

DOI: 10.13718/j.cnki.xdzk.2022.01.010

基于 Bandit 反馈的在线分布式镜面下降算法

朱小梅^{1,2}, 李觉友¹

1. 重庆师范大学 数学科学学院, 重庆 401331; 2. 重庆两江新区博雅小学校, 重庆 401121

摘要: 针对在线分布式优化中一类损失函数梯度信息获取困难的问题, 提出一种基于 Bandit 反馈的在线分布式镜面下降(ODMD-B)算法. 首先, 推广在线分布式镜面梯度下降(ODMD)算法到免梯度的情形, 提出了一种新的仅利用函数值信息来对梯度进行估计的方法即 Bandit 反馈, 其关键在于利用损失函数值信息逼近梯度信息, 能有效克服梯度信息难以获取或计算复杂的困难. 然后, 给出算法的收敛性分析. 结果表明算法的收敛速度为 $O(\sqrt{T})$, 其中 T 是迭代次数. 最后, 使用投资组合选择模型进行了数值仿真实验. 实验结果表明, ODMD-B 算法的收敛速度与已有的 ODMD 算法的收敛速度接近. 对比 ODMD 算法, 本文所提出算法的优点在于仅仅使用了计算花费较小的函数值信息, 使其更适用于梯度信息难以获取的优化问题.

关键词: 在线学习; 分布式优化; 镜面下降算法; Bandit 反馈;

Regret 界

中图分类号: O224; O236

文献标志码: A

开放科学(资源服务)标识码(OSID):

文章编号: 1673-9868(2022)01-0099-09



Online Distributed Mirror Descent Algorithm Based on Bandit Feedback

ZHU Xiaomei^{1,2}, LI Jueyou¹

1. School of Mathematical Sciences, Chongqing Normal University, Chongqing 401331, China;

2. Boya Primary School in Liangjiang New District, Chongqing 401121, China

Abstract: To address the difficulty of obtaining gradient information of a class of loss function in online distributed optimization, an online distributed mirror descent (ODMD-B) algorithm based on bandit feedback was proposed. Firstly, we extend the ODMD algorithm to the gradient-free setup where the gradient of objective functions cannot be directly obtained, and propose the ODMD-B algorithm by using the value of objective function to estimate its gradient. Thus, it overcomes the difficulty in obtaining gradient information involving complicated or expensive computation. Then, the convergence analysis of the algorithm is given. The results show that the convergence rate of the algorithm is order of \sqrt{T} , where T is the total number of iterations. Finally, numerical simulations on the portfolio selection model are used to show the effectiveness of the proposed algorithm. Results of experiment showed that the performance of the ODMD-B algo-

收稿日期: 2020-03-19

基金项目: 国家自然科学基金项目(11871128, 11971083).

作者简介: 朱小梅, 硕士, 主要从事分布式优化算法研究.

通信作者: 李觉友, 博士, 教授.

rithm is very close to that of the existing ODMD algorithm. When compared with the ODMD algorithm, the advantage of our ODMD-B algorithm is that it only uses the function value information with inexpensive computational cost, making it more suitable for more general problems where gradient information is difficult to obtain.

Key words: online learning; distributed optimization; mirror descent algorithm; bandit feedback; regret estimate

随着数据规模的爆炸式增长,集中式优化算法因受限于单机的计算瓶颈而难以求解大规模优化问题.而多机协作的分布式机制可以大大降低单机的计算负担.同时,在分布式网络中,节点之间通过相互协作,可以有效解决诸如智能电网、传感器网络等大规模问题,并能提高数据传递效率,增强网络鲁棒性^[1].但在应用中,分布式网络通常是动态变化的,而在线学习具有实时更新模型的特性,能够根据数据的变化动态地调整模型^[2],进而可高效地完成对大量实时数据的处理,已经在机器学习、在线推荐系统和资源分配等方面体现出了重要的应用价值.

分布式优化算法已成为有效求解大规模优化问题的热点算法之一,如分布式次梯度算法^[3-4]、分布式对偶平均算法和分布式 ADMM 算法^[1]等等,但这些算法是基于 Euclidean 距离设计的,仅能处理一些带有简单约束的优化问题.文献[5]提出了一类镜面梯度下降算法,该算法是将传统的基于 Euclidean 距离的梯度投影算法推广到更一般的基于 Bregman 散度的一类投影算法.针对约束的几何特性,通过适当选择 Bregman 散度,使得镜面梯度下降算法能有效地解决诸如带有单纯形约束的高维优化问题.文献[6]研究表明,镜面梯度下降算法在大规模优化问题中极为有效.文献[7]提出了一种分布式镜面随机梯度下降算法.虽然该算法能够有效处理海量数据,但由于环境的不确定性,损失函数具有时变性和不确定性,导致分布式镜面随机梯度下降算法并不能对网络产生的大量分布式数据流进行实时处理,造成网络资源和时间的浪费.针对损失函数是时变的情形,一种有效的处理方式是在线学习,因为在线学习不仅可以利用具有时变的损失函数来模型化多节点网络系统的不确定性,而且可以方便地对网络节点的动态数据流进行实时处理.最近,文献[8]提出了一种在线分布式镜面梯度下降(ODMD)算法,并给出了 Regret 界的估计是 $O(\sqrt{T})$,这里 T 是迭代次数.

然而,上述大多数算法都需要计算梯度.但在许多实际应用场景中,梯度信息往往难以获取.因此,设计出一种免梯度计算的有效算法显得非常有意义.受文献[9]的启发,本文提出了一种基于 Bandit 反馈的免梯度在线分布式镜面下降算法.不同于 ODMD 算法,我们提出的算法只需要简单计算函数值,主要是利用损失函数在一些随机点的函数值信息去逼近损失函数的梯度,从而避免了直接计算损失函数的梯度.我们设计的算法主要有两个优点:①能有效解决梯度信息难以获取的困难;②能有效处理约束集较复杂(如单纯形约束集)的优化问题.

1 准备工作

1.1 图论

用图 $G = (V, E)$ 表示节点与节点之间的信息交换情况,其中 $V = \{1, \dots, N\}$ 表示由 N 个节点构成的集合, E 表示相邻节点构成的边集,即 $(i, j) \in E$ 当且仅当节点 i 从节点 j 直接获得信息,并用 $N_i = \{j \mid (i, j) \in E\}$ 来表示由节点 i 所有邻居构成的集合.非负矩阵 \mathbf{A} 为图 $G = (V, E)$ 的权重矩阵,满足 $[\mathbf{A}]_{ij} > 0$,当且仅当 $(i, j) \in E$,否则 $[\mathbf{A}]_{ij} = 0$.若对于任意的 $i, j \in V$,有 $\sum_{i=1}^N [\mathbf{A}]_{ij} = \sum_{j=1}^N [\mathbf{A}]_{ij}$,则称图 $G = (V, E)$ 为平衡图,反之为非平衡图.若 $\mathbf{A}^T = \mathbf{A}$,则称图 $G = (V, E)$ 为无向图.

假设 1^[1] 假设图 $G = (V, E)$ 是连通的,即任意两个节点之间都存在一条路径.假设图 $G = (V, E)$ 对应的权重矩阵 \mathbf{A} 是双随机矩阵,即 \mathbf{A} 满足 $\sum_{j=1}^N [\mathbf{A}]_{ij} = 1, \forall i \in V; \sum_{i=1}^N [\mathbf{A}]_{ij} = 1, \forall j \in V$.

1.2 Bregman 散度

定义 1^[8] (μ -强凸函数) 称一个函数 $h: \mathcal{X} \rightarrow \mathbb{R}$ 是 μ -强凸的, 如果对于任意的 $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, 有

$$h(\mathbf{y}) \geq h(\mathbf{x}) + \langle \nabla h(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle + \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|^2$$

定义 2^[8] (Bregman 散度) 设 $h(\cdot)$ 是 1-强凸函数, 定义如下关于函数 h 的 Bregman 散度 $D_h(\cdot, \cdot)$

$$D_h(\mathbf{x}, \mathbf{y}) = h(\mathbf{x}) - h(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla h(\mathbf{y}) \rangle$$

Euclidean 距离和 Kullback-Leiber(KL) 散度为两种常用的 Bregman 散度.

结合定义 1 和 2 可以得到有关 Bregman 散度的一个重要性质^[8]

$$D_h(\mathbf{x}, \mathbf{y}) \geq \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 \quad (1)$$

引理 1^[8] 设 \mathcal{X} 是 \mathbb{R}^m 中的一个凸集, $h: \mathcal{X} \rightarrow \mathbb{R}$ 是一个定义在 \mathcal{X} 中关于 $\|\cdot\|$ 的 1-强凸函数. 定义 $\hat{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \{\langle \mathbf{a}, \mathbf{x} \rangle + D_h(\mathbf{x}, \mathbf{c})\}$, 则有以下不等式成立

$$\langle \hat{\mathbf{x}} - \mathbf{d}, \mathbf{a} \rangle \leq D_h(\mathbf{d}, \mathbf{c}) - D_h(\mathbf{d}, \hat{\mathbf{x}}) - D_h(\hat{\mathbf{x}}, \mathbf{c}), \quad \forall \mathbf{d} \in \mathcal{X} \quad (2)$$

1.3 Bandit 反馈

现有在线分布式算法大多是基于完全信息设计的, 即节点可以获得其决策空间中任何一个位置的梯度信息. 而在许多应用场景中, 节点一般获得的是带噪音的、有缺漏或有限的信息反馈, 称这种不能直接获得损失函数梯度的情形为 Bandit 反馈机制^[9], 即节点只能访问自身损失函数在一些随机点上的函数值, 但并不知道其梯度信息. 众所周知, 大多数优化算法的设计是基于损失函数的梯度设计的, 而在 Bandit 反馈中, 面临着梯度信息不可用的问题. 为此, 我们设计了仅利用损失函数值来逼近梯度的 Bandit 算法. 下面给出根据函数在随机点上的函数值来近似估计其梯度的一些结果.

给定一个函数 $f: \mathbb{R}^m \rightarrow \mathbb{R}$, 对某个很小的数 $\xi > 0$, 定义 $f(\mathbf{x})$ 的一个光滑化近似函数如下

$$\hat{f}(\mathbf{x}) = E_{\mathbf{v} \sim U(B)} [f(\mathbf{x} + \xi \mathbf{v})] \quad (3)$$

其中 $U(B)$ 表示服从单位球 $B = \{\mathbf{x} \in \mathbb{R}^m \mid \|\mathbf{x}\| \leq 1\}$ 的均匀分布, $E[\cdot]$ 表示期望.

引理 2^[9] 设 $S = \{\mathbf{x} \in \mathbb{R}^m \mid \|\mathbf{x}\| = 1\}$ 表示 \mathbb{R}^m 中的单位球面, 有以下结果:

(i) $\nabla \hat{f}(\mathbf{x}) = \frac{m}{\xi} E_{\mathbf{v} \sim U(S)} [f(\mathbf{x} + \xi \mathbf{v}) \mathbf{v}]$, 其中 \mathbf{v} 为服从 S 上均匀分布的随机向量.

(ii) 若 f 是凸的且 G -Lipschitz 连续的函数, 则 \hat{f} 也是凸的且 G -Lipschitz 连续的函数.

(iii) 对某些 $\alpha \in (0, 1)$ 和任意的 $\mathbf{x} \in (1 - \alpha)\mathcal{X}$, 有 $B(\mathbf{x}, r) \subset \mathcal{X}$, 其中 $B(\mathbf{x}, r)$ 表示以 \mathbf{x} 为中心 r 为半径的单位球, 即 $B(\mathbf{x}, r) = \{\mathbf{Z} \in \mathbb{R}^m \mid \|\mathbf{Z} - \mathbf{x}\| \leq r\}$.

从 \hat{f} 的定义知, 函数 \hat{f} 是 f 的一个光滑近似. 由引理 2 知, $\nabla \hat{f}$ 也可以看做是 ∇f 的近似. 为此, 本文主要思想是利用计算函数值 $\frac{m}{2\xi} [f(\mathbf{x} + \xi \mathbf{v}) - f(\mathbf{x} - \xi \mathbf{v})]$ 来近似函数 f 的梯度.

2 问题与算法

2.1 问题

假设 T 是一个事先给定的时间周期, 在每一时刻 $t=1, \dots, T$, 网络中的每个节点 i 需要从凸的紧集 $\mathcal{X} \subseteq \mathbb{R}^m$ 中给出一个本地估计 $\mathbf{x}_{i,t}$, 进而对未知的本地凸损失函数 $f_{i,t}(\mathbf{x}_{i,t})$ 进行观测^[10]. 在时刻 t , 节点 i 仅知道直到前 $t-1$ 时刻的函数信息 $\{f_{i,s}\}_{s=1}^{t-1}$, 而不知道 t 时刻以后的函数信息 $\{f_{i,s}\}_{s=t+1}^T$.

考虑图 $G = (V, E)$ 上的在线分布式凸优化问题如下

$$\min_{\mathbf{x} \in \mathcal{X}} \sum_{t=1}^T f_t(\mathbf{x}) \quad (4)$$

其中: \mathcal{X} 是一个凸紧集, $f_t(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N f_{i,t}(\mathbf{x})$, 且函数 $f_{i,t}(\mathbf{x}): \mathcal{X} \rightarrow \mathbb{R}$ 仅节点 i 自己知道.

本文的目标是生成一系列的决策点 $\{x_{i,t} \mid i \in V, t=1, \dots, T\}$ 相对于拥有完全信息而做出的最优决策

$x^* \in \operatorname{argmin}_{x \in \mathcal{X}} \sum_{i=1}^T \sum_{t=1}^N f_{i,t}(x)$ 的 Regret^[8] 界

$$R_T(x_{i,t}, x^*) = \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T (f_t(x_{i,t}) - f_t(x^*)) \quad (5)$$

尽可能小. 通常要求 Regret 界 $R_T(x_{i,t}, x^*)$ 相对于 T 是次线性的, 即 $\lim_{T \rightarrow \infty} \left(\frac{R_T(x_{i,t}, x^*)}{T} \right) = 0$. 随着 T 的增大, 使其可以逼近最优决策 x^* .

假设 2 (i) 对 $\forall i \in V$ 和 $\forall t \in \{1, \dots, T\}$, $f_{i,t}(\cdot)$ 在 \mathcal{X} 上是 G -Lipschitz 连续的, 即对 $\forall x, y \in \mathcal{X}$, 有 $|f_{i,t}(x) - f_{i,t}(y)| \leq G \|x - y\|$.

(ii) 存在两个正常数 γ 和 Γ 使得: $\gamma B \subset \mathcal{X} \subset \Gamma B$, 其中 $B = \{x \in \mathbb{R}^m \mid \|x\| \leq 1\}$.

由假设 2 知 $f_{i,t}(\cdot)$ 的次梯度 $\nabla f_{i,t}(\cdot)$ 在 \mathcal{X} 上是一致有界的, 即 $\|\nabla f_{i,t}\| \leq G$.

2.2 基于 Bandit 反馈的在线分布式镜面下降算法

由光滑化函数的定义(3), 我们给出问题(4)中节点 i 在时刻 t 的损失函数 $f_{i,t}$ 的一个光滑化函数

$$\hat{f}_{i,t}(x) = E_{v \sim U(B)} [f_{i,t}(x + \xi v)] \quad (6)$$

根据引理 2, 我们可以得到 $\nabla \hat{f}_{i,t}(\cdot)$ 在点 $x_{i,t}$ 的一个无偏估计如下

$$\nabla \hat{f}_{i,t}(x_{i,t}) = \frac{m}{2\xi} E \{ [f_{i,t}(x_{i,t} + \xi u_{i,t}) - f_{i,t}(x_{i,t} - \xi u_{i,t})] u_{i,t} \mid x_{i,t} \} \quad (7)$$

在时刻 t , 只有 $x_{i,t}$ 被确定后才可以获得 $f_{i,t}$, 进而才能估计 $f_{i,t}(x_{i,t} \pm \xi u_{i,t})$, 其中 $u_{i,t} \sim U(S)$.

定义如下的 $\hat{g}_{i,t}$ 为 $\nabla \hat{f}_{i,t}(x_{i,t})$ 在点 $x_{i,t}$ 的一个近似

$$\hat{g}_{i,t} = \frac{m}{2\xi} [f_{i,t}(x_{i,t} + \xi u_{i,t}) - f_{i,t}(x_{i,t} - \xi u_{i,t})] u_{i,t} \quad (8)$$

对 $\hat{g}_{i,t}$ 的估计有以下两个结果^[9]:

$$\|\hat{g}_{i,t}\| \leq mG, E[\hat{g}_{i,t} \mid x_{i,t}] = \nabla \hat{f}_{i,t}(x_{i,t}). \quad (9)$$

算法 1 ODMD-B 算法

- 1) 输入: 最大迭代次数 T , 步长 η_t , 收缩系数 α , 光滑化参数 ξ ;
- 2) 初始化: $x_{i,1} \in (1-\alpha)\mathcal{X}$, $y_{i,1} = x_{i,1}$, $\forall i \in V$;
- 3) for $t=1, \dots, T$ do
- 4) for $i=1, \dots, N$ do
- 5) 利用决策 $x_{i,t}$, 观测局部即时损失函数 $f_{i,t}(x)$;
- 6) 利用(8)式计算梯度估计 $\hat{g}_{i,t}$;
- 6) 更新 $x_{i,t+1} = \operatorname{argmin}_{x \in (1-\alpha)\mathcal{X}} \{ \langle x, \eta_t \hat{g}_{i,t} \rangle + D_h(x, y_{i,t}) \}$;
- 7) 更新 $y_{i,t+1} = \sum_{j=1}^N [A]_{ij} x_{j,t+1}$;
- 8) end for
- 9) end for
- 10) 输出: $\{x_{i,T+1}\}_{i=1}^N$

注 1 算法 1 中步 2) 初始化是为了保证(8)式的 $\hat{g}_{i,t}$ 在点 $x_{i,t} \pm \xi u_{i,t}$ 有定义. 结合算法 1 的步 2)、步 7) 和引理 2 知, 迭代序列 $\{x_{i,t}\}_{t \geq 1}$ 保持在可行域 \mathcal{X} 中.

注 2 算法 1 中步 7) 表示基于 Bregman 的镜面下降算法的局部更新, 其中 $\hat{g}_{i,t}$ 是原来损失函数 $f_{i,t}$ 光滑化后的一个梯度估计, 它仅仅涉及便利的函数值计算.

3 收敛性分析

这一部分主要论述本文提出的 ODMD-B 算法在局部损失函数为凸时的 Regret 界. 分析思路如下: 首先, 估计出各节点的网络化误差, 见引理 4; 然后, 引入一个辅助函数并给出一个重要的引理, 见引理 5; 最后, 根据已建立的结果得出主要结果.

引理 3^[8] 如果假设 1 和 2 均成立, 且 $D_h(\mathbf{x}, \mathbf{y})$ 关于第二个变量 \mathbf{y} 是凸的, 则

$$\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \left(\frac{1}{\eta_t} D_h(\mathbf{x}^*, \mathbf{y}_{i,t}) - \frac{1}{\eta_t} D_h(\mathbf{x}^*, \mathbf{x}_{i,t+1}) \right) \leq \frac{R^2}{\eta_{T+1}} \quad (10)$$

其中 $R^2 = \sup_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} D_h(\mathbf{x}, \mathbf{y})$.

引理 4 如果假设 1 和 2 均成立, 序列 $\{\mathbf{x}_{i,t}\}$ 是由算法 1 迭代产生的, 则对任意的 $i \in V$ 和 $t \geq 1$, 有

$$\|\mathbf{x}_{i,t+1} - \bar{\mathbf{x}}_{t+1}\| \leq mG \sqrt{N} \sum_{l=0}^t \eta_l \sigma_2^{t-l}(\mathbf{A}) \quad (11)$$

其中: $\bar{\mathbf{x}}_{t+1} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_{i,t+1}$, $\{\eta_t\}$ 是非负的且关于 t 呈递减的步长序列, $\sigma_2(\mathbf{A}) \in (0, 1)$ 为权重矩阵 \mathbf{A} 的第二大奇异值.

证 根据算法 1 的步 7) 和引理 1, 有

$$\eta_t \langle \mathbf{x}_{i,t+1} - \mathbf{y}_{i,t}, \hat{\mathbf{g}}_{i,t} \rangle \leq -D_h(\mathbf{y}_{i,t}, \mathbf{x}_{i,t+1}) - D_h(\mathbf{x}_{i,t+1}, \mathbf{y}_{i,t})$$

由(1)式可得

$$\|\mathbf{x}_{i,t+1} - \mathbf{y}_{i,t}\|^2 \leq \eta_t \langle \mathbf{y}_{i,t} - \mathbf{x}_{i,t+1}, \hat{\mathbf{g}}_{i,t} \rangle \leq \eta_t \|\hat{\mathbf{g}}_{i,t}\| \cdot \|\mathbf{y}_{i,t} - \mathbf{x}_{i,t+1}\| \quad (12)$$

从而 $\|\mathbf{x}_{i,t+1} - \mathbf{y}_{i,t}\| \leq mG\eta_t$. 接下来, 令 $\mathbf{e}_{i,t} = \mathbf{x}_{i,t+1} - \mathbf{y}_{i,t}$, $\bar{\mathbf{e}}_t = \frac{1}{N} \sum_{i=1}^N \mathbf{e}_{i,t}$, 根据算法 1 的步 8), 有

$$\mathbf{x}_{i,t+1} = \mathbf{y}_{i,t} + \mathbf{e}_{i,t} = \sum_{j=1}^N [\mathbf{A}]_{ij} \mathbf{x}_{j,t} + \mathbf{e}_{i,t} = \sum_{l=0}^t \sum_{j=1}^N [\mathbf{A}^{t-l}]_{ij} \mathbf{e}_{j,l} \quad (13)$$

由(13)式和假设 1 可得

$$\begin{aligned} \bar{\mathbf{x}}_{t+1} &= \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N [\mathbf{A}]_{ij} \mathbf{x}_{j,t} + \frac{1}{N} \sum_{i=1}^N \mathbf{e}_{i,t} = \\ &= \frac{1}{N} \sum_{j=1}^N \mathbf{x}_{j,t} + \frac{1}{N} \sum_{i=1}^N \mathbf{e}_{i,t} = \bar{\mathbf{x}}_t + \bar{\mathbf{e}}_t = \sum_{l=0}^t \bar{\mathbf{e}}_l = \frac{1}{N} \sum_{l=0}^t \sum_{j=1}^N \mathbf{e}_{j,l} \end{aligned} \quad (14)$$

于是由(13)式和(14)式有

$$\mathbf{x}_{i,t+1} - \bar{\mathbf{x}}_{t+1} = \sum_{l=0}^t \sum_{j=1}^N ([\mathbf{A}^{t-l}]_{ij} - \frac{1}{N}) \mathbf{e}_{j,l}$$

利用双随机矩阵 \mathbf{A} 的性质^[8], 即矩阵 \mathbf{A} 满足

$$\sum_{j=1}^N |[\mathbf{A}^t]_{ij} - \frac{1}{N}| \leq \sqrt{N} \sigma_2^t(\mathbf{A}) \quad (15)$$

这样有

$$\|\mathbf{x}_{i,t+1} - \bar{\mathbf{x}}_{t+1}\| \leq \sum_{l=0}^t \sum_{j=1}^N |[\mathbf{A}^{t-l}]_{ij} - \frac{1}{N}| \|\mathbf{e}_{j,l}\| \leq mG \sqrt{N} \sum_{l=0}^t \eta_l \sigma_2^{t-l}(\mathbf{A}) \quad (16)$$

为了证明定理 1 的结果, 对任意的 $\mathbf{x} \in \mathcal{X}$, 作如下辅助函数

$$F_{i,t}(\mathbf{x}) = \hat{f}_{i,t}(\mathbf{x}) + (\hat{\mathbf{g}}_{i,t} - \nabla \hat{f}_{i,t}(\mathbf{x}_{i,t}))^\top \mathbf{x} \quad (17)$$

引理 5 如果假设 1 和 2 均成立, 序列 $\{\mathbf{x}_{i,t}\}$ 是由算法 1 迭代产生的, 则

$$\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T [F_{i,t}(\mathbf{x}_{i,t}) - F_{i,t}((1-\alpha)\mathbf{x}^*)] \leq \frac{R^2}{\eta_{T+1}} + \frac{m^2 G^2}{2} \sum_{t=1}^T \eta_t + 2m^2 G^2 \sqrt{N} \sum_{t=1}^T \sum_{l=0}^{t-1} \eta_l \sigma_2^{t-l-1}(\mathbf{A}) \quad (18)$$

证 根据函数 $F_{i,t}$ 的定义可得 $\nabla F_{i,t}(\mathbf{x}_{i,t}) = \hat{\mathbf{g}}_{i,t}$. 由定义(17)式知, $F_{i,t}(\cdot)$ 是凸的, 进而有

$$F_{i,t}(\mathbf{x}_{i,t}) - F_{i,t}((1-\alpha)\mathbf{x}^*) \leq \langle \nabla F_{i,t}(\mathbf{x}_{i,t}), \mathbf{x}_{i,t} - (1-\alpha)\mathbf{x}^* \rangle =$$

$$\begin{aligned} & \langle \hat{\mathbf{g}}_{i,t}, \mathbf{x}_{i,t} - (1-\alpha)\mathbf{x}^* \rangle = \\ & \langle \hat{\mathbf{g}}_{i,t}, \mathbf{x}_{i,t} - \mathbf{y}_{i,t} \rangle + \langle \hat{\mathbf{g}}_{i,t}, \mathbf{y}_{i,t} - \mathbf{x}_{i,t+1} \rangle + \\ & \langle \hat{\mathbf{g}}_{i,t}, \mathbf{x}_{i,t+1} - (1-\alpha)\mathbf{x}^* \rangle \end{aligned} \quad (19)$$

接下来对(19)式最后一个不等式右边的三项分别进行估计. 对第一项, 根据算法 1 的步 8 和引理 4, 有

$$\begin{aligned} \langle \hat{\mathbf{g}}_{i,t}, \mathbf{x}_{i,t} - \mathbf{y}_{i,t} \rangle &= \langle \hat{\mathbf{g}}_{i,t}, \mathbf{x}_{i,t} - \bar{\mathbf{x}}_t + \bar{\mathbf{x}}_t - \mathbf{y}_{i,t} \rangle = \langle \hat{\mathbf{g}}_{i,t}, \mathbf{x}_{i,t} - \bar{\mathbf{x}}_t \rangle + \langle \hat{\mathbf{g}}_{i,t}, \bar{\mathbf{x}}_t - \mathbf{y}_{i,t} \rangle \leq \\ & mG \|\mathbf{x}_{i,t} - \bar{\mathbf{x}}_t\| + \sum_{j=1}^N [\mathbf{A}]_{ij} \langle \hat{\mathbf{g}}_{i,t}, \bar{\mathbf{x}}_t - \mathbf{x}_{j,t} \rangle \leq \\ & mG \|\mathbf{x}_{i,t} - \bar{\mathbf{x}}_t\| + mG \sum_{j=1}^N [\mathbf{A}]_{ij} \|\bar{\mathbf{x}}_t - \mathbf{x}_{j,t}\| \leq 2m^2G^2 \sqrt{N} \sum_{l=0}^{t-1} \eta_l \sigma_2^{t-l-1}(\mathbf{A}) \end{aligned} \quad (20)$$

对第二项, 由(9)式有

$$\langle \hat{\mathbf{g}}_{i,t}, \mathbf{y}_{i,t} - \mathbf{x}_{i,t+1} \rangle \leq mG \|\mathbf{y}_{i,t} - \mathbf{x}_{i,t+1}\| \leq \frac{1}{2\eta_t} \|\mathbf{y}_{i,t} - \mathbf{x}_{i,t+1}\|^2 + \frac{\eta_t}{2} m^2G^2 \quad (21)$$

其中(21)式的最后一个不等式由均值不等式得到. 对第三项, 根据引理 1 和(1)式, 有

$$\begin{aligned} \langle \hat{\mathbf{g}}_{i,t}, \mathbf{x}_{i,t+1} - (1-\alpha)\mathbf{x}^* \rangle &\leq \frac{1}{\eta_t} D_h((1-\alpha)\mathbf{x}^*, \mathbf{y}_{i,t}) - \frac{1}{\eta_t} D_h((1-\alpha)\mathbf{x}^*, \mathbf{x}_{i,t+1}) - \frac{1}{\eta_t} D_h(\mathbf{x}_{i,t+1}, \mathbf{y}_{i,t}) \leq \\ & \frac{1}{\eta_t} D_h((1-\alpha)\mathbf{x}^*, \mathbf{y}_{i,t}) - \frac{1}{\eta_t} D_h((1-\alpha)\mathbf{x}^*, \mathbf{x}_{i,t+1}) - \frac{1}{2\eta_t} \|\mathbf{x}_{i,t+1} - \mathbf{y}_{i,t}\|^2 \end{aligned} \quad (22)$$

最后, 将(20)–(22)式代入(19)式, 且在(19)式中对 $t=1, \dots, T$ 求和, 然后再对 $i=1, \dots, N$ 求和, 最后利用引理 3 可证得(18)式成立.

定理 1 如果假设 1 和 2 均成立. 对 $T \geq 1$, 在算法 1 中, 取 $\xi = \frac{1}{T}$, $\alpha = \frac{1}{\gamma T}$, $\eta_t = \frac{\beta}{\sqrt{t}}$, 其中 $\beta > 0$ 是参数. 设序列 $\{\mathbf{x}_{i,t}\}_{t=1}^T$ 是由算法 1 迭代产生的, 则对任意的 $i \in V$, 有如下期望 Regret 上界

$$E[R_T(\mathbf{x}_{i,t}, \mathbf{x}^*)] \leq K_1 + K_2 \sqrt{T} \quad (23)$$

其中

$$\begin{aligned} K_1 &= \left(\frac{\Gamma}{\gamma} + 2\right)G \\ K_2 &= \left(\frac{2R^2}{\beta} + \beta m^2G^2 + \frac{4\beta m(m+1)G^2 \sqrt{N}}{1-\sigma_2(\mathbf{A})}\right) \end{aligned}$$

证 根据 Regret 界的定义(5)式及 f_t 的 G -Lipschitz 连续性, 有

$$\begin{aligned} f_t(\mathbf{x}_{i,t}) - f_t(\mathbf{x}^*) &= f_t(\mathbf{x}_{i,t}) - f_t(\bar{\mathbf{x}}_t) + f_t(\bar{\mathbf{x}}_t) - f_t(\mathbf{x}^*) \leq \\ & G \|\mathbf{x}_{i,t} - \bar{\mathbf{x}}_t\| + \frac{1}{N} \sum_{j=1}^N (f_{j,t}(\mathbf{x}_{j,t}) - f_{j,t}(\mathbf{x}^*)) + \frac{G}{N} \sum_{j=1}^N \|\bar{\mathbf{x}}_t - \mathbf{x}_{j,t}\| \end{aligned} \quad (24)$$

在(24)式中, 利用引理 4, 可得

$$f_t(\mathbf{x}_{i,t}) - f_t(\mathbf{x}^*) \leq \frac{1}{N} \sum_{j=1}^N (f_{j,t}(\mathbf{x}_{j,t}) - f_{j,t}(\mathbf{x}^*)) + 2mG^2 \sqrt{N} \sum_{l=0}^{t-1} \eta_l \sigma_2^{t-l-1}(\mathbf{A}) \quad (25)$$

在(25)式中, 先对 $t=1, \dots, T$ 求和, 然后再对 $i=1, \dots, N$ 求和, 可得

$$\begin{aligned} & \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T [f_t(\mathbf{x}_{i,t}) - f_t(\mathbf{x}^*)] \leq \\ & \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T \left[\frac{1}{N} \sum_{j=1}^N (f_{j,t}(\mathbf{x}_{j,t}) - f_{j,t}(\mathbf{x}^*)) \right] + \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T [2mG^2 \sqrt{N} \sum_{l=0}^{t-1} \eta_l \sigma_2^{t-l-1}(\mathbf{A})] = \\ & \frac{1}{N} \sum_{j=1}^N \sum_{t=1}^T [f_{j,t}(\mathbf{x}_{j,t}) - f_{j,t}(\mathbf{x}^*)] + 2mG^2 \sqrt{N} \sum_{t=1}^T \sum_{l=0}^{t-1} \eta_l \sigma_2^{t-l-1}(\mathbf{A}) \end{aligned} \quad (26)$$

根据 $\hat{f}_{i,t}$ 的定义, 有

$$|\hat{f}_{i,t}(\mathbf{x}_{i,t}) - f_{i,t}(\mathbf{x}_{i,t})| \leq E_{\mathbf{v} \sim U(B)}[|f_{i,t}(\mathbf{x}_{i,t} + \xi \mathbf{v}) - f_{i,t}(\mathbf{x}_{i,t})|] \leq \xi G \quad (27)$$

从而

$$f_{i,t}(\mathbf{x}_{i,t}) - \xi G \leq \hat{f}_{i,t}(\mathbf{x}_{i,t}) \quad (28)$$

根据 $\hat{f}_{i,t}$ 的定义、假设 2 和(27) 式, 有

$$\begin{aligned} |\hat{f}_{i,t}((1-\alpha)\mathbf{x}^*) - f_{i,t}(\mathbf{x}^*)| &\leq |\hat{f}_{i,t}((1-\alpha)\mathbf{x}^*) - f_{i,t}((1-\alpha)\mathbf{x}^*)| + |f_{i,t}((1-\alpha)\mathbf{x}^*) - f_{i,t}(\mathbf{x}^*)| \leq \\ &|E_{\mathbf{v} \sim U(B)}[f_{i,t}((1-\alpha)\mathbf{x}^* + \xi \mathbf{v}) - f_{i,t}((1-\alpha)\mathbf{x}^*)]| + \alpha G \|\mathbf{x}^*\| \leq \\ &\xi G + \alpha G \Gamma \end{aligned} \quad (29)$$

从而由(29) 式可得

$$\hat{f}_{i,t}((1-\alpha)\mathbf{x}^*) \leq f_{i,t}(\mathbf{x}^*) + \xi G + \alpha G \Gamma \quad (30)$$

根据(28) 式和(30) 式, 由(26) 式右边的项得

$$\begin{aligned} E\left\{\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T [f_{i,t}(\mathbf{x}_{i,t}) - f_{i,t}(\mathbf{x}^*)]\right\} &\leq E\left\{\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T [\hat{f}_{i,t}(\mathbf{x}_{i,t}) - \hat{f}_{i,t}((1-\alpha)\mathbf{x}^*) + 2\xi G + \alpha G \Gamma]\right\} = \\ &E\left\{\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T [\hat{f}_{i,t}(\mathbf{x}_{i,t}) - \hat{f}_{i,t}((1-\alpha)\mathbf{x}^*)]\right\} + (2\xi + \alpha \Gamma)GT \end{aligned} \quad (31)$$

由 $F_{i,t}(\cdot)$ 的定义, 有

$$\begin{aligned} \hat{f}_{i,t}(\mathbf{x}_{i,t}) - \hat{f}_{i,t}((1-\alpha)\mathbf{x}^*) &= F_{i,t}(\mathbf{x}_{i,t}) - F_{i,t}((1-\alpha)\mathbf{x}^*) + \\ &(\mathbf{x}_{i,t} - (1-\alpha)\mathbf{x}^*)^\top (\nabla \hat{f}_{i,t}(\mathbf{x}_{i,t}) - \hat{\mathbf{g}}_{i,t}) \end{aligned} \quad (32)$$

注意到 $E[\hat{\mathbf{g}}_{i,t} | \mathbf{x}_{i,t}] = \nabla \hat{f}_{i,t}(\mathbf{x}_{i,t})$, 由(32) 式可得

$$\begin{aligned} E\left\{\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T [\hat{f}_{i,t}(\mathbf{x}_{i,t}) - \hat{f}_{i,t}((1-\alpha)\mathbf{x}^*)]\right\} &= \\ E\left\{\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T [F_{i,t}(\mathbf{x}_{i,t}) - F_{i,t}((1-\alpha)\mathbf{x}^*)]\right\} \end{aligned} \quad (33)$$

对(26) 式取期望, 结合(31) 式和(33) 式, 再根据引理 5, 有

$$\begin{aligned} E\left\{\frac{1}{N} \sum_{i=1}^N \sum_{t=1}^T [f_{i,t}(\mathbf{x}_{i,t}) - f_{i,t}(\mathbf{x}^*)]\right\} &\leq \\ \frac{R^2}{\eta_{T+1}} + \frac{m^2 G^2}{2} \sum_{t=1}^T \eta_t + (2\xi + \alpha \Gamma)GT + 2m(m+1)G^2 \sqrt{N} \sum_{t=1}^T \sum_{l=0}^{t-1} \eta_l \sigma_2^{t-l-1}(\mathbf{A}) \end{aligned} \quad (34)$$

将定理中的 $\xi = \frac{1}{T}$, $\alpha = \frac{1}{\gamma T}$, $\eta_t = \frac{\beta}{\sqrt{t}}$ 代入(34) 式即可得证.

注 3 定理 1 的结果表明, 算法 1 的收敛速度是 $O(\sqrt{T})$, 这与文献[8] 的在线分布式镜面梯度下降算法的收敛速度同阶, 仅仅相差一个光滑化误差项 $(2 + \frac{\Gamma}{\gamma})G$. 通过对比, 本文所提出的算法 1 仅需要计算损失函数的函数值, 并没有涉及复杂的梯度计算, 从而降低了复杂度, 而且算法适用范围更加广泛.

4 数值实验

为了说明算法的有效性, 考虑一个不对股票市场进行任何统计假设的投资组合选择模型, 被称为“通用投资组合选择”模型, 它在在线学习中受到了特别的关注. 下面给出一个投资组合选择模型^[10]

$$\min_{\mathbf{x}_{i,t} \in \Delta_m} \frac{1}{N} \sum_{i=1}^N \sum_{t=1}^N (-\log(\mathbf{r}_{i,t}^\top \cdot \mathbf{x}_{i,t}))$$

其中 Δ_m 为 m 维的单形, 参数 $\mathbf{r}_{i,t} \in \mathbb{R}_+^m$ 表示价格比率是时变的.

我们考虑一个具有 N 个节点的连通网络且满足假设 1. 在 ODMD-B 算法中, 取 $T = 1\,000$, 步长 $\eta_t = \frac{0.06}{\sqrt{t+1}}$, 取光滑化参数 $\xi = \frac{1}{T}$, 压缩系数 $\alpha = \frac{0.4}{T}$. 由于约束集是 m 维的单形, 故采用 KL 散度. 这样, 算

法 1 中步 7) 有如下形式的显式解^[5]

$$[\mathbf{x}_{i,t+1}]_q = \frac{(1-\alpha)[\mathbf{y}_{i,t}]_q \exp(-\eta_i[\hat{\mathbf{g}}_{i,t}]_q)}{\sum_{k=1}^m [\mathbf{y}_{i,t}]_k \exp(-\eta_i[\hat{\mathbf{g}}_{i,t}]_k)} \quad (35)$$

其中 $q=1, \dots, m$. 对于 ODMD-B 算法的性能度量, 考虑平均 Regret 界. 另外, 将本文提出的 ODMD-B 算法与文献[8]提出的 ODMD 算法进行数值对比.

首先, 考虑不同的问题维数对 ODMD-B 算法性能的影响. 图 1a 和图 1b 分别表示 $N=10, m=50$ 和 $N=10, m=100$ 的平均 Regret 界随着迭代次数变化的演化图. 从图 1a 和图 1b 可以看出, 本文提出的 ODMD-B 算法与文献[8]提出的 ODMD 算法都可以收敛. 但 ODMD-B 算法的早期收敛速度稍微慢于 ODMD 算法, 这是因为 ODMD-B 算法只使用了函数值信息, 而 ODMD 算法是直接使用了梯度信息. 但随着迭代次数增大, ODMD-B 算法的平均 Regret 界大致接近 ODMD 算法的平均 Regret 界. 另外, 对比图 1a 和图 1b 中的 ODMD-B 算法, 当 $T=700$ 时, $m=50$ 和 $m=100$ 平均 Regret 界分别为 0.020 52 和 0.027 39, 仅相差 0.007 左右. 这表明问题维数对所提算法性能的影响并不大, 这主要是因为此问题在迭代过程中能够求得显示解.

然后, 考虑不同的节点个数对 ODMD-B 算法性能的影响. 图 1c 和图 1d 分别表示 $N=50, m=100$ 和 $N=100, m=100$ 的平均 Regret 界随着迭代次数变化的演化图. 图 1c 和图 1d 进一步表明两种算法的平均 Regret 界随着迭代次数的增加而趋于一致. 另外, 对比图 1c 和图 1d 可以发现, 随着节点个数的增加, 两种算法的平均 Regret 界随之增加. 相比 ODMD 算法, 本文所提 ODMD-B 算法仅仅使用了函数值信息.

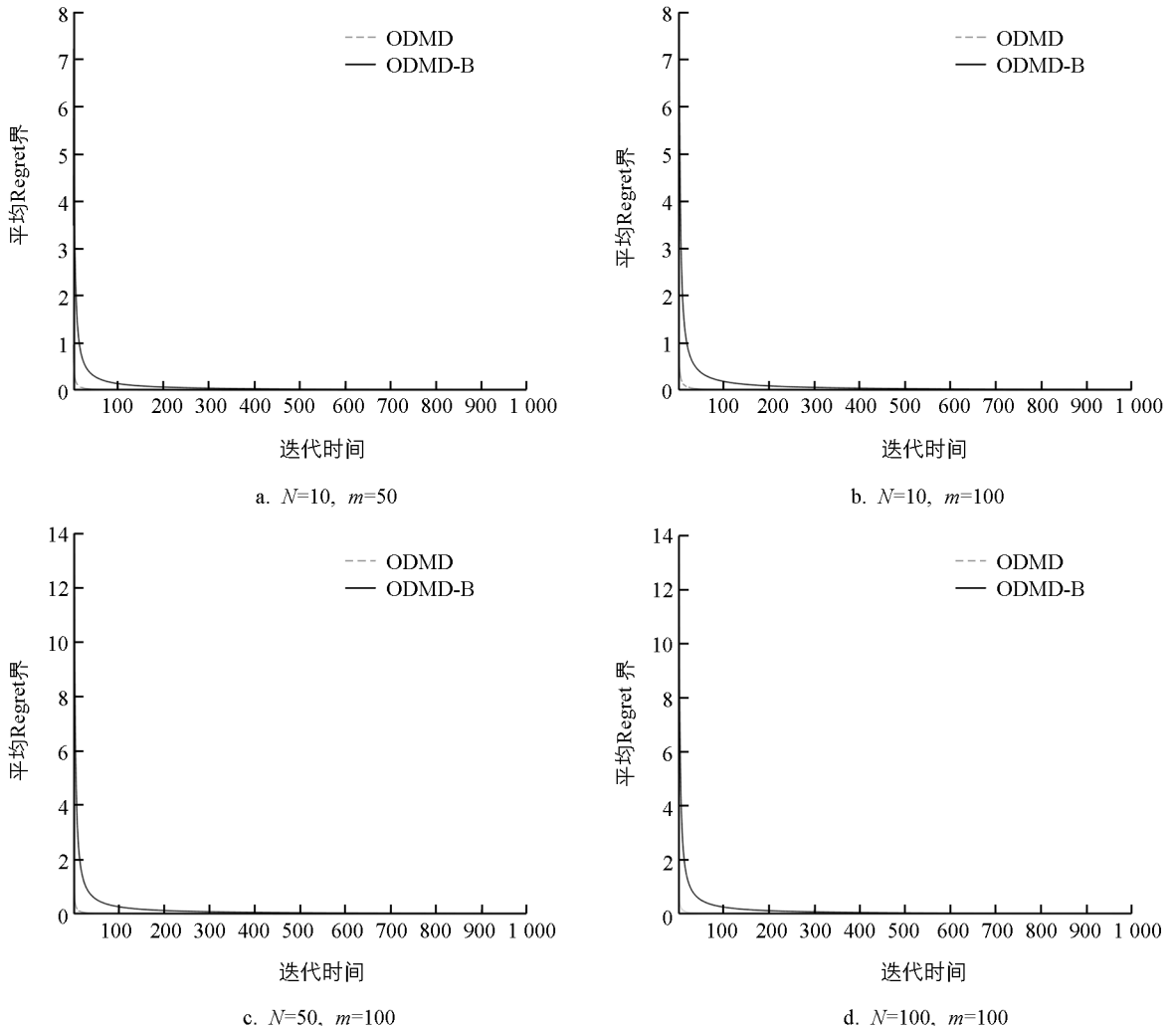


图 1 ODMD-B 算法与 ODMD 算法平均 Regret 界的比较

5 结论

对于梯度获取困难且约束集较复杂的问题, 本文提出了一种 ODMD-B 算法. 该算法与现有的 ODMD 算法的不同点在于将梯度信息反馈改进成基于函数值信息的 Bandit 反馈. 理论结果表明, ODMD-B 算法具有与 ODMD 算法同阶的收敛速度, 即达到 $O(\sqrt{T})$ 的 Regret 界. 但本文提出的 ODMD-B 算法仅仅使用了损失函数值信息, 有效避免了梯度信息获取困难的问题. 未来研究方向可将本文提出的 ODMD-B 算法推广到非平衡有向图切换网络^[11-12]情形.

参考文献:

- [1] 谢佩, 游科友, 洪奕光, 等. 网络化分布式凸优化算法研究进展 [J]. 控制理论与应用, 2018, 35(7): 918-927.
- [2] 张文鹏. 免投影在线学习 [D]. 北京: 清华大学, 2017.
- [3] NEDIC A, OZDAGLAR A. Distributed Subgradient Methods for Multi-Agent Optimization [J]. IEEE Transactions on Automatic Control, 2009, 54(1): 48-61.
- [4] SUNDHAR RAM S, NEDIC A, VEERAVALLI V V. Distributed Stochastic Subgradient Projection Algorithms for Convex Optimization [J]. Journal of Optimization Theory and Applications, 2010, 147(3): 516-545.
- [5] BECK A, TEBoulLE M. Mirror Descent and Nonlinear Projected Subgradient Methods for Convex Optimization [J]. Operations Research Letters, 2003, 31(3): 167-175.
- [6] BEN-TAL A, MARGALIT T, NEMIROVSKI A. The Ordered Subsets Mirror Descent Optimization Method with Applications to Tomography [J]. SIAM Journal on Optimization, 2001, 12(1): 79-108.
- [7] LI J Y, LI G Q, WU Z Y, et al. Stochastic Mirror Descent Method for Distributed Multi-Agent Optimization [J]. Optimization Letters, 2018, 12(6): 1179-1197.
- [8] SHAHRAMPOUR S, JADBABAIE A. Distributed Online Optimization in Dynamic Environments Using Mirror Descent [J]. IEEE Transactions on Automatic Control, 2018, 63(3): 714-725.
- [9] CAO X, LIU K J R. Online Convex Optimization with Time-Varying Constraints and Bandit Feedback [J]. IEEE Transactions on Automatic Control, 2019, 64(7): 2665-2680.
- [10] HAZAN E. Introduction to Online Convex Optimization [J]. Foundations and Trends © in Optimization, 2016, 2(3-4): 157-325.
- [11] 肖丽, 包骏杰, 石熙, 周琳琳. 非平衡有向网络上求解分布式经济分配问题的原始-对偶算法 [J]. 西南大学学报(自然科学版), 2018, 40(11): 48-54.
- [12] 周小清, 李觉友. 一类时变有向图中的 PUSH-SUM 分布式对偶平均优化算法 [J]. 西南师范大学学报(自然科学版), 2019, 44(11): 11-17.

责任编辑 张宥