Privacy Preserving Distributed Maximum Likelihood Estimation

Xiaodong Lin

Distributed Maximum Likelihood Estimation

- A random sample $\mathbf{x^n} = \{x_1, \cdots, x_n\}$, while x_i follows $f(x; \theta)$
- The sample distributed across different agencies
  1. Horizontally partitioned
  2. Vertically partitioned
- Traditionally, the MLE is defined as

$$\hat{\theta} = arg\max_{\theta} l(\theta|\mathbf{x^n}) \tag{1}$$

- Goal: compute $\hat{\theta}$ without sharing data between agencies

Horizontally partitioned, exponential family

- Density $f(x) = b(x)exp\{a(\theta)^T t(x) - c(\theta)\}$
- Log likelihood

$$l(\theta; \mathbf{x^n}) = \sum_{i=1}^{n} \log b(x_i) + \sum_{i=1}^{n} \{a(\theta)^T t(x_i) - c(\theta)\}$$

- The MLE is

$$\hat{\theta} = \arg\max_{\theta} \quad a(\theta)^T \sum_{i=1}^{n} t(x_i) - nc(\theta)$$

- Secure summation of $\sum_{i=1}^{n} t(x_i)$

Horizontally partitioned, Newton Raphson

- Assume the estimates $\theta^{(s-1)}$ from previous step, new estimate is

$$\theta^{(s)} = \theta^{(s-1)} - (D^2 l(\mathbf{x^n}; \theta^{(s-1)}))^{-1} \nabla l(\mathbf{x^n}; \theta^{(s-1)}),$$

  $D^2 l()$ is the Hassian and $\nabla l()$ is the gradient.

- Assume $\theta = \{\theta_1, \cdots, \theta_k\}$,

$$\begin{aligned} \nabla_\theta l(\mathbf{x^n}; \theta^{(s-1)}) &= \left(\frac{\partial l}{\partial \theta_1}, \cdots, \frac{\partial l}{\partial \theta_k} \right) \\ &= \left(\sum_{i=1}^{n} \frac{\frac{\partial f(x_i;\theta)}{\partial \theta_1}}{f(x_i;\theta)}, \cdots, \sum_{i=1}^{n} \frac{\frac{\partial f(x_i;\theta)}{\partial \theta_k}}{f(x_i;\theta)} \right)_{\theta^{(s-1)}}. \end{aligned}$$

Horizontally partitioned, Newton Raphson

- Locally, we can compute L_j, $1 \leq j \leq m$, where

$$L_j = \left(\sum_{i=1}^{m_j} \frac{\frac{\partial f(x_i;\theta)}{\partial \theta_1}}{f(x_i;\theta)}, \cdots, \sum_{i=1}^{m_j} \frac{\frac{\partial f(x_i;\theta)}{\partial \theta_k}}{f(x_i;\theta)} \right)_{\theta^{(s-1)}}.$$

- Similarly we can compute

$$H_j(h,l) = \sum_{i=1}^{m_j} \left(\frac{\frac{\partial^2 f(x_i;\theta)}{\partial \theta_h \partial \theta_l}}{f(x_i;\theta)} - \frac{\frac{\partial f(x_i;\theta)}{\partial \theta_h} \frac{\partial f(x_i;\theta)}{\partial \theta_l}}{f^2(x_i;\theta)} \right)_{\theta^{(s-1)}}.$$

- The iteration step becomes

$$\theta^{(s)} = \theta^{(s-1)} - (\sum_{j=1}^{m} H_j)^{-1} (\sum_{j=1}^{m} L_j),$$

Horizontally partitioned, Newton Raphson

- H_j and L_j can be computed at each agency locally
- If $m > 2$, use secure summation
- Share $\sum_{j=1}^{m} H_j$ and $\sum_{j=1}^{m} L_j$
- Possible problems

  1. $m > 2$
  2. Share more than necessary

- Direct computation of $(\sum_{j=1}^{m} H_j)^{-1}(\sum_{j=1}^{m} L_j)$.

Horizontally partitioned, direct computation

- Without loss of generaility, assume $m = 2$
- Note that when $m = 2$, secure summation can't be applied
- Our goal: Compute $(H_1 + H_2)^{-1}(L_1 + L_2)$ securely
- Our approach: Solving linear equation system
- Denote $X = (H_1 + H_2)^{-1}(L_1 + L_2)$, the problem is equivalent to solve

$$(H_1 + H_2)X = (L_1 + L_2)$$

Direct computation protocol

- Assume two agencies A and B
- A generates $k \times k$ matrix M_1, B generates $k \times k$ matrix M_2, both with rank $k/2$.
- A sends M_1 to B. B computes $M_1 H_2$ and $M_1 L_2$, sent them to A
- A can produce the linear equation system

$$M_1(H_1 + H_2)X = M_1(L_1 + L_2)$$

- Symmetrically, B can produce

$$M_2(H_1 + H_2)X = M_2(L_1 + L_2)$$

Protocol

- Sharing the two linear equation systems directly will reveal H_1 and H_2.
- Solution: A and B generate full rank matrices T_1 and T_2 respectively
- Combine the following two linear equation systems to solve for X

$$T_1 M_1 (H_1 + H_2) X = T_1 M_1 (L_1 + L_2).$$

$$T_2 M_2 (H_1 + H_2) X = T_2 M_2 (L_1 + L_2).$$

Security analysis and discussion

- Agency A sent to B: $M_2H_1, M_2L_1, T_1M_1(H_1 + H_2)$ and $T_1M_1(L_1 + L_2)$
- A can check the rank of M_2. When $K > 2$, H_1 and L_1 are not revealed
- Sharing of $T_1M_1(H_1 + H_2)$ reveals T_1H_1 to B, but not H_1
- Protocol is symmertric
- Protocol works for $m = 2$
- Unsolved case: $k = 1$. Need to compute $(l_1 + l_2)/(h_2 + h_2)$ securely

Vertically partitioned case, independent variable

- Assume $\mathbf{x^n} = \{x_1, \cdots, x_n\}$, where $x_i = (x_i^1, \cdots, x_i^p)$. Each agency owns portion of the variables for all x_i
- Independent case one: assume $f(x_i, \theta) = \Pi_{s=1}^{t} f_s(x_i^s; \theta_s)$
- Log likelihood

$$l = \sum_{s=1}^{t} \left[\sum_{i=1}^{n} \log f_s(x_i^s; \theta_s) \right]$$

- Right hand side can be optimized locally to obtain θ_s

Vertically partitioned case, independent variable

- Indenpendent case two: $f(x_i, \theta) = \Pi_{s=1}^{t} f_s(x_i^s; \theta)$
- Log likelihood

$$l = \sum_{s=1}^{t} \left[\sum_{i=1}^{n} \log f_s(x_i^s; \theta) \right]$$

- Taking first derivative with respect to θ

$$\frac{\partial l}{\partial \theta} = \sum_{s=1}^{t} \left\{ \sum_{i=1}^{n} \left[\frac{1}{f_s(x_i^s; \theta)} \frac{\partial f_s(x_i^s; \theta)}{\partial \theta} \right] \right\}$$

- Compute locally at each agency and use secure summation or the direct computation protocol

Unsolved problems

- Boundary evaluation to identify global maximum
- Opt. out strategy
- General solution to vertical partition case
- Secure computation $(l_1 + l_2)/(h_2 + h_2)$
- Constrained MLE

$$\hat{\theta} = \arg\max l(\theta; \mathbf{x^n}) \;\; s.t. \;\; C_j(\theta)\; 1 \leq j \leq m,$$

where $C_j(\theta)$ are the parameter constraints each agency follows and can not be shared