Understanding Multivariate Adaptive Regression Splines (MARS) through a Case Study

Yutong Jin

June 2nd, 2024

Introduction

Multivariate Adaptive Regression Splines (MARS) is a nonlinear regression method that combines regression splines and adaptive learning to explore complex relationships in data and is good at handling high-dimensional data [2]. It was developed by Jerome H. Friedman in 1991. MARS divides the overall data into many small regions, using recursive partitioning and spline fitting to derive nonlinear relationships between variables [1]. Especially, the MARS model automatically handles nonlinear relationships and interactions in continuous and dichotomous variables without having to make assumptions about the relationship between the independent and dependent variables.

Objectives

The primary objectives of this project are:

1. To understand the theoretical foundation of MARS.
2. To implement MARS using the `earth` package in R and analyze a dataset using this package.

This project focuses on the basic principles of MARS and its application to a simulated dataset.

Theoretical Foundation of MARS

The core idea of the MARS algorithm is to construct and select basis functions in an iterative manner. The model is constructed based on input variables $\mathbf{X}$ of dimension $n \times d$ and a dependent variable Y of dimension $n \times 1$. There is no need to establish an initial hypothesis between $\mathbf{X}$ and Y.

MARS uses piecewise polynomials to construct smooth basis functions (BF) that partition $\mathbf{X}$ into different regions. For each variable, there can be one or more split points, known as knots (t). The basis functions are expressed as follows:

$$(x-t)_+ = \begin{cases} x-t, & \text{if } x \geq t \\ 0, & \text{otherwise} \end{cases}; \quad (t-x)_+ = \begin{cases} t-x, & \text{if } x < t \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

By combining these basis functions, MARS can model complex, non-linear relationships and interactions in the data.

A MARS model with M basis functions can be represented as:

$$\hat{y}(x) = \beta_0 + \sum_{m=1}^{M} \beta_m B_m(x) \tag{2}$$

where β_0 is the intercept, β_m are the coefficients, and $B_m(x)$ are the basis functions, which can be individual truncated basis functions or products of multiple basis functions. For instance, a basis function $B_m(x)$ can be:

$$B_m(x) = (x - t)_+ \quad \text{or} \quad B_m(x) = (t - x)_+ \tag{3}$$

Additionally, basis functions can be products of other basis functions to model interactions between variables. For example:

$$B(X_1, X_2) = (X_1 - t)_+ \cdot (t - X_2)_+ \tag{4}$$

This flexibility allows MARS to effectively capture and model the complex relationships within the data.

Building Procedure

Forward Pass

Begin with a simple model, typically containing only the intercept term. Iteratively add pairs of basis functions that minimize the residual sum of squares (RSS). This process involves three main steps:

1. **Partition the Sample Space**: Use candidate basis functions to divide the sample space. Initially, the basis function $B_0(x) = 1$ is used.
2. **Consider Interactions**: Evaluate potential interactions between variables to determine how they affect the response.
3. **Add Basis Functions Continuously**: Continuously add the optimal pair of basis functions, which reduces the residual error the most, to the model, one pair at a time, to improve accuracy. This iterative process continues until a predefined stopping criterion is met, such as reaching the maximum number of basis functions or achieving the minimum RSS.

Backward Pass

After adding pairs of basis functions stepwise in the forward pass, we obtain a large model of the form (2). This model typically overfits the data, so MARS performs a backward pass to prune the model to reduce the overfitting.

The specific steps are: First, remove the basis functions that contribute the least to the model performance, i.e., the basis functions whose removal results in the smallest increase in the residual sum of squares (RSS) error. Next, the coefficients of the remaining basis function terms are continuously adjusted to maintain the accuracy of the model after each removal. Ultimately, the least important basis functions are eliminated iteration by iteration to ensure that only the most important terms remain in the final model. And this pruning process is guided by the Generalized Cross-Validation (GCV) criterion.

GCV is an effective technique for model validation that can helps avoid overfitting by penalizing models with too many terms. This criterion is defined as

$$\text{GCV}(\lambda) = \frac{\sum_{i=1}^{N}(y_i - \hat{f}_\lambda(x_i))^2}{(1 - \frac{M(\lambda)}{N})^2} \tag{5}$$

where $\hat{f}_\lambda(x_i)$ represents the predicted value for observation i using the model with regularization parameter λ, and $M(\lambda)$ is the effective number of parameters in the model.

The value $M(\lambda)$ is calculated by considering both the number of terms in the model and the parameters used to determine the optimal positions of the knots. For a linear model, the formula for $M(\lambda)$ is given by:

$$M(\lambda) = r + cK \tag{6}$$

where r is the number of linearly independent basis functions in the model, K is the number of knots selected in the forward process, and c is a penalty term (usually 3 for piecewise linear regression).

The backward pass in MARS minimizes the GCV criterion by removing the least significant basis functions, thus ensuring that the final model is both parsimonious and accurate.

Case Study

In this case study, I applied the Multivariate Adaptive Regression Splines (MARS) model to a dataset that summarizes obesity and the number of deaths caused by it in each country. It also includes variables such as mean BMI for males and females, prevalence of hypertension, obesity rates, and other relevant factors for multiple countries over several years. The objective of this case study is to develop and evaluate a model by MARS that predicts the number of deaths attributable to high body-mass index (BMI) per 100,000 people. And I downloaded this dataset from Kaggle.com. Here is the link: `https://www.kaggle.com/datasets/yutodennou/death-and-obesity`

The dataset went through several preprocessing steps before the MARS model was applied. Missing values were handled by removing rows with incomplete data. Categorical variables such as country codes were coded as factors. The dataset is originally provided in three parts:

1. `train.csv` contains obesity and deaths data from 1990 to 2013, which is used to train the regression model.
2. `test.csv` includes the explanatory variables for the year 2014 and these variables were used to predict the outcome for 2014.
3. `answer.csv` holds the actual values of the outcome variable for the year 2014 and it was used to evaluate the model's performance.

Analysis Process

The MARS model was constructed using the `earth` package in R, which employs a two-step process: the forward pass and the backward pass.

Initial Model Building (Figure 1)

I began by training a MARS model on the dataset using the `earth` package in R. This initial model construction involved a forward pass, where basis functions were added iteratively to minimize the residual sum of squares (RSS). The model selection process was guided by the Generalized Cross-Validation (GCV) criterion, which helps prevent overfitting by penalizing models with too many terms.

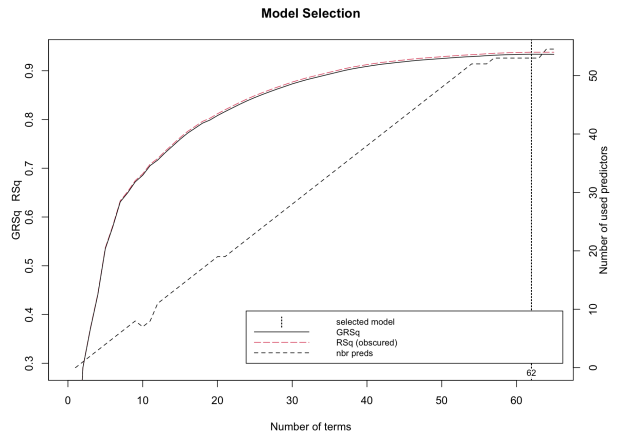

Figure 1: Model selection plot for the MARS model.

Hyperparameter Tuning (Figure 2)

Using cross-validation (using the `train` function from the `caret` package), I determined the optimal hyperparameters, specifically `nprune` (number of pruned terms) and `degree` (degree of interaction), for the MARS

model. We defined a grid of hyperparameter values and evaluated the model's performance using the Root Mean Squared Error (RMSE) metric. Figure 2 shows the RMSE curves for different degrees of the MARS model as the number of terms increases.

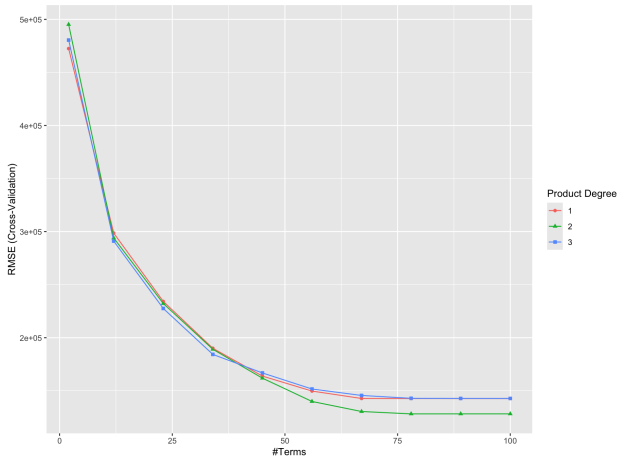

Figure 2: Cross-validation results for hyperparameter tuning of the MARS model.

Variable Importance Analysis (Figure 3)

After hyperparameter tuning, I explored the relative importance of different features in the MARS model. Figure 3 displays two variable importance plots based on the GCV score and the RSS, respectively. These plots rank the variables in descending order of importance, with longer bars indicating higher importance. The "Mean.BMI.Female" and "CountryFiji" features emerged as the most important predictors, followed by other country names and variables related to hypertension prevalence, obesity, and overweight percentages.

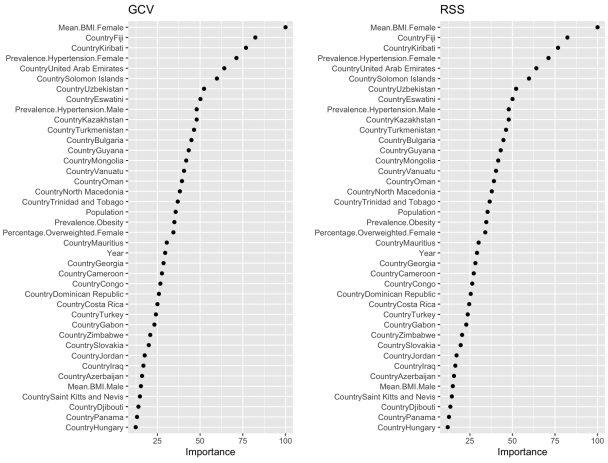

Figure 3: Variable importance plots for the MARS model.

Final Model Training (Figure 4)

Based on the cross-validation results, I identified the optimal hyperparameters as `degree = 2` and `nprune = 78`, which yielded the lowest GCV score. Using these values, I trained the final MARS model on the entire training dataset. Figure 4 shows the model selection process for the final model, with the vertical line indicating the selected model (`nprune = 78`) with the lowest GCV score.

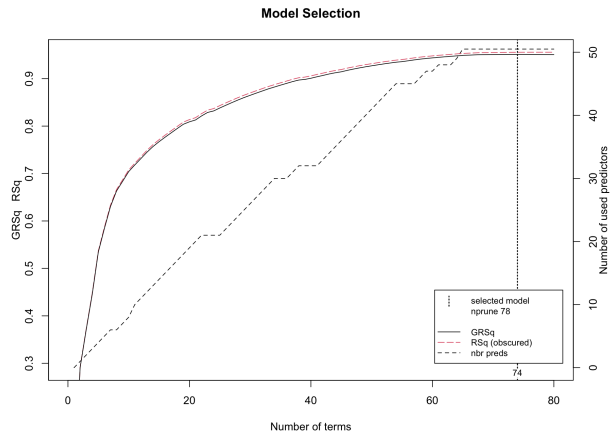

Figure 4: Model selection for the final MARS model with the optimal degree and number of pruned terms (degree = 2, nprune = 78).

Model Evaluation (Figure 5)

After training the final MARS model, I evaluated its performance on a separate test dataset. Figure 5 shows a scatter plot of the actual versus predicted values of deaths caused by high BMI for the year 2014. The data points generally follow a linear trend along the diagonal line, which means the model captured the overall relationship between the independent variables and the target variable reasonably well. However, there are some outliers and deviations from the perfect fit line, suggesting potential areas for further improvement.

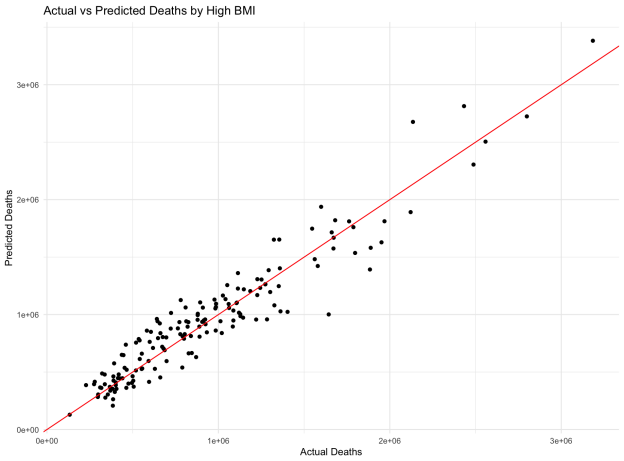

Figure 5: Actual vs. Predicted Deaths by High BMI.

Therefore, the MARS model proved to be an effective and flexible approach for predicting deaths caused by high BMI, capturing complex nonlinear relationships and interactions while automatically selecting relevant features and preventing overfitting.

References

[1] Jerome H Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

[2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.