal{A}}$;

2) 最优间隙，$\|\mathcal{A}_\infty\boldsymbol{x}_{k+1}-\mathbf{1}_n\otimes x^*\|$;

3) 梯度跟踪误差，$\|\boldsymbol{z}_{k+1}-\mathcal{B}_\infty\boldsymbol{z}_{k+1}\|_{\mathcal{B}}$.

本文首先建立了关于上述3个量的一个线性不等式，然后结合该线性不等式和引理1的结果证明了$\mathcal{AB}$-BB算法具有R-线性收敛速度.

为了证明$\mathcal{AB}$-BB算法的线性收敛性质，本文首先需要给出跟踪梯度的上界.

引理 4[29] 对所有的$k\geqslant 0$，有

$$\|\boldsymbol{z}_k\|\leqslant rL\|\mathcal{B}_\infty|||\boldsymbol{x}_k-\mathcal{A}_\infty\boldsymbol{x}_k\|_{\mathcal{A}}+$$

$$L\|\mathcal{B}_\infty\|\|\mathcal{A}_\infty \boldsymbol{x}_k - \boldsymbol{1}_n \otimes x^*\| + t\|\boldsymbol{z}_k - \mathcal{B}_\infty \boldsymbol{z}_k\|_{\mathcal{B}}.$$

引理 5　设假设1–3成立, 若$0 < \bar{\alpha} \leqslant \dfrac{1}{nL_f \pi_r^{\mathrm{T}} \pi_c}$, 则有下列线性不等式成立:

$$v_{k+1} \leqslant G_c v_k,\ \forall k \geqslant 1, \tag{9}$$

其中$v_k \in \mathbb{R}^3$和$G_c \in \mathbb{R}^{3\times 3}$有下列定义:

$$v_k = \begin{bmatrix} \|\boldsymbol{x}_k - \mathcal{A}_\infty \boldsymbol{x}_k\|_{\mathcal{A}} \\ \|\mathcal{A}_\infty \boldsymbol{x}_k - \boldsymbol{1}_n \otimes x^*\| \\ \|\boldsymbol{z}_k - \mathcal{B}_\infty \boldsymbol{z}_k\|_{\mathcal{B}} \end{bmatrix},$$

$$G_c = \begin{bmatrix} \sigma_{\mathcal{A}} + r d_1 a_1 \bar{\alpha} & d_1 a_1 \bar{\alpha} & t a_1 \bar{\alpha} \\ r L a_2 \bar{\alpha} & 1 - \dfrac{\mu_f a_2}{cL} & t a_3 \bar{\alpha} \\ d_2(a_4 + r d_1 a_5 \bar{\alpha}) & d_1 d_2 a_5 \bar{\alpha} & \sigma_{\mathcal{B}} + d_2 a_5 t \bar{\alpha} \end{bmatrix},$$

上述矩阵中的常数$a_i, i = 1,2,3,4,5$和$d_i, i = 1,2$由下列式子计算得到: $a_1 = \sigma_{\mathcal{A}} l \|I_{np} - \mathcal{A}_\infty\|$, $a_2 = (\pi_r^{\mathrm{T}} \pi_c) n$, $a_3 = \|\mathcal{A}_\infty\|$, $a_4 = r\|\mathcal{A} - I_{np}\|$, $a_5 = \|\mathcal{A}\|$, $d_1 = L\|\mathcal{B}_\infty\|$, $d_2 = uL\sigma_{\mathcal{B}}\|I_{np} - \mathcal{B}_\infty\|$.

证　首先, 给出一致误差$\|\boldsymbol{x}_{k+1} - \mathcal{A}_\infty \boldsymbol{x}_{k+1}\|_{\mathcal{A}}$的上界. 注意到$\mathcal{A}_\infty \mathcal{A} = \mathcal{A}_\infty$. 由引理2和式(6a)可知

$$\begin{aligned} &\|\boldsymbol{x}_{k+1} - \mathcal{A}_\infty \boldsymbol{x}_{k+1}\|_{\mathcal{A}} = \\ &\|\mathcal{A}(\boldsymbol{x}_k - D_k \boldsymbol{z}_k) - \mathcal{A}_\infty \mathcal{A}(\boldsymbol{x}_k - D_k \boldsymbol{z}_k)\|_{\mathcal{A}} \leqslant \\ &\sigma_{\mathcal{A}} \|\boldsymbol{x}_k - \mathcal{A}_\infty \boldsymbol{x}_k\|_{\mathcal{A}} + \bar{\alpha} \sigma_{\mathcal{A}} l \|I_{np} - \mathcal{A}_\infty\| \|\boldsymbol{z}_k\|. \end{aligned}$$

结合引理4可得

$$\begin{aligned} &\|\boldsymbol{x}_{k+1} - \mathcal{A}_\infty \boldsymbol{x}_{k+1}\|_{\mathcal{A}} \leqslant \\ &(\sigma_{\mathcal{A}} + \bar{\alpha} \sigma_{\mathcal{A}} r l L \|\mathcal{B}_\infty\| \|I_{np} - \mathcal{A}_\infty\|) \|\boldsymbol{x}_k - \mathcal{A}_\infty \boldsymbol{x}_k\|_{\mathcal{A}} + \\ &\bar{\alpha} \sigma_{\mathcal{A}} l L \|\mathcal{B}_\infty\| \|I_{np} - \mathcal{A}_\infty\| \|\mathcal{A}_\infty \boldsymbol{x}_k - \boldsymbol{1}_n \otimes x^*\| + \\ &\bar{\alpha} \sigma_{\mathcal{A}} l t \|I_{np} - \mathcal{A}_\infty\| \|\boldsymbol{z}_k - \mathcal{B}_\infty \boldsymbol{z}_k\|_{\mathcal{B}}. \end{aligned} \tag{10}$$

其次, 给出最优间隙$\|\mathcal{A}_\infty \boldsymbol{x}_{k+1} - \boldsymbol{1}_n \otimes x^*\|$的上界. 由式(6a)有

$$\begin{aligned} &\|\mathcal{A}_\infty \boldsymbol{x}_{k+1} - \boldsymbol{1}_n \otimes x^*\| = \\ &\|\mathcal{A}_\infty(\mathcal{A}(\boldsymbol{x}_k - D_k \boldsymbol{z}_k) + (D_k - D_k)\mathcal{B}_\infty \boldsymbol{z}_k) - \\ &\boldsymbol{1}_n \otimes x^*\| \leqslant \\ &\|\mathcal{A}_\infty \boldsymbol{x}_k - \mathcal{A}_\infty D_k \mathcal{B}_\infty \boldsymbol{z}_k - \boldsymbol{1}_n \otimes x^*\| + \\ &\bar{\alpha} t \|\mathcal{A}_\infty\| \|\boldsymbol{z}_k - \mathcal{B}_\infty \boldsymbol{z}_k\|_{\mathcal{B}}. \end{aligned} \tag{11}$$

考虑式(11)的右边第1项, 令

$$\widetilde{x}_k = (\pi_r^{\mathrm{T}} \otimes I_p)\boldsymbol{x}_k,$$

$$\nabla \boldsymbol{f}((\boldsymbol{1}_n \otimes I_p)\widetilde{x}_k) = [\nabla f_1^{\mathrm{T}}(\widetilde{x}_k)\ \cdots\ \nabla f_n^{\mathrm{T}}(\widetilde{x}_k)]^{\mathrm{T}}.$$

再者,

$$\mathcal{A}_\infty D_k \mathcal{B}_\infty = (\pi_r^{\mathrm{T}} \mathrm{diag}\{\alpha_k\} \pi_c)(\boldsymbol{1}_n \otimes I_p)(\boldsymbol{1}_n^{\mathrm{T}} \otimes I_p),$$

故

$$\begin{aligned} &\|\mathcal{A}_\infty \boldsymbol{x}_k - \mathcal{A}_\infty D_k \mathcal{B}_\infty \boldsymbol{z}_k - \boldsymbol{1}_n \otimes x^*\| = \\ &\|\mathcal{A}_\infty \boldsymbol{x}_k - \mathcal{A}_\infty D_k \mathcal{B}_\infty \nabla \boldsymbol{f}(\boldsymbol{x}_k) - \boldsymbol{1}_n \otimes x^*\| = \\ &\|(\boldsymbol{1}_n \otimes I_p)(\widetilde{x}_k - (\pi_r^{\mathrm{T}} \mathrm{diag}\{\alpha_k\} \pi_c)(\boldsymbol{1}_n^{\mathrm{T}} \otimes I_p)\nabla \boldsymbol{f}(\boldsymbol{x}_k) - \\ &x^*)\| \leqslant \\ &\|(\boldsymbol{1}_n \otimes I_p)(\widetilde{x}_k - n(\pi_r^{\mathrm{T}} \mathrm{diag}\{\alpha_k\} \pi_c)\nabla f(\widetilde{x}_k) - x^*)\| + \\ &\pi_r^{\mathrm{T}} \mathrm{diag}\{\alpha_k\} \pi_c \|(\boldsymbol{1}_n \otimes I_p)(n\nabla f(\widetilde{x}_k) - \\ &(\boldsymbol{1}_n^{\mathrm{T}} \otimes I_p)\nabla \boldsymbol{f}(\boldsymbol{x}_k))\| \triangleq \theta_1 + \theta_2, \end{aligned} \tag{12}$$

其中第1个等式利用了引理3.

考虑θ_1. 利用文献[14]中的引理10, 可知如果$0 < n(\pi_r^{\mathrm{T}} \pi_c)\bar{\alpha} \leqslant \dfrac{1}{L_f}$, 则

$$\begin{aligned} \theta_1 = {} &\sqrt{n}\|\widetilde{x}_k - n(\pi_r^{\mathrm{T}} \mathrm{diag}\{\alpha_k\} \pi_c)\nabla f(\widetilde{x}_k) - x^*\| \leqslant \\ &\sqrt{n}(1 - \mu_f n(\pi_r^{\mathrm{T}} \mathrm{diag}\{\alpha_k\} \pi_c))\|\widetilde{x}_k - x^*\| \leqslant \\ &\sqrt{n}(1 - \frac{\mu_f n(\pi_r^{\mathrm{T}} \pi_c)}{cL})\|\widetilde{x}_k - x^*\| = \\ &(1 - \frac{\mu_f n(\pi_r^{\mathrm{T}} \pi_c)}{cL})\|\mathcal{A}_\infty \boldsymbol{x}_k - \boldsymbol{1}_n \otimes x^*\|, \end{aligned} \tag{13}$$

其中最后1个不等式利用了$\pi_r^{\mathrm{T}} \mathrm{diag}\{\alpha_k\} \pi_c \geqslant \dfrac{\pi_r^{\mathrm{T}} \pi_c}{cL}$.

考虑θ_2. 由$\nabla f(\widetilde{x}_k) = \dfrac{1}{n}(\boldsymbol{1}_n^{\mathrm{T}} \otimes I_p)\nabla \boldsymbol{f}((\boldsymbol{1}_n \otimes I_p)\widetilde{x}_k)$有

$$\begin{aligned} &\theta_2 \leqslant \\ &(\pi_r^{\mathrm{T}} \mathrm{diag}\{\alpha_k\} \pi_c) n \|\nabla \boldsymbol{f}((\boldsymbol{1}_n \otimes I_p)\widetilde{x}_k) - \nabla \boldsymbol{f}(\boldsymbol{x}_k)\| \leqslant \\ &(\pi_r^{\mathrm{T}} \mathrm{diag}\{\alpha_k\} \pi_c) n L r \|\boldsymbol{x}_k - \mathcal{A}_\infty \boldsymbol{x}_k\|_{\mathcal{A}} \leqslant \\ &\bar{\alpha}(\pi_r^{\mathrm{T}} \pi_c) n L r \|\boldsymbol{x}_k - \mathcal{A}_\infty \boldsymbol{x}_k\|_{\mathcal{A}}. \end{aligned} \tag{14}$$

结合式(11)–(14), 可得到

$$\begin{aligned} &\|\mathcal{A}_\infty \boldsymbol{x}_{k+1} - \boldsymbol{1}_n \otimes x^*\| \leqslant \\ &\bar{\alpha}(\pi_r^{\mathrm{T}} \pi_c) n L r \|\boldsymbol{x}_k - \mathcal{A}_\infty \boldsymbol{x}_k\|_{\mathcal{A}} + \\ &(1 - \frac{\mu_f n(\pi_r^{\mathrm{T}} \pi_c)}{cL})\|\mathcal{A}_\infty \boldsymbol{x}_k - \boldsymbol{1}_n \otimes x^*\| + \\ &\bar{\alpha} t \|\mathcal{A}_\infty\| \|\boldsymbol{z}_k - \mathcal{B}_\infty \boldsymbol{z}_k\|_{\mathcal{B}}. \end{aligned} \tag{15}$$

最后, 给出梯度跟踪误差$\|\boldsymbol{z}_{k+1} - \mathcal{B}_\infty \boldsymbol{z}_{k+1}\|_{\mathcal{B}}$的上界. 根据$\mathcal{B}_\infty \mathcal{B} = \mathcal{B}_\infty$, 引理2、假设1和式(6b), 得

$$\begin{aligned} &\|\boldsymbol{z}_{k+1} - \mathcal{B}_\infty \boldsymbol{z}_{k+1}\|_{\mathcal{B}} = \\ &\|\mathcal{B}(\boldsymbol{z}_k + \nabla \boldsymbol{f}(\boldsymbol{x}_{k+1}) - \nabla \boldsymbol{f}(\boldsymbol{x}_k)) - \\ &\mathcal{B}_\infty \mathcal{B}(\boldsymbol{z}_k + \nabla \boldsymbol{f}(\boldsymbol{x}_{k+1}) - \nabla \boldsymbol{f}(\boldsymbol{x}_k))\|_{\mathcal{B}} \leqslant \\ &\sigma_{\mathcal{B}} \|\boldsymbol{z}_k - \mathcal{B}_\infty \boldsymbol{z}_k\|_{\mathcal{B}} + \sigma_{\mathcal{B}} u L \|I_{np} - \mathcal{B}_\infty\| \|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|. \end{aligned} \tag{16}$$

考虑式(16)的右边第2项. 由$\mathcal{A}\mathcal{A}_\infty = \mathcal{A}_\infty$可知, $\mathcal{A}\mathcal{A}_\infty - \mathcal{A}_\infty$是零矩阵. 又由式(6a)有

$$\begin{aligned} &\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\| = \|\mathcal{A}(\boldsymbol{x}_k - D_k \boldsymbol{z}_k) - \boldsymbol{x}_k\| = \\ &\|(\mathcal{A} - I_{np})(\boldsymbol{x}_k - \mathcal{A}_\infty \boldsymbol{x}_k) - \mathcal{A} D_k \boldsymbol{z}_k\| \leqslant \\ &r\|\mathcal{A} - I_{np}\| \|\boldsymbol{x}_k - \mathcal{A}_\infty \boldsymbol{x}_k\|_{\mathcal{A}} + \bar{\alpha}\|\mathcal{A}\| \|\boldsymbol{z}_k\|. \end{aligned} \tag{17}$$

将式(17)代入式(16)并使用引理4, 可得

$$\|\boldsymbol{z}_{k+1} - \mathcal{B}_\infty \boldsymbol{z}_{k+1}\|_{\mathcal{B}} \leqslant$$

$(ur\sigma_{\mathcal{B}}L\|I_{np}-\mathcal{B}_\infty\|\|\mathcal{A}-I_{np}\|+$
$\bar{\alpha}urL^2\sigma_{\mathcal{B}}\|\mathcal{A}\|\|\mathcal{B}_\infty\|\|I_{np}-\mathcal{B}_\infty\|)\|\boldsymbol{x}_k-\mathcal{A}_\infty\boldsymbol{x}_k\|_{\mathcal{A}}+$
$\bar{\alpha}uL^2\sigma_{\mathcal{B}}\|\mathcal{A}\|\|\mathcal{B}_\infty\|\|I_{np}-\mathcal{B}_\infty\|\|\mathcal{A}_\infty\boldsymbol{x}_k-\mathbf{1}_n\otimes x^*\|+$
$(\sigma_{\mathcal{B}}+\bar{\alpha}utL\sigma_{\mathcal{B}}\|\mathcal{A}\|\|I_{np}-\mathcal{B}_\infty\|)\|\boldsymbol{z}_k-\mathcal{B}_\infty\boldsymbol{z}_k\|_{\mathcal{B}}.\quad(18)$

结合式(10), 式(15)和式(18)即可得证.

以下定理给出了$\mathcal{AB}$-BB算法的主要收敛性结果.

定理 1 设假设1–3成立. 令$\Psi=rd_1\psi_1+d_1\psi_2+t\psi_3$. 定义

$$\widehat{\alpha}=\min\{\frac{1}{nL_f\pi_r^{\mathrm{T}}\pi_c},\frac{\psi_1(1-\sigma_{\mathcal{A}})}{a_1\Psi},\frac{a_4(1-\sigma_{\mathcal{B}})}{2a_5\Psi}\}.$$

若参数c满足$c>\dfrac{1}{\mu\widehat{\alpha}}$, 则$\rho(G_c)<1$, 那么由$\mathcal{AB}$-BB算法产生的迭代点列$\{\boldsymbol{x}_k\}_{k\geqslant1}$以$O(\rho(G_c)^k)$的速度线性地收敛到最优解$\mathbf{1}_n\otimes x^*$。

证 根据文献[38]中的推论8.1.29, 本文推导保障参数c的范围和正向量$\psi=[\psi_1\ \psi_2\ \psi_3]^{\mathrm{T}}$的值使得$G_c\psi<\psi$. 上式可等价地写为

$$a_1\Psi\bar{\alpha}<\psi_1(1-\sigma_{\mathcal{A}}),\tag{19}$$

$$(rLa_2\psi_1+ta_3\psi_3)\bar{\alpha}<\frac{\mu_fa_2\psi_2}{cL},\tag{20}$$

$$d_2a_5\Psi\bar{\alpha}<\psi_3(1-\sigma_{\mathcal{B}})-d_2a_4\psi_1,\tag{21}$$

其中$\Psi=rd_1\psi_1+d_1\psi_2+t\psi_3$. 由于式(21)的右边项必须为正, 则有

$$\psi_1<\frac{\psi_3(1-\sigma_{\mathcal{B}})}{d_2a_4}.\tag{22}$$

再考虑到ψ_1,ψ_2,ψ_3为正数, 则取$\psi_1=\dfrac{1-\sigma_{\mathcal{B}}}{2}$和$\psi_3=d_2a_4$. 那么, 由式(19)–(21)和$\bar{\alpha}\leqslant\dfrac{1}{nL_f\pi_r^{\mathrm{T}}\pi_c}$, 可知

$$\bar{\alpha}<\min\{\frac{1}{nL_f\pi_r^{\mathrm{T}}\pi_c},\frac{\psi_1(1-\sigma_{\mathcal{A}})}{a_1\Psi},\frac{a_4(1-\sigma_{\mathcal{B}})}{2a_5\Psi},\frac{\mu_fa_2\psi_2}{cL(rLa_2\psi_1+td_2a_3a_4)}\}.\tag{23}$$

另一方面, 由引理1得$\bar{\alpha}\leqslant\dfrac{1}{c\mu}$. 进一步, 令$\psi_2=\dfrac{L(rLa_2(1-\sigma_{\mathcal{B}})+2td_2a_3a_4)}{\mu_f\mu a_2}$, 则有下式成立:

$$\frac{\mu_fa_2\psi_2}{cL(rLa_2\psi_1+td_2a_3a_4)}>\frac{1}{c\mu}.\tag{24}$$

定义

$$\widehat{\alpha}=\min\{\frac{1}{nL_f\pi_r^{\mathrm{T}}\pi_c},\frac{\psi_1(1-\sigma_{\mathcal{A}})}{a_1\Psi},\frac{a_4(1-\sigma_{\mathcal{B}})}{2a_5\Psi}\}.$$

结合式(23)–(24)和$\bar{\alpha}\leqslant\dfrac{1}{c\mu}$, 可推出$c>\dfrac{1}{\mu\widehat{\alpha}}$. 证毕.

注 1 定理1表明, 当保障参数c满足某个下界时, 所提出的$\mathcal{AB}$-BB算法可以线性地收敛到问题(1)的最优解. 由定理1可知, 最大步长$\bar{\alpha}$满足$\bar{\alpha}\leqslant\dfrac{1}{nL_f\pi_r^{\mathrm{T}}\pi_c}$, 其中$\pi_r$和$\pi_c$是元素和为1的正向量. 当智能体的个数$n$比较大时, $\pi_r^{\mathrm{T}}\pi_c$的值相当小. 因此, 参数c的值不必取为n, 并且可以是更小的. 从这一点来看, 本文所提出的BB步长比ADBB算法中使用的BB步长更加灵活, 甚至可能更大. 当然, 参数c的取值还依赖于一些其他信息, 例如范数等价常数和收缩系数, 这些信息通常是无法获取的. 因此, 在实际中, c的上界是不可计算的, 本文通过手动调整c的值使得算法达到满意的表现.

注 2 本文所提出的算法和$\mathcal{AB}$算法在理论上都保证了线性收敛性质, 但是它们的收敛速度并不能显式表达. 因此无法在理论上明确比较本文的算法与已有算法的收敛速度的快慢. 与使用固定步长的$\mathcal{AB}$算法相比, 本文所提出的算法的亮点在于每个智能体利用其局部迭代点信息和梯度信息自动地计算其步长. 本文通过仿真实验验证了所提出的算法极大地改善了$\mathcal{AB}$算法的数值表现.

5 数值仿真

本节通过测试实际数据集上的分布式逻辑回归问题来验证所提出的$\mathcal{AB}$-BB算法的有效性, 并且将其性能与其他算法进行对比. 问题模型有下列形式:

$$\min_{x\in\mathbb{R}^p}f(x)=\frac{1}{n}\sum_{i=1}^{n}f_{ij}(x)+\frac{\nu}{2}\|x\|^2,$$

其中: $f_{ij}(x)=\dfrac{1}{m_i}\sum\limits_{j=1}^{m_i}\log(1+\exp(-(M_{ij}^{\mathrm{T}}x)y_{ij}))$, $M_{ij}\in\mathbb{R}^p$是特征向量, $y_{ij}\in\{-1,+1\}$是类标签, $\nu>0$为正则参数. 测试数据集a9a和w8a是从UCI机器学习数据库官方网站下载[1]. 数据集a9a有32500个训练数据, 每个数据的维数为123. 数据集w8a有48000个训练数据, 每个数据的维数为300. 考虑含500个节点的有向图, 其随机生成的边数少于4%. 本文采用一致权重策略生成行和列随机权重矩阵A,B, 即$a_{ij}=1/|\mathcal{N}_i^{\mathrm{in}}|,\forall i$和$b_{ij}=1/|\mathcal{N}_j^{\mathrm{out}}|,\forall j$. 数据集上的数据被均匀地分布到每个智能体上. 在下面所有的图中, 纵轴代表平均残差$\dfrac{1}{n}\sum\limits_{i=1}^{n}\|x_k^i-x^*\|$, 其中$x^*$是通过执行集中式的梯度算法得到的. 设置$\nu=0.01$和初始迭代点$x_0^i=0$.

当初始步长$\alpha_0^i=1.5,i\in\mathcal{V}$和间隔参数$h=3$时, 本文研究了不同的参数$c$对$\mathcal{AB}$-BB算法的影响. 当$c=0.5,1,50,100$和500时, 图1测试了$\mathcal{AB}$-BB在数据集a9a上的表现. 由图1发现, 参数c的取值过小使得分布式BB步长过大, 从而导致算法可能发散. 而参数c的取值过大使得分布式BB步长过小, 这导致算法表现很差. 例如, 当参数c取智能体个数$n=500$时, 由于BB步长过小, $\mathcal{AB}$-BB算法使得平均残差下降不明显. 参数$c=1$表明每个智能体自动计算BB步长, 不受保障参数影响. 此时, $\mathcal{AB}$-BB算法的表现非常好.

[1] https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/.

所以, 在下面的所有实验中, 设置$c=1$.

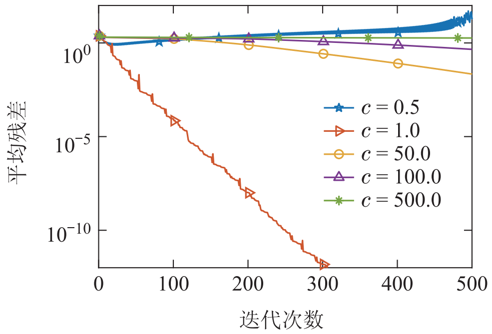


图 1　不同的参数c对$\mathcal{AB}$-BB算法的影响

Fig. 1　Comparison of $\mathcal{AB}$-BB with different parameter c

为了研究初始步长对算法的影响, 基于a9a数据集, 图2对比了$\mathcal{AB}$-BB算法在取不同的初始步长时的结果. 从图2中可以看出, $\mathcal{AB}$-BB算法的性能对初始步长的选取并不敏感.

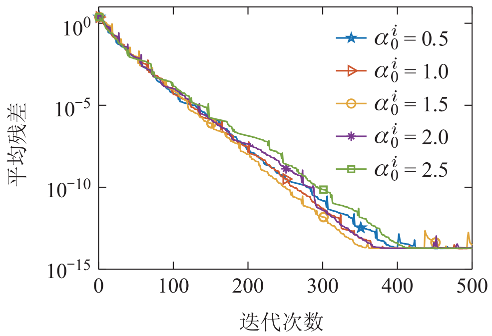


图 2　不同的初始步长对$\mathcal{AB}$-BB算法的影响

Fig. 2　Comparison of $\mathcal{AB}$-BB with different initial step sizes

为了观察间隔参数h对算法的影响, 基于a9a数据集, 本文针对$h=1,2,3,5,8,10$对$\mathcal{AB}$-BB算法进行了测试. 当$h=1$时, $\mathcal{AB}$-BB算法中的BB步长退化为BB1步长. 从图3可以看出, 与$h=1$相比, 当$h=2,3,5,8,10$时, $\mathcal{AB}$-BB算法的表现更好, 这也说明了算法往往比采用单个BB1步长效果好.

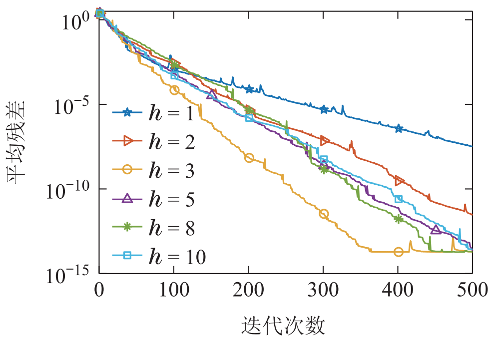
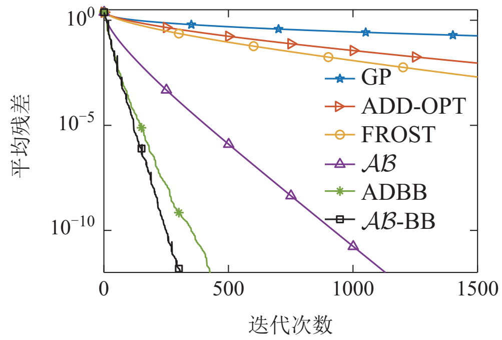


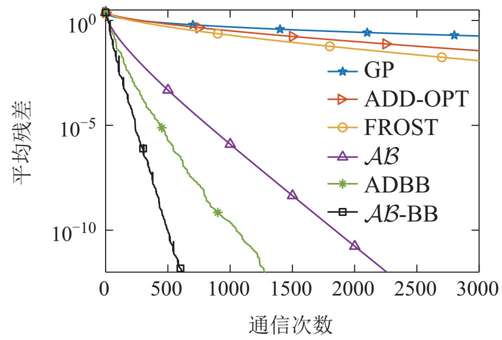
图 3　不同的间隔参数h对$\mathcal{AB}$-BB算法的影响

Fig. 3　Comparison of $\mathcal{AB}$-BB with different parameter h

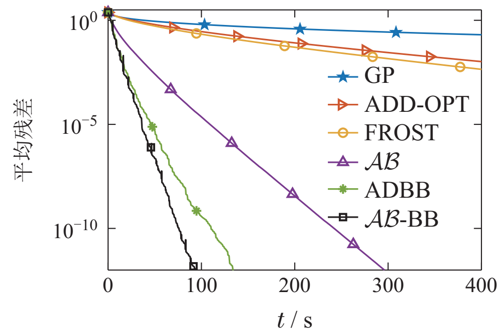
为了进一步说明所提出的算法的有效性, 基于a9a和w8a数据集, 本文将$\mathcal{AB}$-BB算法与GP[21], ADD-OPT[25], FROST[28], $\mathcal{AB}$[29]和ADBB[36]算法作了比较.



(a) 迭代次数



(b) 通信次数



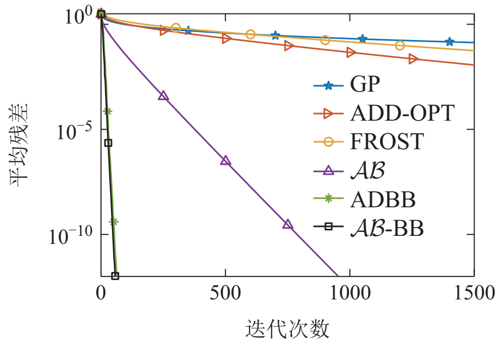
(c) 运行时间

图 4　$\mathcal{AB}$-BB算法与其他算法在a9a上的对比

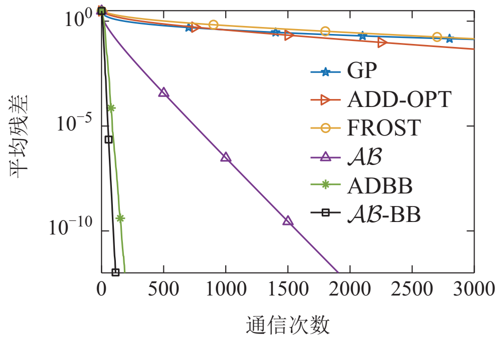
Fig. 4　Comparison of $\mathcal{AB}$-BB and other methods on a9a

需要注意的是ADD-OPT和FROST每次迭代都有3次信息通信, 而GP, $\mathcal{AB}$和$\mathcal{AB}$-BB每次迭代只有2次信息通信. 为了公平起见, 对于ADBB算法, 取多次一致步内循环迭代次数为1, 这样ADBB每次迭代有3次信息通信. 对于a9a和w8a数据集, $\mathcal{AB}$-BB算法分别选取初始步长$\alpha_0^i=1.5, i\in\mathcal{V}$和$\alpha_0^i=2.4, i\in\mathcal{V}$, 其他算法通过多次手动调试得到最优的步长. 图4和图5在2个数据集上对比了所有算法的迭代次数、通信次数和运行时间. 平均残差与迭代次数的关系图直观地反映了算法的收敛速度. 从图4(a)和图5(a)可以看出, 本文所提出的算法实现了线性收敛速度, 并且比采用固定步长的分布式算法的收敛速度更快. 由图4和图5可以看出, 本文所提出的$\mathcal{AB}$-BB算法在迭代次数, 通信次数和时间方面都比使用固定步长的算法的表现好很多; 与现有的BB方法相比ADBB在迭代次数方面有相似的数值表现, 但是在通信次数和时间方面,

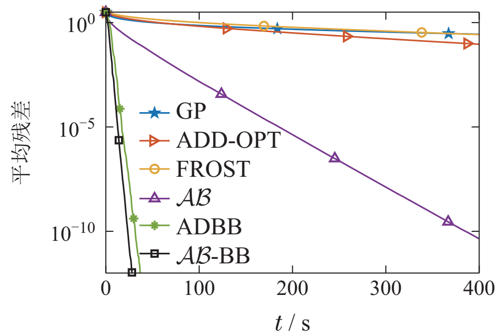
本文算法的表现比ADBB算法更好些. 这是因为本文算法不需要特征向量学习所需的计算和通信成本.



(a) 迭代次数



(b) 通信次数



(c) 运行时间

图 5 $\mathcal{AB}$-BB算法与其他算法在w8a上的对比

Fig. 5 Comparison of $\mathcal{AB}$-BB and other methods on w8a

进一步, 本文对算法进行了定量分析. 基于数据集a9a和w8a, 本文对比了所提出的算法与其他算法在平均残差的精度达到10^{-12}时所需迭代次数, 通信次数和运行时间. 表1给出了所有算法的实验对比数据, 其中"–"表示未统计数据. GP算法是次线性收敛, 收敛速度较慢, 耗时太长, 所以本文没有给出关于GP算法的相关数据. 从表1中可以观察到, 当平均残差达到某个精度时, 与其他算法相比, 本文所提出的算法需要最少的迭代次数, 通信次数和运行时间. 这反映了本文所提出的算法比其他算法更加具有优势.

表 1 不同算法的实验数据结果

Table 1 Experimental data of different algorithms

算法	a9a			w8a		
	迭代次数	通信次数	时间/s	迭代次数	通信次数	时间/s
GP	–	–	–	–	–	–
ADD-OPT	12119	36357	2981.366	11182	33546	5390.115
FROST	8657	25971	2303.388	17158	51474	8468.526
$\mathcal{AB}$	1130	2260	278.150	954	1908	430.352
ADBB	429	1287	118.951	63	189	30.352
$\mathcal{AB}$-BB	**305**	**610**	**77.957**	**58**	**116**	**26.854**

6 结论

本文提出了一种求解基于强连通有向图的优化问题的分布式Barzilai-Borwein梯度跟踪方法. 该方法是在同时使用行和列随机矩阵的$\mathcal{AB}$算法的基础上加入分布式的BB步长得到的方法. 本文所提出的方法中每个智能体利用局部梯度信息自动地计算步长, 使得算法在不增加计算和通信成本的情况下获得更好的数值表现. 当目标函数是光滑和强凸函数时, 本文证明了该算法可以线性地收敛到最优解. 仿真实验表明, 本文算法比使用最优固定步长的$\mathcal{AB}$算法的表现好很多, 并且优于一些常用的分布式梯度算法.

参考文献:

[1] CEVHER V, BECKER S, SCHMIDT M. Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics. *IEEE Signal Processing Magazine*, 2014, 31(5): 32 – 43.

[2] REN W, BEARD R W, ATKINS E M. Information consensus in multivehicle cooperative control. *IEEE Control Systems Magazine*, 2007, 27(2): 71 – 82.

[3] PU S, GARCIA A, LIN Z. Noise reduction by swarming in social foraging. *IEEE Transactions on Automatic Control*, 2016, 61(12): 4007 – 4013.

[4] COHEN K, NEDIĆ A, SRIKANT R. Distributed learning algorithms for spectrum sharing in spatial random access wireless networks. *IEEE Transactions on Automatic Control*, 2017, 62(6): 2854 – 2869.

[5] WANG Long, LU Kaihong, GUAN Yongqiang. Distributed optimization via multi-agent systems. *Control Theory & Applications*, 2019, 36(11): 1820 – 1833.

(王龙, 卢开红, 关永强. 分布式优化的多智能体方法. 控制理论与应用, 2019, 36(11): 1820 – 1833.)

[6] NEDIĆ A, PANG J S, SCUTARI G, et al. *Multi-agent Optimization: Cetraro, Italy 2014*. New York, Springer: 2018.

[7] YANG Zhengquan, PAN Xiaofang, ZHANG Qing, et al. Distributed optimization of multi-agent with communication delay and directed network. *Control Theory & Applications*, 2021, 38(9): 1414 – 1420.
(杨正全, 潘小芳, 张青, 等. 具有通信时延和有向网络的多智能体分布式优化. 控制理论与应用, 2021, 38(9): 1414 – 1420.)

[8] XIE Pei, YOU Keyou, HONG Yiguang, et al. A survey of distributed convex optimization algorithms over networks. *Control Theory & Applications*, 2018, 35(7): 918 – 927.
(谢佩, 游科友, 洪奕光, 等. 网络化分布式凸优化算法研究进展. 控制理论与应用, 2018, 35(7): 918 – 927.)

[9] NEDIĆ A, OZDAGLAR A. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 2009, 54(1): 48 – 61.

[10] YUAN K, LING Q, YIN W. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 2016, 26(3): 1835 – 1854.

[11] SHI W, LING Q, WU G, et al. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 2015, 25(2): 944 – 966.

[12] XU J, REM S, SOH Y C, et al. Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. *The 54th IEEE Conference on Decision and Control*. Osaka, Japan: IEEE, 2015: 2055 – 2060.

[13] NEDIĆ A, OLSHEVSKY A, SHI W, et al. Geometrically convergent distributed optimization with uncoordinated step-sizes. *American Control Conference*. Seattle, WA, USA: IEEE, 2017: 3950 – 3955.

[14] QU G, LI N. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 2018, 5(3): 1245 – 1260.

[15] BERAHAS A S, BOLLAPRAGADA R, KESKAR N S, et al. Balancing communication and computation in distributed optimization. *IEEE Transactions on Automatic Control*, 2019, 64(8): 3141 – 3155.

[16] LI H, LIN Z C. Revisiting EXTRA for smooth distributed optimization. *SIAM Journal on Optimization*, 2020, 30(3): 1795 – 1821.

[17] QU G, LI N. Accelerated distributed Nesterov gradient descent. *IEEE Transactions on Automatic Control*, 2020, 65(6): 2566 – 2581.

[18] LI H, FANG C, YIN W, et al. Decentralized accelerated gradient methods with increasing penalty parameters. *IEEE Transactions on Signal Processing*, 2020, 68: 4855 – 4870.

[19] ZHU M, MARTINEZ S. Discrete-time dynamic average consensus. *Automatica*, 2010, 46(2): 322 – 329.

[20] NEDIĆ A, OLSHEVSKY A. Distributed optimization of strongly convex functions on directed time-varying graphs. *Global Conference on Signal and Information Processing*. Austin, TX, USA: IEEE, 2013: 329 – 332.

[21] NEDIĆ A, OLSHEVSKY A. Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 2015, 60(3): 601 – 615.

[22] KEMPE D, DOBRA A, GEHRKE J. Gossip-based computation of aggregate information. *The 44th Annual IEEE Symposium on Foundations of Computer Science*. Cambridge, MA, USA: IEEE, 2003: 482 – 491.

[23] ZENG J, YIN W. Extrapush for convex smooth decentralized optimization over directed networks. *Journal of Computational Mathematics*, 2017, 35(4): 383 – 396.

[24] XI C, KHAN U A. DEXTRA: A fast algorithm for optimization over directed graphs. *IEEE Transactions on Automatic Control*, 2017, 62(10): 4980 – 4993.

[25] XI C, XIN R, KHAN U A. ADD-OPT: Accelerated distributed directed optimization. *IEEE Transactions on Automatic Control*, 2018, 63(5): 1329 – 1339.

[26] NEDIĆ A, OLSHEVSKY A, SHI W. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 2017, 27(4): 2597 – 2633.

[27] XI C, MAI V S, XIN R, et al. Linear convergence in optimization over directed graphs with row-stochastic matrices. *IEEE Transactions on Automatic Control*, 2018, 63(10): 3558 – 3565.

[28] XIN R, XI C, KHAN U A. Frost-fast row-stochastic optimization with uncoordinated step-sizes. *EURASIP Journal on Advances in Signal Processing*, 2019, 2019(1): 1 – 14.

[29] XIN R, KHAN U A. A linear algorithm for optimization over directed graphs with geometric convergence. *IEEE Control Systems Letters*, 2018, 2(3): 315 – 320.

[30] PU S, SHI W, XU J, et al. Push-pull gradient methods for distributed optimization in networks. *IEEE Transactions on Automatic Control*, 2021, 66(1): 1 – 16.

[31] XIN R, KHAN U A. Distributed heavy-ball: A generalization and acceleration of first-order methods with gradient tracking. *IEEE Transactions on Automatic Control*, 2020, 65(6): 2627 – 2633.

[32] GAO J, LIU X W, DAI Y H, et al. A family of distributed momentum methods over directed graphs with linear convergence. *IEEE Transactions on Automatic Control*, 2023, 68: 1085 – 1092. DOI: 10.1109/TAC.2022.3160684.

[33] LI H, CHENG H, WANG Z, et al. Distributed Nesterov gradient and heavy-ball double accelerated asynchronous optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 2021, 32(12): 5723 – 5737.

[34] GAO J, LIU X W, DAI Y H, et al. Achieving geometric convergence for distributed optimization with Barzilai-Borwein step sizes. *Science China Information Sciences*, 2022, 65(4): 149204.

[35] BURDAKOV O, DAI Y H, HUANG N. Stabilized Barzilai-Borwein method. *Journal of Computational Mathematics*, 2019, 37(6): 916 – 936.

[36] HU J, CHEN X, ZHENG L, et al. The Barzilai-Borwein method for distributed optimization over unbalanced directed networks. *Engineering Applications of Artificial Intelligence*, 2021, 99: 104151.

[37] DAI Y H, HUANG Y K, LIU X W. A family of spectral gradient methods for optimization. *Computational Optimization and Applications*, 2019, 74(1): 43 – 65.

[38] HORN R A, JOHNSON C R. *Matrix Analysis*. 2nd ed. New York, NY, USA: Cambridge University Press, 2013.

[39] BARZILAI J, BORWEIN J M. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 1988, 8(1): 141 – 148.

[40] DAI Y H. Alternate step gradient method. *Optimization*, 2003, 52: 395 – 415.

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

作者简介:

高　娟　博士研究生, 目前研究方向为多智能体系统的分布式优化算法和理论, E-mail: gaojuan514@163.com;

刘新为　博士, 教授, 博士生导师, 目前研究方向为最优化算法、理论, 及其在控制与人工智能中的应用, E-mail: mathlxw@hebut.edu.cn.