>>> Understanding VADER
>>> Using LIME to interpret black-box models

Name: Arturo Soberón
Date: June 6, 2022

>>> Table of Contents

>>> Black-Box Models

High-complexity Machine Learning models are capable of making really accurate predictions. Sadly, this gain in accuracy is often accompanied by a loss in interpretability.

A *black-box* model is a software that receives an input, returns an output and everything that happens in between is a mystery.

Local Interpretable Model-agnostic Explanations (LIME) makes many modifications to a given observation (x_0) and fits an interpretable model on these new data points to explain the decision making process of a black-box model.

>>> VADER

VADER is a Natural Language Processing (NLP) model that receives a text and, among other things, replies with a score that measures its negativity.[1]

$$x_o \rightarrow f(\cdot) \rightarrow y_0$$

As with most NLP models, VADER serves requests as a black-box model.

[1] Hutto et al., 2014

>>> Example

For example, the text:

> *Dear Dice,*
> *This game sucks. The specialists are trash. This trailer was a lie. I want my money back.*

receives a negative score of 0.112.

What drove this black-box model to make this prediction?

* Does the word *Dear* reduce the negative score?
* Do the words *sucks* and *trash* increase the negative score?

>>> General procedure

LIME uses simple *surrogate* models to explain *individual* predictions of a black-box Machine Learning model.

* LIME does not train a simple global model

Given a data point x_0 and a predicted value $f(x_0)$, LIME makes many slight variations to x_0 to test what happens to the predictions made by the complex model.[2]

$$\big(x_0, f(x_0)\big) \rightarrow \{\big(x_0^1, f(x_0^1)\big), ..., \big(x_0^n, f(x_0^n)\big)\}$$

LIME then fits an interpretable model to the data such that

$$\min_g \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

[2] Ribeiro et al., 2016

>>> Pseudo-code

1. Select data point of interest x_0
2. Feed x_0 to $f(\cdot)$ to obtain y_0
3. Make multiple modifications around x_0
4. Get the predictions from the perturbed data set
5. Fit a simple model to this data set
6. Explain $f(x_0)$ by interpreting the local model

>>> LIME for tabular data

1. Generate the perturbed data set by sampling new observations from a multivariate distribution centered around x_0.
2. Fit a local regression with an exponential kernel.

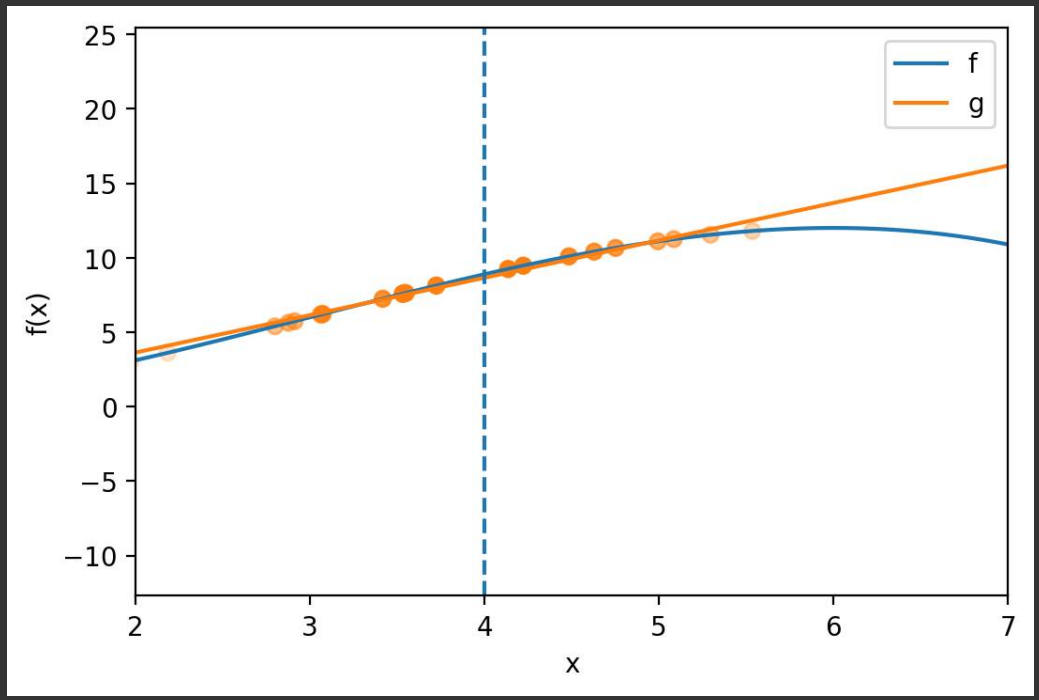

Figure: Local regression with exponential kernel

>>> LIME for image recognition

1. Left: Original image (high predicted probability of *wolf*)
2. Middle: No snow (low predicted probability of *wolf*)
3. Middle: No trees (high predicted probability of *wolf*)

Figure: Removal of interpretable components

$$(x_0, f(x_0)) \rightarrow \mathcal{D}_0 = \{(x_0^1, f(x_0^1)), ..., (x_0^n, f(x_0^n))\}$$

* For any $i \in \{1, 2, ..., n\}$, x_0^i is a vector of size $1 \times m$, where m is the number of words in x_0.
* Each feature $w_j^i \in x_0^i$ is a dichotomous variable that represent the j-th word in x_0 and indicates if it is switched on or off in the i-th perturbation.
* In its most basic form, the weight corresponding to each instance is given by the share of words included in x_0^i.

$$\omega_i = \frac{1}{m} \sum_{j=1}^{m} w_j^i$$

>>> Battlefield 2042

I mined a few thousand comments from Battlefield 2042's launch trailer. I chose to study Battlefield 2042 because it was a highly-anticipated yet poorly-received game.

> *One of the biggest games of the year on one of the most popular digital stores in the world on one of the biggest gaming platforms in the world [...] isn't able to keep up with Farming Simulator 22.*[3]

Due to the game's popularity and poor reception by the general public, the comment section is large and overwhelmingly negative, making it a good case study for this project.

---

[3] Kotaku, 2021

>>> Example

Given x_0:

How tf they managed to screw this up?

We create the following perturbed data set:

How	tf	they	managed	to	screw	this	up?	weight	score^0	score^1
1	0	1	1	1	1	1	0	0.750	0.167	0.167
1	1	1	1	1	0	1	1	0.875	0.167	0.000
0	1	1	1	1	1	1	1	0.875	0.167	0.149
1	0	1	1	1	1	1	1	0.875	0.167	0.149
1	0	1	1	1	1	0	1	0.750	0.167	0.167

>>> Example (continued)

By fitting

$$\min_{\beta} \sum_{i=1}^{n} (y_i - \sum_{p=1}^{8} \beta_p x_{ip})^2 + 0.1 \sum_{p=1}^{8} |\beta_p|$$

we get the following explanation at x_0:

Word	Coef
How	0.43
tf	0.43
they	0.00
managed	0.11
to	0.00
screw	1.62
this	0.00
up?	0.00

>>> Non-negative comment with high score

I still miss bad company 2

Word	Importance
bad	0.37
miss	0.23

The explanation suggests that VADER is unable to understand that *bad company* is a video game and not a negative sentiment.

>>> Non-negative comment with low score

Honestly you all gotta chill. I'm going to say it. Battlefield 2042 is a good game.

Word	Importance
battlefield	0.13

Given that the comments come from a *Battlefield* video, many of them contain the word *battlefield* and will therefore receive high negative scores.

>>> Negative comment with high score

This game is a disgrace

Word	Importance
disgrace	0.47

The sole factor that drove this prediction is the word *disgrace*.

>>> Negative comment with low score

The game has been delayed from October to November. Thanks again, Covid. What would be of our lives without you?

Word	Importance
battlefield	0.13

This explanation suggests that VADER is unable to detect sarcasm.