

Algorithmic Recourse for Machine Learning

Reviewing the state-of-the-art, novel research directions & an hands-on tutorial

by **Martin Pawelczyk**, Data Science and Analytics Research Group, University of Tübingen



Why do we need (counterfactual) explanations?

**Understand ML
decision making**

**Provide algorithmic
recourse**

**Understand
fairness issues**



Why do we need (counterfactual) explanations?

**Understand ML
decision making**

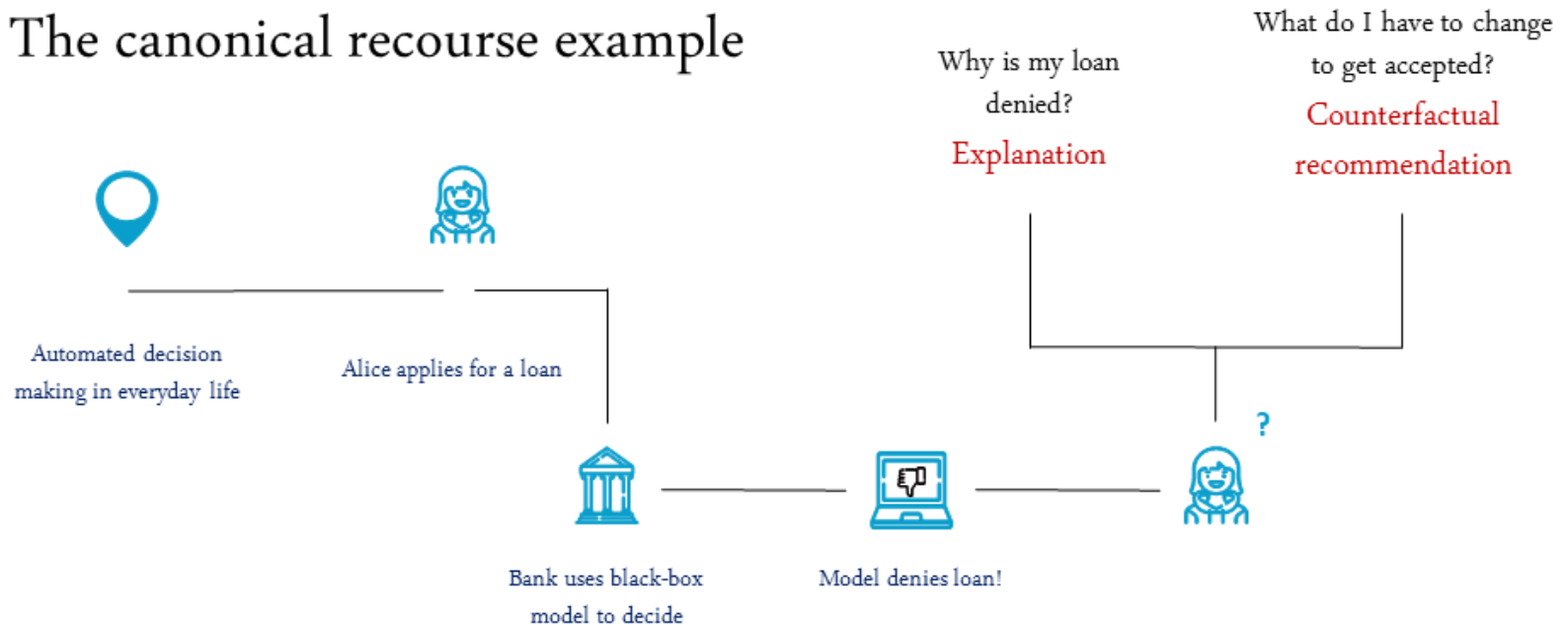
**Provide algorithmic
recourse**

**Understand
fairness issues**

Create Trust in the ML system

Practical Example: Finance

The canonical recourse example





Outline

PART 1 **Introduction to** **Algorithmic** **Recourse**

IMF recourses: Wachter et al (2018)
Manifold recourses: Pawelczyk et al (2020)
Causal recourses: Karimi et al (2021)

Outline

PART 1 **Introduction to** **Algorithmic** **Recourse**

IMF recourses: Wachter et al (2018)
Manifold recourses: Pawelczyk et al (2020)
Causal recourses: Karimi et al (2021)

PART 2a **Robust recourses in** **the presence of noisy** **human responses**

„Let Users Decide: Navigating the Tradeoff Between Costs and Robustness in Algorithmic recourse“
(Pawelczyk et al 2022 a)

PART 2b **On the tradeoff betw.** **recourses and the** **„right to be forgotten“**

„On the trade-off between actionable explanations and the right to be forgotten“ (Pawelczyk et al 2022 b)

Outline

PART 1 Introduction to Algorithmic Recourse

IMF recourses: Wachter et al (2018)
Manifold recourses: Pawelczyk et al (2020)
Causal recourses: Karimi et al (2021)


PART 2a Robust recourses in the presence of noisy human responses

„Let Users Decide: Navigating the Tradeoff Between Costs and Robustness in Algorithmic recourse“
(Pawelczyk et al 2022 a)

PART 2b On the tradeoff betw. recourses and the „right to be forgotten“

„On the trade-off between actionable explanations and the right to be forgotten“ (Pawelczyk et al 2022 b)

PART 3 (Tutorial) CARLA: a recourse library

„CARLA: A python library to benchmark algorithmic recourse and counterfactual explanation algorithms“
(Pawelczyk et al 2021) 



Part 1 – An introduction to algorithmic recourse: challenges and solutions



Challenges for Alg. Recourse on Tabular Data

Desiderata

Potential Issues

Challenges for Alg. Recourse on Tabular Data

Desiderata

Generate „realistic“
looking recourses

Potential Issues

1. Features are dependent of each other
2. Features can have various feature types (e.g., discrete, continuous)
3. Recourses should be „human interpretable“

Challenges for Alg. Recourse on Tabular Data

Desiderata

Generate „realistic“
looking recourses

Recourses should not be
fragile

Potential Issues

1. Features are dependent of each other
2. Features can have various feature types (e.g., discrete, continuous)
3. Recourses should be „human interpretable“

Distribution shifts / models updates / etc. can
invalidate prescribed recourses

Challenges for Alg. Recourse on Tabular Data

Desiderata

Generate „realistic“
looking recourses

Recourses should not be
fragile

Potential Issues

1. Features are dependent of each other
2. Features can have various feature types (e.g., discrete, continuous)
3. Recourses should be „human interpretable“

Distribution shifts / models updates / etc. can
invalidate prescribed recourses

Algorithmic recourse should be compatible with other GDPR principles!?

Prescribed recourses should not leak private information about other users



Issue 1 – „Human interpretability of recourses“

Sparsity is a fundamental requirement of algorithmic recourse.

Issue 1 – „Human interpretability of recourses“

Sparsity is a fundamental requirement of algorithmic recourse.

Proposal	Input subset	current value		required
1	# credit cards	5	→	3
2	current debt	\$3250	→	\$1000
3	has savings account	0	→	1
	has retirement account	0	→	1

Ustun et al (2019)

Changing fewer features
means less things can go
wrong

Clear and short instructions
help building trust in the
system

Issue 1 – „Human interpretability of recourses“

Sparsity is a fundamental requirement of algorithmic recourse.

Proposal	Input subset	current value		required
1	# credit cards	5	→	3
2	current debt	\$3250	→	\$1000
3	has savings account	0	→	1
	has retirement account	0	→	1

Ustun et al (2019)

Changing fewer features
means less things can go
wrong

Clear and short instructions
help building trust in the
system

Takeaway: Sparse solutions help to build trust in the recourse system.



Issue 2 – „Features are dependent of each other“

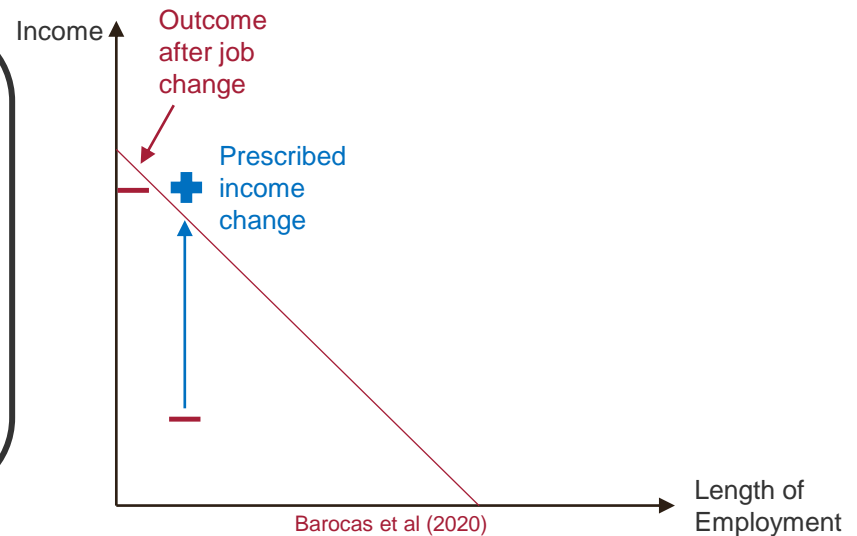
Input dependencies matter for precise instructions!

Issue 2 – „Features are dependent of each other“

Input dependencies matter for precise instructions!

Example

- Data on income & tenure
- Use f for loan approval
- Suggest income change, but ignore the data (e.g., input correlations)
- Leads to *sparse recourse*

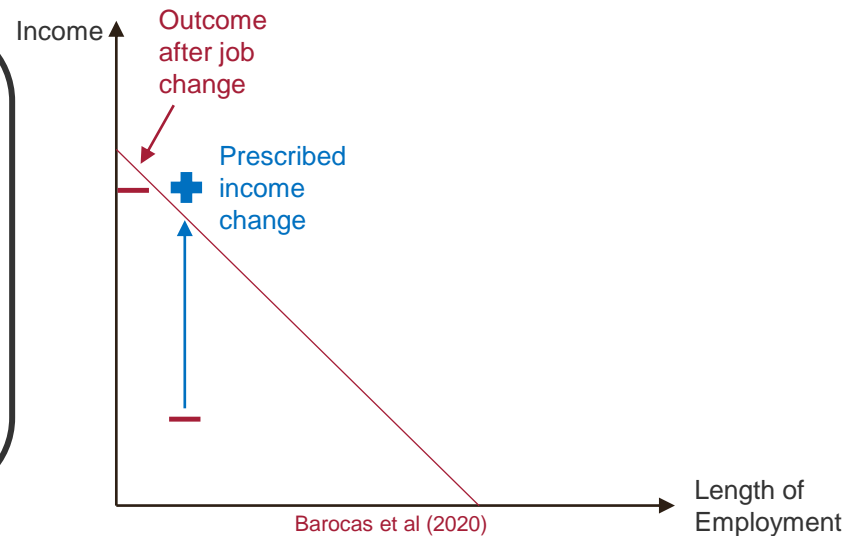


Issue 2 – „Features are dependent of each other“

Input dependencies matter for precise instructions!

Example

- Data on income & tenure
- Use f for loan approval
- Suggest income change, but ignore the data (e.g., input correlations)
- Leads to *sparse recourse*



Takeaway: Relying too heavily on the sparsity principle can make recourses fragile.

Issue 3 – „Features can have various types“

ordinal categorical categorical binary

Age	Education	Occupation	Sex	Income
39	Bachelors	Adm-clerical	Male	≤50K
50	Bachelors	Exec-managerial	Male	>50K
38	HS-grad	Handlers-cleaners	Male	≤50K
53	11th	Handlers-cleaners	Male	≤50K
28	Bachelors	Prof-specialty	Female	>50K

Borisov et al (2021)

Issue 3 – „Features can have various types“

ordinal categorical categorical binary

Age	Education	Occupation	Sex	Income
39	Bachelors	Adm-clerical	Male	≤50K
50	Bachelors	Exec-managerial	Male	>50K
38	HS-grad	Handlers-cleaners	Male	≤50K
53	11th	Handlers-cleaners	Male	≤50K
28	Bachelors	Prof-specialty	Female	>50K

Borisov et al (2021)

How can we efficiently search over this space and still produce recourses that lie on the data manifold?

Issue 3 – „Features can have various types“

ordinal categorical categorical binary

Age	Education	Occupation	Sex	Income
39	Bachelors	Adm-clerical	Male	≤50K
50	Bachelors	Exec-managerial	Male	>50K
38	HS-grad	Handlers-cleaners	Male	≤50K
53	11th	Handlers-cleaners	Male	≤50K
28	Bachelors	Prof-specialty	Female	>50K

Borisov et al (2021)

How can we efficiently search over this space and still produce recourses that lie on the data manifold?

Takeaway: On tabular data, the search for counterfactual explanations should take the feature type in account.

Solution I: Algorithmic Recourse under the IMF Assumption (Wachter et al (2018)) – Objective

Goal

Find a ‚counterfactual‘ on
(the other side of) the
decision boundary

Subject to

Low recourse costs

Solution I: Algorithmic Recourse under the IMF Assumption (Wachter et al (2018)) – Objective

Goal

Find a 'counterfactual' on
(the other side of) the
decision boundary

Subject to

Low recourse costs

$$\min_{\delta_x} \underbrace{(s - f(x + \delta_x))^2}_{\text{Distance from the prediction at the recourse to the target score } s} + \lambda \cdot \underbrace{d(x, x + \delta_x)}_{\text{Distance / cost from the factual } x \text{ to the recourse: } x + \delta_x}$$

Distance from the prediction at the
recourse to the target score s

Distance / cost from the factual x
to the recourse: $x + \delta_x$

Solution I: Algorithmic Recourse under the IMF Assumption (Wachter et al (2018)) – Objective

Goal

Find a ‚counterfactual‘ on (the other side of) the decision boundary

Subject to

Low recourse costs

$$\min_{\delta_x} \underbrace{(s - f(x + \delta_x))^2}_{\text{Distance from the prediction at the recourse to the target score } s} + \lambda \cdot \underbrace{d(x, x + \delta_x)}_{\text{Distance / cost from the factual } x \text{ to the recourse: } x + \delta_x}$$

Distance from the prediction at the recourse to the target score s

Distance / cost from the factual x to the recourse: $x + \delta_x$

Typical Distance / Cost functions

- ℓ_1, ℓ_2 norms (see Wachter et al (2018), and follow up works)
- Percentile shifts (see Ustun et al (2019))

Solution I: Algorithmic Recourse under the IMF Assumption (Wachter et al (2018)) – Examples

Cost functions

$$d(x, x + \delta_x) = \sum_j \frac{\delta_j^2}{std(X_j)}$$

Examples

	Original data			Counterfactual continuous			Counterfactual hybrid		
score	GPA	LSAT	Race	GPA	LSAT	Race	GPA	LSAT	Race
0.17	3.1	39.0	0	3.0	37.0	0.2	3.0	34.0	0
0.54	3.7	48.0	0	3.5	39.5	0.4	3.5	33.1	0
-0.77	3.3	28.0	1	3.5	39.8	0.4	3.4	33.4	0
-0.83	2.4	28.5	1	2.7	37.4	0.2	2.6	35.7	0
-0.57	2.7	18.3	0	2.8	28.1	-0.4	2.9	34.1	0

Wachter et al (2018)

$$d(x, x + \delta_x) = \sum_j \frac{|\delta_j|}{median(X_j)}$$

	Original data			Counterfactual continuous			Counterfactual hybrid		
score	GPA	LSAT	Race	GPA	LSAT	Race	GPA	LSAT	Race
0.17	3.1	39.0	0	3.1	35.0	0.1	3.1	34.0	0
0.54	3.7	48.0	0	3.7	33.5	0.0	3.7	32.4	0
-0.77	3.3	28.0	1	3.3	34.4	0.1	3.3	33.5	0
-0.83	2.4	28.5	1	2.4	39.3	0.2	2.4	35.8	0
-0.57	2.7	18.3	0	2.7	35.8	0.1	2.7	34.9	0

Wachter et al (2018)

Solution 2: Algorithmic Recourse that is likely to occur (Pawelczyk et al (2020)) – Objective

Goal

Find a ‚counterfactual‘ on (the other side of) the decision boundary

Subject to

1. Low recourse costs
2. The recourse being „likely to occur“

Solution 2: Algorithmic Recourse that is likely to occur (Pawelczyk et al (2020)) – Objective

Goal

Find a ‚counterfactual‘ on (the other side of) the decision boundary

Subject to

1. Low recourse costs
2. The recourse being „likely to occur“

- Introduce a **data model**: $x = g(z)$, i.e., **generative model**: $g: R^k \rightarrow R^d$
- If g is expressive enough, it will be able to faithfully model the data distribution (usually done with a VAE, GAN, diffusion model etc.)

Solution 2: Algorithmic Recourse that is likely to occur (Pawelczyk et al (2020)) – Objective

Goal

Find a ‚counterfactual‘ on (the other side of) the decision boundary

Subject to

1. Low recourse costs
2. The recourse being „likely to occur“

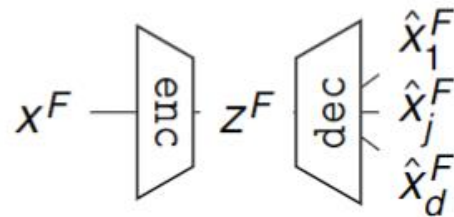
- Introduce a **data model**: $x = g(z)$, i.e., **generative model**: $g: R^k \rightarrow R^d$
- If g is expressive enough, it will be able to faithfully model the data distribution (usually done with a VAE, GAN, diffusion model etc.)

$$\min_{\delta_z} \underbrace{\left(s - f(g(z + \delta_z)) \right)^2}_{\text{Distance from the prediction at the recourse to the target score } s} + \lambda \cdot \underbrace{d(x, g(z + \delta_z))}_{\text{Distance / cost from the factual } x \text{ to the recourse: } g(z + \delta_z)}$$

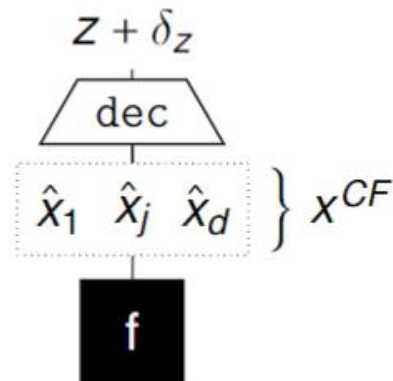
Distance from the prediction at the recourse to the target score s

Distance / cost from the factual x to the recourse: $g(z + \delta_z)$

Solution 2: Algorithmic Recourse that is likely to occur (Pawelczyk et al (2020)) – Generative Model

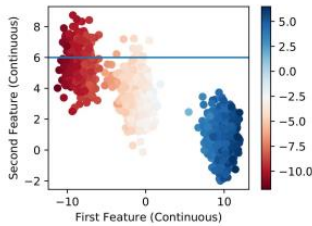


(a) End-to-End VAE training with **data-type specific loss(es)**



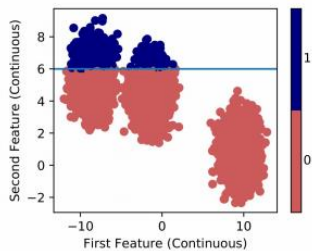
(b) At Recourse Intervention time, after encoding x^F into z^F , $g(\cdot)$ is a deterministic function and can be used to sample from $p(x)$

An illustrative experimental comparison



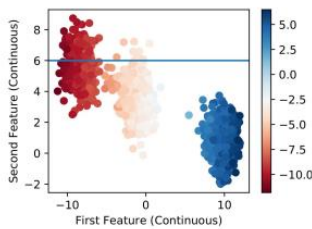
Pawelczyk et al (2020)

(a) Reconstructed train data generated by different \hat{z} (coloured).

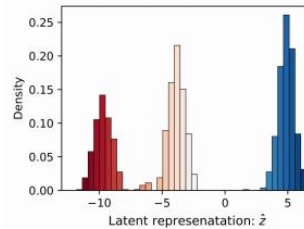


(d) True DGP.

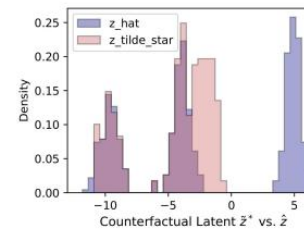
An illustrative experimental comparison



(a) Reconstructed train data generated by different \hat{z} (coloured).

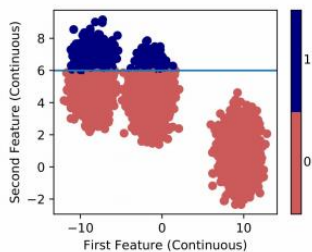


(b) Density of \hat{z} . Colours aligned so that left red \hat{z} generates left \hat{x} in 2(a).



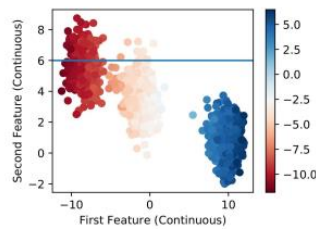
(c) Density of \hat{z} from 2(b) in blue. Density of \tilde{z}^* (red) belongs to $E(x)$ from 2(f) .

Pawelczyk et al (2020)

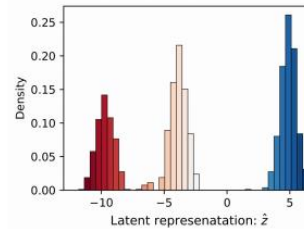


(d) True DGP.

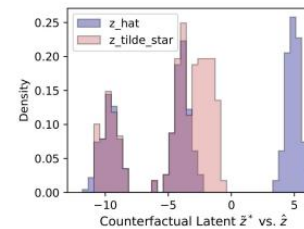
An illustrative experimental comparison



(a) Reconstructed train data generated by different \hat{z} (coloured).

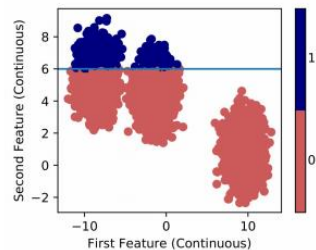


(b) Density of \hat{z} . Colours aligned so that left red \hat{z} generates left \hat{x} in 2(a).



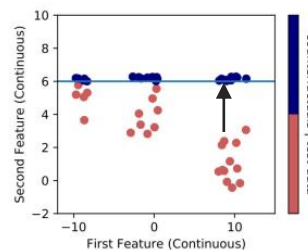
(c) Density of \hat{z} from 2(b) in blue. Density of \tilde{z}^* (red) belongs to $E(x)$ from 2(f) .

Pawelczyk et al (2020)



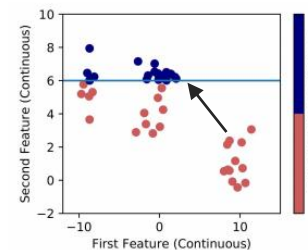
(d) True DGP.

Standard approaches



(e) Test data and $E(x)$ by GS/AR (not shown). Upper right $E(x)$ lie where no data is expected.

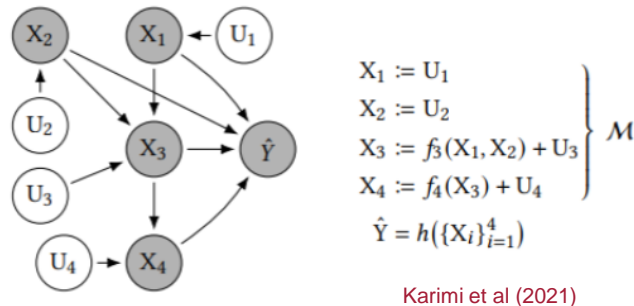
Our's



(f) Test data and $E(x)$ by our cchvae. Most $E(x)$ lie in high-density areas and are connected.

Solution 3: Algorithmic Recourse under Causal Assumptions (Karimi et al (2021))

Assume a structural causal model



- Dark nodes (observed)
- White nodes (latent)

Abduction – Step 1:

$$\begin{array}{ll} U_1 := X_1 & U_2 := X_2 \\ U_3 := X_3 - f_3(X_1, X_2) & U_4 := X_4 - f_4(X_3) \end{array}$$

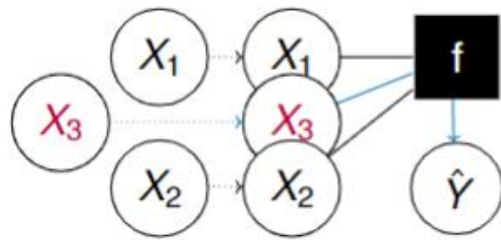
Action – Step 2: Hypothetical interventions:
 $do(\{X_j := a_j\})$ where $a_j = x_j + \delta_j$

$$\begin{array}{l} X_1 := [1 \in I] \cdot a_1 + [1 \notin I] \cdot U_1 \\ X_2 := [2 \in I] \cdot a_2 + [2 \notin I] \cdot U_2 \\ X_3 := [3 \in I] \cdot a_3 + [3 \notin I] \cdot (f(X_1, X_2) + U_3) \\ X_4 := [4 \in I] \cdot a_4 + [4 \notin I] \cdot (f(X_3) + U_4) \end{array}$$

Prediction – Step 3: Counterfactuals

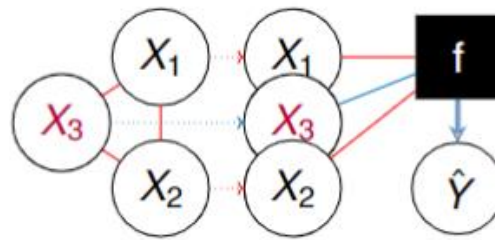
$$\begin{array}{l} x_1^{SCM} := [1 \in I] \cdot a_1 + [1 \notin I] \cdot u_1 \\ x_2^{SCM} := [2 \in I] \cdot a_2 + [2 \notin I] \cdot u_2 \\ x_3^{SCM} := [2 \in I] \cdot a_3 + [2 \notin I] \cdot f(x_1^{SCM}, x_2^{SCM}) + u_3 \\ x_4^{SCM} := [2 \in I] \cdot a_4 + [2 \notin I] \cdot (f(x_3^{SCM}) + u_4) \end{array}$$

Summary of Underlying Model Assumptions



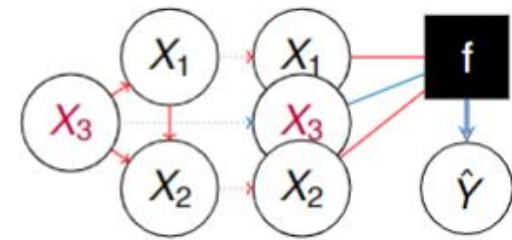
(a) Independent World

Wachter et al (2018)



(b) Dependent World

Pawelczyk et al (2020)



(c) Causal World

Karimi et al (2021)

Summary – PART 1

1. Desiderata

2. Potential
challenges

3. Recourse
objectives

Summary – PART 1

1. Desiderata

2. Potential
challenges

3. Recourse
objectives

Next?

Summary – PART 1

1. Desiderata

2. Potential
challenges

3. Recourse
objectives

Next?

Is algorithmic recourse compatible with other GDPR principles such as data minimization?

And

How can we generate recourses when we anticipate that end-users will likely implement prescribed recourses imprecisely?



Part 2a – Generating robust recourses



LET USERS DECIDE: NAVIGATING THE TRADE-OFFS BETWEEN
COSTS AND ROBUSTNESS IN ALGORITHMIC RECOURSE

Martin Pawelczyk
University of Tübingen
first.last@uni-tuebingen.de

Teresa Datta
Harvard University
tdatta@g.harvard.edu

Johannes van-den-Heuvel
University of Tübingen
first.last@uni-tuebingen.de

Gjergji Kasneci
University of Tübingen
first.last@uni-tuebingen.de

Himabindu Lakkaraju
Harvard University
hlakkaraju@hbs.edu

ABSTRACT

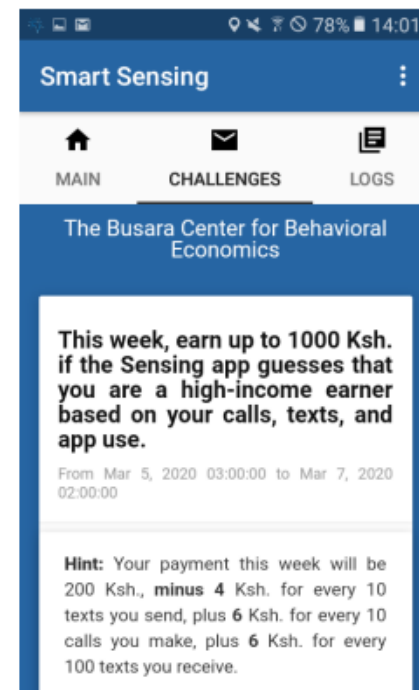
As machine learning (ML) models are increasingly being deployed in high-stakes applications, there has been growing interest in providing recourse to individuals adversely impacted by model predictions (e.g., an applicant whose loan has been denied). To this end, several post hoc techniques have been proposed in recent literature. These techniques generate recourses under the assumption that the affected individuals will implement the prescribed recourses *exactly*. However, recent studies suggest that individuals often implement recourses in a noisy and inconsistent manner – e.g., raising their salary by \$505 if the prescribed recourse suggested an increase of \$500. Motivated by this, we introduce and study the problem of recourse invalidation in the face of noisy human responses. More specifically, we theoretically and empirically analyze the behavior of state-of-the-art algorithms, and demonstrate that the recourses generated by these algorithms are very likely to be invalidated if small changes are made to them. We further propose a novel framework, EXPECTing noisy responses (EXPECT), which addresses the aforementioned problem by explicitly minimizing the probability of recourse invalidation in the face of noisy responses. Experimental evaluation with multiple real world datasets demonstrates the efficacy of the proposed framework, and supports our theoretical findings.

Motivation 1 – How often do recourses become invalidated?

Björkegreen et al (2020) developed an app that mimicked “digital credit” & deployed in Kenya

Figure 2: Smart Sensing App

(b) Challenge with Hint



Björkegreen et al (2020)

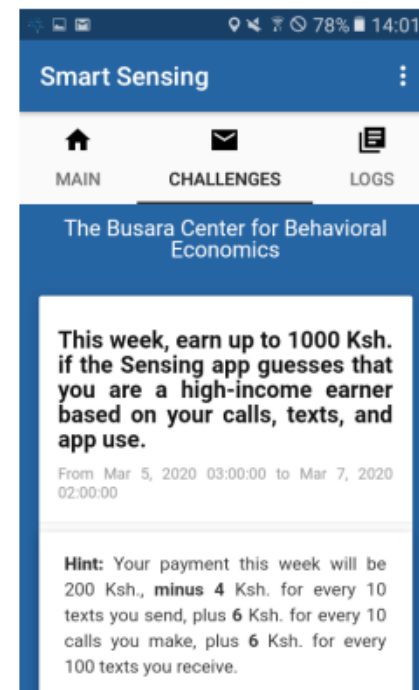
Motivation 1 – How often do recourses become invalidated?

Björkegreen et al (2020) developed an app that mimicked “digital credit” & deployed in Kenya

After participants were given recourse, they tended to change their features in the right direction

Figure 2: Smart Sensing App

(b) Challenge with Hint



Björkegreen et al (2020)

Motivation 2 – How often do recourses become invalidated?

To be classified high-income earner

- Do more calls
- Text less
- Have incoming texts

Users fail to appear 'high income' when model is opaque black-box

Users walk into the right direction when given recourse, but with high variance!

	# Calls (outgoing)	# Texts (outgoing)	# Texts (incoming)	# Calls w Non-Contacts (incoming + outgoing)	Mean Call Duration (evening, seconds)
Weekly Challenge: Use your phone like a high-income earner!					
Panel I: Incentives Generated by Algorithm (€/action)					
β^{LASSO}	0.625	-0.395	0.065	0	0
Panel II: x_{it}					
Assigned to challenge, algorithm opaque	-6.5573 (9.949)	14.3701 (16.405)	12.0135 (20.583)	1.1672 (3.473)	-6.8104 (7.002)
Assigned to challenge, algorithm transparent	11.8231 (9.083)	-15.69 (14.976)	-11.907 (18.79)	0.6706 (3.17)	-4.5744 (6.392)
N (Person-weeks)	1664	1664	1664	1664	1664

Notes: The first panel reports the decision rule associated with the challenge. The second reports the results of a regression of behavior on challenge assignment. Regressions estimated based on dummy indicators for complex challenge assignment for participants assigned “income” challenge, over person-weeks when the income challenge was assigned or when no challenge was assigned (“control” weeks). Simple challenge assignment person-weeks, used in estimating costs, are not included. Standard errors in parentheses.

Björkegreen et al (2020)



Motivation 3 – Invalidation rates for SOTA methods

Study robustness to perturbations of prescribed recourses

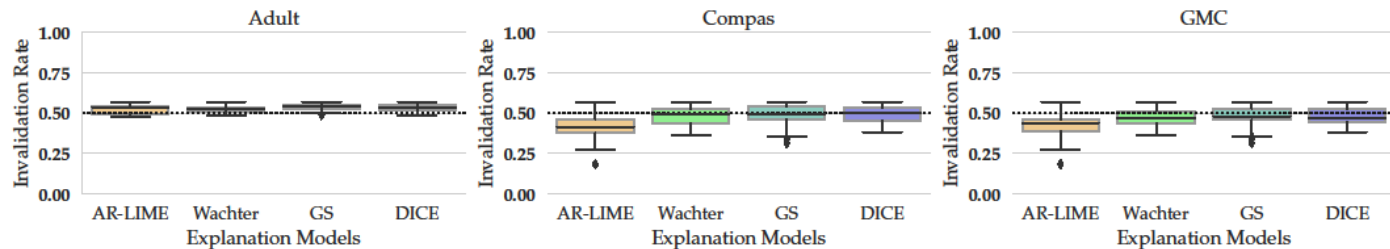
Perturb recourses by adding Gaussian RV with mean 0 and variance 0.01

Motivation 3 – Invalidation rates for SOTA methods

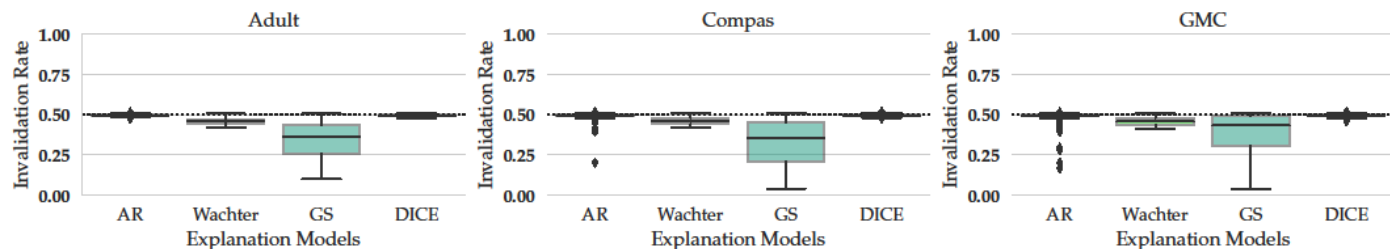
Study robustness to perturbations of prescribed recourses

Perturb recourses by adding Gaussian RV with mean 0 and variance 0.01

Logistic
regression
model



2-layer neural
network model

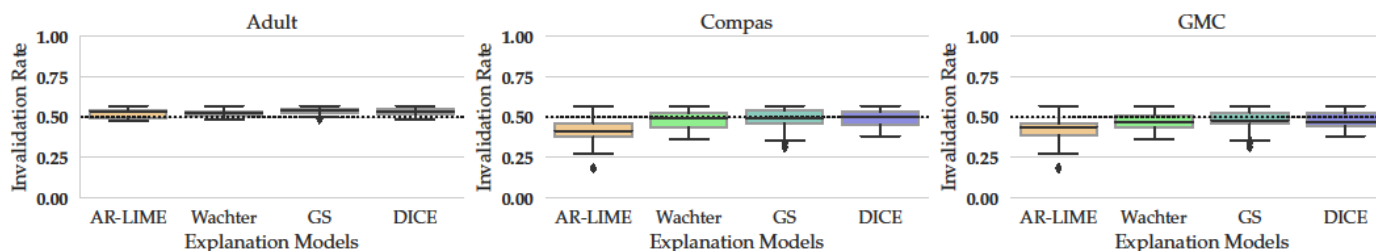


Motivation 3 – Invalidation rates for SOTA methods

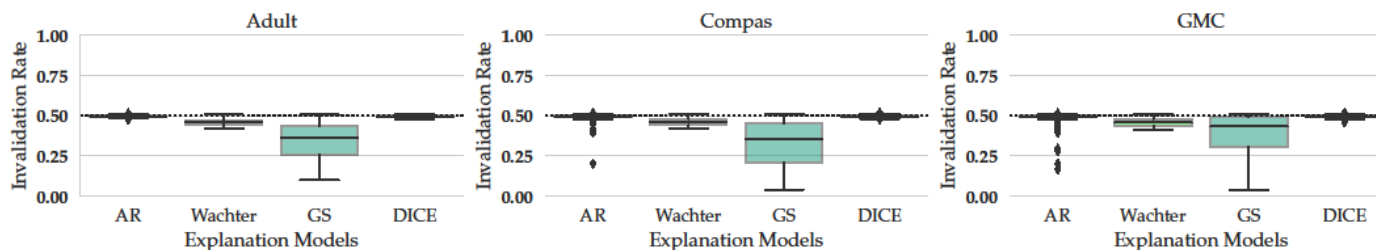
Study robustness to perturbations of prescribed recourses

Perturb recourses by adding Gaussian RV with mean 0 and variance 0.01

Logistic
regression
model



2-layer neural
network model



Median recourse invalidation rates are ~50%. Thus, if the recourse responses were noisy, then recourse success would often be **equivalent to a random coin flip**.



Question 1 : Can we formally study the IR for ML model / method pairs?

Question 2 : Can we devise an algorithm to generate more robust recourses in the presence of noisy human responses?

Defining the Recourse Invalidation Rate (IR)

Goal

Formally study the recourse invalidation rate

Idea

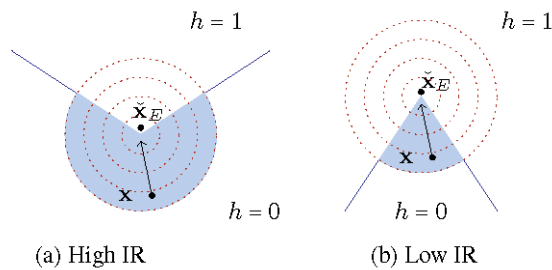
Perturb prescribed recourse \tilde{x}_E by (Gaussian) RV & observe how often recourse is invalidated

Invalidation Rate (IR)

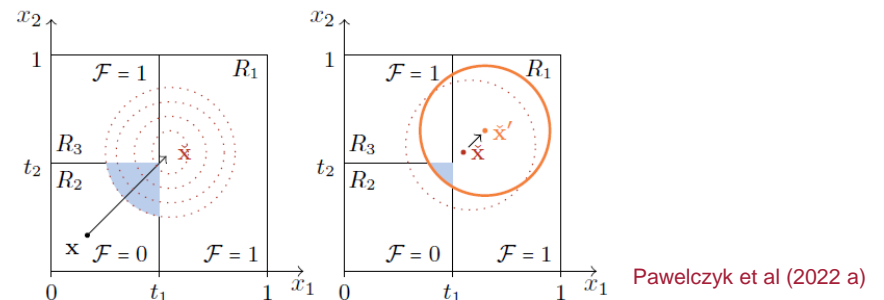
For a given classifier $h \in \{0,1\}$ the recourse invalidation rate for the counterfactual \tilde{x}_E is given by :

$$\Delta = E_{\epsilon \sim N(0, \sigma^2)} [h(\tilde{x}_E) - h(\tilde{x}_E + \epsilon)]$$

IR on neural network models



IR on decision trees



Robust Recourse in the Face of Noisy Human Responses (Pawelczyk et al (2022))

Control invalidation rate

Find 'counterfactual' close to the decision boundary

Low cost recourse

$$\min_{\delta_x} \underbrace{\max(0, \Delta(x + \delta_x) - r)}_{\text{Encourage invalidation rate to become close to } r} + \underbrace{(s - f(x + \delta_x))^2}_{\text{Distance from prediction at the recourse to the target score } s} + \underbrace{\lambda \cdot d(x, x + \delta_x)}_{\text{Distance / cost from the factual } x \text{ to the recourse: } x + \delta_x}$$

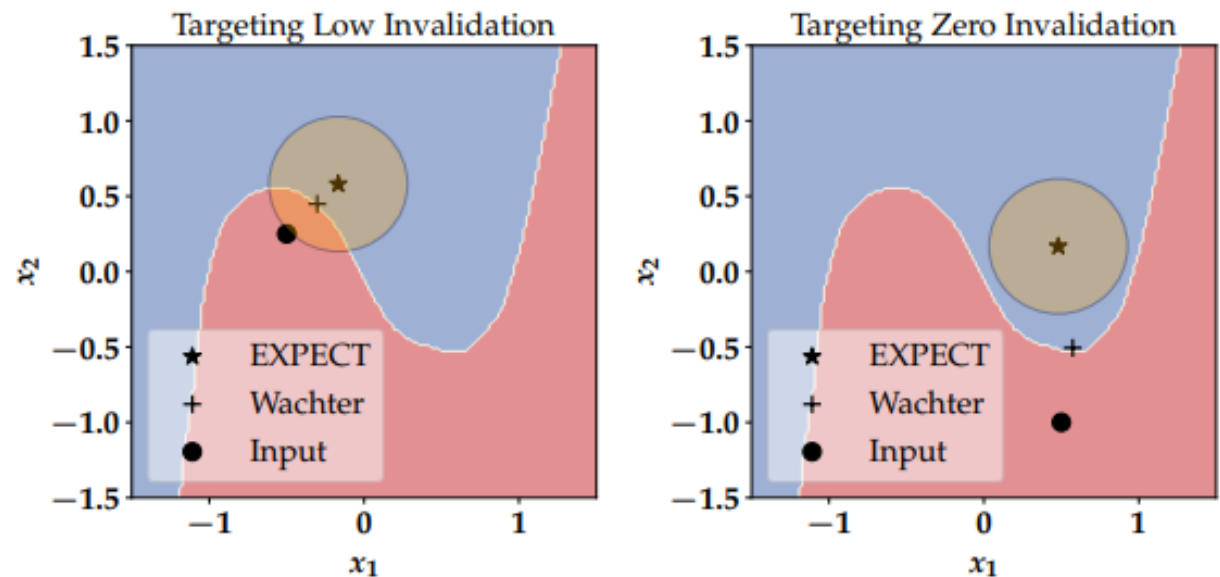
On the invlitation target

- The invalidation target r can be chosen by the end user
- It trades-off costs for increased robustness
- If the user is risk-averse, should choose low r

A Simple Example of our EXPECT Framework

σ^2 controls size of the orange ball

r controls how far the recourse is pushed towards the blue area



Pawelczyk et al (2022 a)

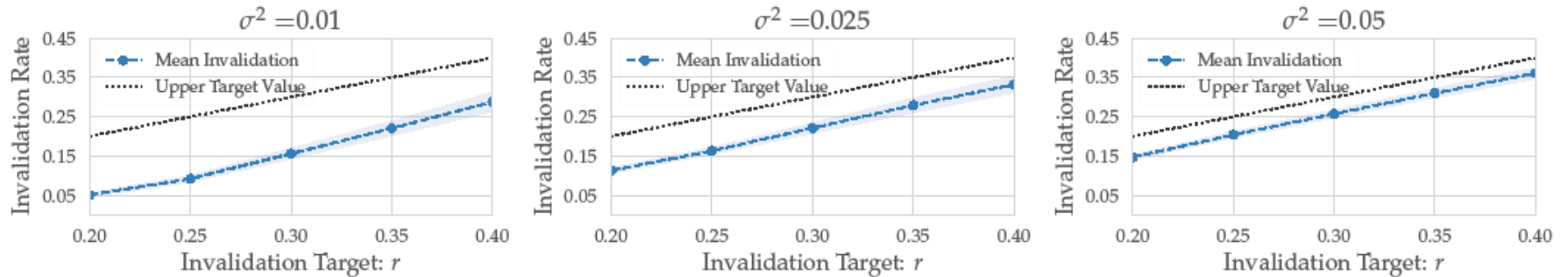
Central requirements for our new method

(1) We can control the invalidation rate via r and σ^2 !

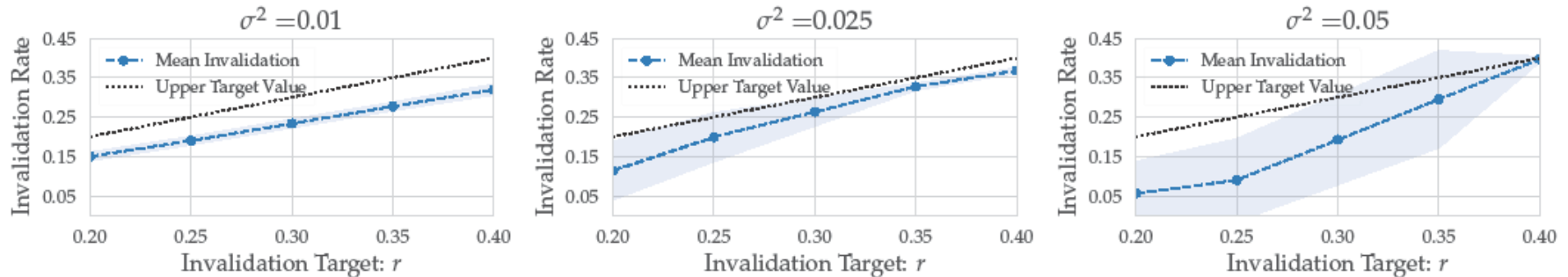
(2) Ideally, costs are low at controlled invalidation rates.

Req. 1: Controlling the recourse invalidation rate (Adult)

Logistic regression model



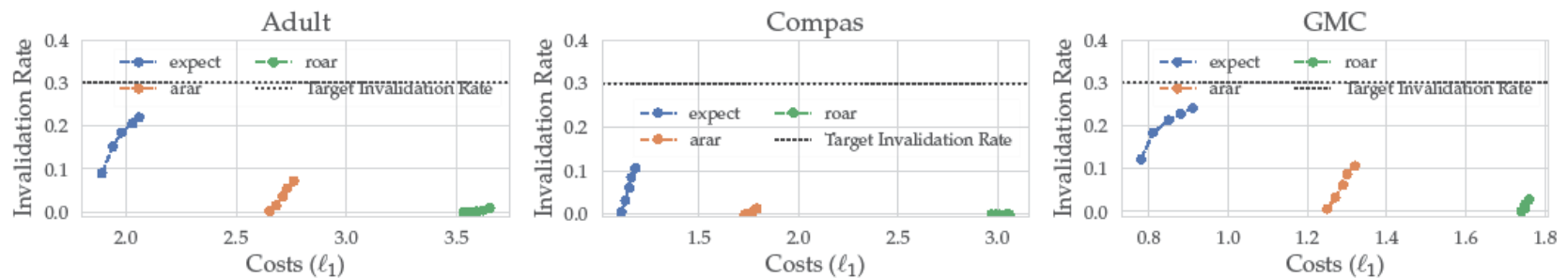
2-layer neural network model



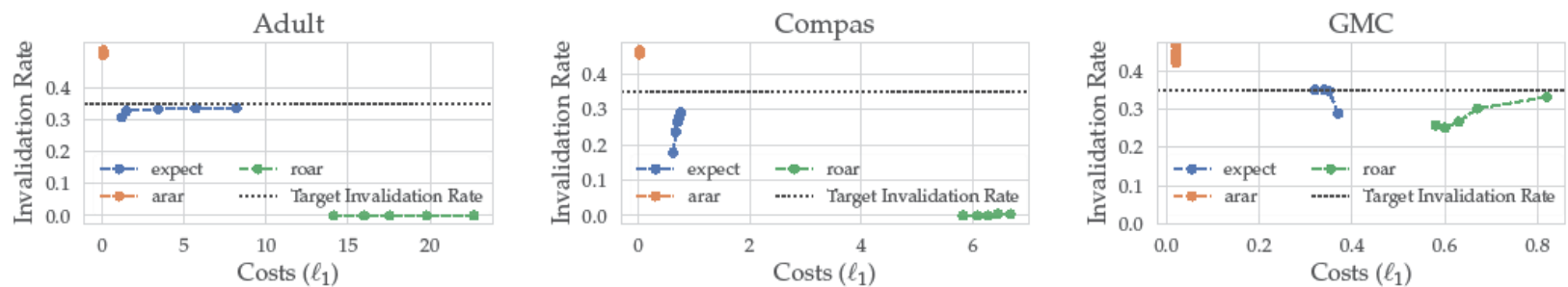
Pawelczyk et al (2022 a)

Req. 2: Low costs at controlled invalidation rates

Logistic regression model



2-layer neural network model



Pawelczyk et al (2022 a)



Summary – PART 2a

1. Noisy responses
are a relevant
problem in practice

2. Devised a
method to generate
robust recourses

3. Experiments
show desirable
properties

Summary – PART 2a

1. Noisy responses
are a relevant
problem in practice

2. Devised a
method to generate
robust recourses

3. Experiments
show desirable
properties

Next?

Is algorithmic recourse compatible with other GDPR principles such as data minimization?



Part 2b – Algorithmic recourse in the face of competing GDPR principles



ON THE TRADE-OFF BETWEEN ACTIONABLE EXPLANATIONS AND THE RIGHT TO BE FORGOTTEN

Martin Pawelczyk
University of Tübingen
first.last@uni-tuebingen.de

Tobias Leemann
University of Tübingen
first.last@uni-tuebingen.de

Asia Biega*
Max-Planck Institute for Security and Privacy
first.last@acm.org

Gjergji Kasneci*
University of Tübingen
first.last@uni-tuebingen.de

ABSTRACT

As machine learning (ML) models are increasingly being deployed in high-stakes applications, policy-makers have suggested tighter data protection regulations (e.g., GDPR, CCPA). One key principle is the “right to be forgotten” which gives users the right to have their data deleted. Another key principle is the right to an actionable explanation, also known as algorithmic recourse, allowing users to reverse unfavorable decisions. To date it is unknown whether these two principles can be operationalized simultaneously. Therefore, we introduce and study the problem of recourse invalidation in the context of data deletion requests. More specifically, we theoretically and empirically analyze the behavior of popular state-of-the-art algorithms and demonstrate that the recourses generated by these algorithms are likely to be invalidated if a small number of data deletion requests (e.g., 1 or 2) warrant updates of the predictive model. For the setting of linear models and overparameterized neural networks – studied through the lens of neural tangent kernels (NTKs) – we suggest a framework to identify a minimal subset of critical training points, which when removed, would lead to maximize the fraction of invalidated recourses. Using our framework, we empirically establish that the removal of as little as 2 data instances from the training set can invalidate up to 95 percent of all recourses output by popular state-of-the-art algorithms. Thus, our work raises fundamental questions about the compatibility of “the right to an actionable explanation” in the context of the “right to be forgotten”.

Motivation: A Closer Look Into The GDPR

Observation

- Not only „a right to an actionable explanation“ featured
- But also „the right to be forgotten“

Question

Can we get these two desiderata at the same time?



Recourse Robustness & Data Deletion Requests (I)

Goal

Study robustness in the presence of deletion requests

Idea

- Assign every training point a *data weight* $\omega \in \{0,1\}$
- Study changes to recourse \tilde{x} (**action stability**) or $f(\tilde{x})$ (**outcome stability**) when ω changes

Recourse Robustness & Data Deletion Requests (I)

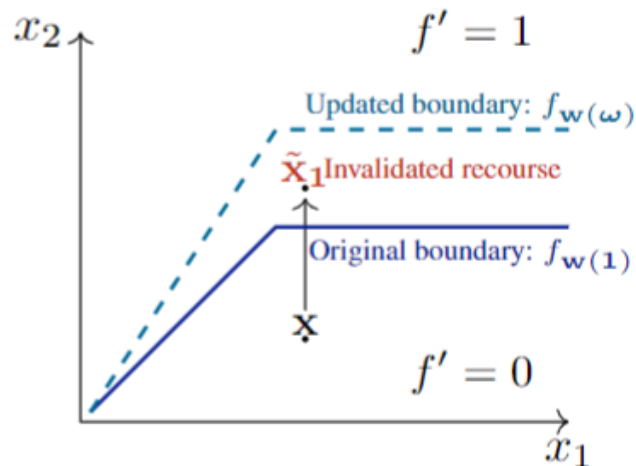
Goal

Study robustness in the presence of deletion requests

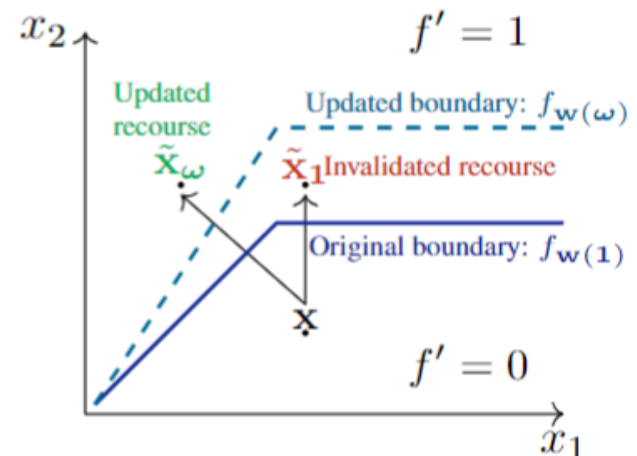
Idea

- Assign every training point a *data weight* $\omega \in \{0,1\}$
- Study changes to recourse \tilde{x} (**action stability**) or $f(\tilde{x})$ (**outcome stability**) when ω changes

Recourse outcome stability



Recourse action stability



Pawelczyk et al (2022 b)

Recourse Robustness & Data Deletion Requests (II)

Definition 1. (*Recourse outcome instability*) The recourse outcome instability with respect to a factual instance \mathbf{x} , where at least one data weight is set to 0, is defined as follows:

$$\Delta(\omega) = |f_{\mathbf{w}(1)}(\tilde{\mathbf{x}}_1) - f_{\mathbf{w}(\omega)}(\tilde{\mathbf{x}}_1)|, \quad (3)$$

where $f_{\mathbf{w}(1)}(\tilde{\mathbf{x}}_1)$ is the prediction at the prescribed recourse $\tilde{\mathbf{x}}_1$ based on the model that uses the full training set (i.e., $f_{\mathbf{w}(1)}$) and $f_{\mathbf{w}(\omega)}(\tilde{\mathbf{x}}_1)$ is the prediction at the prescribed recourse for an updated model and data deletion requests have been incorporated into the predictive model (i.e., $f_{\mathbf{w}(\omega)}$).

- Recourse held constant.
- Model parameters change due to data deletion.
- Will recourse become invalidated?

Definition 2. (*Recourse action instability*) The Recourse action instability with respect to a factual input \mathbf{x} , where at least one data weight is set to 0, is defined as follows:

$$\Phi_p(\omega) = \|\tilde{\mathbf{x}}_1 - \tilde{\mathbf{x}}_\omega\|_p, \quad (4)$$

where $p \in [1, \infty)$, and $\tilde{\mathbf{x}}_\omega$ is the recourse obtained for the model trained on the data instances that remain present in the data set after the deletion request.

- Model parameters change due to data deletion.
- To stay validate, recourse must also change.
- But by how much?

Identifying the smallest set of critical points that maximize recourse instability

Goal

Find the minimum set of critical point that has most impact on recourse instability

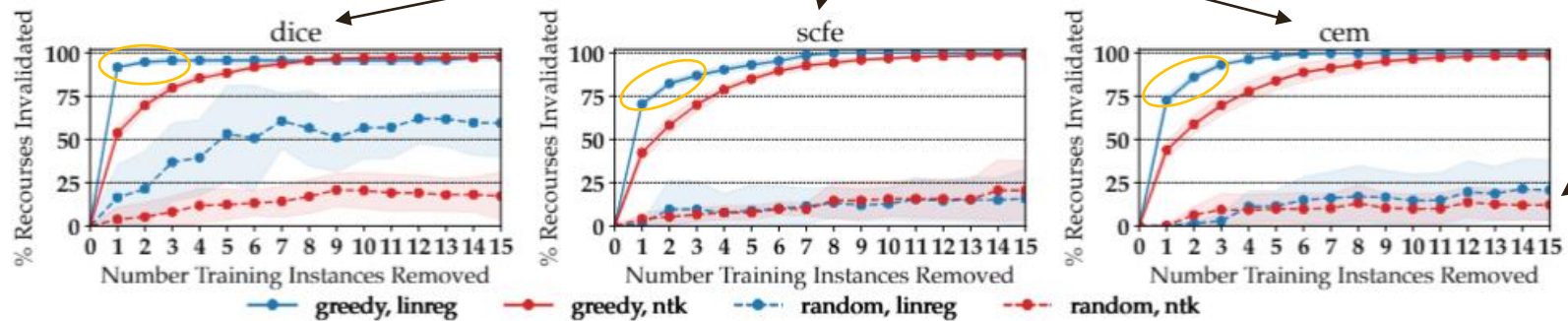
Objective

$\Gamma_\alpha = \{\omega: \text{Maximally floor}(\alpha n) \text{ entries of } \omega \text{ are } 0, \text{ and the remainder is } 1.\}$

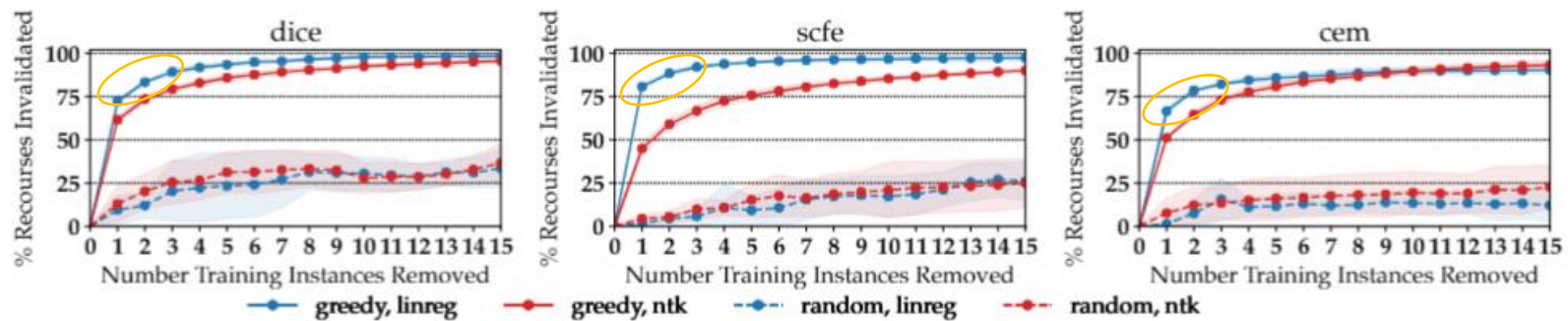
$$\omega^* = \operatorname{argmax}_{\omega \in \Gamma_\alpha} m(\omega) \quad m \in \{\Delta, \Phi_p\}$$

Experimental Results – Outcome Instability

Recourse methods



(a) Admission ($n = 17301$)

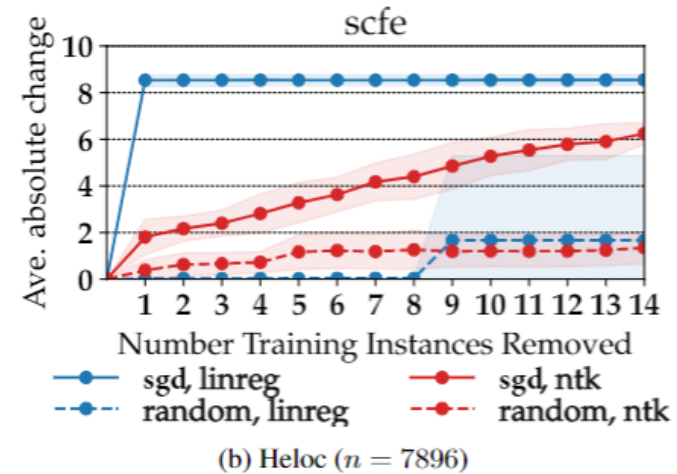
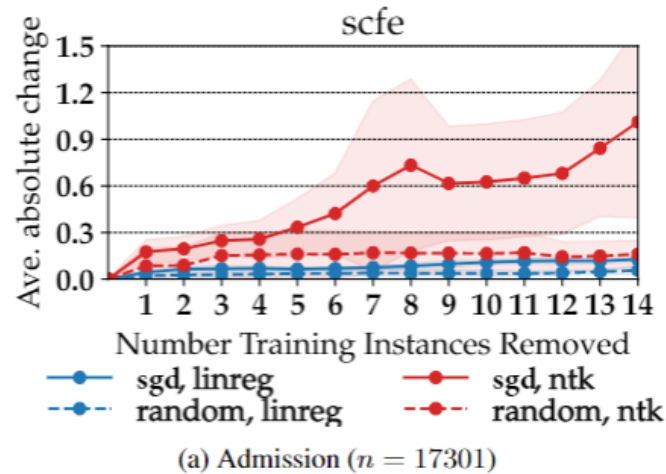


(b) Heloc ($n = 7896$)

Random removal strategy

Pawelczyk et al (2022 b)

Experimental Results – Action Instability



Pawelczyk et al (2022 b)

Experimental Results – Summary

Interesting results

Across multiple data sets & methods, recourses are fragile to a very small number (2-5) of carefully chosen data deletion requests!

Between the lines

The new robustness definitions allow you study counterfactuals from a theoretical viewpoint > check out the paper! 😊

Loads of interesting follow up questions

- Can we devise special training strategies to mitigate these issues?
- Do we need new robust recourse methods to defend against these threads?
- How to deal with invalidation in practice? ...

Summary – PART 2b

1. Two new robustness definitions of algorithmic recourse

2. Optimization procedure to identify most critical points

3. Experiments: two deletions can invalidate up to 95% of explanations

Summary – PART 2b

1. Two new robustness definitions of algorithmic recourse

2. Optimization procedure to identify most critical points

3. Experiments: two deletions can invalidate up to 95% of explanations

Next?

Hands-on tutorial on counterfactual explanations and recourse algorithms



Part III – Hands on session with



CARLA: A Python Library to Benchmark Algorithmic Recourse and Counterfactual Explanation Algorithms

Martin Pawelczyk*

University of Tübingen

`martin.pawelczyk@uni-tuebingen.de`

Sascha Bielawski

University of Tübingen

`sascha.bielawski@uni-tuebingen.de`

Johannes van den Heuvel

University of Tübingen

`johannes.van-den-heuvel@uni-tuebingen.de`

Tobias Richter †

CarePay International

`t.richter@carepay.com`

Gjergji Kasneci †

University of Tübingen

`gjergji.kasneci@uni-tuebingen.de`

Abstract

Counterfactual explanations provide means for prescriptive model explanations by suggesting actionable feature changes (e.g., increase income) that allow individuals to achieve favourable outcomes in the future (e.g., insurance approval). Choosing an appropriate method is a crucial aspect for meaningful counterfactual explanations. As documented in recent reviews, there exists a quickly growing literature with available methods. Yet, in the absence of widely available open-source implementations, the decision in favour of certain models is primarily based on what is readily available. Going forward – to guarantee meaningful comparisons across explanation methods – we present CARLA (Counterfactual And Recourse LibrAry), a python library for benchmarking counterfactual explanation methods across both different data sets and different machine learning models. In summary, our work provides the following contributions: (i) an extensive benchmark of 11 popular counterfactual explanation methods, (ii) a benchmarking framework for research on future counterfactual explanation methods, and (iii) a standardized set of integrated evaluation measures and data sets for transparent and extensive comparisons of these methods. We have open sourced CARLA and our experimental results on [Github](#), making them available as competitive baselines. We welcome contributions from other research groups and practitioners.

What is CARLA? And why is it useful?

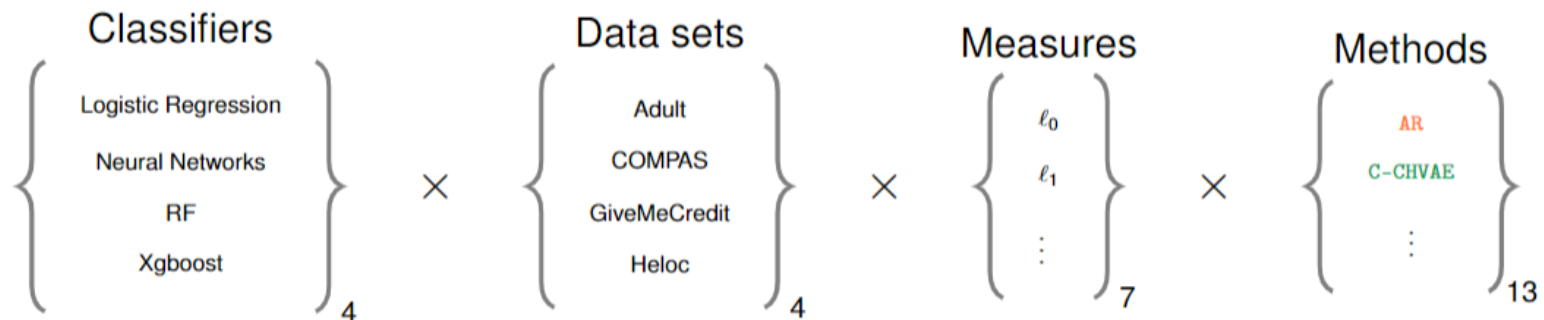
A Standardized Framework

Apply SOTA recourse methods

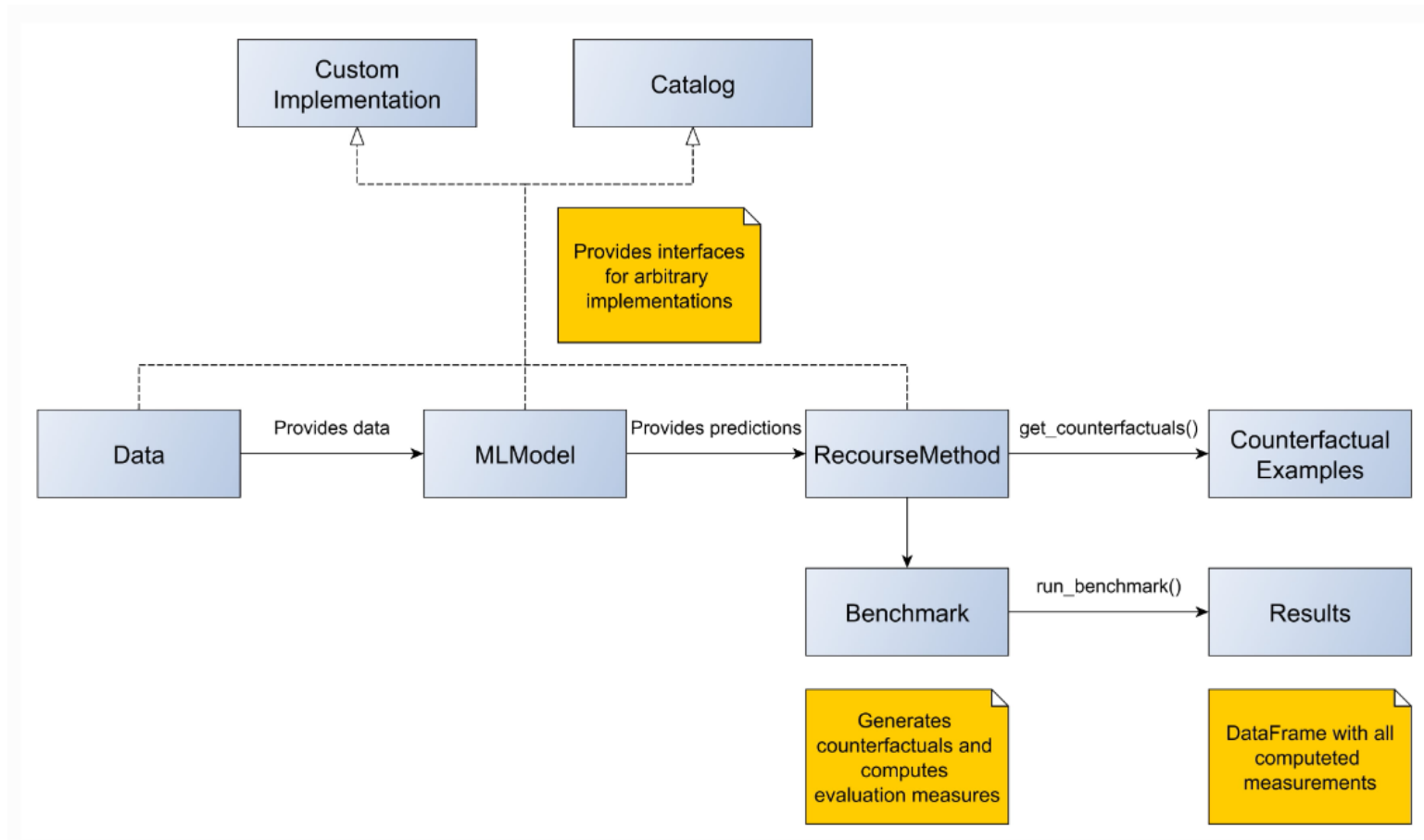
- 1) Data sets
- 2) Complex models
- 3) Evaluation measures

CARLA speeds up your research

- Easy integration of novel recourse methods
- Integrated evaluation measures
- Automated benchmarking



CARLA's structure



Pawelczyk et al (2021)

Are you looking for inspirations for the Hackathon?

Develop a new evaluation measure

- Existing evaluation measures do not evaluate semantic meaningfulness of recourses
- Benchmark the existing methods wrt to your new measure(s)
- Submit a PR

Status Quo & Next Steps

Status Quo

Apply SOTA recourse methods

- 1) Data sets
- 2) Complex models
- 3) Evaluation measures

Next Steps

- Easy integration of novel recourse methods
- Integrated evaluation measures
- Automated benchmarking

How to find CARLA?

- Documentation: <https://carla-counterfactual-and-recourse-library.readthedocs.io/en/latest/>
- Github: <https://github.com/carla-recourse/CARLA>
- Full paper: <https://arxiv.org/abs/2108.00783>

We welcome contributions 😊

The Hands-On Session

- 1) Make sure you have a google account
- 2) Open the following 'google colab' notebook:
https://colab.research.google.com/drive/1T_Au7dY-qv8ZK-VrLOy_oayENTUDDaAR?usp=sharing
- 3) Using CARLA's documentation website, the quickstart tutorials, and what's already there, work on the 9 tasks.

References

Solon Barocas, Andrew Selbst, Manish Raghavan, “**The Hidden Assumptions Behind Counterfactual Explanations and Principal Reasons**”, [FAccT 2020](#)

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci, “**Deep neural networks and tabular data: a survey**”, [arXiv: 2110:01889, 2021](#)

Daniël Björkegren, Joshua E. Blumenstock, and Samsun Knight, “**Manipulation-Proof Machine Learning**”, [arxiv:2004.03865, 2020](#)

Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci, “**Learning Model-Agnostic Counterfactual Explanations for Tabular Data**”, [The Web Conference \(WWW\), 2020](#).

Martin Pawelczyk, Sascha Bielawski, Johannes van-den-Heuvel, Tobias Richter* and Gjergji Kasneci*. “**CARLA: A library to benchmark counterfactual explanation and algorithmic recourse methods**”, [NeurIPS \(Benchmark track\), 2021](#).

Martin Pawelczyk, Teresa Datta, Johannes van-den-Heuvel, Gjergji Kasneci, and Hima Lakkaraju. “**Navigating the Tradeoff between Costs and Robustness in Algorithmic Recourse**”, [arXiv:2203.06768, 2022](#)

Martin Pawelczyk, Tobias Leemann, Asia Biega*, Gjergji Kasneci*, “**On the Tradeoff between Actionable Explanations and the Right to be forgotten**”, [arXiv:2208.14137, 2022](#)

Berk Ustun, Alexander Spangher, and “**Actionable Recourse**”, [FAccT, 2019](#)

Sandra Wachter, Bernt Mittelstadt, and Chris Russel, “**Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR**”, [Harvard Journal of Law & Technology, 2018](#)