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

Yongchan Kwon [1] **James Zou** [2 3]

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. 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. 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. We demonstrate that the proposed method significantly outperforms existing state-of-the-art data valuation methods in identifying mislabeled data, 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. The main goal of data valuation 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. Data Shapley (Ghorbani & Zou, 2019), Distributional Shapley (Ghorbani et al., 2020), and CS-Shapley (Schoch et al., 2022) belong to this category. These methods have shown promising results in many downstream tasks by leveraging every possible marginal contribution (Ghorbani & Zou, 2019; Jia et al., 2019). 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.

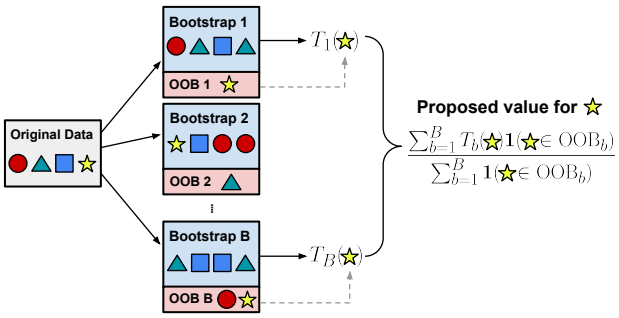

Figure 1. Illustration of the proposed data valuation method. The OOB stands for the out-of-bag set. For each bootstrap sampling procedure, we evaluate an estimate $T_b(\star)$ if the datum $\star$ is in the OOB set. Here, $T_b(\star)$ is a score of the model trained with the b-th bootstrap dataset evaluated at $\star$. The proposed data value summarizes scores $T_b(\star)$ from the B bootstrap datasets. Details are provided in Section 2.

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. Our comprehensive experiments demonstrate that the proposed method significantly better identifies mislabeled data than existing state-of-the-art data valuation methods.

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

Suppose we have a trained bagging model that consists of B weak learner models. For $b \in [B] := \{1, \ldots, B\}$, we denote the b-th weak learner by $\hat{f}_b : \mathcal{X} \to \mathcal{Y}$, which is trained on the b-th bootstrap dataset, *i.e.*, $\hat{f}_b := \mathrm{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum_{j=1}^{n} w_{bj} \ell(y_j, f(x_j))$, where $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function and $w_{bj} \in \mathbb{Z}$ 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

*Equal contribution [1]Columbia University [2]Stanford University [3]Amazon AWS. Correspondence to: James Zou <jamesz@stanford.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

Dataset	$n = 1000$					$n = 10000$		
	KNN Shapley	Data Shapley	Beta Shapley	AME	Data-OOB	KNN Shapley	AME	Data-OOB
pol	0.28 ± 0.003	0.50 ± 0.011	0.46 ± 0.010	0.09 ± 0.009	$\mathbf{0.73 \pm 0.004}$	0.28 ± 0.000	0.10 ± 0.012	$\mathbf{0.88 \pm 0.000}$
jannis	0.25 ± 0.004	0.23 ± 0.003	0.24 ± 0.003	0.09 ± 0.012	$\mathbf{0.30 \pm 0.001}$	0.28 ± 0.001	0.06 ± 0.012	$\mathbf{0.33 \pm 0.000}$
lawschool	0.45 ± 0.014	0.94 ± 0.003	0.94 ± 0.003	0.10 ± 0.009	$\mathbf{0.96 \pm 0.002}$	0.39 ± 0.005	0.08 ± 0.012	$\mathbf{0.95 \pm 0.000}$
fried	0.28 ± 0.005	0.32 ± 0.003	0.32 ± 0.004	0.09 ± 0.011	$\mathbf{0.44 \pm 0.004}$	0.35 ± 0.001	0.08 ± 0.012	$\mathbf{0.54 \pm 0.001}$
vehicle_sensIT	0.20 ± 0.004	0.37 ± 0.006	0.39 ± 0.006	0.07 ± 0.011	$\mathbf{0.49 \pm 0.004}$	0.21 ± 0.004	0.09 ± 0.012	$\mathbf{0.52 \pm 0.001}$
electricity	0.26 ± 0.006	0.32 ± 0.004	0.34 ± 0.004	0.08 ± 0.010	$\mathbf{0.35 \pm 0.002}$	0.29 ± 0.001	0.08 ± 0.012	$\mathbf{0.43 \pm 0.001}$
2dplanes	0.30 ± 0.007	0.57 ± 0.006	0.54 ± 0.006	0.10 ± 0.009	$\mathbf{0.58 \pm 0.004}$	0.42 ± 0.004	0.10 ± 0.012	$\mathbf{0.61 \pm 0.001}$

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.

$\Theta_B := \{(w_b, \hat{f}_b)\}_{b=1}^B$, we propose Data-OOB as follows.

$$\psi((x_i, y_i), \Theta_B) := \frac{\sum_{b=1}^B \mathbb{1}(w_{bi} = 0) T(y_i, \hat{f}_b(x_i))}{\sum_{b=1}^B \mathbb{1}(w_{bi} = 0)}. \quad (1)$$

where $T : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ 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.

3. Experiments

In this section, we systematically investigate the practical effectiveness of the proposed data valuation method Data-OOB in the mislabeled data detection. In Appendix C, we provide additional experimental results, demonstrating that our method is computationally efficient and highly effective in identifying mislabeled data.

Experimental settings We use seven classification datasets that are publicly available in OpenML (Feurer et al., 2021) 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., 2019), 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. 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.

3.1. 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.

We 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 classification datasets. Overall, Data-OOB significantly outperforms other state-of-the-art methods. In particular, when dataset is 'pol' and $n = 10000$, Data-OOB achieves 3.1 and 8.7 times greater F1-score than KNN Shapley and AME, respectively. As noted by Lin et al. (2022), the F1-score for AME can be improved if the Model-X Knockoff procedure is incorporated (Candes et al., 2018). However, it requires additional training LASSO models with dummy variables, resulting in extra computational costs. We demonstrate that Data-OOB shows strong performance in detecting mislabeled data points without such procedures.

4. 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.

References

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

Athey, S., Tibshirani, J., and Wager, S. Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178, 2019.

Breiman, L. Bagging predictors. *Machine learning*, 24(2): 123–140, 1996.

Breiman, L. Random forests. *Machine learning*, 45(1): 5–32, 2001.

Candes, E., Fan, Y., Janson, L., and Lv, J. Panning for gold:'model-x'knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(3):551–577, 2018.

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

Feurer, M., van Rijn, J. N., Kadra, A., Gijsbers, P., Mallik, N., Ravi, S., Muller, A., Vanschoren, J., and Hutter, F. Openml-python: an extensible python api for openml. *Journal of Machine Learning Research*, 22(100):1–5, 2021. URL http://jmlr.org/papers/v22/19-920.html.

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

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

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

Hassine, K., Erbad, A., and Hamila, R. Important complexity reduction of random forest in multi-classification problem. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 226–231. IEEE, 2019.

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

Kwon, Y. and Zou, J. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 8780–8802. PMLR, 2022a.

Kwon, Y. and Zou, J. WeightedSHAP: analyzing and improving shapley based feature attributions. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022b.

Lin, J., Zhang, A., Lécuyer, M., Li, J., Panda, A., and Sen, S. Measuring the effect of training data on deep learning predictions via randomized experiments. In *International Conference on Machine Learning*, pp. 13468–13504. PMLR, 2022.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pp. 4768–4777, 2017.

Rozemberczki, B., Watson, L., Bayer, P., Yang, H.-T., Kiss, O., Nilsson, S., and Sarkar, R. The shapley value in machine learning. *arXiv preprint arXiv:2202.05594*, 2022.

Schoch, S., Xu, H., and Ji, Y. CS-shapley: Class-wise shapley values for data valuation in classification. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022.

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

Stier, J., Gianini, G., Granitzer, M., and Ziegler, K. Analysing neural network topologies: a game theoretic approach. *Procedia Computer Science*, 126:234–243, 2018.

Wager, S., Hastie, T., and Efron, B. Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *The Journal of Machine Learning Research*, 15(1):1625–1651, 2014.

Wang, G. Interpret federated learning with shapley values. *arXiv preprint arXiv:1905.04519*, 2019.

Wang, T. and Jia, R. Data banzhaf: A data valuation framework with maximal robustness to learning stochasticity. *arXiv preprint arXiv:2205.15466*, 2022.

Wang, T., Rausch, J., Zhang, C., Jia, R., and Song, D. A principled approach to data valuation for federated learning. In *Federated Learning*, pp. 153–167. Springer, 2020.

Yan, T. and Procaccia, A. D. If you like shapley then you'll love the core. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 5751–5759, 2021.

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 seven classification datasets in Section 3. Every 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.

Name	Sample size	Input dimension	Number of Classes	OpenML ID	Minor class proportion
law-school-admission-bianry	20800	6	2	43890	0.321
electricity	38474	6	2	44080	0.5
fried	40768	10	2	901	0.498
2dplanes	40768	10	2	727	0.499
pol	15000	48	2	722	0.336
jannis	57580	54	2	43977	0.5
vehicle_sensIT	98528	100	2	357	0.5

Table 2. A summary of seven 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. (2019).

- 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 semivalues. 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. 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), collaborative learning (Sim et al., 2020), 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).

C. Additional results

C.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 η once, and the same η is used to generate different data points. A set of sample sizes n is $\{10^4, 2.5 \times 10^4, 5 \times 10^4, 10^5, 2.5 \times 10^5, 5 \times 10^5\}$. 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.

C.2. Mislabeled data detection

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.

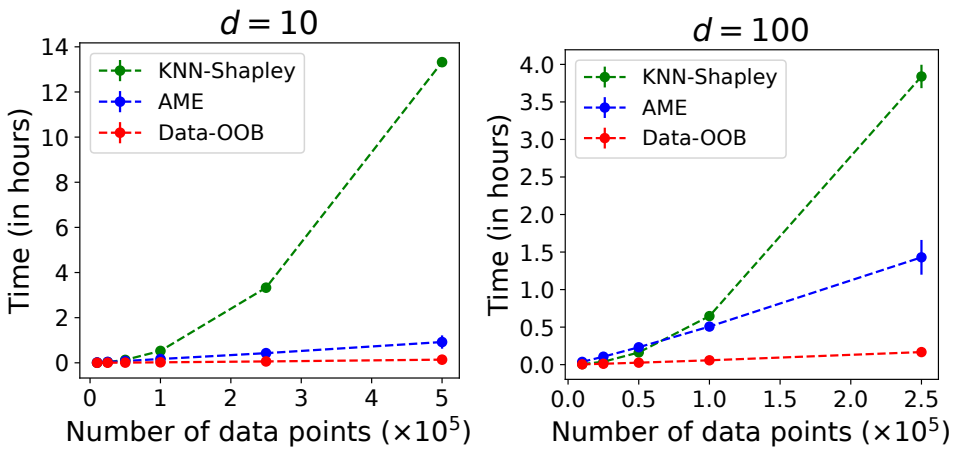

Figure 2. Elapsed time comparison between `KNN Shapley`, `AME`, and `Data-OOB`. We use a synthetic binary classification dataset with (left) $d = 10$ and (right) $d = 100$. We exclude the setting $(n, d) = (5 \times 10^5, 100)$ as `KNN Shapley` exceeds 24 hours. The error bar indicates a 95% confidence interval based on 5 independent experiments. `Data-OOB` is significantly faster than `KNN Shapley` and `AME`. The time for training the random forest is included in the time for `Data-OOB`.

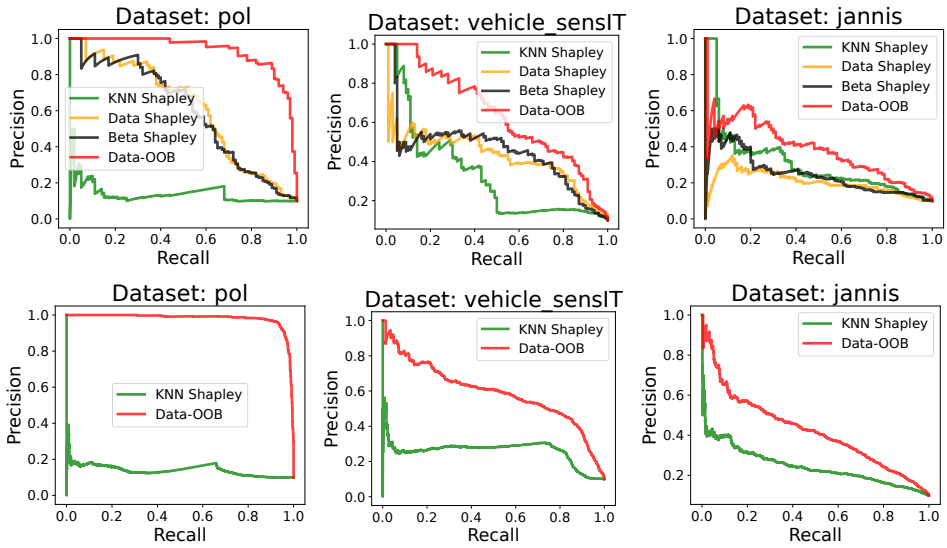

Figure 3. Precision-recall curves of different data valuation methods on the three datasets when (top) $n = 1000$ and (bottom) $n = 10000$. The larger area under the curve is, the better method is. The proposed method shows superior or comparable identification performance in various settings.