PADERBORN UNIVERSITY

Local Model-Agnostic Methods

Explainable Artificial Intelligence

Dr. Stefan Heindorf

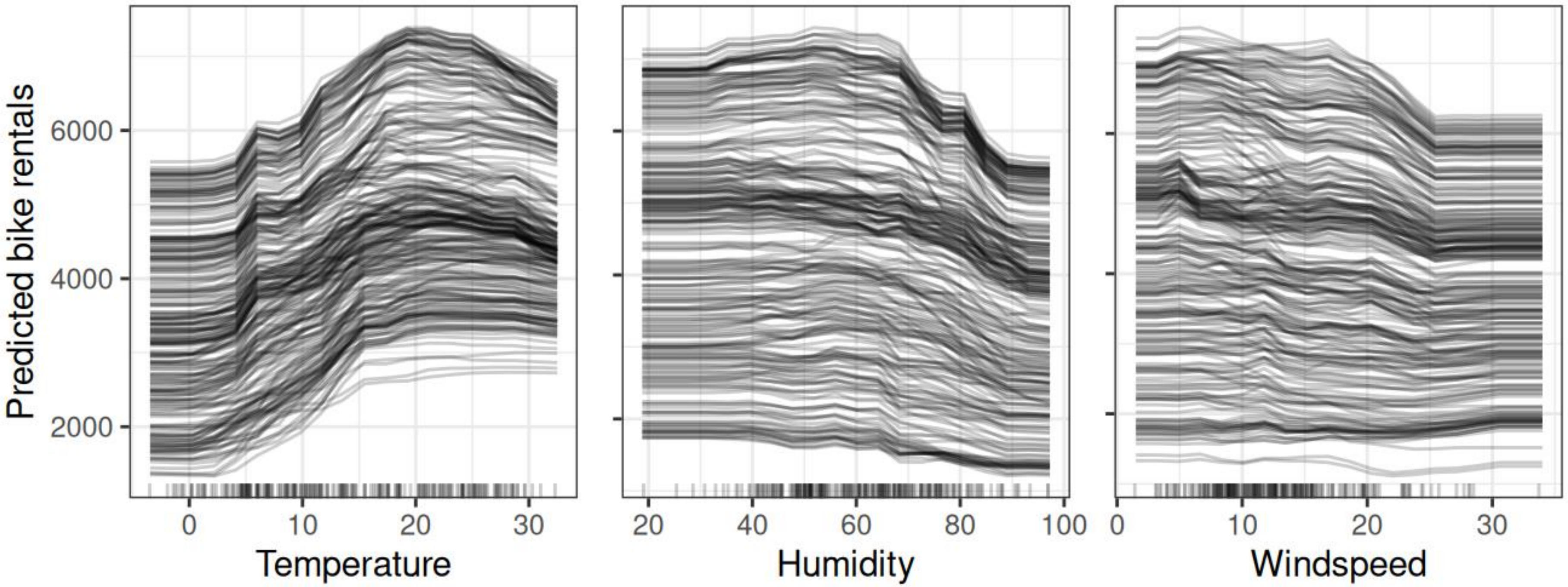
Outlook

- ICE
- LIME
- Anchors

Individual Conditional Expecctation (ICE)

Individual conditional expectation (ICE)

Example



Individual conditional expectation (ICE)

Display one line per instance showing how the instance's prediction changes when a feature changes

Similar to partial dependence plots

- PDP averages over all instances (global method)
- ICE shows single instances (local method)

Why consider single instances instead of all?

- Partial dependence plots can obscure a heterogeneous relationship created by interactions (average can hide important relationships)

Formal definition

- $\hat{f}_S^{(i)}(\mathbf{x}_S) = \hat{f}(\mathbf{x}_S, \mathbf{x}_C^{(i)})$ (in contrast to PDP, there is no average)
- The curve $\hat{f}_S^{(i)}(\mathbf{x}_S)$ is plotted for every instance i

Centered ICE plot (c-ICE)

Problem of ICE plots

- Can be hard to tell whether the ICE curves differ between individuals (because they start at different predictions)

Solution

- Display only the difference in the prediction to an anchor point (usually curves are anchored to the lower end)

Definition

$$\hat{f}^{(i)}_{S,\,cent}(\mathbf{x}_S) = \hat{f}^{(i)}_S(\mathbf{x}_S) - \hat{f}^{(i)}_S(\mathbf{x}^a_S)$$

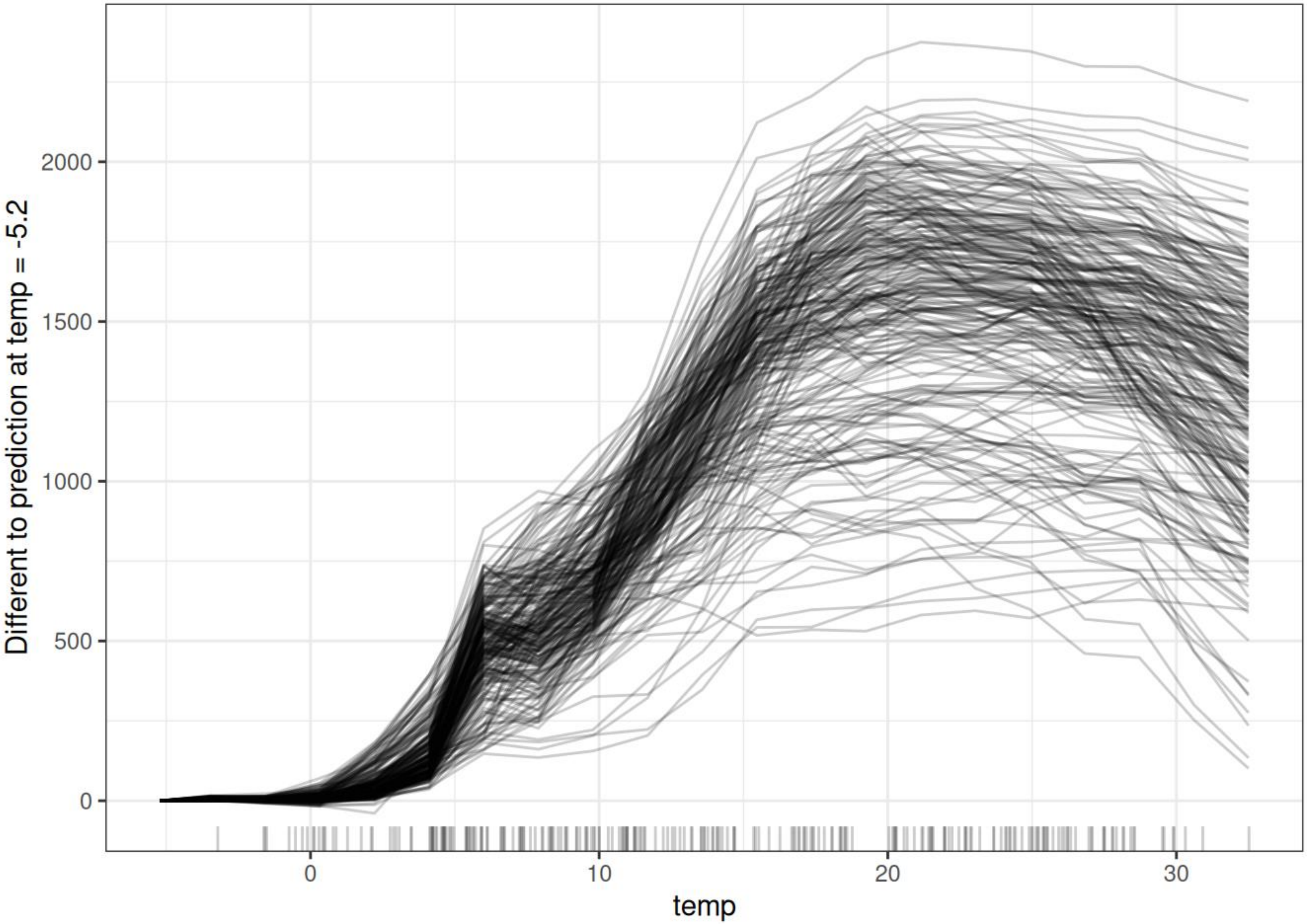
Where

$\hat{f}^{(i)}_S$ ICE plot

$\mathbf{x}^a_S$ anchor point (usually the smallest value $\mathbf{x}_S$ that is plotted)

Centered ICE Plot (c-ICE)

Example: Bike rental



Individual conditional expectation (ICE)

Advantages and disadvantages

Advantages

- Even more intuitive than partial dependence plots
- One line per instance
- Can uncover heterogeneous relationships
  (e.g., some ICE curves go up and some down)

Disadvantages

- Can only display one feature meaningfully
- If feature is correlated with other features
  → Some points in the lines might be invalid
- If many ICE curves are drawn
  → Plot can become overcrowded (and you will not see anything)
  → Solution 1: add transparency to the lines
  → Solution 2: draw a sample of the lines
- It might be difficult to see the average
  → Solution: Combine ICE plot and PDP plot

Local Surrogate (LIME)

LIME

Local interpretable model-agnostic explanations [Ribeiro, 2016]

Assumptions

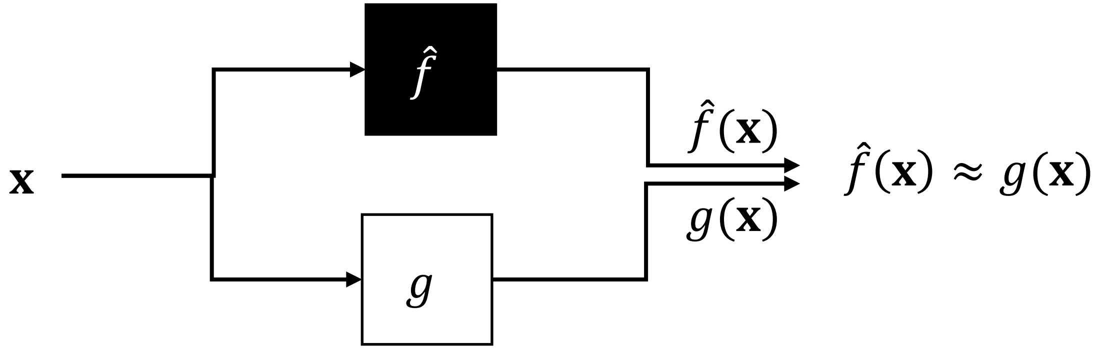
- Underlying model is a black box
- Original data is not available anymore
- Only the inputs and outputs of the black box model can be observed

Task

- Explain single prediction

Solution

- Probe the black box model $\hat{f}$ near the prediction
- Train local surrogate model g
  (surrogate model can be any interpretable model, e.g., LASSO, decision tree)



LIME: Example for tabular data

A

- Black box predictions given features x_1 and x_2
- Predicted classes: 1 (dark) or 0 (light)

B

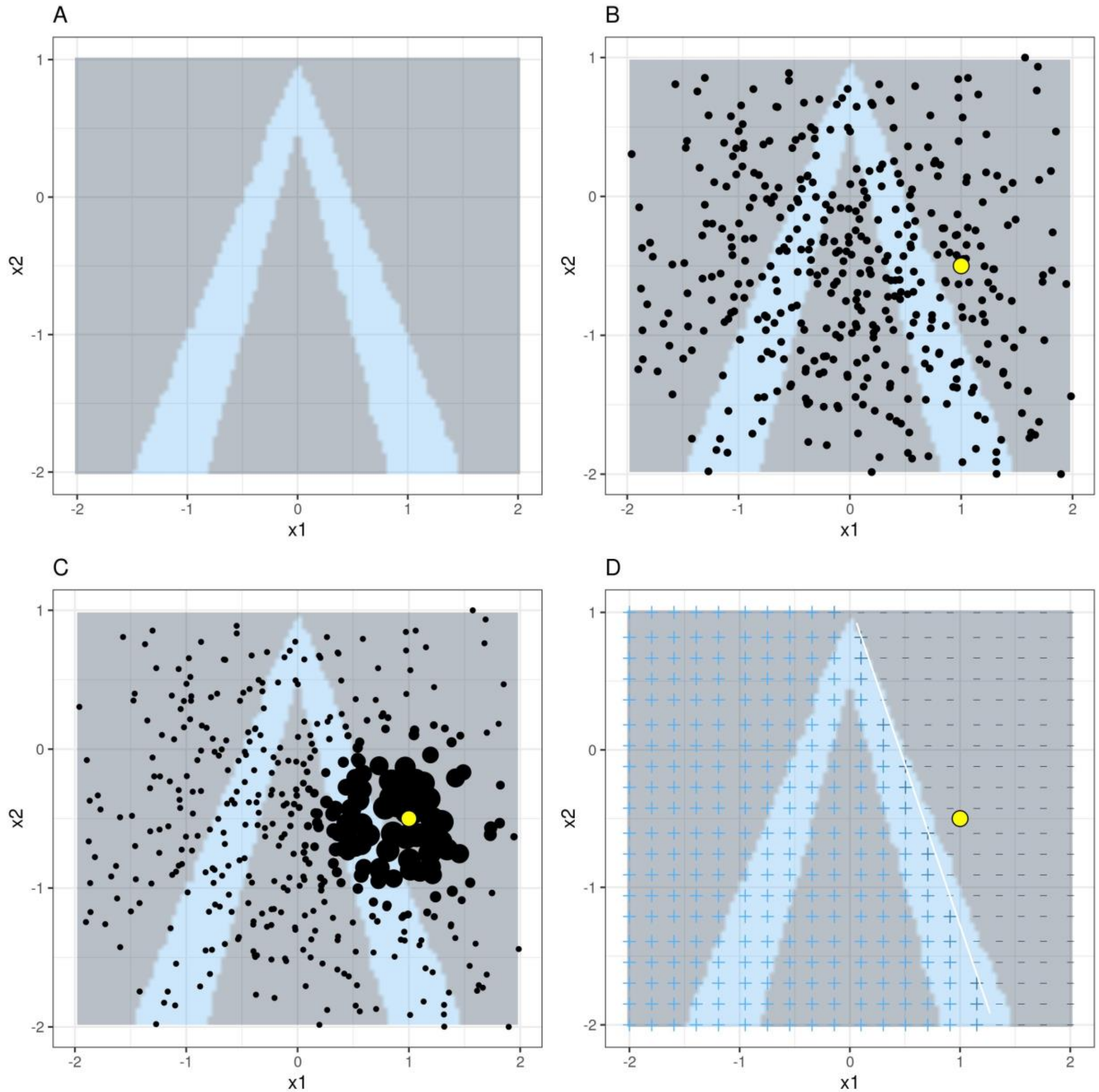
- Instance of interest (big dot)
- Data sampled from a normal distribution (small dots)

C

- Assign higher weight to points near the instance of interest

D

- Signs of the grid show the classifications of the locally learned model from the weighted samples
- The white line marks the decision boundary (P(class=1) = 0.5)



Christoph Molnar (https://christophm.github.io/interpretable-ml-book/images/lime-fitting-1.jpeg), https://creativecommons.org/licenses/by-nc-sa/4.0/

LIME: General framework

$$\text{explanation}(\mathbf{x}) = \underset{g \in G}{arg\,min}\, L\left(\hat{f}, g, \pi_{\mathbf{x}}\right) + \Omega(g)$$

G	family of interpretable models (e.g., linear regression models)
L	loss (e.g., mean squared error)
$g: \mathbb{R}^M \rightarrow \mathbb{R}$	interpretable explanation model
$\hat{f}: \mathbb{R}^p \rightarrow \mathbb{R}$	original model (usually complex model, e.g., xgboost)
$\pi_{\mathbf{x}}(z)$	proximity measure between instance $\mathbf{x}$ and $\mathbf{z}$ (to define "size of neighborhood" of $\mathbf{x}$)
$\Omega(g)$	model complexity (e.g., number of features)

Note:

- feature space of interpretable model can be different from original model (e.g., g uses set-of-word features and $\hat{f}$ uses features after PCA)
- Interpretable model g often uses binary features $\{0, 1\}$ (e.g., features of f are discretized via binning)

LIME: Algorithm for training local surrogate models
[Ribeiro et al., 2016]

Algorithm

- **Require:** model $\hat{f}$, number of samples N ($\hat{f}$ is black box model)
- **Require:** instance $\mathbf{x}$, its interpretable version $\mathbf{x}'$ ($\mathbf{x}$ is instance of interest)
- **for** $\mathrm{i} \in \{1, 2, \dots, \mathrm{N}\}$ **do**:
  - $\mathbf{z}_i' \leftarrow perturb(\mathbf{x}')$
  - $Z \leftarrow Z \cup \langle \mathbf{z}_i', \hat{f}(\mathbf{z}_i), \pi_{\mathbf{x}}(\mathbf{z}_i) \rangle$ (interpretable version $\mathbf{z}_i'$ is transformed to $\mathbf{z}_i$)
- **end for**
- train weighted, interpretable model g on perturbed instances Z
- **return** g

What surrogate model g to use?

- Common choice: linear regression
- The number of features K is often chosen in advance (LASSO)
- Decrease the regularization parameter, until K features reached

How to perturb the data?

- Text: "Turn single words on or off"
- Image: "Turn single (super)pixels on or off"
- Tabular: Perturb each feature individually: Draw from a normal distribution with mean and standard deviation taken from the feature (mean and stddev of column)

LIME for sparse linear explanations

[Ribeiro et al., 2016]

Let G be the class of linear models, such that

$$g(\mathbf{z}') = \beta_0 + \sum_{j=1}^{M} \beta_j z'_j = \mathbf{z}'^{\mathbf{T}}\, \boldsymbol{\beta}$$

We use the locally weighted square loss:

$$L(\hat{f}, g, \pi_{\mathbf{x}}) = \sum_{\mathbf{z},\mathbf{z}' \in Z} \pi_{\mathbf{x}}(\mathbf{z})(\hat{f}(\mathbf{z}) - g(\mathbf{z}'))^2$$

where

$$\pi_{\mathbf{x}}(\mathbf{z}) = exp\left(-\frac{D(\mathbf{x}, \mathbf{z})^2}{\sigma^2}\right)$$

and

- D is a distance function (e.g., L2 distance)
- σ a width (an arbitrary hyperparameter)

Note: The loss measures local fidelity, i.e., how well the surrogate model approximates the original model in the neighborhood of instance $\mathbf{x}$

LIME: Example for tabular data

x_1	x_2	y
1	6	9000
3	2	11000

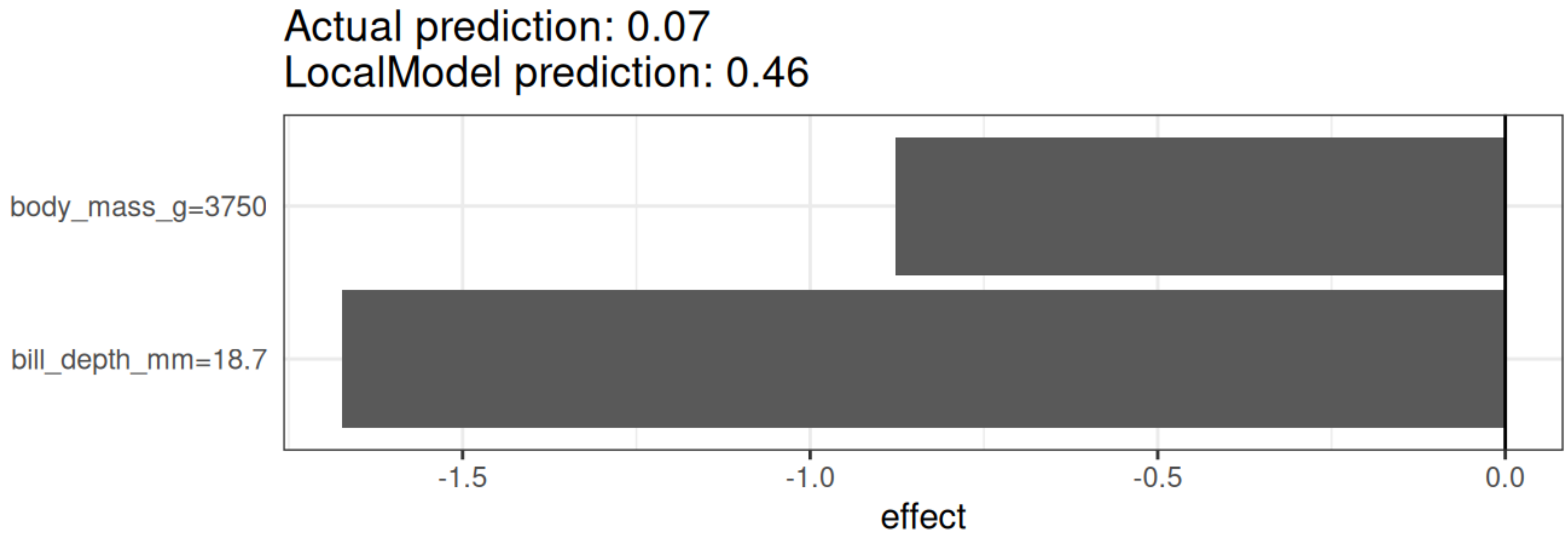
- Task: Explain row $(3,2)$ of the linear model $3000x_1 + 1000x_2$, with distance measure L2 and kernel width $\sigma = 0.75\sqrt{2} \approx 1.1$ (LIME default: 0.75 times sqrt of feature count)
- $Mean(x_1) = 2,\ stddev(x_1) = \sqrt{\frac{(1^2+1^2)}{2-1}} = \sqrt{2} \approx 1.4$
- $Mean(x_2) = 4,\ stddev(x_2) = \sqrt{\frac{(2^2+2^2)}{2-1}} = \sqrt{8} \approx 2.8$
- Sample new datapoints Z that serve as training data for new explainable model

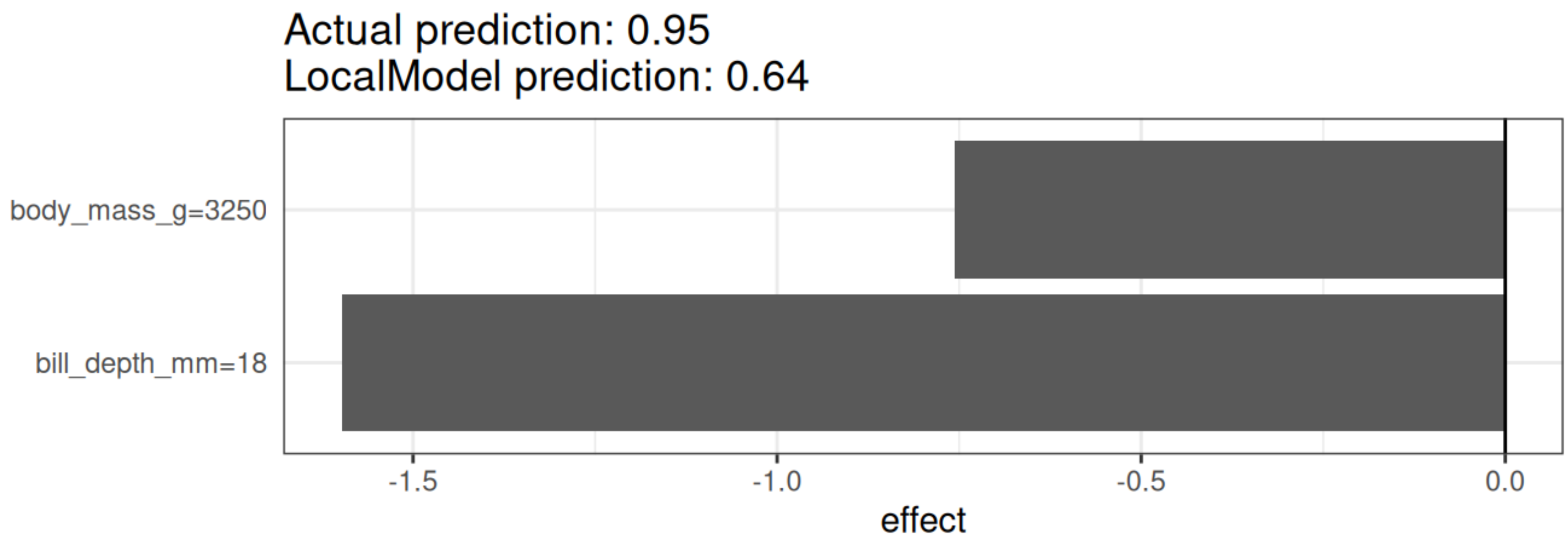
z_1	z_2	$\hat{y} = \hat{f}(z_1, z_2)$	$\boldsymbol{\pi_x(z)}$
2	4	10000	?
2	2	8000	?

- $\pi_{(3,2)}((2,4)) = \exp\left(-\frac{\|(2,4)-(3,2)\|_2^2}{1.1^2}\right) = \exp\left(-\frac{5}{1.1^2}\right) \approx 0.02$
- $\pi_{(3,2)}((2,2))\exp\left(-\frac{\|(2,2)-(3,2)\|_2^2}{1.1^2}\right) = \exp\left(-\frac{1}{1.1^2}\right) \approx 0.44$ → then train g as on previous slide

LIME: Example for tabular data

Example of two penguin instances: predict whether female





LIME: Example for text

Classify YouTube comments as spam / no spam

Predictions g(x) of black box model g for instance x

	CONTENT (x)	CLASS
267	PSY is a good guy	0
173	For Christmas Song visit my channel! ;)	1

Perturbations of instance 173

For	Christmas	Song	visit	my	channel!	;)	prob	weight
1	0	1	1	0	0	1	0.17	0.57
0	1	1	1	1	0	1	0.17	0.71
1	0	0	1	1	1	1	0.99	0.71
1	0	1	1	1	1	1	0.99	0.86
0	1	1	1	0	0	1	0.17	0.57

- prob predicted score $g(\mathbf{z})$ of model g for perturbed instance $\mathbf{z}$
- weight $\pi_{\mathbf{x}}(\mathbf{z}) := 1$ – proportion of removed words (e.g., $1 - 3 / 7 = 0.57$)

LIME: Example for text

Two sentences

- case 1: no spam
- case 2: spam

case	label_prob	feature	feature_weight
1	0.1701170	is	0.000000
1	0.1701170	good	0.000000
1	0.1701170	a	0.000000
2	0.9939024	channel!	6.180747
2	0.9939024	;)	0.000000
2	0.9939024	visit	0.000000

Visualization by coloring single words

- *is a good*
- *visit channel ;)*

LIME example for images

Image classification

Problem: perturbations of single pixels

- Hardly changes prediction
- Hardly visible

Solution: perturbations of super pixels

- Super pixel: interconnected pixels with similar colors
- Obtained via segmentation

Middle and right: LIME explanations for the top 2 classes (bagel, strawberry)

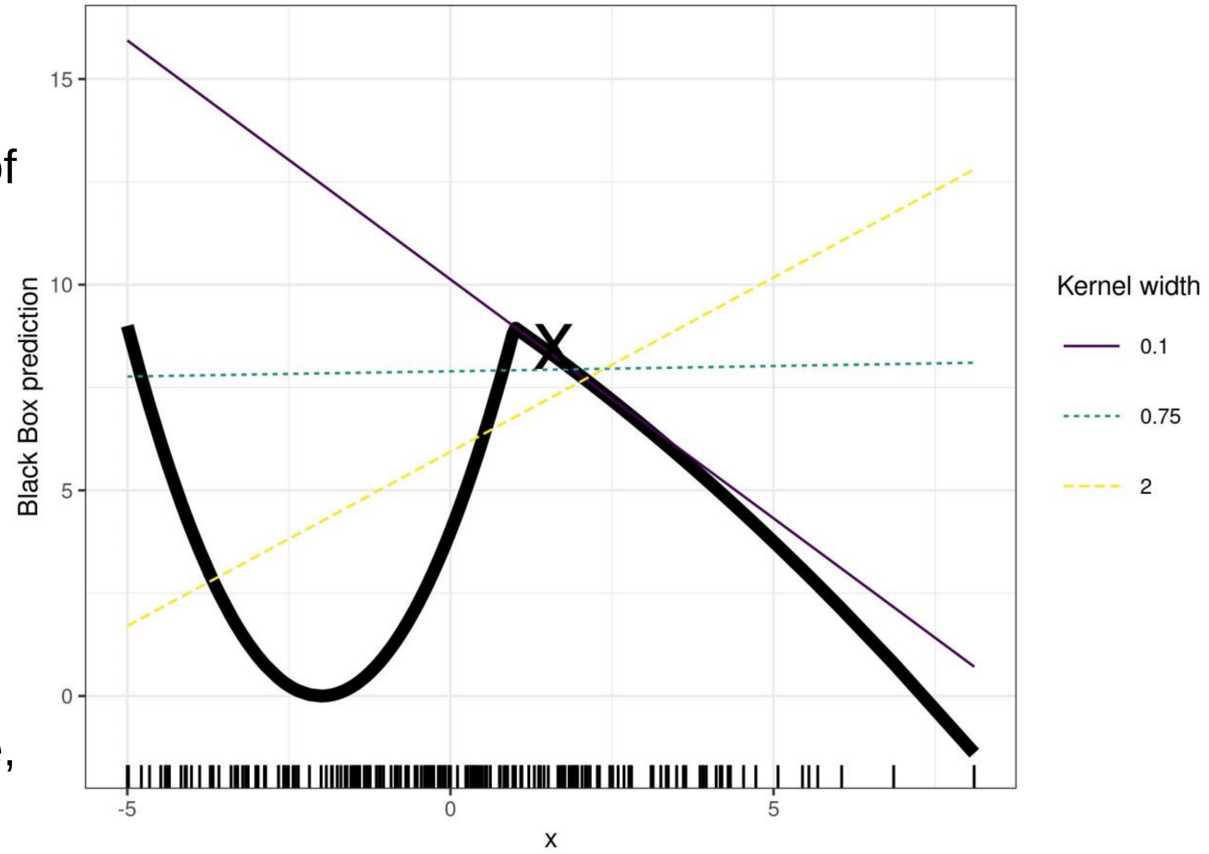
Christoph Molnar (https://christophm.github.io/interpretable-ml-book/images/lime-images-package-example-1.png), https://creativecommons.org/licenses/by-nc-sa/4.0/

Issue with LIME

Dependence on kernel width σ

- X: Explanation for instance x = 1.6
- Thick line: predictions of black box model (on a single feature)
- Rugs: data distribution
- Thin lines: local surrogate models (with different kernel widths)

- Problem: Does the feature have a negative, positive or no effect for x = 1.6?



$$\pi_x(z) = \exp\left(-\frac{D(x,z)^2}{\sigma^2}\right)$$

LIME
Advantages

Model $\hat{f}$ can be changed independently of explanation model g

- Example: Replace BB neural network by xgboost or vice versa (if it gives better predictions)
- Example: Replace linear model by short decision trees (if preferred by users)

Generates human-friendly explanations

- When using LASSO or short trees: explanations are short (=selective)
- Suitable for lay persons
- Not suitable for complete attributions (e.g., compliance scenarios requiring a full explanation)

LIME works for tabular data, text and images

The explanations created with local surrogate models can use other (interpretable) features than the original model was trained on

- Example: explanation model relies on word features, original model on word embeddings
- Example: features can be normalized/transformed, but the original features serve as explanations

LIME
Disadvantages

“Correct” definition of neighborhood is an unsolved problem

- Try many different kernel widths
- See what kernel width makes sense for your dataset and task

“Correct” perturbation function is an unsolved problem

- Data points sampled from Gaussian distribution (in current LIME implementation)
- Correlation between features is ignored

→ Unlikely data points might be sampled

Instability of the explanations [Alvarez-Melis, 2018]

- Explanations of two close points can vary greatly
- Different explanations for two repetitions of the sampling process (perturbation function)

→ instability makes it difficult to trust the explanations

Explanations can be manipulated to hide biases [Slack, 2020]

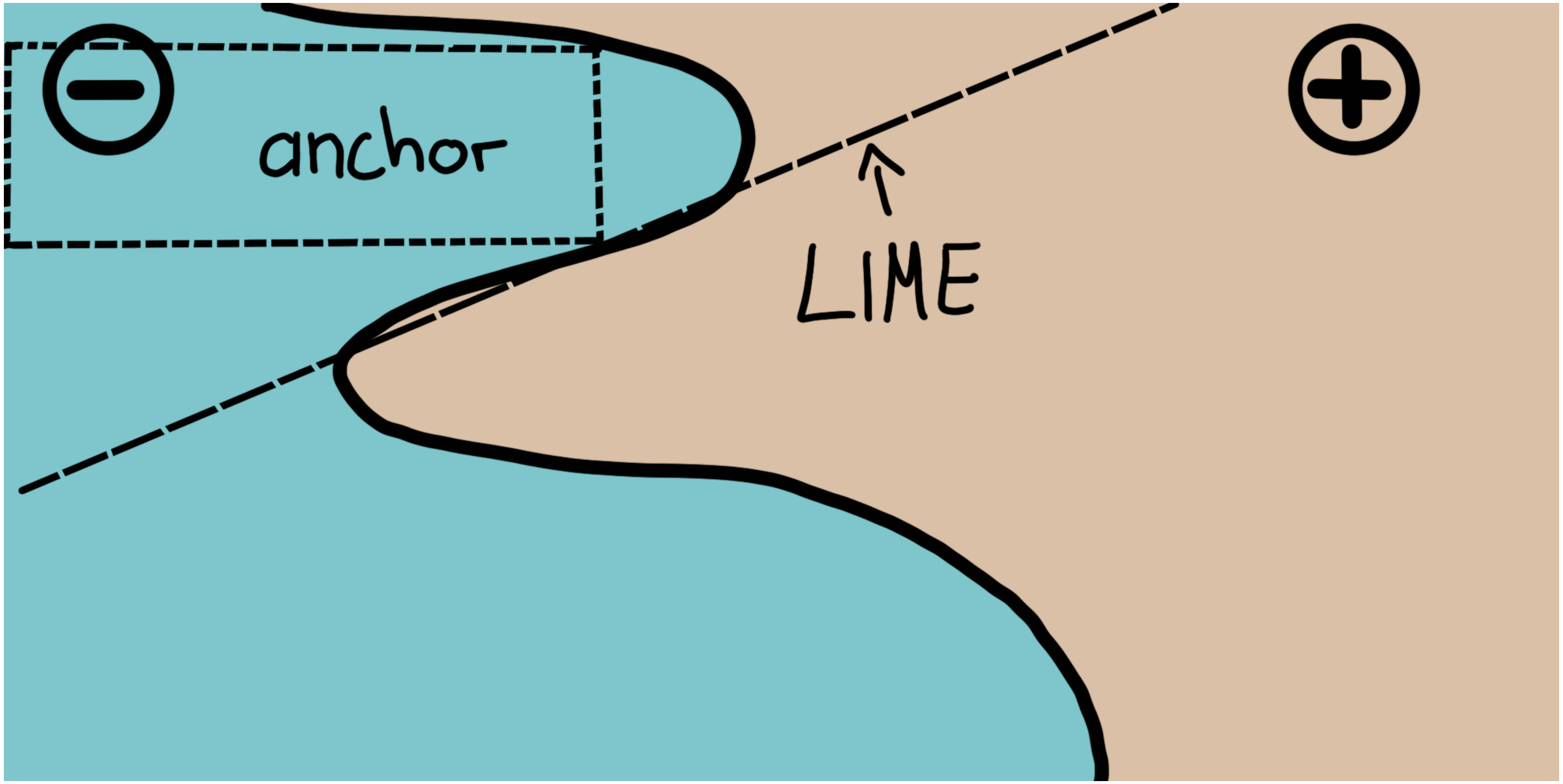
→ Difficult to trust explanations

Scoped Rules (Anchors)

Scoped rules (Anchors)

Introduction: A toy visualization [Ribeiro, 2018]

- Explanations in terms of IF-THEN rules, called anchors
  (in contrast to linear regression and feature weights)
- Each rule has a clear scope (expressing when it holds and when not)
  (in contrast to LIME's problematic neighborhood definition)



Anchors

Example: Predict whether a passenger survived the Titanic disaster

Feature	Value
Age	20
Sex	female
Class	first
Ticket price	300$
More attributes	…
Survived	true

Anchor explanation

IF `SEX = female` AND `Class = first` THEN

PREDICT `Survived = true`

WITH PRECISION 97% AND COVERAGE 15%

(rule covers 15% of instances in perturbation space, 97% accurate for these)

Anchors: Rules and feature predicates

Examples [Ribeiro et al, 2018]

How exactly do rules (conjunctions of feature predicates) look like?

- Depends on data type (e.g., table, image, text, …)
- Depends on use case, …

Example for tabular data (from previous slide):

IF `SEX = female` AND `Class = first` THEN

Example for NLP data

Instance	If	Predict
I want to play(V) ball.	previous word is PARTICLE	play is VERB.
I went to a play(N) yesterday.	previous word is DETERMINER	play is NOUN.
I play(V) ball on Mondays.	previous word is PRONOUN	play is VERB.

Table 1: Anchors for Part-of-Speech tag for the word “play”

Definition of scoped rules (Anchors)

General idea

A is an anchor of $\mathbf{x}$ if

$$\mathbb{E}_{\mathcal{D}_{\mathbf{x}}(\mathbf{z}|A)}\left[1_{\hat{f}(\mathbf{x})=\hat{f}(\mathbf{z})}\right] \geq \tau, A(\mathbf{x}) = 1$$

wherein

- $\mathbf{x}$ represents the instance being explained (e.g., one row in a tabular data set)
- $A(\mathbf{x})$ is a set of feature predicates such that $A(\mathbf{x}) = 1$ when all feature predicates defined by A correspond to $\mathbf{x}$'s feature values
- $\hat{f}$ denotes the classification model to be explained. It can be queried to predict a label for $\mathbf{x}$ and its perturbations $\mathbf{z}$
- $\boldsymbol{\mathcal{D}}_{\mathbf{x}}(\cdot\,|\boldsymbol{A})$ indicates the distribution of neighbors of $\mathbf{x}$, matching A
- $\mathbf{0} \leq \boldsymbol{\tau} \leq \mathbf{1}$ specifies a precision threshold. Only rules that achieve a local fidelity of at least τ are considered a valid result

Informally: A is an anchor of an instance x if A is a set of feature predicates that are all fulfilled for x, and most neighbors $\mathbf{z}$ of x that fulfill A, too, yield the same prediction as x.

Finding anchors

- Problem: Finding exact solution is infeasible
  - Evaluating $1_{\hat{f}(\mathbf{x})=\hat{f}(\mathbf{z})}$ for all $\mathcal{D}_{\mathbf{x}}(\mathbf{z}|A)$ is infeasible in infinite/large input spaces
- Solution: Probabilistic definition
- Probabilistic precision threshold:

$$P(prec(A) \geq \tau) \geq 1 - \delta \quad \text{with} \quad prec(A) = \mathbb{E}_{\mathcal{D}_x(\mathbf{z}|A)}\left[1_{\hat{f}(\mathbf{x})=\hat{f}(\mathbf{z})}\right]$$

- Coverage: an anchor's probability of applying to its neighbors

$$cov(A) = \mathbb{E}_{\mathcal{D}_{(\mathbf{z})}}[A(\mathbf{z})]$$

- Anchor's final definition

$$\boxed{\max_{A \text{ s.t. } P(prec(A) \geq \tau) \geq 1-\delta} \mathrm{cov}(A)}$$

- Goal: find rule that has the highest coverage among all eligible rules (rules that satisfy probabilistic precision threshold)
- **Still a problem:** Number of possible anchors A is exponential in the number of potential feature predicates

  → Efficient methods necessary to find suitable anchor
  (e.g., using multi-armed bandits and beam search, see Ribeiro 2018)

Some remarks

- Rules with more predicates tend to have higher precision
- In particular: If a rule fixes every feature of $\mathbf{x}$, only identical instances are evaluated
- Thus, all neighbors are classified equally and the rule's precision is 1 (if there is no noise in the training data)
- A rule that fixes many features tends to be very specific and only applicable to a few instances

→ There is a tradeoff between precision and coverage

Tabular data example

Bike rental data

Task

- Predict whether the number of bicycles lies ‘above’ or ‘below the trend line

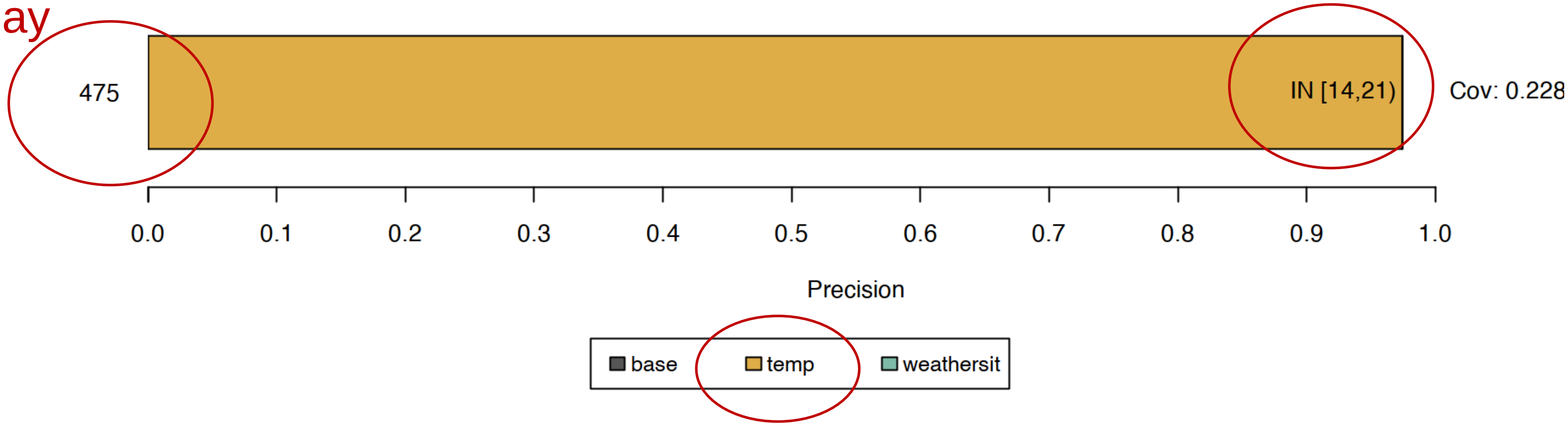
Candidate generation

- Maintain the feature’s values that are subject to the anchors’ predicates
- Replace non-fixed features with values from another randomly sampled instance

→ New instances are similar to the explained one, but have some feature values from other random instances

(Discretized) feature
temperature in interval [14,21) degrees celsius

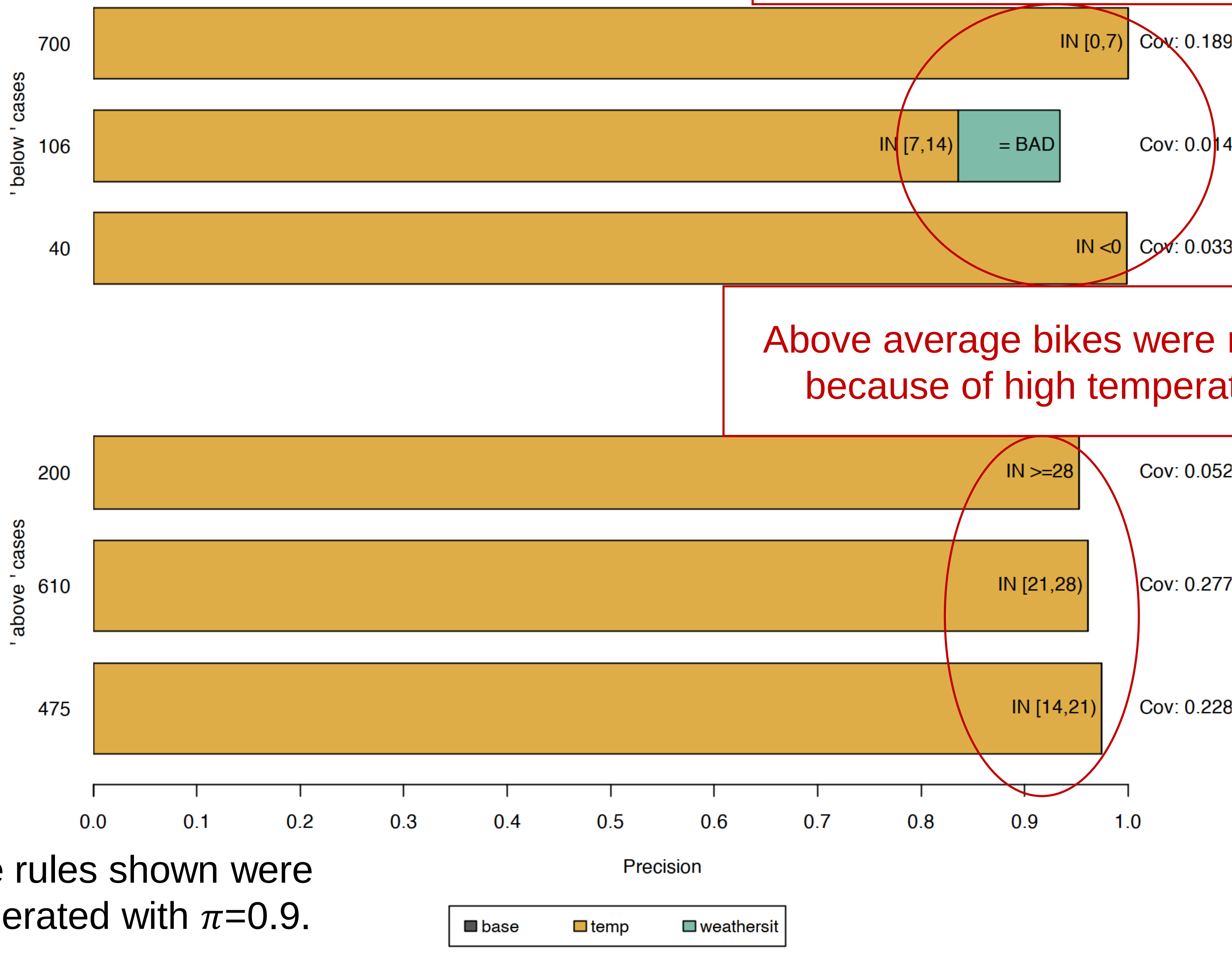
Instance number, day



Tabular data example

Anchors for instances with **simple** explanations

Below average bikes were rented because of low temperature/bad weather



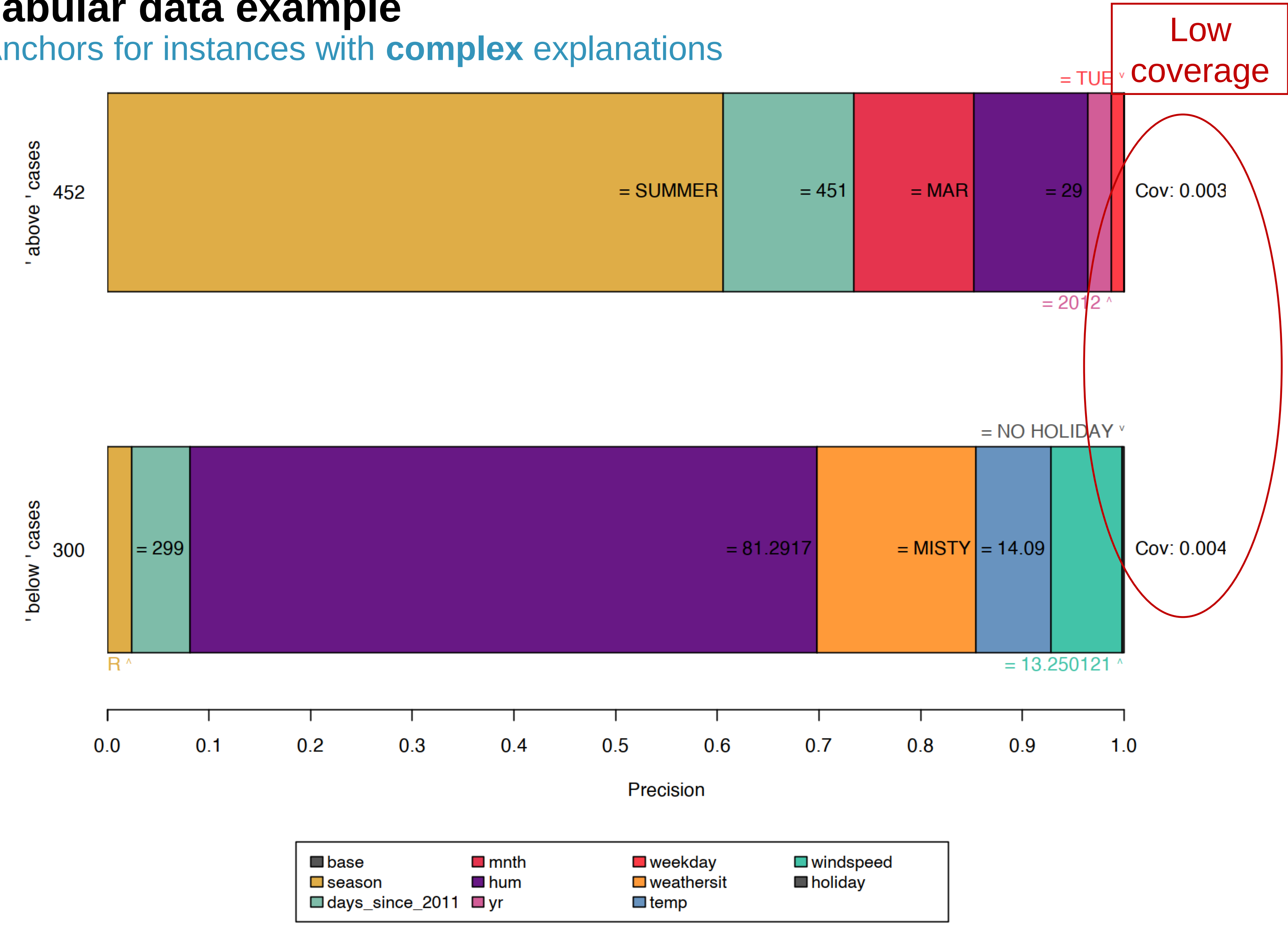
Above average bikes were rented because of high temperature

The rules shown were generated with π=0.9.

Christoph Molnar (https://christophm.github.io/interpretable-ml-book/images/anchors1-1.png), https://creativecommons.org/licenses/by-nc-sa/4.0/

Tabular data example

Anchors for instances with **complex** explanations



Anchors
Advantages

Rules are easy to interpret
- Anchors state a measure of importance (coverage)
- Anchors state a measure of accuracy (precision)
- Works when model predictions are non-linear or complex

Model-agnostic
- Works with every model

Runtime
- Highly efficient
- Can be parallelized with multi-armed bandits (MAB) that support batch sampling (e.g., BatchSAR)

Libraries
- For example: integrated in Alibi library: https://github.com/SeldonIO/alibi

Anchors
Disadvantages

Highly configurable and impactful setup (many hyperparameters that strongly influence the explanations)

- Beam width
- Perturbation functions

Discretization of features and target might be necessary

- For example, binning of similar feature values
- Otherwise, rules are too specific and have low coverage
- Best discretization technique might depend on dataset

Many calls to the black-box model necessary

- Is somewhat mitigated by MAB
- But: anchor’s runtime still depends on model’s runtime

Coverage is undefined in some domains

- Unclear how superpixels in one image compare to superpixels in other images

References

- **Alvarez** Melis, David, and Tommi Jaakkola. "Towards robust interpretability with self-explaining neural networks." *Advances in neural information processing systems* 31 (**2018**).
- **Kaufmann**, Emilie, and Shivaram Kalyanakrishnan. "Information complexity in bandit subset selection." In *Conference on Learning Theory*, pp. 228-251. PMLR, **2013**.
- **Ribeiro**, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" 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. **2016**.
- **Ribeiro**, Marco Tulio, Sameer Singh, and Carlos Guestrin. "Anchors: High-precision model-agnostic explanations." In *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1. **2018**.
- **Slack**, Dylan, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. "Fooling lime and shap: Adversarial attacks on post hoc explanation methods." In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180-186. **2020**.
- **Sutton**, Richard S., and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, **2018**.