**Basic summarization of what is present in this paper**

1. We introduce the perspective of viewing any explanation of a model’s prediction as a model itself, which we term the explanation model. This lets us define the class of additive feature attribution methods (Section 2), which unifies six current methods.

1https://github.com/slundberg/shap 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

2. We then show that game theory results guaranteeing a unique solution apply to the entire class of additive feature attribution methods (Section 3) and propose SHAP values as a unified measure of feature importance that various methods approximate (Section 3. We propose new SHAP value estimation methods and demonstrate that they are better aligned with human intuition as measured by user studies and more effectually discriminate among model.

1. **Additive feature attribution method:**
2. **Lime Basics**

LIME (Local Interpretable Model-Agnostic Explanations) is a popular model interpretation technique used to explain the predictions of any machine learning model. The goal of LIME is to provide interpretable explanations for individual predictions made by a model, which can help users understand how the model is making its predictions.

LIME works by creating a simpler, interpretable model that approximates the behavior of the original model in a local region around the prediction. This local model is then used to explain the prediction by highlighting the features that are most important in the local region.

The basic steps in the LIME process are:

1. Select a prediction to explain.
2. Define a "local" region around the prediction that will be used to create the interpretable model.
3. Sample a set of data points from the local region and use them to train an interpretable model (such as a linear regression model or decision tree).
4. Use the interpretable model to identify the features that are most important in the local region, and generate an explanation based on these features.

One of the key benefits of LIME is that it is model-agnostic, meaning that it can be used to explain the predictions of any machine learning model regardless of its architecture or complexity. LIME has been used in a wide range of applications, including image classification, natural language processing, and fraud detection, among others.

**Lime example:**

Suppose we have a binary classification problem where we want to predict whether a bank loan application will be approved or not based on various input features such as the applicant's income, credit score, employment status, and so on. We have trained a machine learning model on historical loan data to make these predictions, but we want to understand why the model is making a particular prediction for a new loan application.

To do this, we can use LIME to generate an interpretable explanation for the prediction. Here's how we might apply LIME in this scenario:

1. Select a loan application that we want to explain the prediction for.
2. Define a local region around the loan application, which could be a subset of the training data that is similar to the input application in terms of the input features, or a synthetic dataset generated to represent the local region.
3. Sample a set of data points from the local region and use them to train an interpretable model, such as a logistic regression model or a decision tree.
4. Use the interpretable model to identify the most important features that are driving the prediction in the local region.
5. Generate an explanation for the prediction based on these features.

For example, suppose we select a loan application with an income of $50,000, a credit score of 650, and full-time employment status. We define the local region around this application by selecting the 10 loan applications in the training data that are most similar in terms of these features.

We sample a set of data points from the local region and use them to train a logistic regression model. The model shows that the most important features for predicting loan approval in the local region are credit score and employment status, with income being less important.

Based on this information, we can generate an explanation for the prediction by saying that the loan application is likely to be approved because the applicant has a reasonably good credit score and a stable job, despite having a lower income. This explanation can help us understand why the model is making its prediction and provide insights into the factors that are driving the prediction.

In summary, LIME can be used to provide interpretable explanations for machine learning models, which can help us understand how the models are making their predictions and provide insights into the factors that are driving the predictions.

1. **DeepLIFT**

DeepLIFT (Deep Learning Important FeaTures) is another model interpretation technique used to explain the predictions of deep neural networks. It aims to identify which input features are most important for a given prediction, and to what extent each feature contributes to the prediction.

The basic idea behind DeepLIFT is to compare the activations of a given neuron in the network for a given input to its activations for a reference input, and to assign importance scores to the input features based on the differences in the activations. The reference input is typically chosen to be a baseline input that represents the "default" or "background" state of the input features.

Here's a high-level overview of the DeepLIFT algorithm:

1. Choose a neuron in the network whose output you want to explain.
2. Choose a reference input that represents the "default" or "background" state of the input features.
3. Compute the contribution of each input feature to the output of the neuron using the following formula:

contribution = (input - reference) \* (output - reference\_output) / (reference - epsilon)

where input is the input to the network for which you want to compute feature importance, reference is the reference input, output is the output of the neuron for the input, reference\_output is the output of the neuron for the reference input, and epsilon is a small constant to prevent division by zero.

1. Sum the contributions across all input features to obtain a final importance score for each feature.

These importance scores can be visualized as heatmaps or bar charts to help understand which input features are most important for a given prediction.

1. **Classic Shapley Value Estimation**

**Text

Description automatically generated**

**Basics About SHAP**

The statement is referring to the SHAP (SHapley Additive exPlanations) approach, which is a method for interpreting the predictions of machine learning models. The main idea behind SHAP is to assign an importance value to each feature in a particular prediction, which can help explain how the model arrived at that prediction.

The statement highlights two novel components of the SHAP approach:

1. Additive feature importance measures: The SHAP approach identifies a new class of additive feature importance measures, which means that the importance value for a particular feature is calculated based on the contributions of that feature and all other features in the model. This is different from other feature importance measures that consider each feature independently.
2. Theoretical results: The SHAP approach provides theoretical results that show there is a unique solution in this class of additive feature importance measures that has a set of desirable properties. This means that the SHAP approach is grounded in a solid mathematical framework, which gives it credibility and reliability.

Overall, the SHAP approach is a powerful tool for interpreting the predictions of machine learning models. By assigning importance values to each feature in a prediction, it can help explain how the model arrived at that prediction, which can be useful for identifying sources of bias, improving transparency, and building trust in machine learning systems.