Prediction without Preclusion: Recourse Verification with Reachable Sets

Avni Kothari UC San Diego
Bogdan Kulynych EPFL
Lily Weng UC San Diego
Berk Ustun UC San Diego

Motivation

Prediction Task to Repay a Loan

Can only increase from 0 to 1

Most accurate classifier

Features		Action Set	Reachable Set	Dataset		ERM	
has_phd	age ≥ 60	$A(x_1, x_2)$	$R_A(\boldsymbol{x})$	n^-	n^+	$\hat{f}$	$\hat{R}(\hat{f})$
0	0	$\{(1,1),(0,1),(1,0),(0,0)\}$	$\{(1,1),(0,1),(1,0),(0,0)\}$	10	25	+	10
0	1	$\{(1,0),(0,0)\}$	$\{(1,1),(0,1)\}$	11	25	+	11
1	0	$\{(0,1),(0,0)\}$	$\{(1,1),(1,0)\}$	12	25	+	12
1	1	$\{(0,0)\}$	$\{(1,1)\}$	27	15	-	15

Fixed Prediction

Why care?

- Consumers who are denied loans and interviews may be permanently locked out from access to credit and employment
- Models use features without accounting for how individuals can change them

Recourse Verification

Existing Solution: Recourse Provision

Test if $\boldsymbol{x} \in \mathcal{X}$ can attain a target prediction from a given model by performing actions on their features

$$\begin{aligned} \min \quad & \text{cost}(\boldsymbol{a} \mid \boldsymbol{x}) \\ \text{s.t.} \quad & f(\boldsymbol{x} + \boldsymbol{a}) = +1, \\ & \boldsymbol{a} \in A(\boldsymbol{x}), \end{aligned}$$

Actionability Constraints

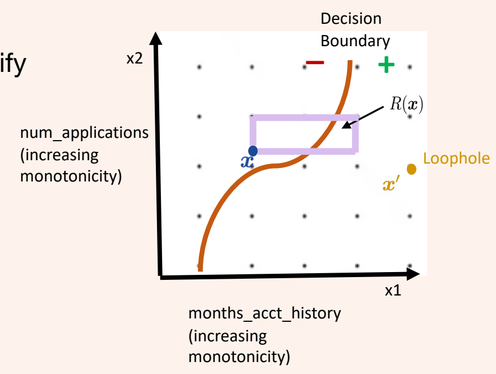
Feature constraints can be expressed in natural language and encoded as a constraint in an optimization problem

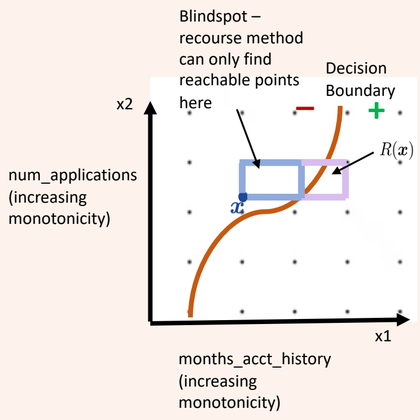
Constraint Type	Sample Constraint	Features	Encoding
Monotonicity	prior_applicant can only increase	x_j = prior_applicant	$a_j \geq 0$
Integrality	n_accounts must be positive integer ≤ 10	x_j = n_accounts	$a_j \in \mathbb{Z} \cap [0 - x_j, 10 - x_j]$
Encoding: Categorical Features	preserve one-hot encoding of married, single	x_j = married x_k = single	$a_j + a_k = 1,\ \{a_j, a_k\} \in \{0, 1\}$
Logical Implications	if is_employed = TRUE then work_hrs_per_week ≥ 0 else work_hrs_per_week $= 0$	x_j = is_employed, x_k = work_hrs_per_week	$a_j = x'_j - x_j \quad a_k = x'_k - x_k$ $x'_j \in \{0, 1\} \quad x'_k \in [0, 168],$ $x'_k \leq 168\, x'_j$

Pitfalls of Recourse Feasibility with Recourse Provision

Blindspots: An instance where a method fails to certify feasibility.

Loopholes: An instance where a method returns an action that violates actionability constraints.





Our Approach: Recourse Verification

We formally verify if a model assigns predictions that *every* individual can change

For all $\boldsymbol{x} \in \mathcal{X}$ we determine the feasibility of

$$\text{Recourse}(\boldsymbol{x}, f, A) = \begin{cases} \texttt{Yes}, & \text{only if there exists } \boldsymbol{a} \in A(\boldsymbol{x}) \text{ such that } f(\boldsymbol{x} + \boldsymbol{a}) = +1 \\ \texttt{No}, & \text{only if } f(\boldsymbol{x} + \boldsymbol{a}) = -1 \text{ for all } \boldsymbol{a} \in A(\boldsymbol{x}) \\ \perp, & \text{otherwise} \end{cases}$$

Why Verification?

- Infeasibility in recourse is not often discussed
- Recourse may not exist under realistic actionability constraints

Models in lending and hiring can assign fixed predictions that preclude access to credit and employment

Code

Paper

Verification with Reachable Sets

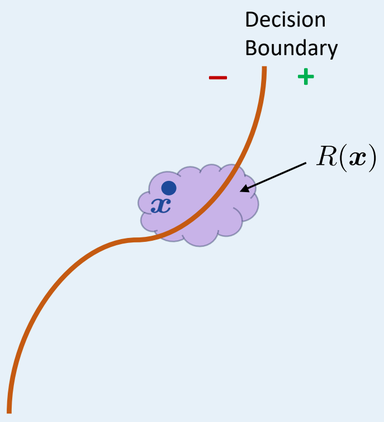
Reachable Set

Given a point $\boldsymbol{x}$ and action set $A(\boldsymbol{x})$, its reachable set contains all feature vectors that can be attained using an action

$$\boldsymbol{a} \in A(\boldsymbol{x}) : R(\boldsymbol{x}) := \{\boldsymbol{x} + \boldsymbol{a} | \boldsymbol{a} \in A(\boldsymbol{x})\}$$

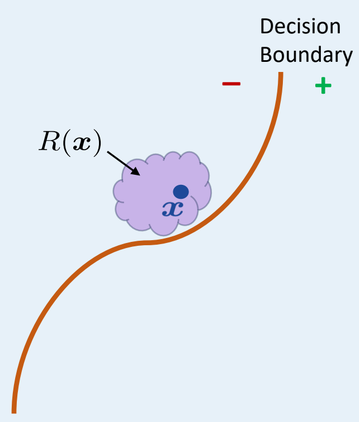
When is recourse feasible?

There exists a $\boldsymbol{x}' \in R(\boldsymbol{x})$ that is positively classified

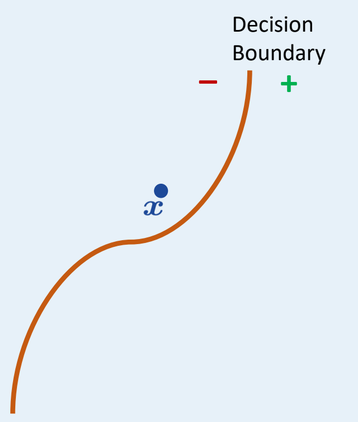


When is recourse infeasible?

For all $\boldsymbol{x}' \in R(\boldsymbol{x})$ they are negatively classified

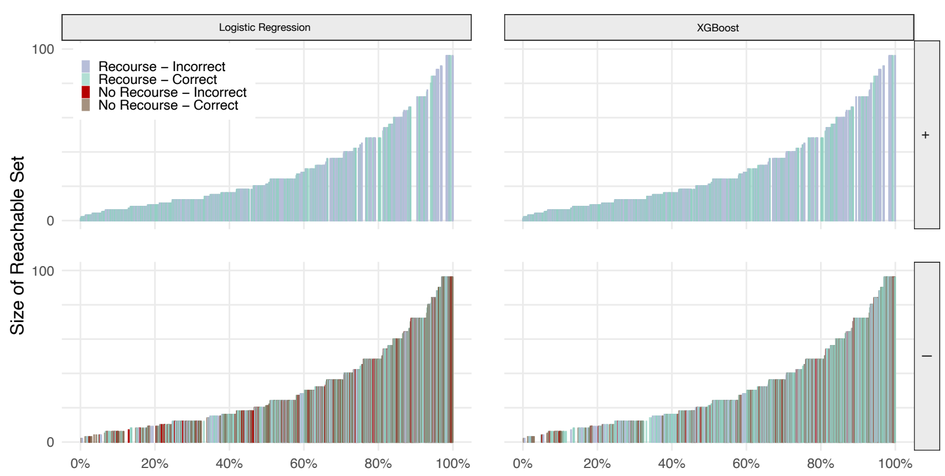


$R(\boldsymbol{x})$ is empty and $\boldsymbol{x}$ is negatively classified



Empirical Results

Dataset	Model Type	Metrics	Simple Reach	Simple AR	Simple DiCE	Separable Reach	Separable AR	Separable DiCE	Non-Separable Reach	Non-Separable AR	Non-Separable DiCE
heloc FICO [17] $n = 3,184$ $d = 29$	LR	Recourse	24.7%	85.2%	51.7%	29.0%	62.2%	48.2%	57.3%	23.5%	20.4%
		No Recourse	0.0%	14.8%	—	0.0%	37.8%	—	41.9%	37.8%	—
		Abstain	75.3%	—	—	71.0%	—	—	0.8%	—	—
		Loopholes	—	0.0%	0.0%	—	0.0%	0.0%	—	38.8%	27.8%
		Blindspots	—	0.0%	48.3%	—	0.0%	51.8%	—	0.0%	51.8%
	XGB	Recourse	84.4%		99.8%	35.1%		57.5%	73.2%		23.4%
		No Recourse	0.0%		—	0.0%		—	26.4%		—
		Abstain	15.6%	—	—	64.9%	—	—	0.5%	—	—
		Loopholes	—		0.0%	—		0.0%	—		34.1%
		Blindspots	—		0.2%	—		42.5%	—		42.5%



- Recourse may not exist under actionability constraints
- Recourse is feasible under simple constraints
- Infeasibility arises once we consider more complex constraints
- Recourse provision methods fail when used to determine recourse verification

- Predictions without recourse are prevalent amongst all sizes of reachable sets
- Predictions without recourse can vary significantly between different types of classifiers
- Heavily weighted immutable features can have significant impact on recourse