DO NOT TRUST ADDITIVE EXPLANATIONS

A PREPRINT

Alicja Gosiewska
Faculty of Mathematics and Information Science
Warsaw University of Technology
alicjagosiewska@gmail.com
https://orcid.org/0000-0001-6563-5742

Przemyslaw Biecek
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw
Faculty of Mathematics and Information Science
Warsaw University of Technology
przemyslaw.biecek@gmail.com
https://orcid.org/0000-0001-8423-1823

December 2, 2019

ABSTRACT

Explainable Artificial Intelligence (XAI) brings a lot of attention recently. Explainability is being presented as a remedy for a lack of trust in model predictions. Model agnostic tools such as LIME, SHAP, or Break Down promise instance level interpretability for any complex machine learning model. But how certain are these additive explanations? Can we rely on additive explanations for non-additive models?

In this paper, we (1) examine the behavior of the most popular instance-level explanations under the presence of interactions, (2) introduce a new method that can handle interactions for instance-level explanations, (3) perform a large scale benchmark to see how frequently additive explanations may be misleading.

1 INTRODUCTION

Predictive models are used in almost every aspect of our life, in school, at work, in hospitals, police stations, or dating services. They are useful, yet, at the same time can be a serious threat. Models that make unexplainable predictions may be harmful (O'Neil, 2016). Need for higher transparency and explainability of models is a hot topic of the recent year both in the Machine Learning community (Gill and Hall, 2018) as well as in the legal community that coined the phrase „Right to Explain" in the discussion around General Data Protection Regulation (Wachter et al., 2017; Edwards and Veale, 2018). Since models affect our lives so much, we should have the right to know what drives their predictions.

In recent years, several methods for model explanations have been developed. New techniques were proposed for image data (Simonyan et al., 2013), text data (Marco Tulio Ribeiro and Sameer Singh and Carlos Guestrin, 2018), or tabular data (Molnar, 2019; Biecek, 2018). The main idea behind local explanations is to create an understandable representation of the local behavior of an underlying model. Yet, since predictive models are complex and good explanations should be simple, there is always a trade-off between fidelity and readability of explanations. Sparse explanations will be only approximations and simplifications of the underlying model and the simpler explanation, the more we lose in the fidelity This causes an increasing number of statements to avoid using explanations in high-stakes decisions (Rudin, 2019). That is why it is important to assess not only the accuracy of a model but also assess the accuracy of such explanations and reduce their uncertainty (Alvarez-Melis and Jaakkola, 2018).

In this article, we focus only on tabular data which are the most frequent in real world applications. One of the most known local explanations for tabular data are SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017), Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016), and Break Down (Staniak and Biecek, 2018). Such tools are widely adopted, but little is said about the quality of their explanations (Guidotti and Ruggieri, 2018; Yeh et al., 2019).

The idea of LIME is to fit a locally-weighted interpretable linear model in the neighborhood of a particular observation. Numerical and categorical features are converted into binary vectors for their interpretable representations. Such

interpretable representation may be a binary vector indicating the presence or absence of a word in the text classification task or super-pixel in the image classification task. For tabular data and continuous features, quantile-based discretization is performed. A linear model is then fitted on simplified binary variables sampled around the instance of interest. Therefore, the coefficients of this model can be considered as variable effects.

There are several modifications of the `LIME` (Ribeiro et al., 2016) approach, for example `live` (Staniak and Biecek, 2018) that aims for regression problems and tabular data. There are two main differences between `live` and `LIME`. In `live`, similar instances around original observation are generated by perturbing one feature at the time and original variables are used as interpretable inputs. Another variant of `LIME` is `localModel` (Staniak and Biecek, 2019). In this method, local sampling is based on decision trees and Ceteris Paribus Profiles (Kuzba et al., 2019). Categorical variables are dichotomized due to the splits of a decision tree, which models the marginal relationship between the feature and response. Numerical variables are transformed into a binary via discretization of Ceteris Paribus Profiles for observation under consideration. On the contrary to other approaches, `localModel` creates interpretable features based on a model, not only based on the distribution of underlying data.

The SHapley Additive exPlanations (`SHAP`) (Lundberg and Lee, 2017) are unification of `LIME` and several other methods for local explanations, such as `DeepLIFT` (Shrikumar et al., 2017) and layer-wise relevance propagation (Bach et al., 2015). `SHAP` is based on Shapley values, a technique used in the game theory. In this method, we calculate the contribution of variable as an average of contributions of each possible ordering of variables.

Another local method is `Break Down` (Staniak and Biecek, 2018). The main idea of `Break Down` is to generate order-specific explanations of features' contributions. It is important to consider ordering for two reasons.

- For non-additive models the order of features in explanation matters, this means that an interpretation of the model-reasoning depends on the order in which explanation is read. An example of different interpretations is presented in Section 2.4.
- Setting a proper order helps to increase the understanding of prediction. Human perception usually associates the prediction with only a few variables. Therefore, it is important to highlight only the most important features and set insignificant variables at the end of the explanation.

In the `Break Down` method, contributions of variables are calculated sequentially. The effects of consecutive variables depend on the change of expected model prediction while all previous variables are fixed. Contributions of features are presented in the form of waterfall plots. This form of visualization is appreciated and is widely used to present results in oncology clinical trials (Gillespie, 2012).Waterfall plots allow the interpretation of the explanation in the form of a scenario, in which the prediction comes from successive contributions of variables.

There are also other approaches to local explainability, which base on rules, for example, `Anchors` (High-Precision Model-Agnostic Explanations) (Marco Tulio Ribeiro and Sameer Singh and Carlos Guestrin, 2018). Anchors are rules that describe subspaces of model features where model prediction is (almost) the same. Another approach is to provide interpretable decision sets of if-then rules (Lakkaraju et al., 2016). However, there is a trade-off between the simplicity of explanation and its fidelity. Covering a complex model in the form of a small set of short rules may be too much simplification. On the other hand, many sets with complex rules will be no longer interpretable. Additionally, these two methods do not produce numerical effects of variables, which makes the effects of features incomparable.

The key issue of local explanations, such as `SHAP` and `LIME`, is that they show additive local representations, while complex models are usually non-additive. Therefore, current methods often turn out to be too imprecise and we need to find approaches that are more accurate to explain the underlying model. One of the possible ways of solving this problem is to take into account the interactions between features.

Contributions in this article are the following:

1. In Section 2, we point out three main problems with additive explanations, such as inconsistency, uncertainty, and infidelity. We identify the reasons behind this issues, such as ignoring interactions. We introduce visual representation of additive explanation uncertainty .
2. In Section 3, we offer an `iBreakDown` method to capture local interactions and generate non-additive explanations with interactions with visualization by waterfall plots.
3. In Section 4, we performed a large scale benchmark to show that our method reduces the uncertainty of explanations.
4. We have released R and Python libraries with the implementation of the `iBreakDown` algorithm, supplementary visual explanations, and plots that assess the uncertainty of explanations.

2 WHAT IS WRONG WITH ADDITIVE EXPLANATIONS?

In this section, we generate the state of the art methods for additive explanations on the example of the toy data set *Titanic*. We show inconsistency in their results, we describe a method to assess the uncertainty of them and clear the idea of local interaction in a model that may be the reason of infidelity.

2.1 A toy example

For the purpose of the example, we have trained a random forest model to predict whether a passenger survived or not, and used different additive methods to explain model's predictions for the same passenger (observation number 274).

Graphical presentation of `LIME` explanation is presented in Figure 1. Results of `SHAP` for Titanic data set generated with Python library are presented in Figure 2. Two `Break Down` explanations are presented in Figure 3. Contributions of variables differ between scenarios because each scenario relies on a different order of variables. For an additive model, regardless of the order, the contribution values are equal in each scenario. Changes in values suggest that the model is non-additive, thus, there is an interaction between variables.

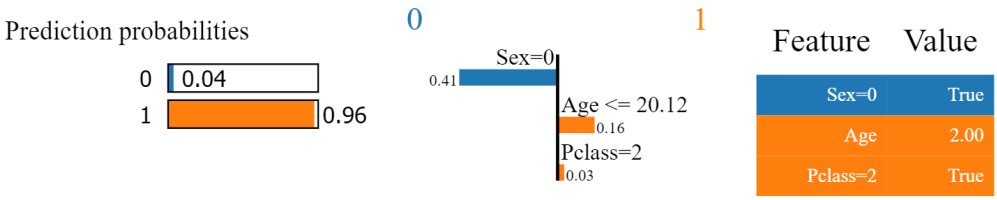

Figure 1: `LIME` explanation for an observation from Titanic data set. The underlying model is a random forest. The model predicts that the probability of survival is 0.96. Blue color indicates the reasons for passenger's death, orange indicates reasons for his survival.

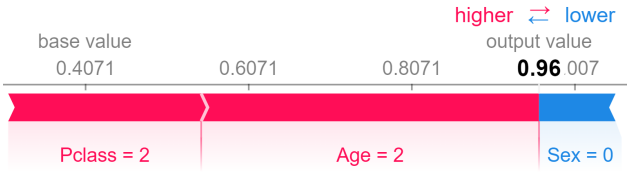

Figure 2: `SHAP` explanations for the underlying random forest model. Features, which decrease the probability of survival are blue, features which increase this probability are red. Base value and effects of variables sum up to the output value.

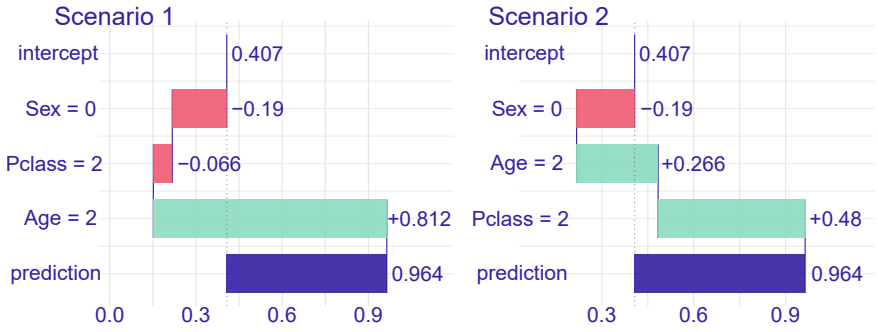

Figure 3: Two `Break Down` explanations for the same observation from Titanic data set. The underlying model is random forest. Scenarios differ due to the order of variables. Blue bar indicates the difference between the model's prediction for a particular observation and an average model prediction. Other bars show contributions of variables. Red color means a negative effect on the survival probability, while green color means a positive effect. Order of variables on the y-axis corresponds to their sequence.

Table 1: Effects of features calculated with LIME, SHAP, and two Break Down scenarios. Break Down and SHAP calculate feature contributions which sum up to model prediction, while LIME calculates only relative importance.

	Feature Effect		
Method	Age	Sex	Pclass
LIME	-0.16	0.41	-0.03
SHAP	0.41	-0.09	0.24
Break Down, Scenario 1	0.81	-0.19	-0.07
Break Down, Scenario 2	0.27	-0.19	0.48

2.2 Inconsistency of additive explanations

The common approaches to local explanations consider the effect of each variable separately. However, when interactions occur in the model, relationships between variables should be also taken into account. Omitting influence of interactions causes that we do not only lose a part of the information about the effects of the variables, we also add undesired randomness in the evaluation of these effects.

Values of feature importance LIME and contributions for SHAP and Break Down are summarized in Table 1. Size of effects differs between methods, there are even differences in the judgment of whether the impact is positive or negative. It is not clear which explanation should be considered as the most reliable.

LIME approximates the underlying model with a linear model, while SHAP averages across all possible combinations of variable contributions. Break Down calculate contributions based on the specified order of variables. For additive models, the results of LIME, SHAP, and Break Down would be similar. The cause of differences between explanations can be the interaction of variables. What is more, in Figure 3, we see that values of contributions even differ for different orders of variables. The differences between Break Down scenarios also leads to the conclusion that the reason for inconsistency can be the interaction between variables. Visualization of different variable orders in Break Down method allowed to identify the source of differences in LIME and SHAP predictions and thus better-explained model prediction. However, interactions are not included in any of these three methods, thus we should not rely on their results.

Detecting interactions would reduce the uncertainty and increase the trust in explanation. One approach to capturing interactions may be analyzing different orders of features in the Break Down algorithm. However, comparing many scenarios is highly ineffective. As the number of variables increases, the number of cases to review increases factorially. The solution to this problem is iBreakDown, a local explanation method that captures interactions. We introduce iBreakDown in Section 3.

2.3 Uncertainty of additive explanations

When generating an explanation for a model, it is important to know how much we can rely on it. Therefore, the uncertainty of the explanation also should be assessed. We propose a methodology for assessing the uncertainty of Break Down explanations. The idea is to use bootstrapping to generate a sample of different explanations and measure the stability of contribution values.

In this setup, we have one fixed underlying model and one baseline explanation of this model. The first step is to generate m random samples of variables orders. Next, we generate a Break Down explanation concerning each sampled variables order. As a result, we obtain m new explanations. The procedure of computing uncertainty of explanation is presented in Algorithm 1.

The example summary plot of bootstrapping explanations is presented in Figure 4. Uncertainty is realized as a variation of contribution values between explanations. Error bars show the range of contribution values for explanations generated on different variable orders. Widths of error bars indicate uncertainty of variables' contributions. The wider bar, the less certain contribution is.

We impose randomness of explanations by forcing different variable orders while the model and explained instance are fixed. The whole variability is the result of the uncertainty of the explanation. What is more, as SHAP method is average over all Break Down scenarios and SHAP is the unification of different explanations, bars show also the uncertainty of the SHAP and many other explanations.

Since Break Down is an additive method of explanation, the high variability of contribution, realized by wide error bars, is related to the occurrence of interaction. To reduce the uncertainty of explanation, the interaction should be taken into account.

Algorithm 1 Explanation level uncertainty

Input: $X_{n \times p}$ - data; f - model; x^* - new observation
for k in $\{1, 2, ..., K\}$ **do**
    sample $path_k$ of features as random permutation
    Calculate explanations $[\Delta_1^{*,k}, ..., \Delta_p^{*,k}]$ of model f, observation x^*, data set X, and $path_k$
A matrix of contributions $\Delta_i^{*,j}$
Shappley additive contribution for feature i is an average of vector $[\Delta_i^{*,1}, ..., \Delta_i^{*,K}]$
Explanation level uncertainty for feature i is interquartile range of vector $[\Delta_i^{*,1}, ..., \Delta_i^{*,K}]$

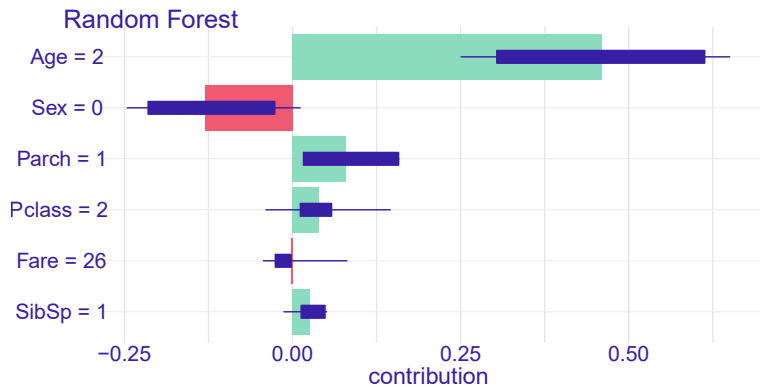

Figure 4: The summary of contribution values for `Break Down` explanations generated for the random forest model for one observation. Green and red bars correspond to contribution values of baseline explanation. Thin blue error bars represent range of contribution values for bootstrapped 100 explanations. Thick blue error bars shows first quartile and third quartile.

2.4 Infidelity of additive explanations

Now, we will broaden the example for Titanic data and explain the interaction in the underlying model. We also showing an `iBreakDown` explanation.

In our example, the training data set consists of 4 variables.

- Survival - binary variable indicates whether passenger survived, 1 for survival and 0 for death.
- Age - numerical variable, age in years.
- Sex - binary variable, 0 for male and 1 for female.
- PClass - categorical variable, ticket class, 1, 2, or 3.

We explain the model's prediction for a 2-year old boy that travels in the second class. The model predicts survival with a probability of 0.964. We would like to explain this probability and understand which factors drive this prediction. In Figure 3, we showed two `Break Down` explanations. Each of them may be interpreted differently.

Scenario 1: The passenger is a boy, and this feature alone decreases the chances of survival. He traveled in the second class which also lower survival probability. Yet, he is very young, which makes odds higher. The reasoning behind such an explanation on this level is that most passengers in the second class are adults, therefore a kid from the second class has high chances of survival.

Scenario 2: The passenger is a boy, and this feature alone decreases survival probability. However, he is very young, therefore odds are higher than adult men. Explanation in the last step says that he traveled in the second class, which make odds of survival even more higher. The interpretation of this explanation is that most kids are from the third class and being a child in the second class should increase chances of survival.

Note that the effect of *the second class* is negative in explanations for scenario 1 but positive in explanations for scenario 2. Two interpretations of the above scenarios imply the existence of an interaction between age and ticket class. The algorithm introduced in the previous section founds this interaction. Corresponding explanation is presented in Figure 5.

Scenario 3 (with interactions): The passenger is a boy in second class, which increases the chance of survival because the effect of age depends on passenger class.

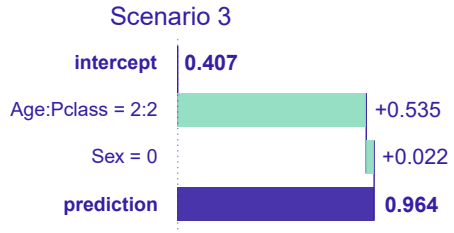

Figure 5: The `iBreakDown` explanation of the non-additive random forest model for 2-year old boy that travels in the second class. Bars show contributions of feature Sex and interaction between Age and Pclass.

The inclusion of the interaction in the explanation allowed to produce different reasoning, that better reflects the way how the underlying model predicts the probability of survival.

3 HOW TO EXPLAIN INTERACTIONS?

If the uncertainty of model explanations is linked with the presence of interactions, then we have to include interactions to model explanations. This way we will have more stable and reliable explanations. In this section, we introduce a novel methodology for the identification of interactions in instance level explanations. The algorithm works in a similar vein with `SHAP` or `Break Down` but is not restricted to additive effects. The intuition is the following:

1. Calculate a single-step additive contribution for each feature.
2. Calculate a single-step contribution for every pair of features. Subtract additive contribution to assess the interaction specific contribution.
3. Order interaction effects and additive effects in a list that is used to determine sequential contributions.

This simple intuition may be generalized into higher order interactions.

Let $f : \mathbb{X} \to \mathbb{R}$ be a predictive model being explained and $x^* \in \mathbb{X}$ be an observation to explain. For the sake of simplicity, we consider a univariate model output, more suited for classification or regression, but every step can be easily generalized into multiclass classification or multivariate regression.

For a feature x_i we may define a single-step contribution.

$$\Delta_i = score_i(f, x^*) = \mathbb{E}[f(x)|x_i = x_i^*] - \mathbb{E}[f(x)]. \tag{1}$$

Expected model prediction $\mathbb{E}[f(x)]$ is sometimes called baseline or intercept and may be denoted as $\Delta_\varnothing$.

Expected value $\mathbb{E}[f(x)|x_i = x_i^*]$ corresponds to an average prediction of a model f if feature x_i is fixed on x_i^* coordinate from the observation to explain x^*. Δ_i measures a naive single-step local variable importance, it indicates how much the average prediction of model f changes if feature x_i is set on x_i^*.

Algorithm 2 is a procedure for calculation of Δ_i, i.e. single-step contributions for each feature.

For a pair of variables x_i, x_j we introduce a single-step contribution as

$$\Delta_{ij} = score_{i,j}(f, x^*) = \mathbb{E}[f(x)|x_i = x_i^*, x_j = x_j^*] - \mathbb{E}[f(x)]. \tag{2}$$

In similar fashion we introduce a corresponding interaction specific contribution as

$$\Delta_{ij}^I = \mathbb{E}[f(x)|x_i = x_i^*, x_j = x_j^*] - \mathbb{E}[f(x)|x_i = x_i^*] - \mathbb{E}[f(x)|x_j = x_j^*] + \mathbb{E}[f(x)]. \tag{3}$$

It is an equivalent to

$$\Delta_{ij}^I = \mathbb{E}[f(x)|x_i = x_i^*, x_j = x_j^*] - score_i(f, x^*) - score_j(f, x^*) - \mathbb{E}[f(x)] = \Delta_{ij} - \Delta_i - \Delta_j. \tag{4}$$

A value of $\mathbb{E}[f(x)|x_i = x_i^*, x_j = x_j^*]$ is an average model output if feature x_i and x_j are fixed on x_i^* and x_j^* respectively. Δ_{ij}^I is the difference between collective effect of variables x_i and x_j denoted as Δ_{ij} and their additive effects Δ_i and Δ_j. Therefore, Δ_{ij}^I measures the importance of local lack-of-additivnes (aka. interaction) between features i and j. For additive models Δ_{ij}^I is small for any i, j.

Algorithm 3 is a procedure for calculation of Δ_{ij} and Δ_{ij}^I, i.e. single-step contributions and interactions for each pair.

Calculating Δ_i for each variable is Step 1, computing Δ_{ij}^I for each pair of variables is Step 2. Note that contributions Δ_i do not sum to the final model prediction. We only use them to determine the order of features in which the instance shall be explained.

We need to provide one more symbol, that corresponds to the added contribution of feature i to the set of features J.

$$\Delta_{i|J} = \mathbf{E}[f(X)|x_{J\cup\{i\}} = x^*_{J\cup\{i\}}] - \mathbf{E}[f(X)|x_J = x^*_J] = \Delta_{J\cup\{i\}} - \Delta_J. \tag{5}$$

And for pairs of features

$$\Delta_{ij|J} = \mathbf{E}[f(X)|x_{J\cup\{i,j\}} = x^*_{J\cup\{i,j\}}] - \mathbf{E}[f(X)|x_J = x^*_J] = \Delta_{J\cup\{i,j\}} - \Delta_J. \tag{6}$$

Once the order of single-step importance is determined based on Δ_i and Δ_{ij}^I scores, the final explanation is the attribution to the sequence of $\Delta_{i|J}$ scores. These contributions for all p features sum up to the model predictions, because

$$\Delta_{1,2...p} = f(x^*) - E[f(X)].$$

Algorithm 4 applies consecutive conditioning to ordered variables. It consists of setting a path due to the calculated effects, then calculating contributions.

This approach can be generalized to interactions between any number of variables.

The introduced method takes into account the interactions between variables. A large difference between the sum of consecutive effects of features and the effect of a pair of features indicates interaction. There is a similar idea of calculating differences between sum of independent effects of variables and join effect to calculate `SHAP` interaction values (Lundberg et al., 2018). However, their approach is based on averaging contributions over all possible ordering of features. Such an approach makes it hard to assess the uncertainty or stability of the explanation.

Algorithm 2 Single-step contributions of features

1: **Input:** $X_{n\times p}$ - data; f - model; x^* - new observation
2: Calculate average model response
3: $\Delta_\varnothing = mean(f(X))$
4: **for** i in $\{1, 2, ...p\}$ **do**
5: Calculate contribution of the i-th feature
6: $avg\_yhat = mean(f(X_{x_i=x_i^*}))$
7: $\Delta_i = avg\_yhat - \Delta_\varnothing$
8: $[\Delta_1, ..., \Delta_p]$ contains contributions of features

4 HOW FREQUENT INTERACTIONS ARE IN REAL DATA SETS?

We have performed the `iBreakDown` method on several classification data sets. The aim of the experiment was to justify the need to include interactions in local explanations. We address the following questions: (1) Are the additive explanation methods faithful enough? (2) Are the interactions useful for local explanations?

4.1 Setup of the benchmark on OpenML

We have performed experiment on 28 data sets from OpenML100 (Bischl et al., 2017) collection of data sets. We have selected data sets for binary classification that do not contain missing values and consist of less than 100 features. For

Algorithm 3 Single-step contributions of pair of features

Input: $X_{n \times p}$ - data; f - model; x^* - new observation; Δ_i - vector of single-step contributions.
for i in $\{1, 2, ...p\}$ **do**
    for j in $\{1, 2, ...p\}/\{i\}$ **do**
        Calculate contribution of pair i,j.
        $avg\_yhat = mean(f(X_{x_i = x_i^*, x_j = x_j^*}))$
        $\Delta_{ij} = avg\_yhat - \Delta_{\varnothing}$
        $\Delta^I_{ij} = \Delta_{ij} - \Delta_i - \Delta_j$
Δ^I contains a matrix with interaction contributions for pairs of features (Δ^I_{ij}).

Algorithm 4 Sequential explanations

Input: $X_{n \times p}$ - data; f - model; x^* - new observation; $[\Delta_1, ..., \Delta_p]$ - vector of single-step feature contributions; Δ^I - table of single-step feature interactions (Δ^I_{ij});
Calculate Δ^* which is a sorted union of Δ_i and Δ^I_{ij} ordered by absolute values of elements.
$features$ - a table of features and pairs in order corresponding to Δ^*.
$open = \{1, 2, ..., p\}$
for $candidates$ in $features$ **do**
    if $candidates$ in $open$ **then**
        $path = append(path, candidates)$
        $open = setdiff(open, candidates)$
        $yhat = mean(f(X | x_{\neg open} = x^*_{\neg open}))$
        $avg\_yhats = append(avg\_yhats, yhat)$
Explanation order is determined in the $path$ vector.
$history = \varnothing$
for k in $\{1, 2, ...length(path)\}$ **do**
    I is a single variable or pair of variables
    $I = path[k]$
    $history = history \cup I$
    $attribution[i] = \Delta_{I|history} = \Delta_{I \cup history} - \Delta_{history} = avg\_yhats[k] - avg\_yhats[k-1]$
Explanations are in the $attribution$ vector.

each data set, we have fitted random forest and three gradient boosting machines (GBM) models with maximum depth of trees equal 1, 2, and 3. A depth of 1 implies an additive model, a depth of 2 implies a model with up to 2-way interactions, and a depth of 3 implies a model with up to 3-way interactions. Performance of models for all of the data sets is presented in Figure 6. AUC was calculated for the first train and test split defined in each OpenML task. For the benchmark, we have taken 50 observations from each data set (28) and for each model (4), then we have calculated iBreakDown explanations of these observations. That gave us in total $50 * 28 * 4 = 5600$ explanations.

4.2 Results

Results of the experiment are in Tables 3 and 2. For GBM models with interactions and random forest, interactions were identified in most of the tasks. One can see a dependency that the more complex interactions included in the model, the more local interactions were detected by `iBreakDown`.

Let us focus on the task 3493, due to the fact that AUC of the models trained on this task was significantly different (see Figure 6). Including interaction in the models increased performance, therefore we expected to identify interactions in explanations. Indeed, explanations of models with interactions contains at least one interaction. Figure 7 consists of models' explanations for the same observation. An additive GBM model do not have any interaction in the Break Down path, while more complex models include interactions in their paths. The detection of different interactions is due to the fact that the models could learn different relationships between the variables.

According to the results in Table 3, the `iBreakDown` method detected local interactions, although the models under consideration were additive. We find the reason for this in the correlations in the data that are reflected in the detection of interaction. Taking into account the overall results and the above example, we can answer the questions stated in the beginning of this section.

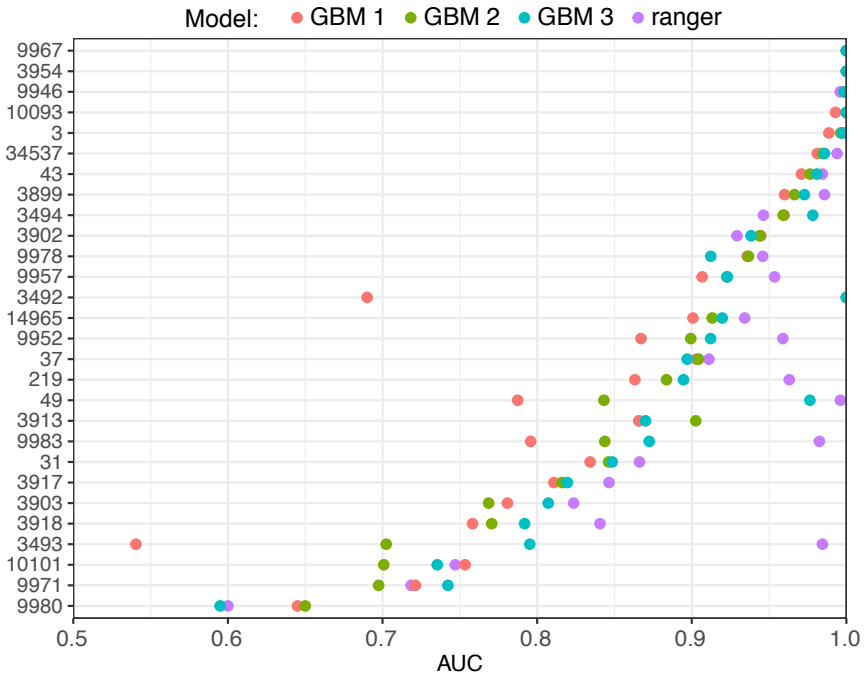

Figure 6: AUC for random forest, GBM with depths of trees equal 1, 2, and 3. Numbers on the x-axis are tasks from the OpenML data base (https://www.openml.org/) (Vanschoren et al., 2013), each task corresponds to a different data set.

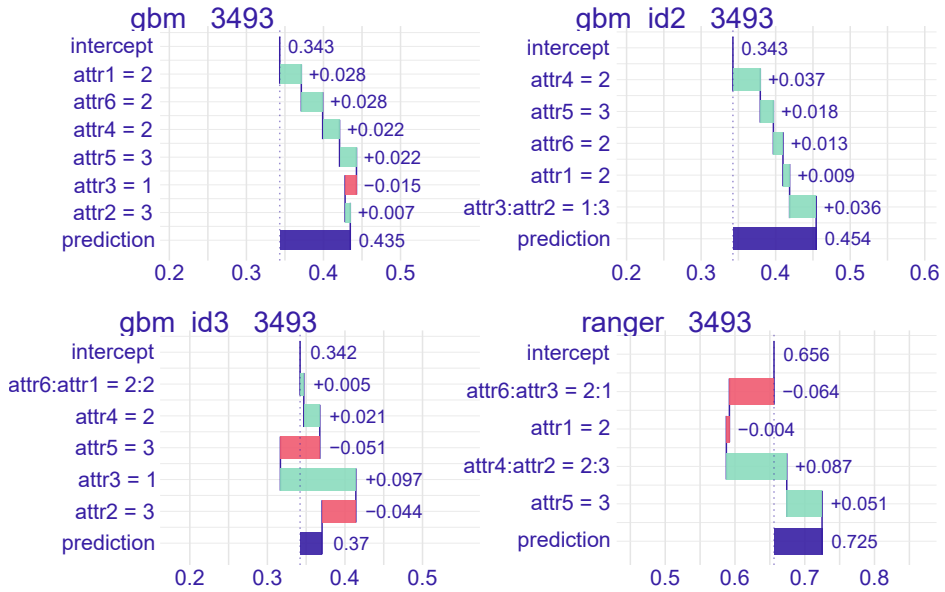

Figure 7: The iBreakDown explanations for one of the observations from the data set corresponding to the task 3493. Each plot corresponds to different model.

(1) Are the additive methods faithful enough? Many of the explanations from the benchmark consist of interactions. Therefore, usage of additive explanations would strongly simplify the explanations, which would make them less accurate. As a result, there would be considerable uncertainty in these explanations.

(2) Are the interactions useful for local explanations? Local interactions included in the explanations allow to grab more nuances of the model, which both increase the trust in the predictions and reduce the uncertainty. The experiment

Table 2: Interactions identified by gradient boosting machines with interaction depths 2 and 3. Columns correspond to the number of interactions identified in iBreakDown path. For example, for task 3 and model GBM 2, there are 49 explanations that do not contain interactions and 1 explanation with 1 interaction.

	GBM, 2 depth interactions					GBM, 3 depth interactions				
Task	0	1	2	3	4+	0	1	2	3	4+
3	49	1	0	0	0	50	0	0	0	0
31	33	16	1	0	0	37	10	3	0	0
37	50	0	0	0	0	42	8	0	0	0
43	50	0	0	0	0	48	2	0	0	0
49	48	2	0	0	0	40	10	0	0	0
219	48	2	0	0	0	47	3	0	0	0
3492	0	50	0	0	0	0	37	13	0	0
3493	32	18	0	0	0	0	39	11	0	0
3494	50	0	0	0	0	50	0	0	0	0
3899	48	2	0	0	0	48	2	0	0	0
3902	45	5	0	0	0	21	20	6	3	0
3903	23	21	5	1	0	0	3	8	35	4
3913	7	32	8	2	1	26	7	10	3	4
3917	21	14	15	0	0	10	13	26	1	0
3918	28	15	4	3	0	16	27	6	0	1
3954	50	0	0	0	0	0	0	0	0	50
9946	1	24	20	4	1	0	16	25	9	0
9952	45	5	0	0	0	36	11	3	0	0
9957	40	10	0	0	0	28	19	3	0	0
9967	50	0	0	0	0	50	0	0	0	0
9972	35	14	1	0	0	30	17	3	0	0
9978	7	35	7	1	0	10	24	11	4	1
9980	33	13	4	0	0	31	16	3	0	0
9983	39	10	1	0	0	26	20	4	0	0
10093	39	11	0	0	0	39	11	0	0	0
10101	40	10	0	0	0	40	10	0	0	0
14965	0	48	2	0	0	0	40	8	2	0
34537	23	25	2	0	0	25	25	0	0	0

show that interactions were detected for many observations. The `iBreakDown` method allows us to better explain models' predictions for these instances.

5 CONCLUSION

In this article, examined the behaviour of the common local additive explanations, such as `SHAP`, `LIME`, and `Break Down`. For the same random forest model, each method generated inconsistent explanations, sometimes even with opposite signs. As we showed, some of the uncertainty and infidelity of the explanations is linked with the lack of additives in the model, which cannot be grasped by the additive explanations. Simple explanations may omit some important parts of model behavior, therefore we introduced the procedure to measure and visualize this type of the uncertainty.

To solve the problems with uncertainty and infidelity of explanations, we introduced a new `iBreakDown` method, which identify local interactions and generates not-only-additive explanations. The theoretical backbone of this algorithm is similar to `SHAP` and `Break Down` methods, yet, in contrast to them we consider also pairwise interactions. It should be noted that for simple linear models interactions may be included directly in model terms, yet we do not have that control in case of more complex models. Such models grab interactions due to their elastic structure and we showed how such interactions can be identified and presented.

Finally, we performed the `iBreakDown` on the several data sets and showed that in majority of the explanations our method detected local interactions, therefore additive explanations were not fiddle enough.

For tabular data, most of the local explanation methods are additive. Applying them to non-additive models increase the uncertainty of such explanations. Tools in the area of Interpretable Machine Learning are developed to explain complex

Table 3: Interactions identified by random forest and gradient boosting machines with trees of depth 1.

	Random forest					GBM with trees of depth 1				
Task	0	1	2	3	4+	0	1	2	3	4+
3	21	23	6	0	0	50	0	0	0	0
31	36	13	1	0	0	50	0	0	0	0
37	26	18	5	1	0	50	0	0	0	0
43	47	3	0	0	0	0	0	0	0	50
49	6	23	19	2	0	50	0	0	0	0
219	18	22	8	2	0	0	50	0	0	0
3492	0	37	13	0	0	50	0	0	0	0
3493	0	7	39	4	0	50	0	0	0	0
3494	50	0	0	0	0	50	0	0	0	0
3899	31	17	2	0	0	48	2	0	0	0
3902	28	16	5	0	1	50	0	0	0	0
3903	36	14	0	0	0	50	0	0	0	0
3913	23	15	10	2	0	47	3	0	0	0
3917	50	0	0	0	0	50	0	0	0	0
3918	31	14	5	0	0	50	0	0	0	0
3954	26	24	0	0	0	50	0	0	0	0
9946	36	11	3	0	0	8	35	7	0	0
9952	8	34	8	0	0	50	0	0	0	0
9957	45	5	0	0	0	50	0	0	0	0
9967	21	18	7	4	0	50	0	0	0	0
9971	24	22	3	1	0	50	0	0	0	0
9978	9	10	16	14	1	50	0	0	0	0
9980	15	30	5	0	0	48	2	0	0	0
9983	5	16	18	6	5	50	0	0	0	0
10093	37	12	1	0	0	44	6	0	0	0
10101	32	18	0	0	0	50	0	0	0	0
14965	11	28	11	0	0	0	0	0	0	50
34537	13	17	17	3	0	50	0	0	0	0

black-box models. We cannot assume that such complex models will be additive, we should expect, identify and handle interactions in these models. A solution to handle interactions and explain uncertainty linked with feature contributions is the `iBreakDown` algorithm.

5.1 Future work

The `iBreakDown` method identifies interactions and measures their contributions. However, the main effects of variables and interaction between them are currently presented as a single value. It would be desirable to separate the main effects and the contribution of an interaction and present deeper visual clues that help to understand the role of interaction.

Presented approach for handling the explanation level uncertainty also needs further examination. The inclusion of interactions in the explanation improves its certainty, yet at the same time, explanations may become more difficult to understand than the additive representations. It is a field for extensive cognitive studies of visual presentation of explanations.

5.2 Software

The Break Down with interactions algorithm and plots are implemented and available as open source R package `iBreakDown` [1] and Python library `piBreakDown` [2]. R package `iBreakDown` provides also interactive versions of plots implemented in D3.js JavaScript library and diagnostic plots for Break Down explanations.

Code that generates examples included in this article and performs experiment can be found in the GitHub repository: https://github.com/agosiewska/iBreakDown_article.

[1] https://github.com/ModelOriented/iBreakDown

[2] https://github.com/ModelOriented/piBreakDown

6 ACKNOWLEDGEMENTS

We would like to acknowledge and thank Hubert Baniecki for his valuable contribution to the development of the `iBreakDown` package, especially the interactive plots. We would like to thank Mateusz Staniak, Anna Gierlak and Katarzyna Kobylińska for valuable discussions.

This work was supported by the Polish National Science Centre under Opus Grant number 2017/27/B/ST6/01307 and 2016/21/B/ST6/02176.

References

D. Alvarez-Melis and T. S. Jaakkola. On the Robustness of Interpretability Methods. *arXiv e-prints*, art. arXiv:1806.08049, Jun 2018.

S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):1–46, 07 2015. doi: 10.1371/journal.pone.0130140. URL https://doi.org/10.1371/journal.pone.0130140.

P. Biecek. DALEX: Explainers for Complex Predictive Models in R. *Journal of Machine Learning Research*, 19(84): 1–5, 2018. URL http://jmlr.org/papers/v19/18-416.html.

B. Bischl, G. Casalicchio, M. Feurer, F. Hutter, M. Lang, R. G. Mantovani, J. N. van Rijn, and J. Vanschoren. OpenML benchmarking suites and the OpenML100. 2017.

L. Edwards and M. Veale. Enslaving the algorithm: From a "right to an explanation" to a "right to better decisions"? *IEEE Security Privacy*, 16(3):46–54, May 2018. doi: 10.1109/MSP.2018.2701152.

N. Gill and P. Hall. *An Introduction to Machine Learning Interpretability*. O'Reilly Media, Incorporated, 2018. ISBN 9781492033158. URL https://www.oreilly.com/library/view/an-introduction-to/9781492033158/.

T. W. Gillespie. Understanding waterfall plots. *Journal of the advanced practitioner in oncology*, Mar 2012. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4093310/.

R. Guidotti and S. Ruggieri. On The Stability of Interpretable Models. *arXiv e-prints*, art. arXiv:1810.09352, Oct 2018.

M. Kuzba, E. Baranowska, and P. Biecek. pyCeterisParibus: explaining Machine Learning models with Ceteris Paribus Profiles in Python. *Journal of Open Source Software*, 4(37), 2019.

H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1675–1684, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939874. URL http://doi.acm.org/10.1145/2939672.2939874.

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

S. M. Lundberg, G. G. Erion, and S.-I. Lee. Consistent Individualized Feature Attribution for Tree Ensembles. *arXiv e-prints*, art. arXiv:1802.03888, Feb 2018.

Marco Tulio Ribeiro and Sameer Singh and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*, 2018. URL https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16982.

C. Molnar. *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/.

C. O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown Publishing Group, 2016. ISBN 0553418815, 9780553418811.

M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144, 2016.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, May 2019. doi: 10.1038/s42256-019-0048-x. URL https://doi.org/10.1038/s42256-019-0048-x.

A. Shrikumar, P. Greenside, and A. Kundaje. Learning Important Features Through Propagating Activation Differences. *arXiv e-prints*, art. arXiv:1704.02685, Apr 2017.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv e-prints*, art. arXiv:1312.6034, Dec 2013.

M. Staniak and P. Biecek. Explanations of Model Predictions with live and breakDown Packages. *The R Journal*, 10(2): 395–409, 2018. doi: 10.32614/RJ-2018-072. URL https://doi.org/10.32614/RJ-2018-072.

M. Staniak and P. Biecek. *LIME-based Explanations With Interpretable Inputs Based on Ceteris Paribus Profiles*, 2019. URL https://modeloriented.github.io/localModel/. R package version 0.3.10.

J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198. URL http://doi.acm.org/10.1145/2641190.2641198.

S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. *arXiv e-prints*, art. arXiv:1711.00399, Nov 2017.

C.-K. Yeh, C.-Y. Hsieh, A. Sai Suggala, D. Inouye, and P. Ravikumar. On the (In)fidelity and Sensitivity for Explanations. *arXiv e-prints*, art. arXiv:1901.09392, Jan 2019.