EcoVal: An Efficient Data Valuation Framework for Machine Learning

Ayush Tarun*
Ola Krutrim
Bengaluru, India
ayush.tarun@olakrutrim.com

Vikram Chundawat*
RespAI Lab
India
vikram2000b@gmail.com

Murari Mandal †
RespAI Lab
Kalinga Institute of Industrial
Technology
Bhubaneswar, India
murari.mandalfcs@kiit.ac.in

Hong Ming Tan
NUS Business School
National University of Singapore
Singapore
thm@nus.edu.sg

Bowei Chen
Adam Smith Business School
University of Glasgow
Glasgow, United Kingdom
bowei.chen@glasgow.ac.uk

Mohan Kankanhalli
School of Computing
National University of Singapore
Singapore
mohan@comp.nus.edu.sg

ABSTRACT

Quantifying the value of data within a machine learning workflow can play a pivotal role in making more strategic decisions in machine learning initiatives. The existing Shapley value based frameworks for data valuation in machine learning are computationally expensive as they require considerable amount of repeated training of the model to obtain the Shapley value. In this paper, we introduce an efficient data valuation framework EcoVal, to estimate the value of data for machine learning models in a fast and practical manner. Instead of directly working with individual data sample, we determine the value of a cluster of similar data points. This value is further propagated amongst all the member cluster points. We show that the overall value of the data can be determined by estimating the intrinsic and extrinsic value of each data. This is enabled by formulating the performance of a model as a *production function*, a concept which is popularly used to estimate the amount of output based on factors like labor and capital in a traditional free economic market. We provide a formal proof of our valuation technique and elucidate the principles and mechanisms that enable its accelerated performance. We demonstrate the real-world applicability of our method by showcasing its effectiveness for both in-distribution and out-of-sample data. This work addresses one of the core challenges of efficient data valuation at scale in machine learning models. The code is available at https://github.com/respai-lab/ecoval.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Information systems → Information systems applications**.

KEYWORDS

Data valuation, Shapley value, machine learning

1 INTRODUCTION

Data valuation is a pivotal concern in modern machine learning (ML) and data analytics, where the quality and worth of data have profound implications for decision-making, model performance, and data marketplace. Quantifying the worth of data plays an important role in data pricing and regulation compliance [39, 49], removing low-value/noisy data from the training set [28, 48], and incentivizing data sharing by personal data monetization [17, 18, 23, 25]. In a ML framework, the quality of data determines the effectiveness of the final model. Therefore, identifying high and low value data through data valuation would yield significant benefits for a wide range of machine learning applications.

Background: In recent studies, a cooperative game theory concept, Shapley value [41] has been frequently used for data valuation in supervised ML [17, 18, 25]. It offers a desirable property of equitable reward allocation. The data Shapley and its extensions [17, 18, 31, 32] have empirically shown the effectiveness of Shapley value based valuation in a fixed dataset as well as in a particular distribution of data, allowing for out-of-time data valuation. The value of a data point in ML relies on its individual contribution to the model's performance and its relationship with other data points utilized during training. The presence of similar data in the training set can dilute the significance of individual points. To account for these interactions, data Shapley methods evaluate the contribution of each point by determining how its absence affects the overall performance of the model. This process

---

*These authors contributed equally to this work

†Corresponding author

usually involves repeatedly training the model with the selective exclusion of certain instances or subsets, thereby identifying those with the most substantial impact. The impact is measured by the observing the change in the performance score of the ML model. However, this incurs a high computational cost, typically requiring model training runs in the order of $O(n^2)$ in current methodologies, where n is the total number of data points in the dataset.

Motivation: While offering insightful analyses of data point significance and alleviating the issue of poor discrimination of data quality in leave-one-out (LOO) error methods, existing data Shapley based frameworks [17, 18, 32, 50] suffer from a high computational cost. The need for a higher number of repeated training sessions for a model, as required by these methods, leads to inefficiencies in both time and resource utilization. Furthermore, this inefficiency translates to an increased carbon footprint due to the energy requirements of training, thereby exacerbating climate change concerns [33]. The development of scalable algorithms capable of handling extensive datasets is essential for practical use of data valuation in real-world applications.

Our Contribution: We adopt a two-step approach where the valuation is performed at cluster-level first and the value is further divided among the cluster members. The similar data points are represented through a cluster which significantly reduces the total number of data points to deal with during training phase of the valuation process. At cluster level, we can use a simple LOO error for valuation since there is minimal possibility (almost zero) of a similar datum to be found in other clusters. The difficulty however, is to divide the value at each cluster among the cluster members. To address this issue, a novel approach is proposed based on *production functions* in economics. Our two-step approach aims to significantly speed up the valuation process in comparison to the Truncated Monte Carlo (TMC) Shapley.

In this paper, we introduce a a novel framework based on Leave Cluster Out (LCO) and production functions for data valuation in machine learning. The framework is computationally efficient, with theoretical and empirical verifications. The following are the key contributions of our work:

Novel Framework: The intuition behind our framework is that we find a group of similar items and estimate this cluster's marginal contribution. As similar data items are bound to have similar values, we extend this principle to estimate cluster-level value through Leave Cluster Out (LCO).

We introduce a *production function* formulation representing the relation between the data and its utility in a model. We show that this formulation can be used to estimate the value of individual data based on the value of each cluster.

Computational Efficiency: We estimate the intrinsic and extrinsic value of each data point to determine the individual data value. By checking only the marginal contribution of the representative data point of a cluster, we substantially reduce the overhead of creating multiple subsets containing similar data points. Our approach is scalable to large datasets without being limited by the presence of similar data points in the dataset.

Theoretical Proof: We provide a theoretical proof of our data valuation method. We also show that the valuation obtained by our method has negligible error margin when compared with the vanilla Shapley value approximation method.

Empirical Evaluation: We conduct experiments with machine learning models on MNIST, CIFAR10, and CIFAR100. We compare the value rankings of our method with the existing state-of-the-art data valuation approaches data Shapley [18], LOO error, and Distributional Shapley [17] and notice similar or better performance with significant speed-up in data valuation process.

2 RELATED WORK

Literature Review of Shapley Value. Shapley value as formalized in [42] establishes the axiomatic properties and demonstrates its unique ability to fairly allocate gains from cooperation among players. This seminal contribution laid the theoretical groundwork for subsequent developments in cooperative game theory [4, 20, 27, 37]. Shapley value has been extensively used for applications in economics, [21, 36, 40], management science [14, 29], online advertising [46]. In machine learning, it has been utilized for addressing the challenges in pricing ML training data, feature selection, and ML explainability. [11, 53] proposed to employ Shapley value properties for feature selection. [1, 16] use Shapley value in market mechanism to price training data and match buyers to sellers data marketplace design. [34] introduced the SHAP framework, leveraging Shapley values to provide interpretable explanations for machine learning models. Other works have also explored its utility in explaining black-box model predictions [6, 9, 19, 47].

Data Valuation in ML. Recently, the subfield of data valuation in ML models has attracted significant attention and the existing works have shown promising outcomes. Data Shapley [18, 25] proposed to use Shapley value from cooperative game theory for valuation of training data. KNN Shapley [24] improved the efficiency of data Shapley by using a k-nearest neighborhood model. Distributional Shapley [17] expanded the scope of valuation to the underlying data distribution instead of only considering the data points. Beta Shapley [32] relaxes the efficiency axiom in DataShapley and reports utility of data valuation in detecting mislabeled images in the training data. Data Banzhaf [51] propose to estimate the Banzhaf value to improve results on noisy label detection. Several works have attempted to improve the efficiency of the Shapley value computation through approximation techniques [31]. Apart from this, other aspects of data value has been studied in [12, 13, 23, 38, 52]. However, approximation of Shapley value still remains a computationally expensive process, making it difficult to adapt for large models and datasets. The main goal of this work is to develop an alternative efficient data valuation framework to overcome this problem.

Literature Review of Production Functions. [35] offers a detailed outline of the evolution and econometrics of the production function. Aggregate production functions are used in macroeconomics to represent the relationship between total output of an economy (GDP) and the inputs used to produce that output. These inputs typically include capital (K), labor (L), and sometimes other factors like technology or natural resources [5, 43]. The simplest production function used in economics, is the Cobb-Douglas production function introduced by [3]. [30] identifies all multi-factor production functions with given elasticity of output and from given elasticity of production. Production functions have been used in various domains, including health, education, and energy, to name

a few [2, 7, 22]. In our study, we adopt the concept of a production function and adapt it for data valuation. This approach draws inspiration from foundational works and recent advancements in the field. [26] develops a theoretical framework that applies the production function to the economics of data, particularly employing data as an input for training machine learning models. Moreover, [15] highlights the role of data as information aimed at reducing forecast errors, which hints at a production function characterized by bounded returns to data. In our paper, we align with these perspectives and further the discourse by specifically focusing on the application of the production function concept in the valuation of data.

3 PRELIMINARIES

Let an ML model M, intended for a task T, is trained on a dataset B of size m. Let U denote the performance metric and U^T denote the performance obtained on task T. The overall performance U is achieved after training a sufficient number of epochs e. Here the sufficient number of epochs means $|U_{e+i+1} - U_{e+i}| < \gamma$ for all $i \geq 0$, where γ is an arbitrarily small value. It should be noted that γ arises due to the randomness within the learning algorithm and not further training. The value of a data point is denoted by Φ.

Leave-One-Out (LOO) Error. The LOO error computes the value of a datum z based on the increase in performance obtained by adding it to the training set:

$$\Phi_{LOO}(z; U, B) = U(B) - U(B \setminus \{z\}). \tag{1}$$

It struggles in differentiating data quality when similar data samples exist in the dataset. For example, if each sample has a duplicate copy in the dataset, the LOO will return a value 0 for all of the samples. Shapley value overcomes this limitation by checking the marginal distribution over many subsets of the dataset.

Shapley Value. Shapley value [18] measures the value of a data point z as the weighted average of the performance increase when z is added to different subsets of the dataset B:

$$\Phi_s(z; U, B) = \frac{1}{m} \sum_{k=1}^{m} \frac{1}{\binom{m-1}{k-1}} \sum_{S \subseteq B \setminus \{z\}} \Delta(z; U, S), \tag{2}$$

where $|S| = k - 1$ for $k \in N$ and $\Delta(z; U, S) = U(S \cup \{z\}) - U(S)$. Thus, data Shapley value is the weighted average of the marginal contribution $\Delta(z; U, S)$. It satisfies the following Shapley value axioms:

- **Dummy Player**: If $U(S \cup \{z\}) = U(S) + e$ for all $S \subseteq B$ß and some $e \in R$, then $\Phi(z; U, B) = e$.
- **Symmetry**: If $U(S \cup \{z\}) = U(S \cup \{z'\})$ for all $S \subseteq B \setminus \{z, z'\}$, then $\Phi(z; U, B) = \Phi(z'; U, B)$.
- **Linearity**: $\Phi(z; \alpha_1 U_1 + \alpha_2 U_2, B) = \alpha_1 \Phi(z; U_1, B) + \alpha_2 \Phi(z; U_2, B)$ for $\alpha_1, \alpha_2 \in R$.
- **Efficiency**: $\sum_{z \in N} \Phi(z; U, B) = \Phi(U, B)$.

Further details regarding the interpretation of the above axioms in the context of machine learning can be referred to Ghorbani and Zou [18] and Jia et al. [25].

Production Function. In economics, a production function expresses the relationship between the specific quantities and combinations of different inputs a company uses and the amount of output it produces. Commonly used production functions include

Linear, Leontief, Cobb–Douglas [8, 10], CES, and CRESH [45], each varying in their assumptions for the input and the output. The widespread usage of the Cobb-Douglas production function is attributed to its simplicity and adaptability. It assumes homogeneity of inputs and this principle is consistent with many machine learning setups.

Let $P(g)$ denote the production over a set of goods $g = (g_1, g_2, \ldots, g_n)$, the Cobb-Douglas production function is defined as

$$P(g) = A \prod_{i=1}^{n} g_i^{x_i}, \tag{3}$$

where x_i is an elastic parameter for good i, and A is the total factor productivity or the quality factor. If inputs are just labor L and capital K, the production function is then

$$P = AL^x K^y. \tag{4}$$

It should be noted that the Cobb-Douglas production function also supports the diminishing returns in terms of both labor and capital. The *Law of Diminishing Returns* [44] states that as the amount of a single factor of production is incrementally increased, the marginal output of a production process decreases. This property is analogous to how more data points have diminishing effects on a machine learning models performance. We therefore adapt the formulation of production functions in our proposed method to efficiently distribute the value of a cluster among its data members.

4 PROPOSED METHOD

A two-stage approach is proposed for efficient data valuation. First, data points are clustered together based on shared characteristics. Then, a leave cluster out (LCO) technique is applied to estimate the value of each cluster. This cluster value is then distributed among its members to obtain the preliminary individual data valuations. In the following, we delve into the building blocks of the proposed method and discuss its properties compared to the original Shapley methods.

4.1 Leave-Cluster-Out

Cluster analysis is firstly performed on the given data and the marginal contribution of a cluster c can be expressed as

$$V_c = U(B) - U(B \setminus c). \tag{5}$$

The simple LOO error may provide an underestimated view of the true impact of specific data points, especially when similar data points remain in the dataset even after removal. Data Shapley alleviates this issue but suffers from high computational cost. By organizing data points into clusters based on their similarity, we ensure that when an entire cluster is removed, there are no closely-related points to mask the effect of its absence in other clusters. Consequently, this leads to a more precise assessment of the cluster's marginal contribution, effectively approximating its value. Furthermore, this clustering approach significantly reduces the number of model training iterations needed in comparison to Data Shapley since evaluations are conducted at the cluster level instead of for each individual data point. Once we have obtained cluster-level valuations, the subsequent step involves efficiently approximating the values of individual data points within each cluster.

4.2 Value Propagation within a Cluster

Production Function for ML. We adapt the Cobb-Douglas production function to approximate the data value for ML. In this context, we can draw an analogy: the labor L corresponds to the available data points for the model; the learning capacity or the number of parameters in the model represents the capital K; and the final output is the obtained performance on the test set U^T. As both data quantity and model complexity exhibit diminishing returns, the Cobb-Douglas production function can be leveraged to effectively model learning performance. Therefore, we propose to approximate the model's performance after e epochs as

$$U^T(S, N) = Af(S)h^T(N),$$ (6)

where $f(S)$ quantifies the informational utility of the dataset S to the predictive efficacy of the model U, T denotes the task, and $h^T(N)$ represents the effect of the model capacity which is dependent on N, the number of parameters of the model.

Then, for a new point z, the performance change ΔU in the model incurred by the small increase ($\Delta S = \{z\}$) in S can be computed by

$$\begin{aligned}
&\Delta U^T(S, N) \\
=&Af(S + \Delta S)h^T(N) - Af(S)h^T(N) \\
=&A\left[f(S + \Delta S) - f(S)\right]h^T(N) \\
=&A\left[\frac{f(S + \Delta S) - f(S)}{o(z)}\right]h^T(N)o(z).
\end{aligned}$$ (7)

To better understand Eq. (7), let us consider f as a smooth function of x as specified in Eq. (6), i.e., $U^T(x, N) = Af(x)h^T(N)$. Thus, a minor change in x leads to a change in U^T, which can be approximated by $Af'(x)h^T(N)\Delta x$. This allows us to interpret the expression enclosed in square brackets of Eq. (7) as effectively serving as the derivative of f with respect to the set S, especially when considering incremental changes to S.

Also, in Eq. (7), $o(z)$ serves as an indicator of how a single data point enhances the model's overall performance and is a proxy to Δx discussed above. Therefore, the difference $f(S + \Delta S) - f(S)$ captures the marginal impact on the model's performance when dataset S is augmented by a new data point. Analogous to the concept of derivatives in calculus, this difference, when normalized by the contribution $o(z)$ of the individual point, can be interpreted as the "rate-of-change" of f upon the addition of a new data point. This rate is contingent on both the existing dataset S and the new data point being added. That is

$$U(S \cup \{z\}) - U(S) = \alpha^T(z)\beta(z, S),$$ (8)

where

$$\begin{aligned}
\alpha^T(z) =&Ah^T(N)o(z), \\
\beta(z, S) =&\frac{f(S + \Delta S) - f(S)}{o(z)}.
\end{aligned}$$

Substituting the above into Eq. (2) then gives

$$\begin{aligned}
\Phi_s(z; U^T, B) &= \frac{1}{m}\sum_{k=1}^{m}\frac{1}{\binom{m-1}{k-1}}\sum_{\substack{S \subset B\setminus\{z\} \\ |S|=k-1}}\alpha^T(z)\beta(z, S) \\
&= \alpha^T(z)\beta^*(z, B)
\end{aligned}$$ (9)

where

$$\beta^*(z, B) = \frac{1}{m}\sum_{k=1}^{m}\frac{1}{\binom{m-1}{k-1}}\sum_{\substack{S \subset B\setminus\{z\} \\ |S|=k-1}}\beta(z, S)$$ (10)

PROPOSITION 1. **(Production Function Based Valuation for ML).** *Let $\alpha^T(z)$ denote the intrinsic value of a datum z, i.e., $\alpha^T(z)$ is only dependent on the characteristics of z. The interaction of z with rest of the data points in B is captured by $\beta^*(z, B)$. From equitable properties of data valuation in [18], we postulate that for every datum z having an intrinsic value $\alpha^T(z)$, the $\beta^*(z, B)$ acts as a multiplier or extrinsic factor that decreases the value of z if similar data points are present in the dataset. Similarly, it increases the data value if z is a unique datum. Then the data valuation can be performed as below*

$$\Phi(z; U^T, B) = \alpha^T(z)\beta^*(z, B).$$ (11)

To simplify notation, we denote $\alpha^T(z)$ with $\alpha(z)$, and $\Phi(z; U^T, B)$ with $\Phi(z; U, B)$ for the rest of the discussion, since T is invariant.

Fast Data Valuation. Based on the above setup, we propose an efficient data valuation method that also works as an efficient proxy to Distributional Shapley [17] to predict valuation for unseen datapoints in the distribution. The existing Data Shapley adheres to two fundamental axioms [50]: symmetry and efficiency. *Symmetry* states that for points z and z' that contribute similarly to the model's performance should have the same value, i.e. $U(S \cup \{z\}) = U(S \cup \{z'\})$ for all $S \in B \setminus \{z, z'\}$. *Efficiency*, on the other hand, ensures that the aggregate value of all data points aligns with the overall performance achieved after training on the entire dataset.

PROPOSITION 2. **(Fast Data Valuation of Cluster Data Members)** *The symmetry and efficiency properties when applied to a specific cluster implies the data points within a cluster, characterized by similar features, will likely possess similar values and a cluster's value can be accurately represented as the sum of its constituent data points' valuations.*

Let $V_c (= \Phi_c)$ be the value of cluster c, the initial value assigned to any data point z_i within this cluster is:

$$V_i = V_c/n_c,$$ (12)

where n_c is the number of data points in cluster c. Using this cluster-level assignment of initial data value, we estimate the actual data value based on Eq. (11) as

$$V_i^* = \alpha_i\beta_i^*.$$ (13)

Estimating α and β^*. Assuming each cluster contains an equal number of data points, the distribution of similar and dissimilar samples encountered by each datum becomes roughly uniform. This results in a near-constant *extrinsic factor*, $\beta^*(z, B)$, across all data points. Thus, the value of these data points are directly proportional to $\alpha(z_i)$. We use Q_i to denote the value of individual datum to differentiate it from V_i value that is initialized by the cluster value in Eq. (12).

THEOREM 4.1. *For data point z_i, assuming there is no error in β_i^*, its adjusted value $V_i^{\Delta\alpha_i}$ is*

$$V_i^{\Delta\alpha_i} = (\alpha_i + \Delta\alpha_i)\beta_i^* = \Gamma_{\alpha_i}\alpha_i\beta_i^*,$$ (14)

where Γ_{α_i} is an adjustment factor for α_i

$$\Gamma_{\alpha_i} = 1 + \frac{Q_i}{\sum_{z_j \in c} Q_j} V_c. \tag{15}$$

The adjustment factor Γ_{α_i} ensures that value of individual data point is adjusted not only based on its personal contribution (Q_i) but also in proportion to the corresponding cluster's overall impact (V_c). This dual consideration is crucial for accurately reflecting the true value of each data point, balancing internal cluster contributions with the broader context of the cluster's role within the entire dataset. Including V_c in the adjustment ensures that the recalibration of α_i remains sensitive to both intra-cluster dynamics and the comparative significance of each cluster, providing a more nuanced and equitable valuation of data points across a heterogeneous dataset.

COROLLARY 4.1.1. *When all data points in a cluster are exactly the same, the adjustment factor should be equal to 1 so that for each point in c, the value becomes V_i. But the above formulation of Γ_{α_i} yields $1 + 1/n$ when all the points are identical as V_i and V_j will be equal for any i, j. Thus, we normalize Γ_{α_i} as follows*

$$\Gamma_{\alpha_i} = \frac{1}{1 + V_c/n_c} \left(1 + \frac{Q_i}{\sum_{z_j \in c} Q_j} V_c \right). \tag{16}$$

Similar to α_i, we find the adjustment factor for β_i^*, i.e. $\Gamma_{\beta_i^*}$. β_i^* measures the interaction of z_i with all other data points in B. As all data points similar to z_i belong to the same cluster and β_i^* is only affected by the other members in z_i's cluster. We use the distance between z_i and cluster centroid as a measure to it's belongingness to the cluster or similarity to other points in the cluster.

THEOREM 4.2. *For data point z_i, assuming no error in α_i, its adjusted value $V_i^{\Delta\beta_i^*}$ is*

$$V_i^{\Delta\beta_i^*} = \alpha_i(\beta_i^* + \Delta\beta_i^*) = \Gamma_{\beta_i^*} \alpha_i \beta_{1i}, \tag{17}$$

where d_i is the distance of z_i and $\Gamma_{\beta_i^*}$ is the adjustment factor represented as

$$\Gamma_{\beta_i^*} = \frac{1}{1 + V_c/n_c} \left(1 + \frac{d_i}{\sum_{z_j \in c} d_j} V_c \right). \tag{18}$$

Production Function based Data Value Estimation. The final approximation value $\hat{\Phi}_i$ of the data point is

$$\hat{\Phi}_i = (\alpha_i + \Delta\alpha_i)(\beta_i^* + \Delta\beta_i^*). \tag{19}$$

Ignoring $\Delta\alpha_i \Delta\beta_i^*$ then gives

$$\hat{\Phi}_i \approx (\alpha_i + \Delta\alpha_i)\beta_i^* + \alpha(\beta_i^* + \Delta\beta_i^*) - \alpha_i\beta_i^*. \tag{20}$$

By substituting Eq. (13), Eq. (14), Eq. (17), we obtain.

$$
\begin{aligned}
\hat{\Phi}_i &= V_i^{\Delta\alpha_i} + V_i^{\Delta\beta_i^*} - V_i \\
&= V_i(\Gamma_{\alpha_i} + \Gamma_{\beta_i^*} - 1) \\
&= V_i \left[\left(\frac{1}{1 + V_c/n_c} \right) \left(1 + \frac{Q_i}{\sum_{z_j \in c} Q_j} V_c \right) + \right. \\
&\quad \left. \left(\frac{1}{1 + V_c/n_c} \right) \left(1 + \frac{d_i}{\sum_{z_j \in c} d_j} V_c \right) - 1 \right].
\end{aligned}
\tag{21}
$$

For the reader's convenience, Algorithm 1 outlines the implementation steps of the EcoVal efficient data valuation framework.

4.3 Discussion: Comparison with Original Shapley

Let $E(z)$ denote the appropriate embedding from a machine learning model or the pre-final layer of a deep learning model for a data point z. We extend the notion of *Lipschitz Stability* of data Shapley introduced in [17] to estimate the difference in value of different data points. We use proximity of the embeddings $E(z)$ as a proxy to the closeness in the underlying data distribution and formalize the same in the following Theorem.

THEOREM 4.3. *For any z_j, z_k if $||E(z_j) - E(z_k)|| < \epsilon$ then, $|\Phi(z_j) - \Phi(z_k)| \le \epsilon_1$ for very small $\epsilon, \epsilon_1 \ge 0$*

From the principle of clustering, a datum z_j belongs to cluster c if

$$||E(z_j) - E(z_k)|| \le \epsilon, \forall z_k \in c, \tag{22}$$

then for this cluster

$$|\Phi(z_j) - \Phi(z_k)| \le \epsilon_1, \forall z_k, z_j \in c. \tag{23}$$

It means all Shapley values lie within an ϵ_1 interval. Therefore, $\Phi(z_j)$ for any $z_j \in c$ can be expressed as

$$\Phi(z_j) = \bar{\Phi}_c + \delta(z_j), \tag{24}$$

where $\delta(z_j) \le \epsilon/2$ and $\bar{\Phi}_c$ lies somewhere in the ϵ_1 interval.

COROLLARY 4.3.1. *The value V_c of the a cluster by Shapley axioms is defined as*

$$V_c = \sum_{z_j \in c} \Phi(z_j) = n_c \bar{\Phi}_c + \sum \delta(z_j). \tag{25}$$

The detailed proof to 4.3.1 is provided in the Appendix.

THEOREM 4.4. *The difference between the original Shapley value and our proposed approximated data value is*

$$\Delta\Phi_i \approx \frac{n_c \bar{\Phi}_c \delta_R}{\sum_{z_j \in c} Q_j} + \frac{n_c^2 \bar{\Phi}_c Q_i \delta_R}{(\sum_{z_j \in c} Q_j)^2}. \tag{26}$$

Due to the intrinsic limitations on the magnitudes of average Shapley value within a cluster $\bar{\Phi}_c$ and individual point contribution Q_i, both values inherently remain within a bounded range. As cluster size n_c increases, the predicted aggregate value $\sum_{z_j \in c} Q_j$ proportionately grows, naturally restricting the potential expansion of $\frac{n_c}{\sum_{z_j \in c} Q_j}$. Additionally, a moderately accurate regression model ensures a low δ_R error. Therefore, our method produces Shapley value estimates Φ_i with minimal margin of error.

The detailed proof to 4.4 is provided in the Appendix.

5 EXPERIMENTS

We show the broad effectiveness of the proposed valuation framework and its general applicability to machine learning models through empirical evidence. We estimate the value of data in a machine learning model in MNIST, CIFAR10, and CIFAR100 datasets. We compare our method with Data Shapley [18] and Distributional Shaply [17].

Experiment Settings: Following the common practice in previous works, we extract the features from last layer of a pre-trained network and apply Shapley on this embedded vector. We sample


Algorithm 1 EcoVal Data Valuation

1: $M(.;\psi)$: Fully Trained Model
2: B: Training Dataset
3: B_D: Set of available points from the underlying distribution of B
4: $M_{-n}(x;\psi) \leftarrow$ Embedding of data x obtained from the nth last layer of the model
5: Let $E(x) = M_{-n}(x;\psi)$
6: Let A_c be a clustering algorithm then $(x_i, c_j) \leftarrow A_c(B_D) \forall x_i \in B_D$ where $c_j \in C$ is the cluster associated with x_i and C is the set of all clusters
7: Find valuation at cluster level
8: $V_{c_j} = U(B) - U(B \setminus c_j) \ \forall c_j \in C$
9: Initialize value V_i for each cluster member x_i
10: $V_i = V_{c_j}/n_{c_j}$, where n_{c_j} is the number of elements in cluster c_j to which x_i belongs
11: Initialize: $D \leftarrow []$
12: **for** $c_j \in C$ **do**
13:     Sample $X_j = \{x_1^j, x_2^j, \dots x_{n_c}^j\}$ from c_j
14:     $D \leftarrow D \cup X_j$
15: **end for**
16: Run TMC Shapley [18]
17: $(x_k, v_{TMC_k}) \leftarrow TMC(U^T, D) \forall x_k \in D$
18: Train a regression model R on the sampled data $\{(x_1, v_{TMC_1}), (x_2, v_{TMC_2})\dots(x_{|D|}, v_{TMC_{|D|}})\}$
19: **for** $c_j \in C$ **do**
20:     $(x_i^j, q_i^j) \leftarrow R(x_i^j) \forall x_i^j \in c_j$
21:     Let $\bar{x}_{c_j}$ be the centroid of the cluster c_j
22:     $(x_i^j, d_i^j) \leftarrow distance(x_i^j, \bar{x}_{c_j}) \forall x_i \in c_j$
23: **end for**
24: **for** $x_i \in B$ **do**
25:     Find correction term for α
26:     $\Gamma_{\alpha_i} = \frac{1}{1+V_{c_j}/n_{c_j}}(1 + \frac{q_i^j}{\sum_{z_k \in c_j} q_k^j} V_{c_j})$
27:     Find correction term for β_i^*
28:     $\Gamma_{\beta_i^*} = \frac{1}{1+V_{c_j}/n_{c_j}}(1 + \frac{d_i^j}{\sum_{z_k \in c_j} d_k^j} V_{c_j})$
29:     Final valuation = $V_i * (\Gamma_{\alpha_i} + \Gamma_{\beta_i^*} - 1)$
30: **end for**

---

a small subset, i.e. 200 samples from the original training data and run the baseline methods TMC-Shapley (Data Shapley) and distributional Shapley. 2000 samples are used for testing and hold-out for Shapley calculation. We keep 10,000 samples which are never seen by model or valuation method at any point, we call this *out-of-sample (OOS) set*. The rest of the samples are used as data distribution and exposed to Distributional Shapley, and our method during the clustering step and α correction step. We use Gaussian Mixture Models (GMM) for clustering. Our proposed method works for both in-distribution and OOS samples. As Data Shapley only works for in-distribution samples, we compare our results with Distribution Shapley for out-of-sample data. We use Gaussian Mixture Model (GMM) clustering with default parameters of
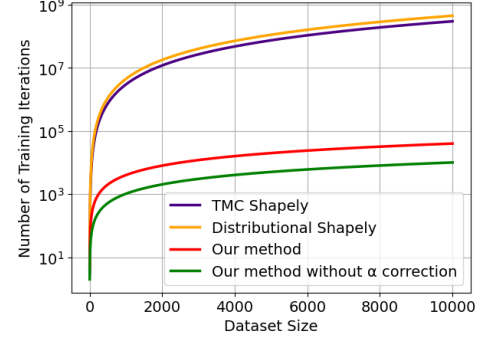

Figure 1: Computation cost in terms of number of training iterations required for the given dataset size. We compare EcoVal with TMC Shapley (also known as Data Shapley), distributional Shapley, and a lighweight version of EcoVal. Our method requires substantially lower number of training iterations for data valuation.

scikit-learn's implementation, covariance-type='full', tol=0.001, reg-covar=1e-06, max-iter=100, n-init=1, init-params='kmeans', and 30 mixtures/clusters.

5.1 Comparative Analysis of the Computational Time

The Data Shapley approximation method TMC Shapley [18] converges in approximately $3|B|$ (or $3 \times m$ in Eq. 2) Monte Carlo samples. Each Monte Carlo sample is a random permutation of the data points in the training set. The marginal contribution of a data point z in a given permutation is obtained as the performance difference between the model trained on data points before this datum, say S, and the model trained on $S \cup \{z\}$. Each point is added sequentially meaning $|B|$ training runs are required in a single Monte Carlo sample. This makes the number of training runs in the order of $O(|B|^2)$. Distributional Shapley's [17] time complexity is similar with T runs to get an unbiased estimate using different subsets S from the underlying data distribution. This makes the number of training runs of Distributional Shapley $O(T * |B|^2)$.

Our method performs clustering that takes less time than training a machine learning or deep learning model in most real-world scenarios. This is a one time effort, so the complexity is in the order of $O(1)$. Estimating the value of each cluster requires $O(p)$ training runs. Apart from that, our method involves running Data Shapley on a curated subset p containing an equal number of points from each cluster, this take $O(p^2)$ time. The size of this subset p is much smaller than $|B|$. The total number of training runs required is in the order of $O(1) + O(p^2) + O(p)$. We compare the computational cost associated with the TMC Shapley/Data Shapley with our EcoVal method. As illustrated in Figure 1, EcoVal requires significantly fewer training iterations, approximately 10^3 to 10^5, compared to 10^7 to 10^9 required by traditional TMC Shapley. Our method reduces computational overhead by employing a curated subset approach, which involves running Data Shapley on a subset p containing an equal number of points from each cluster. This curated subset approach requires computational resources in the order of $O(p^2)$, significantly less than the $O(|B|^2)$ required by the standard TMC
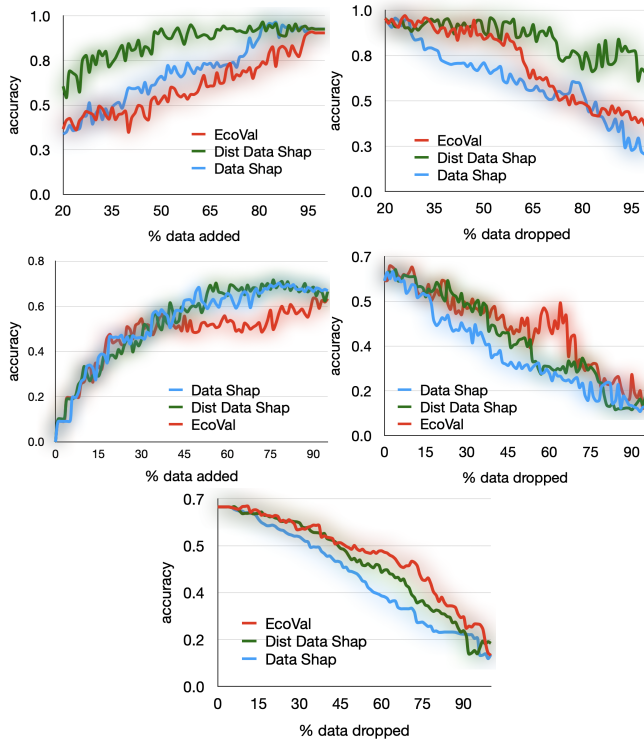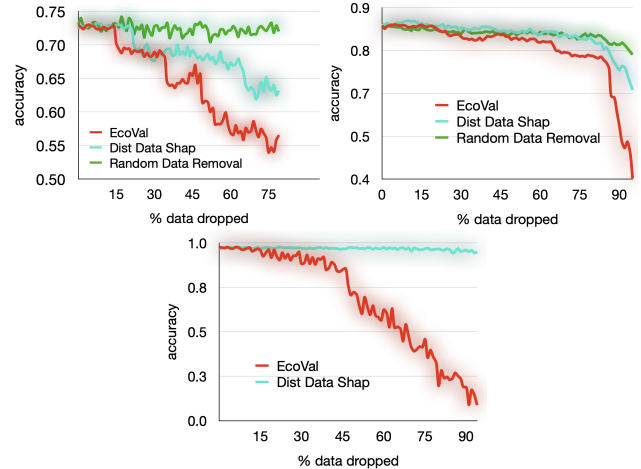
Figure 2: Accuracy difference with respect to the % of data points added or removed. We add or remove the highest valued data points first and then subsequently add or remove the lesser value data, respectively. The top, middle and bottom rows show the results for MNIST, CIFAR10, CIFAR100, respectively with in-distribution valuation. The EcoVal gives comparable or better performance when compared to Data Shapley and Distribution Data Shapley.

Figure 3: Data valuation on out-of-sample data (top left to bottom right: CIFAR10, CIFAR100, MNIST, respectively). Our EcoVal method outperforms Distributed Data Shapley and Random Data Removal by getting steeper performance drop with increasing % valuable data removal.

Shapley method, where $|B|$ represents the size of the full dataset. The size of subset p is much smaller than $|B|$, which explains our method's efficiency and scalability. With the increase in the dataset size, the utility of our EcoVal becomes more evident. Our method without α correction is even faster with negligible loss in valuation quality.

5.2 Data Point Addition and Removal Experiments

We evaluate the data valuation methods by running the data point addition and removal experiments as proposed in [18]. For a given model and dataset, the data points are added in the order of predicted value, i.e. from largest to lowest values, and the model is retrained for each addition. Similarly, another experiment is conducted where we remove samples with high values and observe the performance drop. The impact of removal and addition of high value data-points help us measure the effectiveness of data valuation techniques. We compare our results with state-of-the-art Data Shapley and Distributed Data Shapley valuation methods.

Removing most valued data points. We predict values of data-points using each valuation method and we measure the drop

in performance of model by removing most-valued data-points for each method. A better valuation method's high value data-points will result in a higher drop in performance. So, for removal of most valued points, the method resulting in higher performance drop is a better valuation method.

Adding most valued data points. This approach is vice-versa of the previous approach, we add most valued data-points into the training set and observe the increase in the performance. A higher increase on adding the top data-points shows better valuation method. Figure 2 shows the performance drop and increase upon adding and removing most valued points, respectively. It can be observed that EcoVal performance drop is slightly less than that of Data Shapley but significantly higher than the Distributed Data Shapley which is desired. Similar patterns can be observed in the data addition graph also.

Removing most valuable data points from out-of-sample set. The discussed earlier, EcoVal supports data valuation for out-of-sample data as well which is supported only by Distributed Data Shapley. Therefore, we compare the OOS valuation results between them. Figure 3 shows the performance drop by removing the most valuable points from an out-of-sample set of size 10, 000. It can be observed that EcoVal's performance drop is very high as compared to Distributed Data Shapley. The steep drop in the performance after removing the most valuable data points implies better precision for data valuation in our EcoVal framework.

5.3 Effect of the Adjustment Terms

We observe the effect of removing different adjustment terms $\Gamma_{\beta_i^*}$, Γ_{α_i} or both in the EcoVal framework and show the results in Figure 4. The overall EcoVal framework with the terms α and β performs the best, in general. Eliminating one of the adjustment terms deteriorates the quality of the valuation by a small margin. Removing both corrections significantly impacts the quality of data valuation. This is particularly visible in the initial phase of adding the most
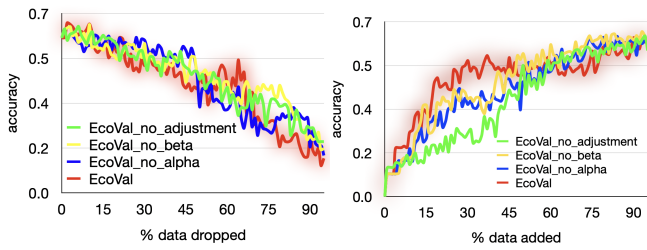
Figure 4: Effect of adding different adjustment terms (refer Section 4.2). EcoVal: the full proposed method, EcoVal_no_alpha: removal of α adjustment term, EcoVal_no_beta: removal of β adjustment term, EcoVal_no_adjustment: EcoVal without any adjustment terms, i.e. the mean of the cluster value used as the data value.

significant data points. It should be noted that eliminating Γ_{α_i} only affects the valuation quality marginally, but completely removes the need for model training, giving an even more efficient version of our valuation method.

The performance of different variations of our method (EcoVal, EcoVal-no-α, EcoVal-no-β, EcoVal-no-adj) would vary depending on the intra-class and inter-class variations present in a dataset. In Figure 4, the performance differences are not sometimes consistent. We report our observation based on the experiments with CIFAR-10 dataset. More complex datasets with larger variations may better reveal the impact of these adjustment factors, which is the future scope of this work.

5.4 Mutual Influence of Clustering Methods and Adjustment Terms

The effectiveness of EcoVal is intrinsically related to the success of the clustering method, as the initial valuation is carried out at the cluster level. The adjustment terms are meant to marginally correct the data value distribution within each cluster. The similarity within a single cluster compared to the similarity between clusters significantly influences the structure and application of the adjustment factors. We assume that data points within the same cluster have a higher degree of similarity compared to points across different clusters, a standard assumption in clustering algorithms such as Gaussian mixture models (GMM). However, the variance in the degree of similarity across different clusters (inter-cluster dissimilarity) justifies the need for a scaling factor such as Γ_{α_i}, introduced in Equation 15. This accounts for the relative contribution of an entire cluster to the model's performance, acknowledging that some clusters may be more pivotal due to their positioning, density, or the nature of the data points they contain. This process depends on the intra-cluster variation present in a dataset. For some dataset, a simple distribution of the value of the cluster without correction factors can also give a good valuation, leaving a minimal scope of correction.

6 CONCLUSION

This work presents a focused study on improving the speed of data valuation in machine learning models. We develop an efficient data valuation method that is significantly fast and practical for working with large datasets. Our method works for both in-distribution and out-of-sample data. The proposed EcoVal data valuation framework shows comparable and sometimes even better results than the existing approaches for in-distribution data. For out-of-sample data points, our method significantly outperforms competing methods, thus establishing a new state-of-the-art. This proves our method's utility in a data market, where new data points analogous to our out-of-sample set are generated every passing instant. Our valuation also shows negligible error margin with the vanilla Shapley value approximation. The points mentioned above collectively make the proposed method a robust and scalable approach to estimate the value of data across a variety of machine learning models.

REFERENCES

[1] Anish Agarwal, Munther Dahleh, and Tuhin Sarkar. 2019. A marketplace for data: An algorithmic solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. 701–726.
[2] Grant Allan, Michelle Gilmartin, Peter McGregor, Karen Turner, and J Kim Swales. 2009. Economics of energy efficiency. In *International Handbook on the Economics of Energy*. Edward Elgar Publishing.
[3] Kenneth J Arrow, Hollis B Chenery, Bagicha S Minhas, and Robert M Solow. 1961. Capital-labor substitution and economic efficiency. *The review of Economics and Statistics* (1961), 225–250.
[4] Robert J Aumann and Lloyd S Shapley. 2015. *Values of non-atomic games*. Princeton University Press.
[5] Robert J Barro. 1997. *Macroeconomics*. MIT Press.
[6] Eric Bax. 2019. Computing a data dividend. *arXiv preprint arXiv:1905.01805* (2019).
[7] David E Bloom, David Canning, and Jaypee Sevilla. 2004. The effect of health on economic growth: a production function approach. *World development* 32, 1 (2004), 1–13.
[8] Lawrence Blume, Steven Durlauf, and Lawrence E Blume. 2008. *The new Palgrave dictionary of economics*. Palgram Macmillan.
[9] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. 2018. L-Shapley and C-Shapley: Efficient Model Interpretation for Structured Data. In *International Conference on Learning Representations*.
[10] CW COBB. 1928. A Theory of Production. *American Economic Review* 18 (1928), 139–165.
[11] Shay Cohen, Eytan Ruppin, and Gideon Dror. 2005. Feature selection based on the Shapley value. In *Proceedings of the 19th international joint conference on Artificial intelligence*. 665–670.
[12] Ian Covert and Su-In Lee. 2021. Improving kernelshap: Practical shapley value estimation using linear regression. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 3457–3465.
[13] Ian Covert, Scott Lundberg, and Su-In Lee. 2021. Explaining by removing: A unified framework for model explanation. *Journal of Machine Learning Research* 22, 209 (2021), 1–90.
[14] Pradeep Dubey, Abraham Neyman, and Robert James Weber. 1981. Value theory without efficiency. *Mathematics of Operations Research* 6, 1 (1981), 122–128.
[15] Maryam Farboodi and Laura Veldkamp. 2021. *A model of the data economy*. Technical Report. National Bureau of Economic Research.
[16] Raul Castro Fernandez, Pranav Subramaniam, and Michael J Franklin. [n. d.]. Data Market Platforms: Trading Data Assets to Solve Data Problems. *Proceedings of the VLDB Endowment* 13, 11 ([n. d.]).
[17] Amirata Ghorbani, Michael Kim, and James Zou. 2020. A distributional framework for data valuation. In *International Conference on Machine Learning*. PMLR, 3535–3544.

[18] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*. PMLR, 2242–2251.

[19] Amirata Ghorbani and James Y Zou. 2020. Neuron shapley: Discovering the responsible neurons. *Advances in neural information processing systems* 33 (2020), 5922–5932.

[20] Michel Grabisch and Marc Roubens. 1999. An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory* 28 (1999), 547–565.

[21] Faruk Gul. 1989. Bargaining foundations of Shapley value. *Econometrica: Journal of the Econometric Society* (1989), 81–95.

[22] Eric A Hanushek. 2020. Education production functions. In *The economics of education*. Elsevier, 161–170.

[23] Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. 2022. Datamodels: Understanding predictions with data and data with predictions. In *International Conference on Machine Learning*. PMLR, 9525–9587.

[24] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li4 Ce Zhang, and Costas Spanos1 Dawn Song. [n. d.]. Efficient Task-Specific Data Valuation for Nearest Neighbor Algorithms. *Proceedings of the VLDB Endowment* 12, 11 ([n. d.]).

[25] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. 2019. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 1167–1176.

[26] Charles I Jones and Christopher Tonetti. 2020. Nonrivalry and the Economics of Data. *American Economic Review* 110, 9 (2020), 2819–2858.

[27] Ehud Kalai and Dov Samet. 1987. On weighted Shapley values. *International journal of game theory* 16 (1987), 205–222.

[28] Bojan Karlaš, David Dao, Matteo Interlandi, Bo Li, Sebastian Schelter, Wentao Wu, and Ce Zhang. 2022. Data debugging with shapley importance over end-to-end machine learning pipelines. *arXiv preprint arXiv:2204.11131* (2022).

[29] Eda Kemahlıoğlu-Ziya and John J Bartholdi III. 2011. Centralizing inventory in supply chains by using Shapley value to allocate the profits. *Manufacturing & Service Operations Management* 13, 2 (2011), 146–162.

[30] Guennadi A Khatskevich and Andrei F Pranevich. 2018. Production functions with given elasticities of output and production. (2018).

[31] Yongchan Kwon, Manuel A Rivas, and James Zou. 2021. Efficient computation and analysis of distributional shapley values. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 793–801.

[32] Yongchan Kwon and James Zou. 2022. Beta Shapley: a Unified and Noise-reduced Data Valuation Framework for Machine Learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 8780–8802.

[33] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the Carbon Emissions of Machine Learning. *arXiv preprint arXiv:1910.09700* (2019).

[34] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[35] Sudhanshu K Mishra. 2007. A brief history of production functions. *Available at SSRN 1020577* (2007).

[36] Hervé Moulin. 1992. An application of the Shapley value to fair division with money. *Econometrica: Journal of the Econometric Society* (1992), 1331–1349.

[37] Roger B Myerson. 1977. Graphs and cooperation in games. *Mathematics of operations research* 2, 3 (1977), 225–229.

[38] Ki Nohyun, Hoyong Choi, and Hye Won Chung. 2022. Data valuation without training of a model. In *The Eleventh International Conference on Learning Representations*.

[39] Jian Pei. 2020. Data Pricing – From Economics to Data Science. Association for Computing Machinery.

[40] Alvin E Roth and Robert E Verrecchia. 1979. The Shapley value as applied to cost allocation: a reinterpretation. *Journal of Accounting Research* (1979), 295–303.

[41] LS Shapley. 1953. A Value for n-Person Games. In *Contributions to the Theory of Games (AM-28), Volume II*. Princeton University Press, 307–318.

[42] Lloyd S Shapley et al. 1953. A value for n-person games. (1953).

[43] Ronald William Shephard. 2015. *Theory of cost and production functions*. Princeton University Press.

[44] Ronald W Shephard and Rolf Färe. 1974. The law of diminishing returns. In *Production Theory: Proceedings of an International Seminar Held at the University at Karlsruhe May–July 1973*. Springer, 287–318.

[45] Robin C Sickles and Valentin Zelenyuk. 2019. *Measurement of productivity and efficiency*. Cambridge University Press.

[46] Raghav Singal, Omar Besbes, Antoine Desir, Vineet Goyal, and Garud Iyengar. 2019. Shapley meets uniform: An axiomatic framework for attribution in online advertising. In *The World Wide Web Conference*. 1713–1723.

[47] Mukund Sundararajan and Amir Najmi. 2020. The many Shapley values for model explanation. In *International conference on machine learning*. PMLR, 9269–9278.

[48] Siyi Tang, Amirata Ghorbani, Rikiya Yamashita, Sameer Rehman, Jared A Dunnmon, James Zou, and Daniel L Rubin. 2021. Data valuation for medical imaging using Shapley value and application to a large-scale chest X-ray dataset. *Scientific reports* 11, 1 (2021), 8366.

[49] Tarun Wadhwa. 2020. Economic impact and feasibility of data dividends.

[50] Jiachen T Wang and Ruoxi Jia. 2023. Data banzhaf: A robust data valuation framework for machine learning. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 6388–6421.

[51] Jiachen T. Wang and Ruoxi Jia. 2023. Data Banzhaf: A Robust Data Valuation Framework for Machine Learning. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 206)*, Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent (Eds.). PMLR, 6388–6421.

[52] Rui Wang, Xiaoqian Wang, and David I Inouye. 2020. Shapley Explanation Networks. In *International Conference on Learning Representations*.

[53] Mohammad Zaeri-Amirani, Fatemeh Afghah, and Sajad Mousavi. 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)*. IEEE, 319–323.

<div style="text-align:center">

Appendix

</div>

A PROOFS

A.1 Theorem 3.4

Let equal number of samples are used from each cluster to run TMC Shapley. Then the *intrinsic value* of a datum is independent of proportion and bias in the data distribution. If n_s such samples exist, the value is divided into these n_s samples. From Shapely axioms, the data Shapley at the current stage of TMC becomes approximately $\frac{\alpha_i}{n_s}$.

For rest of the samples in the TMC, we train a regression model R for predicting $\frac{\alpha_i}{n_s}$ for an input data. If the predicted Shapley for any $z_i \in c$ is Q_i, then assuming no error is introduced due to the TMC Shapely algorithm, this gives us

$$Q_i = \frac{\alpha_i}{n_s} + \delta_{R_i}, \tag{27}$$

where δ_{R_i} is error introduced by the regression model. Q_i denotes $b * \alpha_i$, where b is some constant. We use this to obtain the adjustment factor for α_i in Eq. 14. Assuming V_i and d_i do not have any error, we take the differentiation of Eq. 21 with respect to z

$$\frac{\partial \Phi_i}{\partial z} = \frac{1}{\partial z} \left(\frac{V_i}{1 + V_c/n_c} \right) \left[\frac{n_c \partial Q_i}{\sum_{z_j \in c} Q_j} - \frac{n_c Q_i \partial (\sum_{z_j \in c} Q_j)}{(\sum_{z_j \in c} Q_j)^2} \right]. \tag{28}$$

Intuition. V_i does not have any error as this is the difference between the performance with and without the cluster c divided by a constant. Both the values can be directly computed from the model. Similarly, d_i is the distance of the datum from the centroid of the cluster c which can be calculated without any error.

Comparing Eq. 28 with the change in the Shapley value leads to the following inequality.

$$\Delta \Phi_i \le \frac{V_c/n_c}{1 + V_c/n_c} \left[\frac{\delta_R n_c}{\sum_{z_j \in c} Q_j} + \frac{n_c Q_i (n_c \delta_R)}{(\sum_{z_j \in c} Q_j)^2} \right],$$

where $\delta_R = \max_i \delta_{R_i}$ is the maximum error of the regression model

$$\Delta \Phi_i \le V_c \left[\frac{\delta_R}{\sum_{z_j \in c} Q_j} + \frac{n_c Q_i \delta_R}{(\sum_{z_j \in c} Q_j)^2} \right],$$

as $V_c \ge 0$, $n_c \ge 1$ therefore, $\frac{1}{1+V_c/n_c} \le 1$. From Eq. 24 and Eq. 30

$$\Delta \Phi_i \le (n_c \bar{\Phi}_c + n_c \epsilon/2) \left[\frac{\delta_R}{\sum_{z_j \in c} Q_j} + \frac{n_c Q_i \delta_R}{(\sum_{z_j \in c} Q_j)^2} \right].$$

Ignoring factors with multiples of ϵ and δ as these values are very small. We get the final difference between the original Shapley value and our proposed approximated Value as below

$$\Delta \Phi_i \approx \frac{n_c \bar{\Phi}_c \delta_R}{\sum_{z_j \in c} Q_j} + \frac{n_c^2 \bar{\Phi}_c Q_i \delta_R}{(\sum_{z_j \in c} Q_j)^2}, \tag{29}$$

$\bar{\Phi}_c$ and Q_i cannot be arbitrarily large as they are the average Shapely value for a cluster and change in performance due to a data point z_i. With increasing cluster size n_c, the corresponding predicted value $\sum_{z_j \in c} Q_j$ will increase. Thus, $n_c / \sum_{z_j \in c} Q_j$ can not be very large. The δ_R error will be low for a moderately good regression model. Thus, our method estimates the Shapely value Φ_i with negligible error.

A.2 Corollary 4.3.1

The value V_c of the a cluster by Shapley axioms is defined as

$$V_c = \sum_{z_j \in c} \Phi(z_j) = n_c \bar{\Phi}_c + \sum \delta(z_j). \tag{30}$$

The sum of the Shapley values within a cluster equalling the leave-cluster-out value of the cluster follows logically from the axioms of the Shapley value, in particular, additivity and efficiency.

Additivity: If we treat each data point as contributing a separate game to the performance, the total Shapley value of a cluster should naturally be the sum of the Shapley values of each data point within the cluster.

Efficiency: This axiom ensures that the total value generated by the coalition is fully distributed among the players. If the Shapley value calculation respects this axiom, then the allocation to a cluster should match the cumulative contribution of its members.

We can consider a simple proof by induction on the number of data points in the cluster:

1. Base Case: If a cluster has only one data point, then the Shapley value of the cluster is clearly equal to the Shapley value of the single data point.

2. Inductive Step: Assume the proposition holds for all clusters of size k. For a cluster of size $k + 1$, if you remove one data point, by the induction hypothesis, the sum of the Shapley values of the remaining k data points equals the value of these k data points. Adding the $k+1$-st point, by the additivity axiom, the overall Shapley value would be the sum of the individual Shapley values.