If You Like Shapley Then You'll Love the Core

Tom Yan,[1] **Ariel D. Procaccia** [2]
[1] Carnegie Mellon University
[2] Harvard University

Abstract

The prevalent approach to problems of credit assignment in machine learning — such as feature and data valuation — is to model the problem at hand as a cooperative game and apply the Shapley value. But cooperative game theory offers a rich menu of alternative solution concepts, which famously includes the core and its variants. Our goal is to challenge the machine learning community's current consensus around the Shapley value, and make a case for the core as a viable alternative. To that end, we prove that arbitrarily good approximations to the least core — a core relaxation that is always feasible — can be computed efficiently (but prove an impossibility for a more refined solution concept, the nucleolus). We also perform experiments that corroborate these theoretical results and shed light on settings where the least core may be preferable to the Shapley value.

1 Introduction

As machine learning systems become more capable, they are increasingly used in our society to automate tasks and generate value. This has lead to a surge in the attention given to explainability for machine learning: how features and data contribute to the performance of ML models. To ensure ML models are functioning as intended, much work has been devoted to studying *feature attribution*: how the features used to represent the data influence the model's predictions (Cohen, Dror, and Ruppin 2007; Štrumbelj and Kononenko 2010; Datta et al. 2015; Datta, Sen, and Zick 2016; Lundberg and Lee 2017; Chen et al. 2019). Related to feature attribution is *data valuation* (Ghorbani and Zou 2019; Jia et al. 2019a,b; Ohrimenko, Tople, and Tschiatschek 2019; Agarwal, Dahleh, and Sarkar 2019), which studies how data points contribute to model performance. With ML models now generating profit for enterprises, this understanding is important in order to fairly compensate data suppliers for their training data. Central to both pursuits is an equitable means of *credit assignment*.

Virtually all papers, including every single paper cited above, deem the Shapley value (or close variants thereof) to be the "right" way to carry out credit assignment. The Shapley value is a solution concept from cooperative game theory in which players — in this case features or data points — are assigned payoffs in a way that satisfies four axioms; roughly speaking, a player's payoff is their average marginal contribution to a coalition consisting of other players.

This intense focus on the Shapley value is surprising, however, as — once we have accepted that problems of credit assignment in machine learning can be modeled as cooperative games — there are a plethora of other solution concepts (Peleg and Sudhölter 2007). In particular, there is a seminal solution concept in cooperative game theory that is as prominent as the Shapley value: the core. This solution concept seeks to achieve maximal stability amongst all possible coalitions of the players in the game — an idea that dates back to the writings of Edgeworth on market equilibrium theory in 1881. Since then, it has found extensive applications in economics and beyond (Telser 1994).

Specifically, according to the core, the total payoff to each coalition should be at least its value. When this is not possible, the maximum deficit (difference between value and payoff) of any coalition should be minimized — this is known as the *least core*. The (least) core can be seen as a notion of *group fairness*, in that each group of players (or coalition) gets its dues. Moreover, it is especially apt in the valuation setting, where the data vendors or feature annotators are *paid* in a way that disincentivizes (to the extent possible) any coalition of vendors from choosing to opt out and not contribute; if a coalition S if not paid at least its value $v(S)$ then the coalition would be better off separating from the so-called grand coalition. Indeed, the core values may be seen as *the* set of all *economically plausible* payoffs to participants that compensate them for their contributions.

In this paper, we aim to show that the (least) core is, practically and conceptually, an attractive alternative to the Shapley value for credit assignment in machine learning. In doing so, we hope to raise awareness of the core as a natural solution concept for fair credit assignment, challenge the wide-ranging usage of the Shapley value and inspire a closer examination of cases where one solution concept should be preferred over the other. It is worth emphasizing that, to the best of our knowledge, we are the first to consider using the core for explainability of machine learning models.

1.1 Our Results

Much like the Shapley value, the primary obstacle in applying the concept of least core is computational complexity.

Indeed, it is the solution to a linear program whose number of constraints is exponential in the number of players. Nevertheless, we construct a Monte Carlo algorithm that runs in polynomial time and (with given confidence) outputs a payoff allocation in the *δ-probable least core* — a slightly relaxed version of the least core where the payoff constraints may be violated by up to a δ-fraction of coalitions. When the number of players is large, though, this may still be intractable; we therefore show that it is possible to find a solution in the *(ϵ, δ)-probably approximate least core* — whose constraints are additionally relaxed by ϵ each — in time that is polylogarithmic in the number of players.

We also study a well-known refinement of the least core called the *nucleolus*. However, it turns out that results that are analogous to those for the least core are essentially unattainable. Informally, we prove that *any* algorithm would have to require access to the values of an *exponentially* large number of coalitions to compute a payoff allocation in the (ϵ, δ)-probably approximate nucleolus, which again relaxes all relevant constraints by ϵ and allows a δ-fraction of the constraints to be violated. The juxtaposition of the positive computational results for the least core and the negative result for the nucleolus provides a strong endorsement of the former (somewhat coarser) notion over the latter.

In our experiments, we verify these theoretical results and confirm that our algorithm can compute the least core easily and that the nucleolus is difficult to compute. Next, we compare algorithms one would use to compute the Shapley value against our least core algorithm in data valuation tasks. Our results suggest that the least core algorithm compares favorably with those of the Shapley value in low-resource settings that are typical of analysts without access to large-scale computational resources.

1.2 Related Work

There is an entire area of algorithmic game theory devoted to the computation of solutions of cooperative games (Chalkiadakis, Elkind, and Wooldridge 2011). In particular, a slew of papers have studied the complexity of the core, the least core, and the nucleolus in specific classes of cooperative games (Deng and Papadimitriou 1994; Conitzer and Sandholm 2006; Bachrach and Rosenschein 2008; Elkind and Pasechnik 2009; Elkind et al. 2009).

Our work is most closely related to that of Balkanski et al. (2017). They study settings where solutions to cooperative games — specifically, the Shapley value and the core — are learned from samples consisting of coalitions and their values. Like Balcan et al. (2015), they are motivated by the observation that in classical applications of cooperative games values of coalitions cannot be accessed via queries; for example, if the game represents company employees working together to complete tasks, it is impossible to know which tasks would be completed had a specific coalition worked alone. Importantly, they do not consider explainability at all. Under the assumption that the underlying game has a nonempty core, Balkanski et al. (2017) give bounds on the sample complexity of three approximations of the core.

On a technical level, our definition of approximate notions of least core (Theorems 1 and 2) follow those of Balkanski et al. (2017) for the core, by eschewing the assumption that the core is nonempty; our proofs of these results directly build on theirs. Our interpretation of these results is quite different, though, because in our setting coalition values *can* be queried — for example, one can run a black-box predictor with a specific subset of features and measure its accuracy — so we think of our results as guarantees on the performance of Monte Carlo algorithms. Balkanski et al. (2017) did not study the nucleolus, so our negative result for the nucleolus (Theorem 3) — which we view as our main theoretical result — is entirely new and has no analog in their work. Finally, the work of Balkanski et al. (2017) is purely theoretical, whereas our empirical results study and demonstrate the applicability of the least core to credit assignment in machine learning.

2 Preliminaries

A *cooperative game* consists of a set of players $N = \{1, \ldots, n\}$ and a *characteristic function* $v : 2^N \to \mathbb{R}$ which assigns a value to each *coalition* $S \subseteq N$, such that $v(\emptyset) = 0$; we assume that $v(S) \geq 0$ and $v(S) \leq 1$ for all $S \subseteq N$ for ease of exposition. We think of $v(S)$ as the payoff the coalition S could obtain if it went it alone. Given such a game, we are interested in finding a *payoff allocation* (also known as an *imputation*) $\mathbf{x} = (x_1, \ldots, x_n)$, where x_i is the payoff of player $i \in N$. The payoff allocation must be *efficient*, that is,

$$\sum_{i \in N} x_i = v(N).$$

A payoff allocation is in the *e-core* if and only if the total payoff of each coalition is at least its value, up to e:

$$\forall S \subseteq N, \sum_{i \in S} x_i + e \geq v(S).$$

The core itself, by this definition, satisfies these constraints with $e = 0$. Unfortunately, there are cooperative games whose core is empty. But clearly the e-core is nonempty if e is large enough.

The idea behind the *least core* (Maschler, Peleg, and Shapley 1979) is to choose the smallest e possible. It may be defined as the set of all solutions to the following linear program.

$$\begin{array}{ll} \min & e \\ \text{s.t.} & \sum_{i \in N} x_i = v(N) \\ & \sum_{i \in S} x_i + e \geq v(S) \quad \forall S \subseteq N \end{array} \tag{1}$$

One can think of the least core as the set of payoff allocations that require the smallest subsidy $e^\star$ (the value of e in the optimal solution to (1)) to each coalition so that, if the payoff to each coalition was boosted by $e^\star$, the allocation would be in the core. The core is nonempty if and only if $e^\star \leq 0$.

We next consider a refinement of the least core, the *nucleolus*, first proposed by (Schmeidler 1969). Define the *deficit* of a payoff allocation $\mathbf{x}$ for a coalition $S \subseteq N$ to be $v(S) - \sum_{i \in S} x_i$. The nucleolus is the payoff allocation whose sorted list of deficits across all coalitions lexicographically dominates the list of deficits for any other payoff allocation. That is, the largest deficit (which will be positive

if the core is empty) should be as small as possible; subject to that, the second largest deficit should be as small as possible, and so on. Notice that, in particular, the nucleolus minimizes the largest deficit and so its allocation does lie in the least core. In contrast to the least core, which may contain multiple payoff allocations, the nucleolus is known to be unique (Schmeidler 1969).

3 Theoretical Results

Exact computation of the least core and the nucleolus requires solving linear programs with as many constraints as there are coalitions, which would typically be prohibitively expensive. Our strategy, therefore, is to sample a relatively small number of coalitions from an underlying distribution, and compute the desired solution concept on the sampled coalitions — this can be done in time that is polynomial in the number of samples, via the linear program (1) for the least core, and via a sequence of such linear programs for the nucleolus (Kopelowitz 1967). The hope is that this Monte Carlo algorithm would give us a payoff allocation that approximates the desired one with respect to the underlying distribution.

3.1 Computing the Least Core

We know from the work of Balkanski et al. (2017) that computing the least core exactly is a nonstarter — they prove an impossibility even for the core, under the assumption that it is nonempty. We therefore consider approximate versions of the least core.

Given a cooperative game, let $\mathcal{D}$ be a distribution over 2^N, and let $e^\star$ be the subsidy defined by the least core — the optimal solution to Equation (1). A payoff allocation $\mathbf{x}$ is in the *δ-probable least core* if and only if

$$\Pr_{S\sim\mathcal{D}}\left[\sum_{i\in S} x_i + e^\star \geq v(S)\right] \geq 1-\delta.$$

That is, the least core constraint is violated with probability at most δ when coalitions are drawn from $\mathcal{D}$.

We have the following result, whose proof appears in Appendix A.

Theorem 1. *Given a cooperative game (N, v), distribution $\mathcal{D}$ over 2^N, and $\delta, \Delta > 0$, solving the linear program (1) over $O((n + \log(1/\Delta))/\delta^2)$ coalitions sampled from $\mathcal{D}$ gives a payoff allocation in the δ-probable least core with probability at least $1 - \Delta$.*

It may seem surprising that solving the linear program (1) with respect to a subset of the coalitions gives a guarantee with respect to the unknown subsidy $e^\star$. But the estimated deficit $\hat{e}$ with respect to a subset of coalitions (that is, a subset of constraints) satisfies $\hat{e} \leq e^\star$ due to monotonicity.

Also note that the choice of $\mathcal{D}$ rests with the algorithm designer. In other words, we can sample coalitions from any distribution $\mathcal{D}$ and compute an allocation in the least core on the sample; the probable least core guarantee would then hold with respect to that same $\mathcal{D}$. In particular, if the uniform distribution over coalitions is used, the guarantee holds with respect to a $(1-\delta)$-fraction of all coalitions.

While Theorem 1 is encouraging, a potential drawback is that the algorithm's running time is polynomial in the number of players n. While this is an exponential improvement over naïve least core computation, it can still be a nonstarter when the players are features in a high-dimensional space or data points. We therefore define the *(ϵ, δ)-probably approximate least core* to be payoff allocations such that

$$\Pr_{S\sim\mathcal{D}}\left[\sum_{i\in S} x_i + e^\star + \epsilon \geq v(S)\right] \geq 1-\delta.$$

With this additional relaxation, we can obtain running time that is polynomial in $\log(n)$; the proof is relegated to Appendix B.

Theorem 2. *Given a cooperative game (N, v), distribution $\mathcal{D}$ over 2^N, and $\delta, \Delta, \epsilon > 0$, solving the linear program (1) over*

$$O\left(\frac{\tau^2\left(\log n + \log\left(\frac{1}{\Delta}\right)\right)}{\epsilon^2\delta^2}\right)$$

coalitions sampled from $\mathcal{D}$, where $\tau = \frac{\max_S v(S)}{\min_{S\neq\emptyset} v(S)}$, gives a payoff allocation in the (ϵ, δ)-probably approximate least core with probability at least $1 - \Delta$.

We note that τ may be considered a constant in general. For example, in multiclass classification it is no bigger than $\frac{1}{1/m} = m$, where m is the number of classes.

3.2 Computing the Nucleolus

The probably approximate least core can be seen as requiring the deficit of "most" coalitions to be approximately at most the maximum deficit $e^\star$ that defines the least core. In the (unique) nucleolus, though, that deficit is associated only with the worst-off coalition. It is natural to ask, instead, that the deficit of "most" coalitions be approximately their *own* deficit under the nucleolus allocation.

Formally, as before fix a cooperative game and a distribution $\mathcal{D}$. Denote by $d^\star(S)$ the deficit of coalition $S \subseteq N$ under the unique nucleolus allocation. A payoff allocation $\mathbf{x}$ is in the *(ϵ, δ)-probably approximate nucleolus* if and only if

$$\Pr_{S\sim\mathcal{D}}\left[\left|\sum_{i\in S} x_i + d^\star(S) - v(S)\right| \leq \epsilon\right] \geq 1-\delta.$$

Unfortunately, it turns out that any algorithm that computes the probably approximate nucleolus requires a number of samples that is *exponential* in the number of players n — a doubly exponential increase over the probably approximate least core! — as the following theorem shows.

Theorem 3. *Let $n \geq 9$, $\epsilon < 1/50$, $\delta < 1/200$ and $\Delta < 4/5$. Then any deterministic algorithm that for all games (N, v) on n players, and all distributions $\mathcal{D}$ on N, computes a payoff allocation in the (ϵ, δ)-probably approximate nucleolus with probability at least $1 - \Delta$ requires access to the values of $\Omega(2^{n/3})$ coalitions sampled from $\mathcal{D}$.*

The importance of Theorem 3 lies in the practical guidance it provides. Indeed, the stark contrast between Theorem 2 and 3 suggests that we should focus on approximations of the least core, as natural approximations of the

(stronger notion of) nucleolus are essentially beyond reach. Even though the theoretical result is worst-case in nature, we show in Section 5 that its implication holds in practice.

We also note that the theorem statement deals with algorithms that are deterministic, up to the random sampling of coalitions from $\mathcal{D}$. However, it is not difficult to extend the theorem to deal with randomized algorithms too, at the cost of complicating the proof further. Moreover, the constants in the theorem statement can certainly be improved, but we do not view their exact values as being important.

4 Interlude: A Comparison of the Core and the Shapley Value

Now that we have established that it is viable to compute the least core, we turn to the conceptual part of our argument. Before going into how the least core and the Shapley value differ (we include a comment on when the two are known to coincide in Section 5), one thing to note about the least core is that it is a set of solutions, whereas the Shapley value is a point solution concept. To compare the two conceptually (and experimentally as well), we break ties by selecting the payoff allocation in the least core with the smallest ℓ_2 norm. This is known as the *egalitarian least core*.

Axiomatic Properties. The Shapley value has almost always been justified through its four axiomatic properties (Cohen, Dror, and Ruppin 2007; Štrumbelj and Kononenko 2010; Datta, Sen, and Zick 2016; Lundberg and Lee 2017; Chen et al. 2019): (i) efficiency (ii) symmetry (iii) null player (iv) linearity. If we accept this argument, then the egalitarian least core is quite attractive in satisfying all but the last axiom (linearity).

While the least core's lack of linearity is ostensibly a disadvantage, it is unclear to us why it is an essential property for importance scores. The necessity of linearity is commonly justified by defining a cooperative game for each test point with the coalitional value being the model accuracy with respect to that point. And so, one would desire that summing the importance scores across these games would yield the score of the game corresponding to the entire test set. But one can simply define the latter game, with the coalitional value being the model accuracy with respect to the entire test set, in the very beginning, thus obviating the need for this property to hold.[1] Further questions regarding the usefulness of the linearity axiom are raised by Kumar et al. (2020), who highlight the uninformativeness of Shapley for explaining non-additive models.

By contrast, the stability axiom, which the egalitarian least core does satisfy, is crucial if we are to adopt the economic motivation behind data valuation, as described in data market papers such as that of Ghorbani and Zou (Ghorbani and Zou 2019). Put another way, if the goal is to output scores that reflect and may be *interpreted* as *economically plausible payments* in a competitive market, then the scores should be such that every coalition is compensated for at least its market value. This is so that the agents in the coalitions, who are rational, do not elect to leave the grand coalition. Contrast this with the Shapley value, which confers only a generic notion of "importance" (where relatively bigger means more "important") and may not necessarily correspond to an economically feasible set of payoffs (as we will see in the experiments).

Behavioral Studies. Studies in behavioral game theory have found the core to be predictive of payment distribution in market settings, suggesting that people perceive the core as a fair scheme for dividing up the total payoffs; by contrast, the Shapley value has received "weaker empirical support" (Williams 1988). This is an especially compelling reason to prefer the core over the Shapley value: since the *stakeholders* involved with machine learning are often people, it is imperative to employ a solution concept that is consistent with their behavior and intuition (Bhatt et al. 2019) (Kumar et al. 2020). Indeed, while much is still unclear as to how to assign "importance scores" in interpretability so as to truly aid stakeholders, there exists ample economic literature on how to equitably pay people and the core is one such prominent concept, which we champion as a principled way to assign these scores in the valuation setting.

Negative Computational Results for Shapley. Similar to our negative result for the nucleolus in Theorem 3, prior work has also produced negative results for the computation of the Shapley value. Indeed, the Shapley value is difficult to approximate, not to mention compute exactly. Informally, Bachrach et al. (Bachrach et al. 2010) show that no polynomial-time randomized algorithm can build a confidence interval with small accuracy. And Balkanski et al. (Balkanski, Syed, and Vassilvitskii 2017) show that there exist games such that the Shapley value cannot be approximated from samples over the uniform distribution.

In light of these negative results, the latest state of the art algorithms for computing the Shapley value (Ghorbani and Zou 2019; Jia et al. 2019a) either turn to simpler Monte-Carlo approaches that do not enjoy theoretical guarantees (e.g on convergence) (Ghorbani and Zou 2019) or more complicated algorithms that leverage assumptions such as sparsity to obtain sizable savings in sample complexity (Jia et al. 2019a). By contrast, we provide a simpler algorithm for computing the approximate least core with probable guarantees.

But do these theoretical results translate into practice? In the next section we show, among other things, that in low-resource settings (where the algorithm has limited computational power) our least core algorithm outperforms state-of-the-art algorithms for the Shapley value, thereby bolstering the computational case in favor of the least core.

5 Empirical Results

The purpose of this section is twofold. First, We empirically verify our theoretical conclusions about the computability of the least core and nucleolus (which are worst case in nature). Second, we compare the algorithms that one would use to approximate the Shapley value with that for least core.

[1] We do note that the core satisfies "approximate linearity" in the following sense: An e_1-core under coalition function v_1 and an e_2-core under coalition function function v_2 can be combined into an allocation that satisfies the $(e_1 + e_2)$-core under coalition function $v_1 + v_2$ (though certainly the least core could be better than just summing the least core allocations across the two games).

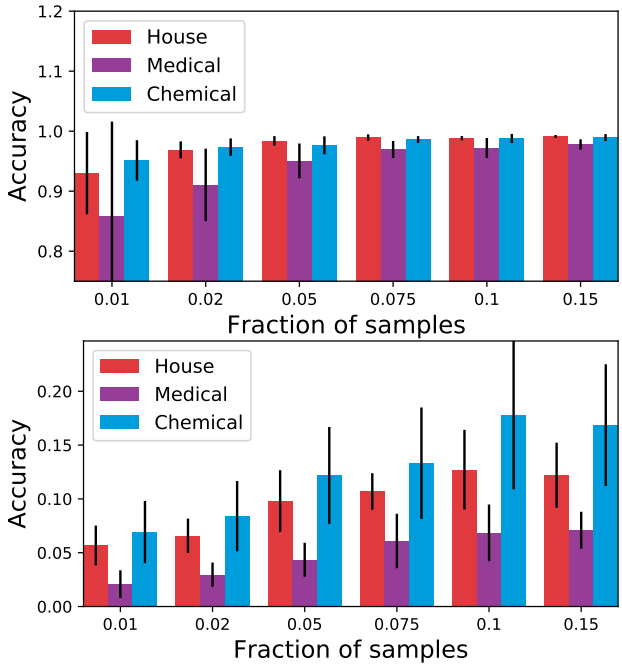


Figure 1: Top Panel: Least core accuracy (satisfaction of the core constraint) over coalitions. Bottom Panel: Nucleolus accuracy (satisfaction of the core constraint) over coalitions ($\epsilon = 0.01$).

Our experiments are conducted on feature valuation and data valuation tasks. Following previous work in the area, our primary aim is to use these tasks to confirm that least core values are predictive of importance, albeit in an indirect way (as the ultimate test of human-centered AI must be how the system interacts with people).

5.1 Feature Valuation

We choose three smaller-scale UCI datasets (Dua and Graff 2017) that have 10–14 features: this makes it computationally feasible to train a logistic regression classifier on all possible subsets of features and to compute the *exact* Shapley and least core values. To define the cooperative game, the players are the features and the value of a coalition is the test accuracy of a logistic regression classifier that is trained on those features. The three real-world datasets are of different domains: *house* (classifying the party of Congressmen based on their votes on issues), *medical* (predicting the presence of breast cancer based on features of images, and *chemical* (classifying the origin of wine based on chemical analysis).

To empirically verify Theorem 1 from Section 3 (which deals with the probable least core), we sample a small fraction of coalitions uniformly at random from all possible coalitions, and compute the least core by restricting Equation (1) to these coalitions. We then determine what fraction of all coalitions satisfy the least core constraints with respect to the true deficit $e^\star$ — that gives us *accuracy* $1-\delta$, which, in turn, leads to δ-probable least core. To obtain error bars, we repeat this ten times. As can be seen in Figure 1, even with a small fraction of sampled coalitions, the resultant allocations are δ-probable least core allocations with very small δ.

Theorem 3, by contrast, asserts that many samples are needed to compute the probably approximate nucleolus. Since this is a worst-case result, one may wonder whether it holds in practice. To check this, we apply the same methodology as above. As can be seen in Figure 1, even when a sizable fraction of samples are used to compute the (ϵ, δ)-probably approximate nucleolus , most coalitions do not satisfy its constraints.

5.2 Data Valuation

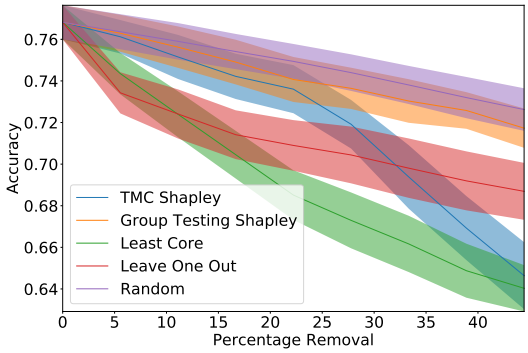
Our second set of experiments deals with data valuation. We focus on low-resource settings in which we assume the analyst who is looking to understand data importance has access to limited computational resources (e.g., few cores, no pun intended). We examine the performance of existing algorithms that one would use. To compare, we elect to fix the sample complexity (the number of $v(S)$ queries) that the algorithms are permitted to use. This sidesteps comparing the actual runtimes of the algorithms, which may vary depending on the details of the implementation. The two data valuation Shapley algorithms we compare against are TMC (Ghorbani and Zou 2019) and Group Testing (Jia et al. 2019a).

Data Removal. We emulate the data removal experiments as described in (Ghorbani and Zou 2019). In this set of experiments, the data is ranked from most valuable to the least valuable using the solution concepts, and the model performance is charted as the most valuable/least valuable five percent of the data is removed at a time. In addition to the two Shapley algorithms we also include two baselines: leave one out (LOO), defined as $v(N) - v(N \setminus \{i\})$ for each player i, and random score assignment.

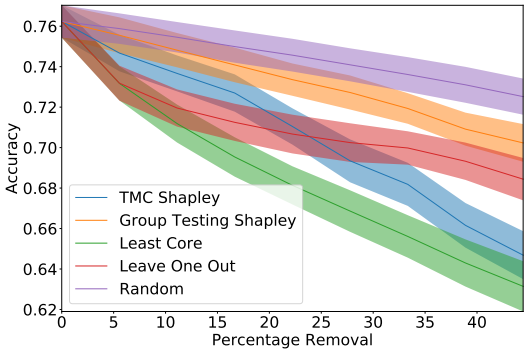
For the synthetic data generation, we sample 200 data points from 50-dimensional Gaussian, the 50-dimensional parameters are sampled from a uniform distribution and the feature-label relationship is set to be linear. To define the cooperative game, we take the players to be the data and the value of a coalition to be the test accuracy of the model trained only on the data in the coalition. The model used here is logistic regression; we relegate results for neural networks to Appendix E.2. We repeat the procedure 20 times and obtain 95 percent confidence intervals for the mean model performance.

For the natural dataset, we use the dog-vs-fish classification dataset as in the work of Koh and Liang (2017) and Ghorbani and Zou (2019). We randomly sample 600 data points and obtain features of the images using Inception network. The model used for training is logistic regression and we vary the budget as before. This entire process is repeated five times to obtain the error bars.

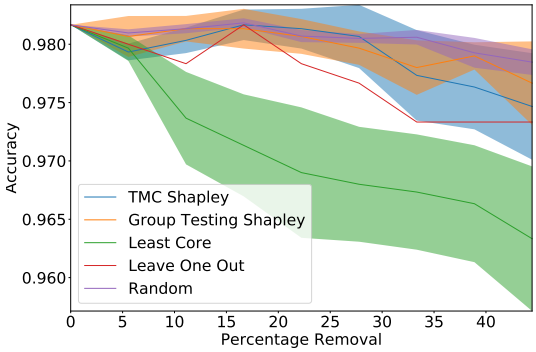
We experiment with a budget of $5K, 10K, 25K, 50K$ for samples as in a low-resource setting. As a point of reference, for the synthetic data experiment, computing the exact least core uses 2^{200} samples. The TMC Algorithm with a stopping threshold of less than one percent change in the estimated Shapley value uses $2.17M$ samples when run until conver-



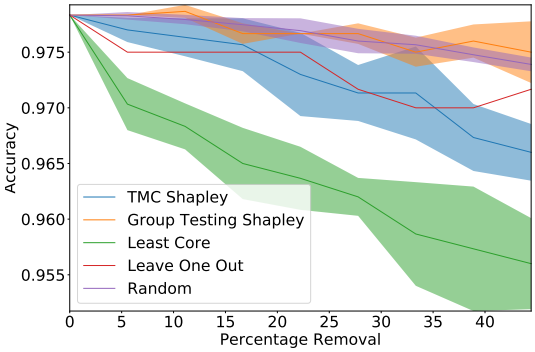
(a) Synthetic data, remove best, $10K$ samples



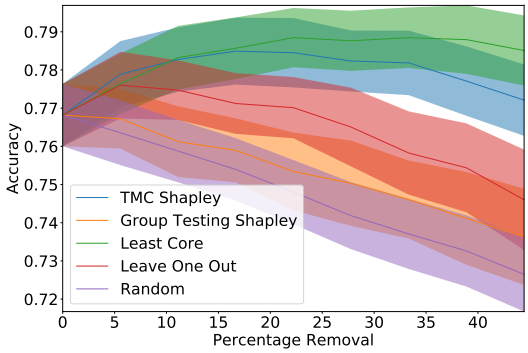
(b) Synthetic data, remove best, $50K$ samples



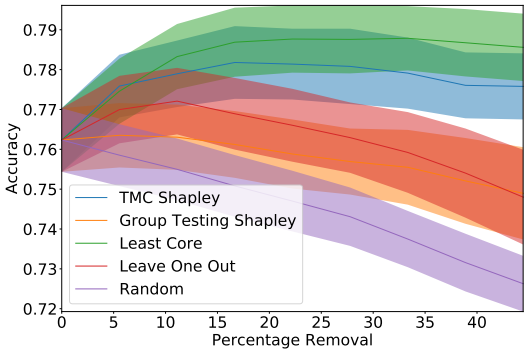
(c) Natural data, remove best, $10K$ samples



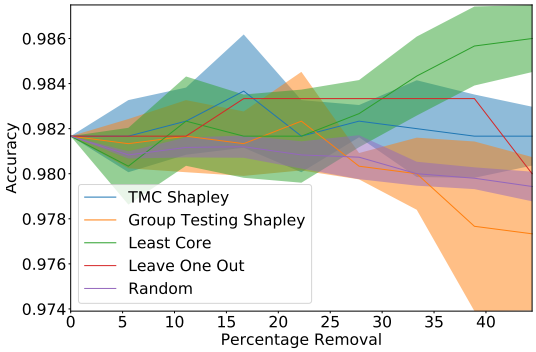
(d) Natural data, remove best, $50K$ samples



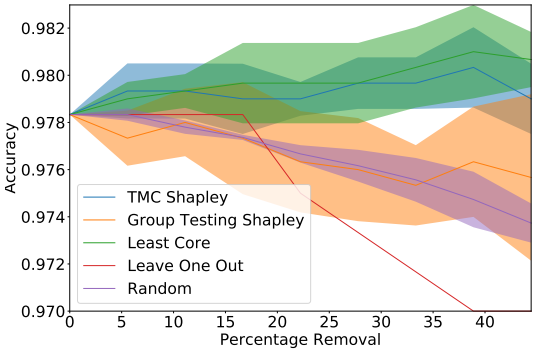
(e) Synthetic data, remove worst, $10K$ samples



(f) Synthetic data, remove worst, $50K$ samples



(g) Natural data, remove worst, $10K$ samples



(h) Natural data, remove worst, $50K$ samples

Figure 2: Curves of logistic regression test performance when the best and worst data points ranked according to the solution concepts are removed. In (a)–(d) the best data points are removed: the steeper the drop, the better. In (e)–(h) the worst data points are removed: the sharper the rise, the better.

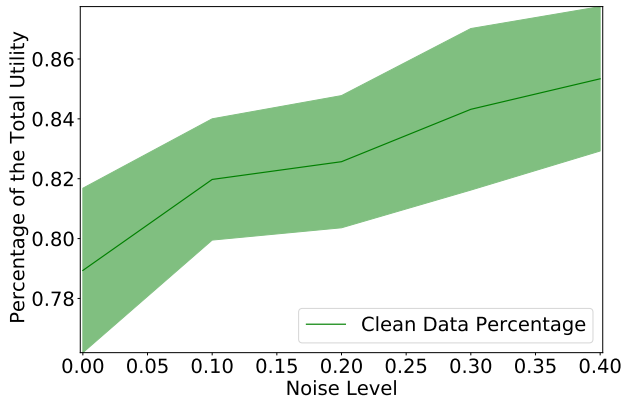


Figure 3: Plotting noise level against percentage of total utility assigned to clean data.

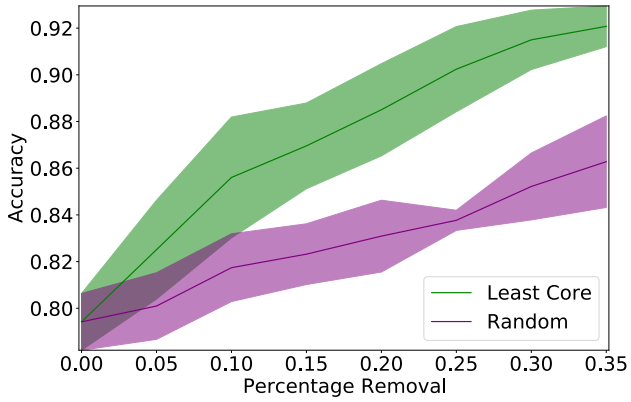


Figure 4: Test performance as we correct more and more training data guided by the least core vs. random selection.

gence. For the Group Testing Algorithm, using the sample complexity derived, running till convergence uses $11.05M$ samples.

As can be seen in Figure 2 (with similar figures for other parameter settings given in Appendix E.2), the least core algorithm compares favorably with the Shapley algorithms in terms of predicting the most and least important (in a sense) data points in these settings. Specifically, the least core's performance is significantly better than the baselines in the synthetic setting, whereas in the natural setting it is slightly better than Shapley value computation via the stronger of the two algorithms.

It is worth pointing out that the formulation of least core is such that it captures a group measure of value, whereas the Shapley value is more of an individual measure. Therefore, this data removal setup should *conceptually* favor Shapley, and yet the least core outperforms it to some degree.

As one more sanity check, we conduct an experiment studying the percentage of utility allocated by the core to noisy data. We divide the dataset into two: a clean portion and a noised portion. We increase the Gaussian noise added to the noised portion and compute the percentage of utility allocated by the core to the clean data. As expected and seen in Figure 3, with higher noise, the noised data become less "valuable" and are thus allocated a lower percentage of the overall utility by the core.

Fixing Mislabeled Data. We perform another set of experiments to verify that the magnitude of the least core values strongly correlate with the importance of the data point. In this experiment, we assume we have a dataset with flipped labels and would like to use the importance scores assigned to expedite the correction of "flipped" data points, which should correspond to the lower scores. The specific dataset we use is the Enron Dataset, as in previous work (Ghorbani and Zou 2019; Koh and Liang 2017). In total, 1000 data points are used for training a Naive Bayes model which takes as input a bag-of-words representation of emails. We randomly flip the label for twenty percent of the data and allot a budget of 5000 samples for computing the solution concepts. The coalitional values are defined as performance on the validation set, and then the final performance in the plot is assessed on the test set. As can be seen in Figure 4, the least core values are much better at picking out lower quality data points than random selection.

Is the Approximate Shapley value in the Approximate Least Core? It is known that the Shapley value coincides with the egalitarian core for convex games, where there is a super-additive effect in players coming together. This effect is not typically present in what we call "supervised-learning" games, in which there are diminishing returns as more and more data or features are added and used. However, in theory it may still be the case that the two solutions usually coincide, which would make it redundant to discuss the core. We therefore test, in the valuation experiments mentioned above, whether approximate Shapley values are close to being in the approximate least core. Our results suggest that this is not the case and therefore the approximate Shapley cannot serve as a proxy for the least core. Details are relegated to Appendix E.2

6 Conclusion

In our paper, we demonstrate that the least core can be approximated in a computationally tractable manner, and question the broad usage of the Shapley value with the hope of invoking further discussion on when and why one solution concept is to be preferred.

Our theoretical and empirical results, taken together with our conceptual arguments (Section 4), suggest that the least core is a principled alternative means of doing credit assignment in ML. Currently, it appears that virtually all papers on feature and data valuation use the Shapley value for this purpose. In light of the many uses of the core as an economically plausible method of payoff assignment, we introduce this alternative approach to the AI community in the hope that researchers and practitioners would take a closer look.

References

Agarwal, A.; Dahleh, M. A.; and Sarkar, T. 2019. A Marketplace for Data: An Algorithmic Solution. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, 701–726.

Bachrach, Y.; Markakis, E.; Resnick, E.; Procaccia, A. D.; Rosenschein, J. S.; and Saberi, A. 2010. Approximating power indices: theoretical and empirical analysis. *Autonomous Agents and Multi-Agent Systems* 20(2): 105–122.

Bachrach, Y.; and Rosenschein, J. S. 2008. Coalitional Skills Games. In *Proceedings of the 7th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 1023–1030.

Balcan, M.-F.; Procaccia, A. D.; and Zick, Y. 2015. Learning Cooperative Games. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 475–482.

Balkanski, E.; Syed, U.; and Vassilvitskii, S. 2017. Statistical Cost Sharing. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NeurIPS)*, 6221–6230.

Bhatt, U.; Xiang, A.; Sharma, S.; Weller, A.; Taly, A.; Jia, Y.; Ghosh, J.; Puri, R.; Moura, J. M. F.; and Eckersley, P. 2019. Explainable Machine Learning in Deployment. arXiv:1909.06342.

Chalkiadakis, G.; Elkind, E.; and Wooldridge, M. 2011. *Computational Aspects of Cooperative Game Theory*. Morgan & Claypool.

Chen, J.; Song, L.; Wainwright, M. J.; and Jordan, M. I. 2019. L-Shapley and C-Shapley: Efficient Model Interpretation for Structured Data. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*.

Cohen, S.; Dror, G.; and Ruppin, E. 2007. Feature Selection via Coalitional Game Theory. *Neural Computation* 19(7): 1939–1961.

Conitzer, V.; and Sandholm, T. 2006. Complexity of Constructing Solutions in the Core Based on Synergies Among Coalitions. *Artificial Intelligence* 170(6–7): 607–619.

Datta, A.; Datta, A.; Procaccia, A. D.; and Zick, Y. 2015. Influence in Classification via Cooperative Game Theory. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 511–517.

Datta, A.; Sen, S.; and Zick, Y. 2016. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In *Proceedings of the 37th IEEE Symposium on Security and Privacy (S&P)*, 598–617.

Deng, X.; and Papadimitriou, C. H. 1994. On the Complexity of Cooperative Solution Concepts. *Mathematics of Operations Research* 19(2): 257–266.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml.

Elkind, E.; Goldberg, L. A.; Goldberg, P. W.; and Wooldridge, M. J. 2009. On the Computational Complexity of Weighted Voting Games. *Annals of Mathematics and Artificial Intelligence* 56: 109–131.

Elkind, E.; and Pasechnik, D. B. 2009. Computing the Nucleolus of Weighted Voting Games. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 327–335.

Ghorbani, A.; and Zou, J. 2019. Data Shapley: Equitable Valuation of Data for Machine Learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2242–2251.

Jia, R.; Dao, D.; Wang, B.; Hubis, F. A.; Hynes, N.; Gürel, N. M.; Li, B.; Zhang, C.; Song, D.; and Spanos, C. 2019a. Towards Efficient Data Valuation Based on the Shapley Value. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 1167–1176.

Jia, R.; Dao, D.; Wang, B.; Hubis, F. A.; Hynes, N.; Gürel, N. M.; Li, B.; Zhang, C.; Spanos, C.; and Song, D. 2019b. Efficient Task-Specific Data Valuation for Nearest Neighbor Algorithms. *Proceedings of the VLDB Endowment* 12(11): 1610–1623.

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

Kopelowitz, A. 1967. Computation of the kernels of simple games and the nucleolus of N-person games. RM 31, Department of Mathematics, the Hebrew University of Jerusalem.

Kumar, I. E.; Venkatasubramanian, S.; Scheidegger, C.; and Friedler, S. 2020. Problems with Shapley-value-based explanations as feature importance measures. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*.

Lundberg, S. M.; and Lee, S.-I. 2017. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 4768–4777.

Maschler, M.; Peleg, B.; and Shapley, L. S. 1979. Geometric Properties of the Kernel, Nucleolus, and Related Solution Concepts. *Mathematics of Operations Research* 4(4): 303–338.

Ohrimenko, O.; Tople, S.; and Tschiatschek, S. 2019. Collaborative Machine Learning Markets with Data-Replication-Robust Payments. arXiv:1911.09052.

Peleg, B.; and Sudhölter, P. 2007. *Introduction to the Theory of Cooperative Games*. Springer, 2nd edition.

Schmeidler, D. 1969. The Nucleolus of a Characteristic Function Game. *SIAM Journal on Applied Mathematics* 17(6): 1163–1170.

Shalev-Shwartz, S.; and Ben-David, S. 2014. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Štrumbelj, E.; and Kononenko, I. 2010. An Efficient Explanation of Individual Classifications using Game Theory. *Journal of Machine Learning Research* 11: 1–18.

Telser, L. G. 1994. The Usefulness of Core Theory in Economics. *Journal of Economic Perspectives* 8(2): 151–164.

Williams, M. A. 1988. An empirical test of cooperative game solution concepts. *Behavioral Science* 33(3): 224–237.

A Proof of Theorem 1

This proof is a direct extension of the proof of Theorem 1 of Balkanski et al. (Balkanski, Syed, and Vassilvitskii 2017). Like them, we employ the following known lemmas (Shalev-Shwartz and Ben-David 2014).

Lemma 1. *Let $\mathcal{H}$ be a function class from $\mathcal{X}$ to $\{-1,1\}$, and let f be the true underlying function. If $\mathcal{H}$ has VC-dimension d, then with*

$$m = O\left(\frac{d + \log\left(\frac{1}{\Delta}\right)}{\delta^2}\right)$$

i.i.d. samples $\mathbf{x}^1, ..., \mathbf{x}^m \sim \mathcal{D}$,

$$\left|\Pr_{\mathbf{x}\sim D}[h(\mathbf{x}) \neq f(\mathbf{x})] - \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{h(\mathbf{x}^i)\neq f(\mathbf{x}^i)}\right| \leq \delta$$

for all $h \in \mathcal{H}$ and with probability $1-\Delta$ over the samples.

Lemma 2. *The function class $\{\mathbf{x} \mapsto sign(\mathbf{w}\cdot\mathbf{x}) :\ \mathbf{w} \in \mathbb{R}^n\}$ has VC-dimension n.*

We now turn to the proof. Given a coalition S sampled from $\mathcal{D}$, we convert it into a vector $\mathbf{y}^S = (\mathbf{x}^S, -v(S), 1)$ where $x_i^S = 1$ if $i \in S$ and $x_i^S = 0$ otherwise.

Consider a linear classifier h define by $\mathbf{w}^h = (\mathbf{z}, 1, e)$ where $\mathbf{z} \in \mathbb{R}^n$ and $e \in \mathbb{R}$. If $\text{sign}(\mathbf{w}^h \cdot \mathbf{y}^S) = 1$ then $\sum_{i\in S} z_i - V(S) + e \geq 0$. And if there exist a linear classifier h that satisfies this property for all coalitions $S \in 2^N$, and in addition $\mathbf{z}$ is efficient, then it represents a payoff allocation in the e-core. This allows us to define a class of functions that contains the e-core for all e. This class is:

$$\mathcal{H} = \left\{\mathbf{y} \mapsto \text{sign}(\mathbf{w}\cdot\mathbf{y}) :\ \mathbf{w} = (\mathbf{z}, 1, e), \mathbf{z} \in \mathbb{R}^n, e \in \mathbb{R}, \sum_{i=1}^{n} z_i = v(N)\right\}.$$

This class $\mathcal{H}$ is a subset of the class of all linear classifiers of dimension $n+2$ and thus, by Lemma 2, it has VC-dimension at most $n+2$.

Now, suppose that we run the linear program (1) on our samples $S_1, \ldots, S_m$, which gives us a payoff allocation $\hat{\mathbf{z}}$ and a value $\hat{e}$. Define the corresponding classifier $\hat{h}$; notice that $\hat{h}(\mathbf{y}^{S_i}) = 1$ for all $i = 1, \ldots, m$. In addition, let $\mathbf{z}^\star$ be a payoff allocation in the least core, and $e^\star$ the required subsidy, and define the corresponding classifier $f^\star$. It holds that $f^\star(\mathbf{y}^S) = 1$ for all $S \in 2^N$.

By Lemma 1 we have uniform convergence for all classifiers with probability $1-\Delta$, and in particular for $\hat{\mathbf{h}}$ it holds that

$$\begin{aligned}
\Pr_{S\sim\mathcal{D}}\left[\sum_{i\in S}\hat{z}_i - v(S) + e^\star \geq 0\right] &\geq \Pr_{S\sim\mathcal{D}}\left[\sum_{i\in S}\hat{z}_i - v(S) + \hat{e} \geq 0\right] \\
&= 1 - \Pr_{S\sim\mathcal{D}}\left[\text{sign}(\mathbf{w}^{\hat{h}}\cdot\mathbf{y}^S) = -1\right] \\
&= 1 - \Pr_{S\sim\mathcal{D}}\left[\hat{h}(\mathbf{y}^S) \neq f^\star(\mathbf{y}^S)\right] \\
&= 1 - \left(\Pr_{S\sim\mathcal{D}}\left[\hat{h}(\mathbf{y}^S) \neq f^\star(\mathbf{y}^S)\right] - \frac{1}{m}\sum_{i=1}^{m}\mathbb{1}_{\hat{h}(\mathbf{y}^{S_i})\neq f^\star(\mathbf{y}^{S_i})}\right) \\
&\geq 1 - \delta
\end{aligned}$$

where the first transition holds because $\hat{e} \leq e^\star$ and the fourth transition holds because $\hat{h}$ and $f^\star$ agree on $S_1, \ldots, S_m$. □

B Proof of Theorem 2

This proof directly extends the proof of Theorem 5 of Balkanski et al. (Balkanski, Syed, and Vassilvitskii 2017). Like them, we use the following result (Shalev-Shwartz and Ben-David 2014).

Lemma 3. *Let $\mathcal{H} = \{\mathbf{w} : ||\mathbf{w}||_1 \leq B\}$ be the hypothesis class, and $\mathcal{Z} = \mathcal{X}\times\mathcal{Y}$ be the examples domain. Suppose $\mathcal{D}_Z$ is a distribution over $\mathcal{Z}$ s.t $||\mathbf{x}||_\infty \leq R$. Let the loss function $\ell : \mathcal{H}\times\mathcal{Z} \to \mathbb{R}$ be of the form $\ell(\mathbf{w}, (\mathbf{x}, y)) = \phi(\langle\mathbf{w}, \mathbf{x}\rangle, y)$ and $\phi : \mathbb{R}\times\mathcal{Y} \to \mathbb{R}$ is such that for all $y \in \mathcal{Y}$, the scalar function $a \to \phi(a, y)$ is ρ-Lipschitz and such that $\max_{a\in[-BR,BR]}|\phi(a, y)| \leq c$. Then for any $\Delta \in (0,1)$, with probability of at least $1-\Delta$ over the choice of an iid sample of size m, $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m)$:*

$$\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_Z}[\ell(\mathbf{w}, (\mathbf{x}, y))] \leq \frac{1}{m}\sum_{i=1}^{m}\ell(\mathbf{w}, (\mathbf{x}^i, y^i)) + 2\rho BR\sqrt{\frac{2\log(2d)}{m}} + c\sqrt{\frac{2\log(2/\Delta)}{m}}.$$

for all $\mathbf{w} \in \mathcal{H}$.

We also require the observation that if an (ϵ, δ)-probably approximate least core holds in expectation, then it is likely to hold.

Lemma 4. *For any $\epsilon > 0$, $\delta < 1$ and e-core allocation $\mathbf{x}$ computed from samples,*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}}\left[\left[1 - \frac{\sum_{i\in S} z_i + e}{v(S)}\right]_+\right] \leq \frac{\epsilon\delta}{1+\epsilon} \Rightarrow \Pr_{S\sim\mathcal{D}}\left[\sum_{i\in S} z_i + e^\star + \epsilon \geq v(S)\right] \geq 1 - \delta.$$

Proof. Recall Markov's inequality: for $a > 0$, random variable $X \geq 0$,

$$\Pr[X \leq a] \geq 1 - \frac{\mathbb{E}[X]}{a}.$$

To use it, let $a = \frac{\epsilon}{1+\epsilon}$ and define a nonnegative random variable

$$X = \left[1 - \frac{\sum_{i\in S} z_i + e}{v(S)}\right]_+.$$

Then event $X \leq a$ is such that

$$\begin{aligned}
X \leq a &\Leftrightarrow 1 - \frac{\sum_{i\in S} z_i + e}{v(S)} \leq \frac{\epsilon}{1+\epsilon} \\
&\Leftrightarrow \sum_{i\in S} z_i + e \geq \frac{1}{1+\epsilon} v(S) \\
&\Leftrightarrow \sum_{i\in S} z_i + e + \frac{\epsilon}{1+\epsilon} v(S) \geq v(S) \\
&\Rightarrow \sum_{i\in S} z_i + e + \epsilon \geq v(S) \\
&\Rightarrow \sum_{i\in S} z_i + e^\star + \epsilon \geq v(S)
\end{aligned}$$

where the penultimate step uses $v(S) \leq 1$ for all $S \subseteq N$, and the last step uses that $e^\star \geq e$ since e is the least core value obtained from only a sample of all coalitional constraints.

We conclude that

$$\Pr\left[\sum_{i\in S} z_i + e^\star + \epsilon \geq v(S)\right] \geq \Pr[X \leq a] \geq 1 - \frac{\mathbb{E}[X]}{a} \geq 1 - \frac{\delta a}{a} = 1 - \delta.$$

□

Turning to the theorem's proof, in order to use Lemma 3, we begin by bounding the L_1 norm of every allocation and e in the e-core to obtain B.

Suppose $\mathbf{z}$ is an allocation in the e-core, then $||(\mathbf{z}, e)||_1 = v(N) + e$. This holds because $z_i \geq 0$ for all $i \in N$ and, by efficiency, $||\mathbf{z}||_1 = v(N)$. Therefore:

$$||(\mathbf{z}, e)||_1 = v(N) + e \leq v(N) + \max_S v(S) \leq 2 \max_S v(S)$$

Then, we can take our hypothesis class to be:

$$\mathcal{H} = \left\{\mathbf{z} \in \mathbb{R}^n : \ ||\mathbf{z}||_1 \leq 2 \max_S v(S)\right\}$$

Given $S \sim \mathcal{D}$, define the corresponding $\mathbf{x}^S = (\frac{\mathbb{1}_{i\in S}}{v(S)}, \frac{1}{v(S)})$ and the label to be $y^S = 1$. This allows us define to $\mathcal{D}_Z$ to be the uniform distribution over all $(\mathbf{x}^S, y^S)$ pairs. Next, suppose we obtain m samples $S_1, \ldots, S_m$ from $\mathcal{D}$, the uniform distribution over all coalitions, we may again run the linear program (1) on the m samples, which gives us a payoff allocation $\hat{\mathbf{z}}$ and a value $\hat{e}$. We take our classifier to be of the form $\mathbf{w} = (\hat{\mathbf{z}}, \hat{e})$ and we may define its loss ℓ to be:

$$\begin{aligned}
\ell(\mathbf{w}, (\mathbf{x}^S, y^S)) &= \ell\left((\hat{\mathbf{z}}, \hat{e}), \left(\left(\frac{\mathbb{1}_{i\in S}}{v(S)}, \frac{1}{v(S)}\right), y^S\right)\right) \\
&= \left[y^S - (\hat{\mathbf{z}}, \hat{e}) \cdot \left(\frac{\mathbb{1}_{i\in S}}{v(S)}, \frac{1}{v(S)}\right)\right]_+ \\
&= \left[1 - \frac{\sum_{i\in S} \hat{z}_i + \hat{e}}{v(S)}\right]_+.
\end{aligned} \tag{2}$$

Now, we may utilize Lemma 3 with the remaining variables being $R = \frac{1}{\min_{S\neq\emptyset} v(S)}$, $B = 2\max_S v(S)$, $\phi(a,y) = [y-a]_+$, $\rho = 1$ and $c = 1+2\tau$. This is legal because, ignoring the empty set, by definition of $\mathbf{x}^S$, $||\mathbf{x}^S||_\infty \leq \frac{1}{\min_{S\neq\emptyset} v(S)}$. By definition of the hypothesis class, $||(\mathbf{z},e)||_1 \leq 2\max_S v(S)$ for all $(\mathbf{z},e) \in \mathcal{H}$. $\phi(a,y) = [y-a]_+$ is 1-Lipschitz as:

$$\begin{aligned}
[y-a_1]_+ - [y-a_2]_+ &= \max\{y-a_1,0\} - \max\{y-a_2,0\} \\
&= \frac{|y-a_1|+y-a_1}{2} - \frac{|y-a_2|+y-a_2}{2} \\
&= \frac{|y-a_1|-|y-a_2|+a_2-a_1}{2} \\
&\leq \frac{|y-a_1-(y-a_2)|+a_2-a_1}{2} \\
&\leq |a_2-a_1|
\end{aligned}$$

Lastly, because our example domain $\mathcal{Z}$ is such that $\mathcal{Y} = \{1\}$. We may obtain upper bound c:
$c = \max_{a\in[-BR,BR]} |\phi(a,y)| = \max_{a\in[-BR,BR]}[1-a]_+ \leq (1--BR) = 1+BR = 1+2\tau$.
Moreover, since for all S_t in our sample it holds that $\sum_{i\in S_t} \hat{z}_i + \hat{e} \geq v(S)$, Equation (2) implies that

$$\frac{1}{m}\sum_{t=1}^{m} \ell\left((\hat{\mathbf{z}},\hat{e}), \left(\left(\mathbf{x}^{S_t}, \frac{1}{v(S_t)}\right), 1\right)\right) = 0.$$

Therefore by Lemma 3,

$$\begin{aligned}
\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[l(\mathbf{w},(\mathbf{x},y))] &= \mathbb{E}_{S\sim D}\left[\left[1 - \frac{\sum_{i\in S}\hat{\mathbf{z}}_i + \hat{e}}{v(S)}\right]_+\right] \\
&\leq 0 + 2\cdot 1\cdot 2\tau\sqrt{\frac{2\log(2(n+1))}{m}} + (1+2\tau)\sqrt{\frac{2\log(2/\Delta)}{m}}
\end{aligned} \tag{3}$$

Using Lemma 4, we need the number of samples m to be such that

$$4\tau\sqrt{\frac{2\log(2(n+1))}{m}} + (1+2\tau)\sqrt{\frac{2\log(2/\Delta)}{m}} \leq \frac{\delta\epsilon}{1+\epsilon},$$

and we get that

$$O\left(\frac{\tau^2\left(\log n + \log\left(\frac{1}{\Delta}\right)\right)}{\epsilon^2\delta^2}\right)$$

samples suffice. □

C Proof of Theorem 3

On a high level, we will construct a set of cooperative games $\mathcal{G}$ over the same set of players N, and a distribution $\mathcal{D}$ over the coalitions, such that no deterministic algorithm can compute a payoff allocation in the (ϵ,δ)-approximate nucleolus with probability $1-\Delta$ using $m \leq \frac{1}{6}\cdot 2^{n/3+1}$ samples with respect to *every* game in $\mathcal{G}$.

The idea of the proof is as follows. We construct the class of games $\mathcal{G}$ in a way that it is likely to observe $v(S_i) = 0$ for the coalitions $S_1,\ldots,S_m$ sampled from $\mathcal{D}$. Lemma 5 shows that at least half of the games in our class are consistent with such an observation. But Lemma 7 asserts that *any* payoff allocation would be in the (ϵ,δ)-probably approximate nucleolus of only a small fraction of the games in $\mathcal{G}$. Intuitively, then, when such an input is observed, the algorithm does not have enough information about the underlying game and is likely to violate the (ϵ,δ)-probably approximate nucleolus requirement. In the theorem's proof itself, we formalize this intuition by first assuming that the game itself is drawn from a uniform distribution over $\mathcal{G}$; the theorem statement follows from an averaging argument.

Formally, the class of games $\mathcal{G}$ is defined as follows. Let N be a set of n players; we assume without loss of generality that n is divisible by 3. Let C_1 be a set of 3 players $\{i,j,k\}$. Define C_2, C_3, C_4 to be sets of $n/3-1$ players such that $C_1\cup C_2\cup C_3\cup C_4 = N$. Each cooperative game G_{C_1,C_2,C_3,C_4} in our class $\mathcal{G}$ is such that $v(S) = 1$ if $\{i,j\}\cup C_2 \subseteq S$ or $\{i,k\}\cup C_3 \subseteq S$ or $\{j,k\}\cup C_4 \subseteq S$; $v(S) = 0$ otherwise. The important thing to note is that all coalitions of size $n/3+1$ have value 0, except for exactly three that have value 1: $\{i,j\}\cup C_2$, $\{i,k\}\cup C_3$, and $\{j,k\}\cup C_4$. We call C_1 the *critical set* of game G_{C_1,C_2,C_3,C_4}.

Next, we define the distribution $\mathcal{D}$ to be the uniform distribution over all coalitions of size $n/3+1$.

Lemma 5. *For any m coalitions $S_1, \ldots, S_m$ of size $n/3+1$, at least half of the games in $\mathcal{G}$ satisfy $v(S_i) = 0$ for all $i = 1, \ldots, m$.*

Proof. To count the number of such games, we can count the number of games in which the value of S_i is 1. By symmetry, the number of games in which a coalition S has value 1 is the same for all coalitions S of size $n/3+1$. Moreover, for each game in $\mathcal{G}$ there are three coalitions of size $n/3+1$ with value 1. Therefore, for each S_i, the number of games in $\mathcal{G}$ with $v(S_i) = 1$ is $3|\mathcal{G}|/\binom{n}{n/3+1}$. It follows that the number of games for which it does *not* hold that $v(S_i) = 0$ for all $i = 1, \ldots, m$ is at most $3m|\mathcal{G}|/\binom{n}{n/3+1}$. Since $\binom{n}{n/3+1} \geq 2^{n/3+1}$, by our choice of m this is at most $|\mathcal{G}|/2$. □

We next characterize the nucleolus of games in $\mathcal{G}$.

Lemma 6. *For every game $G_{C_1,C_2,C_3,C_4} \in \mathcal{G}$ and every $S \subseteq N$,*

$$d^*(S) = \begin{cases} 1/3 & S \in \{\{i,j\} \cup C_2, \{i,k\} \cup C_3, \\ & \quad \{j,k\} \cup C_4\} \\ -\frac{|S \cap \{i,j,k\}|}{3} & \textit{otherwise} \end{cases}$$

Proof. Let us compute the least core first since we know the nucleolus lies within it. Summing the constraints of linear program (1) for the coalitions $\{i,j\} \cup C_2$, $\{i,k\} \cup C_3$, $\{j,k\} \cup C_4$, we get that

$$\sum_{t \in N} x_t + (x_i + x_j + x_k) \geq 3 - 3e.$$

Since $1 = \sum_{t \in N} x_t \geq x_i + x_j + x_k$, we have that $2 \geq 3 - 3e$, and hence $e \geq 1/3$. Moreover, $e = 1/3$ is achieved if $x_i = x_j = x_k = 1/3$.

We claim that this payoff allocation is the only one that achieves $e = 1/3$. Indeed, the total payoff to each of the coalitions $\{i,j\} \cup C_2$, $\{i,k\} \cup C_3$, $\{j,k\} \cup C_4$ must be at least $2/3$, which means that the payoff of players at the intersection of each pair of these coalitions must be at least $1/3$. But the intersection of each pair is exactly one of the players i, j, k.

Since the payoff allocation $\mathbf{x}$ is the unique solution to the least core program, it must be the nucleolus. The statement of the lemma directly follows. □

Lemma 6 implies that two games G_{C_1,C_2,C_3,C_4} and $G_{C_1',C_2',C_3',C_4'}$ have the same nucleolus if and only if $C_1 = C_1'$. Let us, therefore, partition $\mathcal{G}$ into *equivalence classes*, where the games in an equivalence class have the same critical set.

Lemma 7. *Any payoff allocation is in the (ϵ, δ)-probably approximate nucleolus for games from at most one equivalence class.*

Proof. Let $\mathbf{x}$ be a payoff allocation. We consider two cases, based on the number of players $i \in N$ with $x_i > \epsilon$.

Case 1: There are at least three players with $x_i > \epsilon$. Let those three players be $\{i,j,k\}$, and consider a game in $\mathcal{G}$ whose critical set is not $\{i,j,k\}$. Then there exists a player ℓ not in the critical set such that $x_\ell > \epsilon$.

Consider all coalitions of size $n/3+1$ containing ℓ but no player from the critical set. By Lemma 6, under the nucleolus of the game, all such coalitions have deficit 0, but under $\mathbf{x}$ they would have a deficit of at at most $-x_{a'} < -\epsilon$. There are $\binom{n-4}{n/3}$ such coalitions, which accounts for the following portion of all coalitions of size $n/3+1$:

$$\frac{\binom{n-4}{n/3}}{\binom{n}{n/3+1}} = \frac{(n/3+1)(2n/3-1)(2n/3-2)(2n/3-3)}{n(n-1)(n-2)(n-3)}$$
$$> (1/3 \cdot 1/2 \cdot 1/2 \cdot 1/2) = \frac{1}{24} \geq \delta.$$

Case 2: There are less than three players with $x_i > \epsilon$.

In this case, for any game in $\mathcal{G}$, $\mathbf{x}$ is such that there exists at least one player in its critical set with allocation at most ϵ. We show that this means $\mathbf{x}$ cannot satisfy the (ϵ, δ)-probably approximate nucleolus property with respect to the game.

Fix a game in $\mathcal{G}$, let the critical set of the game be $\{i,j,k\}$, and let $x_i \leq \epsilon$. Assume for the sake of contradiction that $\mathbf{x}$ satisfies the (ϵ, δ)-probably approximate nucleolus property for this game.

Consider the set of all coalitions of size $n/3+1$ that contain i, j but not k. There are $\binom{n-3}{n/3-1}$ such coalitions. We know by Lemma 6 that all but one of these coalitions have value 0 and deficit $-2/3$. In order for the property

$$\left| \sum_{i \in S} x_i + d^\star(S) - v(S) \right| \leq \epsilon \tag{4}$$

to hold for such coalitions, we would need their payoff to be at least $2/3 - \epsilon$.

Overall, there are at least $\binom{n-3}{n/3-1} - \delta\binom{n}{n/3+1} - 1$ many coalitions containing i, j but not k for which Equation (4) applies and have value 0. The middle term comes from factoring in that at most a δ fraction of all $\binom{n}{n/3+1}$ coalitions will not satisfy the probably approximate nucleolus property. By summing over the total payoffs of all such coalitions we have

$$\begin{aligned}&\binom{n-3}{n/3-1}(x_i + x_j) + \binom{n-4}{n/3-2}\left(\sum_{t\notin\{i,j,k\}} x_t\right)\\ &\geq \left(\binom{n-3}{n/3-1} - \delta\binom{n}{n/3+1} - 1\right)(2/3-\epsilon)\end{aligned}$$

since each player that is not i, j or k shows up $\binom{n-4}{n/3-2}$ times. Dividing by $\binom{n-3}{n/3-1}$ and using the fact that $\binom{n-4}{n/3-2}/\binom{n-3}{n/3-1} = 1/3$, we have

$$\begin{aligned}&x_i + x_j + \frac{1}{3}\left(\sum_{t\notin\{i,j,k\}} x_t\right)\\ &\geq \left(1 - \frac{\binom{n}{n/3+1}}{\binom{n-3}{n/3-1}}\cdot\delta - \frac{1}{\binom{n-3}{n/3-1}}\right)(2/3-\epsilon).\end{aligned}$$

With $n \geq 9$, $\frac{1}{\binom{n-3}{n/3-1}} \leq \frac{1}{15}$ and so we obtain

$$x_i + x_j + \frac{1}{3}\left(\sum_{t\notin\{i,j,k\}} x_t\right) \geq \left(\frac{14}{15} - \frac{\binom{n}{n/3+1}}{\binom{n-3}{n/3-1}}\delta\right)(2/3-\epsilon).$$

Using efficiency, $\sum_{t\notin\{i,j,k\}} x_t = 1 - x_i - x_j - x_k$, and using the fact that

$$\frac{\binom{n}{n/3+1}}{\binom{n-3}{n/3-1}} = \frac{n(n-1)(n-2)}{(n/3+1)(n/3)(2n/3-1)} \leq 27$$

we get

$$\frac{2}{3}x_i + \frac{2}{3}x_j + \frac{1}{3} - \frac{1}{3}x_k \geq \left(\frac{14}{15} - 27\delta\right)(2/3-\epsilon).$$

Similarly, by considering the set of all coalitions that contain i, k but not j, we see that

$$\frac{2}{3}x_i + \frac{2}{3}x_k + \frac{1}{3} - \frac{1}{3}x_j \geq \left(\frac{14}{15} - 27\delta\right)(2/3-\epsilon).$$

Summing both inequalities, we conclude that

$$\frac{4}{3}x_i + \frac{1}{3}(x_j + x_k) + \frac{2}{3} \geq \frac{4}{3}\cdot\frac{14}{15} - 36\delta - \frac{28}{15}\cdot\epsilon + 54\delta\epsilon.$$

Since $x_j + x_k \leq 1$,

$$\frac{4}{3}x_i \geq \frac{11}{45} - 36\delta - \frac{28}{15}\epsilon + 54\delta\epsilon,$$

which is impossible for $x_i \leq \epsilon$ since $\epsilon < 1/50$ and $\delta < 1/200$. □

We are now ready to prove the theorem.

Proof of Theorem 3. Fix the set of players N. Let $\mathcal{U}$ be the uniform distribution over games in $\mathcal{G}$. Since N is fixed, we think of $\mathcal{U}$ as a distribution over characteristic functions and write $v \sim \mathcal{U}$.

Suppose that we draw coalitions $S_1, \ldots, S_m$ from $\mathcal{D}$, and v from $\mathcal{U}$. Let $\mathcal{A}((S_1, v(S_1)), \ldots, (S_m, v(S_m)))$ be the payoff allocation returned by the given algorithm $\mathcal{A}$ on this input. Consider the event $\mathcal{E}$ that occurs when $\mathcal{A}((S_1, v(S_1)), \ldots, (S_m, v(S_m))$ is in the (ϵ, δ)-probably approximate nucleolus of the game (N, v). We wish to upper-bound the probability of $\mathcal{E}$.

To this end, instead of drawing v from $\mathcal{U}$ directly, it will be useful to use the following generative process. First, decide whether it holds that $v(S_i) = 0$ for all $i = 1, \ldots, m$; call this event $\mathcal{F}$. If $\mathcal{F}$ occurred, condition $\mathcal{U}$ on $\mathcal{F}$ and draw v from this posterior distribution. As we will see shortly, there is no need to explicitly define the process for the case where $\mathcal{F}$ did not occur.

Denoting the complement of $\mathcal{F}$ by $\bar{\mathcal{F}}$, it holds that

$$\begin{aligned}\Pr[\mathcal{E}] &= \Pr[\mathcal{E} \mid \mathcal{F}] \cdot \Pr[\mathcal{F}] + \Pr[\mathcal{E} \mid \bar{\mathcal{F}}] \cdot \Pr[\bar{\mathcal{F}}] \\ &\leq \Pr[\mathcal{E} \mid \mathcal{F}] + \Pr[\bar{\mathcal{F}}].\end{aligned} \tag{5}$$

Since for every $S_1, \ldots, S_m$, the probability of drawing v from $\mathcal{U}$ such that $\mathcal{F}$ occurs is the same by symmetry, we can compute $\Pr[\mathcal{F}]$ by reversing the coin flips, first drawing v and then $S_1, \ldots, S_m$. Only three of the $\binom{n}{n/3+1}$ coalitions of size $n/3+1$ have non-zero value; therefore

$$\Pr[\bar{\mathcal{F}}] = 1 - \left(1 - \frac{3}{\binom{n}{n/3+1}}\right)^m < 1/10, \tag{6}$$

where the inequality holds for $n \geq 9$ and $m \leq \frac{1}{6} \cdot 2^{n/3+1}$.

As for $\Pr[\mathcal{E} \mid \mathcal{F}]$, by Lemma 5 at least half of the games in $\mathcal{G}$ (or, equivalently, at least half of the corresponding characteristic functions) are in the support of $\mathcal{U}$ conditioned on $\mathcal{F}$. But by Lemma 7, the payoff allocation $\mathcal{A}((S_1, v(S_1)), \ldots, (S_m, v(S_m)))$ can be in the (ϵ, δ)-probably approximate nucleolus of at most one of the $\binom{n}{3}$ equivalence classes. It follows that

$$\Pr[\mathcal{E} \mid \mathcal{F}] \leq \frac{2}{\binom{n}{3}} < 1/10. \tag{7}$$

Plugging Equations (6) and (7) into Equation (5), we conclude that $\Pr[\mathcal{E}] < 1/5$.

To recap, when drawing $S_1, \ldots, S_m$ from $\mathcal{D}$ and v from $\mathcal{U}$, the probability that the output of $\mathcal{A}$ is in the (ϵ, δ)-probably approximate nucleolus of $G = (N, v) \in \mathcal{G}$ is at most $1/5$. But since this is true for a random game $G \in \mathcal{G}$, there must exist a game $G^\star \in \mathcal{G}$ where the same is true when only drawing $S_1, \ldots, S_m$ from $\mathcal{D}$. That is, m samples are insufficient to compute a payoff allocation in the (ϵ, δ)-probably approximate nucleolus with probability at least $1 - \Delta$ for $\Delta < 4/5$. □

D Approximate Least Core Implementation

The approximate least core algorithm works as follows: compute the approximate least core value $\hat{e}$ from the samples via linear program (1), then minimize the ℓ_2 norm over all allocations $\mathbf{x}$ s.t $\mathbf{x}$ is in the $\hat{e}-$core:

$$\begin{array}{ll}\min & \|\mathbf{x}\|_2 \\ \text{s.t.} & \sum_{i \in N} x_i = v(N) \\ & \sum_{i \in S} x_i + \hat{e} \geq v(S) \quad \forall S \subseteq N\end{array}$$

This may be easily done with any standard optimization library and it is not hard to argue that the resultant $\mathbf{x}$ satisfies null player and symmetry in addition to efficiency.

E Additional Experimental Results

E.1 Feature Valuation

Maximum Deficit. By definition, the maximum deficit $e^\star$ under the least core should be at most as large as that under the Shapley value. However, we wish to verify that the difference is significant in practice. To that end, we compute the least core, the Shapley value, and (as a baseline) equal payoffs on our three datasets. Figure 5 shows the difference between the maximum deficit of each of the solution concepts (including the least core itself) and the maximum deficit of the least core. It can be seen that there is a sizable gap between the Shapley value and the least core, considering that the maximum value of any coalition is 1. Note that no sampling (indeed, no randomness) is involved in this experiment.

Standard Deviation. On each of our three datasets, we compute the empirical standard deviation of payoff allocations given by the least core and the Shapley value (again no sampling is involved). Interestingly, we observe that the least core has considerably higher standard deviation and may thus be considered more discriminating; see Figure 6.

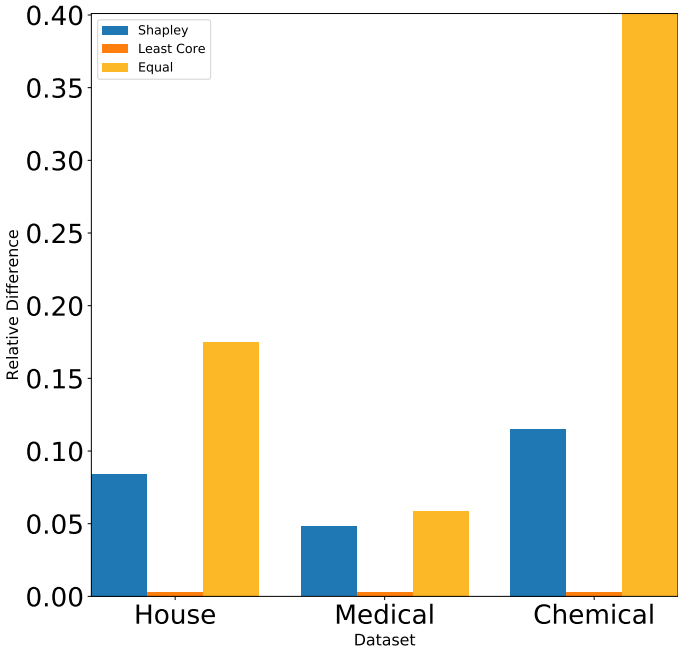


Figure 5: Relative difference between different solution concepts' largest deficits and the least core's largest deficit

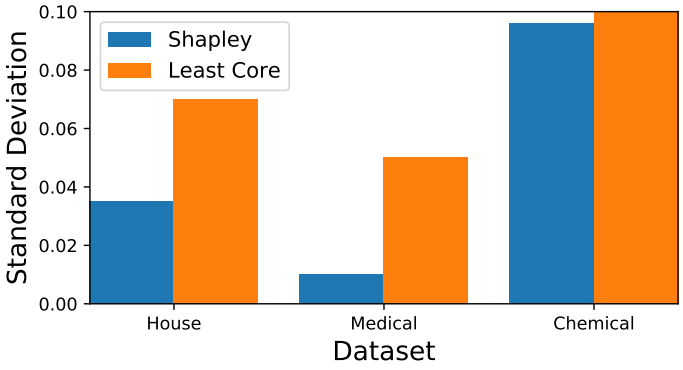


Figure 6: Standard deviation of solution concepts

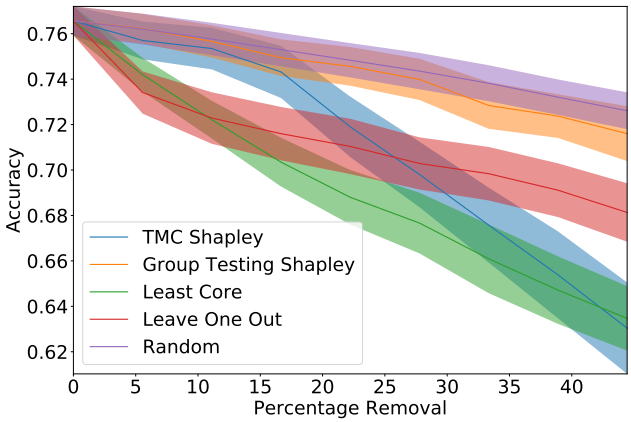
E.2 Data Valuation

Additional Experimental Details Below include attach plots for the synthetic and natural experiments that were not included in the main body due to space constraints. We observe that in the synthetic settings, as depicted in Figures 7 and 8, the approximate least core values are decidedly better than the other importance scores. Under the natural setting, as portrayed by Figure 9, it seems that the change in performance is small and the least core and the Shapley value are roughly comparable across all budgets. Lastly, we note that LOO does not have an error bar in the natural experiment since all the runs are based on the same random sample of data points, and so the error bars are only due to the randomness in the sample of $v(S)$'s that are drawn to approximate the solution concepts.

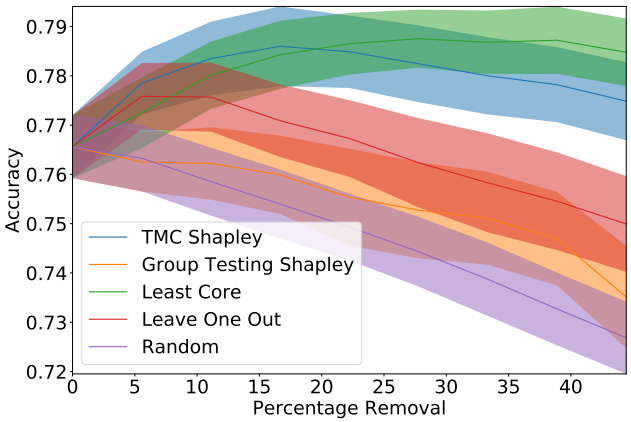
Data Quality vs. Score Lastly, we repeat one more experiment that assesses data quality vs solution concept value. We randomly sample 200 dog-vs-fish data points to form an equally balanced training set. We corrupt 20 percent of train data by adding varying levels of white noise to the features and compute the Least Core value of clean and noisy images. The 5 noise levels are such that it leads to a monotonic decrease in test performance. Then, we plot the percentage of total utility that is assigned to clean scores (since the total utility goes down with noise, using the absolute scale makes it harder to interpret the result). This procedure is repeated 20 times and a budget of 1000 is alloted for approximating the least core.

As can be seen in Figure 3, under the no-noise setting, the clean data account for roughly 80 percent of the total utility and with increasing noise added the proportion grows bigger. The slight trend is due to the fact that the test performance does not drop by too much, going from 96.3 to 92.7.

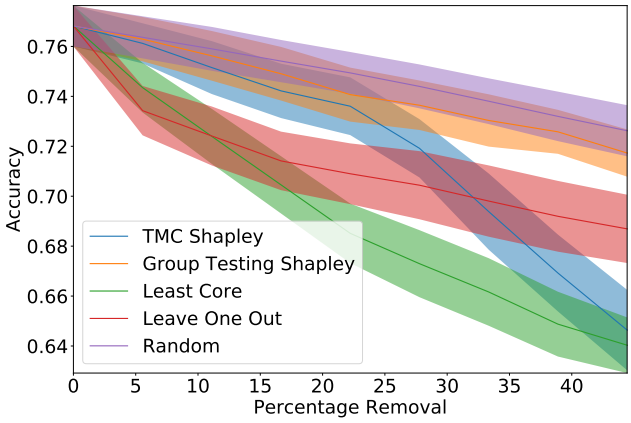
Is the Approximate Shapley Value in the Approximate Least Core Our test procedure is as follows: for each randomly sampled coalition value $v(S)$ used in approximating the least core and estimated Shapley value $\mathbf{x}_S$, we compute $(\sum_{i \in S} x_i + \hat{e})/v(S)$. We count the number of samples for which the ratio is below 0.95. Indeed, if we find one, then the approximate Shapley value $\mathbf{x}$ is not close to being in the approximate least core. Overall, we find that in all the settings we checked, the approximated Shapley does not lie in the approximated least core. For most experiments, at least one percent of all sampled coalitions has its ratio below 0.95. Other trends include that Group Testing tends to produce many more violations than TMC and that the percentage of violations decreases with a larger budget.



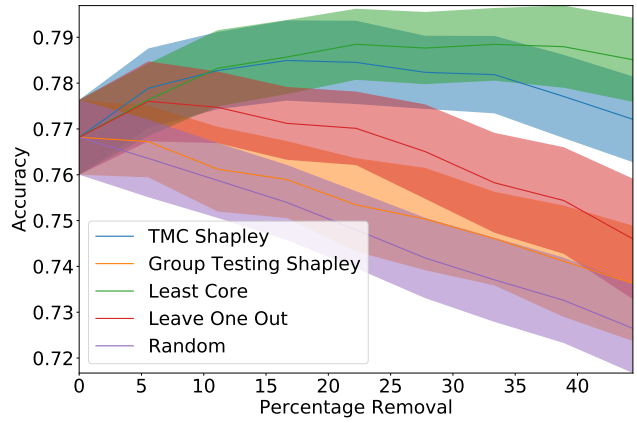
(a) Dropping best data curve at budget $5K$



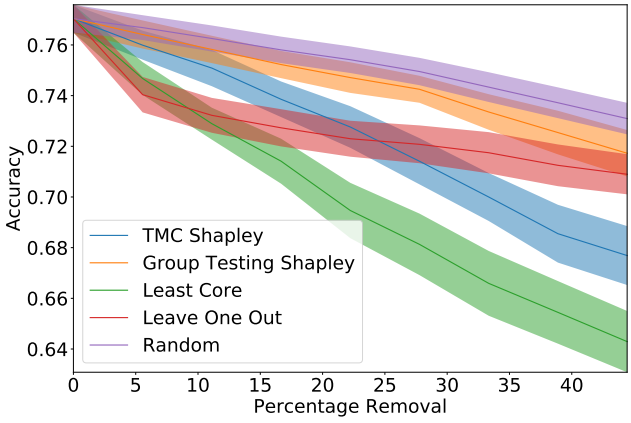
(b) Dropping worst data curve at budget $5K$



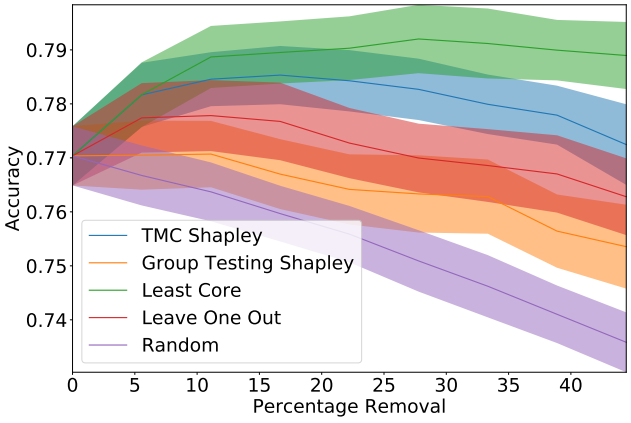
(c) Dropping best data curve at budget $10K$



(d) Dropping worst data curve at budget $10K$

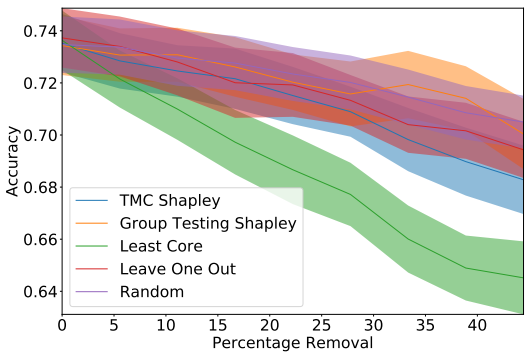


(e) Dropping best data curve at budget $25K$

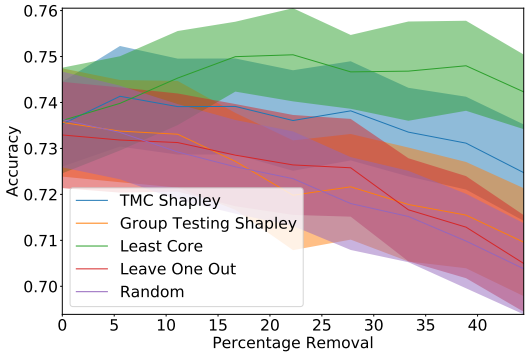


(f) Dropping worst data curve at budget $25K$

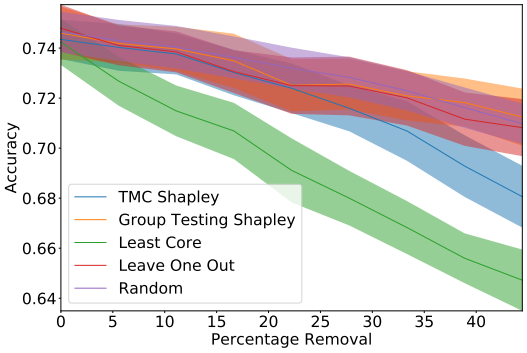
Figure 7: Curves of synthetic dataset (under a logistic regression model) test performance when the best and worst data points ranked according to the solution concepts are removed. For the left column, the steeper the drop, the better. For the right column, the sharper the rise, the better.



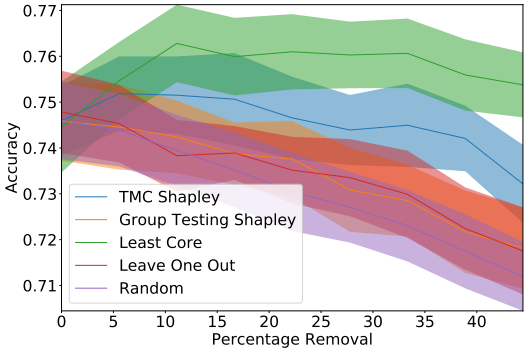
(a) Dropping best data curve at budget $5K$



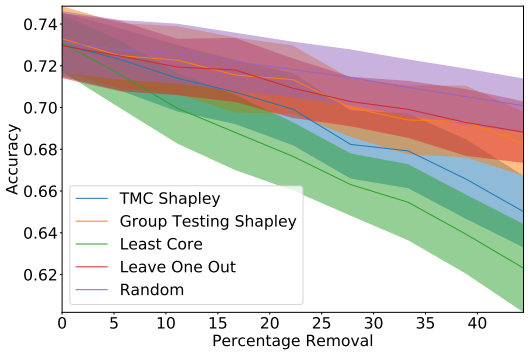
(b) Dropping worst data curve at budget $5K$



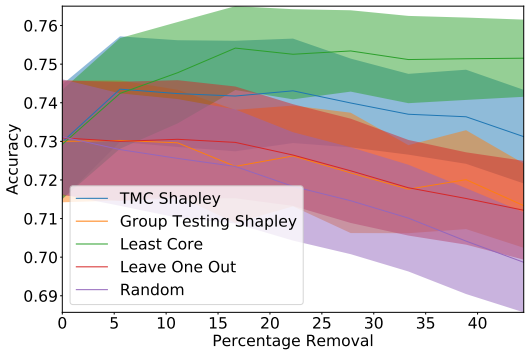
(c) Dropping best data curve at budget $10K$



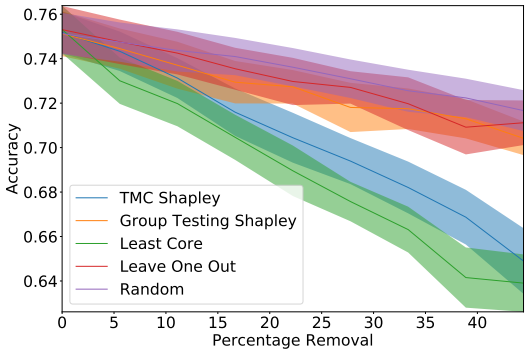
(d) Dropping worst data curve at budget $10K$



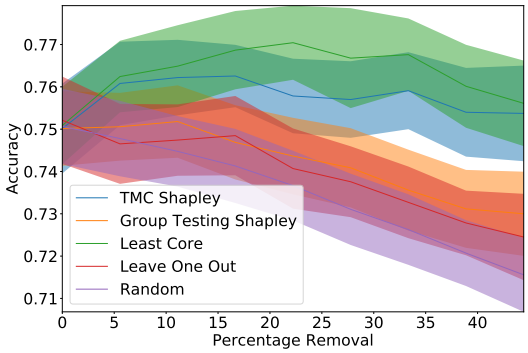
(e) Dropping best data curve at budget $25K$



(f) Dropping worst data curve at budget $25K$

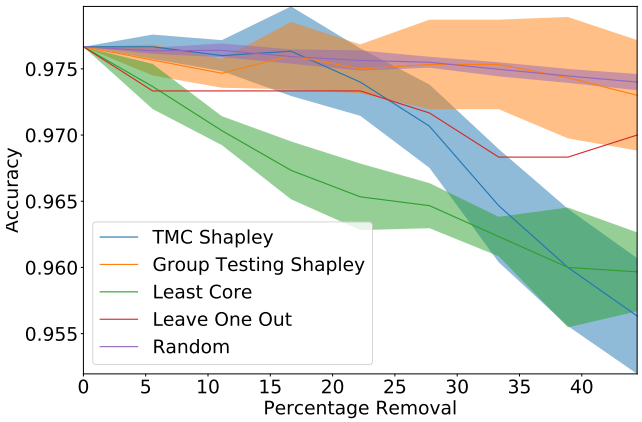


(g) Dropping best data curve at budget $50K$

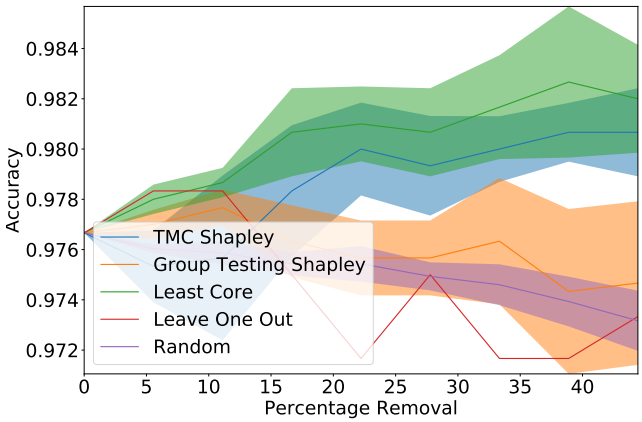


(h) Dropping worst data curve at budget $50K$

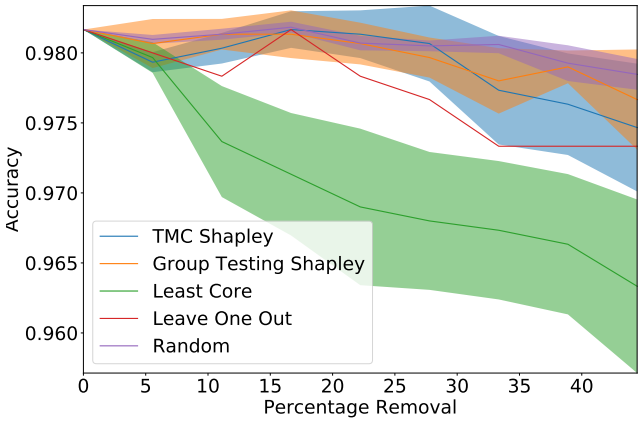
Figure 8: Curves of synthetic dataset (under a feedforward neural network model) test performance when the best and worst data points ranked according to the solution concepts are removed. For the left column, the steeper the drop, the better. For the right column, the sharper the rise, the better.



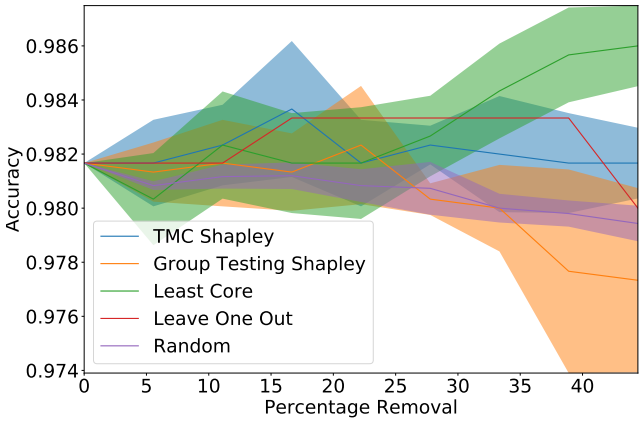
(a) Dropping best data curve at budget $5K$



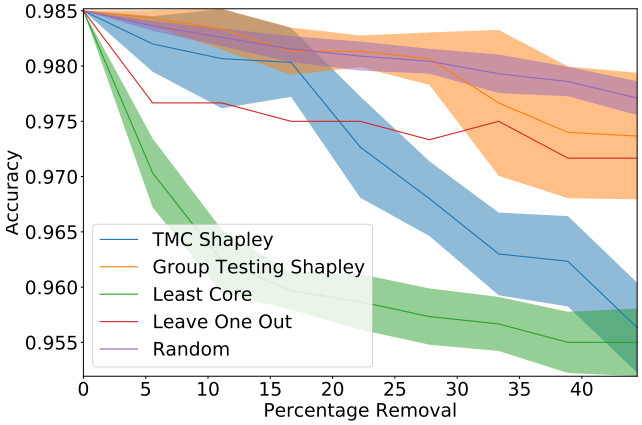
(b) Dropping worst data curve at budget $5K$



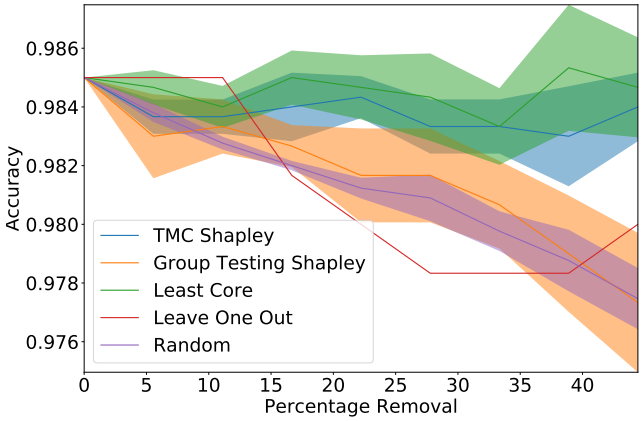
(c) Dropping best data curve at budget $10K$



(d) Dropping worst data curve at budget $10K$



(e) Dropping best data curve at budget $25K$



(f) Dropping worst data curve at budget $25K$

Figure 9: Curves of natural, dog-vs-fish dataset (under a logistic regression model) test performance when the best and worst data points ranked according to the solution concepts are removed. For the left column, the steeper the drop, the better. For the right column, the sharper the rise, the better.