Data Shapley : Equitable Valuation of Data for Machine Learning

2023.06.20 (Tue.)

Superb AI Machine Learning Team

Presenter : Kyeongryeol, Go

Overview

- Data generated by individuals is a key component of the market place
  - E.g. health care, advertising

- 3 ingredients in case of supervised machine learning
  - A fixed training dataset : $D = \{(x_i, y_i)\}_{i=1}^{n}$
  - A learning algorithm : $\mathcal{A}$, which solves $\theta^* = \arg\min_{\theta} \sum l(f(x_i; \theta), y_i)$
  - A performance metric : V, test performance of f
    - Mean Squared Error in regression
    - Accuracy in classification

1. What is an equitable measure of the value of each (x_i, y_i) to $\mathcal{A}$ w.r.t. V
2. How do we efficiently compute this data value in practical settings

Equitable data valuation

- Notation
  - $\phi_i(V)$: value of i-th train datum in terms of V
  - $V(S)$: test performance of f trained on $S \subseteq D$

    c.f. $V(S = \Phi)$ indicates test performance of randomly initialized classifier

- What properties make ϕ_i "equitable"
  1. Null player : $\forall S \subseteq D - \{i\},\ V(S \cup \{i\}) = V(S),\ then\ \ \phi_i = 0$
  2. Symmetry : $\forall S \subseteq D - \{i, j\},\ V(S \cup \{i\}) = V(S \cup \{j\}),\ then\ \ \phi_i = \phi_j$
  3. Linearity : $\forall \alpha_1, \alpha_2 \in \mathbb{R},\ \phi_i(\alpha_1 V_1 + \alpha_2 V_2) = \alpha_1 \phi_i(V_1) + \alpha_2 \phi_i(V_2)$
  4. Efficiency : $V(D) = \sum_i \phi_i(V)$

Shapely is a <u>unique</u> measure satisfying the above four properties

Shapley, Lloyd S. "A value for n-person games." (1953): 307-317.

Shapely formulation

- Formulation

$$(Shapely)\ \phi_i \coloneqq \frac{1}{n} \sum_{S \subseteq D - \{i\}} \binom{n-1}{|S|}^{-1} \left(V(S \cup \{i\}) - V(S)\right)$$

  - Sum of marginal gains divided by # of subsets with cardinality $|S|$
    - assign uniform weights to different subset size $|S|$
    - note that such normalization is derived from the efficiency property

1. Traversing all the possible $S \subseteq D - \{i\}$ is computationally infeasible for large dataset
2. Every computation of V requires a separate model training

Approximating ϕ

- Monte-Carlo method

$$\phi_i := \frac{1}{n}\sum_{S\subseteq D-\{i\}} \binom{n-1}{|S|}^{-1} \left(V(S\cup\{i\}) - V(S)\right)$$

$$= \frac{1}{n}\sum_{j=0}^{n-1}\binom{n-1}{j}^{-1}\sum_{\substack{S\subseteq D-\{i\}\\ |S|=j}} V(S\cup\{i\})-V(S)$$

$$= \mathbb{E}_{j\sim U(0,n-1)}\mathbb{E}_{\substack{S\subseteq D-\{i\}\\ |S|=j}}[V(S\cup\{i\})-V(S)] = \mathbb{E}_{\pi\sim\Pi}\left[V\left(S_\pi^i\cup\{i\}\right) - V\left(S_\pi^i\right)\right]$$

t=1, π = [2,1,5,3,4]

$\phi_2^1 = V(\{2\})$

$\phi_1^1 = V(\{2,1\}) - V(\{2\})$

…

$\phi_4^1 = V(\{2,1,5,3,4\}) - V(\{2,1,5,3\})$

…

t=T, π = [5,3,1,2,4]

$\phi_5^T = V(\{5\})$

$\phi_3^T = V(\{5,3\}) - V(\{5\})$

…

$\phi_4^T = V(\{5,3,1,2,4\}) - V(\{5,3,1,2\})$

$$\phi_i = \frac{1}{T}\sum_{t=1}^{T}\phi_i^t \quad \forall i \in [5]$$

- π : random permutation of indices $[n]$
- S_π^i : subset of indices coming before datum i

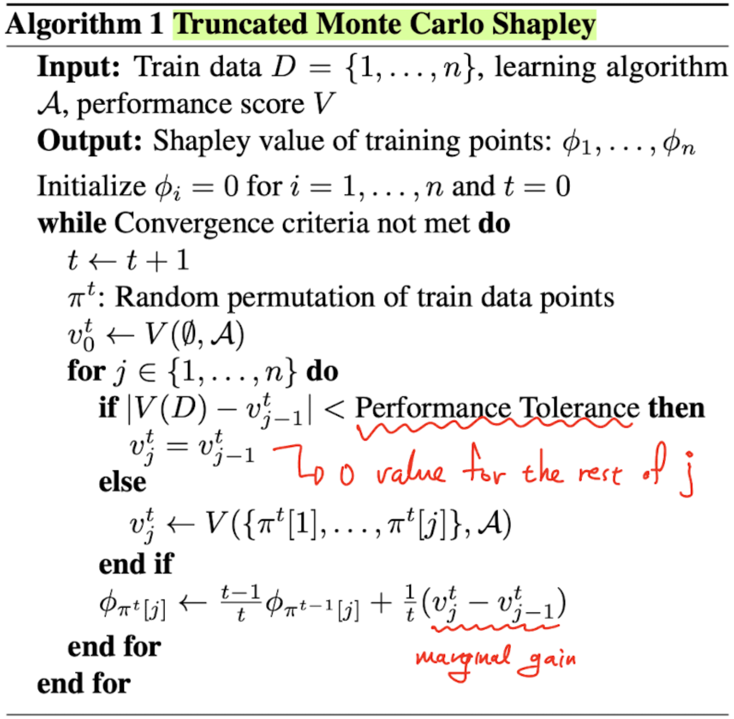
If π = [2,1,5,3,4], then $S_\pi^5 = \{2,1\}$

1. Sample a random permutation
2. Scan the permutation from the first to the last element and calculate the marginal gain (truncate the calculation of marginal gain whenever $\left|V(D) - V\left(S_\pi^i\right)\right| < tolerance$)
3. Repeat 1 and 2, then obtain final estimation by average

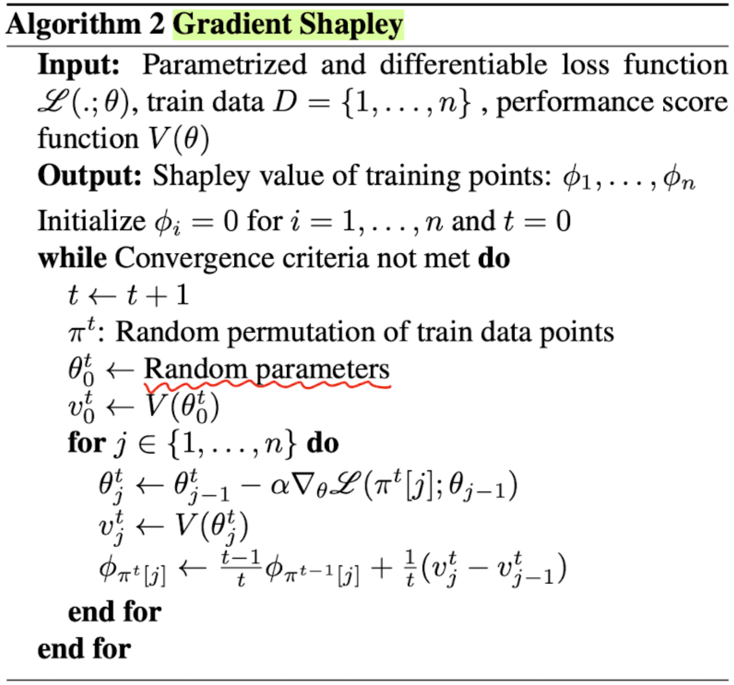
Approximating V

1. Train the model with <u>only one epoch</u> and with <u>mini-batch size of 1</u> → Algorithm 2
   - marginal gain can be computed between iterations
   - require HPO to find the one resulting best performance by one epoch (high learning rate)
2. Estimate value by <u>group of data points</u> → Algorithm 1

   e.g. group the patients into discrete bins based on age, gender, ethnicity, etc.



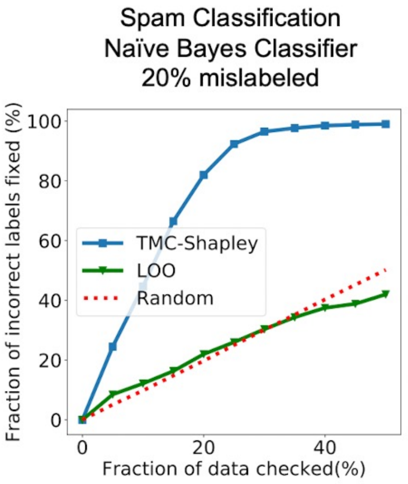
```
Algorithm 1 Truncated Monte Carlo Shapley
Input: Train data $D = \{1, \ldots, n\}$, learning algorithm $\mathcal{A}$, performance score $V$
Output: Shapley value of training points: $\phi_1, \ldots, \phi_n$
Initialize $\phi_i = 0$ for $i = 1, \ldots, n$ and $t = 0$
while Convergence criteria not met do
    $t \leftarrow t + 1$
    $\pi^t$: 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
            $v_j^t = v_{j-1}^t$   → 0 value for the rest of j
        else
            $v_j^t \leftarrow V(\{\pi^t[1], \ldots, \pi^t[j]\}, \mathcal{A})$
        end if
        $\phi_{\pi^t[j]} \leftarrow \frac{t-1}{t}\phi_{\pi^{t-1}[j]} + \frac{1}{t}(v_j^t - v_{j-1}^t)$
                                                   marginal gain
    end for
end for
```



```
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
    $t \leftarrow t + 1$
    $\pi^t$: 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
```

Convergence criteria
(for generating random permutation)

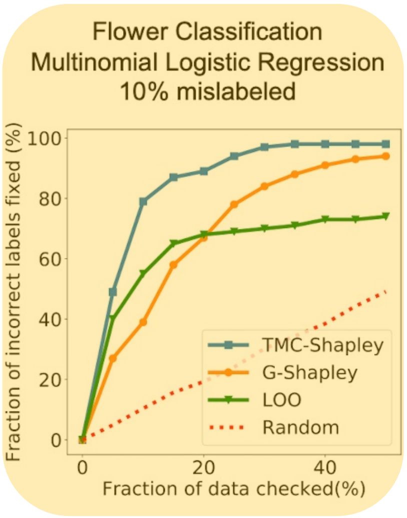
$$\Rightarrow \frac{1}{n}\sum_{i=1}^{n}\frac{|\phi_i^t - \phi_i^{t-100}|}{|\phi_i^t|} < 0.05$$



Spam Classification
Naïve Bayes Classifier
20% mislabeled

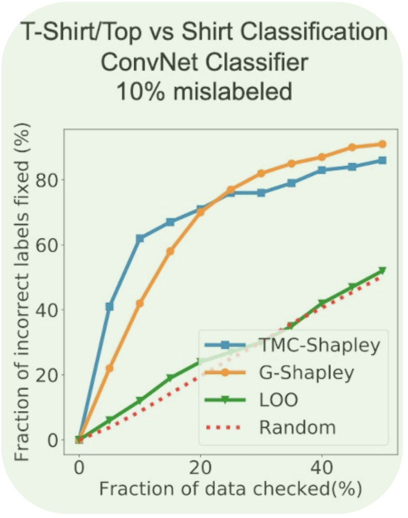
2 classes
3000 data points
5000 iterations

< From lowest valued data >



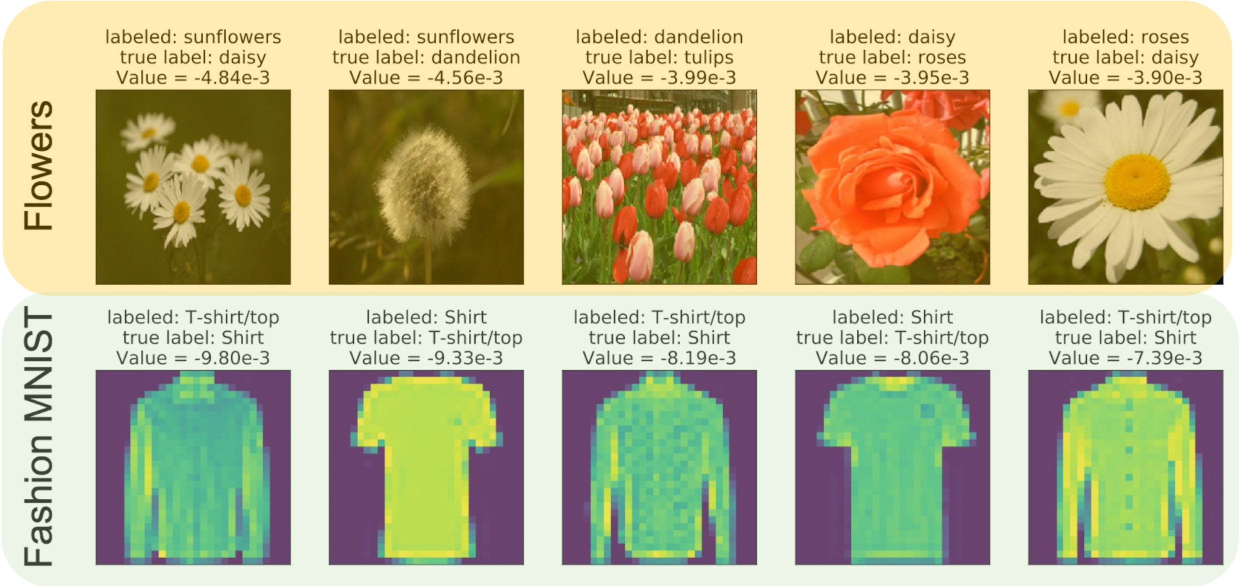
Flower Classification
Multinomial Logistic Regression
10% mislabeled

5 classes
1000 images
2000 iterations



T-Shirt/Top vs Shirt Classification
ConvNet Classifier
10% mislabeled

2 classes
1000 images
2000 iterations

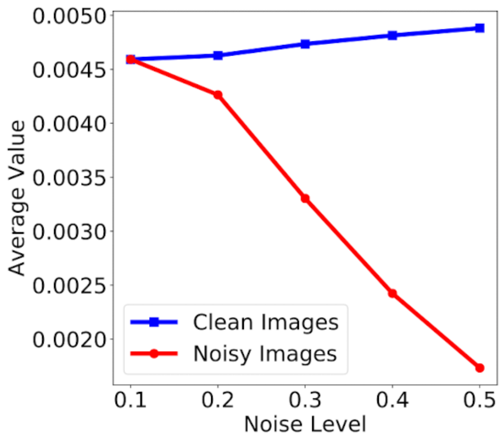


5 images with the lowest value : all mis-labeled

Use representation from Inception-V3 model

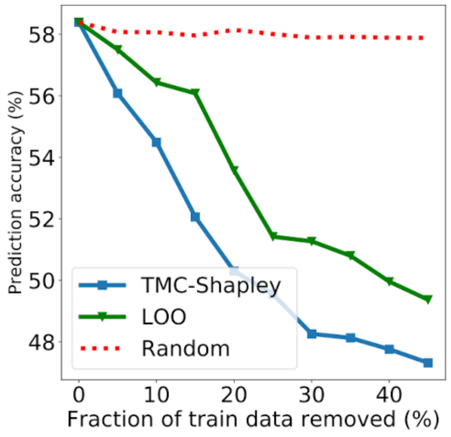
Use one convolution + 2 feed-forward layers

1. Mis-labeled data has low values
2. Discrepancy b/t clean and noisy gets larger as noise level increases
3. Deletion of groups of high values leads to huge performance drop



2 classes
100 images

< From largest valued group >



2 classes
60000 data points
146 groups

Use Inception-V3 model unfreezing the last layer

Use a gradient boosting classifier

Distributional Shapely

- Data Shapely assumed a fixed dataset D
  - no guarantee on consistency between the different, but similar datasets
  - it may be very sensitive to the exact choice of D and not transferrable across datasets

- Idea : Suggest an unbiased estimator for the size of D

$$v_i = \mathbb{E}_{B \subseteq D}[\phi_i] = \mathbb{E}_{B \subseteq D} \mathbb{E}_{j \sim U(0,|B|-1)} \mathbb{E}_{\substack{S \subseteq B - \{i\} \\ |S| = j}} [V(S \cup \{i\}) - V(S)]$$

  - more stable under perturbations to inputs as well as the underlying data distribution

  - Thm. under Lipschitz Stable performance metric V, the following holds
    (c.f. $|V(S \cup \{i\}) - V(S \cup \{j\})| \leq \beta(|S|) \cdot d(i,j)$)
    1. Similar distributions yield similar value functions $\Rightarrow |v_i(D_s) - v_i(D_t)| \leq C \cdot W(D_s, D_t)$
    2. Similar points receive similar values $\Rightarrow |v_i - v_j| \leq \mathrm{C} \cdot d(i,j)$

Distributional Shapely

- Distributional Shapely assigns relatively <u>larger weights to small sized subsets</u>

  (note : (i) $S \subseteq B - \{i\},\ B \subseteq D,$ (ii) Data shapely : $B = D$)

	weight	$\lvert S\rvert=0$	$\lvert S\rvert=1$	…	$\lvert S\rvert=n\text{-}2$	$\lvert S\rvert=n\text{-}1$	*sum*
Data shapely →	$\lvert B\rvert=n$	$1/\mathrm{n}$	$1/\mathrm{n}$	…	$1/\mathrm{n}$	$1/\mathrm{n}$	1
	$\lvert B\rvert=n\text{-}1$	$1/(\mathrm{n}\text{-}1)$	$1/(\mathrm{n}\text{-}1)$	…	$1/(\mathrm{n}\text{-}1)$	0	1
	…	…	…	…	…	…	…
	$\lvert B\rvert=1$	$1/2$	$1/2$	…	0	0	1
	$\lvert B\rvert=0$	1	0	…	0	0	1

larger

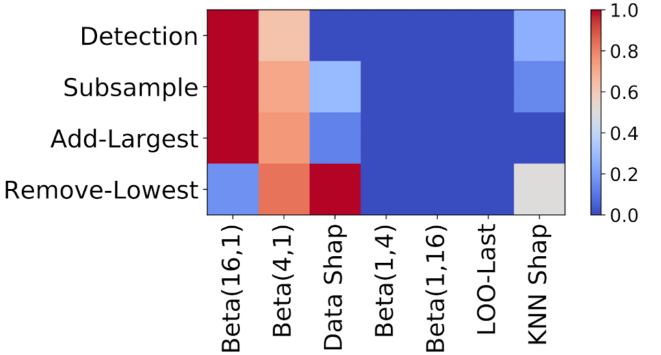
smaller

- no more equal weights to subset size → no more satisfy the efficiency property ≔ semi-values
- It is controversial whether it is beneficial to satisfy the efficiency in machine learning

  ∵ the value itself doesn't matter, but the rank matters

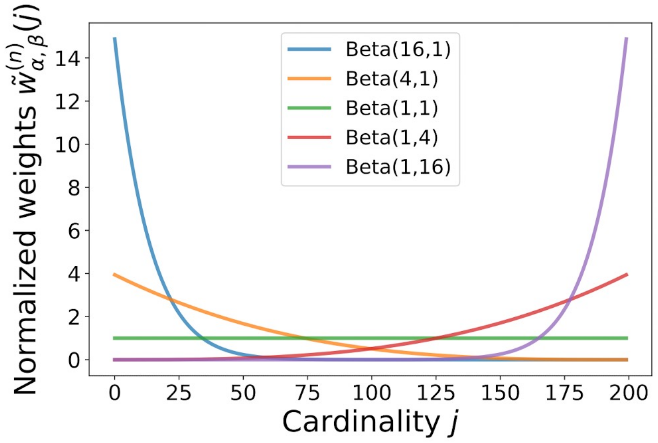
Beta Shapely

- Data Shapely assumed a uniform weight over the subset size $|S|$
  - Thm. When $|S|$ is large enough, the marginal gain becomes negligible
    (signal-to-noise ratio($:= \mu/\sigma$) increases as the cardinality increases)
    (more likely to be perturbed by noise)



- General representation of semi-value
  - $\psi_i = \frac{1}{n}\sum_{j=0}^{n-1} w(j) \sum_{\substack{S\subseteq D-\{i\} \\ |S|=j}} V(S\cup\{i\}) - V(S)$ s.t. $\sum_{j=0}^{n-1}\binom{n-1}{j} w(j) = n$

    (c.f. $\phi_i = \frac{1}{n}\sum_{j=0}^{n-1}\binom{n-1}{j}^{-1} \sum_{\substack{S\subseteq D-\{i\} \\ |S|=j}} V(S\cup\{i\}) - V(S)$)



- Idea : Assign large weights to small cardinality
  - $w_{\alpha,\beta}(j) = n\frac{Beta(j+\beta, n-j-1+\alpha)}{Beta(\alpha,\beta)}$ $\Rightarrow (\alpha,\beta) = (1{,}1)$ suffices to Data Shapely
  - $\alpha \geq \beta = 1$ assigns large weights on the small cardinality and remove noise from large cardinality

Data Banzhaf

- Stochastic training methods (e.g. SGD) make the performance metric unreliable
  - noise is shown to be substantial to make different runs produce inconsistent value rankings

- Idea 1 : Suggest to use Banzhaf value as a robust data value notion
  - $\psi_i = \frac{1}{n}\sum_{j=0}^{n-1} \frac{n}{2^{n-1}} \sum_{\substack{S\subseteq D-\{i\} \\ |S|=j}} V(S\cup\{i\}) - V(S) = \mathbb{E}_{S\subseteq D-\{i\}}[V(S\cup\{i\}) - V(S)]$
  - Note that it is also the semi-value s.t. $\sum_{j=0}^{n-1}\binom{n-1}{j}\frac{n}{2^{n-1}} = n$
  - $\frac{n}{2^{n-1}}$ is not a function of cardinality, which leads large weights to subsets of large cardinality
  - Thm. Banzhaf value achieves the largest safety margin among the semi-values
    - Safety margin : the largest tolerable perturbation on V that keeps the value order unchanged

- Idea 2 : Maximum Sample Reuse (MSR) Monte Carlo
  - Sample a number of subsets with various cardinalities $\mathcal{S} = \{S_1, \dots, S_m\} = \mathcal{S}_{\ni i} \cup \mathcal{S}_{\not\ni i}$
  - $\psi_i = \frac{1}{|\mathcal{S}_{\ni i}|}\sum_{S\in\mathcal{S}_{\ni i}} V(S) - \frac{1}{|\mathcal{S}_{\not\ni i}|}\sum_{S\in\mathcal{S}_{\not\ni i}} V(S)$ ⇒ enable faster estimation

Banzhaf III, John F. "Weighted voting doesn't work: A mathematical analysis." *Rutgers L. Rev.* 19 (1964): 317.

E.O.D