Noah Podgurski, Yusef Mustafa, Sergio Montes Correa

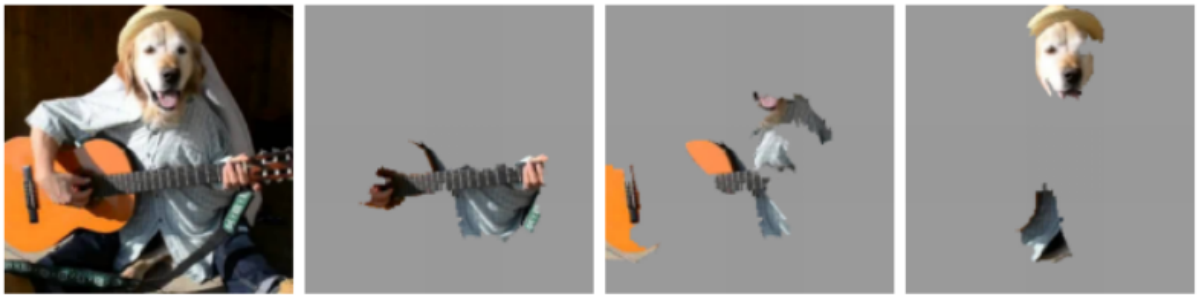
CS301 Project

For many years, machine learning models have been able to make accurate predictions based on several different classification methods. To most people, however, complex machine learning models are difficult to understand which leads to a distrust in the predictions. LIME attempts to remove any trace of doubt by providing in-depth explanations as to how a prediction is generated.

LIME stands for Local Interpretable Model-Agnostic Explanations. Local, referring to the classifier and data we are using. Interpretable, meaning, that the explanations are presented in an easy-to-understand manner. Model-agnostic, meaning that LIME can be used on any type of model. Lastly, Explanations is quite obvious.

LIME is an algorithm that explains the predictions of any machine learning model so that the user can trust the predictions. In other words, it is an easier way to understand the reasoning behind machine learning models classifications. For example, say a doctor needs help diagnosing a patient. Rather than a model giving a classification saying the patient has influenza, which could lead to some doubt, the model gives a classification saying which symptoms the patient is experiencing, and whether or not they support the claim that the patient has influenza. The doctor is then given the final say to determine the diagnosis.

How is LIME utilized? It essentially breaks problems down into smaller problems.



(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 network, highlighting positive pixels. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

Using the example above, LIME wants to classify the image as three separate things. If the three grey pictures were not shown, it might say Electric Guitar (p=.32), Acoustic Guitar (p=.24) and Labrador (p=.21). Without the image being split up into smaller sections with only the matching pixels being shown the user might become confused as to why it is detecting electric guitar. Of course, this is an obvious example, but one could see how this could make a difference with much bigger data. LIME is incredibly useful and incredibly easy to use. Simply run "pip install lime" to be able to import it into your python code.

To get started with LIME, there is a notebook which has a very basic example of explaining random forest classifiers using a LIME library. The link to the notebook is Lime - basic usage, two class case (marcotcr.github.io). First, the lime package is installed by using *pip install lime* command; the package is from PyPI. In the project, the *lime_text* is imported from the lime library, as well as *LimeTextExplainer*. *LimeTextExplainer* is passed an array of class names and the result is stored in a variable, which in the example is called *explainer*. Next, the *explainer* variable is used to call the *explain_instance* function which takes three

parameters: the document to be explained, the probability of the target being 0 or 1 (*predict_proba* is used to get this value*)*, and the number of features to consider for the classification. The result is stored in another variable named *exp*. The LIME explanations can then be printed by accessing the *exp* variable and calling *as_list()* function from it. Other useful functions that can be accessed from *exp* include saving to file, *save_to_file()*, and *show_in_notebook()*. Using the show in notebook function will return classes with prediction probabilities containing graphics. Passing *text=true* to *show_in_notebook()* is also extremely helpful because it highlights words in the document/text that affect the prediction probabilities. There are more advanced examples of projects using LIME in the following repository: GitHub - marcotcr/lime: Lime: Explaining the predictions of any machine learning classifier.

Although with high potential, LIME does have some drawbacks that must be taken into consideration. For example, there have been some negative comments on LIME's potential due to possible wrong and unexpected interpretations because it suffers from label and data shift, and its explanations depend on how hyperparameters are picked. When the distribution of training and test are different, this would be called a data shift. In the validation stage of the ML pipeline, an underestimation or overestimation could be seen if there is data shift present.

Additionally, explanations generated from LIME are not considered robust. There is a paper written by David Alvarez-Melis and Tommi S. Jaakkola, "On the Robustness of Interpretability Methods", where they explain how explanations

from LIME could dramatically change when very minor changes in the input data are performed.

Sources

https://paperswithcode.com/method/lime

https://homes.cs.washington.edu/~marcotcr/blog/lime/

https://towardsdatascience.com/understanding-model-predictions-with-lime-a582fdff3a3b

https://towardsdatascience.com/whats-wrong-with-lime-86b335f34612

https://medium.com/xebia-france/explain-your-ml-model-predictions-with-local-interpretable-model-agnostic-explanations-lime-82343c5689db

https://github.com/marcotcr/lime

https://marcotcr.github.io/lime/tutorials/Lime%20-%20basic%20usage%2C%20two%20class%20case.html