研究与开发

SWTA-Shapley：一种高效的联邦学习贡献评估方法

鲍世豪，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：联邦学习很好地解决了当数据隐私保护导致的“数据孤岛”问题，为了维持联邦学习系统的长期运行，需要通过适当的激励计划来吸引高质量的数据拥有者参与联邦学习训练。如何公平评估参与者对最终联邦学习模型性能的贡献是研究的重点。Shapley值是贡献评估的重要方法，已经被广泛应用。然而，Shapley值的计算复杂度高，这使Shapley值很难应用到现实的场景中。现有许多近似Shapley值算法，如引导截断梯度Shapley（guided trunction gradient Shapley，GTG-Shapley），结合了引导随机抽样、梯度重建子模型、截断3种机制，能够降低计算复杂度，得到接近准确的近似Shapley值。这样的近似算法使Shapley值在现实场景中的应用变成可能，但是GTG-Shapley依旧存在许多不足，所以基于GTG-Shapley提出了一种新的近似Shapley值算法——分层加权截断与自适应规划Shapley（stratified weighted truncation and adaptive programming Shapley，SWTA-Shapley），该算法将GTG-Shapley中的引导随机抽样换成了分层组合的方法，并将截断优化为加权截断，同时为了进一步提高准确率，采用自适应规划的方法充分利用了被截断部分的效用值。实验结果表明，相较于GTG-Shapley，SWTA-Shapley在多种场景中都能取得更高的计算效率，更适用于现实的场景。

关键词：联邦学习；Shapley值；贡献评估；激励机制；近似算法
中图分类号：TP30；TN91
文献标志码：A
doi: 10.11959/j.issn.1000−0801.2025211

SWTA-Shapley: an efficient contribution evaluation method for federated learning

BAO Shihao, NI Zhengwei
College of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

Abstract: Federated learning has effectively addressed the “data silo” issue caused by data privacy protection. To maintain the long-term operation of federated learning systems, it is necessary to attract high-quality data owners to participate in federated learning training through appropriate incentive programs. The focus of researches is on how to fairly evaluate the contribution of participants to the performance of the final federated learning model. The Shapley value is an important method for contribution assessment and has been widely applied. However, the computational

收稿日期：2025−03−17；修回日期：2025−04−12
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)

complexity of Shapley values is particularly high, making it difficult to apply Shapley value in real-world scenarios. Many approximate Shapley value algorithms, such as guided trunction gradient Shapley (GTG-Shapley) that combines guided random sampling, gradient reconstruction of sub-models, and truncation mechanisms, can reduce computational complexity and yield approximate Shapley values that are very close to the accurate values. Such approximation algorithms make the application of Shapley value in real-world scenarios possible, but GTG-Shapley still has many shortcomings. Therefore, based on GTG-Shapley, a new approximate Shapley value algorithm called stratified weighted truncation and adaptive programming Shapley (SWTA-Shapley) was proposed. This algorithm replaced the guided random sampling in GTG-Shapley with a stratified combination method and optimized truncation to weighted truncation. At the same time, to further improve accuracy, adaptive programming method was combined to fully utilize the utility values of the truncated parts. Extensive experimental results show that, compared with GTG-Shapley, SWTA-Shapley can achieve higher computational efficiency in various scenarios and is more applicable to real-world situations.

Key words: federated learning, Shapley value, contribution assessment, incentive mechanism, approximation method

0　引言

近年来，机器学习在多个领域都得到了广泛的应用，数据隐私也成为了大众所关注的问题。为了保护隐私，各国都制定了相应的数据隐私保护法，如欧盟的《通用数据保护条例》[1]，这些法规的颁布在保护用户数据隐私的同时，也造成了数据孤岛的问题，对机器学习技术的应用产生影响。具体来说，机器学习中使用的数据通常由用户生成，这些数据会包含用户的敏感信息，如身份信息、性格、信用、习惯，甚至健康信息。为了遵循保护数据隐私的法律法规，收集这些数据来训练机器学习的模型变得不再可行。

为了解决上述问题，打破“数据孤岛”的限制，由Google最先提出了一种新的机器学习范式——联邦学习[2]。这种新的人工智能模型训练范式一出现就受到广泛的关注，它可以使用在不同规模上，从轻量级边缘设备到企业级数据[3]。在联邦学习中，本地参与训练的联邦参与者在联邦服务器下载全局模型，而后利用本地数据对模型进行更新，更新后上传至服务器，服务器将所有上传的本地模型进行聚合得到新的全局模型。不难看出，这样的过程不需要参与者上传自己的数据，可以很好地保护参与者的数据隐私，使所有参与者都可以在不泄露自己数据的前提下完成模型训练并从中获得收益。

联邦学习中的一个基本问题是如何评估每个参与者的数据价值。联邦学习训练使用来自不同实体的数据，为了激励更多实体参与训练，必须根据他们对训练过程的贡献公平地分配最终模型的收益[4]。例如，联邦学习可以被应用于金融领域[5-6]的风险预测，许多可能是竞争对手的公司联合他们的所有数据来进行模型训练，所获得的模型会产生一定的收益，越多公司的参与可以使训练出来的模型更有价值，只有对所有参与训练的公司按照他们的贡献公平地分配收益，才能激励更多的公司参与到训练中。所以，需要有一个统一的方案对各个公司在训练过程中的贡献进行正确评估，以便公平地对训练收益进行分配。

Shapley值是合作博弈论中分配所有参与者联盟产生的总收益的经典方法[7]，2012年由诺贝尔经济学奖获得者Lloyd Shapley提出，于1953年首次被引入[8]，现在已经应用于各个领域的问题，从经济学[9-11]、反恐[12-13]、环境科学[14-16]到机器学习[17-19]。Shapley值定义了一个独特的利润分配方案，并且具有多种理想的属性：（1）它确保模型的所有增益分布在数据源之间；（2）分配给参与者的值符合他们对学习过程的实际贡献；（3）数据的价值在多次使用时累积。基于这

些理想属性，Shapley值被广泛地应用于贡献评估与收益分配中。

然而，尽管Shapley值具有理想的属性，它也有缺点，由于需要评估联邦学习参与者中所有组合的效用值，即所有组合计算出的子模型性能得分，Shapley值的计算过程中随着参与者数量的增加，所需要进行效用评估的组合数量会呈指数级增长。同时，对于机器学习任务来说，效用评估首先需要训练子模型，这一过程本身也需要耗费大量的时间，特别是对于拥有大体量数据的参与者（例如医院、银行等），需要花费更多的时间来训练子模型。因此，准确的Shapley值计算方法有着极高的计算复杂度，且复杂度会随着参与者数量和数据量的增加而呈指数级的增长，将Shapley值应用到现实场景中面临着巨大的挑战。

为了降低Shapley值的计算复杂度，现有许多Shapley值近似算法，通过计算近似值来降低计算复杂度。例如，引导截断梯度Shapley（guided truncation gradient Shapley，GTG-Shapley）结合了引导随机抽样、梯度重建子模型、截断3种机制，首先对所有参与者的所有排列进行随机抽样，再按照抽样出的排列顺序将参与者加入到训练中，效用评估所需要的子模型由梯度生成代替了重新训练。得到当前参与者的边际效用（即该参与者加入训练后的效用变化），如果边际效用过小则将这个排列中剩余部分的效用设为0。完成所有轮次和组合的计算之后，每个参与者将边际效用进行加权求和就能得到他们各自的Shapley值。这个方法很好地提高了计算效率。但是，GTG-Shapley依然存在不足之处，它的引导随机抽样无法避免抽样时的不确定性，这种不确定性可能导致重复效用评估，从而引起不必要的计算负担，也可能丢失重要的排列从而影响准确率，为了提高准确率必须大幅度地提高抽样次数，导致计算成本显著增加。同时，GTG-Shapley的截断机制仅基于边际效用值的大小来判断，但是最终的Shapley值的大小还与参与者在排列中不同位置时的权重有关，只通过边际效用来判断是否截断可能会忽略那些边际效用小但权重高的参与者组合，这些组合对最终Shapley值的贡献可能很大，但容易被错误地截断，从而影响结果的准确性。

本文在GTG-Shapley的基础上提出了一种新的近似Shapley值算法——分层加权截断与自适应规划Shapley（stratified weighted truncation and adaptive programming Shapley，SWTA-Shapley）。该方法对GTG-Shapley进行了改进。（1）使用分层组合的方法来代替GTG-Shapley中的引导随机抽样，将联盟中所有可能的子集按照大小进行分层，每层代表不同大小的子集。这种方法不仅避免了重复评估，还大幅地减少了需要考虑的组合数量，从而节约了计算时间。且评估的组合覆盖了全部层级，不会造成重要的组合丢失，所有可能的组合都会被考虑，保证了效用评估的公平性，提高了计算的准确率。（2）将截断机制优化为加权截断，通过权重和边际效用来共同判断是否进行截断，更准确地判断是否进行截断，确保贡献大的组合不会被错误地抛弃。（3）为了进一步增强算法，对被截断后的部分进行充分利用，采用自适应规划的方法，通过使用上一层级组合效用值的平均值来近似得到被截断部分的效用值，可以做到在不增加计算时间的情况下提高计算结果的准确率。

本文在5种不同的联邦学习设置下进行了大量的实验，用来比较本文所提的SWTA-Shapley方法和包括GTG-Shapley在内的5种近似Shapley值算法之间的性能差异。实验结果表明，SWTA-Shapley在降低计算复杂度，提高Shapley值计算准确率上明显优于GTG-Shapley，在所有实验方法中性能也是最好的。

本文的主要贡献如下。

（1）基于GTG-Shapley提出了一种新的近似Shapley值算法SWTA-Shapley，该算法相较于GTG-Shapley算法拥有更高的计算效率。对于应用Shapley值来评估联邦学习中参与者的贡献以及解决收益分配问题有很大的帮助。

（2）使用了分层组合的方式代替GTG-Shapley中的引导随机抽样，将截断机制优化为加权截断，最后结合自适应规划方法，使SWTA-Shapley相较于GTG-Shapley在计算效率上得到了明显的提升。

（3）在多种不同的联邦学习设置下进行了大量的实验，其中包括了不同分布、不同数据集以及不同参与者数量的设置，实验结果表明SWTA-Shapley的计算效率优于所有参与实验的方法，相较于GTG-Shapley有显著提升。同时还进行了消融实验，证明了本文所提的3种改进方法均对提升计算效率有很大的帮助。

1　相关工作

Shapley值是评估联邦学习中参与者贡献最公平有效的方法，但是准确Shapley值指数级的计算成本使得将Shapley值应用于现实场景中十分困难，如何降低Shapley值的计算复杂度是当前研究的重点。

现在已经存在许多用于降低Shapley值计算复杂度的方法。文献[20]提出了一种随机抽样蒙特卡罗估计方法，对参与者的所有排列进行随机抽样，以减少效用评估的数量来近似得到Shapley值。文献[7]提出了使用分组检验来加速Shapley值估计。文献[21]利用Shapley值来量化单个数据点对学习任务的贡献。此外，该文献还提出了一种有效的方法截断蒙特卡罗Shapley（truncated Monte Carlo Shapley，TMC-Shapley），这个方法使用随机蒙特卡罗抽样和在每一个轮次中截断不必要的效用评估，很好地减少了计算量。这些方法都致力于减少计算Shapley值过程中效用评估的数量。然而，随机抽样并非针对每个排列定制的，在抽样时会有很大的不确定性。这可能会导致欠抽样或者过抽样，丢失重要的联邦学习参与者或者排列，使Shapley值的准确率受到影响。

在效用评估时，每次都需要训练子模型，这导致在计算过程中经常会出现冗余的再训练过程，为了解决这个问题，文献[21]提出了梯度Shapley方法，参与者在每轮的梯度更新中都储存在联邦服务器上，在评估效用函数时通过梯度来更新子模型，代替了从头开始训练子模型。文献[22]提出了两种基于梯度的Shapley值计算方案：单轮重构算法（one-round reconstruction，OR）和多轮重构算法（mulit-round reconstruction，MR）。这两种方案都使用梯度的方法，通过梯度更新来重建子模型，取代了实际的联邦学习模型训练过程。不同之处在于MR在每轮联邦学习模型训练期间计算Shapley值，最后将所有轮次聚合得到最终结果，而OR在所有训练轮次完成后再计算一次得到最终的Shapley值。文献[2]提出了截断多轮重构算法（truncated mulit-round reconstruction，TMR）方法，该方法通过截断衰减因子λ值低于预定义阈值的所有轮次来减少计算轮次。这3种方法都显著地提高了计算近似Shapley值的效率。文献[23]提出了GTG-Shapley方法，融合了这些机制，进一步提高了计算效率。但是GTG中使用的抽样方法是由蒙特卡罗抽样改进而来的引导随机抽样，这个抽样方法只考虑了排列头部的重要性，并没有解决随机抽样有很大不确定性的问题，并且为了提高准确率还需要增加抽样的次数，增加计算时间。除此之外，GTG-Shapley中的截断方法过于简化，导致有价值的参与者组合被丢弃，影响最后的计算准确率。

本文在GTG-Shapley的基础上提出了SWTA-Shapley方法，使用了分层组合、加权截断和自

适应规划3种方法，很好地改进了GTG-Shapley的不足之处，计算效率有了明显的提升。

2 预备知识

联邦学习框架如图1所示，假设在联盟N中有n个参与者，每个参与者$i\in\{1, 2, \cdots, n\}$都具有私有的本地数据集D_i用于参与联邦训练。一共有T个全局轮次来训练联邦模型。在每一轮$t\in\{1, 2, \cdots, T\}$中，联邦参与者i下载联邦服务器的全局模型$M^{(t)}$，而后在自己的本地数据集上进行训练，得到自己的本地子模型$M_i^{(t+1)}$，并将更新的梯度$\Delta_i^{(t+1)}$发送联邦服务器，梯度的获得如下：

$$\Delta_i^{(t+1)} = M_i^{(t+1)} - M^{(t)} \tag{1}$$

联邦服务器收集所有参与者发来的梯度更新，聚合所有的梯度用于更新全局模型$M^{(t+1)}$。

$$M^{(t+1)} = M^{(t)} + \sum_i \frac{|D_i|}{|D_N|}\Delta_i^{(t+1)} \tag{2}$$

其中，$|D_i|$表示数据集D_i的大小，$|D_N| = \sum_i^n |D_i|$表示联盟的所有数据集大小。

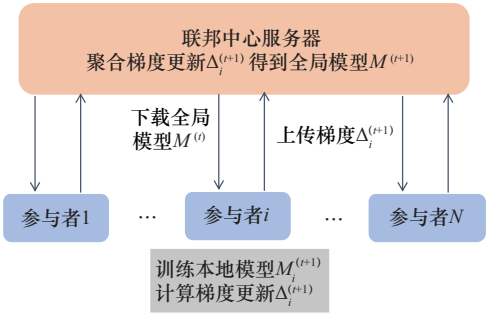

图1 联邦学习框架

参与者i的Shapley值Φ_i可以用于评估所有参与者的贡献度，最终的Shapley值结果是所有T个轮次Shapley值的和，轮次t中的Shapley值Φ_i^t定义为：

$$\Phi_i^t(N, V) = \frac{1}{|N|}\sum_{S\subseteq N\setminus\{i\}} \frac{V(S\cup\{i\}) - V(S)}{\binom{|N|-1}{|S|}} \tag{3}$$

$$\Phi_i = \sum_{t\in\{1, 2, \cdots, T\}} \Phi_i^t \tag{4}$$

其中，S表示来自所有参与者联盟N的不包含i的子集，V表示效用函数，用于评估子集S的效用。在机器学习中，效用评估函数$V(\cdot)$基于使用S在单独测试集上训练得到的子模型来预测性能。

$$V(S) = V(M_S) = V(A(M^{(0)}, D_S)) \tag{5}$$

其中，M_S是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。

式（3）中的$V(S\cup\{i\}) - V(S)$表示参与者i加入子集S后的边际效用，用Δv_i来指代。$1/|N| \times \binom{|N|-1}{|S|}$表示子集$S$的权重，由$w_{|S|}$来指代，大小只与此时的子集$S$的长度有关。所以式（3）也可表示为：

$$\Phi_i(N, V) = \sum_{S\subseteq N\setminus\{i\}} \frac{1}{|N| \times \binom{|N|-1}{|S|}} \times (V(S\cup\{i\}) - V(S)) = \sum_{S\subseteq N\setminus\{i\}} w_{|S|} \times \Delta v_i \tag{6}$$

式（6）代表参与者i的Shapley值是由他在每个不包含他的子集的边际效用加权求和得到的。

主要符号及含义见表1。

3 Shapley值近似算法研究

本文基于GTG-Shapley提出了一种新的Shapley值近似算法SWTA-Shapley，本节具体介绍这个方法。

Shapley值的另一个定义是参与者i在所有可能顺序加入训练时的边际效用平均值，设$\pi(N)$为联盟中所有参与者的排列集合，总共有$|N|!$种排列。给定$\pi(N)$中一个排列$O = [O[1], O[2], \cdots, O[n]]$，设排列$O$中的某个参与者$i = O[j]$，$\mathrm{Pre}^i(O) = [O[1], O[2], \cdots, O[j-1]]$表示在$i$之前加入联盟的所有参与者。因此式（3）中Shapley值的定义可以改

表 1　主要符号及含义

符号	含义
i，n，N	任一参与者，参与者总数，所有参与者的联盟
t，T，k	任一训练轮次，训练的轮次总数，抽样次数
D_i，D_N	参与者i的本地数据集，联盟的总数据集
$M^{(t)}$，$M_i^{(t)}$	t轮次的全局模型，t轮次参与者i的本地子模型
Φ_i，Φ_i^t	参与者i的Shapley值，参与者i在t轮次的Shapley值
C，S	联盟中的任意子集，联盟任意不包含i的子集
$\pi(N)$，$\pi_{S,i}(N)$	N中所有可能的排列集合，N中所有满足参与者i在第$\|S\|$个位置的排列集合
O，$\mathrm{Pre}^i(O)$	$\pi(N)$中的任意一个排列，排列O中参与者i之前的所有参与者组成的子排列
v_N，v_0，$w_{\|S\|}$	完整联盟的效用值，空联盟效用值，排列中不同位置的权重
$V(\cdot)$，Δv_i	效用评估函数，参与者i的边际效用
η，ϵ，ϵ_1	轮内截断阈值的比例系数，轮内截断阈值，轮间截断阈值

写为：

$$\Phi_i(N,V)=\sum_{O\in\pi(N)}\frac{1}{|N|!}[V(\mathrm{Pre}^i(O)\cup\{i\}-V(\mathrm{Pre}^i(O))] \tag{7}$$

其中，$V(\mathrm{Pre}^i(O)\cup\{i\}-V(\mathrm{Pre}^i(O))$表示参与者$i$加入训练后带来的额外价值，也就是$i$在排列$O$中的边际效用$\Delta v_i$，由此可以得知：

$$\Delta v_i=[V(S\cup\{i\})-V(S)]=V(\mathrm{Pre}^i(O)\cup\{i\}-V(\mathrm{Pre}^i(O)),\ \forall O[\in\pi S,i](N) \tag{8}$$

其中，$\pi_{S,i}(N)$是所有满足$\mathrm{Pre}^i(O)$包含的元素与S相同的排列集合，即排列O的前$|S|$个位置是子集S中的参与者的随机填充，第$|S|+1$个位置是参与者i，其余位置是N中剩下的参与者的随机填充。通过式（6）和式（8）可以得到：

$$w_{|S|}{}^*\Delta v_i=\sum_{O\in\pi_{S,i}(N)}\frac{1}{|N|!}[V(\mathrm{Pre}^i(O)\cup\{i\}-V(\mathrm{Pre}^i(O))] \tag{9}$$

由式（9）可以观察出：（1）无论参与者i前面的参与者加入训练的顺序如何，他的边际效用都是一样的，这也说明了对于任意的排列来说，他的效用只与排列中的参与者组成有关，而与这些参与者的顺序无关，因此，无论$\mathrm{Pre}^i(O)$中的参与者以何种顺序排列，$V(\mathrm{Pre}^i(O))$的值都是相同的；（2）边际效用Δv_i的权重$w_{|S|}$是Shapley值计算过程中的关键因素，对最后的计算结果有很大的影响。基于这两个结论，本文对GTG-Shapley方法进行改进。

3.1　分层组合

在GTG-Shapley方法中，引导随机抽样的方法随机抽取排列O来近似计算Shapley值，过程中需要将排列中的值按顺序逐个加入训练，如对于排列[3, 2, 1]，就需要逐个对[3]、[3, 2]、[3, 2, 1]进行效益评估，所以过程中需要多次对子排列进行效用评估，但由于排列的效用值仅与他的参与者组成有关，和顺序无关，所以会对很多仅顺序不同但参与者组成相同的子排列进行重复的效用评估，比如对[1, 3]和[3, 1]进行效用评估的结果是相同的，这会严重浪费计算资源，影响计算的效率。同时引导随机抽样的随机性，还会导致丢失重要排列，增加计算结果与真实值之间的误差，为了提高最后计算结果的精度，需要增加抽样次数k。随着k的增加，近似值的误差会逐渐减小。但是随着抽样次数的增加，需要的效用评估次数也会越多，尤其是当参与者数量较大时，效

用评估的次数会指数式增加，需要的计算成本非常高。

本文在SWTA-Shapley使用了分层组合的方法来代替引导随机抽样，分层组合就是将联盟中的所有子集按照大小进行分层，第1层存放所有长度为1的子集，第2层存放长度为2的子集，以此类推，对每一层的组合进行效用评估。假设一个拥有4个参与者的联盟{1, 2, 3, 4}，使用SWTA-Shapley来计算，分层组合会将所有的子集分为4层，并对每一层中的子集进行效用评估，共需要进行15次无重复的效用评估。使用GTG-Shapley来计算时，需要使用引导随机抽样来随机抽取排列，为了使效用评估次数与使用SWTA-Shapley方法时接近，假设抽取出4个排列，一共需要对16个子排列进行效用评估，分层组合与引导随机抽样得到的效用评估组合对比如图2所示。引导随机抽样随机抽取了[1, 2, 3, 4]，[3, 2, 4, 1]，[2, 1, 4, 3]，[4, 1, 2, 3]这4个排列，所有参与效用评估的子排列对应的子集如图2右侧所示。对比图2左右两侧可以看出，虽然两个方法中参与效用评估的子集数量接近，一个为15个，一个为16个，但是很明显，分层组合的方法对每一个有可能的子集都进行了评估，且不存在重复评估，而引导随机抽样则出现了很多子集未评估或者重复评估的现象。重复的评估会导致计算资源的严重浪费，而大量子集未评估则会影响最终计算结果的准确率。分层组合的方法解决了这些问题，不会出现重复的评估，而且覆盖了全层级的所有子集，不会出现重要评估遗漏，可以在计算准确率和计算时间上取得良好的平衡，有效地提高了Shapley值的计算效率。

3.2 加权截断

采用截断机制可以机会性地丢弃对最终Shapley值贡献较小的计算过程，在保证准确率的前提下减少计算所需要的时间。在GTG-Shapley的截断策略中，设置了一个截断阈值$\epsilon = \eta \times |v_N - v_0|$，其中，$v_N = V(N)$是完整联盟的效用值，$v_0$是空联盟时的效用值，$\eta$是一个比例系数，用于调整截断阈值的严格程度。设置剩余边际效用$\Delta\bar{v}_i = |V(N) - V(\mathrm{Pre}^i(O)|$来表示剩下未加入训练的参与者拥有的边际效用和。根据截断条件，边际效用Δv_i被分为两种情况：

$$\Delta v_i = \begin{cases} 0, & \Delta\bar{v}_i < \epsilon \\ V(\mathrm{Pre}^i(O) \cup \{i\}) - V(\mathrm{Pre}^i(O)), & \Delta\bar{v}_i \geqslant \epsilon \end{cases} \quad (10)$$

即，如果当参与者i加入训练后，$\Delta\bar{v}_i$小于阈值则将其视为0，否则保留其原始值。GTG-Shapley的截断策略对所有的Δv_i采取了一视同仁的处理。但是，对最终Shapley值的贡献大小并不仅取决于边际效用Δv_i，权重$w_{|S|}$也发挥着至关重要的作用。

为了探究权重的重要性，本文进行了一个实验，将MNIST数据集[24]以独立同分布与相同数据量大小的方式分配给一个拥有5名参与者的联盟中的每个参与者，使用GTG-Shapley来计算5名参与者在联邦训练中的Shapley值。不同位置

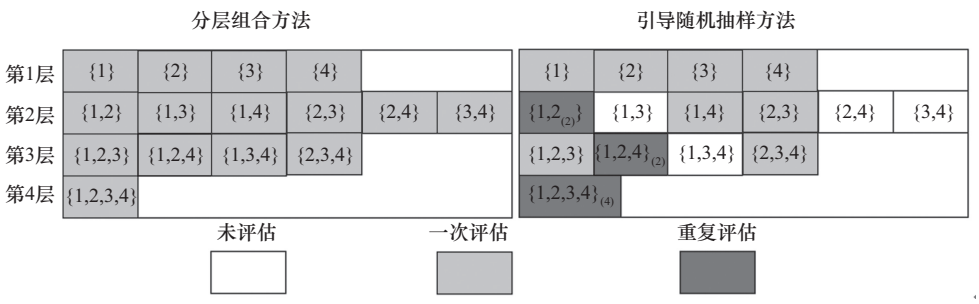

图2 分层组合与引导随机抽样得到的效用评估组合对比

下的权重、边际效用与乘积之间的关系如图3所示。可以看出，虽然位置越靠后边际效用越低，但是权重是中间低两边高，在第5个位置上的边际效用虽然是最低的，但是加权之后则超越了前面，由式（6）可知Shapley值是边际效用的加权和，说明虽然靠后位置的边际效用小，但因为权重高，还是有可能对最终结果有较大贡献的。GTG-Shapley的截断策略过于简化，仅考虑了边际效用的大小却忽略了不同位置上的权重变化，导致对Δv_i的处理不够精确，增加最终计算结果与真实值之间的误差。所以，本文提出了加权截断的策略，将截断条件改为$w_{|S|}*\Delta\bar{v}_i<\epsilon$时进行截断，这个截断条件结合了边际效用$\Delta v_i$与此时对应的权重$w_{|S|}$，使截断的结果更准确，防止对最终贡献较大的边际效用因为不够精确的截断策略而被错误地丢弃，提高了计算准确率。

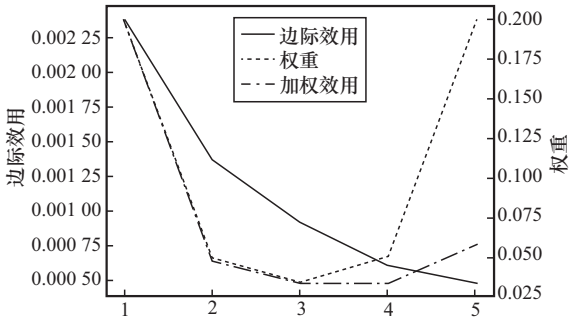

图3　不同位置下的权重、边际效用与乘积之间的关系

3.3　基于效用值的自适应规划

在GTG-Shapley方法的截断机制中，对被截断的部分直接将效用值设为0，然而这一做法忽视了即便微小的效用值也可以对最终结果的准确率有所贡献。为了在保持计算效率的同时充分利用这些被截断的效用值，本文提出了基于效用值自适应规划的方法。该方法的核心思想在于自适应地调整效用值的计算方式，具体方程为：

$$V(C)=\begin{cases}\text{average}(V(C\backslash\{i\})), & w_{|C\backslash\{i\}|}*\left|V(N)-V(C\backslash\{i\})\right|<\epsilon,\ \forall i\in C\\ \text{效用评估}, & \text{其他}\end{cases}\tag{11}$$

式（11）表示，对于一个子集C，如果它的每一个成员所拥有的加权边际效用都小于阈值，说明它的所有成员在$|C|$位置加入训练都不会使模型性能有明显的提升，为了利用好这部分效用值又不增加额外的计算开销，在子集C中任选一个i，得到子集$S=C\backslash\{i\}$，这样的子集S一共有$|C|$个，而$|S|=|C|-1$，说明子集S是子集C的上一层级子集，可以对所有的$|C|$个$V(S)$取平均来近似得到$V(C)$，因为这些成员的边际效用足够小。这样一来，不仅利用了这些被截断部分的效用值，还不会增加计算的时间，而对于不需要截断的部分依旧使用常规的效用评估来获得效用值。通过这个方法，可以自适应地调整获得效用值的方式，在计算时间和准确率之间取得平衡。

3.4　SWTA-Shapley算法

SWTA-Shapley算法如算法1所示。根据算法1可知，首先需要确定参与训练的人数n，性能评估函数$V(\cdot)$和初始的联邦学习模型$M^{(0)}$，初始化一个长度为n全为0的表格来存放各个客户的Shapley值Φ_i。在某一轮次t中，客户端需要各个参与者先下载全局模型$M^{(t-1)}$，而后训练好本地模型后得到更新的梯度$\Delta_i^{(t)}$用于在服务器上通过联邦平均算法得到下一个轮次的全局模型$M^{(t)}$，得到这些后从第8行开始正式计算第t轮次中各个参与者的Shapley值$\Phi_i^{(t)}$。先计算出$M^{(t-1)}$的效用值v_0和$M^{(t)}$的效用值v_N，以及后面轮内截断时需要的权重值$w_{|S|}$，然后进入轮间截断，判断当前轮次是否需要被丢弃，如果当前轮次的效用值$|v_N-v_0|>\epsilon 1$则继续进行计算。对每一个层级进行遍历，设空字典v_{lut}^j用于存储当前层级中所有子集的效用值，对联盟N中的所有大小为j的子集C进行遍历，对于C中的任意参与者i，获得C中不含i的子集S，S是C的上一层级子集。然后，第（16）行开始轮内截断，当加权的剩余边际效用$w_{|S|}\times|$

$v_N - v_{\text{lut}}[S]$大于阈值，说明参与者i对最后的结果有足够的影响，不需要截断，于是先构建子模型$\tilde{M}_C^{(t)}$，再对其进行效用评估得到子集C的效用值v_C。当所有i对应的S都满足截断条件时，则会将这些子集S的效用值$v_{\text{lut}}[S]$放入集合v_{S_visited}中，并计算平均值来得到当前子集C的效用值v_C，最后将v_C存入字典v_{lut}^j中用于下一层级的计算。第（27）～（30）行对所有参与者在当前子集C中的贡献值$w_{|S|}\times(v_C - v_{\text{lut}}[S])$进行累加，再选取下一个子集$C$。完成第$j$个层级的计算后将当前层级的所有效用值$v_{\text{lut}}^j$更新到字典$v_{\text{lut}}$中，然后进入下一个层级的计算。在完成所有层级的计算后便得到了所有参与者在t轮次的Shapley值$\Phi_i^{(t)}$，最后将所有轮次的Shapley值求和得到所有参与者的Shapley值$\{\Phi_1, \Phi_2, \cdots, \Phi_n\}$。

算法1 SWTA-Shapley算法

输入 初始联邦模型$M^{(0)}$，性能评估函数$V(\cdot)$，共n个参与者，轮间截断阈值ϵ_1、轮内截断阈值比例系数η

输出 Shapley值Φ_i，$i \in \{1, 2, \cdots, n\}$

（1）初始化$\Phi_i = 0$，$i \in \{1, 2, \cdots, n\}$

（2）**for** $t = 0, 1, \cdots, T-1$ **do**

（3） **for** 每一个参与者i **do** #客户端

（4） $\Delta_i^{(t)} = \text{ClientUpdate}(i, M^{(t-1)})$;

（5） **end for**

（6） $M^{(t)} = \text{FedAvg}(\{\Delta_i^{(t)}\}, M^{(t-1)})$; #服务器端

（7） $\Phi_i^{(t)} = 0$，$i \in \{1, 2, \cdots, n\}$;

（8） $v_0 = V(M^{(t-1)})$，$v_N = V(M^{(t)})$，$w_{|S|} = \dfrac{1}{|N| * \binom{|N|-1}{|S|}}$;

（9） **if** $|v_N - v_0| > \epsilon_1$ **then** #轮间截断

（10） **for** $j = 1, 2, \cdots, |N| - 1$ **do**

（11） 设$v_{\text{lut}}^j = \{\}$;

（12） **for** 任意子集$C \subseteq N$，$|C| = j$ **do**

（13） 设$v_{S_\text{visited}} = \{\}$;

（14） **for** $\forall i \in C$ **do**

（15） 设子集$S = C \backslash \{i\}$;

（16） **if** $w_{|S|} \times |v_N - v_{\text{lut}}[S]| \geqslant \epsilon_2 \times |v_N - v_0|$ #轮内截断

（17） $v_C = V(\tilde{M}_C^{(t)}) = V(\text{FedAvg}(\{\Delta_C\}, M^{(t-1)}))$;

（18） **break**;

（19） **else**

（20） 将$v_{\text{lut}}[S]$加入v_{S_visited};

（21） $v_C = \sum v_{S\_\text{visited}} / |v_{S\_\text{visited}}|$;

（22） **end if**

（23） **end for**

（24） 更新$v_{\text{lut}}^j[C] = v_C$;

（25） **for** $\forall i \in C$ **do**

（26） 设子集$S = C \backslash \{i\}$;

（27） $\Phi_i^{(t)} = \Phi_i^{(t)} + w_{|S|} \times (v_C - v_{\text{lut}}[S])$;

（28） **end for**

（29） **end for**

（30） 更新$v_{\text{lut}} = v_{\text{lut}}^j$;

（31） **end for**

（32） **for** $\forall i \in N$ **do**

（33） 设子集$S = N \backslash \{i\}$;

（34） $\Phi_i^{(t)} = \Phi_i^{(t)} + w_{|S|} \times (v_N - v_{\text{lut}}[S])$;

（35） **end for**

（36） **end if**

（37） $\Phi_i = \Phi_i + \Phi_i^{(t+1)}$，$i \in \{1, 2, \cdots, n\}$;

（38）**end for**

（39）**return** $\{\Phi_1, \Phi_2, \cdots, \Phi_n\}$

SWTA-Shapley方法的计算过程如图4所示，可以直观地看出SWTA-Shapley的具体计算过程，首先在t轮次开始时先判断本轮次效用值是否小于阈值，如果小于则说明本轮次训练的性价比不高可以忽略，进入下一个轮次，反之则开始本轮次的计算。先进行分层组合，将联盟中所有可能

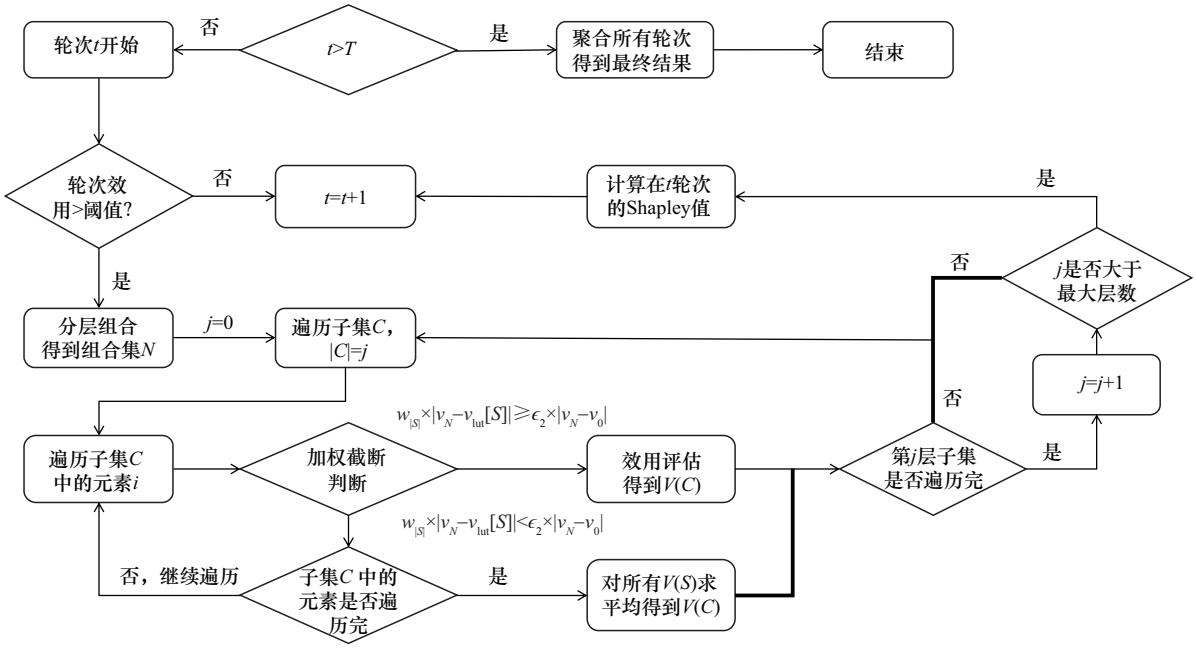

图4　SWTA-Shapley 方法的计算过程

的子集按照大小划分并存放到列表 N 中。遍历一个层级中的子集 C，然后遍历子集 C 中的所有元素 i，得到 $S=C\backslash\{i\}$，进行加权截断的判断，如果子集 C 的全部 S 都满足加权截断条件，则通过对所有 $V(S)$ 求平均得到 $V(C)$，否则就通过效益评估来得到 $V(C)$，遍历完这个层级的子集后对下一个层级进行遍历，直到所有的子集都被遍历完，所有的 $V(C)$ 都得到后就可以计算本轮次所有用户的 Shapley 值，然后进入下一轮次。在所有轮次结束后将每个轮次的结果相加得到所有用户最终的 Shapley 值。

4　实验结果与分析

在本节中，将 SWTA-Shapley 方法与 GTG-Shapley 算法以及现有的其他 4 种先进算法在不同数据集和数据分布情况下进行性能比较。

4.1　实验设置

4.1.1　数据集和分布场景

在本实验中，使用了 MNIST 数据集和 Fashion-MNIST 数据集，MNIST 数据集是一个手写数字的图像数据集，包含了 0 到 9 的数字，每一个样本都是 28 像素 × 28 像素的灰度图像，图像中的数字经过尺寸标准化并位于图像中心，每一个图像都与其相应的标签一起存储，标签指示图像代表的数字，该数据集由 60 000 个训练样本和 10 000 个测试样本组成。Fashion-MNIST 数据集与 MNIST 数据集类似，每个样本也都是图像，但是与 MNIST 的数字不同，Fashion-MNIST 包含的是 10 个类别的时尚物品，如鞋子、衣服、包包等。相较于 MNIST 中的数字，Fashion-MNIST 更复杂，更难以识别。

本文对这两个数据集进行预处理，从 MNIST 和 fashion-MNIST 的每个标签中随机选择 5 421 个样本（即每个数据集选择 54 210 个样本）。然后，对每个数字和每个物品随机抽样 892 个图像作为测试数据集（即每个数据集共选择 8 920 个样本作为测试集）。为了验证不同数量参与者对算法的影响，在实验中设置了 5 用户和 10 用户的两种情况。为了在不同的联邦学习设置下评估本文所提算法，设计了 5 种不同的分布场景，包含独立同分布和非独立同分布。

（1）相同数据量大小与独立同分布：对于 5

用户情况，对每个类别随机抽取1/5的图片分配给每个用户，每个用户拥有10 842个图像的数据，对于10用户情况来说，每个类别选取1/10的图片分配给每个用户，每个用户有5 421个图像。这样确保每个用户都能拥有相同数量且相同类别的数据。

（2）相同数据量大小与非独立同分布：每个参与者都拥有相同数量的样本，但是5用户时，参与者1的数据集包含80%的类别1和2，剩余20%的类别1和2平均分配给剩余的参与者，10用户时，参与者1和2拥有80%的类别1和2，剩余的平均分配给其他的用户。其他用户的分配也参照此模式，最终每个用户拥有相同数量的数据，但是这些数据来自不同的类别。

（3）不同数据量大小和独立同分布：根据预定义的比例对整个数据集进行抽样，例如，5用户时，参与者1抽取每个类别的10%，参与者2抽取每个类别的15%，后面参与者的比例分别为20%、25%、30%。10用户时参与者1和2的比例为10%，参与者3和4的比例为15%，以此类推。最终使每个用户的数据都在每个类别中按一定的比例抽取出来，分布相同但是数据量的大小不同。

（4）标签噪声和相同数据量大小：采用分布（1）设置的数据集，然后对数据集中的样本标签进行翻转来对标签添加噪声，对每个参与者数据集中的标签翻转采用不同的比例，5用户时，参与者1为0%，参与者2为5%，参与者3为10%，参与者4为15%，参与者5为20%。10用户时，参与者1和2为0%，参与者3和4为5%，以此类推。

（5）特征噪声和相同数据量大小：采用分布（1）设置的数据集，然后在输入图像中加入不同比例的高斯噪声，5用户时，参与者1为0%，参与者2为5%，参与者3为10%，参与者4为15%，参与者5为20%。10用户时，参与者1和2为0%，参与者3和4为5%，参与者5和6为10%，参与者7和8为15%，参与者9和10为20%。

4.1.2 方法比较

将SWTA-Shapley与以下方法进行比较。

（1）ExactShapley[8]：该方法遵循了式（3）计算最准确的Shapley值，对参与者的所有可能组合进行评估，每个子模型都是从数据集中重新训练出来的，所耗费的时间最长。

（2）TMC-Shapley[21]：在这种方法下，联邦学习参与者子模型是用本地数据集和初始联邦学习模型训练的，采用了蒙特卡罗抽样和截断不必要的子模型训练和评估来提高计算效率。

（3）MR[22]：在该方法下，联邦参与者基于其梯度更新每一轮重建子模型。在每个轮次中计算每名参与者的Shapley值。参与者的最终Shapley值是在所有轮次的Shapley值的总和。

（4）OR[22]：在该方法下，联邦学习参与者基于其梯度更新每一轮重建子模型，在所有轮次结束之后最终进行一次Shapley值计算来得到所有用户的Shapley值。

（5）GTG-Shapley[23]：在该方法下，参与者通过梯度重建得到子模型，并且使用引导随机抽样的方法来选取每次评估的组合，采用截断机制来避免不必要的效用评估过程。

（6）TMR[2]：在该方法下，在每轮中使用参与者的梯度更新独立计算Shapley值，衰减参数λ为权重，以放大来自早期轮的Shapley值，通过截断机制来避免不必要的轮次，减少计算时间。

4.1.3 性能评估指标

设联盟中所有参与者真实的Shapley值为$\boldsymbol{\Phi}^*=\{\Phi_1^*, \Phi_2^*, \cdots, \Phi_n^*\}$，使用近似方法计算出的近似Shapley值为$\boldsymbol{\Phi}=\{\Phi_1, \Phi_2, \cdots, \Phi_n\}$，通过以下指标对所有方法的性能进行评估。

（1）时间：计算总时间来评估每个方法的计算效率。

（2）欧几里得距离：欧几里得距离是常见的衡

量空间中两点之间距离的方法，$\boldsymbol{\Phi}=\{\Phi_1,\Phi_2,\cdots,\Phi_n\}$ 与 $\boldsymbol{\Phi}^*=\{\Phi_1^*,\Phi_2^*,\cdots,\Phi_n^*\}$ 之间的欧几里得距离越小，说明这两个向量之间越接近，也就说明近似值与真实值之间的相似程度越高，那么算出这个近似值的近似方法计算准确率越高。欧几里得距离的定义为

$$\sqrt{\sum_{i=1}^{n}(\Phi_i^*-\Phi_i)^2} \tag{12}$$

（3）余弦相似度：余弦相似度是通过测量两个向量在向量空间中的夹角来评估它们的相似性。余弦相似度取值范围为–1 ~ 1，越接近 1 表示两个向量越相似，$\boldsymbol{\Phi}$ 与 $\boldsymbol{\Phi}^*$ 之间的余弦相似度越接近 1，说明它们之间越相似，说明算出 $\boldsymbol{\Phi}$ 的这个近似方法计算准确率越高。其定义如下：

$$\cos(\boldsymbol{\Phi},\boldsymbol{\Phi}^*)=\frac{\sum_{i=1}^{n}\Phi_i^*\Phi_i}{\sqrt{\sum_{i=1}^{n}(\Phi_i^*)^2}\sqrt{\sum_{i=1}^{n}(\Phi_i)^2}} \tag{13}$$

结合这 3 个指标，计算时间越短，与真实值之间欧几里得距离越短，余弦相似度越高的近似方法性能越好。

4.2　结果分析

本节按不同的数据分布对实验的结果进行分析。由于 Shapley 值计算时的随机性，在实验中改变了随机种子的数值，对每个方法都进行了至少 50 次的重复实验，将 50 次的结果取平均，以确保实验结果的稳定性。

4.2.1　相同数据量大小和独立同分布

在相同数据量大小和独立同分布场景中的实验结果见表 2，相较于准确的 ExactShapley 方法，这些方法在计算时间上都有所减少。SWTA-Shapley 相较于 GTG-Shapley，欧几里得距离减少约 8% ~ 17%，余弦相似度提高了约 50%，也就是余弦相似度接近 1 的程度提高了约 50%，计算时间减少了约 30%。同时，SWTA-Shapley 方法的计算时间在所有方法中表现最好，准确率上仅次于 MR 与 TMR，所有方法中 TMC-Shapley 在计算时间上表现最差，其次是 MR 方法。这两个方法相较于其余 4 个方法要慢得多，这是由于 TMC-Shapley 仅采用了轮内截断机制，而 MR 仅采用了梯度代替子模型重头训练，并且还要在每一个轮次进行聚合操作，所以计算时间上的改善有限，与其他方法有着较大差距。

4.2.2　相同数据量大小和非独立同分布

相同数据量大小和非独立同分布场景中的实验结果见表 3，在非独立同分布的环境中所有方法的准确性都有所下降，计算时间也有所增加。其中，SWTA-Shapley 方法拥有比 GTG-Shapley 更好的性能表现，相较于 GTG-Shapley 计算时间在 5 用户时减少约 20%，当参与者数量增加时则效果更明显，减少了约 50% 的计算时间，这是由于分层组合方法减少了需要考虑的组合数量，尤其

表 2　在相同数据量大小和独立同分布场景中的实验结果

Shapley 值近似算法	MNIST 数据集						Fshion-MNIST 数据集					
	5 用户			10 用户			5 用户			10 用户		
	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s
ExactShapley	—	—	3 364	—	—	111 506	—	—	3 674	—	—	178 566
GTG-Shapley	0.013 6	0.999 948	303	0.011 4	0.999 902	1 666	0.013 5	0.999 958	237	0.011 3	0.999 917	1 372
OR	0.029 2	0.997 003	345	0.014 1	0.998 544	1 666	0.013 8	0.999 128	247	0.014 7	0.998 117	1 338
MR	0.010 6	0.977 523	1 022	0.088 2	0.999 892	8 820	0.011 4	0.999 648	644	0.009 4	0.999 152	10 411
TMR	0.011 1	0.996 061	522	0.009 7	0.999 460	2 720	0.010 7	0.999 935	295	0.009 8	0.999 835	3 072
TMC-Shapley	0.068 4	0.980 792	2 004	0.083 8	0.953 308	16 451	0.072 1	0.976 471	2 553	0.071 3	0.953 345	17 579
SWTA-Shapley	0.011 2	0.999 972	288	0.009 7	0.999 946	935	0.012 3	0.999 970	206	0.010 2	0.999 949	1 026

是参与者数量增加时减少的数量更多。同时相较于GTG-Shapley，欧几里得距离也减少了8%~10%，余弦相似度提高了30%~50%。在所有方法中，原本准确率最高的TMR与MR方法被SWTA-Shapley方法超越，尤其是MR成为了所有方法中准确率最低的方法。SWTA-Shapley方法成为了准确率最高的方法，同时计算时间也是最少的，说明SWTA-Shapley方法虽然在非独立同分布的环境中性能有所下降，但是相较于其他方法还有最好的性能表现。

4.2.3　不同数据量大小和独立同分布

不同数据量大小和独立同分布场景中的实验结果见表4，当实验场景重新变为独立同分布后，所有方法的准确率与计算时间相较分布不同的实验场景有了很大的改善。SWTA-Shapley相较于GTG-Shapley在欧几里得距离上减少了6%~10%，在余弦相似度上提高了30%~60%，在计算时间上减少了约12%~54%，也是在增加参与者数量后有了更好的改善，说明对于现实中参与者数量很大的场景中，SWTA-Shapley也能取得很好的效果。SWTA-Shapley在所有方法中取得了最好的性能，除了TMR与GTG-Shapley和SWTA-Shapley性能接近，其他方法都存在较大的差距，尤其是OR方法在这个场景中欧几里得距离相较于之前增加了2～3倍，为准确率最低的方法，说明了OR不适合使用在独立同分布不同数据量大小的场景中。

4.2.4　标签噪声和相同数据量大小

标签噪声和相同数据量大小场景中的实验结果见如表5，在标签加入噪声后，SWTA-Shapley依旧拥有极高的准确率，相较于GTG-Shapley欧几里得距离减少了约6%~10%，余弦相似度提高了20%~50%，计算时间减少了20%~50%。此时在Fashion-MINIST数据集上MR方法的准确率超越了SWTA-Shapley成为准确率最高的方法，但是其仅优于TMC-Shapley的计算时间影响了计算效率，SWTA-Shapley依旧是性能最好的方法。

4.2.5　特征噪声和相同数据量大小

特征噪声和相同数据量大小场景中的实验结果见表6，在特征加入噪声的情况下，SWTA-Shapley相较于GTG-Shapley欧几里得距离减少了5%~10%，余弦相似度提高了约50%，计算时间减少了约50%，可以看出，SWTA-Shapley的性能始终优于GTG-Shapley，在所有方法中，GTG-Shapley与TMR在准确率上接近SWTA-Shapley，但是需要更多的计算时间，SWTA-Shapley依旧是性能最优的方法。

表3　在相同数据量大小和非独立同分布场景中的实验结果

Shapley值近似算法	MNIST数据集						Fashion-MNIST数据集					
	5用户			10用户			5用户			10用户		
	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s
ExactShapley	—	—	4 071	—	—	162 779	—	—	3 676	—	—	179 804
GTG-Shapley	0.032 1	0.996 494	403	0.038 1	0.989 091	4 707	0.063 5	0.981 776	262	0.081 6	0.970 021	5 907
OR	0.080 5	0.971 156	350	0.056 4	0.990 000	2 888	0.065 2	0.009 988	244	0.079 8	0.963 916	3 064
MR	0.092 2	0.996 233	1 038	0.088 1	0.978 752	8 935	0.107 1	0.856 918	647	0.093 2	0.833 807	10 439
TMR-Shapley	0.036 4	0.988 521	512	0.035 3	0.987 574	2 775	0.064 6	0.982 103	295	0.075 5	0.965 952	2 970
TMC	0.071 5	0.980 235	2 168	0.075 5	0.960 265	31 412	0.061 9	0.981 138	2 958	0.075 9	0.955 132	38 027
SWTA-Shapley	0.029 8	0.997 437	348	0.034 8	0.994 352	2 478	0.059 0	0.983 067	223	0.074 5	0.971 378	2 580

表4　在不同数据量大小和独立同分布场景中的实验结果

Shapley值近似算法	MNIST数据集						Fashion-MNIST数据集					
	5用户			10用户			5用户			10用户		
	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s
ExactShapley	—	—	3 716	—	—	121 255	—	—	3 682	—	—	162 106
GTG-Shapley	0.014 8	0.999 487	449	0.012 3	0.999 450	1 623	0.015 2	0.999 436	246	0.013 4	0.999 078	1 417
OR	0.396 8	0.682 380	356	0.228 1	0.970 473	1 425	0.344 8	0.009 913	252	0.335 3	0.987 356	1 351
MR	0.032 7	0.997 848	1 027	0.012 9	0.997 318	12 311	0.026 7	0.009 968	655	0.022 7	0.995 147	10 442
TMR	0.032 7	0.999 213	431	0.010 3	0.999 344	2 721	0.016 8	0.999 270	260	0.016 8	0.998 778	2 983
TMC-Shapley	0.073 1	0.975 468	2 043	0.085 4	0.955 008	17 599	0.066 5	0.980 117	2 746	0.068 5	0.958 960	25 568
SWTA-Shapley	0.013 2	0.999 755	366	0.011 1	0.999 615	740	0.014 2	0.999 770	215	0.011 8	0.999 743	1 017

表5　标签噪声和相同数据量大小场景中的实验结果

Shapley值近似算法	MNIST数据集						Fashion-MNIST数据集					
	5用户			10用户			5用户			10用户		
	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s
ExactShapley	—	—	3 570	—	—	231 132	—	—	3 664	—	—	177 664
GTG-Shapley	0.012 5	0.999 921	460	0.011 1	0.999 907	1 440	0.011 7	0.999 901	223	0.011 1	0.999 954	1 411
OR	0.124 5	0.682 380	340	0.228 1	0.764 489	1 500	0.249 1	0.809 717	250	0.166 1	0.823 714	1 338
MR	0.024 7	0.997 848	1 038	0.024 7	0.998 795	12 764	0.011 2	0.999 499	638	0.009 2	0.999 031	10 507
TMR	0.019 7	0.999 213	432	0.019 7	0.999 863	2 687	0.011 8	0.999 054	233	0.010 9	0.999 810	2 067
TMC-Shapley	0.082 1	0.975 468	2 042	0.082 1	0.951 780	14 371	0.076 8	0.979 335	2 539	0.070 3	0.955 465	22 924
SWTA-Shapley	0.011 1	0.999 961	309	0.010 4	0.999 942	656	0.010 8	0.999 922	207	0.010 1	0.999 965	915

表6　特征噪声和相同数据量大小场景中的实验结果

Shapley值近似算法	MNIST数据集						Fashion-MNIST数据集					
	5用户			10用户			5用户			10用户		
	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s
ExactShapley	—	—	3 516	—	—	249 689	—	—	3 700	—	—	177 746
GTG-Shapley	0.012 6	0.999 921	457	0.010 9	0.999 833	1 344	0.011 6	0.999 827	334	0.009 5	0.999 647	1 797
OR	0.111 2	0.998 358	345	0.058 4	0.953 828	1 447	0.108 4	0.950 334	250	0.048 6	0.979 441	1 356
MR	0.047 4	0.999 394	1 031	0.030 9	0.993 235	12 752	0.071 1	0.978 862	647	0.012 5	0.998 553	10 507
TMR	0.013 4	0.999 579	433	0.010 5	0.999 893	2 624	0.012 5	0.999 823	239	0.011 5	0.999 443	2 040
TMC-Shapley	0.084 9	0.982 232	2 054	0.081 2	0.673 875	27 352	0.066 2	0.979 613	2 882	0.080 8	0.941 897	27 612
SWTA-Shapley	0.011 9	0.999 951	293	0.010 1	0.999 916	656	0.010 4	0.999 885	174	0.008 6	0.999 772	1 035

4.3 实验结果总结

总的来说，本文所提的SWTA-Shapley方法在5种不同的场景中均展现出优异的性能，尤其是在后3种场景中，SWTA-Shapley都达到了最高的准确率和最少的计算时间。而作为以GTG-Shapley为基础的改进方法，SWTA-Shapley在所有实验设置下，都能取得比GTG-Shapley更好的性能，欧几里得距离上减少了6%~20%，余弦相似度上提高了20%~60%，也就是相较于GTG-Shapley接近1的程度提高了20%~60%，在计算时间上减少了30%~50%。本文使用分层组合的方法来选择用于效用评估的组合，很好地降低了计算复杂度，而使用加权截断来提高截断准确度和使用自适应规划来充分利用被截断部分，更是提高了计算的准确率。在所有方法中，TMC-Shapley由于仅使用了轮内截断，对计算时间的改善一直处于最差的水平，并且准确率也与SWTA-Shapley方法有较大的差距。TMR虽然在第1种场景中准确率超越过了SWTA-Shapley，但是需要花费更多的计算时间，相较来说性能上还是有所不如。OR方法仅使用一次聚合，虽然提高了计算速度，但是也对准确率造成了很大的影响，甚至在后3种分布中准确率与其他方法出现了较大差距。MR方法与OR相反，在每个计算轮次都会聚合，这提高了计算的准确率，使其可以在少部分情况下获得比SWTA-Shapley更高的准确率，但是也是因为每轮都需要聚合，计算时间是SWTA-Shapley的5 ~ 10倍，在实际应用时需要很大的计算成本。可以看出，SWTA-Shapley方法在计算时间与准确率之间取得了很好的平衡，性能上不仅超越了GTG-Shapley，也能在其他方法中达到最优，它的优越性能对应用Shapley值来评估联邦学习中参与者的贡献以及解决收益分配问题有很大的帮助。

4.4 消融实验

为了探究SWTA-Shapley方法各个组成部分对Shapley值计算结果的影响，本文进行了消融实验，实验设置与之前完全相同。本文将SWTA-Shapley方法的完整版本与以下的变体进行比较。

（1）WT-Shapley：该方法只保留SWTA-Shapley中的加权截断方法，排除分层组合和自适应规划的方法，依旧使用GTG-Shapley中的引导随机抽样，用于探究加权截断的优势。

（2）STA-Shapley：本文在该方法中不使用加权截断方法，只使用普通的截断，分层组合与自适应规划依旧保留，用于探究分层组合与自适应规划组合对性能的提升。

（3）SWT-Shapley：本文在该方法中保留分层组合与加权截断的方法，排除自适应规划的方法，用以探究自适应规划方法对计算准确率的提升。

这3种变体方法与GTG-Shapley以及SWTA-Shapley的实验结果见表7 ~ 11。变体算法在相同数据量大小和独立同分布场景中的实验结果见表7，变体算法在相同数据量大小和非独立同分布场景中的实验结果见表8，变体算法在不同数据量大小和独立同分布场景中的实验结果见表9，

表7 变体算法在相同数据量大小和独立同分布场景中的实验结果

SWTA变体算法	MNIST数据集						Fashion-MNIST数据集					
	5用户			10用户			5用户			10用户		
	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s
WT	0.012 4	0.999 957	296	0.010 9	0.999 926	1 375	0.013 2	0.999 962	231	0.010 6	0.999 929	1 365
STA	0.012 3	0.999 951	298	0.010 3	0.999 931	1 481	0.013 2	0.999 961	209	0.010 4	0.999 925	1 187
SWT	0.011 5	0.999 965	281	0.009 9	0.999 941	902	0.012 5	0.999 968	201	0.010 2	0.999 945	998
SWTA	0.011 2	0.999 972	288	0.009 7	0.999 946	935	0.012 3	0.999 971	206	0.010 7	0.999 949	1 026

表 8　变体算法在相同数据量大小和非独立同分布场景中的实验结果

SWTA 变体算法	MNIST 数据集						Fashion-MNIST 数据集					
	5 用户			10 用户			5 用户			10 用户		
	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s
WT	0.031 1	0.996 954	388	0.036 7	0.990 821	3 893	0.061 9	0.982 703	289	0.076 7	0.970 825	5 098
STA	0.030 4	0.997 179	363	0.035 4	0.992 556	3 982	0.062 3	0.982 756	212	0.078 7	0.970 628	3 418
SWT	0.030 1	0.997 409	344	0.034 9	0.994 312	2 276	0.060 3	0.983 013	209	0.075 4	0.971 259	2 497
SWTA	0.029 8	0.997 437	348	0.034 8	0.994 352	2 478	0.059 0	0.983 067	223	0.074 5	0.971 378	2 580

表 9　变体算法在不同数据量大小和独立同分布场景中的实验结果

SWTA 变体算法	MNIST 数据集						Fashion-MNIST 数据集					
	5 用户			10 用户			5 用户			10 用户		
	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s
WT	0.014 1	0.999 645	431	0.011 8	0.999 546	1 362	0.014 52	0.999 666	267	0.012 54	0.999 346	1 652
STA	0.013 9	0.999 672	396	0.011 7	0.999 558	1 203	0.014 45	0.999 603	199	0.012 49	0.999 337	980
SWT	0.013 3	0.999 748	361	0.114 3	0.999 595	712	0.014 27	0.999 705	203	0.012 03	0.999 638	1 002
SWTA	0.013 2	0.999 755	366	0.011 1	0.999 615	740	0.014 19	0.999 771	215	0.011 84	0.999 743	1 017

表 10　变体算法在标签噪声和相同数据量大小场景中的实验结果

SWTA 变体算法	MNIST 数据集						Fashion-MNIST 数据集					
	5 用户			10 用户			5 用户			10 用户		
	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s
WT	0.012 08	0.999 948	437	0.010 95	0.999 912	1 666	0.011 01	0.999 917	280	0.010 68	0.999 957	1 372
STA	0.012 19	0.999 941	423	0.010 98	0.999 918	8 820	0.011 08	0.999 911	198	0.010 81	0.999 959	992
SWT	0.011 53	0.999 954	303	0.010 72	0.999 938	2 720	0.010 91	0.999 921	202	0.010 34	0.999 962	901
SWTA	0.011 17	0.999 961	309	0.010 47	0.999 942	656	0.010 81	0.999 922	207	0.010 18	0.999 965	915

表 11　变体算法在特征噪声和相同数据量大小场景中的实验结果

SWTA 变体算法	MNIST 数据集						Fashion-MNIST 数据集					
	5 用户			10 用户			5 用户			10 用户		
	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s	欧几里得距离	余弦相似度	时间/s
WT	0.012 58	0.999 935	457	0.010 99	0.999 887	1 344	0.011 62	0.999 836	334	0.009 52	0.999 729	1 797
STA	0.012 09	0.999 933	419	0.010 52	0.999 888	989	0.011 03	0.999 853	308	0.009 28	0.999 699	1 529
SWT	0.012 26	0.999 948	397	0.010 61	0.999 902	912	0.011 37	0.999 862	209	0.008 92	0.999 745	1 392
SWTA	0.012 05	0.999 951	287	0.010 39	0.999 912	621	0.010 62	0.999 885	162	0.008 61	0.999 772	983

变体算法在标签噪声和相同数据量大小场景中的实验结果见表 10，变体算法在特征噪声和相同数据量大小场景中的实验结果见表 11。

4.4.1　分层组合的作用

根据表 7 ~ 11 中 WT-Shapley 和 SWT-Shapley 的实验结果对比可以看出，WT-Shapley 方法将原

本的引导随机抽样改为分层组合方法之后，在所有的实验设置中，计算需要的时间都有了明显的减少，准确率也有所提升，这得益于分层组合很好地避免了重复的组合评估以及重要组合丢失的问题。

4.4.2 加权截断的作用

从表 7 ~ 11 中 SWTA-Shapley 和 STA-Shapley 方法的实验结果对比可以看出，SWTA-Shapley 将截断机制重新改为 GTG-Shapley 中简化的截断机制之后，计算的准确率有所下降，这是因为加权截断可以避免一些错误的截断，原本的截断机制会造成错误截断导致误差增加。可以看出加权截断对于提高计算准确率有很重要的作用。

4.4.3 自适应规划的作用

提出自适应规划方法的目的是进一步强化本文所提算法，由表 7 ~ 11 中 SWTA-Shapley 和 SWT-Shapley 方法的实验结果对比可以看出，在增加了自适应规划后，SWTA-Shapley 的计算准确率相较于 SWT-Shapley 有一定的提升，并且计算时间几乎没有变化，自适应规划使本文所提算法的计算效率有了进一步的提升。

4.4.4 消融实验结果总结

消融实验的结果表明，分层组合可以减少计算时间，同时提高计算的精度。加权截断的方法提高了截断的准确性，相应地减少了最终计算结果的误差。而自适应规划的方法是在不增加计算时间的基础上提高了计算的准确度。SWTA-Shapley 中的这 3 种方法对于减少计算时间与提高计算准确率都十分重要。

5 结束语

本文以 GTG-Shapley 为基础提出了一种新的 Shapley 值近似算法 SWTA-Shapley，以解决当前对联邦学习中参与者贡献进行评估时计算成本过高的问题。GTG-Shapley 方法通过结合引导随机抽样、梯度重建子模型、截断 3 种机制很好地降低了 Shapley 值的计算复杂度，但随机抽样方法的不确定性和高抽样次数以及过于简化的截断机制会影响 Shapley 值的计算效率。本文对 GTG-Shapley 进行改进，提出了 SWTA-Shapley 算法，用分层组合的方式代替了引导随机抽样，避免了随机抽样时的不确定性，使需要考虑的组合数量更少且不会造成遗漏，保证了所有组合都能参与效用评估且不会重复。同时将截断机制优化为加权截断，提高了截断结果的准确率，为了进一步优化使用了自适应规划的方法，将被截断的部分充分利用，利用上一层级的效用平均值来近似被截断部分的效用值，不仅未增加计算时间，还提高了计算结果的准确率。这些改进很好地提高了 SWTA-Shapley 的计算效率。本文选取了 5 种不同场景、2 种数据集以及不同数量的联邦参与者，在多种设置下进行了大量的实验，结果表明，改进后的 SWTA-Shapley 在准确率和计算时间之间取得了很好的平衡，在多种应用场景中都取得了比 GTG-Shapley 更好的计算结果，同时在其他所有的比较方法中也取得了最优的性能。SWTA-Shapley 对降低 Shapley 值计算复杂度，提高准确率，解决现实中的贡献评估与收入分配问题有很大的帮助。

在未来的工作中，将探究如何进一步降低 SWTA-Shapley 方法的计算复杂度，减少计算成本，并且在更多应用场景中进行实验，根据实验结果作出改进。

参考文献：

[1] HEMSLEY M . Why the general data protection regulation is likely to disrupt core digital marketing channels in Europe[J]. Journal of Digital & Social Media Marketing, 2018, 6(2): 137-142.

[2] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[C]// Proceeding of the Int. Symp. on Artificial Intelligence and Statistics (AISTATS). 2017, arXiv: 1602. 05629.

[3] YANG Q, LIU Y, CHENG Y, et al. Federated learning[M]. Heidelberg: Springer, 2019.

[4] WANG T H, RAUSCH T, ZHANG C, et al. A principled approach to data valuation for federated learning[J]. Federated

Learning: Privacy and Incentive, 2020: 153-167.

[5] SHI Y Y, SONG H J, XU J. Responsible and Effective federated learning in financial services: a comprehensive survey[C] // Proceedings of the 2023 62nd IEEE Conference on Decision and Control (CDC). Singapore: IEEE Press, 2023: 4229-4236.

[6] LYU X C, HOU X Y, CHEN S, et al. Secure and efficient federated learning with provable performance guarantees via stochastic quantization[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 4070-4085.

[7] JIA R, DAO DAVID, WANG B X, et al. Towards efficient data valuation based on the Shapley value[C]//Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics. 2019, arXiv: 1902. 10275.

[8] SHAPLEY L S. A value for n-person games[J]. Contributions to the Theory of Games, 1953, 307–317.

[9] GUL F. Bargaining foundations of Shapley value [J]. Econometrica: Journal of the Econometric Society, 1989, 57(1): 81-95.

[10] AUMANN R J. Economic applications of the shapley value[M]// Game-Theoretic Methods in General Equilibrium Analysis. Dordrecht: Springer, 1994: 121-133.

[11] ALGABA E, FRAGNELLI V, SANCHEZ-SORIANO. Handbook of the shapley value[J]. Chapman and Hall/CRC, 2019.

[12] MICHALAK T P, RAHWAN T, JENNINGS N R, Szczepański P L, et al. Computational analysis of connectivity games with applications to the investigation of terrorist networks[C]//Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI '13). Palo Alto: AAAI Press, 2013: 293-301.

[13] LINDELAUF R, HAMERS H, HUSSLAGE B. Cooperative game theoretic centrality analysis of terrorist networks: the cases of Jemaah Islamiyah and al-Qaeda[J]. European Journal of Operational Research, 2013, 229(1): 230-238.

[14] PETROSJAN L, ZACCOUR G. Time-consistent Shapley value allocation of pollution cost reduction[J]. Journal of Economic Dynamics and Control, 2003, 27(3): 381-398.

[15] XU Z W, PENG Z X, YANG L, et al. An improved Shapley Value Method For A Green Supply Chain Income Distribution Mechanism[J]. International Journal of Environmental Research and Public Health, 2018, 15(9): 1976.

[16] LIAO Z L, ZHU X L. Case study on initial allocation of Shanghai carbon emission trading based on Shapley value[J]. Journal of Cleaner Production, 2015, 103: 338-344.

[17] COHEN S, RUPPIN E, DROR G. Feature selection based on the Shapley value[C]//Proceedings of the 19th International Joint Conference on Artificial Intelligence. 2005.

[18] ROZEMBERCZKI B, WATSON L, BAYER P, et al. The Shapley value in machine learning[J]. arXiv preprint, arXiv: 2202.05594, 2022.

[19] CHEN H, IAN C, SCOTT M, et al. Algorithms to estimate Shapley value feature attributions [J]. Nature Machine Intelligence, 2022, 5: 590-601.

[20] CASTRO J ,GÓMEZ D ,TEJADA J .Polynomial calculation of the Shapley value based on sampling[J].Computers and Operations Research, 2008, 36(5): 1726-1730.

[21] GHORBANI A ,ZOU Y J .Data Shapley: equitable valuation of data for machine learning[J]. CoRR, 2019, abs/1904. 02868.

[22] SONG T S, TONG Y X, WER S Y. Profit allocation for federated learning[C]//Proceedings of the 2019 IEEE International Conference on Big Data (Big Data). Piscataway: IEEE Press, 2019: 2577-2586.

[23] 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 (TIST), 2022, 13(4): 1-21.

[24] LECUN Y, BOTTOU L. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[作者简介]

鲍世豪（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。