Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods

Authors: Dylan Slack*, Sophie Hilgard*, Emily Jia, Sameer Singh, Himabindu Lakkaraju

Presented by: Chelsea (Zixi) Chen, Xin Tang, Prayaag Venkat

Motivation

- ML has been applied for **critical** decision making
    - Healthcare
    - Criminal justice
    - Finance
- The decision makers must clearly **understand the model behavior** to
    - Diagnose the error and potential biases
    - Decide when and how much these ML models should be trusted

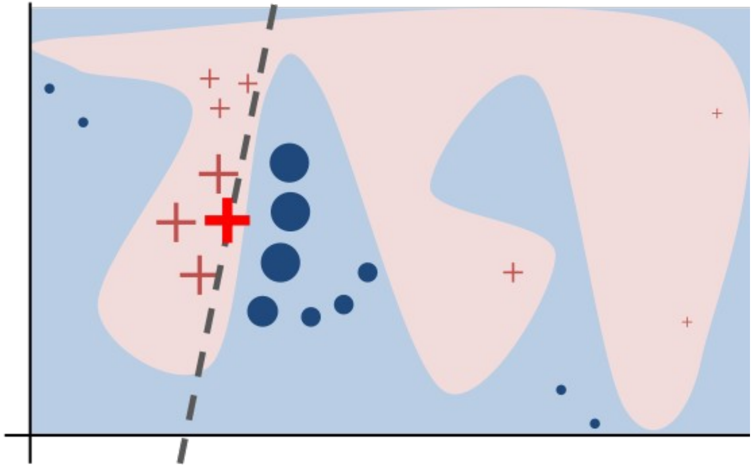
Motivation

- Trade-off between interpretability and accuracy
  - **Simple** models can be easily **interpreted** (e.g., linear regression)
  - **Complex** but also black-box model has much **better performance** (e.g., deep neural network)
- Can a ML method be both **interpertable** and **accurate**?
- ***Post hoc*** explanation can seemingly solve this problem:
  - First build **complex and accurate** ML models for good performance
  - Then use post hoc explanation for model **interpretation**
- The question is:
  - How robust and reliable is the *post hoc* explanation methods?

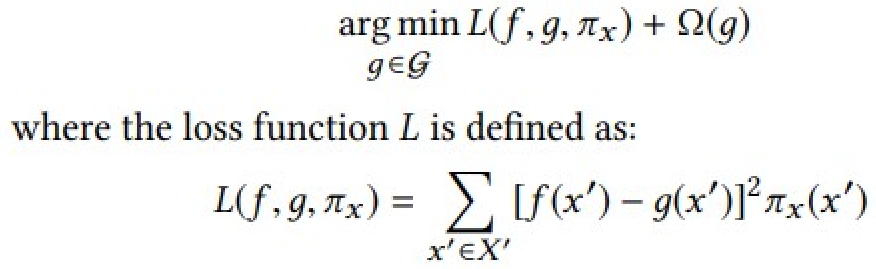
Contribution: A framework to ‘fool’ the post hoc explanation method

- A novel framework that can effectively **mask the discriminatory biases** of any black box classifier
    - Fooling the **perturbation based** post hoc explanation method
    - LIME and SHAP
- Allowing an adversarial entity to control and generate an arbitrary desired explanation
- Demonstration using real-world datasets with extremely biased classifier
- Existing post hoc explanation techniques are **NOT** sufficiently robust for ascertaining discriminatory behavior of classifiers in sensitive applications

Perturbation-based post hoc explanation method



Preliminaries & Background



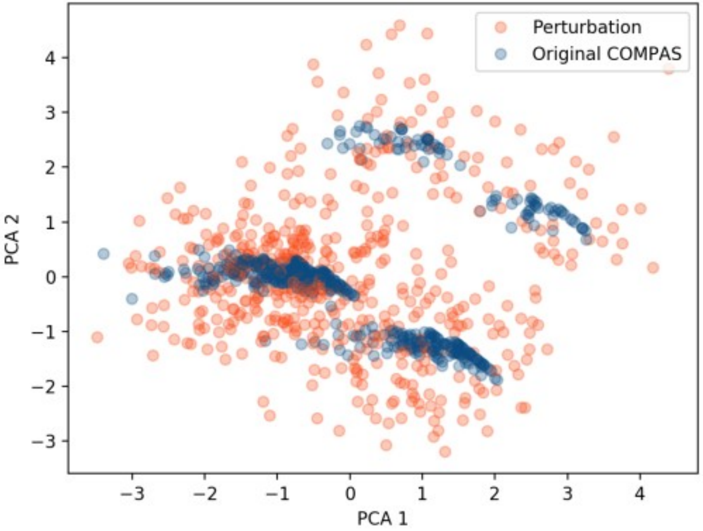
$$\underset{g \in G}{\arg\min}\, L(f, g, \pi_x) + \Omega(g)$$

where the loss function L is defined as:

$$L(f, g, \pi_x) = \sum_{x' \in X'} [f(x') - g(x')]^2 \pi_x(x')$$

- f is the original classifier and x is the datapoint we want to explain
- g is the explanation we want to learn, **Ω**(g) is the “complexity” of g
- π is the proximity measure
- X’ is a **synthetic dataset**, consisting of perturbations of x

Intuition



● Perturbation
● Original COMPAS
PCA 2
4
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
4
PCA 1

Approach: Set-up

Adversary would like to deploy a biased classifier **f**!

- Background: the biased model **f** uses sensitive attributes to make critical decisions
- Requirement: give access of black-box models to customers and regulators who use post-hoc explanations
- Goal: hide bias of the classifier **f**

Approach: Set-up

What do we need?

- Input: dataset sampled from real-world distribution
- Target Product: an adversarial classifier **e**
  - **f** is the biased model to be explained, while **ψ** is an unbiased model

$$e(x) = \begin{cases} f(x), & \text{if } x \in \mathcal{X}_{dist} \\ \psi(x), & \text{otherwise} \end{cases}$$

Approach: OOD Detection

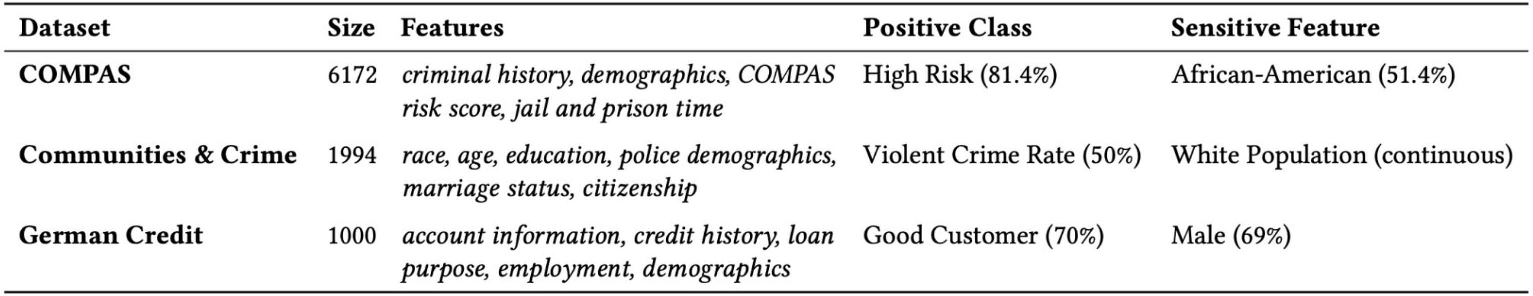
Which of the inputs belong to the real-world distribution?

- Build another classifier for OOD detection
  - Assign label “False” (not OOD) to all instance in the dataset **X**
  - Perturb all instances in **X** and assign them label “True” (OOD)
    - Exceptions: instances too close to observations from **X**
  - Combine data and train OOD detection classifier

$$e(x) = \begin{cases} f(x), & \text{if } x \in X_{dist} \\ \psi(x), & \text{otherwise} \end{cases}$$

Experiment: Set-up

90% training & 10% test

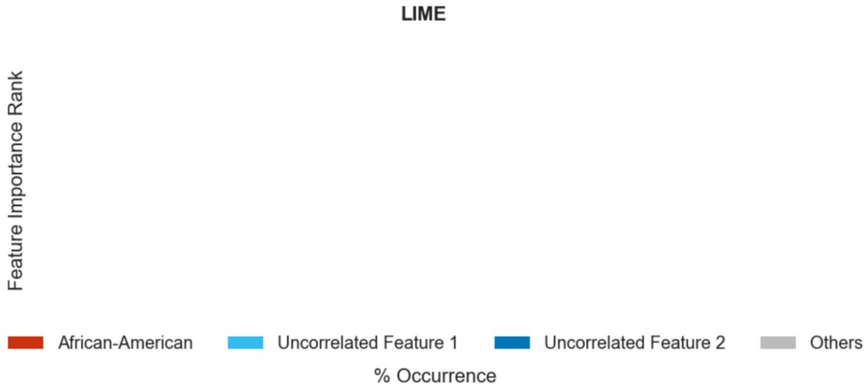


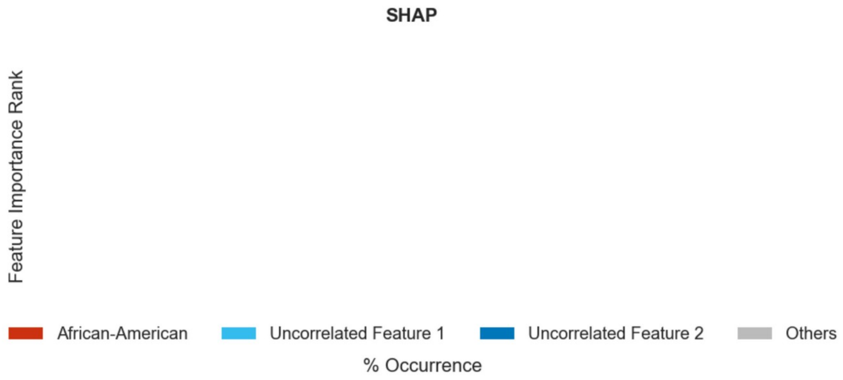
Dataset	Size	Features	Positive Class	Sensitive Feature
COMPAS	6172	*criminal history, demographics, COMPAS risk score, jail and prison time*	High Risk (81.4%)	African-American (51.4%)
Communities & Crime	1994	*race, age, education, police demographics, marriage status, citizenship*	Violent Crime Rate (50%)	White Population (continuous)
German Credit	1000	*account information, credit history, loan purpose, employment, demographics*	Good Customer (70%)	Male (69%)

Biased classifier **f** makes predictions purely based on sensitive attributes (race, gender)

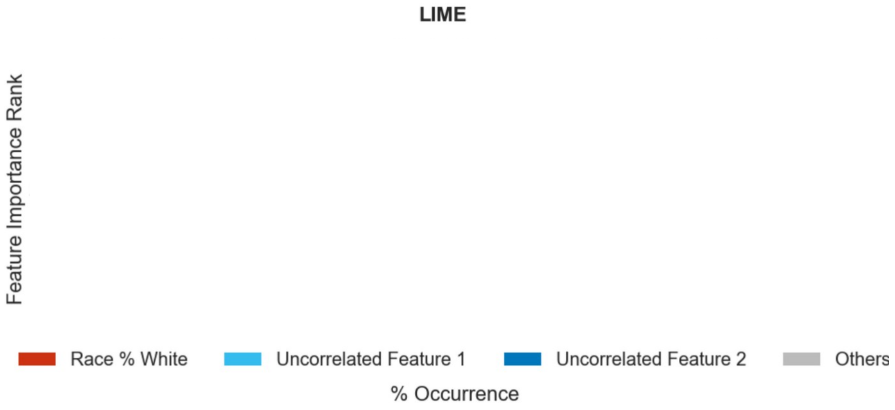
Unbiased classifier **ψ** uses only features uncorrelated with the sensitive attributes

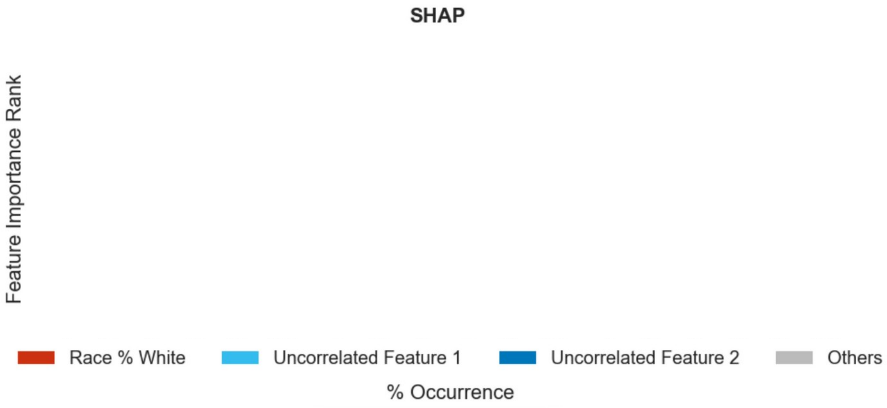
Experiment: Results - COMPAS



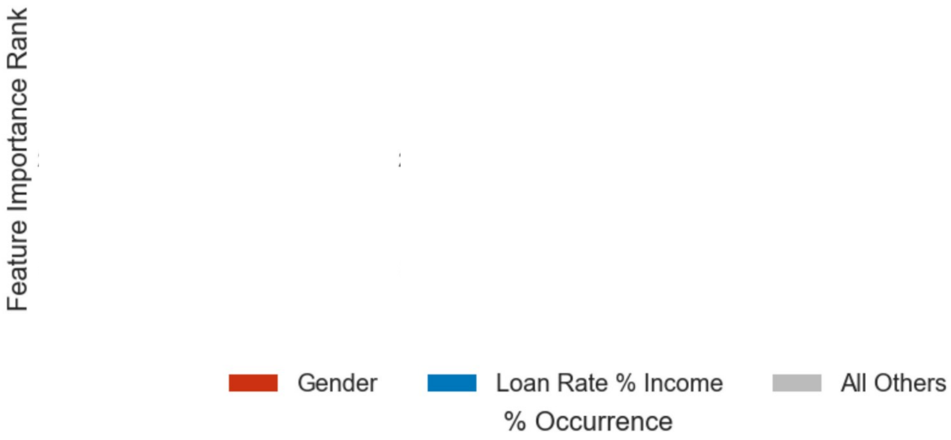


Experiment: Results - Communities and Crime





Experiment: Results - German credit



Takeaway from experiments

1. Accuracy of the OOD classifier -> success of the adversarial attack
2. Adversarial classifiers to LIME are ineffective against SHAP explanations
    a. Any sufficiently accurate OOD classifier is sufficient to fool LIME, while fooling SHAP requires more accurate classifiers
3. SHAP less successful when using two features <- local accuracy property
    a. Distribute attributions among several features

$$e(x) = \begin{cases} f(x), & \text{if } \boxed{x \in \mathcal{X}_{dist}} \\ \psi(x), & \text{otherwise} \end{cases}$$

Conclusions

- Main contribution: A framework for converting any black-box classifier into a *scaffolded* classifier that fools perturbation-based post-hoc explanation techniques like LIME and SHAP
- Effectiveness of this framework demonstrated on sensitive real-world data (criminal justice and credit scoring)
- Perturbation-based post-hoc explanation techniques are not sufficient to test whether classifiers discriminate based on sensitive attributes

Related Works

- Issues with post-hoc explanations:
  - [Doshi-Velez and Kim] identify explainability of predictions as a potentially useful feature of interpretable models.
  - [Lipton] and [Rudin] argues post-hoc explanations can be misleading and are not trustworthy for sensitive applications.
  - [Ghorbani et al.] and [Mittelstadt et al.] identified further weaknesses of post-hoc explanations.
- Adversarial explanations
  - [Dombrowski et al.] and [Heo et al.] show how to change saliency maps in arbitrary ways by imperceptibly changing inputs.

Q&A

- Are the experimental results sufficient to justify the conclusions?
  - In particular, how can we explain the discrepancy in results for LIME vs. SHAP?
- What about fooling other classes of post-hoc explanation methods?
  - Past work: gradient-based methods
- Alternatively, can one design post-hoc explanations that are *adversarially robust*?