What is Your Data Worth? Equitable Valuation of Data

Amirata Ghorbani and James Y. Zou

Max Nadeau, Max Li, and Xander Davies

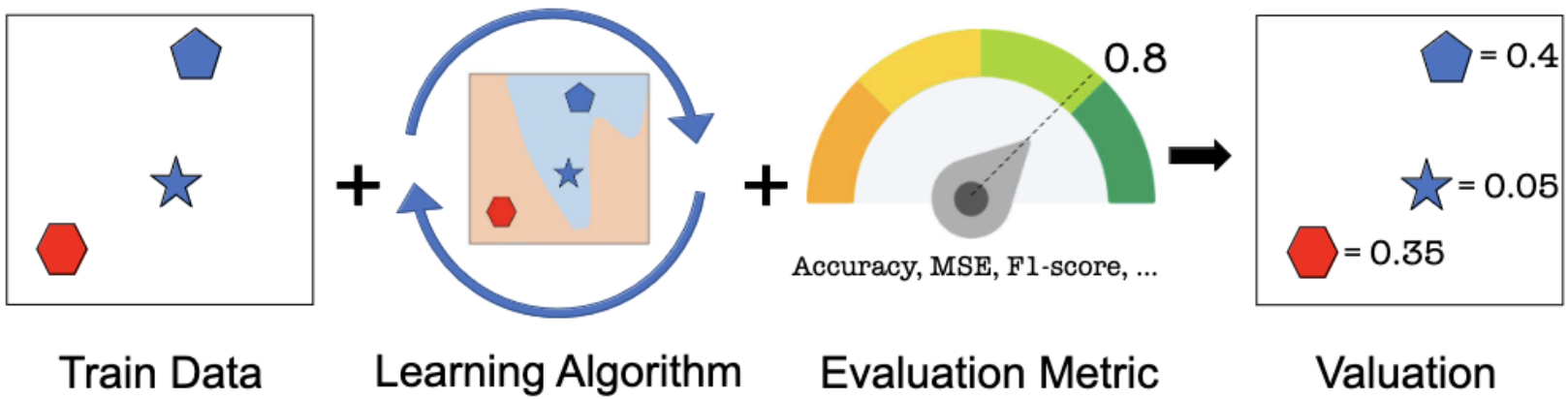
March 22, 2023

Outline

Allocating credit among training data points

- How do we compensate people for the data that they generate?
- How do we quantify the value of data in algorithmic predictions and decisions?
  - Quantity of data
  - Quality of data
- We'd like a principled way of quantifying the value of each training datum.
- Given a labeled training set, a learning algorithm, and a evaluation metric, produce valuations for each datum.

Allocating credit among training data points (cont)



Desiderata for Allocation Procedure

Assume our model is trained on dataset D and is worth $V(D)$. We want to assign each data point $i \in D$ a valuation v_i.

- **Efficiency**: the sum of the data points' valuations add up to the total valuation. $\sum_{i \in D} v_i = V$.
- **Symmetry**: if two data points' marginal contributions are always the same, then their valuations are the same. I.e. if $\forall S \subset D,\ i, j \notin S,\ V(S \sqcup \{i\}) = V(S \sqcup \{j\})$, then $v_i = v_j$.
- **Additivity**: if the overall valuation V is the sum of two separate valuations $U + W$, then the overall valuation v_i should be the sum of its valuations $u_i + w_i$ calculated using the valuations U and W.
- If data point i makes no difference in training performance on any subset $S \subset D$, then i is worth zero.

Cooperative games

- Suppose we have a game with d players, where each player can choose whether or not to cooperate.
- Assume a reward function $V : \mathcal{P}([d]) \to \mathbb{R}$. If $S \subset [d]$ is the set of players that choose to cooperate, then the group receives reward $V(S)$.
- We want to determine how much each player "contributes" to the reward. However, the marginal contribution of player i may depend on which other players have also chosen to cooperate. V does not need to be monotonic!

Shapley values

Maybe we can just take the average of player i's marginal contribution over all subsets. In fact, this calculation gives player i's **Banzhaf power index**:

$$\frac{1}{2^{d-1}} \sum_{S \subset [d] \setminus \{i\}} \left(V(S \sqcup \{i\}) - V(S) \right) \tag{1}$$

The **Shapley value** reweights the marginal contributions based on the size of the subset S:

$$\frac{1}{d} \sum_{j=0}^{d-1} \left[\binom{d-1}{j}^{-1} \sum_{S \subset [d] \setminus \{i\},\ |S|=j} \left(V(S \sqcup \{i\}) - V(S) \right) \right] \tag{2}$$

or, equivalently, $\sum_{\sigma \in \mathbb{S}_d} \left(V(\sigma([\sigma(i)])) - V(\sigma([\sigma(i)-1])) \right)$

Adapting model training to cooperative games

- Assume we have d data points. The order of the training data matters. The value function takes in an *ordered* subset rather than a unordered subset. Thus, it is a function $V : \mathcal{P}([d]) \times \mathbb{S}_d \to \mathbb{R}$, **not** just $\mathcal{P}([d]) \to \mathbb{R}$, where $\mathbb{S}_d$ is the set of permutations on d elements.
- One way to adapt model-training to the cooperative game setup is to take the expectation over all permutations of the training data, i.e. we can define the model value on an *unordered* subset to be

$$V^*(S) = E_{\sigma \sim \text{Unif}(\mathbb{S}_d)}[V(S, \sigma)]$$

Monte Carlo approximation of Shapley Values

Here is one way to sample Shapley values:

- Sample a random permutation of the players σ.
- For each $j \in \{0, ..., d\}$, compute the value of the subset consisting of the first j players by re-training the model on only the first j data points in the permutation (in some random order); this value represents one sample estimate of V^*.
- For every player i, update the sample average of the Shapley value with the marginal contribution of player i when it is included.

Gradient-based approximation

We can approximate the Shapley values by permuting the data points and then training on the dataset in the permuted order, then considering the marginal contribution of each training point to performance. However, this method is *not* an unbiased or consistent estimator of the Shapley value using values V^*.

- Each training point's marginal contribution is only measured when it is the *last* training point being trained on.

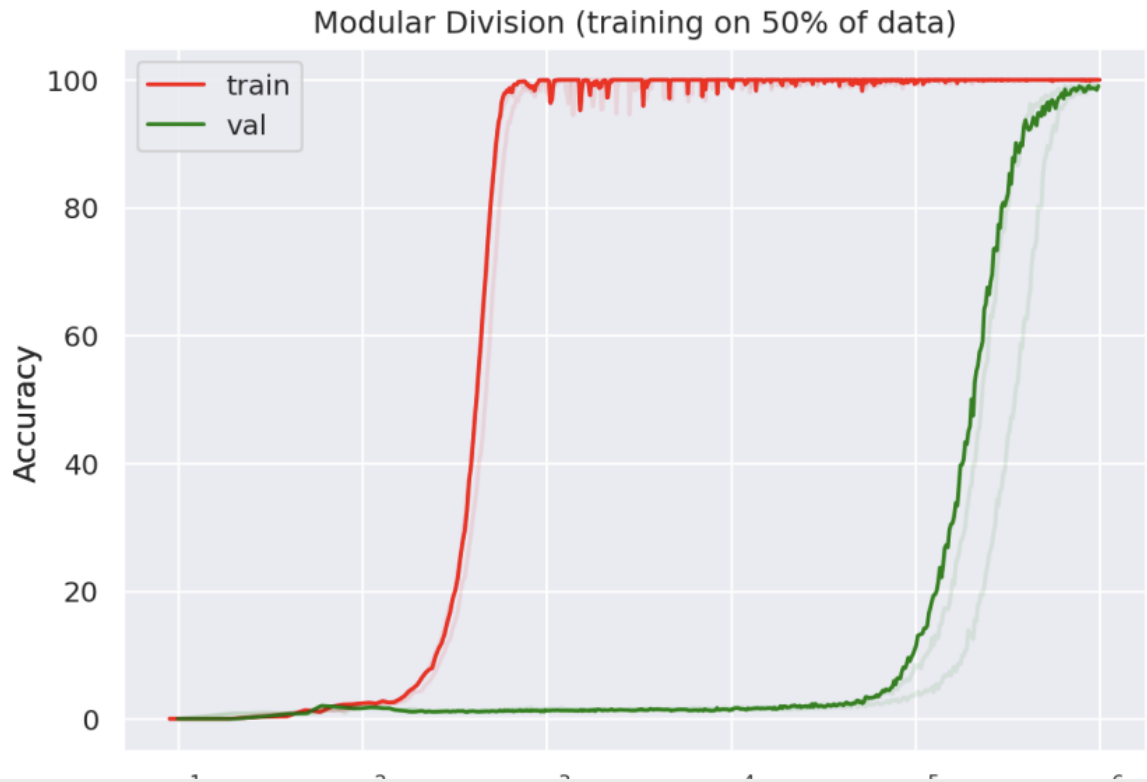
We think taking the average of a player's marginal contribution over all permutations when it is the last player to be added **is** a principled way to estimate the valuations of each of the data points, and we think it is likely easy to show that it satisfies analogues of the above desiderata. But the values calculated are *not* Shapley values.

Truncation

- After training on a large number of training points, the marginal contributions of additional data becomes small.
- To save compute, after the change in loss becomes small (below a threshold computed with bootstrap variance), the authors' algorithm declares the marginal contribution of all additional data points to be zero.
- How much does this distort things? The authors claim that the truncated and non-truncated Shapley values have a correlation of .98.

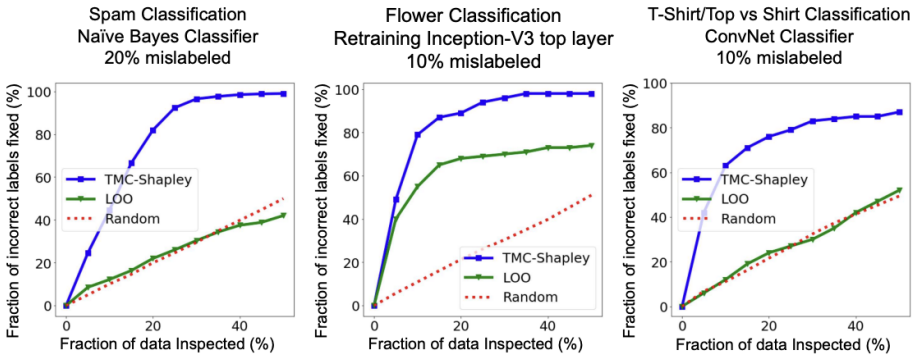
Truncation (cont'd)

However, it is unclear whether certain training points having low marginal contributions implies further training points having low marginal contributions. E.g. grokking.



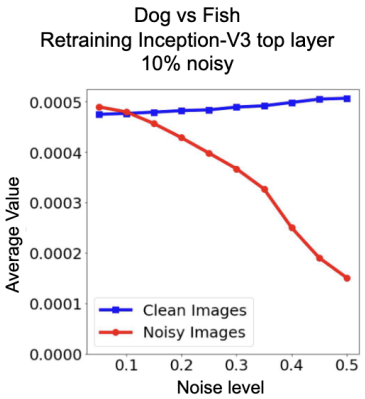
Removing mislabeled data

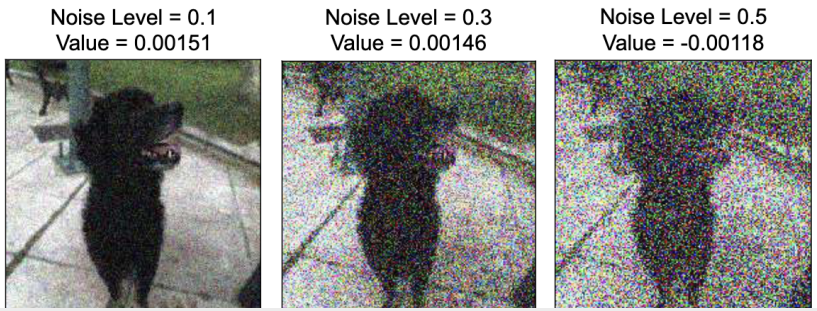
- The authors train classifiers on datasets with 10 or 20% of the data mislabeled, then calculate Shapley values for the points to automatically locate the mislabeled examples.



Noisy data is less informative

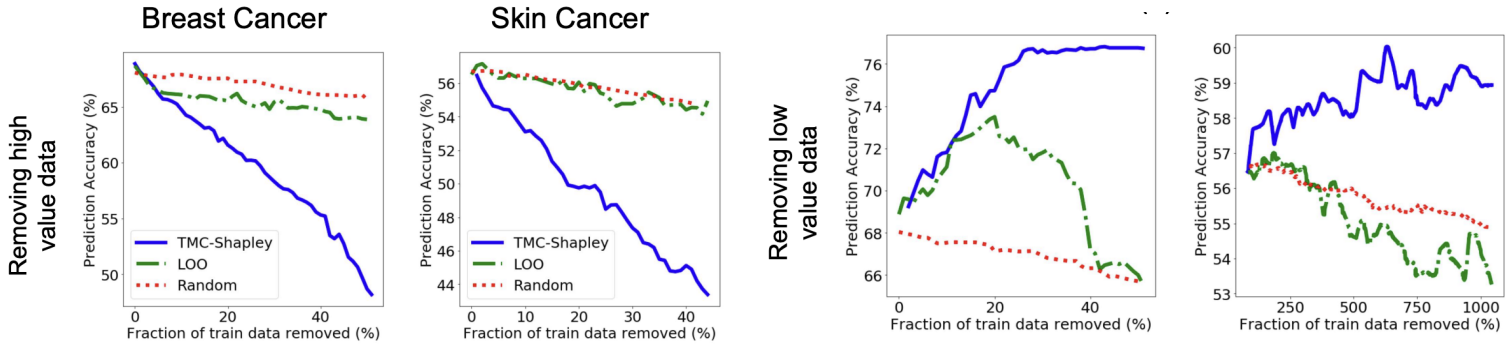
- We'd also like noisy data (like blurry images) to be considered less valuable by our method. To evaluate this, they corrupt 10% of training data with white noise, and show this causes decreased values for targeted data.





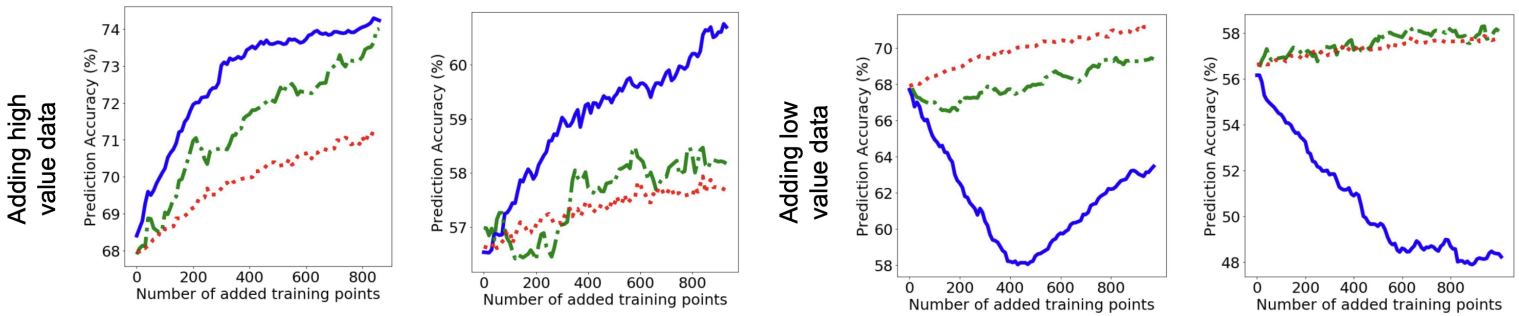
High- and low-quality train set data

- They find that classifier performance decreases fast when you don't train on the highest-value data points, and classifier quality improves.
- These experiments are for logistic regression models for cancer diagnosis that start out pretty close to random.



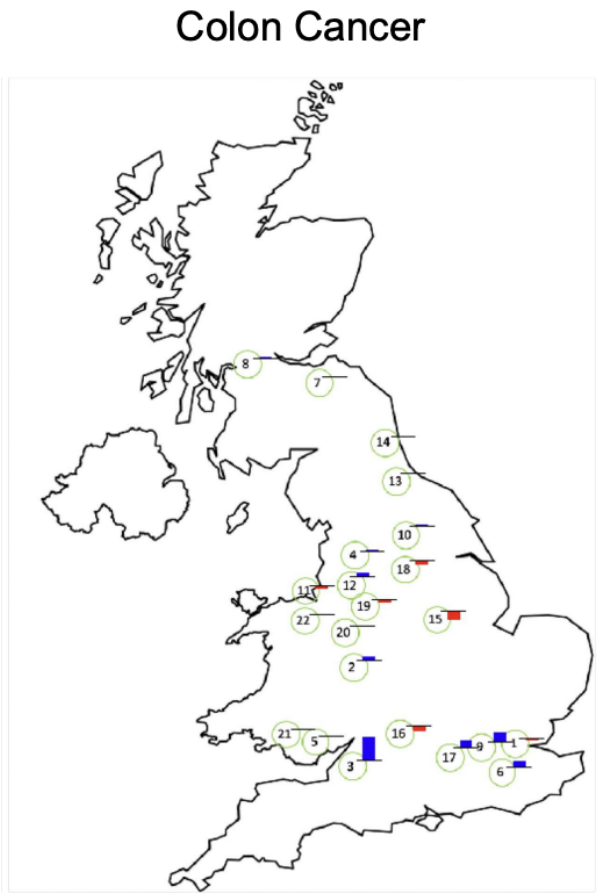
Finding more data to add

- For the same classification tasks and logistic regression models used on the previous slide, the authors train a Random Forest model on the training set to predict the Shapley values.
- They then evaluate new datapoints with this RF model. Adding datapoints to the training set that have high predicted values increases performance a lot, and adding datapoints with low predicted value can decrease performance.



Not all data sources are equal

- For the same classification set-up from the last two slides (binary diagnosis, logistic regression), the authors assess the average Shapley value of data from different locations in the UK.
- Nottingham has a very negative Shapley value, which they speculate is due to different patient demographics in that city.



Invariance of Shapley values across model classes

- For the UK biobank data, they train different classes of binary classifiers: logistic regression (LR), random forests (RF), and k-nearest neighbors (KNN).
- Computing the Shapley values for the data points across the three different model classes indicates that the Shapley values for given data points are only somewhat positively correlated between model classes.

Average correlation of Shapley Values

LR	RF	0.52
RF	KNN	0.42
LR	KNN	0.32

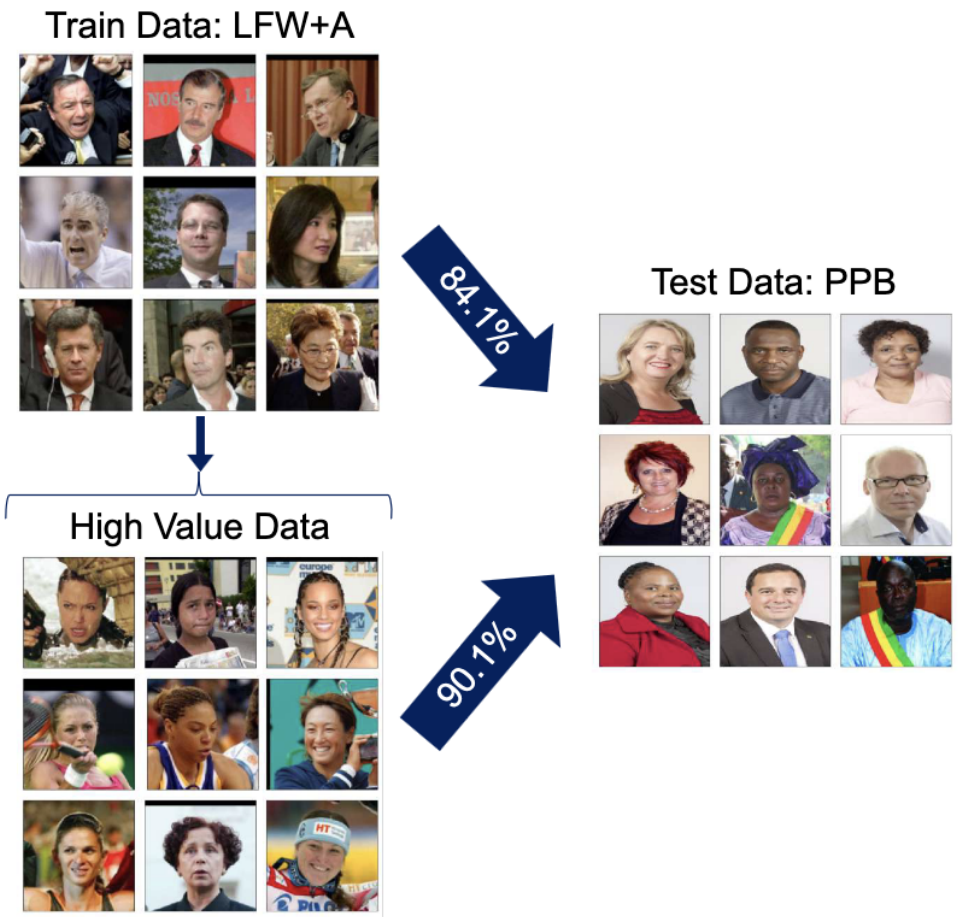
Using value to adapt to new data

- Often, we will train on importantly different data from what our model will eventually be deployed and evaluated on.
- For example, we might have a large dataset with a large range in data quality, and then a small high-quality dataset.
- To achieve better performance on our small high-quality dataset, we:
  - Compute data Shapley values based on performance on a portion of our small high-quality dataset.
  - Remove data points with negative value.
  - Train a new model using a weighted loss function, where each point's weight in the new loss function is its relative data Shapley value.

Example: removing gender bias

- Training a CNN on the CelebA gender detection task results in degraded performances for minorities and women.
- Our goal is to train (the final layer) on images from the LFW+A data set to achieve good performance on a different race- & sex-balanced dataset (PPB).
- Using Data Shapley (calculating with a portion of the PPB dataset), gender detection accuracy increases from 84.1% to 91.5% on a held-out set of 400 PPB image.
- All of the LFW+A images with negative value are from male subjects (the over-represented groups) while the top 20% most valuable images are female subjects!

Example: removing gender bias



Data Adaptation Problem!

- There is an obvious baseline here: fine-tuning on the data points in the target dataset you are using to calculate data Shapley values.
- Unfortuantely, they don't compare to this baseline. This makes it hard to interpret their results.
- It's possible this is still interesting, even if it (for now) under-performs the fine-tuning baseline.

Related Work

- This work connects to past work by these authors on assigning Shapley values to neurons or filters of the finished model. Both papers remove negative-value components to improve performance either on the train set or an OOD test set.
- The Monte Carlo approximation method for Shapley values used in this paper is one of a number of methods used in the literature. In other contexts, Monte Carlo approximation of Shapley value can be greatly improved upon.