Beta Shapley: a Unified and Noise-reduced Data Valuation Framework for Machine Learning

Yongchan Kwon **James Zou**
Department of Biomedical Data Science, Stanford University

Abstract

Data Shapley has recently been proposed as a principled framework to quantify the contribution of individual datum in machine learning. It can effectively identify helpful or harmful data points for a learning algorithm. In this paper, we propose Beta Shapley, which is a substantial generalization of Data Shapley. Beta Shapley arises naturally by relaxing the efficiency axiom of the Shapley value, which is not critical for machine learning settings. Beta Shapley unifies several popular data valuation methods and includes data Shapley as a special case. Moreover, we prove that Beta Shapley has several desirable statistical properties and propose efficient algorithms to estimate it. We demonstrate that Beta Shapley outperforms state-of-the-art data valuation methods on several downstream ML tasks such as: 1) detecting mislabeled training data; 2) learning with subsamples; and 3) identifying points whose addition or removal have the largest positive or negative impact on the model.

1 INTRODUCTION

Getting appropriate training data is often the biggest and most expensive challenge of machine learning (ML). In many real world applications, a fraction of the data could be very noisy due to outliers or label errors. Furthermore, data are often costly to collect and understanding of what types of data are more useful for training a model can help to guide data curation. For all of these motivations, data valuation has emerged as an important area of ML research. The goal of data valuation is to quantify the contribution of each training datum to the model's performance.

Recently, inspired by ideas from economics and game theory, data Shapley has been proposed to represent the notion of which datum helps or harms the predictive performance of a model (Ghorbani and Zou, 2019). Data Shapley has several benefits compared to existing data valuation methods. It uniquely satisfies the natural properties of fairness in cooperative game theory. Also, it better captures the influence of individual datum, showing superior performance on multiple downstream ML tasks, including identifying mislabeled observations in classification problems or detecting outliers in regression problems (Ghorbani and Zou, 2019; Jia et al., 2019a,b).

Data Shapley is defined as a function of marginal contributions that measure the average change in a trained model's performance when a particular point is removed from a set with a given cardinality. The marginal contribution is a basic ingredient in many data valuation approaches. For example, the commonly used leave-one-out (LOO) analysis is equivalent to estimating a point's marginal contribution when it is removed from the entire training set. The marginal contribution of a point can vary if the cardinality of a given set changes, and data Shapley takes a simple average of the marginal contributions on all the different cardinalities. In this way, data Shapley can avoid the dependency on a specific cardinality, but it is unclear whether this uniform weight is optimal for quantifying the impact of individual datum. As we will show both theoretically and through experiments, this is in fact sub-optimal. The uniform averaging arises from the efficiency axiom of Shapley values, which is not essential in ML settings. The axiom requires the sum of data values to equal the total utility, but it might not be sensible nor verifiable in practice.

Our contributions In this paper, we propose Beta Shapley, a unified data valuation framework that naturally arises by relaxing the efficiency axiom. Our theoretical analyses show that Beta Shapley is char-

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

acterized by reduced noise compared to data Shapley and can be applied to find optimal importance weights for subsampling. We develop an efficient algorithm to estimate it based on Monte Carlo methods. We demonstrate that Beta Shapley outperforms state-of-the-art data valuation methods on several downstream ML tasks including noisy label detection, learning with subsamples, and point addition and removal experiments.

Related works The Shapley value was introduced in a seminar paper as a method of fair division of rewards in cooperative games (Shapley, 1953b). It has been applied to various ML problems, for instance, variable selection (Cohen et al., 2005; Zaeri-Amirani et al., 2018), feature importance (Lundberg and Lee, 2017; Covert et al., 2020; Lundberg et al., 2020; Covert et al., 2021; Covert and Lee, 2021), model interpretation (Chen et al., 2019; Sundararajan and Najmi, 2020; Ghorbani and Zou, 2020; Wang et al., 2021), model importance (Rozemberczki and Sarkar, 2021), and the collaborative learning problems (Sim et al., 2020). As for the data valuation problem, data Shapley was introduced by Ghorbani and Zou (2019) and Jia et al. (2019b), and many extensions have been studied in the literature. For example, KNN Shapley was proposed to address the computational cost of data Shapley by using the k-nearest neighborhood model (Jia et al., 2019a), and distributional Shapley value was studied to deal with the random nature of data Shapley (Ghorbani et al., 2020; Kwon et al., 2021).

The relaxation of the Shapley axioms has been one of the central topics in the field of economics (Kalai and Samet, 1987; Weber, 1988). When the symmetry axiom is removed, the quasivalue and the weighted value have been studied (Shapley, 1953a; Banzhaf III, 1964; Gilboa and Monderer, 1991; Monderer et al., 1992). When the efficiency axiom is removed, the semivalue has been studied (Dubey and Weber, 1977; Dubey et al., 1981; Ridaoui et al., 2018). We refer to Monderer and Samet (2002b) for a complementary literature review of variations of Shapley value. Our work is based on the semivalue and characterizes its statistical properties in ML settings.

2 PRELIMINARIES

We review the marginal contribution, a key component for analyzing the impact of one datum, and various data valuation methods based on it. We first define some notations. Let Z be a random variable for data defined on a set $\mathcal{Z} \subseteq \mathbb{R}^d$ for some integer d and denote its distribution by P_Z. In supervised learning, we can think of $Z = (X, Y)$ defined on a set $\mathcal{X} \times \mathcal{Y}$, where X and Y describe the input and its label, respectively. Throughout this paper, we denote a set of independent and identically distributed (i.i.d.) samples from P_Z by $\mathcal{D} = \{z_1, \ldots, z_n\}$. We denote a utility function by $U : \cup_{j=0}^{\infty} \mathcal{Z}^j \to \mathbb{R}$. Here, we use the conventions $\mathcal{Z}^0 := \{\emptyset\}$ and $U(\emptyset)$ is the performance based on the best constant predictor. The utility function represents the performance of a model trained on a set of data points. In regression, for instance, one choice for $U(S)$ is the negative mean squared error of a model (e.g. linear regression) trained on the subset $S \subseteq \mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$. Similarly, in classification, $U(S)$ can be the classification accuracy of a model (e.g. logistic regression) trained on S. Note that the utility depends on which model is used. Throughout this paper, the dependence on the model is omitted for notational simplicity, but it does not affect our results. Lastly, for a set S, we denote its cardinality by $|S|$, and for $m \in \mathbb{N}$, we use $[m]$ to denote a set of integers $\{1, \ldots, m\}$.

Data valuation has been studied as a problem to evaluate the impact of individual datum, and many existing metrics measure how much the model output or model performance changes after removing one data point of interest. These concepts can be formalized by the marginal contribution defined below.

Definition 1 (Marginal contribution)**.** *For a function h and $j \in [n]$, we define the marginal contribution of $z^* \in \mathcal{D}$ with respect to $j-1$ samples as*

$$\Delta_j(z^*; h, \mathcal{D}) := \frac{1}{\binom{n-1}{j-1}} \sum_{S \in \mathcal{D}_j^{\backslash z^*}} h(S \cup \{z^*\}) - h(S),$$

where $\mathcal{D}_j^{\backslash z^*} := \{S \subseteq \mathcal{D} \backslash \{z^*\} : |S| = j-1\}$.

The marginal contribution $\Delta_j(z^*; h, \mathcal{D})$ considers all possible subsets with the same cardinality $S \in \mathcal{D}_j^{\backslash z^*}$ and measures the average changes of h when datum of interest z^* is removed from $S \cup \{z^*\}$. Note that when $j = n$, the marginal contribution $\Delta_n(z^*; h, \mathcal{D})$ equals to $h(\mathcal{D}) - h(\mathcal{D} \backslash \{z^*\})$, and it captures the effect of deleting z^* from the entire training dataset $\mathcal{D}$.

Many existing data valuation methods can be explained by the marginal contribution $\Delta_j(z^*; h, \mathcal{D})$. Specifically, Cook's distance (Cook and Weisberg, 1980, 1982) is proportional to the squared ℓ_2-norm of the marginal contribution $\left\|\Delta_n(z^*; h, \mathcal{D})\right\|_2^2$ when h outputs predictions of the given dataset $\mathcal{D}$, and the LOO method uses $\Delta_n(z^*; h, \mathcal{D})$ as data values when h is a utility function U. Also, the influence function can be regarded as an approximation of LOO (Koh and Liang, 2017).

Data Shapley is another example that can be expressed as a function of marginal contributions (Ghorbani and Zou, 2019; Jia et al., 2019b,a). To be more specific,

data Shapley of datum $z^* \in \mathcal{D}$ is defined as

$$\psi_{\text{shap}}(z^*; U, \mathcal{D}) := \frac{1}{n}\sum_{j=1}^{n} \Delta_j(z^*; U, \mathcal{D}). \tag{1}$$

Unlike Cook's distance or LOO methods, data Shapley in (1) considers all cardinalities and takes a simple average of the marginal contributions. By assigning the constant weight on different marginal contributions, it avoids the dependency on a specific cardinality and can capture the effect of one data point for small cardinality.

Data Shapley provides a principled data valuation framework in that it uniquely satisfies the natural properties of a fair division of rewards in cooperative game theory. Shapley (1953b) showed that the Shapley value is the unique function ψ that satisfies the following four axioms.

- Linearity: for functions U_1, U_2 and $\alpha_1, \alpha_2 \in \mathbb{R}$, $\psi(z^*; \alpha_1 U_1 + \alpha_2 U_2, \mathcal{D}) = \alpha_1 \psi(z^*; U_1, \mathcal{D}) + \alpha_2 \psi(z^*; U_2, \mathcal{D})$.
- Null player: if $U(S \cup \{z^*\}) = U(S) + c$ for any $S \subseteq \mathcal{D} \backslash \{z^*\}$ and some $c \in \mathbb{R}$, then $\psi(z^*; U, \mathcal{D}) = c$.
- Symmetry: for every U and every permutation π on $\mathcal{D}$, $\psi(\pi^* U) = \pi^* \psi U$ where $\pi^* U$ is defined as $(\pi^* U)(S) := U(\pi(S))$ for every $S \subseteq \mathcal{D}$.
- Efficiency: for every U, $\sum_{z \in \mathcal{D}} \psi(z; U, \mathcal{D}) = U(\mathcal{D})$.

Although data Shapley provides a fundamental framework for data values, there are some critical issues. In particular, it is unclear whether the uniform weight in (1) is optimal to represent the influence of one datum. When the cardinality $|S|$ is large enough, the performance change $U(S \cup \{z^*\}) - U(S)$ is near zero, and thus the marginal contribution $\Delta_{|S|}(z^*; U, \mathcal{D})$ becomes negligible. In particular, when U is a negative log-likelihood function, it can be shown that $U(S \cup \{z^*\}) - U(S) = O_p(|S|^{-2})$ under mild conditions. This can make it hard to tell which data points contribute more to predictive performance. In the following section, we rigorously analyze the marginal contribution and show that using the uniform weight in (1) can be detrimental to capturing the influence of individual data.

3 THEORETICAL ANALYSIS OF MARGINAL CONTRIBUTION

In this section, we study asymptotic properties of the marginal contribution. To this end, we define a set $\mathfrak{D} = \{z^*, Z_1, \ldots, Z_{n-1}\}$ where Z_i's be i.i.d. random variables from P_Z, *i.e.*, all elements of $\mathfrak{D}$ are random except for z^*. Given that $\Delta_j(z^*; U, \mathfrak{D})$ has the form of U-statistics (Hoeffding, 1948), Theorem 12.3 in Van der Vaart (2000) implies that for a fixed cardinality j, as n approaches to infinity, we have

$$(j^2\zeta_1/n)^{-1}\text{Var}(\Delta_j(z^*; U, \mathfrak{D}) \to 1, \tag{2}$$

where $\zeta_1 = \text{Var}\left(\mathbb{E}[U(S \cup \{z^*\}) - U(S) \mid Z_1]\right)$ and S is a random subset such that $|S| = j - 1$ and each element in S is chosen from $\{Z_1, \ldots, Z_{n-1}\} = \mathfrak{D} \backslash \{z^*\}$ uniformly at random. All expectation and variance computations are under the P_Z. The result (2) shows that the asymptotic variance of $\Delta_j(z^*; U, \mathfrak{D})$ scales $O(j^2\zeta_1/n)$ for a fixed cardinality j. However, since the data Shapley is a simple average of marginal contributions across all cardinalities $j \in \{1, \ldots, n\}$, an analysis of marginal contribution for large j is important to examine the statistical properties of the data Shapley.

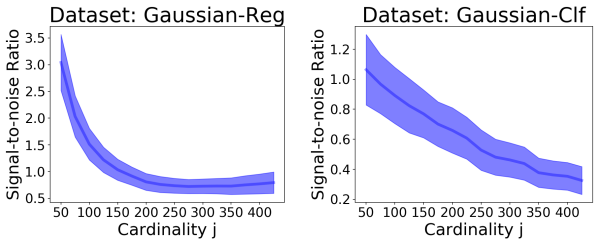

Figure 1: The signal-to-noise ratio of $\Delta_j(z^*; U, \mathfrak{D})$ as a function of the cardinality j when $n = 500$ in (left) regression and (right) classification settings. The data are generated from a generalized linear model. The signal-to-noise ratio generally decreases as the cardinality j increases, showing that when j is large, the signal of the marginal contribution at large cardinality is more likely to be perturbed by noise.

In the following theorem, we provide an asymptotic variance when the cardinality j is allowed to increase to infinity. To begin with, for $j \in [n]$ we set $\zeta_j := \text{Var}\left(U(S \cup \{z^*\}) - U(S)\right)$ where S is a random subset such that $|S| = j - 1$ and $S \subseteq \mathfrak{D} \backslash \{z^*\}$.

Theorem 1 (Asymptotic distribution of the marginal contribution)**.** *Suppose the cardinality $j = o(n^{1/2})$ and assume that $\lim_{j\to\infty} \zeta_j/(j\zeta_1)$ is bounded. Then, $(j^2\zeta_1/n)^{-1}\text{Var}(\Delta_j(z^*; U, \mathfrak{D})) \to 1$ as n increases.*

Theorem 1 extends the previous asymptotic result in (2) to the case of diverging cardinality j. Note that for any $0 \leq \gamma < 1/2$, the cardinality $j = n^\gamma$ satisfies the condition $j = o(n^{1/2})$. This result provides a mathematical insight into the signal-to-noise ratio of $\Delta_j(z^*; U, \mathfrak{D})$ as the following remark.

Remark 1. *Given that $|\mathbb{E}[\Delta_j(z^*; U, \mathfrak{D})]|$ usually decreases as j increases in ML settings, the signal-to-noise ratio $|\mathbb{E}[\Delta_j(z^*; U, \mathfrak{D})]|/\sqrt{\text{Var}(\Delta_j(z^*; U, \mathfrak{D}))}$ is expected*

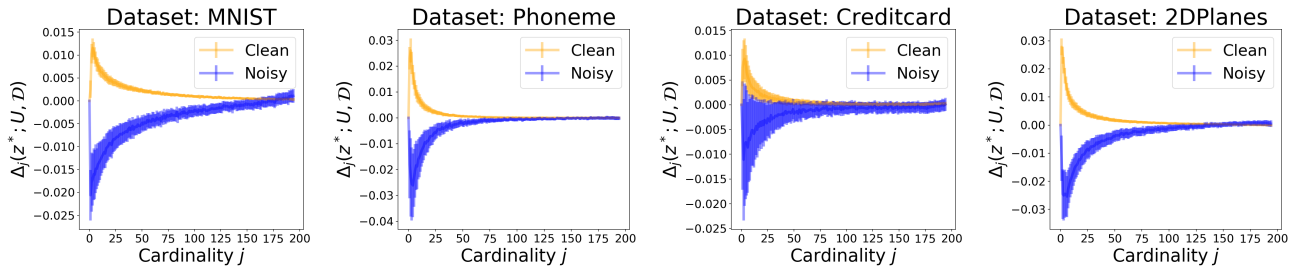

Figure 2: Illustrations of the marginal contribution $\Delta_j(z^*; U, \mathcal{D})$ as a function of the cardinality j on the four datasets. Each color indicates a clean data point (yellow) and a noisy data point (blue). When the cardinality j is large, the marginal contributions of the two groups become similar, so it is difficult to determine whether a data point is noisy by $\Delta_j(z^*; U, \mathcal{D})$. We provide additional results on different datasets in Appendix D.

to decrease as j increases because $\mathrm{Var}(\Delta_j(z^*; U, \mathfrak{D}))$ *is* $O(j^2\zeta_1/n)$.

Figure 1 illustrates the signal-to-noise ratio of $\Delta_j(z^*; U, \mathfrak{D})$ as a function of the cardinality j for two example datasets with $n = 500$. We denote a 95% confidence band based on 50 repetitions under the assumption that the results follow the identical Gaussian distribution. We use a generalized linear model to generate data and use the negative mean squared error and the classification accuracy as a utility for regression and classification settings, respectively. Figure 1 clearly shows the signal-to-noise ratio decreases as j increases. In other words, when j is large, the signal of marginal contribution is more likely to be perturbed by noise. This result motivates us to assign large weight to small cardinality instead of using the uniform weight used in data Shapley. We provide details on implementation and additional results on real datasets in Appendix A.

In Theorem 1, we fix one data point z^* and show that noise can be introduced when the cardinality is large. We now consider the entire dataset $\mathcal{D}$ and examine which cardinality is useful to capture the signal. To do so, we directly compare marginal contribution $\Delta_j(z^*; U, \mathcal{D})$ of mislabeled and correctly labeled data points in various classification settings. We set the sample size $n = 200$ and assume that observed data can be mislabeled. As for mislabeled data, we flip the original label for a random 10% of data points in $\mathcal{D}$. The utility function is the classification accuracy. Implementation details are provided in Appendix A.

Figure 2 shows that there is a significant gap between the marginal contributions of the two groups when j is small, but the gap becomes zero when j is large. In particular, the 95% confidence band for the clean data point (yellow) overlaps the confidence band for the mislabeled data point (blue) when j is greater than 150 in all datasets. This shows that using the uniform weight used in data Shapley (1) makes it difficult to tell whether or not a data point belongs to the clean group, and as a result, it can lead to undesirable decisions.

One potential limit of Theorem 1 is that it is unknown whether the bound condition $\lim_{j\to\infty} \zeta_j/(j\zeta_1)$ holds. We empirically show that this condition is plausible in Appendix B.

4 PROPOSED BETA-SHAPLEY METHOD

Motivated by the results in Section 3, assigning large weights to small cardinality is expected to capture the impact of one datum better than data Shapley. In the following subsection, motivated by the idea of semivalue (Dubey et al., 1981), we show that removing the efficiency axiom gives a new form of data value that can assign larger weight on the marginal contribution based on small cardinality than large cardinality.

4.1 Data valuations without efficiency axiom

The efficiency axiom, which requires the total sum of data values to be equal to the utility, is not essential in ML settings. For example, multiplying data Shapley by a positive constant changes the sum of the data values but does not change the order between the data values. In other words, there are many data values that do not satisfy the efficiency but can equivalently identify low-quality data points as data Shapley. In this respect, we define a semivalue, which is characterized by all Shapley axioms except the efficiency axiom.

Definition 2 (semivalue)**.** *We say a function ψ is a semivalue if ψ satisfies the linearity, null player, and symmetry axioms.*

In the following theorem, we now show how data values can be formulated without the efficiency axiom.

Theorem 2 (Representation of semivalues)**.** *A value function ψ_{semi} is a semivalue, if and only if, there*

exists a weight function $w^{(n)} : [n] \to \mathbb{R}$ *such that* $\sum_{j=1}^{n} \binom{n-1}{j-1} w^{(n)}(j) = n$ *and the value function* ψ_{semi} *can be expressed as follows.*

$$\psi_{\text{semi}}(z^*; U, \mathcal{D}, w^{(n)}) := \frac{1}{n} \sum_{j=1}^{n} \binom{n-1}{j-1} w^{(n)}(j) \Delta_j(z^*; U, \mathcal{D}). \tag{3}$$

Theorem 2 shows that every semivalue ψ_{semi} can be expressed as a weighted mean of marginal contributions without the efficiency axiom. Compared to data Shapley, a semivalue provides flexible formulation of data valuation and includes various existing data valuation methods. For example, when $w^{(n)}(j) = \binom{n-1}{j-1}^{-1}$, the semivalue $\psi_{\text{semi}}(z^*; U, \mathcal{D}, w^{(n)})$ becomes the data Shapley (Ghorbani and Zou, 2019), and when $w^{(n)}(j) = n\mathbb{1}(j = n)$, and it reduces to the LOO method. Moreover, for any Borel probability measure ξ defined on $[0, 1]$, a function $w^{(n)}$ defined as

$$w^{(n)}(j) := n \int_0^1 t^{j-1}(1-t)^{n-j} d\xi(t) \tag{4}$$

satisfies the condition $\sum_{j=1}^{n} \binom{n-1}{j-1} w^{(n)}(j) = n$, and thus the corresponding function $\psi_{\text{semi}}(z^*; U, \mathcal{D}, w^{(n)})$ is a semivalue by Theorem 2. We provide a detailed proof in Appendix C.

The semivalue expressed in (3) provides a unified data valuation framework, but loses the uniqueness of the weights. This can be a potential drawback, but the following proposition shows that a semivalue is unique up to the sum of data values.

Proposition 3. *Let* ψ_1 *and* ψ_2 *be two semivalues such that for any* U *the sum of data values are same, i.e.,*

$$\sum_{z \in \mathcal{D}} \psi_1(z; U, \mathcal{D}) = \sum_{z \in \mathcal{D}} \psi_2(z; U, \mathcal{D}).$$

Then, the two semivalues are identical, i.e., $\psi_1 = \psi_2$.

Proposition 3 shows that any two semivalues are identical if they have the same sum of data values. In other words, if there is a weight function that could reflect a practitioner's prior knowledge on the total sum of values, then the semivalue based on the weight function is unique.

4.2 Beta Shapley: Efficient Semivalue

The exact computation of $w^{(n)}(j)$ in Equation (4) can be expensive or infeasible due to the integral. To address this, we propose Beta Shapley that has a closed form for the weight. To be more specific, we consider

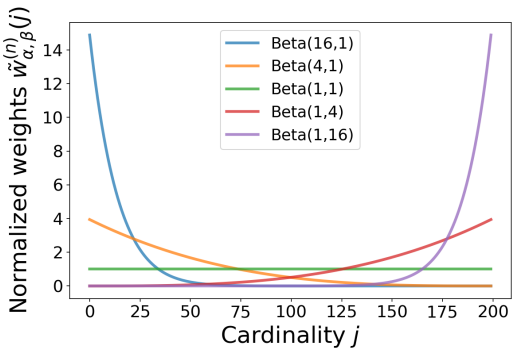

Figure 3: Illustration of the normalized weight $\tilde{w}^{(n)}_{\alpha,\beta}(j)$ for various pairs of (α, β) when $n = 200$. Each color indicates a different hyperparameter pair (α, β).

the Beta distribution with a pair of positive hyperparameters (α, β) for ξ. Then, the weight can be expressed as

$$w^{(n)}_{\alpha,\beta}(j) := n \int_0^1 t^{j-1}(1-t)^{n-j} \frac{t^{\beta-1}(1-t)^{\alpha-1}}{\text{Beta}(\alpha, \beta)} dt = n \frac{\text{Beta}(j + \beta - 1, n - j + \alpha)}{\text{Beta}(\alpha, \beta)},$$

where $\text{Beta}(\alpha, \beta) = \Gamma(\alpha)\Gamma(\beta)/\Gamma(\alpha + \beta)$ is the Beta function and $\Gamma(\cdot)$ is the Gamma function. This can be further simplified as the following closed form.

$$w^{(n)}_{\alpha,\beta}(j) = n \frac{\prod_{k=1}^{j-1}(\beta + k - 1) \prod_{k=1}^{n-j}(\alpha + k - 1)}{\prod_{k=1}^{n-1}(\alpha + \beta + k - 1)}. \tag{5}$$

We propose to use $\psi_{\text{semi}}(z^*; U, \mathcal{D}, w^{(n)}_{\alpha,\beta})$ and call it Beta(α, β)-Shapley value.

The pair of hyperparameters (α, β) decides the weight distribution on $[n]$. For instance, when $(\alpha, \beta) = (1, 1)$, the normalized weight $\tilde{w}^{(n)}_{\alpha,\beta}(j) := \binom{n-1}{j-1} w^{(n)}_{\alpha,\beta}(j) = 1$ for all $j \in [n]$, giving the uniform weight on marginal contributions, *i.e.*, Beta(1,1)-Shapley $\psi_{\text{semi}}(z^*; U, \mathcal{D}, w^{(n)}_{1,1})$ is exactly the original data Shapley. Figure 3 shows various weight distributions for different pairs of (α, β). For simplicity, we fix one of the hyperparameter to be one. When $\alpha \geq \beta = 1$, the normalized weight assigns large weights on the small cardinality and remove noise from the large cardinality. Conversely, Beta(1,β) puts more weights on large cardinality and it approaches to the LOO as β increases.

Beta(α, β)-Shapley for optimal subsampling weights When some data points in a given dataset are noisy or there are too many data points that could cause heavy computational costs, subsampling is an effective method to learn a model with a small number of high-quality samples. We show how Beta(α, β)-Shapley

can be used to find the optimal importance weight that produces an estimator with the minimum variance in an asymptotic manner. We consider the Beta Shapley $\psi_{\text{semi}}(z^*; h, \mathcal{D}, w^{(n)}_{\alpha,\beta})$ for an M-estimator h. The M-estimator h can include various estimators such as the maximum likelihood estimator (Van der Vaart, 2000).

Theorem 4 (Informal)**.** *Let λ_i be the importance weight for i-th sample z_i and suppose $\lambda_i \propto \left\|\psi_{\text{semi}}(z_i; h, \mathcal{D}, w^{(n)}_{\alpha,\beta})\right\|_2$ for some $\alpha \geq 1$. If we subsample based on the importance weight λ_i, then the M-estimator obtained by subsamples asymptotically achieves the minimum variance.*

We present a formal version of Theorem 4 with some technical conditions in Appendix B. This shows that the data value-based importance weight can be useful to capture the impact of one datum, and leads to the optimal estimator in the sense that it asymptotically achieves the minimum variance.

Efficient estimation of Beta(α, β)-Shapley Although $w^{(n)}_{\alpha,\beta}(j)$ is easy-to-compute, the exact computation of Beta(α, β)-Shapley can be expensive because it requires an exponential number of model fittings. This could be a major challenge of using Beta(α, β)-Shapley in practice. To address this, we develop an efficient algorithm by adapting the Monte Carlo method (Maleki et al., 2013; Ghorbani and Zou, 2019).

Beta(α, β)-Shapley can be expressed in the form of a weighted mean as follows.

$$\frac{1}{n}\sum_{j=1}^{n}\frac{1}{|\mathcal{D}_j^{\setminus z^*}|}\sum_{S\in\mathcal{D}_j^{\setminus z^*}}\tilde{w}^{(n)}_{\alpha,\beta}(j)(h(S\cup\{z^*\})-h(S)).$$

We note that $\tilde{w}^{(n)}_{\alpha,\beta}(j) = \binom{n-1}{j-1} w^{(n)}_{\alpha,\beta}(j)$ only needs to be calculated once using the closed form in Equation (5). As a result, it can be efficiently approximated by the Monte Carlo method: at each iteration, we first draw a number j from the discrete uniform distribution from $[n]$, and randomly draw a subset S is from a class of set $\mathcal{D}_j^{\setminus z^*}$ uniformly at random. We then compute $\tilde{w}^{(n)}_{\alpha,\beta}(j)(h(S\cup\{z^*\}) - h(S))$ and update the Monte Carlo estimates. We provide a pseudo algorithm in Appendix A.

5 NUMERICAL EXPERIMENTS

In this section, we demonstrate the practical efficacy of Beta Shapley on various classification datasets. We conduct the three different ML tasks: noisy label detection, learning with subsamples, and point addition and removal experiments. We compare the eight methods: the `LOO-First` $\Delta_2(z^*; U, \mathcal{D})$, the five variations of Beta(α, β)-Shapley, the `LOO-Last` $\Delta_n(z^*; U, \mathcal{D})$ (which is the standard LOO), and the `KNN Shapley` proposed in Jia et al. (2019a). We use 15 standard datasets that are commonly used to benchmark classification methods and use a logistic regression classifier. Additional results with a support vector machine model and detailed information about experiment settings are provided in Appendix D.

5.1 Noisy label detection

We first investigate the detection ability of Beta Shapley. As for detection rules, we use a clustering-based procedure as the number of mislabeled data points and the threshold for detecting noisy samples are usually unknown in practice. Specifically, we first divide all data values into two clusters using the K-Means clustering algorithm (Arthur and Vassilvitskii, 2007) and then classify a data point as a noisy sample if its value is less than the minimum of the two cluster centers. After this selection procedure, the F1-score is evaluated as a performance metric. We consider the two different types of label noise: synthetic noise and real-world label noise. As for the synthetic noise, we generate noisy samples by flipping labels for a random 10% of training data points.

Synthetic noise Table 1 shows the F1-score of the eight data valuation methods. `Beta(16,1)`, which assigns larger weights to small cardinality, outperforms other variations of Beta Shapley as well as the baseline data valuation methods. In contrast, `Beta(1,4)` or `LOO-Last` perform much worse than other methods because they focus heavily on large cardinalities. `LOO-First` $\Delta_2(z^*; U, \mathcal{D})$, which only considers the first marginal contribution, often suffers from a failure of training due to a small number of samples, and as a result, it performs worse than the `Beta(16,1)` and `Beta(4,1)` methods. This shows that focusing too much on only small cardinality can also degrade performance. We further compare Beta Shapley with the uncertainty-based method proposed by Northcutt et al. (2021a). Although this method is not intended for data valuation, it is a state-of-the-art method of noise label detection and achieves average $F_1 = 0.42$ across the 15 datasets. This score is worse than `Beta(16,1)` or `Beta(4,1)`, showing `Beta(16,1)` Shapley values are very competitive in identifying mislabeled data points.

Real-world label noise We next apply data valuation methods to detect real-world label errors in the CIFAR100 test dataset. Northcutt et al. (2021b) estimated 5.85% of the images in the CIFAR100 test dataset were indeed mislabeled. We choose the ten most confusing class pairs in the CIFAR100 test dataset. For each pair of classes, we detect mislabeled points

Table 1: Comparison of mis-annotation detection ability of the eight data valuation methods on the fifteen classification datasets. The average and standard error of the F1-score are denoted by 'average±standard error'. All the results are based on 50 repetitions. Boldface numbers denote the best method.

Dataset	LOO-First	Beta(16,1)	Beta(4,1)	Data Shapley	Beta(1,4)	Beta(1,16)	LOO-Last	KNN Shapley
Gaussian	0.465 ± 0.010	$\mathbf{0.470 \pm 0.010}$	0.454 ± 0.012	0.416 ± 0.012	0.255 ± 0.012	0.204 ± 0.011	0.147 ± 0.013	0.398 ± 0.012
Covertype	0.324 ± 0.011	$\mathbf{0.355 \pm 0.013}$	0.347 ± 0.013	0.337 ± 0.012	0.252 ± 0.008	0.236 ± 0.008	0.180 ± 0.010	0.278 ± 0.012
CIFAR10	0.252 ± 0.011	0.272 ± 0.011	$\mathbf{0.276 \pm 0.011}$	0.272 ± 0.012	0.238 ± 0.010	0.213 ± 0.008	0.169 ± 0.008	0.259 ± 0.010
FMNIST	0.487 ± 0.012	0.547 ± 0.012	$\mathbf{0.555 \pm 0.011}$	0.523 ± 0.013	0.356 ± 0.011	0.271 ± 0.011	0.187 ± 0.013	0.484 ± 0.017
MNIST	0.412 ± 0.010	0.482 ± 0.011	$\mathbf{0.504 \pm 0.012}$	0.477 ± 0.012	0.345 ± 0.011	0.284 ± 0.011	0.203 ± 0.013	0.446 ± 0.012
Fraud	$\mathbf{0.623 \pm 0.009}$	0.591 ± 0.016	0.550 ± 0.017	0.427 ± 0.022	0.221 ± 0.012	0.233 ± 0.015	0.177 ± 0.021	0.491 ± 0.013
Apsfail	0.624 ± 0.015	$\mathbf{0.643 \pm 0.011}$	0.606 ± 0.013	0.494 ± 0.019	0.229 ± 0.018	0.236 ± 0.017	0.242 ± 0.020	0.483 ± 0.014
Click	0.216 ± 0.008	$\mathbf{0.218 \pm 0.009}$	0.214 ± 0.009	0.201 ± 0.009	0.174 ± 0.009	0.167 ± 0.009	0.140 ± 0.011	0.204 ± 0.009
Phoneme	0.388 ± 0.011	0.409 ± 0.011	0.399 ± 0.011	0.350 ± 0.012	0.224 ± 0.011	0.192 ± 0.011	0.126 ± 0.014	$\mathbf{0.446 \pm 0.011}$
Wind	0.515 ± 0.013	$\mathbf{0.521 \pm 0.015}$	0.515 ± 0.016	0.501 ± 0.015	0.248 ± 0.011	0.226 ± 0.012	0.163 ± 0.016	0.505 ± 0.015
Pol	0.432 ± 0.012	0.451 ± 0.011	$\mathbf{0.471 \pm 0.011}$	0.461 ± 0.012	0.229 ± 0.010	0.210 ± 0.010	0.184 ± 0.015	0.446 ± 0.014
Creditcard	0.268 ± 0.010	$\mathbf{0.270 \pm 0.010}$	0.265 ± 0.012	0.238 ± 0.012	0.194 ± 0.011	0.180 ± 0.011	0.152 ± 0.010	0.259 ± 0.010
CPU	$\mathbf{0.659 \pm 0.013}$	0.638 ± 0.012	0.613 ± 0.014	0.555 ± 0.021	0.281 ± 0.018	0.250 ± 0.015	0.212 ± 0.021	0.559 ± 0.012
Vehicle	0.448 ± 0.015	0.469 ± 0.015	$\mathbf{0.484 \pm 0.016}$	0.456 ± 0.014	0.360 ± 0.013	0.287 ± 0.012	0.217 ± 0.015	0.310 ± 0.014
2Dplanes	0.518 ± 0.009	0.526 ± 0.012	0.505 ± 0.012	0.460 ± 0.013	0.278 ± 0.013	0.240 ± 0.012	0.177 ± 0.018	$\mathbf{0.568 \pm 0.014}$
Average	0.442	**0.458**	0.451	0.411	0.259	0.228	0.178	0.409

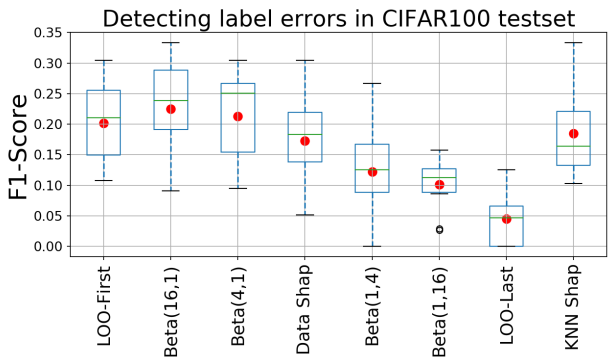

Figure 4: A boxplot of the F1-score of the data valuation methods on the CIFAR100 test dataset. The red dot indicates the mean of the F1-score. Beta Shapley that focuses on small cardinality detects mislabeled data points better than other baseline methods.

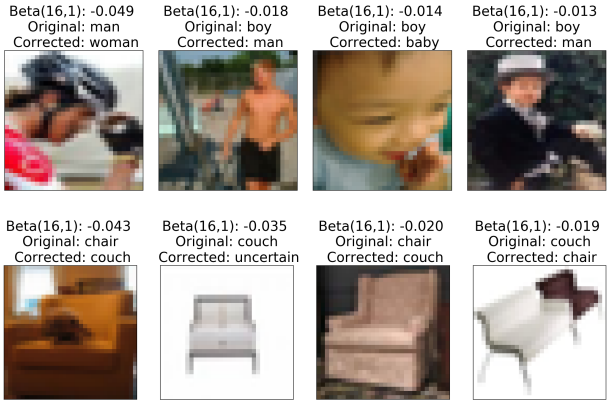

Figure 5: Examples of mislabeled images in the CIFAR100 test dataset. The corrected label suggested by Northcutt et al. (2021b) is provided for comparison. `Beta(16,1)` values for the mislabeled samples are negative, meaning that this type of labeling error can harm the model.

in a binary classification setting. Figure 4 shows a boxplot of the F1-score of the different data valuation methods. `Beta(16,1)` achieves 0.225 and outperforms other methods. For the three pairs of classes with the largest number of mislabels—(willow tree, maple tree), (pine tree, oak tree), (oak tree, maple tree)—we also compared the detection performance of `Beta(16,1)` with the state-of-the-art uncertainty-based method proposed in Northcutt et al. (2021a). The uncertainty-based method was developed in the same paper that identified the CIFAR100 misannotations, so it is a strong benchmark. `Beta(16,1)` and the uncertainty-based methods achieve 0.307 and 0.273 F1-score, respectively, showing `Beta(16,1)` is effective in identifying real-world label errors. Figure 5 shows representative examples of mislabeled images and Beta Shapley rightfully assigned negative values for them.

5.2 Learning with subsamples

We now examine how the data value-based importance weight can be applied to subsample data points. We train a model with 25% of the given dataset by using the importance weight $\max(\psi_{\text{semi}}(z_i; U, \mathcal{D}, w_{\alpha,\beta}^{(n)}), 0)$ for the i-th sample. With this importance weight, data points with higher values are more likely to be selected and data points with negative values are not used. We train a classifier to minimize the weighted loss then evaluate the accuracy on the held-out test dataset. As Table 2 shows, `Beta(4,1)` shows the best overall performance, but `Beta(16,1)` also shows very similar performance. `Beta(1,4)` and `LOO-Last` perform worse than uniform sampling, suggesting that the marginal

Table 2: Accuracy comparison of models trained with subsamples. We compare the eight data valuation methods on the fifteen classification datasets. The `Random` denotes learning a model with subsamples drawn uniformly at random. The average and standard error of classification accuracy are denoted by 'average±standard error'. All the results are based on 50 repetitions. Boldface numbers denote the best method.

Dataset	LOO-First	Beta(16,1)	Beta(4,1)	Data Shapley	Beta(1,4)	LOO-Last	KNN Shapley	Random
Gaussian	0.763 ± 0.003	$\mathbf{0.765 \pm 0.002}$	0.760 ± 0.003	0.732 ± 0.005	0.598 ± 0.008	0.569 ± 0.015	0.749 ± 0.005	0.727 ± 0.007
Covertype	0.645 ± 0.005	0.661 ± 0.005	$\mathbf{0.670 \pm 0.005}$	0.661 ± 0.004	0.607 ± 0.007	0.567 ± 0.008	0.635 ± 0.006	0.636 ± 0.006
CIFAR10	0.625 ± 0.004	$\mathbf{0.628 \pm 0.003}$	0.624 ± 0.003	0.617 ± 0.003	0.575 ± 0.004	0.553 ± 0.006	0.580 ± 0.004	0.582 ± 0.005
FMNIST	0.823 ± 0.004	$\mathbf{0.842 \pm 0.003}$	0.840 ± 0.004	0.830 ± 0.003	0.726 ± 0.008	0.614 ± 0.012	0.801 ± 0.006	0.752 ± 0.007
MNIST	0.762 ± 0.004	$\mathbf{0.773 \pm 0.004}$	0.770 ± 0.003	0.753 ± 0.004	0.673 ± 0.006	0.607 ± 0.009	0.725 ± 0.005	0.702 ± 0.006
Fraud	0.883 ± 0.003	0.881 ± 0.003	0.883 ± 0.002	0.873 ± 0.006	0.567 ± 0.019	0.637 ± 0.037	$\mathbf{0.886 \pm 0.003}$	0.866 ± 0.004
Apsfail	0.866 ± 0.003	0.877 ± 0.003	$\mathbf{0.878 \pm 0.003}$	0.870 ± 0.004	0.671 ± 0.023	0.477 ± 0.034	0.864 ± 0.003	0.858 ± 0.003
Click	0.566 ± 0.004	$\mathbf{0.567 \pm 0.003}$	0.566 ± 0.003	0.561 ± 0.004	0.535 ± 0.004	0.520 ± 0.004	0.551 ± 0.004	0.538 ± 0.005
Phoneme	0.741 ± 0.002	$\mathbf{0.744 \pm 0.003}$	0.743 ± 0.002	0.738 ± 0.003	0.581 ± 0.009	0.567 ± 0.017	0.727 ± 0.004	0.712 ± 0.005
Wind	0.801 ± 0.003	0.804 ± 0.002	0.809 ± 0.003	0.796 ± 0.004	0.569 ± 0.013	0.549 ± 0.028	$\mathbf{0.811 \pm 0.003}$	0.800 ± 0.003
Pol	0.750 ± 0.004	0.731 ± 0.004	0.748 ± 0.004	0.746 ± 0.005	0.543 ± 0.013	0.532 ± 0.018	$\mathbf{0.762 \pm 0.006}$	0.734 ± 0.005
Creditcard	0.625 ± 0.003	0.632 ± 0.003	$\mathbf{0.637 \pm 0.003}$	0.632 ± 0.004	0.571 ± 0.006	0.528 ± 0.006	0.595 ± 0.004	0.584 ± 0.007
CPU	0.848 ± 0.004	0.870 ± 0.003	$\mathbf{0.872 \pm 0.004}$	0.862 ± 0.004	0.628 ± 0.015	0.545 ± 0.029	0.862 ± 0.004	0.858 ± 0.004
Vehicle	0.754 ± 0.005	0.770 ± 0.003	$\mathbf{0.772 \pm 0.004}$	0.761 ± 0.005	0.675 ± 0.009	0.592 ± 0.010	0.729 ± 0.005	0.728 ± 0.007
2Dplanes	0.802 ± 0.003	$\mathbf{0.806 \pm 0.002}$	0.803 ± 0.003	0.796 ± 0.003	0.631 ± 0.009	0.615 ± 0.018	0.777 ± 0.005	0.755 ± 0.006
Average	0.750	**0.757**	**0.757**	0.749	0.612	0.564	0.735	0.722

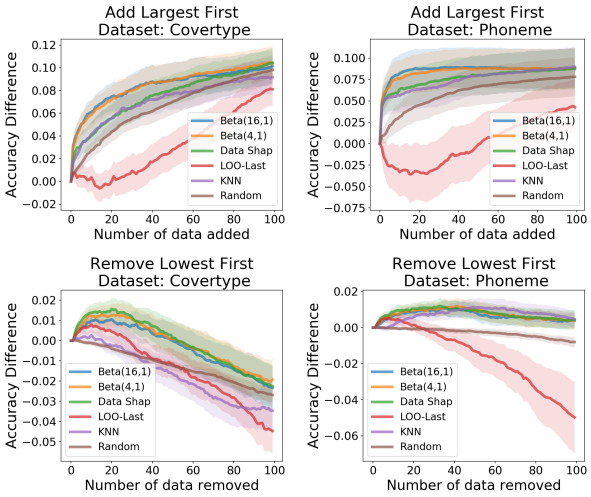

Figure 6: Accuracy difference as a function of the number of data points (top) added or (bottom) removed. We add (*resp.* remove) data points whose value is large (*resp.* small) first. We denote a 95% confidence band based on 50 repetitions. We provide additional results on different datasets in Appendix D.

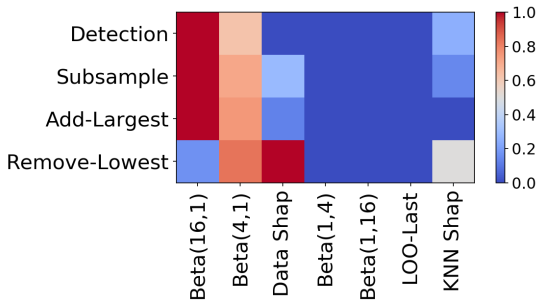

Figure 7: A summary of performance comparison on the fifteen datasets. Each element of the heatmap represents a linearly scaled frequency for each task to be between 0 and 1. Better and worse methods are depicted in red and blue respectively.

contributions at large cardinality are not useful to capture the importance of data.

5.3 Point addition and removal experiments

We now conduct point addition and removal experiments which are used to evaluate previous data valuation methods Ghorbani and Zou (2019). For point addition experiments, we add data points from largest to lowest values because adding helpful data points first is expected to increase performance (similar to active learning setup). For the removal experiments, we remove data points from lowest to largest values because it is desirable to remove harmful or noisy data points first to increase model performance. At each step of addition or removal, we retrain a model with the current dataset and evaluate the accuracy changes on the held-out test dataset. For point addition, `Beta(16,1)` shows the most rapid gain from identifying valuable points by putting more weights on small cardinality marginals (Figure 6). For removal, data Shapley performs slightly better than other methods. This is because we remove data points from the entire dataset, so the uniform weight can capture the effect of large cardinality parts better than other methods.

Finally, we summarize all of our experiments in a heatmap (Figure 7). For each ML task and data valuation method, we count the number of datasets where

the method is the best performer and linearly transform it to be between 0 and 1. `Beta(16,1)`, which focuses on small cardinalities, is consistently the best method on the detection, subsampling, and point addition tasks. Data Shapley, which is `Beta(1,1)` and puts equal weights on all cardinalities, is the top performer in the point removal task.

6 CONCLUDING REMARKS

This work develops Beta Shapley to unify and extend popular data valuation methods like LOO and Data Shapley. Beta Shapley has desirable statistical and computational properties. We find that marginal contributions based on small cardinality are likely to have larger signal-to-noise, which is why `Beta(16,1)` works well in many settings. Our extensive experiments show that Beta Shapley that weighs small cardinalities more (e.g. `Beta(16,1)`) outperforms Data Shapley, LOO and other state-of-the-art methods.

There are many interesting future works in this area. Our sampling-based algorithm provides an efficient implementation of data valuation, but the development of scalable algorithms that can be applied to a large-scale dataset is critical for the practical use of data values in practice. As an orthogonal direction, Beta Shapley opens up a new question about how to define and obtain the optimal weight representing data values. We believe it can depend on several factors including ML task or data distribution.

Acknowledgments

We thank AISTATS 2022 anonymous reviewers for their helpful feedback. This research is supported by funding from Stanford AI Lab, Chan-Zuckerberg Biohub and the NSF CAREER #1942926.

References

Arthur, D. and Vassilvitskii, S. (2007). k-means++ the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035.

Banzhaf III, J. F. (1964). Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.*, 19:317.

Blackard, J. A. (1998). *Comparison of neural networks and discriminant analysis in predicting forest cover types*. Colorado State University.

Chen, J., Song, L., Wainwright, M. J., and Jordan, M. I. (2019). L-shapley and c-shapley: Efficient model interpretation for structured data. In *International Conference on Learning Representations*.

Cohen, S., Ruppin, E., and Dror, G. (2005). Feature selection based on the shapley value. In *Proceedings of the 19th international joint conference on Artificial intelligence*, pages 665–670.

Cook, R. D. and Weisberg, S. (1980). Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22(4):495–508.

Cook, R. D. and Weisberg, S. (1982). *Residuals and influence in regression*. New York: Chapman and Hall.

Covert, I. and Lee, S.-I. (2021). Improving kernelshap: Practical shapley value estimation using linear regression. In *International Conference on Artificial Intelligence and Statistics*, pages 3457–3465. PMLR.

Covert, I., Lundberg, S., and Lee, S.-I. (2021). Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research*, 22(209):1–90.

Covert, I., Lundberg, S. M., and Lee, S.-I. (2020). Understanding global feature contributions with additive importance measures. *Advances in Neural Information Processing Systems*, 33.

Dal Pozzolo, A., Caelen, O., Johnson, R. A., and Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166. IEEE.

DiCiccio, C. and Romano, J. P. (2020). Clt for u-statistics with growing dimension.

Duarte, M. F. and Hu, Y. H. (2004). Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826–838.

Dubey, P., Neyman, A., and Weber, R. J. (1981). Value theory without efficiency. *Mathematics of Operations Research*, 6(1):122–128.

Dubey, P. and Weber, R. J. (1977). Probabilistic values for games. Technical report, YALE UNIV NEW HAVEN CONN COWLES FOUNDATION FOR RESEARCH IN ECONOMICS.

Gautschi, W. (1959). Some elementary inequalities relating to the gamma and incomplete gamma function. *J. Math. Phys*, 38(1):77–81.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1995). *Bayesian data analysis*. Chapman and Hall/CRC.

Ghorbani, A., Kim, M., and Zou, J. (2020). A distributional framework for data valuation. In *International Conference on Machine Learning*, pages 3535–3544. PMLR.

Ghorbani, A. and Zou, J. (2019). Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251.

Ghorbani, A. and Zou, J. Y. (2020). Neuron shapley: Discovering the responsible neurons. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5922–5932. Curran Associates, Inc.

Gilboa, I. and Monderer, D. (1991). Quasi-values on subspaces. *International Journal of Game Theory*, 19(4):353–363.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distribution. *The Annals of Mathematical Statistics*, 19(3):293–325.

Hoeffiding, W. (1948). A class of statistics with asymptotically normal distributions. *Annals of Mathematical Statistics*, 19(3):293–325.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Gurel, N. M., Li, B., Zhang, C., Spanos, C., and Song, D. (2019a). Efficient task-specific data valuation for nearest neighbor algorithms. *Proceedings of the VLDB Endowment*, 12(11):1610–1623.

Jia, R., Dao, D., Wang, B., Hubis, F. A., Hynes, N., Gürel, N. M., Li, B., Zhang, C., Song, D., and Spanos, C. J. (2019b). Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176.

Kalai, E. and Samet, D. (1987). On weighted shapley values. *International journal of game theory*, 16(3):205–222.

Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR.

Krizhevsky, A. et al. (2009). Learning multiple layers of features from tiny images.

Kwon, Y., Rivas, M. A., and Zou, J. (2021). Efficient computation and analysis of distributional shapley values. In *International Conference on Artificial Intelligence and Statistics*, pages 793–801. PMLR.

LeCun, Y., Cortes, C., and Burges, C. (2010). Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2020). From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774.

Maleki, S., Tran-Thanh, L., Hines, G., Rahwan, T., and Rogers, A. (2013). Bounding the estimation error of sampling-based shapley value approximation. *arXiv preprint arXiv:1306.4265*.

Monderer, D. and Samet, D. (2002a). Chapter 54 variations on the shapley value. volume 3 of *Handbook of Game Theory with Economic Applications*, pages 2055–2076. Elsevier.

Monderer, D. and Samet, D. (2002b). Variations on the shapley value. *Handbook of game theory with economic applications*, 3:2055–2076.

Monderer, D., Samet, D., and Shapley, L. S. (1992). Weighted values and the core. *International Journal of Game Theory*, 21(1):27–39.

Northcutt, C., Jiang, L., and Chuang, I. (2021a). Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411.

Northcutt, C. G., Athalye, A., and Mueller, J. (2021b). Pervasive label errors in test sets destabilize machine learning benchmarks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ridaoui, M., Grabisch, M., and Labreuche, C. (2018). An axiomatisation of the banzhaf value and interaction index for multichoice games. In *International Conference on Modeling Decisions for Artificial Intelligence*, pages 143–155. Springer.

Rozemberczki, B. and Sarkar, R. (2021). The shapley value of classifiers in ensemble games. *arXiv preprint arXiv:2101.02153*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Särndal, C.-E., Swensson, B., and Wretman, J. (2003). *Model assisted survey sampling*. Springer Science & Business Media.

Shapley, L. S. (1953a). *Additive and non-additive set functions*. Princeton University.

Shapley, L. S. (1953b). A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317.

Sim, R. H. L., Zhang, Y., Chan, M. C., and Low, B. K. H. (2020). Collaborative machine learning with incentive-aware model rewards. In *International Conference on Machine Learning*, pages 8927–8936. PMLR.

Sundararajan, M. and Najmi, A. (2020). The many shapley values for model explanation. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9269–9278. PMLR.

Ting, D. and Brochu, E. (2018). Optimal subsampling with influence functions. In *Advances in neural information processing systems*, pages 3650–3659.

Tricomi, F. G., Erdélyi, A., et al. (1951). The asymptotic expansion of a ratio of gamma functions. *Pacific Journal of Mathematics*, 1(1):133–142.

Van der Vaart, A. W. (2000). *Asymptotic statistics*, volume 3. Cambridge university press.

Vats, D. and Knudson, C. (2021). Revisiting the gelman–rubin diagnostic. *Statistical Science*, 36(4):518–529.

Wang, R., Wang, X., and Inouye, D. I. (2021). Shapley explanation networks. In *International Conference on Learning Representations*.

Weber, R. J. (1988). Probabilistic values for games. *The Shapley Value. Essays in Honor of Lloyd S. Shapley*, pages 101–119.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Yeh, I.-C. and Lien, C.-h. (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2):2473–2480.

Zaeri-Amirani, M., Afghah, F., and Mousavi, S. (2018). A feature selection method based on shapley value to false alarm reduction in icus a genetic-algorithm approach. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 319–323. IEEE.

Supplementary Material: Beta Shapley: a Unified and Noise-reduced Data Valuation Framework for Machine Learning

In Appendix, we provide implementation details in Section A, additional explanations in Section B, and proofs in Section C. In addition, we provide additional numerical results and demonstrate robustness of our results against different datasets and a model using a support vector machine in Section D. Our implementation codes are available at `https://github.com/ykwon0407/beta_shapley`.

A IMPLEMENTATION DETAILS

The proposed algorithm We propose to use the sampling-based Monte Carlo (MC) method to approximate Beta Shapley value: at each iteration, we first draw a number k from the discrete uniform distribution from $[n]$, and randomly draw a subset S is from a class of set $\mathcal{D}_k^{\backslash z^*}$ uniformly at random. We then compute $\tilde{w}_{\alpha,\beta}^{(n)}(k)(h(S\cup\{z^*\})-h(S))$ and update the MC estimates. As for the utility computation, we can use a held-out validation set of samples from P_Z.

Accuracy of the proposed algorithms The propose algorithm is based on the MC method, and thus it guarantees to converge to the true value if we repeat the sampling procedure. In our experiments, we stop the sampling procedure when the new increment is small enough compared to the current MC estimate. To do this, we evaluate the Gelman-Rubin statistic for data values, which is well known for one of the most popular convergence diagnostic methods (Vats and Knudson, 2021, Equation (4)). We set the number of Markov chains as 10 and terminate the sampling procedure if the Gelman-Rubin statistic for all data values is less than 1.0005 to ensure accurate approximation, which is much less than a typical terminating threshold 1.1 (Gelman et al., 1995). We provide a pseudo algorithm in Algorithm 1.

Algorithm 1 Efficient computation algorithm for Beta(α,β)-Shapley

Require: A set to be valued $\mathcal{D}=\{z_1,\ldots,z_n\}$. A utility function U. A terminating threshold ρ (in our experiment $\rho=1.0005$).
procedure
  Initialize $\hat{\rho}=2\rho$, $B=1$, $\nu(j)=0$ for all $j\in[n]$.
  Compute $\tilde{w}_{\alpha,\beta}^{(n)}(j)=\binom{n-1}{j-1}w_{\alpha,\beta}^{(n)}(j)$ for all $j\in[n]$.
  while $\hat{\rho}\geq\rho$ **do**
    for $j\in[n]$ **do**
      Sample k a uniform distribution from $[n]$.
      Sample $S\in\mathcal{D}_k^{\backslash z_j}$ uniformly at random.
      Update $\nu(j)\leftarrow\frac{B-1}{B}\nu(j)+\frac{1}{B}\tilde{w}_{\alpha,\beta}^{(n)}(k)(U(S\cup\{z_j\})-U(S))$
    end for
    Update the Gelman-Rubin statistic $\hat{\rho}$.
    $B\leftarrow B+1$.
  end while
end procedure

Datasets used in Figure 1 of the manuscript We use the two synthetic datasets, regression and classification settings. As for the regression dataset, we generate input data from a 10-dimensional multivariate Gaussian with zero mean and the identity covariance matrix, *i.e.*, $x_i\sim\mathcal{N}(0,I_{10})$. The output is generated as $y_i=x_i^T\beta_0+\varepsilon_i$, where $\beta_0\sim\mathcal{N}(0,I_{10})$ and $\varepsilon_i\sim\mathcal{N}(0,1)$. As for the classification dataset, we generate input data $x_i\sim\mathcal{N}(0,I_3)$. For

Table 3: A summary of datasets used in numerical experiments.

Dataset	Sample size	Input dimension	Source
`Gaussian`	50000	5	Synthetic dataset
`Covertype`	581012	54	Blackard (1998)
`CIFAR10`	60000	32	Krizhevsky et al. (2009)
`Fashion-MNIST`	60000	32	Xiao et al. (2017)
`MNIST`	60000	32	LeCun et al. (2010)
`Fraud`	284807	31	Dal Pozzolo et al. (2015)
`Creditcard`	30000	24	Yeh and Lien (2009)
`Vehicle`	98528	101	Duarte and Hu (2004)
`Apsfail`	76000	171	`https://www.openml.org/d/41138`
`Click`	1997410	12	`https://www.openml.org/d/1218`
`Phoneme`	5404	6	`https://www.openml.org/d/1489`
`Wind`	6574	15	`https://www.openml.org/d/847`
`Pol`	15000	49	`https://www.openml.org/d/722`
`CPU`	8192	22	`https://www.openml.org/d/761`
`2DPlanes`	40768	11	`https://www.openml.org/d/727`

outputs, we draw from a Bernoulli distribution $y_i = \mathbf{Bern}(\pi_i)$ for all $i \in [n]$. Here $\pi_i := \exp(x_i^T\beta)/(1+\exp(x_i^T\beta))$ for $\beta = (5, 0, 0)$.

Datasets used in Figure 2 and Section 5 of the manuscript We use the one synthetic dataset and the fourteen real datasets. For the synthetic dataset, `Gaussian`, we generate data as follows. Given a sample size n, we generate input data from a 5-dimensional multivariate Gaussian with zero mean and the identity covariance matrix, *i.e.*, $x_i \sim \mathcal{N}(0, I_5)$. For outputs, we draw from a Bernoulli distribution $y_i = \mathbf{Bern}(\pi_i)$ for all $i \in [n]$. Here $\pi_i := \exp(x_i^T\beta)/(1+\exp(x_i^T\beta))$ for $\beta = (2, 1, 0, 0, 0)$. For the real datasets, we collect datasets from multiple sources including `OpenML`[1]. A comprehensive list of datasets and details on sample size and data source are provided in Table 3. We preprocess datasets to ease the training.

If the original dataset is the multi-class classification dataset (e.g. `Covertype`), we binarize the label by considering $\mathbb{1}(y = 1)$. For `OpenML` and `Covertype` datasets, we consider oversampling a minor class to balance positive and negative labels. For the image datasets `Fashion-MNIST`, `MNIST` and `CIFAR10`, we follow the common procedure in prior works (Ghorbani et al., 2020; Kwon et al., 2021): we extract the penultimate layer outputs from the pre-trained ResNet18 (He et al., 2016). The pre-training is done with the ImageNet dataset (Russakovsky et al., 2015) and the weight is publicly available from `Pytorch` (Paszke et al., 2019). Using the extracted outputs, we fit a principal component analysis model and select the first 32 principal components.

A.1 Model

Throughout the experiment, we use a logistic regression model or a support vector machine model using the Python module `scikit-learn` (Pedregosa et al., 2011). As for KNN Shapley (Jia et al., 2019b), we used the k-nearest neighborhood classifier with $k = 10$.

A.2 Experiment settings

As for Figure 1 of the manuscript, we consider 500, 500, and 2000 samples for the dataset to be valued $\mathcal{D}$, the validation dataset, and the held-out test dataset, respectively. The validation dataset is used to estimate utility, and all the results are based on this held-out dataset. Except for this experiment, we use 200 samples for $\mathcal{D}$ and 200 samples for the validation dataset. For the held-out test dataset, we randomly choose 1000 samples. As for the experiments in Figure 2 and Section 5, we randomly flip a label for 10% of samples for $\mathcal{D}$ and the validation datasets. Below we provide details on ML tasks.

[1] https://www.openml.org/

Noisy label detection Suppose $z^{(1)}, \ldots, z^{(n)}$ are data points such that they satisfy the ordering $\psi(z^{(1)}) \leq \cdots \leq \psi(z^{(n)})$. We fit the K-Means clustering algorithm on $\{\psi(z^{(1)}), \ldots, \psi(z^{(n)})\}$, diving into the two non-intersect sets $\{\psi(z^{(1)}), \ldots, \psi(z^{(B)})\}$ and $\{\psi(z^{(B+1)}), \ldots, \psi(z^{(n)})\}$. Note that B is not necessarily the number of noisy samples. We define a detection rule as follows: we select z is noisy if the data value is less than or equal to the lower cluster mean. That is, if $\psi(z) \leq \frac{1}{B}\sum_{i=1}^{B}\psi(z^{(i)})$, then z is a noisy data point. After that we compute a F1-score, a harmonic mean of precision and recall of the rule, where

$$\text{Recall} = \frac{|\{z : z \text{ is flipped and selected by the rule}\}|}{|\{z : z \text{ is flipped}\}|}$$
$$\text{Precision} = \frac{|\{z : z \text{ is flipped and selected by the rule}\}|}{|\{z : z \text{ is selected by the rule}\}|}.$$

Learning with subsamples We consider a situation where we select 50 samples among 200 samples, which is 25% of $\mathcal{D}$. For a data valuation ν, let $\lambda_i(\nu) = \max(\nu(z_i), 0)$ be the importance weight for sample $z_i = (x_i, y_i)$. Then, we compare the test accuracy of a weighted risk minimizer f_ν defined as

$$f_\nu := \text{argmin}_f \sum_{j \in \mathcal{S}_{50}} \frac{1}{\lambda_i(\nu)}(y_i \log f(x_i) + (1 - y_i)\log(1 - f(x_i))),$$

where $\mathcal{S}_{50}$ is a set of the 50 subsamples. Here, the inverse weight $\frac{1}{\lambda_i(\nu)}$ is used to consider an unbiased risk minimizer. Note that this inverse propensity is used in the Horvitz-Thompson empirical measure. After obtaining f_ν, we compute unweighted test accuracy using the held-out test dataset.

Point addition and removal experiments In Figure 7, we use the relative area as the performance of point addition and removal experiments. Specifically, for the removal task, we compute the relative area as follows.

$$\text{Relative area-removal}(\psi) := \sum_{k=1}^{n/2}\left\{U(\mathcal{D}\backslash\{z^{(1)}, \ldots, z^{(k)}\}) - U(\mathcal{D})\right\},$$

where $z^{(1)}, \ldots, z^{(n)}$ satisfy the ordering $\psi(z^{(1)}) \leq \cdots \leq \psi(z^{(n)})$. Similarly, for the addition task, we consider

$$\text{Relative area-addition}(\psi) := \sum_{k=1}^{n/2}\left\{U(\mathcal{S} \cup \{z^{(n)}, \ldots, z^{(n-k+1)}\}) - U(\mathcal{S})\right\},$$

where $\mathcal{S}$ is the initial set. In our experiment, $\mathcal{S}$ is randomly selected from $\mathcal{D}$ and $|\mathcal{S}| = 10$.

B ADDITIONAL DETAILS

In this section, we provide further details regarding Theorem 1, Theorem 4, and the Equation 4 of the manuscript.

B.1 A bound condition in Theorem 1

One drawback of Theorem 1 is that it is unknown whether $\lim_{j\to\infty}\zeta_j/(j\zeta_1)$ is bounded. Although Theorem 1 in Hoeffiding (1948) showed a lower bound is greater than 1, *i.e.*, $1 \leq \zeta_j/(j\zeta_1)$ for any j, the existence of an upper bound has not been shown in literature. In Figure 8, we show that this condition is plausible in our numerical examples. The details on dataset are given in Appendix A.

B.2 Details on Theorem 4.

For fixed positive constants c_1 and c_2, let $\Pi(c_1, c_2)$ be a set of measure Q such that (i) the total measure of Q is c_1 and (ii) it mutually absolutely continuous with P_Z and $c_2 \leq dQ/dP_Z \leq 1$. For $Q \in \Pi(c_1, c_2)$, let $\mathbb{P}_n^Q := \frac{1}{n}\sum_{i=1}^{n} A_i \frac{dP_Z}{dQ}(Z_i)\delta_{Z_i}$ be the Horvitz–Thompson empirical measure, where A_i be a Bernoulli random variable with a probability $dQ/dP_Z(Z_i)$, and δ_z be the Dirac delta measure on $\mathcal{Z}$ (Särndal et al., 2003).

Ting and Brochu (2018) showed $\sqrt{\frac{n}{c_1}}(h(\mathbb{P}_n^Q) - h(P_Z))$ converges in distribution to a Gaussian distribution with zero mean and ν^Q variance, where $\nu^Q := \int \mathcal{I}(z)\mathcal{I}(z)^T(dP_Z/dQ(z))dP_Z(z)$ and $\mathcal{I}$ is the Hadamard derivative of

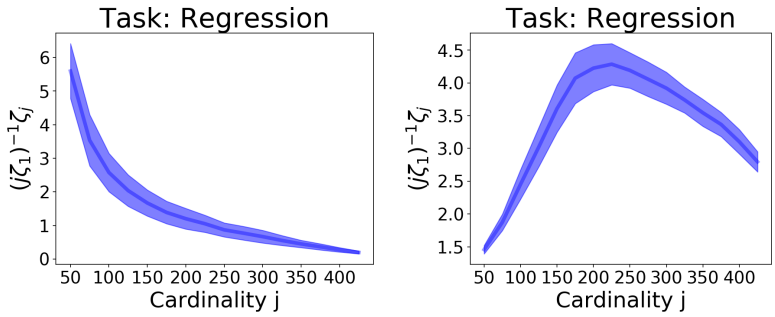

Figure 8: Illustration of $\zeta_j/(j\zeta_1)$ as a function of cardinality j in (left) linear regression and (right) logistic regression settings. The curve and band indicate mean and 95% standard error of $\zeta_j/(j\zeta_1)$ among different samples. Note that the quantity $\zeta_j/(j\zeta_1)$ decreases as the cardinality increases and it empirically shows the validity of the condition in Theorem 1.

h at P_Z, which is the influence function when it exists. Since the variance ν^Q is the function of $Q \in \Pi(c_1, c_2)$, the problem of finding the optimal subsampling weights can be formulated as finding Q that minimizes the trace of variance, *i.e.*, $\text{argmin}_{Q\in\Pi(c_1,c_2)}\text{Tr}(\nu^Q)$. In the following theorem, we show that the Beta Shapley-based Horvitz-Thompson empirical measure produces the optimal estimator with the smallest variance.

Theorem 5 (Formal version of Theorem 4)**.** *Suppose h is Hadamard differentiable at P_Z and the importance weight λ_i for i-th sample z_i is $\lambda_i \propto \left\|\psi_{\text{semi}}(z_i; h, \mathcal{D}, w^{(n)}_{\alpha,\beta})\right\|_2$ and $n^{-1}\sum_{i=1}^n \lambda_i = c_1$ for some $\beta \geq 1$. If there is $\mathbb{P}_n^{Q_\psi} \in \Pi(c_1, c_2)$ such that $dQ_\psi/dP_Z(Z_i) = \lambda_i$, then the asymptotic variance of $h(\mathbb{P}_n^{Q_\psi})$ is $\min_{Q\in\Pi(c_1,c_2)}\text{Tr}(\nu^Q)$.*

Theorem 5 shows asymptotic convergence of $\text{Var}(h(\mathbb{P}_n^{Q_\psi}))$, unfortunately, its convergence rate is unknown. We believe different choices of (α, β) can affect the convergence rate, and thus it can be used to choose the optimal hyperparameter.

B.3 Derivation of Equation 4

Proof of Equation 4. For $j \in [n]$ and any probability density function ξ defined on $[0, 1]$, we set

$$w^{(n)}(j, \xi) := n\int_0^1 t^{j-1}(1-t)^{n-j}\xi(t)dt.$$

Then, we have

$$\begin{aligned}\frac{1}{n}\sum_{j=1}^n w^{(n)}(j,\xi)\binom{n-1}{j-1} &= \int_0^1 \sum_{j=1}^n \binom{n-1}{j-1} t^{j-1}(1-t)^{n-j}\xi(t)dt \\ &= \int_0^1 \xi(t)dt = 1.\end{aligned}$$

The last equality is due to the definition of ξ. This concludes a proof. □

C PROOFS

Proof of Theorem 1. The first result, $(j^2\zeta_1/n)^{-1}\text{Var}(\Delta_j(z^*, U, \mathfrak{D})) \to 1$ as n increases, is from the central limit theorem of U-statistics when j increases. Our result is from Theorem 3.1(i) of DiCiccio and Romano (2020). □

Proof of Theorem 2. If there exists a weight function $w^{(n)} : [n] \to \mathbb{R}$ such that $\sum_{j=1}^n \binom{n-1}{j-1} w^{(n)}(j) = n$ and the value function ψ_{semi} can be expressed as

$$\psi_{\text{semi}}(z^*; U, \mathcal{D}, w^{(n)}) = \frac{1}{n}\sum_{j=1}^n \binom{n-1}{j-1} w^{(n)}(j)\Delta_j(z^*; U, \mathcal{D}).$$

Then by the form of the function $\psi_{\text{semi}}(z^*; U, \mathcal{D}, w^{(n)})$, it satisfies the linearity, null player and symmetry axioms.

Now, we show the inverse. By Theorem 3 of Dubey and Weber (1977), when a function satisfies the linearity and null player, there exists a weight function $\mu : \cup_{j=0}^{\infty} \mathcal{Z}^j \to \mathbb{R}$ such that $\sum_{S \subseteq \mathcal{D}\setminus\{z^*\}} \mu^{(n)}(S) = 1$ and the value function ψ can be expressed as

$$\psi(z^*; U, \mathcal{D}, \mu^{(n)}) = \sum_{S \subseteq \mathcal{D}\setminus\{z^*\}} \mu^{(n)}(S)(U(S \cup \{z^*\}) - U(S)).$$

In addition, due to the symmetry axiom, if there are two subsets S_1 and S_2 such that $|S_1| = |S_2|$ and $S_1, S_2 \subseteq \mathcal{D}\setminus\{z^*\}$, then $\mu(S_1) = \mu(S_2)$.[2] Therefore, for $\nu : [n] \to \mathbb{R}$, we can further simplify ψ as follows.

$$\begin{aligned}\psi(z^*; U, \mathcal{D}, \mu^{(n)}) &= \sum_{j=1}^{n} \sum_{S \subseteq \mathcal{D}_j^{\setminus\{z^*\}}} \mu^{(n)}(S)(U(S \cup \{z^*\}) - U(S)) \\ &= \sum_{j=1}^{n} \nu(j) \sum_{S \subseteq \mathcal{D}_j^{\setminus\{z^*\}}} (U(S \cup \{z^*\}) - U(S)),\end{aligned}$$

where $\sum_{j=1}^{n} \nu(j)\binom{n-1}{j-1} = 1$. Thus, considering $w^{(n)}(j) := n^{-1}\nu(j)$, it concludes a proof. □

Proof of Proposition 3. The uniqueness of semivalues shown in Proposition 3 is directly from Theorem 10 of Monderer and Samet (2002a). □

Proof of Theorem 4. We provide a proof for Theorem 5, which is a detailed version of Theorem 4. Theorem 5 is specified in Appendix B.2. A key component of this proof is to show for some sequence of explicit constants $(\gamma_{\alpha,\beta}^{(j)})$, $(\gamma_{\alpha,\beta}^{(n)}/n)^{-1}\psi_{\text{semi}}(z^*; U, \mathcal{D}, w_{\alpha,\beta}^{(n)})$ converges to the influence function as n increases by using the Silverman-Toeplitz theorem.

To be more specific,

$$\begin{aligned}\psi_{\text{semi}}(z^*; h, \mathcal{D}, w_{\alpha,\beta}^{(n)}) &= \frac{1}{n}\sum_{j=1}^{n} w_{\alpha,\beta}^{(n)}(j) \sum_{S \subseteq \mathcal{D}_j^{\setminus z^*}} h(S \cup \{z^*\}) - h(S) \\ &= \frac{1}{n}\sum_{j=1}^{n} w_{\alpha,\beta}^{(n)}(j)\binom{n-1}{j-1}\frac{1}{j}\frac{1}{\binom{n-1}{j-1}} \sum_{S \subseteq \mathcal{D}_j^{\setminus z^*}} \frac{h(S \cup \{z^*\}) - h(S)}{1/j} \\ &= \frac{\gamma_{\alpha,\beta}^{(n)}}{n}\sum_{j=1}^{n} \frac{w_{\alpha,\beta}^{(n)}(j)\binom{n-1}{j-1}\frac{1}{j}}{\gamma_{\alpha,\beta}^{(n)}}\psi_j(z^*, h),\end{aligned}$$

where

$$\begin{aligned}\gamma_{\alpha,\beta}^{(n)} &:= \sum_{j=1}^{n} w_{\alpha,\beta}^{(n)}(j)\binom{n-1}{j-1}\frac{1}{j} \\ \psi_j(z^*, h) &:= \frac{1}{\binom{n-1}{j-1}} \sum_{S \subseteq \mathcal{D}_j^{\setminus z^*}} \frac{h(S \cup \{z^*\}) - h(S)}{1/j}.\end{aligned}$$

Note that $\psi_j(z^*, h) \to \mathcal{I}(z^*; h, P_Z)$ as $j \to \infty$ because $\mathcal{I}$ is a Hadamard derivative of h at P_Z. We formally denote this by $\mathcal{I}(z^*; h, P_Z)$. That is, if

$$\lim_{n\to\infty} (\gamma_{\alpha,\beta}^{(n)})^{-1}\left(w_{\alpha,\beta}^{(n)}(j)\binom{n-1}{j-1}\frac{1}{j}\right) = 0$$

[2] For a function $v_S(A) := \mathbb{1}(S \subsetneq A)$ and a permutation π^* that sends $\pi^*(S_1) = S_2$ and fixes others, $\mu(S_1) = \psi(z^*; v_{S_1}, \mathcal{D}, \mu^{(n)}) = \psi(z^*; v_{S_2}, \mathcal{D}, \mu^{(n)}) = \mu(S_2)$.

for all fixed $j \in [n]$, then by the Silverman-Toeplitz theorem, we have $(\gamma_{\alpha,\beta}^{(n)}/n)^{-1}\psi_{\text{semi}}(z^*; h, \mathcal{D}, w_{\alpha,\beta}^{(n)}) \to \mathcal{I}(z^*; h, P_Z)$.

Note that

$$\begin{aligned} w_{\alpha,\beta}^{(n)}(j)\binom{n-1}{j-1}\frac{1}{j} &= \frac{\text{Beta}(j+\beta-1, n-j+\alpha)}{\text{Beta}(\alpha,\beta)}\binom{n}{j} \\ &= \frac{1}{\text{Beta}(\alpha,\beta)}\binom{n}{j}\int_0^1 t^{j+\beta-2}(1-t)^{n-j+\alpha-1}dt, \end{aligned}$$

and thus

$$\begin{aligned} \gamma_{\alpha,\beta}^{(n)} &= \sum_{j=1}^{n} w_{\alpha,\beta}^{(n)}(j)\binom{n-1}{j-1}\frac{1}{j} \\ &= \frac{1}{\text{Beta}(\alpha,\beta)}\int_0^1 \sum_{j=1}^{n}\binom{n}{j}t^{j+\beta-2}(1-t)^{n-j+\alpha-1}dt \\ &= \frac{1}{\text{Beta}(\alpha,\beta)}\int_0^1 t^{\beta-2}(1-t)^{\alpha-1}\sum_{j=1}^{n}\binom{n}{j}t^{j}(1-t)^{n-j}dt \\ &= \frac{1}{\text{Beta}(\alpha,\beta)}\int_0^1 t^{\beta-2}(1-t)^{\alpha-1}\left(1-(1-t)^n\right)dt \\ &= \frac{1}{\text{Beta}(\alpha,\beta)}\int_0^1 (1-t)^{\beta-2}t^{\alpha-1}\left(1-t^n\right)dt \\ &= \frac{1}{\text{Beta}(\alpha,\beta)}\int_0^1 (1-t)^{\beta-1}t^{\alpha-1}\frac{1-t^n}{1-t}dt \\ &= \frac{1}{\text{Beta}(\alpha,\beta)}\int_0^1 (1-t)^{\beta-1}\sum_{k=0}^{n-1}t^{\alpha-1+k}dt \\ &= \frac{1}{\text{Beta}(\alpha,\beta)}\sum_{k=0}^{n-1}\text{Beta}(\alpha+k,\beta). \end{aligned}$$

[Step 1] When $\beta > 1$, due to $\text{Beta}(\alpha+k,\beta-1) + \text{Beta}(\alpha+k-1,\beta) = \text{Beta}(\alpha+k-1,\beta-1)$ for any $k \in [n]$, we have

$$\begin{aligned} \gamma_{\alpha,\beta}^{(n)} &= \frac{1}{\text{Beta}(\alpha,\beta)}\left(\text{Beta}(\alpha,\beta-1) - \text{Beta}(\alpha+n,\beta-1)\right) \\ &= \left(\frac{\text{Beta}(\alpha,\beta-1)}{\text{Beta}(\alpha,\beta)} - \frac{\Gamma(\beta-1)}{\text{Beta}(\alpha,\beta)}\frac{\Gamma(n+\alpha)}{\Gamma(n+\beta+\alpha-1)}\right) \\ &\approx \left(\frac{\text{Beta}(\alpha,\beta-1)}{\text{Beta}(\alpha,\beta)} - \frac{\Gamma(\beta-1)}{\text{Beta}(\alpha,\beta)}n^{1-\beta}\right) = O(1). \end{aligned}$$

The last approximation is due to Equation (1) of Tricomi et al. (1951) when n is large enough.

[Step 2] When $\beta \leq 1$, we have

$$\begin{aligned} \gamma_{\alpha,\beta}^{(n)} &= \sum_{k=0}^{n-1}\frac{\text{Beta}(\alpha+k,\beta)}{\text{Beta}(\alpha,\beta)} \\ &= \sum_{k=0}^{n-1}\frac{\Gamma(\alpha+k)\Gamma(\beta)}{\Gamma(\alpha+\beta+k)}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \\ &= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)}\sum_{k=0}^{n-1}\frac{\Gamma(\alpha+k)}{\Gamma(\alpha+\beta+k)} \end{aligned}$$

$$= \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)} \sum_{k=0}^{n-1} \frac{\Gamma(1+\alpha+k)}{\Gamma(\alpha+\beta+k)} \frac{1}{\alpha+k}$$
$$\approx \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)} \sum_{k=0}^{n-1} (\alpha+k)^{1-\beta} \frac{1}{\alpha+k}.$$

The approximation is from Equation (7) of Gautschi (1959) when $0 < \beta \leq 1$. Therefore, we have

$$\gamma_{\alpha,\beta}^{(n)} \approx \begin{cases} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)} \log \frac{n+\alpha}{\alpha} = O(\log n), & \text{if } \beta = 1 \\ \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)} \frac{(n+\alpha)^{1-\beta} - \alpha^{1-\beta}}{1-\beta} = O(n^{1-\beta}), & \beta < 1 \end{cases}.$$

[Step 3] For a fixed s, we have

$$\begin{aligned} w_{\alpha,\beta}^{(n)}(j) \binom{n-1}{j-1} \frac{1}{j} &= n \frac{\text{Beta}(j+\beta-1, n-j+\alpha)}{\text{Beta}(\alpha,\beta)} \frac{(n-1)!}{(j-1)!(n-j)!} \frac{1}{j} \\ &= n \frac{\Gamma(j+\beta-1)\Gamma(n-j+\alpha)}{\Gamma(n+\alpha+\beta-1)} \frac{\Gamma(n)}{j!\Gamma(n-j+1)} \frac{1}{\text{Beta}(\alpha,\beta)} \\ &= n \frac{\Gamma(n-j+\alpha)\Gamma(n)}{\Gamma(n+\alpha+\beta-1)\Gamma(n-j+1)} \frac{\Gamma(j+\beta-1)}{\Gamma(j+1)} \frac{1}{\text{Beta}(\alpha,\beta)} \\ &\approx n^{1-\beta} \frac{\Gamma(j+\beta-1)}{\Gamma(j+1)} \frac{1}{\text{Beta}(\alpha,\beta)} = O(n^{1-\beta}). \end{aligned}$$

[Step 4] Now we consider $\lim_{n\to\infty} (\gamma_{\alpha,\beta}^{(n)})^{-1} w_{\alpha,\beta}^{(n)}(j) \binom{n-1}{j-1} \frac{1}{j}$. In case of $\beta > 1$, from [Step 1] and [Step 3], we have $\gamma_{\alpha,\beta}^{(n)} = O(1)$ and $w_{\alpha,\beta}^{(n)}(j) \binom{n-1}{j-1} \frac{1}{j} = O(n^{1-\beta})$, thus $\lim_{n\to\infty} (\gamma_{\alpha,\beta}^{(n)})^{-1} w_{\alpha,\beta}^{(n)}(j) \binom{n-1}{j-1} \frac{1}{j} = 0$. Similarly, in case of $\beta = 1$, from [Step 2] and [Step 3], we also have $\lim_{n\to\infty} (\gamma_{\alpha,\beta}^{(n)})^{-1} w_{\alpha,\beta}^{(n)}(j) \binom{n-1}{j-1} \frac{1}{j} = 0$. Thus, by Silverman-Toeplitz theorem, if $\beta \geq 1$, then

$$\frac{n}{\gamma_{\alpha,\beta}^{(n)}} \psi_{\text{semi}}(z^*; h, \mathcal{D}, w_{\alpha,\beta}^{(n)}) = \mathcal{I}(z^*; h, P_Z) + o_p(1).$$

Moreover, by Theorem 2 of Ting and Brochu (2018), learning with a Horvitz–Thompson empirical measure defined with the importance weight λ_i such that $\lambda_i \propto \left\| \frac{n}{\gamma_{\alpha,\beta}^{(n)}} \psi_{\text{semi}}(z^*; h, \mathcal{D}, w_{\alpha,\beta}^{(n)}) \right\|_2 \propto \left\| \psi_{\text{semi}}(z^*; h, \mathcal{D}, w_{\alpha,\beta}^{(n)}) \right\|_2$ gives an estimator with the minimum variance. □

D ADDITIONAL EXPERIMENTS

D.1 Additional results using different datasets

In Figure 9, we shows the signal-to-noise ratio of $\Delta_j(z^*; U, \mathfrak{D})$ as a function of the cardinality j for the four real datasets. As in other figures, we denote a 95% confidence band based on 50 repetitions under the assumption that the results follow the identical Gaussian distribution. In Figure 10, we illustrate the marginal contributions as a function of the cardinality using the eleven datasets. This figure shares the same setting with Figure 2 of the manuscript, but different datasets are used. In Figures 11 and 12, we present additional point addition and removal experiment results using the thirteen datasets. Details on datasets and experiments settings are provided in Section A.

D.2 Beta Shapley with a support vector machine model

Throughout our experiments, we considered a logistic model to compute a utility. In this section, we demonstrate our results are robust against different models, in particular, a support vector machine model. In the following experiments, we use the same experiment setup but a model. Figure 13 shows the marginal contributions for clean and noisy samples as a function of the cardinality. Similar to the case of a logistic regression model, the

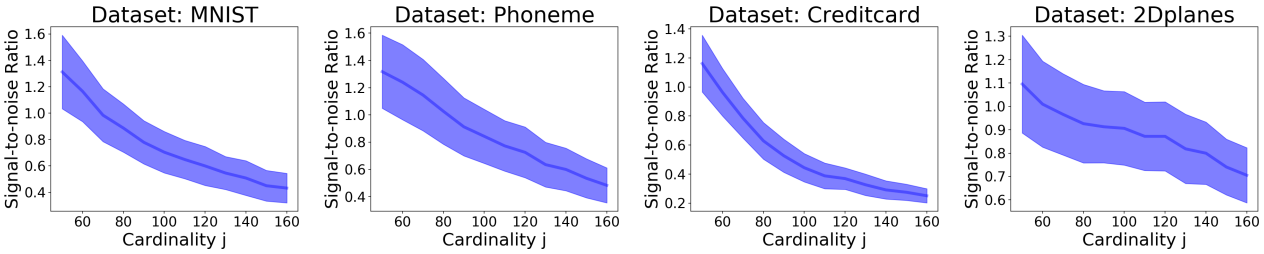

Figure 9: The signal-to-noise ratio of $\Delta_j(z^*; U, \mathfrak{D})$ as a function of the cardinality j for four real datasets. Similar to Figure 1, the signal-to-noise ratio decreases as the cardinality j increases.

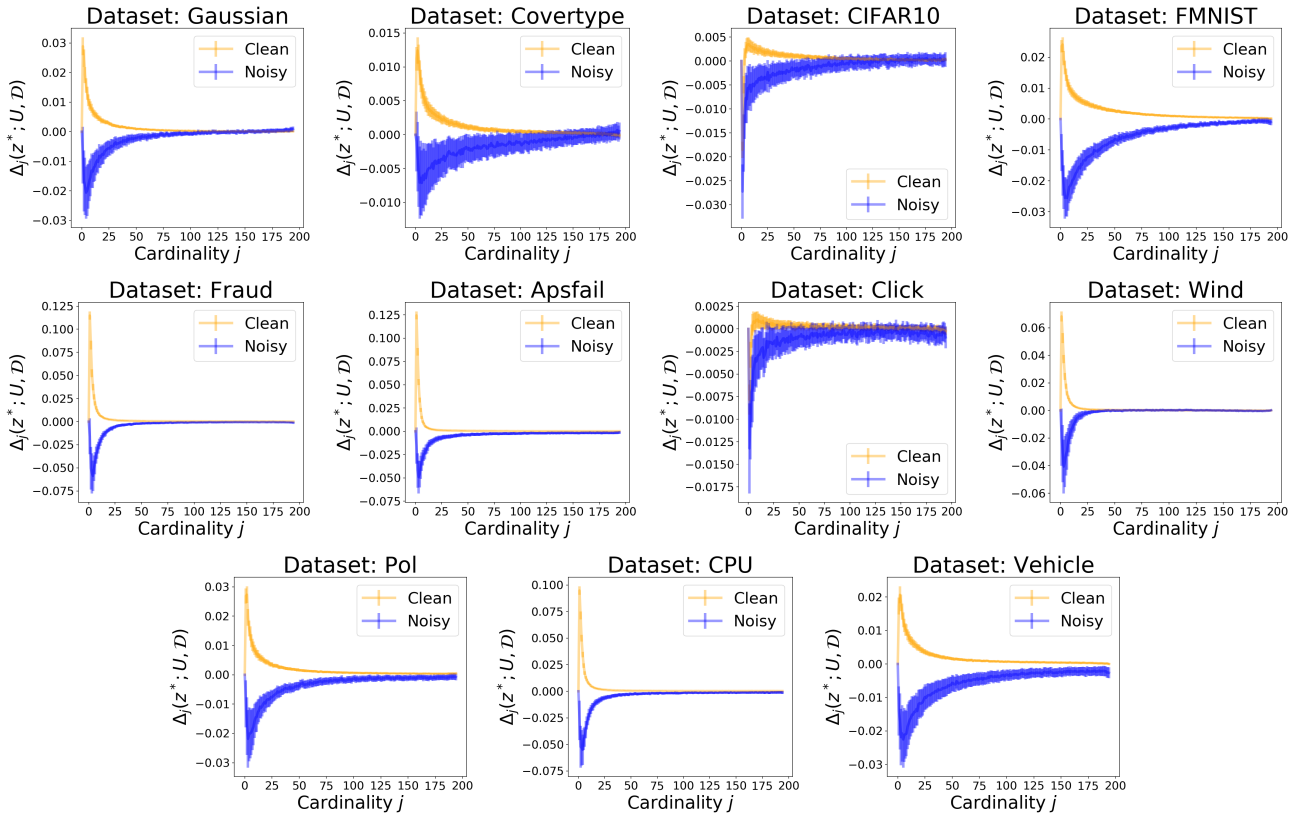

Figure 10: Illustrations of the marginal contributions $\Delta_j(z^*; U, \mathcal{D})$ as a function of the cardinality j on the eleven datasets. Each color indicates a noisy (blue) and a clean (yellow) data point. When the cardinality j is large, it is hard to tell if point is noisy or not as they become similar or even reversed.

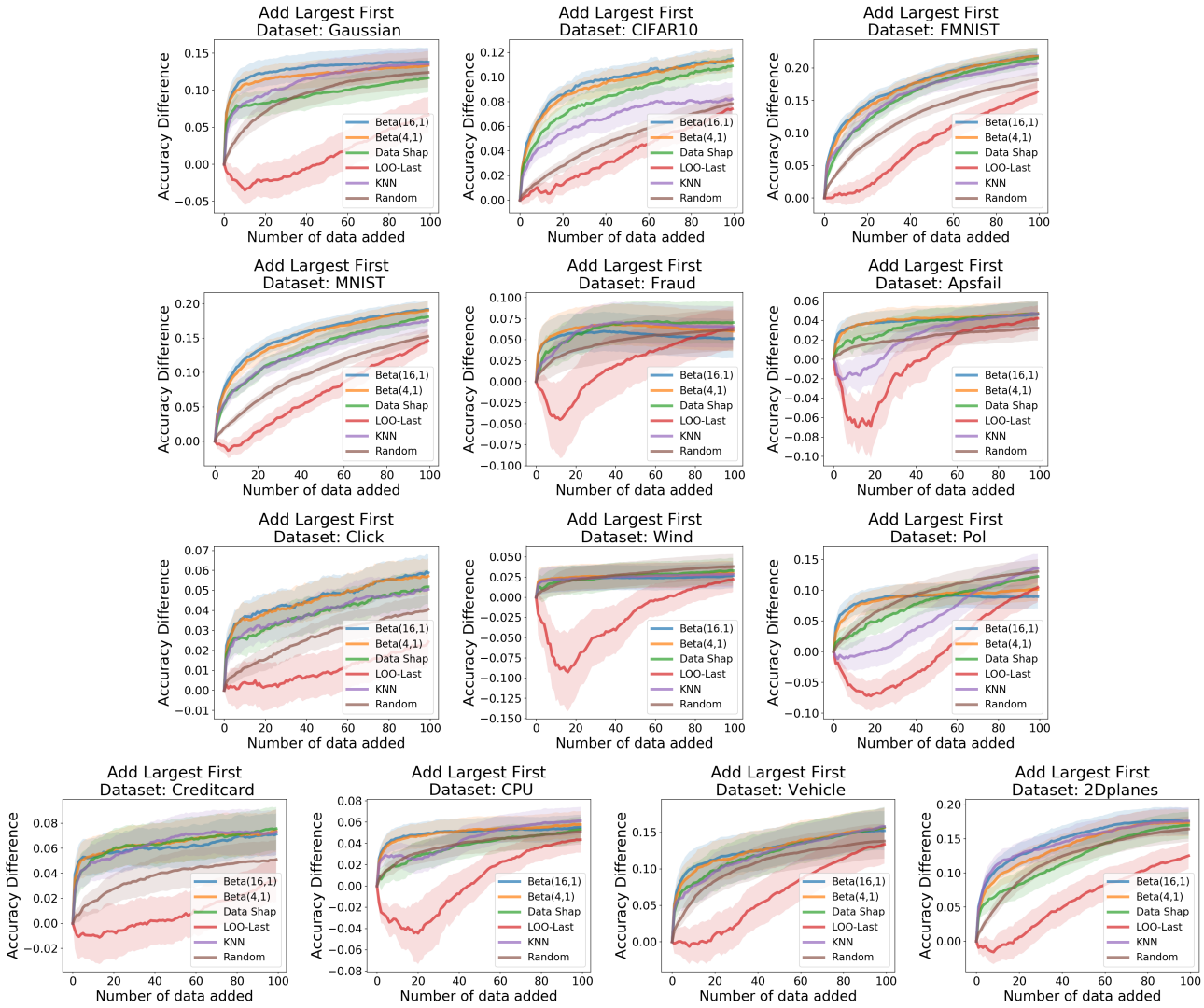

Figure 11: Accuracy change as a function of the number of data points added on the thirteen datasets. We add data points whose value is small first.

difference between clean and noisy groups is significantly big when the cardinality is small, but they overlap when the cardinality is big. This suggests the uniform weight used in data Shapley might not be optimal, but Beta Shapley can be more effective. Figure 14 shows a summary of performance comparison on the fifteen datasets when a support vector machine is used.

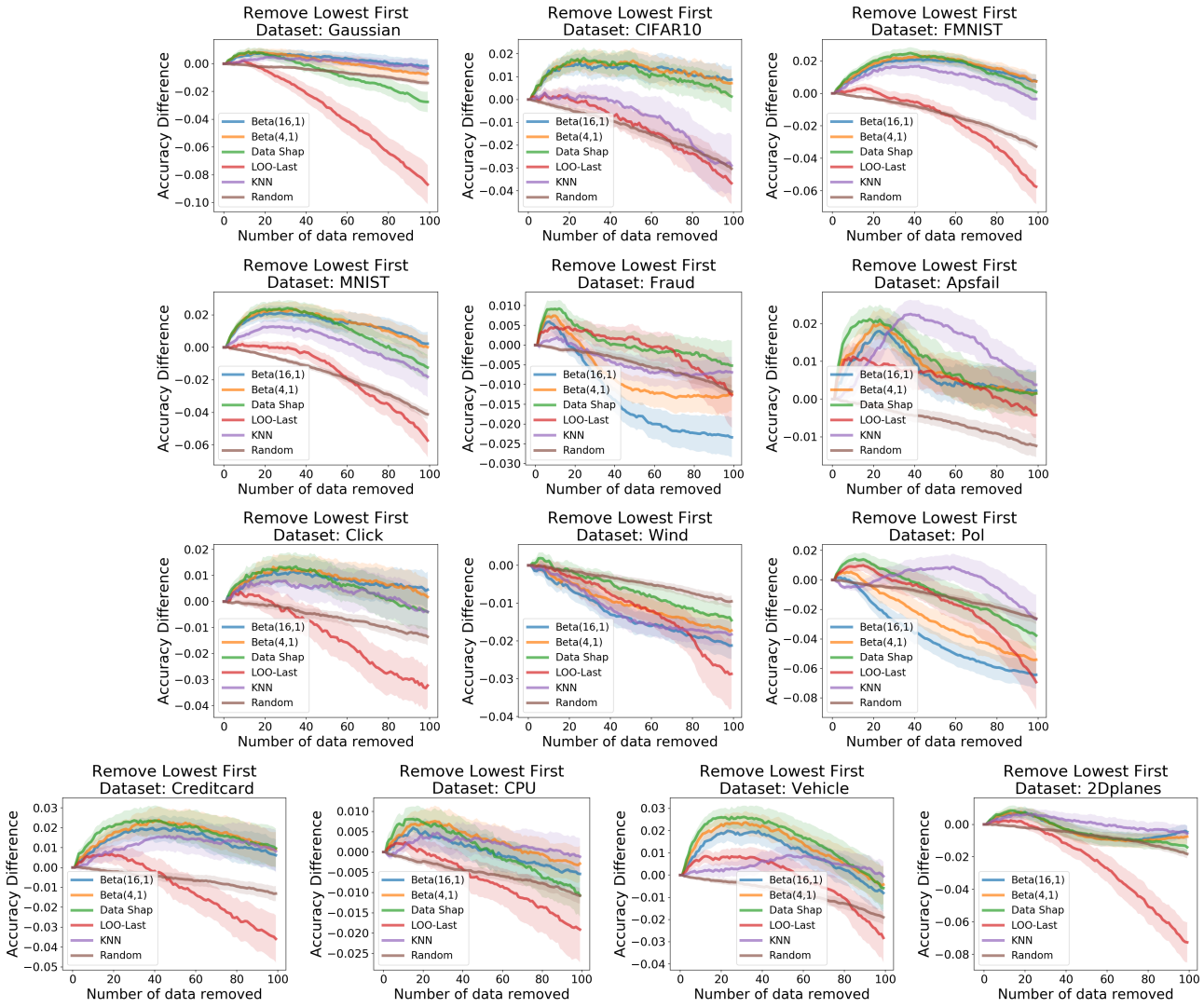

Figure 12: Accuracy change as a function of the number of data points removed on the thirteen datasets. We remove data points whose value is small first.

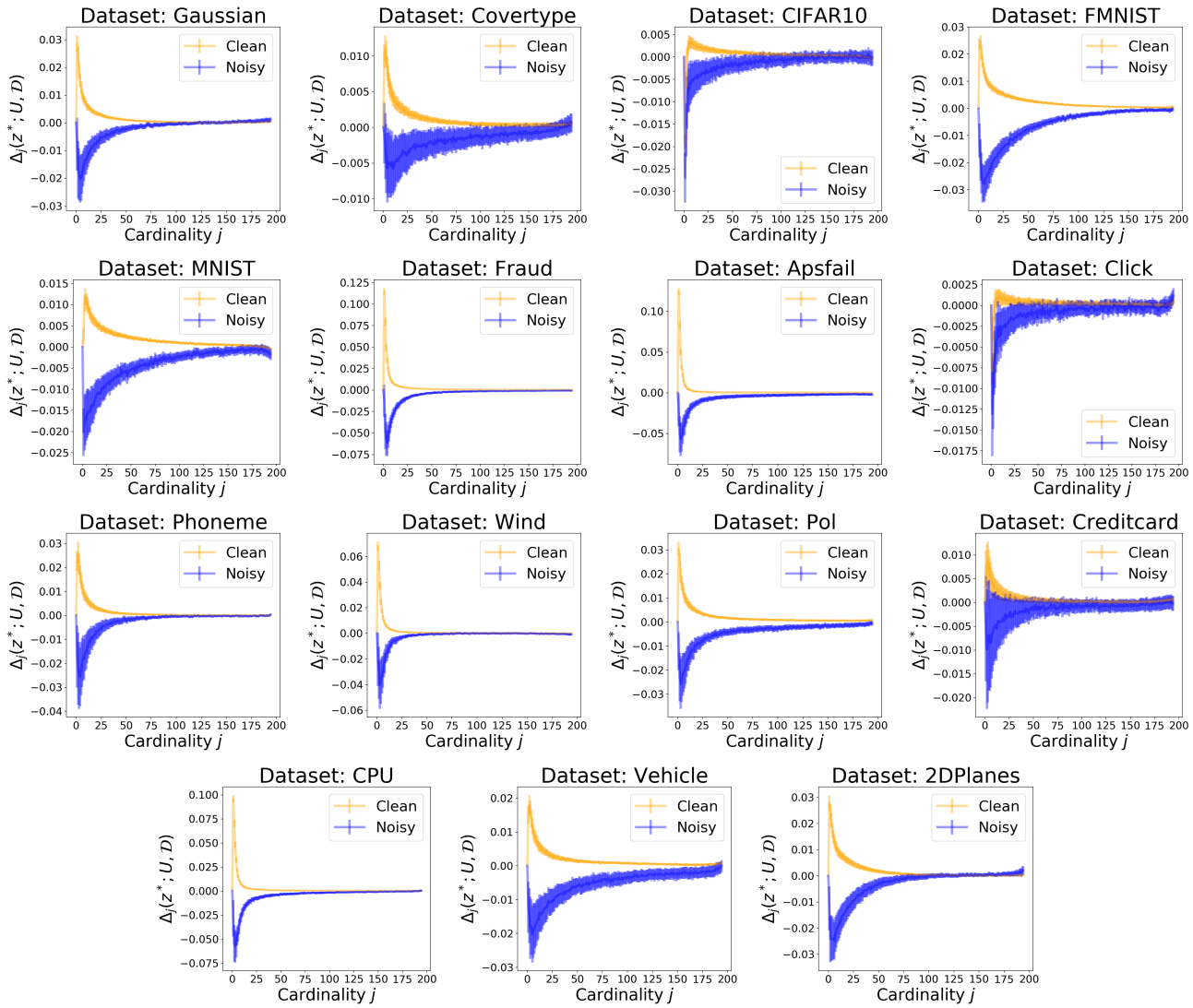

Figure 13: Illustrations of the marginal contributions $\Delta_j(z^*; U, \mathcal{D})$ as a function of the cardinality j on the eleven datasets when a support vector machine is used. Each color indicates a noisy (blue) and a clean (yellow) data point. We denote a 99% confidence band based on 50 independent runs. When the cardinality j is large, it is hard to tell if point is noisy or not as they become similar or even reversed.

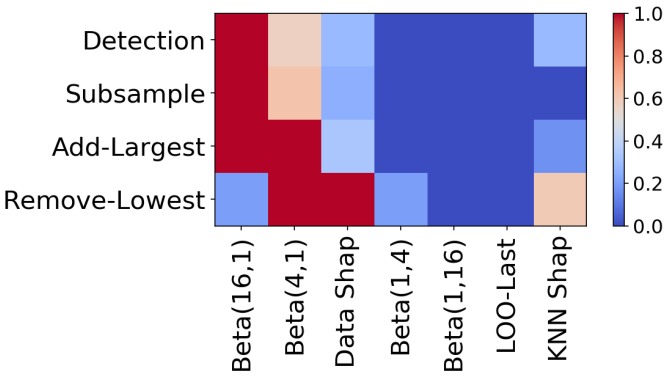

Figure 14: A summary of performance comparison on the fifteen datasets when a support vector machine is used. Each element of the heatmap represents a linearly scaled frequency for each task to be between 0 and 1. Better and worse methods are depicted in red and blue respectively.