Interpretability with SHAP

Deep Learning - 677

John Morrison

Introduction

Many times, models will sacrifice accuracy for interpretability. For instance, a linear model is much easier to explain compared to a more complex one like a deep neural network. On the other hand, more complex models like NN's and ensemble tree methods have very strong predictive capabilities. Due to the complexity, these models often fall short when it comes to interpretability, which is why many refer to these as black box models. Information is entered and a prediction is spit out, and it is somewhat a mystery on how the different features influence the prediction. This fact subsequently prevents the use of these complex models in domains where the stakes are very high (i.e. finance, medicine, regulatory systems, etc.). Luckily, ancillary models which are much simpler can be applied at a local level to help explain what is really going on under the hood. Two predominate approaches in the interpretability toolkit are LIME (Local Interpretable Model-Agnostic Explanations) and SHAP (Shapley Additive exPlanations). LIME uses linear regressions while SHAP uses game theory methods. SHAP will be the focus of this paper.

Background

Lloyd Shapley proposed the original idea in 1953 and has since been honored for his findings with a Nobel Prize. I always wondered how game theory related to machine learning and it has now become clear to me that it’s used for explainability! Game theory is used to test combinations of features and see how they affect predictions. One example is the following – say your company wants to examine profitability based on adding certain employees to a project. It’s easy to intuit that certain combinations of personnel could lead to greater profits. Maybe Lisa is very technical while Tom is more creative – this combination might be more valuable than say two technical folks. This thought experiment can be mapped to a machine learning problem. Instead we would swap the employees for features of a dataset and instead of profitability it would be predicted outcomes.

Okay, so now that we have a high-level understanding of the approach now it might be worth describing some other components to this framework. Two main pieces are *baselines* and *counterfactuals*. Both of these components help produce *shapley values*. The baseline might be the expected value of an outcome or average case. The counterfactuals are the different combinations of features. The baseline can be compared against the counterfactuals. The counterfactuals might uncover interaction effects within the data. The shapley values are not so much a statistical value they are more of a score – the values can be compared against each other to examine the features. The SHAP approach is very reliable because its additive and exhaustive – every potential combination of features are evaluated.

Caveats

Generating these shapley values can be computationally expensive. The approach uses additive feature attribution methods which is a similar approach to LIME however rather than using linear regressions it uses linear combinations of features and game theory. Actually, it might be more accurate to say that LIME is a subset of SHAP. This could lead to millions of operations. Also, it might be worth mentioning that this approach is model agnostic however there are different optimizations for different types of models. The best optimized model really cater to tree-based models, but there are also solutions for neural networks and deep frameworks.

Another limitation which might go hand in hand with explainers more generally is the fact that you will sometimes need an expert opinion. If your data is showing a feature to be very important you might want to verify that that makes sense in the real world. In the case of X-Ray analysis, a specialized technician or doctor could be brought in to confirm if the predicted areas in the lungs are actually healthy or compromised. The expert could also uncover biases in the data or maybe discover issues in the data collection process.

Tools

The model is referred to as the "explainer" – this would be chosen based on what the complex model type is – in our case we would use a GradientExplainer or a DeepExplainer. After the explainer is constructed, you'll pass in your observations or some subset of them. The library to be imported is called "shap" and there are a number of visualization tools that aid in the the analysis. A lot of the visualization tools seem to be better suited for tabular data but there are image-based ones as well. The "image_plot" method could be used to examine the image features – the features are essentially the parts of the image. The shapley values are represented by two different colors and the more contrasted they are the more likely the feature is predicting for or against a certain outcome. Another similar tool in deep models is deepLift which is a modified approach to backpropagation that uses multipliers. One interesting experiment that illustrates the power of SHAP is where they took mnist and asked what are the feature that would contribute to turning the number 8 into a 3. Humans sometimes do this exercise in their own head when they miswrite something and they write over it and preserve the good parts that are already there. SHAP, in this experiment, outperformed the deepLift and LIME.

References

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems.

Google. AI Explanations Whitepaper. https://storage.googleapis.com/cloud-ai-whitepapers/AI%20Explainability%20Whitepaper.pdf