7.3 Feature Interaction

When features interact with each other in a prediction model, the prediction cannot be expressed as the sum of the feature effects, because the effect of one feature depends on the value of the other feature. Aristotle's predicate "The whole is greater than the sum of its parts" applies in the presence of interactions.

整体大于部分的总和

7.3.1 Feature Interaction?

$y = c + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2$

If a machine learning model makes a prediction based on two features, we can decompose the prediction into four terms: a constant term, a term for the first feature, a term for the second feature and a term for the interaction between the two features.
The interaction between two features is the change in the prediction that occurs by varying the features after considering the individual feature effects.

For example, a model predicts the value of a house, using house size (big or small) and location (good or bad) as features, which yields four possible predictions:

Location	Size	Prediction
good	big	300,000
good	small	200,000
bad	big	250,000
bad	small	150,000

We decompose the model prediction into the following parts: A constant term (150,000), an effect for the size feature (+100,000 if big; +0 if small) and an effect for the location (+50,000 if good; +0 if bad). This decomposition fully explains the model predictions. There is no interaction effect, because the model prediction is a sum of the single feature effects for size and location. When you make a small house big, the prediction always increases by 100,000, regardless of location. Also, the difference in prediction between a good and a bad location is 50,000, regardless of size.

Let's now look at an example with interaction:

Location	Size	Prediction
good	big	400,000
good	small	200,000
bad	big	250,000
bad	small	150,000

We decompose the prediction table into the following parts: A constant term (150,000), an effect for the size feature (+100,000 if big, +0 if small) and an effect for the location (+50,000 if good, +0 if bad). For this table we need an additional term for the interaction: +100,000 if the house is big and in a good location. This is an interaction between size and location, because in this case the difference in prediction between a big and a small house depends on the location.

One way to estimate the interaction strength is to measure how much of the variation of the prediction depends on the interaction of the features. This measurement is called H-statistic, introduced by Friedman and Popescu (2008)[10].

评估交互影响力的一个方法是测量预测值到底受交互的影响变化有多大

H-statistic

7.3.2 Theory: Friedman's H-statistic

We are going to deal with two cases: First, a two-way interaction measure that tells us whether and to what extent two features in the model interact with each other; second, a total interaction measure that tells us whether and to what extent a feature interacts in the model with all the other features. In theory, arbitrary interactions between any number of features can be measured, but these two are the most interesting cases.

① The intereaction between two features.

② The intereaction between a feature and the rest of the features.

If two features do not interact, we can decompose the partial dependence function as follows (assuming the partial dependence functions are centered at zero):

$$PD_{jk}(x_j, x_k) = PD_j(x_j) + PD_k(x_k)$$

where $PD_{jk}(x_j, x_k)$ is the 2-way partial dependence function of both features and $PD_j(x_j)$ and $PD_k(x_k)$ the partial dependence functions of the single features.

Likewise, if a feature has no interaction with any of the other features, we can express the prediction function $\hat{f}(x)$ as a sum of partial dependence functions, where the first summand depends only on j and the second on all other features except j:

How to get $PD_{-j}(x_{-j})$?

$$\hat{f}(x) = PD_j(x_j) + PD_{-j}(x_{-j})$$

where $PD_{-j}(x_{-j})$ is the partial dependence function that depends on all features except the j-th feature.

This decomposition expresses the partial dependence (or full prediction) function without interactions (between features j and k, or respectively j and all other features). In a next step, we measure the difference between the observed partial dependence function and the decomposed one without interactions. We calculate the variance of the output

measure the difference between the partial dependence function and the decomposed one without interactions.

[10] Friedman, Jerome H, and Bogdan E Popescu. "Predictive learning via rule ensembles." The Annals of Applied Statistics. JSTOR, 916–54. (2008).

of the partial dependence (to measure the interaction between two features) or of the entire function (to measure the interaction between a feature and all other features). The amount of the variance explained by the interaction (difference between observed and no-interaction PD) is used as interaction strength statistic. The statistic is 0 if there is no interaction at all and 1 if all of the variance of the PD_{jk} or $\hat{f}$ is explained by the sum of the partial dependence functions. An interaction statistic of 1 between two features means that each single PD function is constant and the effect on the prediction only comes through the interaction. The H-statistic can also be larger than 1, which is more difficult to interpret. This can happen when the variance of the 2-way interaction is larger than the variance of the 2-dimensional partial dependence plot.

被交互所解释的方差数量
可以被当作交互影响力的统计

interaction statistic is 1
means their single PD is
Constant and the effected
prediction only comes from
interaction.

Mathematically, the H-statistic proposed by Friedman and Popescu for the interaction between feature j and k is:

$$H^2_{jk} = \frac{\sum_{i=1}^n \left[PD_{jk}(x_j^{(i)}, x_k^{(i)}) - PD_j(x_j^{(i)}) - PD_k(x_k^{(i)})\right]^2}{\sum_{i=1}^n PD^2_{jk}(x_j^{(i)}, x_k^{(i)})}$$

The same applies to measuring whether a feature j interacts with any other feature:

$$H^2_j = \frac{\sum_{i=1}^n \left[\hat{f}(x^{(i)}) - PD_j(x_j^{(i)}) - PD_{-j}(x_{-j}^{(i)})\right]^2}{\sum_{i=1}^n \hat{f}^2(x^{(i)})}$$

The H-statistic is expensive to evaluate, because it iterates over all data points and at each point the partial dependence has to be evaluated which in turn is done with all n data points. In the worst case, we need 2n^2 calls to the machine learning models predict function to compute the two-way H-statistic (j vs. k) and 3n^2 for the total H-statistic (j vs. all). To speed up the computation, we can sample from the n data points. This has the disadvantage of increasing the variance of the partial dependence estimates, which makes the H-statistic unstable. So if you are using sampling to reduce the computational burden, make sure to sample enough data points.

计算它的代价很大：$2n^2$

take samples, but make
H-statistic unstable.

Friedman and Popescu also propose a test statistic to evaluate whether the H-statistic differs significantly from zero. The null hypothesis is the absence of interaction. To generate the interaction statistic under the null hypothesis, you must be able to adjust the model so that it has no interaction between feature j and k or all others. This is not possible for all types of models. Therefore this test is model-specific, not model-agnostic, and as such not covered here.

必须要可以修改 model ?

so it is model-specific、
can we do this by masking
the feature with empty input?

The interaction strength statistic can also be applied in a classification setting if the prediction is a probability.

7.3.3 Examples

Let us see what feature interactions look like in practice! We measure the interaction strength of features in a support vector machine that predicts the number of rented bikes based on weather and calendrical features. The following plot shows the feature interaction H-statistic:

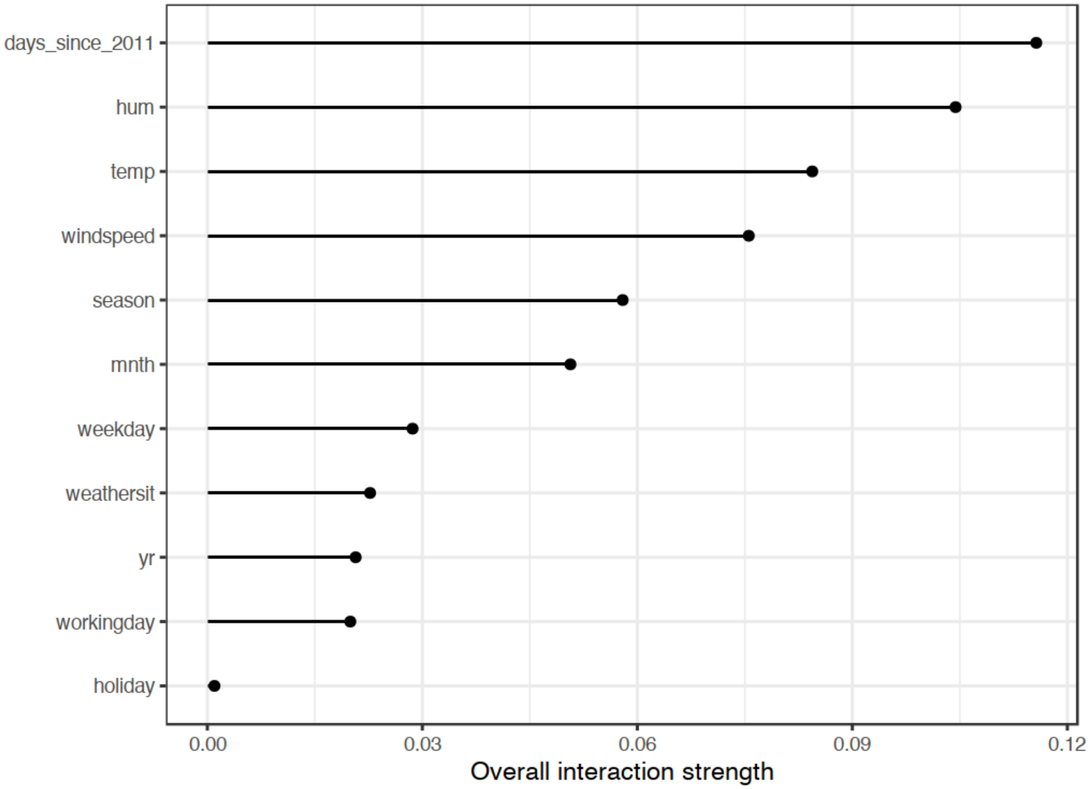

Figure 7.19: The interaction strength (H-statistic) for each feature with all other features for a support vector machine predicting bicycle rentals. Overall, the interaction effects between the features are very weak (below 10% of variance explained per feature).

each feature with all other features

In the next example, we calculate the interaction statistic for a classification problem. We analyze the interactions between features in a random forest trained to predict cervical cancer, given some risk factors.

→ Fig 7.20

After looking at the feature interactions of each feature with all other features, we can select one of the features and dive deeper into all the 2-way interactions between the selected feature and the other features.

→ Fig 7.21

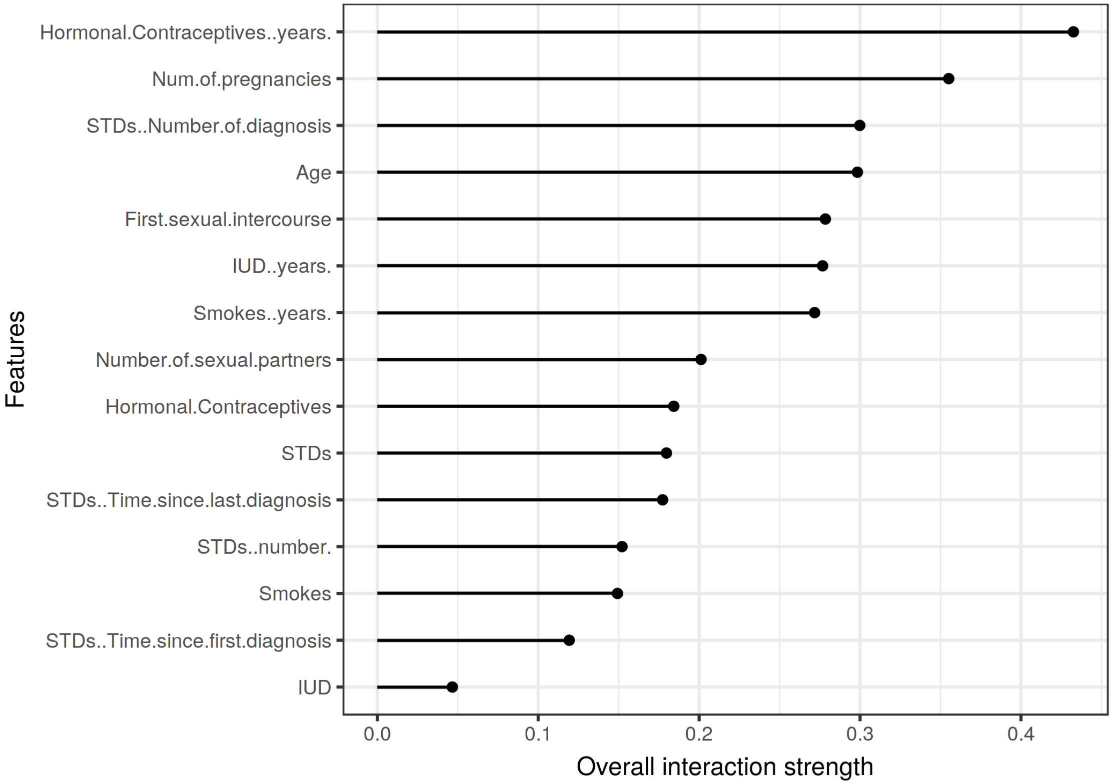

Figure 7.20: The interaction strength (H-statistic) for each feature with all other features for a random forest predicting the probability of cervical cancer. The years on hormonal contraceptives has the highest relative interaction effect with all other features, followed by the number of pregnancies.

7.3.4 Advantages

The interaction H-statistic has an **underlying theory** through the partial dependence decomposition.

The H-statistic has a **meaningful interpretation**: The interaction is defined as the share of variance that is explained by the interaction.

Since the statistic is **dimensionless**, it is comparable across features and even across models.

The statistic **detects all kinds of interactions**, regardless of their particular form.

With the H-statistic it is also possible to analyze arbitrary **higher interactions** such as the interaction strength between 3 or more features.

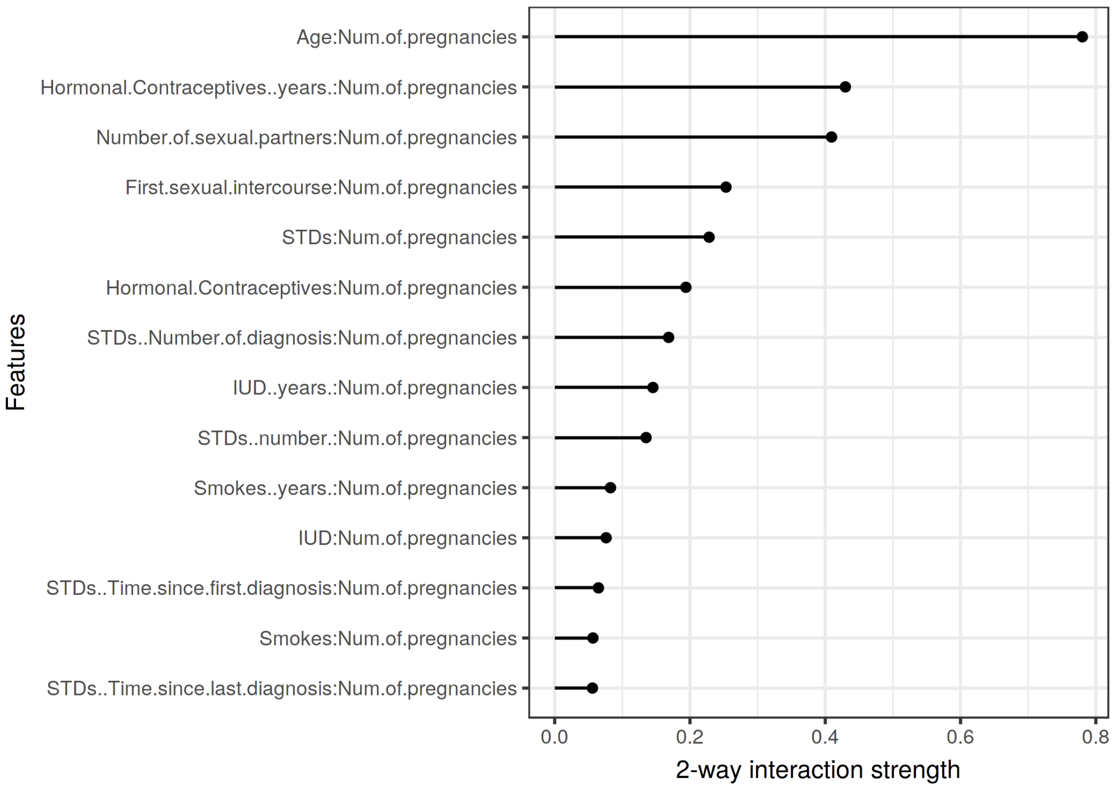

Figure 7.21: The 2-way interaction strengths (H-statistic) between number of pregnancies and each other feature. There is a strong interaction between the number of pregnancies and the age.

7.3.5 Disadvantages

The first thing you will notice: The interaction H-statistic takes a long time to compute, because it is **computationally expensive**.

计算成本高.

The computation involves estimating marginal distributions. These **estimates also have a certain variance** if we do not use all data points. This means that as we sample points, the estimates also vary from run to run and the **results can be unstable**. I recommend repeating the H-statistic computation a few times to see if you have enough data to get a stable result.

如果没有用所有数据计算的话，
H-statistic 会很不稳定.

It is unclear whether an interaction is significantly greater than 0. We would need to conduct a statistical test, but this **test is not (yet) available in a model-agnostic version**.

还不知道如果有交互的值是否会
远大于0.
很难说多大的 H-statistic 值
才说明是否是强交互.

Concerning the test problem, it is difficult to say when the H-statistic is large enough for us to consider an interaction "strong".

Also, the H-statistic can be larger than 1, which makes the interpretation difficult.

When the total effect of two features is weak, but mostly consists of interactions, than the H-statistic will be very large. These spurious interactions require a small denominator of the H-statistic and are made worse when features are correlated. A spurious interaction can be easily overinterpreted as a strong interaction effect, when in reality both features play a minor role in the model. A possible remedy is to visualize the unnormalized version of the H-statistic, which is the square root of the numerator of the H-statistic [11]. This scales the H-statistic to the same level as the response, at least for regression, and puts less emphasis on spurious interactions.

如两个features的effect很弱，但主要是由交互构成时，H值会很大.

spurious 虚假的 remedy 方案

denominator 分母

虚假的交互会被过度解释.

用平方根的H-statistic有可能解决这个情况.

$$H^*_{jk} = \sqrt{\sum_{i=1}^{n} \left[PD_{jk}(x_j^{(i)}, x_k^{(i)}) - PD_j(x_j^{(i)}) - PD_k(x_k^{(i)})\right]^2}$$

The H-statistic tells us the strength of interactions, but it does not tell us how the interactions look like. That is what partial dependence plots are for. A meaningful workflow is to measure the interaction strengths and then create 2D-partial dependence plots for the interactions you are interested in.

它只反映交互的存在，不反映交互的过程.

The H-statistic cannot be used meaningfully if the inputs are pixels. So the technique is not useful for image classifier.

不处理图像.

The interaction statistic works under the assumption that we can shuffle features independently. If the features correlate strongly, the assumption is violated and **we integrate over feature combinations that are very unlikely in reality**. That is the same problem that partial dependence plots have. Correlated features can lead to large values of the H-statistic.

它基于我们可以打乱features的假设下.且如果features的关联非常强，那么这个假设不成立.

Sometimes the results are strange and for small simulations **do not yield the expected results**. But this is more of an anecdotal observation.

7.3.6 Implementations

For the examples in this book, I used the R package `iml`, which is available on CRAN[12] and the development version on Github[13]. There are other implementations, which focus

[11] Inglis, Alan, Andrew Parnell, and Catherine Hurley. "Visualizing Variable Importance and Variable Interaction Effects in Machine Learning Models." arXiv preprint arXiv:2108.04310 (2021).

[12] https://cran.r-project.org/web/packages/iml

[13] https://github.com/christophM/iml

on specific models: The R package pre[14] implements RuleFit and H-statistic. The R package gbm[15] implements gradient boosted models and H-statistic.

7.3.7 Alternatives

The H-statistic is not the only way to measure interactions:

Variable Interaction Networks (VIN) by Hooker (2004)[16] is an approach that decomposes the prediction function into main effects and feature interactions. The interactions between features are then visualized as a network. Unfortunately no software is available yet.

Partial dependence based feature interaction by Greenwell et al. (2018)[17] measures the interaction between two features. This approach measures the feature importance (defined as the variance of the partial dependence function) of one feature conditional on different, fixed points of the other feature. If the variance is high, then the features interact with each other, if it is zero, they do not interact. The corresponding R package `vip` is available on Github[18]. The package also covers partial dependence plots and feature importance.

2features 的 PDP 可以反映
features 之间的关系.

7.4 Functional Decompositon

A supervised machine learning model can be viewed as a function that takes a high-dimensional feature vector as input and produces a prediction or classification score as output. Functional decomposition is an interpretation technique that deconstructs the high-dimensional function and expresses it as a sum of individual feature effects and interaction effects that can be visualized. In addition, functional decomposition is a fundamental principle underlying many interpretation techniques – it helps you better understand other interpretation methods.

Let us jump right in and look at a particular function. This function takes two features as input and produces a one-dimensional output:

$$y = \hat{f}(x_1, x_2) = 2 + e^{x_1} - x_2 + x_1 \cdot x_2$$

[14] https://cran.r-project.org/web/packages/pre/index.html
[15] https://github.com/gbm-developers/gbm3
[16] Hooker, Giles. "Discovering additive structure in black box functions." Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. (2004).
[17] Greenwell, Brandon M., Bradley C. Boehmke, and Andrew J. McCarthy. "A simple and effective model-based variable importance measure." arXiv preprint arXiv:1805.04755 (2018).
[18] https://github.com/koalaverse/vip