Efficient Sampling Approaches to Shapley Value Approximation

JIAYAO ZHANG, Zhejiang University, China
QIHENG SUN, Zhejiang University, China
JINFEI LIU*, Zhejiang University; ZJU-Hangzhou Global Scientific and Technological Innovation Center, China
LI XIONG, Emory University, USA
JIAN PEI, Duke University, USA
KUI REN, Zhejiang University, China

Shapley value provides a unique way to fairly assess each player's contribution in a coalition and has enjoyed many applications. However, the exact computation of Shapley value is #P-hard due to the combinatoric nature of Shapley value. Many existing applications of Shapley value are based on Monte-Carlo approximation, which requires a large number of samples and the assessment of utility on many coalitions to reach high quality approximation, and thus is still far from being efficient. Can we achieve an efficient approximation of Shapley value by smartly obtaining samples? In this paper, we treat the sampling approach to Shapley value approximation as a stratified sampling problem. Our main technical contributions are a novel stratification design and two sample allocation methods based on Neyman allocation and empirical Bernstein bound, respectively. Experimental results on several real data sets and synthetic data sets demonstrate the effectiveness and efficiency of our novel stratification design and sampling approaches.

CCS Concepts: • **Information systems** → *Information integration*; *Mediators and data integration.*

Additional Key Words and Phrases: Shapley value; Sampling; Data Market

ACM Reference Format:
Jiayao Zhang, Qiheng Sun, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. 2023. Efficient Sampling Approaches to Shapley Value Approximation. *Proc. ACM Manag. Data* 1, 1, Article 48 (May 2023), 24 pages. https://doi.org/10.1145/3588728

1 INTRODUCTION

The well-celebrated Shapley value [40] is the unique metric for fair allocation of rewards to contributors based on their contribution towards a collective utility that satisfies all four desirable properties in fairness, including allocation efficiency, symmetry, zero element, and additivity. Shapley value is general and flexible to support various utility functions. Therefore, it has been extensively employed in many applications, such as machine learning model explanation [32], data/feature selection [14, 19], and data pricing in data markets [1, 9, 10, 28, 31]. For example, for

*the corresponding author.

Authors' addresses: Jiayao Zhang, Zhejiang University, China, jiayaozhang@zju.edu.cn; Qiheng Sun, Zhejiang University, China, qiheng_sun@zju.edu.cn; Jinfei Liu, Zhejiang University; ZJU-Hangzhou Global Scientific and Technological Innovation Center, China, jinfeiliu@zju.edu.cn; Li Xiong, Emory University, USA, lxiong@emory.edu; Jian Pei, Duke University, USA, j.pei@duke.edu; Kui Ren, Zhejiang University, China, kuiren@zju.edu.cn.

2836-6573/2023/5-ART48 $15.00
https://doi.org/10.1145/3588728

data valuation and pricing, each data record can be viewed as a player, and the utility function can be defined as the accuracy score of a machine learning model trained on the collective data.

Intuitively, the Shapley value of a player z is the expectation of the marginal utility contribution that z makes for a coalition of players, that is $\frac{\mathcal{U}(\mathcal{S}\cup\{z\})-\mathcal{U}(\mathcal{S})}{\binom{n-1}{|\mathcal{S}|}}$, where $\mathcal{S}$ is a coalition of players such that $z \notin \mathcal{S}$, $\mathcal{U}$ is a utility function, and n is the total number of players (see Section 3 for the concrete technical details). One major challenge of applying Shapley value is the prohibitive computational cost associated with exact computation, which in general involves evaluating the utility of an exponential number of coalitions and the corresponding marginal contributions $\mathcal{U}(\mathcal{S}\cup\{z\})-\mathcal{U}(\mathcal{S})$ and thus is #P-hard [11]. In many applications, such as data pricing, training and testing large-scale machine learning models for utility assessment is very costly. Consequently, using the exact Shapley value is impractical in many large-scale applications that involve many players.

Naturally many large-scale applications turn to approximate Shapley value. A series of sampling techniques have been proposed to efficiently estimate Shapley value [6–8, 19, 34]. Most of the existing methods mainly focus on sampling marginal contributions following the original definition of Shapley value. The proposed methods can be generally categorized based on their sampling mechanisms: simple random sampling and stratified random sampling. Simple random sampling is designed as sampling random permutations and computing average marginal contributions as the Shapley value [8, 19, 36]. Alternatively, stratified random sampling is designed as stratifying marginal contributions based on coalition cardinality and computing the expectation of the strata average marginal contributions as the Shapley value [6, 7, 34]. Still, those methods have to obtain a large number of samples and evaluate the utility of many coalitions in order to reach high quality approximation and thus is not efficient.

Since utility evaluation in many applications is costly, the major bottleneck of sampling based on marginal contributions is that one sample of marginal contributions $\mathcal{U}(\mathcal{S} \cup \{z\}) - \mathcal{U}(\mathcal{S})$ can only be used to update the Shapley value estimate for one player z, although coalition $\mathcal{S}$ may contain many other players.

Can we design a new sampling strategy that makes good use of the utility assessment of one coalition as much as possible? In this paper, we develop a novel stratification design based on a new notion, *complementary contribution*, defined as $\mathcal{U}(\mathcal{S}) - \mathcal{U}(\mathcal{N} \setminus \mathcal{S})$, where $\mathcal{N}$ is the set of all players. We show that the Shapley value is the expectation of weighted complementary contributions. One unique advantage is that a complementary contribution can be used to update the estimate of Shapley value for every player. Therefore, the number of samples and utility evaluation can be dramatically reduced to achieve a good approximation.

To further improve the effectiveness of sampling, we develop two methods that explore Neyman allocation [38] and the empirical Bernstein bound [3], respectively, to achieve better sample allocation, i.e., the number of samples to allocate for each stratum. Specifically, to minimize the estimated variance of Shapley value for a better approximation, we categorize complementary contributions and derive an optimum sample allocation scheme based on the variance of the strata following Neyman's approach [38]. Moreover, the variance of the strata is unobservable and therefore requires to be estimated. A sample allocation method has to incorporate the uncertainty of the estimated variance of strata inherently. Therefore, we design an online sample allocation method that selects an appropriate sample from a finite pool of samples during the sequential sampling process in the hope of gradually reducing estimation errors empirically. When a sample is drawn, the empirical Bernstein-Serfling inequality [3] is employed to evaluate the error bounds of stratified estimators as a guide for the next sampling. We then propose an algorithm to select the next appropriate sample in polynomial time. Our proposed methods are model-agnostic – they can approximate Shapley value in the general class of games with any utility functions. Our extensive experimental results on real and synthetic data sets show that the proposed algorithms outperform the baseline algorithms significantly, and the techniques designed for enhancing sample allocation can further improve the approximation performance.

Shapley value is important for many applications across different domains, so there is a rich body of studies on approximating Shapley value. The main novelty of the paper is that we propose a new notion of complementary contributions for the first time for reformulating and computing Shapley value, which allows reuse of the computation and hence enables drastic improvement of the sampling cost. Concretely, we summarize our contributions as follows.

- We propose a novel stratification design by sampling complementary contributions, which can dramatically reduce samples compared to marginal contributions.
- We further develop two sample allocation methods to improve the performance of the complementary contribution-based stratified sampling algorithm.
- Experiments on the cooperative games and data valuation tasks are conducted, which verify the efficiency and effectiveness of our proposed algorithms.

The rest of the paper is organized as follows. Section 2 reviews the related work on Shapley value and approximation. Section 3 discusses the preliminaries. We develop the novel notion of complementary contributions and the new stratification algorithm for Shapley value computation based on complementary contributions in Section 4. In Section 5 we present the sampling allocation methods based on the Neyman approach and the empirical Bernstein bound. We report the experimental results and findings in Section 6. Finally, we conclude the paper in Section 7.

2 RELATED WORK

Shapley value [40] has an incredible impact on the cooperative game theory, which has been applied in tackling many problems, such as terrorist network [29], profit allocation [41], query answering [12], data/feature selection [14, 19], and data pricing [1, 9, 10, 28, 31, 43].

Computing the exact Shapley value was proved to be #P-hard [11]. To address the challenge, several techniques [6–8, 34, 36] were developed to approximate Shapley value. Castro et al. [8] presented a permutation sampling method that estimates Shapley value as the expectation of marginal contributions. Mitchell et al. [36] improved the permutation sampling via Quasi Monte Carlo techniques. Maleki et al. [34] provided a stratified sampling algorithm that relies on an assumption about the range of utilities and gives the sample size of each stratum based on the Hoeffding bound [21], which was improved by Castro et al. [7]. Burgess and Chapman [6] provided a stratified sampling algorithm that takes an assumption about the sample variance and sequentially chooses strata to sample based on an empirical bound.

Shapley value has recently been used to quantify the contributions of data points towards training machine learning models [1, 16–20, 23, 24, 26, 28]. The performance of a model trained using a subset of the training data and tested on another test set is often used as the utility function. Ghorbani and Zou [19] proposed two algorithms to accelerate the estimation of Shapley value in this context. The first method truncates the calculation of the near-zero marginal contributions, since the change in performance by adding one more training data point becomes smaller and smaller as data increases. The second method updates the model by performing gradient descent on one data point at a time to approximate marginal contributions. Jia et al. [23] focused on one family of models relying on k-nearest neighbors, which are lazy models, and developed an algorithm based on Locality Sensitive Hashing with sublinear complexity. Ghorbani et al. [17] proposed distributional Shapley value to measure the value of data points where the dataset is drawn in an independent and identically distributed (i.i.d.) manner from the underlying distribution. On the basis of this work, Kwon et al. [26] derived the analytic expressions for distributional Shapley for

Table 1. Some frequently used notations.

Notation	Definition
n	the number of players
$\mathcal{U}(\cdot)$	utility function
$\mathcal{N}$	a set of n players $\mathcal{N} = \{z_1, \ldots, z_n\}$
z_i	the i^{th} player
$\mathcal{S}$	a coalition
m	the total number of samples
$\mathcal{SV}_i$	Shapley value of z_i
$\overline{\mathcal{SV}_i}$	approximate Shapley value of z_i
$\mathcal{SV}_{i,j}$	the expected complementary contributions of (z_i, j)-coalitions
$\overline{\mathcal{SV}_{i,j}}$	the estimation of $\mathcal{SV}_{i,j}$

linear regression, binary classification, and non-parametric density estimation. Ghorbani et al. [18] used Shapley value for annotation in batch active learning. They used Shapley values computed on labeled data points to train a regression model that predicts Shapley values for unlabeled data points. Ghorbani and Zou [20] applied Shapley value to identify responsible neurons and developed a multi-armed bandit algorithm to explore neurons with high Shapley value.

The above studies design algorithms based on sampling marginal contributions following the original definition of Shapley value, whereas in this paper we focus on developing novel and much more efficient algorithms based on sampling complementary contributions with great potential in reducing computational costs.

3 PRELIMINARIES

In this section, we review the notion of Shapley value and the classical approximation method. Table 1 summarizes some frequently used notations.

Consider a set of n players $\mathcal{N} = \{z_1, \ldots, z_n\}$. A *coalition* is a subset of players $\mathcal{S} \subseteq \mathcal{N}$ that cooperate to complete a task. We assume a utility function $\mathcal{U}(\mathcal{S})$ ($\mathcal{S} \subseteq \mathcal{N}$) that evaluates the utility of a coalition $\mathcal{S}$ for a task. The *marginal contribution* of z_i with respect to a coalition $\mathcal{S}$ is $\mathcal{U}(\mathcal{S} \cup \{z_i\}) - \mathcal{U}(\mathcal{S})$.

Shapley [40] laid out the fundamental requirements of fair reward allocation, including balance, symmetry, zero element, and additivity. Specifically, *balance* (also known as *efficiency*) requires that the total payoff should be fully distributed to all players. *Symmetry* specifies that two players should receive the same reward if they have the same marginal contributions. *Additivity* indicates that the reward value on two tasks should be the sum of the values on individual tasks. *Zero element* specifies that a player should not be rewarded anything if the player does not make any marginal contribution.

Shapley value measures the expectation of marginal contribution by z_i in all possible coalitions. That is,

$$\mathcal{SV}_i = \frac{1}{n} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{z_i\}} \frac{\mathcal{U}(\mathcal{S} \cup \{z_i\}) - \mathcal{U}(\mathcal{S})}{\binom{n-1}{|\mathcal{S}|}} \tag{1}$$

Shapley value is the only existing measure that satisfies all the four fundamental requirements.

Computing the exact Shapley value has to enumerate all the subsets of players as all possible coalitions and thus is prohibitively expensive. The Monte Carlo simulation method [8] is commonly

Algorithm 1: Monte Carlo Shapley value computation.

input : players $\mathcal{N} = \{z_1, \ldots, z_n\}$ and $m > 0$
output: approximate Shapley value $\overline{\mathcal{SV}}_i$ for each player z_i $(1 \leq i \leq n)$
1 $\overline{\mathcal{SV}}_i \leftarrow 0$ $(1 \leq i \leq n)$;
2 **for** *k=1 to m* **do**
3   let π^k be a random permutation of $\{1, \ldots, n\}$;
4   **for** *i=1 to n* **do**
5     $\mathcal{SV}(z_{\pi^k(i)}) = \mathcal{U}(\{z_{\pi^k(1)}, \ldots, z_{\pi^k(i)}\}) - \mathcal{U}(\{z_{\pi^k(1)}, \ldots, z_{\pi^k(i-1)}\})$;
6     $\overline{\mathcal{SV}}_{\pi^k(i)} += \mathcal{SV}(z_{\pi^k(i)})$;
7 **for** *i=1 to n* **do**
8   $\overline{\mathcal{SV}}_i = \overline{\mathcal{SV}}_i / m$;
9 **return** $\overline{\mathcal{SV}}_1, \ldots, \overline{\mathcal{SV}}_n$.

used to compute the approximate Shapley value. The pseudo-code of the Monte Carlo method is shown in Algorithm 1. It samples random permutations of players, scans each sample permutation, and calculates the marginal contribution of every player in the order of the permutation (Lines 3-6). By examining a sufficiently large set of sample permutations, the final estimation of Shapley value is the average of all the calculated marginal contributions in the samples (Lines 7-8). This Monte Carlo simulation gives an unbiased estimation of the Shapley value. In practice, we can conduct Monte Carlo simulation iteratively until the average empirically converges. The larger the number of sample permutations, the smaller error bound between the computed Shapley value and the exact Shapley value. The estimation quality is established by the following result [33].

Theorem 3.1 (Monte Carlo marginal contribution approximation quality [33]). *According to Hoeffding's inequality, given the range r of the utility function, an error bound ϵ, and a confidence level $1 - \delta$, if the sample size of marginal contributions, i.e., the number of permutations, satisfies $m \geq \frac{2r^2 \log 2/\delta}{\epsilon^2}$, then $P(|\overline{\mathcal{SV}}_i - \mathcal{SV}_i| \geq \epsilon) \leq \delta$.*

In Algorithm 1 and any Shapley value approximation algorithms based on sampling marginal contributions, one sample of marginal contributions $\mathcal{U}(\mathcal{S} \cup \{z_i\}) - \mathcal{U}(\mathcal{S})$ can only be used to update the Shapley value estimate for one player z, although coalition $\mathcal{S}$ may contain many other players. As the evaluation of utility function is often costly in many applications, such as building machine learning models, the limitation that one marginal contribution can only be used by one player becomes the efficiency bottleneck.

4 SHAPLEY VALUE COMPUTATION BASED ON COMPLEMENTARY CONTRIBUTIONS

To tackle the efficiency bottleneck in Shapley value approximation based on marginal contributions, in this section we develop a novel approach based on complementary contribution. We first describe the definition of complementary contribution and discuss the related properties in Section 4.1 and then present a Shapley value computation algorithm based on sampling complementary contributions in Section 4.2.

4.1 Complementary Contributions versus Marginal Contributions

Shapley value is a weighted sum of marginal contributions, a kind of utility difference. Our observation is that the utility weights of each pair of complementary coalitions are opposite in the

Shapley value formula, thus intuitively the Shapley value can be reformulated based on complementary coalitions by regrouping utilities. In this section, we introduce the notion of complementary contribution and show that Shapley value can be computed using complementary contributions. Then, we compare complementary contributions and marginal contributions in Shapley value computation.

Definition 4.1. Given a set of players $\mathcal{N} = \{z_1, \ldots, z_n\}$ and a coalition $\mathcal{S} \subseteq \mathcal{N}$, the **complementary contribution** of $\mathcal{S}$ is

$$CC_{\mathcal{N}}(\mathcal{S}) = \mathcal{U}(\mathcal{S}) - \mathcal{U}(\mathcal{N} \setminus \mathcal{S}).$$

When $\mathcal{N}$ is clear from context, we also write $CC_{\mathcal{N}}(\mathcal{S})$ as $CC(\mathcal{S})$.

Shapley value can be computed using complementary contributions.

Theorem 4.2. *Given a set of players $\mathcal{N} = \{z_1, \ldots, z_n\}$, the Shapley value of z_i $(1 \leq i \leq n)$ is*

$$\mathcal{SV}_i = \frac{1}{n} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{z_i\}} \frac{CC_{\mathcal{N}}(\mathcal{S} \cup \{z_i\})}{\binom{n-1}{|\mathcal{S}|}}. \tag{2}$$

Proof. We rewrite Equation 1 to

$$\begin{aligned}
\mathcal{SV}_i &= \frac{1}{n} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{z_i\}} \frac{\mathcal{U}(\mathcal{S} \cup \{z_i\}) - \mathcal{U}(\mathcal{S})}{\binom{n-1}{|\mathcal{S}|}} \\
&= \frac{1}{n} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{z_i\}} \frac{\mathcal{U}(\mathcal{S} \cup \{z_i\})}{\binom{n-1}{|\mathcal{S}|}} - \frac{1}{n} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{z_i\}} \frac{\mathcal{U}(\mathcal{S})}{\binom{n-1}{|\mathcal{S}|}}.
\end{aligned}$$

Let $\mathcal{S}' = (\mathcal{N} \setminus \{z_i\}) \setminus \mathcal{S}$, that is, $\mathcal{S} = (\mathcal{N} \setminus \{z_i\}) \setminus \mathcal{S}'$. We have

$$\begin{aligned}
\mathcal{SV}_i &= \frac{1}{n} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{z_i\}} \frac{\mathcal{U}(\mathcal{S} \cup \{z_i\})}{\binom{n-1}{|\mathcal{S}|}} - \frac{1}{n} \sum_{(\mathcal{N} \setminus \{z_i\}) \setminus \mathcal{S}' \subseteq \mathcal{N} \setminus \{z_i\}} \frac{\mathcal{U}((\mathcal{N} \setminus \{z_i\}) \setminus \mathcal{S}')}{\binom{n-1}{|\mathcal{N} \setminus \mathcal{S}'|-1}} \\
&= \frac{1}{n} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{z_i\}} \frac{\mathcal{U}(\mathcal{S} \cup \{z_i\})}{\binom{n-1}{|\mathcal{S}|}} - \frac{1}{n} \sum_{\mathcal{S}' \subseteq \mathcal{N} \setminus \{z_i\}} \frac{\mathcal{U}((\mathcal{N} \setminus \{z_i\}) \setminus \mathcal{S}')}{\binom{n-1}{|\mathcal{N} \setminus \mathcal{S}'|-1}}.
\end{aligned}$$

In the second term of the above, rename variable $\mathcal{S}'$ to $\mathcal{S}$. Moreover, $\binom{n-1}{|\mathcal{N} \setminus \mathcal{S}'|-1} = \binom{n-1}{|\mathcal{S}'|}$, since $|\mathcal{N} \setminus \mathcal{S}'| - 1 + |\mathcal{S}'| = n - 1$. We have

$$\begin{aligned}
\mathcal{SV}_i &= \frac{1}{n} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{z_i\}} \frac{\mathcal{U}(\mathcal{S} \cup \{z_i\})}{\binom{n-1}{|\mathcal{S}|}} - \frac{1}{n} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{z_i\}} \frac{\mathcal{U}(\mathcal{N} \setminus (\mathcal{S} \cup \{z_i\}))}{\binom{n-1}{|\mathcal{S}|}} \\
&= \frac{1}{n} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{z_i\}} \frac{\mathcal{U}(\mathcal{S} \cup \{z_i\}) - \mathcal{U}(\mathcal{N} \setminus (\mathcal{S} \cup \{z_i\}))}{\binom{n-1}{|\mathcal{S}|}} \\
&= \frac{1}{n} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{z_i\}} \frac{CC_{\mathcal{N}}(\mathcal{S} \cup \{z_i\})}{\binom{n-1}{|\mathcal{S}|}}.
\end{aligned}$$

□

Shapley value can be computed using marginal contributions (Equation 1) or complementary contributions (Equation 2). Are there any differences?

As analyzed at the end of Section 3, each marginal contribution $\mathcal{U}(\mathcal{S} \cup \{z_i\}) - \mathcal{U}(\mathcal{S})$ can be used only for player z_i in Equation 1. However, each complementary contribution $\mathcal{U}(\mathcal{S}) - \mathcal{U}(\mathcal{N} \setminus \mathcal{S})$ can be used for all players $z_j \in \mathcal{S}$ in computing their Shapley values using Equation 2. Moreover, complementary contribution $\mathcal{U}(\mathcal{N} \setminus \mathcal{S}) - \mathcal{U}(\mathcal{S}) = -(\mathcal{U}(\mathcal{S}) - \mathcal{U}(\mathcal{N} \setminus \mathcal{S}))$ can be used in computing the Shapley values for players $z_j \in \mathcal{N} \setminus \mathcal{S}$. In other words, a complementary contribution $\mathcal{U}(\mathcal{S}) - \mathcal{U}(\mathcal{N} \setminus \mathcal{S})$ can be used by every player in its Shapley value computation.

When there are many players or the evaluation of coalition utility is expensive, we have to conduct a Monte Carlo approach to approximate Shapley value based on a sample of utility values. The fact that a complementary contribution can be used in computing the Shapley value of every player provides a promising way to estimate Shapley value more efficiently.

One may think that it may be possible to memorize and maximally reuse utility computations when computing Shapley value based on marginal contributions. Below we analyze why this is not feasible. When using complementary contributions, each sample of complementary contributions and its corresponding coalition utilities are used n times, which achieves the sample utilization maximization. To maximize sample utilization when using marginal contributions, each sampled coalition utility $\mathcal{U}(\mathcal{S})$ should be memorized and (re)used for all $z_i \in \mathcal{N}$ to compute marginal contributions, which are $\{\mathcal{U}(\mathcal{S} \cup \{z_i\}) - \mathcal{U}(\mathcal{S}) | z_i \notin \mathcal{S}\} \cup \{\mathcal{U}(\mathcal{S}) - \mathcal{U}(\mathcal{S} \setminus \{z_i\}) | z_i \in \mathcal{S}\}$ $(1 \le i \le n)$. Let $\mathcal{A} = \{\mathcal{S}_1, \ldots, \mathcal{S}_\tau\}$ denote τ sampled coalitions. Given $\mathcal{S}_j \in \mathcal{A}$, to fully utilize $\mathcal{S}_j$ for all $z_i \in \mathcal{N}$, when $z_i \in \mathcal{S}_j$, $\mathcal{S}_j$ is used to construct marginal contribution of $\mathcal{U}(\mathcal{S}_j) - \mathcal{U}(\mathcal{S}_j \setminus \{z_i\})$ and thus $\mathcal{S}_j \setminus \{z_i\}$ needs to be sampled and belongs to $\mathcal{A}$, which implies any subset of $\mathcal{S}_j$ belongs to $\mathcal{A}$; when $z_i \notin \mathcal{S}_j$, $\mathcal{S}_j$ is used to construct marginal contribution of $\mathcal{U}(\mathcal{S}_j \cup \{z_i\}) - \mathcal{U}(\mathcal{S}_j)$ and thus $\mathcal{S}_j \cup \{z_i\} \in \mathcal{A}$, which implies any superset of $\mathcal{S}_j$ belongs to $\mathcal{A}$, and thus $\mathcal{N} \in \mathcal{A}$. Since any subset of $\mathcal{S}_j$ belongs to $\mathcal{A}$ for $\mathcal{S}_j \in \mathcal{A}$, we can infer that any subset of $\mathcal{N}$ belongs to $\mathcal{A}$. Therefore, a sampling method based on marginal contributions can maximize the sample utilization if and only if all coalition utilities are memorized. However, this is not feasible because the total number of coalition utilities is 2^n, growing exponentially.

4.2 Computing Shapley Value Using Complementary Contributions

In this section, we develop an algorithm to compute the approximate Shapley value based on sampling complementary contributions.

Definition 4.3. Given a set of players $\mathcal{N} = \{z_1, \ldots, z_n\}$, a coalition of j players $(1 \le j \le n)$ is called a j-**coalition**. Moreover, for a player z_i $(1 \le i \le n)$, a j-coalition that contains z_i is called a (z_i, j)-**coalition**. Denote by $\mathfrak{S}^{i,j} = \{\mathcal{S} \cup \{z_i\} | \mathcal{S} \subseteq \mathcal{N} \setminus \{z_i\}, |\mathcal{S}| = j-1\}$ $(1 \le j \le n)$ the set of (z_i, j)-coalitions, and by $\mathcal{SV}_{i,j}$ **the expected complementary contributions of** (z_i, j)-coalitions. That is,

$$\mathcal{SV}_{i,j} = \sum_{\mathcal{S} \in \mathfrak{S}^{i,j}} \frac{CC_{\mathcal{N}}(\mathcal{S})}{\binom{n-1}{j-1}} = \sum_{\mathcal{S} \in \mathfrak{S}^{i,j}} \frac{\mathcal{U}(\mathcal{S}) - \mathcal{U}(N \setminus \mathcal{S})}{\binom{n-1}{j-1}}. \tag{3}$$

Using Theorem 4.2 and Definition 4.3, we have the following immediately.

Corollary 4.4. *Given a set of players $\mathcal{N} = \{z_1, \ldots, z_n\}$, the Shapley value of z_i $(1 \le i \le n)$ is $\mathcal{SV}_i = \frac{1}{n} \sum_{j=1}^{n} \mathcal{SV}_{i,j}$.*

According to Corollary 4.4, approximating the Shapley value of $z_i \in \mathcal{N}$ $(1 \le i \le n)$ can be reformulated as a stratified sampling process of all complementary contributions containing z_i. The stratification design is to divide all complementary contributions into n strata such that the j-th stratum contains all (z_i, j)-coalitions. Then, to approximate $\mathcal{SV}_i$ $(1 \le i \le n)$, we can first estimate $\mathcal{SV}_{i,j}$ $(1 \le j \le n)$ by sampling with replacement. Let $CC_{\mathcal{N}}^{i,j}$ be a random variable with uniform distribution on the set $\{CC_{\mathcal{N}}(\mathcal{S}) | \mathcal{S} \in \mathfrak{S}^{i,j}\}$. The expectation of $CC_{\mathcal{N}}^{i,j}$ is $\mathcal{SV}_{i,j}$. Given a random sample of $CC_{\mathcal{N}}^{i,j}$ of size $m_{i,j}$ $\{CC_{\mathcal{N}}(\mathcal{S}_1), \ldots, CC_{\mathcal{N}}(\mathcal{S}_{m_{i,j}})\}$, where $\mathcal{S}_1, \ldots, \mathcal{S}_{m_{i,j}} \in \mathfrak{S}^{i,j}$, the mean over the sample is $\overline{\mathcal{SV}_{i,j}} = \frac{1}{m_{i,j}} \sum_{k=1}^{m_{i,j}} CC_{\mathcal{N}}(\mathcal{S}_k)$, which is an estimation of $\mathcal{SV}_{i,j}$. Then, an estimation of $\mathcal{SV}_i$ is $\overline{\mathcal{SV}_i} = \frac{1}{n} \sum_{j=1}^{n} \overline{\mathcal{SV}_{i,j}}$.

Algorithm 2: Shapley value computation based on complementary contributions.

input : players $\mathcal{N} = \{z_1, \ldots, z_n\}$ and $m > 0$

output: approximate Shapley value $\overline{\mathcal{SV}}_i$ for each player z_i $(1 \le i \le n)$

1 $\overline{\mathcal{SV}}_i \leftarrow 0$ $(1 \le i \le n)$; $\overline{\mathcal{SV}}_{i,j}, m_{i,j} \leftarrow 0$ $(1 \le i, j \le n)$;

2 **for** *k=1 to m* **do**

3   let π^k be a random permutation of $\{1, \ldots, n\}$;

4   let i be a random value drawn from $\{1, \ldots, n\}$;

5     $\mathcal{S} \leftarrow \{z_{\pi^k(1)}, \ldots, z_{\pi^k(i)}\}$;

6     $\mathcal{N} \setminus \mathcal{S} \leftarrow \{z_{\pi^k(i+1)}, \ldots, z_{\pi^k(n)}\}$;

7     $u \leftarrow \mathcal{U}(\mathcal{S}) - \mathcal{U}(\mathcal{N} \setminus \mathcal{S})$;

8   **for** *j= 1 to i* **do**

9         $\overline{\mathcal{SV}}_{\pi^k(j),i} += u$;

10         $m_{\pi^k(j),i} += 1$;

11   **for** *j=i+1 to n* **do**

12         $\overline{\mathcal{SV}}_{\pi^k(j),n-i} -= u$;

13         $m_{\pi^k(j),n-i} += 1$;

14 **for** *i=1 to n* **do**

15     $\overline{\mathcal{SV}}_i = \frac{1}{n} \sum_{j=1}^{n} \overline{\mathcal{SV}}_{i,j} / m_{i,j}$;

16 **return** $\overline{\mathcal{SV}}_1, \ldots, \overline{\mathcal{SV}}_n$.

The detailed algorithm is shown in Algorithm 2. We first randomly generate a pair of complementary coalitions $\mathcal{S}$ and $\mathcal{N} \setminus \mathcal{S}$, calculate the complementary contribution u, assign the value u to n players in $\mathcal{N}$, and update the corresponding counts of $m_{i,j}$ (Lines 3-13). By drawing m samples of complementary contributions, the final estimation of the Shapley value is the average of complementary contribution means (Lines 14-15).

Now we show that the estimated Shapley value $\overline{\mathcal{SV}}_i$ in Algorithm 2 is unbiased.

Theorem 4.5. *Given a set of players $\mathcal{N} = \{z_1, \ldots, z_n\}$, Algorithm 2 gives an unbiased estimation of Shapley value for every player, that is, $E[\overline{\mathcal{SV}}_i] = \mathcal{SV}_i$ $(1 \le i \le n)$.*

Proof. Denote by $CC_{\mathcal{N}}(\mathcal{S}_1), \ldots, CC_{\mathcal{N}}(\mathcal{S}_{m_{i,j}})$ a sample of $CC_{\mathcal{N}}^{i,j}$ $(1 \le i, j \le n)$ drawn by Algorithm 2. The expectation of the sample $\overline{\mathcal{SV}}_{i,j} = \frac{1}{m_{i,j}} \sum_{k=1}^{m_{i,j}} CC_{\mathcal{N}}(\mathcal{S}_k)$. We have

$$E[\overline{\mathcal{SV}}_{i,j}] = E[\frac{1}{m_{i,j}} \sum_{k=1}^{m_{i,j}} CC_{\mathcal{N}}(\mathcal{S}_k)] = \frac{1}{m_{i,j}} \sum_{k=1}^{m_{i,j}} E[CC_{\mathcal{N}}(\mathcal{S}_k)]$$

According to Equation 3, $E[CC_{\mathcal{N}}(\mathcal{S}_k)] = \mathcal{SV}_{i,j}$. Thus, $E[\overline{\mathcal{SV}}_{i,j}] = \mathcal{SV}_{i,j}$.

Now, consider the estimate $\overline{\mathcal{SV}}_i$ produced by Algorithm 2. We have

$$E[\overline{\mathcal{SV}}_i] = E[\frac{1}{n} \sum_{j=1}^{n} \overline{\mathcal{SV}}_{i,j}] = \frac{1}{n} \sum_{j=1}^{n} E[\overline{\mathcal{SV}}_{i,j}] = \frac{1}{n} \sum_{j=1}^{n} \mathcal{SV}_{i,j} = \mathcal{SV}_i.$$

That is, $\overline{\mathcal{SV}}_i$ is an unbiased estimation of $\mathcal{SV}_i$. □

5 SAMPLE ALLOCATION

In stratified sampling, it is important to allocate samples to strata properly. We allocate the sample size in each stratum uniformly in Algorithm 2. In principle, we should allocate larger sample sizes to the larger or more variable strata. In this section, to strengthen the efficiency in our stratified sampling approach to Shapley value approximation using complementary contributions, we incorporate the Neyman allocation in Section 5.1 to better allocate samples to each stratum and the empirical Bernstein bound in Section 5.2, which can dynamically collect samples based on previous sampling results.

5.1 Allocation Based on the Neyman Approach

In Section 4.2, complementary contributions $CC(\mathcal{S})$ are naturally stratified into n strata $\mathfrak{S}^{i,1}, \ldots, \mathfrak{S}^{i,n}$ according to the coalition size. We need to estimate the expectation of complementary contribution $\mathcal{SV}_{i,j}$ in every stratum $\mathfrak{S}^{i,j}$ $(1 \le j \le n)$. How should we allocate sample sizes to these estimators? To address the problem, we develop an approach following Neyman allocation [38]. Neyman allocation is the optimal allocation that allocates samples to strata and minimizes the sample variance of the estimator.

We start from the relationship between the variance of $\overline{\mathcal{SV}_i}$ and the sample size of $\mathfrak{S}^{i,j}$ $(1 \le j \le n)$.

Lemma 5.1. *Given a set of players $\mathcal{N} = \{z_1, \ldots, z_n\}$, for player z_i $(1 \le i \le n)$, the variance of $\overline{\mathcal{SV}_i}$ is $Var[\overline{\mathcal{SV}_i}] = \frac{1}{n^2}\sum_{j=1}^{n}\frac{\sigma_{i,j}^2}{m_{i,j}}$, where $\sigma_{i,j}^2$ is the variance of random variable $CC_{\mathcal{N}}^{i,j}$ and $m_{i,j}$ is the sample size assigned to stratum $\mathfrak{S}^{i,j}$ $(1 \le j \le n)$.*

Proof. For a sample of $m_{i,j}$ j-coalitions $\{\mathcal{S}_1, \ldots, \mathcal{S}_{m_{i,j}}\} \subseteq \mathfrak{S}^{i,j}$ $(1 \le j \le n)$, we obtain a sample of $m_{i,j}$ complementary contributions $\{CC_{\mathcal{N}}(\mathcal{S}_1), \ldots, CC_{\mathcal{N}}(\mathcal{S}_{m_{i,j}})\}$. From the sample we can estimate random variable $CC_{\mathcal{N}}^{i,j}$. The expectation of the sample $\overline{\mathcal{SV}_{i,j}} = \frac{1}{m_{i,j}}\sum_{k=1}^{m_{i,j}} CC_{\mathcal{N}}^{i,j}(\mathcal{S}_k)$.

Thus, we have

$$Var[\overline{\mathcal{SV}_{i,j}}] = Var[\frac{1}{m_{i,j}}\sum_{k=1}^{m_{i,j}} CC_{\mathcal{N}}^{i,j}(\mathcal{S}_k)] = \frac{1}{m_{i,j}^2}\sum_{k=1}^{m_{i,j}} Var[CC_{\mathcal{N}}^{i,j}(\mathcal{S}_k)] = \frac{\sigma_{i,j}^2}{m_{i,j}}.$$

Thus,

$$Var[\overline{\mathcal{SV}_i}] = Var[\frac{1}{n}\sum_{j=1}^{n}\overline{\mathcal{SV}_{i,j}}] = \frac{1}{n^2}\sum_{j=1}^{n} Var[\overline{\mathcal{SV}_{i,j}}] = \frac{1}{n^2}\sum_{j=1}^{n}\frac{\sigma_{i,j}^2}{m_{i,j}}.$$

□

According to Lemma 5.1, the sum of variance of approximate Shapley value is given by $\sum_{i=1}^{n} Var[\overline{\mathcal{SV}_i}] = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{\sigma_{i,j}^2}{m_{i,j}}$, which is under the effect of the sample size $m_{i,j}$ and variance $\sigma_{i,j}^2$ $(1 \le i, j \le n)$. Since a sample of complementary contributions can be used in computing the Shapley value of all players, it is difficult to find the sampling scheme given the required sample size $m_{i,j}$ to minimize $\sum_{i=1}^{n} Var[\overline{\mathcal{SV}_i}]$. In order to overcome the difficulty, we seek to approximately solve the problem by relaxing the exact sample size $m_{i,j}$ to the expected sample size $E[m_{i,j}]$.

Denote by $\mathfrak{S}^j = \{\mathcal{S} | \mathcal{S} \subseteq \mathcal{N}, |\mathcal{S}| = j\}$ the set of j-coalitions $(1 \le j \le n)$. We draw samples from $\mathfrak{S}^j$ independently. Let m_j be the sample size in $\mathfrak{S}^j$ $(\lceil n/2 \rceil \le j \le n)$. After drawing a coalition $\mathcal{S}$ from $\mathfrak{S}^j$, we can estimate the complementary contribution $CC_{\mathcal{N}}(\mathcal{S})$, which can be used in $\mathcal{SV}_{i,j}$ for z_i in $\mathcal{S}$ and $\mathcal{SV}_{i,n-j}$ for z_i in $\mathcal{N} \setminus \mathcal{S}$. The probability that a sample belongs to $CC_{\mathcal{N}}^{i,j}$ is the probability that z_i belongs to the sampled j-coalition, i.e., $\frac{j}{n}$ $(1 \le i, j \le n)$. Thus, it is easy to see that the expected sample size of $CC_{\mathcal{N}}^{i,j}$ is $E[m_{i,j}] = \frac{j}{n} m_{\max\{j,n-j\}}$ $(1 \leq i, j \leq n)$ after drawing $m_{\max\{j,n-j\}}$ samples. Considering the sum of variance of approximate Shapley value when the sample size of $\mathfrak{S}^{i,j}$ is $\frac{j}{n} m_{\max\{j,n-j\}}$ $(1 \leq i, j \leq n)$, we have:

$$\sum_{i=1}^{n} Var[\overline{\mathcal{SV}_i}] = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\sigma_{i,j}^2 \cdot \frac{n}{j}}{m_{\max\{j,n-j\}}} = \frac{1}{n} \sum_{j=\lceil n/2 \rceil}^{n} \frac{\sum_{i=1}^{n} (\frac{\sigma_{i,j}^2}{j} + \frac{\sigma_{i,n-j}^2}{n-j})}{m_j}.$$

Given the total number of samples m, the relaxed variance minimization problem is formulated as follows.

$$\min \sum_{j=\lceil n/2 \rceil}^{n} \frac{\sum_{i=1}^{n} (\frac{\sigma_{i,j}^2}{j} + \frac{\sigma_{i,n-j}^2}{n-j})}{m_j},$$
$$\text{s.t.} \sum_{j=\lceil n/2 \rceil}^{n} m_j = m.$$

Using the method of Lagrange multipliers, we can get:

$$m_j = \frac{m \sqrt{\sum_{i=1}^{n} (\frac{\sigma_{i,j}^2}{j} + \frac{\sigma_{i,n-j}^2}{n-j})}}{\sum_{j=\lceil n/2 \rceil}^{n} \sqrt{\sum_{i=1}^{n} (\frac{\sigma_{i,j}^2}{j} + \frac{\sigma_{i,n-j}^2}{n-j})}}. \quad (4)$$

Since the sample allocation method shown in Equation 4 depends on the unobservable true variance, we use the unbiased sample variance instead. We divide the Shapley value computation process into two stages and correspondingly divide the samples into two parts. The pseudo-code is given in Algorithm 3.

In the first stage (Lines 2-11), we sample at least m_{init} samples for $\mathcal{SV}_{i,j}$. We then compute unbiased estimations of $\sigma_{i,j}^2$ using Bessel's correction based on samples collected in the first stage (Line 12). Let $CC_{\mathcal{N}}(\mathcal{S}_1 \cup \{z_i\}), \ldots, CC_{\mathcal{N}}(\mathcal{S}_{m_{i,j}} \cup \{z_i\})$ be $m_{i,j}$ samples for computing $\overline{\mathcal{SV}_{i,j}}$, then $\widehat{\sigma_{i,j}^2} = \frac{1}{m_{i,j}-1} \sum_{k=1}^{m_{i,j}} (CC_{\mathcal{N}}(\mathcal{S}_k \cup \{z_i\}) - \frac{1}{m_{i,j}} \sum_{k=1}^{m_{i,j}} CC_{\mathcal{N}}(\mathcal{S}_k \cup \{z_i\}))$. Let m_{first} be the number of samples used in the first stage, the number of remaining samples is $m - m_{first}$. We calculate m_j according to Equation 4 using the unbiased sample variance $\widehat{\sigma_{i,j}^2}$ (Lines 13-14). In the second stage (Lines 15-22), we complete the remaining sampling according to m_j. The final estimation of Shapley value is the average of all complementary contribution means (Lines 23-24).

5.2 Allocation Based on the Empirical Bernstein Bound

The algorithm based on the Neyman approach aims to minimize the sample variance using a sample allocation scheme, but the two-stage sampling is heavily controlled by the results of the first-stage sampling, which limits the potential of dynamic updates. In this section, we adopt the empirical Bernstein-Serfling inequality [3] to design a dynamic sample allocation method based on the sample variance in order to compute the approximate Shapley value more effectively in practice.

The precision of $\overline{\mathcal{SV}_i}$ $(1 \leq i \leq n)$ crucially depends on the estimation quality of $\overline{\mathcal{SV}_{i,j}}$ $(1 \leq j \leq n)$. We obtain a bound on the total absolute error of approximate Shapley value as follows.

Theorem 5.2. *Given a set of players* $\mathcal{N} = \{z_1, \ldots, z_n\}$, *if for* $1 \leq i, j \leq n$, $|\overline{\mathcal{SV}_{i,j}} - \mathcal{SV}_{i,j}| \leq \epsilon_{i,j}$ *holds with probability at least* $1 - \delta$ $(0 < \delta \leq 1, \epsilon_{i,j} > 0)$, *then* $\sum_{i=1}^{n} |\overline{\mathcal{SV}_i} - \mathcal{SV}_i| \leq \epsilon$ *holds with probability at least* $(1-\delta)^{2n}$, *where* $\epsilon = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \epsilon_{i,j}$.

Algorithm 3: Shapley value computation based on the Neyman approach.

input : players $\mathcal{N} = \{z_1, \ldots, z_n\}$, $m_{init} > 1$, and $m > 0$
output: approximate Shapley value $\overline{\mathcal{SV}_i}$ for each player z_i $(1 \le i \le n)$

$\overline{\mathcal{SV}_i}, \overline{\mathcal{SV}_{i,j}}, m_{i,j} \leftarrow 0$ $(1 \le i \le n)$; $c \leftarrow -1$;
while $c \neq \sum_{j=1}^{n} m_{1,j}$ **do**
  $c = \sum_{j=1}^{n} m_{1,j}$;
  for *i=1 to n, j = 1 to n* **do**
    if $m_{i,j} < m_{init}$ **then**
      let $\mathcal{S}$ be a sample drawn from $\mathfrak{S}^j$;
      $u \leftarrow \mathcal{U}(\mathcal{S}) - \mathcal{U}(\mathcal{N} \setminus \mathcal{S})$;
      for $z_i \in \mathcal{S}$ **do**
        $\overline{\mathcal{SV}_{i,|\mathcal{S}|}} += u$; $m_{i,|\mathcal{S}|} += 1$;
      for $z_i \in \mathcal{N} \setminus \mathcal{S}$ **do**
        $\overline{\mathcal{SV}_{i,|\mathcal{N} \setminus \mathcal{S}|}} -= u$; $m_{i,|\mathcal{N} \setminus \mathcal{S}|} += 1$;
compute $\widehat{\sigma^2_{i,j}}$ $(1 \le i, j \le n)$;
$m_{first} \leftarrow \sum_{j=1}^{n} m_{1,j}$;
$m_j = \left\lceil \frac{(m - m_{first})\sqrt{\sum_{i=1}^{n}\left(\frac{\widehat{\sigma^2_{i,j}}}{j} + \frac{\widehat{\sigma^2_{i,n-j}}}{n-j}\right)}}{\sum_{j=\lceil n/2 \rceil}^{n}\sqrt{\sum_{i=1}^{n}\left(\frac{\widehat{\sigma^2_{i,j}}}{j} + \frac{\widehat{\sigma^2_{i,n-j}}}{n-j}\right)}} \right\rceil$ $(\lceil n/2 \rceil \le j \le n)$;
for $j = \lceil n/2 \rceil$ *to n* **do**
  for k *= 1 to* m_j **do**
    let $\mathcal{S}$ be a sample drawn from $\mathfrak{S}^j$;
    $u \leftarrow \mathcal{U}(\mathcal{S}) - \mathcal{U}(\mathcal{N} \setminus \mathcal{S})$;
    for $z_i \in \mathcal{S}$ **do**
      $\overline{\mathcal{SV}_{i,|\mathcal{S}|}} += u$; $m_{i,|\mathcal{S}|} += 1$;
    for $z_i \in \mathcal{N} \setminus \mathcal{S}$ **do**
      $\overline{\mathcal{SV}_{i,|\mathcal{N} \setminus \mathcal{S}|}} -= u$; $m_{i,|\mathcal{N} \setminus \mathcal{S}|} += 1$;
for *i=1 to n* **do**
  $\overline{\mathcal{SV}_i} = \frac{1}{n}\sum_{j=1}^{n} \overline{\mathcal{SV}_{i,j}} / m_{i,j}$;
return $\overline{\mathcal{SV}_1}, \ldots, \overline{\mathcal{SV}_n}$.

Proof. Since $Pr(|\overline{\mathcal{SV}_{i,j}} - \mathcal{SV}_{i,j}| \le \epsilon_{i,j}) \ge 1 - \delta$, we have

$$\begin{aligned} &Pr\Big(\sum_{i=1}^{n}\sum_{j=1}^{n} |\overline{\mathcal{SV}_{i,j}} - \mathcal{SV}_{i,j}| \le \sum_{i=1}^{n}\sum_{j=1}^{n} \epsilon_{i,j}\Big) \\ \ge &Pr(\cap_{i,j=1,\ldots,n}\{|\overline{\mathcal{SV}_{i,j}} - \mathcal{SV}_{i,j}| \le \epsilon_{i,j}\}) \\ = &\prod_{i=1}^{n}\prod_{j=1}^{n} Pr(|\overline{\mathcal{SV}_{i,j}} - \mathcal{SV}_{i,j}| \le \epsilon_{i,j}) = (1-\delta)^{2n}. \end{aligned}$$

Since $\sum_{i=1}^{n} |\overline{\mathcal{SV}_i} - \mathcal{SV}_i| \leq \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} |\overline{\mathcal{SV}_{i,j}} - \mathcal{SV}_{i,j}|$, we have $Pr(\sum_{i=1}^{n} |\overline{\mathcal{SV}_i} - \mathcal{SV}_i| \leq \epsilon) \geq (1-\delta)^{2n}$. □

Theorem 5.2 guides us to allocate more samples to the estimator $\mathcal{SV}_{i,j}$ that is relatively farther from its expected value in order to reduce the error bound of Shapley value ϵ. It is worth considering developing an approach that monitors the estimation quality of $\mathcal{SV}_{i,j}$ in an online manner and allocates samples adaptively. Thus, we use the empirical Bernstein-Serfling inequality to estimate $\epsilon_{i,j}$. The empirical Bernstein-Serfling inequality is suitable to this online scenario, since it is based on the up-to-date and observable information via replacing the variance by the sample variance [2, 3, 35, 37]. The empirical Bernstein-Serfling inequality [3] on Shapley value is shown as follows.

Theorem 5.3. ***(Empirical Bernstein-Serfling Inequality*** *[3]) Given a set of players* $\mathcal{N} = \{z_i, \dots, z_n\}$, *the range* r *of the utility function, a sample without replacement of* $CC_{\mathcal{N}}^{i,j}$ *of size* $m_{i,j}$ $\{CC_{\mathcal{N}}(\mathcal{S}_1), \dots, CC_{\mathcal{N}}(\mathcal{S}_{m_{i,j}})\}$, *where* $\mathcal{S}_1, \dots, \mathcal{S}_{m_{i,j}} \in \mathfrak{S}^{i,j}$ $(1 \leq i, j \leq n; 1 < m_{i,j} \leq \binom{n-1}{j-1})$, *with probability at least* $1-\delta$ $(\delta > 0)$ *we have*

$$
\begin{aligned}
&|\frac{1}{m_{i,j}}\sum_{k=1}^{m_{i,j}} CC_{\mathcal{N}}(\mathcal{S}_k) - \mathcal{SV}_{i,j}| = |\overline{\mathcal{SV}_{i,j}} - \mathcal{SV}_{i,j}| \\
\leq &\frac{1}{m_{i,j}}[\sqrt{2(m_{i,j}-1)\widehat{\sigma_{i,j}^2}\rho_{m_{i,j}}\log(10/(1+\delta))} + \kappa r \log(10/(1+\delta))],
\end{aligned}
$$

where $\kappa = \frac{14}{3} + 3\sqrt{2}$, $\widehat{\sigma_{i,j}^2} = \frac{1}{m_{i,j}-1}\sum_{k=1}^{m_{i,j}}(CC_{\mathcal{N}}(\mathcal{S}_k) - \mathcal{SV}_{i,j})^2$, *and*

$$
\rho_{m_{i,j}} = \begin{cases} 1 - \frac{m_{i,j}-1}{\binom{n-1}{j-1}} & \text{if } m_{i,j} \leq \binom{n-1}{j-1}/2 \\ \left(1 - \frac{m_{i,j}}{\binom{n-1}{j-1}}\right)(1 + 1/m_{i,j}) & \text{if } m_{i,j} > \binom{n-1}{j-1}/2 \end{cases}.
$$

To obtain a better estimation of Shapley value by reducing the bound ϵ, it is intuitive to assign the next sample to the estimators $\overline{\mathcal{SV}_{i,j}}$ with a larger bound estimated by the empirical Bernstein-Serfling inequality [3] $\epsilon_{i,j} = \frac{1}{m_{i,j}}[\sqrt{2(m_{i,j}-1)\widehat{\sigma_{i,j}^2}\rho_{m_{i,j}}\log(10/(1+\delta))} + \kappa r \log(10/(1+\delta))]$. Considering that a sample of complementary contributions can be used to update Shapley value of all players, we propose a method to select the next specific sample from a finite pool of complementary contributions. Given the previous sampling results, the next sample of complementary contributions is selected as

$$
\arg\max_{CC_{\mathcal{N}}(\mathcal{S})}\{\sum_{z_i \in \mathcal{S}} \epsilon_{i,|\mathcal{S}|} + \sum_{z_i \in \mathcal{N}\setminus\mathcal{S}} \epsilon_{i,|\mathcal{N}\setminus\mathcal{S}|}\}. \tag{5}
$$

Equation 5 provides a way to appraise the gain of samples from a view of Theorem 5.3. Taking the complementary contributions with the largest sum of error bounds contributes to tightening the bounds of most estimators with larger error bounds, which helps to reduce approximation errors of Shapley value. Thus, the complementary contribution with the largest sum of bounds is the most cost-effective selection in this sense. A combination of the empirical Bernstein-Serfling bound and sample selection is promising.

Finding the answer to Equation 5 is not easy. A brute-force solution is to enumerate complementary contributions, calculate the sum of corresponding bounds, and take the one with the largest sum of bounds. Note that there are 2^{n-1} pairs of complementary coalitions, the time cost of such a brute-force solution is prohibitively high, $O(n \cdot 2^n)$. To tackle the problem, we develop a polynomial time algorithm. The key idea is to find the j-coalition with the largest sum of bounds in $\mathfrak{S}^j$ for $j \in [\lceil n/2 \rceil, n]$, then select the coalition with the largest sum of bounds among the coalitions obtained in the first step and evaluate the corresponding complementary contribution. By converting

$\epsilon_{i,n-j}$ to the sum of $\epsilon_{i,j}$ and the difference $\Delta_{i,n-j} = \epsilon_{i,n-j} - \epsilon_{i,j}$, Equation 5 is equivalent to finding

$$\begin{aligned} &\arg\max_{CC_{\mathcal{N}}(\mathcal{S})} \{ \sum_{z_i \in \mathcal{S}} \epsilon_{i,|\mathcal{S}|} + \sum_{z_i \in \mathcal{N} \setminus \mathcal{S}} (\epsilon_{i,|\mathcal{S}|} + \Delta_{i,|\mathcal{N} \setminus \mathcal{S}|}) \} \\ =&\arg\max_{CC_{\mathcal{N}}(\mathcal{S})} \{ \sum_{z_i \in \mathcal{N}} \epsilon_{i,|\mathcal{S}|} + \sum_{z_i \in \mathcal{N} \setminus \mathcal{S}} \Delta_{i,|\mathcal{N} \setminus \mathcal{S}|} \}. \end{aligned} \tag{6}$$

Thus, the question of finding the j-coalition with the largest sum of bounds in $\mathfrak{S}^j$ ($\lceil n/2 \rceil \le j \le n$) is equivalent to finding

$$\arg\max_{CC_{\mathcal{N}}(\mathcal{S})} \sum_{z_i \in \mathcal{N} \setminus \mathcal{S}} \Delta_{i,n-j}, \textit{ where } |\mathcal{S}| = j. \tag{7}$$

We can obtain $\lceil n/2 \rceil$ complementary contributions while j traverses from $\lceil n/2 \rceil$ to n, and then select the complementary contribution with the largest sum of bounds.

The detailed algorithm is shown in Algorithm 4. We first compute $\Delta_{i,j}$ (Lines 1-3). Then, we find the first $n-k$ players with the largest $\Delta_{i,n-k}$ to construct $\mathcal{S}$ and compute the sum of bounds $\epsilon_{\{\mathcal{S},\mathcal{N}\setminus\mathcal{S}\}}$ when $|\mathcal{S}| = n-k$ (Lines 6-20). ϵ_{max} is set to record the largest sum of bounds. We update ϵ_{max} and the corresponding $\mathcal{S}'$ (Lines 21-22). Finally, we output $CC_{\mathcal{N}}(\mathcal{S}')$ and $\mathcal{S}'$ as the result (Line 23).

Theorem 5.4. *The time complexity of Algorithm 4 is* $O(n^2)$.

Proof. Because the $(n-k)^{th}$ largest $\Delta_{i,n-k}$ can be selected in $O(n)$ time by the classic selection algorithm [5] and there are $O(n)$ such operations as k goes from $\lceil n/2 \rceil$ to n, overall the algorithm takes $O(n^2)$ time. □

We show a running example of Algorithm 4.

Example 5.5. Given $\mathcal{N} = \{z_1, z_2, z_3, z_4, z_5\}$, let us assume the current bound matrix is $\mathbf{M}$, where the element in row i and column j is error bound $\epsilon_{i,j}$ ($1 \le i, j \le 5$).

$$\mathbf{M} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 3 & 4 & 2 & 5 & 1 \\ 4 & 1 & 5 & 3 & 2 \\ 5 & 2 & 4 & 1 & 3 \end{bmatrix}$$

Since $\Delta_{i,j}$ is computed as $\epsilon_{i,j} - \epsilon_{i,n-j}$ ($1 \le i \le 5, 1 \le j \le 2$), we can get a matrix Δ, where the element in row i and column j is $\Delta_{i,j}$ ($1 \le i \le 5, 1 \le j \le 2$).

$$\Delta = \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 3 & 4 \\ 4 & 1 \\ 5 & 2 \end{bmatrix} - \begin{bmatrix} 4 & 3 \\ 4 & 3 \\ 5 & 2 \\ 3 & 5 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} -3 & -1 \\ -3 & -1 \\ -2 & 2 \\ 1 & -4 \\ 4 & -2 \end{bmatrix}$$

Since $\epsilon_{i,j} = \epsilon_{i,n-j} + \Delta_{i,j}$ ($1 \le i \le 5, 1 \le j \le 2$), $\mathbf{M}$ can be written as follows.

$$\mathbf{M} = \begin{bmatrix} 4 & 3 & 3 & 4 & 5 \\ 4 & 3 & 3 & 4 & 5 \\ 5 & 2 & 2 & 5 & 1 \\ 3 & 5 & 5 & 3 & 2 \\ 1 & 4 & 4 & 1 & 3 \end{bmatrix} + \begin{bmatrix} -3 & -1 & 0 & 0 & 0 \\ -3 & -1 & 0 & 0 & 0 \\ -2 & 2 & 0 & 0 & 0 \\ 1 & -4 & 0 & 0 & 0 \\ 4 & -2 & 0 & 0 & 0 \end{bmatrix}$$

Algorithm 4: Sample Selection.

input : players $\mathcal{N} = \{z_1, \ldots, z_n\}$ and bounds $\epsilon_{1,1}, \ldots, \epsilon_{n,n}$
output: complementary contribution $CC_{\mathcal{N}}(\mathcal{S}')$ and coalition $\mathcal{S}'$

for *i=1 to n* **do**
  for *j=1 to* $\lfloor n/2 \rfloor$ **do**
    $\Delta_{i,j} = \epsilon_{i,j} - \epsilon_{i,n-j}$;
$\epsilon_{max} \leftarrow \sum_{i=1}^{n} \epsilon_{i,n}$; $\mathcal{S}' \leftarrow \mathcal{N}$;
for k=$\lceil n/2 \rceil$ *to* $n-1$ **do**
  $u \leftarrow$ the $(n-k)^{th}$ largest $\Delta_{i,n-k}$;
  $\mathcal{S} \leftarrow \emptyset$;
  for *i=1 to n* **do**
    if $\Delta_{i,n-k} > u$ **then**
      $\mathcal{S} = \mathcal{S} \cup \{z_i\}$;
  $i \leftarrow 1$
  while $|\mathcal{S}| < n-k$ **do**
    if $\Delta_{i,n-k} == u$ **then**
      $\mathcal{S} = \mathcal{S} \cup \{z_i\}$;
    $i+=1$;
  $\epsilon_{\{\mathcal{S},\mathcal{N}\setminus\mathcal{S}\}} \leftarrow 0$;
  for $z_i \in \mathcal{S}$ **do**
    $\epsilon_{\{\mathcal{S},\mathcal{N}\setminus\mathcal{S}\}} += \epsilon_{i,n-k}$;
  for $z_i \in \mathcal{N} \setminus \mathcal{S}$ **do**
    $\epsilon_{\{\mathcal{S},\mathcal{N}\setminus\mathcal{S}\}} += \epsilon_{i,k}$;
  if $\epsilon_{\{\mathcal{S},\mathcal{N}\setminus\mathcal{S}\}} > \epsilon_{max}$ **then**
    $\epsilon_{max} = \epsilon_{\{\mathcal{S},\mathcal{N}\setminus\mathcal{S}\}}$; $\mathcal{S}' \leftarrow \mathcal{S}$;
return $CC_{\mathcal{N}}(\mathcal{S}'), \mathcal{S}'$.

For $k = 3$, the second largest $\Delta_{i,2}$ is -1. Thus, we select $\mathcal{S} = \{z_1, z_3\}$ and $\epsilon_{\{\{z_1,z_3\},\{z_2,z_4,z_5\}\}}$=3+3+2+5+4-1+2=18. For $k = 4$, the first largest $\Delta_{i,1}$ is 4. We select $\mathcal{S} = \{z_5\}$ and $\epsilon_{\{\{z_5\},\{z_1,z_2,z_3,z_4\}\}}$=4+4+5+3+1+4=21. For $k = 5$, $\epsilon_{\{\{z_1,z_4,z_2,z_3,z_4,z_5\},\emptyset\}}$=5+5+1+2+3=16. Since $21 > 18 > 16$, $\mathcal{S}'$ is $\{z_5\}$ and the sample of complementary contributions is $CC_{\mathcal{N}}(\{z_5\})$ after running Algorithm 4.

The pseudo-code of approximate Shapley value computation using sample selection is shown in Algorithm 5. We initialize the empirical Bernstein-Serfling bound $\epsilon_{i,j}$ with at least two samples for computing the variance (Lines 2-12). Then, we sequentially select samples based on up-to-date $\epsilon_{i,j}$. We run Algorithm 4 to get the sample that yields the largest sum of bounds (Line 14). We update the estimators and recompute $\epsilon_{i,j}$ after drawing a new sample (Lines 15-20). Repeating the process until the total number of samples is reached, the final estimation of the Shapley value is simply the average of all the calculated means (Lines 21-22).

6 EXPERIMENTS

In this section, we report experimental results evaluating our proposed algorithms for computing Shapley value, including the effectiveness and the efficiency.

Algorithm 5: Shapley value computation based on the Bernstein bound.

input : players $\mathcal{N} = \{z_1, \ldots, z_n\}$ and $m > 0$

output: approximate Shapley value $\overline{\mathcal{SV}}_i$ for each player z_i $(1 \le i \le n)$

$\overline{\mathcal{SV}}_i, \overline{\mathcal{SV}}_{i,j}, m_{i,j} \leftarrow 0$ $(1 \le i, j \le n)$; $c \leftarrow -1$;
while $c \neq \sum_{j=1}^{n} m_{1,j}$ **do**
  $c = \sum_{j=1}^{n} m_{1,j}$;
  for *i=1 to n,j = 1 to n* **do**
    if $m_{i,j} < 2$ **then**
      let $\mathcal{S}$ be a sample drawn from $\mathfrak{S}^j$;
      $u \leftarrow \mathcal{U}(\mathcal{S}) - \mathcal{U}(\mathcal{N} \setminus \mathcal{S})$;
      for $z_i \in \mathcal{S}$ **do**
        $\overline{\mathcal{SV}}_{i,|\mathcal{S}|} += u$; $m_{i,|\mathcal{S}|} += 1$;
      for $z_i \in \mathcal{N} \setminus \mathcal{S}$ **do**
        $\overline{\mathcal{SV}}_{i,|\mathcal{N} \setminus \mathcal{S}|} -= u$; $m_{i,|\mathcal{N} \setminus \mathcal{S}|} += 1$;
compute $\epsilon_{i,j}$ $(1 \le i, j \le n)$;
for *k=1 to* $m - \sum_{j=1}^{n} m_{1,j}$ **do**
  run Algorithm 4 to get u and $\mathcal{S}'$;
  for $z_i \in \mathcal{S}'$ **do**
    $\overline{\mathcal{SV}}_{i,|\mathcal{S}'|} += u$; $m_{i,|\mathcal{S}'|} += 1$;
    recompute $\epsilon_{i,|\mathcal{S}'|}$;
  for $z_i \in \mathcal{N} \setminus \mathcal{S}'$ **do**
    $\overline{\mathcal{SV}}_{i,|\mathcal{N} \setminus \mathcal{S}'|} -= u$; $m_{i,|\mathcal{N} \setminus \mathcal{S}'|} += 1$;
    recompute $\epsilon_{i,|\mathcal{N} \setminus \mathcal{S}'|}$;
for *i=1 to n* **do**
  $\overline{\mathcal{SV}}_i = \frac{1}{n} \sum_{j=1}^{n} \overline{\mathcal{SV}}_{i,j} / m_{i,j}$;
return $\overline{\mathcal{SV}}_1, \ldots, \overline{\mathcal{SV}}_n$.

6.1 Experiment Setup

We conduct experiments on a computer with an Intel(R) Xeon(R) Silver 4214 CPU running Ubuntu with 128GB memory. The code for experiments is available in https://github.com/ZJU-DIVER/ShapleyValueApproximation.

6.1.1 Methods Compared. We compare our proposed algorithms with several baseline algorithms as follows.

- **MC**: the Monte Carlo simulation algorithm, which randomly samples permutations, developed by Castro et al. [8].
- **MCN**: the algorithm that improves MC by stratified random sampling of marginal contributions with optimum allocation, developed by Castro et al. [7].
- **MCH**: the algorithm with Hoeffding bound based on sampling of marginal contributions, developed by Maleki et al. [34].
- **CC**: the proposed algorithm based on complementary contributions (Algorithm 2).

- **CCN**: the enhanced algorithm with sample allocation based on Neyman approach (Algorithm 3). We set $m_{init} = max(2, \lfloor \frac{m}{2n^2} \rfloor)$ in the experiment results reported in Sections 6.2 and 6.3, where m is the sample size and n is the number of players.
- **CCB**: the enhanced algorithm with sample allocation based on empirical Bernstein bound (Algorithm 5).

6.1.2 Test Cases. To evaluate the efficiency and effectiveness of the methods, we employ both payoff allocation in cooperative games and data valuation in machine learning. Three cooperative game examples and a data valuation task are used as follows. The setting of cooperative game examples is following the same way by Castro et al. [8].

A Voting Game [39]. In a voting game, the principle of minority obeying majority is used. Shapley value in this game can be thought of as an index of voting power. The set of players of a non-symmetric voting game defined by Owen [39] for a voting process of a presidential election in the United States is $\mathcal{N} = \{1, \ldots, 51\}$ and the utility function in this game is given by

$$\mathcal{U}_v(\mathcal{S}) = \begin{cases} 1, & \text{if } \sum_{i \in \mathcal{S}} w_i > \frac{1}{2} \sum_{j \in \mathcal{N}} w_j \\ 0, & \text{otherwise} \end{cases}$$

where w_i is the weight of votes for player i, $\{w_1, \ldots, w_{51}\} = \{45, 41, 27, 26, 26, 25, 21, 17, 17, 14, 13, 13, 12, 12, 12, 11, \underbrace{10, \ldots, 10}_{4}, \underbrace{9, \ldots, 9}_{4}, 8, 8, \underbrace{7, \ldots, 7}_{4}, \underbrace{6, \ldots, 6}_{4}, 5, \underbrace{4, \ldots, 4}_{9}, \underbrace{3, \ldots, 3}_{7}\}$, and the subscript v annotates that the utility function is for this voting game.

An Airport Game [30]. In an airport game, an airstrip accommodating a given plane can accommodate any smaller plane at no additional cost. Shapley value in this game is a fair distribution of costs, deciding how to distribute the cost of an airstrip among different planes who need airstrips of different lengths. The set of players of an airport game is $\mathcal{N} = \{1, \ldots, 100\}$ and the utility function in this game is given by

$$\mathcal{U}_a(\mathcal{S}) = max_{i \in \mathcal{S}}\{c_i\},$$

where $\{c_1, \ldots, c_{100}\} = \{\underbrace{1, \ldots, 1}_{8}, \underbrace{2, \ldots, 2}_{12}, \underbrace{3, \ldots, 3}_{6}, \underbrace{4, \ldots, 4}_{14}, \underbrace{5, \ldots, 5}_{8}, \underbrace{6, \ldots, 6}_{9}, \underbrace{7, \ldots, 7}_{13}, \underbrace{8, \ldots, 8}_{10}, \underbrace{9, \ldots, 9}_{10}, \underbrace{10, \ldots, 10}_{10}\}$ and the subscript a annotates that the utility function is for this airport game.

A minimum spanning tree game [4]. In a minimum spanning tree game, a group of agents located at different geographical places share some services that can only be provided by a common supplier. Shapley value in this game is used to allocate the cost associated with the minimum spanning tree among the agents. The set of players of a minimum spanning tree game is $\mathcal{N} = \{1, \ldots, 100\}$ and the cost associated with an edge (i, j) is

$$C_{i,j} = \begin{cases} 1, & \text{if } i = j+1, i = j-1, i = 1 \land j = 100, i = 100 \land j = 1 \\ 101, & \text{if } i = 0 \text{ or } j = 0 \\ \infty, & \text{otherwise.} \end{cases}$$

The utility function in this game is the sum of the edge cost of the minimum spanning tree, i.e., $\mathcal{U}_m(\mathcal{S})$ = the minimum spanning tree of the graph $G|_{\mathcal{S} \cup \{0\}}$, where $G|_{\mathcal{S} \cup \{0\}}$ is the partial graph restricted to the players in coalition $\mathcal{S}$ and the source node 0.

A data valuation task. In this task, Shapley value is used to evaluate the contribution of each data point toward training a machine learning model. We used a real Breast Cancer Wisconsin dataset from the UCI machine learning repository [13]. We randomly sampled 600 data points for the task of training models and 99 points as the test dataset. Support Vector Machine (SVM) is employed as the machine learning model, and the utility function is the accuracy score of the trained SVM model on the test dataset. We choose SVM mainly because SVM has been successful in various applications, and is often considered one of the classic classifiers. It has also been used by recent works [28, 31] for Shapley value computation.

We used the above settings in Section 6.2 and 6.4 and the extended settings with varying numbers of players in Section 6.3.

6.1.3 Evaluation metric.

Average error ratio. Given benchmark Shapley value $\mathcal{SV}_i$ and estimated Shapley value $\overline{\mathcal{SV}_i}$ ($1 \leq i \leq n$), the average error ratio for the estimated Shapley value compared to the benchmark Shapley value is

$$\text{average error ratio} = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{\overline{\mathcal{SV}_i} - \mathcal{SV}_i}{\mathcal{SV}_i}\right|.$$

Maximum error ratio. Given benchmark Shapley value $\mathcal{SV}_i$ and estimated Shapley value $\overline{\mathcal{SV}_i}$ ($1 \leq i \leq n$), the maximum error ratio for the estimated Shapley value compared to the benchmark Shapley value is

$$\text{maximum error ratio} = \max_i \left|\frac{\overline{\mathcal{SV}_i} - \mathcal{SV}_i}{\mathcal{SV}_i}\right|.$$

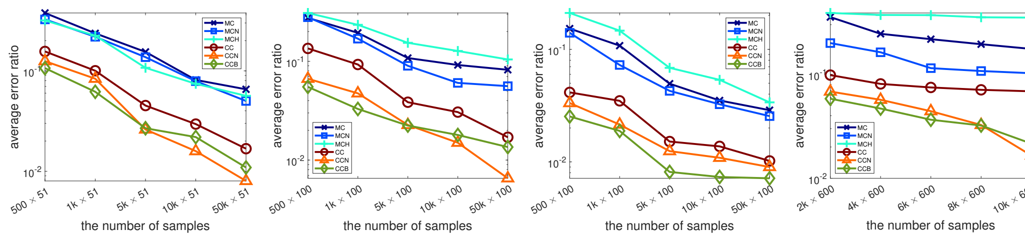
Computing the exact Shapley value $\mathcal{SV}_i$ for evaluation purposes is prohibitively expensive because it grows exponentially with the number of players. Therefore, we use the true Shapley value reported in [8] as the benchmark Shapley value in Figures 1(a)(b)(c), Figures 2(a)(b)(c), and Figures 6(a)(b)(c) and use the estimated Shapley value computed by the classic Monte Carlo simulation algorithm with 100000 permutations as the benchmark Shapley value for all other experiments.

Average coefficient of variation. Given a set of estimated Shapley value $\{\overline{\mathcal{SV}^1_i}, \ldots, \overline{\mathcal{SV}^k_i}\}$ ($1 \leq i \leq n, \overline{\mathcal{SV}^k_i} > 0$) obtained by computing k times using the same algorithm under the same setting, where $\overline{\mathcal{SV}^j_i}$ denotes the j^{th} estimated Shapley value of z_i computed by the algorithm, the average coefficient of variation is

$$\text{average CV} = \frac{1}{n}\sum_{i=1}^{n} \frac{\sqrt{\frac{1}{k}\sum_{j=1}^{k}(\overline{\mathcal{SV}^j_i} - \frac{1}{k}\sum_{j=1}^{k}\overline{\mathcal{SV}^j_i})^2}}{|\frac{1}{k}\sum_{j=1}^{k}\overline{\mathcal{SV}^j_i}|}.$$

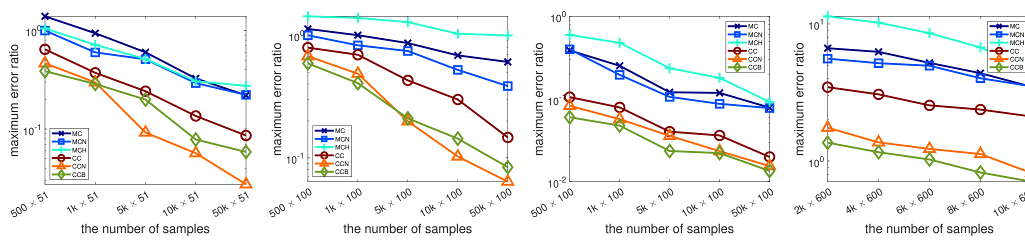
6.2 Effectiveness

We experimentally study the effectiveness of the proposed algorithms in four test cases, including the voting game, the airport game, the minimum spanning tree game, and the data valuation task. Figures 1(a)(b)(c)(d) and Figures 2(a)(b)(c)(d) show the average error ratios and the maximum error ratios of the estimated Shapley value with varying numbers of samples m for the four test cases, respectively. Both error ratios decrease with the increasing number of samples, indicating that the estimated Shapley value becomes closer to the accurate Shapley value. We observed that players with smaller Shapley values usually have larger error ratios. This is because even when the absolute error value is small, it can result in a large error ratio on a small Shapley value. CC, CCN, and CCB



(a) the voting game (b) the airport game (c) the minimum spanning tree game (d) the data valuation task

Fig. 1. Shapley value computation effectiveness (average error ratio).



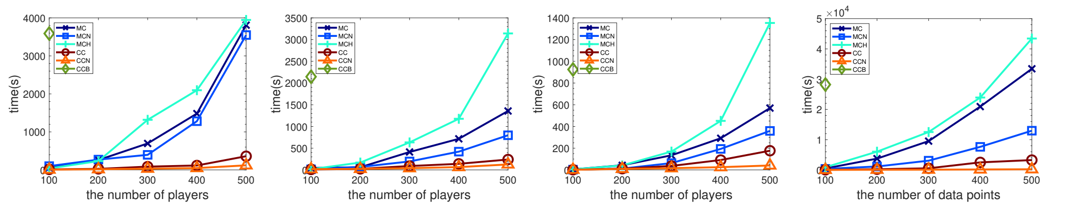
(a) the voting game (b) the airport game (c) the minimum spanning tree game (d) the data valuation task

Fig. 2. Shapley value computation effectiveness (maximum error ratio).

significantly outperform the baselines, including MC, MCN, and MCH, and achieve a small error even given a small number of samples. Boosted by improvements in sample allocation, both CCN and CCB outperform CC. As CCB constantly monitors the bounds of estimators and dynamically picks samples, it shows an advantage over CCN when given a relatively small number of samples. We note that this performance gain does come at a higher computation cost. Given the increasing number of samples, CCN obtains a more accurate estimation of the variance of each stratum since the number of samples in the first stage increases and the final sample allocation is closer to the theoretical optimum allocation. Thus, as the number of samples increases, CCN outperforms CCB in some cases.

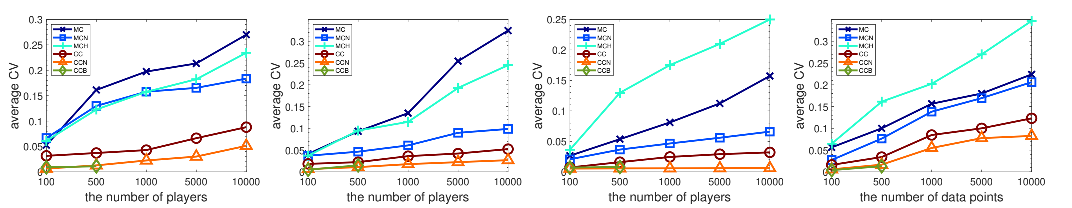
6.3 Efficiency

We experimentally study the efficiency of the proposed algorithms. We simulate four test cases on varying numbers of players. For the voting game, the airport game, and the minimum spanning tree game, we generated 100, 200, 300, 400, and 500 players, where each player has randomly generated $w_i/c_i/C_{i,j}$. For the data valuation task, we randomly sampled 100, 200, 300, 400, and 500 data points from the Breast Cancer Wisconsin dataset [13] to form different numbers of players and adopted the accuracy of the SVM model on the test dataset of size 99 as the utility function. Figures 3(a)(b)(c)(d) investigate the time cost for the algorithms to achieve an average error ratio $\leq 10\%$. Since CCB needs to update bounds and choose samples sequentially, which leads to a high time cost for large datasets, we omit some experimental results for CCB. The time cost required for the baselines increases sharply with the increasing number of players, while CC and CCN require significantly less time to achieve the same approximation error ratio, which verifies the efficiency and scalability of our algorithms.



(a) the voting game (b) the airport game (c) the minimum spanning tree game (d) the data valuation task

Fig. 3. Shapley value computation efficiency.



(a) the voting game (b) the airport game (c) the minimum spanning tree game (d) the data valuation task

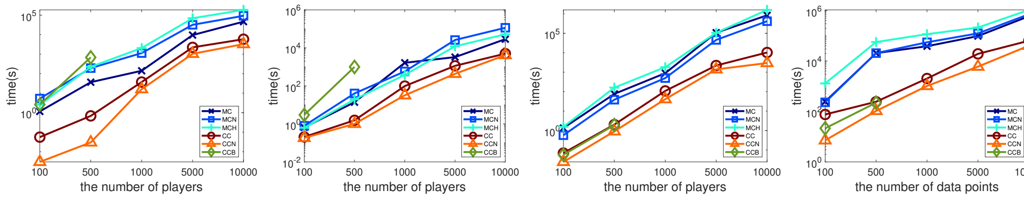
Fig. 4. Shapley value computation scalability.

Since it is hard to obtain a sufficiently accurate Shapley value as the benchmark Shapley value for comparison in tolerable time on large datasets, we perform an analysis of the proposed algorithms by measuring the average coefficient of variation. We simulated the voting game, the airport game, and the minimum spanning tree game with 100, 500, 1000, 5000, and 10000 players and randomly generated $w_i/c_i/C_{i,j}$ for each player. For the data valuation task, we used a real Adult dataset from the UCI machine learning repository [13], randomly sampled 100, 500, 1000, 5000, and 10000 data points for the task of training models and computing Shapley value, and randomly sampled 1000 points as the test dataset. SVM is employed as the machine learning model and the utility function is set to the accuracy score of the SVM model on the test dataset. Figures 4(a)(b)(c)(d) present the average CV of MC, MCN, MCH, CC, CCN, and CCB with $10000n$ samples. The average CV of CC, CCN, and CCB are much smaller than MC, MCN, and MCH, which confirms the convergence of the estimated Shapley value computed by our proposed algorithms. Thus, CC and CCN are scalable on larger data sets.

Moreover, Figures 5(a)(b)(c)(d) show the time cost for the algorithms to achieve an average CV ≤ 0.25 on four test cases with the varying number of players. The time cost increases with the increasing number of players. Some experimental results for CCB are not shown for large number of players due to the high time cost. CC and CCN perform well, beating all the other algorithms for all the scenarios tested, especially on data valuation tasks.

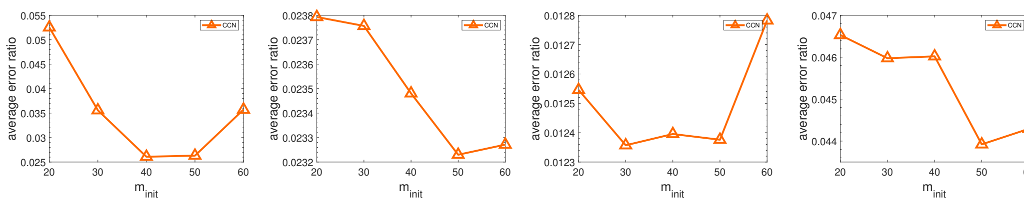
6.4 Effect of m_{init} in CCN

We study the performance of CCN with varying initialization sample size m_{init}. Given a total number of samples $5000n$, we set m_{init} to 20, 30, 40, 50, and 60, respectively. Figures 6(a)(b)(c)(d) show the average error ratio of CCN in four test cases. The average error ratio of CCN first decreases and then increases with the increase of m_{init}. As m_{init} increases, the estimation of $\sigma_{i,j}^2$ becomes more accurate so that a sample allocation scheme that is closer to the theoretically optimal allocation can



(a) the voting game (b) the airport game (c) the minimum spanning tree game (d) the data valuation task

Fig. 5. Shapley value computation scalability.



(a) the voting game (b) the airport game (c) the minimum spanning tree game (d) the data valuation task

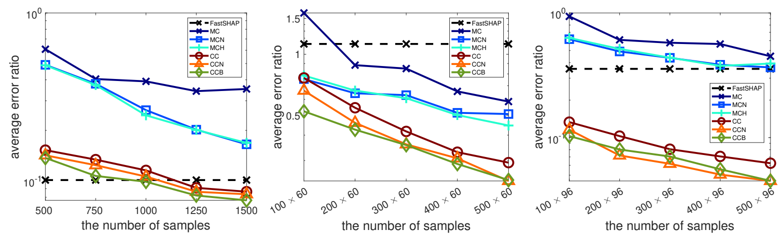
Fig. 6. Effect of m_{init} in CCN.

be obtained, thereby improving the performance. Moreover, since the number of samples allocated in the second stage decreases as m_{init} increases, the average error ratio of CCN increases.

6.5 Comparison

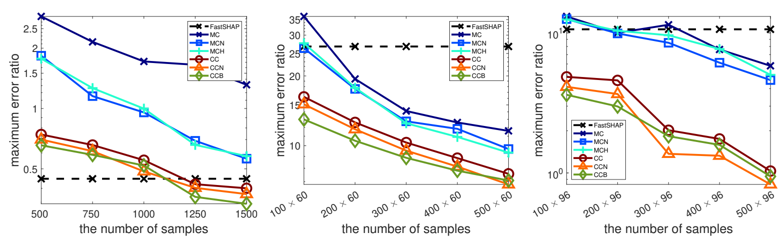
We analyze the performance of the proposed algorithms by comparing them to FastSHAP [22], a state-of-the-art learning-based method for Shapley value approximation. We would like to emphasize that our algorithms have significant advantages compared to FastSHAP. 1) FastSHAP studies a special task of model explanation. In this task, Shapley value assigns a value to each feature to represent the effect of that feature on the model prediction. The features of each data tuple are treated as players, and the utility function of a subset of features (players) is the model prediction of these features. It requires additional effort to adapt the method to the wide range of scenarios that we consider. 2) FastSHAP trains a neural network based on sampling coalition utilities on the training dataset to predict Shapley values on the test dataset. It cannot output an unbiased estimation of Shapley value even given a huge amount of training time. In contrast, CC can output more accurate Shapley value as the number of samples increases, and gives an unbiased estimation as shown in Theorem 4.5.

Besides the two advantages, our algorithms still outperform FastSHAP following the specific experimental settings in FastSHAP. We test the algorithms on the model explanation task using Census dataset [25] which contains 12 features, News dataset [15] which contains 60 features, and Bankruptcy dataset [27] which contains 96 features. For the original models to be explained, gradient boosted trees are used for Census dataset and Bankruptcy dataset, and neural networks are used for News dataset. Figures 7(a)(b)(c) and Figures 8(a)(b)(c) show the average error ratios and the maximum error ratios of each method for the three datasets, respectively. FastSHAP has been trained to converge, and its result is shown as horizontal lines. Training FastSHAP takes 46



(a) Census (b) News (c) Bankruptcy

Fig. 7. Comparison to FastSHAP (average error ratio).



(a) Census (b) News (c) Bankruptcy

Fig. 8. Comparison to FastSHAP (maximum error ratio).

minutes, 25 minutes, and 124 minutes for the three datasets, respectively, enough for the sample-based algorithms to take more than 300 million samples. As the number of samples increases, our algorithms provide Shapley values with smaller error ratios and outperform FastSHAP.

6.6 Discussion

Complementary contributions, each of which can be used to compute Shapley value for each player, provide a promising and more efficient method for estimating Shapley value. Experimental results show that CC and its variants significantly outperform the baselines. CCN and CCB improve the accuracy of CC by incorporating variance estimation. CCB involves additional overhead time due to sample selection, which does not scale well compared to CCN. Therefore, it costs more time to achieve a level of convergence, as shown in Figures 5(a)(b)(c)(d). However, given a small number of samples, CCB has the best performance, as shown in Figures 1(a)(b)(c)(d) and Figures 2(a)(b)(c)(d). In many applications, such as evaluating client contribution in cross-silo federated learning [42], the time cost for utility evaluation is prohibitively high. The major concern in computing Shapley value is how to reduce and choose samples to achieve a more accurate approximation with limited samples. CCB is favorable in this scenario because the cost of choosing samples is much smaller than the cost of evaluating utilities.

7 CONCLUSION

In this paper, we proposed the first stratified sampling method based on complementary contributions for approximating Shapley value, economizing valuable computational resources by conveniently reusing evaluated complementary contributions. We further proposed two sample allocation methods to improve sampling performance. The Neyman allocation-based method derives the sample allocation scheme with minimum variance for sampling complementary contributions.

The empirical Bernstein bound-based method monitors Shapley value estimators in an online fashion and picks samples to reduce approximation errors. Experimental results on real and synthetic datasets show that the proposed algorithms based on sampling complementary contributions with sample allocation strategies clearly outperform baseline algorithms based on sampling marginal contributions in effectiveness and efficiency.

There are several interesting directions for future research. CCB while effective in some scenarios, incurs significant computational overhead. It would be interesting to explore the approximate CCB method to achieve a good tradeoff between the sampling cost and the sample size required. In addition, there are several related and practical challenges: 1) how to quickly recalculate Shapley value when some players join/leave the coalition, and 2) how to fairly compute Shapley value when the utilities of some coalitions are uncertain.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported in part by NSFC grants (62102352), the National Key R&D Program of China (2021YFB3101100, 2022YFB310340), NSF grants (CNS-2124104, CNS-2125530), and NIH grants (R01LM013712, UL1TR002378).

REFERENCES

[1] Anish Agarwal, Munther A. Dahleh, and Tuhin Sarkar. 2019. A Marketplace for Data: An Algorithmic Solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, Anna Karlin, Nicole Immorlica, and Ramesh Johari (Eds.). ACM, 701–726. https://doi.org/10.1145/3328526.3329589

[2] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. 2007. Tuning Bandit Algorithms in Stochastic Environments. In *Algorithmic Learning Theory, 18th International Conference, ALT 2007, Sendai, Japan, October 1-4, 2007, Proceedings (Lecture Notes in Computer Science, Vol. 4754)*, Marcus Hutter, Rocco A. Servedio, and Eiji Takimoto (Eds.). Springer, 150–165. https://doi.org/10.1007/978-3-540-75225-7_15

[3] Rémi Bardenet and Odalric-Ambrym Maillard. 2015. Concentration inequalities for sampling without replacement. *Bernoulli* 21, 3 (2015), 1361–1385.

[4] Charles G. Bird. 1976. On cost allocation for a spanning tree: A game theoretic approach. *Networks* 6 (1976), 335–350.

[5] Manuel Blum, Robert W. Floyd, Vaughan R. Pratt, Ronald L. Rivest, and Robert Endre Tarjan. 1973. Time Bounds for Selection. *J. Comput. Syst. Sci.* 7, 4 (1973), 448–461. https://doi.org/10.1016/S0022-0000(73)80033-9

[6] Mark Alexander Burgess and Archie C. Chapman. 2021. Approximating the Shapley Value Using Stratified Empirical Bernstein Sampling. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, Zhi-Hua Zhou (Ed.). ijcai.org, 73–81. https://doi.org/10.24963/ijcai.2021/11

[7] Javier Castro, Daniel Gómez, Elisenda Molina, and Juan Tejada. 2017. Improving polynomial estimation of the Shapley value by stratified random sampling with optimum allocation. *Comput. Oper. Res.* 82 (2017), 180–188. https://doi.org/10.1016/j.cor.2017.01.019

[8] Javier Castro, Daniel Gómez, and Juan Tejada. 2009. Polynomial calculation of the Shapley value based on sampling. *Computers & OR* 36, 5 (2009), 1726–1730.

[9] Lingjiao Chen, Paraschos Koutris, and Arun Kumar. 2019. Towards Model-based Pricing for Machine Learning in a Data Marketplace. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, Peter A. Boncz, Stefan Manegold, Anastasia Ailamaki, Amol Deshpande, and Tim Kraska (Eds.). ACM, 1535–1552. https://doi.org/10.1145/3299869.3300078

[10] Lingjiao Chen, Hongyi Wang, Leshang Chen, Paraschos Koutris, and Arun Kumar. 2019. Demonstration of Nimbus: Model-based Pricing for Machine Learning in a Data Marketplace. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, Peter A. Boncz, Stefan Manegold, Anastasia Ailamaki, Amol Deshpande, and Tim Kraska (Eds.). ACM, 1885–1888. https://doi.org/10.1145/3299869.3320231

[11] Xiaotie Deng and Christos H. Papadimitriou. 1994. On the Complexity of Cooperative Solution Concepts. *Math. Oper. Res.* 19, 2 (1994), 257–266. https://doi.org/10.1287/moor.19.2.257

[12] Daniel Deutch, Nave Frost, Benny Kimelfeld, and Mikaël Monet. 2022. Computing the Shapley Value of Facts in Query Answering. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022,*

Zachary G. Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 1570–1583. https://doi.org/10.1145/3514221.3517912
[13] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml
[14] Eitan Farchi, Ramasuri Narayanam, and Lokesh Nagalapatti. 2021. Ranking Data Slices for ML Model Validation: A Shapley Value Approach. In *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021*. IEEE, 1937–1942. https://doi.org/10.1109/ICDE51399.2021.00180
[15] Kelwin Fernandes, Pedro Vinagre, and Paulo Cortez. 2015. A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. In *Progress in Artificial Intelligence - 17th Portuguese Conference on Artificial Intelligence, EPIA 2015, Coimbra, Portugal, September 8-11, 2015. Proceedings (Lecture Notes in Computer Science, Vol. 9273)*, Francisco C. Pereira, Penousal Machado, Ernesto Costa, and Amílcar Cardoso (Eds.). Springer, 535–546. https://doi.org/10.1007/978-3-319-23485-4_53
[16] Raul Castro Fernandez. 2022. Protecting Data Markets from Strategic Buyers. In *SIGMOD '22: International Conference on Management of Data, Philadelphia, PA, USA, June 12 - 17, 2022*, Zachary G. Ives, Angela Bonifati, and Amr El Abbadi (Eds.). ACM, 1755–1769. https://doi.org/10.1145/3514221.3517855
[17] Amirata Ghorbani, Michael P. Kim, and James Zou. 2020. A Distributional Framework For Data Valuation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 3535–3544.
[18] Amirata Ghorbani, James Zou, and Andre Esteva. 2022. Data Shapley Valuation for Efficient Batch Active Learning. In *56th Asilomar Conference on Signals, Systems, and Computers, ACSSC 2022, Pacific Grove, CA, USA, October 31 - Nov. 2, 2022*. IEEE, 1456–1462. https://doi.org/10.1109/IEEECONF56349.2022.10064696
[19] Amirata Ghorbani and James Y. Zou. 2019. Data Shapley: Equitable Valuation of Data for Machine Learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 2242–2251. http://proceedings.mlr.press/v97/ghorbani19c.html
[20] Amirata Ghorbani and James Y. Zou. 2020. Neuron Shapley: Discovering the Responsible Neurons. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/41c542dfe6e4fc3deb251d64cf6ed2e4-Abstract.html
[21] Wassily Hoeffding. 1994. Probability inequalities for sums of bounded random variables. In *The collected works of Wassily Hoeffding*. Springer, 409–426.
[22] Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. 2022. FastSHAP: Real-Time Shapley Value Estimation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net. https://openreview.net/forum?id=Zq2G_VTV53T
[23] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gürel, Bo Li, Ce Zhang, Costas J. Spanos, and Dawn Song. 2019. Efficient Task-Specific Data Valuation for Nearest Neighbor Algorithms. *Proc. VLDB Endow.* 12, 11 (2019), 1610–1623.
[24] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J. Spanos. 2019. Towards Efficient Data Valuation Based on the Shapley Value. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan (Proceedings of Machine Learning Research, Vol. 89)*, Kamalika Chaudhuri and Masashi Sugiyama (Eds.). PMLR, 1167–1176.
[25] Ron Kohavi. 1996. Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*, Evangelos Simoudis, Jiawei Han, and Usama M. Fayyad (Eds.). AAAI Press, 202–207. http://www.aaai.org/Library/KDD/1996/kdd96-033.php
[26] Yongchan Kwon, Manuel A. Rivas, and James Zou. 2021. Efficient Computation and Analysis of Distributional Shapley Values. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 130)*, Arindam Banerjee and Kenji Fukumizu (Eds.). PMLR, 793–801.
[27] Deron Liang, Chia-Chi Lu, Chih-Fong Tsai, and Guan-An Shih. 2016. Financial ratios and corporate governance indicators in bankruptcy prediction: A comprehensive study. *Eur. J. Oper. Res.* 252, 2 (2016), 561–572. https://doi.org/10.1016/j.ejor.2016.01.012
[28] Qiongqiong Lin, Jiayao Zhang, Jinfei Liu, Kui Ren, Jian Lou, Junxu Liu, Li Xiong, Jian Pei, and Jimeng Sun. 2021. Demonstration of Dealer: An End-to-End Model Marketplace with Differential Privacy. *Proc. VLDB Endow.* 14, 12 (2021), 2747–2750.
[29] Roy Lindelauf, Herbert Hamers, and Bart Husslage. 2013. Cooperative game theoretic centrality analysis of terrorist networks: The cases of Jemaah Islamiyah and Al Qaeda. *Eur. J. Oper. Res.* 229, 1 (2013), 230–238. https://doi.org/10.1016/j.ejor.2013.02.032

[30] S.C. Littlechild and G.F. Thompson. 1977. Aircraft Landing Fees: A Game Theory Approach. *Bell Journal of Economics* 8, 1 (Spring 1977), 186–204. https://ideas.repec.org/a/rje/bellje/v8y1977ispringp186-204.html

[31] Jinfei Liu, Jian Lou, Junxu Liu, Li Xiong, Jian Pei, and Jimeng Sun. 2021. Dealer: An End-to-End Model Marketplace with Differential Privacy. *Proc. VLDB Endow.* 14, 6 (2021), 957–969.

[32] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 4765–4774. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

[33] Sasan Maleki. 2015. *Addressing the computational issues of the Shapley value with applications in the smart grid.* Ph. D. Dissertation. University of Southampton, UK. http://eprints.soton.ac.uk/383963/

[34] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. 2013. Bounding the Estimation Error of Sampling-based Shapley Value Approximation With/Without Stratifying. *CoRR* abs/1306.4265 (2013). arXiv:1306.4265 http://arxiv.org/abs/1306.4265

[35] Andreas Maurer and Massimiliano Pontil. 2009. Empirical Bernstein Bounds and Sample-Variance Penalization. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*. http://www.cs.mcgill.ca/%7Ecolt2009/papers/012.pdf#page=1

[36] Rory Mitchell, Joshua Cooper, Eibe Frank, and Geoffrey Holmes. 2022. Sampling Permutations for Shapley Value Estimation. *J. Mach. Learn. Res.* 23 (2022), 43:1–43:46. http://jmlr.org/papers/v23/21-0439.html

[37] Volodymyr Mnih, Csaba Szepesvári, and Jean-Yves Audibert. 2008. Empirical Bernstein stopping. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008 (ACM International Conference Proceeding Series, Vol. 307)*, William W. Cohen, Andrew McCallum, and Sam T. Roweis (Eds.). ACM, 672–679. https://doi.org/10.1145/1390156.1390241

[38] Jerzy Neyman. 1992. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. In *Breakthroughs in statistics*. Springer, 123–150.

[39] G. Owen. 2013. *Game Theory*. Emerald Group Publishing Limited. https://books.google.com.ph/books?id=OfnLkgEACAAJ

[40] Lloyd S Shapley. 1953. A value for n-person games. *Contributions to the Theory of Games* 2, 28 (1953), 307–317.

[41] Tianshu Song, Yongxin Tong, and Shuyue Wei. 2019. Profit Allocation for Federated Learning. In *2019 IEEE International Conference on Big Data (IEEE BigData), Los Angeles, CA, USA, December 9-12, 2019*, Chaitanya Baru, Jun Huan, Latifur Khan, Xiaohua Hu, Ronay Ak, Yuanyuan Tian, Roger S. Barga, Carlo Zaniolo, Kisung Lee, and Yanfang Fanny Ye (Eds.). IEEE, 2577–2586. https://doi.org/10.1109/BigData47090.2019.9006327

[42] Tianhao Wang, Johannes Rausch, Ce Zhang, Ruoxi Jia, and Dawn Song. 2020. A Principled Approach to Data Valuation for Federated Learning. In *Federated Learning - Privacy and Incentive*, Qiang Yang, Lixin Fan, and Han Yu (Eds.). Lecture Notes in Computer Science, Vol. 12500. Springer, 153–167. https://doi.org/10.1007/978-3-030-63076-8_11

[43] Jiayao Zhang, Haocheng Xia, Qiheng Sun, Jinfei Liu, Li Xiong, Jian Pei, and Kui Ren. 2023. Dynamic Shapley Value Computation. In *37th IEEE International Conference on Data Engineering, ICDE 2023, California, USA, April 3-7, 2023*. IEEE.

Received April 2022; revised July 2022; accepted August 2022