doi:10.19920/j.cnki.jmsc.2023.05.004

大数据下分位数回归通讯有效算法及其应用①

周　勇，张澍一*，李子洋
（华东师范大学统计与数据科学前沿理论及应用教育部重点实验室，
统计交叉科学研究院和统计学院，上海 200062）

摘要：考虑风险度量中常见的分位数回归模型，给出在超大容量数据且复杂数据类型下的几类快速分布式算法. 虽然仅考虑分位数回归模型，但本文提供的算法大多数可以应用到其它更一般的模型中. 由于分位数回归模型的目标函数为非光滑函数，通常的分块集成法和光滑函数高效通讯算法并不适用. 本文首先针对完整观测数据，给出了分位数回归模型参数估计的等度连续法，光滑函数逼近法和改进的数萃（Meta）方法三种分布式通讯有效算法. 进一步，考虑了非平衡半监督数据，分别针对无标签数据样本量较小和较大两种情形，提出了加权损失函数法和改进的数萃方法两种数据融合方法. 所提出的方法可以把分散在不同机器上的半监督数据进行数据融合，从而实现不同数据类型和不同样本量情形下的高效通讯分布式计算，提高算法的精度和参数估计的效率. 本文通过大量仿真模拟研究了所提出的算法在有限样本下的表现，并将其应用到了洛杉矶流浪人口数的实际数据分析中，发现其均具有较好的准确性.

关键词：大数据分析；数据融合；通讯有效算法；分位数回归
中图分类号：C93－03；O212.1　**文献标识码**：A　**文章编号**：1007－9807(2023)05－0070－33

0　引　言

近年来，“大数据”已成为互联网、新闻媒体、学术机构、政府企业管理人员等多方关注的热点. 随着现代科学技术——尤其是计算机、网络信息、生物工程等技术的发展，海量数据出现在许多不同的自然科学和人文科学领域，包括生物学、医学、信息技术、经济学、金融学、环境科学等，并以前所未有的速度产生和积累. 大数据涵盖的数据量大、包罗万象、变化速度快、存储的形式多种多样，可以是包括文字、图片、视频等多种信息的集合. 在新一轮科技和产业竞争中，大数据资源已经是与自然资源、人力资源一样重要的战略资源，将推动整个人类社会的创新发展，促进社会生产力的发展，改善人们的生活方式、企业的决策流程和国家的治理模式.

大数据的分析与应用，关键是发展出大数据分析的理论方法和技术，特别是高效算法、大数据金融计量模型及决策模型，同时应用这些理论方法解决金融大数据中的重要问题. 从理论上取得金融大数据分析方法的创新突破，是目前大数据分析理论与应用中亟需解决的问题. 此外，除了金融大数据，大数据分析在系统控制、数据安全等领域也有着迫切的需求与广泛的应用前景，并且已经形成了一些算法基础[1,2].

目前主流的三大分布式计算系统 Hadoop、

① 收稿日期：2021－12－31；修订日期：2022－12－26.
基金项目：国家自然科学基金资助重点项目(71931004)；科技部国家重点研发计划资助项目(2021YFA1000101；2021YFA1000102；2021YFA1000104)；国家自然科学基金资助项目(72201101)；教育部人文社会科学研究资助青年基金项目(22YJC910013)；上海市浦江人才计划资助项目(21PJC034).
通讯作者：张澍一(1991—)，女，山东济南人，博士，助理教授. Email：syzhang@fem.ecnu.edu.cn

Storm、Spark 被广泛运用于大数据领域，但是这些系统无法处理带复杂数据结构类型的大数据计算，例如带有缺失的非平衡数据和半监督数据等. 因此，本文对不同类型的数据进行信息挖掘，同时也考虑并行计算，在统计学环境下构建模型，发展出可行的分布式计算算法.

在大数据环境下，如何进行统计学的计算实现是目前研究的一大难点. 传统统计学理论和统计计算是分不开的，很多传统统计学的理论与方法从提出到在实际中应用，都需要算法上的实现. 这些传统算法在很多情况下不再适用于分布式计算，同时大数据分析的实时性与高效性使得统计学理论在大数据应用与国家大数据战略制定中面临新的挑战. 事实是，统计学家们正在为解决大数据分析中的主要挑战——大数据分析理论与方法，与实现大数据应用而努力着，并逐渐形成大数据背景下统计学理论研究的新视角与新方向. 面对复杂的超大容量数据，其主要障碍可以概括为以下四点：1）大数据的海量特点使得数据不能够被一台计算机内存一次读入；2）计算过程太过缓慢，无法在有限的时间内得出结果；3）数据结构复杂，例如，在不同机器存放的数据是非平衡的数据或是部分无标签的数据；4）所建立模型的目标函数不是光滑函数，使得通常基于求导的算法不再适用，更谈不上不同机器上结果的传输. 这些障碍对传统的统计方法带来了挑战，需要发展新的统计理论和计算方法，特别是新的分布式算法. 而经济、管理与金融大数据相比于传统数据具有大量化、多样化和低值化等显著特点. 面对这些复杂特征，原有的计量模型理论和计算方法有诸多局限性而难以实现，需要创建有针对性的灵活建模方法，发展能高效稳定地进行数据处理和分析的算法.

在统计学应用中，大数据的出现为更有效的统计推断提供了契机. 但是与此同时，大数据的高样本量和高生成速度限制了传统统计方法的实际应用. 这主要是因为大数据分析通常伴随着极高的计算代价、通讯成本和内存需求. 目前大数据计算的研究重点是并行化的分布式计算，包括分布式架构以及算法上采用分而治之的方法，这对统计推断和应用具有重要意义. 对于分布式环境下的参数估计，一个简单直观的分治方法（以下简称 naive 分布式方法）是，首先把数据随机分成若干子数据集并分配到不同的子计算机上，在所有子计算机上并行计算出相应的估计量，其中每个估计量都通过基于子集的目标函数或估计方程获得. 然后，所有子计算机将它们的估计量发送回核心计算机，核心计算机再通过这些基于子机器的估计量的简单平均来获得一个聚合的估计量. 尽管这种 naive 分布式方法可以应用于各种参数估计问题，但此方法通常会给估计的有效性带来损失. 特别是，Zhang 等[3] 在经验风险最小化的背景下研究了这种基于简单平均的 naive 估计量. 其中，未知参数是通过最小化一个由所有单样本损失函数的平均值定义的目标函数来估计的. 他们说明，如果要使得聚合后的估计量的收敛速度达到最优，划分的子计算机的数量应远小于每台计算机数据子集的大小. 这是一个相当严格的条件，限制了此方法的应用场景.

在传统问题中，整个数据集是存储在单台计算机中的，如果将整个数据集进行简单划分并基于传统统计方法直接进行并行计算，将会产生过高的通讯成本. 除此之外，在许多实际情况下，海量数据集本身就存储在分布式的环境中，而不是集中式地存储在一台机器中. 例如，一家公司可能会从处于不同位置的传感器中收集数据，并将收集到的各个子数据集存储在不同的局域服务器中. 而且，不同位置的传感器每天产生的数据量可能很大. 为了在一台计算机上实现统计方法，需要将所有这些存储在不同服务器中的数据发送到同一台核心计算机中，这可能导致难以承受的通讯成本. 即使将分布在不同机器上的数据分别在本地服务器上进行计算，然后再将各个机器上的结果送回到一台主机器进行合并，目前设计的传统算法通常需要传递二阶导数矩阵，例如牛顿迭代法，这样也会带来极大的通讯成本. 此外，Jaggi 等[4] 指出，在不同计算机之间进行数据通讯可能比在一台计算机上进行复杂的内部计算还要消耗更多的时间成本.

在传统的统计学和计量经济学中，分位数回归模型是一类应用广泛的基础性模型. 例如，分位数回归模型可以应用到在险价值（value at risk，

VaR)的度量中,其作为一类强有力的风险度量模型,在金融风险度量领域有着重要的应用,文献众多在此不一一列出. 考虑带复杂因素的条件 VaR (conditional VaR,CVaR,也即分位数回归),目前也已有一些研究. Cai 和 Wang[5] 在考虑包括宏观经济因素、市场风险暴露因素和过去收益的各种风险因素下,提出了 CVaR 和条件期损 (conditional expected shortfall,CES) 两个风险度量,并研究了其统计性质. 刘晓倩和周勇[6] 研究了一类类似于 VaR 模型的 CES 风险度量模型,王子剑等[7] 将其扩展到了相依时间序列的框架. 本文的方法可以推广到 CES 风险度量模型. 针对现实应用背景较强的管理学数据,例如伏红勇等[8] 通过 CVaR 准则评估了农户天气看跌期权契约的相关管理模式,黄潇和黄守军[9] 利用类似的分位数回归模型,评估了流动人口自雇问题的不同决定机制与其收入差异的相关关系. 这些都是 VaR 模型或者其同源的分位数回归模型在近年来被频繁采用的典型案例.

通常,分位数回归模型的参数估计是基于核验 (check) 损失函数的优化问题,或可从由其一阶条件导出的一组估计方程获得参数估计量. 它们的计算过程都需要一次性地对整个数据集进行集中式的处理. 当数据量很大时,这往往难以实现. 现有的针对分位回归模型的大数据分析算法大多都是基于“分块集成”(divide and conquer, DC) 的加权平均方法,参见 Chen 等[10],Chen 和 Zhou[11]. 这些方法仅需一次传输,估计效率便可以很高. 但是由于需要传输权重矩阵,就对传输带宽提出了较高的要求,因此具有进一步改进算法的需要. 分块集成算法是一类迭代算法,此外还有基于数萃 (Meta) 方法的非迭代算法,参见 Singh 等[12]. 但是这类算法也需要传输协方差矩阵,具有较高的通讯成本. 图 1 给出了分块集成迭代算法和数萃算法的算法示意图. 潘莹丽等[13] 通过构造全局损失函数的替代损失函数,设计了 Proximal-ADMM 算法进行参数估计,然而这种算法在变量维数较高的情形并不适用. 还有一类算法是基于梯度信息传输的通讯有效算法,参见 Jordan 等[14] 和 Fan 等[15],它们在降低通讯成本方面具有巨大的优势,但是无法直接用于分位数回归模型等具有非光滑目标函数的模型. 后来有例如 Wang 和 Lian [16] 的工作将此通讯有效算法推广至分位数回归. 但他们的文章缺少对算法相关统计性质的讨论,比如推广后的通讯有效算法的效率如何等问题没有明确解答,也没有进一步深入讨论对于不同源数据存在异质性的场景,算法是否仍然适用. 而 Duan 等[17] 的文章专门研究了通讯有效算法在异质性数据下如何做统计推断,但并没有针对分位数回归模型做进一步展开. 就此,本文提出了新的分布式并行计算方法,同时也考虑了非平衡半监督数据等复杂数据,其中最重要的是大数据下分位回归模型的通讯有效算法的设计. 本文将围绕这个问题展开,给出几种有效的分位数回归模型的分布式计算方法,同时针对非平衡半监督数据提出了两种新的通讯有效算法. 同时,本文将给出算法的伪代码,以便可以方便地应用到实际问题中. 虽然仅考虑分位数回归模型,但本文提供的大多数算法可以应用到其它更一般的模型中. 特别是改进的数萃 (Meta) 方法,可以应用到更一般的基于最小损失或最小风险的估计中.

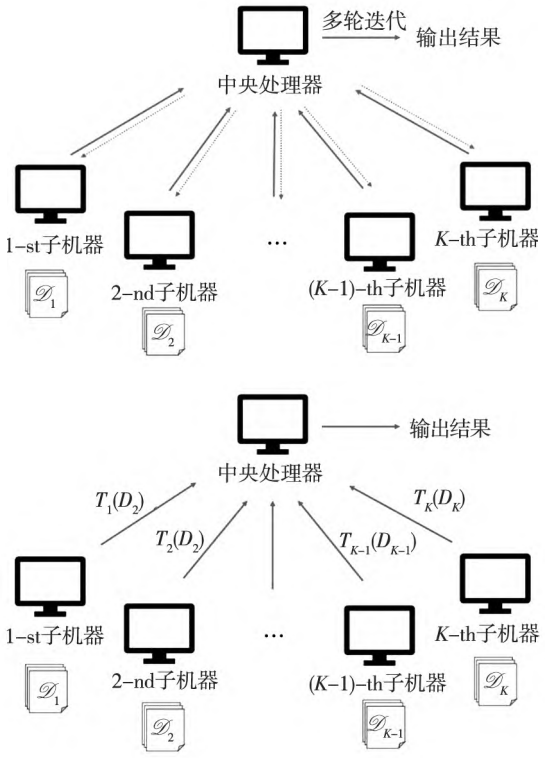

图 1　分块集成算法(上)与数萃算法(下)的算法示意图

Fig. 1 Diagram of DC algorithm (top) and Meta algorithm (bottom)

1　模型及数据

下面介绍分位数回归的基本方法与本文的问题框架. 设 $(\boldsymbol{X}_1^{\mathrm{T}},Y_1)^{\mathrm{T}},(\boldsymbol{X}_2^{\mathrm{T}},Y_2)^{\mathrm{T}},\cdots,(\boldsymbol{X}_N^{\mathrm{T}},Y_N)^{\mathrm{T}}$ 为随机向量 $(\boldsymbol{X}^{\mathrm{T}},Y)^{\mathrm{T}}$ 的 N 个独立同分布观测样本，其中 $\boldsymbol{X}_i\in\mathbb{R}^p$ 为第 i 个观测的 p 维协变量，Y_i 为第 i 个观测的响应变量. 假设样本量 N 非常大，以致不能在一台机器上用传统方法计算得到结果. 令 $\vec{\boldsymbol{X}}=(1,\boldsymbol{X}^{\mathrm{T}})^{\mathrm{T}}$. 对于给定的分位数水平 $\tau\in(0,1)$，线性分位数回归模型假设数据 $(\boldsymbol{X}^{\mathrm{T}},Y)^{\mathrm{T}}$ 满足

$$Y=\boldsymbol{\theta}(\tau)^{\mathrm{T}}\vec{\boldsymbol{X}}+\varepsilon_\tau,P(\varepsilon_\tau\leqslant 0\mid\boldsymbol{X})=\tau \tag{1}$$

其中 $\boldsymbol{\theta}(\tau)$ 为未知参数，ε_τ 为不可观测的误差项. 这等价于认为在给定 $\boldsymbol{X}$ 的条件下，Y 的 τ 条件分位数为 $Q_\tau(Y\mid\boldsymbol{X})=\boldsymbol{\theta}(\tau)^{\mathrm{T}}\vec{\boldsymbol{X}}$，即真实的条件分位数与协变量 $\boldsymbol{X}$ 具有线性关系. 更一般的情形则假定 Y 在给定 $\boldsymbol{X}$ 的条件下的 τ 条件分位数不一定与协变量 $\boldsymbol{X}$ 具有线性关系. 这可以描述为以下模型，对于可测函数 $g(\cdot;\tau):\mathbb{R}^p\to\mathbb{R}$，变量 $(\boldsymbol{X}^{\mathrm{T}},Y)^{\mathrm{T}}$ 满足

$$Y=g(\boldsymbol{X};\tau)+\varepsilon_\tau,P(\varepsilon_\tau\leqslant 0\mid\boldsymbol{X})=\tau \tag{2}$$

此时，在给定 $\boldsymbol{X}$ 的条件下，Y 的 τ 条件分位数为 $Q_\tau(Y\mid\boldsymbol{X})=g(\boldsymbol{X};\tau)$. 经典分位数回归估计定义为

$$\widehat{\boldsymbol{\theta}}_N=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\,\mathcal{L}_N(\boldsymbol{\theta})$$

$$\mathcal{L}_N(\boldsymbol{\theta})=\frac{1}{N}\sum_{i=1}^{N}\rho_\tau(Y_i-\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i) \tag{3}$$

其中 $\rho_\tau(x)=x\{\tau-I(x\leqslant 0)\}$ 称为核验（check）函数，$I(\cdot)$ 为示性函数. $\widehat{\boldsymbol{\theta}}_N$ 称为非对称最小二乘（Asymmetric Least Square，ALS）估计. 由经典的分位数估计的理论可知，若数据满足线性模型（1），则 $\widehat{\boldsymbol{\theta}}_N\xrightarrow{\mathrm{P}}\boldsymbol{\theta}(\tau)$. 若线性模型（1）不满足，则需要考虑更一般的风险度量模型（2）. 此时定义 $\boldsymbol{\theta}_0(\tau)$ 为平均损失函数 $\mathrm{E}\{\rho_\tau(Y-\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}})\}$ 的极小值点，即

$$\boldsymbol{\theta}_0(\tau)=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\,\mathrm{E}\{\rho_\tau(Y-\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}})\} \tag{4}$$

则按照式(3)定义的 $\widehat{\boldsymbol{\theta}}_N$ 为其样本类似. 在一些一般性条件成立的情况下，$\widehat{\boldsymbol{\theta}}_N\xrightarrow{\mathrm{P}}\boldsymbol{\theta}_0(\tau)$. 注意到，在线性模型（1）成立时，$\boldsymbol{\theta}(\tau)=\boldsymbol{\theta}_0(\tau)=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\,\mathrm{E}\{\rho_\tau(Y-\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}})\}$. 故不论对于线性模型（1），还是更一般的模型（2），总可以从式（4）出发定义真值 $\theta_0(\tau)$，且能够保证式（3）中的 $\widehat{\boldsymbol{\theta}}_N$ 为 $\boldsymbol{\theta}_0(\tau)$ 的一致估计（也称为相合估计）. 故本文将不假设线性模型（1）成立，而是在模型（2）的基础上进行展开，待估参数的真值定义由式（4）给出. 值得注意的是，利用式（4）的方式定义参数真值等价于用线性模型逼近真实模型. 在实际应用中，真实模型可能不具有线性形式，此时包含更广泛情况的模型（2）显然对数据的描述更加精确，但是由于线性模型具有良好的解释性，往往仍然选择用模型（1）作为模型（2）的近似. 虽然这不会影响参数估计量的值，但需要对估计量的收敛值 $\boldsymbol{\theta}_0(\tau)$ 有更加谨慎的解读.

更一般地，可以从损失函数的角度来理解式（4）. 在人工智能和大数据分析中，人们往往更关心损失，而预报对应于某类损失函数下的最小风险. 具体而言，通常会考虑一个损失函数 $L\{Y,f(\boldsymbol{X};\boldsymbol{\theta})\}$，其中 $\{f(\boldsymbol{X};\boldsymbol{\theta}):\boldsymbol{\theta}\in\boldsymbol{\Theta}\}$ 为一族感兴趣的参数模型，考虑在该损失下参数的最小风险真值 $\boldsymbol{\theta}_0=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\,\mathrm{E}[L\{Y,f(\boldsymbol{X};\boldsymbol{\theta})\}]$，以及预报 $f(\boldsymbol{X};\boldsymbol{\theta}_0)$. 而分位数损失 $\rho_\tau(\cdot)$ 就是其中的一个非光滑非对称损失函数，这么做的目的是在分位数损失下求最小风险线性预报 $\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}$. 因此，真值与预报的定义本质上不依赖于真实模型，而是在感兴趣的模型族下最小化风险得到，即无论真实的条件分位数与协变量之间是否符合线性关系，都可以计算出一个最小风险线性预报. 若与条件分位数模型相对应，那么当真实模型是线性的，则最小化分位数风险就等价于通常的分位数回归；当真实模型是非线性的，则通过最小化线性分位数风险可以得到真实条件分位数的一阶线性逼近.

基于损失函数的最小风险预报相比于传统分位数回归具有巨大优势. 传统分位数回归是从模型出发定义真实参数与预报，需要条件分位数满足线性假设，然而这一假设在实际应用中显得过于苛刻，从模型出发所得结果往往与实际存在一定差距，使其在人工智能与大数据分析中存在很大的局限性. 最小风险预报恰好可以解决这一问题. 从损失函数的角度出发，最小风险预报不需要对真实模型做任何假设，从而具有更大的模型灵活性和稳健性，这也是无模型（model-free）方

法相比于有模型（model-based）方法的优势所在. 基于损失函数,将搜索范围限制在线性模型族中,又可以降低计算复杂度并提高预报的可解释性. 这相当于对真实条件分位数进行一阶泰勒展开,所得最小风险预报为真实条件分位数的最优线性逼近. 进一步,还可以推广到二阶或更高阶多项式逼近. 此时,预报误差由模型逼近误差与估计误差两部分组成,其中模型逼近误差是无法从传统分位数回归中体现的. 特别地,最小风险预报可以归为统计学 M 估计的框架中,从而具有更多理论工具与方法来进行统计分析.

在数据量过大或者数据分布式存储于不同子机器上时,对 $\boldsymbol{\theta}_0(\tau)$ 的估计量往往无法直接得到. 分布式计算假设 N 个样本分布于 K 个子机器上,或者将全样本分成 K 个子数据集. 一般的做法是,在每台机器上利用本地数据进行建模和估计,然后将信息返回到中央机器上进行整合. 设第 k 台机器上的样本量为 n_k ,则 $\sum_{k=1}^{K} n_k = N$. 在第 2 节至第4 节中,记第 k 个子机器或者第 k 个子数据集为 $\mathscr{U}_k = \{(\boldsymbol{X}_{k,i}^{\mathrm{T}}, Y_{k,i})^{\mathrm{T}}: i = 1,2,\cdots,n_k\}$. 在第5 节中,由于考虑了更为复杂的非平衡半监督数据,因而记号有所改变,详见第 5 节的内容. 由于总是对给定的分位数水平 τ 进行分析,即 τ 是固定的,因而在不致产生歧义的前提下,为了记号简便,下文将统一用 $\boldsymbol{\theta}_0$ 来简记真值 $\boldsymbol{\theta}_0(\tau)$.

2 基于等度连续的通讯有效算法

在“分块集成”的设定下,假设针对第 k 个子机器或者第 k 个子数据集,有相同结构的目标函数 $\ell_{n_k}(\boldsymbol{\theta})$ 以及基于子机器上样本的估计 $\widehat{\boldsymbol{\theta}}_k$,那么由分位数回归理论可知

$$\widehat{\boldsymbol{\theta}}_k = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\, \ell_{n_k}(\boldsymbol{\theta})$$

$$\ell_{n_k}(\boldsymbol{\theta}) = \frac{1}{n_k}\sum_{i=1}^{n_k} \rho_\tau(Y_{k,i} - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i}) \tag{5}$$

由于目标函数 $\rho_\tau(\cdot)$ 并非光滑函数,不存在二阶以上的导数,因而不能直接使用泰勒展开的方式来获得高效通讯的分布式算法. 但是可以通过等度连续变换把目标函数转化为期望函数,此期望函数通常是高阶可导的,因而通过高阶泰勒展开其期望函数可以获得分位数回归的分布式高效通讯算法. 通过目标函数 $\ell_{n_k}(\boldsymbol{\theta})$ 求解 $\widehat{\boldsymbol{\theta}}_k$ 等价于极小化

$$\begin{aligned} U_{n_k}(\boldsymbol{\theta}) &= \ell_{n_k}(\boldsymbol{\theta}) - \ell_{n_k}(\widehat{\boldsymbol{\theta}}_k) \\ &= \frac{1}{n_k}\sum_{i=1}^{n_k}[\mathrm{E}\{\rho_\tau(Y_{k,i} - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i})\} - \\ &\quad \mathrm{E}\{\rho_\tau(Y_{k,i} - \widehat{\boldsymbol{\theta}}_k^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i})\}] + \\ &\quad R_1^k(\boldsymbol{\theta}) - R_1^k(\widehat{\boldsymbol{\theta}}_k) \end{aligned}$$

其中,余项 $R_1^k(\boldsymbol{\theta}) = \frac{1}{n_k}\sum_{i=1}^{n_k}[\rho_\tau(Y_{k,i} - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i}) - \mathrm{E}\{\rho_\tau(Y_{k,i} - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i})\}]$. 类似地,求解全样本最优解 $\widehat{\boldsymbol{\theta}}_N$ 等价于极小化

$$\begin{aligned} U_N(\boldsymbol{\theta}) &= \ell_N(\boldsymbol{\theta}) - \ell_N(\widehat{\boldsymbol{\theta}}_N) \\ &= \frac{1}{N}\sum_{i=1}^{N}[\mathrm{E}\{\rho_\tau(Y_i - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i)\} - \\ &\quad \mathrm{E}\{\rho_\tau(Y_i - \widehat{\boldsymbol{\theta}}_N^{\mathrm{T}}\vec{\boldsymbol{X}}_i)\}] + R_1(\boldsymbol{\theta}) - R_1(\widehat{\boldsymbol{\theta}}_N) \end{aligned}$$

其中 $R_1(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}[\rho_\tau(Y_i - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i) - \mathrm{E}\{\rho_\tau(Y_i - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i)\}]$.

直观上看,基于经验过程理论中等度连续的性质,不难想到在 N 和 n_k 都充分大时, $R_1(\boldsymbol{\theta})$, $R_1^k(\boldsymbol{\theta})$ 将变得足够的小. 事实上,可以给出余项 $R_1(\boldsymbol{\theta})$ 和 $R_1^k(\boldsymbol{\theta})$ 的收敛速度. 由经验过程理论,可以得到 $\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}_\delta} | R_1(\boldsymbol{\theta}) - R_1(\widehat{\boldsymbol{\theta}}_N) | = o_p(N^{-1})$ 和 $\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}_k} | R_1^k(\boldsymbol{\theta}) - R_1^k(\widehat{\boldsymbol{\theta}}_k) | = o_p(n_k^{-1})$,其中 $\boldsymbol{\Theta}_\delta = \{\boldsymbol{\theta}\in\boldsymbol{\Theta}: |\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_N| < CN^{-1/2}\}$, $\boldsymbol{\Theta}_k = \{\boldsymbol{\theta}\in\boldsymbol{\Theta}: |\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_k| < Cn_k^{-1/2}\}$. 注意到,这个收敛速度关于 $\boldsymbol{\theta}$ 是一致的,所以 $R_1(\widehat{\boldsymbol{\theta}}_N)$, $R_1^k(\widehat{\boldsymbol{\theta}}_k)$ 都将随 N 和 n_k 的增大而依概率收敛到零. 因此,优化 $U_N(\boldsymbol{\theta})$, $U_{n_k}(\boldsymbol{\theta})$ 的问题,可以近似等价于优化

$$\mathscr{L}_N(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\mathrm{E}\{\rho_\tau(Y_i - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i)\}$$

$$\mathscr{L}_{n_k}(\boldsymbol{\theta}) = \frac{1}{n_k}\sum_{i=1}^{n_k}\mathrm{E}\{\rho_\tau(Y_{k,i} - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i})\} \tag{6}$$

注意到,虽然 $\rho_\tau(Y_i - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i)$ 关于 $\rho_\tau(Y_i - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i)$ 并不可微,但因为 Y 给定 $\boldsymbol{X}$ 的条件下的条件分布函数 $F_{Y|\boldsymbol{X}}(y|\boldsymbol{X})$ 往往可微,因此可以认为 $\mathrm{E}\{\rho_\tau(Y_i - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i)\}$ 关于 $\boldsymbol{\theta}$ 可微,即 $\mathscr{L}_N(\boldsymbol{\theta})$, $\mathscr{L}_{n_k}(\boldsymbol{\theta})$ 关于 $\boldsymbol{\theta}$ 可微.

给定 $\boldsymbol{\theta}_0$ 的初值，即某一种初始化估计 $\bar{\boldsymbol{\theta}}$，例如 $\bar{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}_1$，意味着初值定为基于第一块子数据集极小化目标函数 $\ell_{n_1}(\boldsymbol{\theta})$ 得到的估计，由 $\mathscr{L}_N(\boldsymbol{\theta})$ 的泰勒展开可得

$$\mathscr{L}_N(\boldsymbol{\theta}) = \mathscr{L}_N(\bar{\boldsymbol{\theta}}) + \langle \nabla \mathscr{L}_N(\bar{\boldsymbol{\theta}}), \boldsymbol{\theta} - \bar{\boldsymbol{\theta}} \rangle + \sum_{j=2}^{\infty} \frac{1}{j!} \nabla^j \mathscr{L}_N(\bar{\boldsymbol{\theta}}) (\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})^{\otimes j} \quad (7)$$

当 N 和 n_k 都充分大时，上述展开中 $j \geqslant 2$ 的高阶项 $\nabla^j \mathscr{L}_N(\boldsymbol{\theta})$ 与对应子块的高阶项 $\nabla^j \mathscr{L}_{n_1}(\boldsymbol{\theta})$ 之间的差别也将是充分小的. 更精确地说，两者差的范数可控制在 $O_p(n_1^{-1})$ 内. 又因为 $\bar{\boldsymbol{\theta}} = \widehat{\boldsymbol{\theta}}_1$ 是利用子块数据集得到的一致估计，所以展开式又可以近似地等价于

$$\widetilde{\mathscr{L}}_N(\boldsymbol{\theta}) = \mathscr{L}_N(\bar{\boldsymbol{\theta}}) + \langle \nabla \mathscr{L}_N(\bar{\boldsymbol{\theta}}), \boldsymbol{\theta} - \bar{\boldsymbol{\theta}} \rangle + \sum_{j=2}^{\infty} \frac{1}{j!} \nabla^j \mathscr{L}_{n_1}(\bar{\boldsymbol{\theta}}) (\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})^{\otimes j} \quad (8)$$

另一方面，由 $\mathscr{L}_{n_1}(\boldsymbol{\theta})$ 的泰勒展开，又有

$$\mathscr{L}_{n_1}(\boldsymbol{\theta}) = \mathscr{L}_{n_1}(\bar{\boldsymbol{\theta}}) + \langle \nabla \mathscr{L}_{n_1}(\bar{\boldsymbol{\theta}}), \boldsymbol{\theta} - \bar{\boldsymbol{\theta}} \rangle + \sum_{j=2}^{\infty} \frac{1}{j!} \nabla^j \mathscr{L}_{n_1}(\bar{\boldsymbol{\theta}}) (\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})^{\otimes j} \quad (9)$$

上述式(8)和式(9)作差，并忽略常数项，则可以将式(8)中的近似目标函数 $\widetilde{\mathscr{L}}_N(\boldsymbol{\theta})$ 等价地写为新的目标函数

$$\widetilde{\mathscr{L}}_N(\boldsymbol{\theta}) = \mathscr{L}_{n_1}(\boldsymbol{\theta}) - \langle \boldsymbol{\theta}, \nabla \mathscr{L}_{n_1}(\bar{\boldsymbol{\theta}}) - \nabla \mathscr{L}_N(\bar{\boldsymbol{\theta}}) \rangle \quad (10)$$

这种方法受到了 Jordan 等[14]基于线性回归问题所讨论的代理损失函数思想的启发. 然而，Jordan 等[14]要求目标函数具有足够阶数的光滑性，而在分位数回归中，由于目标函数不可导，须利用经验过程中的等度连续理论，对目标函数进行近似.

进一步，下面给出 $\mathscr{L}_{n_1}(\boldsymbol{\theta})$，$\nabla \mathscr{L}_{n_1}(\boldsymbol{\theta})$ 与 $\nabla \mathscr{L}_N(\boldsymbol{\theta})$ 的具体表达式. 由大数定律可知，当 N，n_1 都足够大时，可以用相应的样本近似去代替式(10)中目标函数对应的项，即

$$\mathscr{L}_{n_1}(\boldsymbol{\theta}) \approx \ell_{n_1}(\boldsymbol{\theta})$$

$$\nabla \mathscr{L}_{n_1}(\boldsymbol{\theta}) \approx D\ell_{n_1}(\boldsymbol{\theta}), \nabla \mathscr{L}_N(\boldsymbol{\theta}) \approx D\ell_N(\boldsymbol{\theta})$$

其中 $D\ell_N(\boldsymbol{\theta})$ 和 $D\ell_{n_1}(\boldsymbol{\theta})$ 分别表示 $\ell_N(\boldsymbol{\theta})$ 和 $\ell_{n_1}(\boldsymbol{\theta})$ 在 $\boldsymbol{\theta}$ 处的一阶次导数，即

$$\ell_{n_k}(\boldsymbol{\theta}) = \frac{1}{n_k} \sum_{i=1}^{n_k} \rho_\tau(Y_{k,i} - \boldsymbol{\theta}^{\mathrm{T}} \vec{\boldsymbol{X}}_{k,i})$$

$$D\ell_{n_k}(\boldsymbol{\theta}) = -\frac{1}{n_k} \sum_{i=1}^{n_k} \vec{\boldsymbol{X}}_{k,i} \psi_\tau(Y_{k,i} - \boldsymbol{\theta}^{\mathrm{T}} \vec{\boldsymbol{X}}_{k,i})$$

$$D\ell_N(\boldsymbol{\theta}) = \frac{1}{N} \sum_{k=1}^{K} n_k D\ell_{n_k}(\boldsymbol{\theta})$$

而 $\psi_\tau(x) = \tau - I(x \leqslant 0)$ 为核验函数 $\rho_\tau(x)$ 的次导数. 注意到，直接采用次导数来近似 $\nabla \mathscr{L}_N(\boldsymbol{\theta})$，避免了对条件分布 $F_{Y|\boldsymbol{X}}(\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}} \mid \boldsymbol{X})$ 进行非参数估计，从而可以降低计算复杂度. 由此，最终的优化目标函数可写为

$$\widetilde{\mathscr{L}}_N(\boldsymbol{\theta}) = \ell_{n_1}(\boldsymbol{\theta}) - \langle \boldsymbol{\theta}, D\ell_{n_1}(\bar{\boldsymbol{\theta}}) - D\ell_N(\bar{\boldsymbol{\theta}}) \rangle \quad (11)$$

表 1　基于等度连续的通讯有效算法

Table 1 Communication-efficient algorithm based on equicontinuity

算法 1 基于等度连续的通讯有效算法	
1	初始化 $\boldsymbol{\theta}^{(0)} = \bar{\boldsymbol{\theta}}$ 和迭代循环次数 T；
2	for $t = 0, 1, \cdots, T-1$ do
3	传输当前迭代循环的 $\boldsymbol{\theta}^{(t)}$ 到 K 个机器 $\{\mathscr{M}_k\}_{k=1}^K$
4	for $k = 1, \cdots, K$ do
5	在子机器 $\mathscr{M}_k$ 上计算 $D\ell_{n_k}(\boldsymbol{\theta}^{(t)})$；
6	将 $(D\ell_{n_k}(\boldsymbol{\theta}^{(t)}), n_k)$ 传输到主机器 $\mathscr{M}_1$；
7	end for
8	在主机器 $\mathscr{M}_1$ 上计算 $D\ell_N(\boldsymbol{\theta}^{(t)}) = \frac{1}{N}\sum_{k=1}^{K} n_k \times D\ell_{n_k}(\boldsymbol{\theta}^{(t)})$，构建目标函数 $\widetilde{\mathscr{L}}_N^{1,(t)}(\boldsymbol{\theta}) = \ell_{n_1}(\boldsymbol{\theta}) - \langle \boldsymbol{\theta}, D\ell_{n_1}(\boldsymbol{\theta}^{(t)}) - D\ell_N(\boldsymbol{\theta}^{(t)}) \rangle$；
9	求解 $\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmin}} \widetilde{\mathscr{L}}_N^{1,(t)}(\boldsymbol{\theta})$；
10	end for
11	返回最终结果 $\widehat{\boldsymbol{\theta}}_{\mathrm{EC}} = \boldsymbol{\theta}^{(T)}$.

则 $\boldsymbol{\theta}_0$ 的估计量可以定义为 $\widehat{\boldsymbol{\theta}}_{\mathrm{EC}} = \underset{\boldsymbol{\theta} \in \Theta}{\operatorname{argmin}} \widetilde{\mathscr{L}}_N^1(\boldsymbol{\theta})$. 观察式(11)不难发现，如果该核心步骤的计算在第一台机器上运行，仅需要从第 k 台子机器传给第一台机器相应的梯度向量 $D\ell_{n_k}(\bar{\boldsymbol{\theta}})$，这将大大减少通讯交互的必要信息量. 基于此，可以提出第一种基于等度连续的通讯有效算法，见算法 1.

在算法 1 中，因为 $\boldsymbol{\theta}$ 为 $p+1$ 维向量，传输

($D\ell_{n_k}(\boldsymbol{\theta}^{(t)})$, n_k) 相当于传输 $p+2$ 个数值. 而采用一般“分块集成”的加权平均方法 (参见 Chen 等[10], Chen 和 Zhou[11]), 除了传输基于子数据集的估计 $\widehat{\boldsymbol{\theta}}_k$, 即一个 $p+1$ 维向量, 还需传输相应的 $p+1$ 维对称权重矩阵. 因此, 每台子机器至少需要向主机器传输 $p+1+(p+1)(p+2)/2$ 个数值. 可见, 新提出的算法确实能大大降低通讯成本. 特别地, 与 Chen 和 Zhou[11] 的方法相比, 新算法避免了估计条件密度函数 $E\{\vec{\boldsymbol{X}}\vec{\boldsymbol{X}}^{\mathrm{T}} f_{Y|X}(\boldsymbol{\theta}_0^{\mathrm{T}}\vec{\boldsymbol{X}}\mid \boldsymbol{X})\}$ 这个积分式, 也不需要传输此矩阵, 因此计算复杂度与通讯成本得以进一步降低. 同时还注意到, 在算法 1 中取 $T=1$ 时, $\boldsymbol{\theta}^{(T)}$ 对应了参数的一步 (one-shot) 估计; 而取 $K=1$ 时, 则自然地退化为不分块数据对应的直接估计.

理论上, 要求 N 和 n_k 需要满足如下假设: 存在常数 $\alpha\geqslant 1$ 使得 $N=O\{(\min_k n_k)^{\alpha}\}$. 此条件与 Chen 等[10] 的条件类似, 但是 Chen 等[10] 仅考虑了 $n_1=\cdots=n_k=n$ 的情形, 且涉及窗宽估计, 因此需要的条件更强. 本文给出的关于 N 和 n_k 的假设比很多分布式算法的条件要弱. 比如, “分块集成法” 一般要求 $n_k/\sqrt{N}\to\infty$, 即每台机器上的样本量 n_k 相比机器数 K 要足够大. Chen 和 Zhou[11] 要求存在常数 $\gamma\in[0,1/3)$ 使得 $K=O\{(\min_k n_k)^{\gamma}\}$. 这个条件显然也比本文所需的条件更强.

在实际应用中, 合适的样本量 n_k 需要实现统计估计效率与计算复杂度之间的平衡, 同时需要考虑实际的内存限制、通讯成本、计算时限等, 因此确定 n_k 并非易事. 一个基本的原则是 n_k 不能过小. 事实上, 虽然机器数量 K 越大, 所得估计量的方差越小, 但是当每台机器上的样本量 n_k 过少时, 会出现不可忽略偏差, 从而导致估计量的一致性 (也称为相合性) 无法保证. 此外, 为了保证对式 (10) 的近似达到足够的精度, 实际中要求 n_k 不能过小. 进一步, 可以给出如下的样本量 n_k 的选择策略: 在固定的计算时限、内存限制、通讯成本的基础上, 在满足一定的统计推断效率的前提下最小化计算时间, 从而实现数据驱动的 n_k 选择.

3 基于光滑逼近的通讯有效算法

再次关注到求解式 (11) 的优化问题, 式 (11) 的前半部分为第一台机器上的非光滑核验函数, 后半部分为关于 $\boldsymbol{\theta}$ 的线性函数, 这使得式 (11) 可以利用传统分位数回归的线性优化算法求解. 但是, 利用线性优化算法求解, 对内存空间的要求较高. 同时本文的方法希望在优化算法这个关键的步骤上有更高的计算效率. 因此, 本节将提出一种光滑化的方式进一步改良式 (11) 对应的优化问题. 从第 2 节的初步推导中, 可知求解通讯有效算法的关键是目标函数的可微性. 而由于示性函数 $I(\cdot)$ 不可微, 使得分位数回归模型的目标函数中的核验函数 $\rho_\tau(x)=x\{\tau-I(x\leqslant 0)\}$ 不可微. 第 2 节的解决方案是用次导数代替导数, 本节提出另一种方法, 采用可微函数对 $\rho_\tau(x)$ 进行逼近, 类似想法可参见 Chen 和 Hall[18].

令 $\phi_h(x)=x\{H(x/h)+\tau-1\}$, 其中 $h>0$ 为窗宽. 通过选取适当的函数 $H(x)$, 可以使得当 $h\to 0$ 时 $\phi_h(x)\to\rho_\tau(x)$, 例如,

$$H(x)=\begin{cases}0, & x\leqslant -1\\ \dfrac{1}{2}+\dfrac{15}{16}\left(x-\dfrac{2}{3}x^3+\dfrac{1}{5}x^5\right), & -1<x<1\\ 1, & x\geqslant 1\end{cases}\quad (12)$$

则 $\ell_{n_k}(\boldsymbol{\theta})$ 可用 $\ell_{n_k,h_k}(\boldsymbol{\theta})=\dfrac{1}{n_k}\sum_{i=1}^{n_k}\phi_{h_k}(Y_{k,i}-\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i})$ 来近似. 这里依旧采用同样的估计方法给出 $\boldsymbol{\theta}$ 的初始化估计 $\bar{\boldsymbol{\theta}}$. 例如, 基于第一块子数据集的估计 $\bar{\boldsymbol{\theta}}=\widehat{\boldsymbol{\theta}}_1^{h_1}$, 即

$$\bar{\boldsymbol{\theta}}=\widehat{\boldsymbol{\theta}}_1^{h_1}=\operatorname*{argmin}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\ell_{n_1,h_1}(\boldsymbol{\theta})$$

重复第 2 节中基于泰勒展开的推导, 类似地可以得到近似的优化目标函数

$$\widetilde{\mathscr{L}}_N^2(\boldsymbol{\theta})=\ell_{n_1,h_1}(\boldsymbol{\theta})-\langle\boldsymbol{\theta},\nabla\ell_{n_1,h_1}(\bar{\boldsymbol{\theta}})-\nabla\ell_{N,h}(\bar{\boldsymbol{\theta}})\rangle\quad (13)$$

其中

$$\nabla\ell_{n_k,h_k}(\boldsymbol{\theta})=-\frac{1}{n_k}\sum_{i=1}^{n_k}\zeta_{h_k}(Y_{k,i}-\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i})\vec{\boldsymbol{X}}_{k,i}$$

$$\nabla\ell_{N,h}(\boldsymbol{\theta})=\frac{1}{N}\sum_{k=1}^{K}n_k\nabla\ell_{n_k,h_k}(\boldsymbol{\theta})$$

$$\zeta_h(x)=H(x/h)+\tau-1+(x/h)H'(x/h)$$

$H'(\cdot)$ 为 $H(\cdot)$ 的导数. 由此可见, 近似后可微的

目标函数，已经具有了显式的微分表达. 因此可以运用传统的牛顿迭代法求解相应的优化问题，所得估计量为 $\widehat{\boldsymbol{\theta}}_{\mathrm{SJ1}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}} \widetilde{\mathscr{L}}_N^2(\boldsymbol{\theta})$.

表 2　基于光滑逼近的通讯有效算法

Table 2 Communication-efficient algorithm based on smooth approximation

算法 2 基于光滑逼近的通讯有效算法	
1	初始化 $\boldsymbol{\theta}^{(0)} = \bar{\boldsymbol{\theta}}$，迭代循环次数 T 和每台子机器的窗宽
2	for $t = 0,1,\cdots,T-1$ do
3	传输当前迭代循环的 $\boldsymbol{\theta}^{(t)}$ 到 K 个机器 $\{\mathscr{M}_k\}_{k=1}^K$
4	for $k = 1,\cdots,K$ do
5	在子机器 $\mathscr{M}_k$ 上计算 $\nabla \ell_{n_k,h_k}(\boldsymbol{\theta}^{(t)})$；
6	将 $(\nabla \ell_{n_k,h_k}(\boldsymbol{\theta}^{(t)}), n_k)$ 传输到主机器 $\mathscr{M}_1$；
7	end for
8	方法一：在主机器 $\mathscr{M}_1$ 上计算 $\nabla \ell_{N,h}(\boldsymbol{\theta}^{(t)}) = \frac{1}{N}\sum_{k=1}^K n_k \nabla \ell_{n_k,h_k}(\boldsymbol{\theta}^{(t)})$，构建目标函数 $\widetilde{\mathscr{L}}_N^{2,(t)}(\boldsymbol{\theta}) = \ell_{n_1,h_1}(\boldsymbol{\theta}) - \langle \boldsymbol{\theta}, \nabla \ell_{n_1,h_1}(\boldsymbol{\theta}^{(t)}) - \nabla \ell_{N,h}(\boldsymbol{\theta}^{(t)}) \rangle$；
9	求解并更新 $\boldsymbol{\theta}^{(t+1)} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}} \widetilde{\mathscr{L}}_N^{2,(t)}(\boldsymbol{\theta})$；
10	方法二：在主机器 $\mathscr{M}_1$ 上更新 $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \nabla^2 \ell_{n_1,h_1}(\boldsymbol{\theta}^{(t)})^{-1} \cdot \nabla \ell_{N,h}(\boldsymbol{\theta}^{(t)})$；
11	end for
12	返回最终结果 $\widehat{\boldsymbol{\theta}}_{SJ} = \boldsymbol{\theta}^{(T)}$；记方法一所得估计为 $\widehat{\boldsymbol{\theta}}_{SJ1}$，方法二所得估计为 $\widehat{\boldsymbol{\theta}}_{SJ2}$.

进一步，此优化问题可以通过一种更简洁的近似迭代来求解. 考虑式(13) 的泰勒展开，若仅取其中二阶及其以下的项有

$$\widetilde{\mathscr{L}}_N^2(\boldsymbol{\theta}) = \widetilde{\mathscr{L}}_N^2(\bar{\boldsymbol{\theta}}) + \langle \nabla \widetilde{\mathscr{L}}_N^2(\bar{\boldsymbol{\theta}}), \boldsymbol{\theta} - \bar{\boldsymbol{\theta}} \rangle + \frac{1}{2}(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})^{\mathrm{T}} \nabla^2 \widetilde{\mathscr{L}}_N^2(\bar{\boldsymbol{\theta}})(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}) \quad (14)$$

注意到 $\nabla \widetilde{\mathscr{L}}_N^2(\bar{\boldsymbol{\theta}}) = \nabla \ell_{N,h}(\bar{\boldsymbol{\theta}})$ 和 $\nabla^2 \widetilde{\mathscr{L}}_N^2(\bar{\boldsymbol{\theta}}) = \nabla^2 \ell_{n_1,h_1}(\bar{\boldsymbol{\theta}})$，其中

$$\nabla^2 \ell_{n_k,h_k}(\boldsymbol{\theta}) = \frac{1}{n_k}\sum_{i=1}^{n_k} \eta_{h_k}(Y_{k,i} - \boldsymbol{\theta}^{\mathrm{T}} \vec{\boldsymbol{X}}_{k,i}) \vec{\boldsymbol{X}}_{k,i} \vec{\boldsymbol{X}}_{k,i}^{\mathrm{T}}$$

$$\eta_h(x) = \frac{1}{h}\{2H'(x/h) + (x/h)H''(x/h)\}$$

在式(14) 中省去与优化不相关的常数项，可得新的近似目标函数

$$\widetilde{\mathscr{L}}_N^H(\boldsymbol{\theta}) = \langle \nabla \ell_{N,h}(\bar{\boldsymbol{\theta}}), \boldsymbol{\theta} - \bar{\boldsymbol{\theta}} \rangle + \frac{1}{2}(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})^{\mathrm{T}} \nabla^2 \ell_{n_1,h_1}(\bar{\boldsymbol{\theta}})(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}) \quad (15)$$

那么关于式(13) 的优化问题可近似转化为关于 $\boldsymbol{\theta}$ 的二次函数，即式 (15) 的 $\widetilde{\mathscr{L}}_N^{\mathrm{H}}(\boldsymbol{\theta})$ 的优化问题. 而这个问题的解存在显式表达式

$$\widehat{\boldsymbol{\theta}}_{\mathrm{SJ2}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}} \widetilde{\mathscr{L}}_N^{\mathrm{H}}(\boldsymbol{\theta}) = \bar{\boldsymbol{\theta}} - \nabla^2 \ell_{n_1,h_1}(\bar{\boldsymbol{\theta}})^{-1} \cdot \nabla \ell_{N,h}(\bar{\boldsymbol{\theta}})$$

由于分位数回归中的原始目标函数并不光滑，以上显式解中的 $\nabla^2 \ell_{n_1,h_1}(\bar{\boldsymbol{\theta}})$ 矩阵并不存在，所以无法直接套用此简化算法. 但现在已经对目标函数进行了光滑化处理，就能克服这个困难.

综上，$\boldsymbol{\theta}_0$ 的估计量可由两种方式得到，一种为直接优化求解出 $\widehat{\boldsymbol{\theta}}_{\mathrm{SJ1}}$. 另一种为显式解 $\widehat{\boldsymbol{\theta}}_{\mathrm{SJ2}}$. 基于此，可以提出基于光滑逼近的通讯有效算法，见算法 2.

这里还有个关键问题是如何选择窗宽. 窗宽有两种选择方式，一种是所有子机器使用相同的窗宽 ($h_1 = h_2 = \cdots = h_K$)，另一种是分别使用自己特定的窗宽. 在典型的光滑函数方法中，类似的窗宽选取并不像一些非参数估计那样，依据最小化均方误差来得出最优窗宽. 在光滑方法中仅需要考虑使得光滑化之后的函数与原函数之间的偏差能足够小，从而不影响到估计结果的统计性质. 换言之，只要能保证光滑函数合理工作的窗宽都是合适的窗宽. 这里采用拇指法则来获得光滑窗宽的选取方法，参见 Chen 等[10]，Li 和 Peng[19]. 特别是 Chen 等[10]同样也研究了分位数回归的分块光滑方法，其中对窗宽的选择给出了讨论. 但其提出的方法要求每一次迭代都更新窗宽，计算量较大. 基于这些相关工作，本文在模拟与实证部分会优先考虑 $O((p/n_k)^{1/4})$ 量级的窗宽.

算法 2 通过光滑逼近使得目标函数更易优化，且通过代理损失进一步近似目标函数，从而实现高效通讯. Chen 等[10]提出了基于“分块集成”思想的光滑逼近算法，该算法涉及到 $p+1$ 维权重矩阵的传输，会产生较高的通讯成本. 与基于“分块集成”思想的方法相比，本节提出的算法 2 在光滑逼近的基础上进一步通过高阶泰勒展开对光滑化的目标函数进行简化，构造代理损失函数，使

得优化步骤只需在第一台机器上运行. 相比于 Chen 等[10]的方法,本节由于不需要传输目标函数二阶导数矩阵,因而将不同机器间的通讯成本由 $O(Kp^2)$ 降为 $O(Kp)$,从而在降低计算复杂度和提高通讯效率方面更具有优势. 值得一提的是,虽然本文中给出的算法是针对分布式系统设计的,但是可以很容易地推广到流数据的情形. 此时,通讯有效算法可以大大降低数据的存储成本.

4 改进的数萃(Meta)方法

除了第 2 节和第 3 节中提出的算法之外,还有一种可以整合不同机器上的数据进行总体分析的方法,称为数萃(Meta)方法. 数萃(Meta)方法是将不同来源的信息进行综合而对总体参数进行统计推断的一类方法. 其中,基于置信分布(confidence distribution,CD)函数的数萃方法在分布式计算中是一种高效的分析手段. 但是,在统计学上考虑的数萃方法的分块数 K 通常是有限的,而对于分布式系统,K 可以随样本量趋于无穷.

在本节中,基于估计方程的思想,将提出一种改进的数萃方法. 该方法给出了分布式计算的一般框架,可以将经典数萃方法和许多现有的分布式算法涵盖在内. 首先,在第 k 台机器上,求解

$$D\mathscr{U}_{n_k}(\boldsymbol{\theta})=\frac{1}{n_k}\sum_{i=1}^{n_k}\psi_\tau(Y_{k,i}-\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i})\vec{\boldsymbol{X}}_{k,i}=\boldsymbol{0} \tag{16}$$

得到估计量 $\widehat{\boldsymbol{\theta}}_k$,此估计量称为 ALS 估计. 一般地,可以把 $\{\widehat{\boldsymbol{\theta}}_k\}_{k=1}^K$ 作为 K 个观察值,但可能具有不同的分布(通常有不同的方差). 这里采用估计方程的思想来组合不同机器上获得的估计值 $\{\widehat{\boldsymbol{\theta}}_k\}_{k=1}^K$,以获得最终结果. 在估计方程的思想中,最重要的是寻找到一个合适的无偏(或渐近无偏)的估计函数,具体可以见周勇[20]的第一章和第九章的内容. 考虑一类 p 维渐近无偏估计函数 $\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})$ 满足 $\mathrm{E}\,\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta}_0)\approx\boldsymbol{0}$. 由广义矩方法(generalized method of moments,GMM)可得如下 GMM 估计

$$\widehat{\boldsymbol{\theta}}_{\mathrm{M}}=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\left\{\sum_{m}^{K}\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})\right\} \tag{17}$$

$$=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\,\boldsymbol{\Psi}(\boldsymbol{\theta},\widehat{\boldsymbol{\Xi}})^{\mathrm{T}}\mathbb{W}(\widehat{\boldsymbol{\Xi}})\boldsymbol{\Psi}(\boldsymbol{\theta},\widehat{\boldsymbol{\Xi}}) \tag{18}$$

其中 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 是正定矩阵,通常依赖于第 k 组子样本. 为保证估计的一致性,一般要求 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 依概率收敛到样本无关的正定矩阵 $\boldsymbol{W}_k(\boldsymbol{\theta}_0)$. 此外,$\widehat{\boldsymbol{\Xi}}=\{\widehat{\boldsymbol{\theta}}_k\}_{k=1}^K$,$\widehat{\boldsymbol{\xi}}_k(\cdot,\boldsymbol{\theta})$ 为 $\boldsymbol{\xi}_k(\cdot,\boldsymbol{\theta})$ 在第 k 台机器上的估计,$\boldsymbol{\Psi}(\boldsymbol{\theta},\widehat{\boldsymbol{\Xi}})=\{\widehat{\boldsymbol{\xi}}_1(\widehat{\boldsymbol{\theta}}_1,\boldsymbol{\theta})^{\mathrm{T}},\cdots,\widehat{\boldsymbol{\xi}}_K(\widehat{\boldsymbol{\theta}}_K,\boldsymbol{\theta})^{\mathrm{T}}\}^{\mathrm{T}}$,$\mathbb{W}(\widehat{\boldsymbol{\Xi}})=\mathrm{Diag}\{\boldsymbol{W}_1(\widehat{\boldsymbol{\theta}}_1),\cdots,\boldsymbol{W}_K(\widehat{\boldsymbol{\theta}}_K)\}$ 为分块对角矩阵. 此外,若直接由式(18)出发,还可以考虑更一般的情形,即允许 $\widehat{\boldsymbol{\theta}}_k$ 之间存在相关性,例如下一节中考虑的非平衡半监督分布式系统. 此时,可在式(18)中选取 $\mathbb{W}(\widehat{\boldsymbol{\Xi}})$ 为 $\widehat{\boldsymbol{\Xi}}$ 的协方差矩阵的逆矩阵,其不一定具有分块对角的形式. 算法 3 给出了基于式(17)的迭代算法.

在算法 3 中,初值的选取可以更加简单,例如,可以选取由主机器 $\mathscr{M}_1$ 通过求解方程(16)得到的估计值 $\widehat{\boldsymbol{\theta}}_1$ 来作为初值. 此外,如果选取的权重矩阵与第 k 台机器的样本值无关,就不需要再进行迭代. 算法 3 给出了一般形式的改进的数萃算法. 但在具体问题中,由于一般都可以选取到一个简单可行的渐近无偏估计函数 $\widehat{\boldsymbol{\xi}}_k(\cdot,\boldsymbol{\theta})$ 及权重矩阵 $\boldsymbol{W}_k(\cdot)$,因此可以大大简化上面的算法,具体可参见算法 4 和算法 5.

在独立同分布的设定下,由经典 GMM 方法可考虑如下 GMM 估计

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_{\mathrm{GMM}}&=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\left\{\sum_{k=1}^{K}\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})\right\}^{\mathrm{T}}\boldsymbol{W}\left\{\sum_{k=1}^{K}\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})\right\}\\&=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\sum_{k_1,k_2=1}^{K}\widehat{\boldsymbol{\xi}}_{k_1}(\widehat{\boldsymbol{\theta}}_{k_1},\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{W}\widehat{\boldsymbol{\xi}}_{k_2}(\widehat{\boldsymbol{\theta}}_{k_2},\boldsymbol{\theta})\end{aligned}$$

其中 $\boldsymbol{W}$ 为正定的权重矩阵(此矩阵可以依赖于未知参数 $\boldsymbol{\theta}$). 若 $\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})$,$k=1,\cdots,K$ 相互独立,则

$$\mathrm{E}\{\widehat{\boldsymbol{\xi}}_{k_1}(\widehat{\boldsymbol{\theta}}_{k_1},\boldsymbol{\theta}_0)^{\mathrm{T}}\boldsymbol{W}\widehat{\boldsymbol{\xi}}_{k_2}(\widehat{\boldsymbol{\theta}}_{k_2},\boldsymbol{\theta}_0)\}\approx 0,k_1\neq k_2$$

因此,若令 $\widetilde{\boldsymbol{\theta}}_{\mathrm{GMM}}=\underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmin}}\sum_{k=1}^{K}\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{W}\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})$,则 $\widehat{\boldsymbol{\theta}}_{\mathrm{GMM}}$ 与 $\widetilde{\boldsymbol{\theta}}_{\mathrm{GMM}}$ 渐近等价. 本质上,经典 GMM 方法要求 $\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})$ 同分布,因此才可以对不同 k 选取一致的权重矩阵 $\boldsymbol{W}$. 而本节提出的方法允许 $\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})$ 具有不同的渐近分布,即允许不同机器上的数据存在异质性,因此需要考虑权重矩阵 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 与 k 相关. 由此可知,本节所提出的方法

比经典 GMM 方法具有更一般的适用性，可以适用于独立不同分布数据（式（17））乃至不独立不同分布数据（式（18））的情形. 下面先关注独立数据的分布式系统，而存在相关性的情形留在下一节半监督数据进行讨论. 对于独立数据的分布式系统，只需考虑式(17).

表 3　基于改进的数萃（Meta）方法的通讯有效算法

Table 3 Communication-efficient algorithm based on the improved Meta method

算法 3 基于改进的数萃（Meta）方法的通讯有效算法	
1	选取适当的估计函数 $\widehat{\boldsymbol{\xi}}_k(\cdot,\boldsymbol{\theta})$ 和权重矩阵函数 $\boldsymbol{W}_k(\cdot)$；
2	for $k=1,\cdots,K$ do
3	在机器 $\mathscr{M}_k$ 上计算 ALS 估计量 $\widehat{\boldsymbol{\theta}}_K$，即解方程 $D\mathscr{L}_{n_k}(\boldsymbol{\theta})=\frac{1}{n_k}\sum_{i=1}^{n_k}\psi_\tau(Y_{k,i}-\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i})\vec{\boldsymbol{X}}_{k,i}=\boldsymbol{0}$；
4	计算 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$，并将 $\widehat{\boldsymbol{\theta}}_k,\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 传输到主机器 $\mathscr{M}_1$ 上；
5	end for
6	在主机器 $\mathscr{M}_1$ 上计算 $\bar{\boldsymbol{\theta}}=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\left\{\sum_{k=1}^{K}\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})\right\}$；
7	初始化 $\boldsymbol{\theta}^{(0)}=\bar{\boldsymbol{\theta}}$ 和迭代循环次数 T；
8	for $t=0,1,\cdots,T-1$ do
9	传输当前迭代循环的 $\boldsymbol{\theta}^{(t)}$ 到 K 个机器 $\{\mathscr{M}_k\}_{k=1}^{K}$；
10	for $k=1,\cdots,K$ do
11	计算 $\boldsymbol{W}_k(\boldsymbol{\theta}^{(t)})$，将 $\boldsymbol{W}_k(\boldsymbol{\theta}^{(t)})$ 传输到主机器 $\mathscr{M}_1$；
12	end for
13	在主机器 $\mathscr{M}_1$ 上计算 $\boldsymbol{\theta}^{(t+1)}=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\left\{\sum_{k=1}^{K}\widehat{\boldsymbol{\xi}}_k(\boldsymbol{\theta}^{(t)},\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{W}_k(\boldsymbol{\theta}^{(t)})\widehat{\boldsymbol{\xi}}_k(\boldsymbol{\theta}^{(t)},\boldsymbol{\theta})\right\}$；
14	end for
15	返回最终结果 $\widehat{\boldsymbol{\theta}}_{\mathrm{M}}=\boldsymbol{\theta}^{(\mathrm{T})}$.

另外，可以寻找到更简单的渐近无偏估计函数，例如，可选取渐近无偏估计函数 $\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})=\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k$，就满足假设 $\mathrm{E}\,\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta}_0)\approx\boldsymbol{0}$. 则由式(17)的一阶条件，可得如下更简单的加权估计方程

$$\sum_{k=1}^{K}\nabla_{\boldsymbol{\theta}}\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})^{\mathrm{T}}\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})=\sum_{k=1}^{K}\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})=\boldsymbol{0}\qquad(19)$$

其中 $W_k(\widehat{\boldsymbol{\theta}}_k)$ 是一正定矩阵. 直接从式(19)出发，便可得到 $\boldsymbol{\theta}$ 的有效估计. 以上基于估计方程的思想给出了子机器估计量的一般融合方法，这些方法不仅对分位数回归成立，而且对一般损失下的最优解问题仍然成立. 下面分析在线性分位数损失下的具体应用.

对分位数回归，基于第 3 节中的光滑逼近法，考虑光滑化的目标函数 $\mathscr{L}_{n_k,h_k}(\boldsymbol{\theta})=\frac{1}{n_k}\sum_{i=1}^{n_k}\phi_{h_k}(Y_{k,i}-\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i})$，其中，$\phi_h(x)=x\{H(x/h)+\tau-1\}$，$h>0$ 为窗宽，$H(\cdot)$ 为第 3 节定义的光滑函数. 在一定条件下，可得 $\widehat{\boldsymbol{\theta}}_k$ 与 $\widetilde{\boldsymbol{\theta}}_k=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\mathscr{L}_{n_k,h_k}(\boldsymbol{\theta})$ 渐近等价，即可以等价地将对估计 $\widehat{\boldsymbol{\theta}}_k$ 和 $\widetilde{\boldsymbol{\theta}}_k$ 的整理相融合.

若考虑一阶条件，可得 $\mathrm{d}\,\phi_h(x)/\mathrm{d}x=H(x/h)+\tau-1+(x/h)H'(x/h)$. 在估计式（17）或式（19）中，权重矩阵可以根据算法的需要来选取. 选择不同的权重矩阵不会影响估计的一致性（也称为相合性）和渐近正态性. 这些性质仍然成立，而仅会影响其渐近方差. 根据估计方程中的 GMM 估计理论，如果 $\boldsymbol{W}_k$ 选取为渐近无偏估计函数的协方差矩阵的逆，则所获得的估计是有效估计. 但通常此 GMM 估计的渐近方差具有较复杂的三明治形式，需要重抽样的方法来估计. 在允许一些估计效率损失的情况下，可以选取方便计算的权重函数.

在式(19) 中，选取估计函数

$$\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})=\frac{1}{n_k}\sum_{i=1}^{n_k}\vec{\boldsymbol{X}}_{k,i}[H\{\Delta_{k,i}(\widehat{\boldsymbol{\theta}}_k;h)\}+\tau-1+\Delta_{k,i}(\boldsymbol{\theta};h)H'\{\Delta_{k,i}(\widehat{\boldsymbol{\theta}}_k;h)\}]\qquad(20)$$

其中 $\Delta_{k,i}(\boldsymbol{\theta};h)=(Y_{k,i}-\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i})/h$，$h>0$ 为窗宽，$H(\cdot)$ 为第 3 节定义的光滑函数. 令

$$\boldsymbol{U}_k(\widehat{\boldsymbol{\theta}}_k)=\frac{1}{n_k}\sum_{i=1}^{n_k}\vec{\boldsymbol{X}}_{k,i}[H\{\Delta_{k,i}(\widehat{\boldsymbol{\theta}}_k;h)\}+\tau-1+Y_{k,i}H'\{\Delta_{k,i}(\widehat{\boldsymbol{\theta}}_k;h)\}/h]$$

$$\boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)=\frac{1}{n_kh}\sum_{i=1}^{n_k}\vec{\boldsymbol{X}}_{k,i}\vec{\boldsymbol{X}}_{k,i}^{\mathrm{T}}H'\{\Delta_{k,i}(\widehat{\boldsymbol{\theta}}_k;h)\}\qquad(21)$$

则 $\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})=\boldsymbol{U}_k(\widehat{\boldsymbol{\theta}}_k)-\boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\boldsymbol{\theta}$. 注意到 $\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta}_0)\approx\boldsymbol{0}$，因此由式(17) 可得估计量

$$\widehat{\boldsymbol{\theta}}_{\mathrm{MS}} = \left\{\sum_{k=1}^{K} \boldsymbol{V}_k^{\mathrm{T}}(\widehat{\boldsymbol{\theta}}_k)\, \boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)\, \boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\right\}^{-1} \times \left\{\sum_{k=1}^{K} \boldsymbol{V}_k^{\mathrm{T}}(\widehat{\boldsymbol{\theta}}_k)\, \boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)\, \boldsymbol{U}_k(\widehat{\boldsymbol{\theta}}_k)\right\} \quad (22)$$

表 4　基于改进的数萃（Meta）方法的通讯有效算法

Table 4 Communication-efficient algorithm based on the improved Meta method

算法 4 基于改进的数萃（Meta）方法的通讯有效算法	
1	输入权重矩阵 $\boldsymbol{W}_k(\cdot)$，初始化 $\boldsymbol{\theta}^{(0)} = \bar{\boldsymbol{\theta}}$ 和迭代循环次数 T；
2	for $t = 0,1,\cdots,T-1$ do
3	传输当前迭代循环的 $\boldsymbol{\theta}^{(t)}$ 到 K 个机器 $\{\mathscr{M}_k\}_{k=1}^{K}$；
4	for $k = 1,\cdots,K$ do
5	在机器 $\mathscr{M}_k$ 上计算 $\boldsymbol{W}_k(\boldsymbol{\theta}^{(t)})$ 及 $\boldsymbol{U}_k(\boldsymbol{\theta}^{(t)}) = \frac{1}{n_k}\sum_{i=1}^{n_k} \vec{\boldsymbol{X}}_{k,i}[H\{\Delta_{k,i}(\boldsymbol{\theta}^{(t)};h)\} + \tau - 1 + Y_{k,i}H'\{\Delta_{k,i}(\boldsymbol{\theta}^{(t)};h)\}/h]$，$\boldsymbol{V}_k(\boldsymbol{\theta}^{(t)}) = \frac{1}{n_k}\sum_{i=1}^{n_k} \vec{\boldsymbol{X}}_{k,i}\vec{\boldsymbol{X}}_{k,i}^{\mathrm{T}}H'\{\Delta_{k,i}(\boldsymbol{\theta}^{(t)};h)\}$；
6	将 $\boldsymbol{W}_k(\boldsymbol{\theta}^{(t)})$，$\boldsymbol{U}_k(\boldsymbol{\theta}^{(t)})$，$\boldsymbol{V}_k(\boldsymbol{\theta}^{(t)})$ 传输到主机器 $\mathscr{M}_1$；
7	end for
8	在主机器 $\mathscr{M}_1$ 上计算 $\boldsymbol{\theta}^{(t+1)} = \left\{\sum_{k=1}^{K} \boldsymbol{V}_k^{\mathrm{T}}(\boldsymbol{\theta}^{(t)})\, \boldsymbol{W}_k(\boldsymbol{\theta}^{(t)})\, \boldsymbol{V}_k(\boldsymbol{\theta}^{(t)})\right\}^{-1}\left\{\sum_{k=1}^{K} \boldsymbol{V}_k^{\mathrm{T}}(\boldsymbol{\theta}^{(t)})\, \boldsymbol{W}_k(\boldsymbol{\theta}^{(t)})\, \boldsymbol{U}_k(\boldsymbol{\theta}^{(t)})\right\}$；
9	end for
10	返回最终结果 $\widehat{\boldsymbol{\theta}}_{\mathrm{MS}} = \boldsymbol{\theta}^{(\mathrm{T})}$.

算法 4 给出了此估计的迭代算法.

在算法 4 中，如果选取 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k) = \boldsymbol{V}_k^{-1}(\widehat{\boldsymbol{\theta}}_k)$，则可得如下估计量

$$\widehat{\boldsymbol{\theta}}_{\mathrm{ML}} = \left\{\sum_{k=1}^{K} \boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\right\}^{-1}\left\{\sum_{k=1}^{K} \boldsymbol{U}_k(\widehat{\boldsymbol{\theta}}_k)\right\} \quad (23)$$

事实上，Chen 等[10]中的估计就具有式(23) 的形式. 因此，Chen 等[10]的方法可以看作本文提出的改进的数萃方法的特例. 此外，由式 (19) 出发，还可以选取估计函数为 $\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta}) = \boldsymbol{U}_k(\widehat{\boldsymbol{\theta}}_k) - \boldsymbol{V}\boldsymbol{D}_k(\widehat{\boldsymbol{\theta}}_k)\boldsymbol{\theta}$，且选取权重矩阵 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 为单位阵，也能得式 (23) 中的估计量.

若由式(17) 出发，且选取估计函数 $\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta}) = \boldsymbol{U}_k(\widehat{\boldsymbol{\theta}}_k) - \boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\boldsymbol{\theta}$，上述算法 4 所选取的权重矩阵 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k) = \boldsymbol{V}_k^{-1}(\widehat{\boldsymbol{\theta}}_k)$ 有时会增加计算复杂度. 为了快速计算，一种方便的取法是选取权重矩阵 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 为单位阵 $\boldsymbol{I}_{p+1}$，很容易得到 $\widehat{\boldsymbol{\theta}}_{\mathrm{M}}$ 的另一种显式估计表达式

$$\widehat{\boldsymbol{\theta}}_{\mathrm{MV}} = \left\{\sum_{k=1}^{K} \boldsymbol{V}_k^{\mathrm{T}}(\widehat{\boldsymbol{\theta}}_k)\, \boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\right\}^{-1} \times \left\{\sum_{k=1}^{K} \boldsymbol{V}_k^{\mathrm{T}}(\widehat{\boldsymbol{\theta}}_k)\, \boldsymbol{U}_k(\widehat{\boldsymbol{\theta}}_k)\right\} \quad (24)$$

这个估计由于选取了简单的单位矩阵作为权重，根据估计方程理论，其估计效率将有一定损失②，但可以大大降低计算复杂度. 这个结论在数值模拟中也可以看出，且当不同机器的数据存在异质性时，其表现与已有文献中的方法相比并不差.

进一步，令 $\widehat{\varepsilon}_{k,i} = Y_{k,i} - \widehat{\boldsymbol{\theta}}_k^{\mathrm{T}}\vec{\boldsymbol{X}}_{k,i}$ 和

$$\boldsymbol{A}_k(\widehat{\boldsymbol{\theta}}_k) = \frac{1}{n_k}\sum_{i=1}^{n_k} \vec{\boldsymbol{X}}_{k,i}[H\{\Delta_{k,i}(\widehat{\boldsymbol{\theta}}_k;h)\} + \tau - 1 + \widehat{\boldsymbol{\varepsilon}}_{k,i}H'\{\varepsilon_{k,i}(\widehat{\boldsymbol{\theta}}_k;h)\}/h] \quad (25)$$

则可得 $\widehat{\boldsymbol{\theta}}_{\mathrm{MV}} - \boldsymbol{\theta}_0 \approx \left\{\sum_{k=1}^{K} \boldsymbol{V}_k^{\mathrm{T}}(\widehat{\boldsymbol{\theta}}_k)\, \boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\right\}^{-1} \times \left\{\sum_{k=1}^{K} \boldsymbol{V}_k^{\mathrm{T}}(\widehat{\boldsymbol{\theta}}_k)\, \boldsymbol{A}_k(\widehat{\boldsymbol{\theta}}_k)\right\}$. 由此出发可以得到其渐近协方差矩阵的估计

$$\widetilde{\boldsymbol{\Sigma}}_{\mathrm{MV}} = \left\{\sum_{k=1}^{K} \boldsymbol{V}_k^{\mathrm{T}}(\widehat{\boldsymbol{\theta}}_k)\, \boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\right\}^{-1} \times \left\{\sum_{k=1}^{K} \boldsymbol{V}_k^{\mathrm{T}}(\widehat{\boldsymbol{\theta}}_k)\, \widetilde{\boldsymbol{D}}_{0k}(\widehat{\boldsymbol{\theta}}_k)\, \boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\right\} \times \left\{\sum_{k=1}^{K} \boldsymbol{V}_k^{\mathrm{T}}(\widehat{\boldsymbol{\theta}}_k)\, \boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\right\}^{-1}$$

其中 $\widetilde{\boldsymbol{D}}_{0k}(\widehat{\boldsymbol{\theta}}_k)$ 由 $\boldsymbol{A}_k(\widehat{\boldsymbol{\theta}}_k)$ 的渐近协方差矩阵导出，定义为

$$\widetilde{\boldsymbol{D}}_{0k}(\widehat{\boldsymbol{\theta}}_k) = \frac{1}{n_k}\sum_{i=1}^{n_k} (\vec{\boldsymbol{X}}_{k,i}[H\{\Delta_{k,i}(\widehat{\boldsymbol{\theta}}_k;h)\} + \tau - 1 + \widehat{\varepsilon}_{k,i}H'\{\Delta_{k,i}(\widehat{\boldsymbol{\theta}}_k;h)\}/h])^{\otimes 2} \quad (26)$$

其中对于一个向量 $\boldsymbol{a}$，定义 $\boldsymbol{a}^{\otimes 2} = \boldsymbol{a}\boldsymbol{a}^{\mathrm{T}}$.

下面给出基于上述算法导出的一种通讯有效算法. 注意到，以上方法需要从第 k 台机器传输 $p+1$ 维矩阵 $\boldsymbol{V}_k$ 和向量 $\boldsymbol{U}_k$ 到第一台机器，通讯成本较高. 当不同机器上的数据分布相同，即数据同质时，可以对以上算法提出改进，使之具有通讯有效性. 特别地，为了降低通讯成本，可以分别利用第一台机器的数据和第 k 台机器的估计量 $\widehat{\boldsymbol{\theta}}_k$，通

② 估计效率是由估计的方差大小来称量，方差越小效率越高.

过 $\boldsymbol{V}_1(\widehat{\boldsymbol{\theta}}_k)$ 对 $\boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)$ 进行近似逼近，直接由 $\boldsymbol{V}_1(\widehat{\boldsymbol{\theta}}_k)$ 代替 $\boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)$，以此节约不同机器向主机器传输矩阵的通讯成本. 虽然由于使用的信息少了很多，因此估计效率将会降低，但此种方法可以大大节约通讯成本. 因此在不太注重估计效率的情况下，此种方法仍然有较大的价值，特别在协变量维数较高的同质数据下，是一种非常有效的方法.

类似地，在式(19) 中，只需要 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 正定，就能保证估计 $\widehat{\boldsymbol{\theta}}_M$是一致估计(也称为相合估计). 为了提高通讯效率与计算效率，可以允许损失一些估计效率，通过简单且易于计算的 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 获得一个快速估计. 特别地，只要 $\widehat{\boldsymbol{\theta}}_k$是$\boldsymbol{\theta}_0$的一致估计(相合估计)，那么总是可以构造估计函数 $\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k, \boldsymbol{\theta}) = \boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_k$，则可得显式表达式

$$\widehat{\boldsymbol{\theta}}_{\mathrm{MW}} = \left\{\sum_{k=1}^{K} \boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)\right\}^{-1}\left\{\sum_{k=1}^{K} \boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)\,\widehat{\boldsymbol{\theta}}_k\right\} \tag{27}$$

式中，若选取权重矩阵 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 为与参数无关的 $p+1$ 维单位矩阵 $\boldsymbol{I}_{p+1}$和 $\boldsymbol{n}_k$的乘积时（注意到此时允许每台机器上的样本量是不等的，第 k 台机器的样本量为 n_k)，则

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_{\mathrm{AM}} &\approx \left(\sum_{k=1}^{K} n_k \boldsymbol{I}_{p+1}\right)^{-1}\left(\sum_{k=1}^{K} n_k \boldsymbol{I}_{p+1}\,\widehat{\boldsymbol{\theta}}_k\right)\\ &= \left(\sum_{k=1}^{K} n_k\right)^{-1}\left(\sum_{k=1}^{K} n_k\,\widehat{\boldsymbol{\theta}}_k\right)\end{aligned} \tag{28}$$

如果每台机器上的样本量相等，即 $n_1 = \cdots = n_K$，则 $\widehat{\boldsymbol{\theta}}_{\mathrm{AM}}$为简单平均估计量. 如果每台机器的权重矩阵 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 满足，存在与 k 无关的常矩阵 $\boldsymbol{W}_0$使得 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k) \approx n_k \boldsymbol{W}_0$，例如当 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)/n_k$的收敛值与 k 无关时，不同的$\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 所得的估计均与以上估计渐近等价. 特别地，若 $\widehat{\boldsymbol{\theta}}_k$的渐近分布相同，则以上估计就是有效估计. 此时，就可以选取 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k) = \boldsymbol{I}_{\boldsymbol{p}+1}$ 或者 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k) = n_k \widehat{E}_k(\vec{\boldsymbol{X}}\vec{\boldsymbol{X}}^{\mathrm{T}}) = \sum_{i=1}^{n_k} \vec{\boldsymbol{X}}_{k,i}\vec{\boldsymbol{X}}_{k,i}^{\mathrm{T}}$，假如 Wang 等[21]，在 $\widehat{\boldsymbol{\theta}}_k$渐近独立同分布时均可得有效估计.

然而，若 $\widehat{\boldsymbol{\theta}}_k$的渐近分布不同，最优权重矩阵应选取为 $\widehat{\boldsymbol{\theta}}_k$的协方差矩阵的逆矩阵. 在分位数回归中，这等价于设定 $\widehat{\boldsymbol{\theta}}_k$有相同的渐近均值和不同的渐近协方差矩阵，即 $\sqrt{n_k}(\widehat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_0) \xrightarrow{d} \mathcal{N}(0, \boldsymbol{Q}_k(\boldsymbol{\theta}_0))$，其中渐近协方差矩阵 $\boldsymbol{Q}_k(\boldsymbol{\theta}_0) = \boldsymbol{D}_{1k}^{-1}(\boldsymbol{\theta}_0)\boldsymbol{D}_{0k}(\boldsymbol{\theta}_0)\boldsymbol{D}_{1k}^{-1}(\boldsymbol{\theta}_0)$，而$\boldsymbol{D}_{0k}(\boldsymbol{\theta}) = \mathrm{E}[\{\tau^2 + (1-2\tau)F_{Y_k|\boldsymbol{X}_k}(\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_k \mid \boldsymbol{X}_k)\}\vec{\boldsymbol{X}}_k\vec{\boldsymbol{X}}_k^{\mathrm{T}}]$，$\boldsymbol{D}_{1k}(\boldsymbol{\theta}) = \mathrm{E}\{\vec{\boldsymbol{X}}_k\vec{\boldsymbol{X}}_k^{\mathrm{T}} f_{Y_k|\boldsymbol{X}_k}(\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_k \mid \boldsymbol{X}_k)\}$，并且 $F_{Y_k|\boldsymbol{X}_k}(y \mid \boldsymbol{X}_k)$ 与 $f_{Y_k|\boldsymbol{X}_k}(y \mid \boldsymbol{X}_k)$ 分别表示在给定 $\boldsymbol{X}_k$的条件下 Y_k的条件分布函数与条件密度函数. 实际中，这类数据分块的形式广泛存在. 特别地，在多中心大数据获取中，由于数据来源不同、数据收集方式不同等原因，不同机器上数据的协变量 $\boldsymbol{X}$ 的分布常常具有不同程度的差异性，就会出现以上所述的 $\widehat{\boldsymbol{\theta}}_k$有相同的渐近均值和不同的协方差矩阵的情况. 进一步，若考虑不同机器上的模型也可以不相同，则 Y 的分布也可能存在差异，此时数据的异质性会更加明显，从而将协方差矩阵的逆矩阵作为权重矩阵所具有的估计效率优势得以体现.

此时，一个关键的问题是如何得到协方差矩阵的估计$\widehat{\boldsymbol{Q}}_k(\widehat{\boldsymbol{\theta}}_k)$. 这并非易事，尤其还需考虑分布式系统下的通讯成本与计算成本. 一种方法是对 $\boldsymbol{D}_{0k}(\boldsymbol{\theta}_0)$ 和 $\boldsymbol{D}_{1k}(\boldsymbol{\theta}_0)$ 进行直接估计. 对于 $\boldsymbol{D}_{1k}(\boldsymbol{\theta}_0)$ 的表达式中所涉及的条件密度函数 $f_{Y_k|\boldsymbol{X}_k}(\cdot \mid \boldsymbol{X}_k)$，可以采用非参数核平滑方法进行估计，具体可以参见 Hall 等[22]. 然而，这种方法需要的计算量较大. 另一种方法是利用第 3 节中提出的光滑化方法，直接对 $\widehat{\boldsymbol{\theta}}_k$ 的渐近协方差矩阵进行等价表示. 令 $\widetilde{\boldsymbol{\theta}}_k = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}} \ell_{n_k,h_k}(\boldsymbol{\theta})$ 为第 k 台机器上的光滑化估计. 则基于第 3 节中的光滑逼近法，在一定条件下，可得 $\widehat{\boldsymbol{\theta}}_k$ 与 $\widetilde{\boldsymbol{\theta}}_k$ 渐近等价. 则 $\widehat{\boldsymbol{\theta}}_k$ 满足展开 $\widehat{\boldsymbol{\theta}}_k - \boldsymbol{\theta}_0 \approx \boldsymbol{V}_k^{-1}(\widehat{\boldsymbol{\theta}}_k)\,\boldsymbol{A}_k(\widehat{\boldsymbol{\theta}}_k)$，其中，$\boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)$ 与$\boldsymbol{A}_k(\widehat{\boldsymbol{\theta}}_k)$ 分别由式 (21) 与式 (25) 定义. 则 $\widehat{\boldsymbol{\theta}}_k$ 的渐近协方差矩阵的一个估计可以选取为

$$\widehat{\boldsymbol{Q}}_k(\widehat{\boldsymbol{\theta}}_k) = \boldsymbol{V}_k^{-1}(\widehat{\boldsymbol{\theta}}_k)\,\widetilde{\boldsymbol{D}}_{0k}(\widehat{\boldsymbol{\theta}}_k)\,\boldsymbol{V}_k^{-1}(\widehat{\boldsymbol{\theta}}_k)$$

其中$\widetilde{\boldsymbol{D}}_{0k}(\widehat{\boldsymbol{\theta}}_k)$ 由式 (26) 定义. 为了减少矩阵求逆运算的次数，可直接计算最优权重矩阵

$$\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k) = n_k\,\widehat{\boldsymbol{Q}}_k^{-1}(\widehat{\boldsymbol{\theta}}_k) = n_k\,\boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\,\widetilde{\boldsymbol{D}}_{0k}^{-1}(\widehat{\boldsymbol{\theta}}_k)\,\boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)$$

为了便于比较，记基于此权重矩阵和渐近无偏估计函数$\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k, \boldsymbol{\theta}) = \boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_k$的估计量为 $\widehat{\boldsymbol{\theta}}_{SP}$，相应的迭代算法 3 可以简化为算法 5.

由于涉及矩阵传输，以上方法的通讯成本较高. 与算法 4 同理，当数据同质时，为了降低通讯

成本,还可以基于第 k 台机器的估计 $\widehat{\boldsymbol{\theta}}_k$ 与第一台机器上的数据来计算估计,选取 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k) = n_k \widetilde{\boldsymbol{D}}_{11}(\widehat{\boldsymbol{\theta}}_k) \widetilde{\boldsymbol{D}}_{01}^{-1}(\widehat{\boldsymbol{\theta}}_k) \widetilde{\boldsymbol{D}}_{11}(\widehat{\boldsymbol{\theta}}_k)$,其中

$$\widetilde{\boldsymbol{D}}_{01}(\widehat{\boldsymbol{\theta}}_k) = \frac{1}{n_1}\sum_{i=1}^{n_1}\left[\vec{\boldsymbol{X}}_{1,i}\left\{H\left(\frac{Y_{1,i}-\widehat{\boldsymbol{\theta}}_k^{\mathrm{T}}\vec{\boldsymbol{X}}_{1,i}}{h}\right)+\tau-1+\frac{\widehat{\varepsilon}_{1,i}}{h}H'\left(\frac{Y_{1,i}-\widehat{\boldsymbol{\theta}}_k^{\mathrm{T}}\vec{\boldsymbol{X}}_{1,i}}{h}\right)\right\}\right]^{\otimes 2}$$

$$\widetilde{\boldsymbol{D}}_{11}(\widehat{\boldsymbol{\theta}}_k) = \frac{1}{n_1 h}\sum_{i=1}^{n_1}\vec{\boldsymbol{X}}_{k,i}\vec{\boldsymbol{X}}_{k,i}^{\mathrm{T}}H'\left(\frac{\widehat{\varepsilon}_{1,i}}{h}\right)$$

且 $\hat{\varepsilon}_{1,i} = Y_{1,i} - \widehat{\boldsymbol{\theta}}_k^{\mathrm{T}}\vec{\boldsymbol{X}}_{1,i}$. 从而获得一个不需要传输矩阵的权重矩阵定义

$\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k) = n_k \widehat{\boldsymbol{Q}}_1^{-1}(\widehat{\boldsymbol{\theta}}_k) = n_k \boldsymbol{V}_1(\widehat{\boldsymbol{\theta}}_k)\widetilde{\boldsymbol{D}}_{01}^{-1}(\widehat{\boldsymbol{\theta}}_k)\boldsymbol{V}_1(\widehat{\boldsymbol{\theta}}_k)$. 估计量由式(27)给出. 记基于上述方法得到的估计为 $\widehat{\boldsymbol{\theta}}_{\mathrm{SP-Local}}$.

另外,估计权重矩阵还可以采用重抽样的方法,使用 Zeng 和 Lin[23] 中的 LS 方法来估计 $\boldsymbol{D}_{0k}(\boldsymbol{\theta}_0)$ 以及 $\boldsymbol{D}_{1k}(\boldsymbol{\theta}_0)$,在第 k 台机器上计算 $\widehat{\boldsymbol{D}}_{1k}(\widehat{\boldsymbol{\theta}}_k)$, $\widehat{\boldsymbol{D}}_{0k}(\widehat{\boldsymbol{\theta}}_k)$ 与渐近协方差矩阵的估计 $\widehat{\boldsymbol{Q}}_k(\widehat{\boldsymbol{\theta}}_k) = \widehat{\boldsymbol{D}}_{1k}^{-1}(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{D}}_{0k}(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{D}}_{1k}^{-1}(\widehat{\boldsymbol{\theta}}_k)$. 然而,这种方法需要传输 $p+1$ 维矩阵 $\widehat{\boldsymbol{Q}}_k(\widehat{\boldsymbol{\theta}}_k)$,包含 $(p+1)(p+2)/2$ 个参数,如果把所有 $k \geqslant 2$ 台子机器上的协方差矩阵都传输到第一台机器将涉及 $O(Kp^2)$ 个参数,当维数 p 较高时,这将大大增加通讯成本. 因此,当不同机器上的数据分布相同时,可以利用第一台机器上的数据和第 k 台机器上的参数估计对协方差矩阵进行近似. 将第 k 台子机器上的参数估计 $\widehat{\boldsymbol{\theta}}_k$ 传输到第一台机器,基于第一台机器上的数据与 $\widehat{\boldsymbol{\theta}}_k$,估计 $\widehat{\boldsymbol{D}}_{11}(\widehat{\boldsymbol{\theta}}_k)$ 和 $\widehat{\boldsymbol{D}}_{01}(\widehat{\boldsymbol{\theta}}_k)$ 分别代替 $\widehat{\boldsymbol{D}}_{1k}(\widehat{\boldsymbol{\theta}}_k)$ 和 $\widehat{\boldsymbol{D}}_{0k}(\widehat{\boldsymbol{\theta}}_k)$. 具体而言,令 $\boldsymbol{S}_{1,i}(\boldsymbol{\theta}) = \{I(Y_{1,i} \leqslant \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_{1,i}) - \tau\}\vec{\boldsymbol{X}}_{1,i}$. 对 $k = 1,\cdots,K$,令 $\widehat{\boldsymbol{D}}_{01}(\widehat{\boldsymbol{\theta}}_k) = n_1^{-1}\sum_{i=1}^{n_1}\boldsymbol{S}_{1,i}(\widehat{\boldsymbol{\theta}}_k)\boldsymbol{S}_{1,i}^{\mathrm{T}}(\widehat{\boldsymbol{\theta}}_k)$. 生成 B 个独立同分布随机向量 $\{\boldsymbol{Z}_{k,b}\}_{b=1}^{B}$(这里 B 的取值可以根据实际情况来选取,一般在 200 以上效果就很好),使得 $\boldsymbol{Z}_{k,b} \sim \mathcal{N}(\boldsymbol{0}, \widehat{\boldsymbol{D}}_{01}^{-1}(\widehat{\boldsymbol{\theta}}_k))$. 基于式(16),计算 $\{n_1^{1/2}\boldsymbol{D}\, l_{n_1(\widehat{\boldsymbol{\theta}}_k + n_1^{-1/2}\boldsymbol{Z}_{k,b})}\}_{b=1}^{B}$ 并将其作为响应变量对 $\{\boldsymbol{Z}_{k,b}\}_{b=1}^{B}$ 做回归,由此得到的回归系数即为 $\widehat{\boldsymbol{D}}_{11}(\widehat{\boldsymbol{\theta}}_k)$. 选取 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k) = n_k \widehat{\boldsymbol{D}}_{11}(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{D}}_{01}^{-1}(\widehat{\boldsymbol{\theta}}_k)\boldsymbol{D}_{11}(\widehat{\boldsymbol{\theta}}_k)$. 需要指出的是,虽然此方法的计算成本仅是线性的,但是由于需要进行重抽样,且当维数 p 较大时,涉及多个高维矩阵的逆运算,仍然具有较大的计算复杂度. 但其可以保证在数据不同分布时仍能得到有效估计. 因此,这里还是将此方法列出,作为一种可行的估计方法. 将基于重抽样方法获得权重矩阵估计并应用算法 5 获得的参数 $\boldsymbol{\theta}_0$ 的估计量记为 $\widehat{\boldsymbol{\theta}}_{\mathrm{MB}}$,将仅传输参数估计量 $\widehat{\boldsymbol{\theta}}_k$ 到第一台主机器,并在主机器上得出近似的 $\widehat{\boldsymbol{Q}}_1(\widehat{\boldsymbol{\theta}}_k)$,而得到的估计量记为 $\widehat{\boldsymbol{\theta}}_{\mathrm{MB-Local}}$.

表 5 基于改进的数萃(Meta)方法的通讯有效算法

Table 5 Communication-efficient algorithm based on the improved Meta method

算法 5 基于改进的数萃(Meta)方法的通讯有效算法	
1	输入权重矩阵 $\boldsymbol{W}_k(\cdot)$,初始化 $\boldsymbol{\theta}^{(0)} = \bar{\boldsymbol{\theta}}$ 和迭代循环次数 T;
2	for $t = 0,1,\cdots,T-1$ do
3	传输当前迭代循环的 $\boldsymbol{\theta}^{(t)}$ 到 K 个机器 $\{\mathcal{M}_k\}_{k=1}^{K}$;
4	for $k = 1,\cdots,K$ do
5	在机器 $\mathcal{M}_k$ 上计算 $\boldsymbol{V}_k(\boldsymbol{\theta}^{(t)}) = \frac{1}{n_k}\sum_{i=1}^{n_k}\vec{\boldsymbol{X}}_{k,i}\vec{\boldsymbol{X}}_{k,i}^{\mathrm{T}}H'\{\Delta_{k,i}(\boldsymbol{\theta}^{(t)};h)\}$, $\widetilde{\boldsymbol{D}}_{0k}(\boldsymbol{\theta}^{(t)}) = \frac{1}{n_k}\sum_{i=1}^{n_k}(\vec{\boldsymbol{X}}_{1,i}[H\{\Delta_{k,i}(\boldsymbol{\theta}^{(t)};h)\} + \tau - 1 + \widehat{\varepsilon}_{1,i}H'\{\Delta_{k,i}(\boldsymbol{\theta}^{(t)};h)\}])^{\otimes 2}$, $\boldsymbol{W}_k(\boldsymbol{\theta}^{(t)}) = n_k \boldsymbol{V}_k(\boldsymbol{\theta}^{(t)})\widetilde{\boldsymbol{D}}_{0k}^{-1}(\boldsymbol{\theta}^{(t)})\boldsymbol{V}_k(\boldsymbol{\theta}^{(t)})$.
6	将 $\boldsymbol{W}_k(\boldsymbol{\theta}^{(t)}), \boldsymbol{U}_k(\boldsymbol{\theta}^{(t)}), \boldsymbol{V}_k(\boldsymbol{\theta}^{(t)})$ 传输到主机器 $\mathcal{M}_1$;
7	end for
8	在主机器 $\mathcal{M}_1$ 上计算 $\boldsymbol{\theta}^{(t+1)} = \left\{\sum_{k=1}^{K}\boldsymbol{W}_k(\boldsymbol{\theta}^{(t)})\right\}^{-1}\left\{\sum_{k=1}^{K}\boldsymbol{W}_k(\boldsymbol{\theta}^{(t)})\boldsymbol{\theta}^{(t)}\right\}$;
9	end for
10	返回最终结果 $\widehat{\boldsymbol{\theta}}_{\mathrm{SP}} = \boldsymbol{\theta}^{(T)}$.

事实上,改进的数萃方法可以统一许多现有的方法. 例如,在不同机器上的数据独立同分布的设定下,Chen 等[10] 的光滑化方法,Chen 和 Zhou[11] 的分块集成法,Wang 等[20] 的针对线性分位数回归模型的方法等. 虽然切入点不同,但是最终都可以化为式(17)的形式. 在式(17)的框架下,Chen 等[10] 对应选取了估计函数 $\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k, \boldsymbol{\theta}) = \boldsymbol{U}_k(\widehat{\boldsymbol{\theta}}_k) - \boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\boldsymbol{\theta}$ 和权重矩阵 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k) = \boldsymbol{V}_k^{-1}(\widehat{\boldsymbol{\theta}}_k)$, Chen 和 Zhou[11] 与 Wang 等[21] 都可视为选取了估计函数 $\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k, \boldsymbol{\theta}) = \boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_k$,不同之处

仅在于两种方法的权重矩阵 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 不同. 具体而言,当 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)=n_k\widehat{\boldsymbol{D}}_{1k}(\widehat{\boldsymbol{\theta}}_k)$ 时,则估计 $\widehat{\boldsymbol{\theta}}_{\mathrm{M}}$ 化为 Chen 和 Zhou[11]的分块集成法估计. 若考虑线性分位数回归模型,并假定误差与自变量 $\boldsymbol{X}$ 独立(特别地,此时没有异方差),可以得到 $\boldsymbol{D}_0(\boldsymbol{\theta}_0)=\tau(1-\tau)\mathrm{E}(\vec{\boldsymbol{X}}\vec{\boldsymbol{X}}^{\mathrm{T}})$,$\boldsymbol{D}_1(\boldsymbol{\theta}_0)=f_\varepsilon(0)\mathrm{E}(\vec{\boldsymbol{X}}\vec{\boldsymbol{X}}^{\mathrm{T}})$. 因为未知的密度函数 $f_\varepsilon(0)$ 与参数且与 k 无关,此时若选取权重矩阵 $\boldsymbol{W}_{\boldsymbol{k}}(\widehat{\boldsymbol{\theta}}_k)=n_k\widehat{E}_k(\vec{\boldsymbol{X}}\vec{\boldsymbol{X}}^{\mathrm{T}})=\sum_{i=1}^{n_k}\vec{\boldsymbol{X}}_{k,i}\vec{\boldsymbol{X}}_{k,i}^{\mathrm{T}}$,并代入式(17)就得到

$$\widehat{\boldsymbol{\theta}}_{\mathrm{WW}}=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\left\{n_k\sum_{k=1}^{K}(\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k)^{\mathrm{T}}\widehat{E}_k(\vec{\boldsymbol{X}}\vec{\boldsymbol{X}}^{\mathrm{T}})(\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k)\right\}$$

$$=\left\{\sum_{k=1}^{K}n_k\widehat{E}_k(\vec{\boldsymbol{X}}\vec{\boldsymbol{X}}^{\mathrm{T}})\right\}^{-1}\left\{\sum_{k=1}^{K}n_k\widehat{E}_k(\vec{\boldsymbol{X}}\vec{\boldsymbol{X}}^{\mathrm{T}})\widehat{\boldsymbol{\theta}}_k\right\}\quad(29)$$

这与 Wang 等[21]所提出的估计一致. 以上两种方法都可认为是本节提出的改进的数萃方法,即算法 3 的特例.

因此,本节提出的改进数萃方法式(17)是实际上可视为此类估计的一般框架,该框架包含了一批现有算法. 有时为了提高计算效率和通讯效率,可以允许损失一定估计效率,通过选择简单且易于计算的 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 就可得到快捷的方法. 例如,当 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 选取为 $p+1$ 维单位阵时就是平均估计. 当选取 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)=n_k\widehat{E}_k(\vec{\boldsymbol{X}}\vec{\boldsymbol{X}}^{\mathrm{T}})=\sum_{i=1}^{n_k}\vec{\boldsymbol{X}}_{k,i}\vec{\boldsymbol{X}}_{k,i}^{\mathrm{T}}$ 时,也可以大大提高运算速度. 因此,在允许估计效率损失的情况下,上面的方法可以更方便快捷地计算估计量. 但是在非同分布数据下,估计效率的损失可能较大.

最后还要指出,经典的数萃方法同样可以看作改进的数萃方法的特例. 对于分位数回归,在一般性条件下,$\sqrt{n_k}(\widehat{\boldsymbol{\theta}}_k-\boldsymbol{\theta}_0)\xrightarrow{d}\mathcal{N}(\boldsymbol{0},\boldsymbol{Q}_k(\boldsymbol{\theta}_0))$. 记第 k 台子机器上的估计量 $\widehat{\boldsymbol{\theta}}_k$ 的渐近置信密度函数为 $h_k(\boldsymbol{\theta};\mathscr{D}_k)$. 则对 $k=1,\cdots,K$,

$$h_k(\boldsymbol{\theta};\mathscr{D}_{\boldsymbol{k}})\propto\frac{1}{(2\pi)^{p/2}\mid\widehat{\boldsymbol{Q}}_k(\widehat{\boldsymbol{\theta}}_k)\mid^{1/2}}\times$$

$$\exp\left\{-\frac{n_k}{2}(\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k)^{\mathrm{T}}\widehat{\boldsymbol{Q}}_{\boldsymbol{k}}^{-1}(\widehat{\boldsymbol{\theta}}_k)(\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k)\right\}\quad(30)$$

其中 $\widehat{\boldsymbol{Q}}_k(\widehat{\boldsymbol{\theta}}_k)$ 为 $\widehat{\boldsymbol{Q}}_k(\boldsymbol{\theta}_0)$ 的一致估计(相合估计). Singh 等[12]提出了按照乘积的方式来整合置信密度函数的方法,是一种十分高效的处理手段. 这种想法与似然函数的构造不谋而合. 此时,可以获得关于 $\boldsymbol{\theta}_0$ 的一个估计量,记为 CD(置信分布,confidence distribution)估计量,定义为

$$\widehat{\boldsymbol{\theta}}_{\mathrm{CD}}=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmax}}\prod_{k=1}^{K}h_k(\boldsymbol{\theta};\mathscr{D}_k)$$

$$=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\left\{\sum_{k=1}^{K}n_k(\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k)^{\mathrm{T}}\widehat{\boldsymbol{Q}}_k^{-1}(\widehat{\boldsymbol{\theta}}_k)(\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k)\right\}\quad(31)$$

这就是经典的数萃方法. 可以看到,经典的数萃方法等价于在式(17)中选取 $\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})=\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k$ 和 $\boldsymbol{W}_{\boldsymbol{k}}(\widehat{\boldsymbol{\theta}}_k)=n_k\widehat{\boldsymbol{Q}}_k^{-1}(\widehat{\boldsymbol{\theta}}_k)$.

值得注意的是,改进的数萃方法的关键是找到渐近无偏估计函数 $\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})$,使其满足正则方程 $\mathrm{E}\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta}_0)\approx\boldsymbol{0}$,以及选取合适的正定的权重矩阵 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$,而与 $\widehat{\boldsymbol{\theta}}_k$ 的估计方法并非直接相关. 事实上,无论在每台机器上用何种估计方法得到 $\widehat{\boldsymbol{\theta}}_k$,只要 $\widehat{\boldsymbol{\theta}}_k$ 是 $\boldsymbol{\theta}_0$ 的渐近无偏估计,即有 $\mathrm{E}(\widehat{\boldsymbol{\theta}}_k)\approx\boldsymbol{\theta}_0$,就可以选取估计函数 $\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})=\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k$,则对于任意的权重矩阵 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$,都可以保证 $\widehat{\boldsymbol{\theta}}_M$ 是 $\boldsymbol{\theta}_0$ 的相合估计,而当 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$ 选取为 $\widehat{\boldsymbol{\theta}}_k$ 的渐近协方差阵时,$\widehat{\boldsymbol{\theta}}_M$ 是 $\boldsymbol{\theta}_0$ 的有效估计. 当然,估计函数的取法不是唯一的,只要满足正则条件 $\mathrm{E}\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta}_0)\approx\boldsymbol{0}$ 的函数 $\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})$ 都可以作为估计函数. 因此,$\widehat{\boldsymbol{\theta}}_k$ 的具体估计方法并非决定改进的数萃方法是否适用的关键条件,即不论是分位数回归,还是更一般的 M 估计,还是其他估计方法,只要存在渐近无偏估计函数 $\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})$,就可以利用改进的数萃方法进行数据融合.

在实际中,如何选取估计方程 $\boldsymbol{\xi}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})$ 和权重矩阵 $\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k)$,并选择哪些参数进行传输决定了算法的计算复杂度与通讯成本. 这里,本节所给出的算法 4,算法 5 以及重抽样算法,在同分布数据下的通讯成本均可达到 $O(Kp)$(若考虑迭代,则为 $O(TKp)$),不涉及高维矩阵的传输,因此均为通讯有效的. 其中,算法 4 和算法 5 均基于光滑化估计方程的思想导出,其优势在于其具有较低的通讯成本与计算复杂度,但是需要进行窗宽选择. 重抽样方法相比其他两种方案不需要进行窗宽选择,收敛速度更快,但是由于需要进行

重抽样,因此计算复杂度较高. 值得注意的是,这里给出的三种算法在同分布数据下均为通讯有效的,而不论数据的分布是否相同,算法 5 和重抽样算法总可以得到有效估计. 对于所有算法,若考虑用第一台机器上的样本对相应矩阵进行局部近似,还需要第一台机器上具有充足的样本量. 在实际应用中,应当根据不同的数据分布式结构、通讯成本阈值、计算复杂度阈值,从中选择合适的方法. 表 6 给出了以上三种算法与现有算法的比较.

表 6 改进的数萃方法与几种现有方法的比较

Table 6 Comparison of the improved Meta method over existing methods

估计方法	$\widehat{\boldsymbol{\xi}}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})$	$\boldsymbol{W}_k(\widehat{\boldsymbol{\theta}}_k,\boldsymbol{\theta})$	重抽样	最优通信成本	异质数据	选窗宽
算法 4	$\boldsymbol{U}_k(\widehat{\boldsymbol{\theta}}_k)-\boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\boldsymbol{\theta}$	$\boldsymbol{I}_{p+1}$	否	$O(TKp)$	不一定有效	需要
算法 5	$\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k$	$n_k\boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\widetilde{\boldsymbol{D}}_{01}^{-1}\boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)$	否	$O(TKp)$	有效估计	需要
重抽样方法	$\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k$	$n_k\widehat{\boldsymbol{D}}_{11}(\widehat{\boldsymbol{\theta}}_k)\widehat{\boldsymbol{D}}_{01}^{-1}\widehat{\boldsymbol{D}}_{11}(\widehat{\boldsymbol{\theta}}_k)$	是	$O(TKp)$	有效估计	不需要
经典 Meta	$\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k$	$n_k\widehat{\boldsymbol{Q}}_k^{-1}(\widehat{\boldsymbol{\theta}}_k)$	是	$O(TKp^2)$	有效估计	不需要
Chen 等[10]	$\boldsymbol{U}_k(\widehat{\boldsymbol{\theta}}_k)-\boldsymbol{V}_k(\widehat{\boldsymbol{\theta}}_k)\boldsymbol{\theta}$	$\boldsymbol{V}_k^{-1}(\widehat{\boldsymbol{\theta}}_k)$	否	$O(TKp^2)$	不一定有效	需要
Chen 和 Zhou[11]	$\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k$	$n_k\widehat{\boldsymbol{D}}_{11}(\widehat{\boldsymbol{\theta}}_k)$	否	$O(Kp^2)$	不一定有效	不需要
Wang 等[21]	$\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k$	$n_k\widehat{E}_k(\vec{\boldsymbol{X}}\vec{\boldsymbol{X}}^{\mathrm{T}})$	否	$O(Kp^2)$	不一定有效	不需要

与现有的方法相比,本节所提出的改进的数萃方法主要具有以下三点优势. 1) 从更一般的损失函数的角度出发,本质上对模型没有假设和要求,因此适用范围更广. 此时,估计量 $\widehat{\boldsymbol{\theta}}_k$ 的渐近协方差矩阵依赖于条件密度 $F_{Y|\boldsymbol{X}}(\cdot|\ \boldsymbol{X})$,渐近协方差矩阵的形式更复杂,且估计难度更大. 2) 从估计方程的角度出发,提出了一般性框架,可以将经典数萃方法以及其他很多现有方法涵盖其中. 通过选择不同的估计方程与权重矩阵,可以将经典数萃方法、Chen 和 Zhou[11] 分块集成法、Chen 等[10] 的光滑化方法、Wang 等[21] 的对线性分位数回归模型的方法都视为本节所提出的方法的特例. 因此,本节的改进的数萃方法更具有一般性. 3) 现有的考虑数萃方法的算法大多要求 K 固定或者是针对流数据所设计的,因此未考虑不同机器间的通讯成本. 然而,在分布式系统下,如何提高算法的通讯有效性,却正是本文的出发点和侧重点. 若将现有的针对流数据的方法直接应用于分布式系统,需要在每台机器上估计 $p+1$ 维渐近协方差矩阵,并将其传输到第一台机器,这会带来不可接受的通讯成本. 而在本节提出的改进的数萃算法中,整个过程不涉及矩阵的传输,从而通讯成本得以大大降低.

此外,改进的算法除了不需要子机器上的任何原始数据外,甚至协方差阵的估计都不需要,因此可以大大提高通讯效率. 这里也说明,即使在子机器上(或数据中心)仅有以前的研究获得的参数估计而原始数据缺失的情况下,也可以使用改进的数萃方法来得到分布式计算结果. 这种情况在实际中广泛存在,即只要有摘要数据,就可以通过改进的数萃方法来得到数据融合的估计. 下一节考虑更为一般的非平衡半监督数据,并基于本节中给出的改进的数萃方法的思想提出一种新的通讯有效算法.

5 非平衡半监督数据的通讯有效算法

前面几节考虑了数据完全观测的情况,即每台机器上都可以完整地观测到响应变量和所有协变量. 而在实际应用中,不同机器上的数据,无论协变量或是响应变量都可能存在不同形式的缺失,对应着非平衡半监督数据. 这就要求发展新的大数据分布式计算方法来处理这种复杂的数据类型. 本节将针对非平衡半监督数据的问题,分别在

两种情形下提出不同的通讯有效算法.

回忆前面的记号 $Y \in \mathbb{R}$ 表示响应变量，$\boldsymbol{X} \in \mathbb{R}^p$ 表示 p 维协变量. 本节中关于变量的记号将与前面几节有所不同. 在半监督学习的框架下，数据由两部分组成：(i) 有标签数据 $\mathscr{U}_1 = \{(\boldsymbol{X}_i^{\mathrm{T}}, Y_i)^{\mathrm{T}}: i = 1,2,\cdots,n\}$ 为来自总体分布 $\mathbb{P}$ 的独立同分布样本，存放在第一台机器上；(ii) 对 $k = 2,3,\cdots,K$，第 k 台子机器上没有响应变量的数据，只存放了部分观测的协变量信息. 这里所提到的部分观测的具体含义如下. 对向量 $\boldsymbol{Z} = (Z_1, Z_2,\cdots, Z_p)^{\mathrm{T}}$ 和指标集 $A \subset \{1,2,\cdots,p\}$，令 $\boldsymbol{Z}_A = \{Z_j: j \in A\}$ 表示指标属于集合 A 的 Z 的分量组成的向量. 对 $k = 2,3,\cdots,K$，设第 k 台子机器上观测到的协变量的指标集为 $A_k \subset \{1,2,\cdots,p\}$. 记 $p_k = |A_k| \leqslant p$ 表示指标集 A_k 中元素个数. 则第 k 台子机器上存放的数据可以表示为 $\mathscr{U}_k = \{\boldsymbol{X}_{A_k,i}: i = 1,2,\cdots, n_k\}$，其中 $\{\boldsymbol{X}_{A_k,1},\cdots, \boldsymbol{X}_{A_k,n_k}\}$ 为来自分布 $\mathbb{P}_{\boldsymbol{X}_{A_k}}$ 的独立同分布样本. 图 2 给出了非平衡数据的数据结构示意图. 在实际应用中，标签的获取往往成本较高，这导致有标签的样本量 n 相对较小. 此时，在不同机器间传输有标签数据并不会造成过高的通讯成本. 本节正是在这种情形下展开讨论的.

表 7　在非平衡半监督数据下，基于加权损失函数的通讯有效算法

Table 7 Communication-efficient algorithm based on the weighted loss function for imbalanced semi-supervised data

算法 6 在非平衡半监督数据下，基于加权损失函数的通讯有效算法	
1	初始化 $\boldsymbol{\theta}^{(0)} = \bar{\boldsymbol{\theta}}$ 和迭代循环次数 T；
2	for $t = 0,1,\cdots,T-1$ do
3	在主机器 $\mathscr{M}_1$ 上计算 $\boldsymbol{\alpha}_k^{(t)}$ 和 $\mathscr{M}_k^{(t)}$，并将 $\boldsymbol{\alpha}_k^{(t)}$ 传到子机器 $\mathscr{M}_k$；
4	for $k = 2,\cdots,K$ do
5	在子机器 $\mathscr{M}_k$ 上计算 $\boldsymbol{X}_{A_k,1}^{\mathrm{T}}\boldsymbol{\alpha}_k^{(t)},\cdots,\boldsymbol{X}_{A_k,n_k}^{\mathrm{T}}\boldsymbol{\alpha}_k^{(t)}$，并将 $\boldsymbol{X}_{A_k,1}^{\mathrm{T}}\boldsymbol{\alpha}_k^{(t)}$，$\cdots,\boldsymbol{X}_{A_k,n_k}^{\mathrm{T}}\boldsymbol{\alpha}_k^{(t)}$ 传回主机器 $\mathscr{M}_1$；
6	end for
7	在主机器 $\mathscr{M}_1$ 上计算如下目标函数： $\varphi^{(t)}(\boldsymbol{\theta}) = \frac{\lambda_1}{n}\sum_{i=1}^{n}\rho_\tau(Y_i - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i) + \sum_{k=2}^{K}\frac{\lambda_k}{n_k}\sum_{i=1}^{n_k}\widehat{m}_k(\boldsymbol{X}_{A_k,i}^{\mathrm{T}}\boldsymbol{\alpha}_k^{(t)},\boldsymbol{\theta})$；
8	更新 $\boldsymbol{\theta}^{(t+1)} = \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\varphi^{(t)}(\boldsymbol{\theta})$；
9	end for
10	返回 $\widehat{\boldsymbol{\theta}}_{\mathrm{WE}} = \boldsymbol{\theta}^{(\mathrm{T})}$.

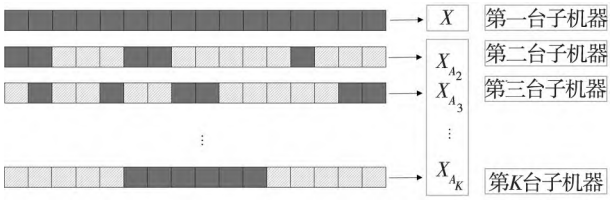

图 2　非平衡数据的数据结构示意图（实心深色方框与虚线浅色方框分别表示可观测与不可观测分量）

Fig. 2 Diagram of data structure of covariates under imbalanced semi-supervised distributed system

注意到，对任意 $\boldsymbol{\alpha}_k \in \mathbb{R}^{p_k}, k = 2,\cdots,K$，

$$\mathrm{E}\{\rho_\tau(Y - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}})\} = \mathrm{E}[\mathrm{E}\{\rho_\tau(Y - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}) \mid \boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{X}_{A_k}\}].$$

令 $m_k(s,\boldsymbol{\theta}) = \mathrm{E}\{\rho_\tau(Y - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}) \mid \boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{X}_{A_k} = s\}$. 当 $\boldsymbol{\theta}$ 给定时，可以利用数据 $\mathscr{U}_1 = \{(\boldsymbol{X}_i^{\mathrm{T}}, Y_i)^{\mathrm{T}}: i = 1,2,\cdots,n\}$ 来估计 $m_k(s,\boldsymbol{\theta})$ 和 $\boldsymbol{\alpha}_k$，且 $m_k(s,\boldsymbol{\theta})$ 可以将条件变量的维数由 p_k 降到 1. 为了模型的可识别性，假设 $\|\boldsymbol{\alpha}_k\| = 1$. 为了估计 $\boldsymbol{\alpha}_k$ 和 h，首先，在第一台机器上计算

$$\widehat{\boldsymbol{\theta}}_1 = \operatorname*{argmin}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\sum_{i=1}^{n}\rho_\tau(Y_i - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i) \tag{32}$$

将 $\widehat{\boldsymbol{\theta}}_1$ 作为初始值并记为 $\bar{\boldsymbol{\theta}}$，利用非线性最小二乘的方法，可以通过最小化如下残差平方和对 $\boldsymbol{\alpha}_k$ 和 h 进行估计，

$$(\widehat{\boldsymbol{\alpha}}_k, h_k) = \operatorname*{argmin}_{\substack{\|\alpha_k\|=1,\\ h_k>0}}\frac{1}{n}\sum_{i=1}^{n}\{\rho_\tau(Y_i - \bar{\boldsymbol{\theta}}^{\mathrm{T}}\vec{\boldsymbol{X}}_i) - \widetilde{m}_{k,h}(\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{X}_{k,i}, \bar{\boldsymbol{\theta}}, \boldsymbol{\alpha}_k)\}^2 \tag{33}$$

其中

$$\widetilde{m}_{k,h}(s,\boldsymbol{\theta},\boldsymbol{\alpha}_k) = \frac{\sum_{i=1}^{n}K_h(\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{X}_{k,i} - s)\rho_\tau(Y_i - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i)}{\sum_{i=1}^{n}K_h(\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{X}_{k,i} - s)} \tag{34}$$

为 $m_k(s,\boldsymbol{\theta})$ 的非参数核平滑（kernel smoothing）估计. 令 $\widehat{m}_k(s,\boldsymbol{\theta}) = \widetilde{m}_{k,h_k}(s,\boldsymbol{\theta},\boldsymbol{\alpha}_k)$. 然后，将 $\widehat{\boldsymbol{\alpha}}_k$ 传到第 k 台子机器上，并在第 k 台子机器上计算 $\widehat{\boldsymbol{\alpha}}_k^{\mathrm{T}}\boldsymbol{X}_{A_k,1},\cdots,\widehat{\boldsymbol{\alpha}}_k^{\mathrm{T}}\boldsymbol{X}_{A_k,n_k}$.

下面，针对两种不同的情形，分别提出两种方法来估计 $\boldsymbol{\theta}_0$. 1）情形一：相对于协变量的维数 p，每一台机器上的样本量 n_k 都比较小. 2）情形二：相对于协变量的维数 p，存在某一台机器，其上的样本量 n_k 比较大. 首先来考虑情形一. 将

$\widehat{\boldsymbol{\alpha}}_k^{\mathrm{T}}\boldsymbol{X}_{A_k,1},\cdots,\widehat{\boldsymbol{\alpha}}_k^{\mathrm{T}}\boldsymbol{X}_{A_k,n_k}$ 由第 k 台子机器传回到第一台机器. 然后,在第一台机器上,构造目标函数

表 8 在非平衡半监督数据下,基于改进的数萃 (Meta) 方法的通讯有效算法

Table 8 Communication-efficient algorithm based on the improved Meta method for imbalanced semi-supervised data

算法 7 在非平衡半监督数据下,基于改进的数萃 (Meta) 方法的通讯有效算法	
1	初始化 $\boldsymbol{\theta}^{(0)}=\bar{\boldsymbol{\theta}}$ 和迭代循环次数 T;
2	for $t=0,1,\cdots,T-1$ do
3	在主机器 $\mathscr{M}_1$ 上估计算 $\boldsymbol{\alpha}_k^{(t)}$ 和 $h_k^{(t)}$, 并将 $\boldsymbol{\theta}^{(t)},\boldsymbol{\alpha}_k^{(t)},h_k^{(t)}$ 和主机器 $\mathscr{M}_1$ 上的数据 $\mathscr{D}_1$ 传到子机器 $\mathscr{M}_k$;
4	for $k=2,\cdots,K$ do
5	在子机器 $\mathscr{M}_k$ 上构造目标函数: $\varphi_k^{(t)}(\boldsymbol{\theta})=\frac{1}{n_k}\sum_{i=1}^{n_k}\widehat{m}_k(\boldsymbol{X}_{A_k,i}^{\mathrm{T}}\boldsymbol{\alpha}_k^{(t)},\boldsymbol{\theta})$;
6	计算 $\boldsymbol{\theta}_k^{(t+1)}=arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}}\varphi_k^{(t)}(\boldsymbol{\theta})$,并将 $\boldsymbol{\theta}_k^{(t+1)}$ 传回主机器 $\mathscr{M}_1$;
7	end for
8	在主机器 $\mathscr{M}_1$ 上计算 $Q_k^{(t+1)}$,更新: $\boldsymbol{\theta}^{(t+1)}=\left(\sum_{k=1}^{K}(\boldsymbol{Q}_k^{(t+1)})^{-1}\right)^{-1}\left(\sum_{k=1}^{K}(\boldsymbol{Q}_k^{(t+1)})^{-1}\boldsymbol{\theta}_k^{(t+1)}\right)$;
9	end for
10	返回 $\widehat{\boldsymbol{\theta}}_{\mathrm{CD}}=\boldsymbol{\theta}^{(\mathrm{T})}$.

$$\varphi(\boldsymbol{\theta})=\frac{\lambda_1}{n}\sum_{i=1}^{n}\rho_\tau(Y_i-\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i)+\sum_{k=2}^{K}\frac{\lambda_k}{n_k}\sum_{i=1}^{n_k}\widehat{m}_{k,h}(\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{X}_{A_k,i},\boldsymbol{\theta}) \tag{35}$$

其中 $\lambda_k\in[0,1],k=1,2,\cdots,K$,满足 $\sum_{k=1}^{K}\lambda_k=1$. 进一步,通过下式估计 $\boldsymbol{\theta}_0$,

$$\widehat{\boldsymbol{\theta}}_{\mathrm{WE}}=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\,\varphi(\boldsymbol{\theta}) \tag{36}$$

算法 6 给出了基于估计量 $\widehat{\boldsymbol{\theta}}_{\mathrm{WE}}$ 的迭代算法.

再看情形二,当每台机器上的样本量 n_k 比较大时,上述方法需要在不同机器之间传输的参数个数为 $O(N)$,而这显然不是通讯有效的. 为了降低通讯成本,这里提出了一种基于伪置信分布 (pseudo confidence distribution, PCD) 的方法. 首先,如式 (32) 和式 (33) 所示,得到 $\widehat{\boldsymbol{\theta}}_1,\widehat{\boldsymbol{\alpha}}_k,h_k$ 并将其与第一台机器上的数据传到第 k 台子机器. 在第 k 台子机器上,基于下式估计 $\boldsymbol{\theta}_0$,

$$\widehat{\boldsymbol{\theta}}_k=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\frac{1}{n_k}\sum_{i=1}^{n_k}\widehat{m}_{k,h}(\boldsymbol{\alpha}_k^{\mathrm{T}}\boldsymbol{X}_{A_k,i},\boldsymbol{\theta})$$

并将估计量 $\widehat{\boldsymbol{\theta}}_k$ 传回第一台机器. 在一般性条件下,当 $\min(n,n_k)\to\infty$ 时,$\widehat{\boldsymbol{\theta}}_k$ 往往近似服从正态分布. 记 $\sqrt{n}(\widehat{\boldsymbol{\theta}}_k-\boldsymbol{\theta}_0)$ 的渐近方差为 $\boldsymbol{Q}_k$,$\widetilde{\boldsymbol{Q}}_k$ 表示 $\boldsymbol{Q}_k$ 的相合估计量. 则可以基于 $\widehat{\boldsymbol{\theta}}_k$ 和 $\widetilde{\boldsymbol{Q}}_k$ 构造如下伪渐近置信密度函数,

$$h_k(\boldsymbol{\theta};\mathscr{D}_k)\propto\frac{1}{(2\pi)^{p/2}\,|\widetilde{\boldsymbol{Q}}_k|^{1/2}}\times\exp\left\{-\frac{n}{2}(\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k)^{\mathrm{T}}\widetilde{\boldsymbol{Q}}_k^{-1}(\boldsymbol{\theta}-\widehat{\boldsymbol{\theta}}_k)\right\}$$

进一步,通过乘积形式构造联合密度函数,并得到 $\boldsymbol{\theta}_0$ 的置信分布 (CD) 估计量,定义为

$$\widetilde{\boldsymbol{\theta}}_{\mathrm{CD}}=\left(\sum_{k=1}^{K}\widetilde{\boldsymbol{Q}}_k^{-1}\right)^{-1}\left(\sum_{k=1}^{K}\widetilde{\boldsymbol{Q}}_k^{-1}\widehat{\boldsymbol{\theta}}_k\right)$$

然而,若在每台机器上都估计 $\widetilde{\boldsymbol{Q}}_k$ 并将其传回第一台机器,就要涉及矩阵的传输. 为了提高通讯效率,允许对矩阵 $\boldsymbol{Q}_k$ 的估计不那么精确,即用估计量 $\widehat{\boldsymbol{Q}}_k$ 代替 $\widetilde{\boldsymbol{Q}}_k$,使得 $\widetilde{\boldsymbol{Q}}_k$ 可以基于第一台机器上的数据和 $\widehat{\boldsymbol{\theta}}_k$ 得到. 估计量 $\widetilde{\boldsymbol{Q}}_k$ 可通过 Zeng 和 Lin[22] 中提出的 LS 算法或 SV 算法得到. 第 4 节已经给出了 LS 算法的步骤,此处给出基于 SV 算法的高效通讯算法步骤. 具体而言,令 $\boldsymbol{S}_i(\boldsymbol{\theta})=\{I(Y_i\leqslant\boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i)-\tau\}\vec{\boldsymbol{X}}_i$. 对 $k=1,\cdots,K$,计算 $\widehat{\boldsymbol{V}}_k=n^{-1}\sum_{i=1}^{n}\boldsymbol{S}_i(\widehat{\boldsymbol{\theta}}_k)\boldsymbol{S}_i^{\mathrm{T}}(\widehat{\boldsymbol{\theta}}_k)$. 生成 B 个独立同分布随机向量 $\{\boldsymbol{Z}_{k,b}\}_{b=1}^{B}$,使得 $\boldsymbol{Z}_{k,b}\sim\mathscr{N}(\boldsymbol{0},\widehat{\boldsymbol{V}}_k^{-1})$. 令 $\widetilde{\boldsymbol{\theta}}_{k,b}=\widehat{\boldsymbol{\theta}}_k+n^{-1/2}\boldsymbol{Z}_{k,b}$ 以及 $\boldsymbol{U}_{k,b}=\sum_{i=1}^{n}\boldsymbol{S}_i(\widetilde{\boldsymbol{\theta}}_{k,b})$. 计算 $\{n_1^{-1/2}\boldsymbol{U}_{k,b}\}_{b=1}^{B}$ 的样本协方差矩阵记为 $\widehat{\boldsymbol{\Sigma}}_k$,取 $\boldsymbol{Q}_k$ 的估计为 $\widehat{\boldsymbol{Q}}_k=\widehat{\boldsymbol{\Sigma}}_k^{-1}$. 相应地 CD 估计量为

$$\widehat{\boldsymbol{\theta}}_{\mathrm{CD}}=\left(\sum_{k=1}^{K}\widehat{\boldsymbol{Q}}_k^{-1}\right)^{-1}\left(\sum_{k=1}^{K}\widehat{\boldsymbol{Q}}_k^{-1}\widehat{\boldsymbol{\theta}}_k\right) \tag{37}$$

这里 $\widehat{\boldsymbol{Q}}_k$ 不一定与 $\boldsymbol{Q}_k$ 相合,但并不会影响 $\widehat{\boldsymbol{\theta}}_{\mathrm{CD}}$ 的相合性,只影响其估计效率. 相应迭代算法呈现为算法 7. 图 3 给出了算法 6 和算法 7 的示意图.

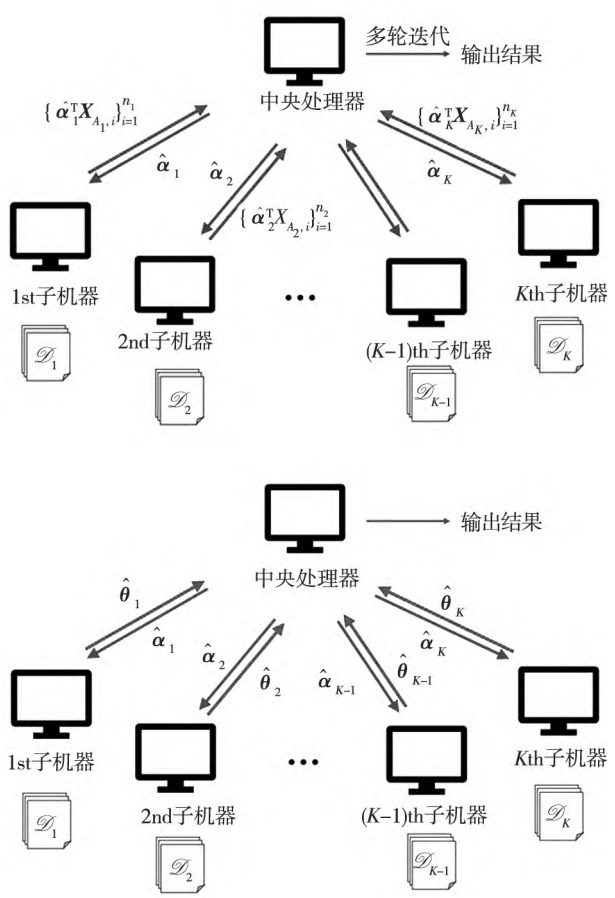

图 3　非平衡半监督数据的通讯有效算法示意图

Fig. 3 Diagram of communication-efficient algorithm for imbalanced semi-supervised data

6　数值模拟分析

本节将通过大量数值模拟对第 2 节至第 5 节中提出的算法验证其在有限样本下的表现. 模拟将分为两个部分呈现,第一部分包括本文所有可用于完全观测数据的分布式算法,以及其它相关研究工作中提出的算法. 这里将展示所有这些方法的估计效果并做出对照分析. 第二部分包括本文可应用于非平衡半监督数据的高效通讯算法相关结果.

第一部分考虑有完全观测数据的分布式算法. 不失一般性地,可以选择不同机器数据等分的简洁设定,同时选用充分大的样本量以较好地贴合大数据算法的现实背景. 模拟采用十万级别的数据规模,在计算中采用十二核的 CPU 进行并行计算,以模仿类似多台计算机的实现效果. 实际计算过程中占用计算空间约 16 GiB.

具体而言,总机器的个数设为 $K = 50$,每台机器上的样本量 $n_1 = \cdots = n_K = 2\,000$,则总样本量 $N = 100\,000$. 模拟的重复次数为 200 次. 作为对比,也在附录中给出了总机器数 $K = 10$,每台机器上的样本量 $n_1 = \cdots = n_K = 1\,000$,总样本量 $N = 10\,000$ 的模拟结果. 对于需要迭代的算法,固定迭代步数为 $T = 5$ 步. 对于光滑化算法（SJ 算法,对应于改进的数萃方法中的算法 4 与算法 5),选择 $H(x)$ 为式（12）中的形式,同时设定不同机器上的窗宽相等,等于 $h_k = (p/n_k)^{1/4}$. 对于需要进行重抽样的方法,选取重抽样样本的样本量为 $B = 200$. 模拟时,结果主要关注两组分位数水平 $\tau = 0.5$ 和 $\tau = 0.25$,其它分位数下的结果类似,为节省篇幅在此不再列出. 每一种算法都将呈现以下结果指标：基于 200 次模拟得到的估计量相对于参数真值的平均偏差（Bias）和标准误差（standard error,SE). 同时,将集中所有数据在一台假设有无限算力的超级计算机上,通过传统分位数回归的线性优化方法得到的估计量 $\widehat{\boldsymbol{\theta}}_N = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N} \rho_\tau(Y_i - \boldsymbol{\theta}^{\mathrm{T}}\vec{\boldsymbol{X}}_i)$ 作为基准估计,称为 Oracle 估计. 通过计算各方法得到的估计量的平均平方误差（mean squared error,MSE）与 Oracle 估计的 MSE 的比值,得出各种算法所得估计相对基准估计的估计效率. 估计效率越大表明估计越有效,特别地,若估计效率接近 1（当不同机器数据分布不同时,可以大于 1),表明估计是渐近有效估计. 此外,计算中也会测量计算时长对标算法的计算复杂程度,这里定义为在分布式计算中,每台机器运算 200 次模拟中,单次运算花费的平均时长,并定义通讯成本为在核心计算步骤中,机器之间的数据传输量,由传输量的主阶数表示.

考虑线性模型与平方模型的两种设定,并在线性模型下,进一步考虑异方差结构,且允许不同机器上的数据分布相同或不同. 首先,考虑如下线

性数据生成模型

$$\begin{aligned} &Y_{k,i} = \boldsymbol{\theta}_0^{\mathrm{T}} \vec{\boldsymbol{X}}_{k,i} + \alpha_0^{\mathrm{T}} \vec{\boldsymbol{X}}_{k,i}\, \varepsilon_{k,i} \\ &\varepsilon_{k,i} \overset{\mathrm{IID}}{\sim} \mathcal{N}(0,1), \varepsilon_{k,i} \perp \boldsymbol{X}_{k,i} \end{aligned} \tag{38}$$

其中 $\boldsymbol{A} \perp \boldsymbol{B}$ 表示随机向量 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 独立，$\vec{\boldsymbol{X}} = (1,\boldsymbol{X}^{\mathrm{T}})^{\mathrm{T}} = (1,X_1,\cdots,X_p)^{\mathrm{T}}$ 包含截距项，$\tilde{\boldsymbol{\theta}}_0 = \{(-1)^l, l = 0,\cdots,p\}$ 为 $p+1$ 维线性系数，$\boldsymbol{\alpha}_0^{\mathrm{T}}\vec{\boldsymbol{X}}$ 决定了模型的异方差结构，可以通过设定系数 $\boldsymbol{\alpha}_0$ 来对是否存在条件异方差与条件异方差的形式进行控制. 特别地，当 $\boldsymbol{\alpha}_0^{\mathrm{T}} \vec{\boldsymbol{X}}_{k,i}$ 恒取正值时，待估参数的真值 $\boldsymbol{\theta}_0 = \tilde{\boldsymbol{\theta}}_0 + c_\tau \boldsymbol{\alpha}_0$，其中 c_τ 为标准正态分布的 τ 分位数. 设定协变量维数 $p = 2$. 在线性模型(38)下，将分别考虑不存在条件异方差与条件异方差具有线性结构两种情形，并在线性条件异方差下，考虑不同机器上的数据分布不相同的情况. 为此，考虑如下四种设定.

(a) 同方差模型，每台机器上的变量 $(\boldsymbol{X}_{k,i}^{\mathrm{T}}, Y_{k,i})^{\mathrm{T}}$ 分布相同. 设协变量 $\boldsymbol{X}_{k,i}$ 的各个分量相互独立，服从(0,1)上的均匀分布 $U(0,1)$. 设 $\boldsymbol{\alpha}_0 = (1,0,\cdots,0)^{\mathrm{T}}$，即除截距项以外，其它维度系数为零，则此时总体噪声 $\tilde{\varepsilon} = \boldsymbol{\alpha}_0^{\mathrm{T}}\vec{\boldsymbol{X}}\varepsilon$，$\varepsilon$ 与 $\boldsymbol{X}$ 独立，即不存在条件异方差. 待估的参数真值为 $\boldsymbol{\theta}_0 = \tilde{\boldsymbol{\theta}}_0 + c_\tau \boldsymbol{e}_1$，其中 $\boldsymbol{e}_k$ 表示第 k 个分量为 1，其他分量均为 0 的单位向量.

(b) 异方差模型，每台机器上的变量 $(\boldsymbol{X}_{k,i}^{\mathrm{T}}, Y_{k,i})^{\mathrm{T}}$ 分布相同. 协变量 $\boldsymbol{X}_{k,i}$ 与 (a) 的设定相同. 设 $\boldsymbol{\alpha}_0 = (1,0.5,0,\cdots,0)^{\mathrm{T}}$，则总体噪声 $\tilde{\varepsilon}$ 在给定 $\boldsymbol{X}$ 的条件下的条件分布为 $\tilde{\varepsilon} \mid \boldsymbol{X} \sim \mathcal{N}(0, (1+0.5X_1)^2)$. 待估参数的真值为 $\boldsymbol{\theta}_0 = \tilde{\boldsymbol{\theta}}_0 + c_\tau (\boldsymbol{e}_1 + 0.5\,\boldsymbol{e}_2)$.

(c) 异方差模型，每台机器上的协变量 $\boldsymbol{X}_{k,i}$ 分布不同. 令 $\boldsymbol{X}_{k,i} = (X_{k,i,1},\cdots,X_{k,i,p})^{\mathrm{T}}$. 在第 k 台机器上，设 $X_{k,i,p} \overset{\mathrm{IID}}{\sim} U(k/10, k/5)$，即不同机器上协变量的第一个分量 X_1 的分布不同，而 $\boldsymbol{X}_{-1} = (X_2,\cdots,X_p)^{\mathrm{T}}$ 在不同机器上的分布相同，各个分量相互独立且服从 $U(0,1)$ 分布. 设 $\boldsymbol{\alpha}_0 = (0,0.5,0,\cdots,0)^{\mathrm{T}}$，则异方差项由 X_1 决定，总体噪声 $\tilde{\varepsilon}$ 在给定 $\boldsymbol{X}$ 的条件下的条件分布为 $\tilde{\varepsilon} \mid \boldsymbol{X} \sim N(0,(0.5X_1)^2)$. 待估参数的真值为 $\boldsymbol{\theta}_0 = \tilde{\boldsymbol{\theta}}_0 + 0.5\, c_\tau\, \boldsymbol{e}_2$.

(d) 平方模型，每台机器上的变量 $(\boldsymbol{X}_{k,i}^{\mathrm{T}}, Y_{k,i})^{\mathrm{T}}$ 分布相同. 考虑如下数据生成模型

$$\begin{aligned} &Y_{k,i} = (\tilde{\boldsymbol{\theta}}_0^{\mathrm{T}} \vec{\boldsymbol{X}}_{k,i})^2 + \varepsilon_{k,i} \\ &\boldsymbol{X}_{k,i} \overset{\mathrm{IID}}{\sim} \mathcal{N}(0,\Sigma) \\ &\varepsilon_{k,i} \overset{\mathrm{IID}}{\sim} \mathcal{N}(0,1), \varepsilon_{k,i} \perp \boldsymbol{X}_{k,i} \end{aligned} \tag{39}$$

其中，协方差矩阵 $\boldsymbol{\Sigma} = (\sigma_{ij})_{p\times p}$ 满足当 $i = j$ 时 $\sigma_{ii} = 1$，当 $i \neq j$ 时 $\sigma_{ij} = 0.5$. 此时，Y 在给定 $\boldsymbol{X}$ 的条件下的 τ 条件分位数为 $\boldsymbol{Q}_\tau(Y \mid \boldsymbol{X}) = (\tilde{\boldsymbol{\theta}}_0^{\mathrm{T}}\vec{\boldsymbol{X}})^2 + c_\tau$. 选取 $\tilde{\boldsymbol{\theta}}_0 = \boldsymbol{I}_p$，并考虑维数 $p = 2$ 的情况. 通过蒙特卡洛模拟得到待估参数的真值如下

$$\boldsymbol{\theta}_0 = \begin{cases} (2.643,1.999,2.001)^{\mathrm{T}}, & \tau = 0.5 \\ (1.359,1.999,2.001)^{\mathrm{T}}, & \tau = 0.25 \end{cases}$$

注意到，在此情况下线性分位数回归模型不成立.

表 9 至表 12 给出了不同模型下算法的估计效率，计算复杂度和通讯成本. 其中，EC 表示基于等度连续的通讯有效算法；SJ 表示基于光滑逼近的通讯有效算法，SJ1 和 SJ2 分别表示结合替代似然和牛顿算法的两种优化方法；在改进的数萃方法中，MV 和 SP 分别对应算法 4 和算法 5，MB 表示基于重抽样方法估计权重矩阵的算法. 特别地，在 MV 算法中，可选取权重矩阵为单位阵，得到一个可以进行快速计算的算法. 当不同机器上的数据同分布时，SP 和 MB 可以通过第一台机器的数据进行相应的矩阵近似，得到对应的高效通讯算法，分别记为 SP-Local 和 MB-Local. 作为对照，记基于式(28)的朴素平均算法为 AM，以及 Chen 等[10]，Chen 和 Zhou[11]，Wang 等[20]的方法分别为 ML，DC，WW. 为了简洁，在不致产生混淆的前提下，下文中均用以上简写表示相应算法所得估计量.

表 9 与表 10 展示了在线性模型下，不同机器数据同分布时的模拟结果. 首先，从标准误差（SE）的量级上看，本文所给出的方法均可以得到真值的一致估计，只是在不同的场景下，各种估计的效率有一定的差别. 大部分算法所得估计量的估计效率都接近为 1，除了 MV 算法和基于等度连续的 EC 算法所得估计量的估计效率略低于其他几种方法. 其中，若将 ML 算法在 GMM 估计的框架下看作两步估计，则 MV 算法可以看作其第一步估计后得到的结果，是在对估计效率要求不高时的一种加速算法，在此给出仅作为比较. EC 算法因为进行了大量高阶梯度近似，用估计效率的降低换取了通讯效率的提升，因此在给定的迭代步数下估计效率略低于其他算法. 事实上，通过增加迭代步数，其所得估计量的估计效率可以进一步提升.

表 9　同质性分布式系统下线性同方差模型（a）的模拟结果

Table 9 Simulation result of homoscedastic linear model (a) under homogenous distributed system

分位数水平	算法	Bias0 (10^{-3})	Bias1 (10^{-3})	Bias2 (10^{-3})	SE0 (10^{-2})	SE1 (10^{-2})	SE2 (10^{-2})	估计效率	计算时长（秒）	通讯成本
$\tau = 0.50$	Oracle	1.08	−1.42	−1.13	1.02	1.35	1.37	1.00	3.55	−
	EC	0.60	−2.03	0.58	1.55	2.26	2.20	0.62	8.76	$O(TKp)$
	SJ1	1.01	−1.34	−1.00	1.03	1.35	1.38	1.00	1.15	$O(TKp)$
	SJ2	1.00	−1.34	−1.00	1.03	1.35	1.38	1.00	0.48	$O(TKp)$
	MV	−0.03	−0.58	0.25	1.68	1.98	2.12	0.65	0.42	$O(TK(p^2+p))$
	SP	1.11	−1.56	−1.08	1.05	1.37	1.37	0.99	0.43	$O(K(p^2+p))$
	MB	1.18	−1.69	−1.11	1.05	1.39	1.39	0.97	1.33	$O(K(p^2+p))$
	SP-Local	1.37	−2.11	−1.01	1.13	1.46	1.44	0.93	0.46	$O(Kp)$
	MB-Local	1.30	−1.93	−1.04	1.24	1.59	1.57	0.85	1.48	$O(Kp)$
	AM	1.10	−1.54	−1.03	1.05	1.37	1.34	1.00	0.25	$O(Kp)$
	ML	0.99	−1.33	−0.99	1.03	1.35	1.37	1.00	0.65	$O(TK(p^2+p))$
	DC	1.10	−1.59	−1.04	1.05	1.36	1.36	0.99	1.53	$O(K(p^2+p))$
	WW	1.09	−1.57	−1.00	1.05	1.36	1.33	1.00	0.27	$O(K(p^2+p))$
$\tau = 0.25$	Oracle	−0.44	0.77	0.86	1.09	1.50	1.45	1.00	1.92	−
	EC	−2.09	3.64	1.17	2.00	2.91	2.75	0.52	9.16	$O(TKp)$
	SJ1	5.18	1.60	0.58	1.12	1.56	1.49	0.94	1.31	$O(TKp)$
	SJ2	5.29	1.39	0.59	1.11	1.54	1.49	0.95	0.53	$O(TKp)$
	MV	5.54	1.17	2.85	1.96	2.44	2.29	0.60	0.39	$O(TK(p^2+p))$
	SP	1.26	0.95	0.85	1.13	1.52	1.51	0.97	0.41	$O(K(p^2+p))$
	MB	1.22	0.92	0.93	1.15	1.54	1.52	0.96	1.46	$O(K(p^2+p))$
	SP-Local	1.32	0.30	1.10	1.23	1.63	1.71	0.88	0.46	$O(Kp)$
	MB-Local	1.31	0.70	1.00	1.31	1.71	1.82	0.83	1.67	$O(Kp)$
	AM	−0.18	0.75	0.86	1.08	1.48	1.49	1.00	0.22	$O(Kp)$
	ML	5.44	1.02	0.65	1.11	1.51	1.48	0.96	0.64	$O(TK(p^2+p))$
	DC	0.55	0.82	0.85	1.10	1.50	1.50	0.98	1.69	$O(K(p^2+p))$
	WW	−0.14	0.71	0.84	1.08	1.48	1.49	1.00	0.23	$O(K(p^2+p))$

表 10 同质性分布式系统下线性异方差模型（b）的模拟结果

Table 10 Simulation result of heteroscedastic linear model (b) under homogenous distributed system

分位数水平	算法	Bias0 (10^{-3})	Bias1 (10^{-3})	Bias2 (10^{-3})	SE0 (10^{-2})	SE1 (10^{-2})	SE2 (10^{-2})	估计效率	计算时长（秒）	通讯成本
$\tau = 0.50$	Oracle	−0.33	−1.92	1.59	1.32	1.84	1.59	1.00	3.07	−
	EC	−1.43	−2.23	4.83	2.15	2.74	2.81	0.62	8.87	$O(TKp)$
	SJ1	−0.35	−1.98	2.04	1.34	1.98	1.69	0.94	1.19	$O(TKp)$
	SJ2	−0.05	−2.43	1.55	1.29	1.88	1.61	0.99	0.50	$O(TKp)$
	MV	−1.64	−0.94	3.42	2.21	2.78	2.70	0.62	0.42	$O(TK(p^2+p))$
	SP	0.09	−2.09	1.10	1.28	1.82	1.58	1.01	0.42	$O(K(p^2+p))$
	MB	0.05	−2.08	1.11	1.32	1.86	1.59	1.00	1.36	$O(K(p^2+p))$
	SP-Local	0.36	−1.89	0.32	1.38	2.06	1.77	0.91	0.44	$O(Kp)$
	MB-Local	0.61	−1.67	−0.06	1.61	2.31	2.15	0.78	1.51	$O(Kp)$
	AM	0.06	−1.94	0.91	1.27	1.84	1.58	1.01	0.25	$O(Kp)$
	ML	−0.17	−1.99	1.50	1.29	1.83	1.59	1.01	0.67	$O(TK(p^2+p))$
	DC	0.01	−1.95	1.00	1.29	1.84	1.58	1.01	1.53	$O(K(p^2+p))$
	WW	0.04	−1.85	0.85	1.26	1.84	1.57	1.02	0.26	$O(K(p^2+p))$
$\tau = 0.25$	Oracle	1.70	0.08	−2.07	1.35	2.00	1.81	1.00	1.43	−
	EC	3.22	−1.67	−3.77	2.16	3.26	2.97	0.61	9.27	$O(TKp)$
	SJ1	7.04	−0.89	−2.19	1.40	2.06	1.85	0.95	1.37	$O(TKp)$
	SJ2	7.12	−1.01	−2.23	1.39	2.05	1.84	0.96	0.52	$O(TKp)$
	MV	5.44	2.14	0.87	2.23	3.07	2.89	0.63	0.40	$O(TK(p^2+p))$
	SP	3.90	0.86	−2.86	1.39	2.05	1.84	0.97	0.42	$O(K(p^2+p))$
	MB	3.98	0.44	−2.80	1.39	2.07	1.86	0.96	1.42	$O(K(p^2+p))$
	SP-Local	2.32	1.76	−1.43	1.62	2.45	2.23	0.82	0.45	$O(Kp)$
	MB-Local	2.41	1.18	−0.92	1.76	2.71	2.45	0.75	1.63	$O(Kp)$
	AM	1.67	1.04	−2.74	1.42	2.03	1.87	0.97	0.22	$O(Kp)$
	ML	7.24	−1.27	−2.24	1.38	2.03	1.84	0.96	0.69	$O(TK(p^2+p))$
	DC	3.04	0.34	−2.76	1.39	2.04	1.85	0.98	1.71	$O(K(p^2+p))$
	WW	1.66	1.06	−2.76	1.41	2.02	1.87	0.98	0.24	$O(K(p^2+p))$

表 11 异质性分布式系统下线性异方差模型（c）的模拟结果

Table 11 Simulation result of heteroscedastic linear model (c) under heterogeneous distributed system

分位数水平	算法	Bias0 (10^{-3})	Bias1 (10^{-3})	Bias2 (10^{-3})	SE0 (10^{-2})	SE1 (10^{-2})	SE2 (10^{-2})	估计效率	计算时长（秒）	通讯成本
$\tau = 0.50$	Oracle	−0.20	0.04	−0.27	0.31	0.26	0.35	1.00	4.72	−
	MV	−0.08	0.04	−0.38	0.54	0.34	0.27	0.76	0.42	$O(TK(p^2+p))$
	SP	−0.03	−0.04	−0.22	0.16	0.22	0.19	1.59	0.43	$O(K(p^2+p))$
	MB	−0.01	−0.03	−0.23	0.17	0.23	0.19	1.56	1.48	$O(K(p^2+p))$
	AM	−6.41	0.81	−0.24	4.54	1.04	0.90	0.11	0.33	$O(Kp)$
	ML	−0.23	0.06	−0.27	0.33	0.27	0.35	0.97	0.60	$O(TK(p^2+p))$
	DC	−0.26	0.08	−0.23	0.36	0.28	0.41	0.87	1.50	$O(K(p^2+p))$
	WW	−1.20	0.31	−0.23	1.30	0.47	0.90	0.32	0.28	$O(K(p^2+p))$

续表 11

Table 11 Continues

分位数水平	算法	Bias0 (10^{-3})	Bias1 (10^{-3})	Bias2 (10^{-3})	SE0 (10^{-2})	SE1 (10^{-2})	SE2 (10^{-2})	估计效率	计算时长（秒）	通讯成本
$\tau = 0.25$	Oracle	0.10	−0.22	0.39	0.36	0.33	0.35	1.00	4.62	−
	MV	4.43	0.09	0.19	0.55	0.42	0.24	0.71	0.43	$O(TK(p^2+p))$
	SP	−0.01	0.70	0.00	0.20	0.25	0.18	1.62	0.45	$O(K(p^2+p))$
	MB	0.08	0.66	0.04	0.21	0.26	0.19	1.55	1.68	$O(K(p^2+p))$
	AM	−5.81	1.52	1.29	4.41	1.05	0.88	0.13	0.31	$O(Kp)$
	ML	5.11	−0.34	0.37	0.36	0.34	0.33	0.77	0.57	$O(TK(p^2+p))$
	DC	0.40	0.13	0.56	0.42	0.35	0.37	0.90	1.62	$O(K(p^2+p))$
	WW	−0.03	−0.06	1.29	1.40	0.55	0.88	0.35	0.28	$O(K(p^2+p))$

表 12　同质性分布式系统下平方模型（d）的模拟结果

Table 12 Simulation result of quadratic model (d) under homogenous distributed system

分位数水平	算法	Bias0 (10^{-3})	Bias1 (10^{-3})	Bias2 (10^{-3})	SE0 (10^{-2})	SE1 (10^{-2})	SE2 (10^{-2})	估计效率	计算时长（秒）	通讯成本
$\tau = 0.50$	Oracle	0.50	−0.84	0.02	0.49	0.98	1.02	1.00	5.48	−
	EC	−0.06	−0.78	−0.21	0.80	1.49	1.40	0.68	8.89	$O(TKp)$
	SJ1	−1.18	−0.76	−0.24	0.52	1.08	1.15	0.90	1.10	$O(TKp)$
	SJ2	−1.17	−0.72	−0.03	0.51	1.01	1.05	0.97	0.46	$O(TKp)$
	MV	−1.31	−0.74	0.03	0.51	1.03	1.05	0.96	0.44	$O(TK(p^2+p))$
	SP	0.23	−0.67	0.07	0.51	1.03	1.06	0.96	0.46	$O(K(p^2+p))$
	MB	0.21	−0.66	0.06	0.52	1.04	1.06	0.95	1.34	$O(K(p^2+p))$
	SP-Local	0.46	−1.69	1.29	0.59	1.24	1.26	0.80	0.49	$O(Kp)$
	MB-Local	0.32	−1.78	1.38	0.76	1.62	1.58	0.63	1.48	$O(Kp)$
	AM	0.13	−0.64	0.15	0.49	0.99	1.03	0.99	0.29	$O(Kp)$
	ML	−1.21	−0.74	0.06	0.49	0.98	1.02	0.99	0.55	$O(TK(p^2+p))$
	DC	0.68	−0.68	0.16	0.50	1.00	1.04	0.98	1.47	$O(K(p^2+p))$
	WW	1.12	−0.63	0.17	0.49	0.98	1.03	1.00	0.30	$O(K(p^2+p))$
$\tau = 0.25$	Oracle	−0.03	−1.10	0.99	0.46	1.16	1.21	1.00	3.16	−
	EC	−0.69	−1.53	1.12	0.74	1.66	2.10	0.63	8.97	$O(TKp)$
	SJ1	3.98	−1.10	0.50	0.50	1.32	1.33	0.88	1.24	$O(TKp)$
	SJ2	3.94	−0.82	0.81	0.48	1.19	1.22	0.96	0.50	$O(TKp)$
	MV	6.29	−1.00	0.77	0.49	1.30	1.37	0.85	0.41	$O(TK(p^2+p))$
	SP	2.27	−1.16	0.91	0.47	1.19	1.25	0.97	0.43	$O(K(p^2+p))$
	MB	2.25	−1.01	0.71	0.48	1.19	1.25	0.96	1.39	$O(K(p^2+p))$
	SP-Local	2.58	−1.48	1.38	0.55	1.45	1.46	0.81	0.46	$O(Kp)$
	MB-Local	2.57	−1.52	1.35	0.64	1.63	1.62	0.72	1.51	$O(Kp)$
	AM	−0.19	−1.27	1.31	0.47	1.16	1.20	1.00	0.25	$O(Kp)$
	ML	3.98	−1.13	0.99	0.47	1.16	1.22	0.97	0.57	$O(TK(p^2+p))$
	DC	1.62	−1.19	1.05	0.47	1.17	1.22	0.99	1.53	$O(K(p^2+p))$
	WW	0.81	−1.31	1.29	0.47	1.16	1.20	1.00	0.26	$O(K(p^2+p))$

关于 EC 算法的计算复杂度在此有一点补充：从结果上看，EC 算法的运算时间略高于其它算法，这并非是 EC 算法本身的缺陷，而是由其所采用的优化算法导致. 在本文处理关键的迭代步骤时，针对于这个新提出的问题，求解线性优化时采用了运算稳定且可解性更为清晰的单纯形（simplex）法. 如果在此换成经典的内点（interior point）法求解线性优化问题，也可以达到毫不逊色于表中其它迭代方法的计算效率，但是解的稳定性不如本文所采用的方法.

正如本文第 4 节中所讨论的一样，当已知数据独立同分布时，改进的数萃方法所得 SP 与 MB 估计，Chen 和 Zhou[11] 所得 DC 估计，Wang 等[20] 所得 WW 估计，本质上都与 AM 估计渐近等价. 事实上，此时采用以上任意一种估计量都是渐近有效的，而直接采用朴素平均就可以得到有效估计，且计算难度也是最低的，运算时间也最短. 由表中结果可知，以上不同方法的效率几乎都近似为 1，都可以认为与 Oracle 估计之间没有本质区别. 这也说明，如果预先已经有额外的信息，可以确定样本不存在非同分布的情况，那么以上所有算法均可以实现有效估计并都具有较小的计算成本.

下面给出所提出的几种算法之间的比较. 仅采用中心机器的本地信息进行运算的 Local 方法，由于机器与机器间传输的信息量大大减少，多少会伴随一定的估计效率损失，一般在 10% 左右，但其优点是通讯成本的大大降低. 另一方面，光滑化处理之后的方法，估计效率往往优于不做光滑直接针对非光滑函数做优化的方法，即 SP 估计将在估计效率上优于 MB 估计. 同理，SJ 算法也比直接对核验函数做线性优化的 EC 算法更优. 采用重抽样方法对协方差矩阵进行估计的方案，则会因为重抽样而相对花费更多的计算时间. 由此可见，直接代入数值的 SP 算法总体上最有优势. 从组间对照的角度，仅看两种重抽样方法，MB 估计由于包含对方差阵的完整估计，也会在估计效率上略优于 Chen 和 Zhou[11] 的 DC 估计. 仅看高效通讯算法，本地化的 SP-Local 和 MB-Local 估计在不做迭代的情况下，估计效率表现不如做迭代的 SJ 估计. 而且 SJ 算法在使用迭代的同时计算效率也很高. 特别是直接代入光滑近似黑塞矩阵（Hessian matrix）的 SJ2 方法运算速度很快，在同类算法中表现得较有优势. 尽管如此，SP-Local 和 MB-Local 算法在仅从本地传输一次向量的较低通讯成本下，能达到 80% 甚至 90% 的估计效率，也较好地平衡了通讯成本与计算效率.

表 11 给出了异方差模型下，不同机器上的数据分布不同时的模拟结果. 当不同机器上的数据分布不同时，由于 EC 算法与 SJ 算法都在本地通过第一台机器上的数据进行高阶梯度近似，需要数据同分布假设，因此在理论上并不适用，使得在计算过程中往往出现优化问题无解的情况. 为此，在此情形下将不考虑 EC 算法与 SJ 算法. 从表中结果可知，改进的数萃方法相比现有的 ML 算法，DC 算法和 WW 算法，所得估计量的估计效率显著提升，且计算时间显著减小. 由第 4 节中的讨论，从理论上讲，当不同机器上的数据分布不同时，最优权重矩阵将依赖于机器指标 k，此时，改进的数萃算法，DC 算法，WW 算法所得估计与朴素平均估计将不再渐近等价. 此时，由于本文的算法采用了最优权重矩阵，因此不论数据是否同分布，所得估计均为渐近有效估计. 本文的算法具有的估计有效性以及在数据分布异质性下的稳健性，由模拟结果得到了进一步的验证.

在平方模型下，由表 12 中的结果可知，改进的数萃方法相比 EC 算法与 SJ 算法，所得估计量保持了更高的估计效率. 特别地，MV 算法的优势在平方模型下得以体现，表现为在计算时长差不多的情况下，相比所提出的其他算法具有并不逊色的估计效率，特别是在分位数水平等于 0.5 时，在几种改进的数萃方法中表现最佳. 与此同时，若将 MV 算法看作 ML 算法在 GMM 估计框架下的第一步，则可以看到，与 ML 算法相比，仅通过一步计算得到的 MV 估计的估计效率并未有太大降低，而算法的计算时长分别减小了 28.57%（$\tau = 0.50$）与 27.27 %（$\tau = 0.25$）. 因此，MV 算法在平方模型下可以实现估计的有效性与计算复杂度之间的平衡.

第二部分考虑非平衡半监督数据的通讯有效算法. 在非平衡半监督分布式系统下，仅针对不同机器上数据分布相同的情况进行模拟. 同时增加

总机器的个数到 $K = 100$，并允许第一台机器上有标签数据的样本量减小到 $n = 100$，第 $k(k \geqslant 2)$ 台机器上的半监督数据的样本量为 $n_k = 1\,000$，则总样本量为 $N = 99\,100$. 由于本文提出的算法可以适用于多维协变量，因此也将协变量的维数增加到 $p = 9$. 第一台机器上为有标签数据且协变量完整观测，第 $k(k \geqslant 2)$ 台机器上为无标签数据，且协变量存在缺失. 表 13 给出了协变量的具体缺失设置. 分位数水平设为 $\tau = 0.5$ 和 $\tau = 0.25$. 分别考虑线性模型和平方模型两种数据生成方式，分别对应模型（1）正确设定和误设的两种情形.

表 13　非平衡半监督分布式系统下协变量缺失模式

Table 13 Missing pattern of covariates under imbalanced semi-supervised distributed system

机器	$\boldsymbol{X}_1$	$\boldsymbol{X}_2$	$\boldsymbol{X}_3$	$\boldsymbol{X}_4$	$\boldsymbol{X}_5$	$\boldsymbol{X}_6$	$\boldsymbol{X}_7$	$\boldsymbol{X}_8$	$\boldsymbol{X}_9$
1	*	*	*	*	*	*	*	*	*
2	*	*							
3				*		*			
4								*	*
5	*		*		*				
6		*		*		*			
7			*		*		*		
8	*	*	*	*	*	*			
9		*	*	*	*	*	*		*
10				*	*	*	*	*	*

注：‘*’表示变量可观测，否则表示变量缺失.

（e）线性模型. 考虑如下数据生成模型

$$Y_i = \widetilde{\boldsymbol{\theta}}_0^{\mathrm{T}} \vec{\boldsymbol{X}}_i + \varepsilon_i$$

第 1 台机器：$\boldsymbol{X}_i \overset{\mathrm{IID}}{\sim} \mathcal{N}(0, \boldsymbol{\Sigma})$

$$\varepsilon_i \overset{\mathrm{IID}}{\sim} \mathcal{N}(0,1), \varepsilon_i \perp \boldsymbol{X}_i$$

第 k 台机器：$\boldsymbol{X}_{A_k,j} \overset{\mathrm{IID}}{\sim} \mathcal{N}(0, \boldsymbol{\Sigma}_{A_k})$.

其中协方差矩阵 $\boldsymbol{\Sigma}$ 的设定如平方模型（d），$\boldsymbol{\Sigma}_{A_k}$ 表示 $\boldsymbol{\Sigma}$ 对应分量 A_k 的子矩阵. 此时，待估参数的真值为 $\boldsymbol{\theta}_0 = \widetilde{\boldsymbol{\theta}}_0 + c_\tau \boldsymbol{e}_1$，其中 c_τ 为标准正态分布的 τ 分位数. 这里选取了两组 $\widetilde{\boldsymbol{\theta}}_0$ 的取值，分别为 $\boldsymbol{\theta}_1 = \boldsymbol{1}_p$，$\boldsymbol{\theta}_2 = (1, 1, 1.1, 2.2, -3.3, 4.4, -5.5, 6.6, -7.7, 8.8)^{\mathrm{T}}$.

（f）平方模型. 考虑如下数据生成模型

$$Y_i = (\widetilde{\boldsymbol{\theta}}_0^{\mathrm{T}} \vec{\boldsymbol{X}}_i)^2 + \varepsilon_i$$

第 1 台机器：$\boldsymbol{X}_i \overset{\mathrm{IID}}{\sim} \mathcal{N}(0, \boldsymbol{\Sigma})$

$$\varepsilon_i \overset{\mathrm{IID}}{\sim} \mathcal{N}(0,1), \varepsilon_i \perp \boldsymbol{X}_i$$

第 k 台机器：$\boldsymbol{X}_{A_k,j} \overset{\mathrm{IID}}{\sim} \mathcal{N}(0, \boldsymbol{\Sigma}_{A_k})$

其中协方差矩阵 $\boldsymbol{\Sigma}$ 的设定如平方模型（d），$\boldsymbol{\Sigma}_{A_k}$ 的定义如线性模型（e）. 此时，Y 在给定 $\boldsymbol{X}$ 的条件下的 τ 条件分位数为 $Q_\tau(Y \mid \boldsymbol{X}) = (\widetilde{\boldsymbol{\theta}}_0^{\mathrm{T}} \vec{\boldsymbol{X}})^2 + c_\tau$. 这里选取 $\widetilde{\boldsymbol{\theta}}_0 = \boldsymbol{1}_p$，通过蒙特卡洛模拟得到待估参数的真值如下

$$\boldsymbol{\theta}_0 = \begin{cases} (21.507, 1.997, 2.008, 1.980, 2.010, \\ 2.009, 2.000, 2.019, 2.007, 1.976)^{\mathrm{T}}, & \tau = 0.5 \\ (5.638, 1.997, 2.008, 1.980, 2.010, \\ 2.009, 2.000, 2.019, 2.007, 1.976)^{\mathrm{T}}, & \tau = 0.25 \end{cases}$$

表 14 与表 15 分别给出了非平衡半监督数据下线性模型的模拟结果. 由表 14 与表 15 中的结果可知，所有四种算法的偏差（Bias）和标准误差（SE）都可以控制在较小的范围. 下面，将基于加权损失函数的通讯有效算法记为 WE 算法，将基于改进的数萃方法的通讯有效算法记为 CD 算法.

表 14　线性模型（e）在非平衡半监督数据下，基于加权损失函数的通讯有效算法的模拟结果

Table 14 Simulation result of linear model (e) for imbalanced semi-supervised data by using the method based on weighted loss functions

参数设定	评价指标	$\boldsymbol{\theta}_0$	$\boldsymbol{\theta}_1$	$\boldsymbol{\theta}_2$	$\boldsymbol{\theta}_3$	$\boldsymbol{\theta}_4$	$\boldsymbol{\theta}_5$	$\boldsymbol{\theta}_6$	$\boldsymbol{\theta}_7$	$\boldsymbol{\theta}_8$	$\boldsymbol{\theta}_9$
$\boldsymbol{\theta}=\boldsymbol{\theta}_1, \tau=0.50$	Bias	0.000	−0.002	0.003	−0.014	−0.009	0.004	0.019	0.000	−0.007	−0.007
	SE	0.131	0.204	0.176	0.170	0.182	0.190	0.166	0.178	0.176	0.187
$\boldsymbol{\theta}=\boldsymbol{\theta}_1, \tau=0.25$	Bias	0.007	−0.010	0.033	0.007	−0.018	0.005	−0.003	−0.002	−0.002	0.005
	SE	0.145	0.216	0.196	0.187	0.195	0.191	0.180	0.181	0.206	0.202
$\boldsymbol{\theta}=\boldsymbol{\theta}_2, \tau=0.50$	Bias	0.002	−0.003	0.004	−0.002	−0.004	0.029	0.003	−0.003	0.008	−0.025
	SE	0.115	0.170	0.171	0.191	0.177	0.208	0.207	0.172	0.177	0.172
$\boldsymbol{\theta}=\boldsymbol{\theta}_2, \tau=0.25$	Bias	0.027	0.006	0.008	0.008	0.026	−0.007	−0.006	0.006	−0.035	−0.017
	SE	0.154	0.196	0.183	0.207	0.182	0.211	0.193	0.208	0.195	0.179

表 15 线性模型（e）在非平衡半监督数据下，基于改进的数萃（Meta）方法的通讯有效算法的模拟结果

Table 15 Simulation result of linear model (e) for imbalanced semi-supervised data by using the improved Meta method

参数设定	评价指标	$\boldsymbol{\theta}_0$	$\boldsymbol{\theta}_1$	$\boldsymbol{\theta}_2$	$\boldsymbol{\theta}_3$	$\boldsymbol{\theta}_4$	$\boldsymbol{\theta}_5$	$\boldsymbol{\theta}_6$	$\boldsymbol{\theta}_7$	$\boldsymbol{\theta}_8$	$\boldsymbol{\theta}_9$
$\boldsymbol{\theta}=\boldsymbol{\theta}_1,\tau=0.50$	Bias	0.017	0.000	0.016	0.021	−0.003	−0.003	−0.001	0.001	0.012	−0.025
	SE	0.137	0.169	0.170	0.189	0.167	0.176	0.155	0.161	0.173	0.171
$\boldsymbol{\theta}=\boldsymbol{\theta}_1,\tau=0.25$	Bias	0.007	−0.004	−0.004	−0.016	−0.005	0.034	0.005	−0.024	−0.003	−0.010
	SE	0.140	0.174	0.187	0.191	0.181	0.176	0.169	0.197	0.184	0.190
$\boldsymbol{\theta}=\boldsymbol{\theta}_2,\tau=0.50$	Bias	−0.012	0.016	0.010	0.002	−0.012	0.017	0.012	−0.005	−0.014	0.004
	SE	0.128	0.178	0.170	0.183	0.177	0.167	0.166	0.164	0.159	0.173
$\boldsymbol{\theta}=\boldsymbol{\theta}_2,\tau=0.25$	Bias	0.018	−0.037	0.008	0.001	0.032	−0.003	−0.001	0.013	−0.007	0.002
	SE	0.132	0.175	0.196	0.186	0.185	0.184	0.189	0.180	0.186	0.200

对于模型误设的情形，模拟主要关心的是与有监督学习相比，利用无标签数据能使得估计量的估计效率提高多少. 因为 CD 算法的估计量的方差估计涉及到对其渐近性质的进一步推导，因此这里只考虑 WE 算法对应的估计量 $\widehat{\boldsymbol{\theta}}_{\mathrm{WE}}$. 表 16 给出了估计量 $\widehat{\boldsymbol{\theta}}_{\mathrm{WE}}$ 的模拟结果. 除了表 14 与表 15 中给出的评价指标之外，表 16 进一步给出了 $\widehat{\boldsymbol{\theta}}_{\mathrm{WE}}$ 的估计效率. 此处估计效率的定义与数据完整观测的情况有所不同. 记仅利用无标签数据得到的有监督学习下的估计量为 $\widehat{\boldsymbol{\theta}}_S$，即 $\widehat{\boldsymbol{\theta}}_S=\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{\operatorname{argmin}}\frac{1}{n}\sum_{i=1}^{n}\rho_\tau(Y_i-\boldsymbol{\theta}^{\mathrm{T}}\times\vec{\boldsymbol{X}}_i)$. 对于 $\boldsymbol{\theta}_0$ 的任一估计量 $\widehat{\boldsymbol{\theta}}_S$，令 $\mathrm{MES}(\widehat{\boldsymbol{\theta}})$ 表示 $\widehat{\boldsymbol{\theta}}$ 的平均平方误差. 定义 $\widehat{\boldsymbol{\theta}}_{\mathrm{WE}}$ 相对于 $\widehat{\boldsymbol{\theta}}_S$ 的估计效率为 $\mathrm{Efficiency}(\widehat{\boldsymbol{\theta}}_{\mathrm{WE}})=\{\mathrm{MES}(\widehat{\boldsymbol{\theta}}_S)-\mathrm{MES}(\widehat{\boldsymbol{\theta}}_{\mathrm{WE}})\}/\mathrm{MES}(\widehat{\boldsymbol{\theta}}_S)$.

由表 16 可知，在平方模型的设定下，截距项估计量的偏差较线性模型有所增加，参数估计量的标准误差也较线性模型有比较明显的增加. 进一步，$\widehat{\boldsymbol{\theta}}_{\mathrm{WE}}$ 相比$\widehat{\boldsymbol{\theta}}_S$ 具有更高的估计效率，且对于大部分参数，相对效率能达到 15% 以上. 这说明通过 WE 算法的方式加入无标签数据可以显著提高估计效率.

表 16 平方模型（f）在非平衡半监督数据下，基于加权损失函数的通讯有效算法的模拟结果

Table 16 Simulation result of quadratic model (f) for imbalanced semi-supervised data by using the method based on weighted loss functions

分位数水平	评价指标	$\boldsymbol{\theta}_0$	$\boldsymbol{\theta}_1$	$\boldsymbol{\theta}_2$	$\boldsymbol{\theta}_3$	$\boldsymbol{\theta}_4$	$\boldsymbol{\theta}_5$	$\boldsymbol{\theta}_6$	$\boldsymbol{\theta}_7$	$\boldsymbol{\theta}_8$	$\boldsymbol{\theta}_9$
$\tau=0.50$	Bias	2.566	0.680	0.084	0.070	−0.227	−0.031	−0.569	−0.172	0.123	−0.263
	SE	4.633	5.636	5.341	5.450	5.406	5.432	5.845	5.880	5.266	6.754
	估计效率	0.389	0.175	0.272	0.172	0.137	0.157	0.164	0.102	0.047	0.128
$\tau=0.25$	Bias	1.699	0.051	0.052	0.108	−0.150	−0.265	0.150	−0.005	−0.019	0.068
	SE	2.007	2.459	2.657	2.741	2.578	2.623	2.335	2.391	2.613	2.417
	估计效率	0.244	0.267	0.137	0.207	0.207	0.226	0.187	0.331	0.197	0.248

7 实际数据分析

本节将本文所提出的算法运用到一个实际的数据场景中. 数据集取自 2004 年至 2005 年间，洛杉矶流浪人口服务部（Los Angeles Homeless Services Authority）关于洛杉矶城一共 2 054 个管理区域的流浪人口的追踪调查. 由于即时而完备的街道考查是一项非常困难的工作，该项调查选取了 244 个几乎依概率 1 观察到流浪汉人群的地区为重点区域. 这些重点区域都有比较可靠的人口普查统计，而余下的 1 810 个区域则不一定存在完整的追踪调查. 在这 1 810 个非重点区域中，又随机挑选了 265 个区域抽查，同样也获得了较可靠的人口普查统计. 所以一共有 509 个城市区域已经完成了其中流浪人口的追踪统计，而余下 1 545 个区域则没有统计具体的流浪人口数量. 在本次的研究中，每个地区的流浪人口数量将视为响应变量 Y，那么其中 1 545 个响应变量存在缺失.

同时，所有的区域均考查了一些被认为与流浪人口数量相关的变量. Kriegler 和 Berk[24] 对该

调查的相关变量有较详细的说明与分析，他们提到被认为对流浪人口数量影响最大的因素有：工业用地占比、商业用地占比、居住区用地占比、空置土地面积占比、非高加索人口占比、有户主房屋占比、空置房屋占比、中位居民收入，一共八项. 其中，由于有户主房屋占比与空置房屋占比可以认为有强相关关系，所以仅考虑有户主房屋占比一项即可. 因此可将上述七项重要的相关变量选取为协变量 X. 值得注意的是，在进行实际建模时对数据做了标准化处理，而对应的线性模型并不包含截距项. 本次分析考虑到了五个不同的分位数水平 $\tau = 0.10, 0.25, 0.50, 0.75, 0.90$，并计算各水平下的参数真值及其估计量.

接下来运用本文所提出的算法进行参数估计. 首先，运用第 2 节和第 3 节给出的 EC 算法与 SJ 算法. 由于 EC 算法和 SJ 算法并不考虑数据缺失的问题，所以关于这两个算法，仅针对无缺失的 509 个观测样本进行分析. 分析时先对完整数据进行分块. 由于原数据中重点区域与非重点区域的记录比较规整，为了使得分块之间的数据更加均匀. 在设置分块前会重新将数据集随机打乱，以得到分布更为相似的数据分块. 将 509 条数据随机均分为 3 块，分别包含 170 条，170 条与 169 条观测. 默认以第一分块为算法的中心机器所在的分块，并且为了符合分块场景的设定，初值估计仅由第一子块数据的估计产生. 而对于设定的线性模型，为事先消除截距项与量纲的影响，对数据集做标准化处理. 对于迭代算法，这里也采用 5 步迭代. 光滑化方法中的窗宽按照第 3 节中所述方法统一取为 $h = (7/169)^{1/4} = 0.45$. 表 17 给出了不同分位数水平下，本文所提出的所有非半监督算法的计算结果. 同时，作为对照组，结果中也列出直接用全部数据估计的结果，并且同时也给出朴素平均（AM）的结果，以及 Wang 等[21]的 WW 算法的结果，用以评估不同算法间的差别.

研究希望本文提出的分布式算法的估计结果能与 Oracle 估计相似，即符号相同，数量级相近. 表 17 显示②，在同分布以及非异方差模型的模拟中一致地有着高效率表现的 WW 算法，应用到这个实例数据上时，无论在哪个分位数水平下，都与 Oracle 估计存在相当的差异. 这就表明，在真实数据下，往往难以得到明确具有同质性且严格满足同分布条件的场景. 而在这种现实场景中，本文提出的估计方法则能表现出很明显的优势. 首先是基于光滑逼近的 SJ 算法，在任何一个分位数水平下基本都能与 Oracle 估计保持一致，而且其仅需要在低通讯成本下就可以完成计算. 这就大大地优于 WW 算法的实际效果. 而各种基于改进的数萃方法所得的估计则在各种异质性数据下有稳健的表现. 特别地，注意到 0.5 分位数水平下，MV，SP 与 MB 算法对应的三种估计都一致地与 Oracle 估计保持很高的相似程度. 比较遗憾的是，在实际数据中，子块间的数据分布存在一定的差异，由上述方法推广出的仅运用一个子块上的信息近似协方差矩阵的 Local 算法相对逊色. 但是对于需要高效通讯的场合，本实例还是说明了 SJ 算法的优势，即仍然能为如同本实例一样的现实分布式计算场景提供降低通讯成本的可行性.

进一步，考虑加入无标签数据的信息，即考虑非平衡半监督数据的通讯有效算法，采用第 5 节中提出的 WE 算法和 CD 算法，重新进行参数估计. 这里将所有有标签数据放在第一台机器，并将其作为中心机器，则一号机上的样本量为 $n = 509$. 无标签数据共有 1 545 条，它被随机均分到 3 台子机器上，每台子机器上的样本量为 $n_k = 515$.

表 18 的前 10 行给出了本文提出的 EC 算法与 SJ 算法的参数估计结果，作为非平衡半监督算法的对照组. 后 10 行给出了 WE 算法与 CD 算法的参数估计结果. 从中可知，由于加入了无标签数据，模型的不确定性增大，使得此时关于不同分位数水平的参数估计结果出现了差异，但是大部分差异较小，即参数估计仍可以保持较好的一致性. 个别差异稍大的参数估计结果可能是由于方差较大导致的，需要通过进一步针对方差估计的研究. 由此，综合分析表 17 与表 18 中的结果可

② 在表 17 中，SJ2 算法在 0.75 与 0.90 分位数下的‘–’记号表示，在此分位数下，难以得到可逆的黑塞矩阵完成计算，故算法失效. 但此处的失效，仅能说明光滑逼近算法在此数据集中存在算数意义上的偶然不可行性. 与 SJ2 算法同源的 SJ1 算法，因无需求解黑塞矩阵而可以直接进行优化，可以得到很好的结果. 而在其它可以正常计算的分位数水平下，SJ2 算法表现良好.

知,对于所考虑的数据集,利用本文所提出的四种算法均可以得到较为准确的参数估计结果,主要表现为符号的一致性与数值的准确性.

通过对表 17 和表 18 的数据进行总结可知,在本文所考虑的七个协变量中,居住区用地占比,有户主房屋占比和非高加索人口占比对流浪人口数具有负向作用,即随着这三个变量取值的增加,流浪人口数会减少,而其他四个变量的增加则伴随着流浪人口数的增加. 因此,在进行公共管理政策研究时,特别是有关城区管理与资源调配的政策制定时,若无法通过直接的大规模调查统计流浪人口数量,则可以通过本文所考虑的七个经济社会变量对其进行估计,或通过观察这七个协变量的情况对不同区域的流浪人口数进行比较,以通过较低的调查成本得到相对精确的估计,从而实现更合理的社会资源调配.

表 17 完整观测的实际数据下不同算法的分析结果

Table 17 Result of real data analysis of different algorithms for fully observe data

分位数水平	算法	工业用地占比	居住区用地占比	空置土地面积占比	商业用地占比	有户主房屋占比	非高加索人口占比	中位居民收入
$\tau = 0.10$	EC	0.014	−0.045	0.317	0.122	−0.062	−0.019	0.020
	SJ1	0.037	−0.021	0.224	0.068	−0.065	−0.021	0.021
	SJ2	−0.132	0.710	−2.080	−0.342	−0.554	0.043	−0.217
	MV	−0.118	−0.049	0.075	0.363	0.020	0.049	0.014
	SP	0.012	−0.188	0.069	0.124	−0.146	0.042	0.090
	MB	−0.029	0.003	0.397	0.121	−0.232	0.116	0.297
	SP-Local	−0.017	0.355	0.084	−0.044	−0.085	0.003	0.004
	MB-Local	−0.110	−0.553	0.969	0.032	−0.715	0.444	0.935
	WW	0.004	0.064	0.075	−0.045	−0.069	−0.016	0.001
	AM	0.008	−0.374	0.122	−0.135	−0.050	−0.017	−0.004
	Oracle	0.041	−0.023	0.253	0.060	−0.087	−0.038	0.020
$\tau = 0.25$	EC	−0.002	0.027	0.152	0.025	−0.131	−0.042	0.025
	SJ1	0.034	−0.011	0.160	0.031	−0.101	−0.042	0.026
	SJ2	0.036	−0.005	0.174	0.042	−0.108	−0.040	0.046
	MV	−0.118	−0.049	0.075	0.363	0.020	0.049	0.014
	SP	0.054	−0.055	0.170	0.078	−0.118	0.002	0.038
	MB	−0.029	0.003	0.397	0.121	−0.232	0.116	0.297
	SP-Local	−0.019	0.191	0.078	−0.018	−0.129	−0.023	0.059
	MB-Local	0.032	−0.329	0.354	0.114	−0.079	0.155	0.194
	WW	−0.011	−0.011	0.075	−0.031	−0.087	−0.021	0.015
	AM	0.023	−0.317	0.112	−0.121	−0.075	−0.026	0.011
	Oracle	0.041	−0.023	0.253	0.060	−0.087	−0.038	0.020
$\tau = 0.50$	EC	0.019	−0.087	0.285	0.118	−0.172	−0.027	0.146
	SJ1	0.039	−0.019	0.184	0.045	−0.086	−0.033	0.024
	SJ2	0.054	−0.019	0.096	0.052	−0.111	−0.006	0.042
	MV	0.004	−0.010	0.100	0.094	−0.069	0.004	0.015
	SP	0.057	−0.029	0.161	0.070	−0.121	−0.021	0.033
	MB	0.043	−0.022	0.209	0.057	−0.141	−0.014	0.106
	SP-Local	0.019	0.109	0.074	−0.003	−0.128	0.001	0.035
	MB-Local	0.046	−0.037	0.190	−0.022	−0.097	0.034	0.055
	WW	0.010	0.000	0.101	0.001	−0.102	−0.016	0.014
	AM	0.033	−0.152	0.168	−0.058	−0.091	−0.032	0.018
	Oracle	0.041	−0.023	0.253	0.060	−0.087	−0.038	0.020

续表 17

Table 17 Continues

分位数水平	算法	工业用地占比	居住区用地占比	空置土地面积占比	商业用地占比	有户主房屋占比	非高加索人口占比	中位居民收入
$\tau = 0.75$	EC	0.028	−0.029	0.329	0.069	−0.049	0.021	0.050
	SJ1	0.042	−0.023	0.195	0.049	−0.076	−0.025	0.021
	SJ2	–	–	–	–	–	–	–
	MV	0.096	0.008	0.110	0.034	−0.087	−0.050	−0.012
	SP	0.098	−0.007	0.237	0.042	−0.059	−0.024	0.028
	MB	0.106	−0.014	0.189	0.053	−0.101	−0.008	0.087
	SP-Local	0.039	0.067	0.239	0.014	−0.087	0.025	0.062
	MB-Local	0.082	0.090	0.218	−0.010	−0.107	0.035	0.083
	WW	0.128	−0.031	0.188	0.173	−0.120	0.010	0.101
	AM	0.188	0.053	0.261	0.077	−0.115	−0.031	0.093
	Oracle	0.041	−0.023	0.253	0.060	−0.087	−0.038	0.020
$\tau = 0.90$	EC	0.068	−0.015	0.224	0.039	−0.131	−0.060	0.045
	SJ1	0.037	−0.025	0.189	0.045	−0.081	−0.027	0.022
	SJ2	–	–	–	–	–	–	–
	MV	−0.023	−0.175	−0.079	−0.138	−0.446	0.034	0.208
	SP	0.002	−0.056	0.010	0.104	−0.075	−0.078	−0.006
	MB	−0.179	−0.040	0.063	0.011	−0.033	−0.088	0.036
	SP-Local	0.215	−0.328	−0.145	0.134	−0.332	−0.194	0.223
	MB-Local	−0.785	0.277	0.112	−0.097	0.402	−0.023	−0.444
	WW	0.243	−0.034	0.377	0.239	−0.064	0.056	0.147
	AM	0.095	0.350	0.530	0.188	−0.064	−0.078	0.114
	Oracle	0.041	−0.023	0.253	0.060	−0.087	−0.038	0.020

表 18　非平衡半监督实际数据下不同算法的分析结果

Table 18 Result of real data analysis of different algorithms for imbalanced semi-supervised data

算法	分位数水平	工业用地占比	居住区用地占比	空置土地面积占比	商业用地占比	有户主房屋占比	非高加索人口占比	中位居民收入
EC	0.10	0.014	−0.045	0.317	0.122	−0.062	−0.019	0.020
	0.25	−0.002	0.027	0.152	0.025	−0.131	−0.042	0.025
	0.50	0.019	−0.087	0.285	0.118	−0.172	−0.027	0.146
	0.75	0.028	−0.029	0.329	0.069	−0.049	0.021	0.050
	0.90	0.068	−0.015	0.224	0.039	−0.131	−0.060	0.045
SJ1	0.10	0.037	−0.021	0.224	0.068	−0.065	−0.021	0.021
	0.25	0.034	−0.011	0.160	0.031	−0.101	−0.042	0.026
	0.50	0.039	−0.019	0.184	0.045	−0.086	−0.033	0.024
	0.75	0.042	−0.023	0.195	0.049	−0.076	−0.025	0.021
	0.90	0.037	−0.025	0.189	0.045	−0.081	−0.027	0.022
WE	0.10	0.013	−0.015	0.303	0.061	−0.070	−0.044	0.026
	0.25	0.008	−0.027	0.299	0.067	−0.061	−0.030	0.016
	0.50	0.048	−0.022	0.305	0.046	−0.060	−0.026	0.017
	0.75	0.041	−0.026	0.248	0.026	−0.090	−0.024	0.010
	0.90	0.126	−0.021	0.225	0.023	−0.083	−0.030	0.013

续表 18

Table 18 Continues

算法	分位数水平	工业用地占比	居住区用地占比	空置土地面积占比	商业用地占比	有户主房屋占比	非高加索人口占比	中位居民收入
CD	0.10	0.007	−0.019	0.294	0.062	−0.071	−0.042	0.027
	0.25	0.014	−0.026	0.291	0.064	−0.068	−0.033	0.022
	0.50	0.058	−0.024	0.285	0.051	−0.069	−0.030	0.020
	0.75	0.032	−0.028	0.257	0.032	−0.090	−0.016	0.028
	0.90	0.078	−0.034	0.240	0.025	−0.094	−0.033	0.026

注：在半监督问题中 EC 和 SJ 算法只考虑了有标签数据. WE 和 CD 算法利用了所有有标签与无标签数据.

8 结束语

本文针对分位回归模型提出了几种大数据下的通讯有效算法. 从损失函数的角度出发,考虑了分位数损失下的最小风险线性预报. 本文的方法不对真实模型做任何假设,即为无模型（model-free）设定,从而可以提高算法的稳健性. 从线性预报的角度出发,给出的最小风险预报可以看作真实条件分位数的一阶线性逼近,具有更快的计算效率和更好的可解释性. 换言之,本文允许分位数回归模型存在模型误设,即真实的条件分位数与协变量之间并不符合线性关系,而具有更一般的形式. 在这种情况下,线性模型可以看成对真实模型的近似. 对于近似的线性模型,仍可以定义参数的真值和估计量.

本文针对完整观测数据和非平衡半监督数据两种情形进行了考虑. 一方面,对完整观测数据,本文提出了三种通讯有效的数据融合算法. 首先,利用目标函数的次导数,提出了基于等度连续的数据融合算法. 其次,通过对非光滑的目标函数进行光滑近似,提出了基于光滑逼近的数据融合算法. 以上两种算法均可以将通讯成本控制在 $O(TKp)$ 的阶数,从而实现了通讯有效性. 进一步,基于广义矩估计（GMM）的思想,提出了改进的数萃（Meta）方法. 本文将每台机器上的估计量看作样本观测,构造估计方程和最优化目标函数. 改进的数萃方法可以广泛应用于数据同分布与存在异质性的分布式系统,作为一般性框架,通过选取不同的估计函数和权重矩阵,可以将许多现有方法涵盖在内. 不论不同机器上的数据的分布是否相同,其均可以得到有效估计,且在数据同分布时实现有效通讯. 另一方面,针对非平衡半监督数据,本文根据每台子机器上数据样本量的不同情况讨论,提出了两种通讯有效算法. 当每台子机器上数据的样本量较小时,利用数据插补和变量降维的思想,将子机器上的无标签数据加入到待优化的目标函数中,来提高总体参数的估计效率. 当存在一台子机器上数据的样本量较大时,基于改进的数萃方法通过对协方差矩阵进行近似,提出了另一种新的分布式算法,同样可以显著提高通讯效率.

本文对所提出的算法进行了大量的仿真模拟,分析结果表明这些算法都具有较好的统计推断表现. 最后,还针对实际数据集进行了分析,验证了所提出的算法在实际数据中的良好表现. 目前,团队正在针对这些算法的理论性质进行研究,如渐近正态性的理论证明和命题成立的技术性条件等. 值得注意的是,虽然本文的出发点仅考虑分位数损失,但所提出的大多数算法都可以应用到其它更一般的模型中.

参 考 文 献：

[1]王璞玉, 张 海. 分布式隐私保护——Logistic 回归[J]. 中国科学：信息科学, 2020, 50(10)：1511 − 1528.
Wang Puyu, Zhang Hai. Distributed logistic regression with differential privacy[J]. Scientia Sinica Informations, 2020, 50(10)：1511 − 1528. (in Chinese)

[2]衣 鹏, 洪奕光. 分布式合作优化及其应用[J]. 中国科学：数学, 2016, 46(10)：1547 − 1564.
Yi Peng, Hong Yiguang. Distributed cooperative optimization and its applications[J]. Scientia Sinica Mathematica, 2016,

46(10)：1547 – 1564. (in Chinese)
[3]Zhang Y C, Duchi J C, Wainwright M J. Communication-efficient algorithms for statistical optimization[J]. The Journal of Machine Learning Research, 2013, 14(1)：3321 – 3363.
[4]Jaggi M, Smith V, Takac M, et al. Communication-efficient distributed dual coordinate ascent[C]. Advances in Neural Information Processing Systems, 2014：27.
[5]Cai Z W, Wang X. Nonparametric estimation of conditional VaR and expected shortfall[J]. Journal of Econometrics, 2008, 147：120 – 130.
[6]刘晓倩, 周　勇. 风险度量半参数变系数复合 Expectile 回归模型及其应用[J]. 系统工程理论与实践, 2020, 40(8)：2176 – 2192.
Liu Xiaoqian, Zhou Yong. The semiparametric varying-coefficient composite expectile regression model in risk measurement and its application[J]. Systems Engineering：Theory and Practice, 2020, 40(8)：2176 – 2192. (in Chinese)
[7]王子剑, 周　勇, 曾凡平. 在相依样本下半参数变系数期望分位数风险度量模型统计推断[J]. 中国科学：数学, 2021, 51(9)：1377 – 1406.
Wang Zijian, Zhou Yong, Zeng Fanping. Semiparametric varying-coefficient expectile model for estimating value at risk on dependent samples[J]. Scientia Sinica Mathematica, 2021, 51(9)：1377 – 1406. (in Chinese)
[8]伏红勇, 但　斌, 王　磊, 等. CVaR 准则下“公司 + 农户”模式的天气看跌期权契约[J]. 管理科学学报, 2020, 23(11)：59 – 73.
Fu Hongyong, Dan Bin, Wang Lei, et al. Weather put option contract for “company & farmer” pattern under CVaR criterion [J]. Journal of Management Sciences in China, 2020, 23(11)：59 – 73. (in Chinese)
[9]黄　潇, 黄守军. 流动人口自雇的决定机制及收入差异[J]. 管理科学学报, 2021, 24(6)：57 – 75.
Huang Xiao, Huang Shoujun. The determination mechanism and income difference of migrants’ self-employment in China [J]. Journal of Management Sciences in China, 2021, 24(6)：57 – 75. (in Chinese)
[10]Chen X, Liu W, Zhang Y. Quantile regression under memory constraint[J]. The Annals of Statistics, 2019, 47(6)：3244 – 3273.
[11]Chen L, Zhou Y. Quantile regression in big data：A divide and conquer based strategy[J]. Computational Statistics & Data Analysis, 2020, 144：106892.
[12]Singh K, Xie M, Strawderman W E, et al. Combining information from independent sources through confidence distributions[J]. The Annals of Statistics, 2005, 33(1)：159 – 183.
[13]潘莹丽, 刘　展, 蔡　雯. 大数据背景下基于 expectile 回归模型的分布式优化方法研究[J]. 数学的实践与认识, 2020, 50(14)：259 – 268.
Pan Yingli, Liu Zhan, Cai Wen. Research on distributed optimization method based on expectile regression model under the background of big data[J]. Mathematics in Practice and Theory, 2020, 50(14)：259 – 268. (in Chinese)
[14]Jordan M, Lee J, Yang Y. Communication-efficient distributed statistical inference[J]. Journal of the American Statistical Association, 2019, 114(526)：668 – 681.
[15]Fan J, Guo Y, Wang K. Communication-efficient accurate statistical estimation[J]. Journal of the American Statistical Association, 2021：1 – 11.
[16]Wang L, Lian H. Communication-efficient estimation of high-dimensional quantile regression[J]. Analysis and Applications, 2020, 18(6)：1057 – 1075.
[17]Duan R, Ning Y, Chen Y. Heterogeneity-aware and communication-efficient distributed statistical inference[J]. Biometrika, 2022, 109(1)：67 – 83.
[18]Chen S X, Hall P. Smoothed empirical likelihood confidence intervals for quantiles[J]. The Annals of Statistics, 1993, 21：1166 – 1181.
[19]Lin H, Peng H. Smoothed rank correlation of the linear transformation regression model[J]. Computational Statistics & Data Analysis, 2012, 57：615 – 630.
[20]周　勇. 广义估计方程估计方法[M]. 北京：科学出版社, 2013.
Zhou Yong. Generalized Estimating Equation Methods[M]. Beijing：Science Press, 2013. (in Chinese)
[21]Wang K L, Wang H W, Li S M. Renewable quantile regression for streaming datasets[J]. Knowledge-Based Systems, 2022, 235：10767.
[22]Hall P, Racine J, Li Q. Cross-validation and the estimation of conditional probability densities[J]. Journal of the American Statistical Association, 2004, 99(468)：1015 – 1026.
[23]Zeng D, Lin D Y. Efficient resampling methods for nonsmooth estimating functions[J]. Biostatistics, 2008, 9(2)：355 – 363.
[24]Kriegler B, Berk R. Small area estimation of the homeless in Los Angeles：An application of cost-sensitive stochastic gradi-

ent boosting[J]. The Annals of Applied Statistics, 2010: 1234 – 1255.

Communication-efficient algorithms for quantile regression and their applications in the framework of big data analysis

ZHOU Yong, *ZHANG Shu-yi**, *LI Zi-yang*

Key Laboratory of Advanced Theory and Application in Statistics and Data Science, Ministry of Education, Academy of Statistics and Interdisciplinary Sciences and School of Statistics, East China Normal University, Shanghai 200062, China

Abstract: This paper considers the widely used quantile regression models in risk measurement, and proposes several fast distributed algorithms to address huge datasets and complex data types. Although our proposed algorithms are based on quantile regression models, most of them can be applied to more general models. Since the objective function of quantile regression is non-smooth, the usual divide-and-conquer strategy and communication-efficient algorithms for smooth functions are not applicable. This paper first gives three communication-efficient algorithms for completely observed data: The equicontinuity-based method, the smooth approximation method and the improved Meta method. Second, for imbalanced semi-supervised data, the method based on weighted loss function and the improved Meta method are proposed for small-sized and large sized unlabeled datasets, respectively. The proposed algorithms can integrate the semi-supervised data scattered on different machines, so as to realize communication-efficient distributed computing for different data types and different sample sizes, and improve the accuracy of the algorithm and the efficiency of the parameter estimation. The finite sample performance of the proposed algorithms is investigated through extensive simulation studies. Applications to the homeless data in Log Angeles are presented, which illustrate the accuracy of the proposed algorithms.

Key words: big data analysis; data fusion; communication-efficient algorithm; quantile regression

附录

本附录给出了第七节数值模拟分析的补充结果, 包含了数据完整观测情形下线性同方差模型 (a), 不同机器数据同分布的线性异方差模型 (b), 不同机器数据分布不同的线性异方差模型 (c), 平方模型 (d) 在总样本量 $N = 10\,000$ 和总机器数 $K = 10$ 设定下的模拟结果. Bias 和 SE 基于参数的各个分量分别计算, 标号 0, 标号 1, 标号 2 分别表示截距项和对应协变量 X 的分量.

附表 1 同质性分布式系统下线性同方差模型 (a) 的补充模拟结果

Attached Table 1 Additional simulation result of homoscedastic linear model (a) under homogenous distributed system

分位数水平	算法	Bias0 (10^{-3})	Bias1 (10^{-3})	Bias2 (10^{-3})	SE0 (10^{-2})	SE1 (10^{-2})	SE2 (10^{-2})	估计效率	计算时长/s	通讯成本
$\tau = 0.50$	Oracle	1.47	3.45	1.09	3.60	4.22	4.83	1.00	9.46	–
	EC	3.16	−4.15	−4.74	4.43	5.22	5.67	0.68	0.44	$O(TKp)$
	SJ1	1.33	−2.95	−1.64	3.64	4.15	4.83	1.01	0.46	$O(TKp)$
	SJ2	1.51	−3.22	−1.77	3.61	4.13	4.82	1.01	0.40	$O(TKp)$
	MV	5.44	−5.23	−0.75	6.06	6.95	6.50	0.42	0.42	$O(TK(p^2+p))$
	SP	1.50	−4.54	−0.69	3.54	4.20	4.85	1.00	0.36	$O(K(p^2+p))$
	MB	1.75	−4.76	−0.74	3.57	4.30	4.87	0.98	1.33	$O(K(p^2+p))$
	SP-Local	1.17	−3.76	1.16	3.54	4.47	5.10	0.93	0.36	$O(Kp)$
	MB-Local	1.14	−5.59	1.76	3.52	4.60	5.02	0.92	1.33	$O(Kp)$
	ML	1.70	−3.52	−1.50	3.62	4.26	4.84	0.98	0.70	$O(TK(p^2+p))$
	DC	2.11	−5.85	−0.45	3.60	4.43	4.89	0.95	1.22	$O(K(p^2+p))$
	WW	1.44	−4.96	−0.73	3.45	4.20	4.66	1.05	0.32	$O(K(p^2+p))$

续附表 1

Attached Table 1 Continues

分位数水平	算法	Bias0 (10^{-3})	Bias1 (10^{-3})	Bias2 (10^{-3})	SE0 (10^{-2})	SE1 (10^{-2})	SE2 (10^{-2})	估计效率	计算时长/秒	通讯成本
$\tau = 0.25$	Oracle	-2.37	-2.70	7.55	3.23	5.15	4.23	1.00	10.20	-
	EC	-0.60	-1.72	1.16	4.82	5.60	6.19	0.60	0.48	$O(TKp)$
	SJ1	-6.61	-0.62	6.47	3.36	5.17	4.08	1.00	0.49	$O(TKp)$
	SJ2	-7.20	-1.63	6.62	3.30	5.18	4.10	1.00	0.43	$O(TKp)$
	MV	3.73	-8.63	6.47	5.31	7.27	5.86	0.43	0.55	$O(TK(p^2+p))$
	SP	1.24	-3.13	6.50	3.46	5.33	4.12	0.96	0.39	$O(K(p^2+p))$
	MB	1.23	-2.97	6.48	3.53	5.38	4.19	0.93	1.34	$O(K(p^2+p))$
	SP-Local	1.25	-3.15	5.19	3.38	5.63	4.57	0.86	0.39	$O(Kp)$
	MB-Local	1.41	-3.55	5.82	3.65	5.83	4.63	0.80	1.33	$O(Kp)$
	ML	-2.36	-2.70	7.55	3.22	5.13	4.22	1.01	0.72	$O(TK(p^2+p))$
	DC	-0.51	-2.60	7.42	3.40	5.18	4.38	0.95	1.26	$O(K(p^2+p))$
	WW	-2.02	-2.42	6.78	3.23	4.13	4.06	1.02	0.34	$O(K(p^2+p))$

附表 2　同质性分布式系统下线性异方差模型（b）的补充模拟结果

Attached Table 2 Additional simulation result of heteroscedastic linear model (b) under homogenous distributed system

分位数水平	算法	Bias0 (10^{-3})	Bias1 (10^{-3})	Bias2 (10^{-3})	SE0 (10^{-2})	SE1 (10^{-2})	SE2 (10^{-2})	估计效率	计算时长/秒	通讯成本
$\tau = 0.50$	Oracle	-3.18	-5.27	9.16	3.92	5.11	5.20	1.00	9.67	-
	EC	-5.10	5.03	3.84	4.98	6.75	6.98	0.58	0.44	$O(TKp)$
	SJ1	-2.89	-2.58	7.63	3.81	5.75	5.20	0.92	0.46	$O(TKp)$
	SJ2	-3.08	-3.62	8.51	3.81	5.43	5.05	0.99	0.41	$O(TKp)$
	MV	-0.50	-2.85	3.28	6.69	8.28	8.97	0.36	0.52	$O(TK(p^2+p))$
	SP	-4.46	-4.12	10.40	3.94	5.48	5.26	0.93	0.36	$O(K(p^2+p))$
	MB	-4.30	-4.26	9.98	3.96	5.60	5.35	0.91	1.35	$O(K(p^2+p))$
	SP-Local	0.18	-11.30	9.78	4.01	5.93	5.00	0.88	0.36	$O(Kp)$
	MB-Local	0.88	-13.30	8.86	4.10	5.88	5.66	0.81	1.34	$O(Kp)$
	ML	-2.86	-5.58	8.74	3.94	5.10	5.24	0.99	0.70	$O(TK(p^2+p))$
	DC	-4.42	-3.07	10.70	4.01	5.56	5.18	0.92	1.23	$O(K(p^2+p))$
	WW	-3.88	-3.44	10.40	3.82	5.29	4.93	1.02	0.32	$O(K(p^2+p))$
$\tau = 0.25$	Oracle	1.81	7.36	-6.58	4.15	5.91	5.57	1.00	9.86	-
	EC	3.72	0.40	-8.50	5.01	7.61	7.45	0.60	0.37	$O(TKp)$
	SJ1	12.10	0.59	-6.97	4.83	6.14	6.75	0.78	0.42	$O(TKp)$
	SJ2	12.30	2.67	-7.48	4.19	5.92	5.78	0.96	0.36	$O(TKp)$
	MV	42.00	-12.40	-4.84	6.72	8.12	8.82	0.40	0.54	$O(TK(p^2+p))$
	SP	5.36	7.84	-7.00	4.15	6.20	5.62	0.95	0.38	$O(K(p^2+p))$
	MB	5.10	7.77	-7.01	4.21	6.24	5.59	0.94	1.34	$O(K(p^2+p))$
	SP-Local	4.29	8.24	-6.97	4.50	6.94	6.38	0.76	0.38	$O(Kp)$
	MB-Local	4.77	4.94	-4.12	4.79	7.11	6.60	0.72	1.33	$O(Kp)$
	ML	1.62	7.56	-6.35	4.17	5.94	5.58	0.99	0.72	$O(TK(p^2+p))$
	DC	2.36	7.20	-4.99	4.38	6.27	5.72	0.92	1.23	$O(K(p^2+p))$
	WW	2.69	5.59	-6.43	3.98	5.99	5.33	1.04	0.34	$O(K(p^2+p))$

附表 3 异质性分布式系统下线性异方差模型(c) 的补充模拟结果

Attached Table 3 Additional simulation result of heteroscedastic linear model (c) under heterogeneous distributed system

分位数水平	算法	Bias0 (10^{-3})	Bias1 (10^{-3})	Bias2 (10^{-3})	SE0 (10^{-2})	SE1 (10^{-2})	SE2 (10^{-2})	估计效率	计算时长/秒	通讯成本
$\tau = 0.50$	Oracle	-0.15	1.49	-0.14	0.72	1.17	1.22	1.00	9.73	-
	MV	-0.86	1.31	1.65	0.74	1.16	1.17	1.02	0.54	$O(TK(p^2+p))$
	SP	-0.03	0.69	0.46	0.59	1.02	0.94	1.48	0.38	$O(K(p^2+p))$
	MB	-0.01	0.67	0.28	0.64	1.07	1.03	1.29	1.39	$O(K(p^2+p))$
	ML	-0.06	1.55	-0.35	0.68	1.20	1.13	1.05	0.76	$O(TK(p^2+p))$
	DC	-0.21	1.61	-0.16	0.73	1.23	1.25	0.93	1.27	$O(K(p^2+p))$
	WW	-0.94	3.58	-1.86	1.22	1.71	1.98	0.40	0.33	$O(K(p^2+p))$
$\tau = 0.25$	Oracle	-0.51	1.02	0.16	0.80	1.19	1.20	1.00	10.30	-
	MV	-0.93	3.53	0.24	0.67	1.08	1.06	1.22	1.35	$O(TK(p^2+p))$
	SP	3.41	3.86	-0.74	0.76	1.10	1.23	0.98	0.55	$O(K(p^2+p))$
	MB	0.54	2.84	0.05	0.62	1.06	0.99	1.36	0.40	$O(K(p^2+p))$
	ML	4.45	2.29	-0.34	0.79	1.16	1.14	0.99	0.76	$O(TK(p^2+p))$
	DC	-0.82	2.29	0.37	0.88	1.21	1.27	0.90	1.24	$O(K(p^2+p))$
	WW	-1.07	1.81	-0.91	1.74	1.79	2.19	0.32	0.35	$O(K(p^2+p))$

附表 4 同质性分布式系统下平方模型 (d) 的补充模拟结果

Attached Table 4 Additional simulation result of quadratic model (d) under homogenous distributed system

分位数水平	算法	Bias0 (10^{-3})	Bias1 (10^{-3})	Bias2 (10^{-3})	SE0 (10^{-2})	SE1 (10^{-2})	SE2 (10^{-2})	估计效率	计算时长/秒	通讯成本
$\tau = 0.50$	Oracle	-6.45	-0.71	-3.23	3.19	5.73	5.79	1.00	10.10	-
	EC	-1.05	-0.86	0.33	3.95	7.16	6.17	0.72	0.45	$O(TKp)$
	SJ1	-1.78	4.26	-4.37	3.80	6.74	6.71	0.70	0.47	$O(TKp)$
	SJ2	-1.22	1.69	-1.37	3.31	6.07	5.90	0.91	0.40	$O(TKp)$
	MV	-1.93	1.46	-1.22	3.34	5.83	5.90	0.92	0.50	$O(TK(p^2+p))$
	SP	-1.49	0.16	-4.39	3.45	6.06	6.11	0.87	0.37	$O(K(p^2+p))$
	MB	-1.52	-0.72	-4.97	3.51	6.10	6.07	0.86	1.33	$O(K(p^2+p))$
	SP-Local	-14.70	-4.07	-5.35	3.86	7.50	7.36	0.60	0.37	$O(Kp)$
	MB-Local	-13.10	-6.77	-8.94	4.32	9.73	9.62	0.37	1.33	$O(Kp)$
	ML	-6.50	0.69	-3.07	3.18	5.73	5.79	0.99	0.70	$O(TK(p^2+p))$
	DC	-8.13	2.38	-4.17	3.34	5.98	5.87	0.93	1.23	$O(K(p^2+p))$
	WW	0.02	0.50	-3.79	3.25	5.98	5.89	0.95	0.32	$O(K(p^2+p))$
$\tau = 0.25$	Oracle	7.31	4.67	-0.32	2.02	4.93	5.26	1.00	9.85	-
	EC	4.51	6.72	-3.57	3.05	5.95	6.06	0.69	0.39	$O(TKp)$
	SJ1	7.61	8.84	-4.33	2.55	5.99	5.66	0.75	0.43	$O(TKp)$
	SJ2	8.17	7.57	-0.59	2.20	5.68	5.72	0.80	0.36	$O(TKp)$
	MV	15.20	5.18	-1.08	2.06	5.05	5.26	0.95	0.48	$O(TK(p^2+p))$
	SP	11.90	3.21	-0.66	2.13	5.43	5.59	0.85	0.33	$O(K(p^2+p))$
	MB	11.70	3.86	-0.01	2.19	5.45	5.56	0.85	1.28	$O(K(p^2+p))$
	SP-Local	12.20	4.17	0.24	2.34	5.95	6.44	0.68	0.33	$O(Kp)$
	MB-Local	12.90	3.99	1.27	2.55	6.94	7.34	0.51	1.28	$O(Kp)$
	ML	7.37	4.50	-0.46	2.01	4.96	5.24	0.99	0.66	$O(TK(p^2+p))$
	DC	11.50	6.31	1.52	2.20	5.31	5.41	0.88	1.17	$O(K(p^2+p))$
	WW	10.30	6.41	1.94	2.14	5.08	5.18	0.96	0.28	$O(K(p^2+p))$