Data Shapley: Equitable Valuation of Data for Machine Learning

Amirata Ghorbani, James Zou

Presenter: Arshdeep Sekhon
https://qdata.github.io/deep2Read

Summary

Assign an *equitable* importance score ϕ_i to every data point / data sources in a scalable way for ML

- Assume a supervised ML setting
- Given N data sources $D = \{x_i, y_i\}$ $i = \{1, \ldots, N\}$, a learning alogrithm A(blackbox), metric V(blackbox) computed on fixed test set
- thus the score depends on the metric, learning algorithm, target task

Desired Characteristics of Data Valuation

- Here, V is test set loss / or other metric on test set
- NULL PLAYER: if (x_i, y_i) does not change the performance if i is added to any subset of the train data source, ideal data value is 0.
- EQUAL PLAYERS: If for two data i and j, and any subset $S \subseteq D - \{i, j\}$, $V(S \bigcup \{i\}) = V(S \bigcup \{j\})$, then $\phi_i = \phi_j$.
- SUM OF TWO GAMES : If the evaluation is from two performance scores V and W $\phi_i(V + W) = \phi_i(V) + \phi_i(W)$

Proposition

View the supervised machine learning problem as a cooperative game : each source is a player and the players work together through the learning algorithm to achieve prediction score $V(D)$

Proposition

View the supervised machine learning problem as a cooperative game : each source is a player and the players work together through the learning algorithm to achieve prediction score $V(D)$
Any data valuation $\phi(D, A, V)$ that satsifies the above properties must have the form :

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

C is an arbitrary constant, ϕ_i is the shapley value of source i

- "same as shapley upto a constant value"

Data Shapley: Introduction

- Shapley Value computation is exponential in the number of train data sources. (2^{N-1} subsets S if N data sources)
- Monte Carlo Sampling for Shapley Value

$$E_\pi V(S_\pi^i \bigcup \{i\}) - V(S_\pi^i) \tag{2}$$

- S_π^i the set of data points before i in permutation π
- Data Shapley : Extend the sampling based idea to data valuation setting

Algorithm

Algorithm 1 **Truncated Monte Carlo Shapley**

Input: Train data $D = \{1, \ldots, n\}$, learning algorithm $\mathscr{A}$, 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
  $v_0^t \leftarrow V(\emptyset, \mathscr{A})$
  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
      $v_j^t \leftarrow V(\{\pi^t[1], \ldots, \pi^t[j]\}, \mathscr{A})$
    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 while

Truncation

- $V(S)$ performance on a test set after being trained on S
- Already an approximation of the true test performance
- Noise in this $V(S)$: variation in the test performance(measure by bootstrapping samples of the test set)
- as S increases, change in performance by adding only one point dectrases
- Truncate based on the marginal contribution within V

Keys

Say 4 data points : A,B,C,D

- Sample a permutation of data points say B, C, A, D
- scan from left to right in one such sample of eprmutaiton $B- > C- > A- > D$
- Marginal Contribution for each sample At Step 3, $V(B, C, A) - V(B, C)$, will be less than $V(B, C) - V(B)$ At step 2
- Truncate at a predefined tolerance: Only do $B- > C- > A$ and assign zero as marginal contribution to the rest

Issues

- learning functions and metrics
- retraining repeatedly, (not done for feature attribution)
- – hyperparameters change as dataset changes

Alternative Data Shapley: G Shapley

To avoid retraining, every time

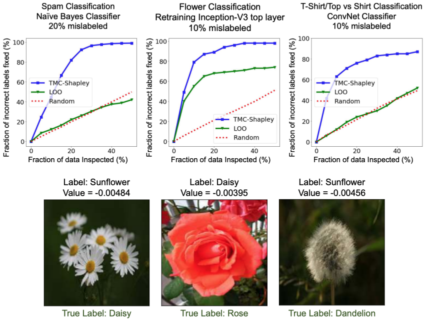
Algorithm 2 Gradient Shapley

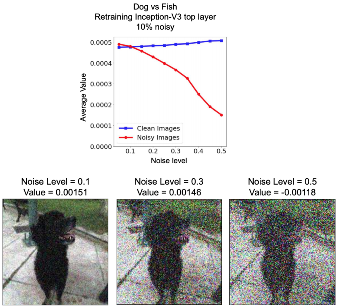
Input: Parametrized and differentiable loss function $\mathscr{L}(.;\theta)$, train data $D = \{1,\ldots,n\}$, performance score function $V(\theta)$
Output: Shapley value of training points: $\phi_1,\ldots,\phi_n$

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 while

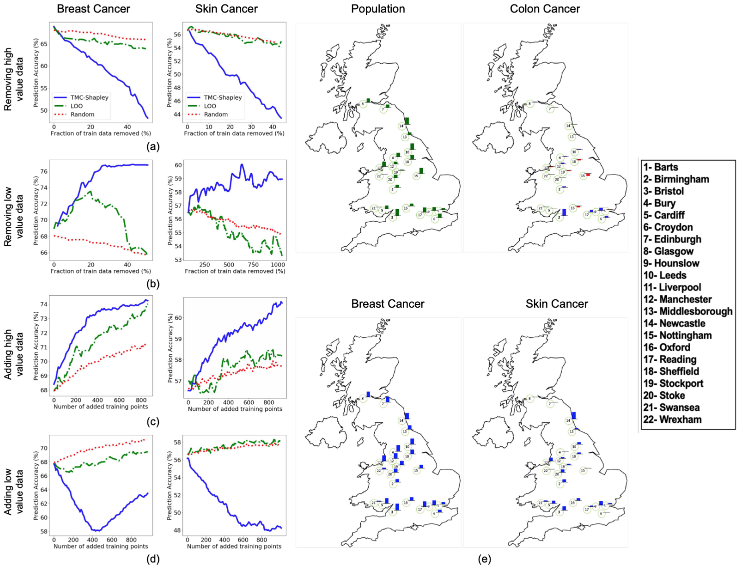
Experiments: Value of low quality data

- data set where some data points are mislabeled
- use value to find the mislabeled data points and correct the labels
- noisy data





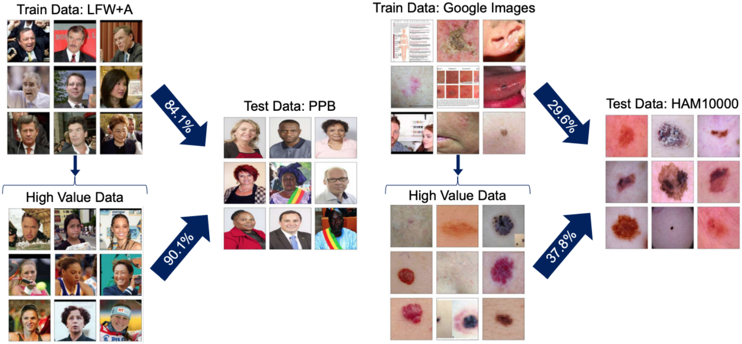
Experiments: value of data



Using value to adapt to new data

Source to Target	Prediction Task	Trained Model	Original Performance (%)	Adapted Performance (%)
Google to HAM1000	Skin Lesion Classification	Retraining Inception-V3 top layer	29.6	37.8
CSU to PP	Disease Coding	Retraining DeepTag top layer	87.5	90.1
LFW+ to PPB	Gender Detection	Retraining Inception-V3 top layer	84.1	91.5
MNIST to UPS	Digit Recognition	Multinomial Logistic Regression	30.8	39.1
Email to SMS	Spam Detection	Niave Bayes	68.4	86.4

(a)



(b)

Conclusions

- the value of individual datum depends on the learning algorithm, evaluation metric as well as other data points in the training set
- how DATA SHAPLEY behaves for different learning functions and metrics