



网络上的加速在线平均算法

献给林正炎教授 85 寿辰

刘卫东^{1,2}, 刘潇¹, 毛晓军^{1,3*}, 武泽宇¹

1. 上海交通大学数学科学学院, 上海 200240;

2. 上海交通大学人工智能教育部重点实验室, 上海 200240;

3. 上海交通大学教育部科学工程计算重点实验室, 上海 200240

E-mail: weidongl@sjtu.edu.cn, shaw_stat@sjtu.edu.cn, maoxj@sjtu.edu.cn, yufeng168@sjtu.edu.cn

收稿日期: 2024-07-04; 接受日期: 2025-01-17; 网络出版日期: 2025-07-14; * 通信作者

国家自然科学基金 (批准号: 11825104, 12422111 和 12371273)、上海市青年科技启明星计划 (批准号: 23QA1404600) 和中国科协青年人才托举工程 (批准号: 2023QNRC001) 资助项目

摘要 由于对实时收集的海量数据进行处理的需求快速增长, 分布式在线学习已经得到了广泛的研究和应用. 大部分现有的工作都是在集中式框架下研究在线问题, 而本文主要关注去中心化在线平均问题, 并提出一种新颖的加速算法. 本文还将所提出的算法扩展到去中心化在线最小二乘回归问题. 通过建立算法收敛性的理论和统计相合性的理论, 本文阐明所提出的算法的加速效果. 此外, 首次提供了去中心化在线最小二乘估计的中心极限定理. 一系列数值模拟展示本文所提出的算法的可行性和有效性.

关键词 加速算法 去中心化网络 分布式在线学习 在线平均

MSC (2020) 主题分类 62R07, 68T09, 68W15, 93A14

1 引言

近些年来, 随着应用场景的多元化与复杂化, 同时处理海量数据已经成为了机器学习领域的一项巨大挑战. 很多数据往往是由个人终端、雷达阵列等分布在多个地点的设备分开收集, 并且由于传输成本和隐私保护的顾虑, 所有的设备只能在一个去中心化网络中与自己的邻居节点进行信息传输. 作为对传统中心化网络的一个推广, 去中心化网络对于攻击是更加稳健的, 并且不需要设置中心节点, 就能在一定的连接性条件下实现更加灵活的通信方式. 去中心化网络上的优化算法已经成为了近些年来饱受关注的研究方向^[18,20,34], 也在许多实际场景中得到了广泛的应用, 如多智体系统^[5,21]、传感器网络^[15,33]、电力网络^[35] 和机器学习^[38,41] 等.

英文引用格式: Liu W D, Liu X, Mao X J, et al. Accelerated online averaging over networks (in Chinese). Sci Sin Math, 2026, 56: 187–216, doi: 10.1360/SSM-2024-0223

由于流式数据的发展和当前信息时代下快速处理数据的需求, 在线算法和相关的分布式问题逐渐变得非常重要 (参见文献 [16, 27, 43]). 这里的“在线”场景具体指的是数据被按照序列实时收集, 并且网络中的每个节点只能获取到自身的局部样本信息. 图 1 直观展示了在整个去中心化在线模式下, 流数据是怎样被每个机器收集的. 图 1 构建了一个包含 4 个节点 A, B, C 和 D 的简单去中心化网络, 其中节点 A 与 B, C 直接相连, 但是与节点 D 没有连接. 节点 B 的连接模式也与 A 类似. 接下来, 以节点 A 为例去展示这一个网络的运行机制. 在第 k 个轮次, 节点 A 收集到了一个流式样本 a_k , 并且用这个样本对自己的局部参数估计值进行了更新. 之后, 节点 A 仅仅可以与自己的邻居 B 和 C 交流, 但是不能和没有与自身直接相连的节点 D 相互传递信息. 至此便完成了节点 A 上去中心化在线算法的第 k 个轮次, 其中 $k = 1, 2, \dots$

目前对去中心化在线算法的研究还是比较有限的. 大多数现存的工作 [12, 23, 32, 37, 42] 主要是基于随机梯度下降方法讨论一般的去中心化在线优化问题, 但是因为使用了衰减步长, 往往都存在收敛速度慢的问题. 例如, 最近的工作 [32] 考虑了用随机梯度追踪算法去解决一般的去中心化优化问题. 他们得到的收敛速率为 $\sqrt{\mathbb{E}(\|\mathbf{x}_k - \mathbf{1}\mathbf{x}^*\|_2^2)} = O(1/\sqrt{k})$, 其中 $\mathbf{x}_k \in \mathbb{R}^p$ 是他们优化算法在第 k 个轮次产生时的去中心化估计值, 而 $\mathbf{x}^* \in \mathbb{R}^p$ 表示的是目标函数真实的最小值. 相比之下, 本文考虑更加具体的去中心化在线平均问题. 记一共 n 个节点在第 k 个轮次接收到的样本数据为 $\{y_i^{(k)}, 1 \leq i \leq n\}$, 其中 $k = 1, 2, \dots$. 本文的主要目的是对齐所有节点上的估计值, 从而在去中心化网络的每个节点上一致地近似全局平均值 $\bar{y}_{[k]} = \sum_{i=1}^n \sum_{t=1}^k y_i^{(t)} / (nk)$. 本文所提方法在第 k 个轮次的估计向量 $\mathbf{z}^{(k)}/k$ 与全局平均向量 $\mathbf{1}_n \bar{y}_{[k]}$ 之间的距离具有收敛速度 $\sqrt{\mathbb{E}(\|\mathbf{z}^{(k)}/k - \mathbf{1}_n \bar{y}_{[k]}\|_2^2)} = O(1/k)$, 其中 $\mathbf{1}_n$ 表示一个 n 维的全 1 向量, 这比文献 [32] 中给出的结果要更快. 尽管去中心化的在线平均问题可以由一般的去中心化在线优化问题推导出来, 但是我们的结果并不能由任何现存的去中心化一阶算法得到. 目前已有很多工作针对去中心化的平均问题进行了探索, 如文献 [11, 28, 30, 31, 39] 等. 然而, 这些工作都只是在离线的情景下研究了去中心化平均问题, 即要求完整的样本数据需要被提前存储在每个节点上, 并不适用于在线场景. 作为分布式在线学习中的一个基本问题, 分布式在线平均在当前的现实世界中有着广泛的应用, 如协作的多智体系统、自动驾驶车辆的障碍物检测及基于无线传感器的区域环境监测等, 相关

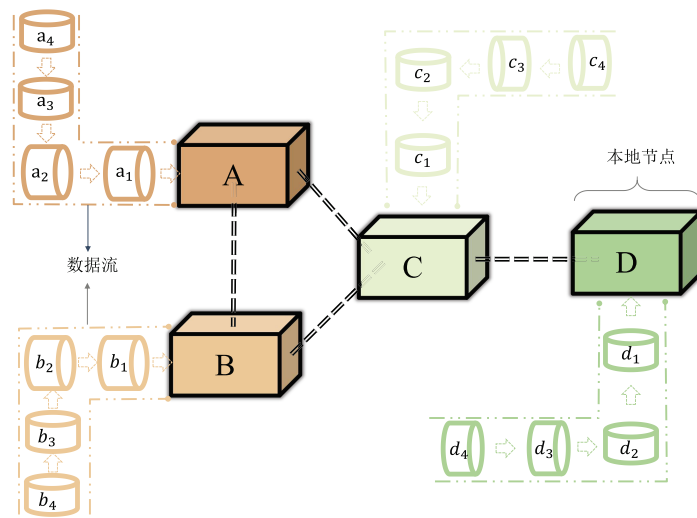


图 1 (网络版彩图) 去中心化在线学习机制示意图. A, B, C 和 D 4 个节点共同构成了去中心化网络, 它们可以与自己的邻居进行信息交互. 图中的小圆柱 a_k, b_k, c_k 和 $d_k, k = 1, 2, \dots$ 代表在第 k 个轮次每个节点分别接收到的流数据

的内容参见文献 [7, 9, 19]. 这些任务都要求我们在从去中心化网络中实时接收信息后, 迅速并有效地作出在线决策. 因此, 探索加速的分布式在线平均算法具有深远的意义.

针对去中心化在线平均问题, 一个自然且可行的方法是直接通过去中心化在线增量求和来近似 n 个节点之间的均值 $\sum_{i=1}^n \sum_{t=1}^k y_i^{(t)} / n$, 然后再除以总共的轮数 k . 这种方法的主要思想是在每一轮中, 每个节点通过添加新收集的样本来更新自己的本地估计值, 然后与邻居节点进行通信以达成共识. 该方法直接由传统的离线去中心化平均拓展而来, 我们将这种估计方法称之为去中心化在线平均估计 (decentralized online average estimation, DOVE). 然而, 这种方法并不能充分利用历史数据, 尤其是当异质性数据存在的情形下具有较慢的收敛速度. 本文的目标是在 DOVE 算法的基础上提出一种更加高效的去中心化在线平均算法. 进一步地, 去中心化在线平均问题还可以推广到去中心化在线最小二乘问题. 最近的工作 [40] 提出了一种去中心化在线最小二乘算法并在理论上为其建立了收敛性结果. 我们也基于所提出的加速去中心化在线平均框架, 将我们的方法拓展到了在网络中的每个节点上在线且一致地近似最小二乘估计.

值得注意的是, 在去中心化网络中, 通信效率很大程度上依赖于网络的拓扑结构 [29]. 一些连接较为稀疏的网络可能会阻碍通信效率, 从而进一步减缓去中心化算法的收敛速度. 因此, 探索是否可以开发具有改进收敛速度的新型去中心化在线算法, 并且在潜在的“不良”网络拓扑结构下仍然能获得理想的收敛速度是非常有意义的. 本文对上述问题提供积极的回答, 并且展示在去中心化在线平均和最小二乘问题中的一种具有广阔前景的加速技术. 本文所提方法的显著贡献总结如下.

- 受滑动平均的启发, 本文建立了一种加速的去中心化在线平均估计 (accelerated decentralized online average estimation, A-DOVE), 并且为该估计的收敛速度建立了扎实的理论保证. 记每次选取的滑动窗宽为 l , 本文证明 A-DOVE 相较于传统方法具有加速因子 $O(1/l)$. 并通过一系列数值实验验证 A-DOVE 方法在收敛速度上具有显著的改进.

- 基于 A-DOVE 方法, 本文提出一种加速去中心化在线最小二乘估计 (accelerated decentralized online least square estimation, A-DOSE). 相关的数值实验表明 A-DOSE 具有加速的收敛速度.

- 在适当的矩条件下, 本文证明 A-DOVE 估计满足均值模型的统计相合性. 除此之外, 本文进一步建立 A-DOSE 估计的中心极限定理. 据我们所知, 这是在去中心化在线框架下首次建立这样的结果.

接下来介绍一些与本文比较相关的工作. 针对在线学习, 目前有很多工作探索了在线中位数回归 [2, 3]、在线排序 [8] 和在线回归 [13, 14] 等问题. 此外, 分布式在线框架下仍然还有很多关于处理海量数据和流数据的重要而且有价值的课题, 特别是对分布式框架下的在线优化问题 [10, 24, 46], 在近近年来收到了广泛关注.

然而上述提到的分布式在线算法都是在中心化的框架下进行的, 容易受到中心节点存储和计算能力的限制, 也存在一定的隐私泄露隐患. 目前也有一部分工作 (如文献 [17, 22, 26, 27, 36, 44, 45]) 研究了去中心化的在线算法. 关于去中心化的在线优化, 绝大部分现有的文献 (如文献 [12, 23, 32, 37, 42]) 依赖于随机梯度下降方法及其变体. 值得注意的是, 文献 [6] 研究了在不平衡网络上的去中心化动态在线线性回归问题. 最近, 文献 [43] 通过梯度下降技术结合全信息反馈或者 Bayes 反馈研究了去中心化在线线性回归. 然而, 他们关注的重点是控制累计遗憾而不是逼近真实参数. 此外, 文献 [1, 4] 从最小二乘的角度解决了去中心化在线线性回归问题.

与本文最相关的工作是文献 [40], 该工作致力于研究去中心化的在线最小二乘算法. 受直接的去中心化在线增量求和的启发, 他们采用矩阵代数设计了算法, 用于回归参数和逆协方差矩阵的扩散策略. 他们在弱合作激励条件下, 建立了算法的殆必收敛理论, 而不需要独立性、平稳性或 Gauss 性质的

假定. 然而, 文献 [40] 的主要贡献于算法的收敛速度, 还未能建立分布式在线估计值的中心极限定理. 相比之下, 我们的工作探讨了去中心化在线最小二乘估计的统计相合性, 并建立了相应的中心极限定理.

本文余下内容的结构如下. 第 2 节提出新颖的去中心化在线平均算法来逼近全局平均值, 并进一步提出新颖的去中心化在线最小二乘回归算法来逼近最小二乘估计. 第 3 节提供针对我们的去中心化在线算法的算法收敛速度和统计性质的坚实理论保证. 第 4 节提供数值模拟的实现, 以展示我们算法的有效性和优势. 第 5 节给出我们的结论并讨论未来可以继续探索的方向.

基本符号和定义 范数 $\|\cdot\|_2$ 被定义为向量的 Euclid 范数或矩阵的 Frobenius 范数. 对于两个随机序列 $\{a_k\}$ 和 $\{b_k\}$, 如果存在一个常数 C 使得 $|a_k| \leq C|b_k|$ 成立, 则记 $a_k = O(b_k)$; 如果当 $k \rightarrow \infty$ 时 $a_k/b_k \rightarrow 0$ 成立, 则记 $a_k = o(b_k)$. 此外, 记 $a_n = O_P(b_k)$, 如果对任意 $0 < \varepsilon < 1$, 存在一个常数 C_ε , 使得 $|a_k| \leq C_\varepsilon|b_k|$ 至少有概率 $1 - \varepsilon$ 成立. 如果 $a_k/b_k \xrightarrow{P} 0$, 则记 $a_k = o_P(b_k)$, 其中 $\xrightarrow{P}$ 表示依概率收敛. 用 $\mathbf{1}_n$ 表示全 1 的 n 维向量, 并在不会产生混淆的前提下将其简写为 $\mathbf{1}$.

2 方法

本节首先介绍一种直观的、通过增量求和的基本去中心化在线平均方法. 在此基础上, 本文提出一种加速的去中心化在线平均算法, 通过滑动平均的思想来改善收敛速度. 此外, 本文还提出一种新颖的加速算法, 通过类似于去中心化在线平均的思想进行去中心化在线最小二乘回归.

2.1 去中心化在线平均

考虑一个被描述为无向图 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 的去中心化网络, 其中 $\mathcal{V}$ 表示节点集合, $|\mathcal{V}| = n$ 表示节点数量, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ 表示边的集合. 节点 i 的邻居节点集合表示为 $\mathcal{N}_i = \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}$. 在网络中, 每个节点 i 只具有对本地数据的计算能力, 并且只能通过在 $\mathcal{E}$ 中的边与其邻居节点 $\mathcal{N}_i$ 交换信息.

在接下来的内容中, 为了方便讨论, 假设每一轮每个机器只收集到一个样本. 值得注意的是, 将一个样本的情形扩展到多个样本的情形并没有技术上的困难. 在第 k 个轮次中, 对于所有 $1 \leq i \leq n$, 节点 i 收集到一个新样本 $y_i^{(k)}$. 定义聚合样本向量 $\mathbf{y}^{(k)} = (y_1^{(k)}, \dots, y_n^{(k)})^T \in \mathbb{R}^n$ 、节点平均值 $\bar{y}^{(k)} = \sum_{i=1}^n y_i^{(k)}/n$ 和全局平均值 $\bar{y}_{[k]} = \sum_{t=1}^k \bar{y}^{(t)}/k = \sum_{t=1}^k \sum_{i=1}^n y_i^{(t)}/(nk)$. 定义一个聚合向量 $\mathbf{z}^{(k)} = (z_1^{(k)}, \dots, z_n^{(k)})^T$, 其中 $z_i^{(k)}/k$ 表示节点 i 在第 k 轮对全局平均值 $\bar{y}_{[k]}$ 的本地估计, 所有节点上估计的平均值记为 $\bar{z}^{(k)} = \sum_{i=1}^n z_i^{(k)}/n$. 用 $z_i^{(k)}/k$ 而不是 $z_i^{(k)}$ 主要是为了表示在时间维数上的平均. 本文的目标是, 随着 k 的增长, 局部估计 $z_i^{(k)}/k$ 能够在所有的机器之间达成共识, 即能够收敛到所有节点之间的平均值 $\bar{z}^{(k)}/k$. 进一步地, 对于所有节点 $1 \leq i \leq n$, $z_i^{(k)}/k$ 能够一致收敛到全局平均值 $\bar{y}_{[k]}$. 为了方便起见, 假设每个节点在本地同时收集样本和更新估计值, 换言之, 节点之间的计算和通信没有延迟.

在传统的离线去中心化平均问题 (即去中心化共识问题) 中, 最常用的方法是去中心化线性迭代 [39], 即在每一轮通信时, 对每一个节点及其邻居的数据作加权平均. 对于给定的网络 $\mathcal{G}$, 用 $\mathbf{W} = (w_{ij})_{1 \leq i, j \leq n} \in [0, 1]^{n \times n}$ 表示一个权重邻接矩阵, 其维数 n 与节点数量相同. 权重邻接矩阵 $\mathbf{W}$ 满足当且仅当 $(i, j) \in \mathcal{E}$ 时 $w_{ij} > 0$, 否则 $w_{ij} = 0$. 每个节点 i 只能访问来自 $\mathcal{N}_i$ 中的节点所传递的本地信息, 但它不需要知道全局信息, 如全局节点数量或 $\mathcal{N}_i$ 以外的节点的网络结构. 我们可以仅依靠关于每个节点 i 的邻居 j 的局部结构信息来确定权重 w_{ij} , $1 \leq i, j \leq n$.

关于权重矩阵的 $\mathbf{W}$ 构造和选取其实是一个备受关注且得到了广泛研究的问题 (参见文献 [29, 31, 39]). 正如文献 [39, 定理 1] 所介绍的, 只要权重矩阵 $\mathbf{W}$ 满足双随机性质 (即 $\mathbf{1}^T \mathbf{W} = \mathbf{1}^T$ 和 $\mathbf{W} \mathbf{1} = \mathbf{1}$) 且所有的对角元素 $w_{ii} > 0$, 则去中心化平均算法就能够收敛. 记 $\rho_i(\cdot)$ 为矩阵的第 i 大奇异值, 文献 [39] 也指出了离线去中心化平均算法的收敛速度为 $\rho_1(\mathbf{W} - \mathbf{1}\mathbf{1}^T/n) = \rho_2(\mathbf{W}) < 1$. 因此不同的 $\mathbf{W}$ 构造方式会影响算法的收敛速度. 理论上, 构造的 $\mathbf{W}$ 的第二大奇异值越小, 算法收敛的就越快.

为了加快收敛, 一类启发式的构造方法是, 利用图的 Laplace 矩阵为每个边分配权重. 具体而言, 记 $\deg(i)$ 表示节点 i 的度, 此时 $\mathbf{W}$ 的具体形式包括常数权重矩阵 [39]、Metropolis 矩阵 [39] 和惰性 Metropolis 矩阵 [29] 等. 这里展示文献 [29] 中所引入的惰性 Metropolis 矩阵, 其具体定义如下:

$$w_{ij} = \begin{cases} 1/(2 \times \max\{\deg(i), \deg(j)\}), & j \in \mathcal{N}_i, \\ 1 - \sum_{k \in \mathcal{N}_i} w_{ik}, & i = j, \\ 0, & \text{其他.} \end{cases} \quad (2.1)$$

拓展到去中心化在线平均问题, 一个自然的想法是通过在线增量求和来实现, 即

$$\mathbf{z}^{(k)} = \mathbf{W}(\mathbf{z}^{(k-1)} + \mathbf{y}^{(k)}). \quad (2.2)$$

称这个基本的方法为 DOVE. 在第 k 轮中, 节点 i 收集一个新样本 $y_i^{(k)}$, 并将其添加到本地的 $z_i^{(k-1)}$ 中. 然后, 每个节点 i 通过与邻居节点通信, 根据权重邻接矩阵 $\mathbf{W}$ 将其本地信息 $z_i^{(k-1)} + y_i^{(k)}$ 更新为本地估计 $z_i^{(k)} = \sum_{j \in \mathcal{N}_i} w_{ij}(z_j^{(k-1)} + y_j^{(k)})$. 由于权重邻接矩阵 $\mathbf{W}$ 具有双随机的性质, 所以有以下关于平均估计值的递归关系式:

$$\bar{\mathbf{z}}^{(k)} = \frac{1}{n} \mathbf{1}^T \mathbf{W}(\mathbf{z}^{(k-1)} + \mathbf{y}^{(k)}) = \bar{\mathbf{z}}^{(k-1)} + \bar{\mathbf{y}}^{(k)}. \quad (2.3)$$

通过使用初始估计 $\mathbf{z}^{(0)} = \mathbf{0}$ 的递归关系 (2.3), 可以推导出 $\bar{\mathbf{z}}^{(k)}/k$ 等于全局平均值 $\bar{\mathbf{y}}_{[k]}$. 因此, 对于 DOVE 方法, 逼近全局平均值 $\bar{\mathbf{y}}_{[k]}$ 等价于实现所有局部估计 $z_i^{(k)}/k$ 的共识. 附录将给出 DOVE 方法的详细算法和理论.

值得注意的是, 去中心化的在线平均估计 $\mathbf{z}^{(k)}/k$ 与全局平均向量 $\mathbf{1}\bar{\mathbf{y}}_{[k]}$ 之间的差异可以通过以下不等式进行控制:

$$\|\mathbf{z}^{(k)}/k - \mathbf{1}\bar{\mathbf{y}}_{[k]}\|_2 \leq \underbrace{\|\mathbf{z}^{(k)}/k - \mathbf{1}\bar{\mathbf{z}}^{(k)}/k\|_2}_{\text{共识误差}} + \underbrace{\|\mathbf{1}\bar{\mathbf{z}}^{(k)}/k - \mathbf{1}\bar{\mathbf{y}}_{[k]}\|_2}_{\text{逼近误差}}.$$

根据平均递归关系 (2.3), 可以得出 DOVE 的逼近误差为零. 至于共识误差, 以下的命题 2.1 表明它与网络拓扑结构及权重邻接矩阵 $\mathbf{W}$ 的谱间隙 $1/(1 - \rho_2)$ 有关 (参见文献 [12, 29]):

命题 2.1 假设图 $\mathcal{G}$ 是连通的. 对于 DOVE 算法产生的估计 $\mathbf{z}^{(k)}/k$, 有

$$\left\| \frac{1}{k} \mathbf{z}^{(k)} - \mathbf{1}\bar{\mathbf{y}}_{[k]} \right\|_2 \leq \frac{\rho_2}{(1 - \rho_2)k} \max_{1 \leq t \leq k} \|\mathbf{y}^{(t)} - \mathbf{1}\bar{\mathbf{y}}^{(t)}\|_2, \quad \text{其中 } \rho_2 < 1.$$

命题 2.1 的具体的证明见附录. 文献 [29, 30] 已经指出, 在一些特殊的网络结构, 如路径网络和双星网络中, ρ_2 可以很接近 1. 这导致了较大的谱间隙 $1/(1 - \rho_2)$, 从而降低了共识误差的收敛速度. 本文的主要目标是改进共识误差, 并在保证连接较为稀疏 (即 ρ_2 接近 1) 的网络下获得更理想的收敛速度.

2.2 加速去中心化在线平均

关于基本的 DOVE 方法的改进主要有两个方面. 命题 2.1 指出,

$$\left\| \frac{1}{k} \mathbf{z}^{(k)} - \mathbf{1} \bar{y}_{[k]} \right\|_2 = O\left(\frac{1}{(1-\rho_2)k} \max_{1 \leq t \leq k} \|\mathbf{y}^{(t)} - \mathbf{1} \bar{y}^{(t)}\|_2\right), \quad (2.4)$$

其中 $\|\mathbf{y}^{(t)} - \mathbf{1} \bar{y}^{(t)}\|_2$ 的上界通常具有 $O(\sqrt{n})$ 的阶, 无法进一步降低. DOVE 算法关键的问题在于只利用了当前轮收集到的样本来更新估计值. 直观地想, 仅在第 k 轮收集到的单个样本 $\mathbf{y}^{(k)}$ 并不足以完整地描述样本平均向量 $\mathbf{1} \bar{y}^{(k)}$. 另一个问题在于, 在第 k 轮时, 局部分量 $y_i^{(k)}$ 在不同节点之间可能差异较大, 从而与平均值 $\bar{y}^{(k)}$ 也相差较大. 这种现象导致了共识误差中的较大偏差, 并且降低了 DOVE 的收敛速度. 为了解决上述的两个问题, 本文提出一种新颖的去中心化在线平均算法, 以实现与 DOVE 相比的加速效果.

为了加快算法的速度, 本文引入了样本滑动平均的概念, 即在第 k 轮使用窗宽 l 的滑动平均 $\sum_{t=k-l+1}^k \mathbf{y}^{(t)}/l$, 并将其简写为 l -MA. 引入一个与窗宽 l 相关联的辅助滑动平均向量 $\mathbf{g}^{(k)}(l) = (g_1^{(k)}(l), \dots, g_n^{(k)}(l))^T$ 来提出一种新颖的分布式在线平均算法, 其中 $g_i^{(k)}(l)$ 表示用于逼近节点 i 中收集的样本的 l -MA 的局部估计. 定义相应的节点平均值为 $\bar{g}^{(k)}(l) = \sum_{i=1}^n g_i^{(k)}(l)/n$. 在这里, 我们强调辅助向量 $\mathbf{g}^{(k)}(l)$ 取决于窗宽 l , 但为了方便起见, 在本文的其余部分中将省略 (l) , 直接写为 $\mathbf{g}^{(k)}$.

用向量 $\mathbf{g}^{(k)}$ 代替原本增量求和方法中的向量 $\mathbf{y}^{(k)}$, 按照如下的方式更新估计向量 $\mathbf{z}^{(k)}$:

$$\mathbf{z}^{(k)} = \mathbf{W}(\mathbf{z}^{(k-1)} + \mathbf{g}^{(k)}).$$

为了减少共识误差, 我们希望 $\|\mathbf{g}^{(k)} - \mathbf{1} \bar{g}^{(k)}\|_2$ 比 $\|\mathbf{y}^{(k)} - \mathbf{1} \bar{y}^{(k)}\|_2$ 小. 首先对迭代轮次大于窗宽, 即 $k > l$ 的情形进行讨论. 而 $k \leq l$ 的情形会在后文通过“校准”的方式进行讨论. 为了实现对 DOVE 方法的加速, 受梯度追踪算法^[32]思想的启发, 用以下方式来更新 $\mathbf{g}^{(k)}$ 这个新的“增量向量”:

$$\mathbf{g}^{(k)} = \mathbf{W} \mathbf{g}^{(k-1)} + \frac{1}{l} \sum_{t=k-l+1}^k \mathbf{y}^{(t)} - \frac{1}{l} \sum_{t=k-l}^{k-1} \mathbf{y}^{(t)} = \mathbf{W} \mathbf{g}^{(k-1)} + \frac{1}{l} \mathbf{y}^{(k)} - \frac{1}{l} \mathbf{y}^{(k-l)}. \quad (2.5)$$

换言之, 可以将 l -MA 中的 $\sum_{t=k-l+1}^k \mathbf{y}^{(t)}/l$ 分解为求和 $\sum_{t=1}^k (\mathbf{y}^{(t)} - \mathbf{y}^{(t-l)})/l$, 其中定义初始伪样本 $\mathbf{y}^{-(l-1)} = \dots = \mathbf{y}^{(0)} = \mathbf{0}$. 差值 $(\mathbf{y}^{(k)} - \mathbf{y}^{(k-l)})/l$, $k = 1, \dots$ 可以被视为一个样本差序列, 用于构建关于 $\mathbf{g}^{(k)}$ 的去中心化在线平均递归公式. 辅助向量 $\mathbf{g}^{(k)}$ 的作用是“追踪”滑动平均值, 即平均估计 $\bar{g}^{(k)} = \sum_{t=k-l+1}^k \bar{y}^{(t)}/l$ 成立. (2.5) 中的因子 $1/l$ 加速了误差 $\|\mathbf{g}^{(k)} - \mathbf{1} \bar{g}^{(k)}\|_2$ 和共识误差的速度.

注意到在 DOVE 方法中, 逼近误差 $\|\mathbf{1} \bar{z}^{(k)}/k - \mathbf{1} \bar{y}_{[k]}\|_2$ 正好等于 0. 但因为在我们的 A-DOVE 算法中引入了 l 项的滑动平均, 逼近误差会在第 $k-l+2$ 步开始出现. 具体而言, 此时的逼近误差可以表示为

$$\frac{1}{k} \bar{z}^{(k)} - \bar{y}_{[k]} = \frac{1}{kl} \sum_{t=k-l+2}^k (k-t+1-l) \bar{y}^{(t)}.$$

直观来讲, 可以定义伪样本 $\mathbf{y}^{(i)} = \mathbf{0}$, 其中 $i \leq 0$, 从而用与 (2.5) 中相同的方式对 $k \leq l$ 的情形定义向量 $\mathbf{g}^{(k)}$. 但是这样会使得迭代初期的 l 滑动平均与总体平均值相差较远, 从而不能得到较好的在线平均效果. 为了解决这个问题, 引入一个“校准”的步骤. 在 A-DOVE 算法中, 采用前端校准, 通过用 $\sum_{t=1}^k \mathbf{y}^{(t)}/k - \sum_{t=1}^{k-1} \mathbf{y}^{(t)}/(k-1)$ 替换原本 (2.5) 中的 $\sum_{t=k-l+1}^k \mathbf{y}^{(t)}/l - \sum_{t=k-l}^{k-1} \mathbf{y}^{(t)}/l$ 来进行 $\mathbf{g}^{(k)}$

的递归. 经过校准后, 在第 $k, k > l$ 轮中, 近似误差可以表示如下形式:

$$\sum_{i=1}^l \left(\frac{1}{i} - \frac{1}{l} \right) \sum_{t=1}^i (\bar{y}^{(t)} - \bar{y}^{(k-i+1)}),$$

并且在初始的 l 步中改进了收敛速度.

以上步骤的更新可以通过将 $\mathbf{z}^{(k)}$ 和 $\mathbf{g}^{(k)}$ 拼接成一个矩阵来进行紧凑的矩阵计算. 然后能够发现, 更新实际上可以在相邻节点之间的一轮通信中实现. 我们将加速去中心化在线平均算子 (accelerated decentralized online average operator, A-DOVO) 定义为以下形式: 当 $k \leq l$ 时,

$$\begin{aligned} & \text{A-DOVO}(\mathbf{g}^{(k-1)}, \mathbf{z}^{(k-2)}, \mathbf{y}, k, l) \\ &= \begin{bmatrix} \mathbf{W} & \mathbf{O}_{n \times n} \\ \mathbf{W} & \mathbf{W} \end{bmatrix} \begin{bmatrix} \mathbf{g}^{(k-1)} \\ \mathbf{z}^{(k-2)} \end{bmatrix} + \begin{bmatrix} \frac{1}{k} \sum_{t=1}^k \mathbf{y}^{(t)} - \frac{1}{k-1} \sum_{t=1}^{k-1} \mathbf{y}^{(t)} \\ \mathbf{0}_n \end{bmatrix}; \end{aligned}$$

当 $k > l$ 时,

$$\begin{aligned} & \text{A-DOVO}(\mathbf{g}^{(k-1)}, \mathbf{z}^{(k-2)}, \mathbf{y}, k, l) \\ &= \begin{bmatrix} \mathbf{W} & \mathbf{O}_{n \times n} \\ \mathbf{W} & \mathbf{W} \end{bmatrix} \begin{bmatrix} \mathbf{g}^{(k-1)} \\ \mathbf{z}^{(k-2)} \end{bmatrix} + \begin{bmatrix} \frac{1}{l} \sum_{t=k-l+1}^k \mathbf{y}^{(t)} - \frac{1}{l} \sum_{t=k-l}^{k-1} \mathbf{y}^{(t)} \\ \mathbf{0}_n \end{bmatrix}, \end{aligned}$$

其中 $\mathbf{O}_{n \times n}$ 是一个 $n \times n$ 的全零矩阵, $\mathbf{0}_n$ 是 n 维的全零向量.

在我们的算法中, 我们根据局部结构信息 w_{ij} (其中 $j \in \mathcal{N}_i, 1 \leq i \leq n$), 为每个节点 i 构造加权邻接矩阵 $\mathbf{W}$. 我们的 A-DOVE 算法的流程在算法 1 中呈现如下.

算法 1 A-DOVE 算法

输入: 初始向量 $\mathbf{g}^{(0)} = \mathbf{0}, \mathbf{z}^{(-1)} = \mathbf{0}$, 滑动窗宽 l .

- 1: 对于 $k = 1, 2, \dots$ 执行
 - 2: $\begin{bmatrix} \mathbf{g}^{(k)} \\ \mathbf{z}^{(k-1)} \end{bmatrix} = \text{A-DOVO}(\mathbf{g}^{(k-1)}, \mathbf{z}^{(k-2)}, \mathbf{y}, k, l).$
 - 3: 结束循环
-

值得注意的是, 在应用 A-DOVE 算法时, 每一个节点最多只需要存储 l 个历史数据. 当迭代轮数 $k \leq l$ 时, 第 i 个节点需要存储的数据为 $\{y_i^{(t)}, t \leq k\}$. 而当 $k \geq l$ 时, 需要存储的数据为 $\{y_i^{(t)}, k-l+1 \leq t \leq k\}$.

注 2.1 A-DOVE 算法利用了滑动平均的思想, 在每个时间点需要存储 l 个历史数据. 则一个非常直观的想法便是, 在每一个时间点收集 l 个新样本, 求出这 l 个新样本的平均值, 再利用 DOVE 算法. 为了方便, 记这种估计为 l -DOVE.

给定轮次总数 k , 假设存在整数 k' 使得 $k = k'l$. 针对 l -DOVE 算法, 节点 i 在时间点 t' 收集到的样本记为 $\tilde{y}_i^{(t')} = \sum_{j=(t'-1)l+1}^{t'l} y_i^{(j)} / l$. 记节点间平均值为 $\bar{\tilde{y}}^{(t')} = \sum_{i=1}^n \tilde{y}_i^{(t')} / n$, 要估计的目标 (全局平均值) 记为 $\bar{y}_{k'}$. 容易验证:

$$\bar{\tilde{y}}_{k'} = \frac{1}{k'} \sum_{t'=1}^{k'} \bar{\tilde{y}}^{(t')} = \frac{1}{k'n} \sum_{t'=1}^{k'} \sum_{i=1}^n \tilde{y}_i^{(t')} = \frac{1}{lk'n} \sum_{t'=1}^{k'} \sum_{i=1}^n \sum_{j=(t'-1)l+1}^{t'l} y_i^{(j)} = \frac{1}{kn} \sum_{i=1}^n \sum_{t=1}^k y_i^{(t)} = \bar{y}_{[k]}.$$

与之前的方法类似, 记 l -DOVE 算法中每个节点上在第 k' 个轮次得到的估计值为 $\tilde{z}_i^{(k')}/k'$ 并定义聚合向量 $\tilde{\mathbf{z}}^{(k')} = (\tilde{z}_1^{(k')}, \dots, \tilde{z}_m^{(k')})^\top$. 根据 DOVE 的理论结果 (命题 2.1) 可以得到 l -DOVE 的收敛速度为

$$\left\| \frac{1}{k'} \tilde{\mathbf{z}}^{(k')} - \mathbf{1} \bar{y}_{[k]} \right\|_2 = \left\| \frac{1}{k'} \tilde{\mathbf{z}}^{(k')} - \mathbf{1} \bar{y}_{k'} \right\|_2 = O\left(\frac{1}{(1-\rho_2)k'} \max_{1 \leq t' \leq k'} \|\tilde{\mathbf{y}}^{(t')} - \mathbf{1} \bar{y}^{(t')}\|_2\right). \quad (2.6)$$

可以看出, l -DOVE 算法的收敛速度与 DOVE 算法的收敛速度类似, 并没有显著改善. 这主要是由于 l -DOVE 算法和 A-DOVE 不同, 本质上并没有重复利用历史数据中蕴含的信息. 我们在数值模拟部分也通过一个实验来验证这一结论.

2.3 加速去中心化在线最小二乘

去中心化在线平均方法可以与许多其他的去中心化在线学习方法结合使用. 这里将重点放在每个节点 i 中的线性回归模型 $y_i = \mathbf{x}_i^\top \boldsymbol{\theta}^* + \varepsilon_i$ 上, 其中 $y_i \in \mathbb{R}$ 是响应变量, $\mathbf{x}_i \in \mathbb{R}^p$ 是协变量, 其均值为 $\mathbb{E}(\mathbf{x}_i) = \boldsymbol{\mu}_{\mathbf{x},i} = (\mu_{\mathbf{x},1i}, \dots, \mu_{\mathbf{x},pi}) \in \mathbb{R}^p$. 在第 k 个轮次中, 第 i 个节点收集独立样本 $(y_i^{(k)}, \mathbf{x}_i^{(k)})$. 本小节的目标是介绍一种去中心化的在线最小二乘估计 (least square estimation, LSE), 该估计可以适应去中心化在线的应用场景, 也能很好地近似经典的 LSE.

为了描述方便, 在第 k 个迭代轮次定义 $\gamma_i^{(k)} = \mathbf{x}_i^{(k)} y_i^{(k)}$ 并且构建相关的聚合矩阵 $\boldsymbol{\Gamma}^{(k)} = (\gamma_1^{(k)}, \dots, \gamma_n^{(k)})^\top \in \mathbb{R}^{n \times p}$. 记 $\boldsymbol{\Phi}_i^{(k)} = \mathbf{x}_i^{(k)} \mathbf{x}_i^{(k)\top}$ 并且令 $\boldsymbol{\Phi}_{i,m}^{(k)}$ 表示 $\boldsymbol{\Phi}_i^{(k)}$ 的第 m 列. 我们进一步分别定义全局平均向量 $\bar{\gamma}_{[k]} = \sum_{i=1}^n \sum_{t=1}^k \gamma_i^{(t)} / (nk)$ 和全局平均矩阵 $\bar{\boldsymbol{\Phi}}_{[k]} = \sum_{i=1}^n \sum_{t=1}^k \boldsymbol{\Phi}_i^{(t)} / (nk)$. 当 k 轮的数据被一次性全部收集时, 经典的 LSE 定义为

$$\hat{\boldsymbol{\theta}}_{LS}(k) = \bar{\boldsymbol{\Phi}}_{[k]}^{-1} \bar{\gamma}_{[k]}.$$

注意到 $\hat{\boldsymbol{\theta}}_{LS}(k)$ 的公式由全局平均矩阵 $\bar{\boldsymbol{\Phi}}_{[k]}$ 和全局平均向量 $\bar{\gamma}_{[k]}$ 组成. 为了在去中心化在线框架下近似 $\hat{\boldsymbol{\theta}}_{LS}(k)$, 这个观察结果启发了我们相应地用 A-DOVE 替换这两个多变量平均组件. 因此, 我们开发了一种 A-DOSE 方法. 值得注意的是, 当 $p = 1$ 且所有协变量 $\mathbf{x}_i = 1$ 时, 最小二乘回归简化为求平均.

接下来, 定义聚合矩阵 $\boldsymbol{\Xi}^{(k)} = (\boldsymbol{\xi}_1^{(k)}, \dots, \boldsymbol{\xi}_n^{(k)})^\top \in \mathbb{R}^{n \times p}$, 其中 $\boldsymbol{\xi}_i^{(k)}/k$ 表示节点 i 中的本地估计, 用于近似全局平均向量 $\bar{\gamma}_{[k]}$. 接下来, 将之前定义的加速平均算子 A-DOVO 拓展到一个矩阵版本, 并以矩阵形式生成聚合估计值 $\boldsymbol{\Xi}^{(k)}$. 也就是说, 直接用矩阵 $\boldsymbol{\Gamma}^{(t)}$ 替换 A-DOVO 中的输入向量 $\mathbf{y}^{(t)}$, 从而得到以下形式的 $\boldsymbol{\Xi}^{(k)}$ 和相应的辅助矩阵 $\mathbf{G}^{(k)}$:

$$\begin{bmatrix} \mathbf{G}^{(k)} \\ \boldsymbol{\Xi}^{(k-1)} \end{bmatrix} = \text{A-DOVO}(\mathbf{G}^{(k-1)}, \boldsymbol{\Xi}^{(k-2)}, \boldsymbol{\Gamma}, k, l).$$

注意, 在拓展后的 A-DOVO 中, 用全零矩阵 $\mathbf{O}_{n \times p}$ 代替为原本的全零向量 $\mathbf{0}_n$. 类似地, 为了估计 $\bar{\boldsymbol{\Phi}}_{[k]}$, 进一步定义矩阵 $\check{\boldsymbol{\Phi}}_m^{(k)} = (\boldsymbol{\Phi}_{1,m}^{(k)}, \dots, \boldsymbol{\Phi}_{n,m}^{(k)})^\top \in \mathbb{R}^{n \times p}$, 其中 $1 \leq m \leq p$. 按相同的方式, 在线平均算法中的向量 $\mathbf{y}^{(k)}$ 可以在第 k 步被矩阵 $\check{\boldsymbol{\Phi}}_m^{(k)}$ 替换. 因此, 可以通过运行 A-DOVO 来获得更新的 $\check{\boldsymbol{\Sigma}}_m^{(k-1)}$ 和相应的辅助矩阵 $\check{\boldsymbol{\Delta}}_m^{(k)}$, 具体过程如下所示:

$$\begin{bmatrix} \check{\boldsymbol{\Delta}}_m^{(k)} \\ \check{\boldsymbol{\Sigma}}_m^{(k-1)} \end{bmatrix} = \text{A-DOVO}(\check{\boldsymbol{\Delta}}_m^{(k-1)}, \check{\boldsymbol{\Sigma}}_m^{(k-2)}, \check{\boldsymbol{\Phi}}_m, k, l).$$

这里记 $\check{\Sigma}_m^{(k)}$ 的第 i 行为 $\sigma_{mi}^{(k)}$, 这也是第 i 个机器上存储的向量. 然后构造 $\Sigma_i^{(k)}/k = (\sigma_{1i}^{(k)T}, \dots, \sigma_{pi}^{(k)T})/k$ 作为 $\Phi_{[k]}$ 在节点 i 上的估计. 根据经验的代入法, 中心化的最小二乘估计 $\hat{\theta}_{LS}(k)$ 可以在节点 i 上被近似为 $\theta_i^{(k)} = (\Sigma_i^{(k)})^{-1} \xi_i^{(k)}$, 其中 $i = 1, 2, \dots, n$.

A-DOSE 的具体过程在算法 2 中呈现. 与上一小节中的 A-DOVE 类似, A-DOSE 的更新步骤可以被写为紧凑的矩阵形式. 此外, 为了得到关于真实参数 θ^* 的可收敛估计 $\theta_i^{(k)}$ 并且建立相关的中心极限定理, 我们将会证明所有节点上的偏差 $\|\theta_i^{(k)} - \hat{\theta}_{LS}(k)\|_2$ 具有阶数 $o_P(1/\sqrt{nk})$. 详细的理论在接下来的第 3 节中呈现.

算法 2 A-DOSE 算法

输入: 初始值 $G^{(0)} = O_{n \times p}$, $\Xi^{(-1)} = O_{n \times p}$, $\check{\Delta}_m^{(0)} = O_{n \times p}$, $\check{\Sigma}_m^{(-1)} = O_{n \times p}$, $1 \leq m \leq p$; 滑动窗宽 l .

- 1: 对于 $k = 1, 2, \dots$ 执行
 - 2: $[G_{(k-1)}^{(k)}] = \text{A-DOVO}(G^{(k-1)}, \Xi^{(k-2)}, \Gamma, k, l)$.
 - 3: 对于 $m = 1, \dots, p$ 执行
 - 4: $[\check{\Sigma}_{(k-1)}^{(k)}] = \text{A-DOVO}(\check{\Delta}_m^{(k-1)}, \check{\Sigma}_m^{(k-2)}, \check{\Phi}_m, k, l)$.
 - 5: 结束循环
 - 6: 对于 $i = 1, \dots, n$ 执行
 - 7: $\theta_i^{(k-1)} = (\Sigma_i^{(k-1)})^{-1} \xi_i^{(k-1)}$.
 - 8: 结束循环
 - 9: 结束循环
-

注 2.2 在我们的算法中, 假设同一个时刻, 每个节点都顺利接收到了数据. 但在实际情形中, 可能会存在某个节点接收不到数据的故障, 此时可以对算法进行一定的调整, 以顺利完成求在线平均和在线最小二乘的任务.

首先考虑某个节点一直没有新数据到来的情形. 以求在线平均为例, 假设某一个节点 i 出现了故障, 一直无法接收到新的数据, 在已经无法调整网络结构的情形下, 可以选择让节点 i 向其他邻居传递伪数据 $\tilde{y}_i^{(k)} = 0$, $k \geq 1$. 此时整个算法仍然正常运行, 最后再将每个正常节点上得到的结果乘上 $n/(n-1)$, 便相当于完成了在 $n-1$ 个正常节点上的在线平均任务. 对在线最小二乘的调整思路也是类似的.

其次考虑某一个节点 i 只是在某个时刻 k' 没有接收到数据的情形. 此时对于在线平均和在线最小二乘问题有不同的调整思路. 针对在线平均问题, 由于我们的目的本身就是估计所有机器上收集数据的平均值, 所以只需要让节点 i 向邻居节点传输样本 $y_i^{(k')} = 0$ 即可, 不影响算法运行. 而对于在线最小二乘问题, 我们的目的是估计线性模型中的参数 θ_* , 直接传输零数据 (即 $x_i^{(k')} = 0$, $y_i^{(k')} = 0$) 可能会对最终的估计造成偏差. 因此可以考虑让节点 i 使用上一个时刻接收到的数据进行替代, 即令

$$x_i^{(k')} = x_i^{(k'-1)}, \quad y_i^{(k')} = y_i^{(k'-1)},$$

然后正常进行算法流程.

3 理论结果

本节为上文讨论的去中心化在线算法提供了一系列理论保障. 若不加额外说明, 我们默认节点数 n 和维数 p 是固定的, 而轮数 k 可以趋于无穷.

3.1 加速去中心化在线平均

3.1.1 A-DOVE 的算法收敛性

首先, 对网络 $\mathcal{G}$ 的结构和权重矩阵 $\mathbf{W}$ 的选取作出一定的假设.

假设 3.1 网络 $\mathcal{G}$ 是连通的. 选取的权重矩阵 $\mathbf{W}$ 满足双随机性质: $\mathbf{1}^T \mathbf{W} = \mathbf{1}^T$ 和 $\mathbf{W} \mathbf{1} = \mathbf{1}$, 并且所有的对角元素都大于 0.

假设 3.1 在去中心化优化的相关工作中非常常用. 网络的连通性保证了所有节点上的信息可以在邻居节点间进行交互. 根据文献 [25], 在假设 3.1 下可以得到矩阵 $\mathbf{W}$ 的第二特征值 $\rho_2 < 1$. 不同的权重矩阵会产生不同的第二大特征值 ρ_2 , 而从之后的理论结果中可以看到, ρ_2 的大小会影响算法的收敛速度. 下面的定理描述了 A-DOVE 算法的收敛速度:

定理 3.1 在假设 3.1 满足的前提下, 对于算法 1 中在第 $k > l$ 个轮次得到的估计 $\mathbf{z}^{(k)}/k$, 有

$$\left\| \frac{1}{k} \mathbf{z}^{(k)} - \mathbf{1} \bar{y}_{[k]} \right\|_2 \leq \frac{\rho_2}{(1 - \rho_2)^2 k l} \max_{l+1 \leq t \leq k} \|\mathbf{y}^{(t)} - \mathbf{y}^{(t-l)}\|_2 + \frac{4\rho_2^{k-l+1}}{(1 - \rho_2)^2 k} \max_{1 \leq t \leq l} \|\mathbf{y}^{(t)} - \mathbf{1} \bar{y}^{(t)}\|_2 + \frac{\sqrt{nl}}{k} \max_{1 \leq i \leq l} \max_{1 \leq t \leq l} |\bar{y}^{(t)} - \bar{y}^{(k-i+1)}|,$$

这里 l 表示滑动窗宽.

注 3.1 假如所有的样本都是一致有界的, 则上述定理中提出的上界可以简化为

$$\frac{\sqrt{n}}{(1 - \rho_2)^2 k l} + \frac{\rho_2^{k-l+1} \sqrt{n}}{(1 - \rho_2)^2 k} + \frac{\sqrt{nl}}{k} \max_{1 \leq i \leq l} \max_{1 \leq t \leq l} |\bar{y}^{(t)} - \bar{y}^{(k-i+1)}|.$$

定理 3.1 中的上界由三个部分组成. 第一项与共识误差有关, 如果令窗宽 $l \gg 1/(1 - \rho_2)$, 则我们的结果相比 (2.4) 确实产生了加速. 第二项可以解释为因为使用滑动平均产生的偏差余项. 注意到这里有 $\rho_2 < 1$, 所以 ρ_2^{k-l+1} 随着 k 的增大会按照指数的速度快速收敛到零. 于是在 k 足够大时, 第二项可以被视为一个可忽略的余项. 由于定理 3.1 中的第二项与 $\|\mathbf{y}^{(t)} - \mathbf{1} \bar{y}^{(t)}\|_2$ 有关, 因此可以被视为是数据异质对收敛速度的影响. 注意到与 (2.4) 相比, 该项能够以指数的速度更快地收敛到 0. 这从一定程度上说明了, 当数据异质性较强 (即 $\|\mathbf{y}^{(t)} - \mathbf{1} \bar{y}^{(t)}\|_2$ 较大) 时, A-DOVE 算法能够改善收敛速度. 第三项是从逼近误差的部分推导出来的, 与谱间隙 $1/(1 - \rho_2)$ 无关. 假设样本序列 $\{y_i^{(t)}, 1 \leq i \leq n\}$, 其中 $t \geq 1$ 是从一个平稳的随机模型中生成的, 即 $\{y_i^{(t)}, 1 \leq i \leq n\}$ 的联合分布在时间 t 上保持不变. 则根据大数定律的收敛速度, $\max_{1 \leq i \leq l} \max_{1 \leq t \leq l} \|\bar{y}^{(t)} - \bar{y}^{(k-i+1)}\|_2$ 通常可以达到 $O(1/\sqrt{n})$ 的阶数. 因此, 如果令 $l \ll \sqrt{n}/(1 - \rho_2)$, 则第三项可以远小于 (2.4) 中的速率. 我们将在后续的内容中严格证明这个观察结果. 虽然我们在理论结果中为滑动窗口设定了上下界, 但在后续的数值模拟中, 我们发现实际上对 l 的选取是非常灵活的. 简单地选取 $l = 2$ 已经可以使算法得到显著的提升.

3.1.2 去中心化在线平均估计的统计性质

在每一个节点 i 上考虑一个去中心化随机均值模型 $y_i^{(t)} = \mu_i + \varepsilon_i^{(t)}$, 其中噪声 $\varepsilon_i^{(t)}, 1 \leq i \leq n$ 是独立的. 将均值向量记为 $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)$, 并假设存在某个常数 $C > 0$, 使得 $\max_{1 \leq i \leq n} |\mu_i - \frac{1}{n} \sum_{j=1}^n \mu_j| \leq C$. 在这个模型下, 本文证明了通过 A-DOVE 算法获得的估计的统计相合性.

首先, 通过定理 3.1 的证明可以看出, 如果用范数 $\sqrt{\mathbb{E}(\|\cdot\|_2^2)} =: \|\cdot\|_{\mathbb{E}}$ 替换 Euclid 范数 $\|\cdot\|_2$, 定理 3.1 中的结果仍然成立, 从而可以得到如下推论:

推论 3.1 在假设 3.1 成立的前提下, 令 $\mathbb{E}(\varepsilon_i^{(t)}) = 0$ 以及 $\mathbb{E}(\varepsilon_i^{(t)})^2 \leq M$ 对于某个常数 M 及所有 i 和 t 成立, 则对于算法 1 在第 $k > l$ 个轮次给出的 A-DOVE 估计值 $\mathbf{z}^{(k)}/k$, 有

$$\left\| \frac{1}{k} \mathbf{z}^{(k)} - \mathbf{1}\bar{y}_{[k]} \right\|_{\mathbb{E}} = O\left(\frac{\sqrt{n}}{(1-\rho_2)^2 kl} + \frac{\rho_2^{k-l+1}\sqrt{n}}{(1-\rho_2)^2 k} + \frac{l}{k} \right),$$

这里 l 表示滑动窗宽.

这个推论表明, 在一些温和的矩条件下, A-DOVE 估计 $\mathbf{z}^{(k)}/k$ 在期望意义下接近于全局平均向量 $\mathbf{1}\bar{y}_{[k]}$. 正如在定理 3.1 之后讨论的那样, 定理 3.1 中的第三项 $\sqrt{nl} \max_{1 \leq i \leq l} \max_{1 \leq t \leq l} \|\bar{y}^{(t)} - \bar{y}^{(k-i+1)}\|_2/k$ 在期望意义上可以进一步简化为 l/k . 只要选取 $l \ll \sqrt{n}/(1-\rho_2)$ 并且轮数 k 足够大以满足 $l\rho_2^{k-l+1} \ll 1$, 第一项就会占主导地位. 当第一项成为主项时, $1/l$ 在加速因子中起着重要的作用.

3.2 加速去中心化在线最小二乘

当研究最小二乘回归问题时, 我们在假设 3.1 的基础上需要额外引入一些假设:

假设 3.2 对于所有节点 $1 \leq i \leq n$ 和所有轮次 $k \geq 1$, 所有样本 $(\mathbf{x}_i^{(k)}, y_i^{(k)})$ 都是独立的. 每个节点 i 中的样本是同分布的.

假设 3.3 假设所有节点上的协变量 $\mathbf{x}_i$ 具有相同的协方差矩阵 $\mathbb{E}(\mathbf{x}_i \mathbf{x}_i^T) = \Sigma$, 并且存在一个正常数 $c \in (0, 1)$, 使得总体协方差矩阵的最小特征值满足 $\lambda_{\min}(\Sigma) \geq c > 0$, 最大特征值满足 $\lambda_{\max}(\Sigma) \leq c^{-1} < \infty$.

假设 3.4 对于所有的协变量 $\mathbf{x}_i$, $1 \leq i \leq n$, 存在一个常数 $C > 0$, 使得对于所有的分量 $1 \leq m \leq p$ 和所有的节点 $1 \leq i \leq n$, $\mathbb{E}(x_{i,m}^4) \leq C$ 成立.

假设 3.2 是一个说明样本独立性的标准假设. 假设 3.3 确保了协方差矩阵 Σ 的正则性. 假设 3.4 是关于协变量 $\mathbf{x}_i$, $1 \leq i \leq n$ 的温和的四阶矩条件. 在上述假设下, 下面的定理讨论了 A-DOSE 对经典 LSE 的逼近误差:

定理 3.2 在假设 3.1–3.4 成立的前提下, 令 $\mathbb{E}(\varepsilon_i^{(t)}) = 0$ 以及 $\mathbb{E}(\varepsilon_i^{(t)})^4 \leq M$ 对于某个常数 M 及所有 i 和 t 都成立. 我们还假设 $\|\boldsymbol{\theta}^*\|_2 \leq r$, 其中 $r > 0$ 是一个正常数. 对于由算法 2 给出的 A-DOSE 估计 $\boldsymbol{\theta}_i^{(k)}$, 当轮数 k 足够大时, 有以下结果:

$$\sqrt{\sum_{i=1}^n \|\boldsymbol{\theta}_i^{(k)} - \hat{\boldsymbol{\theta}}_{LS}(k)\|_2^2} = O_P\left(\frac{\sqrt{np}}{(1-\rho_2)^2 kl} + \frac{\rho_2^{k-l+1}\sqrt{np}}{(1-\rho_2)^2 k} + \frac{pl}{k} \right),$$

其中 l 表示窗宽.

该定理在一些温和的矩条件下, 通过概率界限刻画了 A-DOSE 估计与经典 LSE 之间的差异. 定理 3.2 中的误差上界继承了先前关于去中心化在线平均问题的理论结果中的谱间隙 $1/(1-\rho_2)^2$ 和加速因子 $1/l$. 类似地, 通过以下推论给出 A-DOSE 的统计相合性:

推论 3.2 在假设 3.1–3.4 成立的前提下, 假设存在常数 $r > 0$ 使得 $\|\boldsymbol{\theta}^*\|_2 \leq r$. 对于由算法 2 给出的 A-DOSE 估计 $\boldsymbol{\theta}_i^{(k)}$, 其中 $1 \leq i \leq n$, 当轮数 k 足够大时, 有以下结果:

$$\sqrt{\sum_{i=1}^n \|\boldsymbol{\theta}_i^{(k)} - \boldsymbol{\theta}^*\|_2^2} = O_P\left(\frac{\sqrt{np}}{(1-\rho_2)^2 kl} + \frac{\rho_2^{k-l+1}\sqrt{np}}{(1-\rho_2)^2 k} + \frac{pl}{k} + \sqrt{\frac{p}{k}} \right), \quad (3.1)$$

其中 l 表示窗宽.

这个推论表明我们所有节点的 A-DOSE 估计与真实的参数 θ^* 是一致的. 前三项来自局部估计与经典 LSE 之间的差异, 这在定理 3.2 中也进行了描述. 第四项来自经典 LSE 与真实回归系数之间的误差, 是一个经典的结果.

在一些正则条件下, 经典的最小二乘估计 $\hat{\theta}_{LS}(k)$ 收敛于真实系数 $\theta^* \in \mathbb{R}^p$, 并且满足中心极限定理, 即当 $k \rightarrow \infty$ 时, 有

$$\sqrt{nk}(\hat{\theta}_{LS}(k) - \theta^*) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \Sigma_\mu^{-1} \mathbf{M} \Sigma_\mu^{-1}),$$

其中 $\mu_{x,i} = \mathbb{E}(x_i)$, $\Sigma_\mu = \Sigma + \sum_{i=1}^n \mu_{x,i} \mu_{x,i}^T / n$, $\mathbf{M} = \sum_{i=1}^n \mathbb{E}(x_i x_i^T \varepsilon_i^2) / n$, “ $\xrightarrow{d}$ ” 代表依分布收敛. 在下述的推论中, 同样可以为 A-DOSE 建立类似的渐近性质:

推论 3.3 在假设 3.1–3.4 成立的前提下, 对于每一个节点 i , $1 \leq i \leq n$ 上算法 2 产生的 A-DOSE 估计 $\theta_i^{(k)}$, 在 $k \rightarrow \infty$ 时有以下结果:

$$\sqrt{nk}(\theta_i^{(k)} - \theta^*) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \Sigma_\mu^{-1} \mathbf{M} \Sigma_\mu^{-1}),$$

其中 $\Sigma_\mu = \Sigma + \sum_{i=1}^n \mu_{x,i} \mu_{x,i}^T / n$, $\mathbf{M} = \sum_{i=1}^n \mathbb{E}(x_i x_i^T \varepsilon_i^2) / n$.

这个推论说明, 对于每个节点 i , 当 k 趋向于无穷大时, 局部 A-DOSE 估计 $\theta_i^{(k)}$ 渐近地趋向于一个均值为零、协方差矩阵为 $\Sigma_\mu^{-1} \mathbf{M} \Sigma_\mu^{-1}$ 的多元正态分布随机向量. 据我们所知, 这是在去中心化在线最小二乘回归问题中建立的第一个中心极限定理. 这个结果为去中心化在线估计 $\theta_i^{(k)}$ 的进一步统计推断作出了贡献. 具体而言, 假设可以收集到足够多的样本, 并且节点的个数也足够多. 基于本文的 A-DOSE 算法, 每个节点上都可以得到关于真实参数 θ^* 的一个估计值 $\hat{\theta}_i$. 然后可以利用经典的去中心化平均算法 (如去中心化线性迭代) 得到均值 $\sum_{i=1}^n \hat{\theta}_i / n$, 作为 θ^* 的点估计. 在每个节点上利用本地的样本信息可以得到关于协方差矩阵 $\Sigma_\mu^{-1} \mathbf{M} \Sigma_\mu^{-1}$ 的估计. 然后便可以在每个节点上构造关于 θ^* 的区间估计, 或者进行假设检验.

注 3.2 考虑一个特殊情形, 所有节点 $1 \leq i \leq n$ 上的噪声项满足 $\mathbb{E}(\varepsilon_i^2) = \sigma^2$, 并且与协变量 x_i 独立. 则可以推导出 $\mathbf{M} = \sigma^2 \sum_{i=1}^n \mathbb{E}(x_i x_i^T) / n = \sigma^2 \Sigma_\mu$. 此时推论 3.3 中呈现的中心极限定理可以简化为以下形式:

$$\sqrt{nk}(\theta_i^{(k)} - \theta^*) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \sigma^2 \Sigma_\mu^{-1}).$$

4 数值模拟

本节研究我们提出的加速去中心化在线算法在不同类型网络结构下的数值表现, 通过递增的迭代轮数 $k = 50, 100, \dots, 500$ 来检验算法的收敛速度. 双随机矩阵 $\mathbf{W}$ 被定义为方程 (2.1) 中的惰性 Metropolis 矩阵. 如果没有特别说明, 网络节点的数量选为 $n = 10$.

考虑四种不同的网络结构: Erdős-Rényi 网络、路径网络、集中式网络和双星网络. Erdős-Rényi 网络是一个随机网络, 网络中的任意两个节点以概率 p_c 相连; 路径网络是一棵树, 其中有两个度为 1 的节点, 其他节点的度为 2; 集中式网络包括一个中心节点, 所有其他节点只与中心节点相连; 双星网络由两个星形结构组成, 其中星形结构的中心节点相连, 每个星形结构都是一个包含 5 个节点的树, 其中一个节点的度为 4, 其他 4 个节点的度为 1. 每种网络的详细示例如图 2 所示.

为了通过数值实验验证我们的加速效果, 分别将我们的加速方法与朴素方法 DOVE (用于平均问题) 和去中心化在线最小二乘估计 (decentralized online least square estimation, DOSE) (用于最小二乘

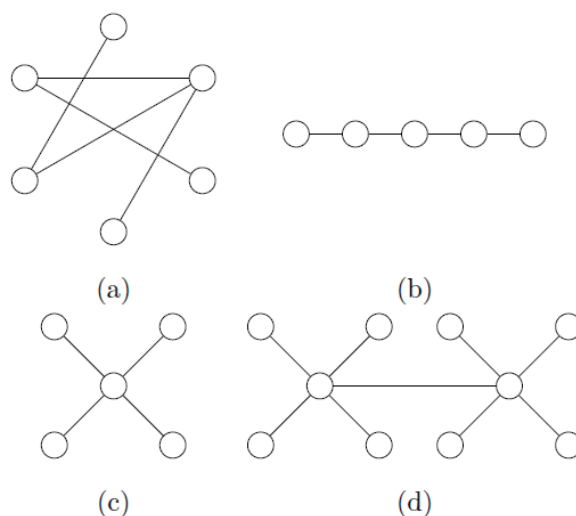


图 2 几种分散网络类型的示例说明. (a) Erdős-Rényi 网络 ($n = 6$); (b) 路径网络 ($n = 5$); (c) 集中式网络 ($n = 5$); (d) 双星网络 ($n = 10$)

问题) 进行比较. 这里的 DOSE 算法是指, 通过插入两个 DOVE 估计来进行基本的去中心化在线最小二乘估计方法. DOSE 算法的构造与 A-DOSE 算法类似, 但没有通过滑动平均加速的步骤. 因为篇幅的限制, 附录将详细描述 DOSE 算法的流程. 此外, 我们还将我们的加速方法与文献 [40] 中提出的去中心化最小二乘方法 (Xie-Zhang-Guo, XZG) 和基于随机梯度的方法进行比较, 包括去中心化随机梯度下降方法 (DSG)、去中心化精确一阶算法 (decentralized exact first-order algorithm, EXTRA) 所对应的随机版本 (参见文献 [37]) 和去中心化随机梯度追踪方法 (DSGT). 值得一提的是, 在数学上, XZG 与 DOSE 其实是等价的. 具体而言, 对于平均问题, XZG 表示 $m = 1$, 且文献 [40] 中的最小二乘方法中 $\varphi_{k,i} \equiv 1$ 的特殊情形也等价于 DOVE. 为了方便比较, 我们在去中心化在线平均问题中选择常数步长 $\alpha = 0.01$, 在分散最小二乘问题中选择 $\alpha = 0.005$ 用于 DSG、EXTRA 和 DSGT. 所有的模拟都是通过 100 次重复实现的.

4.1 去中心化在线平均

在关于平均算法的实验中, 我们使用节点间的平均绝对相对误差 (mean absolute relative error, MARE) 来衡量第 k 个轮次时去中心化平均估计 $z_i^{(k)}/k$ 与全局平均 $\bar{y}_{[k]}$ 之间的差异:

$$\text{MARE}(k) = \frac{1}{n} \sum_{i=1}^n \frac{|z_i^{(k)}/k - \bar{y}_{[k]}|}{|\bar{y}_{[k]}|}.$$

在本小节的所有图表中, 将以 DOVE 产生的 $\text{MARE}(k)$ 为基准, 展示其他方法产生的 $\text{MARE}(k)$ 关于基准的比率.

考虑一种含有异质均值的平均问题, 其中不同节点的本地均值不同, 并且通过均匀分布随机生成. 具体而言, 首先通过 $\mu_i \sim \text{Unif}[0, 10]$ 一次性随机生成均值 μ_i , $1 \leq i \leq n$, 并将其固定为所有重复实验中都相同. 然后, 在第 k 个轮次为第 i 个节点生成样本 $y_i^{(k)} \sim \mathcal{N}(\mu_i, \sigma_i^2)$, 如果没有额外说明, 都设定 $\sigma_i = 1$.

4.1.1 窗宽的影响

第一个实验探究了窗宽 l 对收敛速度的影响. 考虑 4 种不同的网络结构: 连接概率为 $p_c = 0.2$ 的 Erdős-Rényi 网络、路径网络、集中式网络和双星网络. 这 4 种网络的奇异值 ρ_2 都接近于 1, 即都存在连结性不佳的问题. 我们比较了不同窗宽 $l = 1, 2, 3, 5, 10, 15$ 的 A-DOVE 方法对应的 MARE 比率. 从图 3 中可以看出, 当选取 $l = 1$ 时, 我们的算法并不收敛. 因为此时算法相当于计算的是所有样本一步差值之间的平均值, 肯定无法准确地估计样本均值, 这与我们理论分析中认为 l 不能选取太小的结论是一致的. 若选取 $l \geq 2$, 当 k 足够大时, $\text{MARE}(k)$ 比率大致是一个常数, 这说明了我们的算法在 k 足够大时确实是收敛的. 在所有大于 1 的窗宽选择下, $\text{MARE}(k)$ 比率都显著地小于 1, 这反映了我们的方法相较于 DOVE 算法确实具有加速收敛的效果. 同时可以观察到, 不同 l 下 A-DOVE 的性能是相似的, 并且选取 $l = 2$ 时已经取得了足够令人满意的表现. 因此, 后续的模拟中只考虑 $l = 2$ 的情形, 以方便比较.

正如注 2.1 所提到的, 这里也通过一个实验比较了 DOVE, A-DOVE 和 l -DOVE 算法的 $\text{MARE}(k)$

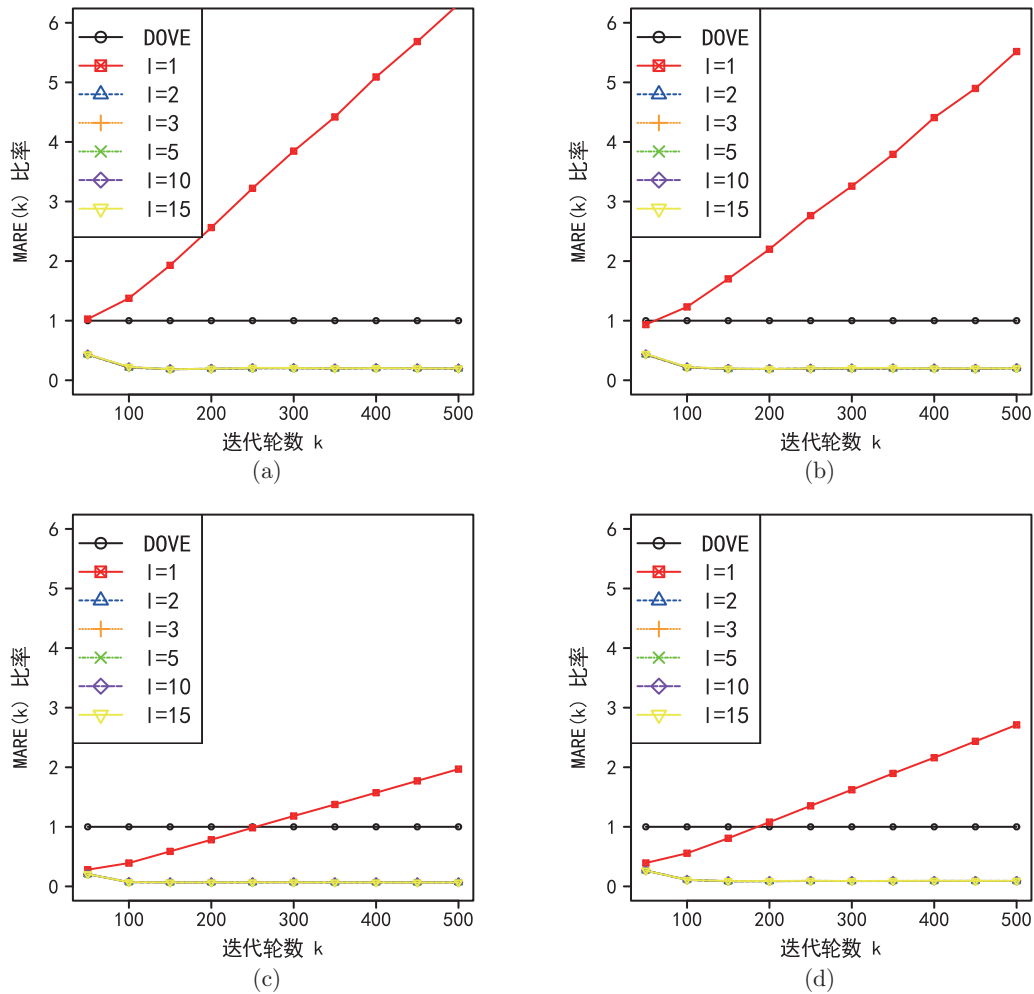


图 3 (网络版彩图) 不同窗宽下加速去中心化在线平均估计的 $\text{MARE}(k)$ 比率随轮数 k 的变化图. (a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) 路径网络; (c) 集中式网络; (d) 双星网络

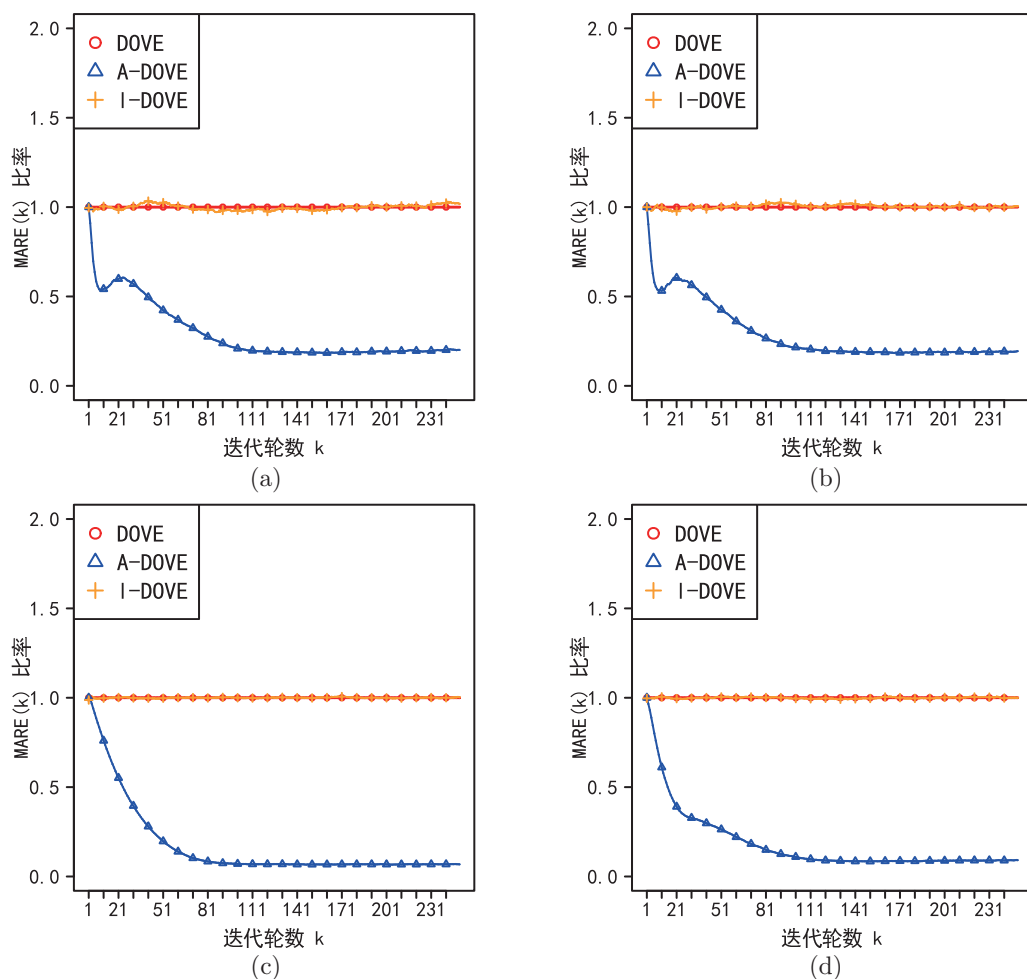


图 4 (网络版彩图) 三种去中心化在线平均估计的 $\text{MARE}(k)$ 比率随轮数 k 的变化图. (a) Erdős-Rényi 网络 ($p_c=0.2$); (b) 路径网络; (c) 集中式网络; (d) 双星网络

比率. 此时仍然选取 $l=2$, 并且比较了 100 次重复实验里前 250 个轮次的平均 $\text{MARE}(k)$ 比率结果, 呈现在图 4 中. 从图中可以看出, 与注 2.1 分析的相同, l -DOVE 算法得到的结果与 DOVE 算法非常接近, 没有改善收敛度的效果. 相比之下, A-DOVE 的加速效果非常明显.

4.1.2 节点数量的影响

这里仍然考虑 4 种不同的网络结构: 连接概率为 $p_c=0.2$ 的 Erdős-Rényi 网络、路径网络、集中式网络和双星网络. 但是在每种网络结构中, 分别考虑网络中节点个数为 $n=10, 20, 50$ 的情形. 图 5-7 展示了对应的实验结果. 如图中所示, 当 k 足够大时, A-DOVE 的 $\text{MARE}(k)$ 比率趋向于一个常数, 展示了算法的收敛性. 相比之下, 基于随机梯度的方法的 $\text{MARE}(k)$ 比率则呈增加趋势, 其中 DSG 方法的表现甚至远不如 DOVE. 总体而言, 除了路径网络 $n=50$ 的情形外, 我们的加速方法 A-DOVE 在几乎所有情形下表现最好, 超过了其他方法. 通过图片间的对比可以看出, 当节点数量 n 上升时, 我们的算法产生的 $\text{MARE}(k)$ 比率也在逐步上升, 这说明了当网络规模增大、数据更加分散时, 去中心化在线平均任务的难度提升, 这与我们所建立的理论中, 误差上界与 n 呈正比的结论是一致的.

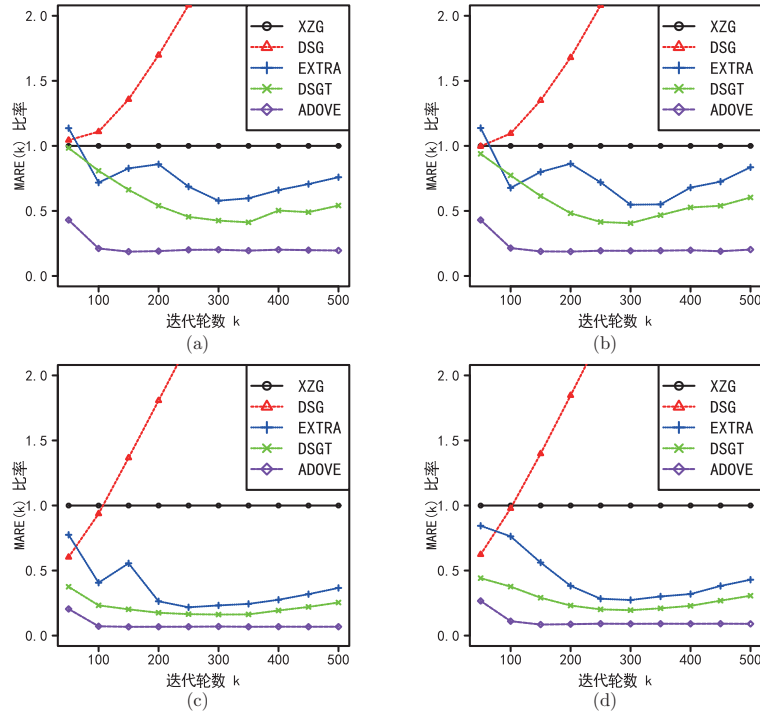


图 5 (网络版彩图) 当 $n = 10$ 时不同的去中心化在线平均估计的 $MARE(k)$ 比率随轮数 k 的变化图. (a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) 路径网络; (c) 集中式网络; (d) 双星网络

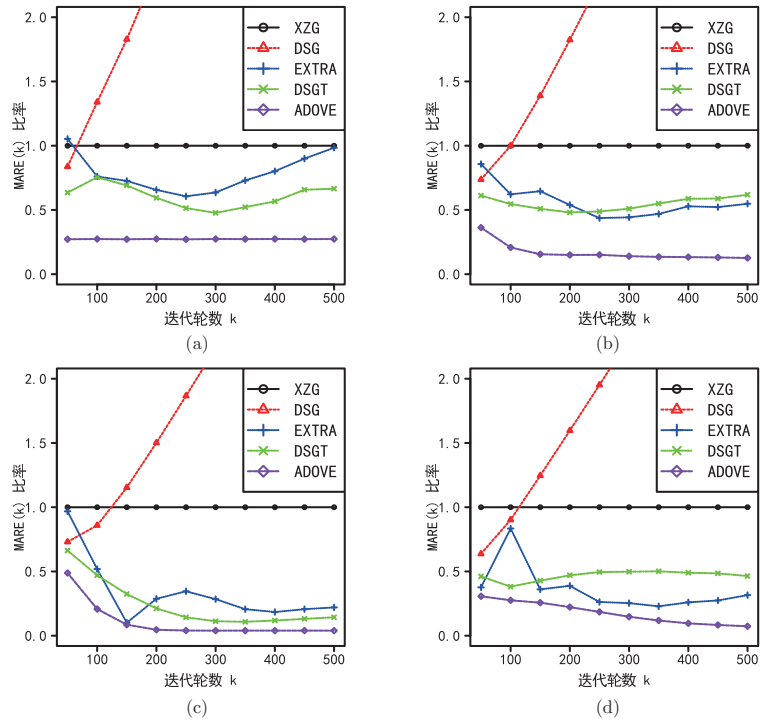


图 6 (网络版彩图) 当 $n = 20$ 时不同的去中心化在线平均估计的 $MARE(k)$ 比率随轮数 k 的变化图. (a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) 路径网络; (c) 集中式网络; (d) 双星网络

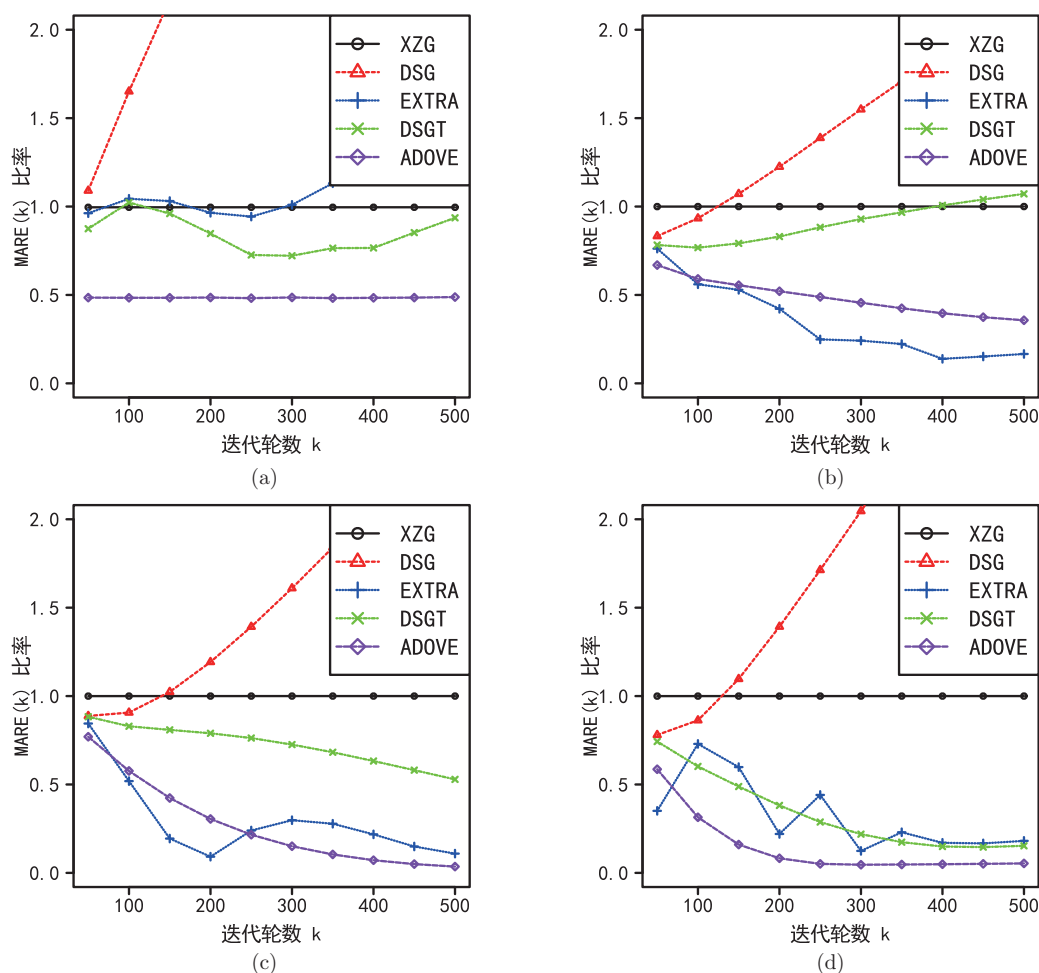


图 7 (网络版彩图) 当 $n = 50$ 时不同的去中心化在线平均估计的 $\text{MARE}(k)$ 比率随轮数 k 的变化图. (a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) 路径网络; (c) 集中式网络; (d) 双星网络

4.1.3 网络连接性的影响

在这个实验中, 将网络节点数量固定为 $n = 20$, 研究去中心化网络中边的数量, 或者说是网络连接性对算法表现产生的影响. 考虑 4 个连接概率分别为 $p_c = 0.2$, $p_c = 0.3$, $p_c = 0.5$ 和 $p_c = 0.8$ 的 Erdős-Rényi 网络. 当连接概率较大时, Erdős-Rényi 网络更有可能拥有更多的边, 连接性更好. 从理论上分析, 较大的 p_c 导致较小的 ρ_2 , 进而能导致更快的收敛速度. 正如图 8 所示, 当节点数量固定时, 我们的加速方法 A-DOVE 在不同连接性的网络上都能表现出色, 所产生的 $\text{MARE}(k)$ 比率低于其他方法.

4.1.4 方差异质性的影响

在这个实验中, 我们旨在探究去中心化网络中节点之间异质方差的影响. 具体而言, 在每个节点 i 中, 噪声方差 σ_i 是从均匀分布 $\text{Unif}[1, 9]$ 中随机生成的, 但是不同节点的方差可能不相同. 图 9 展示了异质性方差情形下不同去中心化在线平均方法产生的 $\text{MARE}(k)$ 比率. 显然, 我们的加速方法 A-DOVE 在所有情形下表现最好, 并且与 XZG 和基于随机梯度的方法相比, A-DOVE 对异质方差并不敏感.

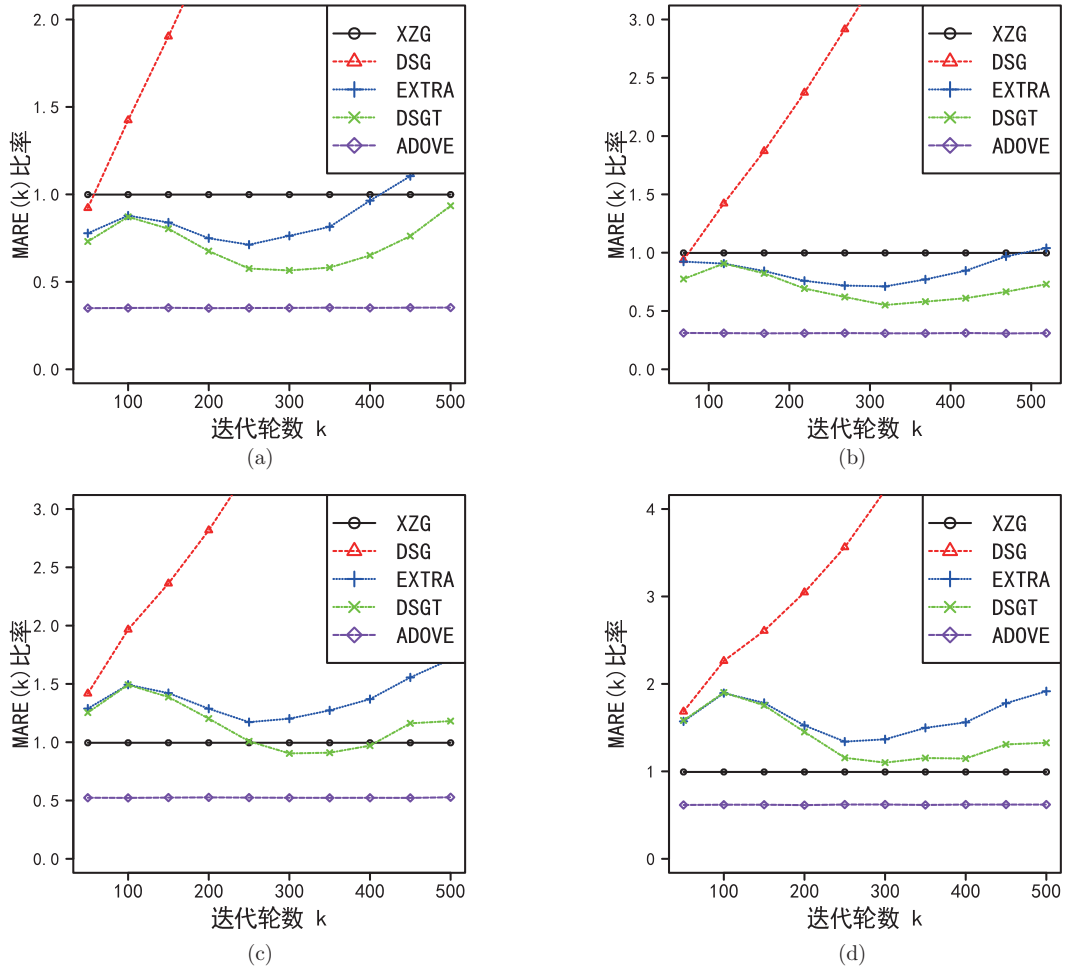


图 8 (网络版彩图) 不同连接概率 p_c 下不同去中心化在线平均估计的 $\text{MARE}(k)$ 比率随轮数 k 的变化图. (a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) Erdős-Rényi 网络 ($p_c = 0.3$); (c) Erdős-Rényi 网络 ($p_c = 0.5$); (d) Erdős-Rényi 网络 ($p_c = 0.8$)

4.2 去中心化在线最小二乘

本小节将真实的回归系数设置为 $\theta^* = (1, (p-2)/(p-1), (p-3)/(p-1), \dots, 0)/\sqrt{p}$, 其中变量维数 $p = 10$. 通过 $\mu_i \sim \text{Unif}[0, 5]$ 一次性随机生成局部均值参数 μ_i , $1 \leq i \leq n$, 并将其固定为所有重复实验中都相同. 此外, 对于每个节点 i 和第 k 个轮次, 从多元正态分布 $\mathcal{N}(\mu_i \theta^*, \Sigma)$ 中生成样本 $x_i^{(k)}$, 其中 $\Sigma = (\sigma_{ts})_{1 \leq t, s \leq p}$ 是一个 Toeplitz 矩阵, 其元素 $\sigma_{ts} = 0.5^{|t-s|}$, 噪声 $\varepsilon_i^{(k)}$ 来自标准正态分布 $\mathcal{N}(0, 1)$. 在去中心化在线回归任务中, 使用去中心化在线估计 $\theta_i^{(k)}$ 与经典的最小二乘估计 $\hat{\theta}_{LS}(k)$ 之间的均方根相对误差 (root mean square relative error, RMSRE) 作为衡量算法表现的标准:

$$\text{RMSRE}(k) = \frac{1}{n} \sum_{i=1}^n \frac{\|\theta_i^{(k)} - \hat{\theta}_{LS}(k)\|_2}{\|\hat{\theta}_{LS}(k)\|_2}.$$

与第 4.1 小节类似, 以 DOSE 产生的 $\text{RMSRE}(k)$ 为基准, 比较不同方法产生的 $\text{RSMRE}(k)$ 比率.

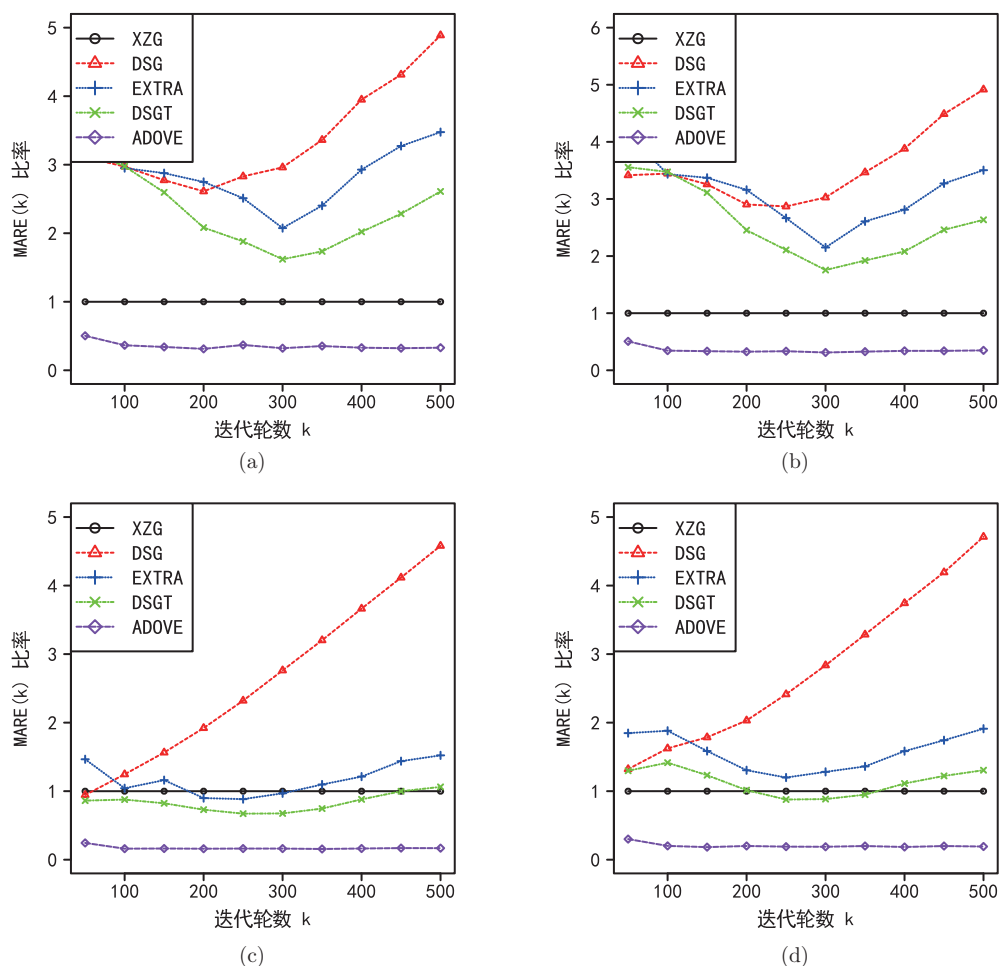


图 9 (网络版彩图) 在异质性方差存在的情形下, 不同网络结构下不同去中心化在线平均估计的 $\text{MARE}(k)$ 比率随轮数 k 的变化图. (a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) 路径网络; (c) 集中式网络; (d) 双星网络

4.2.1 窗宽的影响

这里考虑 4 种不同的网络结构: 连接概率为 $p_c = 0.2$ 的 Erdős-Rényi 网络、路径网络、集中式网络和双星网络. 与在线平均的结果类似, 当选取窗宽 $l = 1$ 时, A-DOSE 无法收敛并得到正确的最小二乘估计结果, 因此这个实验中只比较 A-DOSE 方法在不同窗宽 $l = 2, 3, 5, 10, 15$ 下的结果, 以探索 l 的选取对算法加速的影响. 从图 10 中可以看出, 与平均问题类似, $l = 2$ 的 A-DOSE 方法已经足够好, 为了方便起见, 在后续的模拟中只考虑 $l = 2$ 的情形. 注意到在这个实验中, 选取 $l = 15$ 得到的 RMSRE 要显然大于 $l = 2$ 时的 RMSRE. 这可以通过我们的理论结果进行解释. 当 l 太大时, 误差上界 (3.1) 中由于逼近误差产生的第三项成为主导项, 导致了较大的 RMSRE. 这与我们理论分析中关于 l 的选取不能太大的描述一致.

4.2.2 节点个数的影响

本实验考虑 4 种不同的网络结构: 连接概率为 $p_c = 0.2$ 的 Erdős-Rényi 网络、路径网络、集中式网络和双星网络. 为了探究去中心化网络中节点数量 n 的影响, 分别在节点数量 $n = 10, 20, 50$ 的情

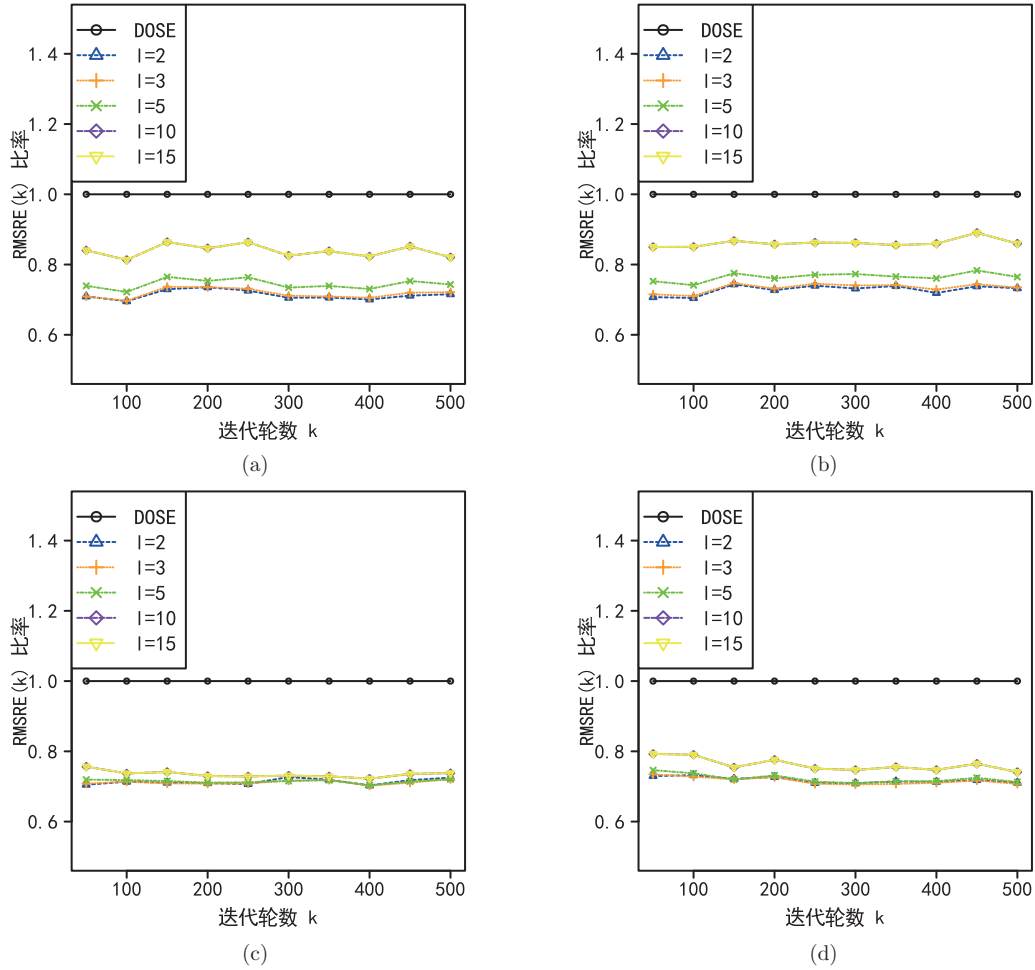


图 10 (网络版彩图) 不同窗宽下加速去中心化在线最小二乘估计的 $\text{RMSRE}(k)$ 比率随轮数 k 的变化图. (a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) 路径网络; (c) 集中式网络; (d) 双星网络

形下进行实验. 图 11–13 展示了我们的 A-DOSE 方法在不同的网络节点数量下相较于其他方法的优越性.

4.2.3 网络连接性的影响

本实验考虑连接概率分别为 $p_c = 0.2$, $p_c = 0.3$, $p_c = 0.5$ 和 $p_c = 0.8$ 的 Erdős-Rényi 网络. 网络节点数量设置为 $n = 20$. 当连接概率越高时, 边的数量会更多, 网络的连接性也更好. 如图 14 显示, 我们的加速方法 A-DOSE 的性能在不同的网络连接性下都优于其他方法.

4.2.4 方差异质性的影响

本实验考虑 4 种不同的网络结构: 连接概率为 $p_c = 0.2$ 的 Erdős-Rényi 网络、路径网络、集中式网络和双星网络. 这个实验的目的是说明去中心化网络节点之间, 如果存在异质方差, 则会对算法表现产生的影响. 方差的异质性来自如下两个方面. (1) 协变量 $\mathbf{x}_i^{(k)}$ 的协方差矩阵设置为 $\Sigma_i = (\rho_i^{|t-s|})_{1 \leq t, s \leq p}$, 其中参数 ρ_i 是从均匀分布 $\text{Unif}[0.2, 0.7]$ 中随机生成的. 该协方差矩阵的设置来源于一阶自回归模型

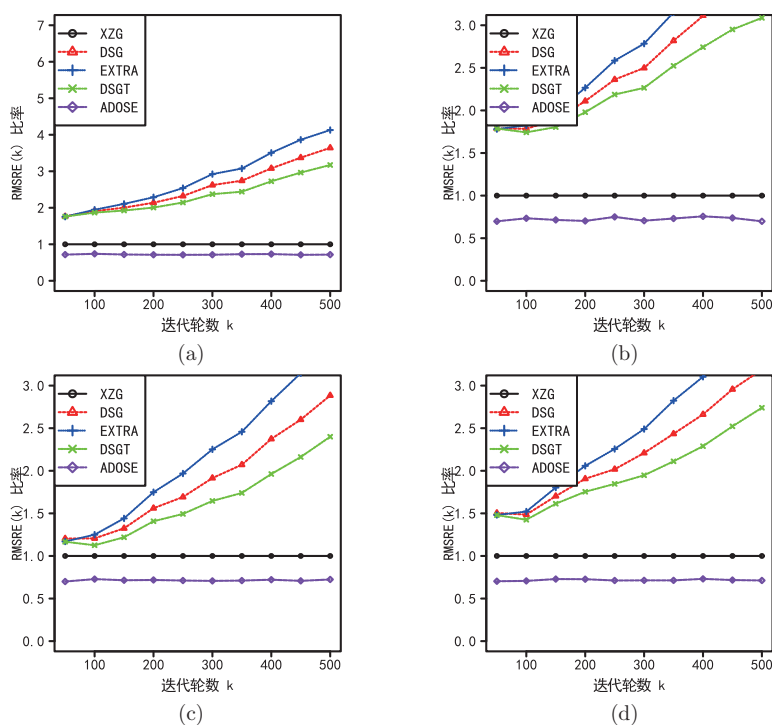


图 11 (网络版彩图) 当 $n = 10$ 时不同的去中心化在线最小二乘估计的 $\text{RMSRE}(k)$ 比率随轮数 k 的变化图. (a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) 路径网络; (c) 集中式网络; (d) 双星网络

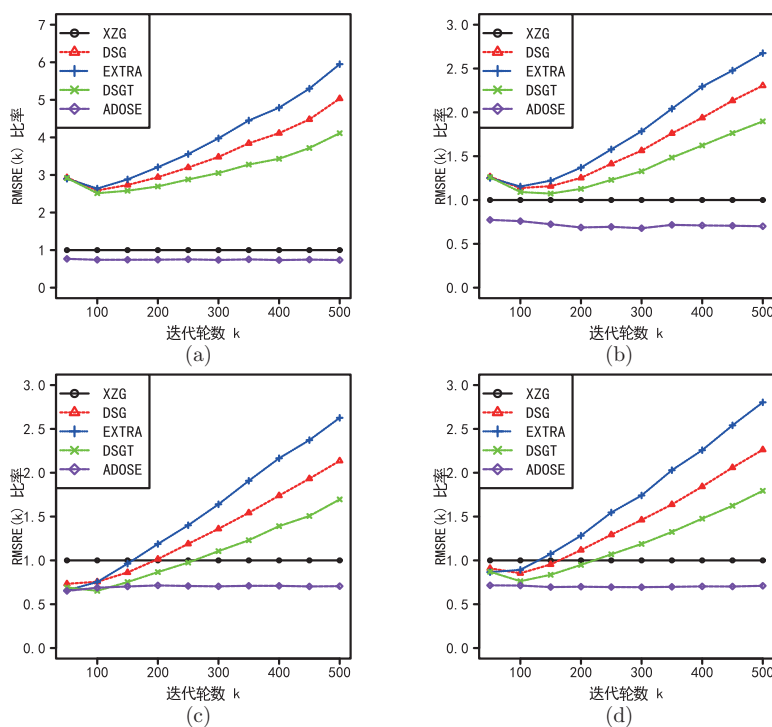


图 12 (网络版彩图) 当 $n = 20$ 时不同的去中心化在线最小二乘估计的 $\text{RMSRE}(k)$ 比率随轮数 k 的变化图. (a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) 路径网络; (c) 集中式网络; (d) 双星网络

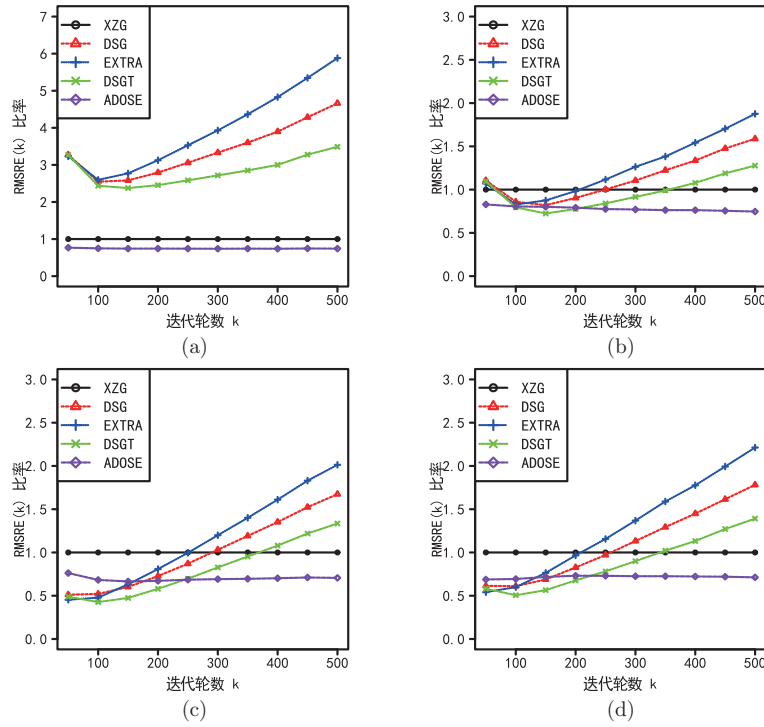


图 13 (网络版彩图) 当 $n = 50$ 时不同的去中心化在线最小二乘估计的 $\text{RMSRE}(k)$ 比率随轮数 k 的变化图。
(a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) 路径网络; (c) 集中式网络; (d) 双星网络

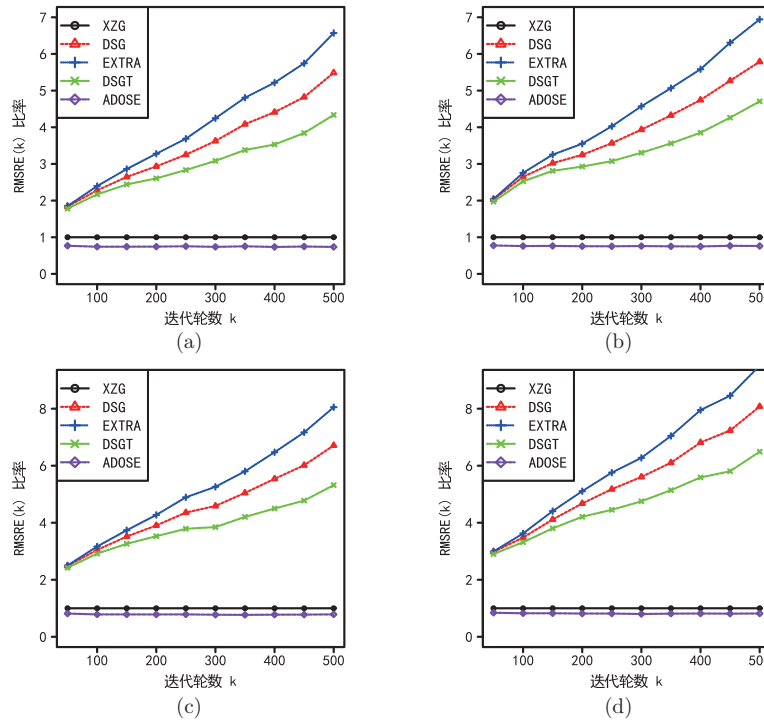


图 14 (网络版彩图) 不同连接概率 p_c 下不同去中心化在线最小二乘估计的 $\text{RMSRE}(k)$ 比率随轮数 k 的变化图。
(a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) Erdős-Rényi 网络 ($p_c = 0.3$); (c) Erdős-Rényi 网络 ($p_c = 0.5$); (d) Erdős-Rényi 网络 ($p_c = 0.8$)

的自协方差矩阵. (2) 噪声方差 σ_i 是从均匀分布 $\text{Unif}[1, 9]$ 中生成的. 如图 15 显示, 我们的加速方法 A-DOSE 在所有情形下表现最佳, 并且对方差的异质性具有较好的适应性.

4.2.5 验证中心极限定理

本小节通过一个简单的实验来验证推论 3.3 中建立的中心极限定理. 基本的实验设置和之前的实验相同. 选取第 1 个节点上得到的最后一个时刻的估计 $\theta_1^{(K)}$, 在 100 次重复实验中记录其第一个维数上的估计值为 $\{\xi_i\}_{i=1}^{100}$. 由于此时协变量 $\mathbf{x}$ 的均值 μ 向量和协方差矩阵 Σ 是已经设定好的, 我们自然也可以计算得到中心极限定理中的协方差矩阵 Σ_μ . 然后可以得到一组随机样本: $\sqrt{nK}(\xi_i - [\theta^*]_1)/([\Sigma_\mu]_{1,1})$, $i = 1, \dots, 100$. 如图 16 所示, 我们绘制了这组随机样本对应的标准密度直方图和 QQ 图. 从图中可以看出, 样本的分布和标准正态分布比较接近.

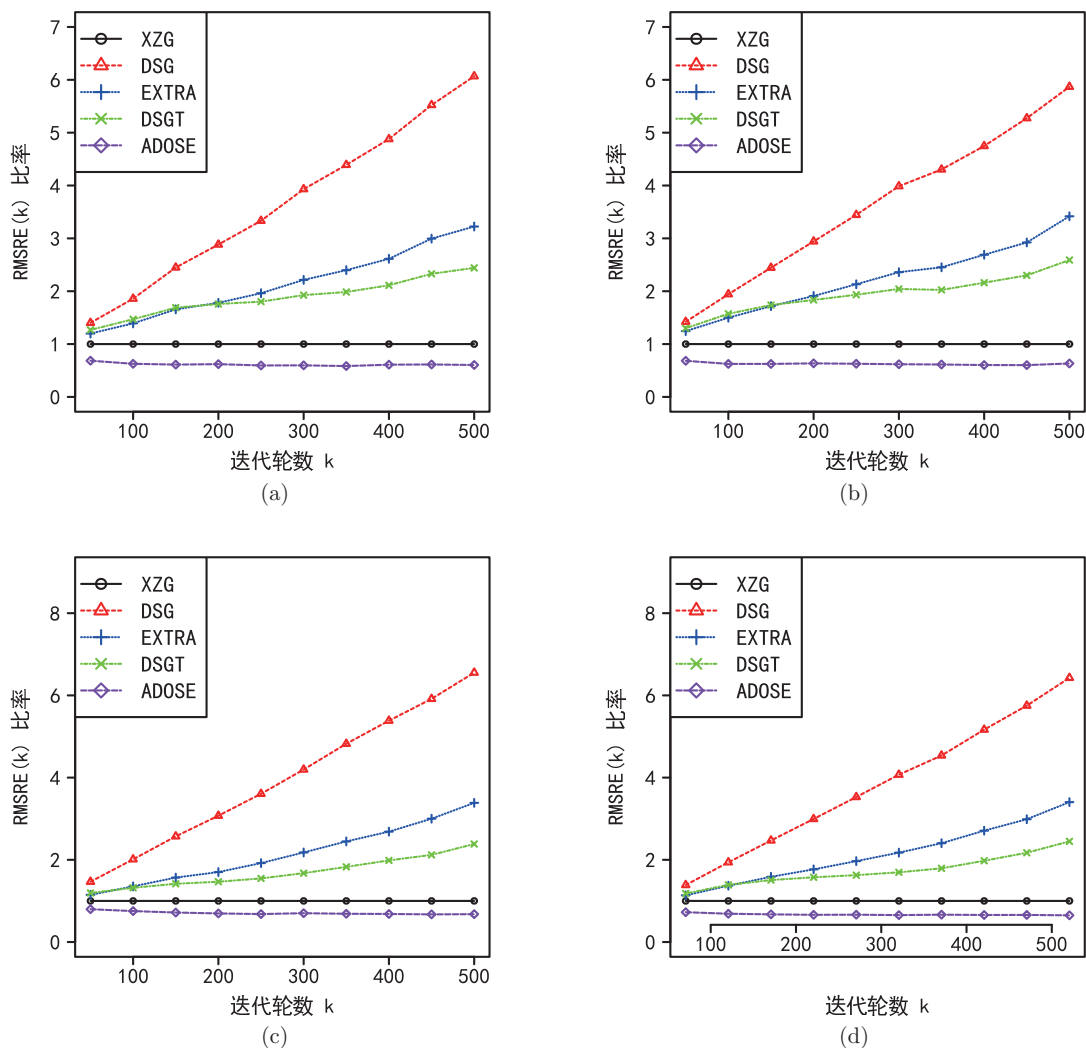


图 15 (网络版彩图) 在异质性方差存在的情形下, 不同网络结构下不同去中心化在线最小二乘估计的 $\text{RMSRE}(k)$ 比率随轮数 k 的变化图. (a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) 路径网络; (c) 集中式网络; (d) 双星网络

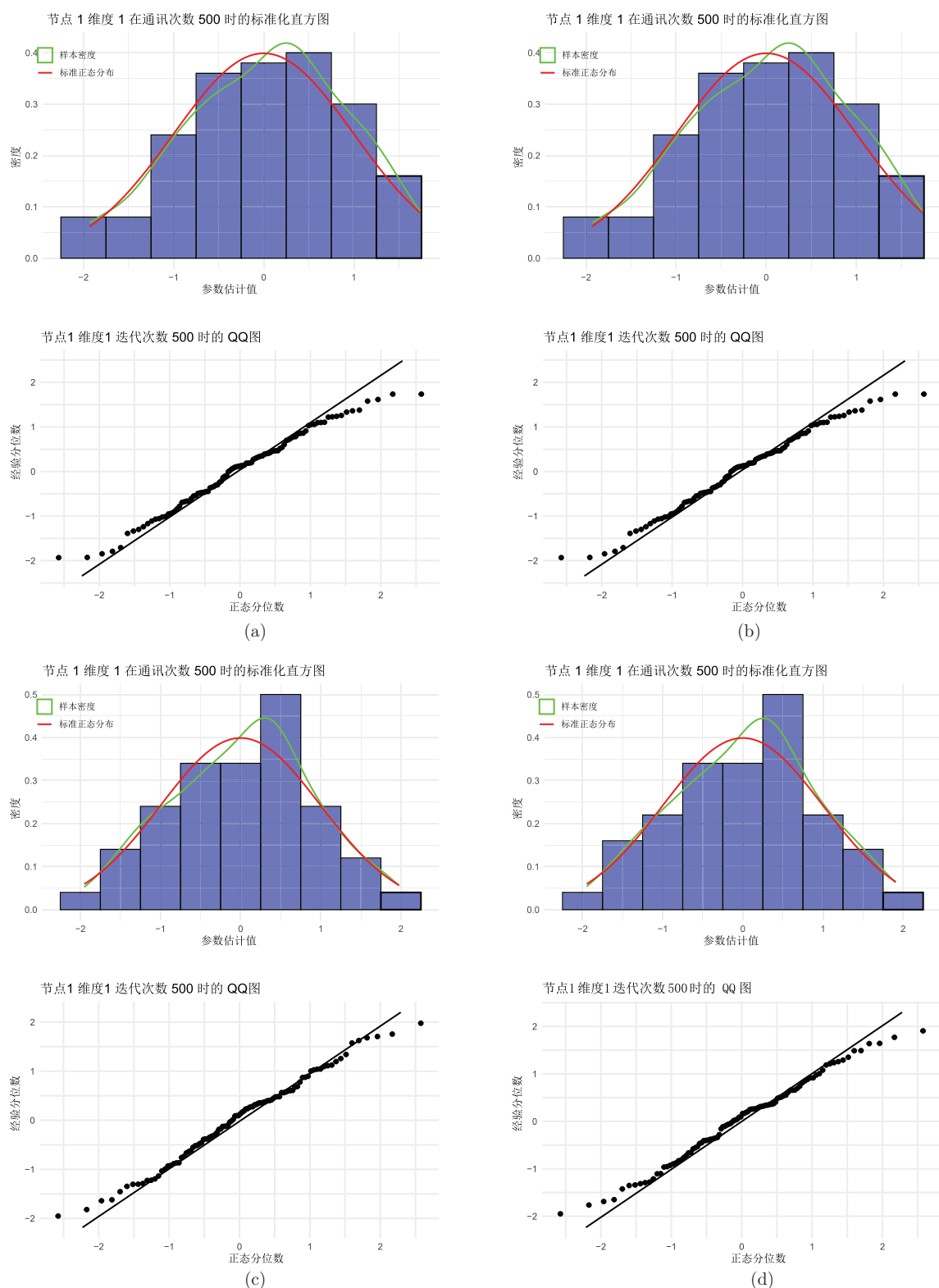


图 16 (网络版彩图) 不同网络结构下第 1 个节点上最终时刻的第一维数据的标准直方图和 QQ 图. (a) Erdős-Rényi 网络 ($p_c = 0.2$); (b) 路径网络; (c) 集中式网络; (d) 双星网络

5 结论

本文探讨了去中心化在线平均和最小二乘回归问题. 本文的主要目的是加速去中心化在线估计在这些问题上的收敛速度. 对于去中心化在线平均问题, 本文提出了一种通过 l -MA 技术实现的 A-DOVE 算法. 基于去中心化在线平均的思想和用两个平均分量逼近经典 LSE 的思路, 本文提出了一种 A-DOSE 算法. 理论和模拟结果中, 我们的方法在加速收敛速度方面的表现也是令人满意的. 在未来的工作中, 可以探索在更复杂的网络上应用加速去中心化在线平均算法, 如有向网络和时变网络.

致谢 感谢审稿人对本文提出宝贵意见.

参考文献

- 1 Arablouei R, Dogancay K, Werner S, et al. Adaptive distributed estimation based on recursive least-squares and partial diffusion. *IEEE Trans Signal Process*, 2014, 62: 3510–3522
- 2 Bylander T, Rosen B. A perceptron-like online algorithm for tracking the median. In: *Proceedings of International Conference on Neural Networks*. New York: IEEE, 1997, 2219–2224
- 3 Cardot H, Cénac P, Godichon-Baggioni A. Online estimation of the geometric median in Hilbert spaces: Nonasymptotic confidence balls. *Ann Statist*, 2017, 45: 591–614
- 4 Cattivelli F S, Lopes C G, Sayed A H. Diffusion recursive least-squares for distributed estimation over adaptive networks. *IEEE Trans Signal Process*, 2008, 56: 1865–1877
- 5 Chen Y, Rosolia U, Ames A D. Decentralized task and path planning for multi-robot systems. *IEEE Robot Autom Lett*, 2021, 6: 4337–4344
- 6 Cheng S, Hong Y. Distributed dynamic online linear regression over unbalanced graphs. In: *Proceedings of the 16th IEEE International Conference on Control & Automation (ICCA)*. New York: IEEE, 2020, 870–875
- 7 Claes D, Oliehoek F, Baier H, et al. Decentralised online planning for multi-robot warehouse commissioning. In: *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems*. Providence: Amer Math Soc, 2017, 492–500
- 8 Crammer K, Singer Y. Online ranking by projecting. *Neural Comput*, 2005, 17: 145–175
- 9 De Carli N, Salaris P, Giordano P R. Online decentralized perception-aware path planning for multi-robot systems. In: *Proceedings of International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. New York: IEEE, 2021, 128–136
- 10 Dekel O, Gilad-Bachrach R, Shamir O, et al. Optimal distributed online prediction using mini-batches. *J Mach Learn Res*, 2012, 13: 165–202
- 11 Delvenne J C, Carli R, Zampieri S. Optimal strategies in the average consensus problem. *Systems Control Lett*, 2009, 58: 759–765
- 12 Duchi J C, Agarwal A, Wainwright M J. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Trans Automat Control*, 2011, 57: 592–606
- 13 Foster D P. Prediction in the worst case. *Ann Statist*, 1991, 19: 1084–1090
- 14 Foster D P, Kale S, Karloff H. Online sparse linear regression. In: *Proceedings of the 29th Annual Conference on Learning Theory*. New York: PMLR, 2016, 960–970
- 15 Gupta V, De S. An energy-efficient edge computing framework for decentralized sensing in WSN-assisted IoT. *IEEE Trans Wireless Commun*, 2021, 20: 4811–4827
- 16 Hong S, Chae J. Distributed online learning with multiple kernels. *IEEE Trans Neural Netw Learn Syst*, 2023, 34: 1263–1277
- 17 Hosseini S, Chapman A, Mesbahi M. Online distributed convex optimization on dynamic networks. *IEEE Trans Automat Control*, 2016, 61: 3545–3550
- 18 Inalhan G, Stipanovic D M, Tomlin C J, et al. Decentralized optimization, with application to multiple aircraft coordination. In: *Proceedings of the 41st IEEE Conference on Decision and Control*. New York: IEEE, 2002, 1147–1155
- 19 Jiménez A C, García-Díaz V, González-Crespo R, et al. Decentralized online simultaneous localization and mapping for multi-agent systems. *Sensors*, 2018, 18: 2612
- 20 Johansson B. On distributed optimization in networked systems. PhD Thesis. Stockholm: KTH Royal Institute of Technology, 2008

- 21 Kontoudis G P, Stilwell D J. Decentralized nested Gaussian processes for multi-robot systems. In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA). New York: IEEE, 2021, 8881–8887
- 22 Koppel A, Paternain S, Richard C, et al. Decentralized online learning with kernels. IEEE Trans Signal Process, 2018, 66: 3240–3255
- 23 Lan G, Lee S, Zhou Y. Communication-efficient algorithms for decentralized and stochastic optimization. Math Program, 2020, 180: 237–284
- 24 Langford J, Smola A J, Zinkevich M. Slow learners are fast. In: Proceedings of the 23rd International Conference on Neural Information Processing Systems. New York: Curran Associates, 2009, 2331–2339
- 25 Levin D A, Peres Y. Markov Chains and Mixing Times. Providence: Amer Math Soc, 2017
- 26 Li X, Yi X, Xie L. Distributed online optimization for multi-agent networks with coupled inequality constraints. IEEE Trans Automat Control, 2020, 66: 3575–3591
- 27 Li X, Yi X, Xie L. Distributed online convex optimization with an aggregative variable. IEEE Trans Control Netw Syst, 2021, 9: 438–449
- 28 Liu J, Morse A S. Accelerated linear iterations for distributed averaging. Annu Rev Control, 2011, 35: 160–165
- 29 Nedić A, Olshevsky A, Rabbat M G. Network topology and communication-computation tradeoffs in decentralized optimization. In: Proceedings of the IEEE. New York: IEEE, 2018, 953–976
- 30 Olshevsky A. Linear time average consensus on fixed graphs. IFAC-PapersOnLine, 2015, 48: 94–99
- 31 Olshevsky A, Tsitsiklis J N. Convergence speed in distributed consensus and averaging. SIAM J Control Optim, 2009, 48: 33–55
- 32 Pu S, Nedić A. Distributed stochastic gradient tracking methods. Math Program, 2021, 187: 409–457
- 33 Rajput K P, Ahmed M F, Venkategowda N K D, et al. Robust decentralized and distributed estimation of a correlated parameter vector in MIMO-OFDM wireless sensor networks. IEEE Trans Commun, 2021, 69: 6894–6908
- 34 Ren W, Beard R W, Atkins E M. Information consensus in multivehicle cooperative control. IEEE Contr Syst Mag, 2007, 27: 71–82
- 35 Roncalli M, Bistaffa F, Farinelli A. Decentralized power distribution in the smart grid with ancillary lines. Mobile Netw Appl, 2019, 24: 1654–1662
- 36 Shahrampour S, Jadbabaie A. Distributed online optimization in dynamic environments using mirror descent. IEEE Trans Automat Control, 2017, 63: 714–725
- 37 Shi W, Ling Q, Wu G, et al. EXTRA: An exact first-order algorithm for decentralized consensus optimization. SIAM J Optim, 2015, 25: 944–966
- 38 Warnat-Herresthal S, Schultze H, Shastry K L, et al. Swarm learning for decentralized and confidential clinical machine learning. Nature, 2021, 594: 265–270
- 39 Xiao L, Boyd S. Fast linear iterations for distributed averaging. Systems Control Lett, 2004, 53: 65–78
- 40 Xie S, Zhang Y, Guo L. Convergence of a distributed least squares. IEEE Trans Automat Control, 2021, 66: 4952–4959
- 41 Xin R, Kar S, Khan U A. Decentralized stochastic optimization and machine learning: A unified variance-reduction framework for robust performance and fast convergence. IEEE Signal Process Mag, 2020, 37: 102–113
- 42 Xin R, Khan U A, Kar S. An improved convergence analysis for decentralized online stochastic non-convex optimization. IEEE Trans Signal Process, 2021, 69: 1842–1858
- 43 Yuan D, Proutiere A, Shi G. Distributed online linear regressions. IEEE Trans Inform Theory, 2020, 67: 616–639
- 44 Yuan D, Proutiere A, Shi G. Distributed online optimization with long-term constraints. IEEE Trans Automat Control, 2022, 67: 1089–1104
- 45 Zhang W, Zhao P, Zhu W, et al. Projection-free distributed online learning in networks. In: Proceedings of the 34th International Conference on Machine Learning. New York: PMLR, 2017, 4054–4062
- 46 Zinkevich M A, Weimer M, Smola A, et al. Parallelized stochastic gradient descent. In: Proceedings of the 24th International Conference on Neural Information Processing Systems. New York: Curran Associates, 2010, 2595–2603

附录 A 基本算法及其理论

在下面的算法 A1 中, 首先对基础的去中心在线平均算法进行更加详细的描述, 并在命题 2.1 中给出了 DOVE 算法的算法收敛性理论.

算法 A1 DOVE 算法

输入: 初始向量 $\mathbf{z}^{(0)} = \mathbf{0}$.

- 1: 对于 $k = 1, 2, \dots$ 执行
- 2: $\mathbf{z}^{(k)} = \mathbf{W}(\mathbf{z}^{(k-1)} + \mathbf{y}^{(k)})$.
- 3: 结束循环

命题 A.1 在假设 3.1 成立的前提下, 对于算法 A1 在第 k 个轮次产生的估计, 有

$$\left\| \frac{1}{k} \mathbf{z}^{(k)} - \mathbf{1} \bar{y}_{[k]} \right\|_2 \leq \frac{\rho_2}{(1 - \rho_2)k} \max_{1 \leq t \leq k} \|\mathbf{y}^{(t)} - \mathbf{1} \bar{y}^{(t)}\|_2.$$

证明 基于假设 3.1 和双随机矩阵的性质, 根据文献 [25], 有 $\rho_2 < 1$ 和 $\|\mathbf{W} - \mathbf{1}\mathbf{1}^T/n\| = \rho_2$, 其中 $\|\cdot\|$ 表示矩阵的谱范数. 由 $(\mathbf{W} - \mathbf{1}\mathbf{1}^T/n)\mathbf{1} = \mathbf{0}$, 可以得到

$$\begin{aligned} \|\mathbf{z}^{(k)} - \mathbf{1} \bar{z}^{(k)}\|_2 &\leq \rho_2 \|\mathbf{z}^{(k-1)} - \mathbf{1} \bar{z}^{(k-1)}\|_2 + \rho_2 \|\mathbf{y}^{(k)} - \mathbf{1} \bar{y}^{(k)}\|_2 \\ &\leq \sum_{t=1}^k \rho_2^{k-t+1} \|\mathbf{y}^{(t)} - \mathbf{1} \bar{y}^{(t)}\|_2 \\ &\leq \frac{\rho_2}{1 - \rho_2} \max_{1 \leq t \leq k} \|\mathbf{y}^{(t)} - \mathbf{1} \bar{y}^{(t)}\|_2. \end{aligned}$$

进一步地, 有

$$\bar{z}^{(k)} = \bar{z}^{(k-1)} + \bar{y}^{(k)} = k \bar{y}_{[k]}.$$

至此命题 2.1 证毕. □

算法 A2 描述了 DOSE 算法的细节. 关于 DOSE 的收敛理论可以类似于定理 3.1 进行建立, 因此在此省略.

算法 A2 DOSE 算法

输入: 初始估计 $\Xi^{(0)} = \mathbf{O}_{n \times p}$ 和 $\check{\Sigma}_m^{(0)} = \mathbf{O}_{n \times p}$, 其中 $1 \leq m \leq p$.

- 1: 对于 $k = 1, 2, \dots$ 执行
- 2: $\Xi^{(k)} = \mathbf{W}(\Xi^{(k-1)} + \Gamma^{(k)})$.
- 3: 对于 $m = 1, \dots, p$ 执行
- 4: $\check{\Sigma}_m^{(k)} = \mathbf{W}(\check{\Sigma}_m^{(k-1)} + \check{\Phi}_m^{(k)})$.
- 5: 结束循环
- 6: 对于 $i = 1, \dots, n$ 执行
- 7: $\theta_i^{(k)} = (\Sigma_i^{(k)})^{-1} \xi_i^{(k)}$.
- 8: 结束循环
- 9: 结束循环

附录 B 相关定理的证明**附录 B.1** 定理 3.1 的证明

我们有

$$\mathbf{g}^{(k)} - \mathbf{1} \bar{g}^{(k)} = \left(\mathbf{W} - \frac{1}{n} \mathbf{1}\mathbf{1}^T \right) (\mathbf{g}^{(k-1)} - \mathbf{1} \bar{g}^{(k-1)}) + \frac{1}{l} (\mathbf{y}^{(k)} - \mathbf{y}^{(k-l)}) - \frac{1}{l} \mathbf{1} (\bar{y}^{(k)} - \bar{y}^{(k-l)}).$$

注意到

$$\left\| \frac{1}{l}(\mathbf{y}^{(k)} - \mathbf{y}^{(k-l)}) - \frac{1}{l}(\bar{\mathbf{y}}^{(k)} - \bar{\mathbf{y}}^{(k-l)}) \right\|_2 = \left\| \left(\mathbf{I}_n - \frac{1}{n} \mathbf{1} \mathbf{1}^T \right) \frac{1}{l}(\mathbf{y}^{(k)} - \mathbf{y}^{(k-l)}) \right\|_2 \leq \frac{1}{l} \|\mathbf{y}^{(k)} - \mathbf{y}^{(k-l)}\|_2,$$

其中 $\mathbf{I}_n$ 表示一个 $n \times n$ 的单位矩阵. 这导致当 $k > l$ 时, 有

$$\|\mathbf{g}^{(k)} - \mathbf{1}\bar{g}^{(k)}\|_2 \leq \rho_2^{k-l} \|\mathbf{g}^{(l)} - \mathbf{1}\bar{g}^{(l)}\|_2 + \frac{1}{(1-\rho_2)l} \max_{l+1 \leq t \leq k} \|\mathbf{y}^{(t)} - \mathbf{y}^{(t-l)}\|_2; \quad (\text{B.1})$$

当 $k \leq l$ 时, 有

$$\|\mathbf{g}^{(k)} - \mathbf{1}\bar{g}^{(k)}\|_2 \leq \rho_2^k \|\mathbf{g}^{(0)} - \mathbf{1}\bar{g}^{(0)}\|_2 + \frac{2}{1-\rho_2} \max_{1 \leq t \leq l} \|\mathbf{y}^{(t)} - \mathbf{1}\bar{y}^{(t)}\|_2 = \frac{2}{1-\rho_2} \max_{1 \leq t \leq l} \|\mathbf{y}^{(t)} - \mathbf{1}\bar{y}^{(t)}\|_2. \quad (\text{B.2})$$

将不等式 (B.1) 和 (B.2) 结合起来, 当 $k > l$ 时, 有下面的不等式成立:

$$\begin{aligned} \|\mathbf{z}^{(k)} - \mathbf{1}\bar{z}^{(k)}\|_2 &\leq \sum_{i=1}^k \rho_2^i \|\mathbf{g}^{(k-i+1)} - \mathbf{1}\bar{g}^{(k-i+1)}\|_2 \\ &\leq \frac{\rho_2}{1-\rho_2} \max_{l+1 \leq i \leq k} \|\mathbf{g}^{(i)} - \mathbf{1}\bar{g}^{(i)}\|_2 + \frac{\rho_2^{k-l+1}}{1-\rho_2} \max_{1 \leq i \leq l} \|\mathbf{g}^{(i)} - \mathbf{1}\bar{g}^{(i)}\|_2 \\ &\leq \frac{\rho_2}{(1-\rho_2)^2 l} \max_{l+1 \leq i \leq k} \|\mathbf{y}^{(i)} - \mathbf{y}^{(i-l)}\|_2 + \frac{2\rho_2^{k-l+1}}{(1-\rho_2)^2} \max_{1 \leq i \leq l} \|\mathbf{y}^{(i)} - \mathbf{1}\bar{y}^{(i)}\|_2. \end{aligned}$$

接下来, 将对逼近误差部分建立上界. 根据辅助变量 $\mathbf{g}^{(k)}$ 的定义和性质 $\mathbf{W}\mathbf{1} = \mathbf{1}$, 对于 $k \leq l$, 针对平均值 $\bar{g}^{(k)} = \mathbf{1}^T \mathbf{g}^{(k)} / n$, 有以下结果:

$$\bar{g}^{(k)} = \bar{g}^{(k-1)} + \frac{1}{k} \sum_{t=1}^k \bar{y}^{(t)} - \frac{1}{k-1} \sum_{t=1}^{k-1} \bar{y}^{(t)} = \frac{1}{k} \sum_{t=1}^k \bar{y}^{(t)}.$$

而对于 $k > l$, 有

$$\bar{g}^{(k)} = \bar{g}^{(l)} + \frac{1}{l} \sum_{t=k-l+1}^k \bar{y}^{(t)} - \frac{1}{l} \sum_{t=1}^l \bar{y}^{(t)} = \frac{1}{l} \sum_{t=k-l+1}^k \bar{y}^{(t)}.$$

综上所述, 对于 $k > l$, 有

$$\bar{z}^{(k)} - k\bar{y}_{[k]} = \sum_{i=1}^k \bar{g}^{(i)} - \sum_{i=1}^k \bar{y}^{(i)} = \sum_{i=1}^l \left(\frac{1}{i} - \frac{1}{l} \right) \sum_{t=1}^i (\bar{y}^{(t)} - \bar{y}^{(k-i+1)}).$$

这引出了上界:

$$\|\mathbf{1}\bar{z}^{(k)} - k \cdot \mathbf{1}\bar{y}_{[k]}\|_2 \leq \sqrt{n}l \max_{1 \leq i \leq l} \max_{1 \leq t \leq l} |\bar{y}^{(t)} - \bar{y}^{(k-i+1)}|.$$

于是可以总结得到

$$\begin{aligned} \left\| \frac{1}{k} \mathbf{z}^{(k)} - \mathbf{1}\bar{y}_{[k]} \right\|_2 &\leq \frac{\rho_2}{(1-\rho_2)^2 k l} \max_{l+1 \leq t \leq k} \|\mathbf{y}^{(t)} - \mathbf{y}^{(t-l)}\|_2 + \frac{4\rho_2^{k-l+1}}{(1-\rho_2)^2 k} \max_{1 \leq t \leq l} \|\mathbf{y}^{(t)} - \mathbf{1}\bar{y}^{(t)}\|_2 \\ &\quad + \frac{\sqrt{n}l}{k} \max_{1 \leq i \leq l} \max_{1 \leq t \leq l} |\bar{y}^{(t)} - \bar{y}^{(k-i+1)}|. \end{aligned}$$

证毕.

附录 B.2 推论 3.1 的证明

可以看出, 如果用范数 $\sqrt{\mathbb{E}(\|\cdot\|_2^2)} =: \|\cdot\|_{\mathbb{E}}$ 替换 Euclid 范数 $\|\cdot\|_2$, 定理 3.1 仍然成立. 然后, 注意到 $\|\bar{y}^{(t)} - \bar{y}^{(k-i+1)}\|_{\mathbb{E}} = O(1/\sqrt{n})$ 和 $\|\mathbf{y}^{(i)} - \boldsymbol{\mu}\|_{\mathbb{E}} = O(\sqrt{n})$ 对于所有 i, k 和 t 一致成立, 推论 3.1 得证.

附录 B.3 定理 3.2 的证明

根据假设 3.3 和 3.4, 有

$$\|\gamma_i^{(k)}\|_{\mathbb{E}}^2 = \|\mathbf{x}_i^{(k)} \mathbf{y}_i^{(k)}\|_{\mathbb{E}}^2 \leq 2\|\mathbf{x}_i^{(k)} \varepsilon_i^{(k)}\|_{\mathbb{E}}^2 + 2\boldsymbol{\theta}^{*\top} \boldsymbol{\Sigma}_i \boldsymbol{\theta}^* = O(1),$$

其中 $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma} + \boldsymbol{\mu}_{\mathbf{x},i} \boldsymbol{\mu}_{\mathbf{x},i}^{\top}$, 这里 $O(1)$ 关于所有的 i 和 k 是一致的. 将定理 3.1 中的 $\mathbf{y}^{(t)}$ 替换为 $\boldsymbol{\Gamma}^{(t)}$ 的每一列, 对于所有 $1 \leq m \leq p$, 一致地有

$$\|\boldsymbol{\xi}_{\cdot,m}^{(k)} - 1\bar{\xi}_m^{(k)}\|_{\mathbb{E}} = O\left(\frac{\sqrt{n}}{(1-\rho_2)^2 kl} + \frac{\rho_2^{k-l+1}\sqrt{n}}{(1-\rho_2)^2 k} + \frac{l}{k}\right),$$

其中 $\boldsymbol{\xi}_{\cdot,m}^{(k)}$ 表示 $\boldsymbol{\Xi}^{(k)}$ 的第 m 列, $\bar{\xi}_m^{(k)} = \sum_{i=1}^n \xi_{i,m}^{(k)}/n$. 观察到 $\bar{\boldsymbol{\xi}}^{(k)} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n \boldsymbol{\xi}_i^{(k)} = \bar{\boldsymbol{\gamma}}_{[k]}$, 可以得到

$$\|\boldsymbol{\Xi}^{(k)} - 1\bar{\boldsymbol{\gamma}}_{[k]}\|_{\mathbb{E}} = O\left(\frac{\sqrt{np}}{(1-\rho_2)^2 kl} + \frac{\rho_2^{k-l+1}\sqrt{np}}{(1-\rho_2)^2 k} + \frac{\sqrt{pl}}{k}\right).$$

类似地, 有

$$\sqrt{\sum_{i=1}^n \|\boldsymbol{\Sigma}_i^{(k)} - \bar{\boldsymbol{\Phi}}_{[k]}\|_{\mathbb{E}}^2} = O\left(\frac{\sqrt{np}}{(1-\rho_2)^2 kl} + \frac{\rho_2^{k-l+1}\sqrt{np}}{(1-\rho_2)^2 k} + \frac{pl}{k}\right).$$

只要轮次 k 满足

$$k \geq c_1 \left(\frac{\sqrt{np}}{(1-\rho_2)^2 l} + \frac{\rho_2^{k-l+1}\sqrt{np}}{(1-\rho_2)^2} + pl \right),$$

这里 c_1 是一个足够大的常数, 就有 $\max_{1 \leq i \leq n} \|\boldsymbol{\Sigma}_i^{(k)} - \bar{\boldsymbol{\Phi}}_{[k]}\|_{\mathbb{E}}^2 \leq \delta$, 其中 δ 足够小. 因此可以看出, 在假设 3.3 下, $\max_{1 \leq i \leq n} |(\boldsymbol{\Sigma}_i^{(k)})^{-1}|_2 = O_P(1)$. 由此可得

$$\sqrt{\sum_{i=1}^n \|(\boldsymbol{\Sigma}_i^{(k)})^{-1} - \bar{\boldsymbol{\Phi}}_{[k]}^{-1}\|_2^2} = O_P(1) \sqrt{\sum_{i=1}^n \|\boldsymbol{\Sigma}_i^{(k)} - \bar{\boldsymbol{\Phi}}_{[k]}\|_2^2} = O_P\left(\frac{\sqrt{np}}{(1-\rho_2)^2 kl} + \frac{\rho_2^{k-l+1}\sqrt{np}}{(1-\rho_2)^2 k} + \frac{pl}{k}\right).$$

于是有

$$\sqrt{\sum_{i=1}^n \|\boldsymbol{\theta}_i^{(k)} - \hat{\boldsymbol{\theta}}_{LS}(k)\|_2^2} = O_P\left(\frac{\sqrt{np}}{(1-\rho_2)^2 kl} + \frac{\rho_2^{k-l+1}\sqrt{np}}{(1-\rho_2)^2 k} + \frac{pl}{k}\right).$$

证毕.

附录 B.4 推论 3.2 的证明

以下的事实成立:

$$\hat{\theta}_{LS}(k) - \theta^* = \bar{\Phi}_{[k]}^{-1} \left(\frac{1}{nk} \sum_{i=1}^n \sum_{t=1}^k \mathbf{x}_i^{(t)} \varepsilon_i^{(t)} \right).$$

注意到 $|\mathbf{x}_i^{(k)} \varepsilon_i^{(k)}|_E = O(1)$, 这里 $O(1)$ 关于所有的 i 和 t 是一致的. 可以证明, 在假设 3.3 下, $\|\bar{\Phi}_{[k]}^{-1}\|_2 = O_P(1)$. 然后可以得到 $\|\hat{\theta}_{LS}(k) - \theta^*\|_2 = O_P(\sqrt{p/nk})$. 根据定理 3.2, 得出如下结论: 当轮数 k 足够大时, 有

$$\sqrt{\sum_{i=1}^n \|\theta_i^{(k)} - \theta^*\|_2^2} = O_P \left(\frac{\sqrt{np}}{(1 - \rho_2)^2 kl} + \frac{\rho_2^{k-l+1} \sqrt{np}}{(1 - \rho_2)^2 k} + \frac{pl}{k} + \sqrt{\frac{p}{k}} \right).$$

证毕.

附录 B.5 推论 3.3 的证明

由连续映射定理和 Slutsky 定理, 结合事实 $\bar{\Phi}_{[k]} \xrightarrow{P} \Sigma_\mu$, 有

$$\sqrt{nk}(\hat{\theta}_{LS}(k) - \theta^*) = \sqrt{nk} \bar{\Phi}_{[k]}^{-1} \left(\frac{1}{nk} \sum_{i=1}^n \sum_{t=1}^k \mathbf{x}_i^{(t)} \varepsilon_i^{(t)} \right) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \Sigma_\mu^{-1} M \Sigma_\mu^{-1}),$$

其中 $k \rightarrow \infty$. 结合这个渐近性质和定理 3.2 中的结果: $\|\theta_i^{(k)} - \hat{\theta}_{LS}(k)\|_2 = O_P(\sqrt{np/k}) = o_P(1/\sqrt{nk})$, 可以验证对于所有节点 $1 \leq i \leq n$, 当 $k \rightarrow \infty$ 时, 有

$$\sqrt{nk}(\theta_i^{(k)} - \theta^*) \xrightarrow{d} \mathcal{N}(\mathbf{0}, \Sigma_\mu^{-1} M \Sigma_\mu^{-1}).$$

Accelerated online averaging over networks

Weidong Liu, Xiao Liu, Xiaojun Mao & Zeyu Wu

Abstract Online learning has been extensively investigated due to the rapidly growing demand for capabilities to process the massive data collected sequentially. Most of the existing works have studied online problems over the centralized framework. In this paper, we mainly focus on the decentralized online averaging problem and propose a novel accelerated algorithm. We also extend our approach to the decentralized online least squares regression. We establish both algorithmic convergence theories and statistical guarantees to demonstrate the acceleration of our algorithms. Moreover, we are the first to provide the central limit theorem of the decentralized online least squares estimator. Several numerical simulations are conducted to demonstrate the feasibility and effectiveness of our proposed algorithms.

Keywords acceleration algorithm, decentralized network, distributed online learning, online averaging

MSC(2020) 62R07, 68T09, 68W15, 93A14

doi: 10.1360/SSM-2024-0223