Explainable AI: LIME & SHAP — I

Dive deep into the Math, intuition & code

Applied AI Course.com

Why does a model in a specific way for a given input ?

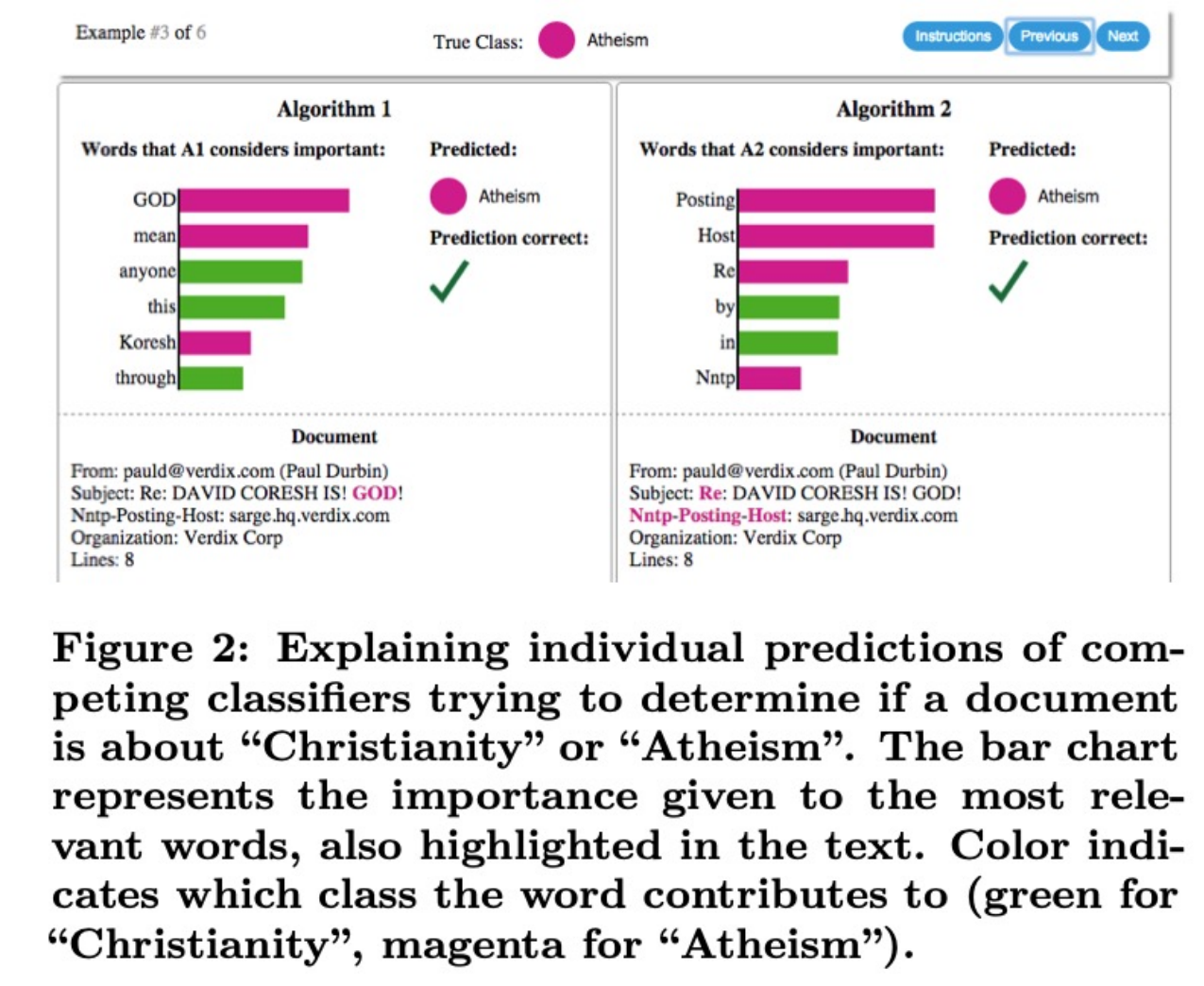


Figure 2: Explaining individual predictions of competing classifiers trying to determine if a document is about "Christianity" or "Atheism". The bar chart represents the importance given to the most relevant words, also highlighted in the text. Color indicates which class the word contributes to (green for "Christianity", magenta for "Atheism").

most images from https://arxiv.org/pdf/1602.04938.pdf

APPLIED ROOTS

Explainability ≈ Feature importance
and/or
Feature contribution

Linear models: magnitude of weights

Tree based models: Feature importance via Entropy or Info Gain

Deep Learning models: Integrated Gradients

https://www.tensorflow.org/tutorials/interpretability/integrated_gradients

APPLIED ROOTS

Can we have model-agnostic explainability

LIME (2016)

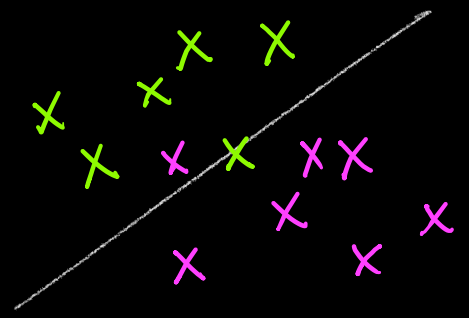
https://arxiv.org/pdf/1602.04938.pdf

Local Interpretable model agnostic explanations

Paper: "Why should I trust you?"

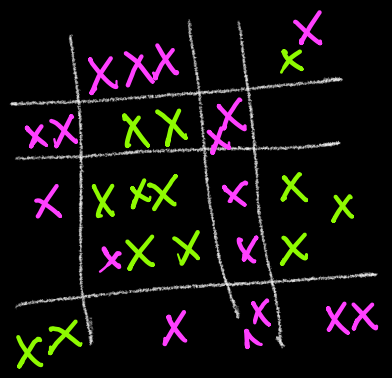
Local vs Global Interpretability

Linear models:



→ Global

Tree based models:

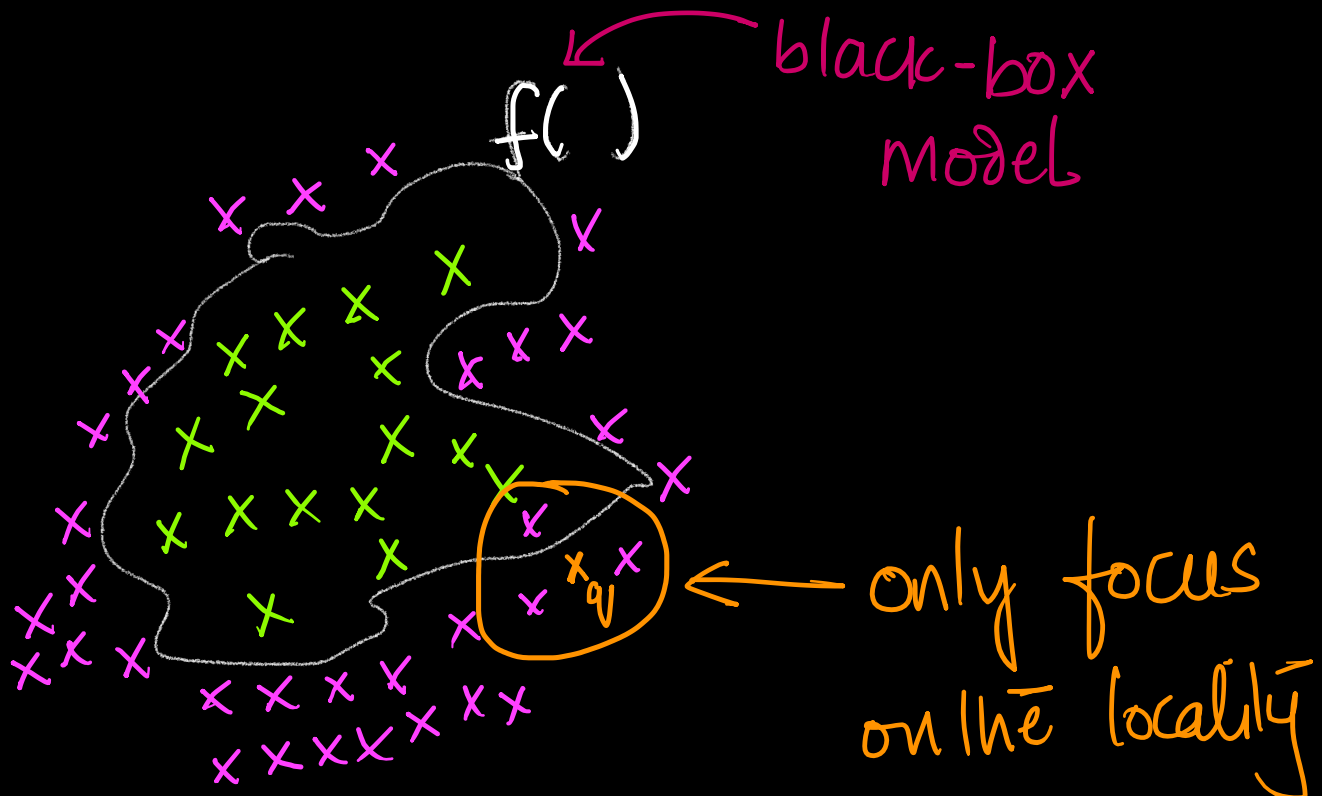


→ Global

Local-interpretability

$f(x) = y \in \mathbb{R}$ → regression

→ probability for classification



why is the model behaving in a specific way in the locality of x_q ?

APPLIED ROOTS

Intuition :

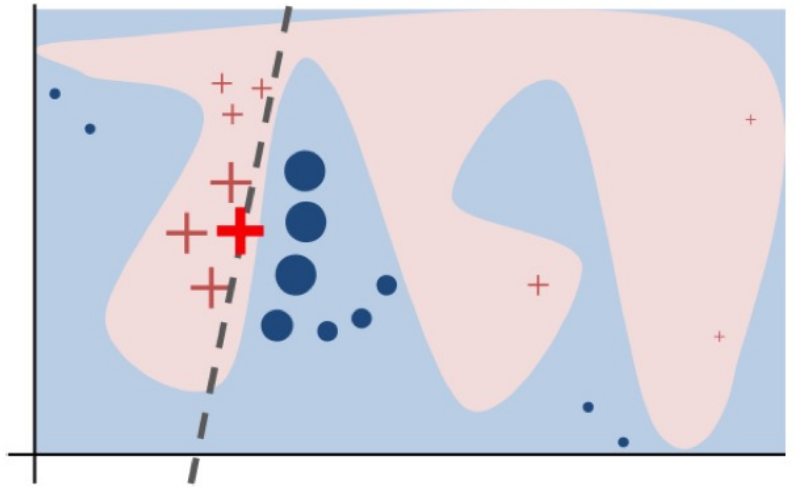


Figure 3: Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.

Math & Algorithm:

$x \in \mathbb{R}^d \longrightarrow f(\,) \longrightarrow y$

$f(\,)$ — BlackBox

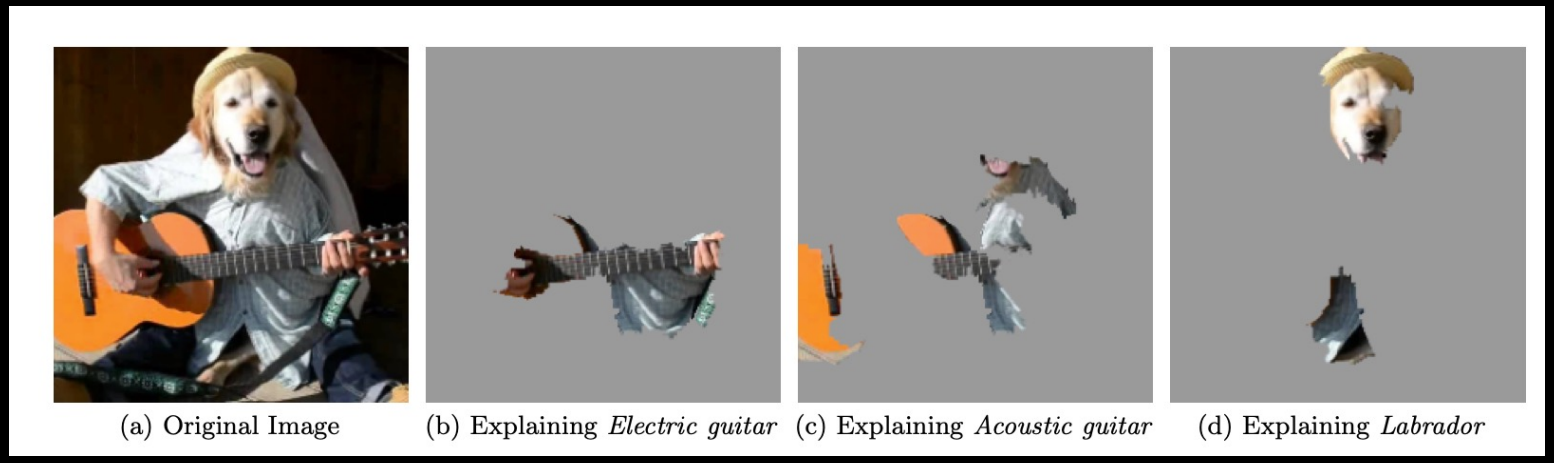
$x' \in \{0,1\}^{d'}$

↳ d'-dim interpretable representation as binary vector (presence/absence)

$x' \in \{0,1\}^{d'}$ → presence | absence of a word (BoW)

↗ Hack: Limit to K-words

→ presence | absence of an image patch or Super-pixel



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

APPLIED ROOTS

Surrogate model : g

model $g \in G$ = class of easily interpretable models (Lr. models, DTs)

$\Omega(g)$ = complexity of g (e.g depth of DT / # of non-zero weights)

$g(x')$ and not $g(x)$

x' ← interpretable representation

$x \in \mathbb{R}^d$, orginal feature vector

APPLIED ROOTS

Proximity Function

$\pi_x(z)$ = proximity measure between x & z

$\pi_x \rightarrow$ locality of x

Local Fidelity:

$\mathcal{L}(f, g, \pi_x)$: How unfaithfully $g()$ approximates $f()$ in the locality defined by π_x

$$\xi(x) = \min_{g \in G} \mathcal{L}(f, g, \Pi_x) + \Omega(g)$$ — 1

G: linear models

$\mathcal{L}$: squared loss as $f(x) \in \mathbb{R}$

$$\Pi_x(z) = \exp\{-\underbrace{D(x,z)}_{\text{dist}}^2 / \sigma^2\} \longrightarrow \text{exponential kernel}$$

APPLIED ROOTS

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in Z} \pi_x(z) \left(f(z) - g(z') \right)^2$$

$$\xi(x) = \min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

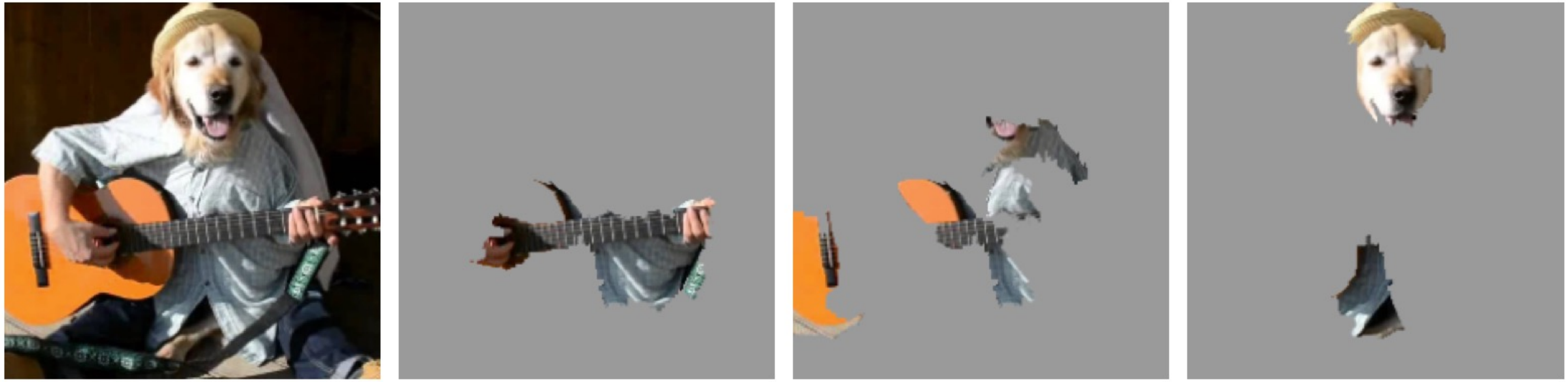
K-Lasso

K-Lasso = L_1-regularized linear regression with only top K-features

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K

$\mathcal{Z} \leftarrow \{\}$
for $i \in \{1, 2, 3, ..., N\}$ **do**
  $z'_i \leftarrow sample\_around(x')$
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
end for
$w \leftarrow$ K-Lasso$(\mathcal{Z}, K)$ ▷ with z'_i as features, $f(z)$ as target
return w



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

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

Advantages

- works on Images, Text & Tabular data
- Lasso or low-depth trees for G → highly interpretable
- x need not be same as x'
  - x → may/maynot be interpretable
  - x' → interpretable feature vector

Drawbacks

- $\Pi_x(z)$: what is neighborhood ?
- Kernel width (σ) → what is the right value (Try various values)
- distance metric : $\Delta(x, z)$ in high-dimensional data
- $x \rightarrow x' \in \{0,1\}^{d'}$ interpretable features may not be straightforward
  e.g: real-valued features → bucketing

— Instability of explanations

https://arxiv.org/pdf/1806.08049.pdf

↳ simulated data: explanations of close points varied

↳ repeat the sampling could give different explanations

⇒ Lack of Robustness

Alternative:

SHAP ⟶ uses Shapley values from Game theory

→ Concrete Mathematical guarantees

→ Time-consuming

→ harder to interpret ⇒ more likely to misinterpret

Nice web-book ↓

https://christophm.github.io/interpretable-ml-book/index.html

[covers breadth well]

for depth → checkout the research papers

Code [very-simple]: https://lime-ml.readthedocs.io/en/latest/

Tabular ↓

https://towardsdatascience.com/lime-how-to-interpret-machine-learning-models-with-python-94b0e7e4432e

Image ↓

https://towardsdatascience.com/interpreting-image-classification-model-with-lime-1e7064a2f2e5

Text ↓

https://towardsdatascience.com/what-makes-your-question-insincere-in-quora-26ee7658b010

APPLIED ROOTS

Next - Session : SHAP

Q&A

APPLIED ROOTS