中国科学: 信息科学 2020 年 第 50 卷 第 10 期: 1511–1528
SCIENTIA SINICA Informationis
论文

分布式隐私保护 – Logistic 回归

王璞玉, 张海*
西北大学数学学院统计系, 西安 710127
* 通信作者. E-mail: zhanghai@nwu.edu.cn

收稿日期: 2018–08–06; 接受日期: 2019–03–19; 网络出版日期: 2020–10–09
国家自然科学基金 (批准号: 11571011) 和 NSFC - 广东省大数据科学研究中心项目 (批准号: U1811461) 资助

摘要 本文关注敏感数据的隐私保护问题, 开展满足差分隐私的分布式 Logistic 回归模型研究. 通过对算法输出结果加扰动, 实现分布式差分隐私. 进一步, 为了防止计算机信息交互过程中可能产生的隐私泄露, 针对算法迭代过程加扰动的方式提出了基于 Alternating Direction Method of Multipliers (ADMM) 算法的分布式 Logistic 变量扰动算法, 并给出算法的理论界估计. 实验表明, 所提算法可有效地处理分布式存储数据并保护其隐私.

关键词 差分隐私, 分布式算法, Logistic 回归, ADMM 算法

1 引言

互联网的普及在方便人们生活的同时, 带来了个人敏感数据隐私泄露的风险. 大量的敏感数据被收集, 例如个人位置轨迹数据、家庭用电数据、网购数据以及医药购买记录等. 基于现代统计机器学习方法对收集到的数据进行分析, 可精准推断个人信息或预测个人行为. 因此, 此类数据如果被不恰当地使用, 将严重威胁到个人隐私安全. 为了避免隐私的泄露, 众多学者提出了不同的隐私保护技术, 例如 Sweeney [1,2] 提出的 k- 匿名方法, Machanavajjhala 等 [3] 提出的 l-diversity 原则以及 Li 等 [4] 提出的 T-closeness 等等. 由于该类隐私保护模型的可靠性与攻击者所掌握的背景知识相关, 当相关数据及背景知识的信息被利用时, 这些方法在面对某些攻击模型时存在失效的可能. Dwork [5] 在 2006 年提出了差分隐私的概念, 与传统的隐私保护方法不同的是, 差分隐私定义了一个严格的与背景知识无关的攻击模型, 并对隐私泄露的程度给出了定量的表示, 可实现在学习总体规律的同时保护个体隐私信息. 差分隐私的定义如下.

定义1 (差分隐私 [5]) 给定一个随机算法 A, Range$(\cdot)$ 为随机算法所有可能输出的集合, 对于任意两个最多相差一个数据的数据集 D 和 D', 以及所有 $S \in \text{Range}(A)$, 随机算法 A 满足 ϵ- 差分隐私, 若

$$\frac{P(A(D) \in S)}{P(A(D') \in S)} \leqslant \exp(\epsilon),$$

引用格式: 王璞玉, 张海. 分布式隐私保护 – Logistic 回归. 中国科学: 信息科学, 2020, 50: 1511–1528, doi: 10.1360/N112018-00214
Wang P Y, Zhang H. Distributed logistic regression with differential privacy (in Chinese). Sci Sin Inform, 2020, 50: 1511–1528, doi: 10.1360/N112018-00214

© 2020《中国科学》杂志社

其中, $P(A(D) \in S)$ 表示随机算法 A 的输出属于集合 S 的概率, 隐私预算参数 ϵ 表示保护隐私的程度, ϵ 越小隐私保护程度越高. 由定义可以看出, 差分隐私实际上要求单个数据对算法输出结果的影响不显著, 使得两个只相差一个数据的数据集输出结果差异非常小, 从而使得攻击者无法辨别个体信息.

差分隐私作为一种有数学理论支撑的隐私保护技术, 适合于大数据时代对个人隐私保护的需要, 具有广泛的应用 [6~8]. 差分隐私通过参数 ϵ 控制隐私的泄露额度, 即使遇到具有背景知识的攻击者, 也有良好的隐私保护能力. 另外, 如果数据发布时满足给定额度的 ϵ- 差分隐私, 其隐私泄露额度不会因为数据的再发布等行为而增加. 上述优点使得差分隐私具有良好的扩展性. 拉普拉斯 (Laplace) 噪声机制 [9] 和指数噪声机制 [10] 是两种常用的隐私保护实现机制. 目前, 差分隐私技术的研究主要集中于数据发布 [11~14] 和查询处理 [15] 等方面. 具有差分隐私的机器学习算法为数不多, 主要有 Chaudhuri 等 [16] 提出的输出扰动和目标函数扰动 ERM 差分隐私方法, Kifer 等 [17] 提出的高维情形下的 ERM 差分隐私方法, Wang 等 [18] 和 Park 等 [19] 提出的差分隐私聚类分析方法, 以及 Abadi 等 [20] 提出的目标扰动因果分析隐私保护方法等. 可以看出, 基于差分隐私的机器学习算法研究尚处于起步阶段, 主要原因在于差分隐私要求算法具有稳健性, 从而大部分算法不适用. 因此需要设计更稳健的算法使之适用于差分隐私框架分析, 目前已有部分此方面工作 [21, 22]. 本文主要关注机器学习算法方面的研究.

随着信息技术的飞速发展, 大数据的分布式存储成为主流的数据存储方式, 即将数据集按照某种规则分割并存储于不同的机器. 针对分布式数据存储的新特点, 大量适合处理该类型数据的分布式机器学习方法被提出. 最简单的分布式框架基于平均混合 (average mixture, AVGM) 的思想, 具体来说, 每台计算机用自身存储的数据做局部估计, 算法最终输出所有局部估计值的平均. 这种分布式方法简单, 且信息交互成本极低. McDonald 等 [23, 24] 和 Zhang 等 [25, 26] 研究了基于平均混合方法的分类及参数估计问题. 另一种分布式框架基于连通的计算机网络, 即计算机之间通过物理方式相连并在算法中通过信息交互得到全局估计. Boyd 等 [27] 基于该框架提出了包含全局变量的分布式方法, Mateos 等 [28] 提出了分布式 Lasso 方法, Wang 等 [29] 将该方法推广至分布式非凸正则化方法, Wang 等 [30] 提出了基于牛顿法的分布式优化方法. 对大数据的处理和分析给人们认识自身和世界带来便利, 给人们的生产生活提供各种服务的同时, 也带来了前所未有的隐私忧患. 例如, 通过对位置轨迹大数据 [31] 的分析, 能够得到用户的兴趣爱好、行为模式、社会地位等信息; 通过对用电大数据 [32] 的分析, 能够得到用户的个人习惯、周边环境等信息; 通过对电商大数据的分析, 能够得到用户的行为喜好、家庭住址、工作单位、联系方式等信息. 这些关乎用户自身或者环境的敏感信息如果被不恰当的使用, 将会给用户带来严重的人身财产安全威胁. 目前在实际应用中, 数据均采用分布式存储. 本文将分布式算法与隐私保护方法相结合, 开展针对于大数据的分布式差分隐私算法研究. 分布式差分隐私的现有工作包括 Han 等 [33] 提出的分布式投影梯度下降算法, Ji 等 [34] 提出的分布式 Newton-Raphson 算法, Zhang 等 [35] 和 Zhang 等 [36] 提出的分布式 ADMM 隐私保护算法等. 本文同样研究基于 ADMM 的差分隐私保护算法, 与文献 [35,36] 不同之处在于, 我们研究优化目标控制一致性的分布式模型并利用三步 ADMM 算法求解, 所用的三步 ADMM 更适合于一般的分布式学习算法 [37], 并通过在迭代过程中加入噪声实现分布式隐私保护.

2 分布式 Logistic 回归

作为一种典型的分类模型, Logistic 回归已被广泛地研究与应用. 本节给出分布式 Logistic 回归模型, 基于 ADMM 算法提出分布式 Logistic 算法, 并证明了算法的收敛性.

2.1 分布式 Logistic 模型

考虑 Logistic 模型

$$\min_{w} \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-y_i w^{\mathrm{T}}x_i))+\frac{\lambda}{2}\|w\|_2^2, \tag{1}$$

其中 $x_i\in\mathbb{R}^p$ 为输入变量, $y_i\in\{-1,1\}$ 为输出变量, $w\in\mathbb{R}^p$ 为模型的系数向量, $\lambda>0$ 为调控参数.

假设数据 D_{all} 被分别存储在 J 台独立的可相互通信的计算机中, 这些计算机自然地构成一个网络. 令 N_j 表示计算机 j 的邻居集, $|N_j|$ 表示其大小. 可以利用无向图 $G=(\mathcal{J},E)$ 表示上述网络, 其中节点集 $\mathcal{J}:=\{1,2,\ldots,J\}$ 代表 J 台计算机, E 为 $J\times J$ 的邻接矩阵, 其表示边的信息, 若计算机 i 与 j 可互相通信, 则 $E_{ij}=1$, 否则, $E_{ij}=0$. 进一步, 我们假设网络中任意两点之间都存在一条路径使之可互相通信, 称之为连通的.

计算机 j 的局部数据为 $D_j=\{x_{ji},y_{ji}\}_{i=1}^{n_j}$, n_j 为其所存储的数据量且 $n=\sum_{j=1}^{J}n_j$. 依据数据所属计算机的不同, 我们将模型 (1) 改写为

$$\min_{w} \frac{1}{n}\sum_{j=1}^{J}\sum_{i=1}^{n_j}\log(1+\exp(-y_{ji}w^{\mathrm{T}}x_{ji}))+\frac{\lambda}{2}\|w\|_2^2. \tag{2}$$

注意到式 (2) 只是将数据拆分为 J 个部分, 显然其与模型 (1) 等价.

为了使每台计算机能够不依赖全局数据独立地进行计算, 我们用局部变量 $\{w_j\}_{j=1}^{J}$ 代替全局变量 w, 其中 w_j 为计算机 j 的局部估计. 模型 (2) 转化为下述分布式 Logistic 模型:

$$\min_{\{w_j\}_{j=1}^{J}} \sum_{j=1}^{J}\left\{\frac{1}{n}\sum_{i=1}^{n_j}\log(1+\exp(-y_{ji}w_j^{\mathrm{T}}x_{ji}))+\frac{\lambda}{2J}\|w_j\|_2^2\right\} \quad \text{s.t. } w_j=w_{j'},\ j\in\mathcal{J},\ j'\in N_j, \tag{3}$$

其中 N_j 为计算机 j 的邻居集, 模型的约束要求互为邻居的两台计算机的局部估计值相等. 由于我们假设整个网络是连通的, 那么该约束使得任意两台计算机的局部变量相等, 即所有计算机拥有相等的局部估计值. 定理 1 给出带约束的分布式 Logistic 模型 (3) 与原始模型 (1) 的等价性.

定理1 若无向图 $G=(\mathcal{J},E)$ 是连通的, 那么模型 (1) 与 (3) 等价.

证明 如果 $G=(\mathcal{J},E)$ 是连通的, 那么任意两台计算机之间都可以互相通信, 由模型 (3) 的约束条件知, $\forall\, j,j'\in\mathcal{J}$, $\exists\, j_1,j_2,\ldots,j_m$, s.t. $\hat{w}_j=\hat{w}_{j_1}=\cdots=\hat{w}_{j_m}=\hat{w}_{j'}$. 那么, $\exists\,\hat{w}$, s.t. $\hat{w}=\hat{w}_1=\cdots=\hat{w}_J$, $\hat{w}$ 既是 (3) 的解, 又是 (2) 的解 (将等式代入可消去 (3) 的约束), 反之亦然, 则模型 (3) 与 (2) 等价, 其与模型 (1) 也等价.

注1 上述分布式方法与 Boyd 等 [27] 提出的分布式 Logistic 方法的不同之处在于, 文献 [27] 的分布式模型包含全局变量, 并且整个计算机网络存在一个可以与所有计算机通信的中心计算机, 全局变量在该中心计算机上进行更新. 由于中心计算机的存在, 基于该分布式模型的算法不具有鲁棒性, 即当中心计算机出现故障或者被攻击时, 该分布式方法失效. 而我们研究基于非中心化网络的分布式方法, 通过引入局部变量代替全部变量, 各计算机独立的更新局部变量. 基于该模型的分布式算法比上述中心化算法更鲁棒, 任何一个节点出现故障时不会大幅影响分布式算法的输出结果.

2.2 分布式 Logistic 算法

本小节给出分布式 Logistic 算法, 并证明其收敛性.

为了求解模型 (3), 我们借助于 ADMM 算法. ADMM 算法是一个交替迭代的过程, 每次迭代包含 3 个步骤, 具体细节见附录 A.1.

首先, 考虑增加局部辅助变量 $\{Z_j\}_{j=1}^J$, 模型 (3) 转化为

$$\min_{\{w_j\}_{j=1}^J,\{Z_j\}_{j=1}^J} \frac{1}{n}\sum_{j=1}^{J}\sum_{i=1}^{n_j}\log(1+\exp(-y_{ji}w_j^{\mathrm{T}}x_{ji}))+\frac{\lambda}{2J}\sum_{j=1}^{J}\|Z_j\|_2^2$$

$$\text{s.t. } Z_i - w_j = 0,\ \forall\ i,j = 1,\ldots,J. \tag{4}$$

上述模型在正则化项引入新的局部变量, 其约束要求对于任意 i 和 j, Z_i 与 w_j 相等, 由于我们假设无向图是连通的, 那么该约束蕴含着任意计算机 j 的局部变量 Z_j 和 w_j 相等. 因此, 模型 (4) 的约束与模型 (3) 的约束是一致的, 两个模型等价.

为了方便表述, 将模型 (4) 改写为向量的形式:

$$\min_{Z,w} f(w)+h(Z) \quad \text{s.t. } AZ+Bw=0, \tag{5}$$

其中损失函数 $f(w)=\frac{1}{n}\sum_{j=1}^{J}\sum_{i=1}^{n_j}\log(1+\exp(-y_{ji}w_j^{\mathrm{T}}x_{ji}))$, 正则化项 $h(Z)=\frac{\lambda}{2J}\sum_{j=1}^{J}\|Z_j\|_2^2$, w 和 Z 为所有局部变量 w_j 和 Z_j 组成的向量, 具体地, $w=[w_1,\ldots,w_J]^{\mathrm{T}}$, $Z=[Z_1,\ldots,Z_J]^{\mathrm{T}}$, 矩阵 A 和 B 为向量 Z 和 w 系数矩阵, 其包含着模型 (4) 中所有约束的信息, 具体地, $A=[A_1^{\mathrm{T}},\ldots,A_J^{\mathrm{T}}]^{\mathrm{T}}\in\mathbb{R}^{(J\times J\times p)\times(J\times p)}$, 其中 $A^*=[I_{p\times p},\ldots,I_{p\times p}]^{\mathrm{T}}\in\mathbb{R}^{(J\times p)\times(p)}$, $\mathbf{0}\in\mathbb{R}^{(J\times p)\times p}$ 为所有元素均为 0 的矩阵, $A_1=[A^*,\mathbf{0},\ldots,\mathbf{0}]$, $A_2=[\mathbf{0},A^*,\ldots,\mathbf{0}]$, $\ldots$, $A_J=[\mathbf{0},\ldots,\mathbf{0},A^*]$, $B=-[I_{(J\times p)\times(J\times p)},\ldots,I_{(J\times p)\times(J\times p)}]^{\mathrm{T}}\in\mathbb{R}^{(J\times J\times p)\times(J\times p)}$.

我们给出模型 (5) 的二次增广 Lagrange 函数:

$$L(Z,w,V)=f(w)+h(Z)+\langle V,AZ+Bw\rangle+\frac{c}{2}\|AZ+Bw\|_2^2, \tag{6}$$

其中 V 为 Lagrange 乘子, $c>0$ 为预选参数.

基于 ADMM 算法, 在每一次迭代中依次更新 Z, w 以及 Lagrange 乘子 V, 直至算法收敛. 下面给出 ADMM 算法的收敛性定理.

定理2 若 $c>\max\{\frac{2L_f}{M'},L_fM',\frac{\beta}{M}\}$, 则 ADMM 算法 (A1), (A2), (A3) 生成的序列收敛到 $L(Z,w,V)$ 的极小值点.

为了证明该定理, 我们首先证明当预选参数 c 满足定理 2 的条件时, 随着 ADMM 算法的迭代, $L(Z,w,V)$ 单调递减. 其次证明对于任意迭代次数 k, $L(Z,w,V)$ 有下界. 最后给出了 $L(Z,w,V)$ 梯度的性质. 具体证明见附录 A.2.

2.3 参数估计

为了求解 ADMM 算法的 (A1), (A2), (A3), 我们将向量形式的二次增广 Lagrange 函数转化为

$$\begin{aligned}L(\{Z_j\}_{j=1}^J,\{w_j\}_{j=1}^J,\{V_j\}_{j=1}^J)=&\frac{1}{n}\sum_{j=1}^{J}\sum_{i=1}^{n_j}\log\left(1+\exp\left(-y_{ji}w_j^{\mathrm{T}}x_{ji}\right)\right)+\frac{\lambda}{2J}\sum_{j=1}^{J}\|Z_j\|_2^2\\&+\frac{c}{2}\sum_{j=1}^{J}\sum_{i=1}^{J}E_{ij}\left(\left\|Z_i-w_j+\frac{V_{ij}}{c}\right\|_2^2-\left\|\frac{V_{ij}}{c}\right\|_2^2\right),\end{aligned} \tag{7}$$

其中 E_{ij} 表示计算机 i 与 j 的连接情况.

首先考虑对变量 Z 的更新. 由 Lagrange 函数 (7) 的可分解性, (A1) 可拆分为 J 个不同的子问题, 对于每台计算机 j,

$$Z_j(k+1)=\arg\min_{Z_j}\frac{\lambda}{2J}\|Z_j\|_2^2+\frac{c}{2}\sum_{i=1}^{J}E_{ij}\left\|Z_j-w_i(k)+\frac{V_{ji}(k)}{c}\right\|_2^2,$$

该优化问题的显式解为

$$Z_j(k+1)=\frac{c\sum_{i=1}^{J}E_{ij}(w_i(k)-\frac{V_{ji}(k)}{c})}{c\sum_{i=1}^{J}E_{ij}+\frac{\lambda}{J}}. \tag{8}$$

下面考虑对变量 w 的更新. 同样的, w 的更新 (A2) 可拆分为 J 个子问题:

$$\begin{aligned}w_j(k+1)=&\arg\min_{w_j}L_j\\=&\arg\min_{w_j}\frac{1}{n}\sum_{i=1}^{n_j}\log(1+\exp(-y_{ji}w_j^{\mathrm{T}}x_{ji}))+\frac{c}{2}\sum_{i=1}^{J}E_{ij}\left\|Z_i(k+1)-w_j+\frac{V_{ij}(k)}{c}\right\|_2^2.\end{aligned} \tag{9}$$

借助于梯度下降法求解上述优化问题, 记 $m=1,2,\ldots$ 为求解 (9) 的内循环标号, 对于 ADMM 的第 $k+1$ 次迭代 (外循环), 令 $w_j^0(k+1)=w_j(k)$, 那么

$$w_j^m(k+1)=w_j^{(m-1)}(k+1)-\alpha\frac{\partial L_j}{\partial w_j}, \tag{10}$$

其中 α 为梯度下降的步长, $\frac{\partial L_j}{\partial w_j}=\frac{1}{n}\sum_{i=1}^{n_j}\frac{-y_{ji}x_{ji}}{1+\exp(y_{ji}w^{\mathrm{T}}x_{ji})}-c\sum_{i=1}^{J}E_{ij}(Z_i(k+1)-w_j^{(m-1)}(k+1)+\frac{V_{ij}(k)}{c})$.

最后, 更新 Lagrange 乘子

$$V_{ji}(k+1)=V_{ji}(k)+c(Z_j(k+1)-w_i(k+1)),\quad i=1,\ldots,J. \tag{11}$$

用式 (8), (10) 和 (11) 分别替代 (A1), (A2), (A3), 我们提出分布式 Logistic 算法, 详见算法 1.

Algorithm 1 Distributed Logistic algorithm (DLA)

1: Input: Data $D_{\mathrm{all}}=\{D_j\}_{j=1}^{J}$, pre-selected parameter c, step size α, iteration number K;
2: For all $j\in\mathcal{J}$, let $w_j(0)=(0,\ldots,0)^{\mathrm{T}}$, $Z_j(0)=(0,\ldots,0)^{\mathrm{T}}$, $V_j(0)=(0,\ldots,0)^{\mathrm{T}}$;
3: Locally run;
4: **for** $k=1$ to K **do**
5: Update $Z_j(k+1)$ via (8), and transmit it to neighbors;
6: **for** $m=1$ to M **do**
7: Update $w_j^m(k+1)$ via (10);
8: **end for**
9: Let $w_j(k+1)=w_j^M(k+1)$, and transmit it to neighbors;
10: Update V_{ji} via (11), where $i=1,\ldots,J$, and transmit it to neighbors;
11: **end for**
12: Output: $w^*=w_j(K)$, $\forall\, j\in\mathcal{J}$.

注2 分布式 Logistic 算法运行如下. 初始化 $w_j(0)=(0,\ldots,0)^{\mathrm{T}}$, $Z_j(0)=(0,\ldots,0)^{\mathrm{T}}$, 以及 $V_j(0)=(0,\ldots,0)^{\mathrm{T}}$. 在第 $k+1$ 次迭代, 每台计算机 j 利用式 (8), (10) 和 (11) 局部更新 Z_j,w_j,V_{ji}, 同时将更新结果与邻居计算机进行交互. 重复该过程直至最大迭代次数 K, 此时 $w^*=w_1=w_2=\cdots=w_J$, 输出 w^*.

3 分布式 Logistic 隐私保护算法

本节首先基于算法输出加扰动的方式提出分布式 Logistic 输出扰动算法. 进一步, 为了保护计算机的局部隐私, 基于在迭代中给变量加扰动的方式提出分布式 Logistic 变量扰动算法.

3.1 分布式 Logistic 输出扰动算法

本小节提出一种简单的满足差分隐私的分布式算法 —— 分布式 Logistic 输出扰动算法. 具体地, 首先基于算法 1 给出模型 (1) 的估计 w^*, 随后给该估计加一个密度函数正比于 $\exp(-\frac{n\epsilon\lambda}{2})$ 的噪声 b, 其中, 从 Gamma 分布 $\Gamma(p, \frac{2}{n\epsilon\lambda})$ 中生成 b 的范数, 并随机赋予其方向, 调控参数 λ 由分布式 K 折交叉验证选择, 具体见附录 A.3, 最后输出 $w^* + b$. 详见算法 2.

Algorithm 2 Distributed Logistic output perturbation algorithm (DLP)

1: Input: Data $D_{\text{all}} = \{D_j\}_{j=1}^J$, pre-selected parameter c, step size α, iteration number K, privacy degree ϵ;
2: For all $j \in \mathcal{J}$, let $w_j(0) = (0, \ldots, 0)^{\mathrm{T}}$, $Z_j(0) = (0, \ldots, 0)^{\mathrm{T}}$, $V_j(0) = (0, \ldots, 0)^{\mathrm{T}}$;
3: Generate noise $b \sim \exp(-\frac{n\epsilon\lambda}{2})$;
4: Run distributed Logistic algorithm, and obtain w^*;
5: Output: $w^* + b$.

下面给出原始模型 (1) 的全局敏感度, 并证明算法 2 满足差分隐私.

引理1 ([16]) 给定 n 个样本 $x_1, \ldots, x_n$, 及其标号 $y_1, \ldots, y_n$. 对于所有 i, 有 $\|x_i\|_2 \leqslant 1, y_i \in \{-1, 1\}$, 那么 Logistic 回归问题的全局敏感度最大为 $\frac{2}{n\lambda}$, 其中 λ 为正则化参数.

定理3 假设对于 D_{all} 中的任意数据 x_i, $\|x_i\|_2 \leqslant 1$, 那么分布式 Logistic 输出扰动算法 (DLP) 满足 ϵ- 差分隐私.

证明 由定理 2 知, 分布式 Logistic 模型 (3) 与原模型 (1) 等价, 其解 w^* 为原问题的解. 那么, 分布式 Logistic 回归问题的全局敏感度和非分布式 Logistic 回归问题的全局敏感度相同. 又由引理 1 知, 当数据集中任意数据 $\|x_i\|_2 \leqslant 1$ 时, 非分布式 Logistic 算法的全局敏感度最大为 $\frac{2}{n\lambda}$, 其中 n 为样本量, λ 为罚参数. 因此, 分布式 Logistic 输出扰动算法的全局敏感度最大为 $\frac{2}{n\lambda}$. 又文献 [16] 的定理 1 证明了给原始问题的输出加噪声 $b \sim \exp(-\frac{n\epsilon\lambda}{2})$ 的算法满足 ϵ- 差分隐私, 故与原始问题解相同的分布式 Logistic 输出扰动算法满足 ϵ- 差分隐私.

注意到, 分布式 Logistic 输出扰动算法给整体算法的输出加噪声, 其只能保护算法最终输出的隐私. 但在分布式算法的每一次迭代中, 计算机与其邻居会进行信息交互, 这涉及到保护各计算机局部隐私的问题. 当攻击者可访问一台计算机的所有迭代过程的数据时, 很容易依据交互信息推测出整个网络中所有计算机的局部信息. 为了解决上述问题, 下面研究不但保护整体输出的隐私而且保护各计算机局部隐私的算法.

3.2 分布式 Logistic 变量扰动算法

本小节提出分布式 Logistic 变量扰动算法, 并证明其满足差分隐私.

为了防止分布式算法在实现过程中各计算机交互时可能产生的隐私泄露, 我们考虑在 ADMM 的每一次迭代过程中对交互信息加扰动, 即给各局部更新变量 $Z_j(k), w_j(k)$ 添加噪声 $b_j(k)$, 其中 $b_j(k)$ 的密度函数正比于 $\exp(-\frac{n\epsilon(k)}{2\alpha}\|b_j(k)\|_2)$, $\epsilon(k)$ 为第 k 次迭代的隐私预算. 我们从 Gamma 分布 $\Gamma(p, \frac{2\alpha}{n\epsilon(k)})$ 中生成噪声 $b_j(k)$ 的范数, 同时赋予其随机方向.

具体的, 在 ADMM 算法的第 $k+1$ 次局部迭代中, 计算机 j 首先利用式 (8) 更新变量 $Z_j(k+1)$, 并给其更新结果添加噪声 $b_j(k+1)$:

$$Z_j(k+1)=\frac{c\sum_{i=1}^{J}E_{ij}(w_i(k)-\frac{V_i(k)}{c})}{c\sum_{i=1}^{J}E_{ij}+\frac{\lambda}{J}}+b_j(k+1). \tag{12}$$

其次, 利用式 (10) 更新局部变量 $w_j(k+1)$, 由于执行式 (10) 直至收敛过于精确, 这里我们对该内循环放松, 只进行一次更新, 即

$$w_j(k+1)=w_j(k)-\alpha\frac{\partial L_j}{\partial w_j}. \tag{13}$$

随后, 给更新过程添加噪声, 那么计算机 j 对 $w_j(k+1)$ 的更新规则为

$$w_j(k+1)=w_j(k)-\alpha\frac{\partial L_j}{\partial w_j}+b_j(k+1), \tag{14}$$

其中 $\frac{\partial L_j}{\partial w_j}=\frac{1}{n}\sum_{i=1}^{n_j}\frac{-y_{ji}x_{ji}}{1+\exp(y_{ji}w_j^{\mathrm{T}}x_{ji})}-c\sum_{i=1}^{J}E_{ij}(Z_i(k+1)-w_j(k)+\frac{V_j(k)}{c})$.
最后, 我们利用更新 Lagrange 乘子

$$V_{ji}(k+1)=V_{ji}(k)+c(Z_j(k+1)-w_i(k+1)), \tag{15}$$

结合 (12), (14) 和 (15), 提出分布式 Logistic 变量扰动算法, 详见算法 3.

Algorithm 3 Distributed Logistic variable perturbation algorithm (DLVP)

1: Input: Data $D_{\mathrm{all}}=\{D_j\}_{j=1}^{J}$, pre-selected parameter c, step size α, iteration number K, privacy degree $\epsilon=\sum_{k=1}^{K}\epsilon(k)$;
2: For all $j\in\mathcal{J}$, let $w(0)=(0,\ldots,0)^{\mathrm{T}}$, $Z(0)=(0,\ldots,0)^{\mathrm{T}}$, $V_{ij}(0)=(0,\ldots,0)^{\mathrm{T}}$;
3: Locally run;
4: **for** $k=1$ to K **do**
5: Generate noise $b_j(k)\sim\exp(-\frac{n\epsilon(k)}{2\alpha}\|b_j(k)\|_2)$;
6: Update $Z_j(k+1)$ via (12), and transmit it to neighbors;
7: Update $w_j(k+1)$ via (14), and transmit it to neighbors;
8: Update $V_{ji}(k)$ via (15), where $i=1,\ldots,J$, and transmit it to neighbors;
9: **end for**
10: Output: $w^*=\frac{1}{J}\sum_{j=1}^{J}w_j(K)$.

分布式 Logistic 变量扰动算法可看作是给每一次局部变量的更新结果的输出加扰动, 相比其他方法更易实现. 在算法的第 k 次迭代, 计算机 j 首先生成噪声 $b_j(k)$, 其次分别利用式 (12) 更新 $Z_j(k)$, 式 (14) 更新 $w_j(k)$, 以及式 (15) 更新 $V_{ji}(k)$, 同时将扰动后的局部变量与邻居进行交互. 上述算法不仅保护最终输出的隐私, 而且在每次信息交互的过程中保护单台计算机的局部隐私, 这是一种比分布式输出扰动算法更严格的隐私保护算法.

注3 由于上述算法是一种保护计算机局部隐私的分布式算法, 其 "输出" 不仅仅是算法最终输出结果, 而且包括每一台计算机在每一次迭代时所交互的信息.

定理4 假设对于 D_{all} 中的任意数据 x_i, $\|x_i\|_2\leqslant 1$, 那么迭代 K 次的分布式 Logistic 变量扰动算法 (DLVP) 满足 ϵ- 差分隐私, 且 $\epsilon=\sum_{k=1}^{K}\epsilon(k)$.

证明 我们证明对于任意两个最多相差一个数据的数据集 $D_{\mathrm{all}}=\{D_j\}_{j=1}^{J}$ 和 $D'_{\mathrm{all}}=\{D'_j\}_{j=1}^{J}$, 分布式 Logistic 变量扰动算法:

$$\frac{P(\{S(k)\}_{k=0}^{K}\in S|D_{\mathrm{all}})}{P(\{S(k)\}_{k=0}^{K}\in S|D'_{\mathrm{all}})}\leqslant\exp(\epsilon). \tag{16}$$

考虑

$$\begin{aligned}
&\frac{P(\{Z(k),w(k)\}_{k=0}^{K}\in S\mid D_{\text{all}})}{P(\{Z(k),w(k)\}_{k=0}^{K}\in S\mid D'_{\text{all}})}\\
&=\frac{P(\{Z(0),w(0)\}\in S(0)\mid D_{\text{all}})}{P(\{Z(0),w(0)\}\in S(0)\mid D'_{\text{all}})}\cdot\prod_{k=1}^{K}\frac{P(\{Z(k),w(k)\}\in S(k)\mid\{Z(r),w(r)\}_{r=0}^{k-1},D_{\text{all}})}{P(\{Z(k),w(k)\}\in S(k)\mid\{Z(r),w(r)\}_{r=0}^{k-1},D'_{\text{all}})}\\
&\overset{\text{(a)}}{=}\prod_{k=1}^{K}\prod_{j=1}^{J}\frac{P(\{Z_j(k),w_j(k)\}\in S_j(k)\mid\{Z_j(r),w_j(r)\}_{r=0}^{k-1},D_j)}{P(\{Z_j(k),w_j(k)\}\in S_j(k)\mid\{Z_j(r),w_j(r)\}_{r=0}^{k-1},D'_j)}\\
&\overset{\text{(b)}}{=}\prod_{k=1}^{K}\frac{P(\{Z_j(k),w_j(k)\}\in S_j(k)\mid\{Z_j(r),w_j(r)\}_{r=0}^{k-1},D_j)}{P(\{Z_j(k),w_j(k)\}\in S_j(k)\mid\{Z_j(r),w_j(r)\}_{r=0}^{k-1},D'_j)}\\
&=\prod_{k=1}^{K}\frac{P(Z_j(k)=Z_j^*(k)\mid\{Z_j(r),w_j(r)\}_{r=0}^{k-1},D_j)}{P(Z_j(k)=Z_j^*(k)\mid\{Z_j(r),w_j(r)\}_{r=0}^{k-1},D'_j)}\cdot\frac{P(w_j(k)=w_j^*(k)\mid\{Z_j(r),w_j(r)\}_{r=0}^{k-1},Z_j^*(k),D_j)}{P(w_j(k)=w_j^*(k)\mid\{Z_j(r),w_j(r)\}_{r=0}^{k-1},Z_j^*(k),D'_j)},
\end{aligned}\tag{17}$$

其中 S 为算法 K 次迭代过程中所有可能输出的集合, (a) 中用到 $w(0)=(0,\dots,0)^{\mathrm{T}}$, $Z(0)=(0,\dots,0)^{\mathrm{T}}$, 并且计算机 j 独立的生成噪声 $b_j(k)$; (b) 中假设 D_{all} 和 D'_{all} 相差的唯一一个数据存储于计算机 j 中.

我们证明算法的第 k 次迭代满足 $\epsilon(k)$- 差分隐私. 首先考虑

$$\frac{P(Z_j(k)=Z_j^*(k)\mid\{Z_j(r),w_j(r)\}_{r=0}^{k-1},D_j)}{P(Z_j(k)=Z_j^*(k)\mid\{Z_j(r),w_j(r)\}_{r=0}^{k-1},D'_j)},$$

由于这部分与数据独立, 则

$$\frac{P(Z_j(k)=Z_j^*(k)\mid\{w(r),Z(r)\}_{r=0}^{k-1},D_j)}{P(Z_j(k)=Z_j^*(k)\mid\{w(r),Z(r)\}_{r=0}^{k-1},D'_j)}=1.\tag{18}$$

其次考虑

$$\frac{P(w_j(k)=w_j^*(k)\mid\{w_j(r),Z_j(r)\}_{r=0}^{k-1},Z_j^*(k),D_j)}{P(w_j(k)=w_j^*(k)\mid\{w_j(r),Z_j(r)\}_{r=0}^{k-1},Z_j^*(k),D'_j)},\tag{19}$$

由 $w_j(k)$ 的更新过程 (15), 有

$$b_j(k)=w_j(k)-w_j(k-1)+\alpha\frac{\partial L_{1j}}{\partial w_j},\tag{20}$$

其中 $b_j(k)$ 是数据集为 D_j 时计算机 j 生成的噪声. 显然, 给定 $\{w_j(r),Z_j(r)\}_{r=1}^{k-1}$ 以及 $Z_j(k)$ 时, 随机变量 $B_j(k)$ 和 $W_j(k)$ 互为双射, $b_j(k)$ 和 $w_j(k)$ 是 $B_j(k)$ 和 $W_j(k)$ 的一个实现. 令 $g_k(\cdot,D_j):\mathbb{R}^p\to\mathbb{R}^p$ 为关于数据集 D_j 的由 $B_j(k)$ 到 $W_j(k)$ 的一一映射, 那么

$$\begin{aligned}
&\frac{P(w_j(k)=w_j^*(k)\mid\{w_j(r),Z_j(r)\}_{r=0}^{k-1},Z_j^*(k),D_j)}{P(w_j(k)=w_j^*(k)\mid\{w_j(r),Z_j(r)\}_{r=0}^{k-1},Z_j^*(k),D'_j)}\\
&=\frac{P(b_j(k)=g_k^{-1}(w_j^*(k),D_j))}{P(b_j(k)=g_k^{-1}(w_j^*(k),D'_j))}\cdot\frac{|\det(J(g_k^{-1}(w_j^*(k),D_j)))|}{|\det(J(g_k^{-1}(w_j^*(k),D'_j)))|},
\end{aligned}\tag{21}$$

其中 $g_k^{-1}(w_j^*(k),D_j)$ 为从 $w_j^*(k)$ 到 $b_j(k)$ 的映射, $J(g_k^{-1}(w_j^*(k),D_j)$ 为其 Jacobi 矩阵.

考虑 w_j 的原始更新过程 (13) 的敏感度

$$\begin{aligned}\triangle &= \max_{|D_j - D_j'|\leqslant 1} \|w_{j,D_j}(k) - w_{j,D_j'}(k)\|_2 = \max_{|D_j - D_j'|\leqslant 1} \left\|\alpha\left(\frac{\partial L_{1j}}{\partial w_j} - \frac{\partial L_{1j}'}{\partial w_j}\right)\right\|_2 \\ &\overset{\text{(c)}}{=} \left\|\frac{\alpha}{n}\left(-\frac{-y_{j1}x_{j1}}{1+\exp(y_{j1}w_j^{\mathrm{T}}(k)x_{j1})} + \frac{-y_{j1}'x_{j1}'}{1+\exp(y_{j1}'w_j^{\mathrm{T}}(k)x_{j1}')}\right)\right\|_2 \\ &\overset{\text{(d)}}{\leqslant} \frac{2\alpha}{n},\end{aligned} \tag{22}$$

(c) 中我们假设两个数据集不同的点为它们的第 1 个数据点, 即 (x_{j1}, y_{j1}) 和 (x_{j1}', y_{j1}'), (d) 中用到 $\|x_{ji}\|_2 \leqslant 1$ 以及 $y_{ji} \in \{-1, 1\}$. 那么,

$$\begin{aligned}\frac{P(b_j(k) = g_k^{-1}(w_j^*(k), D_j))}{P(b_j(k) = g_k^{-1}(w_j^*(k), D_j'))} &= \frac{P(b_j(k) = w_j^*(k) - w_j(k-1) + \alpha\frac{\partial L_{1j}}{\partial w_j})}{P(b_j(k) = w_j^*(k) - w_j(k-1) + \alpha\frac{\partial L_{1j}'}{\partial w_j})} \\ &\overset{\text{(e)}}{\leqslant} \exp\left(\frac{n\epsilon(k)}{2\alpha}\left\|\alpha\left(\frac{\partial L_{1j}}{\partial w_j} - \frac{\partial L_{1j}'}{\partial w_j}\right)\right\|_2\right) \\ &\overset{\text{(f)}}{\leqslant} \exp(\epsilon(k)),\end{aligned} \tag{23}$$

其中, 由于噪声 $b_j(k) \sim \exp(-\frac{n\epsilon(k)}{2\alpha}\|b_j(k)\|_2)$, (e) 成立, 由式 (22) 有不等式 (f) 成立.

考虑式 (21) 的第 2 部分, 因为 $g_k(b_j(k), \cdot)$ 是一个线性函数, 那么

$$|\det(J(g_k^{-1}(w_j^*(k), D_j)))| = |\det(J(g_k^{-1}(w_j^*(k), D_j')))|. \tag{24}$$

结合 (23) 以及 (24),

$$\frac{P(w_j(k) = w_j^*(k) \mid \{w_j(r), Z_j(r)\}_{r=0}^{k-1}, Z_j^*(k), D_j)}{P(w_j(k) = w_j^*(k) \mid \{w_j(r), Z_j(r)\}_{r=0}^{k-1}, Z_j^*(k), D_j')} \leqslant \exp(\epsilon(k)). \tag{25}$$

最后, 由 (17), (18) 以及 (25) 有

$$\frac{P(\{w(k), Z(k)\}_{k=0}^K \in S \mid D_{\text{all}})}{P(\{w(k), Z(k)\}_{k=0}^K \in S \mid D_{\text{all}}')} \leqslant \exp\left(\sum_{k=1}^K \epsilon(k)\right) = \exp(\epsilon).$$

定理得证.

注4 在 DLVP 算法的更新中, 我们固定步骤 7 中梯度下降的步长 α, 其大小不随迭代的变化而变化. 因此, 式 (13) 的敏感度 $\frac{2\alpha}{n}$ 为固定常数, 其大小不随迭代次数改变.

4 实验

本节用单机 (1.2 GHz, 8 Gb RAM, MATLAB 2014b) 模拟一个分布式计算机系统. 首先通过模拟实验证明分布式 Logistic 算法的有效性和高效性. 其次, 通过类似的模拟实验以及基因数据实验证明分布式隐私保护算法的有效性.

$$E=\begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

图 1　计算机对应的无向图的邻接矩阵, 其表示计算机与其邻居的信息交互情况

Figure 1　The adjacency matrix of the undirected graph corresponding to computers, which represents the information interaction between the computers

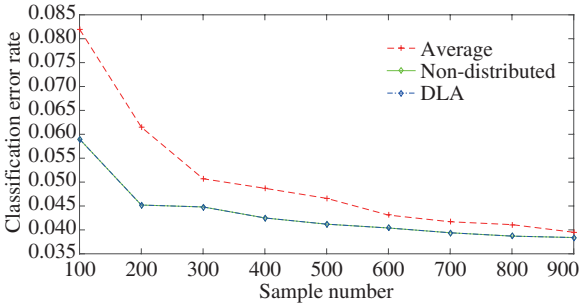

图 2　(网络版彩图) 随着样本量 n 的增加, 非分布式 Logistic 算法, 分布式 Logistic 算法和平均混合算法的错分率变化情况

Figure 2　(Color online) Classification error rate of non-distributed Logistic algorithm, DLA and Average as sample number n increase

4.1　分布式 Logistic 算法的有效性与高效性

我们研究分类模型:

$$p(y_i=1|x_i)=\frac{1}{1+\exp(-x_i^{\mathrm{T}}w)},\quad i=1,\ldots,n,$$

其中 $x_i\in\mathbb{R}^p$ 为输入变量, $w\in\mathbb{R}^p$ 为系数向量, n 为样本数. $x_i\sim N(0,\Sigma)$, 协方差矩阵为 $\Sigma=0.95^{|i-j|}, 1\leqslant i,j\leqslant n$. 若 $p(y_i=1|x_i)\geqslant 0.5$, 令 $y_i=1$, 否则, $y_i=-1$.

4.1.1　分布式 Logistic 算法的有效性

为了研究基于 ADMM 算法提出的分布式 Logistic 算法 (DLA) 的有效性, 我们对比分布式 Logistic 算法、非分布式 Logistic 算法 (令 $J=1$) 和平均混合算法 (Average) 的分类结果. 分别令样本量 $n=100,200,\ldots,1000$, $x_i\in\mathbb{R}^8$, 系数向量 $w=[3,7,-2,1,-5,-1,3,-5]$, 研究不同样本量下 3 种算法的分类结果. 在运行分布式 Logistic 算法和平均混合算法时, 分别将数据不重复且不均衡地分为 $J=5$ 个部分, 使得前 4 部分数据标号为 1 和 -1 的数据量比为 $8:2$ 或 $2:8$. 对于分布式 Logistic 算法, 用分布式 5 折交叉验证选择罚参数 λ (详见附录 A.3), 计算机所组成的无向图的邻接矩阵如图 1. 对于非分布式 Logistic 算法和平均混合算法, 用 5 折交叉验证选择 λ. 对于每一组实验, 分别生成 100 组数据, 运行 3 种算法并统计它们的平均错分率 (classification error rate). 具体结果见图 2.

由图 2 可以看到, 我们提出的分布式 Logistic 算法与非分布式 Logistic 算法的分类结果相同, 两种算法的平均错分率都随着样本量的增加而减小. 平均混合算法的平均错分率也随着样本量的增加而减小, 但其错分率一直大于分布式 Logistic 算法. 这表明在数据划分不均衡的情况下, 我们的算法相比于平均混合算法有更好的分类效果, 其主要原因在于, 我们的算法在优化过程中计算机之间进行了信息交互, 从而消除了样本划分不均衡给估计带来的影响.

4.1.2　分布式 Logistic 算法的高效性

为研究分布式算法在处理大量数据时的高效性, 我们分别生成 $n=10^4,p=10^3$; $n=10^4,p=10^4$; $n=10^4,p=10^5$ 以及 $n=10^5,p=10^3$ 的 4 组数据. 对于每一组数据, 我们分别用 matlab 中的 glmfit

表 1 分布式 Logistic 算法 (DLA) 的高效性

Table 1 Efficiency of distributed Logistic algorithm (DLA)

Data size	$n=10^4,\ p=10^3$	$n=10^4,\ p=10^4$	$n=10^4,\ p=10^5$	$n=10^5,\ p=10^3$
DLA ($J=5$)	175.21 s	1613.15 s	47720.4 s	1533.87 s
DLA ($J=10$)	84.70 s	832.54 s	23988.1 s	782.69 s
DLA ($J=100$)	8.63 s	78.43 s	2754.2 s	72.32 s
Glmfit	114.02 s	6327.79 s	More than 24 hours	1268.64 s

函数以及分布式 Logistic 算法 (固定罚参数 λ) 对其进行处理. 在调用分布式 Logistic 算法时, 选择预选参数 $c=1$, 梯度下降步长 $\alpha=0.4$, 分别将数据不重复的划分为 $J=5$, $J=10$ 以及 $J=100$ 个部分. 当 $J=5$ 时, 邻接矩阵如图 1 所示; 当 $J=10$ 和 $J=100$ 时, 邻接矩阵为 $1^{J\times J}$, 即所有计算机均可互相通信 (此时整个系统的通信成本达到最大). 具体结果见表 1.

由表 1 可以看到, 当 $n=10^4,\ p=10^3$; $n=10^4,\ p=10^4$ 以及 $n=10^5,\ p=10^3$ 时, glmfit 和分布式 Logistic 算法都能有效地对其进行分析; 当 $n=10^4,\ p=10^5$ 时, glmfit 需要超过 24 小时的计算时间, 而分布式算法依旧能够在可接受的时间范围内对其进行处理. 值得注意的是, 随着计算机数量的增加, 每台计算机的计算时间减少, 但它往往会产生不可避免的通信成本.

注5 在 4.1.2 小节的实验中, 当 $n=10^4,\ p=10^5$ 时, 我们调用单机 glmfit, 此时, 程序运行时间过长无法输出结果. 其原因在于计算量超出了单台计算机的运算能力. 调用分布式算法可有效地克服此问题.

4.2 分布式 Logistic 隐私保护算法

本小节通过模拟实验和基因数据实验研究分布式 Logistic 输出扰动算法和分布式 Logistic 变量扰动算法的可行性.

4.2.1 模拟实验

(1) 分布式 Logistic 输出扰动算法的有效性

本例中模型与上例相同, 特别地, 令样本量 $n=10^4$, $x_i\in\mathbb{R}^8$, 系数向量 $w=[3,7,-2,1,-5,-1,3,-5]$. 我们对比没有保护隐私的分布式 Logistic 算法 (DLA), 平均混合算法 (Average), 分布式 Logistic 输出扰动算法 (DLP) 以及平均混合输出扰动算法 (AVEP) 在不同隐私保护预算下的分类情况. 同样的, 对于分布式算法, 将数据不重复且不均衡地分为 $J=5$ 个部分, 预选参数 $c=0.01$, 梯度下降步长 $\alpha=0.8$, 用分布式 5 折交叉验证选择罚参数 λ, 邻接矩阵如图 1. 在运行分布式 Logistic 输出扰动算法和平均混合输出扰动算法时, 隐私预算 ϵ 分别取 $0.75,1,1.25,\ldots,2.75,3$. 具体分类结果见图 3 和 4.

从图 3 和 4 中可以看出, 分布式 Logistic 输出扰动算法和平均混合输出扰动算法的平均错分率和均方误差均大于没有保护隐私的分布式 Logistic 算法和平均混合算法的结果, 并且随着隐私参数 ϵ 的增加, 隐私保护算法和非隐私保护算法的错分率差别逐渐变小. 同时, 对于相同大小的隐私保护预算, 分布式 Logistic 输出扰动算法的分类结果均优于平均混合输出扰动算法, 这说明当数据划分不均衡时, 我们的算法基于交互信息的方式可消除其不利的影响.

(2) 分布式 Logistic 变量扰动算法的有效性

本例中模型与上例相同, 特别地, 令样本量 $n=10^4$, 系数向量 $w=[3,1.5,2,1,5,1,0.5,2]$. 由于分布式 Logistic 变量扰动算法保护了计算机的局部隐私, 其结果与只保护全局隐私的分布式 Logistic 输

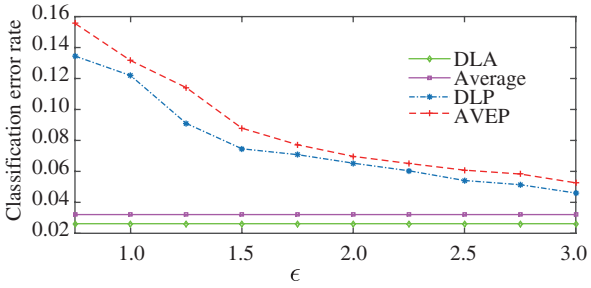

图 3 (网络版彩图) 随着隐私程度 ϵ 大小的变化, 分布式 Logistic 输出扰动算法 (DLP) 和平均混合输出扰动算法 (AVEP) 错分率的变化情况

Figure 3 (Color online) Classification error rate of DLP and AVEP as ϵ change

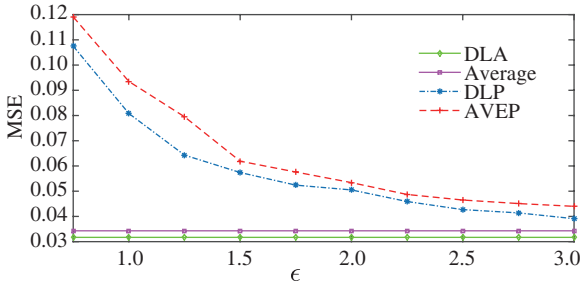

图 4 (网络版彩图) 随着隐私程度 ϵ 大小的变化, 分布式 Logistic 输出扰动算法 (DLP) 和平均混合输出扰动算法 (AVEP) 均方误差的变化情况

Figure 4 (Color online) MSE of DLP and AVEP as ϵ change

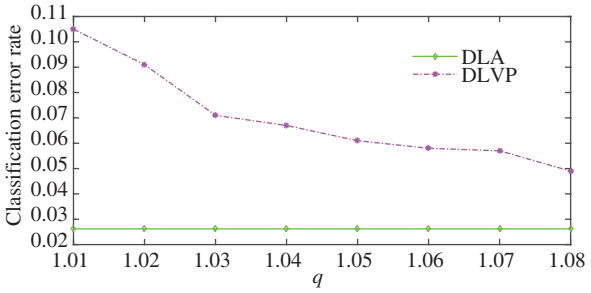

图 5 (网络版彩图) 随着隐私程度 ϵ 大小的变化, 分布式 Logistic 变量扰动算法 (DLVP) 错分率的变化情况

Figure 5 (Color online) Classification error rate of DLVP as ϵ change

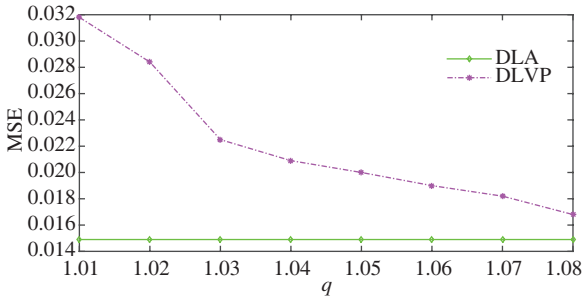

图 6 (网络版彩图) 随着隐私程度 ϵ, 分布式 Logistic 变量扰动算法 (DLVP) 均方误差的变化情况

Figure 6 (Color online) Mean square error of DLVP as ϵ change

出扰动算法的结果没有可比性, 因此我们对比没有保护隐私的分布式 Logistic 算法和分布式 Logistic 变量扰动算法 (DLVP) 的分类结果. 在运行分布式 Logistic 变量扰动算法时, 将数据不重复地分为 $J = 5$ 个部分, 预选参数 $c = 0.1$, 梯度下降步长 $\alpha = 1$, 用分布式 5 折交叉验证选择罚参数 λ, 邻接矩阵如图 1. 我们令第 k 次迭代的隐私保护预算 $\epsilon(k) = \frac{2\alpha(5q^k)}{n}$, q 分别取 $1.01, 1.02, \ldots, 1.08$. 即随着迭代次数的增加, 每一次迭代的隐私预算逐渐增多. 由于噪声的随机性, 对每一个隐私预算重复实验 100 次, 并记录 100 次实验的平均结果. 具体结果见图 5 和 6.

从图 5 和 6 中可以看出, 对于不同的隐私预算, 分布式 Logistic 变量扰动算法错分率和均方误差均大于没有隐私保护的分布式算法, 且随着 ϵ 的增加, 分类误差和均方误差逐渐减小. 这说明保护隐私和精准预测是不能同时满足的, 保护隐私会导致预测的不准确, 泄露的隐私越少 (ϵ 越小), 预测误差越大. 在实际中, 需要根据不同的问题设定不同的隐私额度.

注6 分类结果统计错分样本的个数, 均方误差 $\mathrm{MSE} = \sum_{i=1}^{n_{\mathrm{test}}} (y_i - \frac{1}{1+\exp(-x_i^{\mathrm{T}} w^*)})^2$.

4.2.2 Colon 数据

Colon 数据源自文献 [38], 收集了 62 个组织的样本, 其中包括 24 个正常组织样本以及 38 个肿瘤组织样本, 每个样品含有 2000 个基因表达值. 我们将数据随机分为两部分: 包含 40 个样本的训练集

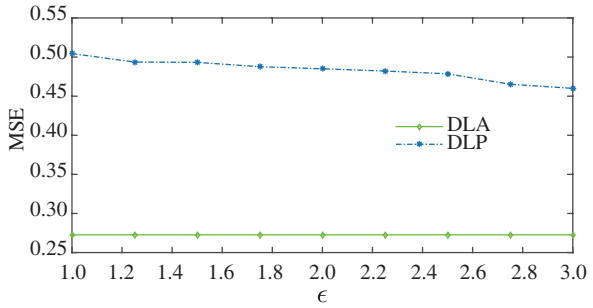

图 7 (网络版彩图) 随着隐私程度 ϵ 大小的变化, 分布式 Logistic 输出扰动算法 (DLP) 的均方误差

Figure 7 (Color online) Mean square error of DLP as ϵ change

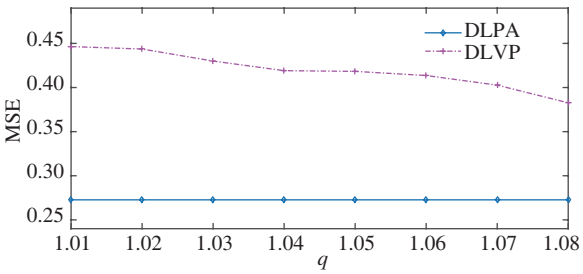

图 8 (网络版彩图) 随着隐私程度 ϵ 大小的变化, 分布式 Logistic 变量扰动算法 (DLVP) 的均方误差

Figure 8 (Color online) Mean square error of DLVP as ϵ change

和包含 22 个样本的测试集. 在数据预处理后, 研究不同的隐私保护预算下分布式 Logistic 输出扰动算法和分布式 Logistic 变量扰动算法对上述数据的处理结果, 并用分布式 Logistic 算法的分类结果与其作对比.

在运用分布式算法时, 我们将训练集分为 $J = 5$ 个部分, 预选参数 $c = 0.01$, 邻接矩阵如图 1, 梯度下降步长 $\alpha = 0.4$, 并用分布式 5 折交叉验证选择 λ. 对于分布式 Logistic 输出扰动算法和平均混合输出扰动算法, 我们令 ϵ 分别取 $1, 1.25, \ldots, 2.75$. 对于每一个 ϵ 的值, 由于噪声 b 的随机性, 我们重复实验 50 次, 并记录其平均错分率. 对于分布式 Logistic 变量扰动算法, 令算法在第 k 次迭代的隐私保护参数 $\epsilon(k) = \frac{2\alpha(5q^k)}{n}$, q 分别取 $1.01, 1.02, \ldots, 1.08$. 同样, 对于每一个 q 的取值重复实验 50 次, 并记录其平均错分率. 实验结果如图 7 和 8.

实验表明, 对于高维基因数据, 分布式 Logistic 输出扰动算法和分布式 Logistic 变量扰动算法仍然可以对其进行有效的分析并进行隐私保护.

4.3 实验结果分析

4.1.1 和 4.1.2 小节的实验表明本文所提分布式 Logistic 方法可以有效并且高效地处理非分布式方法难以处理的高维海量数据; 4.2.1 小节的实验表明分布式 Logistic 输出扰动算法和分布式 Logistic 变量扰动算法可以在分析数据的同时对其进行隐私保护, 随着隐私保护要求的增加 (ϵ 的减小), 预测的精确程度降低. 4.2.2 小节的实验表明分布式 Logistic 输出扰动算法和分布式 Logistic 变量扰动算法可以有效地处理高维基因数据.

5 结论

本文开展满足差分隐私的分布式 Logistic 回归研究. 首先我们提出适合处理分布式存储数据的分布式 Logistic 方法, 基于 ADMM 算法提出了分布式 Logistic 算法, 并证明了算法的收敛性; 其次, 我们提出了一种简单的分布式差分隐私算法 —— 分布式 Logistic 输出扰动算法 (DLP), 并给出其隐私界估计; 最后, 我们提出了可以保护各计算机局部隐私的分布式 Logistic 变量扰动算法 (DLVP). 实验表明, 所提分布式隐私保护算法能够有效地处理分布式存储数据并且对其进行隐私保护. 当数据中的部分数据通过授权等方式转化为公开数据时, 分布式隐私保护算法的效率将有明显的提高. 基于该思想可推广本文分布式方法, 进一步提升算法的效用.

参考文献

1 Sweeney L. k-anonymity: a model for protecting privacy. Int J Unc Fuzz Knowl Based Syst, 2002, 10: 557–570

2 Sweeney L. Achieving k-anonymity privacy protection using generalization and suppression. Int J Uncertain Fuzz Knowl Based Syst, 2008, 10: 571–588

3 Machanavajjhala A, Gehrke J, Kifer D. l-diversity: privacy beyond k-anonymity. In: Proceedings of IEEE 22nd International Conference on Data Engineering, Atlanta, 2006. 24

4 Li N, Li T, Venkatasubramanian S. t-closeness: privacy beyond k-anonymity and l-diversity. In: Proceedings of IEEE 23rd International Conference on Data Engineering, Istanbul, 2007. 106–115

5 Dwork C. Differential privacy. In: Proceedings of International Colloquium on Automata, Languages and Programming, Venice, 2006. 1–12

6 Dwork C, Roth A. The algorithmic foundations of differential privacy. FNT Theor Comput Sci, 2014, 9: 211–407

7 Zhou S G, Li F, Tao Y F, et al. Privacy preservation in database applications: a survey. Chin J Comput, 2009, 32: 847–861 [周水庚, 李丰, 陶宇飞, 等. 面向数据库应用的隐私保护研究综述. 计算机学报, 2009, 32: 847–861]

8 Xiong P, Zhu T Q, Wang X F. A survey on differential privacy and applications. Chin J Comput, 2014, 37: 101–122 [熊平, 朱天清, 王晓峰. 差分隐私保护及其应用. 计算机学报, 2014, 37: 101–122]

9 Dwork C, Mcsherry F, Nissim K. Calibrating noise to sensitivity in private data analysis. In: Proceedings of the 3rd Conference on Theory of Cryptography, New York, 2006. 265–284

10 Mcsherry F, Talwar K. Mechanism design via differential privacy. In: Proceedings of the 48th Annual Symposium on Foundations of Computer Science, Providence, 2007. 94–103

11 Xiao X, Wang G, Gehrke J. Differential privacy via wavelet transforms. IEEE Trans Knowl Data Eng, 2011, 23: 1200–1214

12 Fan L Y, Xiong L. An adaptive approach to real-time aggregate monitoring with differential privacy. IEEE Trans Knowl Data Eng, 2014, 26: 2094–2106

13 Wu Y J, Ge C, Zhang L Q, et al. An algorithm for differential privacy streaming data publication based on matrix mechanism under exponential decay mode. Sci Sin Inform, 2017, 47: 1493–1509 [吴英杰, 葛晨, 张立群, 等. 指数衰减模式下基于矩阵机制的差分隐私流数据发布算法. 中国科学: 信息科学, 2017, 47: 1493–1509]

14 Zhang X J, Meng X F. Differential privacy in data publication and analysis. Chin J Comput, 2014, 37: 927–949 [张啸剑, 孟小峰. 面向数据发布和分析的差分隐私保护. 计算机学报, 2014, 37: 927–949]

15 Cormode G, Procopiuc C, Srivastava D, et al. Differentially private spatial decompositions. In: Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, 2012. 20–31

16 Chaudhuri K, Monteleoni C, Sarwate A. Differentially private empirical risk minimization. J Mach Learn Res, 2011, 12: 1069–1109

17 Kifer D, Smith A, Thakurate A. Private convex empirical risk minimization and high-dimensional regression. J Mach Learn Res, 2012, 23: 1–41

18 Wang Y N, Wang Y X, Singh A. Differentially private subspace clustering. In: Proceedings of the 28th International Conference on Neural Information Processing Systems, Montreal, 2015. 1000–1008

19 Park M, Foulds J, Chaudhuri K, et al. DP-EM: differentially private expectation maximization. In: Proceedings of the 20th International Conference on ArtificialIntelligence and Statistics, Fort Lauderdale, 2017. 54

20 Abadi M, Chu A, Goodfello I, et al. Deep learning with differential privacy. In: Proceedings of the 2016 ACMSIGSAC Conference on Computer and Communication Security, Vienna, 2016. 303–318

21 Dwork C, Lei J. Differential privacy and robust statistics. In: Proceedings of ACM Symposium on Theory of Computing, Bethesd, 2009

22 Wang Y, Lei J, Fienberg S. Learning with differential privacy: stability, learnability and the sufficiency and necessity of ERM principle. J Mach Learn Res, 2016, 17: 6353–6392

23 Mann G, McDonald R, Mohri M, et al. Efficient large-scale distributed training of conditional maximum entropy models. In: Proceedings of the 23rd Annual Conference on Neural Information Processing Systems, Vancouver, 2009. 1231–1239

24 McDonald R, Hall K, Mann G. Distributed training strategies for the structured perceptron. In: Proceedings of the 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Los Angeles, 2010. 456–464

25 Zhang Y, Duchi J C, Wainwright M J. Communication-efficient algorithms for statistical optimization. J Mach Learn Res, 2013, 14: 3321–3363
26 Zhang Y, Duchi J, Wainwright M. Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates. J Mach Learn Res, 2013, 30: 592–617
27 Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn, 2011, 3: 1–122
28 Mateos G, Bazerque J A, Giannakis G B. Distributed sparse linear regression. IEEE Trans Signal Process, 2010, 58: 5262–5276
29 Wang P, Zhang H, Liang Y. Model selection with distributed SCAD penalty. J Appl Stat, 2018, 45: 1938–1955
30 Wang J, Kolar M, Srebro N, et al. Efficient distributed learning with sparsity. In: Proceedings of the International Conference on Machine Learning, Sydney, 2017
31 Huo Z, Meng X F. A survey of trajectory privacy-preserving techniques. Chin J Comput, 2011, 34: 1820–1830 [霍峥, 孟小峰. 轨迹隐私保护技术研究. 计算机学报, 2011, 34: 1820–1830]
32 Drenker S, Kader A. Nonintrusive monitoring of electric loads. IEEE Comput Appl Power, 1999, 12: 47–51
33 Han S, Topcu U, Pappas G J. Differentially private distributed constrained optimization. IEEE Trans Automat Contr, 2017, 62: 50–64
34 Ji Z, Jiang X, Wang S, et al. Differentially private distributed logistic regression using private and public data. BMC Med Genomics, 2014, 7: S14
35 Zhang T, Zhu Q. Dynamic differential privacy for ADMM-based distributed classification learning. IEEE Trans Inform Forensic Secur, 2017, 12: 172–187
36 Zhang X, Khalili M, Liu M. Improving the privacy and accuracy of ADMM-Based distributed algorithms. In: Proceedings of the 35th International Conference on Machine Learning, Stockholm, 2018. 5796–5805
37 Wang Y, Yin W, Zeng J. Global convergence of ADMM in nonconvex nonsmooth optimization. J Sci Comput, 2019, 78: 29–63
38 Alon U, Barkai N, Notterman D A, et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc Natl Acad Sci USA, 1999, 96: 6745–6750

附录 A 附录

A.1 ADMM 算法

考虑下述优化问题:

$$\min_{Z,w} f(w)+h(Z) \quad \text{s.t. } AZ+Bw=0.$$

首先, 我们通过引入参数将有约问题转化为无约束优化:

$$L(Z,w,V)=f(w)+h(Z)+\langle V,AZ+Bw\rangle+\frac{c}{2}\|AZ+Bw\|_2^2,$$

其中 V 为 Lagrange 乘子, $c>0$ 为预选罚参数. 下面我们借助于 ADMM 算法求解上述问题.

ADMM 算法的每一次迭代包括下面 3 个步骤.

步骤 1: 更新 Z,

$$Z(k+1)=\arg\min_{Z} L(Z,w(k),V(k)). \tag{A1}$$

步骤 2: 更新 w,

$$w(k+1)=\arg\min_{w} L(Z(k+1),w,V(k)). \tag{A2}$$

步骤 3: 更新 Lagrange 乘子 V,

$$V(k+1)=V(k)+c(AZ(k+1)+Bw(k+1)). \tag{A3}$$

具体的, 在第 $k+1$ 次迭代过程中, 我们首先关于 Z 最小化 $L(Z,w(k),V(k))$, 给出 $Z(k+1)$ 的更新解, 其次关于 w 最小化 $L(Z(k+1),w,V(k))$, 给出 $w(k+1)$ 最后通过 $V(k+1)=V(k)+c(AZ(k+1)+Bw(k+1))$ 更新 $V(k+1)$. 重复上述过程, 直至算法收敛.

A.2 定理 2 的证明

在分析 ADMM 算法的收敛性之前, 我们给出一些记号. 令 M 表示 $A^{\mathrm{T}}A$ 的最小严格正特征值, M' 表示 $B^{\mathrm{T}}B$ 的最小严格正特征值. 显然 $f(w)$ 是李普希茨 (Lipschitz) 可微的. 对于任意 $j \in \mathcal{J}$, 令

$$h(Z_j) = \frac{\lambda}{2J}\|Z_j\|_2^2,$$

那么存在 $\beta \geqslant 0$, 对于任意 Z_j, Z_j',

$$h(Z_j) + \frac{\beta}{2}\|Z_j - Z_j'\|_2^2 \geqslant h(Z_j') + \langle \nabla h(Z_j'), Z_j - Z_j' \rangle \tag{A4}$$

成立. 为了证明定理 2, 我们给出下面 3 个引理.

引理 A1 对于所有 k, 下述命题成立:

(1) $\nabla f(w(k+1)) = -B^{\mathrm{T}}V(k+1)$;

(2) $\|Z(k) - Z(k+1)\|_2^2 \leqslant \frac{1}{M}\|AZ(k) - AZ(k+1)\|_2^2$;

(3) $\|w(k) - w(k+1)\|_2^2 \leqslant \frac{1}{M'}\|Bw(k) - Bw(k+1)\|_2^2$;

(4) $\|V(k) - V(k+1)\|_2^2 \leqslant \frac{L_f^2}{M'}\|w(k) - w(k+1)\|_2^2$.

证明 由 $w(k+1)$ 的最优性条件:

$$0 = \nabla f(w(k+1)) + B^{\mathrm{T}}V(k) + cB^{\mathrm{T}}(AZ(k+1) + Bw(k+1)),$$

以及

$$V(k+1) = V(k) + c(AZ(k+1) + Bw(k+1)),$$

可知 (1) 成立. 注意到 M 和 M' 分别为 $A^{\mathrm{T}}A$ 和 $B^{\mathrm{T}}B$ 的最小严格正特征值, 那么 (2) 和 (3) 显然成立. 由于

$$\begin{aligned}\|V(k) - V(k+1)\|_2^2 &\leqslant \frac{1}{M'}\|B^{\mathrm{T}}(V(k) - V(k+1))\|_2^2\\ &\overset{\text{(a)}}{=} \frac{1}{M'}\|\nabla f(w(k)) - \nabla f(w(k+1))\|_2^2 \overset{\text{(b)}}{\leqslant} \frac{L_f^2}{M'}\|w(k) - w(k+1)\|_2^2,\end{aligned}$$

其中 (a) 我们利用了等式 (1), (b) 利用了条件 f 是李普希茨可微的. 那么 (4) 成立.

引理 A2 若 $c > \max\{\frac{2L_f}{M'}, L_f M', \frac{\beta}{M}\}$, 那么当 $k \to \infty$ 时 $L(Z^{k+1}, w^{k+1}, V^{k+1})$ 收敛.

证明 首先, 我们证明当预选参数 $c\max\{\frac{2L_f}{M'}, L_f M', \frac{\beta}{M}\}$ 时, 随着 ADMM 算法的迭代, $L(Z, w, V)$ 是单调递减的. 考虑

$$\begin{aligned}&L(Z(k), w(k), V(k)) - L(Z(k+1), w(k+1), V(k+1))\\ &= (L(Z(k), w(k), V(k)) - L(Z(k+1), w(k), V(k))) + (L(Z(k+1), w(k), V(k)) - L(Z(k+1), w(k+1), V(k+1))).\end{aligned}$$

讨论上式第 1 部分,

$$\begin{aligned}&L(Z(k), w(k), V(k)) - L(Z(k+1), w(k), V(k))\\ &= h(Z(k)) - h(Z(k+1)) + \langle V(k), AZ(k) - AZ(k+1)\rangle + \frac{c}{2}\|AZ(k) + Bw(k)\|_2^2 - \frac{c}{2}\|AZ(k+1) + Bw(k)\|_2^2\\ &\overset{\text{(c)}}{=} h(Z(k)) - h(Z(k+1)) + \langle A^{\mathrm{T}}V(k), Z(k) - Z(k+1)\rangle\\ &\quad + \langle cA^{\mathrm{T}}(AZ(k+1) + Bw(k)), Z(k) - Z(k+1)\rangle + \frac{c}{2}\|AZ(k) - AZ(k+1)\|_2^2\\ &\overset{\text{(d)}}{=} h(Z(k)) - h(Z(k+1)) - \langle \partial h(Z(k+1)), Z(k) - Z(k+1)\rangle + \frac{c}{2}\|AZ(k) - AZ(k+1)\|_2^2\\ &\overset{\text{(e)}}{\geqslant} -\frac{\beta}{2}\|Z(k) - Z(k+1)\|_2^2 + \frac{cM}{2}\|Z(k) - Z(k+1)\|_2^2\\ &= \frac{cM - \beta}{2}\|Z(k) - Z(k+1)\|_2^2,\end{aligned}$$

其中, (c) 中我们利用等式: $\|b + c\|_2^2 - \|a + c\|_2^2 = \|b - a\|_2^2 + 2\langle a + c, b - a\rangle$, 并令 $a = AZ(k+1)$, $b = AZ(k)$, $c = Bw(k)$; (d) 中利用 $\nabla h(Z(k+1)) = -(A^{\mathrm{T}}V(k) + cA^{\mathrm{T}}(AZ(k+1) + Bw(k)))$; (e) 中利用引理 A1 的 (2) 以及不等式 (A4).

讨论第 2 部分,

$$
\begin{aligned}
&L(Z(k+1),w(k),V(k))-L(Z(k+1),w(k+1),V(k+1))\\
&=f(w(k))-f(w(k+1))+\langle V(k),AZ(k+1)+Bw(k)\rangle-\langle V(k+1),AZ(k+1)+Bw(k+1)\rangle\\
&\quad+\frac{c}{2}\|AZ(k+1)+Bw(k)\|_2^2-\frac{c}{2}\|AZ(k+1)+Bw(k+1)\|_2^2\\
&=f(w(k))-f(w(k+1))+\langle B^{\mathrm{T}}V(k+1),w(k)-w(k+1)\rangle-\langle c(AZ(k+1)+Bw(k+1)),AZ(k+1)+Bw(k)\rangle\\
&\quad+\frac{c}{2}\langle AZ(k+1)+Bw(k),AZ(k+1)+Bw(k)\rangle-\frac{c}{2}\|AZ(k+1)+Bw(k+1)\|_2^2\\
&\overset{\text{(f)}}{\geqslant}-\frac{L_f}{2}\|w(k)-w(k+1)\|_2^2+\frac{c}{2}\|Bw(k)-Bw(k+1)\|_2^2-\frac{1}{c}\|V(k)-V(k+1)\|_2^2\\
&\overset{\text{(g)}}{\geqslant}\left(\frac{cM'-L_f}{2}-\frac{L_f^2}{cM'}\right)\|w(k)-w(k+1)\|_2^2,
\end{aligned}
$$

其中, (f) 中利用引理 A1 的 (1) 以及 f 是李普希茨可微的; (g) 中利用引理 A1 的 (3) 和 (4). 结合上述两个不等式, ADMM 算法的迭代满足

$$
\begin{aligned}
&L(Z(k),w(k),V(k))-L(Z(k+1),w(k+1),V(k+1))\\
&\geqslant\left(\frac{cM-\beta}{2}\right)\|Z(k)-Z(k+1)\|_2^2+\left(\frac{cM'-L_f}{2}-\frac{L_f^2}{cM'}\right)\|w(k)-w(k+1)\|_2^2,
\end{aligned}
$$

若 $c>\max\{\frac{2L_f}{M'},\frac{\beta}{M}\}$, 那么随着 ADMM 的迭代 $L(Z,w,V)$ 单调递减.

下面我们证明对任意 k, 二次增广 Lagrange 函数有下界.

注意到对任意 $Z(k+1)$, 存在 w' 使得 $AZ(k+1)+Bw'=0$, 则二次增广 Lagrange 函数可表示为

$$
\begin{aligned}
&L(Z(k+1),w(k+1),V(k+1))\\
&=h(Z(k+1))+f(w(k+1))+\langle V(k+1),AZ(k+1)+Bw(k+1)\rangle+\frac{c}{2}\|AZ(k+1)+Bw(k+1)\|_2^2\\
&=h(Z(k+1))+f(w(k+1))+\langle V^{k+1},AZ(k+1)+Bw'+Bw(k+1)-Bw'\rangle+\frac{c}{2}\|AZ(k+1)+Bw(k+1)\|_2^2\\
&=h(Z(k+1))+f(w(k+1))+\langle\nabla f(w(k+1)),w'-w(k+1)\rangle+\frac{c}{2}\|AZ(k+1)+Bw(k+1)\|_2^2\\
&\geqslant h(Z(k+1))+f(w')-\frac{L_f}{2}\|w'-w(k+1)\|_2^2+\frac{c}{2}\|AZ(k+1)+Bw(k+1)\|_2^2\\
&\geqslant h(Z(k+1))+f(w')-\frac{L_fM'}{2}\|Bw'-Bw(k+1)-AZ(k+1)-Bw'\|_2^2+\frac{c}{2}\|AZ(k+1)+Bw(k+1)\|_2^2\\
&\overset{\text{(h)}}{\geqslant}h(Z(k+1))+f(w')+\frac{c-L_fM'}{2}\|AZ(k+1)+Bw(k+1)\|_2^2\\
&\geqslant h(Z(k+1))+f(w')\geqslant 0,
\end{aligned}
$$

其中, 注意到假设 $c>L_fM'$, (h) 成立. 那么 $L(Z(k+1),w(k+1),V(k+1))$ 有下界.

综上所述, 当 $k\to\infty$ 时, $L(Z(k+1),w(k+1),V(k+1))$ 收敛.

最后, 我们给出一个梯度的性质.

引理 A3 存在 $d^{k+1}\in\nabla L(Z(k+1),w(k+1),V(k+1))$, 使得当 $k\to\infty$ 时, $\|d(k+1)\|_2^2\to 0$.

证明 注意到 $\nabla L(Z(k+1),w(k+1),V(k+1))=(\nabla_Z L,\nabla_w L,\nabla_V L)$. 因为 $\nabla_V L=AZ(k+1)+Bw(k+1)=\frac{1}{c}(V(k+1)-V(k))$, 有 $\|\nabla_V L\|_2^2\leqslant\frac{L_f^2}{c^2M'}\|w(k)-w(k+1)\|_2^2$.

并且

$$
\begin{aligned}
\nabla_w L&=\nabla f(w(k+1))+B^{\mathrm{T}}V(k+1)+cB^{\mathrm{T}}(AZ(k+1)+Bw(k+1))\\
&=B^{\mathrm{T}}(V(k+1)-V(k)).
\end{aligned}
$$

那么 $\|\nabla_w L\|_2^2=\|B^{\mathrm{T}}(V(k+1)-V(k))\|_2^2$. 为了给出 $\|\nabla_Z L\|_2^2$ 的界, 注意到

$$
\begin{aligned}
\nabla_Z L&=\nabla_Z h(Z(k+1))+A^{\mathrm{T}}V(k+1)+cA^{\mathrm{T}}(AZ(k+1)+Bw(k+1))\\
&\overset{\text{(I)}}{=}A^{\mathrm{T}}(V(k+1)-V(k))+cA^{\mathrm{T}}(Bw(k+1)-Bw(k)),
\end{aligned}
$$

其中, (I) 利用 $Z(k+1)$ 的最优化条件. 那么 $\|\nabla_Z L\|_2^2\leqslant\|A^{\mathrm{T}}(V(k+1)-V(k))\|_2^2+\|cA^{\mathrm{T}}(Bw(k+1)-Bw(k))\|_2^2$.

由于当 $k\to\infty$ 时, $L(Z(k),w(k),V(k))-L(Z(k+1),w(k+1),V(k+1))\to 0$, 结合引理 A2, 有 $\|Z(k)-Z(k+1)\|_2^2\to 0$, $\|w(k)-w(k+1)\|_2^2\to 0$, 和 $\|V(k)-V(k+1)\|_2^2\to 0$. 那么当 $k\to\infty$ 时, $\|d(k+1)\|_2^2\to 0$. 引理得证.

上述 3 个引理直接可推出定理 2 成立.

A.3 分布式 K 折交叉验证

我们用适合分布式的 K 折交叉验证来选择调控参数 λ. 假设 $\lambda \in \Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_L\}$, 且 $\lambda_1 > \lambda_2 > \cdots > \lambda_L$. 首先将全部数据随机分为 K 部分, 记为 $\{\tilde{X}_i, \tilde{Y}_i\}_{i=1}^K$, 那么对于第 j 台计算机来说, 它的数据可根据第 k 部分数据分为两部分: $\{X_j^k, Y_j^k\} = \{X_j, Y_j\} \cap \{\tilde{X}_k, \tilde{Y}_k\}$ 以及 $\{X_j^{(-k)}, Y_j^{(-k)}\} = \{X_j, Y_j\} \setminus \{\tilde{X}_k, \tilde{Y}_k\}$. 即既属于第 j 台计算机又属于第 k 部分的数据和属于第 j 台计算机但不属于第 k 部分的数据. 用数据集 $\{X_j^{(-k)}, Y_j^{(-k)}\}$ 做训练集, 对于不同的 λ_i 运行分布式算法, 得到 $\{w_j^{(-k)}(\lambda_i)\}_{i=1}^L$. 用 $\{X_j^k, Y_j^k\}$ 做测试集计算均方误差 $e_j^k(\lambda_i)$. 对于每台计算机以及每一部分数据重复上述步骤, 并计算每一个 λ_i 的总预测误差 $e(\lambda_i) = \sum_{j=1}^J \sum_{k=1}^K e_j^k(\lambda_i)$. 最后, 选择预测误差最小的 $\lambda_{\mathrm{dcv}} = \arg\min_{\lambda_i}\{e(\lambda_i)\}_{i=1}^L\}$ 作为调控参数.

Distributed logistic regression with differential privacy

Puyu WANG & Hai ZHANG*

Department of Statistics, School of Mathematics, Northwest University, Xi'an 710127*, China*
* Corresponding author. E-mail: zhanghai@nwu.edu.cn

Abstract In this paper, we focus on the privacy protection of sensitive data and develop a distributed logistic regression that satisfies differential privacy. Distributed differential privacy is achieved by perturbing the distributed algorithm output. Further, to prevent privacy leakage occurring during the computer interaction process, we propose a distributed logistic variable perturbation algorithm based on an alternating direction method of multipliers (ADMM) algorithm. Further, the theoretical bounds of the algorithms are provided. Experiments show that the proposed algorithms can effectively analyze distributed storage data and protect their privacy.

Keywords differential privacy, distributed algorithm, logistic regression, ADMM algorithm

Puyu WANG was born in 1992. At present, she is pursuing her Ph.D. degree in Statistics from Northwest University, Xi'an, China. Her current research interests include differential privacy and distributed computing.

Hai ZHANG was born in 1975. He received his Ph.D. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, in 2012. At present, he is a professor in the Department of Financial Mathematics and Statistics at Northwest University. His research interests include statistical machine learning, high-dimensional statistics, and differential privacy.