DUKE
COMPUTER SCIENCE

CompSci 590.01
Spring 2023

Causal Inference in Data Analysis
with Applications to
Fairness and Explanations

Lecture 9
Attack on Post-hoc ML explanation methods

Sudeepa Roy

Announcements: March 31

- Sign up for project presentations: Last two classes
- **Final report due on LDOC: April 19**
- Your project report will be evaluated after LDOC – does not matter when you present – even if your project is not complete then
- Reminder: You can skip two paper reviews + your own (so 3)
- No more homework will be posted (all homework grades on paper reviews), but may post some self-study material
- Next week: Guest lectures by Harsh Parikh on causal inference: continuous covariates and validating causal inference methods – **no paper review next week**

Reading

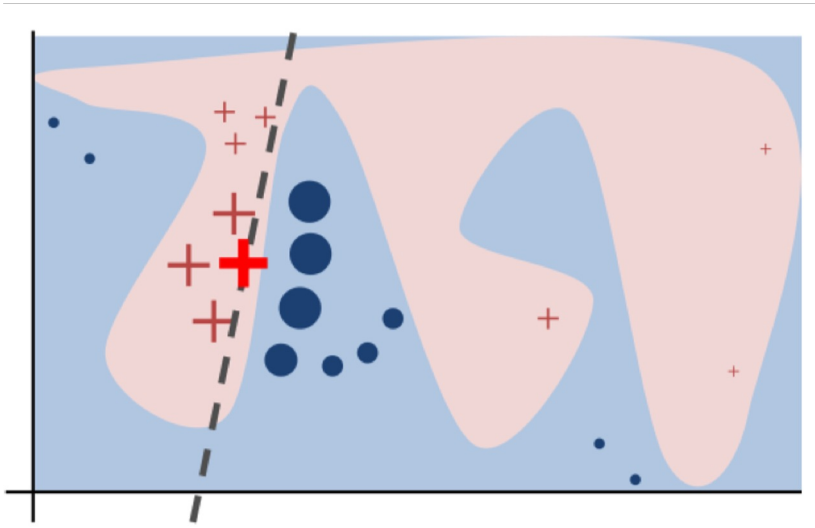
Slack, Dylan, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. "Fooling lime and shap: Adversarial attacks on post hoc explanation methods." In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society,* pp. 180-186 (2020).

Acknowledgement (big thanks!):

The slides closely follow online youtube talks by Hima Lakkaraju and Dylan Slack

Key idea in Post-hoc explanation methods: LIME / SHAP

- For a prediction of a given individual, we learn **a explainable model** (e.g. linear models, decision trees) that uses **interpretable representations** to generate the prediction which **mimic** the **local behaviors** of the original black-box model (in terms of prediction results), while **controlling the complexity** of the learned explainable model.
- Complex models have complex decision surfaces and are harder to explain globally, simpler decision boundaries in local neighborhoods



Perturbation: Among all features set to 1 for a given data point, randomly sample a subset of features and set their values to 0

Class labels of data points will be the predictions of black box

Generate a neighborhood by perturbing each data point and construct a linear model

Vulnerabilities of LIME/SHAP

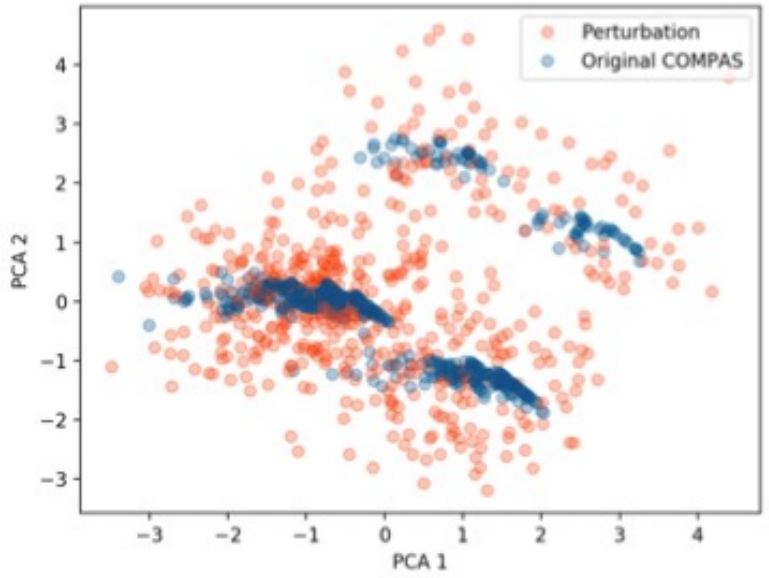


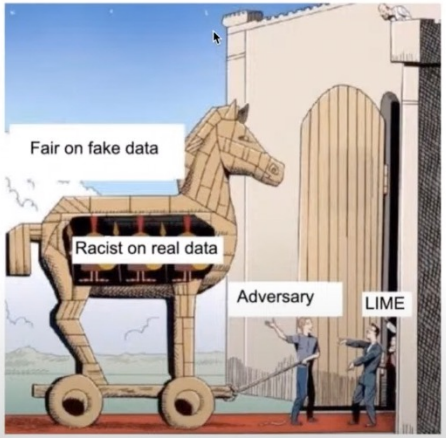
Figure 1: PCA applied to the COMPAS dataset (blue) as well as its LIME style perturbations (red). Even in this low-dimensional space, we can see that data points generated via perturbations are distributed very differently from instances in the COMPAS data. In this paper, we exploit this difference to craft adversarial classifiers.

- Several of these data points are out of distributions even for low dimension
- Adversaries can exploit this and build a classifier that is biased on in-sample data points and unbiased on OOD samples

Adversarial Classifiers

Building Adversarial Classifiers

- Intuition:



(Image credit: Willie Boag)

Volkswagen: The scandal explained

10 December 2015

By Russell Hotten
Business reporter, BBC News

What is Volkswagen accused of?

It's been dubbed the "diesel dupe". In September, **the Environmental Protection Agency (EPA) found** that many VW cars being sold in America had a "defeat device" - or software - in diesel engines that could detect when they were being tested, changing the performance accordingly to improve results. The German car giant has since admitted cheating emissions tests in the US.

Screenshot and example from a youtube talk by Dylan Slack
https://www.bbc.com/news/business-34324772

Building Adversarial Classifiers

- Adversary wants to deploy a biased classifier f in real world (e.g., use gender/race to make a decision)
- Adversary must provide black box access to users who may use post-hoc explanations
- Goal of the adversary is to fool the post-hoc explanation techniques and hide the underlying biases of f

Building Adversarial Classifiers

- Input: Adversary gives biased classifier f, input database X sampled from input distribution X-dist
- Output: “scaffolded” or adversarial classifier e which behaves exactly like f when making predictions on instances sampled from X-dist but will not reveal underlying biases of f when proved with perturbation-based explanations
- u is the unbiased classifier (does not depend on sensitive attributes)
- e(X) = f(X) if x in X-dist
  = u(X) otherwise

How to detect if a sample is OOD?

- Goal decide if _OOD(x) = true or not
- Construct a new dataset X_p from X:
  - Perturb each data point x in X to generate x_p (like LIME)
- For all x in X: assign class label False (not OOD)
- For all x_p in X_p: assign class label True (OOD) unless they are already in X (or very close to some data point in X)
- Then train an off-the-shelf classifier on the combined dataset $X \cup X_p$ and their labels

Experiments: Datasets & LIME/SHAP

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%)

Table 1: Summary of Datasets

Default LIME Tabular implementation without discretization

Default Kernel SHAP implementation with kmeans with 10 clusters as the background distribution.

Experiments: Classifiers

- Biased classifier f: f is perfectly discriminatory and purely uses a sensitive feature to make its prediction
- Perturbations:
  - LIME: for every data point add noise sampled from N(0, 1)
  - SHAP: Randomly choose a subset of features and mark their values as missing, and replace them with a value from the background distribution
- OOD Classifier (as before): Use sklearn-Random Forest 100 trees
- Unbiased classifier u: use synthetic uncorrelated features with zero correlation sensitive attribute
  - One uncorrelated feature: if 1 predict 1, else 0
  - Two uncorrelated features: based on XOR of this
  - Also tested with real Loan Rate % Income feature on German Credit

Effectiveness of Attacks: COMPAS

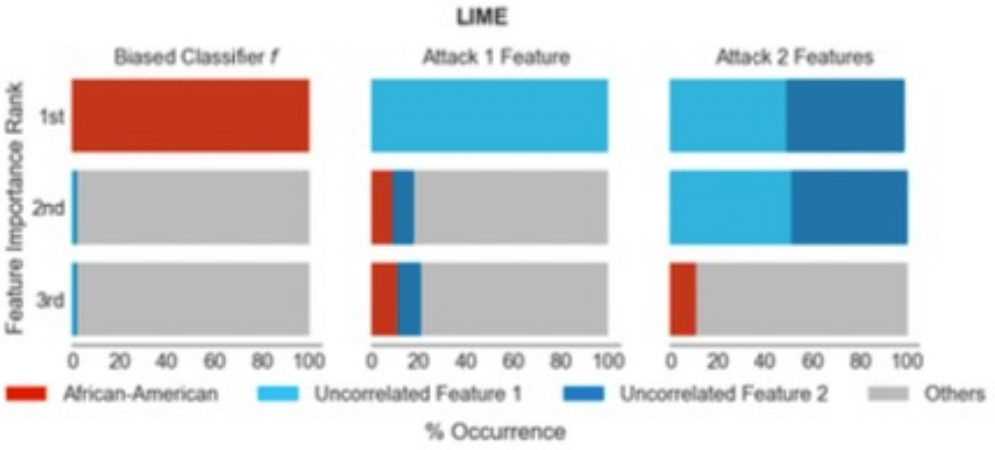
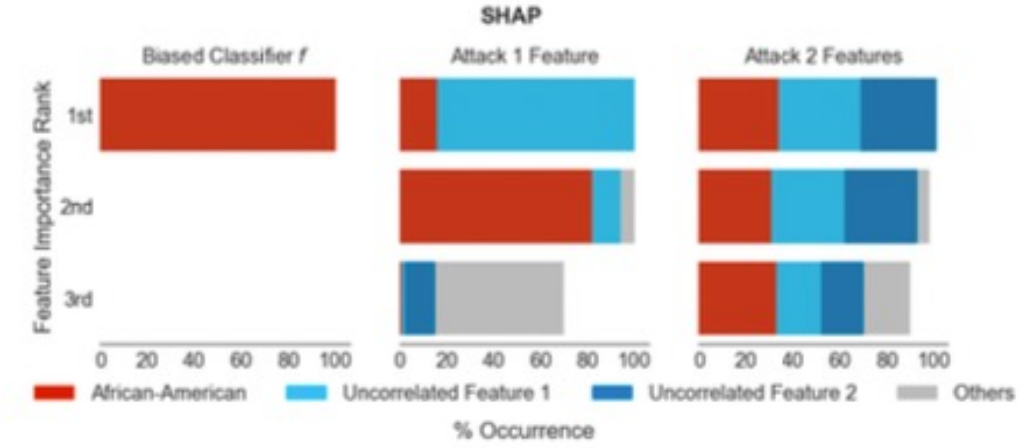


Figure 2: COMPAS: % of data points for which each feature (color coded) shows up in top 3 (according to LIME and SHAP's ranking of feature importances) for the biased classifier f (left), our adversarial classifier where ψ uses only one uncorrelated feature to make predictions (middle), and our adversarial classifier where ψ uses two uncorrelated features to make predictions (right).



- Foold LIME and SHAP
- Race is not among the top features (LIME) or effect is reduced (SHAP)

Effectiveness of Attacks: Community & Crime

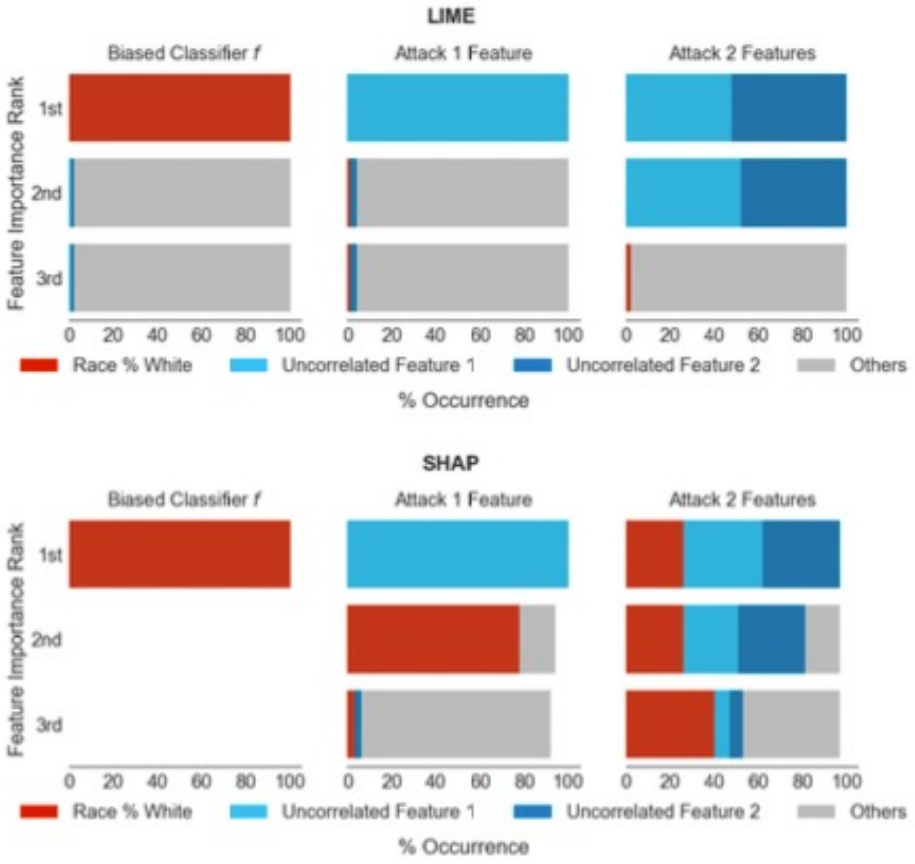


Figure 3: Communities and Crime: Similar to Fig 2; *Race % White* is the sensitive feature here.

Effectiveness of Attacks: German Credit

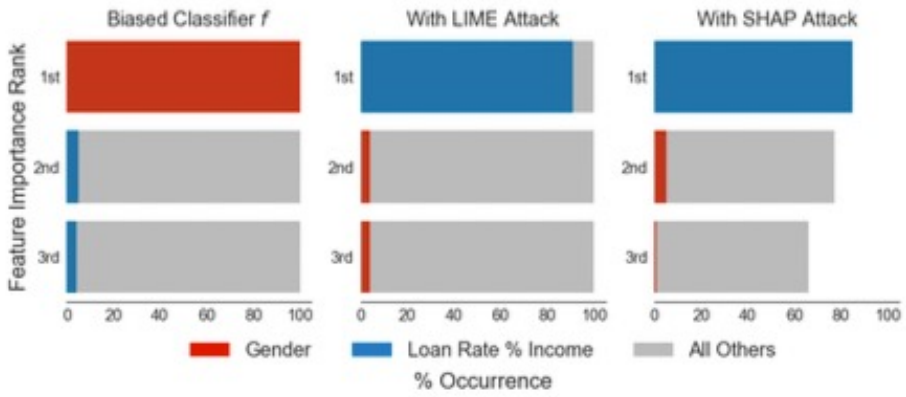


Figure 4: German credit: Similar to Fig 2 and 3, but unbiased classifier ψ uses an existing feature (*Loan Rate % Income*) to make predictions, and *Gender* is the sensitive feature. Feature importances for the biased classifier f shown in the figure (left) are generated using LIME; SHAP also produces similar feature importance values.

Explanations shift to loan rate percentage (not-sensitive)

Follow up work

- ““How do I fool you?": Manipulating User Trust via Misleading Black Box Explanations”
- Lakkaraju & Bastani, AIES 2020

- How misleading explanations influence user trust in black box models
- Theoretical framework & User Study