# Intelligible Models for Classification and Regression

Rich Caruana
Microsoft Research
Microsoft Corporation
rcaruana@microsoft.com

Yin Lou
Dept. of Computer Science
Cornell University
yinlou@cs.cornell.edu

Johannes Gehrke
Dept. of Computer Science
Cornell University
johannes@cs.cornell.edu

The most accurate supervised learning methods for many problems are complex models such as boosted trees, SVMs with RBF kernels, or deep neural nets. In many applications, however, *what* is learned is just as important as model accuracy. Unfortunately, the high accuracy of complex models usually comes at the expense of interpretability. The goal of this work is to construct models that are as accurate as possible while retaining interpretability. Interpretability is not easy to define. Here we mean that users can understand the contribution of individual features in the model. This desiderata permits arbitrary complex relationships between individual features and the target, but excludes models with complex interactions between features. Thus we are interested in *generalized additive models* [3, 4] of the form:

$$g(y) = f_1(x_1) + ... + f_n(x_n), \tag{1}$$

The function $g(\cdot)$ is called the *link function* and $f_i$s are the *feature shape functions*. If the link function is the identity, Equation 1 describes an additive model (e.g., a linear regression model); if the link function is the logit function, Equation 1 describes a generalized additive model (e.g., a logistic regression classification model).

Consider a real dataset (where there may be interactions between features): the Concrete dataset relates the compressive strength of concrete to the age of the concrete and ingredients used to make it. We fit an additive model of the form in Equation 1. Figure 1 shows scatterplots of the shape functions learned for three of the eight features. As we can see from the figure, the compressibility of concrete depends nearly linearly on the Cement feature, but it is a complex non-linear function of the Water and Age features; we say that the model has *shaped* these features. A linear model without the ability to shape features would fit this data much worse because it cannot capture these non-linearities. Moreover, an attempt to interpret the contribution of features by examining the slopes of a simple linear model would be misleading; the additive model yields much better fit to the data while remaining intelligible. Full complexity models such as boosted trees, random forests, or neural nets would fit the data better better than the restricted GAM models, but would be difficult to interpret.

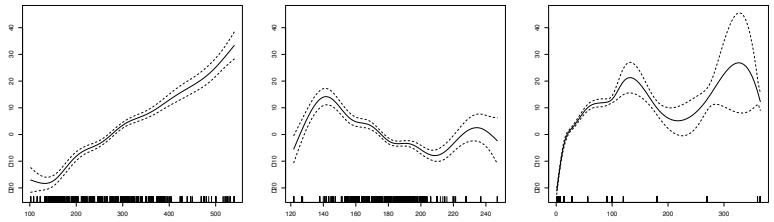


**Figure 1: Shape Functions for Three Features in the Concrete Dataset.**

Our work is the first large-scale study of different methods for training GAMs. We examine shape functions based on splines [2, 4] and boosted stumps [1], as well as novel shape functions based on bagged and boosted ensembles of trees that choose the number of leaves adaptively. We experiment with least squares, iteratively re-weighted least squares, gradient boosting, and backfitting to iteratively refine the shape functions and their contribution to the linear model. We apply these methods to a dozen classification and regression tasks. For comparison, we also fit simple linear models as a baseline, and unrestricted ensembles of trees as full complexity models to get an idea of what accuracy is achievable.

To summarize the results, as expected, the accuracy of GAMs falls between that of linear/logistic regression without feature shaping and full-complexity models such as random forests. Surprisingly, the best GAM models have accuracy much closer to the full-complexity models than to the linear models. In particular, the new GAM model and shaping functions we introduce based on boosted-bagged depth-limited trees outperform all previous GAM models by a significant margin while retaining intelligibility. Increasing the accuracy of these models as much as possible is important for at least two reasons: 1) if one is going to interpret the shaping functions learned for each feature, it is critical that the model is as accurate as possible so that the learned shaping functions are as accurate as possible; and 2) if shaping functions can be learned with high accuracy, it should be easier to detect feature interactions because whatever is left in the residuals *after shaping must be noise or interaction.*

## 1. REFERENCES

*[1] J. Friedman. Stochastic gradient boosting.* Computational Statistics and Data Analysis*, 38:367–378, 2002.*

*[2] T. Hastie and R. Tibshirani. Generalized additive models (with discussion).* Statistical Science*, 1:297–318, 1986.*

*[3] T. Hastie and R. Tibshirani.* Generalized additive models*. Chapman & Hall/CRC, 1990.*

*[4] S. Wood.* Generalized additive models: an introduction with R*. CRC Press, 2006.*