Quantifying Interpretability of Arbitrary Machine Learning Models Through Functional Decomposition

Christoph Molnar, Giuseppe Casalicchio, Bernd Bischl

LMU Munich

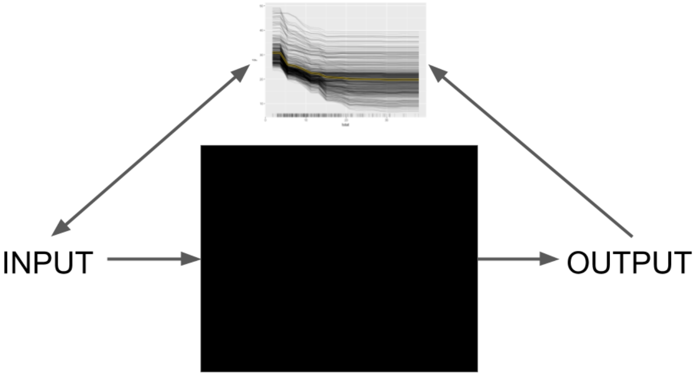
Statistical Computing 2019-07-03

Outline

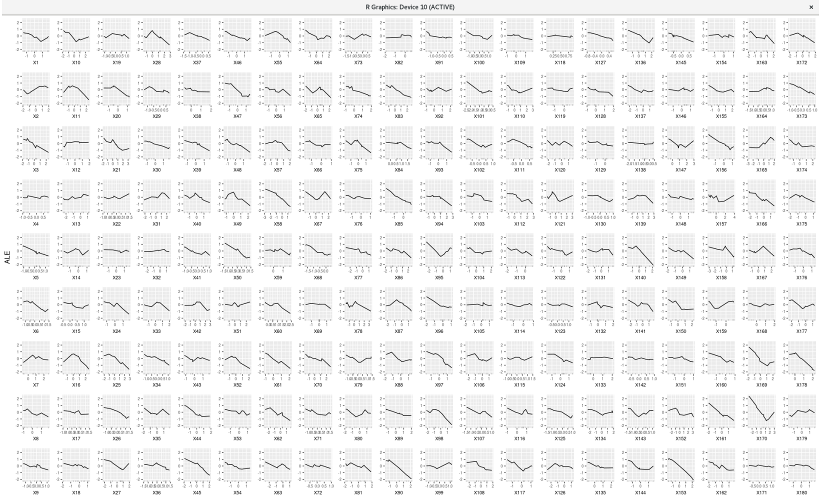
- Minimizing model complexity maximizes interpretability
- Functional Decomposition
- Measures
- Application

Black Box Problem

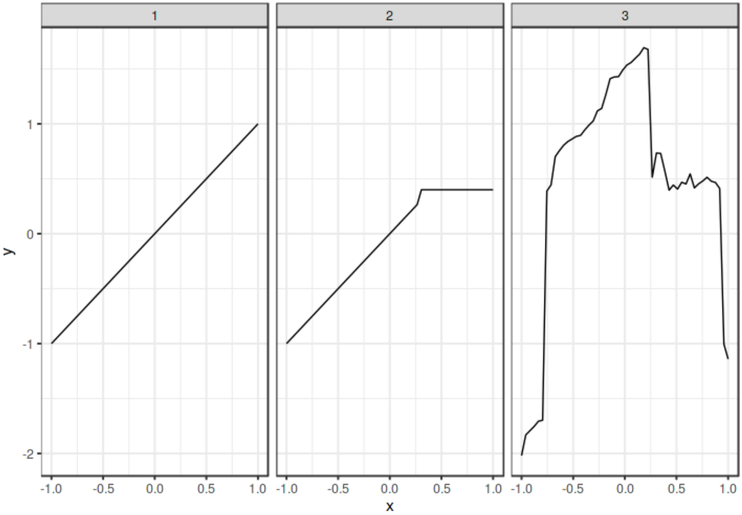
Solution: Post-hoc Interpretation?



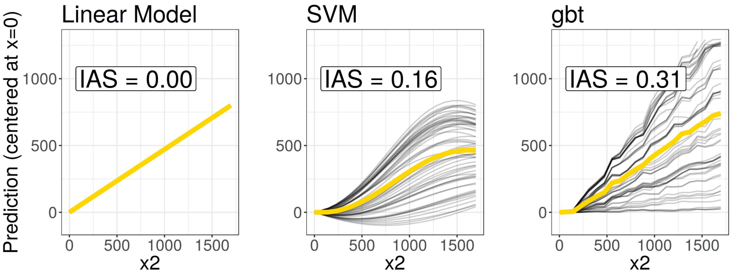
Sparsity Improves Interpretation



Linearity Improves Interpretation



Additivity Improves Interpretation



We Propose Measures of Model Complexity

Measure model complexity in a model-agnostic way: number of features, interaction strength, main effect complexity

$\Rightarrow$ Allows model comparison

$\Rightarrow$ Allows direct optimization for interpretability

$\Rightarrow$ Allows choosing models with improved post-hoc interpretation

Functional Decomposition

$$f(x) = \overbrace{f_0}^{\text{Intercept}} + \overbrace{\sum_{j=1}^{p} f_j(x_j)}^{\text{1st order effects}} + \overbrace{\sum_{j \neq k}^{p} f_{jk}(x_j, x_k)}^{\text{2nd order effects}} + \ldots + \overbrace{f_{1,\ldots,p}(x_1, \ldots, x_p)}^{\text{p-th order effect}}$$

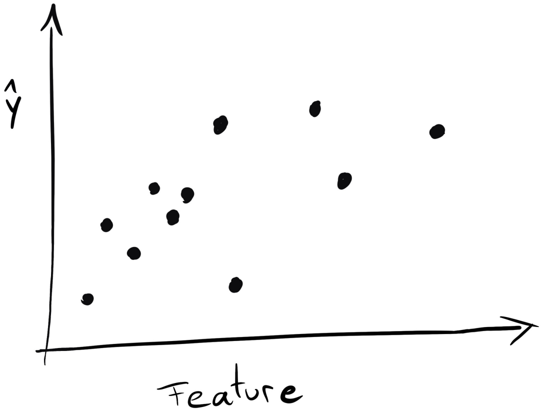
Functional Decomposition

$$f(x) = \overbrace{f_0}^{\text{Intercept}} + \overbrace{\sum_{j=1}^{p} f_j(x_j)}^{\text{1st order effects}} + \overbrace{\sum_{S \subseteq \{1,\ldots,p\}, |S| \geq 2} f_S(x_S)}^{\text{Higher order effects}}$$

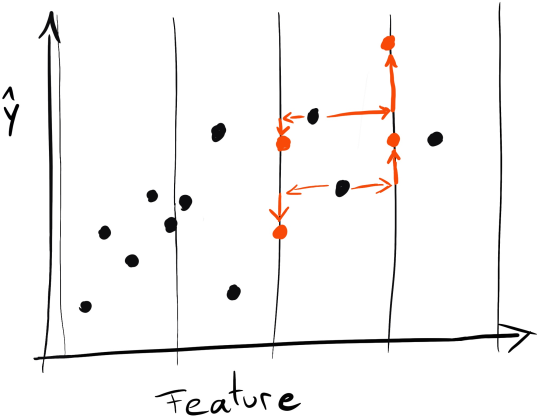
Functional Decomposition

$$f(x) = \underbrace{f_0 + \sum_{j=1}^{p} \overbrace{f_j(x_j)}^{\text{How complex?}} + \overbrace{IA(x)}^{\text{How much interaction?}}}_{\text{How many feature used?}}$$

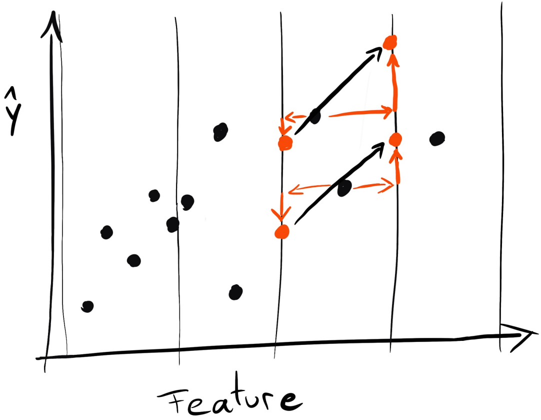
Accumulated Local Effects (ALE)



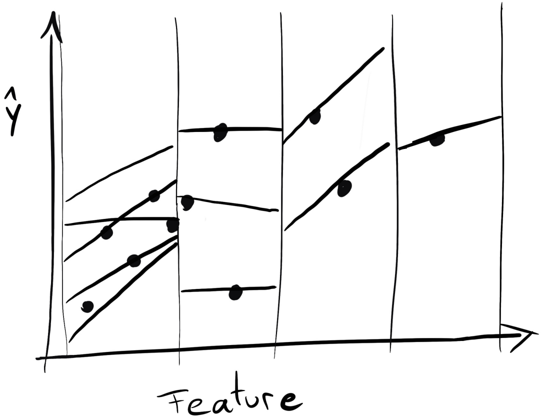
Accumulated Local Effects (ALE)



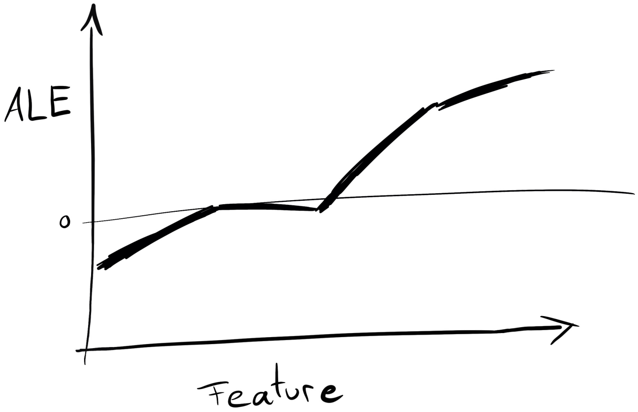
Accumulated Local Effects (ALE)



Accumulated Local Effects (ALE)

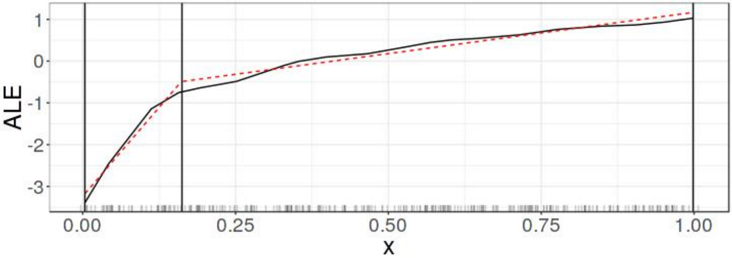


Accumulated Local Effects (ALE)



Main effect complexity

- Approximate ALE plot with linear segments
- Count number of non-zero coefficients
- Average over all features, weight with variance



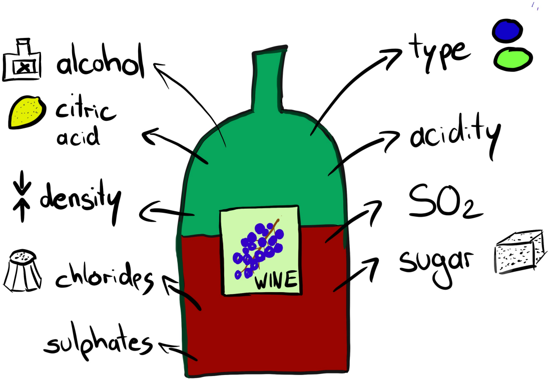
Interaction Strength

Measure main effect model with proportion of error explained:

$$\text{Interaction Strength} = \frac{\mathbb{E}(L(\hat{f}, f_0 + \sum_{j=1}^{p} f_j))}{\mathbb{E}(L(\hat{f}, c))}$$

Application: Multi-Objective Optimization

- Predict wine quality from physicochemical properties
- Minimize MAE, number of features, interaction strength, main effect complexity
- Search across different model classes and hyperparameter settings



Application: Multi-Objective Optimization

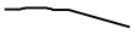
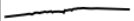
	xgboost (max_depth: 9,nrounds:913)	ksvm (C:0.79)	xgboost (max_depth: 1,nrounds:141)	rpart (maxdepth: 3)
MAE	0.47	0.53	0.57	0.61
NF	11	11	11	3
IA	0.64	0.27	0.02	0.14
MEC	4.12	1.99	2.90	1.81
fixed.acidity				
volatile.acidity				
citric.acid				
residual.sugar				
chlorides				
free.sulfur.dioxide				
total.sulfur.dioxide				
density				
pH				
sulphates				
alcohol				

Table 3. A selection of 4 models from the Pareto optimal set. From left to right, the models with best MAE, best MAE when $MEC \leq 2$, best MAE when $IA = \leq 0.1$, best MAE with $NF \leq 7$.

Summary

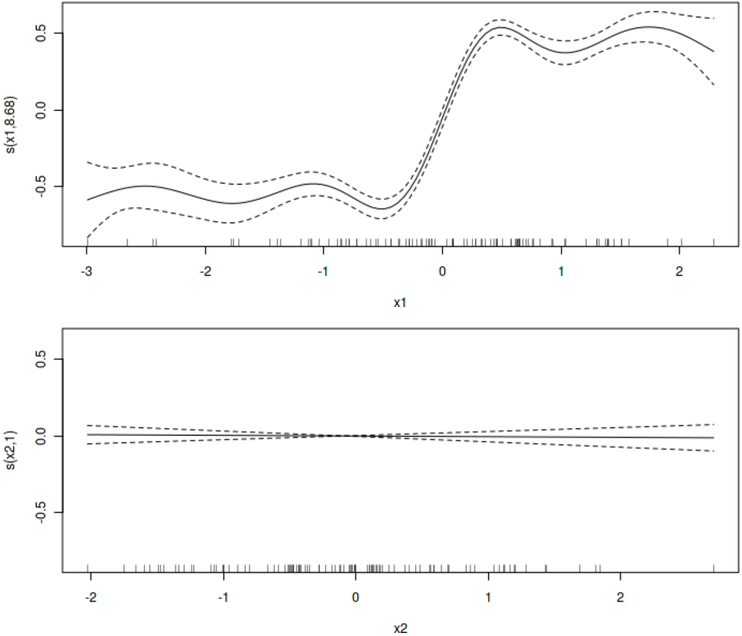
- Model-agnostic complexity measures: number of features, interaction strength, main effect complexity
- Use: model comparison and optimization metric
- Implementation: https://github.com/compstat-lmu/paper_2019_iml_measures
- Paper:https://arxiv.org/abs/1904.03867

The End

Why not GAM?

```
library("mgcv")
library("iml")
set.seed(42)
X = data.frame(x1 = rnorm(100), x2 = rnorm(100))
f = function(model = NULL, newdata) {
    1 *  (newdata[,1] > 0)
}
y = f(newdata = X)
dat = cbind(y, X)
mod.gam = gam(y ~ s(x1) + s(x2), data = dat)
```

Why not GAM?



Why not GAM?