39th Annual Conference on Neural Information Processing Systems (NeurIPS2025)

Learning Gradient Boosted Decision Trees with Algorithmic Recourse

Kentaro Kanamori[1], Ken Kobayashi[2], Takuya Takagi[1]

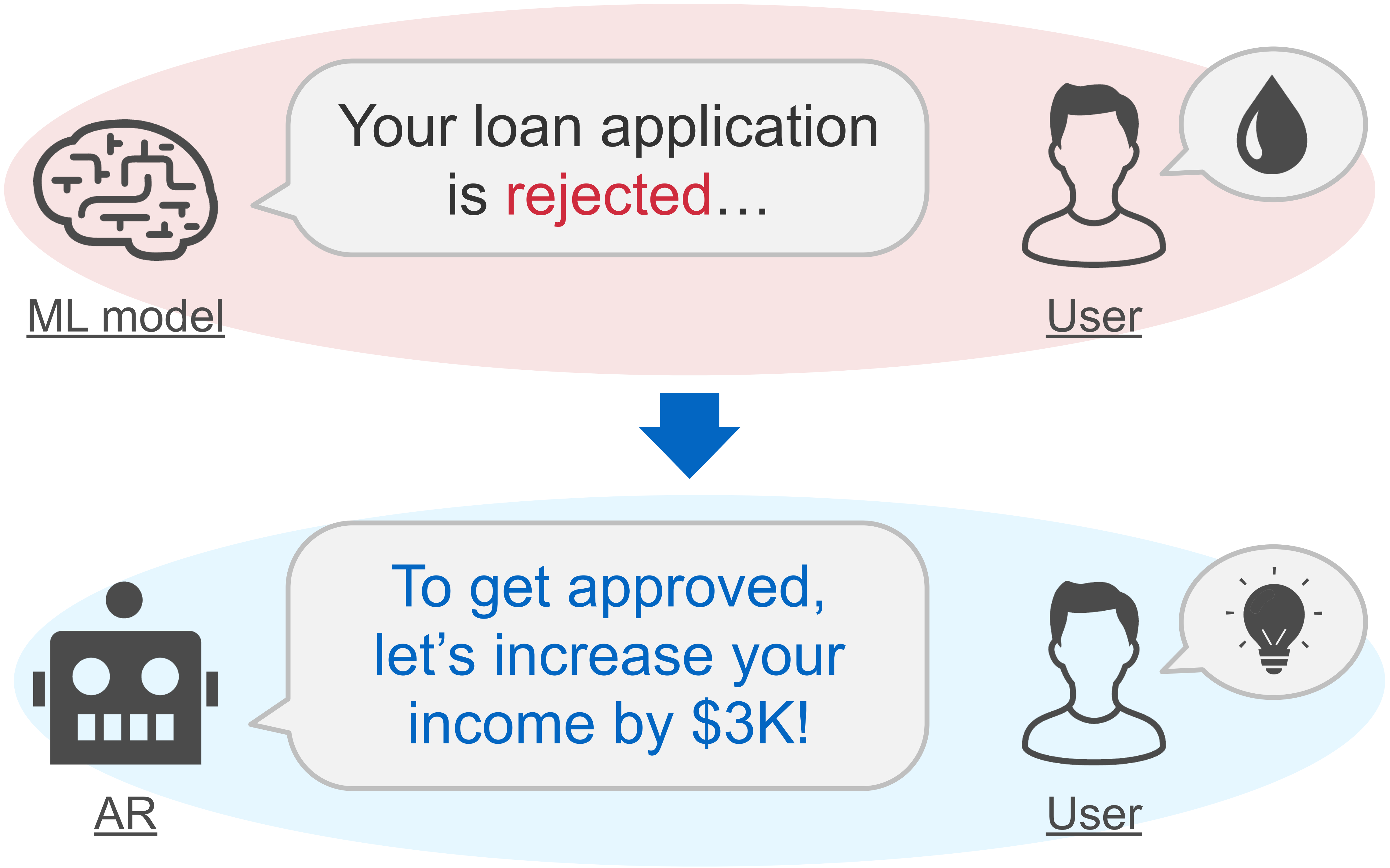
[1]Fujitsu Limited [2]Institute of Science Tokyo

Background

Algorithmic recourse aims to provide “actions” for altering unfavorable predictions

Algorithmic Recourse [Ustun+ 19]

Explaining a “***recourse action***” for obtaining a desirable prediction result from an ML model



Algorithmic Recourse (AR) [Ustun+ 19]

Given a learned model $f\colon \mathcal{X} \to \mathcal{Y}$, an input instance $x \in \mathcal{X}$, and a desirable class $y^* \in \mathcal{Y}$, find an action a^* such that

$$a^* = \arg\min_{a \in \mathcal{A}(x)} c(a \mid x) \quad \text{s.t.} \quad f(x+a) = y^*$$

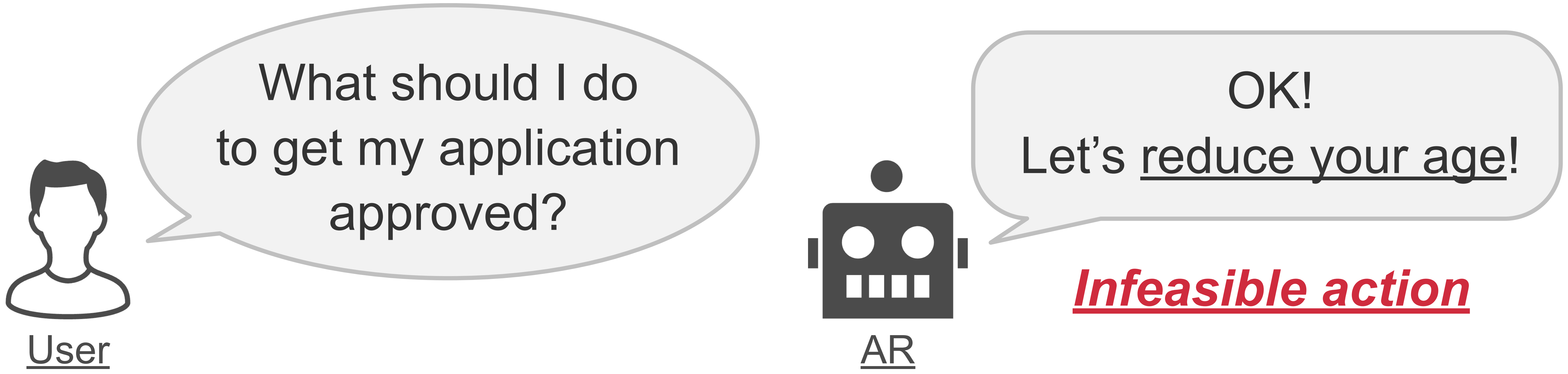
where $\mathcal{A}(x)$ is a set of feasible actions and c is a cost function.

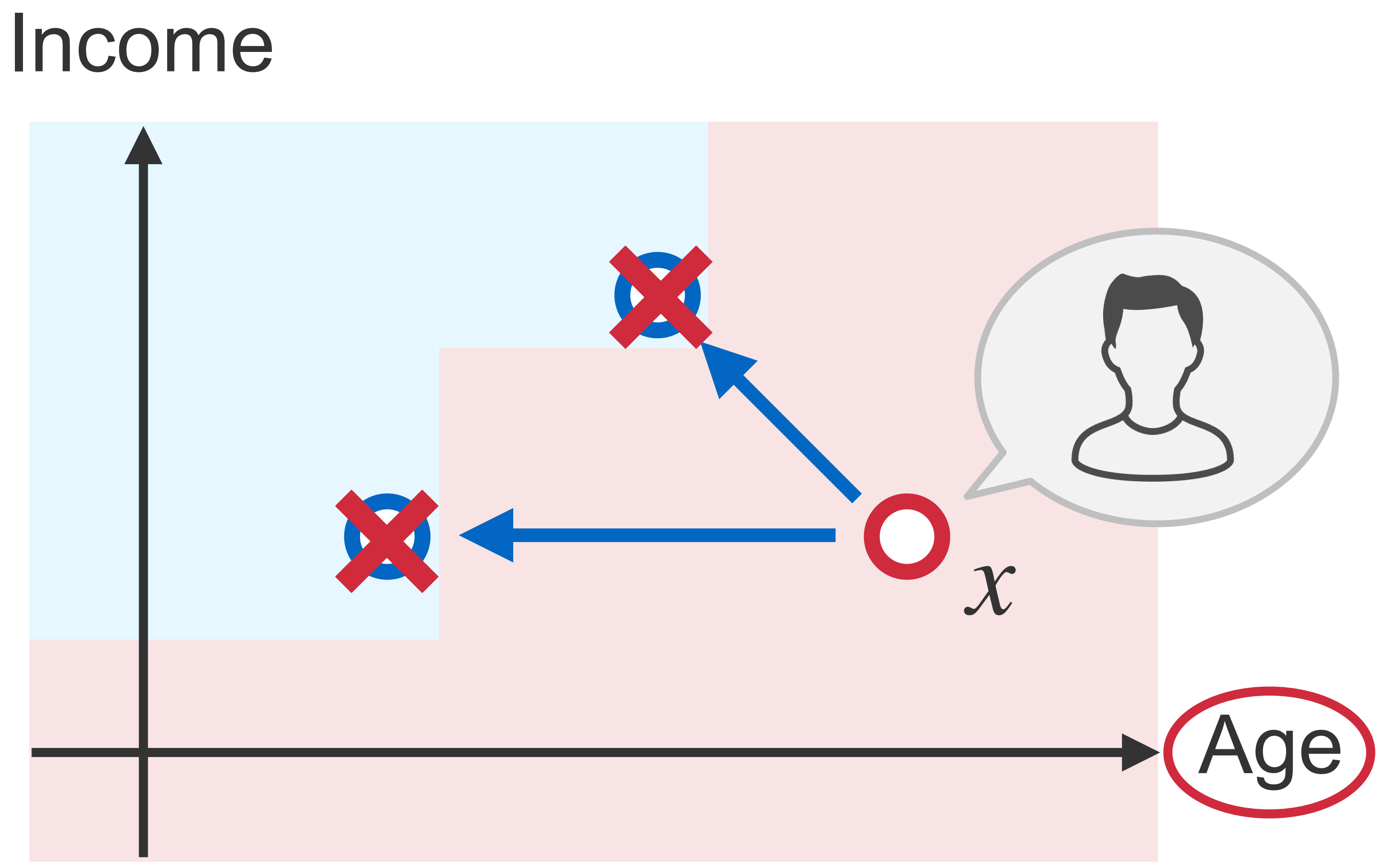
- Provide a recourse action a that
  (1) is feasible with a low-cost (***executable***);
  (2) alters the current prediction result (***valid***)

Motivation

There is no guarantee that executable valid actions always exist for a learned model

Most of existing studies focus on post-hoc methods for a given learned model, however…





- There is no executable and valid action for this model…

Our Goal

Learn a model that <u>guarantees the existence of executable and valid actions</u> with high probability

Problem Formulation

Learning an accurate model while ensuring the existence of executable valid actions

Learning with Algorithmic Recourse [Ross+ 21; Kanamori+ 24]

Given a sample $S = \{(x_n, y_n)\}_{n=1}^{N}$, model class $\mathcal{F}$, and $\gamma \geq 0$, we consider the following learning problem:

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{n=1}^{N} l(y_n, f(x_n)) + \gamma \cdot l_\beta(x_n \mid f)$$

Evaluate whether at least one executable and valid action exists

where $l_\beta(x \mid f) := \min_{a \in \mathcal{A}_\beta(x)} l(y^*, f(x + a))$ is the *recourse loss*, $\beta \geq 0$ is a cost budget parameter, and $\mathcal{A}_\beta(x) := \{a \in \mathcal{A}(x) \mid c(a \mid x) \leq \beta\}$ is the set of feasible actions whose costs are less than β.

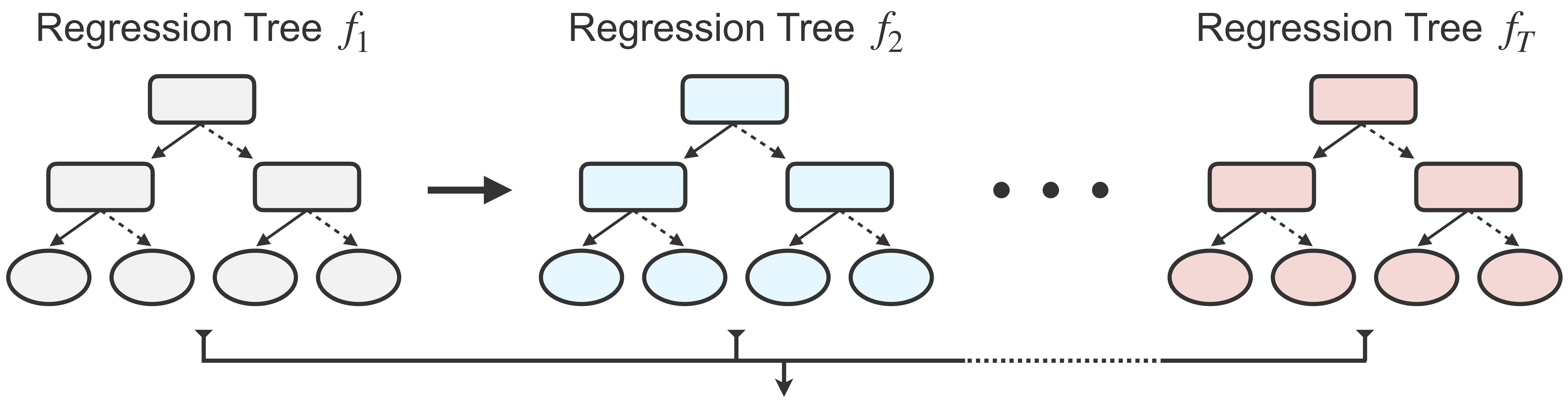
- We aim to learn a model that can guarantee the existence of executable valid actions for as many instances as possible without degrading its accuracy and efficiency

Our Approach: Overview

Learning a tree ensemble model by gradient boosting with the recourse loss

This paper focusses on ***gradient boosted decision trees (GBDTs)*** [Friedman 00; Chen+ 16; Ke+ 17]

∵ GBDTs are the state-of-the-art models for tabular datasets (e.g., finance & justice) [Grinsztajn+ 22]



Ensemble $f(x) = \sum_{t=1}^{T} f_t(x)$

XGBoost

LightGBM

CatBoost

Several popular libraries exist!

Our Framework

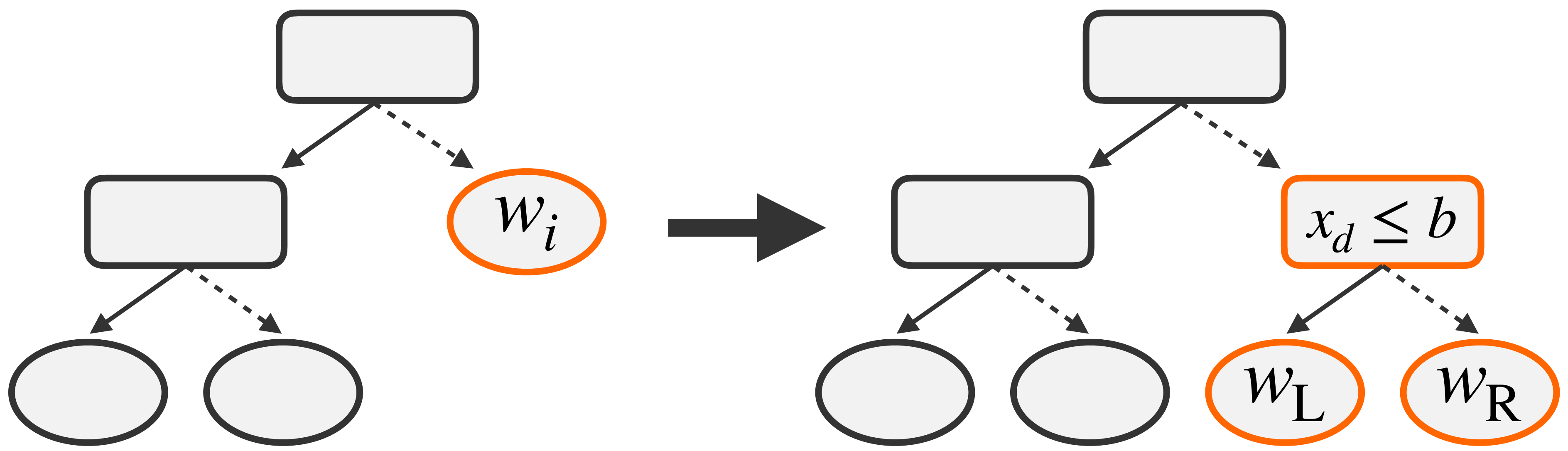
Learning each tree by gradient boosting with recourse loss and post-hoc refinement of leaf weights

Our Approach: Recourse-Aware Gradient Boosting

Deriving closed-form approximate solutions with the differentiable recourse loss

Standard Method (e.g., XGBoost)

Top-down greedy splitting & closed-form approximate solutions by <u>Taylor expansion of loss function</u>



$$\min_{d,b} \min_{w_{\mathrm{L}}, w_{\mathrm{R}}} \Phi_{d,b}(w_{\mathrm{L}}, w_{\mathrm{R}}) \approx G_{\mathrm{L}}^{(d,b)} \cdot w_{\mathrm{L}} + \frac{1}{2} \cdot H_{\mathrm{L}}^{(d,b)} \cdot w_{\mathrm{L}}^2 + G_{\mathrm{R}}^{(d,b)} \cdot w_{\mathrm{R}} + \frac{1}{2} \cdot H_{\mathrm{R}}^{(d,b)} \cdot w_{\mathrm{R}}^2$$

Able to obtain closed-form solutions for w_{L} and w_{R}

Challenge

Closed-form solutions are not trivial for our case since the <u>recourse loss is not differentiable</u>

Our Idea

Deriving a <u>differentiable upper bound</u> of recourse loss

<u>Standard Problem</u>

$$\min_{d,b} \min_{w_{\mathrm{L}}, w_{\mathrm{R}}} \Phi_{d,b}(w_{\mathrm{L}}, w_{\mathrm{R}})$$

<u>Our Problem</u>

$$\min_{d,b} \min_{w_{\mathrm{L}}, w_{\mathrm{R}}} \Phi_{d,b}(w_{\mathrm{L}}, w_{\mathrm{R}}) + \gamma \cdot \Psi_{d,b}(w_{\mathrm{L}}, w_{\mathrm{R}})$$

<u>Our Surrogate Problem (differentiable)</u>

$$\min_{d,b} \min_{w_{\mathrm{L}}, w_{\mathrm{R}}} \Phi_{d,b}(w_{\mathrm{L}}, w_{\mathrm{R}}) + \gamma \cdot \tilde{\Psi}_{d,b}(w_{\mathrm{L}}, w_{\mathrm{R}})$$

Proposition 2. (Time complexity)

Our algorithm approximately solves the above problem in $\mathcal{O}(N \cdot D)$

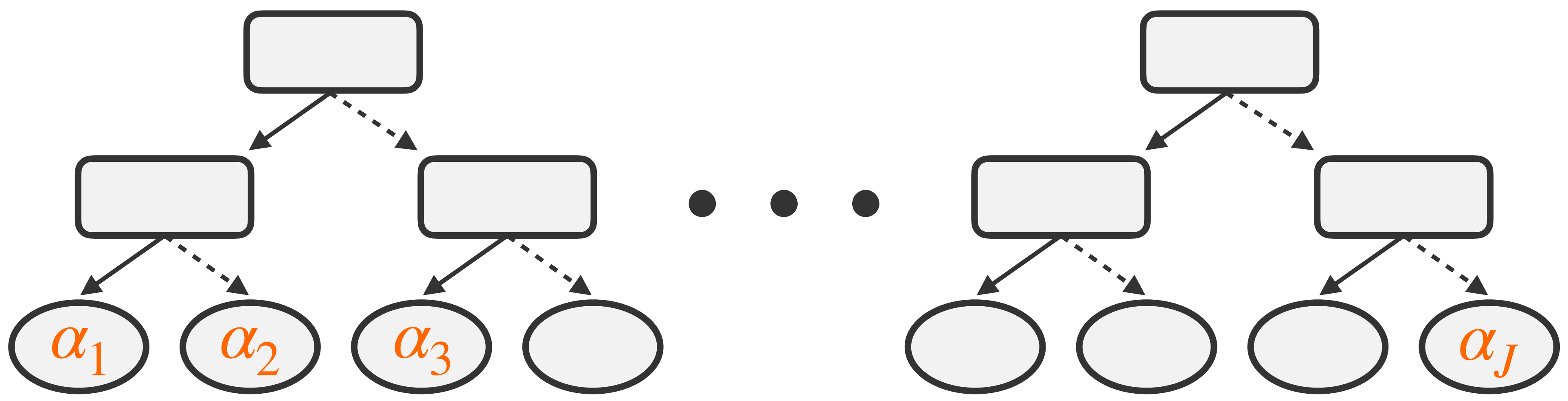
- Achieve the same complexity as the standard one!

Our Approach: Recourse-Aware Leaf Refinement

Modifying leaf weights of a learned model to probably guarantee recourse actions

Our Idea

To control the existence ratio of recourse actions, refining leaf weights of a learned model



Optimizing leaf weights $\alpha = (\alpha_1, \dots, \alpha_J)$ while fixing learned tree structures

Leaf Refinement Problem

$$\min_{\alpha \in \mathbb{R}^J} \frac{1}{N} \sum_{n=1}^{N} l(y_n, f_\alpha(x_n)) \quad \text{s.t.} \quad \frac{1}{N} \sum_{n=1}^{N} l_\beta(x_n \mid f_\alpha) \leq \varepsilon$$

- Can be solved by repeated fitting of linear classifiers

Theoretical Analysis

Deriving a PAC-style bound on the estimation error of the recourse loss over a training sample

Proposition 3. (PAC-style analysis of refined models)

For a model f, let $\mathscr{R}_\beta(f) := \mathbb{P}_x[\forall a \in \mathscr{A}_\beta(x) : f(x+a) \neq y^*]$ be the expected recourse risk. Then, a refined model f_α satisfies the following inequality with probability at least $1 - \delta$:

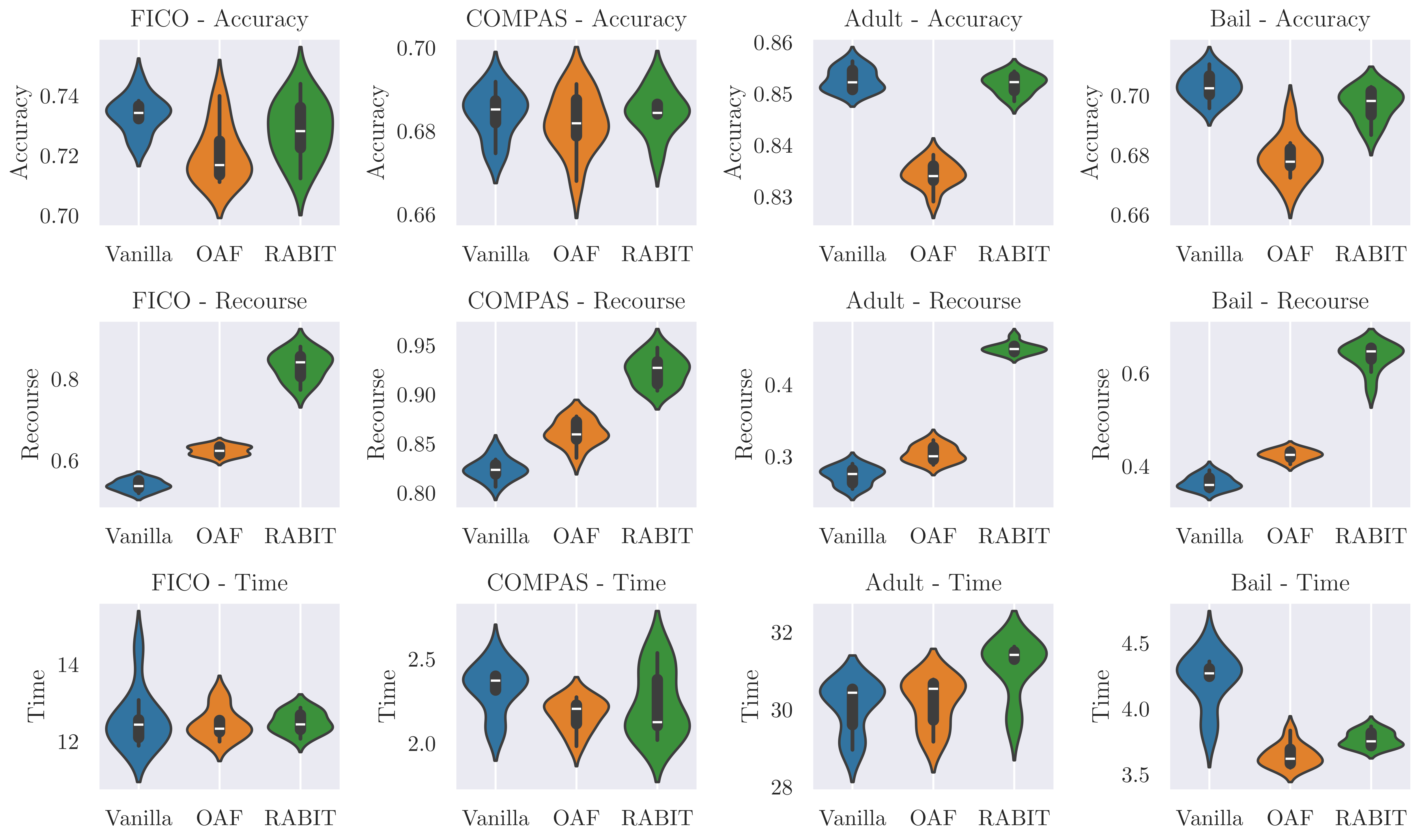
$$\mathscr{R}_\beta(f_\alpha) \leq \frac{1}{N} \sum_{n=1}^{N} l_\beta(x_n \mid f_\alpha) + \sqrt{\frac{8 \cdot \ln \frac{e \cdot N}{4}}{N}} + \sqrt{\frac{\ln \frac{1}{\delta}}{2 \cdot N}}$$

- We can also control the probability of the existence of executable valid actions for unseen instances

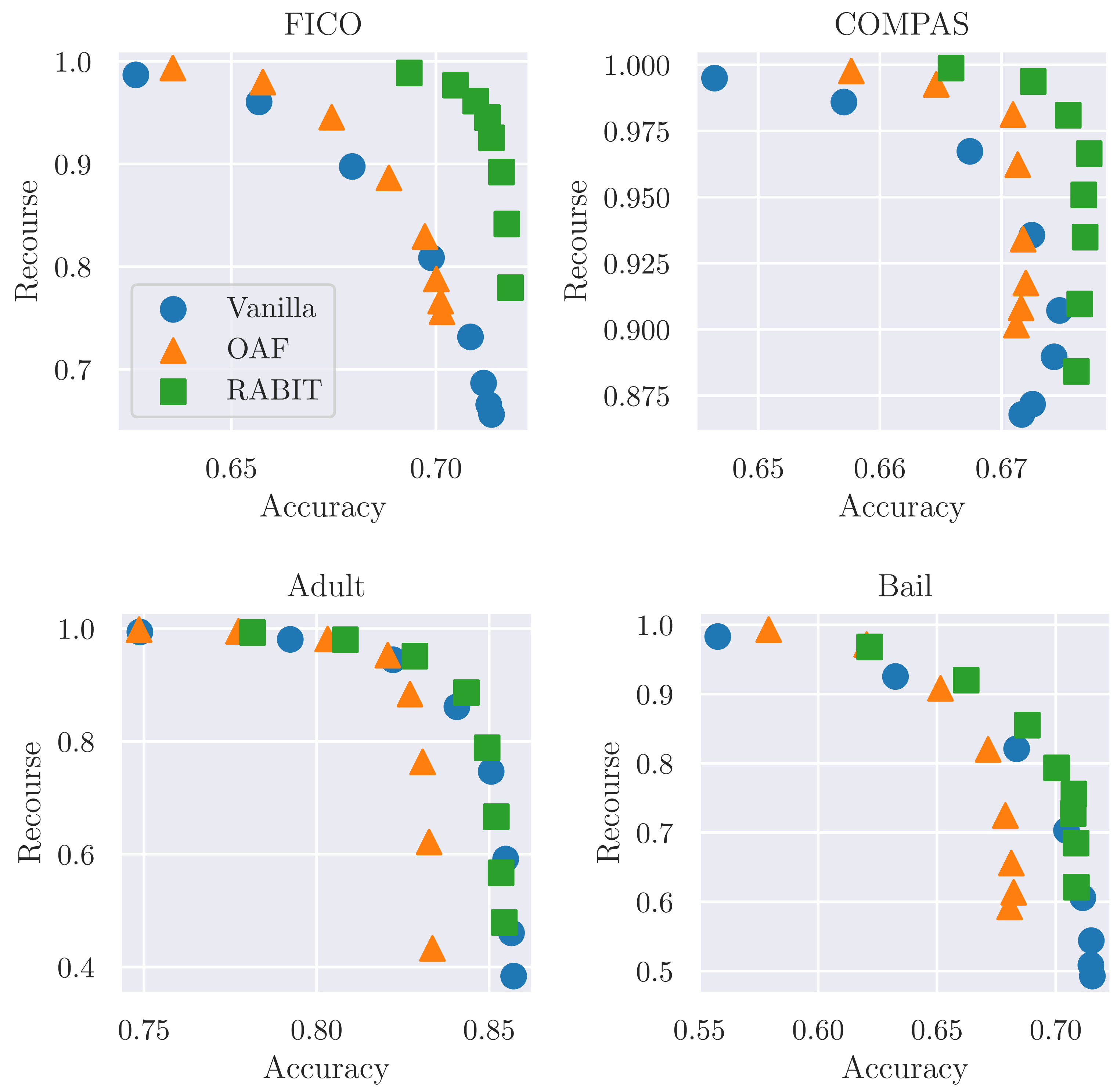
Experiments

Attained higher recourse ratio while maintaining comparable accuracy and efficiency

Exp. 1 Baseline Comparison (w/o leaf refinement)



Exp. 2 Efficacy of Leaf Refinement



- Our method (***RABIT***) significantly improved recourse ratio without degrading accuracy and efficiency

Summary

Learning GBDTs that can provide accurate predictions and executable valid actions

We propose ***Recourse-Aware gradient Boosted decIsion Trees (RABIT)***:

- Propose a new gradient boosting algorithm with the recourse loss
  - Its time complexity is equivalent to the standard ones (e.g., XGBoost)
- Introduce a leaf refinement method under the recourse constraint
  - Provide a PAC-style guarantee of the recourse action existence
- Demonstrate the efficacy of our method by experiments
  - Improve the recourse guarantee while maintaining accuracy and efficiency

RABIT

OpenReview

GitHub