arXiv:2302.03670v1 [cs.IT] 7 Feb 2023

异构数据库中的私有读取更新写入（PRUW）

Sajani Vithana　Sennur Ulukus　电气与计算机工程系　马里兰大学帕克分校，马里兰州 20742
spallego@umd.edu　ulukus@umd.edu

摘要——我们研究联邦子模型学习（FSL）中，采用异构存储受限数据库的私有读取更新写入（PRUW）问题。在 FSL 中，机器学习（ML）模型会根据用于训练模型的数据类型划分为多个子模型。用户根据其本地数据的类型，下载、更新并将更新内容上传回所关注的单个子模型。采用 PRUW 时，读取（下载）和写入（上传）过程会以确保更新子模型索引及更新值的信息论隐私的方式进行。我们考虑一种实际情形：子模型存储在存储约束任意（异构）的数据库中，并提出一种 PRUW方案，通过利用子模型分区和编码的存储机制来最大限度地降低通信成本。

I. 引言

在私有读取更新写入（PRUW） [1]–[8], 中，用户从数据存储中读取选定分段并向其写入数据，同时不泄露读取或写入的内容。我们将 PRUW 用于联邦子模型学习（FSL） [1], [2], [9]–[12] ，以保障学习过程中的用户隐私。FSL 是一种旨在降低联邦学习（FL）通信成本的方法：它根据用于训练的数据类型，将中央 FL 模型划分为多个子模型。在 FSL 中，给定用户只下载与其本地数据相关的子模型，对其进行更新，然后将更新写回（上传至）该子模型。正在更新的子模型索引和上传的更新值会泄露有关用户所持数据类型的信息。借助 PRUW，FSL 中的读写过程能够在保障正在更新的子模型索引和更新值的信息论隐私的同时进行。在 PRUW 中，子模型存储在多个互不串通的数据库中，这与私有信息检索（PIR） [13]–[44]类似。

本文考虑这样一种实际场景：多个互不串通的数据库具有任意存储约束。文献 [3] 研究了具有同构存储约束的 PRUW。文献 [45]–[51]，研究了具有同构和异构存储约束的数据库中的私有信息检索，并提出了最优分区策略，以将消息适配到存储受限的数据库中，从而最大限度地降低下载成本。在本文中，我们将私有信息检索中针对存储受限数据库的思路拓展到 PRUW。私有信息检索领域用于处理存储受限数据库的两种主要方法是：1）编码存储，即使用特定的 (K, R) 最大距离可分码对模型参数进行编码；2）分割存储，即对模型进行分区，

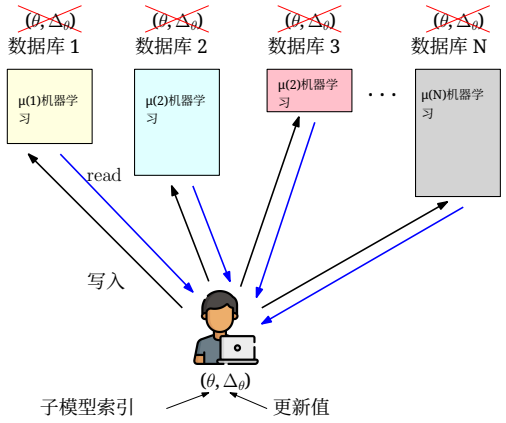

图 1. 系统模型。

并将不同分区复制到不同的数据库子集中。我们首先发现，在异构数据库的 PRUW 中，使用较大 K 值的 (K, R) MDS 编码存储具有优势；随后提出一种 PRUW方案以及一种用于联邦子模型学习的存储机制，该方案结合了子模型分区和 MDS 编码，以有效利用所有数据库中的可用空间，目标是最大限度地降低总通信成本。

II. 问题表述

我们考虑一个包含 M 个独立子模型的联邦子模型学习（FSL）设置，每个子模型包含 L 个参数，其取值来自有限域 $\mathbb{F}_q$，并存储在由 N 个不串通数据库组成的系统中。数据库 n 的存储容量为 μ（n）ML 个符号，其中 μ（n）≤ 1 对于 $n \in \{1, \ldots, N\}$；见图 1。也就是说，

$$H(S_n^{[t]}) \leq \mu(n)ML, \quad n \in \{1, \ldots, N\}, \quad t \in \mathbb{N}, \quad (1)$$

其中，$S_n^{[t]}$ 是时间 t 数据库 n 中存储的内容。在给定时间 t，用户下载所需的子模型，同时不向任何数据库透露其索引。这是 PRUW 过程的读取阶段。随后，用户在本地更新该子模型，并将更新上传回所有数据库中的同一子模型，同时不透露更新值或子模型索引。这是写入阶段。

子模型索引隐私：在时间 t、$\theta^{[t]}$ 更新的子模型索引信息不得泄露给任何数据库，即对于每个 n、$n \in \{1, \ldots, N\}$，

$$I(\theta^{[t]}; Q_n^{[t]}, U_n^{[t]} | Q_n^{[1:t-1]}, S_n^{[0:t-1]}, U_n^{[1:t-1]}) = 0, \quad (2)$$

其中， Q_n 和 U_n 是用户在方括号所标示的时间实例向数据库 n 发送的查询和更新。

更新值隐私：不允许任何数据库获取有关更新值的信息，

$$I(\Delta_\theta^{[t]}; U_n^{[t]} | Q_n^{[1:t]}, S_n^{[0:t-1]}, U_n^{[1:t-1]}) = 0, \quad n \in \{1, \ldots, N\} \tag{3}$$

其中， $\Delta_\theta^{[t]}$ 是用户在时间 t生成的更新。

子模型安全性：不允许任何数据库获取有关子模型值的信息，即

$$I(W_{1:M}^{[t]}; S_n^{[t]}) = 0, \quad n \in \{1, \ldots, N\} \tag{4}$$

其中， $W_k^{[t]}$ 是时间 t的第 k个子模型。

读取阶段正确性：用户应在读取阶段正确下载所需的子模型，即

$$H(W_\theta^{[t-1]} | Q_{1:N}^{[t]}, A_{1:N}^{[t]}) = 0 \tag{5}$$

其中， $A_n^{[t]}$是时间 t从数据库 n 收到的答案。

写入阶段正确性：所有数据库中的子模型都必须正确更新，

$$W_m^{[t]} = \begin{cases} W_m^{[t-1]} + \Delta_m^{[t]}, & \text{if } m = \theta^{[t]} \\ W_m^{[t-1]}, & \text{if } m \neq \theta^{[t]}. \end{cases} \tag{6}$$

读取成本和写入成本分别定义为 $C_R = \frac{\mathcal{D}}{L}$ 和 $C_W = \frac{\mathcal{U}}{L}$，其中 $\mathcal{D}$ 和 $\mathcal{U}$ 分别是读取阶段和写入阶段下载和上传的符号总数。总成本 C_T 是读取成本与写入成本之和，即 $C_T = C_R + C_W$。在本文中，我们使用记号 $C_T(a, b)$ 表示子模型采用 (a, b) 最大距离可分码进行编码时的总成本。

III. 主要结果

定理1 对于给定的 PRUW 设置，其中有 N 个不串通数据库，且存储约束任意为 $\{1, \ldots, N\}$ $k = \frac{1}{n}$ $p = \sum_{n=1}^{N} \mu$ n $r = kp$ (),

, 令 $\max_n \mu()$, (), 并且 $s = \lfloor k \rfloor p$。那么，存在一种存储机制和一种 PRUW方案，能够完全填满这些 N 个数据库，并实现总通信成本 $\mathcal{C} = \min\{C_1, C_2\}$，其中

$$C_1 = (\lceil s \rceil - s) C_T(\lfloor k \rfloor, \lfloor s \rfloor) + (s - \lfloor s \rfloor) C_T(\lfloor k \rfloor, \lceil s \rceil) \tag{7}$$

$$C_2 = \alpha\beta C_T(\lfloor k \rfloor, \lfloor r \rfloor) + \alpha(1-\beta) C_T(\lfloor k \rfloor, \lceil r \rceil) + (1-\alpha)\delta C_T(\lceil k \rceil, \lfloor r \rfloor) + (1-\alpha)(1-\delta) C_T(\lceil k \rceil, \lceil r \rceil) \tag{8}$$

其中，

$$\alpha = \begin{cases} \frac{\lfloor k \rfloor (p\lceil k \rceil - \lceil r \rceil)}{\lceil k \rceil \lfloor r \rfloor - \lfloor k \rfloor \lceil r \rceil}, & \text{if } r - \lfloor r \rfloor > k - \lfloor k \rfloor,\ s \leq \lfloor r \rfloor \\ \frac{\lfloor k \rfloor}{k}(\lceil k \rceil - k), & \text{else} \end{cases} \tag{9}$$

$$\beta = \begin{cases} \frac{\lceil r \rceil - r}{\lceil k \rceil - k}, & \text{if } r - \lfloor r \rfloor > k - \lfloor k \rfloor,\ s > \lfloor r \rfloor \\ 1, & \text{else} \end{cases} \tag{10}$$

$$\delta = \begin{cases} 1 - \frac{r - \lfloor r \rfloor}{k - \lfloor k \rfloor}, & \text{if } r - \lfloor r \rfloor \leq k - \lfloor k \rfloor \\ 0, & \text{else}, \end{cases} \tag{11}$$

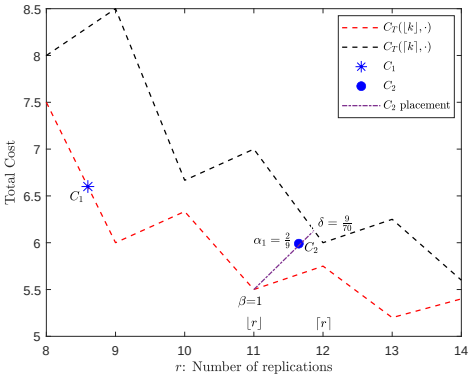

图 2. 包含 $k = 2.7$ 和 $p = 4.3$的示例设置。

if $\lfloor r \rfloor - \lfloor k \rfloor$ *is odd, and*

$$\alpha = \begin{cases} \frac{\lfloor k \rfloor}{k}(\lceil k \rceil - k), & \text{if } r - \lfloor r \rfloor < \lceil k \rceil - k \\ \frac{\lfloor k \rfloor (p\lceil k \rceil - \lfloor r \rfloor)}{\lceil k \rceil \lceil r \rceil - \lfloor k \rfloor \lfloor r \rfloor}, & \text{else} \end{cases} \tag{12}$$

$$\beta = \begin{cases} 1 - \frac{r - \lfloor r \rfloor}{\lceil k \rceil - k}, & \text{if } r - \lfloor r \rfloor < \lceil k \rceil - k \\ 0, & \text{else} \end{cases} \tag{13}$$

$$\delta = 1 \tag{14}$$

当 $\lfloor r \rfloor - \lfloor k \rfloor$ 为偶数时，函数 $C_T(a, b)$ 定义为：

$$C_T(a, b) = \begin{cases} \frac{4b}{b-a-1}, & \text{if } b - a \text{ is odd} \\ \frac{4b-2}{b-a-2}, & \text{if } b - a \text{ is even}. \end{cases} \tag{15}$$

备注1 作为主要结果的说明，考虑一个满足 $\max_n \mu(n) = 0.37$ 和 $p = \sum_{n=1}^{N} \mu(n) = 4.3$的示例。于是， $k = 2.7$、$r = 11.61$ 和 $s = 8.6$，由此得到如图 2 中蓝色星形所示的 $C_1 = 6.6$。由于 $\lfloor r \rfloor - \lfloor k \rfloor = 9$为奇数， $s = 8.6 < 11 = \lfloor r \rfloor$ 和 $r - \lfloor r \rfloor = 0.61 < 0.7 = k - \lfloor k \rfloor$，在计算 C_2时， α、 β和 δ分别由 $\alpha = \frac{\lfloor k \rfloor}{k}(\lceil k \rceil - k) = \frac{2}{9}$、 $\beta = 1$和 $\delta = 1 - \frac{r - \lfloor r \rfloor}{k - \lfloor k \rfloor} = \frac{9}{70}$给出。因此， $C_2 = 5.99$（图 2 中的蓝点）。注意，C_1 通过使用（$\lfloor k \rfloor$、 $\lfloor s \rfloor$）和（$\lfloor k \rfloor$、 $\lceil s \rceil$）最大距离可分码，分别存储所有子模型的 $\lceil s \rceil - s$ 和 $s - \lfloor s \rfloor$ 分数的参数而获得。类似地，如图 2 所示， C_2 通过使用（$\lfloor k \rfloor$、 $\lfloor r \rfloor$）、（$\lceil k \rceil$、 $\lfloor r \rfloor$）和（$\lceil k \rceil$、 $\lceil r \rceil$）MDS 码，分别存储所有子模型的 α、 $(1-\alpha)\delta$ 和 $(1-\alpha)(1-\delta_1)$ 分数而获得。该示例中可实现的最小成本为 $\min\{C_1, C_2\} = C_2$。

注释 2 总成本随着副本数 b 的增加而降低，并随着编码参数 a 的增加而提高，如式 (15) 所示。然而，当副本数（r 或 s）被选为编码参数 k的线性函数，即 $r = kp$、 $s = \lfloor k \rfloor p$时，总成本会随着编码参数的增加而降低；参见第四节-B2。因此，当$k \notin \mathbb{Z}^+$时，主要结果中的 C_2 通过同时考虑 $\lfloor k \rfloor$ 和 $\lceil k \rceil$ ，采用最大的 k 。然而，基于式 (15) 中总成本的波动结构（图 2 中的虚线），对于 k和 p 的特殊情形，仅考虑$\lfloor k \rfloor$也可以实现更低的总成本。这些情形由 C_1处理。

注释 3 对于具有任意存储约束 $\{\mu(\quad p=\sum_{n=1}^{N}\mu\ n\quad)\}_{n=1}^{N}$的给定 PRUW设置，设总可用存储为 ( )。那么，存在一种未编码 PRUW方案（第四节中提出的方案，其中 $k=1$ 和 $r=p$），分别在 $\lfloor p\rfloor$ 和 $\lceil p\rceil$ 个数据库中复制未编码子模型参数的 $\lceil p\rceil-p$和 $p-\lfloor p\rfloor$ 分数，并实现总成本 $(\lceil p\rceil-p)C_T(1,\lfloor p\rfloor)+(p-\lfloor p\rfloor)C_T(1,\lceil p\rceil)$。请注意，该总成本仅取决于所有任意存储约束之和（而不取决于各个约束），这与具有未编码异构存储约束的私有信息检索 *[45]*中的相应结果一致。然而，在 PRUW中，对于给定的 p，当 $k=\quad^{1}\max_{n}\mu(n)$ 最大化时，总成本最小，即所有存储约束相等时。

IV. 提出的方案

针对具有异构存储约束的数据库进行联邦子模型学习（FSL）的所提方案包含两个主要部分，即存储机制和 PRUW 方案。在本研究中，我们采用 [3] and[2], 中提出的 PRUW 方案，并提出一种存储机制，以便在任意给定的存储约束条件下执行联邦子模型学习。本节首先简要介绍 PRUW 方案，以确保内容完整；随后介绍所提出的存储机制。

A. PRUW 方案

该方案针对所有子模型的一个子分包（由 y 个编码比特组成）进行定义，然后以相同方式应用于所有子分包。

存储：在时间 t，数据库 n,$n\in\{1,\dots,R\}$ 中一个子分包的内容为：

$$S_n^{[t]}=\begin{bmatrix}\sum_{i=1}^{K}\frac{1}{f_{1,i}-\alpha_n}\begin{bmatrix}W_{1,1}^{[i]}\\ \vdots\\ W_{M,1}^{[i]}\end{bmatrix}+\sum_{j=0}^{y}\alpha_n^j Z_{1,j}\\ \vdots\\ \sum_{i=1}^{K}\frac{1}{f_{y,i}-\alpha_n}\begin{bmatrix}W_{1,y}^{[i]}\\ \vdots\\ W_{M,y}^{[i]}\end{bmatrix}+\sum_{j=0}^{y}\alpha_n^j Z_{y,j}\end{bmatrix},\quad(16)$$

其中，$W_{m}^{[i]}{}_{,j}$ 是第 i 个编码符号中第 j 个子模型的参数；m, $Z_{i,j}$ 是大小为 $M\times 1$ 的随机噪声向量；$f_{i,j}$ 和 α_n 是来自 $\mathbb{F}_q$ 的全局已知互异常数。注意，第 j 个编码数据符号（不含噪声）以 (K,R) 最大距离可分码的形式存储。

读取阶段：在读取阶段，用户向数据库发送查询，以下载所需的子模型 W_θ。发送给数据库 n, $n\in\{1,\dots,R\}$ 的查询，用于下载 W_θ 中每个子分包的 Ky 参数，其形式如下：

$$Q_{n,\ell}=\begin{bmatrix}\frac{\prod_{i=1,i\neq\ell}^{K}(f_{1,i}-\alpha_n)}{\prod_{i=1,i\neq\ell}^{k}(f_{1,i}-f_{1,\ell})}e_M(\theta)+\prod_{i=1}^{K}(f_{1,i}-\alpha_n)\tilde{Z}_{1,\ell}\\ \vdots\\ \frac{\prod_{i=1,i\neq\ell}^{K}(f_{y,i}-\alpha_n)}{\prod_{i=1,i\neq\ell}^{k}(f_{y,i}-f_{y,\ell})}e_M(\theta)+\prod_{i=1}^{K}(f_{y,i}-\alpha_n)\tilde{Z}_{y,\ell}\end{bmatrix}\quad(17)$$

对于 $\ell\in\{1,\dots,K\}$，其中 $e_M(\theta)$ 是大小为 $M\times 1$ 的全零向量，其第 θ 个位置为 1，且 $Z_{j,\ \ell}$ 是大小为 $M\times 1$ 的随机噪声向量。然后，数据库 n 生成相应的答案如下：

$$A_{n,\ell}=S_n^TQ_{n,\ell}=\sum_{j=1}^{y}\frac{1}{f_{j,\ell}-\alpha_n}W_{\theta,j}^{[\ell]}+P_{\alpha_n}(K+y),\quad(18)$$

对于每个 $\ell\in\{1,\dots,K\}$，其中 $P_{\alpha_n}(h)$ 是关于 α_n 的 h 次多项式。利用 [1] 中的引理 1 得到式 (18)。用户通过从 $R'=2y+K+1$ 个数据库下载式 (18) 中的答案，获得 W_θ 每个子分包的 Ky 个参数，因为对于固定的 ℓ，每组答案中都有 y 个 W_θ 参数和 $K+y+1$ 个多项式系数需要求解。由于所考虑的数据库总数为 R，若 $R-K$ 为奇数，则 $R'=R$；若 $R-K$ 为偶数，则 $R'=R-1$，因为 $y\in\mathbb{Z}^+$。因此，子分包化 y 为 $y=\frac{R'-K-1}{2}$，由此得到读取成本：

$$C_R=\frac{R'\times K}{y\times K}=\frac{2R'}{R'-K-1}.\quad(19)$$

写入阶段：在写入阶段，用户向 R 个数据库中的每个数据库发送 K 个符号。第 ℓ 个符号（在上传的 K 个符号中）是组合更新，由 W_θ 的更新组成，对应于每个子分包中 y 个编码参数各自的第 ℓ 个参数。发送给数据库 n、 $n\in\{1,\dots,R\}$ 的 K 个组合更新为：

$$U_{n,\ell}=\sum_{j=1}^{y}\prod_{i=1,i\neq j}^{y}(f_{i,\ell}-\alpha_n)\tilde{\Delta}_{\theta,j}^{[\ell]}+\prod_{i=1}^{y}(f_{i,\ell}-\alpha_n)\hat{z}_\ell,\quad(20)$$

对于 $\ell\in\{1,\dots,K\}$，其中 $\tilde{\Delta}_{\theta}^{[\ell]}{}_{,\ j}=\frac{\prod_{i=1}^{K}{}_{,\ i\neq\ell}(f_{j,\ i}-f_{j,\ \ell})}{\prod_{i=1,\ i\neq j}^{y}(f_{i,\ \ell}-f_{j,\ \ell})}\Delta_{\theta}^{[\ell]}{}_{,\ j}$对于 $j\in\{1,\dots,y\}$，其中 $\Delta_{\theta}^{[\ell]}{}_{,\ j}$是所考虑的子分包 W_θ 中第 j 个编码符号的第 ℓ 个参数的更新，而 $\hat{}z_\ell$ 是一个随机噪声符号。数据库 n 接收到所有$U_{n,\ \ell}$后，会计算增量更新，如下所示：

$$\bar{U}_{n,\ell}=U_{n,\ell}\times\tilde{D}_{n,\ell}\times Q_{n,\ell}\quad(21)$$

$$=\begin{bmatrix}\frac{1}{f_{1,\ell}-\alpha_n}\Delta_{\theta,1}^{[\ell]}e_M(\theta)+P_{\alpha_n}^{[1]}(y)\\ \vdots\\ \frac{1}{f_{y,\ell}-\alpha_n}\Delta_{\theta,y}^{[\ell]}e_M(\theta)+P_{\alpha_n}^{[y]}(y)\end{bmatrix},\quad(22)$$

˜其中 $D_{n,\ \ell}$ 是一个缩放矩阵，定义如下：

$$\tilde{D}_{n,\ell}=\mathrm{diag}\left(\frac{1_M}{\prod_{i=1}^{K}(f_{1,i}-\alpha_n)},\dots,\frac{1_M}{\prod_{i=1}^{K}(f_{y,i}-\alpha_n)}\right)\quad(23)$$

其中，大小为 $1\times M$ 的全1向量表示为 1_M。由于 ¯增量更新 $(U_{n,\ \ell})$ 与（$\sum_{\ell=1}^{K}\bar{U}_{n\ \ell}^{16}\quad S_n^{[t]}=S_n^{[t-1]}+$）中的存储形式相同，因此通过，更新子模型。由此得到的写入成本和总成本为，

$$C_W=\frac{K\times R}{K\times y}=\frac{2R}{R'-K-1},\quad(24)$$

$$C_T=\frac{2(R+R')}{R'-K-1}=\begin{cases}\frac{4R}{R-K-1}, & \text{if } R-K \text{ is odd}\\ \frac{4R-2}{R-K-2}, & \text{if } R-K \text{ is even.}\end{cases}\quad(25)$$

B. 存储机制

所提出的存储机制由子模型分区和子模型编码组成，其中参数使用特定的（K, R）最大距离可分码进行编码，并分块存储在数据库的不同子集中。本节将求出最优编码参数以及存储在各数据库中的子模型分数，以使总成本最小。

1) 子模型分区：这决定了所有 (K, R) 最大距离可分码编码子模型中，存储在具有任意存储约束的N 个数据库中的分数。根据 (K, R) 最大距离可分码编码结构，每个编码参数必须恰好存储在 R 个数据库中。令 η_i 表示存储在相同 R 个数据库子集（由 i索引的数据库子集）中的所有子模型的分数。令 $\mathcal{B}$ 表示由所有元素属于 $\{0,1\}$ 且恰好含有 R个 1 的 $N \times 1$ 向量组成的基，记为 $\mathcal{B} = \{b_1, b_2, \ldots, b_{|\mathcal{B}|}\}$。注意，每个 b_i 都对应于由 i索引的 R 个数据库的特定子集。因此，需要求出满足以下条件的 η_i:

$$\frac{1}{K}\sum_{i=1}^{|\mathcal{B}|}\eta_i b_i = \mu, \quad \text{where} \quad \mu = [\mu(1), \ldots, \mu(N)]^T \tag{26}$$

$$\sum_{i=1}^{|\mathcal{B}|}\eta_i = 1, \quad 0 \le \eta_1, \ldots, \eta_{|\mathcal{B}|} \le 1, \tag{27}$$

以便将每个编码参数复制到恰好 R 个数据库中，同时确保所有数据库均被完全填满。文献 [45] 和 [46] 给出了该问题在未编码参数 $(K = 1)$ 情况下的解，以及解存在的充要条件，如下所示:

$$\mu(n) \le \frac{\sum_{n=1}^{N}\mu(n)}{R}, \quad n \in \{1, \ldots, N\}. \tag{28}$$

在本工作中，我们求出最优编码参数 K 和R，使 PRUW 过程的总成本最小，同时满足 (28)，并求解 (26)-(27)，以确定要存储在各数据库中的编码子模型分区η_i 。

2) 子模型编码：对于给定的一组存储约束 $\{\mu\ (p = \sum_{n=1}^{N}\mu\ n\ )\}_{n=1}^{N}$，可用存储总量为 ( )，本质上等于模型中每个未编码参数在数据库系统中可复制的最大次数。设模型参数采用 (K, R) 最大距离可分码存储。因此，模型中的编码参数总数为 $\frac{ML}{K}$，这意味着每个编码参数最多可以复制到

$$R \le \frac{\sum_{n=1}^{N}\mu(n)ML}{ML/K} = Kp \tag{29}$$

个数据库中，前提是 $Kp \le N$。同时，每个数据库中的存储量必须满足:

$$\mu(n)ML \le \frac{ML}{K}, \quad n \in \{1, \ldots, N\} \implies K \le \frac{1}{\bar{\mu}}, \tag{30}$$

其中 $\bar{\mu} = \max_n \mu(n)$，以充分利用每个数据库中的可用空间。因此，编码参数 K

case	最大距离可分码	分数 of 子模型	已分配空间在数据库 n中
1	$(\lfloor k \rfloor, \lfloor r \rfloor)$	$\alpha\beta$	$\hat{\mu}_1(n)$
2	$(\lfloor k \rfloor, \lceil r \rceil)$	$\alpha(1-\beta)$	$\hat{\mu}_2(n)$
3	$(\lceil k \rceil, \lfloor r \rfloor)$	$(1-\alpha)\delta$	$\bar{\mu}_1(n)$
4	$(\lceil k \rceil, \lceil r \rceil)$	$(1-\alpha)(1-\delta)$	$\bar{\mu}_2(n)$

表 I 子模型的分数及对应的最大距离可分码。

必须满足,

$$K \le \min\left\{\frac{1}{\bar{\mu}}, \frac{N}{p}\right\} = \frac{1}{\bar{\mu}}, \tag{31}$$

由于 $p = \sum_{n=1}^{N}\mu(n) \le \bar{\mu}N$ 意味着 $\frac{1}{\bar{\mu}} \le \frac{N}{p}$。由 (25) 可知，PRUW方案的总成本随副本数 R 的增加而降低，因此，对于给定的 K，满足 (31) 的 $p \in \mathbb{Z}^+$ 中，使 (K, R) 最大距离可分码达到最优的 R 由 (29) 给出的 Kp 决定，所得总成本为,

$$C_T(K, R) = \begin{cases} \frac{4Kp}{Kp-K-1}, & \text{odd } K, \text{ even } p \\ \frac{4Kp-2}{Kp-K-2}, & \text{otherwise,} \end{cases} \tag{32}$$

它随 K 增加而降低。因此，对于给定的一组存储约束 $\{\mu(n)\}_{n=1}^{N}$，由 (16) 中的 (K, R) 最大距离可分码实现的最小总成本，其中 K 和 R 分别为,

$$K = \frac{1}{\bar{\mu}} = k, \quad R = kp = r, \tag{33}$$

由于对所有 n，$\sum_{n=1}^{N}\mu\ (n)\ _r \quad = \bar{\mu} \ge \mu(n)$，因此它自动满足 (28)。然而，由于 k 和 r 不一定是整数，因此需要进行额外计算，以获得最优的 (k, r) 最大距离可分码，其中 k、$r \in \mathbb{Z}^+$ 共同使总成本达到最低。考虑一般情形，其中 k、$r \notin \mathbb{Z}^+$。这里的总体思路是将每个子模型划分为四个分段，并使用表 I 中给出的四种最大距离可分码对其进行编码。四个分段的大小由参数 α、β 和 δ 定义，其中 $0 \le \alpha$、β、$\delta \le 1$，如表 I 所示。整个数据库系统中为这四种最大距离可分码分配的总空间为,

$$\sum_{n=1}^{N}\hat{\mu}_1(n) = \frac{\alpha\beta\lfloor r \rfloor}{\lfloor k \rfloor}, \qquad \sum_{n=1}^{N}\hat{\mu}_2(n) = \frac{\alpha(1-\beta)\lceil r \rceil}{\lfloor k \rfloor} \tag{34}$$

$$\sum_{n=1}^{N}\bar{\mu}_1(n) = \frac{(1-\alpha)\delta\lfloor r \rfloor}{\lceil k \rceil}, \sum_{n=1}^{N}\bar{\mu}_2(n) = \frac{(1-\alpha)(1-\delta)\lceil r \rceil}{\lceil k \rceil} \tag{35}$$

每个数据库中为各个最大距离可分码已分配的空间都必须分别满足（28），以确保所有数据库都被完全填满，同时每个编码参数都在相应数量的数据库中复制，即:

$$\hat{\mu}_1(n) \le \frac{\alpha\beta}{\lfloor k \rfloor}, \qquad \hat{\mu}_2(n) \le \frac{\alpha(1-\beta)}{\lfloor k \rfloor} \tag{36}$$

$$\bar{\mu}_1(n) \le \frac{(1-\alpha)\delta}{\lceil k \rceil}, \qquad \bar{\mu}_2(n) \le \frac{(1-\alpha)(1-\delta)}{\lceil k \rceil}. \tag{37}$$

每个数据库中的四项存储分配都必须满足,

$$\hat{\mu}_1(n) + \hat{\mu}_2(n) + \bar{\mu}_1(n) + \bar{\mu}_2(n) = \mu(n), \quad \forall n. \tag{38}$$

接下来需要求出 $\hat{}\mu_1(n)$、$\hat{}\mu_2(n)$、 $\bar{\mu}_1(n)$、 $\bar{\mu}_2(n)$、α、 β 和 δ 的值，使总成本最小，如下所示，

$$\begin{aligned}\mathcal{C}=&\alpha\left(\beta C_T(\lfloor k\rfloor,\lfloor r\rfloor)+(1-\beta)C_T(\lfloor k\rfloor,\lceil r\rceil)\right)\\&+(1-\alpha)\left(\delta C_T(\lceil k\rceil,\lfloor r\rfloor)+(1-\delta)C_T(\lceil k\rceil,\lceil r\rceil)\right)\end{aligned}\quad(39)$$

同时满足 (34)–(38)，其中 C_T 定义于 (15)。我们考虑两种情况：1) $\alpha=1$，2) $\alpha<1$。当 $\alpha=1$时，存储仅使用表 I 中的情况 1、2，且 (28) 变为$R\le\lfloor k\rfloor p$，从而得到 $\lfloor k\rfloor p=s$ 个最大副本数。当$\alpha=1$时，这将表 I 和 (34)–(37) 中的所有“r”替换为 s 。

引理 1 当 $\alpha=1$时，为满足 (34) 和 (38)， β固定为 $\beta=\lceil s\rceil-s$，且满足 (36) 的 $\hat{}\mu_1(n)$、$\hat{}\mu_2(n)$ 由下式给出，

$$\hat{\mu}_1(n)=\tilde{m}(n)+(\mu(n)-\tilde{m}(n)-\tilde{h}(n))\tilde{\gamma}\quad(40)$$

$$\hat{\mu}_2(n)=\tilde{h}(n)+(\mu(n)-\tilde{m}(n)-\tilde{h}(n))(1-\tilde{\gamma})\quad(41)$$

对于所有 $n\in\{1,\dots,N\}$，其中，

$$\tilde{m}(n)=\left[\mu(n)-\frac{s-\lfloor s\rfloor}{\lfloor k\rfloor}\right]^+,\quad\tilde{h}(n)=\left[\mu(n)-\frac{\lceil s\rceil-s}{\lfloor k\rfloor}\right]^+\quad(42)$$

$$\tilde{\gamma}=\frac{\frac{\lfloor s\rfloor}{\lfloor k\rfloor}(\lceil s\rceil-s)-\sum_{n=1}^{N}\tilde{m}(n)}{p-\sum_{n=1}^{N}\tilde{m}(n)-\sum_{n=1}^{N}\tilde{h}(n)},\quad(43)$$

其中 $[x]^+=\max\{x,0\}$。由此得到的总成本为 (7).

引理 2 以下 $\hat{}\mu_1(n)$、$\hat{}\mu_2(n)$、 $\bar{\mu}_1(n)$ 和 $\bar{\mu}_2(n)$ 对于 $n\in\{1,\dots,N\}$ 满足 (34)-(38)，且任意 $\alpha<1,\beta$ 以及满足 $\alpha\ge$的 δ $\frac{\lfloor k\rfloor}{k}(\lceil k\rceil-k)$，$\beta\ge\left[1-\frac{\lfloor k\rfloor}{k\alpha}(r-\lfloor r\rfloor)\right]^+$ 以及 $\delta\ge\left[1-\frac{\lceil k\rceil}{k\,1-\alpha()}(r-\lfloor r\rfloor)\right]^+$.

$$\hat{\mu}_1(n)=\begin{cases}\hat{\mu}(n)\beta, & \text{if } \beta\in\{0,1\}\\ \hat{m}(n)+(\hat{\mu}(n)-\hat{m}(n)-\hat{h}(n))\hat{\gamma}, & \text{if } \beta\in(0,1)\end{cases}\quad(44)$$

$$\hat{\mu}_2(n)=\begin{cases}\hat{\mu}(n)(1-\beta), & \text{if } \beta\in\{0,1\}\\ \hat{h}(n)+(\hat{\mu}(n)-\hat{m}(n)-\hat{h}(n))(1-\hat{\gamma}), & \text{if } \beta\in(0,1)\end{cases}\quad(45)$$

$$\bar{\mu}_1(n)=\begin{cases}\bar{\mu}(n)\delta, & \text{if } \delta\in\{0,1\}\\ \bar{m}(n)+(\bar{\mu}(n)-\bar{m}(n)-\bar{h}(n))\bar{\gamma}, & \text{if } \delta\in(0,1)\end{cases}\quad(46)$$

$$\bar{\mu}_2(n)=\begin{cases}\bar{\mu}(n)(1-\delta), & \text{if } \delta\in\{0,1\}\\ \bar{h}(n)+(\bar{\mu}(n)-\bar{m}(n)-\bar{h}(n))(1-\bar{\gamma}), & \text{if } \delta\in(0,1)\end{cases}\quad(47)$$

对于所有 $n\in\{1,\dots,N\}$，其中，

$$\hat{\mu}(n)=m(n)+(\mu(n)-m(n)-h(n))\gamma\quad(48)$$

$$\bar{\mu}(n)=h(n)+(\mu(n)-m(n)-h(n))(1-\gamma)\quad(49)$$

$$m(n)=\left[\mu(n)-\frac{1-\alpha}{\lceil k\rceil}\right]^+,\quad h(n)=\left[\mu(n)-\frac{\alpha}{\lfloor k\rfloor}\right]^+\quad(50)$$

$$\gamma=\frac{\frac{\alpha}{\lceil k\rceil}(\lceil r\rceil-\beta)-\sum_{n=1}^{N}m(n)}{p-\sum_{n=1}^{N}m(n)-\sum_{n=1}^{N}h(n)}\quad(51)$$

$$\hat{m}(n)=\left[\hat{\mu}(n)-\frac{\alpha(1-\beta)}{\lfloor k\rfloor}\right]^+,\quad\hat{h}(n)=\left[\hat{\mu}(n)-\frac{\alpha\beta}{\lfloor k\rfloor}\right]^+\quad(52)$$

$$\hat{\gamma}=\frac{\frac{\alpha\beta}{\lfloor k\rfloor}\lfloor r\rfloor-\sum_{n=1}^{N}\hat{m}(n)}{\frac{\alpha}{\lfloor k\rfloor}(\lceil r\rceil-\beta)-\sum_{n=1}^{N}\hat{m}(n)-\sum_{n=1}^{N}\hat{h}(n)}\quad(53)$$

$$\bar{m}(n)=\left[\bar{\mu}(n)-\frac{(1-\alpha)(1-\delta)}{\lceil k\rceil}\right]^+,\quad\bar{h}(n)=\left[\bar{\mu}(n)-\frac{(1-\alpha)\delta}{\lceil k\rceil}\right]^+\quad(54)$$

$$\bar{\gamma}=\frac{\frac{(1-\alpha)\delta}{\lceil k\rceil}\lfloor r\rfloor-\sum_{n=1}^{N}\bar{m}(n)}{\frac{1-\alpha}{\lceil k\rceil}(\lceil r\rceil-\delta)-\sum_{n=1}^{N}\bar{m}(n)-\sum_{n=1}^{N}\bar{h}(n)}.\quad(55)$$

引理3 对于 $\alpha<1$ 的情况，在满足引理 2 中约束的同时，使式 (39) 中总成本最小化的 α、 β 和 δ 的取值由定理 1给出，相应的总成本见式 (8)。

一旦根据引理 1-3 确定表 I 中的所有参数，就可以使用 [45] 和 [46] 中提出的方法求解 (26)-(27)，即分别求出四组存储分配 $\{\hat{}\mu_1\}_{n=1}^N$、 $\{\hat{}\mu_2\}_{n=1}^N$、 $\{\bar{\mu}_1\}_{n=1}^N$、 $\{\bar{\mu}_2\}_{n=1}^N$ 对应的 η_i。这四组解确定了每个独立编码符号在数据库系统中的复制位置，从而完成存储机制。

作为示例，考虑一个包含 $N=12$ 个数据库的 P RUW设置，其中 $\mu(1)=\ldots=\mu(5)=0.37$，且 $\mu(6)=\ldots=\mu(12)=0.35$。注意，这里 k、 r 和 s 的取值与备注 1中考虑的相同，因此得到的 α、 β、 δ 以及总成本也与备注1中所述相同。利用 (44)-(55)，我们得到存储分配如下:

$$\hat{\mu}_1(n)=\hat{\mu}(n)=\begin{cases}0.1107, & \text{for } n=1,\dots5\\ 0.0951, & \text{for } n=6,\dots12\end{cases}\quad(56)$$

$$\bar{\mu}_1(n)=\begin{cases}0.033, & \text{for } n=1,\dots5\\ 0.029, & \text{for } n=6,\dots12\end{cases}\quad(57)$$

$$\hat{\mu}_2(n)=0,\quad\bar{\mu}_2(n)=0.226,\quad\forall n.\quad(58)$$

注意，根据 (58)，并未使用 $(\lfloor k\rfloor,\lceil r\rceil)$ 最大距离可分码，而且所有数据库中为 $(\lceil k\rceil,\lceil r\rceil)$ 最大距离可分码分配的空间相同。因此，采用 [3] 中提出的、适用于同构存储约束下 PRUW 的最优存储机制，来放置 $(\lceil k\rceil,\lceil r\rceil)$ 编码参数。$(\lfloor k\rfloor,\lfloor r\rfloor)$ 和 $(\lceil k\rceil,\lfloor r\rfloor)$ 码的存储分配在不同数据库之间并不相同，如 (56) 和 (57) 中两组不同的数值所示。因此，对于这两种情况，我们分别求解 (26)–(27)，以确定每个数据库中要存储的编码参数分数。例如，对于 $(\lfloor k\rfloor,\lfloor r\rfloor)$ 最大距离可分码，我们求解 (26)–(27)，其中 $N=12$, $K=\lfloor k\rfloor=2$,$R=\lfloor r\rfloor=11$ 和 $\mu=[\hat{}\mu_1(1):\hat{}\mu_1(12)]$ 分别为 $\eta_1=\ldots=\eta_5=2.73\times10^{-4}$ 和 $\eta_6=\ldots=\eta_{12}=3.15\times10^{-2}$; 其中，(26) 中与每个 η_i 对应的 b_i是一个大小为12×1 的全1向量，其第 i个位置为零。这实质上意味着，该存储机制将 $(\lfloor k\rfloor,\lfloor r\rfloor)$ 编码参数的 η_i 分配给包含除第 i个数据库之外所有数据库的 $\lfloor r\rfloor=11$ 个数据库子集。

参考文献

[1] S. Vithana 和 S. Ulukus。联邦子模型学习（FSL）中的私有读取更新写入（PRUW）：采用和不采用稀疏化的通信高效方案。可在线访问 arXiv:2209.04421。 [2] Z. Jia 和 S. A. Jafar。具有弹性丢弃鲁棒性的 X-安全、 T-私有联邦子模型学习。IEEE 信息论汇刊，68(8):5418–5439，2022年8月。 [3] S. Vithana 和 S. Ulukus。存储受限数据库下的私有读取更新写入（PRUW）。发表于 IEEE 国际信息论研讨会，2022年6月。 [4] S. Vithana 和 S. Ulukus。采用稀疏化的私有联邦子模型学习。发表于 IEEE 信息论研讨会，2022年11月。 [5] S. Vithana 和 S. Ulukus。联邦子模型学习中私有读取更新写入的率失真权衡。发表于阿西洛马会议，2022年10月。[6] S. Vithana 和 S. Ulukus。高效私有联邦子模型学习。发表于 IEEE 国际通信会议，2022年5月。 [7] S. Vithana 和 S. Ulukus。采用 top r 稀疏化、面向存储高效私有联邦学习的模型分割。可在线访问 arXiv:2212.11947。[8] S. Vithana 和 S. Ulukus。采用 top r 稀疏化的联邦学习中的率-隐私-存储权衡。可在线访问 arXiv:2212.09704。 [9] C. Niu、F. Wu、S. Tang、L. Hua、R. Jia、C. Lv、Z. Wu 和 G. Chen。安全联邦子模型学习。可在线访问 arXiv:1911.02254。 [10] M. Kim 和 J. Lee。联邦子模型学习中的信息论隐私。可在线访问 arXiv:2008.07656。 [11] C. Niu、F. Wu、S. Tang、L. Hua、R. Jia、C. Lv、Z. Wu 和 G. Chen。移动客户端上的十亿规模联邦学习：一种具有可调隐私的子模型设计。发表于 MobiCom，2020年4月。 [12] S. Ulukus、S. Avestimehr、M. Gastpar、S. A. Jafar、R. Tandon 和 C. Tian。私有检索、计算与学习：近期进展与未来挑战。IEEE精选通信领域期刊，40(3):729–748，2022年3月。 [13] B. Chor、E. Kushilevitz、O. Goldreich 和 M. Sudan。私有信息检索。美国计算机协会期刊，45(6):965–981，1998年11月。 [14] H. Sun 和 S. A. Jafar。私有信息检索容量。IEEE 信息论汇刊，63(7):4075–4088，2017年7月。 [15] C. Tian、H. Sun 和 J. Chen。具有最优消息大小和上传成本的容量可达私有信息检索码。IEEE 信息论汇刊，65(11):7613–7627，2019年11月。 [16] S. Kumar、H.-Y. Lin、E. Rosnes 和 A. G. i Amat。使用线性码实现最大距离可分私有信息检索容量。IEEE 信息论汇刊，65(7):4243–4273，2019年7月。 [17] H. Sun 和 S. A. Jafar。对称私有信息检索容量。IEEE 信息论汇刊，65(1):322–329，2019年1月。 [18] Z. Wang、K. Banawan 和 S. Ulukus。私有集合求交：多消息对称私有信息检索视角。IEEE 信息论汇刊，68(3):2001–2019，2022年3月。 [19] K. Banawan 和 S. Ulukus。多消息私有信息检索：容量结果与近最优方案。IEEE 信息论汇刊，64(10):6842–6862，2018年10月。 [20] T. Chan、S. Ho 和 H. Yamamoto。编码存储的私有信息检索。发表于 IEEE 国际信息论研讨会，2015年6月。 [21] A. Fazeli、A. Vardy 和 E. Yaakobi。用于低存储开销分布式私有信息检索的编码。发表于 IEEE 国际信息论研讨会，2015年6月。 [22] K. Banawan 和 S. Ulukus。编码数据库上的私有信息检索容量。IEEE 信息论汇刊，64(3):1945–1956，2018年3月。 [23] R. Tajeddine 和 S. El Rouayheb。编码数据上的鲁棒私有信息检索。发表于 IEEE 国际信息论研讨会，2017年6月。 [24] H. Sun 和 S. A. Jafar。多轮私有信息检索：容量与存储开销。IEEE 信息论汇刊，64(8):5743–5754，2018年8月。 [25] H. Sun 和 S. A. Jafar。任意消息长度下私有信息检索的最优下载成本。IEEE 信息取证与安全汇刊，12(12):2920–2932，2017年12月。 [26] Q. Wang 和 M. Skoglund。关于存在对手和窃听者时串通数据库下的 PIR 与对称 PIR。IEEE 信息论汇刊，65(5):3183–3197，2019年5月。 [27] H. Sun 和 S. A. Jafar。串通数据库下鲁棒私有信息检索的容量。IEEE 信息论汇刊，64(4):2361–2370，2018年4月。 [28] R. Freij-Hollanti、O. Gnilke、C. Hollanti 和 D. Karpuk。合谋服务器下编码数据库上的私有信息检索。SIAM 应用代数与几何杂志，1(1):647–664，2017年。 [29] H. Sun 和 S. A. Jafar。合谋服务器下从 MDS 编码数据进行私有信息检索：解决 Freij-Hollanti 等人提出的一个猜想。IEEE 信息论汇刊，64(2):1000–1022，2018年2月。 [30] R. Tandon。缓存辅助私有信息检索的容量。载于阿勒顿会议，2017年10月。 [31] Y.-P. Wei、K. Banawan 和 S. Ulukus。未知且未编码预取时缓存辅助私有信息检索的基本极限。IEEE 信息论汇刊，65(5):3215–3232，2019年5月。 [32] I. Samy、M. Attia、R. Tandon 和 L. Lazos。非对称泄漏私有信息检索。IEEE 信息论汇刊，67(8):5352–5369，2021年8月。 [33] I. Samy、R. Tandon 和 L. Lazos。关于泄漏式私有信息检索的容量。载于 IEEE 信息论研讨会，页码 1262–1266，2019年7月。 [34] S. Kadhe、B. Garcia、A. Heidarzadeh、S. El Rouayheb 和 A. Sprintson。带侧信息的私有信息检索。IEEE 信息论汇刊，66(4):2032–2043，2020年4月。 [35] Q. Wang、H. Sun 和 M. Skoglund。存在窃听者时私有信息检索的容量。IEEE 信息论汇刊，65(5):3198–3214，2019年5月。 [36] Q. Wang、H. Sun 和 M. Skoglund。存在窃听者时私有信息检索的容量。IEEE 信息论汇刊，65(5):3198–3214，2019年5月。 [37] K. Banawan 和 S. Ulukus。拜占庭数据库和合谋数据库上的私有信息检索容量。IEEE 信息论汇刊，65(2):1206–1219，2019年2月。 [38] S. Vithana、K. Banawan 和 S. Ulukus。语义私有信息检索。IEEE 信息论汇刊，68(4):2635–2652，2022年4月。 [39] S. Li 和 M. Gastpar。带侧信息的单服务器多消息私有信息检索：一般情形。载于 IEEE 国际信息论研讨会，2020年6月。 [40] Z. Jia 和 S. A. Jafar。从 MDS 编码存储中进行X安全、 T私有信息检索，且服务器可能为拜占庭服务器或无响应服务器。IEEE 信息论汇刊，66(12):7427–7438，2020年12月。 [41] Z. Jia、H. Sun 和 S. A. Jafar。跨子空间对齐与 X安全、 T私有信息检索的渐近容量。IEEE 信息论汇刊，65(9):5783–5798，2019年9月。 [42] J. Xu 和 Z. Zhang。在小域上构建具有最优子分包的达到容量的 PIR 方案。载于 IEEE 国际信息论研讨会，页码 1749–1753，2018年6月。 [43] H. Yang、W. Shin 和 J. Lee。面向安全分布式存储系统的私有信息检索。IEEE 信息取证与安全汇刊，13(12):2953–2964，2018年12月。 [44] K. Banawan 和 S. Ulukus。多址信道中的私有信息检索。载于 IEEE 信息论研讨会，2018年11月。 [45] K. Banawan、B. Arasli、Y.-P. Wei 和 S. Ulukus。异构未编码缓存数据库上的私有信息检索容量。IEEE 信息论汇刊，66(6):3407–3416，2020年6月。 [46] N. Woolsey、R. Chen 和 M. Ji。针对存储受限数据库上私有信息检索的线性子消息未编码放置。IEEE 通信汇刊，68(10):6039–6053，2020年10月。 [47] K. Banawan、B. Arasli 和 S. Ulukus。用于高效私有信息检索的改进存储。载于 IEEE 信息论研讨会，2019年8月。 [48] Y.-P. Wei、B. Arasli、K. Banawan 和 S. Ulukus。去中心化未编码缓存数据库上的私有信息检索容量。Information，10，2019年12月。 [49] R. Tandon、M. Abdul-Wahid、F. Almoualem 和 D. Kumar。存储受限数据库上的 PIR——编码缓存与 PIR 的结合。IEEE 国际通信会议，2018年5月。 [50] M. A. Attia、D. Kumar 和 R. Tandon。未编码存储受限数据库上的私有信息检索容量。IEEE 信息论汇刊，66(11):6617–6634，2020年11月。 [51] Y.-P. Wei 和 S. Ulukus。存储约束下带私有侧信息的私有信息检索容量。IEEE 信息论汇刊，66(4):2023–2031，2020年4月。