Data Shapley: Equitable Valuation of Data for Machine Learning

Amirata Ghorbani, James Zou

ICML2019

Motivation

Quantify the value of data in algorithmic predictions and decisions

It has been suggested that certain data constitute individual property, and as such individuals should be compensated in exchange for these data.

Shapley value

The Shapley value is one way to distribute the total gains to the players, assuming that they all collaborate.

D	a set of n players
$v: 2^D \rightarrow R$	maps subsets of players to the real numbers, $v(\emptyset) = 0$

$$\phi_i(v) = \frac{1}{n} \sum_{S \subseteq D \setminus \{i\}} \binom{n-1}{|S|}^{-1} (v(S \cup \{i\}) - v(S))$$

Shapley value

Properties

1. $\forall S \subseteq D, v(S \cup \{i\}) = v(S \cup \{j\}) \rightarrow \phi_i(v) = \phi_j(v)$
2. $\forall S \subseteq D, v(S \cup \{i\}) = v(S) \rightarrow \phi_i(v) = 0$
3. $\phi_i(v + w) = \phi_i(v) + \phi_i(w)$
4. $\sum_{i \in N} \phi_i(v) = v(N)$

Extend Shapley value to data valuation.

Given a player set D, the Shapley value is the only map from the set of all gamers to payoff vectors that satisfies all four properties in a **coalitional game**.

$$\forall S_1, S_2 \subseteq N, v(S_1 \cup S_2) \geq v(S_1) + v(S_2)$$
$$\forall S \subseteq N, \sum_{i \in S} \phi_i(v) \geq v(S)$$

Proposed method

Proposition 2.1. *Any data valuation $\phi(D, \mathcal{A}, V)$ that satisfies properties 1-3 above must have the form*

$$\phi_i = C \sum_{S \subseteq D - \{i\}} \frac{V(S \cup \{i\}) - V(S)}{\binom{n-1}{|S|}} \tag{1}$$

where the sum is over all subsets of D not containing i and C is an arbitrary constant. We call ϕ_i the data Shapley value of point i.

$$C = 1/n! \qquad \phi_i = \mathbb{E}_{\pi \sim \Pi}[V(S^i_\pi \cup \{i\}) - V(S^i_\pi)]$$

Proposed method

Algorithm 1 Truncated Monte Carlo Shapley

Input: Train data $D = \{1, \ldots, n\}$, learning algorithm $\mathcal{A}$, performance score V
Output: Shapley value of training points: $\phi_1, \ldots, \phi_n$

Initialize $\phi_i = 0$ for $i = 1, \ldots, n$ and $t = 0$
while Convergence criteria not met **do**
  $t \leftarrow t + 1$
  π^t: Random permutation of train data points
  $v_0^t \leftarrow V(\emptyset, \mathcal{A})$
  for $j \in \{1, \ldots, n\}$ **do**
    if $|V(D) - v_{j-1}^t| <$ Performance Tolerance **then**
      $v_j^t = v_{j-1}^t$
    else
      $v_j^t \leftarrow V(\{\pi^t[1], \ldots, \pi^t[j]\}, \mathcal{A})$
    end if
    $\phi_{\pi^t[j]} \leftarrow \frac{t-1}{t}\phi_{\pi^{t-1}[j]} + \frac{1}{t}(v_j^t - v_{j-1}^t)$
  end for
end for

Proposed method

Algorithm 2 Gradient Shapley

Input: Parametrized and differentiable loss function $\mathscr{L}(.;\theta)$, train data $D = \{1, \ldots, n\}$, performance score function $V(\theta)$

Output: Shapley value of training points: $\phi_1, \ldots, \phi_n$

Initialize $\phi_i = 0$ for $i = 1, \ldots, n$ and $t = 0$

while Convergence criteria not met **do**

$t \leftarrow t + 1$

π^t: Random permutation of train data points

$\theta_0^t \leftarrow$ Random parameters

$v_0^t \leftarrow V(\theta_0^t)$

for $j \in \{1, \ldots, n\}$ **do**

$\theta_j^t \leftarrow \theta_{j-1}^t - \alpha \nabla_\theta \mathscr{L}(\pi^t[j]; \theta_{j-1})$

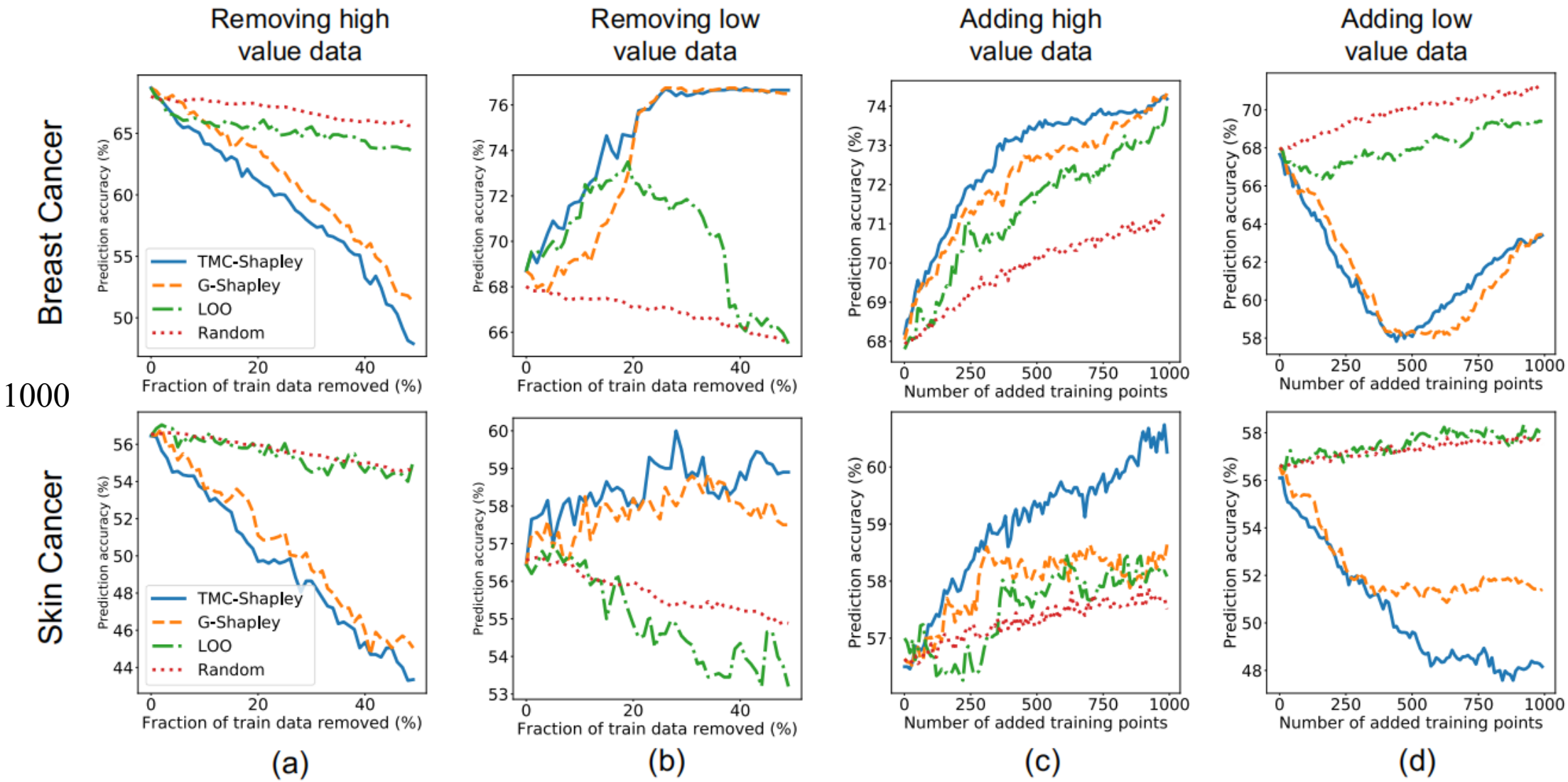
$v_j^t \leftarrow V(\theta_j^t)$

$\phi_{\pi^t[j]} \leftarrow \frac{t-1}{t} \phi_{\pi^{t-1}[j]} + \frac{1}{t}(v_j^t - v_{j-1}^t)$

end for

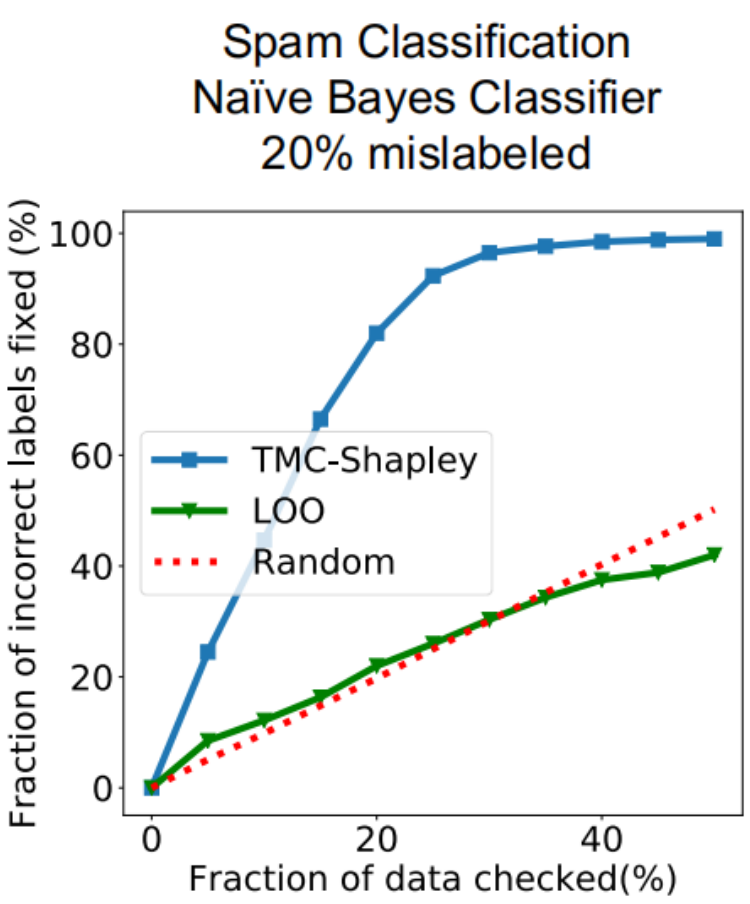
end for

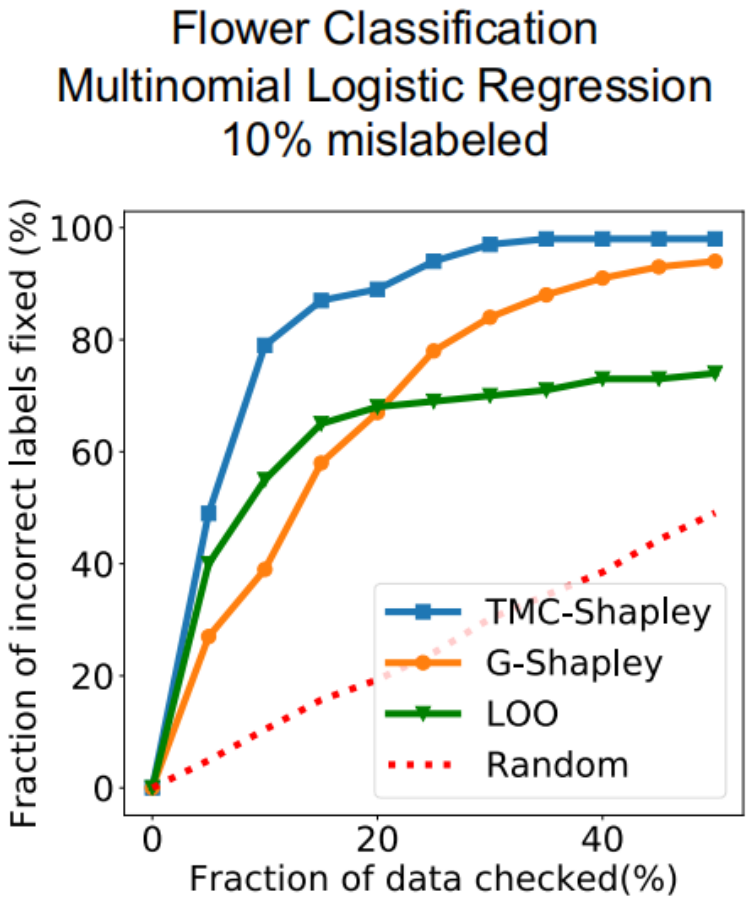
Experiments

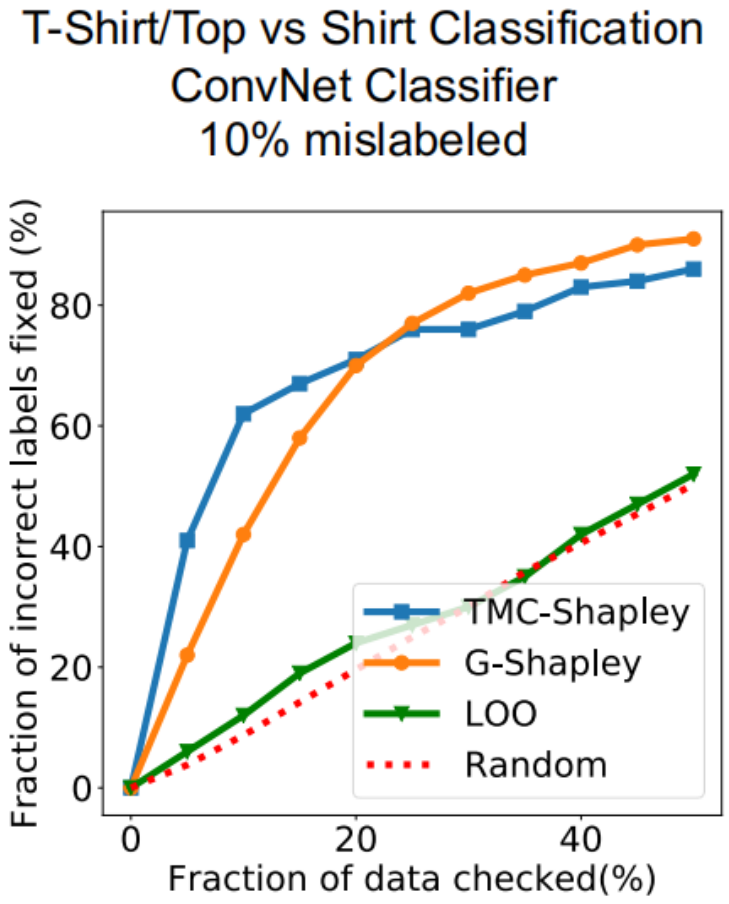


1000

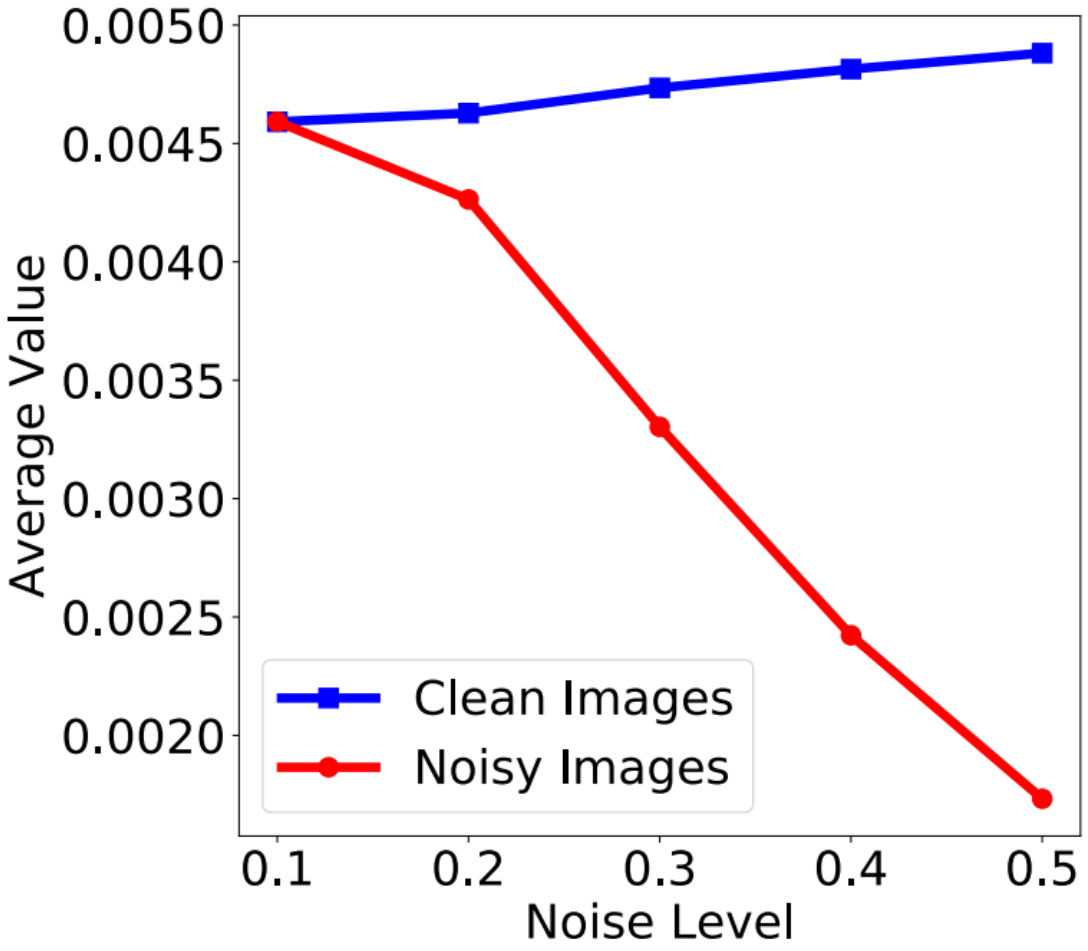
Experiments



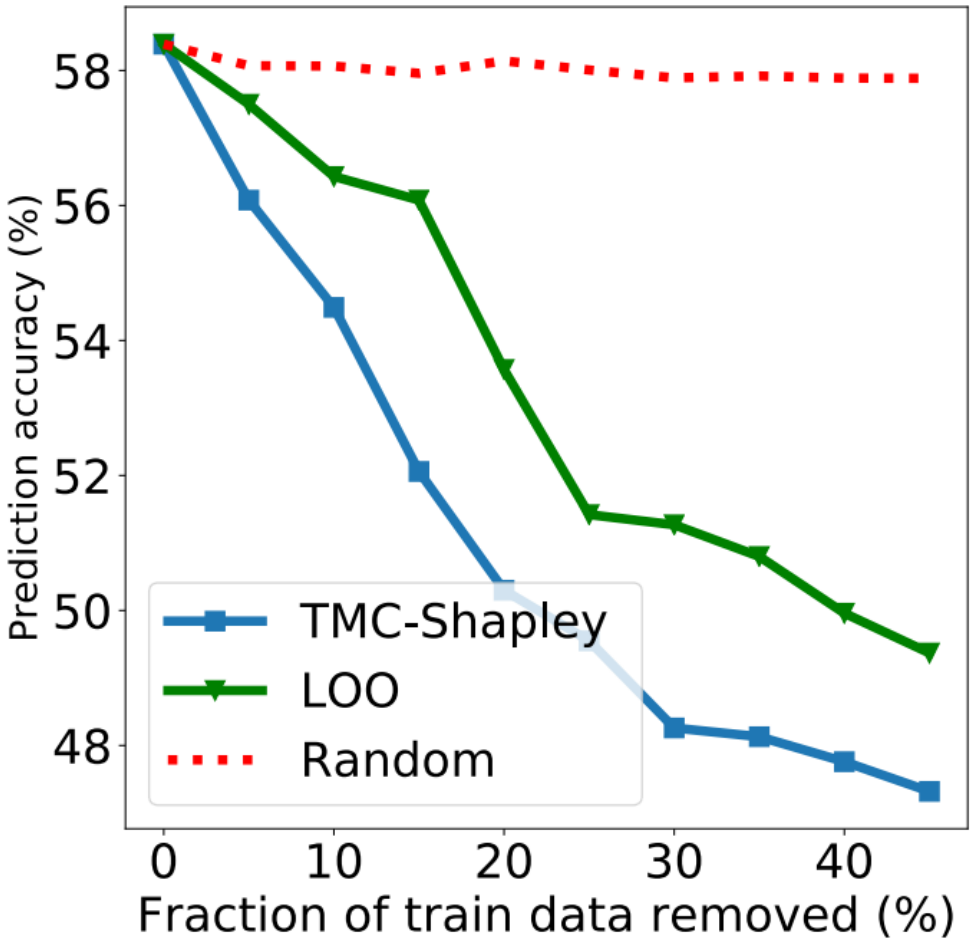




Experiments



(a)



(b)

Thanks