



研究与开发

## Least core: 联邦学习中高效稳定的参与者贡献评价机制

林国鹏, 倪郑威

(浙江工商大学信息与电子工程学院, 浙江 杭州 310018)

**摘要:** 为了应对数据隐私保护引发的“数据孤岛”问题, 联邦学习提供了一种允许多方在不共享原始数据的情况下联合训练模型的技术框架。然而, 为了吸引高质量数据拥有者参与并维持系统的长期稳定运行, 公平合理的价值评估和收益分配机制至关重要。尽管 Shapley 值被广泛用于联邦学习中的贡献评估, 但其单一性限制了对价值的多维考量。为此, 引入最小核心 (Least core) 方法, 通过最小化最大不满意度, 为价值分配提供新视角。然而, Least core 方法的计算复杂性与 Shapley 值相似, 均存在一定的计算限制, 需进一步结合多种抽样方式, 近似计算分配结果, 在显著降低大规模场景下计算成本的同时, 确保分配的准确性。实验结果表明, 该机制在确保公平性和系统稳定性的同时, 有效提升了计算效率, 为多样化的联邦学习场景提供了可行的解决方案。

**关键词:** 联邦学习; 博弈论; 最小核心; 贡献评估; 激励机制

**中图分类号:** TN91; TP309

**文献标志码:** A

**doi:** 10.11959/j.issn.1000-0801.2026011

## Least core: an efficient and stable participant contribution evaluation mechanism in federated learning

Lin Guopeng, Ni Zhengwei

College of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

**Abstract:** To address the “data silo” problem caused by data privacy protection, federated learning offers a technological framework that enables multiple parties to collaboratively train models without sharing raw data. However, to attract high-quality data owners and ensure the long-term stable operation of the system, a fair and reasonable value evaluation and revenue distribution mechanism is essential. Although the Shapley value has been widely used in federated learning for contribution evaluation, relying on a single approach limits the multi-perspective assessment of data contribution and value distribution. For this purpose, the Least core method was introduced, providing a new perspective for value allocation by minimizing the maximum deficit. However, the computational complexity of the Least core method, similar to that of the Shapley value, posed certain computational limitations. By further combining vari-

收稿日期: 2025-04-16; 修回日期: 2025-09-30

通信作者: 倪郑威, zhengwei.ni@zjgsu.edu.cn

基金项目: 桐乡市通用人工智能研究院项目 (No.TAGI2-B-2024-0017)

**Foundation Item:** The Tongxiang General Artificial Intelligence Research Institute Project (No.TAGI2-B-2024-0017)



ous sampling methods to approximate the allocation results, it significantly reduced the computational cost in large-scale scenarios while ensuring the accuracy of the allocation. Experimental results demonstrate that the mechanism ensures fairness and system stability while effectively improving computational efficiency, providing a feasible solution for diverse federated learning scenarios.

**Key words:** federated learning, game theory, Least core, contribution evaluation, incentive mechanism

## 0 引言

近年来,机器学习技术越发成熟,在诸多领域出现了广泛的应用实例,与此同时,数据隐私问题也日益凸显,成为公众所关注的焦点。随着各地区或国家隐私法规的出台,如欧盟的《通用数据保护条例》(General Data Protection Regulation, GDPR)<sup>[1]</sup>、美国的《加州消费者隐私法案》(California Consumer Privacy Act, CCPA)<sup>[2]</sup>、我国的《个人信息保护法》(Personal Information Protection Law, PIPL)<sup>[3]</sup>，“数据孤岛”问题日益严重,限制了机器学习的发展与应用。联邦学习(federated learning, FL)<sup>[4]</sup>作为一种分布式训练框架,能够在不共享原始数据的前提下,实现多方协同训练全局模型,已在医疗、金融等领域展现出广泛的潜力<sup>[5-6]</sup>。然而,联邦学习的长期稳定运行依赖于合理的参与者贡献评估与收益分配机制,若贡献者无法获得公平回报,将导致高质量数据拥有者退出合作,最终影响模型性能与系统可持续性<sup>[7]</sup>。因此,催生了博弈论与联邦学习的深度交叉研究,其中Shapley值的贡献评估方法在联邦学习的贡献评估中被广泛应用<sup>[8]</sup>。

现有研究表明,Shapley值的线性公理假设在联邦学习场景中存在显著局限性<sup>[9]</sup>。当参与者数据呈现互补性或冲突性时,Shapley值难以准确量化非线性协同效应。文献[9]通过理论证明,在存在数据依赖关系的合作博弈中,Shapley值可能低估协同效果下的真实贡献,进而引发分配不公与联盟解体风险。

在此基础上,Shapley值在实际部署中暴露出以下两方面核心问题:一是必须枚举所有参与子集,计算复杂度随参与者数量呈指数增长,难以扩展至大规模系统;二是其分配结果在高度协同的场景下常出现失真,无法有效维持系统稳定性。这一问题在如医疗联合建模、跨机构金融风控等应用中尤为突出,急需一种能够同时兼顾公平性与稳定性的替代性价值评估机制。

对于Shapley值的问题,本文首次将合作博弈理论中的最小核心(Least core)解概念引入联邦学习的贡献评估与收益分配问题中。与Shapley值不同,Least core不依赖边际增益线性模型,可以更好地察觉参与方提供数据之间的协同效果,以通过最小化系统的最大不满度(excess)来构建一个更具鲁棒性与稳定性的分配框架,保证分配方案的稳定性,避免参与者因不满而退出联盟<sup>[10]</sup>。这种方法天然适应非线性协同结构,具备更强的公平性与激励兼容性。

本文与近期研究的区别在于以下两方面。

(1) 与GTG-Shapley等改进Shapley值的工作<sup>[11-12]</sup>不同,本文从根本上摒弃线性公理约束,更契合实际数据交互模式。

(2) Least core通过选择性抛弃部分约束条件,在计算效率与公平性间取得更优权衡。这些创新为联邦学习在隐私敏感场景中的落地提供了新的理论工具。

本文的贡献主要体现在以下3方面。

(1) 构建基于Least core的联邦学习贡献评估框架,从参与方协同效果的角度重新定义数据价值,增强系统联盟稳定性。

(2) 针对 Least core 计算复杂度高的问题, 设计了3种近似优化策略: 分层抽样机制根据联盟规模抽取代表性子联盟; 固定基准抽样方案引入参考联盟作为估值基准; 阈值跳过机制设置不满意度上限, 直接跳过低影响子集。这些技术协同作用, 在不显著影响分配精度的前提下, 使得整体计算效率显著提升, 能够在大规模参与者场景中实现可行的近似求解。

(3) 在多种联邦学习设置下(包括不同抽样比例、算法与参与者数量)的大量实验表明, Least core 分配方案相较传统 Shapley 值在公平性与系统稳定性方面更具优势。本文还系统考查了多种外部因素对 Least core 分配结果的影响, 分析其在不同联邦配置及其他协同计算框架中的适配性与可拓展性。

## 1 相关工作

在合作博弈领域, 大量文献研究了 Shapley 值、核心、Least core 和核仁等收益分配的方式, 并进行了相关的复杂性分析。

文献[13]从计算复杂性的视角出发, 以一个图论博弈为载体展开研究。在该博弈中, 参与者对应图的节点, 边的权重反映了合作产生的协同效应, 而联盟的总价值则由所包含边的权重之和决定。对于具有明显协同效应的情形, 核心的表征相对简单。类似的结果也适用于核仁、Least core 以及谈判集(bargaining set)。

文献[14]则将注意力聚焦于联盟形成过程中的核心计算问题, 提出了一种基于超可加性特征函数的简洁表示方法, 使得在给定分配结果的情况下能够高效验证该结果是否属于核心。尽管验证单个解是否在核心中是可行的, 但判断整个核心是否为空, 却被证明是非确定性多项式(non-deterministic polynomial, NP)完全问题, 其主要难点在于如何确定大联盟中各参与者之间的协作可能性。一旦协作可能性问题得到解决, 相应的

稳定性验证问题便可以在多项式时间内完成。

针对 Shapley 值、核心等解的计算复杂性问题, 文献[15]提出, 对于特定类型的博弈(如任务分配类博弈)可以通过简化特征函数显著减少计算复杂性, 同时分析了不同分配方案在不同类型博弈中的计算可行性。文献[16]提出了一种伪多项式时间算法, 用于计算加权投票博弈实例中的核仁, 这为解决部分特定博弈中的复杂性问题提供了新的思路。

在此基础上, 文献[17]建立了概率近似正确(probably approximately correct, PAC)可学习性(如网络流博弈、阈值任务博弈和诱导子图博弈等)与核心稳定性之间的联系, 证明了对于那些可高效学习的博弈, 通过多项式数量的样本就可以找到满足核心属性的近似分配方案。文献[18]则是在文献[17]的基础上, 给出了确保任何具有非空核心的收益分配函数在高概率下满足核心条件的精确样本复杂度界限。

近年来, 联邦学习中的参与者价值评估成为研究热点, 尤其在2024—2025年涌现出多个以 Shapley 值为核心改进路径的研究方法, 代表性工作包括引入类别敏感机制的 FedMS<sup>[19]</sup>方法、结合上置信边界(upper confidence bound, UCB)策略进行动态激励优化的 Shapley-UCB 方法<sup>[20]</sup>, 以及强调参与者协作结构的 FairFed<sup>[21]</sup>方法(基于合作 Shapley 值)。这些方法多聚焦于提升分配的公平性与对数据的适应能力, 同时通过采样、聚合压缩、注意力机制等手段提升 Shapley 值计算的效率。例如, IPSS(importance-based probabilistic stratified sampling)方法<sup>[22]</sup>基于分层抽样策略在一定程度上降低了传统 Shapley 值计算的复杂度。尽管这些方法在算法机制与应用场景上取得一定进展, 但多数仍然继承了 Shapley 值计算中的边际贡献假设, 其估值过程依赖于大量子联盟评估, 计算复杂度与分布波动性依然较高<sup>[23]</sup>。



文献[10]针对当前围绕 Shapley 值所形成的广泛共识,提出了一种更具可行性的替代方案-最小核心。为此,他们证明: Least core 具有良好的可计算性,即可以通过高效算法获得任意精度的近似解。然而,对于更精细的方案-核仁,则无法实现类似的计算保证。这是因为 Least core 只需要寻找一个最小“最大不满度”的稳定分配,而核仁则要求对所有联盟的不满度进行逐层排序和优化,以寻求唯一的最优分配序列。因此,核仁不仅需要更多的计算步骤,还涉及更复杂的约束与排序,使得其计算复杂度远高于 Least core 与 Shapley 值。

受上述研究启发,本文将合作博弈中的 Least core 方法引入联邦学习,作为收益分配的新方案,并验证了其有效性与广泛适用性。针对传统方法在计算复杂性上的挑战,本文提出一种基于 Least core 的分配机制,不再依赖边际增益序列,而是通过线性规划统一约束所有子联盟的不满值,有效缓解分配极化问题。结合抽样算法优化计算过程,在保持分配公平性与稳定性的同时,显著提升了大规模博弈中的计算效率。

## 2 预备知识

### 2.1 联邦学习

联邦学习是数据保留在本地的分布式机器学习框架,学习过程由多个参与方合作并经过多轮次训练完成<sup>[8]</sup>。联邦学习框架如图1所示,联邦学习由一个联邦服务器和多个参与方组成,每个参与方持有本地数据。假设当前训练轮次为  $t$  并有  $N$  个参与者,各参与方  $i \in N$  下载联邦服务器的全局模型  $M(t)$ ,然后基于各自的本地数据更新本地模型  $M_i^{t+1}$ ,并仅将参数梯度更新  $\Delta_i = M_i^{t+1} - M^t$  上传联邦服务器。服务器则根据数据量加权聚合(如 FedAvg),并广播更新后的全局模型

$M(t+1) = \text{FedAvg}(M(t), \{\Delta_i, i \in N\})$ ,以准备下一轮次训练。

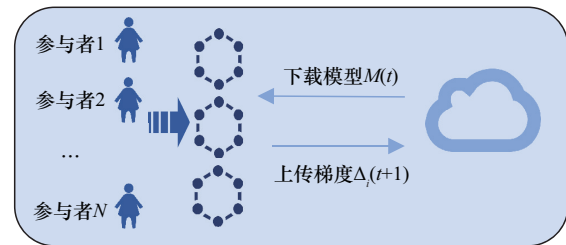


图1 联邦学习框架

### 2.2 Least core 原理

假设有一组参与者  $N$  和一个特征函数  $V(\cdot)$ ,每个参与者都有各自的私有数据集  $D_i, \forall i \in n$ 。对于任意子联盟  $S$  由任意个参与者的数据组成,即  $S \subseteq N$ ,特征函数  $V(S)$  表示通过测量联盟  $S$  中数据集上训练的模型性能来评估子联盟  $S$  的数据价值,  $V(\emptyset) = 0$ ,那么每个参与者单独训练所获数据价值为  $V(i)$ 。

在机器学习中,评估函数  $V(\cdot)$  基于子联盟数据  $S$  在单独测试集上学习得到的模型来预测性能:

$$V(S) = V(M_S) = V(A(M(0), D_S)) \quad (1)$$

其中,  $M_S$  为从头开始使用数据集  $D_S = \sum_{i \in S} D_i$  和算法  $A$  训练出来的子模型,  $M(0)$  表示训练的初始模型。

在子联盟  $S$  内,将特征函数确定的整体价值分配给各参与者的过程视为一个合作博弈中的核心分配问题。个体理性要求每个参与者获得的分配值  $X_i$  至少不低于其独立价值  $V(i)$ ,即  $X_i \geq V(i)$ ,确保其有动力留在联盟中。集体理性则要求联盟总收益被充分分配,即  $\sum_{i \in N} X_i = V(N)$ 。在同时满足个体理性和集体理性约束的前提下,所求得分配解被称为核心。

在合作博弈中,需要将  $V(S)$  在子联盟成员间进行合理分配。核心分配要求满足两个条件:个

体理性  $X_i \geq V(i)$ ，确保成员不退出；集体理性  $\sum_{i \in N} X_i = V(N)$ ，保证资源完全分配。然而，在实践中，核心可能不存在，即无法找到同时满足这两者的方案。

Least core 通过引入最大不满度  $e$  来近似核心：

$$e = \max_{S \subseteq N} [V(S) - \sum_{i \in S} X_i] \quad (2)$$

$e$  是所有子联盟  $S$  的总需求  $V(S)$  与实际分配总和  $\sum_{i \in S} X_i$  的差值中的最大值，即最大不满。

Least core 用  $e$  来衡量参与者的满意度，以近似核心的形式找到一种最优分配， $e$  越小意味着所有联盟的不满度越低，分配方案越接近核心。为了联邦学习的合作稳定，最小化  $e$  就是在所有可能的子联盟中寻找一个分配方案，使得最大不满度  $e$  尽可能小。Least core 将个体理性的分配目标演变成找到满足如下条件的分配方案：

$$\min e, \sum_{i \in S} X_i + e \geq V(S), \forall S \subseteq N \quad (3)$$

即使核心为空，Least core 仍可给出满足近似理性的稳定分配方案，在保障合作持续性的同时，减少成员间的利益冲突。

### 2.3 线性公理缺失对 Least core 的影响

上文分析了 Least core 在分配方案中的可行性，接下来讨论其与 Shapley 值的不同。

Least core 具备自己的个体理性和集体理性，相比之下，Shapley 值则基于 4 个公理：效率、公平性、零贡献和线性。这 4 个公理确保了在分配过程中，每个参与方的贡献能够被准确地识别和评估，而不依赖于最小化不满的策略。效率公理保证总收益完全分配；公平性公理确保相似贡献的参与方获得相等的收益；零贡献公理规定对合作没有贡献的玩家不会获得任何收益；线性公理则在多合作场景下提供了累加性质，从而保证了

在不同合作游戏之间的结果一致性。

尽管线性公理提供了可加性和简单性的优势，但在某些需要考虑参与方之间互动效果的博弈场景中，这种线性关系可能不适用。假设有两个参与方  $i$  和  $j$ ，在单独参加子联盟  $S$  时所带来的额外价值为  $V(S \cup \{i\}) - v(S)$  和  $V(S \cup \{j\}) - v(S)$ ，其中  $V(\cdot)$  为特征函数。当这两个参与方共同合作时，可能会产生更高的总价值，表现为： $V(S \cup \{i, j\}) > V(S \cup \{i\}) + V(S \cup \{j\})$ ，甚至可能双方产生冲突和替代，导致产生更低的总价值，即  $V(S \cup \{i, j\}) < V(S \cup \{i\}) + V(S \cup \{j\})$ 。在这种情况下，线性公理无法捕捉到这种协同效应所带来的价值变化，导致 Shapley 值未能准确反映真实的价值贡献。

核心和 Least core 算法则是将对每个参与方的价值评估转为整体联盟的价值评估。正因如此，相较于 Shapley 值，Least core 不追求严格的线性叠加。

### 2.4 复杂度分析

Least core 算法在合作博弈中的时间复杂度需要综合考虑问题设定、输入参数及算法步骤，以下从联邦学习环境进行分析。

算法输入主要包括参与者数量  $n$  及评估各子联盟收益的特征函数  $V(\cdot)$ 。在联邦学习中，数据共享机制与联盟组合数量是复杂度的关键因素，其核心步骤包括：为所有可能的子联盟  $S \in N$  聚合其数据并计算  $V(S)$ ；据此构建不满值约束；最终通过线性规划求解最小  $e$ ，获得最优分配方案。

算法的主要计算开销来自对子联盟收益  $V(S)$  的评估，由于合作博弈中的子联盟数量为  $2^n - 1$ ，最坏情况下需要进行指数级计算，与 Shapley 值相当，即  $O(2^n)$ 。这种增长趋势限制了算法在大规模场景中的可扩展性。

尽管 Least core 理论上可提供公平且稳定的



分配方案,但随着参与者与数据规模的增加,其复杂度迅速上升。因此,实际应用中需要结合复杂度优化或近似算法,以提升其实用性与可扩展性。

### 3 Least core 及其近似计算

#### 3.1 引入 Least core

在联邦学习中,引入 Least core 旨在提升收益分配的公平性与系统稳定性,避免高价值参与者因回报不公而退出系统。与基于边际贡献的 Shapley 值不同, Least core 通过最小化最大不满度  $e$ ,在保障个体理性与集体理性的同时,优化整体满意度,即使核心为空,也能通过线性规划得到近似稳定解,从而有效维护系统合作。该机制更适用于存在非线性协同或冲突的数据结构,具备良好的激励兼容性,能降低分配失衡导致的节点流失风险,为构建长期稳定的联邦系统提供支持。

Least core 算法模型如图 2 所示。图 2 展示了 Least core 算法在联邦学习训练流程中的嵌入方式。每轮训练开始时,所有参与者  $i \in N$  下载当前

全局模型  $M_t$ ,并基于本地数据集  $D_i$  执行本地训练,生成更新后对应的梯度  $\Delta_{M_t^i}$ ,随后将其上传至联邦服务器。

服务器针对所有可能的子联盟  $S \subseteq N$  构建子模型  $M_S^t$ ,其梯度更新通过加权平均得到:

$$\Delta_{M_S^t} = \sum_{i \in S} \frac{|D_i|}{\sum_{j \in S} |D_j|} \Delta_{M_t^i} \quad (4)$$

其中,  $D_i$  表示参与者  $i$  拥有的本地数据数量,  $\sum_{j \in S} |D_j|$  表示子联盟  $S$  中所有参与者数据集样本数

量的总和。接着,使用子模型  $M_S^t$  在验证集上评估其性能,获得特征函数值  $V(S) = V(M_S^t)$ ,即子联盟  $S$  的整体价值。将所有  $V(S)$  构成线性规划模型的约束条件,目标是求解最小  $e$ ,使得对于任意  $S \subseteq N$ ,满足:

$$\sum_{i \in S} x_i \geq V(S) - e \text{ 且 } \sum_{i \in N} x_i = V(N) \quad (5)$$

其中,  $x_i$  表示单个参与者的收益分配权重,  $e$  为最大不满度。通过求解该优化问题,本文可以获得一组满足联盟稳定性与公平性的 Least core 解,最终将分配结果  $x_i$  与更新后的全局模型  $M_{t+1}$

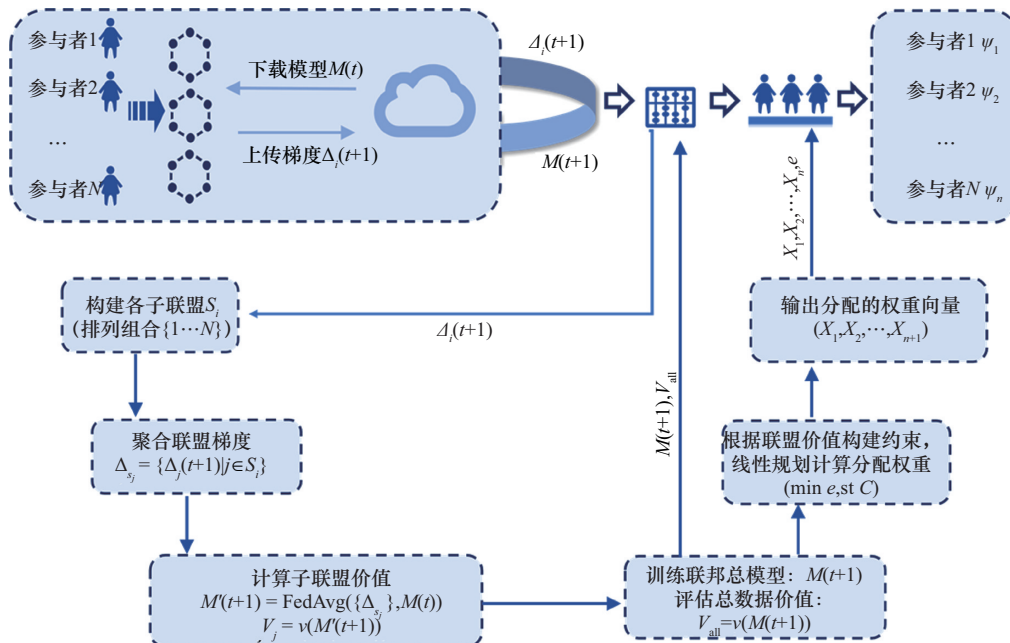


图2 Least core 算法模型

分发给各个参与方。引入 Least core 作为评价机制的联邦学习如算法 1 所示。

**算法 1** 引入 Least core 作为评价机制的联邦学习

**输入** 初始联邦模型  $M(0)$ , 性能评估函数  $V(\cdot)$ , 参与者数量  $N$ , 训练总轮数  $R$

**输出** 最小  $e$  和对应的 Least core 分配  $\psi_i$ , 其中  $i \in \{1, \dots, k\}$

- 步骤 1** 初始化  $\psi_i = 0, i \in \{1, \dots, n\}$
- 步骤 2** 定义约束集  $C = \emptyset$
- 步骤 3** 初始化  $e = \infty$
- 步骤 4** for  $t = 0, \dots, R - 1$ :
- 步骤 5**  $S = \{S_i | S_i \subseteq N, S_i \neq \emptyset\} \quad i = 1, 2, \dots, K$ ;  
# $K$  为子联盟序号,  $S$  为所有联盟组合
- 步骤 6** for  $i \in N$ : #每一个参与者  $i$  客户端
- 步骤 7**  $\Delta_{M_i} \leftarrow \text{ClientUpdate}(i, M_t)$ ;
- 步骤 8** end for
- 步骤 9**  $M_{t+1} \leftarrow \text{FedAvg}\left(\left\{\Delta_{M_i} | i \in N\right\}, M(t)\right)$ ;
- 步骤 10**  $V_{\text{all}} \leftarrow V(M_{t+1})$ ;
- 步骤 11**  $C \leftarrow C \cup \{x_0 + x_1 + \dots + x_n = V_{\text{all}}\}$ ;
- 步骤 12** for  $j \in K$
- 步骤 13**  $\Delta_{M'_j} \leftarrow \{\Delta_j(t+1) | S_j \in S\}$ ;
- 步骤 14**  $M'_{t+1} \leftarrow \text{FedAvg}\left(\left\{\Delta_{S_j}\right\}, M_t\right)$ ;
- 步骤 15**  $V_j \leftarrow V(M'_{t+1})$ ;
- 步骤 16**  $C \leftarrow C \cup \{\dots + x_i + \dots + x_n + e \geq V_j, i \in S\}$ ;
- 步骤 17** end for
- 步骤 18**  $[\psi_1, \dots, \psi_n, e] \leftarrow \text{solver}(\text{mine}, \text{st } C)$ ;
- 步骤 19** end for
- 步骤 20** return  $\psi_1, \dots, \psi_n$  和  $e$

### 3.2 抽样优化策略以及复杂度分析

为解决 Least core 在大规模场景下的计算瓶颈, 本文提出 3 种近似优化策略: 分层抽样、固定基准

抽样和系统抽样。它们通过有代表性地选择子联盟, 减少约束数量和特征函数评估开销, 有效提升分配效率, 同时保持结果稳定性与准确性。

#### 3.2.1 分层抽样

分层抽样将子联盟按规模划分为多个层级, 在每层中按设定比例随机抽取子联盟参与特征函数评估与约束构建。该方法既保证了不同联盟规模的均衡覆盖, 也有效控制了计算复杂度。在包含  $n$  个参与者的系统中, 可分为  $k = n - 1$  个层, 每层包含  $\binom{n}{k}$  联盟。每层抽样比例为  $\{r_i, r_i \in [0, 1]\}$ , 抽样后复杂度降低至  $\sum_{i=1}^k r_i \cdot \binom{n}{i}$ 。

分层抽样方法可以调整各层级的抽样比例  $r_i$ , 从而在精度与计算效率之间找到平衡点。这种方法的合理性主要体现在分层原则与比例分配两个方面。根据子联盟规模  $k$  进行分层, 可以确保不同规模的子联盟得到均衡考虑, 避免对小规模或大规模子联盟产生偏倚, 从而在保障样本覆盖全面性与代表性的同时, 更精准地捕捉参与者之间的协同效应。这种分层设计能够有效反映各层级子联盟的特征, 确保抽样结果的科学性与合理性, 为后续的特征函数评估与约束条件构建提供坚实的数据基础。引用分层抽样优化算法如算法 2 所示。

**算法 2** 引用分层抽样优化算法

**输入** 参与者集合  $N$ , 每层比例  $r_i$ , 训练总轮数  $R$

**输出** 最小  $e$  和对应的 Least core 分配  $\psi_i$ , 其中  $i \in \{1, \dots, k\}$

- 步骤 1** 初始化  $\psi_i = 0, i \in \{1, \dots, n\}$
- 步骤 2** for  $t = 0, \dots, R - 1$ :
- 步骤 3**  $S \leftarrow \{S_i | S_i \subseteq N, S_i \neq \emptyset\} \quad i = 1, 2, \dots, K$ ;
- 步骤 4**  $S\_num \leftarrow \text{len}(S)$  #计算总抽样数
- 步骤 5** for  $i$  in range(1,  $N - 1$ ): #遍历每个子联盟规模



步骤6  $S_i \leftarrow S_i \cup \{S_j \text{ for } S_j \text{ in } S \text{ if } \text{len}(S_j) = i\}$   
 步骤7  $S_r \leftarrow S_r \cup \{\text{random.sample}(S_i, S\_num * r_i)\}$   
 步骤8 end for  
 步骤9 #使用抽样后的集合  $S_r$  进行训练和价值评估 (算法1: 步骤6~步骤18)  
 步骤10 end for  
 步骤11 return  $\psi_1, \dots, \psi_n$  和  $e$

### 3.2.2 固定基准抽样

固定基准抽样通过优先选择一组具有代表性的关键子联盟作为基准子联盟, 减少了非必要子联盟的评估与约束构建, 进而优化了计算复杂度和资源消耗。基准子联盟主要包括单个参与者的子联盟  $\binom{n}{1} = n$  (用于衡量个体贡献) 和两两组合的子联盟  $\binom{n}{2} = \frac{n(n-1)}{2}$  (用于捕捉基本的合作关系)。在包含  $n$  个参与者的场景下, 基准子联盟的数量约为  $n^2 + n$ 。

接着, 确定除基准外的抽样数量  $m = r \cdot (2^n - n^2 - n)$ ,  $r$  为总抽样比例  $r \in [0, 1]$ ,  $m$  为需要抽取的样本子联盟数, 采用随机抽样的方式进行补充。这一策略使得特征函数的计算复杂度降低至  $n^2 + n + m$ 。

固定基准抽样优先选择具有代表性的关键子联盟, 确保了特征函数评估的准确性和公平性。这些基准子联盟主要包括个体子联盟和两两组合子联盟, 能够初步反映个体贡献及其简单协作关系。在此基础上, 非基准子联盟通过蒙特卡洛随机抽样的方式进行补充, 确保了整体子联盟集合的多样性和全局性。引用固定基准抽样优化算法如算法3所示。

**算法3** 引用固定基准抽样优化算法

**输入** 参与者集合  $N$ , 抽样比例  $r$ , 训练总轮数  $R$ , 基准联盟最大规模  $m$

**输出** 最小  $e$  和对应的 Least core 分配  $\psi_i$ , 其中  $i \in \{1, \dots, k\}$

步骤1 初始化  $\psi_i = 0, i \in \{1, \dots, n\}$   
 步骤2 for  $t = 0, \dots, R-1$ :  
 步骤3  $S = \{S_i \mid S_i \subseteq N, S_i \neq \emptyset\} \quad i = 1, 2, \dots, K$ ;  
 步骤4  $S\_num \leftarrow \text{len}(S) * r$  #计算总抽样数  
 步骤5 for  $i$  in range(1,  $m$ ): #选取基准联盟  
 步骤6  $S\_back \leftarrow S\_back \cup \{S_j \text{ for } S_j \text{ in } S \text{ if } \text{len}(S_j) = i\}$   
 步骤7 end for  
 步骤8  $S_r \leftarrow S_r \cup \{\text{random.sample}(S\_back, S\_num - \text{len}(S\_back))\}$  #随机抽取剩余子联盟  
 步骤9 #使用  $S_r$  进行训练和评估 (算法1: 步骤6~步骤18)  
 步骤10 end for  
 步骤11 return  $\psi_1, \dots, \psi_n$  和  $e$

### 3.2.3 系统抽样

在上述两个策略中, 本文观察到, 在 Least core 的计算中会有部分相连或相近的子联盟作为约束条件, 对最终分配结果的影响较小, 甚至可以忽略其影响。这些子联盟可以被视为冗余联盟。由于这些冗余联盟对结果的贡献有限, 对它们的价值评估可以简化或省略。同时, 受到 GTG-Shapley<sup>[11]</sup> 中对于边际效应采用截断方式的启发, 本文提出了加入阈值判断的系统抽样。

具体而言, 首先对子联盟按照字典序进行排序。在对子联盟依次进行特征函数评估和分配结果计算时, 引入一个阈值  $\epsilon$ , 用于衡量当前子联盟与前一个子联盟分配结果之间的差异。当加入当前子联盟  $S_j$  后的分配结果与前一个子联盟  $S_{j-1}$  的分配结果之间的欧几里得距离小于阈值  $\epsilon$ , 即满足以下条件:

$$\|\mathbf{x}(S_j) - \mathbf{x}(S_{j-1})\|_2 < \epsilon \quad (6)$$

其中,  $\mathbf{x}(S_j)$  和  $\mathbf{x}(S_{j-1})$  分别表示当前子联盟和前一个子联盟的分配权重向量,  $\|\cdot\|_2$  表示欧几里得距离。此时认为当前子联盟对最终结果的影响较

小,从而跳过后续 $m$ 个(步长)子联盟的特征函数计算和约束条件构建,直接进入下一个有效子联盟的评估。通过这种方式,系统抽样能够自适应地剔除对结果贡献较小的子联盟,从而显著减少不必要的计算。

原始复杂度为 $2^N$ ,引入“阈值跳过”策略后,假设平均每跳过一次可以省下 $n$ 次评估,而跳过触发率为 $p(\epsilon)$ ,则有效评估次数变为: $(1-p(\epsilon))\cdot(2^N-1)+p(\epsilon)\cdot\frac{2^N-1}{n+1}$ ,其中, $\frac{1}{n+1}$ 近似表示:每触发一次跳过,实际访问1次,跳过 $n$ 次(共 $n+1$ 个子联盟) $\rightarrow$ 触发比约为 $\frac{1}{n+1}$ 。所以系统抽样的期望复杂度为: $\left(1-p(\epsilon)+\frac{p(\epsilon)}{n+1}\right)\cdot 2^N$ 。

引用系统抽样优化算法如算法4所示。

**算法4** 引用系统抽样优化算法

**输入** 参与者集合 $N$ ,子联盟集合 $S$ ,截断阈值 $\epsilon$ ,训练总轮数 $R$ ,步长 $n$

**输出** 最小 $e$ 和对应的Least core分配 $\psi_i$ ,其中 $i\in\{1,\dots,k\}$ 。

- 步骤1 初始化  $\psi_i=0, i\in\{1,\dots,n\}$ ;
- 步骤2 定义约束集  $C=\emptyset$ ;
- 步骤3 初始化  $e=\infty$ ;
- 步骤4 初始化  $x_2(S)=[]$ #记录子联盟的分配结果
- 步骤5 for  $t=0,\dots,R-1$ :
- 步骤6 #子联盟集合 $S$ 、总联盟训练和价值评估(算法1:步骤5~步骤11)
- 步骤7 for  $j\in K$
- 步骤8  $\Delta_{S_j}\leftarrow\{\Delta_j(t+1)|S_j\in S\}$ ;
- 步骤9  $M'_{t+1}\leftarrow\text{FedAvg}(\{\Delta_{S_j}\},M_t)$ ;
- 步骤10  $V_j\leftarrow V(M'_{t+1})$ ;
- 步骤11  $C\leftarrow C\cup\{\dots+x_i+\dots+x_n+e\geq V_j, i\in S\}$ ;
- 步骤12  $[x_0,x_1,\dots,x_n,e]\leftarrow\text{solver}(\min e, \text{st } C)$
- 步骤13  $x_1(S)\leftarrow[x_0,x_1,\dots,x_n,e]$

步骤14 if  $\|x_1(S)-x_2(S)\|_2<\epsilon$ :

步骤15  $j+=n$ ;

步骤16 end if

步骤17  $x_2(S)\leftarrow x_1(S)$

步骤18 end for

步骤19  $[\psi_1,\dots,\psi_n,e]=x_1(S)$

步骤20 end for

步骤21 return  $\psi_1,\dots,\psi_n$ 和 $e$

## 4 实验结果和分析

### 4.1 实验设置

#### 4.1.1 数据集和分布场景

本次实验使用了MNIST数据集,该数据集是一个手写数字的图像数据集,包含了从0到9的数字,每个样本都是 $28\times 28$ 像素的灰度图像。数据集中的图像已经经过标准化处理,数字位于图像的中心位置,方便模型识别。每张图像都附有其对应的标签,用于指示图像中代表的数字类别。整个数据集由60 000个训练样本和10 000个测试样本组成。在数据预处理阶段,本文从MNIST数据集的每个数字类别中随机选择5 421个样本,最终形成包含54 210个样本的训练数据集。同时,从每个类别中随机抽取892个样本,构建包含8 920个样本的测试数据集。在数据处理时,本文将图像像素值归一化到 $[0,1]$ 区间,以提高模型训练的稳定性。

参与者对图像按区域切割,获得不同局部特征,模拟数据价值差异。在6参与方设置中,图像划分为6块,中心区域数据价值较高;10参与方场景则将图像均匀切为上、下各5块,分配给各方。该切分策略可以在保证数据隐私的同时,提供非均衡特征分布,以测试模型的协同能力。

#### 4.1.2 不同设置对Least core解的影响因素分析实验设置

在Least core分配机制中,子联盟 $S$ 的性能评



估函数  $V(S)$  构成了线性规划约束的核心部分，其定义与计算方式对最终分配结果具有决定性影响。因此，合理设计评估模型并明确  $V(S)$  的设定方式，是确保分配机制稳定性与公平性的关键。为系统考查不同建模与求解设置对 Least core 分配结果的影响，本文设计了3组对比实验，分别针对评估模型、求解器类型以及子联盟入场顺序进行分析。实验均基于统一的数据集构建，包含5个参与者，定义了各参与者在不同子联盟中的价值函数  $V(S)$ ，以评估不同建模方式对最终分配结构的影响。

在评估模型方面，本文选取了5种具有代表性的分类模型，涵盖线性与非线性两类结构，具体如下。

(1) **Logistic Regression**: 典型的线性模型，具备良好的凸优化性质，训练过程稳定。

(2) **SGD Classifier**: 基于随机梯度下降的在线学习方法，在大规模数据下具有较高训练效率。

(3) **Logistic RegressionCV**: 在逻辑回归基础上引入交叉验证机制，提升了模型的泛化能力与鲁棒性。

(4) **MLP Classifier**: 浅层前馈神经网络，能够捕捉更复杂的非线性特征交互。

(5) **Random Forest Classifier**: 集成式决策树模型，对特征之间的协同效应尤为敏感，适用于表达多变量间的非线性关系。

在每个子联盟  $S$  上，本文使用对应模型进行训练，并以测试准确率作为其价值  $V(S)$ ；整体联盟价值  $V(N)$  则由全集训练所得的准确率确定。尽管可选用曲线下面积 (area under curve, AUC)、F1-score 等指标，但这类指标往往不满足 Least core 所需的线性可加性与理性分配假设<sup>[10]</sup>，可能导致线性规划模型不收敛或解的经济含义丧失。因此，在本实验设定中，准确率是唯一既满足建

模假设又具实际解释性的评估函数。

在模型训练并确定  $V(S)$  后，通过线性规划求解 Least core 问题。为比较不同求解工具在精度与效率上的表现，本文选用 SciPy 的 Linprog、商业优化器 Gurobi 和基于 Cvxpy 的方案，统一输入与约束设定，确保变量仅为求解器差异。

此外，为评估子联盟入场顺序对结果的影响，设置自然顺序 (1→5)、逆序 (5→1) 和打乱顺序 (3,4,2,5,1) 这3种典型排列，分析其对最大不满度与分配方差的影响。在保持其他实验条件不变的前提下，确保结果差异源自单一变量变化。

#### 4.1.3 抽样设置

分层抽样策略通过为不同规模的子联盟分配特定比例，实现了对抽样过程的精细化控制。该方法根据预设比例，从每个子联盟类别中随机抽取样本，直至所有类别均按比例完成抽样，并确保最终的总样本数量满足实验要求。这种策略的优势在于能够有效平衡组合的多样性与重要性，不同规模的子联盟均有被选中的机会，而预设比例的设计则反映了各类别在实验中的权重与价值。分层抽样特别适用于需要在子联盟种类覆盖与计算效率之间寻求平衡的场景，同时能够较为真实地模拟实际中参与方数据贡献的不均衡性。

本次实验首先确定总体的抽样比例，按比例从大到小进行实验。由于分层抽样是以成员数量对子联盟进行划分，不同层的子联盟数量分布存在显著差异。具体而言，由于子联盟数量在不同规模层次分布不均，中间层（如3人联盟）数量最多，边缘层（如1人或5人联盟）数量最少，因此，按如下比例分配抽样权重：1人和5人联盟各采样10%，2人和4人联盟各采样24%，3人联盟采样32%。10个参与者情形亦采用同样的分布策略。该设置一方面保证了对关键子联盟的覆盖，另一方面保留了原有联盟规模分布，使抽样

近似计算结果更接近于全量计算的性能表现。

固定基准抽样策略优先覆盖小规模组合，确保基础信息的完整性，并从大规模组合中随机补充剩余样本，具体做法：先固定抽取所有 1 成员和 2 成员的组合，以捕捉个体与两两协同效应，再从 3 成员及以上的组合中随机选取，直至达到预设样本量。

与分层抽样策略类似，固定基准抽样策略也基于总体抽样比例进行控制，但其重点更侧重于基础子联盟的全面覆盖。相比之下，分层抽样更注重不同子联盟规模的比例分布，因此两种策略在设计初衷和应用场景上各具特色。通过实验，本文可以更全面地评估不同抽样策略对实验结果和模型性能的影响。

系统抽样通过设定阈值，利用欧几里得距离判断新增子联盟对分配结果的影响，以决定是否继续抽样，具体做法：当一组子联盟加入后，计算其前后分配结果的欧几里得距离，若低于预设阈值，则认为其影响较小，可跳过该组后续子联盟的计算。通过阈值设定，系统可精细控制抽样过程，评估跳过比例对结果的影响，从而优化策略、提升效率、去除冗余子联盟。由于分配基于模型识别准确率（范围为[0,1]），新增联盟引入的影响也不会远超该区间。实验通过在[0,1]中分段取值，得出在[0,0.1]范围内对子联盟数量控制效果最为明显。

#### 4.1.4 性能评估标准

比较方法的性能通过以下指标进行评估。

(1) 时间：计算总时间来评估每个方法的计算效率。

(2) 均方误差（mean-square error, MSE）的定义为：

$$MSE = \frac{\sum_{i=0}^N (\Phi_i^* - \Phi_i)^2}{N} \quad (7)$$

本文通过计算与基线的 MSE 判断所计算出的

近似值的准确程度，其中， $\Phi_i^*$  为基线值， $\Phi_i$  为抽样估算值。

## 4.2 实验结果分析

### 4.2.1 不同设置对 Least core 解的影响因素分析

不同模型对分配结果的影响见表 1。由表 1 可知，评估模型的选择对 Least core 分配结果有显著影响。线性模型（如 Logistic Regression、Logistic RegressionCV）在各子联盟下的性能评估值  $V(S)$  更加平稳，所产生的分配方差较低，分配结构较为平均，体现出良好的公平性，但其最大不满意度  $e$  相对较高（0.244 75），表明在满足所有联盟稳定性方面仍需要容忍一定不满意度。

表 1 不同模型对分配结果的影响

模型类型	分配值范围	分配方差	最大不满意度 $e$	运行时间/s
Logistic Regression	0.279~0.568	0.014 7	0.244 75	9 394.51
SGD Classifier	0.234~0.545	0.018 5	0.287 10	3 913.31
Logistic RegressionCV	0.283~0.584	0.014 9	0.244 75	7 632.96
MLP Classifier	0.440~0.813	0.030 7	0.052 33	5 608.28
Random Forest Classifier	0.458~0.840	0.032 0	0.033 29	3 878.71

相比之下，非线性模型（如 MLP Classifier 和 Random Forest Classifier）对数据特征协同关系的表达能力更强，Least core 能够识别出协同影响力高的参与者。在相同数据下，其最后训练总体价值明显高于线性模型。这类模型也显著压缩了最大不满意度（MLP Classifier: 0.052, Random Forest Classifier: 0.033），表明其更有可能找到联盟满意度更高的最优解。然而，较大的分配方差显示分配结果更易受评估值波动影响，可能在部分联盟中产生偏差。此外，虽然 Logistic Regression 运行时间最长（9 394.51 s），但其在后续关于抽样策略加速效果的实验中正好可作为对照，凸显优化策略的效益。

不同求解器对于分配结果的影响见表 2。由表 2 可知，3 种求解器均给出了近似的最大不满意度，表明问题最优性未受影响，但在参与者收益分配方面仍存在轻微波动。这种差异主要源自不



同求解器内部数值精度控制、迭代策略及容差设置。在运行效率上，三者只有略微的差距，Gurobi与Cvxpy两个商业求解器略优于Linprog，求解器的不同对于最后的运行效率没有太大的影响。从整体来看，不同求解器在保持全局最优性前提下，解的具体结构存在微小偏差，但不会影响Least core的稳定性判定。

表2 不同求解器对于分配结果的影响

求解器类型	分配值范围	分配方差	最大不满度 $e$	运行时间/s
Linprog	0.279~0.568	0.014 7	0.244 75	9 394.51
Gurobi	0.237~0.604	0.013 7	0.237 50	8 943.52
Cvxpy	0.241~0.590	0.014 2	0.241 66	9 036.66

为保持实验的一致性与对比可复现性，本文后续实验（包括与Shapley值的精度对比、3种抽样策略的效率评估）统一采用Linprog作为Least core求解器，其较高的运行时间开销也更能体现抽样策略在复杂场景下的加速效果，从而为策略优化带来更明确的实验量化依据。

不同子联盟入场顺序对于分配结果的影响见表3。由表3可知，尽管分配方差基本一致，不同入场顺序对最大不满度 $e$ 仍产生了微小影响。例如，倒序入场(5,4,3,2,1)方案使最大不满度降至0.227 10，表明协同影响强的参与者优先加入可轻微提升稳定性。该结果提示子联盟构造顺序在Least core求解中虽非主要因素，但在精细调优与大规模场景中可作为优化变量之一。

表3 不同子联盟入场顺序对于分配结果的影响

联盟按规模入场	分配值范围	分配方差	最大不满度 $e$	运行时间/s
(1,2,3,4,5)	0.279~0.568	0.014 7	0.244 75	9 394.51
(5,4,3,2,1)	0.281~0.581	0.014 7	0.227 10	9 283.93
(3,4,2,5,1)	0.278~0.597	0.014 5	0.240 10	9 368.19

综合3组实验结果，评估模型对Least core分配结果影响最大：线性模型分配稳定，非线性模型则更能压缩最大不满度、提升联盟满意度，体

现公平性与稳定性的权衡。求解器类型影响较小，各主流工具在保持全局最优性的前提下，仅在效率和分配细节上略有差异，可按资源灵活选择。子联盟入场顺序虽不显著影响总体稳定性，但在优化最大不满度方面具有一定潜力。综上，这些设置不仅影响解的数值特征，也决定求解效率与可解释性。后续实验将统一采用稳定配置，进一步探索基于抽样与策略设计的效率提升路径。

#### 4.2.2 Least core与Shapley值分配结果对比与分析

精准计算的Least core与Shapley值的分配结果如图3所示。目前，Shapley值作为衡量参与方边际贡献的经典方法，其分配方案已经被广泛认可并已被证明其分配的合理性和有效性<sup>[5-7]</sup>。基于这一前提，通过图3所示实验结果可以看出，在参与方的价值分配上，Shapley值和Least core对于数据价值高的参与者（参与者2和参与者5）均给出较高的价值分配权重，这一结果表明，两种方法在识别和量化关键贡献者方面具有基本一致性，验证了在联邦学习中运用Least core的分配机制在评估数据价值上的有效性。

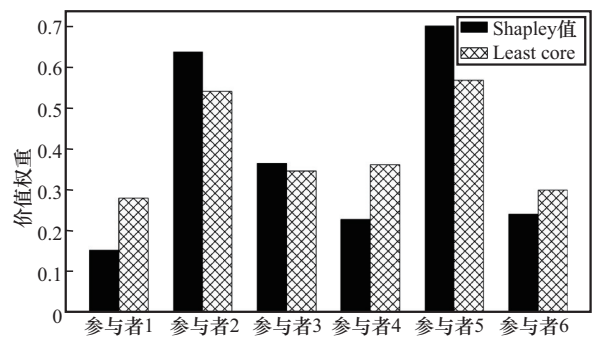


图3 精准计算的Least core与Shapley值的分配结果

精准计算的Least core与Shapley值的箱线图如图4所示。由图4可知，其分配结果呈现出更为集中和稳定的特性，波动性明显降低。这说明Least core在确保各参与方贡献得到合理反映的同时，更能维护联盟的公平性与长期稳定性，从而在实际应用中展现出较Shapley值方法更为突出

的优势。

两种方法的耗时对比见表 4。从表 4 可以看出, 在不同用户规模下, Least core 方法在计算耗时方面整体略优于 Shapley 值方法。需要指出的是, 虽然两种方法在理论上都具有指数级计算复杂度: Shapley 值需要枚举所有可能的用户排列, 而 Least core 方法在最坏情况下需要处理指数级数量的联盟约束, 但在实际实现中, Least core 方法可以借助线性规划求解技术及约束生成策略, 使得它在本实验设定下展现出略优的计算表现。

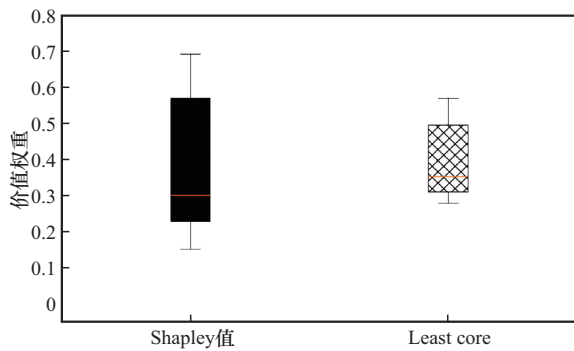


图4 精准计算的 Least core 与 Shapley 值的箱线图

表4 两种方法的耗时对比

算法	计算耗时/min	
	6个用户	10个用户
Least core	156.58	1 679.18
Shapley 值	162.21	1 858.43

#### 4.2.3 分层抽样分析

结合分层抽样的实验结果见表 5。从时间上

看, 相较于准确计算的 Least core 算法, 分层抽样算法对于时间的改善十分明显, 在 6 个用户和 10 个用户的情况下, 都表现出了较好的优化效果。从准确度角度看, 即使抽样比例降低至 25%, 抽样后估算的 Least core 结果与准确计算的结果相比较, 误差也仅为 0.557% 与 0.076%, 这里的误差是通过比较每个参与者的抽样分配值与精确分配值的平方差得到的, 相对误差越低, 说明抽样近似方法在保证计算效率的同时能够保持较高的分配精度。在 10 个用户的情况下, 其 MSE 远低于 6 个用户的场景。这可能是由于用户数量越多, 数据协同效应越明显, 抽样后模型对总体结果的拟合能力更强。这使得该方法在多用户联邦学习场景中更具应用优势, 如在医疗、金融等对数据隐私和计算效率有高要求的场景。

#### 4.2.4 固定基准抽样分析

结合固定基准抽样的结果见表 6。与分层抽样类似, 样本数量的减少能够有效降低分配时间。然而, 与分层抽样不同, 固定基准抽样更倾向于确保基础子联盟 (单个参与者和两两组合) 被优先选择。这种策略使得固定基准抽样在用户数量较少的情况下更具优势, 这是因为对于小规模用户场景, 两两之间的协同效应是主要特征。固定基准抽样能够更精确地捕捉这些协同效应, 因而误差要低于分层抽样。在用户数量较多的情况下, 固定基准抽样的误差开始超过分层抽样。

表5 结合分层抽样的实验结果

抽样比例	6个用户				10个用户			
	子联盟数量	MSE/10 <sup>-3</sup>	时间/min	最大不满度 <i>e</i>	子联盟数量	MSE/10 <sup>-3</sup>	时间/min	最大不满度 <i>e</i>
100%	62	*	156.58	0.244 8	1 022	*	1 679.18	1.062 4
90%	54	1.24	139.91	0.240 9	920	0.349 9	1 434.20	1.043 8
80%	49	2.57	132.13	0.238 6	818	0.083 5	1 203.07	1.053 8
70%	43	2.79	129.64	0.210 8	715	0.366 5	1 186.21	1.055 6
60%	37	3.39	114.47	0.209 5	613	0.458 8	980.97	1.042 5
50%	31	4.18	99.76	0.193 5	511	0.565 2	712.22	1.051 5
25%	15	5.57	44.81	0.161 2	255	0.764 3	421.33	1.031 0

注: 100% 抽样比例对应全量子联盟计算(精确计算), 无误差, 因此标记为“\*”。



表6 结合固定基准抽样的实验结果

抽样比例	6个用户				10个用户			
	子联盟数量	MSE/10 <sup>-3</sup>	时间/min	最大不满度 $e$	子联盟数量	MSE/10 <sup>-3</sup>	时间/min	最大不满度 $e$
100%	62	*	156.58	0.244 9	1 022	*	1 679.18	1.062 4
90%	54	1.04	142.23	0.237 8	920	0.730 6	1 423.65	1.087 8
80%	49	2.31	139.46	0.233 0	818	2.153 9	1 278.09	1.110 7
70%	43	2.47	121.48	0.225 1	715	2.898 5	1 193.08	1.092 8
60%	37	3.04	111.11	0.199 7	613	3.167 5	1 017.56	1.052 9
50%	31	4.16	89.99	0.194 7	511	4.367 9	678.89	1.098 7
25%	*	*	*	*	255	4.890 0	340.03	1.106 9

注:1.100% 抽样比例对应全量子联盟计算(精确计算),无误差,因此标记为“\*”。

2.在6个用户的场景下,由于25%抽样比例对应的抽样数量小于基准子联盟数量,因此无法进行抽样计算,故标记为“\*”。

因此,固定基准抽样更适用于用户数量较少且基础协同效应显著的场景,如小规模物联网数据协作或初期的联邦学习部署。

#### 4.2.5 系统抽样分析

结合设定阈值跳过的系统抽样方法的实验结果见表7。由表7可知,引入阈值跳过的改进系统抽样方法能够显著提升计算效率,并在一定程度上保持结果的准确性和均衡性。在样本数量和计算时间方面,随着阈值的增加,计算开销显著降低。这说明,通过阈值判断跳过冗余子联盟的特征函数计算,系统抽样能够有效减少不必要的计算任务,从而大幅优化性能。在MSE方面,随着阈值的提升,误差略有上升,但总体水平较低,表明分配结果的准确性受影响较小。

然而,系统抽样的主要缺点在于其对阈值设置和步长的敏感性。阈值过高和步长设置过大的

不合理可能导致跳过过多子联盟,影响最终分配的准确性和公平性;阈值过低和步长设置过低则削弱优化效果。此外,由于系统抽样动态调整样本数量,其覆盖范围可能不如其他方法均衡。因此,系统抽样特别适用于计算资源有限或需要实时处理的场景,如大规模分布式计算和实时数据处理环境。在这些场景下,通过合理设置阈值和步长,可以在计算效率与结果精度之间实现平衡,提供高效且灵活的收益分配解决方案。

#### 4.3 实验总结

上述实验结果表明,Least core方法能够通过合理分配参与者收益,实现联盟的稳定性和公平性。在结合优化抽样策略的实验中,分层抽样、固定基准抽样以及系统抽样方法各展其长,适用于不同的场景需求。

分层抽样能够全面覆盖不同规模的子联盟,

表7 结合设定阈值跳过的系统抽样方法的实验结果

判定阈值	6个用户				10个用户			
	子联盟数量	MSE/10 <sup>-3</sup>	时间/min	最大不满度 $e$	子联盟数量	MSE/10 <sup>-3</sup>	时间/min	最大不满度 $e$
0	62	*	156.58	0.244 8	1 022	*	1 679.18	1.062 4
0.01	44	1.13	122.07	0.217 1	738	0.229 0	1 117.78	1.030 7
0.03	39	1.19	117.50	0.219 2	610	0.166 3	983.51	1.042 0
0.05	37	1.23	103.55	0.210 4	577	0.109 4	839.37	1.046 7
0.07	31	1.30	80.17	0.197 8	496	0.300 0	732.85	1.033 2
0.09	27	1.46	62.18	0.188 7	416	0.342 8	569.81	1.049 1
0.1	26	1.49	60.73	0.186 6	411	0.575 7	530.50	1.025 9

特别适用于用户较多的场景，有助于捕捉各层级的协同效应，兼顾精度与效率，适合如医疗、金融等对隐私与准确性要求较高的领域。固定基准抽样重点覆盖基础子联盟（如单个或两两组合），在用户较少或数据分布均匀的场景中效果更佳，如物联网小规模协作或在线广告推荐。系统抽样则通过设定阈值动态优化计算过程，在大规模计算或实时处理场景下显著提升效率，但对阈值和步长设置较为敏感。

3种策略在子联盟规模扩展时表现出显著的复杂度差异。系统抽样在 $N$ 较大时评估跳过高，可有效降低计算成本；固定基准适用于稳定场景但无法规避子联盟遍历瓶颈；分层抽样虽然能提供更细粒度控制，但在中间层子联盟分布密集时评估压力较大。

实验结果进一步验证了3种优化抽样方法在Least core分配问题中的适用性和优势。这些方法为复杂联盟收益分配提供了高效、灵活的解决方案，能够在不同实际应用中发挥重要作用。

## 5 结束语

本文引入合作博弈中的Least core方法，为联邦学习中的收益分配提供了稳定且公平的解决方案。从理论与实践两个层面验证了其在应对数据互补性与冲突性方面的有效性，并提出分层抽样、固定基准抽样与系统抽样优化策略，以应对大规模场景下的计算复杂性。

实验结果表明，Least core能有效最小化联盟不满值，提升分配公平性与系统稳定性，优化策略也显著改善了计算效率，增强了其在复杂场景中的适用性。相关研究不仅拓展了Least core的理论应用，也为多方合作中的利益平衡提供了新思路。

尽管本文提出的Least core近似求解方法结合多种抽样策略，在提升分配效率、兼顾公平性方面展现出良好效果，但仍存在以下待改进之处。

(1) 本文聚焦于Least core在静态收益分配中的应用，其评估结构与Shapley值相似，均可作为系统提供量化的参与者价值度量。已有研究表明，Shapley值已用于动态调整训练频次、上传时机与资源分配，以优化联邦系统效率。受此启发，Least core亦具备拓展为系统激励与调控模块的潜力，有望支持更精细的参与者管理与通信优化。

(2) 虽然抽样策略具备一定灵活性，但对抽样中的参数设置（如比例、阈值）较为敏感，尚未建立统一的自适应调控机制。

我们将以此作为后续研究重点，进一步提升模型的可扩展性、稳定性与系统集成能力。

## 参考文献：

- [1] Magalhes S T D. The European Union's general data protection regulation (GDPR) [R]. 2020.
- [2] Glocker F. Der California consumer privacy act[R]. 2022.
- [3] 中华人民共和国个人信息保护法[J]. 中华人民共和国公安部公报, 2021(4): 28-36.  
Personal information protection law of the People's Republic of China[J]. Gazette of the Ministry of Public Security of the People's Republic of China, 2021(4): 28 - 36.
- [4] McMahan H B, Moore E, Ramage D, et al. Communication-efficient learning of deep networks from decentralized data[PP]. arXiv (2016-02-17)[2025-01-16]. arXiv: 1602.05629.
- [5] Dayan I, Roth H R, Zhong A X, et al. Federated learning for predicting clinical outcomes in patients with COVID-19[J]. Nature Medicine, 2021, 27(10): 1735-1743.
- [6] Oualid A, Maleh Y, Moumoun L. Federated learning techniques applied to credit risk management: a systematic literature review[J]. Edpacs, 2023, 68(1): 42-56.
- [7] Kairouz P, McMahan H B, Avent B, et al. Advances and open problems in federated learning[C]//Proceedings of the Foundations and Trends® in Machine Learning. New York: ACM Press, 2021: 1-210.
- [8] 王勇, 李国良, 李开宇. 联邦学习贡献评估综述[J]. 软件学报, 2023, 34(3): 1168-1192.  
Wang Y, Li G L, Li K Y. Survey on contribution evaluation for federated learning[J]. Journal of Software, 2023, 34(3): 1168-1192.



- [9] Kumar I E, Venkatasubramanian S, Scheidegger C, et al. Problems with Shapley-value-based explanations as feature importance measures[PP]. arXiv(2020-02-25) [2025-01-16]. arXiv: 2002.11097.
- [10] Yan T, Procaccia A D. If you like shapley then you'll love the core[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(6): 5751-5759.
- [11] Liu Z L, Chen Y Y, Yu H, et al. GTG-shapley: efficient and accurate participant contribution evaluation in federated learning[J]. ACM Transactions on Intelligent Systems and Technology, 2022, 13(4): 1-21.
- [12] Yang C Y, Liu J, Sun H, et al. WTDP-shapley: efficient and effective incentive mechanism in federated learning for intelligent safety inspection[J]. IEEE Transactions on Big Data, 2024, 10(6): 1028-1037.
- [13] Deng X T, Papadimitriou C H. On the complexity of cooperative solution concepts[J]. Mathematics of Operations Research, 1994, 19(2): 257-266.
- [14] Conitzer V, Sandholm T. Complexity of constructing solutions in the core based on synergies among coalitions[J]. Artificial Intelligence, 2006, 170(6/7): 607-619.
- [15] Bachrach Y, Rosenschein J S. Coalitional skill games[J]. Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1, 2008: 1023-1030.
- [16] Elkind E, Pasechnik D. Computing the nucleolus of weighted voting games[C]//Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2009: 327-335.
- [17] Balcan M F, Procaccia A D, Zick Y. Learning cooperative games[C]//Proceedings of the 24th International Conference on Artificial Intelligence. New York: ACM Press, 2015: 475-481.
- [18] Balkanski E, Syed U, Vassilvitskii S. Statistical cost sharing[PP]. arXiv (2017-03-09)[2025-01-17]. arXiv: 1703.03111.
- [19] Yang M W, Jarin I, Buyukates B, et al. Maverick-aware shapley valuation for client selection in federated learning[PP]. arXiv (2024-05-21)[2025-01-16]. arXiv: 2405.12590.
- [20] Chen K Y, Xu Z M. Federated learning for data market: shapley-UCB for seller selection and incentives[PP]. arXiv (2024-10-10)[2025-01-16]. arXiv: 2410.09107.
- [21] Liu Y Q, Chang S, Liu Y, et al. FairFed: improving fairness and efficiency of contribution evaluation in federated learning via cooperative shapley value[C]//Proceedings of the IEEE INFOCOM 2024 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2024: 621-630.
- [22] Wei S Y, Tong Y X, Zhou Z M, et al. Efficient data valuation approximation in federated learning: a sampling-based approach[PP]. arXiv (2025-04-23) [2025-05-10]. arXiv: 2504.16668.
- [23] GEIMER A, FIZ B, STATE R. On the volatility of shapley-based contribution metrics in federated learning[PP]. arXiv (2024-05-13)[2025-01-16]. arXiv: 2405.08044.

#### [作者简介]



林国鹏 (2000-), 男, 浙江工商大学信息与电子工程学院硕士生, 主要研究方向为联邦学习。



倪郑威 (1989-), 男, 博士, 浙江工商大学信息与电子工程学院副研究员, 主要研究方向为机器学习、物联网、无线通信等。