Unpack Local Model Interpretation for GBDT

Wenjing Fang[1(✉)], Jun Zhou[1], Xiaolong Li[1], and Kenny Q. Zhu[2]

[1] Ant Financial Services Group, Hangzhou, China
{bean.fwj,jun.zhoujun,xl.li}@antfin.com
[2] Shanghai Jiao Tong University, Shanghai, China
kzhu@cs.sjtu.edu.cn

Abstract. A gradient boosting decision tree (GBDT), which aggregates a collection of single weak learners (i.e. decision trees), is widely used for data mining tasks. Because GBDT inherits the good performance from its ensemble essence, much attention has been drawn to the optimization of this model. With its popularization, an increasing need for model interpretation arises. Besides the commonly used feature importance as a global interpretation, feature contribution is a local measure that reveals the relationship between a specific instance and the related output. This work focuses on the local interpretation and proposes an unified computation mechanism to get the instance-level feature contributions for GBDT in any version. Practicality of this mechanism is validated by the listed experiments as well as applications in real industry scenarios.

1 Introduction

Machine learning has great success in modeling data and making predictions automatically. In many real-world applications, we need an explanation rather than a black-box model. For example, when customers apply for a loan on credit, the loan officers will compute their credit scores based on their historical behaviors. In this case, it's far from enough to only show the customers the final scores, and the loan officers would better give some detailed reasons. While most efforts in data mining have been made on improving the accuracy and efficiency, which results in better models, little attention is paid to model interpretation for these models. Several common measures for the variable significance have been proposed. Gini importance is one of the commonly used importance measure for Random Forest, which is derived from the Gini index [2]. Gini is used to measure impurity between the parent node and two descendent nodes of samples after splitting. The final importance is accumulated from the Gini changes for each feature over all the trees in forest. This general feature importance(FI), also known as *global interpretation* , shows the important factors of the target, which unpacks the general information in the trained models. However, it doesn't take any feature values of an instance into consideration, which is insufficient sometimes. *Local interpretation*, on the other hand, places particular emphasis on a

J. Pei et al. (Eds.): DASFAA 2018, LNCS 10828, pp. 764–775, 2018.
https://doi.org/10.1007/978-3-319-91458-9_48

specific case and reveals the main causes of each record. This type of interpretation makes up for the shortages of the global one. One approach proposed to define the feature contributions(FC) [12] , which is accumulated from label distribution changes, as a measure of the feature impact on the output. The value of feature contribution reveals how much a feature contributes and the sign represents whether it's a positive impact or not.

GBDT [6] is an ensemble model built on top of a bunch of regression decision trees. It has some appealing characteristics. For example, GBDT can naturally handle nonlinearity and tolerate missing values. As a winning model in many data mining challenges [1,3,7], GBDT is a good option for regression, classification and ranking problems with well-known ability to generalize. Besides its wide range of applications, GBDT is also flexible in allowing users to define their own suitable loss functions. Furthermore, there are many implementations [4,8] and much work has been done to speed up the training process.

In most cases, GBDT outperforms linear models and random forest. Given the popularity and high quality of GBDT, it's important to uncover internals of the model. For GBDT, global feature importances calculation is widely used to do the feature selection. For example, Breiman proposed a method to estimate feature importance [6]. However, existing work has largely ignored the exploration of local interpretations, which will be the focus of this paper. Specifically, we will study feature contributions for GBDT. We starts from previous approaches of model interpretation for random forest [12] and update the definition of the feature contribution. The proposed mechanism is flexible enough to interpret all versions of GBDT. The original definition based on label distribution change is proved to be a special case of ours under a particular loss function.

The rest of the paper is organized as follows. Section 2 provides a brief review of related work on local interpretations. Section 3 gives out the formal definition of feature contribution as preliminary and presents the approach for calculating feature contributions for random forests. In Sect. 4, we describe the rationale behind as well as main actions in interpreting GBDT. Section 5 contains experiment settings and the process to examine the proposed methodology. At the end, Sect. 6 concludes our work.

2 Related Work

Local model interpretation provides convincing reasons to the model outputs. One type of interpretations prefer both the good performance of complex models and interpretability of simple models. The pipeline of this type will first make use of advanced models as a black-box and then extract useful information out of it with the help of a more interpretable model. For example, a novel approach in [5] formally treats the interpretation of additive tree models as extracting the optimal actionable plan. It models the optimization problem as an integer linear programming and utilizes existing toolkit as the solver. The constraints are based on both the output score and the objective function. Notice that, this

kind of approaches need extra training process especially for the interpretation and bring new models or tasks to solve.

Some other researchers come up with model-independent local interpretations. They mainly make changes to feature value and test the chain effect to performance loss of predictions. The loss is then taken as the measure of local importance of feature [11]. This method only relies on the output evaluation and provides an unified way to check feature contribution for black-box models. By replacing the actual feature values with missing, zero or average values, the impact of a feature in predicting is then removed. The instance-level contributions of all the features can be calculated separately and compared with each other. Moreover, this method is also work for global feature importance.

As a derivative of decision tree, the random forest goes further on model interpretation than GBDT. The method in [10,12] computes the feature contributions so as to show informative results about the structure of model and provide valuable information for designing new compounds. This method makes full use of the information, not only the training data but also the model structure. It is natural to design the interpretations with the model structures to get a more reasonable result.

This work proposes an easy way to get the feature contributions on the instance-level. Generally, it can be applied to all versions of GBDT implementations with little preprocessing and modification to the prediction process.

3 Preliminary

Additive tree models are a powerful branch of machine learning but are often used as black boxes. Though they enjoy high accuracies, it's hard to explain their predictions from a feature based point of view. Different ensemble strategies bring out different models while sharing the tree structure as a basis. So the model interpretations for different addictive tree models share some key spirits and can spread out from one to another with appropriate adaptation. In this section, we first review a practical interpretation method for random forest (for the binary classification) and introduce the general definition of feature contribution to better illustrate the proposed model interpretation for GBDT.

3.1 Interpretation for Random Forest

Random forest is one of the most popular machine learning models due to its exordinary accuracy utilizing categorical or numerical features on regression and classification problems. A random forest is a bunch of decision trees that are generated respectively and vote together to get a final prediction. Every tree is trained on randomly sampled data and subsampling feature columns to introduce the diversity for better generalization, which is the key weakness of single decision tree models. Random forest is known as a typical bagging model and the bagging strategy works out by averaging the noises to get a lower variance model.

An instance starts a path from the root node all the way down to a leaf node according to its real feature value. All the instances in the training data will fall into several nodes and different nodes have quite different label distributions of the instances in them. Every step after passing a node, the probability of being the positive class changes with the label distributions. All the features along the path contribute to the final prediction of a single tree.

A practical way to evaluate feature contributions is explored [12]. The key idea is taking the distribution change values for the positive class as the feature contribution. Concretely, it takes four procedures to work:

1. Computing the percentage of positive class of every node in a tree;
2. Recording the percentage difference between every parent node and its children;
3. Accumulating the contributions for every feature on each tree;
4. Averaging the feature contribution among all the trees in the forest;

The method consists of an offline preparation embedded in training (steps 1–2) and an online computing with the prediction process (step 3–4). It is easy to record the local contribution (or local increment) and related split feature to every edge on a tree.

3.2 Gradient Boosting Decision Tree

GBDT is another type of ensemble model that consists of a collection of regression decision trees. However, the ensemble is based on gradient boosting which promotes the prediction gradually by reducing the residual. For every iteration, a new model is built up to fit the negative gradient of the loss function until it converges under an acceptable threshold. The final prediction is the summation of all stagewise model predictions. Gradient boosting is a general framework and different models are available to be embedded. GBDT introduces decision tree as the basic weak learner. When square error is chosen as the loss function, the residual between current prediction and target label is the negative gradient which is computational friendly.

From the above definition, we can see the differences between random forest and GBDT, some of which are the main obstacles that prevent us from adapting the model interpretation for random forest to GBDT:

1. Random forest aggregates trees by voting, while GBDT sums up the scores from all the trees. This means that the trees in GBDT are not equal and the trees have to be trained in sequential order. The interpretation should make proper adaptations to deal with this problem.
2. Decision tree in GBDT outputs a score instead of a majority class type for classification problems. Though we can get the label distribution changes as random forest interpretation, the output scores in GBDT should be wisely taken into consideration.

3.3 Problem Statement

Given a training dataset $D = \{x^{(i)}, y^{(i)}\}_{i=1}^{N}$, where N is the total number of training samples, $x = (x_1, x_2, ..., x_S)$ implies a S dimensional feature vector, $x^{(i)}$ is the feature vector for the i-th sample and $y^{(i)}$ is the related label. We can illustrate training process of GBDT as in Algorithm 1. r_{mi} is the residual for sample i in the m-th iteration.

Algorithm 1. Gradient Boosting Decision Tree

function TRAIN(D,M)
  Init $f_0(x) = 0$
  for $m = 1, 2, ..., M$ **do**
    Compute residual:
    $r_{mi} = y_i - f_{m-1}(x_i),\ i = 1, 2, \ldots, N$
    Train a regression decision tree from residual:
    T_m =BUILDTREE(D)
    Cumulated prediction sum:
    $f_m(x) = f_{m-1}(x) + T_m$
  end for
  Get finally boosting function:
  $f_M = \sum_{m=1}^{M} T_m$
  return f_M
end function
function PREDICTINSTANCE(X_i,f_M)
  score = $\sum_{m=1}^{M}$ TREEPREDICT(X_i, T_m)
  return score
end function

Besides the basics of model, the feature contribution(FC) , as the key concept for local interpretation, is clarified below. We introduce the notation of FC by denoting the model interpretation for random forest in Sect. 3.1 :

$$LI_f^c = \begin{cases} Y_{mean}^c - Y_{mean}^p & \text{if the split in the parent is performed} \\ & \text{over the feature } f; \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

LI_f^n in Eq. 1 is the Local Increment(LI) of feature f for node n defined before. For binary classification, Y_{mean}^n represents the percentage of the instances belonging to the positive class in node n.

$$FC_{i,m}^f = \sum_{c \in path(i)} LI_f^c \tag{2}$$

$$FC_i^f = \frac{1}{M} \sum_{m=1}^{M} FC_{i,m}^f \tag{3}$$

On a single tree m, $FC_{i,m}^{f}$ in Eq. 2 cumulates the feature contribution of feature f for a specific instance i. Equation 3 later average all the feature contribution for feature f among all the trees.

4 Mechanism

Looking back at model interpretation for random forest, its central spirit is to establish the idea of feature contribution. By computing label distribution, a measure of the change is then obtained and associated with the split feature. In the case of GBDT, we can expand this computation with a slight modification. Because the targets of the latter trees are the residual, it should replace the instance label while computing label distribution. Nevertheless, the problem of this version is that the average of labels on a leaf node is not always equal to the score on it. So the valuable model information in these scores are not utilized and the method is not appropriate for different GBDT versions [4,6].

In fact, the loss function determines the optimal coefficient and Table 1 shows some common examples. LS and LAD stand for Least Square and Least Absolute Deviation respectively. $\tilde{y}_i$ is the residual updated after each iteration. $F_{m-1}(x_i)$ is the approximation on iteration $(m-1)$. g_i and h_i are the first and second order gradient statistics on the loss. Different from the numerical optimization essence to compute negative gradient (for LS and LAD), XGB [4] first approximates the loss function with its second order Taylor expansion and an analytic solution is then got. So it contains no negative gradient computation and the evaluation of leaf weights is far from the label average. Particularly, only if the LS loss function and traditional GBDT training process is used, the label averages meet the scores.

Table 1. Loss functions of GBDT

Settings	Loss function	Negative gradient	Leaf weight
LS	$\frac{1}{2}[y_i - f(x_i)]^2$	$y_i - f(x_i)$	$ave_{x_i \in R_{jm}} \tilde{y}_i$
LAD	$\mid y_i - f(x_i) \mid$	$sign[y_i - f(x_i)]$	$median_{x_i \in R_{jm}}\{y_i - F_{m-1}(x_i)\}$
XGB	$\sum_{i=1}^{n}[l((y_i, \hat{y}^{(t-1)})) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i))] + \Omega(f_t)$	/	$-\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}$

Without loss of generality, the interpretation for GBDT needs to work on the leaf scores. Since the scores are only assigned to leaf nodes, we have to find a way to propagate them back all the way to the root. The left tree of Fig. 1 shows an example tree in a GBDT model, with split feature and split value marked on arcs. Observing the three nodes in the rounded rectangle, the instances in node 6 will get a score difference as: $S_{n11} - S_{n12} = 0.085 - 0.069 = 0.016$, where S_{nk} is the score on node k. Moreover, this difference is caused by splitting feature $feat5$ branching by a threshold of 1.5. We can allocate this difference to the two branches by assigning the average score of child nodes to their parent node. For

instance, $S_{n6} = \frac{1}{2}(S_{n11} + S_{n12}) = \frac{1}{2} \times (0.085 + 0.069) = 0.0771$. Then, the local increment metrics could be calculated using the scores, $LI^{n11}_{feat5} = S_{n11} - S_{n6} = 0.085 - 0.0771 = 0.0079$. Similarly, the leaf scores as well as the local increment could be spread to the whole tree.

The interpretation process during predicting is the same as that of the random forest. On the right hand side of Fig. 1, all the node average scores and feature contributions on the tree are marked. Supposing an instance gets a final prediction on leaf node 14 of tree t, a cumulation through the path: $n0 \rightarrow n2 \rightarrow n5 \rightarrow n9 \rightarrow n14$ will be executed: $FC^t_{feat5} = LI^{n2}_{feat5} = -0.0201$, $FC^t_{feat2} = LI^{n5}_{feat2} = -0.0073$,$FC^t_{feat4} = LI^{n9}_{feat4} + LI^{n14}_{feat4} = -0.0015 + 0.0010 = 0.0025$.

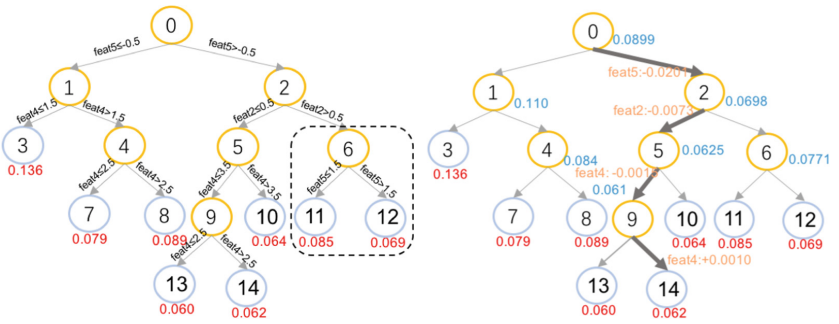

Fig. 1. Feature contribution example for GBDT

By the propagation strategy, the average score is assigned to the node 6 which assumes an instance falls into the left branch or the right with equal probability. So the expectation of intermediate nodes could be revised as in Eq. 4:

$$S_p = \frac{1}{2}(S_{c1} + S_{c2}) \rightarrow \frac{N_{c1} \times S_{c1} + N_{c2} \times S_{c2}}{N_{c1} + N_{c2}}, \tag{4}$$

where the N_{c1} and N_{c2} is the number of the instances fall into child nodes node c1 and c2. These statistics need extra information from training process.

By viewing the computation in this brand new way, we get a flexible interpretation mechanism by only using the leaf node scores and instance distributions, regardless of the implement settings of GBDT. Under the setting of the LS loss function, we can see that not only the label distribution meets the prediction score on leaf node but also the label distribution of the intermediate node meets our back propagated score. That is to say, the label distribution method is a special case of our mechanism with this particular setting. Furthermore, this method also supports the multiple classification problems.

5 Experiment

In this section, we demonstrate the experiments on the proposed interpretation. In the first place, we show the mechanism is reliable and generally agrees with

global feature importance. Then we compare our interpretations to those of random forest and find it accord with the global feature importance better. Finally, we study the interpretations of real cases in our scenario and get a satisfied analysis for them.

5.1 Experiment Setup

The GBDT version in our experiment is the Scalable Multiple Additive Regression Tree(SMART) [13], which is a distributed algorithm under the parameter server. Hundreds of billions of samples with thousands of features could be trained by the algorithm. Not only the storage usage but also the running time cost is optimized without the loss of the accuracy.

The training data is drawn from transactions under the scene of Fast Pay(FP) in Alipay[1]. A transaction is marked as a positive if it is reported as a fraud by the customer. To keep a balanced ratio between positive and negative cases, only 1% of normal transactions are retained by random sampling.

```
<Node id="0">
    <True/>
    <Node id="1">
        <SimplePredicate field="feat5" operator="lessOrEqual" value="-.5"/>
        <Node id="3" score="0.1366064239336732">
            <SimplePredicate field="feat4" operator="lessOrEqual" value="1.5"/>
        </Node>
        <Node id="4">
            <SimplePredicate field="feat4" operator="greaterThan" value="1.5"/>
            <Node id="7" score="0.07905535474853541">
                <SimplePredicate field="feat4" operator="lessOrEqual" value="2.5"/>
            </Node>
            <Node id="8" score="0.08930692524151827">
                <SimplePredicate field="feat4" operator="greaterThan" value="2.5"/>
            </Node>
        </Node>
    </Node>
    <Node id="2">
        <SimplePredicate field="feat5" operator="greaterThan" value="-.5"/>
        <Node id="5">
            <SimplePredicate field="feat2" operator="lessOrEqual" value=".5"/>
            <Node id="9">
                <SimplePredicate field="feat4" operator="lessOrEqual" value="3.5"/>
                <Node id="13" score="0.06043183878498103">
                    <SimplePredicate field="feat4" operator="lessOrEqual" value="2.5"/>
                </Node>
                <Node id="14" score="0.06211634427742983">
                    <SimplePredicate field="feat4" operator="greaterThan" value="2.5"/>
                </Node>
            </Node>
            <Node id="10" score="0.06457350478010203">
                <SimplePredicate field="feat4" operator="greaterThan" value="3.5"/>
            </Node>
        </Node>
        <Node id="6">
            <SimplePredicate field="feat2" operator="greaterThan" value=".5"/>
            <Node id="11" score="0.08556975499229456">
                <SimplePredicate field="feat5" operator="lessOrEqual" value="1.5"/>
            </Node>
            <Node id="12" score="0.0691611364527228">
                <SimplePredicate field="feat5" operator="greaterThan" value="1.5"/>
            </Node>
        </Node>
    </Node>
</Node>
```

Fig. 2. GBDT model in PMML format

[1] https://global.alipay.com/.

Figure 2 is a fraction of GBDT model in Predictive Model Markup Language (PMML) format[2] and the tree embedded in it can be translate as shown in Fig. 1. The element *Node* is an encapsulation for a tree node, which contains a predicative rule to choose itself or its siblings. The attribute *id* assigns a unique number to each node in a tree. The value of *score* in a *Node* is the predicted value for an instance falling into it. *SimplePredicate* is a simple Boolean expression indicating the split information. Our pre-trained model is stored as a PMML file. JPMML[3] is employed as the evaluator and we implement the proposed interpretation based on it.

5.2 Consistency Check

We implement the feature contribution as the previous description in [6]. In order to make the interpretation be independent of the training process of GBDT, the training algorithm is not changed in our experiment. In order to get the distribution of instances in Eq. 4, we use JPMML to predict the training instances and record instance distributions on every node. According to the tree structure in model and instance distributions, the pre-process is done by back propagating the local increments as shown in Sect. 4. With the local increments, the feature contributions of the new instances could be computed. After interpreting lots of instances, we can get a distribution of feature contributions among the instances. The median is a robust estimator for the expectation of the general feature contribution and should somehow keep accordance with the global feature importances metrics [12].

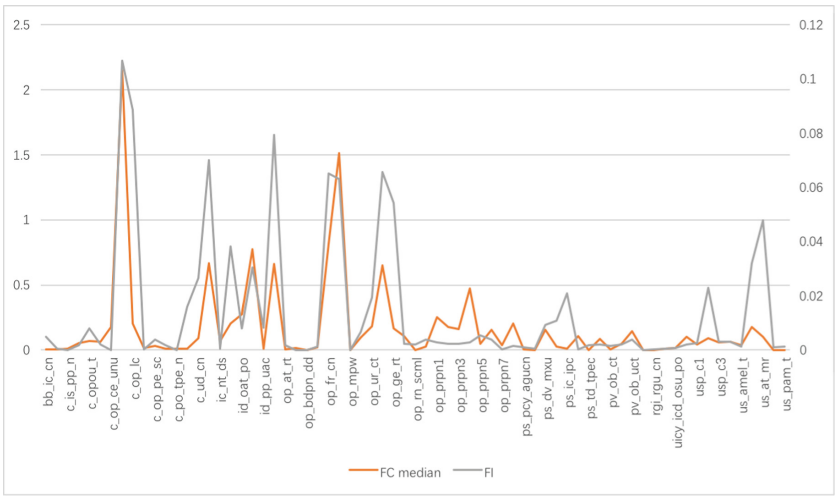

Fig. 3. Feature importance and feature contribution medians

[2] http://dmg.org/pmml/v4-3/GeneralStructure.html.
[3] https://github.com/jpmml/jpmml-evaluator.

Figure 3 plots the global Feature Importance(FI) for GBDT and Feature Contribution(FC) medians for every feature. As we can see, this two statistics have similar distributions and are in good agreement. It proves that the interpretation for GBDT is practical and reasonable.

5.3 Comparison to Random Forest

Following the experiment of last section, we get a ranking of the feature contribution median. This ranking is a measure of feature importance and reflects the quality of local interpretation. We implement the work for random forest in [12] and compare it with our ranking. For justice, we replace the GBDT Feature Importance with Information Value(IV) as the importance metric. IV is a concept from information theory and shows the predictive strength for the features [9].

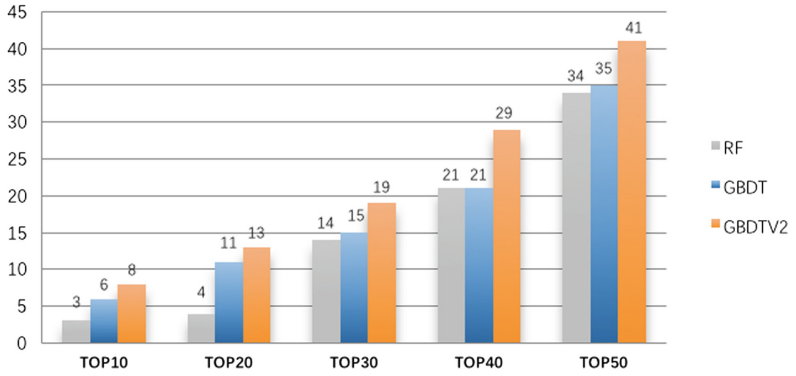

Fig. 4. Interpretation: GBDT v.s RF

In Fig. 4, we compute the intersection size on different variable coverage (i.e. Top 10–50 features of IV). *RF* implies the method explained in Sect. 3.1. *GBDT* is the simple average strategy with only the information in PMML file. *GBDTV*2 is the revised version in Eq. 4. From the result, our interpretations capture the importance better and the revised version works best.

5.4 Case Study

Besides the general evaluation, we analysis the 300 specific instances in the test data. Figure 5 shows a case, we only list some representative fields and divide them into 4 parts. The variables are ranked by IV (general feature importance). Domain experts check the feature risk manually and draw the following conclusions:

- Part I: Variables in this section are with high IV, our interpretation is able to capture the features that are judged to be high risk(marked as blue fields). The feature with high IV but low risk (judging from the feature value) is assigned a lower score, so the interpretation is good for instance-level contributions.
- Part II: There are 2 variables(colored pink) with high IV and marked high risk is missed by the interpretation, which mainly due to its low occurrence in split features. The global importance of these two variables is also low and model interpretations are limit by the model quality.
- Part III: Variables with median or low IVs are not caught by mistake and is assigned a low feature contribution for that case.
- Part IV: Several variables are considered to be high risk for the particular instance, even the general IVs of them are low. Our interpretation finds them out, which shows the superiority of the local feature contribution over the global feature importance.

Variables	IV	IVrank	Feature Importance	Feature Value	Risk	
c_op_ud_lk	2.203	1	0.106791601	2	High	I
opcd_pn_cy	1.762	2	0.065620865	0	High	
id_pp_cy	1.525	3	0.030499262	0	High	
op_st_at_d	1.445	4	0.079317491	1	High	
op_fd_pob	1.315	5	0.063102101	-1	Low	
ic_dcn_dr	1.126	6	0.07008353	0.333333333	High	
op_fr_cn	1.065	7	0.065047636	1	High	
c_op_lc	0.92	8	0.088589539	0	High	
op_ur_ct	0.874	9	0.019405404	2	High	II
id_oat_po	0.842	10	0.008055243	1	High	
c_op_cr_y	0.811	11	0.002003042	2	High	
uicy_icdcy_ud	0.778	12	0.00064423	0.078567	Low	III
c_opcd_cn	0.693	13	0.00193592	0	Median	
uicy_icd_osu_po	0.68	14	0.000859095	0.117214	Low	
ps_pn_ct	0.651	15	0.00016962	15	Low	
op_prpn1	0.624	16	0.002749283	0	Low	
ic_ay_dcn	0.618	17	0.038145332	-1	Low	
us_bste_t	0.608	18	0.03190312	0	High	IV
usp_c1	0.343	29	0.002563547	4	High	
op_ge_rt	0.203	52	0.054343817	0	High	
us_at_mr	0.196	53	0.047708771	56.50257164	High	

Fig. 5. Case study for interpretation

Further more, if we conduct interpretations on a batch of fraud cases which are missed by the model, the local feature contributions will help analysts improve the model.

6 Conclusion

Employing models as a black-box is not enough. A measure for the impact of a feature on the prediction convinces analysts in an intuitive way. The local

interpretation provides an explanation when necessary and contributes to the promotion of the models. We describe a method to unpack the interpretation for the advanced model GBDT. To the delight of analysts, the whole process is independent from the training details and technical optimizations. Only the tree structure and instance distribution are needed, which can be easily extracted by a post-processing after training. The label distribution based method of random forest is proved to be a special case of our method. We explore the distribution of local feature contributions and prove it to be in agreement with global feature importance. The method is applied to real case studies in different scenarios and serves as a good translator of our models.

References

1. Bennett, J., Lanning, S., et al.: The netflix prize. In: Proceedings of KDD Cup and Workshop, New York, NY, USA, vol. 2007, p. 35 (2007)
2. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
3. Chapelle, O., Chang, Y.: Yahoo! learning to rank challenge overview. In: Proceedings of the Learning to Rank Challenge, pp. 1–24 (2011)
4. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM (2016)
5. Cui, Z., Chen, W., He, Y., Chen, Y.: Optimal action extraction for random forests and boosted trees. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 179–188. ACM (2015)
6. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**, 1189–1232 (2001)
7. He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., Shi, Y., Atallah, A., Herbrich, R., Bowers, S., et al.: Practical lessons from predicting clicks on ads at Facebook. In: Proceedings of the Eighth International Workshop on Data Mining for Online Advertising, pp. 1–9. ACM (2014)
8. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: LightGBM: a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems, pp. 3149–3157 (2017)
9. Kullback, S.: Information Theory and Statistics. Courier Corporation, Chelmsford (1997)
10. Kuz'min, V.E., Polishchuk, P.G., Artemenko, A.G., Andronati, S.A.: Interpretation of QSAR models based on random forest methods. Mol. Inform. **30**(6–7), 593–603 (2011)
11. Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R.J., Wasserman, L.: Distribution-free predictive inference for regression. J. Am. Stat. Assoc. (2017, accepted)
12. Palczewska, A., Palczewski, J., Robinson, R.M., Neagu, D.: Interpreting random forest models using a feature contribution method. In: 2013 IEEE 14th International Conference on Information Reuse and Integration (IRI), pp. 112–119. IEEE (2013)
13. Zhou, J., Cui, Q., Li, X., Zhao, P., Qu, S., Huang, J.: PSMART: parameter server based multiple additive regression trees system. In: Proceedings of the 26th International Conference on World Wide Web Companion, pp. 879–880. International World Wide Web Conferences Steering Committee (2017)