VTruST: Controllable value function based subset selection for Data-Centric Trustworthy AI

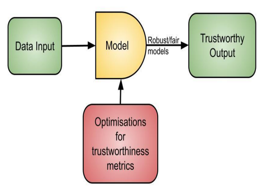
Soumi Das *[1,3], Shubhadip Nag *[1], Shreyyash Sharma [1], Suparna Bhattacharya [2], Sourangshu Bhattacharya [1]

1. Indian Institute of Technology Kharagpur 2. AI Research Lab, Hewlett Packard Labs, Bengaluru, India
3. Max Planck Institute for Software Systems (MPI-SWS), Saarbrücken, Germany

Data Centric Trustworthy AI (DCTAI)

Research Goal

1. Trustworthy AI is mostly model centric [1,2].



2. Data centric AI aims to create high quality datasets by optimizing the error rate [7].
3. Research question: *Can we design a data centric approach for implementing trustworthy AI?*
4. Research challenge:

Designing a "value function" capturing the value of a datapoint towards optimising trustworthiness metrics.

Designing a user controllable framework for providing weightage between the various metrics.

Contributions

We propose the framework **VTruST** that has 2 components:

General value function based framework for different trustworthy metrics: We propose the value functions for fairness and robustness that are used in our framework for approximation.

Algorithm for constructing high quality subsets: We pose the problem of data valuation as an **online sparse approximation** objective using Orthogonal Matching Pursuit(OMP). Our algorithm replaces selected datapoints with incoming ones on the fly, as long as they lead to a better approximation of the value function.

Difference with traditional OMP? OMP selects datapoints after **parsing the entire dataset.** Online OMP **parses the selected set (a much smaller set)** and the incoming datapoint over time to decide on replacement.

Social Datasets

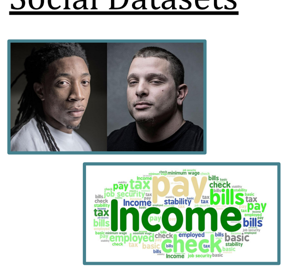
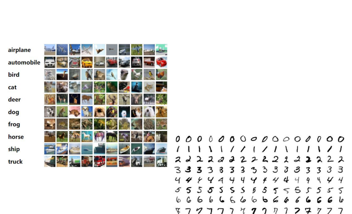
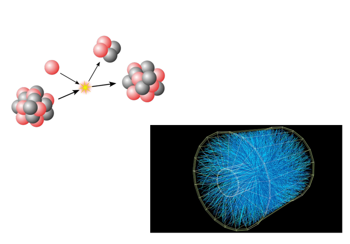


Image Datasets



Scientific Datasets



Controllable value function framework

Training set : $\mathcal{D} = \{d_i | i = 1, 2, .., N\}$; Validation set : $\mathcal{D}' = \{d'_j | j = 1, 2, .., M\}$

The value of a datapoint d_i at an epoch t can be attributed to its influence on the validation set D'.

[4] defined the value as the change in loss between epochs:

$$v_i^t(D') = l(\theta_t^{i-1}, D') - l(\theta_t^i, D') \quad \text{--- Equation 1}$$

The total value function can be written as:

$$\mathcal{V}(\mathcal{D}') = \sum_{t=1}^{T} \sum_{i=1}^{N} v_i^t(\mathcal{D}')$$

$$l(\theta_t^{i-1}, D') - l(\theta_t^i, D')$$

$$= \mathcal{V}_a(\mathcal{D}')$$ Value function for accuracy

The cumulative value function till epoch t can be written as:

$$\vec{y}_t = \sum_{k=1}^{t} \sum_{i=1}^{N} v_i^k(D')$$

Goal : Find a subset $\mathcal{S} \subset D$ $|\,|\mathcal{S}| < \omega$ that can approximate the above value function as :

$$\vec{y}_t \approx \sum_{d_i \in S_t \subseteq D} \alpha_i^t [\sum_{k=1}^{t} v_i^k(D')]$$

Using Taylor series expansion over Equation 1 and using SGD update, we can rewrite the above approximation as

$$\vec{y}_t \approx \sum_{d_i \in S_t \subseteq D} \alpha_i^t [\sum_{k=1}^{t} X_i^k]$$

$\vec{X}_i^k$ defines the feature of a datapoint d_i at epoch k which is derived from the expansion as :

$$\vec{X}_i^k = \nabla l(\theta_k^{i-1}, d_i)^T \nabla l(\theta_k^{i-1}, D') + \frac{(\nabla l(\theta_k^{i-1}, d_i)^T \nabla l(\theta_k^{i-1}, D'))^2}{2}$$

Challenge: Storing the features of all training datapoints over all epochs is prohibitively expensive.

Proposed solution: Online Sparse Approximation (OSA) method, **VTruST** , for solving the approximation problem:

$$\vec{y}_t \approx \sum_{(p,q) \in S_t} \alpha_p^q \vec{X}_p^q$$

The features $\vec{X}$ are derived from the value function $\mathcal{V}(\mathcal{D}')$ that are defined for different trustworthy objectives.

Since we define additive value functions, we can combine them weighed by λ to construct different **user-controlled composite value functions:** $\mathcal{V}(\mathcal{D}') = \sum_f \lambda_f \mathcal{V}_f(\mathcal{D}')$

Value functions for trustworthy AI

Fairness value function

Let $x \in \mathcal{X}$ be the input domain, $\{y_0, y_1\} \in \mathcal{Y}$ be the true binary labels, $\{z_0, z_1\} \in \mathcal{Z}$ be the sensitive binary attributes.

Based on [3], equalised odds disparity (***ed***) is defined as the maximum difference in accuracy between sensitive groups preconditioned on the true labels:

$$ed(\theta, \mathcal{D}') = max(\| l(\theta, \mathcal{D}'_{y_0, z_0}) - l(\theta, \mathcal{D}'_{y_0, z_1}) \|, \| l(\theta, \mathcal{D}'_{y_1, z_0}) - l(\theta, \mathcal{D}'_{y_1, z_1}) \|)$$

We define the **fairness value function** as the change in equalised odds disparity :

$$\mathcal{V}_f(\mathcal{D}') = \sum_{t=1}^{T} \sum_{d_i \in \mathcal{D}} ed(\theta_t^i, \mathcal{D}') - ed(\theta_t^{i-1}, \mathcal{D}')$$

Robustness value function

We use various perturbations to create the augmented training $\mathcal{D}_a$ and validation $\mathcal{D}'_a$ sets.

The perturbations includes adding noise or several transformations.

The selected subset includes a mix of unaugmented and augmented datapoints that aim to optimise the **robustness value function** :

$$\mathcal{V}_r(\mathcal{D}'_a) = \sum_{t=1}^{T} \sum_{d_i \in \{\mathcal{D} \cup \mathcal{D}_a\}} l(\theta_t^i, \mathcal{D}'_a) - l(\theta_t^{i-1}, \mathcal{D}'_a)$$

Controllable composite value functions

Accuracy-Fairness: $\mathcal{V}_{af}(\mathcal{D}') = \lambda \mathcal{V}_a(\mathcal{D}') + (1-\lambda)\mathcal{V}_f(\mathcal{D}')$

Accuracy-Robustness: $\mathcal{V}_{ar}(\mathcal{D}', \mathcal{D}'_a) = \lambda \mathcal{V}_a(\mathcal{D}') + (1-\lambda)\mathcal{V}_r(\mathcal{D}'_a)$

Robustness-Fairness: $\mathcal{V}_{rf}(\mathcal{D}', \mathcal{D}'_a) = \lambda \mathcal{V}_r(\mathcal{D}'_a) + (1-\lambda)\mathcal{V}_f(\mathcal{D}')$

Online sparse approximation algorithm

Online OMP based algorithm adding datapoints sequentially till buffer is full followed by replacement of selected data with incoming ones.

Algorithm 1 : VTruST

- **Input:**
  - i. ω : Total number of datapoints to be selected
  - ii. $\vec{y}$: Targeted value function
  - iii. $\vec{X}_i$: Features of all training points $d_i \in \mathcal{D}$
  - iv. S : Set of selected datapoint indices
  - v. $\vec{\alpha} \in \mathbb{R}^{|S|}$: Weight of selected datapoints
- **Initialize:**
  - $S \longleftarrow \phi$ //Indices of selected datapoints
- **for** each epoch $t \in \{1, 2, ..., T\}$ **do**
  - **for** each datapoint $d_i \in \mathcal{D}$ **do**
    - **Input:** $\vec{y}_t, X_i^t \quad \forall i \in \{1, 2, .., N\}, ||X_i^t||_2 = 1$
    - **Process:**
    - **if** $|S_{t-1}| = \omega$ **then**
      - $S_t \leftarrow$ **DataReplace**$(\vec{y}_t, \vec{\xi}_{t-1}, S_{t-1}, \vec{\alpha}^{t-1}, \vec{X}_i^t)$
    - **else**
      - $S_t \longleftarrow S_{t-1} \cup \{i\}$ // Add datapoints till the cardinality of S_t reaches ω
    - **end if**
    - Update $\vec{\alpha}^t = \text{argmin}_\alpha \|\vec{y}_t - \sum_{p,q \in S_t} (\alpha_q^p \vec{X}_q^p)||_2$
    - Update $\vec{\xi}_t = \sum_{p,q \in S_t} \alpha_q^p \vec{X}_q^p$
  - **end for**
- **end for**
- **Output**:Final set of selected datapoint indices S_T, learned coefficients $\{\alpha_q^p | p, q \in S_T\}$

Selected datapoints

Feature of incoming datapoint

Replacement criteria: Select the datapoints that contribute to a better approximation of the current value function $\vec{y}_t$

Algorithm 2 :DataReplace$(\vec{y}_t, \vec{\xi}_{t-1}, S_{t-1}, \vec{\alpha}^{t-1}, \vec{X}_i^t)$ - *Replace an existing datapoint.*

- $\vec{\rho}_t = \vec{y}_t - \vec{\xi}_{t-1}$
- $\pi_{max} = -\infty$
- $(a, b) = \phi$
- $\pi \longleftarrow \text{abs}(\vec{X}_i^t . \vec{\rho}_t)$ — **Projection of the feature of incoming data**
- **for** each index $p, q \in S_{t-1}$ **do**
  - $\pi' \longleftarrow \text{abs}(\vec{X}_q^p . \vec{\rho}_t)$ — **Projection of the feature of selected data**
  - $\gamma \longleftarrow \alpha_q^p$
  - **if** $\pi > \pi'$ & $\gamma \leq 0$ & $(\pi' + \gamma) > \pi_{max}$ **then**
    - $\pi_{max} \longleftarrow \pi' + \gamma$
    - $a, b \longleftarrow p, q$
  - **end if**
- **end for**
- **if** $(a, b) \neq \phi$ **then**
  - $S_t \longleftarrow S_{t-1} \setminus \{a, b\} \cup \{t, i\}$
- **end if**
- return S_t

The datapoint whose impact is < that of incoming datapoint but > than all others in the selected set gets replaced.

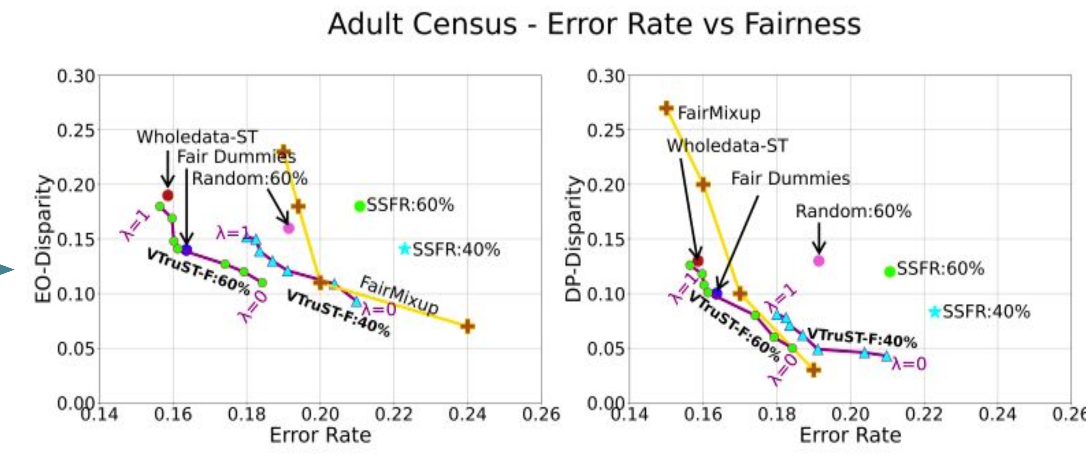
Controlling error rate / accuracy and fairness on social data

Methods	AdultCensus ER ±std	AdultCensus EO Disp ±std	AdultCensus DP Disp ±std
Wholedata-ST	0.16 ±0.002	0.19 ±0.06	0.13 ±0.06
Random	0.19 ±0.002	0.16 ±0.05	0.13 ±0.05
SSFR [5]	0.21 ±0.001	0.18 ±0.03	0.12 ±0.01
Fair- Dummies [2]	0.16 ±0.002	0.14 ±0.01	0.10 ±0.01
Fair- Mixup [6]	0.24 ±0.04	0.11 ±0.05	0.1 ±0.02
VTruST-F	0.18 ±0.001	**0.11 ±0.03**	**0.05 ±0.01**

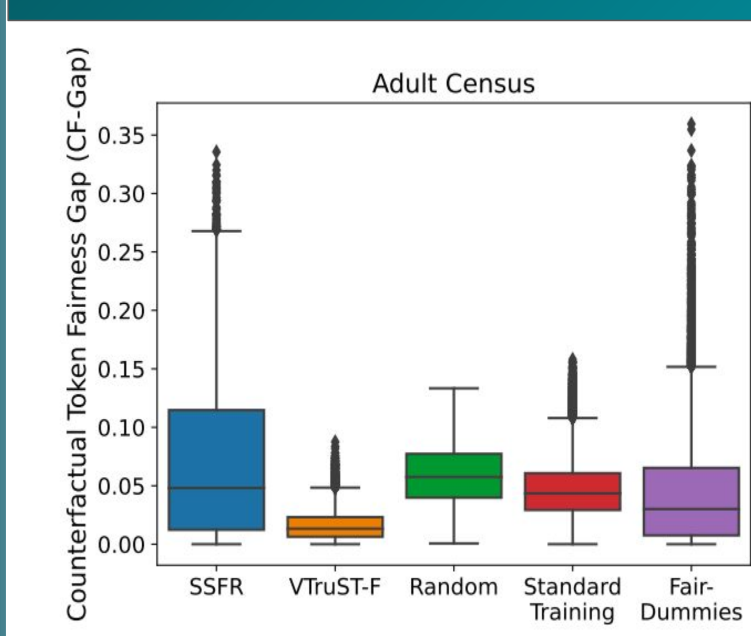
- Measuring error rate (ER) and fairness metrics (EO Disp and DP Disp) after training on 60% subsets.
- VTruST-F performs the closest to Whole Data in terms of utility and the best in terms of fairness.

- Tradeoff between utility (ER) and fairness metrics across different weightage of λ in Accuracy-Fairness value function.
- VTruST-F can achieve lower ER by retaining low disparity compared to other methods.

Adult Census - Error Rate vs Fairness



Post hoc explanations through data centric analysis



- Given a selected instance *x*, we generate a counterfactual instance *x'* by altering its sensitive attribute and define CF-Gap(x) [8] as *|| f(x) – f(x') ||*
- Lower the gap, less is its dependence on sensitive attributes.
- VTruST-F achieves the lowest CF-Gap justifying its retainment of fair subsets that lead to fair models.

- Anecdotal samples on the basis of high CF-Gap.
- SSFR has a large number of redundant samples with similar attribute values.
- VTruST-F which anyway has relatively lower CF-gap contains a diverse set of samples.

VTruST-F *Feat*	VTruST-F Rel	VTruST-F Race	VTruST-F Sex	VTruST-F NC	SSFR *Feat*	SSFR Rel	SSFR Race	SSFR Sex	SSFR NC
D_1	ORel	B	F	JM	D_1	Husb	W	M	US
D_2	NIF	W	M	US	D_2	Husb	W	M	US
D_3	NIF	W	M	US	D_3	Husb	W	M	US
D_4	OC	API	F	TW	D_4	Husb	W	M	US
D_5	Husb	W	M	US	D_5	Husb	W	M	US
D_6	UnM	W	F	US	D_6	Husb	W	M	US
D_7	Wife	W	F	US	D_7	NIF	W	M	US
D_8	OC	W	M	US	D_8	OC	W	M	US
D_9	NIF	AIE	F	Col	D_9	Husb	W	M	DE
D_{10}	UnM	W	F	DE	D_{10}	OC	W	M	US

[1] Wang et al. "Augmax: Adversarial composition of random augmentations for robust training." NeurIPS 2021.
[2] Romano, Yaniv, et al., "Achieving equalized odds by resampling sensitive attributes." NeurIPS 2020.
[3] Roh et al. "Fairbatch: Batch selection for model fairness." ICLR 2021.
[4] Pruthi et al. "Estimating training data influence by tracing gradient descent." NeurIPS 2020.
[5] Roh et al. "Sample selection for fair and robust training." NeurIPS 2021.
[6] Chuang et al.,. "Fair mixup: Fairness via interpolation." ICLR 2021
[7] Paul, et al., "Deep learning on a data diet: Finding important examples early in training." NeurIPS 2021.
[8] Garg et al. "Counterfactual fairness in text classification through robustness." AAAI, AI, Ethics, and Society. 2019.

https://github.com/SoumiDas/VTruST
soumidas@mpi-sws.org
soumi-das
https://soumidas.github.io/