Efficient Computation and Analysis of Distributional Shapley Values

Yongchan Kwon   Manuel A. Rivas   James Zou

{ychkwon,mrivas,jamesz}@stanford.edu

Department of Biomedical Data Science, Stanford University, CA, USA

Abstract

Distributional data Shapley value (DShapley) has recently been proposed as a principled framework to quantify the contribution of individual datum in machine learning. DShapley develops the foundational game theory concept of Shapley values into a statistical framework and can be applied to identify data points that are useful (or harmful) to a learning algorithm. Estimating DShapley is computationally expensive, however, and this can be a major challenge to using it in practice. Moreover, there has been little mathematical analyses of how this value depends on data characteristics. In this paper, we derive the first analytic expressions for DShapley for the canonical problems of linear regression, binary classification, and non-parametric density estimation. These analytic forms provide new algorithms to estimate DShapley that are several orders of magnitude faster than previous state-of-the-art methods. Furthermore, our formulas are directly interpretable and provide quantitative insights into how the value varies for different types of data. We demonstrate the practical efficacy of our approach on multiple real and synthetic datasets.

1 Introduction

Data valuation has emerged as an important topic for machine learning (ML) as well as for the broader discussions around the economics of data. Proposed policies such as the Designing Accounting Safeguard to Help Broaden Oversight and Regulations on Data Act, also known as DASHBOARD Act, and the Data

Dividend in the US would stipulate that companies need to quantify the value of the data that they collect from customers (Warner, 2019; Wadhwa, 2020). Such valuation could have important implications for policy, regulation, taxation and potentially even for individual compensation (Posner and Weyl, 2018). Recently data Shapley, a data valuation framework based on the foundational Shapley value in economics, has gained significant attention (Ghorbani and Zou, 2019; Jia et al., 2019b). Data Shapley is appealing from a policy perspective because it inherits the same fair allocation properties that the original Shapley value uniquely satisfies. Moreover, it has shown to empirically capture the notion of which datum helps or harms the ML model.

A fundamental limitation of data Shapley, however, is that it is defined with respect to a fixed dataset. The statistical and random nature of data is ignored. Accordingly, data Shapley needs to be recalculated even when the dataset changes slightly, which is computationally expensive, and it could also be unstable for randomly drawn datasets. To tackle these challenges, Ghorbani et al. (2020) proposed distributional Shapley value (DShapley) as the natural statistical extension of the Shapley value, by considering the expected value of data Shapley value with respect to the underlying distribution. While DShapley is numerically more stable and does not require the aforementioned recalculation, DShapley is still mathematically challenging to analyze and computationally hard to estimate.

In this paper, we address these challenges by developing rigorous analyses and computationally efficient algorithms for DShapley. **Theoretical contributions:** we develop the first analytic expressions for DShapley for linear regression, binary classification, and non-parametric density estimation, which are widely used canonical examples of supervised and unsupervised learning. Our formulations are also easy to interpret and provide direct insights into how DShapley behaves for different data. **Algorithmic contributions:** based on our theory, we provide new algorithms to efficiently estimate DShapley which is several or-

ders of magnitude faster than previous state-of-the-art methods. We support our analyses with numerical experiments on both real and synthetic datasets.

Related works Shapley value was initially proposed in a seminar paper (Shapley, 1953) and has been studied extensively in the field of cooperative game theory (Dubey et al., 1981; Grabisch and Roubens, 1999; Aumann and Shapley, 2015). Shapley value has been widely applied in economics (Gul, 1989; Moulin, 1992), management science (Dubey, 1982) and has also been appeared in ML literature. Examples include feature selection (Cohen et al., 2005; Zaeri-Amirani et al., 2018), data marketplace design (Agarwal et al., 2019; Fernandez et al., 2020), and model explanation (Lundberg and Lee, 2017; Chen et al., 2019; Sundararajan and Najmi, 2019; Ghorbani and Zou, 2020).

Another body is Shapley value-based data valuation methods, yet most of the literature focuses on data Shapley values (Ghorbani and Zou, 2019; Jia et al., 2019b). Data Shapley value has been shown to empirically work better than other methods of data valuation, such as using leave-one-out residual estimate (Cook and Weisberg, 1982), or influence-based scores (Hampel, 1974; Koh and Liang, 2017), but it can cause expensive computational costs when data are regularly collected. Other promising data valuation schemes have been proposed to leverage reinforcement learning (Yoon et al., 2019). These approaches lack the fairness principles that has uniquely satisfied by the Shapley value.

DShapley was introduced as a rigorous statistical extension of Shapley value (Ghorbani et al., 2020). Previous to our work, the only computationally efficient form for data Shapley is just for the nearest neighbor classifier (Jia et al., 2019a); and similar results are not known for DShapley. Our work develops principled and efficient methods for analyzing and computing DShapley.

2 Preliminaries

We review existing Shapley value-based data valuation methods. To begin with, we define some notations. Let Z be a random variable for data defined on $\mathcal{Z} \subseteq \mathbb{R}^d$ and denote its distribution by P_Z. In supervised learning, we set $Z = (X, Y)$ defined on $\mathcal{X} \times \mathcal{Y}$, where X and Y are the input and its label, respectively, and in unsupervised learning $Z = X$. We denote a utility function by $U : \cup_{j=0}^{\infty} \mathcal{Z}^j \to \mathbb{R}$. Here, the utility function describes model performance. For instance, in classification, $U(S)$ could be the test accuracy of a model trained using a subset $S \subseteq \mathcal{X} \times \mathcal{Y}$. We define the marginal contribution of $z^* \in \mathcal{Z}$ with respect to $S \subseteq \mathcal{Z}$ as $\Delta(z^*; U, S) := U(S \cup \{z^*\}) - U(S)$. We use

the conventions $\mathcal{Z}^0 := \{\emptyset\}$ and $U(\emptyset) = 0$. For a set S, we denote its cardinality by $|S|$, and we use $[m]$ to denote a set of integers $\{1, \ldots, m\}$.

Data Shapley value applies the cooperative game theory concept of Shapley value to ML problems (Ghorbani and Zou, 2019; Jia et al., 2019b). More precisely, given a utility function U and a fixed dataset $B \subseteq \mathcal{Z}$ with $|B| = m$, data Shapley value of a point $z^* \in B$ is defined as

$$\phi(z^*; U, B) := \frac{1}{m} \sum_{j=1}^{m} \frac{1}{\binom{m-1}{j-1}} \sum_{S \in B_j^{\backslash z^*}} \Delta(z^*; U, S), \quad (1)$$

where $B_j^{\backslash z^*} := \{S \subseteq B \backslash \{z^*\} : |S| = j - 1\}$ for $j \in \mathbb{N}$. Note that the cardinality $|B_j^{\backslash z^*}|$ is $\binom{m-1}{j-1}$ for all $j \in [m]$. That is, data Shapley value (1) is a weighted average of the marginal contribution $\Delta(z^*; U, S)$. Data Shapley provides a principled data valuation regime in that the value (1) uniquely satisfies the natural properties of fair valuation, namely, symmetry, null player, and additivity (Ghorbani and Zou, 2019; Jia et al., 2019a). We review these properties and the uniqueness of data Shapley value in the Supplementary Material.

Despite the aforementioned promising theoretical characteristics, data Shapley value has a critical limitation; the original data Shapley value is defined with respect to a fixed dataset B. Even if a single point in B is changed, in principle, all of the values should be recomputed and the exact computation of the value costs exponential computational complexity. This is particularly problematic in typical statistics and ML settings, where the data points are regularly collected from an underlying distribution. In order to resolve this issue and capture the statistical nature of data valuation, DShapley has been proposed where data Shapley is treated as a random variable (Ghorbani et al., 2020). To be more specific, given a utility function U, a data distribution P_Z, and some $m \in \mathbb{N}$, Ghorbani et al. (2020) defined DShapley of a point z^* as

$$\nu(z^*; U, P_Z, m) := \mathbb{E}_{B \sim P_Z^{m-1}}[\phi(z^*; U, B \cup \{z^*\})]. \quad (2)$$

DShapley (2) is the expectation of data Shapley value (1) over random datasets of size m containing z^*. Ghorbani et al. (2020) further showed that DShapley possesses some desirable properties. For instance, DShapley is stable under small perturbations to the data points themselves and to the underlying data distribution (Ghorbani et al. (2020, Theorems 2.7 and 2.8)), which have not been clear with (1). However, estimating DShapley is still computationally expensive and thus it critically hampers the practical use of DShapley. In this paper, we focus on canonical problems of linear regression, binary classification, and non-parametric

density estimation, deriving new expressions for DShapley that lead to new mathematical insights and efficient computation algorithms.

3 Distributional Shapley values for linear regression and classification

We present rigorous analyses of DShapley for linear regression problems. In Sec. 3.1, we first provide a general reformulation of DShapley without distributional assumptions on inputs. In Sec. 3.2, we simplify DShapley as a function of Mahalanobis distance and an error when inputs are Gaussian. In Sec. 3.3, we consider sub-Gaussian inputs and provide upper and lower bounds for DShapley. In Sec. 3.4, we present an application of our theoretical result to binary classification.

3.1 A general reformulation of distributional Shapley values

Throughout this section, we let (X, Y) be a pair of input and output random variables defined on $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^p \times \mathbb{R}$. We assume that $Y = X^T\beta + e$ is the underlying linear model where e is a random error whose mean is zero and variance is σ^2. Here, X can come from an arbitrary distribution with bounded first two moments. For a subset $S \subseteq \mathcal{X} \times \mathcal{Y}$, we denote a design matrix and its corresponding output vector based on S by $X_S \in \mathbb{R}^{|S| \times p}$ and $Y_S \in \mathbb{R}^{|S|}$, respectively. For $\gamma \geq 0$, the ridge regression estimator based on S is defined as $\hat{\beta}_{S,\gamma} := (X_S^T X_S + \gamma I_p)^{-1} X_S^T Y_S$ where I_p is the $p \times p$ identity matrix. For $q \in \mathbb{N}$, a constant $C_{\text{lin}} > 0$, and an estimator $\hat{\beta} \in \mathbb{R}^p$, we define a utility function as $U_q(S, \hat{\beta}) := (C_{\text{lin}} - \int (y - x^T\hat{\beta})^2 dP_{X,Y}(x,y))\mathbb{1}(|S| \geq q)$. Here, $\mathbb{1}(\cdot)$ is the indicator function. To this end, we suppress the notation if the ridge regression estimator is used, i.e., $U_{q,\gamma}(S) := U_q(S, \hat{\beta}_{S,\gamma})$. We denote the Gaussian distribution with mean μ and covariance Σ by $\mathcal{N}(\mu, \Sigma)$. Lastly, we denote the data to be valued by (x^*, y^*) and its error by $e^* := y^* - x^{*T}\beta$.

The DShapley can be equivalently expressed as follows (Ghorbani et al., 2020):

$$\nu((x^*, y^*); U_{q,\gamma}, P_{X,Y}, m)$$
$$= \mathbb{E}_{j \sim [m]} \mathbb{E}_{S \sim P_{X,Y}^{j-1}}[\Delta((x^*, y^*); U_{q,\gamma}, S)] \quad (3)$$

where $j \sim [m]$ denotes j follows a uniform distribution over $[m]$. Using Equation (3), we further derive a general reformulation of DShapley in the following proposition.

Proposition 1 (A general form of DShapley). *Let* $\mathbb{E}[Y \mid X] = X^T\beta$, $\text{Var}(Y \mid X) = \sigma^2$, *and* $\mathbb{E}(XX^T) = \Sigma_X$. *Then, for any* $q \geq 2$ *and some fixed constant* C_{lin}, *DShapley of a point* (x^*, y^*) *with the ridge regression*

estimator is given by

$$\nu((x^*, y^*); U_{q,\gamma}, P_{X,Y}, m)$$
$$= \frac{1}{m} \sum_{j=q}^{m} \mathbb{E}_{X_S \sim P_X^{j-1}} \left[\frac{x^{*T} A_{S,\gamma}^{-1} \Sigma_X A_{S,\gamma}^{-1} x^*}{(1 + x^{*T} A_{S,\gamma}^{-1} x^*)^2} \right.$$
$$\left. \times \left((2 + x^{*T} A_{S,\gamma}^{-1} x^*)\sigma^2 - e^{*2} \right) \right] + h(\gamma),$$

where $A_{S,\gamma}^{-1} := (X_S^T X_S + \gamma I_p)^{-1}$ *and* $h(\gamma)$ *is a term such that* $\lim_{\gamma \to 0+} h(\gamma)/(\gamma \log(\gamma)) = 0$ *and* $h(0) = 0$.

In the expression (3), different choices of C_{lin} in the utility function $U_{q,\gamma}$ cause constant changes in DShapley. To be more specific, for a fixed $C \in \mathbb{R}$ and for all $S \subseteq \mathcal{X} \times \mathcal{Y}$, suppose $U_{q,\gamma}(S) = \tilde{U}_{q,\gamma}(S) + C\mathbb{1}(|S| \geq q)$. Then DShapley is $\nu((x^*, y^*); U_{q,\gamma}, P_{X,Y}, m) = \nu((x^*, y^*); \tilde{U}_{q,\gamma}, P_{X,Y}, m) + C/m$. In this respect, we simply choose a constant C_{lin} that gives the simplest form in Proposition 1 and the following results.

Proposition 1 simplifies the expected value of the marginal contributions of (x^*, y^*) in Equation (3) with a few terms such as the squared error e^{*2} and the ridge leverage score $x^{*T} A_{S,\gamma}^{-1} x^*$ (Cohen et al., 2017; McCurdy, 2018). This new formulation provides mathematical insights and interpretations. For a fixed x^*, DShapley is negatively related to the squared error e^{*2} as long as γ is small enough; as the error decreases, DShapley increases. In addition, DShapley is determined only by the first two conditional moments of Y given X, meaning that it does not rely on other higher moments or a particular distribution of Y. Furthermore, it is noteworthy that Proposition 1 does not require a specific distributional assumption on X except for the moment condition $\mathbb{E}(XX^T) = \Sigma_X$. In the following sections, we pay more attention to the input distribution and propose computationally efficient algorithms for DShapley.

3.2 Distributional Shapley value when inputs are Gaussian

When input data are Gaussian, we introduce a new expression for DShapley in the following theorem. To begin with, for $k \geq 1$, we denote the Chi-squared distribution with k degree of freedom by χ_k^2.

Theorem 2 (DShapley when inputs are Gaussian). *Assume* $\mathbb{E}[Y \mid X] = X^T\beta$, $\text{Var}(Y \mid X) = \sigma^2$ *and* $X \sim \mathcal{N}_p(0, \Sigma_X)$. *For* $j \geq p$, *let* T_j *be a Chi-squared random variable with* $j - p + 1$ *degree of freedom, i.e.,* $T_j \sim \chi_{j-p+1}^2$. *Then, for any* $q \geq p + 3$ *and some fixed constant* C_{lin}, *DShapley of a point* (x^*, y^*) *with the least squares estimator is given by*

$$\nu((x^*, y^*); U_{q,0}, P_{X,Y}, m)$$

$$= -\frac{1}{m} \sum_{j=q}^{m} \mathbb{E}\left[\frac{j-1}{j-p} \frac{\left(x^{*T}\Sigma_X^{-1}x^* e^{*2} + T_j\sigma^2 \right)}{(x^{*T}\Sigma_X^{-1}x^* + T_j)^2} \right], \quad (4)$$

where the expectation is with respect to the Chi-squared distributions.

Theorem 2 presents a new representation of DShapley when $\gamma = 0$ and inputs are Gaussian. The new form (4) depends only on the two terms, the error e^{*2} and the term $x^{*T}\Sigma_X^{-1}x^*$, also known as the Mahalanobis distance of x^* from zero with respect to Σ_X. Likewise Proposition 1, a direct implication is that any points with the same error level have the same DShapley when they have the same Mahalanobis distance. In addition, a role of e^{*2} is also explicitly explained. DShapley for the point with the smaller squared error is higher than the other point, *i.e.*, $\nu((x^*, y_1^*); U_{q,0}, P_{X,Y}, m) \leq \nu((x^*, y_2^*); U_{q,0}, P_{X,Y}, m)$ if $e_1^{*2} \geq e_2^{*2}$. This inequality matches our intuitions that the big error e^{*2} is likely to produce small marginal contributions $\Delta((x^*, y^*); U_{q,\gamma}, S)$. We provide illustrations on how DShapley changes with respect to $x^{*T}\Sigma_X^{-1}x^*$ and e^{*2} in the Supplementary Material.

Efficient estimation of DShapley As for the estimation of DShapley $\nu((x^*, y^*); U_{q,0}, P_{X,Y}, m)$, we propose to use the Monte-Carlo approximation method. We describe a simple version of the proposed algorithm in Alg. 1. A detailed version is provided in the Supplementary Material.

A similar idea has been suggested in a number of algorithms including `TMC-SHAPLEY` (Ghorbani and Zou, 2019) or $\mathcal{D}$`-SHAPLEY` (Ghorbani et al., 2020). Although the previous state-of-the-art algorithms and the proposed algorithm make use of the Monte-Carlo method, there are notable differences. Since the previous algorithms are based on Equation (3), they require the utility evaluation $U_{q,0}(S)$ for every random dataset S. This computation is expensive because it includes the matrix inversion $(X_S^T X_S)^{-1}$. However, the proposed algorithm avoids such computational costs because the new form (4) has nothing to do with a random dataset S. This characteristic is not obtained with Equation (3) and Proposition 1. In terms of the computational complexity, when the maximum number of Monte-Carlo samples is T, the previous state-of-the-art algorithms require $O(mTp^3)$ computations. In contrast, the proposed Alg. 1 only needs to perform the matrix inversion once for Σ_X^{-1}, and the computational complexity for the proposed algorithm is $O(mT + p^3)$, which is substantially smaller since T is usually large.

Algorithm 1 DShapley for the least squares estimator when inputs are Gaussian.

Require: Estimates for $x^{*T}\Sigma_X^{-1}x^*$, e^{*2}, and σ^2. The maximum number of Monte Carlo samples T. A utility hyperparameter $q \geq p + 3$.

    procedure
        for $j \in \{q, \ldots, m\}$ **do**
            Sample $t_{[1]}, \ldots, t_{[T]}$ from the χ^2_{j-p+1}.
            $A_j \leftarrow \frac{1}{T} \sum_{i=1}^{T} \frac{j-1}{j-p} \frac{x^{*T}\Sigma_X^{-1}x^* e^{*2} + t_{[i]}\sigma^2}{(x^{*T}\Sigma_X^{-1}x^* + t_{[i]})^2}$
            $\hat{\nu} \leftarrow \hat{\nu} - A_j/m$
        end for
        $\hat{\nu}((x^*, y^*); U_q, P_{X,Y}, m) \leftarrow \hat{\nu}$
    end procedure

3.3 Distributional Shapley values when inputs are sub-Gaussian

In this section, we develop closed-form bounds for DShapley when inputs are sub-Gaussian. To be more formal, we first define the sub-Gaussian.

Definition 1 (Sub-Gaussian). *We say that a random variable X in $\mathbb{R}$ is sub-Gaussian if there are positive constants C_{sub} and v_{sub} such that for every $t > 0$, $P(|X| > t) \leq C_{\text{sub}} e^{-v_{\text{sub}}t^2}$ holds. In addition, we say that a random vector X in $\mathbb{R}^p$ is sub-Gaussian if the one-dimensional marginals $\langle X, x \rangle$ are sub-Gaussian random variables for all $x \in \mathbb{R}^p$.*

Note that a class of sub-Gaussian includes many useful random variables such as Gaussian and any bounded random variables (Vershynin, 2010). Now we develop bounds for DShapley in the following theorem.

Theorem 3 (Upper and lower bounds for DShapley when inputs are sub-Gaussian). *Assume that $\mathbb{E}[Y \mid X] = X^T\beta$ and $\text{Var}(Y \mid X) = \sigma^2$. Suppose $\mathcal{Y}$ is bounded and X are sub-Gaussian in $\mathbb{R}^p$ with $\mathbb{E}(XX^T) = \Sigma_X$. Then, for $q \geq 2$ and some fixed constant C_{lin}, DShapley of a point (x^*, y^*) with the ridge regression estimator has the following bounds.*

$$h(\gamma) + \frac{1}{m} \sum_{j=q-1}^{m-1} \frac{x^{*T}\Sigma_X^{-1}x^* \Lambda_{\text{lower}}^2(j)}{(1 + x^{*T}\Sigma_X^{-1}x^* \Lambda_{\text{upper}}(j))^2}$$

$$\times \left((2 + x^{*T}\Sigma_X^{-1}x^* \Lambda_{\text{lower}}(j))\sigma^2 - \Lambda_{\text{ratio}}^{-1}(j)e^{*2} \right)$$

$$\leq \nu((x^*, y^*); U_{q,\gamma}, P_{X,Y}, m) + o\left(\frac{1}{m}\right)$$

$$\leq h(\gamma) + \frac{1}{m} \sum_{j=q-1}^{m-1} \frac{x^{*T}\Sigma_X^{-1}x^* \Lambda_{\text{upper}}^2(j)}{(1 + x^{*T}\Sigma_X^{-1}x^* \Lambda_{\text{lower}}(j))^2}$$

$$\times \left((2 + x^{*T}\Sigma_X^{-1}x^* \Lambda_{\text{upper}}(j))\sigma^2 - \Lambda_{\text{ratio}}(j)e^{*2} \right),$$

where the function h is defined in Proposition 1 and

$$\Lambda_{\text{ratio}}(j) = \left(\frac{1 + x^{*T}\Sigma_X^{-1}x^*\Lambda_{\text{lower}}(j)}{1 + x^{*T}\Sigma_X^{-1}x^*\Lambda_{\text{upper}}(j)} \right)^2 ,$$

$\Lambda_{\text{lower}}(j)$ *and* $\Lambda_{\text{upper}}(j)$ *are two explicit constants that scale* $O(1/j)$ *and depend only on* γ *and the sub-Gaussian distribution. The explicit expression for* $\Lambda_{\text{lower}}(j)$ *and* $\Lambda_{\text{upper}}(j)$ *are provided in the Supplementary Material.*

Theorem 3 provides upper and lower bounds for DShapley when inputs are sub-Gaussian. As Theorem 2, the main component of the bounds consists of the Mahalanobis distance $x^{*T}\Sigma_X^{-1}x^*$ and the squared error e^{*2}. Hence, data points with the same Mahalanobis distance lead to having the same bounds if the error levels are the same. Although the new bounds in Theorem 3 are not the exact form of DShapley, they are analytically expressed, and can be efficiently computed without Monte Carlo sampling.

The two assumptions in Theorem 3 Compared to Proposition 1, we additionally assume the boundness of $\mathcal{Y}$ and the sub-Gaussian distribution on inputs in Theorem 3. The former implies the boundness of the marginal contribution $\Delta((x^*, y^*); U_{q,\gamma}, S)$ for all $S \subseteq \mathcal{X} \times \mathcal{Y}$, and the latter ensures that eigenvalues of $A_{S,\gamma}^{-1}$ are in the closed interval $[\Lambda_{\text{lower}}(j), \Lambda_{\text{upper}}(j)]$ with high probability. Combining these two ingredients, we obtain the bounds for DShapley as a function of $\Lambda_{\text{lower}}(j)$ and $\Lambda_{\text{upper}}(j)$.

3.4 Application to binary classification

We now study an efficient DShapley estimation method for binary classification datasets. Our approach is to transform binary classification data and apply Theorem 3. To be more precise, let (X, Y) be a pair of input and output random variables and assume $\mathbb{E}[Y \mid X] = \pi = \text{logit}^{-1}(X^T\beta)$. Here, $\text{logit}(\pi) := \pi/(1 - \pi)$ for $\pi \in (0, 1)$. We define the working dependent variable Z and its corresponding weight w as

$$Z = \eta + (Y - \pi)\frac{\partial\eta}{\partial\pi} \text{ and } w = \pi(1 - \pi), \quad (5)$$

respectively, where $\eta = X^T\beta$. Note that $\partial\eta/\partial\pi = w^{-1}$. We propose to consider DShapley with respect to the transformed random variables $(\tilde{X}, \tilde{Z}) := (w^{1/2}X, w^{1/2}Z)$ instead of (X, Y). In the following corollary, we provide a lower bound of DShapley in binary classification. An upper bound and detailed notations are provided in the Supplementary Material.

Corollary 4 (DShapley in binary classification). *Assume* $\mathbb{E}[Y \mid X] = \text{logit}^{-1}(X^T\beta)$ *and* X *are sub-Gaussian in* $\mathbb{R}^p$ *with* $\mathbb{E}(XX^T) = \Sigma_X$. *For a point*

(x^*, y^*), *let* $\pi^* = \text{logit}^{-1}(x^{*T}\beta)$, $w^* = \pi^*(1 - \pi^*)$, *and* $z^* = x^{*T}\beta + (y^* - \pi^*)/w^*$. *Then, for any* $q \geq p + 3$ *and some fixed constant* C_{lin}, *DShapley of a point* $\left((w^*)^{1/2}x^*, (w^*)^{1/2}z^*\right)$ *has a lower bound given by*

$$\nu\left(\left((w^*)^{1/2}x^*, (w^*)^{1/2}z^*\right); U_{q,0}, P_{\tilde{X},\tilde{Z}}, m\right)$$

$$\geq \frac{1}{m} \sum_{j=q-1}^{m-1} \frac{w^* x^{*T}\tilde{\Sigma}_X^{-1}x^*\tilde{\Lambda}_{\text{lower}}^2(j)}{(1 + w^* x^{*T}\tilde{\Sigma}_X^{-1}x^*\tilde{\Lambda}_{\text{upper}}(j))^2}$$

$$\times \left((2 + w^* x^{*T}\tilde{\Sigma}_X^{-1}x^*\tilde{\Lambda}_{\text{lower}}(j)) - \tilde{\Lambda}_{\text{ratio}}^{-1}(j)e_b^{*2}\right)$$

$$+ o\left(\frac{1}{m}\right),$$

where $e_b^{*2} := (w^*)^{-1}(y^* - \pi^*)^2$ *and the function h is defined in Proposition 1.*

Although a typical choice of the utility function in classification is accuracy, using $U_{q,0}$ with the transformation in Corollary 4 provides sensible data values. This is because our approach can be viewed as using the iteratively re-weighted least squares (IRLS) algorithm (Green, 1984), a classic algorithm for finding the maximum likelihood estimator (MLE) in generalized linear models. To be more specific, for a set of random samples $\{(X_i, Y_i)\}_{i=1}^B$ from $P_{X,Y}$ and their working dependent variables and its corresponding weights $\{(Z_i, w_i)\}_{i=1}^B$ based on (5), the IRLS estimator is defined as

$$\hat{\beta}_{\text{IRLS}} = (\mathbb{X}^T\mathbb{W}\mathbb{X})^{-1}\mathbb{X}^T\mathbb{W}\mathbb{Z}, \quad (6)$$

where $\mathbb{X}$ is a matrix whose i-th row is X_i^T, $\mathbb{W}$ is a diagonal matrix whose i-th element is w_i, and similarly $\mathbb{Z}$ is a vector whose i-th element is Z_i. Note that the estimator (6) is the least squares estimator with the transformation in Corollary 4. Hence, the DShapley captures the contribution to finding the MLE in binary classification.

In practice, we do not know π nor $\mathbb{W}$. To address this issue, we first use the original IRLS algorithm; we iteratively compute (5) and (6) until $\hat{\beta}_{\text{IRLS}}$ converges. After convergence, we apply Corollary 4. A simple version of this process is described in Alg. 2. A detailed version is provided in the Supplementary Material.

4 Distributional Shapley values for non-parametric density estimation

In this section, we study DShapley for non-parametric density estimation problems. We let Z be a random variable defined on $\mathcal{Z} \subseteq \mathbb{R}^d$ as in Sec. 2 and let $p(z)$ be the underlying probability density function. We consider the kernel density estimator (KDE), a fundamental non-parametric density estimator in statistics

---

Algorithm 2 DShapley for binary classification.

Require: A datum to be valued (x^*, y^*). A set of
  random samples $\{(X_i, Y_i)\}_{i=1}^B$ from $P_{X,Y}$.
  procedure
    while until a convergent condition is met **do**
        $\pi_i \leftarrow \text{logit}^{-1}(X_i^T \hat{\beta}_{\text{IRLS}})$
        Update w_i and Z_i based on Equation (5) and
  set $\mathbb{W}$ and $\mathbb{Z}$
        $\hat{\beta}_{\text{IRLS}} \leftarrow (\mathbb{X}^T \mathbb{W} \mathbb{X})^{-1} \mathbb{X}^T \mathbb{W} \mathbb{Z}$
    end while
    $\pi^* \leftarrow \text{logit}^{-1}(x^{*T} \hat{\beta}_{\text{IRLS}})$
    $z^* \leftarrow x^{*T} \hat{\beta}_{\text{IRLS}} + (y^* - \pi^*)/(\pi^*(1 - \pi^*))$
    $w^* \leftarrow \pi^*(1 - \pi^*)$
    Compute a lower bound of DShapley of
  $\left((w^*)^{1/2} x^*, (w^*)^{1/2} z^*\right)$.
  end procedure

(Rosenblatt, 1956; Parzen, 1962). For a kernel function[1]
$k : \mathcal{Z} \to \mathbb{R}$, the KDE based on a dataset $S \subseteq \mathcal{Z}$ is de-
noted by $\hat{p}_{S,k}(z) = \frac{1}{|S|} \sum_{z_i \in S} k(z - z_i)$. By convention,
we assume that a kernel is bounded and parameter-
ized by a bandwidth $h > 0$, *i.e.*, $k_h(\cdot) := h^{-d} k(\cdot/h)$
for a kernel k. For notational convenience, we sup-
press the bandwidth notation and use k instead of
k_h. For a constant $C_{\text{den}} > 0$, and a density esti-
mator $\hat{p}$, we define a utility function as $U(S, \hat{p}) =
(C_{\text{den}} - \int (p(z) - \hat{p}(z))^2 dz) \mathbb{1}(|S| \geq 1)$. When the KDE
is used, we set $U_k(S) := U(S, \hat{p}_{S,k})$. As before, chang-
ing the constant C_{den} simply shifts the value of all the
points by the same constant; therefore we just set C_{den}
to simplify expressions of DShapley.

Before going to the analysis, we define DShapley of
a set, a natural extension of DShapley of a point, by
regarding a set as a point. More precisely, given a
utility function U, a data distribution P_Z, and some
$m \in \mathbb{N}$, we define DShapley of a set as follows.

$$\nu(S^*; U, P_Z, m) := \mathbb{E}_{j \sim [m]} \mathbb{E}_{S \sim P_Z^{j-1}} [U(S \cup S^*) - U(S)].$$

Similar to DShapley for a point, DShapley for a set
describes the expected value of marginal contributions
of set S^* over random datasets S. With this notion,
we present DShapley for the KDE in the following
theorem. To begin, let $A(n, m) := \frac{1}{m} \sum_{j=1}^m \frac{n^2}{(j+n-1)^2}$
and $B(n, m) := \frac{1}{m} \sum_{j=2}^m \frac{2n(j-1)}{(j+n-1)^2}$.

Theorem 5 (DShapley for non-parametric density
estimation). *Let $S^* \subseteq \mathcal{Z}$ be a set to be valued such
that $|S^*| = n$. Then, for some fixed constant C_{den},
DShapley of S^* with the KDE is given by*

$\nu(S^*; U_k, P_Z, m)$

---

[1] For a non-negative function $k : \mathcal{Z} \to \mathbb{R}$, we say k is a
kernel if $\int k(z)dz = 1$ and $k(z) = k(-z)$ for all $z \in \mathcal{Z}$.

$$= -A(n, m) \int (p(z) - \hat{p}_{S^*,k}(z))^2 dz + B(n, m)g(S^*),$$

where $g(S^*) := \int \hat{p}_{S^*,k}(z)(p(z) - \mathbb{E}[k(z - Z)])dz$.

The term $g(S^*)$ Suppose $p(z)$ is twice continuously
differentiable, and for all $i \in [d]$, a kernel satisfies
$\int z_{(i)}^2 k(z)dz < \infty$ and $\int |z_{(i)}|^3 k(z)dz < \infty$, where $z =
(z_{(1)}, \ldots, z_{(d)}) \in \mathbb{R}^d$. Then, the bias $(p(z) - \mathbb{E}[k(z - Z)])$
of the KDE is $O(h^2)$ and thus $g(S^*) = O(h^2)$ (Ghosh,
2018, Equation (1.131)). Many useful kernels such
as the Gaussian kernel or any continuous kernel with
bounded support satisfy the conditions.

Theorem 5 shows the exact form of DShapley of a set
S^*. As discussed above, under the mild conditions, the
second term is $O(h^2)$, so we focus on the first term.
The first term is the negative constant $-A(n, m)$ times
to the integrated squared error (ISE) of $\hat{p}_{S^*,k}$. That
means, DShapley for a set S^* increases as ISE decreases,
and vice versa. Note that the ISE could be interpreted
as performance of S^*.

As for the estimation of DShapley, we use the Monte-
Carlo approximation method based on Theorem 5.
For $m, B \in \mathbb{N}$, sets of random samples $\{\tilde{z}_1, \ldots, \tilde{z}_B\}$
and $\{\tilde{z}_1^*, \ldots, \tilde{z}_B^*\}$ from P_Z and $\hat{p}_{S^*,k}$, respectively, the
DShapley estimator $\hat{\nu}(S^*; U_k, P_Z, m)$ is given by

$$\frac{A(|S^*|, m)}{B} \sum_{i=1}^B \left(\hat{p}_{S^*,k}(\tilde{z}_i^*) - 2\hat{p}_{S^*,k}(\tilde{z}_i)\right)$$

$$+ \frac{B(|S^*|, m)}{B} \sum_{i=1}^B \left(\hat{p}_{S^*,k}(\tilde{z}_i) - k(\tilde{z}_i^* - \tilde{z}_i)\right). \quad (7)$$

We provide more details in the Supplementary Material.
In the following examples, we provide more insights on
DShapley with the uniform kernel. Proofs of Examples
1 and 2 are available in the Supplementary Material.

Example 1 (A set with two elements). *Suppose $S^* =
\{z_1^*, z_2^*\}$, $p(z) = 1$ for all $z \in [0, 1]$ and $k(z - z_i) =
\frac{1}{h} \mathbb{1}(|\frac{z - z_i}{h}| \leq \frac{1}{2})$. We set a bandwidth such that $h \leq
2 \min\{z_1^*, z_2^*, (1 - z_1^*), (1 - z_2^*)\}$. Then, we have a closed-
form expression for DShapley as follows.*

$\nu(S^*; U, P_Z, m)$

$$= \begin{cases} A(2, m)\left(1 - \frac{1}{2h}\right) + C_{\text{set}} & \text{if } \Delta \geq h, \\ A(2, m)\left(1 - \frac{1}{h} + \frac{\Delta}{2h^2}\right) + C_{\text{set}} & \text{if } \Delta < h, \end{cases}$$

*where $\Delta := |z_1^* - z_2^*|$ and C_{set} is some explicit con-
stant independent of S^*. DShapley for a set satisfying
$\Delta < h$ is less than the value of a set with $\Delta \geq h$. In
other words, if the two data points are farther than h,
DShapley gets larger.*

Example 2 (Synergy of two elements). *We suppose
the same setting with Example 1 and now investigate*
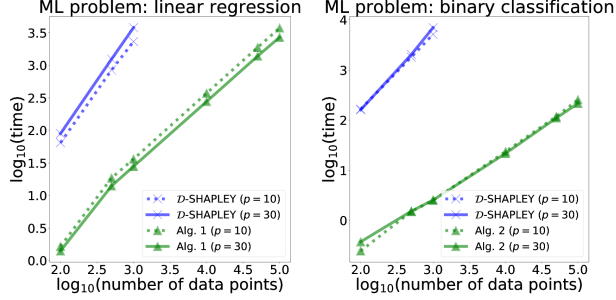
Yongchan Kwon, Manuel A. Rivas, James Zou

Figure 1: Computation time (in seconds) as a function of number of data points to be valued in logarithmic scale. We compare $\mathcal{D}$-SHAPLEY (blue) with the proposed algorithms (green) when the input dimension p is 10 (dashed) or 30 (solid). Alg. 1 is used for regression and Alg. 2 for classification.

the case where DShapley of S^* is greater than the sum of two DShapleys of the point, i.e.,

$$\nu(\{z_1^*, z_2^*\}; U, P_Z, m) \geq \sum_{z \in \{z_1^*, z_2^*\}} \nu(z; U, P_Z, m). \quad (8)$$

We say there is a synergy of z_1^* and z_2^* when the inequality (8) holds. Although a similar analysis used in Example 1 gives $\nu(z_1^*; U, P_Z, m)$ a closed-form expression, it is difficult to know when the inequality (8) holds analytically. With empirical experiments, we show that synergy happens when Δ is bigger than some threshold, i.e., when the two points are not too close.

5 Numerical experiments

We now demonstrate the practical efficacy of the DShapley using real and synthetic datasets. As for the proposed methods, we use Alg. 1, Alg. 2, and Equation (7) for linear regression, binary classification, and non-parametric density estimation problems, respectively. To empirically show the general applicability of the proposed methods, we include complex non-linear models such as convolutional neural networks (CNNs) on our image datasets. Following the common procedure in prior works, we treat the early layers of an off-the-shelf pre-trained network as fixed feature extractors and apply Shapley to the last layer (Ghorbani et al. (2020); Koh and Liang (2017)). Detailed information about datasets and experiment settings are provided in the Supplementary Material. Our implementation codes are available at `https://github.com/ykwon0407/fast_dist_shapley`.

Comparison of the computational time We compare the computational time of $\mathcal{D}$-SHAPLEY by Ghorbani et al. (2020) with the proposed methods in several ML problems. As we mentioned in Sec. 3.2,

Table 1: Computation time (in seconds) of $\mathcal{D}$-SHAPLEY and the proposed algorithms in various ML problems and datasets. The number of data to be valued is fixed to 200 for all datasets.

ML problem (Proposed method)	Dataset	$\mathcal{D}$-SHAPLEY	Proposed
Linear regression (Alg. 1)	Gaussian-R	229.2	6.9
	abalone	226.3	5.2
	airfoil	280.7	4.6
	whitewine	275.4	5.1
Binary classification (Alg. 2)	Gaussian-C	470.4	0.7
	skin-nonskin	788.5	1.7
	CIFAR10	550.7	3.0
	MNIST	536.7	5.3
Density estimation (Equation (7))	diabetes	3307.8	0.6
	australian	5219.8	0.3
	Fashion-MNIST	281.7	23.7
	CIFAR10	338.6	28.4

the existing algorithm requires the utility evaluation, and thus it is anticipated to have much heavier computational costs than the proposed algorithm. All the computation time results in this section are measured with the single Intel®Xeon®E5-2640v4 CPU processor and are an average based on 50 repetitions.

Figure 1 shows the computational time of state-of-the-art $\mathcal{D}$-SHAPLEY and the proposed methods in various the number of data to be valued, denoted by m, and the dimension of input data, denoted by p. We consider linear regression and binary classification problems and use the synthetic Gaussian datasets. For both ML problems, the proposed algorithm is several orders of magnitude faster than $\mathcal{D}$-SHAPLEY. In particular, in case of classification, while $\mathcal{D}$-SHAPLEY requires 7015.7 seconds, Alg. 2 takes 2.6 seconds, which is 2750 times faster, when $(m, p) = (1000, 30)$. Our proposed algorithms is scalable to compute the distribution Shapley values of hundreds of thousands of data points in thousands of dimensions. With $(m, p) = (5 \times 10^5, 10^3)$, Alg. 1 (Alg. 2) takes around 5.3 hours (*resp.* 30 minutes) to compute the DShapley values for all half million data points for linear regression (*resp.* binary classification). This can be further improved with parallel computing and GPU processors. The computation time for the binary classification problem is much smaller because we use a computationally cheap analytic lower bound. The sharpness of this lower bound is examined in the point addition experiment below.

Lastly, Table 1 shows computational time in various tasks and real and synthetic datasets. We here fix the number of data to be valued as 200. This further demonstrates the computational efficiency of the proposed algorithms across all datasets.
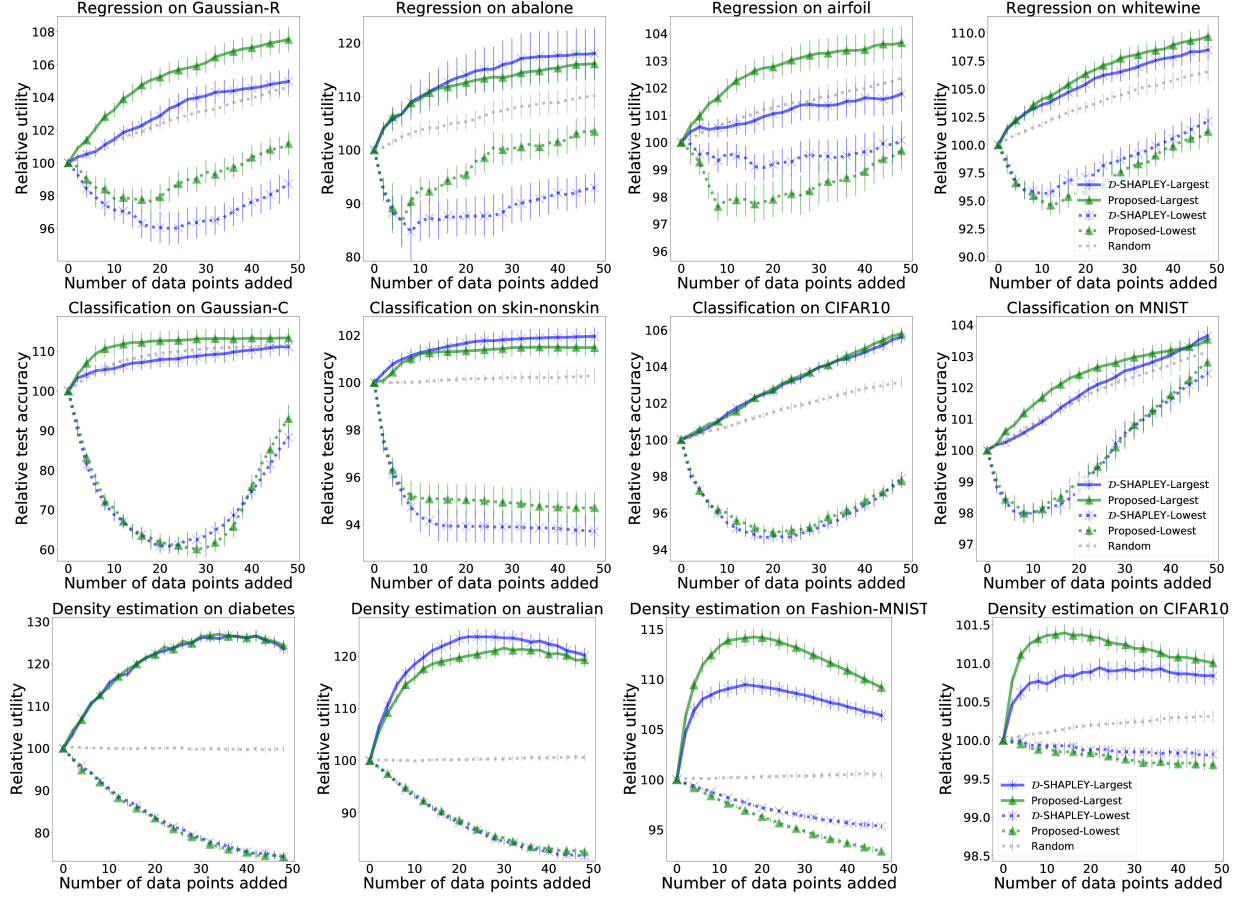
Figure 2: Relative utility and its standard error bar (in %) as a function of the number of data added in (top) linear regression, (middle) binary classification, and (bottom) non-parametric density estimation settings. We examine the state-of-the-art $\mathcal{D}$-SHAPLEY (blue), random order (gray), and our proposed algorithms (green). The solid and dashed curves correspond to adding points with the largest and smallest values first, respectively. The results are based on 50 repetitions.

Point addition experiment We demonstrate the empirical effectiveness of our DShapley approach by running point addition experiments, proposed by Ghorbani et al. (2020). Given a model and a dataset to be valued, we recursively add points given order (e.g. from largest to lowest values), retrain the model with the remained dataset, and observe how the utility changes on the held-out test dataset. We compare the three methods: (i) the random order, (ii) the order based on $\mathcal{D}$-SHAPLEY, and (iii) the order based on the proposed methods. For $\mathcal{D}$-SHAPLEY and the proposed algorithms, we consider the two different types of orders. One is from largest to lowest, denoted by 'Largest', and the other is from lowest to largest, denoted by 'Lowest'. In the case of the 'Largest' order, they are expected to capture points that help improve performance and show a steeper performance boost than the random order. Similarly, in the case of the 'Lowest' order, DShapley is expected to capture outliers first and cause perfor-

mance degradation. As before, we use Alg. 1, Alg. 2, and Equation (7) for linear regression, binary classification, and non-parametric density estimation problems, respectively.

Figure 2 shows point addition experiments in linear regression, binary classification, and non-parametric density estimation problems. As we anticipated, the proposed methods and $\mathcal{D}$-SHAPLEY show reasonable curves; adding data with larger DShapley leads to a greater performance increase than random addition. Also, adding data with lower DShapley causes performance degradation. Moreover, the proposed methods perform similar to or sometimes better than $\mathcal{D}$-SHAPLEY. This phenomenon is because $\mathcal{D}$-SHAPLEY repeatedly evaluates the utility function on a random set, and the utility is unstable when the size of the set is small and it affects instability of DShapley estimation. In contrast, the proposed methods can avoid such instability and provide reasonable values.

6 Concluding remarks

In this work, we derive the first computationally tractable expressions for DShapley for the linear regression, binary classification, and non-parametric density estimation problems. The proposed forms provide new mathematical insights, and lead to efficient algorithms which we demonstrated on large datasets (e.g. 10^5 data points in $\mathbb{R}^{1000}$) and models such as CNN. We validate our results on several commonly used datasets.

Acknowledgements

References

Agarwal, A., Dahleh, M., and Sarkar, T. (2019). A marketplace for data: An algorithmic solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 701–726.

Aumann, R. J. and Shapley, L. S. (2015). *Values of non-atomic games*. Princeton University Press.

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

Cohen, M. B., Musco, C., and Musco, C. (2017). Input sparsity time low-rank approximation via ridge leverage score sampling. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1758–1777. SIAM.

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

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

Dubey, P. (1982). The shapley value as aircraft landing fees–revisited. *Management Science*, 28(8):869–874.

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

Fernandez, R. C., Subramaniam, P., and Franklin, M. J. (2020). Data market platforms: Trading data assets to solve data problems. *arXiv preprint arXiv:2002.01047*.

Ghorbani, A., Kim, M. P., and Zou, J. (2020). A distributional framework for data valuation. *arXiv preprint arXiv:2002.12334*.

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

Ghorbani, A. and Zou, J. (2020). Neuron shapley: Discovering the responsible neurons. *arXiv preprint arXiv:2002.09815*.

Ghosh, S. (2018). *Kernel smoothing: Principles, methods and applications*. John Wiley & Sons.

Grabisch, M. and Roubens, M. (1999). An axiomatic approach to the concept of interaction among players in cooperative games. *International Journal of game theory*, 28(4):547–565.

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2):149–170.

Gul, F. (1989). Bargaining foundations of shapley value. *Econometrica: Journal of the Econometric Society*, pages 81–95.

Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393.

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

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

Koh, P. W. and Liang, P. (2017). Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1885–1894. JMLR. org.

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

McCurdy, S. (2018). Ridge regression and provable deterministic ridge leverage score sampling. In *Advances in Neural Information Processing Systems*, pages 2463–2472.

Moulin, H. (1992). An application of the shapley value to fair division with money. *Econometrica: Journal of the Econometric Society*, pages 1331–1349.

Parzen, E. (1962). On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076.

Posner, E. A. and Weyl, E. G. (2018). *Radical markets: Uprooting capitalism and democracy for a just society.* Princeton University Press.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pages 832–837.

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

Sundararajan, M. and Najmi, A. (2019). The many shapley values for model explanation. *arXiv preprint arXiv:1908.08474*.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.

Wadhwa, T. (2020). Economic impact and feasibility of data dividends.

Warner, M. R. (2019). S.1951 - designing accounting safeguards to help broaden oversight and regulations on data.

Yoon, J., Arik, S. O., and Pfister, T. (2019). Data valuation using reinforcement learning. *arXiv preprint arXiv:1909.11671*.

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