Formal Results in Explainable Machine Learning

Tim van Erven

University of Amsterdam

Masterclass at the 3rd Annual Meeting
for the Dutch Inverse Problems Community

Groningen, October 12, 2023

Outline

Explainable Machine Learning

The Need for Explanations:

Why did the machine learning system

- Classify my company as high risk for money laundering?
- Reject my bank loan?
- Predict this patient can safely leave the intensive care?
- Mistake a picture of a husky for a wolf?
- Reject the profile picture I uploaded to get a public transport card?[1]
- ...

[1] Personal experience

Explainable Machine Learning

The Need for Explanations:

Why did the machine learning system

- Classify my company as high risk for money laundering?
- Reject my bank loan?
- Predict this patient can safely leave the intensive care?
- Mistake a picture of a husky for a wolf?
- Reject the profile picture I uploaded to get a public transport card?[1]
- ...

Information-Theoretic Constraints:

- Cannot communicate millions of parameters!
- Can communicate only some **relevant aspects** and/or need **high-level concepts** in common with user

[1] Personal experience

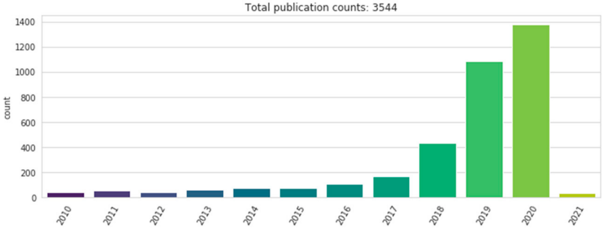
Booming Literature

(Tjoa and Guan, 2021)

Methods
CAM with global average pooling [42], [91]
+ Grad-CAM [43] generalizes CAM, utilizing gradient
+ Guided Grad-CAM and Feature Occlusion [68]
+ Respond CAM [44]
+ Multi-layer CAM [92]
LRP (Layer-wise Relevance Propagation) [13], [53]
+ Image classifications. PASCAL VOC 2009 etc [45]
+ Audio classification. AudioMNIST [47]
+ LRP on DeepLight. fMRI data from Human Connectome Project [48]
+ LRP on CNN and on BoW(bag of words)/SVM [49]
+ LRP on compressed domain action recognition algorithm [50]
+ LRP on video deep learning, *selective relevance method* [52]
+ BiLRP [51]
DeepLIFT [57]
Prediction Difference Analysis [58]
Slot Activation Vectors [41]
PRM (Peak Response Mapping) [59]
LIME (Local Interpretable Model-agnostic Explanations) [14]
+ MUSE with LIME [85]
+ Guidelinebased Additive eXplanation optimizes complexity, similar to LIME [93]
Also listed elsewhere: [56], [69], [71], [94]
Others. Also listed elsewhere: [95]
+ Direct output labels. Training NN via multiple instance learning [65]
+ Image corruption and testing Region of Interest statistically [66]
+ Attention map with autofocus convolutional layer [67]
DeconvNet [72]
Inverting representation with natural image prior [73]
Inversion using CNN [74]
Guided backpropagation [75], [91]
Activation maximization/optimization [38]
+ Activation maximization on DBN (Deep Belief Network) [76]
+ Activation maximization, multifaceted feature visualization [77]
Visualization via regularized optimization [78]
Semantic dictionary [39]
Network dissection [36], [37]
Decision trees
Propositional logic, rule-based [82]
Sparse decision list [83]
Decision sets, rule sets [84], [85]
Encoder-generator framework [86]
Filter Attribute Probability Density Function [87]
MUSE (Model Understanding through Subspace Explanations) [85]

(Tjoa and Guan, 2021)

Methods
Linear probe [101]
Regression based on CNN [106]
Backwards model for interpretability of linear models [107]
GDM (Generative Discriminative Models): ridge regression + least square [100]
GAM, GA^{2}M (Generative Additive Model) [82], [102], [103]
ProtoAttend [105]
Other content-subject-specific models:
+ Kinetic model for CBF (cerebral blood flow) [131]
+ CNN for PK (Pharmacokinetic) modelling [132]
+ CNN for brain midline shift detection [133]
+ Group-driven RL (reinforcement learning) on personalized healthcare [134]
+ Also see [108]–[112]
PCA (Principal Components Analysis), SVD (Singular Value Decomposition)
CCA (Canonical Correlation Analysis) [113]
SVCCA (Singular Vector Canonical Correlation Analysis) [97] = CCA+SVD
F-SVD (Frame Singular Value Decomposition) [114] on electromyography data
DWT (Discrete Wavelet Transform) + Neural Network [135]
MODWPT (Maximal Overlap Discrete Wavelet Package Transform) [136]
GAN-based Multi-stage PCA [118]
Estimating probability density with deep feature embedding [119]
t-SNE (t-Distributed Stochastic Neighbour Embedding) [77]
+ t-SNE on CNN [120]
+ t-SNE, activation atlas on GoogleNet [121]
+ t-SNE on latent space in meta-material design [122]
+ t-SNE on genetic data [137]
+ mm-t-SNE on phenotype grouping [138]
Laplacian Eigenmaps visualization for Deep Generative Model [124]
KNN (k-nearest neighbour) on multi-center low-rank rep. learning (MCLRR) [125]
KNN with triplet loss and *query-result activation map pair* [139]
Group-based Interpretable NN with RW-based Graph Convolutional Layer [123]
TCAV (Testing with Concept Activation Vectors) [96]
+ RCV (Regression Concept Vectors) uses TCAV with Br score [140]
+ Concept Vectors with UBS [141]
+ ACE (Automatic Concept-based Explanations) [56] uses TCAV
Influence function [129] helps understand adversarial training points
Representer theorem [130]
SocRat (Structured-output Causual Rationalizer) [127]
Meta-predictors [126]
Explanation vector [128]
Also listed elsewhere: [14], [43], [85], [94]
Also listed elsewhere: [14], [60], [85] etc
CNN with separable model [142]
Information theoretic: Information Bottleneck [98], [99]
Database of methods v.s. interpretability [10]
Case-Based Reasoning [143]
Integrated Gradients [69], [94]
Input invariance [71]
Application-based [144], [145]
Human-based [146], [147]
Function-based [2], [5], [42]–[44], [96], [97], [144], [145]

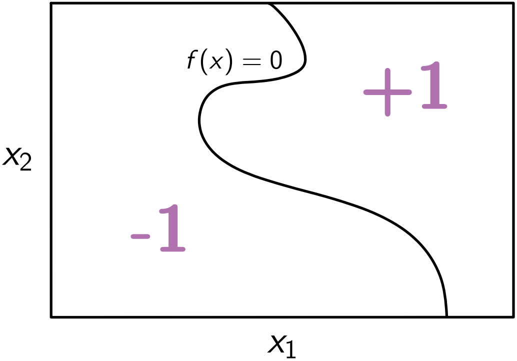


(Zhou et al., 2021)

(Karimi et al., 2021)

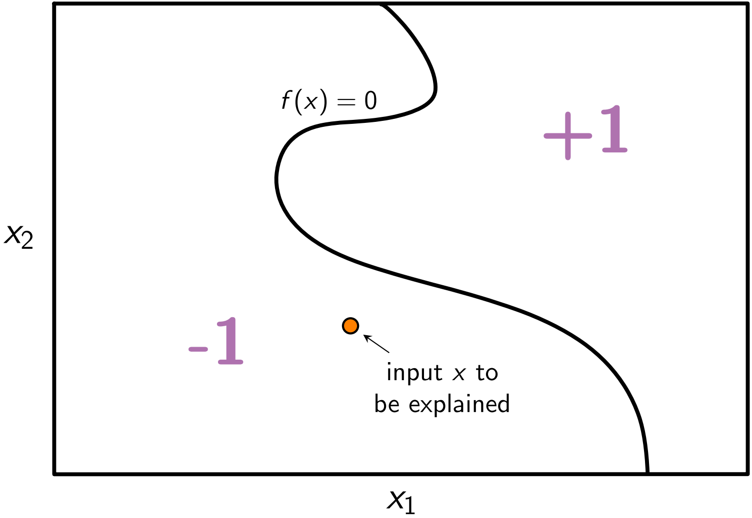
(2014.03)	SEDC [129]
(2015.08)	OAE [51]
(2016.05)	HCLS [110, 112]
(2017.06)	Feature Tweaking [186]
(2017.11)	CF Expl. [196]
(2017.12)	Growing Spheres [114]
(2018.02)	CEM [55]
(2018.02)	POLARIS [209]
(2018.05)	LORE [80]
(2018.06)	Local Foil Trees [190]
(2018.09)	Actionable Recourse [189]
(2018.11)	Weighted CFs [77]
(2019.01)	Efficient Search [175]
(2019.04)	CF Visual Expl. [76]
(2019.05)	MACE [99]
(2019.05)	DiCE [145]
(2019.05)	CERTIFAI [179]
(2019.06)	MACEM [56]
(2019.06)	Expl. using SHAP [165]
(2019.07)	Nearest Observable [201]
(2019.07)	Guided Prototypes [191]
(2019.07)	REVISE [95]
(2019.08)	CLEAR [202]
(2019.08)	MC-BRP [123]
(2019.09)	FACE [162]
(2019.09)	Equalizing Recourse [83]
(2019.10)	Action Sequences [163]
(2019.10)	C-CHVAE [156]
(2019.11)	FOCUS [124]
(2019.12)	Model-based CFs [127]
(2019.12)	LIME-C/SHAP-C [164]
(2019.12)	EMAP [41]
(2019.12)	PRINCE [71]
(2019.12)	LowProFool [18]
(2020.01)	ABELE [79]
(2020.01)	SHAP-based CFs [66]
(2020.02)	CEML [11–13]
(2020.02)	MINT [100]
(2020.03)	ViCE [74]
(2020.03)	Plausible CFs [22]
(2020.04)	SEDC-T [193]
(2020.04)	MOC [52]
(2020.04)	SCOUT [199]
(2020.04)	ASP-based CFs [28]
(2020.05)	CBR-based CFs [103]
(2020.06)	Survival Model CFs [106]
(2020.06)	Probabilistic Recourse [101]
(2020.06)	C-CHVAE [155]
(2020.07)	FRACE [210]
(2020.07)	DACE [96]
(2020.07)	CRUDS [60]
(2020.07)	Gradient Boosted CFs [5]
(2020.08)	Gradual Construction [97]
(2020.08)	DECE [44]
(2020.08)	Time Series CFs [16]
(2020.08)	PermuteAttack [87]
(2020.10)	Fair Causal Recourse [195]
(2020.10)	Recourse Summaries [167]
(2020.10)	Strategic Recourse [43]
(2020.11)	PARE [172]

Machine Learning: Binary Classification



- Goal: classify an input $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ as class -1 or class $+1$
- Usually by **thresholding a real-valued classifier** $f : \mathbb{R}^d \to \mathbb{R}$, e.g. predicted class is $\mathrm{sign}(f(x))$
- Classifier f obtained by minimizing error on **training data**

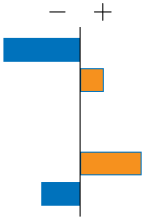
Local Post-hoc Explanations



- **Local:** only explain the part of f that is **(most) relevant for x**.
- **Post-hoc:** ignore explainability concerns when estimating f.

Local Explanations via Attributions

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{d-1} \\ x_d \end{bmatrix}$$

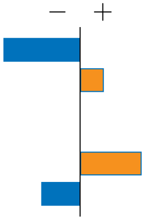


$$\begin{bmatrix} \varphi_f(x)_1 \\ \varphi_f(x)_2 \\ \vdots \\ \varphi_f(x)_{d-1} \\ \varphi_f(x)_d \end{bmatrix} = \varphi_f(x)$$

$\phi_f(x) \in \mathbb{R}^d$ attributes a **weight to each feature**, which explains **how important** the feature is **for the classification of x by f**.

Local Explanations via Attributions

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{d-1} \\ x_d \end{bmatrix}$$



$$\begin{bmatrix} \varphi_f(x)_1 \\ \varphi_f(x)_2 \\ \vdots \\ \varphi_f(x)_{d-1} \\ \varphi_f(x)_d \end{bmatrix} = \varphi_f(x)$$

$\phi_f(x) \in \mathbb{R}^d$ attributes a **weight to each feature**, which explains **how important** the feature is **for the classification of x by f**.

Example: low d, linear f

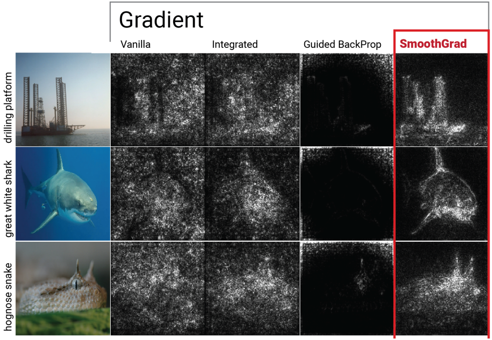
$$f(x) = \theta_0 + \sum_{i=1}^{d} \theta_i x_i$$

$$\phi_f(x)_i = \theta_i \qquad \text{could be } \textbf{coefficient} \text{ of } x_i$$

- NB This example is **too simple!** In general $\phi_f(x)$ will depend on x. But many methods can be viewed as local linearizations of f.

Example: Gradient-based Explanations

Various gradient methods[2]



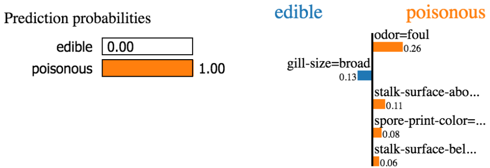
- Vanilla gradient: $\phi_f(x) = \nabla f(x)$
- SmoothGrad: $\phi_f(x) = \mathbb{E}_{Z \sim \mathcal{N}(x,\Sigma)}[\nabla f(Z)]$ (Smilkov et al., 2017)
- ...

[2]Image source: (Smilkov et al., 2017)

Example: LIME

LIME (Ribeiro, Singh, and Guestrin, 2016): Do local linear approximation of f near x (optionally in dimensionality reduced space), and report coefficients

LIME for tabular data:[3]



Feature	Value
odor=foul	True
gill-size=broad	True
stalk-surface-above-ring=silky	True
spore-print-color=chocolate	True
stalk-surface-below-ring=silky	True

(classifying edibility of mushrooms)

[3] Image source: `https://github.com/marcotcr/lime`

Example: LIME

LIME (Ribeiro, Singh, and Guestrin, 2016): Do local linear approximation of f near x (optionally in dimensionality reduced space), and report coefficients

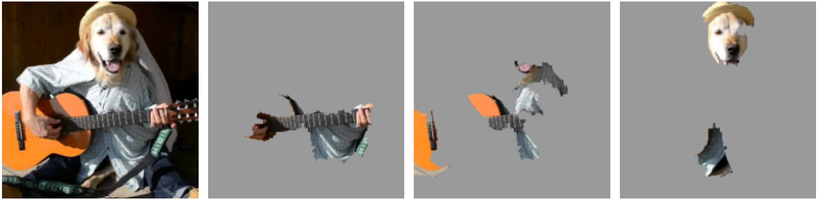
LIME for text:[3]

[3] Image source: `https://towardsdatascience.com/what-makes-your-question-insincere-in-quora-26ee7658b010`

Example: LIME

LIME (Ribeiro, Singh, and Guestrin, 2016): Do local linear approximation of f near x (optionally in dimensionality reduced space), and report coefficients

LIME for images:[3]



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

[3] Image by Ribeiro, Singh, and Guestrin (2016)

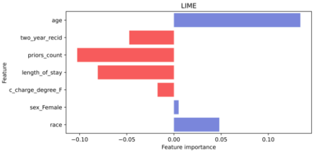
Exciting Times to Work on Explainability

Lots of open issues:

- Easily manipulated
- Explanation methods often disagree
- Plausible looking explanations may not represent model being explained (Adebayo et al., 2018)

Image by Dombrowski et al., 2019

LIME Method



SHAP Method

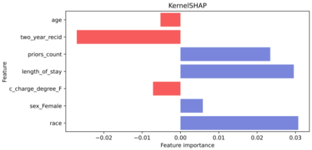


Image by Krishna et al., 2022

Outline

Local Smoothed Function Approximation

$$g^* = \underset{g \in \mathcal{G}}{\arg\min}\ \mathbb{E}_{\xi}[\ell(f, g, x, \xi)]$$

(Han, Srinivas, and Lakkaraju, 2022)

- f: function to be explained at input x
- g: explanation from class of **interpretable functions** $\mathcal{G}$
- ℓ: loss function
- Expectation **smooths** f by random perturbation ξ to x:

$$Z = x \oplus \xi \qquad (e.g.\ addition, multiplication, ...)$$

Local Smoothed Function Approximation

$$g^* = \arg\min_{g \in \mathcal{G}} \mathbb{E}_{\xi}[\ell(f, g, x, \xi)]$$

(Han, Srinivas, and Lakkaraju, 2022)

- f: function to be explained at input x
- g: explanation from class of **interpretable functions** $\mathcal{G}$
- ℓ: loss function
- Expectation **smooths** f by random perturbation ξ to x:

$$Z = x \oplus \xi \qquad (e.g.\ addition, multiplication, ...)$$

Remarks:

- Approximates smoothed version of f, where amount of smoothing depends on distribution of ξ
- Does not approximate the induced decision boundary $\{x : f(x) = 0\}$ (as often suggested)
- In practice: approximate expectation by finite nr. of samples of ξ

Example: C-LIME

$$g^* \;=\; \operatorname*{arg\,min}_{g\in\mathcal{G}} \; \mathbb{E}_{\xi}[\ell(f, g, x, \xi)]$$

Example: C-LIME

$$g^* \;=\; \operatorname*{arg\,min}_{g \in \mathcal{G}} \mathop{\mathbb{E}}_{Z} \left[\big(f(Z) - g(Z) \big)^2 \right]$$

- **Squared error:**

$$\ell(f, g, x, \xi) = \big(f(Z) - g(Z) \big)^2$$

for additive perturbations $Z = x + \xi$

Example: C-LIME

$$\theta^*, \theta_0^* \;=\; \operatorname*{arg\,min}_{\theta,\theta_0} \mathop{\mathbb{E}}_{Z}\left[\left(f(Z) - Z^\intercal\theta - \theta_0\right)^2\right]$$

- **Squared error:**

$$\ell(f, g, x, \xi) = \left(f(Z) - g(Z)\right)^2$$

  for additive perturbations $Z = x + \xi$

- **Linear approximations $\mathcal{G}$:**

$$g(x) = x^\intercal\theta + \theta_0 \qquad (\theta \in \mathbb{R}^d, \theta_0 \in \mathbb{R})$$

  NB: output only feature weights θ^*, not intercept θ_0^*.

Example: C-LIME

$$\theta^*, \theta_0^* \;=\; \arg\min_{\theta,\theta_0} \mathop{\mathbb{E}}_{Z}\left[\big(f(Z) - Z^\intercal\theta - \theta_0\big)^2\right]$$

- **Squared error:**

$$\ell(f, g, x, \xi) = \big(f(Z) - g(Z)\big)^2$$

  for additive perturbations $Z = x + \xi$

- **Linear approximations $\mathcal{G}$:**

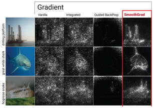
$$g(x) = x^\intercal\theta + \theta_0 \qquad (\theta \in \mathbb{R}^d, \theta_0 \in \mathbb{R})$$

  NB: output only feature weights θ^*, not intercept θ_0^*.

- **Normally distributed perturbations:**

$$\begin{aligned} \xi &\sim \mathcal{N}(0, \Sigma) \qquad \text{for hyperparameter } \Sigma \succ 0 \\ Z &\sim \mathcal{N}(x, \Sigma) \end{aligned}$$

Example: SmoothGrad

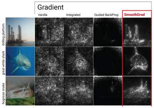


$$\phi_f(x) = \mathop{\mathbb{E}}_{Z\sim\mathcal{N}(x,\Sigma)}[\nabla f(Z)]$$

Theorem (Agarwal et al., 2021)

SmoothGrad and C-Lime are equivalent.

Example: SmoothGrad



$$\phi_f(x) = \mathop{\mathbb{E}}_{Z\sim\mathcal{N}(x,\Sigma)}[\nabla f(Z)]$$

Theorem (Agarwal et al., 2021)

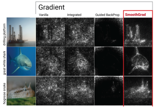
SmoothGrad and C-Lime are equivalent.

Proof sketch:

1. For Gaussian Z, Stein's lemma (proved by a variant of integration by parts) states:

$$\mathop{\mathbb{E}}_{Z\sim\mathcal{N}(x,\Sigma)}[\nabla f(Z)] = \Sigma^{-1}\,\mathbb{E}[f(Z)(Z-x)]$$

Example: SmoothGrad



$$\phi_f(x) = \mathop{\mathbb{E}}_{Z \sim \mathcal{N}(x,\Sigma)}[\nabla f(Z)]$$

Theorem (Agarwal et al., 2021)

SmoothGrad and C-Lime are equivalent.

Proof sketch:

1. For Gaussian Z, Stein's lemma (proved by a variant of integration by parts) states:

$$\mathop{\mathbb{E}}_{Z \sim \mathcal{N}(x,\Sigma)}[\nabla f(Z)] = \Sigma^{-1}\,\mathbb{E}[f(Z)(Z - x)]$$

2. The C-LIME objective is a least-squares problem:

$$\arg\min_{\theta,\theta_0} \mathbb{E}\left[\left(f(Z) - Z^\intercal\theta - \theta_0\right)^2\right]$$

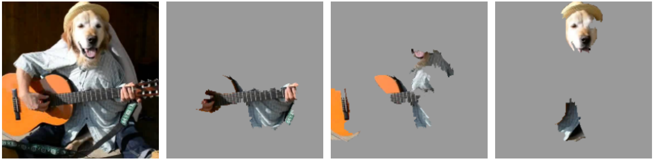
Minimizing first in θ_0 gives $\theta_0 = \mathbb{E}[f(Z)] - x^\intercal\theta$. Then setting the gradient w.r.t. θ to 0 leads to the same solution as SmoothGrad:

$$\theta = \Sigma^{-1}\,\mathbb{E}[f(Z)(Z - x)]$$

Sampling High-level Features

Motivation:

- Low-level features not interpretable (e.g. pixels)
- Want explanation in terms of high-level concepts (e.g. superpixels)

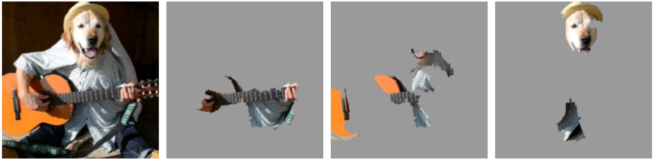


(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Sampling High-level Features

Motivation:

- Low-level features not interpretable (e.g. pixels)
- Want explanation in terms of high-level concepts (e.g. superpixels)



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

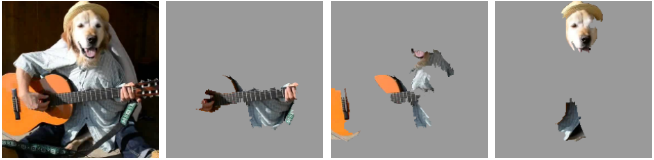
Approach:

- Binary parametrization $h_x : \{0, 1\}^m \to \mathcal{X}$ of variations of x:
  - $\tilde{x}_i = 1$: set i-th interpretable high-level concept from x to be present
  - $\tilde{x}_i = 0$: remove i-th interpretable high-level concept from x (e.g. replace superpixel by gray values)

Sampling High-level Features

Motivation:

- Low-level features not interpretable (e.g. pixels)
- Want explanation in terms of high-level concepts (e.g. superpixels)



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Approach:

- Binary parametrization $h_x : \{0,1\}^m \to \mathcal{X}$ of variations of x:
  - $\tilde{x}_i = 1$: set i-th interpretable high-level concept from x to be present
  - $\tilde{x}_i = 0$: remove i-th interpretable high-level concept from x (e.g. replace superpixel by gray values)
- Approximate the new function of high-level concepts

$$f_x(\tilde{x}) = f(h_x(\tilde{x})) \qquad \text{for } \tilde{x} \in \{0,1\}^m.$$

NB f_x and f have different domains, so an approximation of f_x is not an approximation of f

Example: LIME

$$g^* \;=\; \operatorname*{arg\,min}_{g \in \mathcal{G}} \; \mathbb{E}_{\xi}[\ell(f, g, x, \xi)]$$

Example: LIME

$$g^* \;=\; \arg\min_{g \in \mathcal{G}} \mathop{\mathbb{E}}_{\tilde{Z}} \left[\pi_x(\tilde{Z}) \big(f_x(\tilde{Z}) - g(\tilde{Z}) \big)^2 \right]$$

- **Approximate f_x:** Weighted squared error:

$$\ell(f_x, g, \xi) = \pi_x(\tilde{Z}) \big(f_x(\tilde{Z}) - g(\tilde{Z}) \big)^2$$

Example: LIME

$$g^* \;=\; \operatorname*{arg\,min}_{g \in \mathcal{G}} \mathop{\mathbb{E}}_{\tilde{Z}} \left[\pi_x(\tilde{Z}) \big(f_x(\tilde{Z}) - g(\tilde{Z}) \big)^2 \right]$$

- **Approximate f_x:** Weighted squared error:

$$\ell(f_x, g, \xi) = \pi_x(\tilde{Z}) \big(f_x(\tilde{Z}) - g(\tilde{Z}) \big)^2$$

Let $\bar{x} = h_x^{-1}(x)$ be the high-level representation of x. (Typically $\bar{x} = \mathbb{1}$.) Then $\xi \in \{0,1\}^m$ **masks high-level features**:

$$\tilde{Z}_i = \begin{cases} 1 & \text{if } \bar{x}_i = 1 \text{ and } \xi_i = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Example: LIME

$$\theta^*, \theta_0^* \; = \; \operatorname*{arg\,min}_{\theta, \theta_0} \mathop{\mathbb{E}}_{\tilde{Z}} \left[\pi_x(\tilde{Z}) \big(f_x(\tilde{Z}) - \tilde{Z}^\intercal \theta - \theta_0 \big)^2 \right]$$

- **Approximate f_x:** Weighted squared error:

$$\ell(f_x, g, \xi) = \pi_x(\tilde{Z}) \big(f_x(\tilde{Z}) - g(\tilde{Z}) \big)^2$$

Let $\bar{x} = h_x^{-1}(x)$ be the high-level representation of x. (Typically $\bar{x} = \mathbb{1}$.) Then $\xi \in \{0, 1\}^m$ **masks high-level features**:

$$\tilde{Z}_i = \begin{cases} 1 & \text{if } \bar{x}_i = 1 \text{ and } \xi_i = 1, \\ 0 & \text{otherwise.} \end{cases}$$

- **Linear approximations $\mathcal{G}$ in terms of high-level features**

Example: LIME

$$\theta^*, \theta_0^* \;=\; \arg\min_{\theta,\theta_0} \mathbb{E}_{\tilde{Z}}\Big[\pi_x(\tilde{Z})\big(f_x(\tilde{Z}) - \tilde{Z}^\intercal\theta - \theta_0\big)^2\Big]$$

- **Approximate f_x:** Weighted squared error:

$$\ell(f_x, g, \xi) = \pi_x(\tilde{Z})\big(f_x(\tilde{Z}) - g(\tilde{Z})\big)^2$$

  Let $\bar{x} = h_x^{-1}(x)$ be the high-level representation of x. (Typically $\bar{x} = \mathbb{1}$.) Then $\xi \in \{0,1\}^m$ **masks high-level features**:

$$\tilde{Z}_i = \begin{cases} 1 & \text{if } \bar{x}_i = 1 \text{ and } \xi_i = 1, \\ 0 & \text{otherwise.} \end{cases}$$

- **Linear approximations $\mathcal{G}$ in terms of high-level features**
- **Default weights** downscale distant instances[4]:

$$\pi_x(\tilde{Z}) = \exp\Big(-\frac{\mathrm{d}_{\cos}(\tilde{Z}, \bar{x})^2}{2\nu^2}\Big) \qquad \text{for hyperparameter } \nu > 0.$$

[4] $\mathrm{d}_{\cos}(u, v) = 1 - \frac{u^\intercal v}{\|u\|\|v\|}$ is the cosine distance between vectors

Example: LIME

$$\theta^*, \theta_0^* \;=\; \arg\min_{\theta,\theta_0} \mathop{\mathbb{E}}_{\tilde{Z}}\Big[\pi_x(\tilde{Z})\big(f_x(\tilde{Z}) - \tilde{Z}^\intercal\theta - \theta_0\big)^2\Big]$$

- **Approximate f_x:** Weighted squared error:

$$\ell(f_x, g, \xi) = \pi_x(\tilde{Z})\big(f_x(\tilde{Z}) - g(\tilde{Z})\big)^2$$

Let $\bar{x} = h_x^{-1}(x)$ be the high-level representation of x. (Typically $\bar{x} = \mathbb{1}$.) Then $\xi \in \{0,1\}^m$ **masks high-level features**:

$$\tilde{Z}_i = \begin{cases} 1 & \text{if } \bar{x}_i = 1 \text{ and } \xi_i = 1, \\ 0 & \text{otherwise.} \end{cases}$$

- **Linear approximations $\mathcal{G}$ in terms of high-level features**
- **Default weights** downscale distant instances[4]:

$$\pi_x(\tilde{Z}) = \exp\big(-\frac{\mathrm{d}_{\cos}(\tilde{Z}, \bar{x})^2}{2\nu^2}\big) \qquad \text{for hyperparameter } \nu > 0.$$

- **Default binary masks:** $\xi_i \sim \mathrm{Bernoulli}(1/2)$

[4] $\mathrm{d}_{\cos}(u, v) = 1 - \frac{u^\intercal v}{\|u\|\|v\|}$ is the cosine distance between vectors

Example: SHAP

Axiomatic Characterization of Linear Approximation

(Lundberg and Lee, 2017 translate game-theory result by Young, 1985)

1. **Local accuracy** at input x:

$$f_x(\bar{x}) = \bar{x}^\intercal \theta + \theta_0$$

2. No weight on features **missing** from $\bar{x}$:

$$\bar{x}_i = 0 \implies \theta_i = 0$$

3. **Symmetry:**[5] For any permutation $\pi : [m] \to [m]$

$$\theta(\pi f_x) = \pi\theta(f_x)$$

4. **Strong monotonicity:** For any two functions f_x, f'_x

$$\begin{aligned} &\text{If} \quad f'_x(\tilde{x}) - f'_x(\tilde{x} \setminus i) \geq f_x(\tilde{x}) - f_x(\tilde{x} \setminus i) \qquad \text{for all } \tilde{x} \in \{0,1\}^m, \\ &\qquad \text{then} \quad \theta_i(f'_x) \geq \theta_i(f_x). \end{aligned}$$

[5] Lundberg and Lee, 2017 have incorrect "proof" that symmetry is implied by the other conditions.

Example: SHAP

Axiomatic Characterization of Linear Approximation

(Lundberg and Lee, 2017 translate game-theory result by Young, 1985)

1. **Local accuracy** at input x: $f_x(\bar{x}) = \bar{x}^\intercal \theta + \theta_0$
2. No weight on features **missing** from $\bar{x}$: $\bar{x}_i = 0 \implies \theta_i = 0$
3. **Symmetry:** For any permutation $\pi : [m] \to [m]$: $\theta(\pi f_x) = \pi\theta(f_x)$
4. **Strong monotonicity:** For any two functions f_x, f'_x

$$\text{If} \quad f'_x(\tilde{x}) - f'_x(\tilde{x} \setminus i) \geq f_x(\tilde{x}) - f_x(\tilde{x} \setminus i) \qquad \text{for all } \tilde{x} \in \{0,1\}^m,$$
$$\text{then} \quad \theta_i(f'_x) \geq \theta_i(f_x).$$

Example: SHAP

Axiomatic Characterization of Linear Approximation

(Lundberg and Lee, 2017 translate game-theory result by Young, 1985)

1. **Local accuracy** at input x: $f_x(\bar{x}) = \bar{x}^\intercal \theta + \theta_0$
2. No weight on features **missing** from $\bar{x}$: $\bar{x}_i = 0 \implies \theta_i = 0$
3. **Symmetry:** For any permutation $\pi : [m] \to [m]$: $\theta(\pi f_x) = \pi\theta(f_x)$
4. **Strong monotonicity:** For any two functions f_x, f'_x

$$\text{If} \quad f'_x(\tilde{x}) - f'_x(\tilde{x} \setminus i) \geq f_x(\tilde{x}) - f_x(\tilde{x} \setminus i) \qquad \text{for all } \tilde{x} \in \{0,1\}^m,$$
$$\text{then} \quad \theta_i(f'_x) \geq \theta_i(f_x).$$

Theorem (Young, 1985; Lundberg and Lee, 2017)

The unique θ, θ_0 that satisfy all four axioms are $\theta_0 = f_x(\emptyset)$ and

$$\theta_i = \sum_{\tilde{x} : \tilde{x}_i \leq \bar{x}_i} \frac{|\tilde{x}|!(m - |\tilde{x}| - 1)!}{m!} \big[f_x(\tilde{x}) - f_x(\tilde{x} \setminus i) \big],$$

where $|\tilde{x}|$ is the number of ones in $\tilde{x}$, and $\tilde{x} \setminus i$ is $\tilde{x}$ with the i-th component set to 0.

Kernel SHAP

There is a surprising relation between SHAP and LIME:

Theorem (Lundberg and Lee (2017))

SHAP is equivalent to LIME with the weights set to

$$\pi_x(\tilde{Z}) = \frac{m-1}{\binom{m}{|\tilde{Z}|}|\tilde{Z}|(m-|\tilde{Z}|)}.$$

- NB $\pi_x(\emptyset) = \pi_x(\mathbb{1}) = \infty$. Interpret as hard constraints that $g(\emptyset) = f_x(\emptyset)$ and $g(\mathbb{1}) = f_x(\mathbb{1})$.

Kernel SHAP

There is a surprising relation between SHAP and LIME:

Theorem (Lundberg and Lee (2017))

SHAP is equivalent to LIME with the weights set to

$$\pi_x(\tilde{Z}) = \frac{m-1}{\binom{m}{|\tilde{Z}|}|\tilde{Z}|(m-|\tilde{Z}|)}.$$

- NB $\pi_x(\emptyset) = \pi_x(\mathbb{1}) = \infty$. Interpret as hard constraints that $g(\emptyset) = f_x(\emptyset)$ and $g(\mathbb{1}) = f_x(\mathbb{1})$.

Proof remarks:

- The proof by Lundberg and Lee (2017) is based on evaluating the LIME weighted least squares solution $\theta = (X^\intercal W X)^{-1} X^\intercal W y$
- They omit many non-trivial proof details
- I have checked all steps except their assumption that the weighted least squares solution with the infinite weights is the limit of the least squares solutions for finite weights tending to ∞

Asymptotic Analysis of LIME for Images

Garreau, Mardaoui

What Does LIME Really See in Images?

ICML, 2021

LIME for Images

1. Decompose image into d superpixels (small, homogeneous patches)[5]
2. Can sample perturbed image Z by
   - Sample d Bernoulli(1/2) variables $B = (B^1, \ldots, B^d)$
   - If $B^j = 1$, then keep j-th superpixel from original image
   - If $B^j = 0$, then replace j-th superpixel by its average pixel value.

predicted: trailer_truck (35.2%)

LIME explanation

[5] Image courtesy of Damien Garreau

LIME for Images

1. Decompose image into d superpixels (small, homogeneous patches)
2. Can sample perturbed image Z by
   - Sample d Bernoulli(1/2) variables $B = (B^1, \ldots, B^d)$
   - If $B^j = 1$, then keep j-th superpixel from original image
   - If $B^j = 0$, then replace j-th superpixel by its average pixel value.
3. Query response $\tilde{Y} = f(Z)$
4. Weight image Z by distance to original

$$\pi = \exp\left(-\frac{\mathrm{d}_{\cos}(B, \mathbb{1})^2}{2\nu^2}\right) \qquad \text{for hyperparameter } \nu > 0$$

5. Sample n times and fit weighted ridge regression[5]

$$\hat{\theta}_n = \operatorname*{arg\,min}_{\theta \in \mathbb{R}^d} \min_{\theta_0 \in \mathbb{R}} \sum_{i=1}^{n} \pi_i (\tilde{Y}_i - B_i^\intercal \theta - \theta_0)^2 + \lambda \|\theta\|^2$$

[5] In practice $\lambda = 1$ is tiny; in analysis take $\lambda = 0$ for simplicity.

Asymptotic Analysis of LIME for Images

- Recall that $B = (Z^1, \ldots, Z^d)$ i.i.d. Bernoulli(1/2)
- Induces distribution on weight π and perturbed image Z

Theorem (Garreau, Mardaoui, 2021)

Suppose f bounded and $\lambda = 0$. Then

$$\hat{\theta}_n \to \theta \qquad \text{in probability,}$$

where

$$\theta_j = c_1 \mathop{\mathbb{E}}_B[\pi f(Z)] + c_2 \mathop{\mathbb{E}}_B[\pi B^j f(Z)] + c_3 \sum_{\substack{k \in \{1,\ldots,d\} \\ k \neq j}} \mathop{\mathbb{E}}_B[\pi B^k f(Z)]$$

for some constants c_1, c_2, c_3 that do not depend on f, and which can be computed in closed form.

Consequences

$$\theta_j = c_1 \mathop{\mathbb{E}}_B[\pi f(Z)] + c_2 \mathop{\mathbb{E}}_B[\pi B^j f(Z)] + c_3 \sum_{\substack{k \in \{1,\ldots,d\} \\ k \neq j}} \mathop{\mathbb{E}}_B[\pi B^k f(Z)]$$

Consequence 1

- Apart from sampling noise, LIME explanations are linear in f:

$$\theta^{f+g} = \theta^f + \theta^g$$

Consequence 2: Large Bandwidth

- As $\nu \to \infty$: $c_1 \to -2$, $c_2 \to 4$, $c_3 \to 0$, and $\pi \to 1$ a.s.

$$\theta_j \to 2\left(\mathop{\mathbb{E}}_B[f(Z)|B^j = 1] - \mathop{\mathbb{E}}_B[f(Z)] \right)$$

- Compares value of f with and without fixing the j-th superpixel to be as in the model.

Discussion: What are local approximations good for?

Common question:
Which local approximation method should I use?

Discussion: What are local approximations good for?

Common question:
Which local approximation method should I use?

Current state of affairs:

- **Nobody knows**, because none of the approximation methods specify **under which conditions** or for **what purpose** they can be used
- In practice: people use the method(s) with best software; e.g. SHAP
- And sometimes they are impressed that SHAP has a justification from the economics literature, without considering whether the SHAP axioms are appropriate for their task: motivation by mathematical intimidation.

Discussion: What are local approximations good for?

Common question:
Which local approximation method should I use?

Current state of affairs:

- **Nobody knows**, because none of the approximation methods specify **under which conditions** or for **what purpose** they can be used
- In practice: people use the method(s) with best software; e.g. SHAP
- And sometimes they are impressed that SHAP has a justification from the economics literature, without considering whether the SHAP axioms are appropriate for their task: motivation by mathematical intimidation.

What can be done?

Discussion: What are local approximations good for?

Common question:
Which local approximation method should I use?

One Possible View:

- Doshi-Velez and Kim, 2017: we should provide explanations when the **user's goal is not fully specified**.
- If we take this seriously, then the user should be able to achieve at least some goals using the explanations. What are they?

Outline

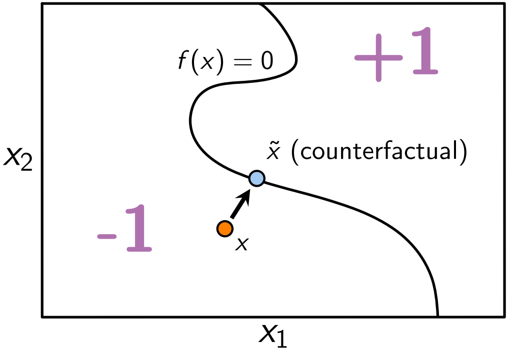
Introduction

Local Function Approximation Methods

Algorithmic Recourse

Example: Counterfactual Explanations

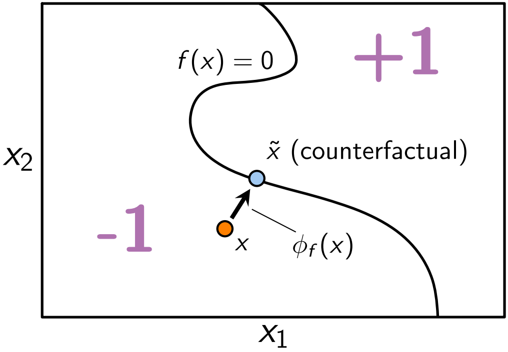
"If you would have had an income of **€40 000** instead of **€35 000**, your loan request would have been approved."



Counterfactual explanation: $\tilde{x} = \underset{x' : \text{sign}(f(x')) = +1}{\arg\min} \; \text{dist}(x', x)$

Example: Counterfactual Explanations

"If you would have had an income of **€40 000** instead of **€35 000**, your loan request would have been approved."

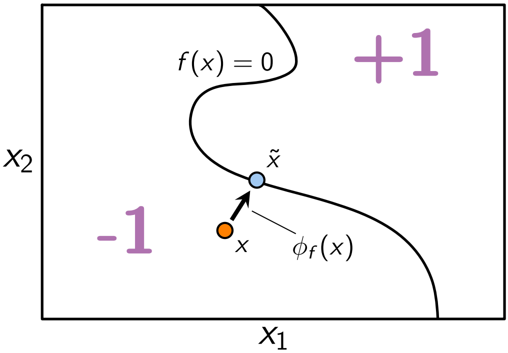


Counterfactual explanation: $\tilde{x} = \underset{x' : \text{sign}(f(x')) = +1}{\arg\min} \text{dist}(x', x)$

Viewed as attribution method: $\phi_f(x) = \tilde{x} - x$

Explanations with Recourse as their Goal

"If you change your current income of **€35 000** to **€40 000**, then your loan request will be approved."



- Attribution methods **provide recourse** if they tell the user how to **change their features** such that f **takes their desired value**.

An Impossibility Result

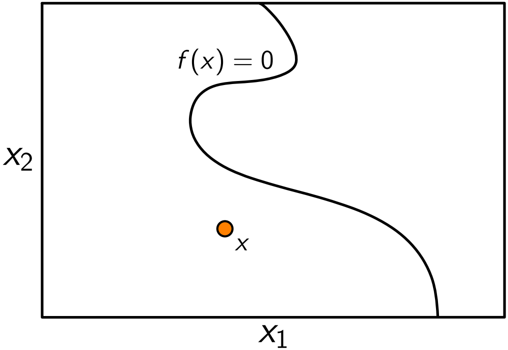
Fokkema, De Heide, Van Erven

Attribution-based Explanations that Provide Recourse Cannot be Robust

ArXiv:2205.15834 preprint, 2023

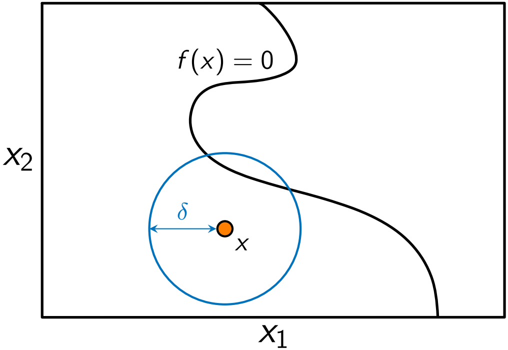
Recourse Sensitivity

- (Fokkema, de Heide, and van Erven, 2023): our approach to define weakest possible requirement for providing recourse.



Recourse Sensitivity

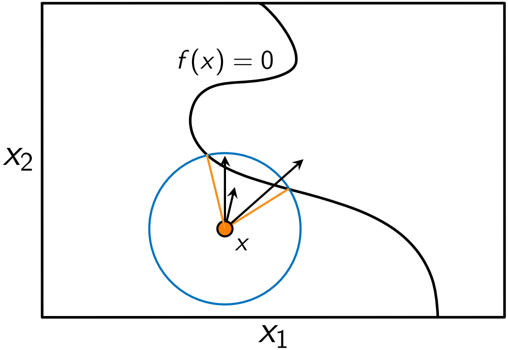
- (Fokkema, de Heide, and van Erven, 2023): our approach to define weakest possible requirement for providing recourse.



1. Assume user can change their features by at most some $\delta > 0$

Recourse Sensitivity

- (Fokkema, de Heide, and van Erven, 2023): our approach to define weakest possible requirement for providing recourse.



1. Assume user can change their features by at most some $\delta > 0$
2. $\phi_f(x)$ can point in **any direction that provides recourse** within distance δ, and length does not matter as long as it is > 0.
3. If no direction provides recourse, then $\phi_f(x)$ can be arbitrary.

Robustness of Explanations

Compare:

1. "If you change your current income of **€35 000** to **€40 000**, then your loan request will be approved."
2. "If you change your current income of **€35 001** to **€45 000**, then your loan request will be approved."

Minor changes in x should not cause big changes in explanations!

Robustness of Explanations

Compare:

1. "If you change your current income of **€35 000** to **€40 000**, then your loan request will be approved."
2. "If you change your current income of **€35 001** to **€45 000**, then your loan request will be approved."

Minor changes in x should not cause big changes in explanations!

Robustness: If f is continuous, then ϕ_f should also be **continuous**.
(e.g. survey of recourse by Karimi et al., 2021)

Robustness of Explanations

Compare:

1. "If you change your current income of **€35 000** to **€40 000**, then your loan request will be approved."
2. "If you change your current income of **€35 001** to **€45 000**, then your loan request will be approved."

Minor changes in x should not cause big changes in explanations!

Robustness: If f is continuous, then ϕ_f should also be **continuous**.
(e.g. survey of recourse by Karimi et al., 2021)

On the robustness of interpretability methods

D Alvarez-Melis, TS Jaakkola

arXiv preprint arXiv:1806.08049, 2018 • arxiv.org

We argue that robustness of explanations---i.e., that similar inputs should give rise to similar explanations---is a key desideratum for interpretability. We introduce metrics to quantify robustness and demonstrate that current methods do not perform well according to these metrics. Finally, we propose ways that robustness can be enforced on existing interpretability approaches.

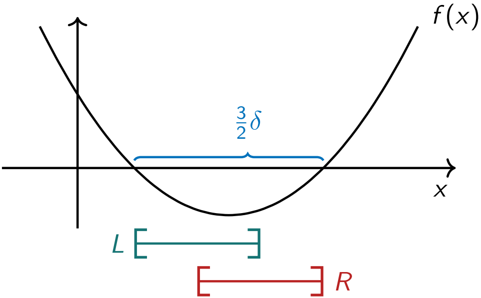
Impossibility in Binary Classification

Theorem (Fokkema, De Heide, Van Erven, 2022)

For any $\delta > 0$ there exists a continuous function f such that no attribution method ϕ_f can be both recourse sensitive and continuous.

- Power of math: can reason about all explanation methods that could possibly exist

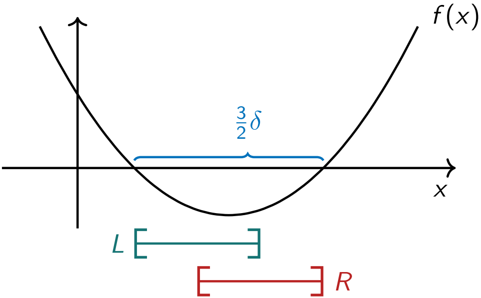
Proof Sketch



$L = \{x : \text{recourse possible by moving at most } \delta \text{ left}\}$

$R = \{x : \text{recourse possible by moving at most } \delta \text{ right}\}$

Proof Sketch

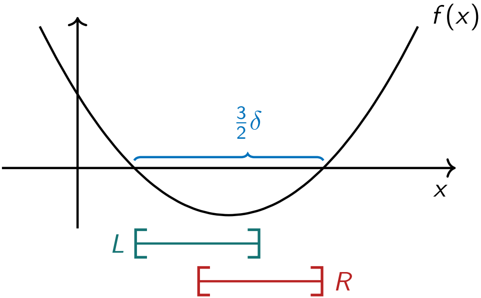


$$L = \{x : \text{recourse possible by moving at most } \delta \text{ left}\}$$
$$R = \{x : \text{recourse possible by moving at most } \delta \text{ right}\}$$

Recourse sensitivity implies:

$$\phi_f(x) \begin{cases} < 0 & \text{for } x \in L \setminus R \\ > 0 & \text{for } x \in R \setminus L \\ \neq 0 & \text{for } x \in L \cap R \end{cases}$$

Proof Sketch



$$L = \{x : \text{recourse possible by moving at most } \delta \text{ left}\}$$
$$R = \{x : \text{recourse possible by moving at most } \delta \text{ right}\}$$

Recourse sensitivity implies:

$$\phi_f(x) \begin{cases} < 0 & \text{for } x \in L \setminus R \\ > 0 & \text{for } x \in R \setminus L \\ \neq 0 & \text{for } x \in L \cap R \end{cases}$$

But this **contradicts continuity**!
(by the mean-value theorem)

Can embed 1D example in higher dimensions as well.

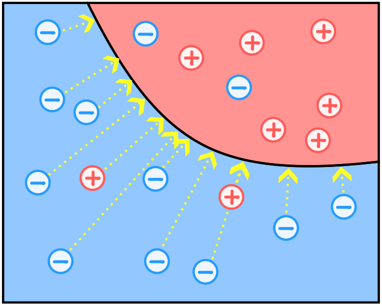
Is Algorithmic Recourse a Good Idea at All?

Fokkema, Garreau, Van Erven

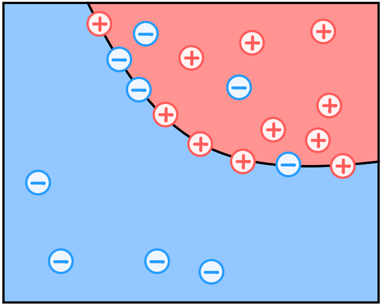
The Risks of Recourse in Binary Classification

ArXiv::2306.00497 preprint, 2023

Effect of Recourse on the Population



Before recourse



After recourse

What happens to the accuracy of the classifier?

- Accuracy matters!
  For example, incorrect $+1$ classifications $=$ users defaulting on loans

Effect of Recourse

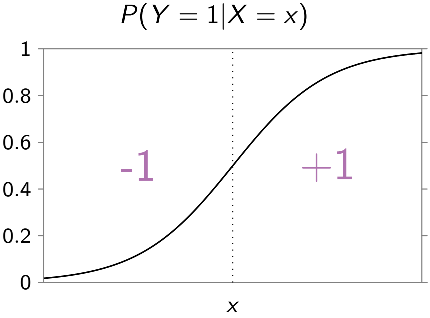
Situation before Recourse:

- User distribution: $(X_0, Y) \sim P$
- Classifier $f : \mathcal{X} \to \{-1, +1\}$
- Risk: $R_P(f) = P(f(X_0) \neq Y)$

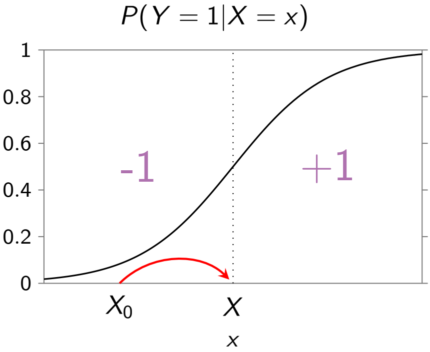
Effect of Recourse:

- User features change from X_0 to X
- Distribution of Y may change

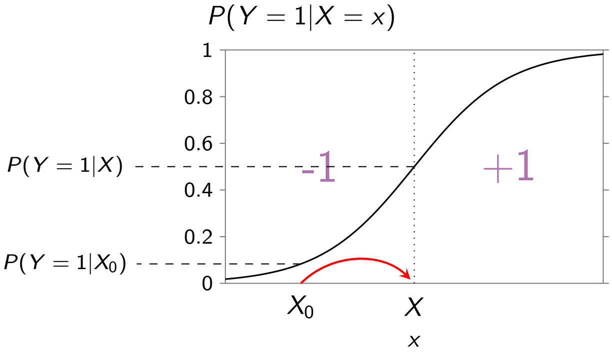
Need to Model User Behavior



Need to Model User Behavior



Need to Model User Behavior



- **Compliant users:** probability of Y after recourse is $P(Y|X)$
- **Defiant users:** probability of Y after recourse is $P(Y|X_0)$

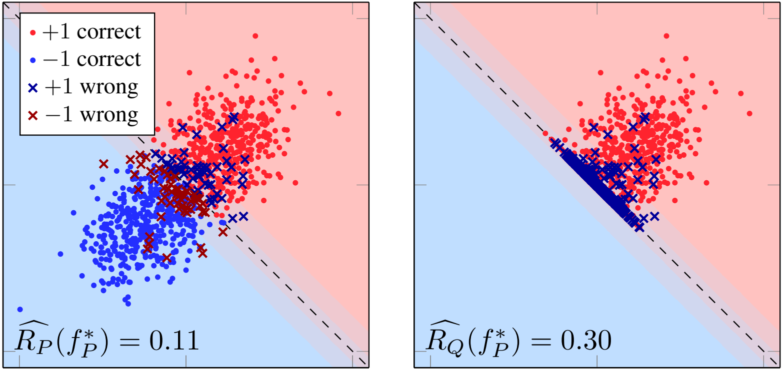
Need to Model User Behavior

Examples:

- Credit loan application:
  - Compliant: Applicant improves risky behaviour
  - Defiant: Applicant tries to "game the system"
- Medical Diagnosis:
  - Compliant: Patient improves their health
  - Defiant: Patient takes medicine to reduce symptoms
- Job applications:
  - Compliant: Applicant improves their skills
  - Defiant: Applicant improves their CV

- **Compliant users:** probability of Y after recourse is $P(Y|X)$
- **Defiant users:** probability of Y after recourse is $P(Y|X_0)$

Effect of Recourse on Population-level Accuracy



Before recourse

After recourse
(compliant users)

- Simulation with Gaussian data
- **Average nr. of mistakes goes up / accuracy goes down**
- Many more customers defaulting on their loans!

Learning-theoretic Framework

Situation before Recourse:

- User distribution: $(X_0, Y) \sim P$
- Classifier $f : \mathcal{X} \to \{-1, +1\}$
- Risk: $R_P(f) = P(f(X_0) \neq Y)$

Learning-theoretic Framework

Situation before Recourse:

- User distribution: $(X_0, Y) \sim P$
- Classifier $f : \mathcal{X} \to \{-1, +1\}$
- Risk: $R_P(f) = P(f(X_0) \neq Y)$
- Users' choice to accept recourse is $B \in \{0, 1\}$ with $\Pr(B = 1 | X_0) = r(X_0)$.

Situation with Recourse:

- Users arrive as before: $X_0 \sim P$
- Recourse proposal: $X^{\text{CF}} = \arg\min_{x : f(x) = +1} \|x - X_0\|$
- Users' choice to accept is $B \in \{0, 1\}$ with $\Pr(B = 1 | X_0) = r(X_0)$:

$$X = (1 - B)X_0 + BX^{\text{CF}}$$

- Q is the resulting distribution of X_0, B, X, Y
- Risk: $R_Q(f) = Q(f(X_0) \neq Y)$

Recourse Increases the Risk

Bayes-optimal classifier under P:

$$f_P^* = \arg\min_f R_P(f)$$

$$f_P^*(x) = \begin{cases} +1 & \text{if } P(Y = 1 | X_0 = x) \geq 1/2, \\ -1 & \text{otherwise.} \end{cases}$$

Recourse Increases the Risk

Bayes-optimal classifier under P:

$$f_P^* = \arg\min_f R_P(f)$$

$$f_P^*(x) = \begin{cases} +1 & \text{if } P(Y = 1 | X_0 = x) \geq 1/2, \\ -1 & \text{otherwise.} \end{cases}$$

Regularity conditions:

- Well-defined setup: $\{x \in \mathcal{X} : f_P^*(x) = +1\}$ is closed
- Continuous conditional probabilities: $P(Y = 1 | X_0 = x) = 1/2$ for all x on the decision boundary of f_P^*

Theorem

Then, both if the users are **defiant** *and if the users are* **compliant**, **recourse always increases the risk***:*

$$R_Q(f_P^*) \geq R_P(f_P^*).$$

The inequality is strict if the probability of recourse in the negative class is non-zero: $P(B = 1, f_P^*(X_0) = -1) > 0$.

Recourse Increases the Risk

Regularity conditions:

- Well-defined setup: $\{x \in \mathcal{X} : f_P^*(x) = +1\}$ is closed
- Continuous conditional probabilities: $P(Y = 1|X_0 = x) = 1/2$ for all x on the decision boundary of f_P^*

Theorem

Then, both if the users are **defiant** *and if the users are* **compliant**, **recourse always increases the risk***:*

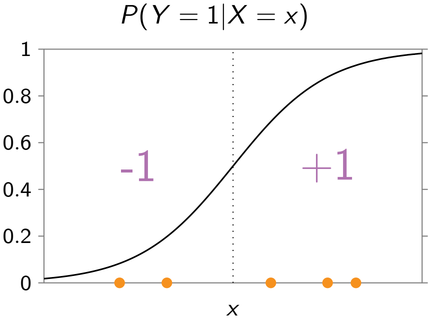
Defiant case:

$$\begin{aligned} R_Q(f_P^*) &= P(B = 1, Y = -1) - P(B = 1, f_P^*(X_0) \neq Y) + R_P(f_P^*) \\ &\geq R_P(f_P^*) \end{aligned}$$

Compliant case:

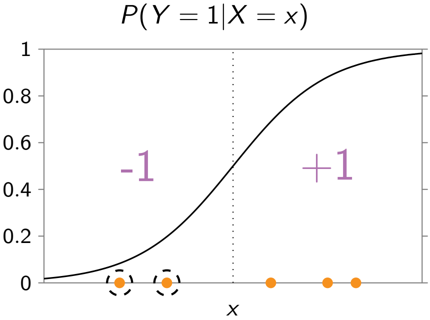
$$\begin{aligned} R_Q(f_P^*) &= \tfrac{1}{2} P(B = 1, f_P^*(X_0) = -1) - P(B = 1, f_P^*(X_0) = -1, Y = 1) \\ &\quad + R_P(f_P^*) \\ &\geq R_P(f_P^*). \end{aligned}$$

Proof Idea: Defiant Case



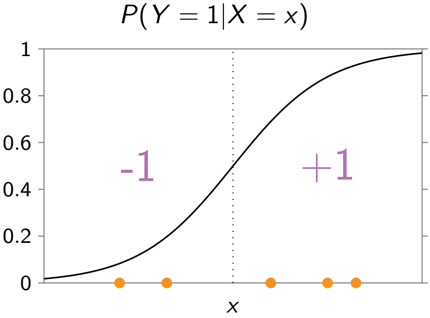
- Defiant case: $Q(Y|X, X_0) = P(Y|X_0)$

Proof Idea: Defiant Case



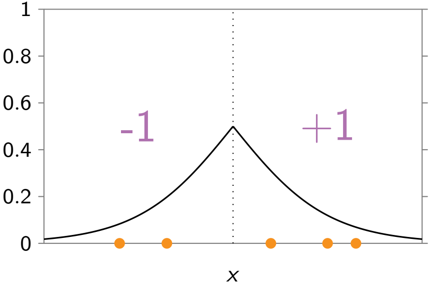
- Defiant case: $Q(Y|X, X_0) = P(Y|X_0)$
- Recourse misclassifies users from class -1 as class $+1$

Proof Idea: Compliant Case



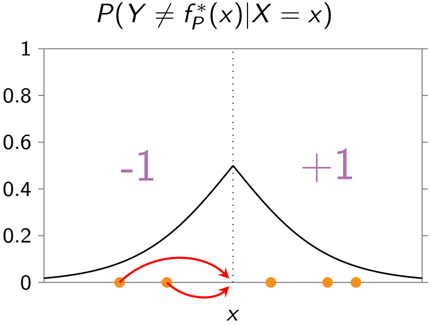
Proof Idea: Compliant Case

$P(Y \neq f_P^*(x) | X = x)$



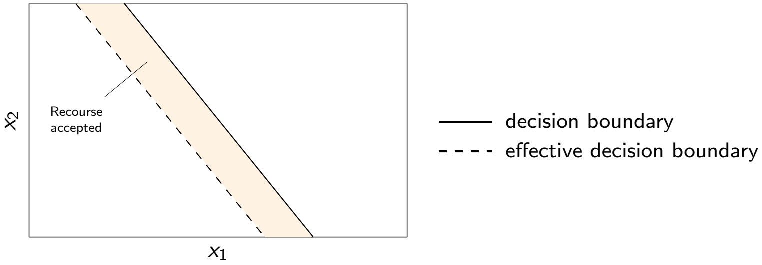
- Compliant case: $Q(Y|X, X_0) = P(Y|X)$

Proof Idea: Compliant Case



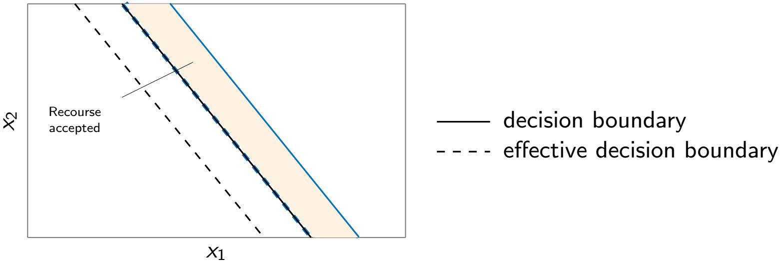
- Compliant case: $Q(Y|X, X_0) = P(Y|X)$
- Recourse moves users from high certainty to lowest certainty region

Strategic Classification



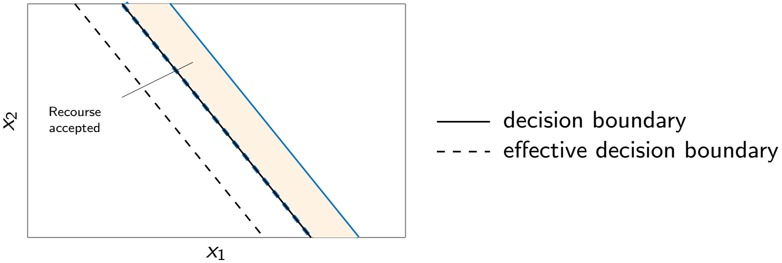
- Suppose recourse accepted deterministically within distance D of decision boundary

Strategic Classification



- Suppose recourse accepted deterministically within distance D of decision boundary
- **Cancel effect of recourse** by moving decision boundary back by distance D

Strategic Classification



- Suppose recourse accepted deterministically within distance D of decision boundary
- **Cancel effect of recourse** by moving decision boundary back by distance D

Definition

A set of classifiers $\mathcal{F}$ is **invariant under recourse** if for any $f \in \mathcal{F}$ there exists a **unique** $f' \in \mathcal{F}$ such that the decision boundary for f without recourse is equal to the effective decision boundary of f' with recourse.

Strategic Classification

Assumptions:

- $\mathcal{F}$ invariant under recourse

Theorem (Defiant Case)

Recourse has no effect:

$$\min_{f \in \mathcal{F}} R_{Q_f}(f) = \min_{f \in \mathcal{F}} R_P(f).$$

- Write Q_f instead of Q to emphasize dependence of the effect of recourse on f.

Strategic Classification

Assumptions:

- $\mathcal{F}$ invariant under recourse

Theorem (Compliant Case)

Recourse may have positive effect:
Let $\bar{f} \in \arg\min_{f \in \mathcal{F}} R_P(f)$ with corresponding $f' \in \mathcal{F}$ that has the same effective decision boundary after recourse. Then

$$\min_{f \in \mathcal{F}} R_{Q_f}(f) \leq R_{Q_{f'}}(\bar{f}).$$

- Think of $Q_{f'}$ as moving users away from the decision boundary compared to P, so plausible that $R_{Q_{f'}}(\bar{f}) < R_P(\bar{f})$.
- Only case where we find that recourse is **beneficial** in terms of accuracy.
- But cancels the effect of recourse and does not help any users from the original -1 class. Not really what we imagined...

Conclusion

Zooming Out

- Most work on explainability is empirical
- Empirical approach has been very successful in deep learning, but struggles to find proper foundations for explainability
- Formal analysis is slow and leads to more modest claims, but builds up **solid foundations**

Where Do We Go From Here?

1. **Formalize the many possible goals** of explainability
2. Bring exaggerated empirical claims down to earth by proving **necessary/sufficient conditions**
3. Better understanding of limitations $\Longrightarrow$ develop better explanations
4. Explainability results for inverse problems? What are the key questions?

References I

Adebayo, Julius, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim (2018). "Sanity Checks for Saliency Maps". In: *Advances in Neural Information Processing Systems*. Vol. 31.

Agarwal, Sushant, Shahin Jabbari, Chirag Agarwal, Sohini Upadhyay, Steven Wu, and Himabindu Lakkaraju (2021). "Towards the Unification and Robustness of Perturbation and Gradient Based Explanations". In: *Proceedings of the 38th International Conference on Machine Learning*, pp. 110–119.

Alvarez-Melis, David and Tommi S. Jaakkola (2018). "On the Robustness of Interpretability Methods". In: *ArXiv:1806.08049 preprint*.

Dombrowski, Ann-Kathrin, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel (2019). "Explanations can be manipulated and geometry is to blame". In: *Advances in Neural Information Processing Systems*. Vol. 32.

Doshi-Velez, Finale and Been Kim (2017). "Towards A Rigorous Science of Interpretable Machine Learning". In: *ArXiv:1702.08608 preprint*.

Fokkema, Hidde, Rianne de Heide, and Tim van Erven (2023). "Attribution-based Explanations that Provide Recourse Cannot be Robust". In: *ArXiv:2205.15834 preprint*.

Fokkema, Hidde, Damien Garreau, and Tim van Erven (2023). "The Risks of Recourse in Binary Classification". In: *ArXiv:2306.00497 preprint*.

References II

Garreau, Damien and Dina Mardaoui (2021). "What does LIME really see in images?" In: *Proceedings of the 38th International Conference on Machine Learning*, pp. 3620–3629.

Han, Tessa, Suraj Srinivas, and Himabindu Lakkaraju (2022). "Which Explanation Should I Choose? A Function Approximation Perspective to Characterizing Post Hoc Explanations". In: *Advances in Neural Information Processing Systems*. Vol. 35, pp. 5256–5268.

Karimi, Amir-Hossein, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera (2021). "A survey of algorithmic recourse: definitions, formulations, solutions, and prospects". In: *arXiv preprint arXiv:2010.04050*.

Krishna, Satyapriya, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju (2022). "The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective". In: *ArXiv 2202.01602 preprint*.

Lundberg, Scott M and Su-In Lee (2017). "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Vol. 30.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). ""Why should I trust you?" Explaining the predictions of any classifier". In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144.

References III

Smilkov, Daniel, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg (2017). "SmoothGrad: removing noise by adding noise". In: *ArXiv:1706.03825*.

Tjoa, Erico and Cuntai Guan (2021). "A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI". In: *IEEE Transactions on Neural Networks and Learning Systems* 32.11, pp. 4793–4813. DOI: 10.1109/TNNLS.2020.3027314.

Young, H. P. (1985). "Monotonic solutions of cooperative games". In: *International Journal of Game Theory* 14, pages 65–72.

Zhou, Jianlong, Amir H. Gandomi, Fang Chen, and Andreas Holzinger (2021). "Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics". In: *Electronics* 10.5. ISSN: 2079-9292. DOI: 10.3390/electronics10050593.