A Model Explanation Framework Aligning Shapley Contributions and Permutation Feature Importance

Fang Wang[1], Tianyi Mao[1], Xinghua Liang[1,] Xin Chen[1]
[1]Amazon, 410 Terry Ave. North, Seattle, WA 98109

Abstract

SHAP (SHapley Additive exPlanations) is widely used in machine learning model explanations nowadays, especially for complex and black-box models (deep learning models, ensemble models). SHAP assigns a feature contribution to every record. Users can check each individual record feature contribution or use the mean absolute SHAP values over the entire dataset as the SHAP feature importance. But it's not uncommon to see contradicting model explanation results using SHAP such as individual record SHAP contributions are counter intuitive and feature contributions are opposite to the feature-to-target correlation relationships; SHAP feature importance ranking is quite different from permutation feature importance (PFI) ranking; top features identified by PFI got little SHAP feature contributions and ranked low by SHAP importance. This paper summarizes a scenario where, empirically, the Shapley value estimations highly varies between different perturbation methods for tree-type models. We present a framework to select the appropriate SHAP methods according to the model and data characteristics and generating the reasonable model explanations by cross-validating top SHAP and PFI features along the tree split process.

Key Words: feature selection, Shapley feature contribution, permutation feature importance, tree-type models

1. Introduction

SHAP (SHapley Additive exPlanations) by Lundberg and Lee (2017) is a method to explain machine learning model predictions especially black-box models such as tree-based models. SHAP is based on the game theory which tells us how to fairly distribute the "payout" among the features. The model prediction can be attributed to all the independent variables as feature contributions. For each record, the sum of the SHAP contributions equal to the model outcome for the record. For a single feature, the mean absolute value of a feature's contributions over all the instances (rows) of the dataset also represents the aggregated contribution of this feature. The aggregated contributions of the features are commonly used to rank the features as the SHAP feature importance ranking. But it also has its disadvantages such as: 1) Shapley value calculation is time consuming, which limited its feasibility to run SHAP for all the records and get the global feature ranking efficiently. 2) The ranking of SHAP assumes feature independence which is hard to reach for real world problems. The two disadvantages could lead to inaccurate global feature ranking using SHAP.

Another commonly adopted feature importance ranking method is permutation feature importance (PFI). Different from SHAP which ranks features based on feature contributions, PFI measures a feature's importance by calculating the model performance reduction after randomly shuffling the feature values. A random shuffled feature breaks the correlation between the feature and the target variable automatically without requiring feature independence. If the model performance dropped significantly after shuffling, the original feature is important to model the target and is ranked high in PFI feature importance ranking. The advantage of PFI is that it provides an accurate global insight of the feature importance by checking the overall model performance decrease. The

disadvantage is that PFI can't explain how much each feature contributed to the model outcome like SHAP.

Due to the different feature ranking mechanisms, SHAP and PFI feature importance rankings do not align on many use cases. Some features may show significant ranking discrepancies between the two methods. To get a global feature ranking, PFI is a more accurate and efficient choice. But to show the feature contributions, SHAP is the most widely adopted option. Different from PFI, SHAP also has various settings and parameters to choose, which made it's even harder to tell which setting yields the accurate feature contributions. This work discusses a method to use PFI global feature importance ranking as the ground truth feature ranking, compare PFI ranking with different SHAP feature contributions to select the right SHAP setting, find the smallest sufficient sample size to calculate SHAP without running SHAP for the entire dataset, align PFI and SHAP to provide feature explanations with both rankings and contributions.

In this paper, our contributions are:

- Provide a method to detect the potential SHAP setting issues by comparing to PFI global ranking.
- Provide a method to select the right SHAP method by analyzing the data and model structures.
- Provide a method to decide the minimum sufficient sample size for running SHAP on a subset of the entire dataset.

2. Methodology

2.1 SHAP Settings

Shapley values are computed by introducing each feature, one at a time, into a conditional expectation function of the model's output,

$$\boldsymbol{f_x(S) = \mathbf{E}[\mathbf{f}(\mathbf{X}) \,|\, \mathbf{do}(X_S = x_S)]}$$

and attributing the change produced at each step to the feature that was introduced; then averaging this process over all feature orderings. This process is called feature perturbation. The subset $\boldsymbol{X_S}$ is the coalition of present features (i.e. we know what values these features take for data instance x). $\mathbf{E}[\mathbf{f}(\mathbf{X}) \,|\, \mathbf{do}(\boldsymbol{X_S = x_S})]$ is the conditional expectation when we intervene to make $\boldsymbol{X_S = x_S}$.

TreeExplainer is the SHAP algorithm to explain the output of ensembled tree models. There are two feature perturbation methods using Tree Explainer: tree-path dependent and interventional. Using tree-path dependent feature perturbation, no prior data is provided to estimate the feature distribution. Tree-path dependent perturbation infers the background distribution based on the structure of the model. The tree-path dependent approach is to follow the decision trees and use the number of training instances that ended up in each leaf to represent the background distribution. In contrast, interventional feature perturbation method requires users to provide a background data, and the absence of a feature is simulated by replacing the feature with one of the values it takes in the background dataset specified during SHAP calculation. Interventional feature perturbation uses the real input sample data (background data) to represent the conditional distribution rather than inferring the distribution using the tree-based model leaf. Tree-path dependent feature perturbation is the default setting for TreeExplainer, and it's much faster than interventional feature perturbation which needs background specification.

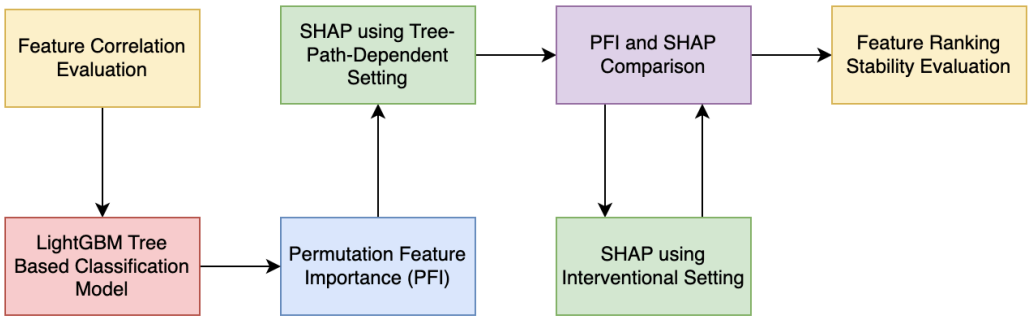
Using the tree-path dependent feature perturbation, due to the nature of calculating the conditional distribution using the number of training instances that ended up in each leaf, features used in later stages of a tree model split receive larger SHAP contributions if the input data has highly correlated features. For two features which are correlated, the more important feature which is

used earlier in the tree split gets smaller feature contribution, and the less important feature which was closer to the tree leaves got higher SHAP contribution values. This leads to the unalignment of PFI and SHAP rankings. To solve the problem, the Interventional feature perturbation could be a better choice. But even though high intercorrelated dataset is an indicator to use Interventional rather than tree-path dependent setting, it still has a lot of challenges using this Interventional feature perturbation setting:

1) Interventional setting is very time consuming. It may not be feasible to finish entire dataset SHAP running having a large data. Tree-path dependent setting is the more feasible for most use cases.
2) Interventional setting doesn't support categorical feature contribution calculation. All the categorical features need to be one-hot encoded to dummy variables for both model training and SHAP calculation.
3) Intercorrelation issue exits for almost all real-world problems. It can't be the sole indicator about which setting to use to get more accurate SHAP feature contributions. Additional benchmark and data analysis should be conducted to detect potential inaccurate SHAP contribution issue.

Given the challenges, we present the solution below to choose which feature perturbation setting to use (tree-path dependent vs interventional), and how to calculate interventional feature perturbation SHAP contributions without running the entire dataset.

2.2 Solution Overview



2.3 Feature Correlation Evaluation

Our original use case is to identify which factors lead to employee attritions. We used a public dataset of IBM HR Analytics Employee Attrition and Performance for this study. This dataset has 37 features: the target variable is the attrition indicator Yes/No, other variables describing an employee are Education, Job Satisfaction, Job Level, Job Role, Income, Performance Rating, Age, Gender etc. We calculated the Pearson correlations among all the feature pairs to identify the feature pairs with high correlations. The highly correlated features may lead to inaccurate SHAP contribution calculations using a tree-based model.

2.4 Tree-Based Classification Model

We trained a tree-based classification model using Attrition (Yes/No) as target variable, and all the other features as predictive variables. We implemented the ensembled LightGBM model training through AutoGluon AutoML library. We randomly split the dataset into 80% for training and 20% for test.

2.5 Permutation Feature Importance

Permutation Feature Importance (PFI) is also implemented using AutoGluon. Using the trained classification model, we randomly shuffled each feature 10 times one by one and calculated the model accuracy decrease using the shuffled feature. The permutation feature importance is calculated on the hold-out test data to highlight the feature impacts on generalized data outside of the training data.

2.6 PFI and SHAP Comparison

Our method is to first run Permutation Feature Importance using the entire dataset to get a global feature importance ranking as the ground truth feature importance ranking. Next, we run SHAP at different settings starting from a smaller sample size (if the dataset is large) and compare the different SHAP rankings with PFI ranking. If the default tree path dependent setting displayed feature ranking discrepancy especially at top features, and the interventional setting ranking aligned with PFI better, we select interventional and incrementally increase the sample size until the SHAP ranking is stable. This way, we solved the SHAP interventional time-consuming challenge by finding the smallest sufficient sample size without running interventional SHAP for the entire dataset.

3. Result

3.1 Feature Correlation Evaluation

There are 35 features in total in the dataset 26 of them are numeric features and 9 are categorical f eatures. 21 pairs of features are highly correlated with each other, with correlation>=0.5.

Table 1 Feature Pairs with Feature Correlation>=0.5

Feature Pairs	Correlation
JobLevel, MonthlyIncome	0.95
JobLevel, TotalWorkingYears	0.78
PercentSalaryHike, PerformanceRating	0.77
MonthlyIncome, TotalWorkingYears	0.77
YearsAtCompany, YearsWithCurrManager	0.77
YearsAtCompany, YearsInCurrentRole	0.76
YearsInCurrentRole, YearsWithCurrManager	0.71
Age, TotalWorkingYears	0.68
TotalWorkingYears, YearsAtCompany	0.63
YearsAtCompany, YearsSinceLastPromotion	0.62
YearsInCurrentRole, YearsSinceLastPromotion	0.55

3.2 Tree-Based Classification Model

We trained an ensemble of LightGBM classification model under 3 different hyperparameter settings with L1 regularization. Each parameter setup has 8 child models, which are later stacked together and ensembled to form the final model. We chose LightGBM model in this experiment for its fastest training speed. Any tree-based model can be explained using TreeExplainer.

Even though LightGBM doesn't require feature one-hot encoding, SHAP Interventional setting requires all the features to be numeric, we one-hot encoded all the categorical variables into dummy variables. After one-hot encoding, we had 52 features in total to train the tree-based classification model. The final model ROC-AUC is 0.81.

3.3 Feature Ranking Comparison

This dataset has1470 rows and 52 variables (after one-hot encoding). For such small data size, we used the entire dataset to run SHAP TreeExplainer. We ran Permutation Feature Importance to get the ground truth feature importance ranking, SHAP feature importance ranking with tree path dependent setting and interventional setting. Below is the top feature ranking comparison of PFI, SHAP Tree Path Dependent, and SHAP Interventional.

Table 2. Top 10 features from PFI, SHAP Tree Path Dependent and SHAP Interventional

Permutation Feature Importance	SHAP Tree Path Dependent	SHAP Interventional
1. OverTime_No,	1. PerformanceRating,	1. MonthlyIncome,
2. EnvironmentSatisfaction,	2. JobRole_Research Scientist,	2. Age,
3. StockOptionLevel,	3. JobRole_Sales Executive,	3. EnvironmentSatisfaction,
4. NumCompaniesWorked,	4. JobLevel,	4. JobInvolvement,
5. RelationshipSatisfaction,	5. OverTime_No,	5. WorkLifeBalance,
6. JobSatisfaction,	6. Age,	6. OverTime_No,
7. DistanceFromHome,	7. YearsWithCurrManager,	7. JobRole_Sales Representative,
8. Age,	8. YearsSinceLastPromotion,	8. DailyRate,
9. BusinessTravel_Travel_Frequently,	9. JobRole_Laboratory Technician,	9. TotalWorkingYears,
10. EmployeeNumber	10. StockOptionLevel	10. JobSatisfaction

Tree-path dependent ranked 3 job role levels Research Scientist, Sales Executive, Laboratory Technician among top 10 features, but missed EnvironmentalSatisfaction and JobSatisfaction. EnviromentalSatisfaction is the 2nd most important features in PFI and 3rd most important in SHAP interventional ranking, but SHAP tree-path dependent ranked it as 27th among all 52 variables. EnvironmentSatisfaction is used early at tree split, and Job Roles are used at later steps in the tree split.

Looking at the correlations between the cases with least environment satisfaction (EnvironmentSatisfaction=1) and Job Roles, focusing on job roles with more than 100 cases. We round that Laboratory Technician, Sales Executive, Research Scientist are 3 roles highest correlated to Environment Satisfaction. Using tree-path dependent feature perturbation method, the feature contributions are assigned to the three job roles, rather than the more important EnvironmentSatisfaction which was used early in tree splits and impacts model performance greatly according to Permutation Feature Importance.

Table 4. Correlation of EnviromentSatisfaction=1 and Job Role

Job Role Dummy Variables	Correlation to EnviromentSatisfaction=1 Dummy Variable
JobRole_Laboratory Technician	0.0134
JobRole_Sales Executive	0.0125
JobRole_Research Scientist	0.0112
Manager	0.0020
JobRole_Manufacturing Director	-0.0405
Healthcare Representative	-0.0200

From this experiment, interventional feature perturbation method aligns with Permutation Feature Importance better, which should be used as the right SHAP TreeExplainer setting. So our next que stion is how to calculate Interventional SHAP feature contribution if running SHAP for the entire dataset is not feasible using interventional feature perturbation method.

3.4 SHAP Sample Size Stability Analysis

SHAP interventional setting is time consuming. This sample analysis helps us decide if running SHAP for a proportion of the entire data is sufficient to represent feature contributions of the entire dataset. The entire dataset has 1470 rows. We ran SHAP for 50, 100, 500, 1000 randomly selected rows and eventually the entire data, to compare the top 20 features overlaps. We defined a feature overlapping rate as number of common top features between SHAP and PFI divided by total number of top features considered (e.g. 20 in our analysis below). We look at the top 20 features and evaluate how many are common between SHAP and PFI. If the common features get stabilized, it means model SHAP feature contributions are stabilized and running SHAP for the current sample could represent entire dataset.

Table 5. SHAP Interventional and PFI Top 20 feature overlap rate at different sample sizes.

Sample Size	**Percent using Interventional Overlap with PFI**
50	70%
100	75%
500	75%
1000	75%
All Data	75%

According our analysis, after running SHAP for only 100 out of 1470 records (6.8% of entire dataset), the overlapping feature percentage between SHAP and PFI are stabilized, which means SHAP rankings are not changing after running SHAP for 100 records. This showed that, running SHAP for a small proportion of the entire dataset could represent feature contribution of the entire data by benchmarking with the ground truth PFI feature ranking. For large dataset, the similar experiment could be implemented by gradually increasing sample size until feature contributions stabilized by comparing to PFI. This method helps users find the smallest sufficient sample size for SHAP feature contribution calculation and avoid running SHAP for the entire dataset.

4. Conclusion

For dataset with intercorrelated features, we provided a method to use permutation feature importance (PFI) to validate the SHAP contribution soundness. This method helps users to detect SHAP contribution issues, select the proper SHAP setting to generate SHAP feature contributions. This study also provided a method to decide smallest sufficient sample size by comparing to global PFI feature ranking given the challenges of running SHAP interventional setting for an entire large dataset.

References

Lundberg, S.M., Erion, G., Chen, H. et al. From local explanations to global understanding with explainable AI for trees. Nat Mach Intell 2, 56–67 (2020).

Lundberg, Scott & Lee, Su-In. (2017). A Unified Approach to Interpreting Model Predictions.

Ajmal M S, TANMAY DESHPANDE, IBM Data Scientists, February 17, 2023, "IBM Analytics Employee Attrition & Performance", IEEE Dataport, doi: https://dx.doi.org/10.21227/2m1g-6v47.