Abstract
Data valuation is a powerful framework for providing statistical insights into which data are beneficial or detrimental to model training. Many Shapley-based data valuation methods have shown promising results in various downstream tasks, however, they are well known to be computationally challenging as it requires training a large number of models. As a result, it has been recognized as infeasible to apply to large datasets. To address this issue, we propose Data-OOB, a new data valuation method for a bagging model that utilizes the out-of-bag estimate. The proposed method is computationally efficient and can scale to millions of data by reusing trained weak learners. Specifically, Data-OOB takes less than 2.25 hours on a single CPU processor when there are $10^6$ samples to evaluate and the input dimension is 100. Furthermore, Data-OOB has solid theoretical interpretations in that it identifies the same important data point as the infinitesimal jackknife influence function when two different points are compared. We conduct comprehensive experiments using 12 classification datasets, each with thousands of sample sizes. We demonstrate that the proposed method significantly outperforms existing state-of-the-art data valuation methods in identifying mislabeled data and finding a set of helpful (or harmful) data points, highlighting the potential for applying data values in real-world applications.

1. Introduction
Assessing the impact of data on a model’s performance is important as it enhances our understanding of the data. Moreover, it has various practical real-world applications including medical image analysis, data curation, and data marketplaces (Tang et al., 2021; Agarwal et al., 2019; Tian et al., 2022). Due to its importance, data valuation has become a primary research topic in machine learning and statistics. The main goal is to establish a practical and principled notion of the influence of individual data points on the process of training a model.

A standard approach for evaluating the impact of data is to use the marginal contribution, which is defined as the average change in a model’s performance when a certain datum is removed from a set of data points. One approach uses the leave-one-out (LOO) method, which evaluates a single marginal contribution when a certain datum is removed from the entire training dataset (Cook & Weisberg, 1982; Koh & Liang, 2017). Another approach is based on the Shapley value in cooperative game theory (Shapley, 1953), assigning a simple average of all marginal contributions by varying the number of data points in a given subset of the training dataset. Data Shapley (Ghorbani & Zou, 2019), Distributional Shapley (Ghorbani et al., 2020), and CS-Shapley (Schoch et al., 2022) belong to this category.

Existing works have shown that Shapley-based methods perform better than LOO in many downstream tasks by leveraging every possible marginal contribution (Ghorbani & Zou, 2019; Jia et al., 2019a). However, it often requires training a significant number of models to accurately estimate marginal contributions. This has been recognized as the primary limitation in practical applications of data valuation. To address this issue, improved approximation algorithms using stratified sampling have been developed (Maleki et al., 2013; Wu et al., 2022), but they still require training multiple models to achieve a small enough approximation error. As an alternative approach, data values with a closed-form expression have been studied. It can easily scale up to millions of data but restricts the model to be either k-Nearest Neighbors (Jia et al., 2019a) or linear models (Kwon et al., 2021), which may not be most suitable for high dimensional data analysis. Recently, Lin et al. (2022) proposed an algorithm that uses the linear coefficient of a LASSO model. Their method is shown to have improved sample efficiency over existing methods, but it still requires unverifiable assumptions and additional computational costs for training LASSO models.

In addition to the computational challenges, the Shapley
value and its variants have critical limitations. They are based on the fair division axioms in cooperative game theory, but the axioms’ relevance to machine learning applications is unclear (Sim et al., 2022; Rozemberczki et al., 2022).

Our contributions In this paper, we propose Data-OOB, a new data valuation framework for a bagging model that uses the out-of-bag (OOB) estimate as illustrated in Figure 1. Our framework is computationally efficient by leveraging trained weak learners and is even faster than KNN-Shapley which has a closed-form expression. Furthermore, Data-OOB is statistically interpretable in that under mild assumptions it identifies the same important data point as the infinitesimal jackknife influence function when two different points are compared. Our comprehensive experiments demonstrate that the proposed method significantly better identifies mislabeled data and determines which data points are beneficial or detrimental for a model’s performance than existing state-of-the-art data valuation methods.

2. Preliminaries

For $d \in \mathbb{N}$, we denote an input space and an output space by $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}$, respectively. We denote a training dataset by $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ are an input and its label for the $i$-th datum. We denote a utility function by $U$, which takes as input a subset of the training dataset $\mathcal{D}$ and outputs a model’s performance that is trained on that subset. In classification, for instance, a common choice for $U$ is the test classification accuracy of an empirical risk minimizer trained on a subset of $\mathcal{D}$, i.e., $U(S) = \mathbb{E}[\mathbb{I}(y = f_S(X))]$ where $\mathbb{I}(A)$ is an indicator whose value is one if a statement $A$ is true and zero otherwise, $f_S := \text{argmin}_{f \in \mathcal{F}} \sum_{j \in S} \mathbb{I}(y_j \neq f(x_j))$ for some class of models $\mathcal{F} = \{f : \mathcal{X} \to \mathcal{Y}\}$, and the expectation $\mathbb{E}$ is taken with respect to a data distribution or is often approximated by a finite holdout validation dataset. When $S = \emptyset$ is the empty set, $U(S)$ is set to be the performance of the best constant model by convention. A utility function depends on the choice of learning algorithms and a class $\mathcal{F}$, but we suppress its dependency as our main focus is on comparing the functional form of data values. For a set $S$, we denote its power set by $2^S$ and its cardinality by $|S|$. We set $[j] := \{1, \ldots, j\}$ for $j \in \mathbb{N}$.

A standard approach for quantifying data values is to use the marginal contribution, which measures the average change in a utility function when a particular datum is removed from a subset of the entire training dataset $\mathcal{D}$.

Definition 2.1 (Marginal contribution). For a utility function $U : 2^\mathcal{D} \to \mathbb{R}$ and $j \in [n]$, the marginal contribution of $z \in \mathcal{D}$ with respect to $j$ samples is defined as follows.

$$\Delta_j(z, U) := \frac{1}{(n-1)} \sum_{S \in \mathcal{D}^j \setminus z} U(S \cup \{z\}) - U(S),$$

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

Many data valuation methods can be expressed as a function of the marginal contribution. The LOO method is $\Delta_{\text{LOO}}(z, U)$, measuring the changes when one particular datum $z$ is removed from the entire dataset $\mathcal{D}$. LOO includes the Cook’s distance and the approximate empirical influence function (Cook & Weisberg, 1980; Koh & Liang, 2017). Another example is Data Shapley (Ghorbani & Zou, 2019), which is expressed as a simple average of marginal contributions $\psi_{\text{Shap}}(z, U) := n^{-1} \sum_{j=1}^n \Delta_j(z, U)$. As its extension, Beta Shapley proposed by Kwon & Zou (2022a) is expressed as a weighted mean of marginal contributions.

$$\psi_{\text{Beta}}(z, U, \beta) := \sum_{j=1}^n \beta_j \Delta_j(z, U), \quad (1)$$

where $\beta = (\beta_1, \ldots, \beta_n)$ is a predefined weight vector such that $\sum_{j=1}^n \beta_j = 1$ and $\beta_j \geq 0$ for all $j \in [n]$. A functional form of Equation (1) is also known as semivalues in cooperative game theory.
Many methods have been proposed to reduce computational costs. Wu et al. (2022) proposed a stratified sampling to optimize the number of utility evaluations, and Jia et al. (2019a) and Kwon et al. (2021) derived a closed-form expression of the Shapley value. However, these methods still have difficulties in scaling to large datasets or require unusual assumptions on the utility function. For instance, Jia et al. (2019a) used a utility function that does not take into account the majority voting in classification settings: for \( S \in \mathcal{D} \), \( U(S) = k^{-1} \sum (x_i, y_i) \in \mathcal{N}(S) \mathbb{1}(\tilde{y} = y_i) \) where \( \mathcal{N}(S) \) is a set of \( \min(k, |S|) \) nearest neighbors of \( \tilde{x} \), and \((\tilde{x}, \tilde{y}) \in \mathcal{X} \times \mathcal{Y} \) is a test datum. Kwon et al. (2021) considered a commonly used utility function (e.g., the negative Euclidean distance), but it is limited to linear regression models, which may not be the most favorable in real-world data analysis.

Recently, Lin et al. (2022) proposed an efficient algorithm to estimate a class of data values called the average marginal effect (AME) given as follows.

\[
\psi_{\text{AME}}(z, U) := \mathbb{E}_S[U(S \cup \{z\}) - U(S)],
\]

where the expectation \( \mathbb{E}_S \) is taken over a random set \( S \) with a user-defined distribution defined on the discrete space \( \cup_{j=1}^n D_j \). They showed that AME can include the Shapley value and semivalues as a special case, and it can be approximated by the linear coefficient of a LASSO model. Specifically, AME is estimated by a minimizer of the following objective function.

\[
\arg\min_{\gamma \in \mathbb{R}^n} \frac{1}{|S|} \sum_{S \in \mathcal{S}} \left( U(S) - g(1S)^T \gamma \right)^2 + \lambda \sum_{i=1}^n |\gamma_i|,
\]

where \( \lambda > 0 \) is a regularization parameter, \( g : \{0, 1\}^n \rightarrow \mathbb{R}^n \) is a predefined transformation function, \( \mathcal{S} = \{ S : S \subseteq D \} \) is a set of subsets of \( D \) and \( 1S \in \{0, 1\}^n \) is \( n \)-dimensional vector whose element is one if its index is an element of \( S \), zero otherwise. Their algorithm is shown to have better computational efficiency in the semivalue estimation than existing methods. However, it requires some sparsity assumption that is difficult to verify, and also, it needs training a LASSO model.

Ilyas et al. (2022) proposed a similar idea called datamodels that use a LASSO model to predict a test data point’s prediction, i.e., the utility \( U(S) \) is evaluated at a particular test data point. Due to its dependency on a particular datum, it is not suitable for capturing the influence on the model performance. Moreover, it needs computational costs for training a LASSO model similar to Lin et al. (2022).

Besides the computational issue, marginal contribution-based methods have another critical issue with theoretical interpretation. Motivated by cooperative game theory, they provide mathematical justifications that are seemingly solid. However, the fair division axioms used in the Shapley value have not been statistically examined, and it raises a fundamental question about the appropriateness of these axioms in machine learning problems (Kumar et al., 2020; Sim et al., 2022; Rozemberczki et al., 2022).

In the following section, we propose a novel data valuation framework for a bagging model that can address aforementioned issues. Our method is computationally efficient by recycling trained weak learners and does not rely on the fair division axioms that can be less relevant to machine learning applications.

3. Data-OOB: Out-Of-Bag Estimate as Data Value

Suppose we have a trained bagging model that consists of \( B \) weak learner models. For \( b \in [B] \), we denote the \( b \)-th weak learner by \( \hat{f}_b : \mathcal{X} \rightarrow \mathcal{Y} \), which is trained on the \( b \)-th bootstrap dataset. It can be expressed as a minimizer of a weighted risk as follows.

\[
\hat{f}_b := \arg\min_{f \in F} \frac{1}{n} \sum_{i=1}^n w_{bi} \ell(y_i, f(x_i)),
\]

where \( \ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R} \) is a loss function and \( w_{bi} \in \mathbb{Z} \) is the number of times the \( j \)-th datum \((x_j, y_j)\) is selected in the \( b \)-th bootstrap dataset. We set \( w_b := (w_{b1}, \ldots, w_{bn}) \) for all \( b \in [B] \). For \( i \in [n] \) and \( \Theta_B := \{(w_{bi}, \hat{f}_b)\}_{b=1}^B \), we propose to use the following quantity as data values.

\[
\psi((x_i, y_i), \Theta_B) := \frac{\sum_{b=1}^B \mathbb{1}(w_{bi} = 0) \mathbb{1}(y_i = \hat{f}_b(x_i))}{\sum_{b=1}^B \mathbb{1}(w_{bi} = 0)}. \tag{2}
\]

where \( T : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R} \) is a score function that represents the goodness of a weak learner \( \hat{f}_b \) at the \( i \)-th datum \((x_i, y_i)\). For instance, we can use the correctness function \( T(y_i, \hat{f}_b(x_i)) = \mathbb{1}(y_i = \hat{f}_b(x_i)) \) in classification settings and the negative Euclidean distance \( T(y_i, \hat{f}_b(x_i)) = -(y_i - \hat{f}_b(x_i))^2 \) in regression settings.

Our proposed data value in Equation (2) measures the average score when the datum \((x_i, y_i)\) is not selected in the bootstrap dataset. Accordingly, it can be interpreted as a partition of the OOB estimate, which is originally introduced to estimate the prediction error (Efron, 1992; Efron & Tibshirani, 1997). Specifically, the OOB estimate is given as

\[
\frac{1}{n} \sum_{i=1}^n \sum_{b=1}^B \mathbb{1}(w_{bi} = 0) T(y_i, \hat{f}_b(x_i)) - \frac{\sum_{b=1}^B \mathbb{1}(w_{bi} = 0)}{n},
\]

and it is equal to the simple average of the proposed data values \( \frac{1}{n} \sum_{i=1}^n \psi((x_i, y_i), \Theta_B) \).

Motivated by this relationship, we call our data valuation method Data-OOB.
Data-OOB has several advantages in computational efficiency. In contrast to existing marginal contribution-based data valuation methods, Data-OOB can leverage trained weak learners \( f_b \). In other words, Data-OOB does not require training multiple models for the utility evaluation and is readily obtained when there is a trained bagging model. Moreover, it has sample efficiency because it does not use additional validation data points for the utility evaluation that can greatly affect the quality of data values.

**Theoretical interpretation** We rigorously examine the statistical implications of our proposed method. We show that Data-OOB identifies the same set of important data points as Jaeckel’s infinitesimal jackknife influence function (Jaeckel, 1972; Mallows, 1975). We denote the empirical influence

\[
\psi(x, y, \Theta_B) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\hat{P}_{\epsilon, i}}(w_{bj} = 0)T(y_j, \hat{f}_b(x_j))
\]

where \( q_b = \frac{1}{n} \sum_{j=1}^{n} 1(w_{bj} = 0)T(y_j, \hat{f}_b(x_j)) \) is the normalized OOB score for the \( b \)-th bootstrap dataset and \( \bar{q} := B^{-1} \sum_{b=1}^{B} q_b \).

**Proposition 3.1** (Order consistency between Data-OOB and the infinitesimal influence function). For \( i \neq j \in [n] \), if \( \psi_{\Omega}(x_i, y_i) > \psi_{\Omega}(x_j, y_j) + 4\sqrt{2V_B^{1/2}} \), then \( \psi((x_i, y_i), \Theta_B) > \psi((x_j, y_j), \Theta_B) \).

A proof is given in Appendix C. Proposition 3.1 provides new statistical insights when two data points are compared: The proposed method and the infinitesimal jackknife influence function have the order if one data point has a large enough influence function value than the other. Here, \( V_B \) is the variance of OOB scores \( q_b \) across different bootstrap datasets, and it is expected to be very small (e.g., \( O(n^{-1}) \)) when \( n \) and \( B \) are large enough (Efron, 1979). In short, when there is a large enough gap between two influence function values, the proposed method will have the same ordering. Given that many applications of data valuation mainly focus on the order of data points, this theoretical result highlights the potential efficacy of the method in downstream tasks.

**4. Experiments**

In this section, we systematically investigate the practical effectiveness of the proposed data valuation method Data-OOB through three sets of experiments, which are frequently used in previous studies: time comparison, mislabeled data detection, and point removal experiment. We demonstrate that our method is computationally efficient and highly effective in identifying mislabeled data. Furthermore, compared to state-of-the-art data valuation methods, Data-OOB better determines which data points are beneficial or detrimental for model training.

**Experimental settings** We use 12 classification datasets that are publicly available in OpenML (Feurer et al., 2021) or the Python package ‘scikit-learn’ (Pedregosa et al., 2011), and have at least 15000 samples. Also, we note that many of these datasets were used in previous data valuation papers (Ghorbani & Zou, 2019; Kwon & Zou, 2022a). We compare Data-OOB with the following four data valuation methods: KNN Shapley (Jia et al., 2019a), Data Shapley (Ghorbani & Zou, 2019), Beta Shapley (Kwon & Zou, 2022a), and AME (Lin et al., 2022). We set the training sample size to \( n \in \{1000, 10000\} \), but Data Shapley and Beta Shapley are computed only when \( n = 1000 \) due to their low computational efficiency. All methods except for Data-OOB require additional validation data to evaluate the utility function \( U \). We set the validation sample size to 10% of the training sample size \( n \). As for Data-OOB, we use a random forest model with \( B = 800 \) decision trees. To make our comparison fair, we use the same number or a
greater number of utility evaluations for Data Shapley, Beta Shapley, and AME compared to Data-OOB. Implementation details are provided in Appendix A.

### 4.1. Elapsed Time Comparison

We first assess the computational efficiency of Data-OOB using a synthetic binary classification dataset. For $d \in \{10, 100\}$, an input $X \in \mathbb{R}^d$ is randomly generated from a multivariate Gaussian distribution with zero mean and an identity covariance matrix, and an output $Y \in \{0, 1\}$ is generated from a Bernoulli distribution with a success probability $p(X)$. Here, $p(X) := 1/(1 + \exp(-X^T \eta))$ and each element of $\eta \in \mathbb{R}^d$ is generated from a standard Gaussian distribution. We only generate $\eta$ once, and the same $\eta$ is used to generate different data points. A set of sample sizes $n$ is $\{10^3, 2.5 \times 10^4, 5 \times 10^4, 10^5, 2.5 \times 10^5, 5 \times 10^5\}$. We measure the elapsed time with a single Intel Xeon E5-2640v4 CPU processor. For a fair comparison, the elapsed time for Data-OOB includes the training time for the random forest.

As Figure 2 shows, Data-OOB achieves better computational efficiency than existing methods KNN Shapley and AME in various $n$ and $d$. Specifically, Data-OOB is 54 times faster than KNN Shapley when $(n, d) = (10^5, 10)$. Interestingly, we find KNN Shapley is slow despite having the closed-form expression because it needs to sort $n$ data points for each validation data point. When $(n, d) = (5 \times 10^5, 100)$ and the validation sample size is $10^4$, KNN Shapley exceeds 24 hours. For this reason, we exclude this setting from Figure 2. KNN Shapley can be more efficient if the validation size is smaller, but it would cost the quality of data values. In comparison with AME, Data-OOB does not require training LASSO models, achieving better computational efficiency.

As for the algorithmic complexity, when a random forest is used, the computational complexity of Data-OOB will be $O(Bdn \log(n))$ where $B$ is the number of trees, $d$ is the number of features and $n$ is the number of data points in the training dataset. This is because the computational cost of Data-OOB is mainly from training a random forest model, and its computational complexity is $O(Bdn \log(n))$ (Hassine et al., 2019). Meanwhile, the computational complexity of KNN Shapley will be $O(n^2 \log(n))$ when the number of data points in the validation dataset is $O(n)$ (e.g. 10% of $n$). These results support why the elapsed time for Data-OOB increases linearly and that of the KNN-Shapley increases polynomially in Figure 2. In addition, it shows that ours can be beneficial when $n$ is increasing but $B$ and $d$ are fixed.

Our method is highly efficient and it takes less than 2.25 hours when $(n, d) = (10^6, 100)$ on a single CPU processor. The proposed method can be more efficient with the use of trained multiple weak learners. For instance, when $(n, d) = (10^5, 10)$, the computation of Data-OOB takes only 13% of the entire training time for a random forest.

### 4.2. Mislabeled Data Detection

Since mislabeled data often negatively affect the model performance, it is desirable to assign low values to these data points. To see the detection ability of Data-OOB, we conduct mislabeled data detection experiment. We randomly choose 10% of the entire data points and change its label to one of other labels. We first compute data values as if the contaminated dataset is the original dataset, and then we evaluate the precision and the recall of data valuation methods. Note that every method is not provided with an annotation about which data point is mislabeled.

Figure 3 compares the precision-recall curves of different data valuation methods. AME is not displayed because it assigns the exactly zero value for most data points, resulting in meaningless precision and recall values. Data-OOB shows better or comparable performance than existing marginal contribution-based methods in various settings. Additional results using different datasets are provided in Appendix B.1, where Data-OOB consistently shows competitive performance over Data Shapley, Beta Shapley, and KNN Shapley.

We further assess the detection ability of different data valuation methods. Following the mislabeled data detection task in Kwon & Zou (2022a), we apply the K-means algorithm to data values and divide data points into two clusters. (Arthur & Vassilvitskii, 2007). We regard data points in a cluster with a lower mean as the prediction for mislabeled data points. Then, the F1-score is evaluated by comparing the prediction with its actual annotations. Table 1 shows the F1-score of different data valuation methods for the twelve...
Figure 3. Precision-recall curves of different data valuation methods on the four datasets when (top) \( n = 1000 \) and (bottom) \( n = 10000 \). The larger area under the curve is, the better method is. The proposed method shows superior or comparable identification performance in various settings. Additional results using different datasets are provided in Appendix B.1.

Table 1. F1-score of different data valuation methods on the twelve datasets when (left) \( n = 1000 \) and (right) \( n = 10000 \). The average and standard error of the F1-score based on 50 independent experiments are denoted by ‘average±standard error’. Bold numbers denote the best method. In almost all situations, the proposed Data-OOB outperforms other methods in detecting mislabeled data.

4.3. Point removal experiment

Data valuation methods can be useful in identifying a small set of the training dataset that is helpful or harmful to a model’s performance. To see this, we conduct the point removal experiment. Following Ghorbani & Zou (2019), we remove data points from the entire dataset one by one starting from lowest to largest. At each time a datum is removed, we fit a logistic regression model with the remaining data and evaluate the test accuracy of the logistic regression model. We include the random removal as one of baseline methods, and use the same contaminated datasets in Section 4.2. All the test accuracy results are evaluated on the fixed holdout dataset with 3000 data points.

Figure 4 shows the test accuracy curves on the four datasets when \( n \in \{1000, 10000\} \). Overall, Data-OOB achieves significantly better test accuracy than other baseline methods in various settings, showing a steeper increase in the first 20% of data removal. We suggest that this increase in performance is due to the mislabeled data detection performance as mislabeled data points are likely to be removed.
Figure 4. Test accuracy curves as a function of the percentage of data removed. We consider the four datasets when (top) \( n = 1000 \) and (bottom) \( n = 10000 \). We remove data points one by one starting from the lowest to the largest. Higher curves indicate better performance for data valuation. The error bar indicates a 95\% confidence interval based on 50 independent experiments. Additional results using 8 different datasets are provided in Appendix B.2.

To further investigate the effect of data removal on the model performance, we illustrate a distribution of the remaining data points when 20\%, 50\%, or 80\% of data points are removed. We compare Data-OOB with KNN Shapley and AME. We use the ‘MiniBooNE’ dataset with \( n = 1000 \). As its input dimension is 50, we display data points using their first two principal component scores.

Figure 5 shows the four different snapshots, namely 0\%, 20\%, 50\%, and 80\% of data removal. Data-OOB shows an increased test accuracy even after 50\% of data removal by effectively removing unhelpful data points. When 80\% of data are removed, it clearly shows a region of each class, giving an intuitive classifier boundary. The test accuracy for KNN Shapley after 80\% of data removal is not measured as there are no blue class data points in top 20\%. This is anticipated in that KNN Shapley overly focuses on the local neighbors and tends to assign large values if there is a homogeneous local neighbors. AME shows a better test accuracy than KNN Shapley, but it almost randomly removes data points due to sparse data values. As a result, it does not give meaningful insights into class regions. In Appendix B.3, we show additional experiments with the datasets ‘electricity’ and ‘fried’. Data-OOB consistently identifies informative data points, showing a better capability in finding beneficial data points than KNN Shapley and AME.

We emphasize that Data-OOB is insensitive to different choices of the number of weak learners \( B \). We conduct a mislabeled data detection experiment with \( B \in \{400, 800, 3200\} \), showing that the F1-score for...
Data-OOB: Out-of-bag Estimate as a Simple and Efficient Data Value

5. Related Works

**Bagging**  Bootstrap aggregation, which is also known as bagging, is an ensemble technique that trains multiple weak learners where each learner is trained using a bootstrap dataset (Breiman, 1996). One popular and powerful bagging model is the random forest in which multiple numbers of decision trees are trained with a randomly selected set of features (Breiman, 2001; Wager et al., 2014; Athey et al., 2019). While the primary usage of bagging is to improve a model’s performance by decreasing the variance of its predictions, the proposed Data-OOB presents a distinct application of bagging.

**Marginal contribution-based methods in machine learning**  Marginal contribution-based methods have been studied and applied to various machine learning problems, for instance, feature attribution problems (Lundberg & Lee, 2017; Covert et al., 2021; Kwon & Zou, 2022b), model explanation (Stier et al., 2018; Ghorbani & Zou, 2020), collaborative learning (Sim et al., 2020; Xu et al., 2021), and federated learning (Wang, 2019; Wang et al., 2020). The Shapley value is one of the most widely used marginal contribution-based methods, and many alternative approaches have been studied by relaxing some of the underlying fair division axioms (Yan & Procaccia, 2021; Kwon & Zou, 2022a; Wang & Jia, 2022; Rozemberczki et al., 2022). Alternatively, there have been approaches that are independent of marginal contributions. In the data valuation literature, for instance, Yoon et al. (2020) proposed a data value estimator model using reinforcement learning and Ilyas et al. (2022) proposed data-
models that capture the influence via predicting a model’s prediction as Lin et al. (2022).

6. Concluding Remarks

In this paper, we propose Data-OOB that is suitable for any tabular machine learning datasets as it is easy to train a random forest model on such datasets. With comprehensive numerical experiments, we demonstrate that Data-OOB is significantly powerful in identifying helpful and harmful data points for model training. Our method does not require additional validation points and is computationally efficient by reusing trained weak learners. Data-OOB is statistically interpretable, showing it has the order consistency with the infinitesimal jackknife influence function.

While Data-OOB has shown promising results in various classification datasets, there are several limitations and it opens several future avenues of research. One potential extension of Data-OOB is to leverage weak learners in boosting models instead of bagging models. We find that boosting models should be treated differently from a regular bagging model. This is because a weak learner in boosting predicts the residuals obtained from the previous optimization steps, not predicting the ground truth labels. In other words, a weak learner in boosting is sequentially dependent on other weak learners, making a direct application of Data-OOB challenging in downstream machine learning tasks. We believe computing data values with a trained boosting model could be very influential as boosting often performs better than a random forest in practice.

One potential caveat is that Data-OOB can assign erroneously high values for detrimental data points if there are many duplicates. This is because when there are multiple duplicate data, the OOB estimate becomes similar to the training accuracy, not the test accuracy. We believe a simple removal of duplicates can address this issue, but we encourage the community to develop a more principled method for duplicate data.

Acknowledgements

The authors would like to thank all anonymous reviewers for their helpful comments. We also would like to thank Mert Yuksekgonul and Young-geun Kim for their constructive feedback.

References


Data-OOB: Out-of-bag Estimate as a Simple and Efficient Data Value


A. Implementation Details

In this section, we provide implementation details. Our Python-based implementation codes are publicly available at https://github.com/ykwon0407/dataoob.

Datasets We use 12 classification datasets in Section 4. The ‘covertype’ dataset is downloaded via the Python package ‘scikit-learn’ (Pedregosa et al., 2011), and every other dataset is downloaded from ‘OpenML’ (Feurer et al., 2021). Table 2 shows a summary of classification datasets.

We apply a standard normalization procedure. Each feature is normalized to have zero mean and one standard deviation. After this preprocessing, we split it into the three datasets, namely, a training dataset, a validation dataset, and a test dataset. We evaluate the value of data in the training dataset and use the validation dataset to evaluate the utility function. Note that the proposed method does not use this validation dataset, and it essentially uses a smaller dataset. The test dataset is used for point removal experiments only when evaluating the test accuracy. The training dataset size \( n \) is either 1000 or 10000, and the validation size is fixed to 10% of the training sample size. The test dataset size is fixed to 3000.

<table>
<thead>
<tr>
<th>Name</th>
<th>Sample size</th>
<th>Input dimension</th>
<th>Number of Classes</th>
<th>OpenML ID</th>
<th>Minor class proportion</th>
</tr>
</thead>
<tbody>
<tr>
<td>law-school-admission-binary</td>
<td>20800</td>
<td>6</td>
<td>2</td>
<td>43890</td>
<td>0.321</td>
</tr>
<tr>
<td>electricity</td>
<td>38474</td>
<td>6</td>
<td>2</td>
<td>44080</td>
<td>0.5</td>
</tr>
<tr>
<td>fried</td>
<td>40768</td>
<td>10</td>
<td>2</td>
<td>901</td>
<td>0.498</td>
</tr>
<tr>
<td>2dplanes</td>
<td>40768</td>
<td>10</td>
<td>2</td>
<td>727</td>
<td>0.499</td>
</tr>
<tr>
<td>default-of-credit-card-clients</td>
<td>30000</td>
<td>23</td>
<td>2</td>
<td>42477</td>
<td>0.221</td>
</tr>
<tr>
<td>pol</td>
<td>15000</td>
<td>48</td>
<td>2</td>
<td>722</td>
<td>0.336</td>
</tr>
<tr>
<td>MiniBooNE</td>
<td>72998</td>
<td>50</td>
<td>2</td>
<td>43974</td>
<td>0.5</td>
</tr>
<tr>
<td>jannis</td>
<td>57580</td>
<td>54</td>
<td>2</td>
<td>43977</td>
<td>0.5</td>
</tr>
<tr>
<td>nomao</td>
<td>34465</td>
<td>89</td>
<td>2</td>
<td>1486</td>
<td>0.285</td>
</tr>
<tr>
<td>vehicle_sensIT</td>
<td>98528</td>
<td>100</td>
<td>2</td>
<td>357</td>
<td>0.5</td>
</tr>
<tr>
<td>webdata_wXa</td>
<td>36974</td>
<td>123</td>
<td>2</td>
<td>350</td>
<td>0.240</td>
</tr>
<tr>
<td>covertype</td>
<td>581012</td>
<td>54</td>
<td>7</td>
<td>Scikit-learn</td>
<td>0.004</td>
</tr>
</tbody>
</table>

Table 2. A summary of 12 classification datasets used in our experiments. We provide the dataset-specific OpenML ID in the column ‘OpenML ID’.

Hyperparameters for data valuation methods

- For **KNN Shapley**, the only hyperparameter is the number of nearest neighbors. Since there is no optimal fixed number for hyperparameter, we set it to be 10% of the sample size \( n \) motivated by Jia et al. (2019a).

- For **Data Shapley** and **Beta Shapley**, following Kwon & Zou (2022a), we use a Monte Carlo-based algorithm. Specifically, it consists of two stages. In the first stage, we estimate every marginal contribution and in the second stage, we compute the Shapley value or semivalue. The second stage is straightforward, so here we explain the first stage. We first randomly draw a cardinality \( j \) from a discrete uniform distribution on \([n]\). Then, we uniformly draw a subset \( S \) from a set of subsets with the cardinality \( j \). After that, we evaluate utility \( U(S) \). We construct 10 independent Monte Carlo chains for this procedure and compute the Gelman-Rubin statistics to check the convergence of a simple average of marginal contributions. For each data point, we can compute the Gelman-Rubin statistics, and we consider the maximum of these statistics across samples. We stop the algorithm if the maximum value is less than the threshold value 1.05, which is less than a usual threshold 1.1 (Gelman et al., 1995). We use a decision tree model for the utility evaluation for a fair comparison with the proposed method.

- For **AME**, we set the number of utility evaluations to be 800. Following Lin et al. (2022), we consider the same uniform distribution for constructing subsets. That is, for each \( p \in \{0.2, 0.4, 0.6, 0.8\} \), we randomly generate 200 subsets such that the probability that a datum is included in the subset is \( p \). The number of utility evaluation is chosen to be same with the number of weak learners \( B \) of the proposed algorithm for a fair comparison. Like **Data Shapley** and **Beta Shapley**, we use a decision tree model for the utility evaluation. As for the Lasso model, we optimize the regularization parameter using ‘LassoCV’ in ‘scikit-learn’ with its default parameter values.
• The proposed method fits a random forest model with $B = 800$ decision trees using ‘scikit-learn’. In classification settings, we use $T(y_1, y_2) = \mathbb{1}(y_1 = y_2)$.

B. Additional Experimental Results

In this section, we present additional experimental results using the eight classification datasets.

B.1. Mislabeled Data Detection

We provide the precision-recall curves for the eight datasets when $n \in \{1000, 10000\}$. Except for datasets, every experimental setting is exactly the same as the one used in Figure 3. Figures 6 and 7 show that the proposed method has superior or at least comparable identification performance in various settings.

**Figure 6.** Precision-recall curves of different data valuation methods on the eight datasets when $n = 1000$. The larger are under the curve is, the better method is.

**Figure 7.** Precision-recall curves of different data valuation methods on the eight datasets when $n = 10000$. The larger are under the curve is, the better method is.
B.2. Point Removal Experiment

Figures 8 and 9 show that point removal experiment results for the eight datasets when \( n \in \{1000, 10000\} \). Like Section 4.3, we remove data points one by one starting from the lowest to the largest. Higher curves indicate better performance for data valuation. The error bar indicates a 95% confidence interval based on 50 independent experiments. The proposed method shows a better or comparable performance than existing state-of-the-art data valuation methods in most settings.

**Figure 8.** Test accuracy curves as a function of the percentage of data removed. We consider the four datasets when \( n = 1000 \).

**Figure 9.** Test accuracy curves as a function of the percentage of data removed. We consider the four datasets when \( n = 10000 \).
B.3. Distribution after data removal

In this section, we present additional qualitative examples using the dataset ‘fried’ in Figure 10 and ‘electricity’ in Figure 11. As in Figure 5, the additional results demonstrate that Data-OOB is more effective in finding beneficial data points and shows competitive test accuracy even after 80% of data removal. In contrast to AME that almost randomly selects data, Data-OOB provides a set of data points that are more sensible in that the remaining data construct clear classification regions.

Figure 10. Distribution after data removal for (top) Data-OOB, (middle) KNN Shapley, and (bottom) AME. We use the ‘fried’ dataset. The details are given in Figure 5.
Figure 11. Distribution after data removal for (top) Data-OOB, (middle) KNN Shapley, and (bottom) AME. We use the ‘electricity’ dataset. The details are given in Figure 5.
B.4. Robustness of the number of weak learners

In this section, we show that our experimental results are robust against the number of weak learners $B$. Our results in this section show that the proposed method can achieve a similar performance with a smaller number $B$.

Mislabeled data detection We compare the mislabeled data detection ability across different number of weak learners $B \in \{400, 800, 3200\}$. Table 3 shows the F1-score of Data-OOB with $B \in \{400, 800, 3200\}$. In almost every dataset, the F1-score is likely to be bigger as $B$ increases, but the difference between the best and the worst is not significant.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$B = 400$</th>
<th>$B = 800$ (used in Section 4)</th>
<th>$B = 3200$</th>
</tr>
</thead>
<tbody>
<tr>
<td>pol</td>
<td>$0.8806 \pm 0.0009$</td>
<td>$0.8822 \pm 0.0009$</td>
<td>$0.8859 \pm 0.0012$</td>
</tr>
<tr>
<td>jannis</td>
<td>$0.3351 \pm 0.0009$</td>
<td>$0.3358 \pm 0.0007$</td>
<td>$0.3377 \pm 0.0009$</td>
</tr>
<tr>
<td>lawsschool</td>
<td>$0.9573 \pm 0.0006$</td>
<td>$0.9595 \pm 0.0006$</td>
<td>$0.9598 \pm 0.0005$</td>
</tr>
<tr>
<td>fried</td>
<td>$0.5398 \pm 0.0014$</td>
<td>$0.5413 \pm 0.0013$</td>
<td>$0.5421 \pm 0.0014$</td>
</tr>
<tr>
<td>electricity</td>
<td>$0.4348 \pm 0.0011$</td>
<td>$0.4354 \pm 0.0012$</td>
<td>$0.4375 \pm 0.0013$</td>
</tr>
<tr>
<td>2planes</td>
<td>$0.6162 \pm 0.0013$</td>
<td>$0.6180 \pm 0.0013$</td>
<td>$0.6188 \pm 0.0010$</td>
</tr>
<tr>
<td>creditcard</td>
<td>$0.4391 \pm 0.0010$</td>
<td>$0.4420 \pm 0.0010$</td>
<td>$0.4431 \pm 0.0010$</td>
</tr>
<tr>
<td>coverty</td>
<td>$0.4720 \pm 0.0011$</td>
<td>$0.4790 \pm 0.0012$</td>
<td>$0.4781 \pm 0.0010$</td>
</tr>
<tr>
<td>nomao</td>
<td>$0.7546 \pm 0.0014$</td>
<td>$0.7564 \pm 0.0014$</td>
<td>$0.7578 \pm 0.0012$</td>
</tr>
<tr>
<td>webdata_wXa</td>
<td>$0.4151 \pm 0.0009$</td>
<td>$0.4160 \pm 0.0010$</td>
<td>$0.4170 \pm 0.0010$</td>
</tr>
</tbody>
</table>

Table 3. F1-score of Data-OOB when the number of weak learners $B \in \{400, 800, 3200\}$. The case $B = 800$ is exactly the same with the one presented in the manuscript. The difference between the best and the worst is not significant, showing the robustness of the choice of $B$.

C. Proof

We provide a proof for Proposition 3.1.

Proof. By Lemma 1 of Efron & Tibshirani (1995), the $\psi_{11}(x_i, y_i)$ is given as

$$\psi_{11}(x_i, y_i) = \left( 2 + \frac{1}{n-1} \right) \frac{\psi((x_i, y_i), \Theta_B) - h(\hat{P})}{n} + \left( 1 - \frac{1}{n} \right)^{-n} \frac{1}{B} \sum_{b=1}^{B} (w_{bi} - 1)q_b,$$

where $q_b = \frac{1}{n} \sum_{j=1}^{n} 1(w_{bj} = 0)T(y_j, \hat{f}_b(x_j))$ for $b \in [B]$. Therefore, for $i \neq j$, we have

$$\psi_{11}(x_i, y_i) - \psi_{11}(x_j, y_j) = \left( 2 + \frac{1}{n-1} \right) \psi((x_i, y_i), \Theta_B) - \psi((x_j, y_j), \Theta_B) + \left( 1 - \frac{1}{n} \right)^{-n} \frac{1}{B} \sum_{b=1}^{B} (w_{bi} - w_{bj})q_b. \quad (3)$$

Let $\bar{q} = B^{-1} \sum_{b=1}^{B} q_b$. Then, $\frac{1}{B} \sum_{b=1}^{B} (w_{bi} - w_{bj})q_b = \frac{1}{B} \sum_{b=1}^{B} (w_{bi} - w_{bj})(q_b - \bar{q})$ and

$$\left| \frac{1}{B} \sum_{b=1}^{B} (w_{bi} - w_{bj})(q_b - \bar{q}) \right| \leq \left( \frac{1}{B} \sum_{b=1}^{B} (w_{bi} - w_{bj})^2 \right)^{1/2} \left( \frac{1}{B} \sum_{b=1}^{B} (q_b - \bar{q})^2 \right)^{1/2}$$

$$= \left( 2 - \frac{4}{n} \right)^{1/2} \{\text{Var}_s(q_b)\}^{1/2}.$$

The first inequality is from Cauchy-Schwarz inequality. The last equation is because $(w_{bi}, w_{bj})$ follows the multinomial distribution with the parameters $(n, (n-1, n-1))$. By the assumption on the influence function, we have

$$\psi_{11}(x_i, y_i) > \psi_{11}(x_j, y_j) + 4\sqrt{2}\{\text{Var}_s(q_b)\}^{1/2}$$

$$> \psi_{11}(x_j, y_j) + \left( 1 - \frac{1}{n} \right)^{-n} \left( 2 - \frac{4}{n} \right)^{1/2} \{\text{Var}_s(q_b)\}^{1/2}.$$

(4)
The second inequality is because \((1 - \frac{1}{n})^{-n} \leq 4\) and \((2 - \frac{4}{n})^{1/2} < \sqrt{2}\). Note that \((1 - \frac{1}{n})^{-n}\) is decreasing with respect to \(n \geq 2\). Therefore, Equations (3) and (4) imply that

\[
\psi((x_i, y_i), \Theta_B) > \psi((x_j, y_j), \Theta_B).
\]