

Statistics 360: Advanced R for Data Science

Multivariate Adaptive Regression Splines (MARS)

Becky Lin

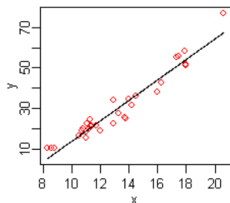
Terminology of MARS

- ▶ MARS stands for Multivariate Adaptive Regression Splines.
- ▶ Multivariate: able to generate model based on several input variables (high dimensionality).
- ▶ Adaptive: Generates flexible models in passes each time adjusting the model.
- ▶ Regression: estimation of relationship among independent and dependent variables.
- ▶ Spline: a piecewise defined polynomial function that is smooth (possesses higher order derivatives) where polynomial pieces connect.
- ▶ Knot: the point at which two polynomial pieces connect.

Introduction to MARS

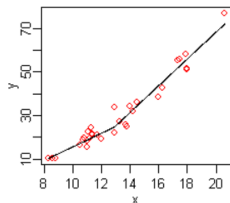
- ▶ MARS is a form of stepwise linear regression.
- ▶ Introduced by Jerome Friedman in 1991.
- ▶ In R, this method is implemented by package `earth`
- ▶ Suitable for higher dimensional inputs
- ▶ Extension of linear model that can model non-linearity.
- ▶ MARS models are simpler as compared to other models like random forest or neural networks.

Normal regression vs MARS



Normal Regression

$$y' = -37 + 5.1x$$



MARS

$$y' = 25 + 6.1 \max(0, x-13) - 3.1 \max(0, 13-x)$$

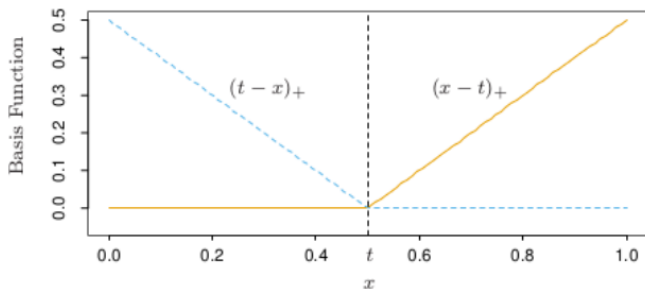
Figure 1: Linear regression vs MARS

In the MARS plot, you could see there is one knot at $x=13$.

Basis Functions

- MARS uses piecewise linear basis functions of the form $(x - t)_+$ and $(t - x)_+$. The $+$ means positive part only. so

$$(x - t)_+ = \max(0, x - t), \quad (t - x)_+ = \max(0, t - x)$$



Basis Functions

- ▶ MARS uses collection of functions comprised of reflected pairs for each input x_j with knots at each observed value x_{ij} of that input

$$C = \{(x_j - t)_+, (t - x_j)_+\}_{t \in \{x_{1j}, x_{2j}, \dots, x_{Nj}\}, j=1, 2, \dots, p.}$$

- ▶ If all input values are distinct, then set C contains $2np$ functions where
 - ▶ n = number of observations.
 - ▶ p = number of predictors or input variables.

MARS Model Equation

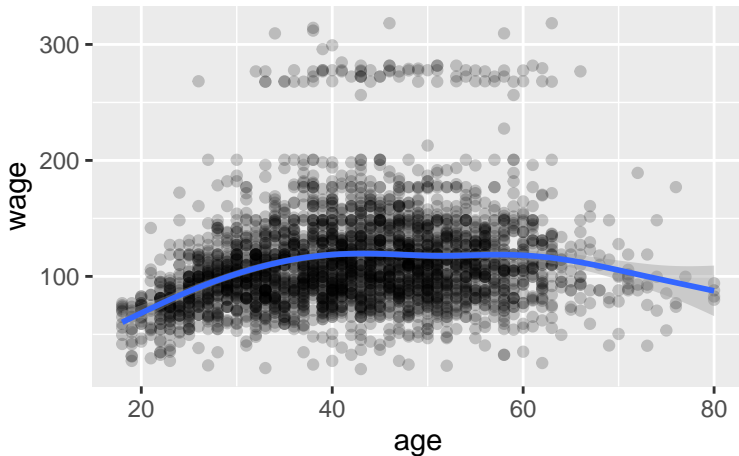
- ▶ MARS model has the general form

$$f(x) = \beta_0 + \sum_{k=1}^M \beta_k h_k(x)$$

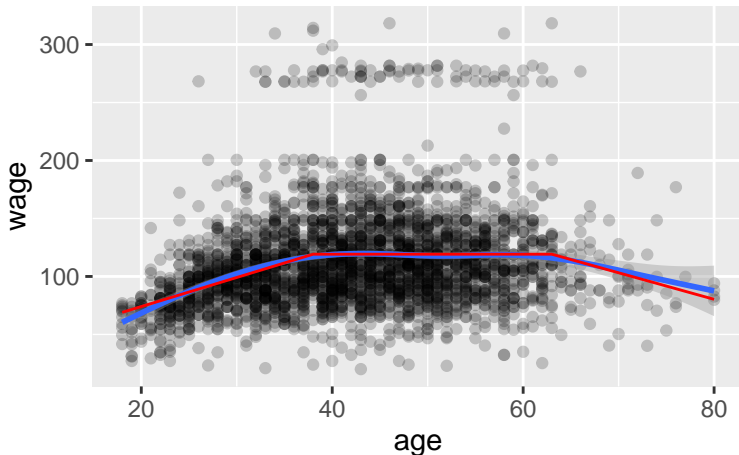
- ▶ $h_k(x)$ is a function from set C of candidate functions or a product of two or more such functions.
- ▶ β s are the coefficients estimated by minimizing the residual sum of squares (standard linear regression).
- ▶ These coefficients can be consider weights that represent the importance of the variable.

Example Data

```
library(tidyverse)
library(ISLR)
data(Wage) # help(Wage) for info
ggplot(Wage, aes(x=age, y=wage)) + geom_point(alpha=.2) + geom_smooth()
```



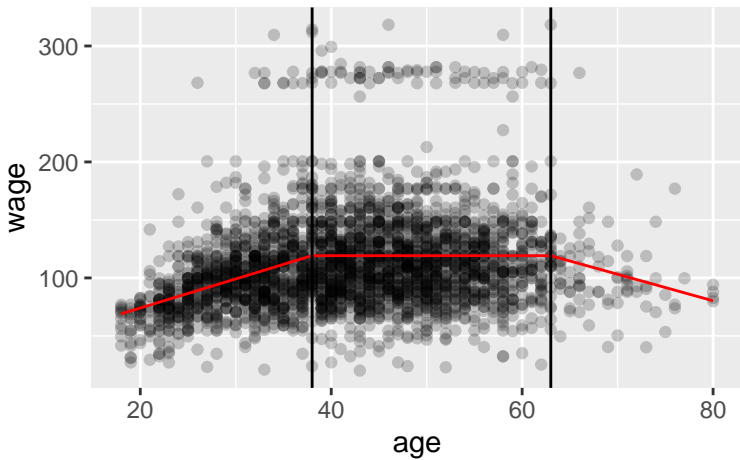

```
library(earth)
ee <- earth(wage ~ age, data=Wage)
Wage <- mutate(Wage, pwage = predict(ee))
ggplot(Wage, aes(x=age, y=wage)) + geom_point(alpha=.2) + geom_smooth()+
  geom_line(aes(y=pwage), color="red")
```



```
summary(ee)
```

```
## Call: earth(formula=wage~age, data=Wage)
##
##               coefficients
## (Intercept)    119.190151
## h(38-age)      -2.508377
## h(age-63)      -2.289070
##
## Selected 3 of 4 terms, and 1 of 1 predictors
## Termination condition: RSq changed by less than 0.001 at 4 terms
## Importance: age
## Number of terms at each degree of interaction: 1 2 (additive model)
## GCV 1595.44    RSS 4770379    GRSq 0.08405764    RSq 0.08649934
```

```
ggplot(Wage,aes(x=age,y=wage)) + geom_point(alpha=.2) +  
  geom_line(aes(y=pwage),color="red") +  
  geom_vline(xintercept=38) +  
  geom_vline(xintercept=63)
```



Hinge functions

- ▶ The points 38 and 63 are “knots” where the piece-wise linear function changes slope.
- ▶ The piece-wise linear fit is a linear model in a constant term (intercept) and two “hinge” functions, $h(38 - \text{age})$ and $h(\text{age} - 63)$, where

$$h(x) = \max(0, x)$$

- ▶ Hinge functions $h(x - c)$ and $h(c - x)$ are called mirror image.
 - ▶ Exercise: Plot two mirror-image hinge functions for `x <- seq(from=0,to=50,length=100)` and `c<-50`. Why are they called mirror image?

Fitting

- Once we are given the knots and hinge functions, the fit can be obtained by least squares.

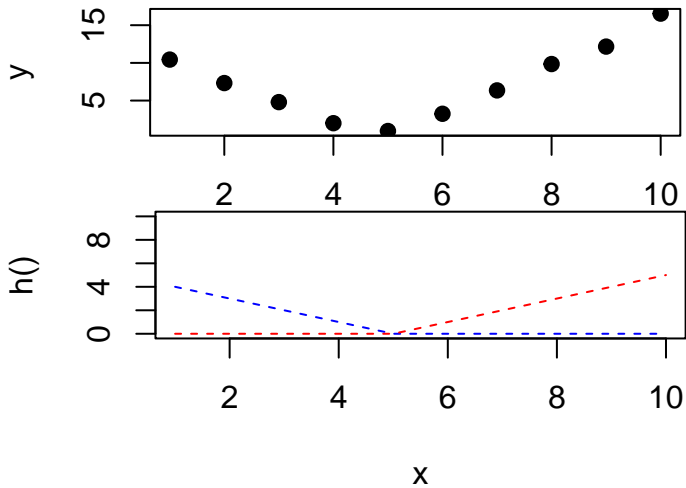
```
Wage <- mutate(Wage, h1=pmax(0, 38-age), h2=pmax(0, age-63))  
ff <- lm(wage ~ h1+h2, data=Wage)  
summary(ff)$coefficients
```

	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	119.190151	0.8539353	139.577500	0.000000e+00
## h1	-2.508377	0.1506008	-16.655805	1.216314e-59
## h2	-2.289070	0.5949343	-3.847601	1.217668e-04

Simulation study - Generating data

```
set.seed(360)
x <- seq(1,10,by=1)
h5x <- pmax(0,5-x);
hx5 <- pmax(0,x-5)
#plot(x,h5x,type="l",col="blue",lty=2,xlab="x",ylim=c(0,10))
#lines(x,hx5, lty=2,col="red")
y <- 1+2*h5x+3*hx5+rnorm(10,0,1);
cbind(x,h5x,hx5,y)
```

##		x	h5x	hx5	y
##	[1,]	1	4	0	10.4374946
##	[2,]	2	3	0	7.3225732
##	[3,]	3	2	0	4.7957034
##	[4,]	4	1	0	2.0009050
##	[5,]	5	0	0	0.9624999
##	[6,]	6	0	1	3.2485689
##	[7,]	7	0	2	6.3494051
##	[8,]	8	0	3	9.8481529
##	[9,]	9	0	4	12.1619673
##	[10,]	10	0	5	16.5373044



Simulation study: fitted coefficients with earth

```
mod <- earth(y ~ x);  
summary(mod)
```

```
## Call: earth(formula=y~x)  
##  
##              coefficients  
## (Intercept)    0.1304094  
## h(5-x)        2.4668639  
## h(x-5)        3.1794562  
##  
## Selected 3 of 3 terms, and 1 of 1 predictors  
## Termination condition: RSq changed by less than 0.001 at 3 terms  
## Importance: x  
## Number of terms at each degree of interaction: 1 2 (additive model)  
## GCV 0.8571923    RSS 2.142981    GRSq 0.9680172    RSq 0.9901288
```


Simulation study: fitted coefficients with `lm()`

```
mod2 <- lm(y~h5x+hx5);  
summary(mod2)
```

```
##  
## Call:  
## lm(formula = y ~ h5x + hx5)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -0.6863 -0.2534 -0.1006  0.3746  0.8321   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept)   0.1304     0.3448   0.378    0.716      
## h5x           2.4669     0.1530  16.123 8.59e-07 ***  
## hx5           3.1795     0.1200  26.489 2.80e-08 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 0.5533 on 7 degrees of freedom  
## Multiple R-squared:  0.9901, Adjusted R-squared:  0.9873   
## F-statistic: 351.1 on 2 and 7 DF,  p-value: 9.557e-08
```

Questions

For a given dataset

- ▶ How do we choose the knots?
- ▶ What happens when there are multiple explanatory variables, and we allow for interactions between them?

MARS Model Building Procedure

1. Gather data: x input variables with y observations each, giving a total of xy data points.
2. Calculate set of candidate functions by generating reflected pairs of basis functions with knots set at observed values.
3. Specify constraints; the number of terms in the model and maximum allowable degree of interaction.
4. Do forward pass: try out new function products and see which product decreases training error.
5. Do backward pass: fix overfit.
6. Do generalized cross validation to estimate the optimal number of terms in the model.

MARS Forward Pass

- ▶ At each step, MARS adds the basis function which reduces the residual error the most
- ▶ Always adds the basis function in “pairs”, both sides of knot
- ▶ Calculate value for knot and function that fit the data, least squares.
- ▶ This is greedy algorithm.
- ▶ The addition of model terms continues until the max number of terms in the model is reached.

MARS Backwards Pass

- ▶ Remove one term at a time from the model
- ▶ Remove the term which increases the residual error the least
- ▶ Continue removing terms until cross validation is statisfied
- ▶ Use the Generalized Cross Validation (GCV) function for this purpose.

Reference

1. Hastie, Tibshirani, and Friedman. The Elements of Statistical Learning (2nd ed.). Springer, 2009.
<http://www-stat.stanford.edu/~hastie/pub.htm>.
2. Jerome H. Friedman. Multivariate Adaptive Regression Splines (with discussion). Annals of Statistics, 1991