Extended abtract

EXTENDED ABSTRACT

Title: The MARS algorithm in the spatial framework. Nonlinearities, spatial effects and nonlinear spatial effect in a hedonic model.

Authors and e-mails: Fernando A. López[1] ; Konstantin Kholodilin[2]

Department: Métodos Cuanitativos e Informáticos.

University: [1]Universidad Politécnica de Cartagena; [2]DIW, Berlin

Subject area: *S11 – Big Data en Ciencia Regional*

Abstract: *(minimum 300 words)*

Multivariate Adaptive Regression Spline (MARS) is a powerful non-parametric technique that automatizes the selection of non-linear terms in regression models. Non-linearities and spatial effects are natural characteristics in numerous spatial hedonic pricing models. In this paper, we propose to use the MARS data-driven methodology in order to account for potential non-linearities, spatial effects, and non-linear spatial effects in a hedonic model. Using a large data set of more than 23000 houses in Madrid urban area we confirm the presence of both effects (non-linearities and spatial autocorrelation) in the hedonic model. The results also show that there is a non-linear effect of the prices of neighbor houses on the price of each house. High prices of neighbor houses has lower impact in the house price that low price of neighbor houses. Finally, an extensive Monte Carlo exercise evaluates the ability of MARS to incorporate the correct spatial spillover terms in spatial regression models including at same time non-linear effects.

Keywords: *Multivariate Adaptive Regression Spline, spatial regression models, hedonic models, pricehouse, Madrid.*

JEL codes: C4 C5 R1.

2 Methodology and data

2.1 Methodology

MARS is a flexible non-parametric piecewise regression technique introduced by Friedman (1991). This data-driven technique is specifically useful to identify non-linearities in regression models without previous assumptions about the functional form or which are the explanatory variables or the number of them. The main characteristic of this methodology is that the econometric model considers different regression slopes in distinct intervals for each predictor. Unlike better known linear regression techniques, MARS does not assume that coefficients are stable across the entire range of each variable and instead uses splines in order to fit piecewise continuous functions to model responses. In general, MARS constructs a piecewise linear function for capture nonlinear relationship with an adaptive manner (Hoang et al., 2017). The principal advantage of this methodology compared with similar algorithms (e.g., polynomial models) is the simplicity of the resulting econometric model and its easy interpret ability. Moreover, the MARS models are reported to work satisfactorily in terms of computational cost irrespective of dimension low, medium or high. This is very useful when it is suspected that model inputs have varying optima across different levels of the model inputs (Crino & Brown, 2007).

2.1.1 The MARS algorithm

Like in any regression model, the objective of this methodology is to build a econometric model in order to explain the variation of a dependent variable $Y = (y_1, \ldots, y_n)'$ with a set of potential independent variables $X = (X_1, \ldots, X_p)$ with $X_i = (x_{1i}, x_{2i}, \ldots, x_{ni})'$. In order to reach this classical objective, the MARS uses the named basic functions (BF) of the form $(x - c)_+ = \max\{0, x - c\}$ and $(c - x)_+ = \max\{0, c - x\}$, where the subscript "+" means that the function takes only the positive value or zero in case of negative difference. Such pairs of linear functions are called "hinge functions" (or two-sided truncated functions) and the constant c denotes a knot, where the slope changes. The collection of all possible BFs, $\mathcal{C}$, is used to construct the following econometric model:

$$\mathcal{C} = \{(x - c)_+, (c - x)_+\} \quad \text{with} \quad c \in \{x_{1i}, x_{2i}, \ldots, x_{ni}\} \quad \text{and } i = 1, \ldots, p \tag{1}$$

Each function is piecewise linear with a knot c at every x_{ij}, and in case of all input values are distinct, there are np hinge functions, or equivalently $2np$ basic functions. Using those BFs, the model-building strategy is similar to a classical forward stepwise regression, using as inputs the functions from the set $\mathcal{C}$ and their products. The final expression of the model is as follows:

$$Y = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(X) + \epsilon \tag{2}$$

where $h_m(X)$ is a BF, or a product of two o more such functions, if interactions between variables are permitted or perhaps the original predictor if have a linear impact on the dependent variable. The coefficients β_m are estimated by minimizing the sum of squared (residual) errors (SSE) similarly to a standard linear regression model.

2.1.2 The learning phase or forward pass

The model-training process will iteratively select and add some of hinge functions into the model (or the original predictor). During the training process at each step, MARS selects new terms that minimize the SSE using ordinary least squares (OLS). In this forward pass, MARS algorithm starts with a single model including only the intercept term β_0. At each subsequent step, a reflected pair of hinge functions (or an original predictor) are selected and added to the model. The selected pair of hinge functions (or original predictor) can enter in the model directly; alternatively, they can be multiplied by an existing basic function that is already in the model and become new basis functions. The second case allows the interaction between/among different predictors to be modelled. Note that a reflected pair of hinge functions always enter the model together (but may be removed separately in the pruning process; see the pruning phase). The forward pass goes on until it meets one of many conditions such as: (i) maximum number of model terms (chosen by the user) before pruning is reached; (ii) the change in the SSE is too small to continue. The search of hinge functions at each steep can be done using brute force, but this search can be sped up with a heuristic that reduces the number of parent terms to consider (see Friedman 1993).

In general, at the end of this process we have a large model of the form of equation (1). The MARS models obtain in this forward pass is adaptive and can exhibit a high degree of flexibility that may ultimately result in over fitting, if no measures are taken to counteract it. To solve the overfitting problem and build a model with better generalization ability a pruning procedure must be applied.

2.1.3 The Pruning Process

Although there are other methods, MARS typically applies a backward deletion procedure to prune the model. Using this procedure, the second phase of this algorithm is the pruning step, where a "one-at-a-time" backward deletion procedure is applied in which the basis functions with the least contribution to the model are eliminated until find the best submodel. This pruning is based on a generalized cross-validation (GCV) criterion originally propose by Craven & Wahba (1979) and adapted by Friedman & Silverman (1989) named Lack-of-Fit (LOF). Note that the raw SSE using on the training data is inadequate for comparing models, because the SSE always increase as MARS terms are dropped and therefore use this criterion in the backward pass always select the largest model. Therefore, the GCV criterion is used to find the overall best model from a sequence of fitted models, where a larger GCV value tends to produce a smaller model, and vice versa. The GCV criterion is estimated as the LOF criterion (Hastie and Tibshirani 2001). At each step, the algorithm removes a term in the model that results in the smallest increase in the sum of squared error, obtaining an optimal model. This criterion is defined as,

$$GCV = \frac{1}{n}\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\left[1 - \frac{\hat{C}(M)}{n}\right]^2} \tag{3}$$

where is a complexity function, and $\hat{C}(M) = C(M) + dK$ and $C(M)$ is the number of parameters being estimated (the number of linearly independent basis functions without the intercep term); K is the number

of knots selected in the forward process, and d represents a cost for each basic function optimization: usually $d = 2$, if the model does not involve interaction terms, and $d = 3$, otherwise (Friedman, 1991). Thus, the GCV formula adjusts the SSE to take into account the flexibility of the model. Larger values of d result in fewer knots and smoother function estimates. The best MARS approximation is the one with the highest GCV value. In order to get a measure similar to R^2, the GCV coefficient can be standardized and a new coefficient is defined as:

$$GRSq = 1 - \frac{GCV}{GCV.tot} \tag{4}$$

where $GCV.tot$ is the GCV of a model with only the intercept term. In this pruning phase, the researcher can determine the maximum number of knots considered, the minimum number of observations between knots, and the highest order of interaction terms. In addition, a critical strength of MARS is that it can train models that are very easily interpretable. However, increasing the flexibility generally reduces the interpretability. If interpretability is not a key consideration, then perhaps a more flexible algorithm such as random forest should be used instead. This is the reason why the degree of interaction is usually limited to one or two but rarely above. A useful option in the MARS procedure is to set an upper limit on the order of interaction. For example, one can set a limit of two, allowing pairwise products of piecewise linear functions, but not three- or higherway products. This can aid in the interpretation of the final model. An upper limit of one results in an additive model.

Parameter options

The basic inputs to the MARS algorithm are the predictors and the response variable. A set of parameter options can be pre-specified by the researcher: The order of interaction; the inclusion of variables linearling; the maximum or minimum of nodes and the number of observations between nodes are the more relevant. (Friedman, 1991) suggest defaults values for this parameters that we will select in the Monte-Carlo section. this doubles the number of predictors, forces that into the range of 20 to 200, and finally adds 1 for the intercept $nk = min(200, max(20, 2 * ncol(x))) + 1$

No regression modeling technique is best for all situations. The algorithm has pros and cons (reference). By example, MARS models are simple to understand and interpret. MARS is suitable for handling fairly large datasets. As an example, the earth test suite has additive models with 8 million cases and 100 variables, and 80 million cases and 2 variables. Bigger models are possible Milborrow. Derived from mda:mars by T. Hastie and R. Tibshirani (2018). The algorithm has overfitting problem and has propension to identify knots in case of linear relationship. In order to identify changes not significant in the slope coefficients, some test can be used (By example LR test). Appendix I include a technical note about this process.

Nota: El algoritmo MARS tiende a 'dividir' variables que son lineales. Esto es debido a que usa un criterio basado en reducir el GCV. En todo caso, aunque MARS 'rompa' una variable en varios trozos, es necesario realizar un test (LR por ejemplo) para evaluar si los coeficientes asociados a la variable dividida son significativamente distintos.

Table 1: Example Forward and Backward of MARS algorithm

	Forward pass					Backward	
	Model1	Model2	Model3	Model4	Model5	Model4	Model3
(Intercept)	1.917^{***}	2.668^{***}	3.766^{***}	3.794^{***}	3.757^{***}	3.794^{***}	3.766^{***}
	(0.061)	(0.063)	(0.371)	(0.370)	(0.370)	(0.370)	(0.371)
$(X - 0.5857)_+$		0.685^{*}	4.096^{***}	4.410^{***}	3.966^{**}	4.410^{***}	4.096^{***}
		(0.281)	(1.169)	(1.180)	(1.208)	(1.180)	(1.169)
$(0.5857 - X)_+$		-5.159^{***}	-7.531^{***}	-7.584^{***}	-7.513^{***}	-7.584^{***}	-7.531^{***}
		(0.202)	(0.814)	(0.812)	(0.812)	(0.812)	(0.814)
$(X - 0.2042)_+$			-3.107^{**}	-3.231^{**}	-3.067^{**}	-3.231^{**}	-3.107^{**}
			(1.034)	(1.034)	(1.036)	(1.034)	(1.034)
$(X - 0.9755)_+$				-20.731	-44.955^{*}	-20.731	
				(11.907)	(18.951)	(11.907)	
$(X - 0.9337)_+$					8.651		
					(5.273)		
$R^2 (= RSq)$	0.0000	0.7513	0.7569	0.7587	0.7604	0.759	0.757
		0.7513	0.0055	0.0019	0.0016		
$GRSq$	0.0000	0.7463	0.7494	0.7488	0.7479	0.7462	
		0.7463	0.0031	-0.0006	-0.0009		

$^{***}p - value < 0.001$, $^{**}p - value < 0.01$, $^{*}p - value < 0.05$

2.1.4 Illustration

To show how the algorithm works, a data set is generated,

$$y_i = 1 + 4(x_i - 0.2)_+ + 8(0.2 - x_i)_+ - 3(0.6 - x_i) + \epsilon_i \tag{5}$$

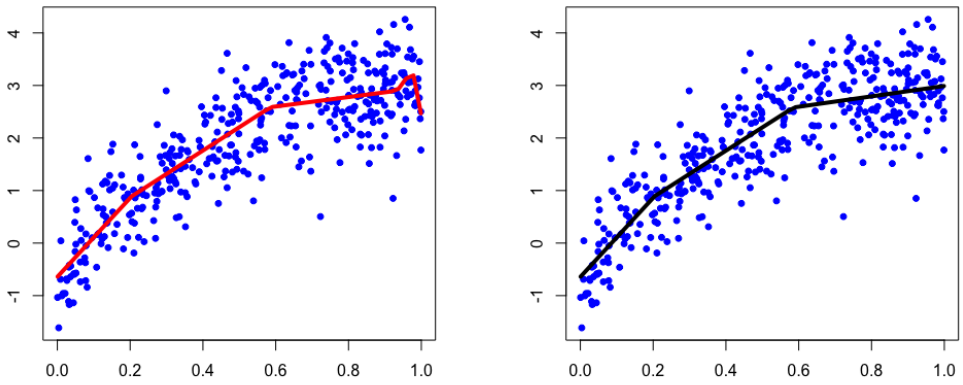
The DGP consider two knots c =0.2, 0.6, with $\epsilon_i = N(0, 1)$ and $i = 1, ..., n = 400$.

Table 1 show the OLS estimate models with the terms selected by MARS algorithm in each step. In the first phase, the algorithm incorporate in an iterative process a total of 6 terms in the forward pass (Models 1-5). In each step select the term that maximize the $RSq = R^2$. The Model 1 is the baseline model and only consider the intercept. The Model 2 select the knot 0.5857 and two terms are incorporate to the model $(X - 0.5857)_+$ and $(0.5857 - X)_+$ with an increase of RSq of $\Delta RSq = 0.7513$. The Model 3 select the knot 0.2042 and include the term $(X - 0.2042)_+$. In this case the the $\Delta RSq = 0.0055$. The Model 4 select the term $(X - 0.9755)_+$ with an increase of $\Delta RSq = 0.0019$ and the Model 5 select $(X - 0.9337)_+$ with an increase of $\Delta RSq = 0.0016$. The process stop because the RSq increase with the next term $(X - 0.73776)_+$ is less than 0.001, a value prefixed by the researcher ,ΔRSq=0.00086 (This model is not include in the Table to save space). In the second phase, the MARS algorithm prune the Model 5 using the $GRSq$ as criteria (with d=2). *A priory*, any of the terms selected in the forward pass can be removed. In our case, note that for Model 4 and Model 5 the $\Delta GRSq$ values are negative. Finally the Model 3 is the model selected by the MARS algorithm.

Figure 1 show the algorithm steps. In firs place the red line in Figure 1a show the model before to pruning. In second place the black line in Figure 1b show the final model after to pruning. The knots identify in teh

final model by the algorithm (0.5857 and 0.2042) are so close to the true knots (0.6 and 0.2).

Figure 1: Example MARS algorithm



(a) Forward pass

(b) Final pruning model