

# Learning Multivariate Adaptive Regression Splines: A Comprehensive Guide

Authored by  
**Mohammed Iooti**

November 6, 2025

## RECOMMENDED CITATION

Mohammed Iooti (2025). *Learning Multivariate Adaptive Regression Splines: A Comprehensive Guide*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=11748>

When analyzing the relationship between a set of predictor variables and a [response variable](#), data scientists often begin with [linear regression](#). This foundational statistical technique is highly effective when the underlying relationship is linear, relying on the core assumption that the relationship between a given predictor variable and the outcome can be expressed simply:

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

However, in real-world scenarios, true relationships between variables are frequently **nonlinear** and structurally complex. Applying a strictly linear model to complex, curved data results in a significant lack of fit, leading to poor predictive accuracy and often misleading inferences. This inherent rigidity necessitates the adoption of more flexible modeling approaches capable of capturing intricate, localized data structures.

## Limitations of Traditional Polynomial Regression

One traditional method for attempting to account for a nonlinear relationship is through [polynomial regression](#). This approach extends the linear model by including polynomial terms (squares, cubes, etc.) of the predictor variables, allowing the fitted model curve to bend and adapt to the data. The general form of the polynomial regression equation is:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \dots + \beta_h X^h + \varepsilon$$

In this equation,  $h$  is defined as the "degree" of the polynomial. As we increase the degree  $h$ , the model gains significant flexibility, enabling it to better contour the observed nonlinear data patterns. While beneficial for capturing moderate curvature, polynomial regression suffers from critical drawbacks that limit its practical utility, especially in datasets characterized by sharp, localized changes.

The primary limitations associated with relying solely on polynomial functions are the inherent risk of excessive complexity and the restrictive global nature of the fit. The first major concern is the susceptibility to [overfitting](#). If the degree,  $h$ , is chosen to be too large, the model begins to fit the random noise and idiosyncrasies inherent in the training data rather than focusing on the true underlying signal. In statistical practice, the degree  $h$  is rarely set higher than three or four because beyond this point, the resulting model typically lacks the crucial ability to generalize well to unseen or new data, drastically reducing its predictive value.

The second drawback is that polynomial regression imposes a single, **global function** across the entire range of the dataset. This assumption is often inaccurate because real-world relationships frequently change behavior abruptly in different regions of the predictor space. A single polynomial equation struggles to represent these distinct local behaviors efficiently, leading to poor approximations in segmented data. This rigidity motivates the search for models that can adapt

locally.

## Introducing Multivariate Adaptive Regression Splines (MARS)

The Multivariate Adaptive Regression Splines ([MARS](#)) algorithm provides a powerful non-parametric alternative specifically designed to model complex, highly nonlinear relationships. Unlike polynomial regression, which forces a smooth curve globally, MARS divides the predictor space into distinct, manageable regions and fits a separate, simpler regression function (often linear) within each region.

MARS is essentially an extension of linear models that automatically models nonlinearities and interactions between variables. It achieves this remarkable flexibility by utilizing a collection of piecewise linear functions, known as **basis functions** or hinge functions, which adaptively adjust to the local features of the data. This adaptive nature allows MARS to capture abrupt changes and localized variations far more effectively than traditional global methods.

The final MARS model is constructed as a weighted sum of these basis functions, each operating over a specific segment of the data determined by strategically placed division points called **knots**. The entire process--identifying the optimal location of these knots and selecting the relevant variables--is highly automated, providing an excellent balance between predictive power and model interpretability.

### The Core Mechanics of MARS: Basis Functions and Knots

The fundamental operation of [Multivariate Adaptive Regression Splines](#) revolves around two key processes: dividing the data space and defining the functional form for each division. This mechanism ensures that the model remains locally adaptive, fitting different curves where necessary to maximize fit.

**1. Defining the Knots and Dividing the Dataset:** First, the algorithm systematically scans the entire dataset to identify optimal split points, referred to as **knots**. A knot is a specific value of a predictor variable where the relationship between that predictor and the response variable changes slope or direction. The initial step involves proposing every unique data point for every predictor variable as a potential knot. The algorithm then tests these candidate features by creating a preliminary linear regression model using the candidate basis functions. The point that results in the greatest reduction in the residual sum of squares (error) is selected as the optimal knot for that step.

Once the first knot is identified, this iterative, greedy process is repeated. The algorithm continues to seek additional knots across all predictor variables, aiming to maximize the fit improvement at each stage. This forward stepwise procedure continues until a specified maximum number of knots

(or basis functions) is reached, resulting in a highly overfit, complex model structure, often divided into  $k$  pieces.

**2. Fitting Hinge Functions (Basis Functions):** Once the dataset is divided by the selected knots, a localized regression function is fitted to each piece. These functions are typically piecewise linear components known as **hinge functions**, which form the building blocks of the MARS model. A hinge function takes the general form  $\max(0, X - t)$  or  $\max(0, t - X)$ , where 't' is the knot location. This structure ensures mathematical continuity at the knot while allowing the slope of the relationship to change abruptly, facilitating local adaptation.

For instance, consider a simple MARS model with one knot placed at the value 4.3. The corresponding hinge functions define two segments, one for values less than 4.3 and one for values greater than 4.3. The functional model would be defined piecewise:

$$y = \beta_0 + \beta_1(4.3 - x) \text{ if } x < 4.3$$

$$y = \beta_0 + \beta_1(x - 4.3) \text{ if } x > 4.3$$

In this specific example, the value **4.3** was determined to be the optimal cutpoint that minimized the model error most effectively. By fitting different regression coefficients ( $\beta_0$ ,  $\beta_1$ ) to the two resulting segments, the model achieves localized adaptation rather than forcing a single curve across the entire data range.

A more complex scenario involving two knots, say at 4.3 and 6.7, would generate three distinct segments, each with its own local fit. The model representation illustrates how the combination of hinge functions covers the entire range:

$$y = \beta_0 + \beta_1(4.3 - x) \text{ if } x < 4.3$$

$$y = \beta_0 + \beta_1(x - 4.3) \text{ if } x > 4.3 \ \& \ x < 6.7$$

$$y = \beta_0 + \beta_1(6.7 - x) \text{ if } x > 6.7$$

The points **4.3** and **6.7** act as the cutpoints, and three distinct regression segments are fitted. This demonstrates the localized fitting capability that makes MARS highly effective for modeling complex, piecewise relationships found in real-world data.

## Building the MARS Model: Model Selection and Pruning

The selection of the final, optimal MARS model is a rigorous, two-stage process designed to balance model complexity with predictive accuracy. This procedure is critical because the initial, greedy construction phase almost always results in a model that is excessively complex and [overfit](#) to the training data.

The first stage, the **\*\*Forward Stepwise Procedure\*\***, is the greedy expansion discussed

previously. It starts with an intercept-only model and iteratively adds pairs of basis functions associated with new knots. This continues until the model is highly complex, incorporating a vast number of terms that cover all possible nonlinearities and interactions. The goal here is simply to ensure that the maximal set of potentially useful basis functions has been identified.

The second stage, the **Backward Stepwise Procedure**, is a crucial pruning step. Starting from the maximal model generated in the forward step, basis functions are systematically removed one by one. The removal criterion is based on minimizing the generalized cross-validation (GCV) score, which functions as an effective complexity penalty. This backward elimination continues until the model producing the lowest GCV score is found. This technique efficiently prevents overfitting by removing terms that only contribute marginally to the training fit but increase the variance of the model, thus improving generalization.

Finally, to rigorously validate the optimal level of complexity (the number of remaining basis functions,  $k$ ), we often employ stringent testing methodologies such as [k-fold cross-validation](#). By fitting several different MARS models, each utilizing a different number of basis functions or knots determined by the pruning step, we can evaluate how well each model performs on data it has never encountered before.

The model that exhibits the lowest test mean squared error (MSE) across the validation folds is ultimately selected. This model is chosen because it demonstrates the best ability to generalize, providing the most reliable predictions for new, previously unobserved data points. This systematic approach ensures the final MARS model is both robust, flexible, and statistically parsimonious.

## Advantages and Disadvantages of MARS

Multivariate Adaptive Regression Splines present several compelling advantages that make them a popular choice in data analysis, particularly when dealing with complex data that resists simple linear modeling.

### Pros of MARS:

**Versatility:** MARS is highly flexible and can be effectively used for both standard [regression and classification problems](#), requiring minimal structural modification between these statistical tasks.

**Computational Efficiency:** The MARS algorithm offers quick computation and scales well, making it highly efficient, particularly when processing large datasets where iterative model training might otherwise be prohibitively time-consuming.

**Automatic Feature Selection:** The critical backward pruning step naturally performs an intrinsic form of feature selection, effectively removing noisy or irrelevant variables (and their associated hinge functions) that do not contribute significantly to reducing the Generalized Cross-Validation score.

**Simplicity in Preprocessing:** Unlike many distance-based algorithms or those sensitive to scale, MARS does not require the user to standardize or normalize the predictor variables prior to modeling, simplifying the initial data preprocessing pipeline.

**High Interpretability:** Because the final model is expressed as a simple sum of basis functions, each clearly delineated by a knot (cutpoint), the contribution of each predictor and the nature of the nonlinearity can be easily analyzed, offering far greater transparency and interpretability than "black-box" models.

Despite these inherent strengths, MARS is not without limitations, especially when compared to cutting-edge ensemble methods that dominate predictive modeling contests:

### Cons of MARS:

**Performance Ceiling:** While a strong and robust modeling technique, MARS tends to not perform quite as well as highly optimized non-linear ensemble methods, such as random forests and gradient boosting machines, especially on large, highly structured data where these methods excel at capturing subtle, high-order interactions that MARS may miss.

**Sensitivity to Knots:** The precise location and definition of the knots can sometimes be sensitive to data noise or isolated outliers, potentially impacting the resulting basis functions and overall model stability, although the pruning step mitigates this risk significantly.

## Implementing MARS in Practice

The implementation of [Multivariate Adaptive Regression Splines](#) is widely supported across major statistical and machine learning programming environments, making it readily accessible for practical application. In R, the dedicated `earth` package is the standard and most comprehensive choice, providing a robust and complete implementation of the MARS algorithm with extensive options for customization and evaluation. Similarly, in Python, implementations are available via specialized libraries and extensions built upon frameworks like scikit-learn, allowing users to easily define and train these adaptive spline models within a modern data science workflow.

For those interested in applying this powerful, adaptive technique to their own datasets, the following resources provide step-by-step examples detailing how to fit multivariate adaptive regression splines (MARS) models effectively in both R and Python environments:

[Multivariate Adaptive Regression Splines in R](#)

[Multivariate Adaptive Regression Splines in Python](#)