A Distributional Framework for Data Valuation

Amirata Ghorbani*
amiratag@stanford.edu

Michael P. Kim*†
mpk@cs.stanford.edu

James Zou
jamesz@stanford.edu

Abstract

Shapley value is a classic notion from game theory, historically used to quantify the contributions of individuals within groups, and more recently applied to assign values to data points when training machine learning models. Despite its foundational role, a key limitation of the data Shapley framework is that it only provides valuations for points within a *fixed data set*. It does not account for statistical aspects of the data and does not give a way to reason about points outside the data set.

To address these limitations, we propose a novel framework – *distributional Shapley* – where the value of a point is defined in the context of an underlying data distribution. We prove that distributional Shapley has several desirable statistical properties; for example, the values are stable under perturbations to the data points themselves and to the underlying data distribution. We leverage these properties to develop a new algorithm for estimating values from data, which comes with formal guarantees and runs two orders of magnitude faster than state-of-the-art algorithms for computing the (non-distributional) data Shapley values. We apply distributional Shapley to diverse data sets and demonstrate its utility in a data market setting.

1 Introduction

As data becomes an essential driver of innovation and service, how to quantify the value of data is an increasingly important topic of inquiry with policy, economic, and machine learning (ML) implications. In the policy arena, recent proposals, such as the Dashboard Act in the U.S. Senate, stipulate that large companies quantify the value of data they collect. In the global economy, the business model of many companies involves buying and selling data. For ML engineering, it is often beneficial to know which type of training data is most valuable and, hence, most deserving of resources towards collection and annotation. As such, a principled framework for data valuation would be tremendously useful in all of these domains.

Recent works initiated a formal study of data valuation in ML [GZ19, JDW$^+$19b]. In a typical setting, a data set $B = \{z_i\}$ is used to train a ML model, which achieves certain performance, say classification accuracy 0.9. The data valuation problem is to assign credit amongst the training set, so that each point gets an "equitable" share for its contribution towards achieving the 0.9 accuracy. Most works have focused on leveraging *Shapley value* as the metric to quantify the contribution of

[*] Equal Contribution. Authors listed alphabetically.

[†] Supported in part by CISPA Center for Information Security and NSF Award IIS-1908774

individual z_i. The focus on Shapley value is in large part due to the fact that Shapley uniquely satisfies basic properties for equitable credit allocation [Sha53]. Empirical experiments also show that data Shapley is very effective – more so than leave-one-out scores – at identifying points whose addition or removal substantially impacts learning [GAZ17, GZ19].

At a high-level, prior works on data Shapley require three ingredients: (1) a fixed training data set of m points; (2) a learning algorithm; and (3) a performance metric that measures the overall value of a trained model. The goal of this work is to significantly reduce the dependency on the first ingredient. While convenient, formulating the value based on a *fixed data set* disregards crucial statistical considerations and, thus, poses significant practical limitations.

In standard settings, we imagine that data is sampled from a distribution $\mathcal{D}$; measuring the Shapley value with respect to a fixed data set ignores this underlying distribution. It also means that the value of a data point computed within one data set may not make sense when the point is transferred to a new data set. If we actually want to buy and sell data, then it is important that the value of a given data point represents some intrinsic quality of the datum within the distribution. For example, a data seller might determine that z has high value based on their data set B_s and sell z to a buyer at a high price. Even if the buyer's data set B_b is drawn from a similar distribution as B_s, the existing data Shapley framework provides no guarantee of consistency between the value of z computed within B_s and within B_b. This inconsistency may be especially pronounced in the case when the buyer has significantly less data than the seller.

Our contributions

Conceptual. Extending prior works on data Shapley, we formulate and develop a notion of *distributional Shapley value* in Section 2. We define the distributional variant in terms of the original data Shapley: the distributional Shapley value is taken to be the expected data Shapley value, where the data set is drawn i.i.d. from the underlying data distribution. Reformulating this notion of value as a statistical quantity allows us to prove that the notion is stable with respect to perturbations to the inputs as well as the underlying data distribution. Further, we show a mathematical identity that gives an equivalent definition of distributional Shapley as an expected marginal performance increase by adding the point, suggesting an unbiased estimator.

Algorithmic. In Section 3, we develop this estimator into a novel sampling-based algorithm, $\mathcal{D}$-Shapley. In contrast to prior estimation heuristics, $\mathcal{D}$-Shapley comes with strong formal approximation guarantees. Leveraging the stability properties of distributional Shapley value and the simple nature of our algorithm, we develop theoretically-principled optimizations to $\mathcal{D}$-Shapley. In our experiments across diverse tasks, the optimizations lead to order-of-magnitude reductions in computational costs while maintaining the quality of estimations.

Empirical. Finally, in Section 4, we present a data pricing case study that demonstrates the consistency of values produced by $\mathcal{D}$-Shapley. In particular, we show that a data broker can list distributional Shapley values as "prices," which a collection of buyers all agree are fair (i.e. the data gives each buyer as much value as the seller claims). In all, our results demonstrate that the distributional Shapley framework represents a significant step towards the practical viability of the Shapley-based approaches to data valuation.

Related works

Shapley value, introduced in [Sha53], has been studied extensively in the literature on cooperative games and economics [SR$^+$88], and has traditionally been used in the valuation of private information and data markets [KPR01, ADS19].

Our work is most directly related to recent works that apply Shapley value to the data valuation problem. [GZ19] developed the notion of "Data Shapley" and provided algorithms to efficiently estimate values. Specifically, leveraging the permutation-based characterization of Shapley value, they developed a "truncated Monte Carlo" sampling scheme (referred to as TMC-Shapley), demonstrating empirical effectiveness across various ML tasks. [JDW$^+$19b] introduce several additional approximation methods for efficient computation of Shapley values for training data; subsequently, [JDW$^+$19a] provided an algorithm for exact computation of Shapley values for the specific case of nearest neighbor classifiers.

Beyond data valuation, the Shapley framework has been used in a variety of ML applications, e.g. as a measure of feature importance [CDR07, K$^+$10, DSZ16, LL17, CSWJ18]. The idea of a distributional Shapley value bears resemblance to the Aumann-Shapley value [AS74], a measure-theoretic variant of the Shapley that quantifies the value of individuals within a continuous "infinite game." Our distributional Shapley value focuses on the tangible setting of finite data sets drawn from a (possibly continuous) distribution.

2 Distributional Data Valuation

Preliminaries

Let $\mathcal{D}$ denote a data distribution supported on a universe $\mathcal{Z}$. For supervised learning problems, we often think of $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ where $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y}$ is the output, which can be discrete or continuous. For $m \in \mathbb{N}$, let $S \sim \mathcal{D}^m$ a collection of k data points sampled i.i.d. from $\mathcal{D}$. Throughout, we use the shorthand $[m] = \{1, \ldots, m\}$ and let $k \sim [m]$ denote a uniform random sample from $[m]$.

We denote by $U : \mathcal{Z}^* \to [0, 1]$ a potential function[1] or performance metric, where for any $S \subseteq \mathcal{Z}$, $U(S)$ represents abstractly the value of the subset. While our analysis applies broadly, in our context, we think of U as capturing both the *learning algorithm* and the *evaluation metric*. For instance, in the context of training a logistic regression model, we might think of $U(S)$ as returning the population accuracy of the empirical risk minimizer when S is the training set.

2.1 Distributional Shapley Value

Our starting point is the data Shapley value, proposed in [GZ19, JDW$^+$19b] as a way to valuate training data equitably.

Definition 2.1 (Data Shapley Value). *Given a potential function U and data set $B \subseteq \mathcal{Z}$ where*

[1] We use $\mathcal{Z}^* = \bigcup_{n \in \mathbb{N}} \mathcal{Z}^n$ to indicates any finite Cartesian product of $\mathcal{Z}$ with itself; thus, U is well-defined on the any natural number of inputs from $\mathcal{Z}$.

$|B| = m$, the data Shapley value of a point $z \in B$ is defined as

$$\phi(z; U, B) \triangleq \frac{1}{m} \sum_{k=1}^{m} \frac{1}{\binom{m-1}{k-1}} \sum_{\substack{S \subseteq B \setminus \{z\}: \\ |S| = k-1}} \left(U(S \cup \{z\}) - U(S) \right).$$

In words, the data Shapley value of a point $z \in B$ is a weighted empirical average over subsets $S \subseteq B$ of the marginal potential contribution of z to each S; the weighting is such that each possible cardinality $|S| = k \in \{0, \ldots, m-1\}$ is weighted equally. The data Shapley value satisfies a number of desirable properties; indeed, it is the unique valuation function that satisfies the Shapley axioms[2]. Note that as the data set size grows, the absolute magnitude of individual data points' values typically scales inversely.

While data Shapley value is a natural solution concept for data valuation, its formulation leads to several limitations. In particular, the values may be very sensitive to the exact choice of B; given another $B' \neq B$ where $z \in B \cap B'$, the value $\phi(z; U, B)$ might be quite different from $\phi(z; U, B')$. At the extreme, if a new point $z' \notin B$ is added to B, then in principle, we would have to rerun the procedure to compute the data Shapley values for all points in $B \cup \{z'\}$.

In settings where our data are drawn from an underlying distribution $\mathcal{D}$, a natural extension to the data Shapley approach would parameterize the valuation function by $\mathcal{D}$, rather than the specific draw of the data set. Such a distributional Shapley value should be more stable, by removing the explicit dependence on the draw of the training data set.

Definition 2.2 (Distributional Shapley Value)**.** *Given a potential function $U : \mathcal{Z}^* \to [0, 1]$, a distribution $\mathcal{D}$ supported on $\mathcal{Z}$, and some $m \in \mathbb{N}$, the distributional Shapley value of a point $z \in \mathcal{Z}$ is the expected data Shapley value over data sets of size m containing x.*

$$\nu(z; U, \mathcal{D}, m) \triangleq \mathop{\mathbf{E}}_{B \sim \mathcal{D}^{m-1}} \left[\phi\left(z; U, B \cup \{z\}\right) \right]$$

In other words, we can think of the data Shapley value as a random variable that depends on the specific draw of data from $\mathcal{D}$. Taking the distributional Shapley value $\nu(z; U, \mathcal{D}, m)$ to be the expectation of this random variable eliminates instability caused by the variance of $\phi(z; U, B)$. While distributional Shapley is simple to state based on the original Shapley value, to the best of our knowledge, the concept is novel to this work.

We note that, while more stable, the distributional Shapley value inherits many of the desirable properties of Shapley, including the Shapley axioms and an expected efficiency property; we cover these in Appendix B. Importantly, distributional Shapley also has a clean characterization as the expected gain in potential by adding $z \in \mathcal{Z}$ to a random data set (of random size).

Theorem 2.3. *Fixing U and $\mathcal{D}$, for all $z \in \mathcal{Z}$ and $m \in \mathbb{N}$,*

$$\nu(z; U, \mathcal{D}, m) = \mathop{\mathbf{E}}_{\substack{k \sim [m] \\ S \sim \mathcal{D}^{k-1}}} \left[U(S \cup \{z\}) - U(S) \right]$$

That is, the distributional Shapley value of a point is its expected marginal contribution in U to a set of i.i.d. samples from $\mathcal{D}$ of uniform random cardinality.

[2] For completeness, the axioms – symmetry, null player, additivity, and efficiency – are reviewed in Appendix A.

Proof. The identity holds as a consequence of the definition of data Shapley value and linearity of expectation.

$$\begin{aligned}
\nu(z; U, \mathcal{D}, m) &= \mathop{\mathbf{E}}_{D \sim \mathcal{D}^{m-1}} \left[\phi(z; U, D \cup \{z\})\right] \\
&= \mathop{\mathbf{E}}_{D \sim \mathcal{D}^{m-1}} \left[\frac{1}{m} \sum_{k=1}^{m} \frac{1}{\binom{m-1}{k-1}} \sum_{\substack{S \subseteq D: \\ |S| = k-1}} (U(S \cup \{z\}) - U(S)) \right] \\
&= \frac{1}{m} \sum_{k=1}^{m} \frac{1}{\binom{m-1}{k-1}} \mathop{\mathbf{E}}_{D \sim \mathcal{D}^{m-1}} \left[\sum_{\substack{S \subseteq D: \\ |S| = k-1}} (U(S \cup \{z\}) - U(S)) \right] \\
&= \frac{1}{m} \sum_{k=1}^{m} \mathop{\mathbf{E}}_{S \sim \mathcal{D}^{k-1}} \left[U(S \cup \{z\}) - U(S)\right] \qquad (1) \\
&= \mathop{\mathbf{E}}_{\substack{k \sim [m] \\ S \sim \mathcal{D}^{k-1}}} \left[U(S \cup \{z\}) - U(S)\right]
\end{aligned}$$

where (1) follows by the fact that $D \sim \mathcal{D}^{m-1}$ consists of i.i.d. samples, so each $S \subseteq D$ with $|S| = k-1$ is identically distributed according to $\mathcal{D}^{k-1}$. □

Example: mean estimation. Leveraging this characterization, for well-structured problems, it is possible to give analytic expressions for the distributional Shapley values. For instance, consider estimating the mean μ of a distribution $\mathcal{D}$ supported on $\mathbb{R}^d$. For a finite subset $S \subseteq \mathbb{R}^d$, we take a potential $U(S)$ based on the empirical estimator $\hat{\mu}_S$.

$$U_\mu(S) = \mathop{\mathbf{E}}_{s \sim \mathcal{D}} \left[\|s - \mu\|^2\right] - \|\hat{\mu}_S - \mu\|^2$$

Proposition 2.4. *Suppose $\mathcal{D}$ has bounded second moments. Then for $z \in \mathcal{Z}$ and $m \in \mathbb{N}$, $\nu(z; U_\mu, \mathcal{D}, m)$ for mean estimation over $\mathcal{D}$ is given by*

$$\frac{\mathbf{E}_{S \sim \mathcal{D}^m}\left[U(S)\right]}{m} + \frac{C_m}{m} \cdot \left(\mathop{\mathbf{E}}_{s \sim \mathcal{D}} \left[\|s - \mu\|^2\right] - \|z - \mu\|^2 \right)$$

for an explicit constant $C_m = \Theta(1)$ determined by m.

Intuitively, this proposition (proved in Appendix B) highlights some desirable properties of distributional Shapley: the expected value for a random $z \sim \mathcal{D}$ is an uniform share of the potential for a randomly drawn data set $S \sim \mathcal{D}^m$; further, a point has above-average value when it is closer to μ than expected. In general, analytically deriving the distributional Shapley value may not be possible. In Section 3, we show how the characterization of Theorem 2.3 leads to an efficient algorithm for estimating values.

2.2 Stability of distributional Shapley values

Before presenting our algorithm, we discuss stability properties of distributional Shapley, which are interesting in their own right, but also have algorithmic implications. We show that when the potential function U satisfies a natural stability property, the corresponding distributional Shapley value inherits stability under perturbations to the data points and the underlying data distribution. First, we recall a standard notion of deletion stability, often studied in the context of generalization of learning algorithms [BE02].

Definition 2.5 (Deletion Stability)**.** *For potential $U : \mathcal{Z}^* \to [0,1]$ and non-increasing $\beta : \mathbb{N} \to [0,1]$, U is $\beta(k)$-deletion stable if for all $k \in \mathbb{N}$ and $S \in \mathcal{Z}^{k-1}$, for all $z \in \mathcal{Z}$*

$$|U(S \cup \{z\}) - U(S)| \leq \beta(k).$$

We can similarly discuss the idea of replacement stability, where we bound $|U(S \cup \{z\}) - U(S \cup \{z'\})|$; note that by the triangle inequality, $\beta(k)$-deletion stability of U implies $2\beta(k)$-replacement stability. To analyze the properties of distributional Shapley, a natural strengthening of replacement stability will be useful, which we call *Lipschitz stability.* Lipschitz stability is parameterized by a metric d, requires the degree of robustness under replacement of z with z' to scale according to the distance $d(z, z')$.

Definition 2.6 (Lipschitz Stability)**.** *Let $(\mathcal{Z}, d)$ be a metric space. For potential $U : \mathcal{Z}^* \to [0,1]$ and non-increasing $\beta : \mathbb{N} \to [0,1]$, U is $\beta(k)$-Lipschitz stable with respect to d if for all $k \in \mathbb{N}$, $S \in \mathcal{Z}^{k-1}$, and all $z, z' \in \mathcal{Z}$,*

$$\left|U(S \cup \{z\}) - U(S \cup \{z'\})\right| \leq \beta(k) \cdot d(z, z').$$

By taking d to be the trivial metric, where $d(z, z') = 1$ if $z \neq z'$, we see that Lipschitz-stability generalizes the idea of replacement stability; still, there are natural learning algorithms that satisfy Lipschitz stability for nontrivial metrics. As one example, we show that Regularized empirical risk minimization over a Reproducing Kernel Hilbert Space (RKHS) – a prototypical example of a replacement stable learning algorithm – also satisfies this stronger notion of Lipschitz stability. We include a formal statement and proof in Appendix C.

Similar points receive similar values. As discussed, a key limitation with the data Shapley approach for fixed data set B is that we can only ascribe values to $z \in B$. Intuitively, however, we would hope that if two points z and z' are similar according to some appropriate metric, then they would receive similar Shapley values. We confirm this intuition for distributional Shapley values when the potential function U satisfies Lipschitz stability.

Theorem 2.7. *Fix a metric space $(\mathcal{Z}, d)$ and a distribution $\mathcal{D}$ over $\mathcal{Z}$; let $U : \mathcal{Z}^* \to [0,1]$ be $\beta(k)$-Lipschitz stable with respect to d. Then for all $m \in \mathbb{N}$, for all $z, z' \in \mathcal{Z}$,*

$$\left|\nu(z; U, \mathcal{D}, m) - \nu(z'; U, \mathcal{D}, m)\right| \leq \mathop{\mathbf{E}}_{k \sim [m]}[\beta(k)] \cdot d(z, z').$$

Proof. For any data set size $m \in \mathbb{N}$, we expand $\nu(z'; U, \mathcal{D}, m)$ to express it in terms of $\nu(z; U, \mathcal{D}, m)$.

$$\begin{aligned}\nu(z'; U, \mathcal{D}, m) &= \mathop{\mathbf{E}}_{\substack{k \sim [m]\\ S \sim \mathcal{D}^{k-1}}}\left[U(S \cup \{z'\}) - U(S)\right]\\ &= \mathop{\mathbf{E}}_{\substack{k \sim [m]\\ S \sim \mathcal{D}^{k-1}}}\left[U(S \cup \{z\}) - U(S)\right] + \mathop{\mathbf{E}}_{\substack{k \sim [m]\\ S \sim \mathcal{D}^{k-1}}}\left[U(S \cup \{z'\}) - U(S \cup \{z\})\right]\\ &\leq \nu(z; U, \mathcal{D}, m) + \mathop{\mathbf{E}}_{k \sim [m]}[\beta(k)] \cdot d(z, z') \qquad (2)\end{aligned}$$

where (2) follows by the assumption that U is $\beta(k)$-Lipschitz stable and linearity of expectation. □

Theorem 2.7 suggests that in many settings of interest, the distributional Shapley value will be Lipschitz in z. This Lipschitz property also suggests that, given the values of a (sufficiently-diverse) set of points Z, we may be able to infer the values of unseen points $z' \notin Z$ through interpolation. Concretely, in Section 3.2, we leverage this observation to give an order of magnitude speedup over our baseline estimation algorithm.

Similar distributions yield similar value functions. The distributional Shapley value is naturally parameterized by the underlying data distribution $\mathcal{D}$. For two distributions $\mathcal{D}_s$ and $\mathcal{D}_t$, given the value $\nu(z; U, \mathcal{D}_s, m)$, what can we say about the value $\nu(z; U, \mathcal{D}_t, m)$? Intuitively, if $\mathcal{D}_s$ and $\mathcal{D}_t$ are similar under an appropriate metric, we'd expect that the values should not change too much. Indeed, we can formally quantify how the distributional Shapley value is stable under distributional shift under the Wasserstein distance.

For two distributions $\mathcal{D}_s, \mathcal{D}_t$ over $\mathcal{Z}$, let Γ_{st} be the collection of joint distributions over $\mathcal{Z} \times \mathcal{Z}$, whose marginals are $\mathcal{D}_s$ and $\mathcal{D}_t$.[3] Fixing a metric d over $\mathcal{Z}$, the Wasserstein distance is the infimum over all such couplings $\gamma \in \Gamma_{st}$ of the expected distance between $(s, t) \sim \gamma$.

$$W_1(\mathcal{D}_s, \mathcal{D}_t) \triangleq \inf_{\gamma \in \Gamma_{st}} \mathop{\mathbf{E}}_{(s,t)\sim\gamma} [d(s,t)] \tag{3}$$

We formalize the idea that distributional Shapley values are stable under small perturbations to the underlying data distribution as follows.

Theorem 2.8. *Fix a metric space $(\mathcal{Z}, d)$ and let $U : \mathcal{Z}^* \to [0,1]$ be $\beta(k)$-Lipschitz stable with respect to d. Suppose $\mathcal{D}_s$ and $\mathcal{D}_t$ are two distributions over $\mathcal{Z}$. Then, for all $m \in \mathbb{N}$ and all $z \in \mathcal{Z}$,*

$$|\nu(z; U, \mathcal{D}_s, m) - \nu(z; U, \mathcal{D}_t, m)| \le \frac{2}{m} \sum_{k=1}^{m-1} k\beta(k) \cdot W_1(\mathcal{D}_s, \mathcal{D}_t).$$

Proof. For notational convenience, for any $z \in \mathcal{Z}$ and subset $S \subseteq \mathcal{Z}$, we denote $\Delta_z U(S) = U(S \cup \{z\}) - U(S)$. Thus, fixing $z \in \mathcal{Z}$, we can write $\nu(z; U, \mathcal{D}, m)$ as $\mathbf{E}_{k\sim[m]} \mathbf{E}_{S\sim\mathcal{D}^{k-1}} [\Delta_z U(S)]$. We analyze $\mathbf{E}_{S\sim\mathcal{D}^{k-1}} [\Delta_z U(S)]$ for each fixed $k \in \{2, \ldots, m\}$ separately.[4]

Let $\gamma \in \Gamma_{st}$ be some coupling of $\mathcal{D}_s$ and $\mathcal{D}_t$. Then, we can expand the expectation as follows.

$$\begin{aligned}
\mathop{\mathbf{E}}_{S\sim\mathcal{D}_s^{k-1}} [\Delta_z U(S)] &= \mathop{\mathbf{E}}_{S\times T\sim\gamma^{k-1}} [\Delta_z U(S)] && (4) \\
&= \mathop{\mathbf{E}}_{S\times T} [\Delta_z U(S) - \Delta_z U(T)] + \mathop{\mathbf{E}}_{S\times T} [\Delta_z U(T)] && (5) \\
&= \mathop{\mathbf{E}}_{S\times T} [\Delta_z U(S) - \Delta_z U(T)] + \mathop{\mathbf{E}}_{T\sim\mathcal{D}_t^{k-1}} [\Delta_z U(T)] && (6)
\end{aligned}$$

where (4) and (6) follow by the assumption that the marginals of γ are $\mathcal{D}_s$ and $\mathcal{D}_t$; and (5) follows by linearity of expectation.

[3] That is, for all $\gamma \in \Gamma_{st}$, if $(s, t) \sim \gamma$, then $s \sim \mathcal{D}_s$ and $t \sim \mathcal{D}_t$.

[4] Note that for a fixed potential U, $m = 1$ is uninteresting because both sides of the inequality are 0; in particular, $|S|$ is always 0, so the LHS is given by the difference $U(z) - U(z)$.

To bound the first term of (6), we expand the difference between $\Delta_z U(S)$ and $\Delta_z U(T)$ into a telescoping sum of k pairs of terms, where we bound each pair to depend on a single draw $(s_i, t_i) \sim \gamma$. For $S, T \in \mathcal{Z}^k$ and $i \in \{0, \ldots, k\}$, denote by $Z_i = \left(\bigcup_{j=i+1}^{k} s_j\right) \cup \left(\bigcup_{j=1}^{i} t_j\right)$; note that $Z_0 = S$ and $Z_k = T$. Then, we can rewrite $\Delta_z U(S) - \Delta_z U(T)$ as follows.

$$\Delta_z U(S) - \Delta_z U(T) = \sum_{i=1}^{k} \Delta_z U(Z_{i-1}) - \Delta_z U(Z_i)$$

Now suppose U is $\beta(k)$-Lipschitz stable with respect to d; note that this implies $\Delta_z U$ is $2\beta(k)$-Lipschitz stable (because β is non-increasing). Then, we obtain the following bound.

$$\begin{aligned}
\mathop{\mathbf{E}}_{S \times T \sim \gamma^{k-1}} \left[\Delta_z U(S) - \Delta_z U(T)\right] &= \mathop{\mathbf{E}}_{S \times T \sim \gamma^{k-1}} \left[\sum_{i=1}^{k-1} \Delta_z U(Z_{i-1}) - \Delta_z U(Z_i)\right] \\
&= \sum_{i=1}^{k-1} \mathop{\mathbf{E}}_{S,T \sim \gamma^{k-1}} \left[\Delta_z U(Z_{i-1}) - \Delta_z U(Z_i)\right] \\
&= \sum_{i=1}^{k-1} \mathop{\mathbf{E}}_{\substack{s_i, t_i \sim \gamma \\ R \in \mathcal{Z}^{k-2}}} \left[\Delta_z U(R \cup \{s_i\}) - \Delta_z U(R \cup \{t_i\})\right] &(7) \\
&\leq 2\beta(k-1) \cdot \sum_{i=1}^{k-1} \mathop{\mathbf{E}}_{(s_i, t_i) \sim \gamma} [d(s_i, t_i)] &(8) \\
&\leq 2(k-1)\beta(k-1) \cdot \mathop{\mathbf{E}}_{(s,t) \sim \gamma} [d(s,t)] &(9)
\end{aligned}$$

where (7) notes Z_{i-1} and Z_i differ on only the ith data point; (8) follows from the assumption that $\Delta_z U$ is $2\beta(k)$-Lischitz stable and linearity of expectation; and finally (9) follows by the fact that each draw from γ is i.i.d.

Finally, we note that the argument above worked for an arbitrary coupling in Γ_{st}; thus, we can express the difference in values in terms of the infimum over Γ_{st}.

$$\begin{aligned}
&\nu(z; U, \mathcal{D}_s, m) - \nu(z; U, \mathcal{D}_t, m) \\
&\leq \inf_{\gamma \in \Gamma_{st}} \mathop{\mathbf{E}}_{k \sim [m]} \left[\mathop{\mathbf{E}}_{S \times T \sim \gamma^{k-1}} \left[\Delta_U(S) - \Delta_z U(T)\right]\right] \\
&\leq \frac{2}{m} \sum_{k=2}^{m} (k-1)\beta(k-1) \inf_{\gamma \in \Gamma_{st}} \mathop{\mathbf{E}}_{(s,t) \sim \gamma} [d(s,t)] \\
&= \frac{2}{m} \sum_{k=1}^{m-1} k\beta(k) \cdot W_1(\mathcal{D}_s, \mathcal{D}_t)
\end{aligned}$$

where the first summation is taken over $k \in \{2, \ldots, m\}$ as the term associated with $k = 1$ is 0. □

Note that the theorem bounds the difference in values under shifts in distribution holding the potential U fixed. Often in applications, we will take the potential function to depend on the underlying data distribution. For instance, we may take to be a measure of population accuracy, e.g. $U_{\mathcal{D}_s} = 1 - \mathbf{E}_{z\sim\mathcal{D}}\left[\ell_S(z)\right]$, where $\ell_S(z)$ is the loss on a point $z \in \mathcal{Z}$ achieved by a model trained on the data set $S \subseteq \mathcal{Z}$. In the case where we only have access to samples from $\mathcal{D}_s$, we still may want to guarantee that $\nu(z; U_{\mathcal{D}_s}, \mathcal{D}_s, m)$ and $\nu(z; U_{\mathcal{D}_t}, \mathcal{D}_t, m)$ are close. Thankfully, such a result follows by showing that $U_{\mathcal{D}_s}$ is close to $U_{\mathcal{D}_t}$, and another application of the triangle inequality. For instance, when the potential is based on the population loss for a Lipschitz loss function, we can bound the difference in the potentials, again, in terms of the Wasserstein distance.

$$\begin{aligned}
U_{\mathcal{D}_t}(Z) - U_{\mathcal{D}_s}(Z) &= \mathop{\mathbf{E}}_{s\sim\mathcal{D}_s}\left[\ell_Z(s)\right] - \mathop{\mathbf{E}}_{t\sim\mathcal{D}_t}\left[\ell_Z(t)\right] \\
&= \inf_{\gamma\in\Gamma_{st}} \mathop{\mathbf{E}}_{(s,t)\sim\gamma}\left[\ell_Z(s) - \ell_Z(t)\right] \\
&\le \inf_{\gamma\in\Gamma_{st}} \mathop{\mathbf{E}}_{s,t}\left[L \cdot d(s,t)\right] \\
&\le L \cdot W_1(\mathcal{D}_s, \mathcal{D}_t).
\end{aligned}$$

3 Efficiently Estimating Distributional Shapley Values

Here, we describe an estimation procedure, $\mathcal{D}$-Shapley, for computing distributional Shapley values. To begin, we assume that we can actually sample from the underlying $\mathcal{D}$. Then, in Section 3.2, we propose techniques to speed up the estimation and look into the practical issues of obtaining samples from the distribution. The result of these considerations is a practically-motivated variant of the estimation procedure, Fast-$\mathcal{D}$-Shapley. In Section 3.3, we investigate how these optimizations perform empirically; we show that the strategies provide a way to smoothly trade-off the precision of the valuation for computational cost.

3.1 Obtaining unbiased estimates

The formulation from Theorem 2.3 suggests a natural algorithm for estimating the distributional Shapley values of a set of points. In particular, the distributional Shapley value $\nu(z; U, \mathcal{D}, m)$ is the expectation of the marginal contribution of z to $S \subseteq \mathcal{Z}$ on U, drawn from a specific distribution over data sets. Thus, the change in performance when we add a point z to a data set S drawn from the correct distribution will be an unbiased estimate of the distributional Shapley value. Consider the Algorithm 1, $\mathcal{D}$-Shapley, which given a subset $Z_0 \subseteq \mathcal{Z}$ of data, maintains for each $z \in Z_0$ a running average of $U(S \cup \{z\}) - U(S)$ over randomly drawn S.

In each iteration, Algorithm 1 uses a fixed sample S_t to estimate the marginal contribution to $U(S_t \cup \{z\}) - U(S_t)$ for each $z \in Z$. This reuse correlates the estimation errors between points in Z, but provides computational savings. Recall that each evaluation of $U(S)$ requires training a ML model using the points in S; thus, using the same S for each $z \in Z$ reduces the number of models to be trained by $|Z|$ per iteration. In cases where the $U(S \cup \{z\})$ can be derived efficiently from $U(S)$, the savings may be even more dramatic; for instance, given a machine-learned model trained on S, it may be significantly cheaper to derive a model trained on $S \cup \{z\}$ than retraining from scratch [GGVZ19].

The running time of Algorithm 1 can naively be upper bounded by the product of the number of iterations before termination T, the cardinality $|Z|$ of the points to valuate, and the expected time

Algorithm 1 $\mathcal{D}$-Shapley

```
Fix: potential $U : \mathcal{Z}^* \to [0,1]$; distribution $\mathcal{D}$; $m \in \mathbb{N}$
Given: data set $Z \subseteq \mathcal{Z}$ to valuate; # iterations $T \in \mathbb{N}$
  for $z \in Z$ do
    $\nu_1(z) \leftarrow 0$                                        // initialize estimates
  end for
  for $t = 1, \ldots, T$ do
    Sample $S_t \sim \mathcal{D}^{k-1}$ for $k \sim [m]$
    for $z \in Z$ do
      $\Delta_z U(S_t) \leftarrow U(S_t \cup \{z\}) - U(S_t)$
      $\nu_{t+1}(z) \leftarrow \frac{1}{t} \cdot \Delta_z U(S_t) + \frac{t-1}{t} \cdot \nu_t(z)$
                                                                 // update unbiased estimate
    end for
  end for
  return $\{(z, \nu_T(z)) : z \in Z\}$
```

to evaluate U on data sets of size $k \sim [m]$. We analyze the iteration complexity necessary to achieve ε-approximations of $\nu(z; U, \mathcal{D}, m)$ for each $z \in Z$.

Theorem 3.1. *Fixing a potential U and distribution $\mathcal{D}$, and $Z \subseteq \mathcal{Z}$, suppose $T \geq \Omega\left(\frac{\log(|Z|/\delta)}{\varepsilon^2}\right)$. Algorithm 1 produces unbiased estimates and with probability at least $1-\delta$, $|\nu(z; U, \mathcal{D}, m) - \nu_T(z)| \leq \varepsilon$. for all $z \in Z$.*

Remark. *When understanding this (and future) formal approximation guarantees, it is important to note that we take ε to be an* absolute *additive error. Recall, however, that $\nu(z; U, \mathcal{D}, m)$ is normalized by m; thus, as we take m larger, the* relative *error incurred by a fixed ε error grows. In this sense, ε should typically scale inversely as $O(1/m)$.*

The claim follows by proving uniform convergence of the estimates for each $z \in Z$. Importantly, while the samples in each iteration are correlated across $z, z' \in Z$, fixing $z \in Z$, the samples $\Delta_z U(S_t)$ are independent across iterations. We include a formal analysis in Appendix D.

3.2 Speeding up $\mathcal{D}$-Shapley: theoretical and practical considerations

Next, we propose two principled ways to speed up the baseline estimation algorithm. Under stability assumptions, the strategies maintain strong formal guarantees on the quality of the learned valuation. We also develop some guiding theory addressing practical issues that arise from the need to sample from $\mathcal{D}$. Somewhat counterintuitively, we argue that given only a fixed finite data set $B \sim \mathcal{D}^M$, we can still estimate values $\nu(z; U, \mathcal{D}, m)$ to high accuracy, for M that grows modestly with m.

Subsampling data and interpolation. Theorem 2.7 shows that for sufficiently stable potentials U, similar points have similar distributional Shapley values. This property of distributional Shapley values is not only useful for inferring the values of points $z \in \mathcal{Z}$ that were not in our original data set, but also suggests an approach for speeding up the computations of values for a fixed $Z \subseteq \mathcal{Z}$. In particular, to estimate the values for $z \in Z$ (with respect to a sufficiently Lipschitz-stable potential U) to $O(\varepsilon)$-precision, it suffices to estimate the values for an ε-cover of Z, and interpolate (e.g. via nearest neighbor search). Standard arguments show that random sampling is an effective way to construct an ε-cover [HP11].

As our first optimization, in Algorithm 2, we reduce the number of points to valuate through subsampling. Given a data set Z to valuate, we first choose a random subset $Z_p \subseteq Z$ (where each $z \in Z$ is subsampled into Z_p i.i.d. with some probability p); then, we run our estimation procedure on the points in Z_p; finally, we train a regression model on $(z, \nu_T(z))$ pairs from Z_p to predict the values of the points from $Z \setminus Z_p$. By varying the choice of $p \in [0, 1]$, we can trade-off running time for quality of estimation: $p \approx 1$ recovers the original $\mathcal{D}$-Shapley scheme, whereas $p \approx 0$ will be very fast but likely produce noisy valuations.

Importance sampling for smaller data sets. To understand the running time of Algorithm 1 further, we denote the time to evaluate U on a set of cardinality $k \in \mathbb{N}$ by $R(k)$.[5] As such, we can express the asymptotic expected running time as $|Z| \cdot T \cdot \mathbf{E}_{k \sim [m]} [R(k)]$. Note that when $U(S)$ corresponds to the accuracy of a model trained on S, the complexity of evaluating $U(S)$ may grow significantly with $|S|$. At the same time, as the data set size k grows, the marginal effect of adding $z \in Z$ to the training set tends to decrease; thus, we should need fewer large samples to accurately estimate the marginal effects. Taken together, intuitively, biasing the sampling of $k \in [m]$ towards smaller training sets could result in a faster estimation procedure with similar approximation guarantees.

Concretely, rather than sampling $k \sim [m]$ uniformly, we can importance sample each k proportional to some non-uniform weights $\{w_k : k \in [m]\}$, where the weights decrease for larger k. More formally, we weight the draw of k based on the stability of U. Algorithm 2 takes as input a set of importance weights $w = \{w_k\}$ and samples k proportionally; without loss of generality, we assume $\sum_k w_k = 1$ and let $k \sim [m]_w$ denote a sample drawn such that $\mathbf{Pr}[k] = w_k$. We show that for the right choice of weights w, sampling $k \sim [m]_w$ improves the overall running time, while maintaining ε-accurate unbiased estimates of the values $\nu(z; U, \mathcal{D}, m)$.

Theorem 3.2 (Informal). *Suppose U is $O(1/k)$-deletion stable and can be evaluated on sets of cardinality k in time $R(k) \geq \Omega(k)$. For $p \in [0, 1]$ and $w = \{w_k \propto 1/k\}$, Algorithm 2 produces estimates that with probability $1 - \delta$, are ε-accurate for all $z \in Z_p$ and runs in expected time*

$$RT_w(m) \leq \tilde{O}\left(p \cdot |Z| \cdot \frac{\log(|Z| / \delta) \cdot R(m)}{\varepsilon^2 m^2}\right).$$

To interpret this result, note that if the subsampling probability p is large enough that Z_p will ε-cover Z, then using a nearest-neighbor predictor as $\mathcal{R}$ will produce $O(\varepsilon)$-estimates for all $z \in Z$. Further, if we imagine $\varepsilon = \Theta(1/k)$, then the computational cost grows as the time it takes to train a model on m points scaled by a factor logarithmic in $|Z|$ and the failure probability. In fact, Theorem 3.2 is a special case of a more general theorem that provides a recipe for devising an appropriate sampling scheme based on the stability of the potential U. In particular, the general

[5] We assume that the running time to evaluate $U(S)$ is a function of the cardinality of S (and not other auxiliary parameters).

Algorithm 2 Fast-$\mathcal{D}$-Shapley

Fix: *potential* $U : \mathcal{Z}^* \to [0,1]$*; distribution* $\mathcal{D}$*;* $m \in \mathbb{N}$

Given: *valuation set* $Z \subseteq \mathcal{Z}$*; database* $B \sim \mathcal{D}^M$*; # iterations* $T \in \mathbb{N}$*;*
subsampling rate $p \in [0,1]$*; importance weights* $\{w_k\}$*; regression algorithm* $\mathcal{R}$

```
Subsample $Z_p \subseteq Z$ s.t. $z \in Z_p$ w.p. $p$ for all $z \in Z$
for $z \in Z_p$ do
    $\nu_1(z) \leftarrow 0$                                   // initialize estimates
end for
for $t = 1, \ldots, T$ do
    Sample $S_t \sim B^{k-1}$ for $k \sim [m]_w$
    for $z \in Z_p$ do
        $\Delta_z U(S_t) \leftarrow U(S_t \cup \{z\}) - U(S_t)$
        $\nu_{t+1}(z) \leftarrow \frac{1}{t} \cdot \frac{\Delta_z U(S_t)}{w_k m} + \frac{t-1}{t} \cdot \nu_t(z)$
                                                             // update unbiased estimate
    end for
end for
$h \leftarrow \mathcal{R}\left(\{(z, \nu_T(z)) : z \in Z_p\}\right)$
                                                             // regress on (z,val(z)) pairs
return $\{(z, h(z)) : z \in Z\}$
```

theorem (stated and proved in Appendix D) shows that the more stable the potential, the more we can bias sampling in favor of smaller sample sizes.

Estimating distributional Shapley from data. Estimating distributional Shapley values $\nu(z; U, \mathcal{D}, m)$ requires samples from the distribution $\mathcal{D}$. In practice, we often want evaluate the values with respect to a distribution $\mathcal{D}$ for which we only have some database $B \sim \mathcal{D}^M$ for some large (but finite) $M \in \mathbb{N}$. In such a setting, we need to be careful; indeed, avoiding artifacts from a single draw of data is the principle motivation for introducing the distributional Shapley framework. In fact, the analysis of Theorem 3.2 also reveals an upper bound on how big the database should be in order to obtain accurate estimates with respect to $\mathcal{D}$. As a concrete bound, if U is $O(1/k)$-deletion stable and we take $\varepsilon = \Theta(1/m)$ error, then the database need only be

$$M \leq \tilde{O}\left(m \cdot \log(|Z|/\delta)\right).$$

In other words, for a sufficiently stable potential U, the data complexity grows modestly with m. Note that, again, this bound leverages the fact that in every iteration, we reuse the same sample $S_t \sim \mathcal{D}^k$ for each $z \in Z$. See Appendix D for a more detailed analysis.

In practice, we find that sampling subsets of data from the database with replacement works well; we describe the full procedure in Algorithm 2, where we denote an i.i.d. sample of k points drawn uniformly from the database as $S \sim B^k$. Finally, we note that ideally, m should be close to the size of the training sets that model developers to use; in practice, these data set sizes may vary widely. One appealing aspect of both $\mathcal{D}$-Shapley algorithms is that when we estimate values with respect to m, the samples we obtain also allow us to simultaneously estimate $\nu(z; U, \mathcal{D}, m')$ for any $m' \leq m$. Indeed, we can simply truncate our estimates to only include samples corresponding to S_t with $|S_t| \leq m'$.

3.3 Empirical performance

We investigate the empirical effectiveness of the distributional Shapley framework by running experiments in three settings on large real-world data sets. The first setting uses the UK Biobank data set, containing the genotypic and phenotypic data of individuals in the UK [SGA+15]; we evaluate a task of predicting whether the patient will be diagnosed with breast cancer using 120 features. Overall, our data has 10K patients (5K diagnosed positively); we use 9K patients as our database (B), and take classification accuracy on a hold-out set of 500 patients as the performance metric (U). The second data set is Adult Income where the task is to predict whether income exceeds \$50K/yr given 14 personal features [DG17]. With 50K individuals total, we use 40K as our database, and classification accuracy on 5K individuals as our performance metric. In these two experiments, we take the maximum data set size $m = 1\text{K}$ and $m = 5\text{K}$, respectively.

For both settings, we first run $\mathcal{D}$-Shapley without optimizations as a baseline. As a point of comparison, in these settings the computational cost of this baseline is on the same order as running the TMC-Shapley algorithm of [GZ19] that computes the data Shapley values $\phi(z; U, B)$ for each z in the data set B. Given this baseline, we evaluate the effectiveness of the proposed optimizations, using weighted sampling and interpolation (separately), for various levels of computational savings. In particular, we vary the sampling weights $\{w_k\}$ and subsampling probability p to vary the computational cost (where weighting towards smaller k and taking p smaller each yield more computational savings). All algorithms are truncated when the average absolute change in value in the past 100 iterations is less than 1%.

To evaluate the quality of the distributional Shapley estimates, we perform a point removal experiment, as proposed by [GZ19], where given a training set, we iteratively remove points, retrain the model, and observe how the performance changes. In particular, we remove points from most to least valuable (according to our estimates), and compare to the baseline of removing random points. Intuitively, removing high value data points should result in a more significant drop in

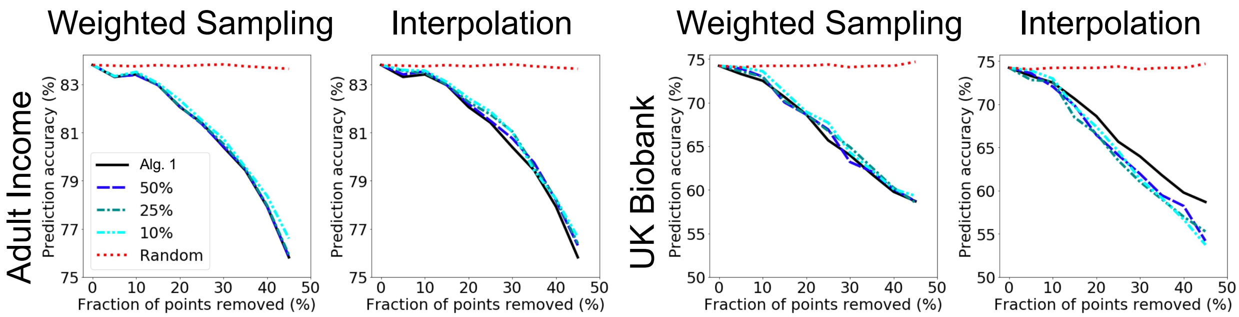


Figure 1: **Point removal performance.** Given a data set and task, we iteratively a point, retrain the model, and evaluate its performance. Each curve corresponds to a different point removal order, based on the estimated distributional Shapley values (compared to random). For example, the 10% curve correspond to estimating values with 10% of the baseline computation of Algorithm 1. We plot classification accuracy vs. fraction of data points removed from the training set, for each task and each optimization method.

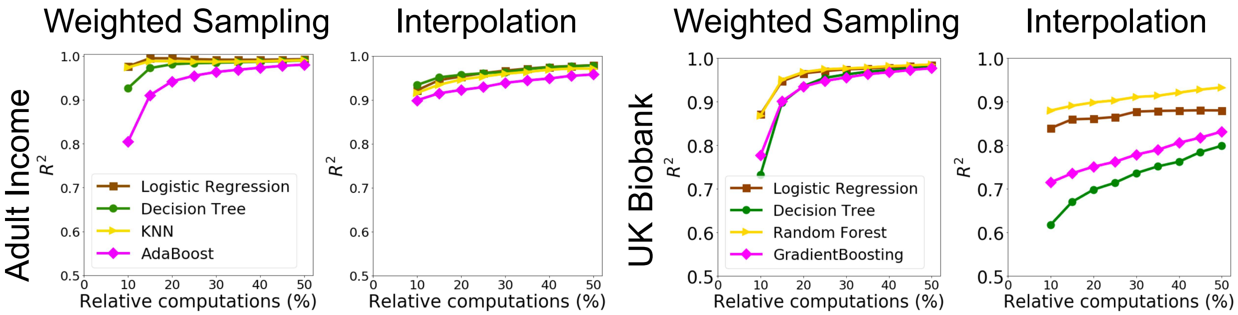


Figure 2: **Smooth trade-off between computation and recovery.** For each task, we plot the R^2 coefficient between the values computed using Algorithm 1 vs. the relative computational cost (as in Figure 1). The results show that there is a smooth trade-off between the recovery precision of the distributional Shapley values and the cost, across a wide range of learning algorithms.

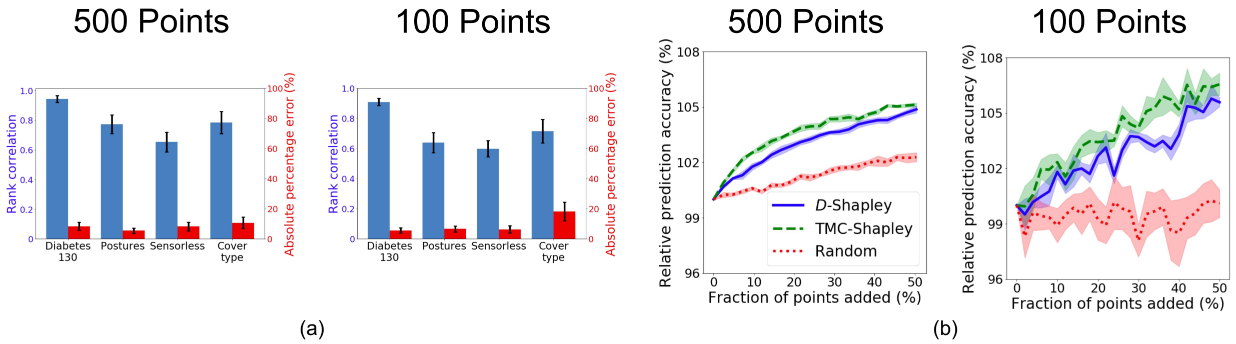
the model's performance. We report the results of this point removal experiment using the values determined using the baseline Algorithm 1, as well as various factor speed-ups (where $t\%$ refers to the computational cost compared to baseline).

As Figure 1 demonstrates, when training a logistic regression model, removing the high distributional Shapley valued points causes a sharp decrease in accuracy on both tasks, even when using the most aggressive weighted sampling and interpolation optimizations. Appendix E reports the results for various other models. As a finer point of investigation, we report the correlation between the estimated values without optimizations and with various levels of computational savings, for a handful of prediction models. Figure 2 plots the R^2 curves and shows that the optimizations provide a smooth interpolation between computational cost and recovery, across every model type. It is especially interesting that these trade-offs are consistently smooth across a variety of models using the 01-loss, which do not necessarily induce a potential U with formal guarantees of stability.

In our final setting, we push the limits of what types of data can be valuated. Specifically, by combining both weighted sampling and interpolation (resulting in a 500× speed-up), we estimate the values of 50K images from the CIFAR10 data set; valuating this data set would be prohibitively expensive using prior Shapley-based techniques. In particular, to obtain accurate estimates for each point, TMC-Shapley would require an unreasonably large number of Monte Carlo iterations due to the sheer size of the data base to valuate. We valuate points based on an image classification task, and demonstrate that the estimates identify highly valuable points, in Appendix E.

4 Case Study: Consistently Pricing Data

Next, we consider a natural setting where a data broker wishes to sell data to various buyers. Each buyer could already own some private data. In particular, suppose the broker plans to sell the set S and a buyer holds a private data set B; in this case, the relevant values are the data Shapley values $\phi(z; U, B \cup S)$ for each $z \in S$. Within the original data Shapley framework, computing these values requires a single party to hold both B and S. For a multitude of financial and legal concerns,



(a)

(b)

Figure 3: **Consistent Pricing.** Each buyer holds a data set B; the seller sells a data set S, where $|B| = |S| = m$. We compare the values estimated by the seller $\nu(z; U, \mathcal{D}, m)$ and $\phi(z; U, B \cup S)$. (a) For various data sets and two data set sizes ($m = 100$ and $m = 500$): in blue, we plot the average rank correlation between $\nu(z)$ and $\phi(z)$ for $z \in S$; in red, we plot the average absolute percentage error between the seller's and buyer's estimates. (b) Points from S are added to B in three different orders: according to ν ($\mathcal{D}$-Shapley), according to ϕ (TMC), and randomly. The plot shows the change in the accuracy of the model, relative to its performance using the buyer's initial dataset, as the points are added; shading indicates standard error of the mean.

neither party may be willing to send their data to the other before agreeing to the purchase. Such a scenario represents a fundamental limitation of the non-distributional Shapley framework that seemed to jeopardize its practical viability. We argue that the distributional Shapley framework largely resolves this particular issue: without exchanging data up front, the broker simply estimates the values $\nu(z; U, \mathcal{D}, m)$; in expectation, these values will accurately reflect the value to a buyer with a private data set B drawn from a distribution close to $\mathcal{D}$.

We report the results of this case study on four large different data sets in Figure 3, whose details are included in Appendix F. For each data set, a set of buyers holds a small data set B (100 or 500 points), and the broker sells them a data set S of the same size; the buyers then valuate the points in S by running the TMC-Shapley algorithm of [GZ19] on $B \cup S$. In Figure 3(a), we show that the rank correlation between the broker's distributional estimates $\nu(z; U, \mathcal{D}, m)$ and the buyer's observed values $\phi(z; U, B \cup S)$ is generally high. Even when the rank correlation is a bit lower (≈ 0.6), the broker and buyer agree on the value of the set as a whole. Specifically, we observe that the seller's estimates are approximately unbiased, and the absolute percentage error is low, where

$$APE = \frac{\left|\sum_{z \in S} \nu(z; U, \mathcal{D}, m) - \phi(z; U, B \cup S)\right|}{\sum_{z \in S} \nu(z; U, \mathcal{D}, m)}.$$

In Figure 3(b), we show the results of a point addition experiment for the Diabetes130 data set. Here, we consider the effect of adding the points of S to B under three different orderings: according to the broker's estimates $\nu(z; U, \mathcal{D}, m)$, according to the buyer's estimates $\phi(z; U, B \cup S)$, and under a random ordering. We observe that the performance (classification accuracy) increase by adding the points according to $\nu(z)$ and according to $\phi(z)$ track one another well; after the addition of all of S, the resulting models achieve essentially the same performance and considerably outperforming random. We report results for the other data sets in Appendix F.

5 Discussion

The present work makes significant progress on understanding statistical aspects in determining the value of data. In particular, by reformulating the data Shapley value as a distributional quantity, we obtain a valuation function that does not depend on a fixed data set; reducing the dependence on the specific draw of data eliminates inconsistencies in valuation that can arise to sampling artifacts. Further, we demonstrate that the distributional Shapley framework provides an avenue to valuate data across a wide variety of tasks, providing stronger theoretical guarantees and orders of magnitude speed-ups over prior estimation schemes. In particular, the stability results that we prove for distributional Shapley (Theorems 2.7 and 2.8) are not generally true for the original data Shapley due to its dependence on a fixed dataset.

One outstanding limitation of the present work is the reliance on a known task, algorithm, and performance metric (i.e. taking the potential U to be fixed). We propose reducing the dependence on these assumptions as a direction for future investigations; indeed, very recent work has started to chip away at the assumption that the learning algorithm is fixed in advance [YGZ19].

The distributional Shapley perspective also raises the thought-provoking research question of whether we can valuate data while protecting the privacy of individuals who contribute their data. One severe limitation of the data Shapley framework, is that the value of every point depends nontrivially on every other point in the data set. In a sense, this makes the data Shapley value an inherently non-private value: the estimate of $\phi(z; U, B)$ for a point $z \in B$ reveals information about the other points in B. By marginalizing the dependence on the data set, the distributional Shapley framework opens the door for to estimating data valuations while satisfying strong notions of privacy, such as differential privacy [DMNS06]. Such an estimation scheme could serve as a powerful tool amidst increasing calls to ensure the privacy of and compensate individuals for their personal data [LNG19].

References

[ADS19] Anish Agarwal, Munther Dahleh, and Tuhin Sarkar. A marketplace for data: An algorithmic solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 701–726, 2019.

[Aga11] Shivani Agarwal. Algorithmic stability. Lecture notes on Statistical Learning Theory, 2011.

[AS74] Robert J Aumann and Lloyd S Shapley. *Values of non-atomic games*. Princeton University Press, 1974.

[BE02] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

[CDR07] Shay Cohen, Gideon Dror, and Eytan Ruppin. Feature selection via coalitional game theory. *Neural Computation*, 19(7):1939–1961, 2007.

[CSWJ18] Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. L-shapley and c-shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610*, 2018.

[DG17] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284, 2006.

[DSZ16] Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 598–617. IEEE, 2016.

[GAZ17] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. *arXiv preprint arXiv:1710.10547*, 2017.

[GGVZ19] Antonio Ginart, Melody Guan, Gregory Valiant, and James Y Zou. Making ai forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems*, pages 3513–3526, 2019.

[GZ19] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251, 2019.

[HP11] Sariel Har-Peled. *Geometric approximation algorithms*. Number 173. American Mathematical Soc., 2011.

[JDW$^+$19a] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *Proceedings of the VLDB Endowment*, 12(11):1610–1623, 2019.

[JDW+19b] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176, 2019.

[K+10] Igor Kononenko et al. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11(Jan):1–18, 2010.

[KPR01] Jon Kleinberg, Christos H Papadimitriou, and Prabhakar Raghavan. On the value of private information. In *Theoretical Aspects Of Rationality And Knowledge: Proceedings of the 8 th conference on Theoretical aspects of rationality and knowledge*, volume 8, pages 249–257. Citeseer, 2001.

[LL17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774, 2017.

[LNG19] Katrina Ligett, Kobbi Nissim, and Ayelet Gordon-Tapiero. Data co-ops. https://csrcl.huji.ac.il/book/data-co-ops, 2019.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[SDG+14] Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international*, 2014, 2014.

[SGA+15] Cathie Sudlow, John Gallacher, Naomi Allen, Valerie Beral, Paul Burton, John Danesh, Paul Downey, Paul Elliott, Jane Green, Martin Landray, et al. Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS medicine*, 12(3):e1001779, 2015.

[Sha53] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[SR+88] Lloyd S Shapley, Alvin E Roth, et al. *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[SVI+16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[YGZ19] Gal Yona, Amirata Ghorbani, and James Zou. Who's responsible? jointly quantifying the contribution of the learning algorithm and training data. *arXiv preprint arXiv:1910.04214*, 2019.

A Review of Shapley Axioms

Here, we provide a high-level review of the axioms that Shapley used to describe an equitable valuation function [Sha53]. We consider the data Shapley setting, letting $\phi(z; U, B)$ denote the value of $z \in B$ for a finite subset $B \subseteq \mathcal{Z}$ with respect to potential $U : \mathcal{Z}^* \to [0, 1]$.

- *Symmetry* – Consider $z_i, z_j \in B$; suppose for all $S \subseteq B \setminus \{z_i, z_j\}$, $U(S \cup \{z_i\}) = U(S \cup \{z_j\})$. Then,
$$\phi(z_i; U, B) = \phi(z_j; U, B).$$
That is, if two data points are equivalent, then they should receive the same value.

- *Null player* – Consider $z \in B$; suppose for all $S \subseteq B \setminus \{z\}$, $U(S \cup \{z\}) = U(S)$. Then,
$$\phi(z; U, B) = 0.$$
That is, if a data point contributes no marginal gain in potential to any nontrivial subset, then it receives no value.

- *Additivity* – Consider two potentials U_1, U_2. For all $z \in B$,
$$\phi(z; U_1 + U_2, B) = \phi(z; U_1, B) + \phi(z; U_2, B).$$
That is, the value of a data point with respect to the combination of two tasks (addition of two potentials) is the sum of the values with respect to each task (potential) separately.

Theorem A.1 ([Sha53]). *The Shapley value is the unique valuation function that satisfies the symmetry, null player, and additivity axioms.*

Additionally, the Shapley value satisfies the desirable property that it allocates all of the value to the contributors.

- *Efficiency* – The sum of the individuals' Shapley values equals the value of the coalition.
$$\sum_{z \in B} \phi(z; U, B) = U(B) - U(\emptyset).$$

It is straightforward to verify that the distributional Shapley value immediately inherits the properties of symmetry, null player, and additivity (by linearity of expectation). Further, it satisfies an on-average variant of efficiency.

Proposition A.2. *Given a potential U and a data distribution $\mathcal{D}$, for $m \in \mathbb{N}$,*
$$\mathop{\mathbf{E}}_{z \sim \mathcal{D}} [\nu(z; U, \mathcal{D}, m)] = \frac{\mathbf{E}_{B \sim \mathcal{D}^m} [U(B)] - U(\emptyset)}{m}.$$

Proof. We expand the expected distributional Shapley value with its definition and then apply linearity of expectation.

$$\begin{aligned}
\mathop{\mathbf{E}}_{z\sim\mathcal{D}}\left[\nu(z;U,\mathcal{D},m)\right] &= \mathop{\mathbf{E}}_{z\sim\mathcal{D}}\left[\mathop{\mathbf{E}}_{\substack{k\sim[m]\\ S\sim\mathcal{D}^{k-1}}}\left[U(S\cup\{z\})-U(S)\right]\right]\\
&= \frac{1}{m}\cdot\sum_{k=1}^{m}\left(\mathop{\mathbf{E}}_{\substack{z\sim\mathcal{D}\\ S\sim\mathcal{D}^{k-1}}}\left[U(S\cup\{z\})\right]-\mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}}\left[U(S)\right]\right)\\
&= \frac{1}{m}\cdot\sum_{k=1}^{m}\left(\mathop{\mathbf{E}}_{S_k\sim\mathcal{D}^{k}}\left[U(S_k)\right]-\mathop{\mathbf{E}}_{S_{k-1}\sim\mathcal{D}^{k-1}}\left[U(S_{k-1})\right]\right)\\
&= \frac{1}{m}\cdot\left(\mathop{\mathbf{E}}_{B\sim\mathcal{D}^{m}}\left[U(B)\right]-U(\emptyset)\right)
\end{aligned}$$

□

B Distributional Shapley Value for Mean Estimation

Proposition (Restatement of Proposition 2.4)**.** *Suppose $\mathcal{D}$ has bounded second moments. Then for $z\in\mathcal{Z}$ and $m\in\mathbb{N}$, $\nu(z;U_\mu,\mathcal{D},m)$ for mean estimation over $\mathcal{D}$ is given by*

$$\frac{\mathbf{E}_{S\sim\mathcal{D}^m}\left[U(S)\right]}{m}+\frac{C_m}{m}\cdot\left(\mathop{\mathbf{E}}_{s\sim\mathcal{D}}\left[\|s-\mu\|^2\right]-\|z-\mu\|^2\right)$$

for an explicit constant $C_m=\Theta(1)$ determined by m.

Proof. Consider the unsupervised learning task of mean estimation using the empirical estimator. Specifically, suppose we receive samples from some distribution $\mathcal{D}$ supported on $\mathbb{R}^d$ with mean $\mu=\mathbf{E}_{s\sim\mathcal{D}}[s]$ and bounded second moments. Given a subset $S\subseteq\mathbb{R}^d$, we consider the empirical estimator $\hat{\mu}_S=\frac{1}{|S|}\cdot\sum_{s\in S}s$. We define a potential $U(S)$ by the performance of the empirical estimator. For notational convenience, let $\mathbf{E}_{s\sim\mathcal{D}}\left[\|s-\mu\|^2\right]=R^2$ for some $R=\Theta(1)$.

$$\begin{aligned}
U(S) &= \mathop{\mathbf{E}}_{s\sim\mathcal{D}}\left[\|s-\mu\|^2\right]-\|\hat{\mu}_S-\mu\|^2\\
&= R^2-\|\hat{\mu}_S-\mu\|^2
\end{aligned}$$

By convention, we will assume that $U(\emptyset)=0$. As such, we can evaluate the difference in potentials as follows.

$$\begin{aligned}
&= \left(R^2-\left\|\mu-\hat{\mu}_{S\cup\{z\}}\right\|^2\right)-\left(R^2-\|\mu-\hat{\mu}_S\|^2\right)\\
&= \|\mu-\hat{\mu}_S\|^2-\left\|\mu-\hat{\mu}_{S\cup\{z\}}\right\|^2
\end{aligned}$$

Importantly, note that we can relate $\hat{\mu}_{S\cup\{z\}}$ to $\hat{\mu}_S$.

$$\hat{\mu}_{S\cup\{z\}} = \hat{\mu}_S + \frac{1}{k}\cdot(z - \hat{\mu}_S)$$

Using these expressions, we can expand the distributional Shapley value into a form that will be convenient to work with.

$$\begin{aligned}
\nu(z;U,\mathcal{D},m) &= \mathop{\mathbf{E}}_{\substack{k\sim[m]\\ S\sim\mathcal{D}^{k-1}}}\left[U(S\cup\{z\}) - U(S)\right]\\
&= \frac{1}{m}\cdot\sum_{k=1}^{m}\mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}}\left[U(S\cup\{z\}) - U(S)\right]\\
&= \frac{1}{m}\cdot\left(U(\{z\}) - U(\emptyset) + \sum_{k=2}^{m}\mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}}\left[U(S\cup\{z\}) - U(S)\right]\right)\\
&= \frac{1}{m}\cdot\left(R^2 - \|z-\mu\|^2 + \sum_{k=2}^{m}\mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}}\left[\|\mu-\hat{\mu}_S\|^2 - \left\|\mu-\hat{\mu}_{S\cup\{z\}}\right\|^2\right]\right)
\end{aligned}$$

We, thus, focus our efforts on bounding the summation from $k=2$ to m. As such, we can evaluate the difference in potentials within the expectation as follows.

$$\begin{aligned}
&\|\mu-\hat{\mu}_S\|^2 - \left\|\mu-\hat{\mu}_{S\cup\{z\}}\right\|^2\\
&= \|\mu-\hat{\mu}_S\|^2 - \left\|\mu-\hat{\mu}_S - \frac{1}{k}\cdot(z-\hat{\mu}_S)\right\|^2\\
&= \|\mu-\hat{\mu}_S\|^2 - \left(\|\mu-\hat{\mu}_S\|^2 + \frac{1}{k^2}\cdot\|z-\hat{\mu}_S\|^2 - \frac{2}{k}\cdot\langle\mu-\hat{\mu}_S, z-\hat{\mu}_S\rangle\right)\\
&= \frac{2}{k}\cdot\langle\mu-\hat{\mu}_S, z-\hat{\mu}_S\rangle - \frac{1}{k^2}\cdot\|z-\hat{\mu}_S\|^2
\end{aligned}$$

Taking an expectation over $S\sim\mathcal{D}^{k-1}$, we can simplify each term in the summation separately; first, some identities that will be useful and hold for all $n\in\mathbb{N}$:

$$\mathop{\mathbf{E}}_{S\sim\mathcal{D}^n}\left[\hat{\mu}_S\right] = \mu \tag{10}$$

$$\mathop{\mathbf{E}}_{S\sim\mathcal{D}^n}\left[\langle\mu-\hat{\mu}_S, q\rangle\right] = 0 \tag{11}$$

$$\mathop{\mathbf{E}}_{S\sim\mathcal{D}^n}\left[\|\hat{\mu}_S\|^2 - \|\mu\|^2\right] = \mathop{\mathbf{E}}_{S\sim\mathcal{D}^n}\left[\|\hat{\mu}_S-\mu\|^2\right] = \frac{1}{n}\cdot\mathop{\mathbf{E}}_{s\sim\mathcal{D}}\left[\|s-\mu\|^2\right] \tag{12}$$

where (10) follows because $\hat{\mu}_S$ an unbiased estimator of μ; (11) holds for all $q\in\mathbb{R}^d$; and (12) is a well-known fact that can be derived using (10) and (11).

Beginning with the first inner product.

$$\begin{aligned}
\mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}}\left[\langle\mu-\hat{\mu}_S, z-\hat{\mu}_S\rangle\right] &= \mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}}\left[\langle\mu-\hat{\mu}_S, \mu-\hat{\mu}_S\rangle\right] \qquad (13)\\
&= \mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}}\left[\|\mu-\hat{\mu}_S\|^2\right]\\
&= \frac{1}{k-1}\cdot\mathop{\mathbf{E}}_{s\sim\mathcal{D}}\left[\|s-\mu\|^2\right] \qquad (14)
\end{aligned}$$

where (13) applies (11) with $q = \mu - z$ and (14) applies (12).

Expanding the next term.

$$\begin{aligned}
\mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}}\left[\|z-\hat{\mu}_S\|^2\right] &= \mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}}\left[\|z\|^2 + \|\hat{\mu}_S\|^2 - 2\langle z,\hat{\mu}_S\rangle + \|\mu\|^2 - \|\mu\|^2\right]\\
&= \mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}}\left[\|z\|^2 - 2\langle z,\hat{\mu}_S\rangle + \|\mu\|^2\right] + \mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}}\left[\|\hat{\mu}_S\|^2 - \|\mu\|^2\right]\\
&= \|z-\mu\|^2 + \frac{1}{k-1}\cdot \mathop{\mathbf{E}}_{s\sim\mathcal{D}}\left[\|s-\mu\|^2\right] \qquad (15)
\end{aligned}$$

where (15) follows by applying linearity of expectation and (10) to the first term and (12) to the second term.

Thus, in all, the value can be expressed as follows.

$$\begin{aligned}
&\sum_{k=2}^{m}\mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}}\left[\frac{2}{k}\cdot\langle\mu-\hat{\mu}_S, z-\hat{\mu}_S\rangle - \frac{1}{k^2}\cdot\|z-\hat{\mu}_S\|^2\right]\\
&= \sum_{k=2}^{m}\left(\frac{2}{k\cdot(k-1)}\cdot\mathop{\mathbf{E}}_{s\sim\mathcal{D}}\left[\|s-\mu\|^2\right] - \frac{1}{k^2\cdot(k-1)}\cdot\mathop{\mathbf{E}}_{s\sim\mathcal{D}}\left[\|s-\mu\|^2\right] - \frac{1}{k^2}\cdot\|z-\mu\|^2\right)\\
&= \sum_{k=2}^{m}\left(\frac{2R^2 - \|z-\mu\|^2}{k\cdot(k-1)} - \frac{R^2-\|z-\mu\|^2}{k^2\cdot(k-1)}\right)\\
&= \frac{m-1}{m}\cdot\left(2R^2 - \|z-\mu\|^2\right) + c(m)\cdot\left(R^2-\|z-\mu\|^2\right)\\
&= \frac{m-1}{m}\cdot R^2 + (1-1/m+c(m))\cdot\left(R^2-\|z-\mu\|^2\right)
\end{aligned}$$

where $\sum_{k=2}^{m}\frac{1}{k\cdot(k-1)} = \frac{m-1}{m}$ and we take $c(m) = \sum_{k=2}^{m}\frac{1}{k^2\cdot(k-1)}$. Thus, plugging this expression back into our original expansion.

$$\begin{aligned}
&\nu(z;U,\mathcal{D},m)\\
&= \frac{1}{m}\cdot\left(\frac{m-1}{m}\cdot\left(2R^2-\|z-\mu\|^2\right) + (1+c(m))\cdot\left(R^2-\|z-\mu\|^2\right)\right)\\
&= \frac{m-1}{m^2}\cdot R^2 + \frac{C(m)}{m}\cdot\left(R^2-\|z-\mu\|^2\right)\\
&= \frac{1}{m}\cdot\left(C(m)\cdot\left(\mathop{\mathbf{E}}_{s\sim\mathcal{D}}\left[\|s-\mu\|^2\right] - \|z-\mu\|^2\right) + \left(\mathop{\mathbf{E}}_{s\sim\mathcal{D}}\left[\|s-\mu\|^2\right] - \mathop{\mathbf{E}}_{S\sim\mathcal{D}^m}\left[\|\hat{\mu}_S-\mu\|^2\right]\right)\right)
\end{aligned}$$

where $C(m) = 2 - 1/m + c(m) = \Theta(1)$ is an explicit function of m, and we use the fact that $\mathbf{E}_{S\sim\mathcal{D}^m}\left[\|\hat{\mu}_S-\mu\|^2\right] = \frac{1}{m}\cdot R^2$. □

C Lipschitz Stability of RKHS

Suppose $\mathcal{Z} = \mathcal{X}\times\mathcal{Y}$; let $\mathcal{F}$ to denote a Reproducing Kernel Hilbert Space (RKHS), with associated feature map $\varphi : \mathcal{X}\to\mathcal{F}$, inner product $\langle\cdot,\cdot\rangle_{\mathcal{F}}$, and norm $\|\cdot\|_{\mathcal{F}}$, such that for all $f\in\mathcal{F}$,

$$f(x) = \langle f, \varphi(x)\rangle_{\mathcal{F}}$$

Given $\mathcal{F}$, we define a natural metric over $\mathcal{Z}$, where for labeled pairs $z_i = (x_i, y_i)$ and $z_j = (x_j, y_j)$,

$$d_{\mathcal{F}}(z_i, z_j) = \begin{cases} \|\varphi(x_i) - \varphi(x_j)\|_{\mathcal{F}} & \text{if } y_i = y_j \\ +\infty & \text{o.w.} \end{cases}$$

That is, the distance is given by the RKHS norm if x_i and x_j have the same label, and are arbitrarily dissimilar otherwise.

We define the potential of a subset $S \subseteq \mathcal{Z}$ for an RKHS learning problem as the performance achieved when training using S. Specifically, suppose $\ell : [0,1] \times [0,1] \to \mathbb{R}^+$ is an L-Lipschitz, convex loss function and $\mathcal{D}$ is a distribution supported on $\mathcal{Z}$. We define the potential function $U_{\mathcal{F}} : \mathcal{Z}^* \to [0,1]$ in terms of the population loss over $\mathcal{D}$ of the following regularized ERM.

$$U_{\mathcal{F}}(S) = 1 - \mathop{\mathbf{E}}_{(x,y)\sim\mathcal{D}}\left[\ell(f_S(x), y)\right]$$
$$\text{where}\quad f_S = \mathop{\mathrm{argmin}}_{f\in\mathcal{F}}\left\{\mathrm{er}_S(f) + \frac{\lambda}{2}\|f\|_{\mathcal{F}}^2\right\}$$

where $\mathrm{er}_S(f) = \frac{1}{|S|}\sum_{(x,y)\in S}\ell(f(x), y)$ and $\lambda > 0$.

Lemma C.1 (Learning RKHS is Lipschitz stable)**.** *Suppose $\mathcal{F}$ is a RKHS with feature map $\phi : \mathcal{X} \to \mathcal{F}$. Let $\mathcal{D}$ be a distribution over $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ such that $\mathbf{E}_{(x,y)\sim\mathcal{D}}\left[\|\phi(x)\|_{\mathcal{F}}\right] = R$. Then, $U_{\mathcal{F}} : \mathcal{Z}^* \to [0,1]$ is $(2L^2R/\lambda k)$-Lipschitz stable with respect to $d_{\mathcal{F}}$.*

In other words, as with the standard notion of replacement stability, the Lipschitz stability depends on the Lipschitz constant of the loss, the expected norm over $\mathcal{D}$, and (inversely on) the degree of regularization.

Proof. We follow the proof of replacement stability of RKHS from [Aga11] closely. For notational convenience, we drop reference to $\mathcal{F}$ in $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$.

Suppose $S \in \mathcal{Z}^{k-1}$ and suppose $z, z' \in \mathcal{Z}$ are two points such that $z = (x, y)$ and $z' = (x', y)$; i.e. they share the same label. By the definition of $d_{\mathcal{F}}$, this is the only case we need to consider. Let $f, f' \in \mathcal{F}$ denote the empirical risk minimizers over $S \cup \{z\}$ and $S \cup \{z'\}$, respectively.

$$f = \mathop{\mathrm{argmin}}_{g\in\mathcal{F}}\left\{\mathrm{er}_{S\cup\{z\}}(g) + \frac{\lambda}{2}\|g\|^2\right\}$$
$$f' = \mathop{\mathrm{argmin}}_{g\in\mathcal{F}}\left\{\mathrm{er}_{S\cup\{z'\}}(g) + \frac{\lambda}{2}\|g\|^2\right\}$$

For $\alpha \in [0,1]$, let

$$f_\alpha = \alpha \cdot f + (1-\alpha)\cdot f'$$
$$f'_\alpha = (1-\alpha)\cdot f + \alpha \cdot f'.$$

By the assumption that ℓ is convex, we can derive the following inequalities for any $(x, y) \in \mathcal{Z}$ and any $S \subseteq \mathcal{Z}$.

$$\begin{aligned} &\ell(f_\alpha(x), y) \le \alpha \cdot \ell(f(x), y) + (1-\alpha) \cdot \ell(f'(x), y) \\ &\quad \Longrightarrow \ \mathrm{er}_S(f_\alpha) \le \alpha \cdot \mathrm{er}_S(f) + (1-\alpha) \cdot \mathrm{er}_S(f') \end{aligned} \tag{16}$$

Note that f_α and f'_α are also feasible hypotheses in the Hilbert space. Thus, by the fact that f, f' are ERMs,

$$\begin{aligned} \mathrm{er}_{S\cup\{z\}}(f) + \frac{\lambda}{2}\|f\|^2 &\le \mathrm{er}_{S\cup\{z\}}(f_\alpha) + \frac{\lambda}{2}\|f_\alpha\|^2 \\ \mathrm{er}_{S\cup\{z'\}}(f') + \frac{\lambda}{2}\left\|f'\right\|^2 &\le \mathrm{er}_{S\cup\{z'\}}(f'_\alpha) + \frac{\lambda}{2}\left\|f'_\alpha\right\|^2 \end{aligned}$$

Rearranging, and applying convexity, we derive the following inequality.

$$\frac{\lambda}{2} \cdot \left(\|f\|^2 + \left\|f'\right\|^2 - \|f_\alpha\|^2 - \left\|f'_\alpha\right\|^2\right) \le \mathrm{er}_{S\cup\{z\}}(f_\alpha) - \mathrm{er}_{S\cup\{z\}}(f) + \mathrm{er}_{S\cup\{z'\}}(f'_\alpha) - \mathrm{er}_{S\cup\{z'\}}(f') \tag{17}$$

We can simplify the inner term of the LHS of (17) as follows.

$$\begin{aligned} \|f\|^2 + \left\|f'\right\|^2 - \|f_\alpha\|^2 - \left\|f'_\alpha\right\|^2 &= \|f\|^2 + \left\|f'\right\|^2 - \left\|f' - \alpha(f' - f)\right\|^2 - \left\|f + \alpha(f' - f)\right\|^2 \\ &= 2\alpha\langle f' - f, f' - f\rangle - 2\alpha^2\left\|f' - f\right\|^2 \\ &= 2\alpha(1-\alpha)\left\|f - f'\right\|^2 \end{aligned}$$

Then, we can bound the RHS of (17) as follows.

$$\begin{aligned} &\mathrm{er}_{S\cup\{z\}}(f_\alpha) - \mathrm{er}_{S\cup\{z\}}(f) + \mathrm{er}_{S\cup\{z'\}}(f'_\alpha) - \mathrm{er}_{S\cup\{z'\}}(f') \\ &\qquad \le (1-\alpha) \cdot \left(\mathrm{er}_{S\cup\{z\}}(f') - \mathrm{er}_{S\cup\{z\}}(f) + \mathrm{er}_{S\cup\{z'\}}(f) - \mathrm{er}_{S\cup\{z'\}}(f')\right) && (18) \\ &\qquad = \frac{1-\alpha}{k} \cdot \left(\ell(f'(x), y) - \ell(f(x), y) + \ell(f(x'), y) - \ell(f'(x'), y) + (k-1)\cdot 0\right) && (19) \\ &\qquad = \frac{(1-\alpha)}{k} \cdot \left(\ell(f'(x), y) - \ell(f'(x'), y) - \ell(f(x), y) + \ell(f(x'), y)\right) \\ &\qquad \le \frac{(1-\alpha)L}{k} \cdot \langle f' - f, \varphi(x) - \varphi(x')\rangle && (20) \\ &\qquad \le \frac{(1-\alpha)L}{k} \cdot \left\|f - f'\right\| \cdot \left\|\varphi(x) - \varphi(x')\right\| && (21) \end{aligned}$$

where (18) follows by expanding according to (16), and then simplifying; (19) follows by the fact that $\mathrm{er}_{S\cup\{z\}}(f) = \frac{1}{k} \cdot \ell(f(x), y) + \frac{k-1}{k}\mathrm{er}_S(f)$, but the errors on S cancel to 0, $\mathrm{er}_S(f') - \mathrm{er}_S(f) + \mathrm{er}_S(f) - \mathrm{er}_S(f')$; (20) follows by the Lipschitzness of ℓ; and (21) follows by Cauchy-Schwarz.

The analysis above applied for any $\alpha \in [0, 1]$; taking $\alpha = 1/2$ implies

$$\begin{aligned} &\frac{\lambda}{4} \cdot \left\|f - f'\right\|^2 \le \frac{L}{2k} \cdot \left\|f - f'\right\| \cdot \left\|\varphi(x) - \varphi(x')\right\| \\ &\qquad \Longrightarrow \ \left\|f - f'\right\| \le \frac{2L}{\lambda k} \cdot d_{\mathcal{F}}(z, z'). \end{aligned}$$

Because ℓ is L-Lipschitz, we obtain Lipschitz-stability of $U_{\mathcal{F}}$.

$$\begin{aligned} U_{\mathcal{F}}(S \cup \{z\}) - U_{\mathcal{F}}(S \cup \{z'\}) &= \mathop{\mathbf{E}}_{(x_0,y_0)\sim\mathcal{D}} \left[\ell(f'(x_0), y_0) - \ell(f(x_0, y_0)\right] \\ &\leq \mathbf{E}\left[\|\varphi(x_0)\|\right] \cdot \frac{2L^2}{\lambda k} \cdot d_{\mathcal{F}}(z, z') \\ &= \frac{2L^2 R}{\lambda k} \cdot d_{\mathcal{F}}(z, z') \end{aligned}$$

□

D Estimating Distributional Shapley Values – Analysis

We require the following standard concentration inequality.

Theorem (Hoeffding's Inequality). *Suppose $X_1, \ldots, X_T$ are independent random variables, where X_t is bounded in the range $[-b_t, b_t]$. Let $\overline{X} = \frac{1}{T}\sum_{t=1}^{T} X_t$. Then,*

$$\mathbf{Pr}\left[\left|\overline{X} - \mathbf{E}[\overline{X}]\right| > \varepsilon\right] \leq 2 \cdot \exp\left(\frac{-T^2\varepsilon^2}{2 \cdot \sum_{t=1}^{T} b_t^2}\right)$$

D.1 Iteration complexity of Algorithm 1.

Theorem (Restatement of Theorem 3.1). *Fixing a potential U and distribution $\mathcal{D}$, and $Z \subseteq \mathcal{Z}$, suppose $T \geq \Omega\left(\frac{\log(|Z|/\delta)}{\varepsilon^2}\right)$. Algorithm 1 produces unbiased estimates and with probability at least $1-\delta$, $|\nu(z; U, \mathcal{D}, m) - \nu_T(z)| \leq \varepsilon$. for all $z \in Z$.*

Proof. The theorem follows from the analysis of Theorem D.1, by taking the stability to be trivial, $\beta(k) = 1$, and using uniform sampling $w_k = 1/m$ for all $k \in [m]$. □

D.2 Running time analysis under stability.

Theorem D.1 (Generalizes Theorem 3.2). *Suppose U is $\beta(k)$-deletion stable and for all sets $S \subseteq \mathcal{Z}$ of cardinality $|S| = k$, $U(S)$ can be evaluated in time $R(k)$; consider a set of positive weights $\{w_k : k \in [m]\}$ and $p \in [0,1]$. Algorithm 2 produces unbiased estimates of $\nu(z; U, \mathcal{D}, m)$ that with probability at least $1-\delta$ are ε-accurate for all $z \in Z_p$ and runs in expected time*

$$RT_w(m) \leq O\left(p \cdot |Z| \cdot \frac{\log(|Z|/\delta)}{\varepsilon^2 m^2} \cdot \left(\sum_{k=1}^{m} \frac{\beta(k)^2}{w_k}\right) \cdot \left(\sum_{k=1}^{m} w_k \cdot R(k)\right)\right)$$

Proof. First, we bound the iteration complexity and time complexity to evaluate models at each iteration within Algorithm 2. The running time bound then follows by the fact that we need to evaluate a model per $z \in Z_p$, per iteration where the expected cardinality of $|Z_p| = p \cdot |Z|$.

Abusing notation, denote by

$$\Delta_z U(k) = \mathop{\mathbf{E}}_{S\sim\mathcal{D}^{k-1}} \left[U(S \cup \{z\}) - U(S)\right].$$

Suppose we sample k according to a possibly non-uniform discrete distribution where $\mathbf{Pr}[k \in m] = w_k$; we denote a random drawn from this distribution by $k \sim [m]_w$. Then, by sampling $k \sim [m]_w$, computing $\Delta_z U(S)$ for $S \sim \mathcal{D}^k$, and reweighting, we obtain an unbiased estimate of $\nu(z; U, \mathcal{D}, m)$.

$$\begin{aligned}
\nu(z; U, \mathcal{D}, m) &= \mathop{\mathbf{E}}_{k\sim[m]} \left[\Delta_z U(k)\right] \\
&= \frac{1}{m}\sum_{k=1}^{m} \Delta_z U(k) \\
&= \sum_{k=1}^{m} w_k \frac{\Delta_z U(k)}{w_k m} \\
&= \mathop{\mathbf{E}}_{k\sim[m]_w} \left[\frac{\Delta_z U(k)}{w_k m}\right]
\end{aligned}$$

For simplicity, we analyze a sampling scheme where we sample T_k sets with cardinality k for $T_k \geq w_k \cdot T$. (By the multiplicative Chernoff bound, this event will occur with high probability.) That is, for all $z \in Z$ and for each $k \in [m]$, we sample T_k subsets $S \sim \mathcal{D}^k$ and compute $\Delta_z U(S)$. For each $z \in Z$, each such sample is an independent unbiased estimate of $\Delta_z U(k)$, so reweighting according to w_k and averaging over $k \in [m]$ gives an unbiased estimate of $\nu(z; U, \mathcal{D}, m)$.

$$\nu_T(z) = \frac{1}{T}\sum_{k=1}^{m}\sum_{t=1}^{T_k} \frac{\Delta_z U(S_t)}{w_k m}$$

Note that by $\beta(k)$-deletion stability, for each k, the terms in the summation associated with $\Delta_z U(k)$ are bounded in magnitude by $\frac{\beta(k)}{w_k m}$. Thus, we can apply Hoeffding's inequality to derive the following bound to obtain ε-error with probability at least $1 - \delta_0$.

$$\begin{aligned}
\delta_0 &\geq 2 \cdot \exp\left(\frac{-\varepsilon^2 T^2}{2 \cdot \sum_{k=1}^{m}\sum_{t=1}^{T_k}\left(\frac{\beta(k)}{w_k m}\right)^2}\right) \\
&\geq 2 \cdot \exp\left(\frac{-\varepsilon^2 T^2}{\frac{2}{m^2} \cdot \sum_{k=1}^{m} w_k \cdot T \cdot \left(\frac{\beta(k)}{w_k}\right)^2}\right) \\
&= 2 \cdot \exp\left(\frac{-\varepsilon^2 m^2 T}{2 \cdot \sum_{k=1}^{m} \frac{\beta(k)^2}{w_k}}\right)
\end{aligned}$$

Thus, taking the failure probability $\delta_0 = \delta / |Z|$ small enough to union bound over all $z \in Z$, we derive the following bound on T.

$$T \geq \Omega\left(\frac{\log(|Z|/\delta)}{\varepsilon^2 m^2} \cdot \sum_{k=1}^{m} \frac{\beta(k)^2}{w_k}\right)$$

Using this bound on T, we can compute the necessary running time for Algorithm 2 in terms of $R(k)$ per $z \in Z_p$.

$$\begin{aligned} T_w(m) &= T \cdot \sum_{k=1}^{m} w_k \cdot R(k) \\ &= \frac{\log(|Z|/\delta)}{\varepsilon^2 m^2} \cdot \left(\sum_{k=1}^{m} \frac{\beta(k)^2}{w_k} \right) \cdot \left(\sum_{k=1}^{m} w_k \cdot R(k) \right) \end{aligned}$$

Thus, we can compare various sampling schemes for different stability factors. Note that in the case of the uniform sampling scheme, where $w_k = 1/m$ for all $k \in [m]$, the sampling probabilities cancel.

$$\begin{aligned} T_u(m) &= \frac{\log(|Z|/\delta)}{\varepsilon^2 m^2} \cdot \left(\sum_{k=1}^{m} \frac{\beta(k)^2}{1/m} \right) \cdot \left(\sum_{k=1}^{m} 1/m \cdot R(k) \right) \\ &= \frac{\log(|Z|/\delta)}{\varepsilon^2 m^2} \cdot \left(\sum_{k=1}^{m} \beta(k)^2 \right) \cdot \left(\sum_{k=1}^{m} R(k) \right) \end{aligned}$$

Thus, the overall running time is given as $RT_w(m) = p \cdot |Z| \cdot \frac{\log(|Z|/\delta)}{\varepsilon^2 m^2} \cdot \left(\sum_{k=1}^{m} \frac{\beta(k)^2}{w_k} \right) \cdot (\sum_{k=1}^{m} w_k \cdot R(k))$. □

Concretely, to see the special case of the theorem stated as Theorem 3.2, suppose that $\beta(k) = k^{-b}$ for $b \geq 1/2$ and $R(k) = k^c$ for $c \geq 1$. With these settings of the parameters, uniform sampling takes time

$$\begin{aligned} T_u(m) &= \frac{\log(|Z|/\delta)}{\varepsilon^2 m^2} \cdot \left(\sum_{k=1}^{m} k^{-2b} \right) \cdot \left(\sum_{k=1}^{m} k^{c} \right) \\ &= \frac{\log(|Z|/\delta)}{\varepsilon^2 m^2} \cdot O(\log(m)) \cdot O(m^{c+1}) \\ &\leq \frac{\log(|Z|/\delta)}{\varepsilon^2} \cdot \tilde{O}\left(m^{c-1}\right). \end{aligned}$$

Suppose, instead, we take $w_k \propto k^{1-2b}$; that is, we choose w_k such that the first summation will still be bounded by $H_m = \Theta(\log(m))$. Under such a sampling scheme, the running time is bounded as

$$\begin{aligned} T_w(m) &= \frac{\log(|Z|/\delta)}{\varepsilon^2 m^2} \cdot \left(\sum_{k=1}^{m} k^{-1} \right) \cdot \left(\sum_{k=1}^{m} k^{c+1-2b} \right) \\ &= \frac{\log(|Z|/\delta)}{\varepsilon^2 m^2} \cdot O(\log(m)) \cdot O(m^{c+2-2b}) \\ &\leq \frac{\log(|Z|/\delta)}{\varepsilon^2} \cdot \tilde{O}\left(m^{c-2b}\right). \end{aligned}$$

In other words, the biased sampling scheme allows us to save roughly a factor-m^{2b-1} in computation time. Thus, if U is $O(1/k)$-deletion stable, then the biased sampling scheme saves roughly a factor m in computation time.

D.3 Finite Sample Approximation to $\mathcal{D}$

While the analysis of Theorem D.1 focuses on the running time of Algorithm 2, we can equally interpret it as a sample complexity bound. In particular, taking $R(k) = k$ corresponds to the sample complexity of taking a fresh sample $S \sim \mathcal{D}^k$ per iteration. Thus, using Algorithm 2, the naive sample complexity given by resampling for each iteration gives the bound of

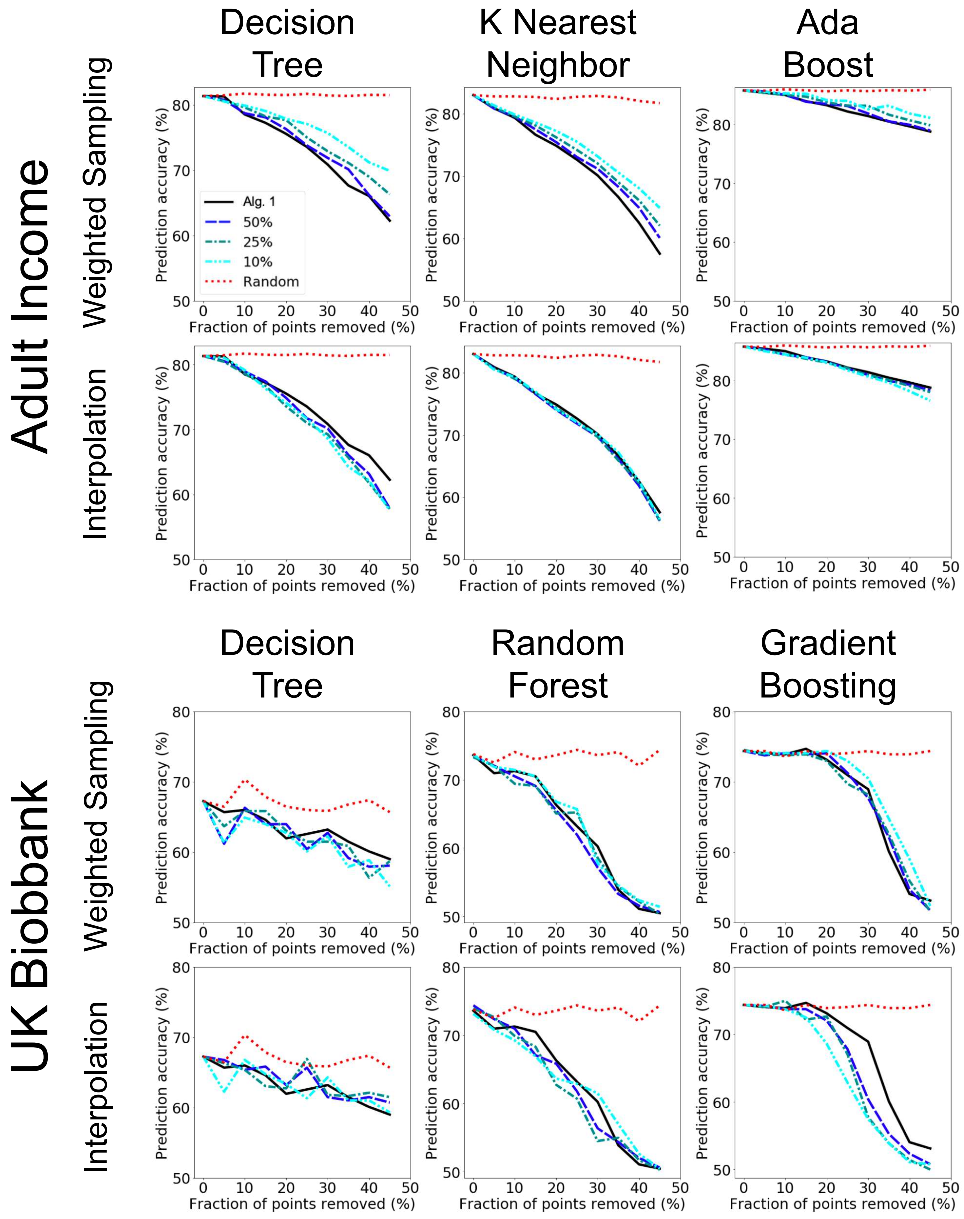
$$M \approx \frac{\log(|Z|/\delta)}{\varepsilon^2 m^2} \cdot \left(\sum_{k=1}^{m} \frac{\beta(k)^2}{w_k}\right) \cdot \left(\sum_{k=1}^{m} w_k \cdot k\right)$$

Taking $\beta(k) = w_k = 1/k$ yields

$$\begin{aligned} M &\approx \frac{\log(|Z|/\delta)}{\varepsilon^2 m} \cdot \left(\sum_{k=1}^{m} \frac{1}{k}\right) \\ &\approx \frac{\log(m) \cdot \log(|Z|/\delta)}{\varepsilon^2 m} \end{aligned}$$

E Additional Performance Experiments

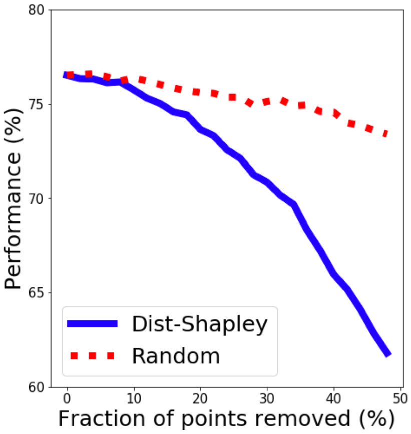
We report the results of additional empirical evaluations of Algoirthm 2, as described in Section 3.3. We depict the results of point removal experiments by plotting model performance vs. fraction of training data removed, where points are removed in order of their Shapley estimates or randomly. We apply the interpolation and weighted sampling speed-ups from Algorithm 2, independently. Both strategies obtain an order-of-magnitude speed-up with no qualitative performance change.



Supplementary Figure 1: Point removal curves for $\mathcal{D}$-Shapley estimates under various speed-ups, using weighted sampling and interpolation across two ML tasks and three learning algorithms.

E.1 Speeding-up Distributional Shapley for Cifar10

In this experiment, we apply both speed-up methods to compute the distributional Shapley values for CIFAR10 dataset. We apply weighted sampling corresponding to a speed-up factor of 10 and subsampling with interpolation corresponding to a speed-up factor of 50, to compute values for 1000 data points. We valuate points based on their effect on an image classification task. We use an Inception-v3 [SVI$^+$16] model, pretrained on the ILSVRC2012 (Imagenet) [RDS$^+$15] dataset; then, we base our potential function $U(S)$ on the performance of the model resulting after retraining the final layer of the network (holding all other layers fixed) using the retraining set S. Figure 2 shows the point-removal results for the complete dataset (e.g. removing 50% of the points with the highest $\mathcal{D}$-Shapley value causes the prediction accuracy to drop from 77% to 68%.)



Supplementary Figure 2: Point removal experiment for CIFAR10, using distributional Shapley estimates computed by Fast-$\mathcal{D}$-Shpaley.

F Additional Case Study Experiments

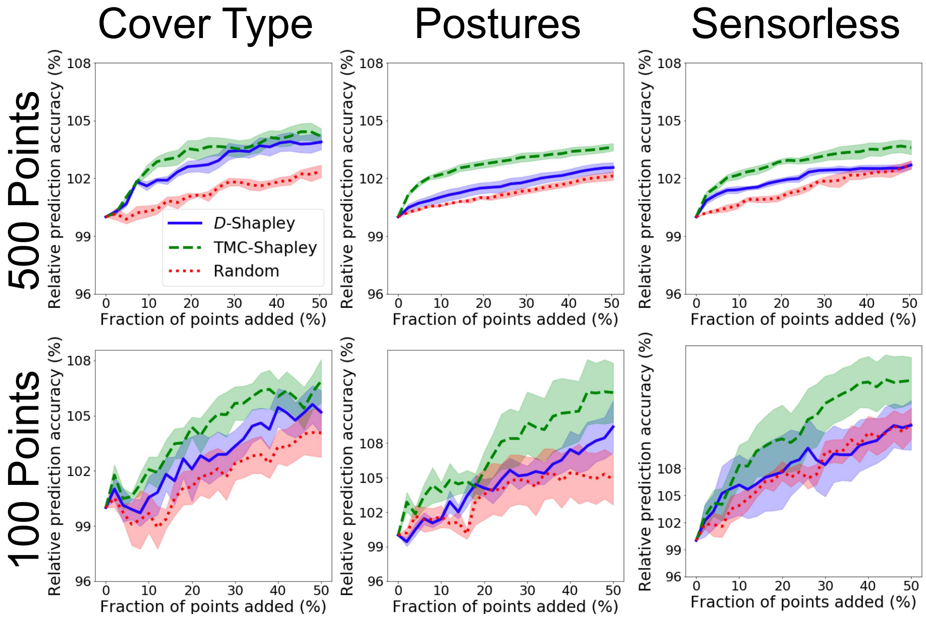
We report the complete results for the caze study from Section 4. We use four large-scale datasets from the UCI repository [DG17]:

(1) Covertype dataset with $581,012$ samples where each sample contains 54 visual features of forest images and the task is to detect the type of forest cover (from a set of 7 different covers). We use a Random Forest model.

(2) Diabetes130 dataset [SDG$^+$14] that with $100,000$ samples. Each sample contains 54 patient and hospital features. The task is to predict whether the patient will be readmitted to the hospital. We use an AdaBoost model.

(3) Wearable Computing: Classification of Body Postures and Movements (PUC-Rio) Data Set which has $165,632$ points where each point has 18 attributes and has one of the 5 postures. We use a multinomial logistic regression model.

(4) Dataset for Sensorless Drive Diagnosis Data Set that contains $58,509$ data points. Each data point has 48 features. The dataset has 11 classes. We use a Gradient Boosting model.



Supplementary Figure 3: Points from an acquired set are added to the buyer's initial dataset in three different orders: according to ν ($\mathcal{D}$-Shapley), according to ϕ (TMC), and randomly. The plot shows the change in the accuracy of the model, relative to its performance using the buyer's initial dataset, as the points are added; shading indicates standard error of the mean.