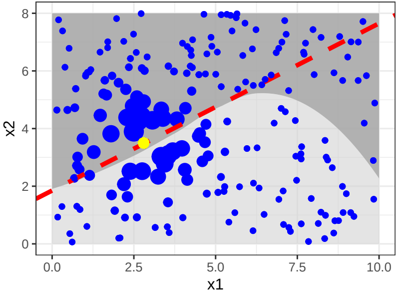
Interpretable Machine Learning

LIME



Learning goals

- Understand motivation for LIME
- Develop a mathematical intuition

LIME

- Local Interpretable Model-agnostic Explanations (LIME) assume that even if a ML model is very complex, the local prediction can be described with a simpler model

LIME

- Local Interpretable Model-agnostic Explanations (LIME) assume that even if a ML model is very complex, the local prediction can be described with a simpler model
- LIME explains **individual** predictions of **any** black-box model by approximating the model **locally** with an interpretable model

LIME

- Local Interpretable Model-agnostic Explanations (LIME) assume that even if a ML model is very complex, the local prediction can be described with a simpler model
- LIME explains **individual** predictions of **any** black-box model by approximating the model **locally** with an interpretable model
- Called local surrogate models $\rightsquigarrow$ often inherently interpretable models such as linear models or classification/regression trees are chosen

LIME

- Local Interpretable Model-agnostic Explanations (LIME) assume that even if a ML model is very complex, the local prediction can be described with a simpler model
- LIME explains **individual** predictions of **any** black-box model by approximating the model **locally** with an interpretable model
- Called local surrogate models $\rightsquigarrow$ often inherently interpretable models such as linear models or classification/regression trees are chosen
- LIME should answer why a ML model predicted $\hat{y}$ for input $\mathbf{x}$

LIME

- Local Interpretable Model-agnostic Explanations (LIME) assume that even if a ML model is very complex, the local prediction can be described with a simpler model
- LIME explains **individual** predictions of **any** black-box model by approximating the model **locally** with an interpretable model
- Called local surrogate models $\rightsquigarrow$ often inherently interpretable models such as linear models or classification/regression trees are chosen
- LIME should answer why a ML model predicted $\hat{y}$ for input $\mathbf{x}$
- LIME is model-agnostic and can handle tabular, image and text data

LIME: CHARACTERISTICS

Definition:
LIME provides a local explanation for a black-box model $\hat{f}$ in form of a model $\hat{g} \in \mathcal{G}$ with $\mathcal{G}$ as the class of potential (interpretable) models

Model g should have two characteristics:

1. **Interpretable**: relation between the input variables and the response are easy to understand
2. **Locally faithful** / **Fidelity**: similar behavior as $\hat{f}$ in the vicinity of the obs. being predicted

Formally, we want to receive a model $\hat{g}$ with **minimal complexity and maximal local-fidelity**

MODEL COMPLEXITY

We can measure the complexity of a model $\hat{g}$ using a complexity measure $J(\hat{g})$

Example: Linear model

- Let $\mathcal{G} = \left\{g : \mathcal{X} \to \mathbb{R} \mid g(\mathbf{x}) = s(\boldsymbol{\theta}^\top \mathbf{x})\right\}$ be the class of linear models
- $s(\cdot)$: identity function for linear regression or logistic sigmoid function for logistic regression

$\rightsquigarrow$ $J(g) = \sum_{j=1}^{p} \mathcal{I}_{\{\theta_j \neq 0\}}$ could be the L_0 loss, i.e., the number of non-zero coefficients

MODEL COMPLEXITY

We can measure the complexity of a model $\hat{g}$ using a complexity measure $J(\hat{g})$

Example: Linear model

- Let $\mathcal{G} = \left\{g : \mathcal{X} \to \mathbb{R} \mid g(\mathbf{x}) = s(\boldsymbol{\theta}^\top \mathbf{x})\right\}$ be the class of linear models
- $s(\cdot)$: identity function for linear regression or logistic sigmoid function for logistic regression

$\rightsquigarrow$ $J(g) = \sum_{j=1}^{p} \mathcal{I}_{\{\theta_j \neq 0\}}$ could be the L_0 loss, i.e., the number of non-zero coefficients

Example: Tree

- Let $\mathcal{G} = \left\{g : \mathcal{X} \to \mathbb{R} \mid g(\mathbf{x}) = \sum_{m=1}^{M} c_m \mathcal{I}_{\{\mathbf{x} \in Q_m\}}\right\}$ be the class of trees
  i.e., the class of additive models (e.g., constant c_m) over the leaf-rectangles Q_m

$\rightsquigarrow$ $J(g)$ could measure the number of terminal/leaf nodes

LOCAL MODEL FIDELITY

- g is locally faithful to $\hat{f}$ w.r.t. $\mathbf{x}$ if for $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^p$ close to $\mathbf{x}$, predictions of $\hat{g}(\mathbf{z})$ are close to $\hat{f}(\mathbf{z})$
- In an optimization task: the closer $\mathbf{z}$ is to $\mathbf{x}$, the closer $\hat{g}(\mathbf{z})$ should be to $\hat{f}(\mathbf{z})$

LOCAL MODEL FIDELITY

- g is locally faithful to $\hat{f}$ w.r.t. $\mathbf{x}$ if for $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^p$ close to $\mathbf{x}$, predictions of $\hat{g}(\mathbf{z})$ are close to $\hat{f}(\mathbf{z})$
- In an optimization task: the closer $\mathbf{z}$ is to $\mathbf{x}$, the closer $\hat{g}(\mathbf{z})$ should be to $\hat{f}(\mathbf{z})$
- Two required measures:
  1. A proximity (similarity) measure $\phi_{\mathbf{x}}(\mathbf{z})$ between $\mathbf{z}$ and $\mathbf{x}$, e.g. the exponential kernel:

     $$\phi_{\mathbf{x}}(\mathbf{z}) = exp(-d(\mathbf{x}, \mathbf{z})^2/\sigma^2)$$

     with σ as the kernel width and d as the Euclidean distance (numeric features) or the Gower distance (mixed features)

LOCAL MODEL FIDELITY

- g is locally faithful to $\hat{f}$ w.r.t. $\mathbf{x}$ if for $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^p$ close to $\mathbf{x}$, predictions of $\hat{g}(\mathbf{z})$ are close to $\hat{f}(\mathbf{z})$
- In an optimization task: the closer $\mathbf{z}$ is to $\mathbf{x}$, the closer $\hat{g}(\mathbf{z})$ should be to $\hat{f}(\mathbf{z})$
- Two required measures:
  1. A proximity (similarity) measure $\phi_{\mathbf{x}}(\mathbf{z})$ between $\mathbf{z}$ and $\mathbf{x}$, e.g. the exponential kernel:

     $$\phi_{\mathbf{x}}(\mathbf{z}) = exp(-d(\mathbf{x}, \mathbf{z})^2/\sigma^2)$$

     with σ as the kernel width and d as the Euclidean distance (numeric features) or the Gower distance (mixed features)
  2. A distance measure or loss function $L(\hat{f}(\mathbf{z}), \hat{g}(\mathbf{z}))$, e.g. the L_2 loss/squared error

     $$L(\hat{f}(\mathbf{z}), \hat{g}(\mathbf{z})) = (\hat{g}(\mathbf{z}) - \hat{f}(\mathbf{z}))^2$$

LOCAL MODEL FIDELITY

- g is locally faithful to $\hat{f}$ w.r.t. $\mathbf{x}$ if for $\mathbf{z} \in \mathcal{Z} \subseteq \mathbb{R}^p$ close to $\mathbf{x}$, predictions of $\hat{g}(\mathbf{z})$ are close to $\hat{f}(\mathbf{z})$
- In an optimization task: the closer $\mathbf{z}$ is to $\mathbf{x}$, the closer $\hat{g}(\mathbf{z})$ should be to $\hat{f}(\mathbf{z})$
- Two required measures:
  1. A proximity (similarity) measure $\phi_{\mathbf{x}}(\mathbf{z})$ between $\mathbf{z}$ and $\mathbf{x}$, e.g. the exponential kernel:

     $$\phi_{\mathbf{x}}(\mathbf{z}) = exp(-d(\mathbf{x}, \mathbf{z})^2/\sigma^2)$$

     with σ as the kernel width and d as the Euclidean distance (numeric features) or the Gower distance (mixed features)
  2. A distance measure or loss function $L(\hat{f}(\mathbf{z}), \hat{g}(\mathbf{z}))$, e.g. the L_2 loss/squared error

     $$L(\hat{f}(\mathbf{z}), \hat{g}(\mathbf{z})) = (\hat{g}(\mathbf{z}) - \hat{f}(\mathbf{z}))^2$$

- Given points $\mathbf{z}$, we can measure local fidelity of g with respect to $\hat{f}$ in terms of a weighted loss

$$L(\hat{f}, g, \phi_{\mathbf{x}}) = \sum_{\mathbf{z} \in \mathcal{Z}} \phi_{\mathbf{x}}(\mathbf{z}) L(\hat{f}(\mathbf{z}), \hat{g}(\mathbf{z}))$$

MINIMIZATION TASK

- Optimization objective of LIME:

$$\arg\min_{g \in \mathcal{G}} L(\hat{f}, \hat{g}, \phi_{\mathbf{x}}) + J(g)$$

- In practice:
  - LIME only optimizes $L(\hat{f}, \hat{g}, \phi_{\mathbf{x}})$ (model-fidelity)
  - Users decide threshold on model complexity $J(g)$ beforehand
- Goal: **model-agnostic** explainer

  $\rightsquigarrow$ optimize $L(\hat{f}, \hat{g}, \phi_{\mathbf{x}})$ without making any assumptions about $\hat{f}$

  $\rightsquigarrow$ learn $\hat{g}$ only approximately

LIME ALGORITHM: OUTLINE

Input:

- Pre-trained model $\hat{f}$
- Observation $\mathbf{x}$ whose prediction $\hat{f}(\mathbf{x})$ we want to explain
- Model class $\mathcal{G}$ for local surrogate (to limit the complexity of the explanation)

LIME ALGORITHM: OUTLINE

Input:

- Pre-trained model $\hat{f}$
- Observation $\mathbf{x}$ whose prediction $\hat{f}(\mathbf{x})$ we want to explain
- Model class $\mathcal{G}$ for local surrogate (to limit the complexity of the explanation)

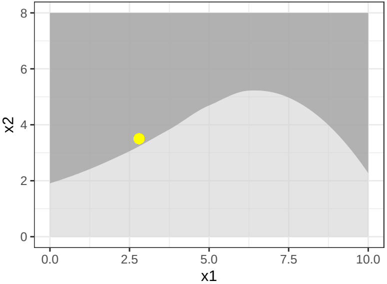
Algorithm:

1. Independently sample new points $\mathbf{z} \in \mathcal{Z}$
2. Retrieve predictions $\hat{f}(\mathbf{z})$ for obtained points $\mathbf{z}$
3. Weight $\mathbf{z} \in \mathcal{Z}$ by their proximity $\phi_{\mathbf{x}}(\mathbf{z})$
4. Train an interpretable surrogate model g on weighted data points $\mathbf{z} \in \mathcal{Z}$
   $\rightsquigarrow$ predictions $\hat{f}(\mathbf{z})$ are the target of this model
5. Return the interpretable model $\hat{g}$ as the explainer

LIME ALGORITHM: EXAMPLE

Illustration of LIME based on a classification task:

- Light/dark gray background: prediction surface of a classifier
- Yellow point: $\mathbf{x}$ to be explained
- $\mathcal{G}$: class of logistic regression models



LIME ALGORITHM: EXAMPLE (STEP 1+2: SAMPLING)

▸ Ribeiro. 2016

Strategies for sampling:

- Uniformly sample new points from the feasible feature range
- Use the training data set with or without perturbations
- Draw samples from the estimated univariate distribution of each feature
- Create an equidistant grid over the supported feature range

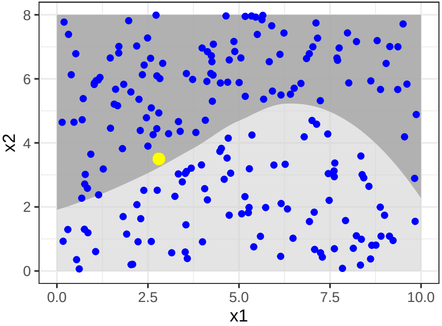


Figure: Uniformly sampled

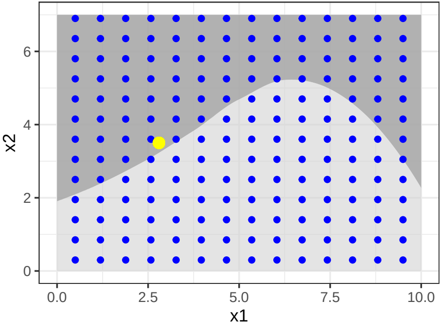
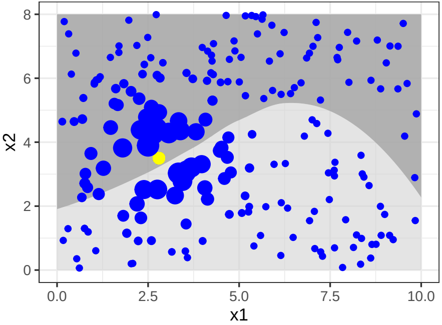


Figure: Equidistant grid

LIME ALGORITHM: EXAMPLE (STEP 3: PROXIMITY) ▸ Ribeiro. 2016

In this example, we use the exponential kernel defined on the Euclidean distance d

$$\phi_{\mathbf{x}}(\mathbf{z}) = exp(-d(\mathbf{x}, \mathbf{z})^2/\sigma^2).$$



LIME ALGORITHM: EXAMPLE (STEP 4: SURROGATE)

▸ Ribeiro. 2016

In our example, we fit a **logistic regression** model (consequently, $L(\hat{f}(\mathbf{z}), \hat{g}(\mathbf{z}))$ is the Bernoulli loss)