Modelos GAM y GAMLSS

Humberto Vaquera **Huerta**

2022-07-12

**Introducción**

Muchos datos en las agropecuarias no se ajustan a modelos lineales simples *y* se describen mejor con "modelos ondulantes", también conocidos como **Modelos aditivos generalizados (GAM)**.

Suponga un modelo de regresion lineal simple:

*y = ßo* +x1ß1 +ɛ, ɛ~ *N*(0,σ2)

Lo que cambia en un **GAM** es la presencia de un término de suavizado:

*y =* ßo + *f(x1*) + *ε*,

~ 3

*N*(0,σ2)

Esto simplemente significa que la contribución al predictor lineal ahora es alguna función *f.* Esto parecido al uso de un término cuadrático (*x2*) o cúbico ( x ) como su predictor.

3 1

la función *ƒ* generalmente se suviza con *splines*.

Puede tener combinaciones de términos lineales y suaves en su modelo, por ejemplo

*y =* Bo+x1ẞ1 + *f(x2*) +ε, ε ~ *N*(0,σ2)

2

**Introducción**

En el modelo lineal generaliado la forma de introducir el efecto de las variables predictivas en el modelo se da usando la funciones*, n*(xi) = Σj=*1 ßjxij*, las cuales pueden ser restrictivas ya que no introducen efectos no

lineales en la relacion.

El **modelo Generalized additive (GAM)**, introducido por **Hastie**, **T. y Tibshirani**, **R. (1986)**., permite una mayor flexibilidad al modelar el predictor lineal de un modelo lineal generalizado como una suma de funciones más generales de cada variable:

*Р*

n(x;) = Σƒj (xij),

Σf;

*j*=1

donde *f;* son funciones desconocidas, se supone que son suaves o de baja complejidad.

Hastie, T. and Tibshirani, R. (1986). Generalized additive models. Statistical science 297-310.

3

**Modelo** GAM

**Generalized Additive Model**

*Yi*

~

EF(μi, *0*)

donde:

*g(μi)* = A¿Y + Σƒj *(x ji)*,

j

yi es una variable de respuesta univariada, EF(μi, ) denota una distribución de **familia exponencial** con media *μ¿* y parámetro de escala *o*, Ai es la *i* — *ésima* fila de una matriz de modelo paramétrico, y son parámetros de regresión, *ƒ;* son funciones suaves a estimar, y x*;* es una covariable (normalmente, pero no necesariamente, univariante).

j

• Se relaja el supuesto de relación lineal entre variable dependiente y predictor

• Relación entre predictores individuales y dependientes (posiblemente transformados)

• La variable se estima mediante una función suave no lineal:

*g*(y) = *s*(x1) + *s*(x2, x3) + ß4x4+...

4

El termino *f(x,)* podrían ser regresiones tipo kernel o splines, pero algún termino lineal en funciones de *xj*.

comp:

•

polinomios creados por poly();

splines de regresión creados por bs ();

splines cúbicos naturales creados por ns ();

• funciones de paso creadas por cut (); o

• variables categóricas (ficticias) creadas por factor

**LO**

5

**Ejemplo Ilustrativo**

**Suponga una función**

*f(x*) = −3.5 + 0.2x11 (10(1 − x))6 + 10(10x)3(1 − x)10

set.seed(0);n <- 400; x <- 0:(n−1) / (n-1)

f <**-** -3.5+0.2\*x^11\*(10\*(1-x))^6+10\*(10\*x)^3\*(1−x)^10

y <- f + rnorm(n, 0, sd = 3)

plot(x,y); lines(x, f, col="red", lwd=2)

y

-5

0

10

**C**

0.0

0.2

0.4

X

0.6

0.8

1.0

6

**Estimación** de la **curva** con **GAM y splines**

**library**(mgcv)

fit.gam <- gam(y~ s(x)) plot(fit.gam)

lines(x, f, col="red", lwd=1.5)

lines (x, y, col="blue")

s(x,7.87)

9

-4

024

0.0

0.2

0.4

X

0.6

0.8

1.0

7

**Ejemplo:** Curva de **lactancia de una vaca**

Rendimientos medios diarios de grasa *(kg/día*) de la leche de una sola vaca durante cada 35 semanas.

**library**(tidyverse)

**library**(agridat)

data (henderson.milkfat)

ggplot (henderson.milkfat, aes(x

=

week)) +

geom\_point (aes (y = yield), col="blue", size

||

=

2, alpha

=

0.5)

||

yield

0.6-

0.4-

0.2-

0.0-

1

10

week

20

30

-10

8

**Ajuste** GAM

~

s(week), data=dat)

dat-henderson.milkfat

libs ("mgcv")

modelo\_milk<- gam(yield summary (modelo\_milk)

##

## Family: gaussian

## Link function: identity

##

## Formula:

## yield

~

s (week)

##

## Parametric coefficients:

##

Estimate Std. Error t value Pr(>|t|)

## (Intercept) 0.398571 0.004763 83.69 <2e-16 \*\*\* ##

## Signif. codes: Θ \*\*\* 0.001 \*\* 0.01 \* 0.05

##

## Approximate significance of smooth terms:

##

edf Ref.df

F p-value

## s (week) 7.79 8.623 235.5 <2e-16 \*\*\*

•

0.1'' 1

##

## Signif. codes:

Θ

\*\*\* 0.001

0.01 '' 0.05 \*\*

' 0.1

' 1

9

yield

0.0-

**Curva ajustada con** GAM

ggplot (henderson.milkfat, aes(x

geom\_point() +

=

week, y=yield)) +

geom\_smooth (method = "gam", formula = y ~ s(x))

0.2-

0.4-

0.6-

0

10

-o

20

week

30

3-0

10

plot(yield ~ week, data = dat)

lines (dat$week, predict (modelo\_milk), lty = 1,col="red")

yield

0.0

0.2

0.4

0.6

0

5

10

15

O

week

Ο

O

O

20

25

O

O

O

O

30

35

11

Curva **ajustada con** GAM

**library**(broom) libs("mgcv")

modelo\_milk<- gam(yield

~

s(week), data=dat)

knitr::kable (tidy (modelo\_milk))

**term**

**edf ref.df statistic p.value**

s(week) 7.790113 8.622723 235.4643

0

knitr::kable (glance (modelo\_milk))

**df**

**logLik**

**AIC**

**BIC deviance df.residual nobs**

8.790113 80.32304 -141.0659 -125.8388 0.020808 26.20989 35

knitr::kable (performance:: performance (modelo\_milk))

**AIC**

**BIC**

**R2 RMSE Sigma**

12

Frequency

0

**Diagnostico** GAM

libs("mgcv")

modelo\_milk<- gam(yield gam.check (modelo\_milk)

~

s(week), data=dat)

deviance residuals

-0.04

-0.06

-0.04 -0.02 0.00 0.02 0.04 0.06

theoretical quantiles

**Histogram of residuals**

-0.06 -0.04 -0.02 0.00 0.02

0.04

0.06

Residuals

Response

residuals

-0.04

0.0 0.1

**Resids vs. linear pred**.

0.2 0.3 0.4 0.5

0.6

0.7

linear predictor

**Response vs. Fitted Values**

: Laagpo q

0.0 0.1 0.2

0.3 0.4 0.5 0.6

0.7

Fitted Values

13

**Ejemplo:** Biomasa

Datos:

Allometric Equations for Aboveground and Belowground Biomass Estimations in an Evergreen Forest in Vietnam Nam VT, van Kuijk M, Anten NPR (2016) Allometric Equations for Aboveground and Belowground Biomass Estimations in an Evergreen Forest in Vietnam. PLOS ONE 11(6): e0156827. https://doi.org/10.1371/journal.pone.0156827

**Ajuste de curva** alometrica **para la** especie ***Endospermum*** chinensis

Datos:

biomasa=read.csv("biomasa.csv"**)**

biomasa

##

N\_individual WD.class

Species\_names DBH\_cm

H\_m WD\_gcm3

AGB\_kg

## 1

17

## 2

18

I Endospermum chinensis 3.2 I Endospermum chinensis

6.0

0.45

1.46

3.6 5.3

0.45

1.76

## 3

19

I Endospermum chinensis

8.3 12.4

0.45

13.96

## 4

20

I Endospermum chinensis

8.9 10.2

0.45

18.61

14

**Modelos alometricos**

Modelos alométricos que relacionan las medidas geométricas (*DAP, H y WD*) con la **biomasa aérea (AGB)**

Several equations are commonly used to develop allometric models for AGB and RB.

By Brown et al. [26] and Brown [1] (pan-tropical):

In*(B*) = a+bln*(DBH*)

**(1)**

In(*B*) = *a+*bln*(DBH*) + *by* In(*DBH*)*2*

**(2**)

By Nelson et al. [27] (central Amazon):

==

In*(B*) = a *+* bln*(DBH*) + *d* In*(H*)

By Chave et al. [3] (pan-tropical):

=

In(*B*) *a+b* ln*(DBH*) + *c* ln(*WD)* + *d* ln*(H*)

In(*B*) = *a* + *b* ln*(DBH*) *+ e*(ln*(DBH*)*)2+f*(In*(DBH*))3 + *c* In(*WD*)

In(*B*) = *a*+*g* ln*(DBH2HWD)*

By Djomo et al. [28] (tropical Africa):

In(*B*) = a*+*bln(*DBH)* + *c* ln(*WD)*

**(**3)

(3)

**(4)**

**(5)**

(6**)**

15

**Curva alometrica para la especie Endospermum chinensis con** GAM

ggplot(biomasa, aes (x = DBH\_cm,y=AGB\_kg)) +

geom\_point () +

geom\_smooth (method = "gam", formula = y ~ s(x))

AGB\_kg

6000-

4000-

2000-

0-

0

25

1-55

50

DBH\_cm

-F5

75

100

16

Ejemplo: **Datos** de propiedades del **suelo**

Varias propiedades químicas del suelo medidas en una cuadrícula regular con 10x25 puntos espaciados por 5 metros. 250 observaciones sobre las siguientes 22 variables:

• Coordenada x de Linha

• Coordenada y de Coluna

• Elevación de la cota

• AGrossa un vecto numérico, una porción de arena de la muestra.

• Silte un vector numérico, porción de limo de la muestra.

•

Argila un vector numérico, una porción de arena de la muestra.

PHAgua un vector numérico, pH del suelo en agua

pHKCI un vector numérico, pH del suelo por KCI

• Ca un vector numérico, contenido de calcio Mg a vector numérico, contenido de magnesio

• K un vector numérico, contenido de potasio

• Al un vector numérico, contenido de aluminio

• H un vector numérico, contenido de hidrógeno

17

Ejemplo: **Datos** de propiedades del **suelo**

Varias propiedades químicas del suelo medidas en una cuadrícula regular con 10x25 puntos espaciados por 5 metros. 250 observaciones sobre las siguientes 22 variables:

cont...

• Cun vector numérico, contenido de carbono

• N un vector numérico, contenido de nitrógeno

• CTC un vector numérico, capacidad de intercambio de cationico

• S un vector numérico, contenido enxofrar

• V un vector numérico

• M un vector numérico

• NC un vector numérico

• CEC un vector numérico

• CN un vector numérico, relación carbono / nitrógeno \_

18

**Sitios de muestreo del suelo**

soil250=read.csv("soil250.csv") ggplot(soil250, aes (Coluna, Linha)) +

geom\_point(size = .25, show. legend coord\_quickmap()

Linha

40-

30-

20-

10-

0-

0

25

=

FALSE) +

6-9

50

Coluna

I

75

100

125

19

**library**(lattice)

p <- cloud (K~ Linha Coluna, pch

\* print(p)

=

".", data =

soil250)

K

Coluna

Linha

20

**Grafica de distribución del Potasio**.

lat=soil250$ Linha

long-soil250$ Coluna

K=soil250$K

**library**(scatterplot3d)

scatterplot3d (lat, long,K, pch=16, highlight.3d=TRUE,

type="h", main="3D Scatterplot")

21

**library** (mgcv)

Ajuste K <- gam(K~te (lat, long)) summary (Ajuste\_K)

##

## Family: gaussian

## Link function: identity

##

## Formula:

## K

##

~

te (lat, long)

**##** Parametric coefficients:

##

Estimate Std. Error t value Pr(>|t|)

0.004897 95.96 <2e-16 \*\*\*

## (Intercept) 0.469920

##

I

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '' 0.05 ' 0.1'' 1

##

## Approximate significance of smooth terms:

##

edf Ref.df

F p-value

## te (lat, long) 20.49 22.82 15.28 <2e-16 \*\*\*

##

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05

##

## R-sq. (adj)

= 0.579 Deviance explained = 61.4%

n

= 250

## GCV = 0.0065595 Scale est. = 0.0059956

1

' 0.1'' 1

22

**library**(mgcv*)*

Ajuste K <- gam (K~te (lat, long)) plot (Ajuste\_K, pers = TRUE)

plot (Ajuste\_K)

te(lat, long,20.49)

-1se

20

120

long

08

40

0

0

Tat

биор

0.15-

-0.05 -0.05-

te(lat, long,20.49)

0.05

10

20

20

0.05

0

0.05

+1se

-0.151

0.05

30

309

lat

10

40

23

numero\_div <- 100

lat-seq (min(lat),max(lat), length=numero\_div)

long-seq (min(long), max(long), length=numero\_div) datos\_malla-expand.grid(long

=

# *prediccion de S04 en* la *malla*

long, lat=lat)

K\_pred =matrix(predict(Ajuste\_K, datos\_malla), numero\_div, numero\_div) *#head (K\_pred, 3)*

p <- persp(lat, long, K\_pred, theta = 10, col = "yellow")

**library** (plot3D)

# *Añadir observaciones de K*

obs <- trans3d (lat, long, K, p)

pred <- trans3d (lat, long, fitted (Ajuste\_K), p)

points (obs, col = "red", pch = 16)

24

**library**(ggplot2)

ggplot(mutate(datos\_malla,

p <- ggplot(mutate (datos\_malla, fit = as.numeric(K\_pred)),

aes(x = lat, y = long, z = fit)) +

coord\_fixed()

*#p* + *geom\_contour()*

p + geom\_tile (aes (fill = fit)) + geom\_contour() + scale\_fill\_gradientn (colors = rev (cm.colors (100)))

125-

100-

**fit**

0.6

75-

0.5

0.4

50-

0.3

long

25-

*0-*

0 10 20 30 40

lat

25

long

**library**(rgl)

*#plot3d (Ajuste\_K)*

plot3d(Ajuste\_K, image = TRUE, contour = TRUE)

**library**(mgcViz)

b <- getViz (Ajuste\_K)

plot(sm(b, 1))+l\_fitRaster () + l\_fitContour() + l\_points() + l\_rug()

120

te(lat, long, 20.49)

40

80

0

+

+

0

10

20

30

40

lat

s(x)

0.1

0.0

-0.1

-0.2

26

**Grafica ajuste GAM Potasio**

**library**(knitr)

include\_graphics("potasio1.png")

te(lat,long, 20.49)

0.0

0

120 100 80

60

40

20 0

10

20

20

30

40

lal

27

Rendimiento de **Maiz en Mexico por Hectarea**

Datos: Datos de FAO http://www.fao.org/faostat/en/#data/QC

**Year Rend\_maiz Year Rend\_maiz**

1961 9934

1989 16929

1962 9946

1990 19942

1963 9867

1991 20515

1964 11332

1992 23450

1965 11578

1993 24401

1966 11188

1994 22255

1967 11304

1995 22883

1968 11806

1996 22387

1969 11840

1970 11935

1997 23840

1998 23429

1971 12723

1999 24720

28

**Rendimiento** de **Maiz en Mexico Hg/Hectarea**

maiz.rend <- read.csv("D:/cursos/no param/noparam/ejemplo gam/maiz-rend.csv") rend\_mex <- ts (maiz.rend$Rend\_maiz, start=c(1961), end=c(2018), frequency=1)

plot(rend\_mex, col="blue", lwd=2,main="Rendimiento por Ha de Maiz en Mexico 1961-1980")

rend\_mex

10000 20000 30000

**Rendimiento por Ha de Maiz en Mexico 1961-1980**

1960

1970

1980

1990

2000

2010

2020

Time

29

**Ajuste** con GAM

gam\_Maiz <- gam(Rend\_maiz~ s(Year,bs="cr",k=55),

data =maiz.rend)

summary (gam\_Maiz)

##

## Family: gaussian

## Link function: identity

##

## Formula:

## Rend\_maiz

##

~

s(Year, bs =

"cr", k = 55)

## Parametric coefficients:

##

## (Intercept)

##

Estimate Std. Error t value Pr(>|t|)

20984.7

81.3 258.1 <2e-16 \*\*\*

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1

##

## Approximate significance of smooth terms:

1

##

edf Ref.df

F p-value

## s(Year) 23.61

29.03 350.9 <2e-16 \*\*\*

##

## Signif. codes:

0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 '

1

30

**Prediccion**

**library**(ggplot2)

layout(matrix (1:2, nrow = 1))

predic=data.frame(cbind(predic\_rend=predict(gam\_Maiz), maiz.rend)) ggplot(predic, aes(x=Year)) +

geom\_line(aes (y = Rend\_maiz), color ="blue") +

geom\_line(aes (y = predic\_rend), color="red", linetype="twodash")

31

prediccion **rendimiento** de Maiz **Hg/**Ha **2019-2020**

a\_2019\_2020-data.frame (Year=c(2019, 2020))

p\_2019\_2020=predict(gam\_Maiz,a\_2019\_2020) data.frame(cbind(a\_2019\_2020,p\_2019\_2020))

##

Year p\_2019\_2020

## 1 2019

## 2 2020

38975.02

39542.21

pred <- predict.gam(gam\_Maiz,a\_2019\_2020, se = TRUE ) $fit se <- predict.gam(gam\_Maiz,a\_2019\_2020, se = TRUE) $se.fit

# *Intervalos de confianza*

—

lcl\_95 <- pred 1.96 \* se

ucl\_95 <- pred + 1.96 \* se

data.frame(cbind(a\_2019\_2020, p\_2019\_2020, lcl\_95, ucl\_95))

##

Year p\_2019\_2020

lcl\_95

ucl\_95

## 1 2019

## 2 2020

38975.02 36801.44 41148.59

39542.21 36164.68 42919.73

32

**Ejemplo** 2

Los GAM son similares a los modelos lineales generalizados (GLM), pero difieren al relajar el supuesto lineal, lo que potencialmente revela relaciones no lineales y una estructura importante en estos datos que de otro modo se perderían. En este sentido, un GAM puede mostrar relaciones lineales, monótonas o más complejas, dependiendo de la forma en que cada variable responda a los cambios en las variables dependientes. El GAM funciona de esta manera al extender un GLM para incluir una función de base de suavizado que puede medir relaciones arbitrariamente no paramétricas. Los datos categóricos se tratan como un término lineal sin suavizado. Este enfoque semiparamétrico hace que los GAM sean muy flexibles y se adapten fácilmente a diferentes tipos de datos.

**Datos**

Con el set de datos **Auto** (información sobre vehículos), trataremos de predecir mpg (consumo de combustible, en millas/galón) en función de la variable predictora displacement (desplazamiento del motor, en pulgadas cúbicas), Junto con la variable horsepower (potencia del motor).

33

**Datos de Autos**

**library** (ISLR)

knitr::kable(head (Auto, 10))

**mpg cylinders displacement horsepower weight acceleration year origin name**

18

8

307

130 3504

12.0 70

1 chevrolet chevelle malibu

15

8

350

165 3693

11.5 70

1 buick skylark 320

18

8

318

150

3436

11.0

70

1 plymouth satellite

16

8

304

150

3433

12.0

70

1 amc rebel sst

17

8

302

140

3449

10.5

70

1 ford torino

15

14

∞ ∞

8

429

198

4341

10.0

70

1 ford galaxie 500

8

454

220

4354

9.0

70

1 chevrolet impala

14

8

440

215

4312

8.5

70

1 plymouth fury iii

14

8

455

225

4425

10.0

70

1 pontiac catalina

15

8

390

190

3850

8.5

70

1 amc ambassador dpl

34

par (mfrow=c(2,3))

**attach** (Auto)

plot(mpg

~

as.factor(cylinders))

plot(mpg~ weight)

plot(mpg

~

displacement)

plot(mpg plot(mpg plot(mpg

~

~

~

horsepower)

as.factor(year))

as.factor(origin))

10

mpg

30

10

mpg

30

3

4

5

6

as.factor(cylinders)

50

100

150

200

horsepower

++

∞

mpg

10

30

mpg

30

10

1500

2500

3500

4500

weight

+--0-

22

· -0- -·

+H

+H

H

HIH

+--+

-

72

74

76

78 80

82

-H

++

+2

70

as.factor(year)

mpg

10 30

mpg

30

10

ודד

8

80 80 8

100

200

300

400

displacement

2

3

as.factor(origin)

35

**library**(caret)

set.seed(12345)

inTraining <- createDataPartition (Auto$name, p=.6, list = FALSE)

datosA.train <- Auto[ inTraining,]

datosA.test <- Auto [-inTraining,]

datosA.test

mpg cylinders displacement horsepower weight acceleration year origin

##

## 9

14.0

## 16 22.0

8

455

225

4425

10.0 70

1

6

198

95

2833

15.5

70

1

## 25 21.0

6

199

90

2648

15.0

70

1

## 38 18.0

6

232

100 3288

15.5

71

1

## 61 20.0

4

140

90

2408

19.5

72

1

## 63

13.0

8

350

165

4274

12.0

72

1

## 65 15.0

8

318

150

4135

13.5

72

1

## 66 14.0

8

351

153

4129

13.0

72

1

## 92 13.0

8

400

150

4464

12.0

73

1

## 128 19.0

6

232

100

2901

16.0

74

1

## 129 15.0

6

250

100

3336

17.0

74

1

## 131 26.0

4

122

80

2451

16.5

74

1

## 168 29.0

4

97

75

2171

16.0

75

3

## 172 24.0

4

134

96

2702

13.5

75

3

## 175 18.0

## 186 26.0

## 190 15.5

## 191 14.5

## 194 24.0

64∞ ∞ ∞

171

97 2984

14.5

75

1

98

79

2255

17.7

76

1

8

304

120

3962

13.9

76

1

8

351

152

4215

12.8

76

1

6

200

81

3012

17.6

76

1

36

**library**(leaps) **library**(mgcv*)*

*#Modelo GAM*

modelo.gam <- gam(mpg

par (mfrow

=

c(2, 3))

~

cylinders + year + origin + s(displacement) **+** s(horsepower) + s (weight),

plot(modelo.gam, se = T, col = "red")

da

100

200

300

400

50

100

150

200

1500

2500

3500

4500

displacement

horsepower

weight

37

summary (modelo.gam)

##

## Family: gaussian

## Link function: identity

##

## Formula:

## mpg

~

##

##

cylinders + year + origin + s(displacement) + s(horsepower) +

s (weight)

## Parametric coefficients:

##

## (Intercept) -38.48255

Estimate Std. Error t value Pr(>|t|)

4.06060 -9.477 <2e-16 \*\*\*

## cylinders

0.23970

## year

0.78511

## origin

0.64628

0.27448

0.36425 0.658

0.04694 16.725

2.355

0.5109

<2e-16 \*\*\*

0.0191 \*

##

## Signif. codes:

Θ \*\*\* 0.001 \*\* 0.01 \* 0.05

0.1 1

## s(horsepower)

##

## Approximate significance of smooth terms:

##

edf Ref.df

F p-value

## s(displacement) 1.948 2.484 2.255 0.0849

## s(weight)

##

2.421 3.076 9.984 2.45e-06 \*\*\*

2.281 2.904 28.129 < 2e-16 \*\*\*

## Signif. codes:

0 '\*\*\*' 0.001 '\*\* 0.01 \* 0.05

' 0.1

1

##

38

**Modelos aditivos generalizados** para posición, **escala** y forma (**GAMLSS)**

Los modelos lineales generalizados (GLM) y los modelos generalizados aditivos (GAM) asumen que la variable respuesta Y sigue una distribución de la familia exponencial, cuya media μ puede ser modelada en función de otras variables (predictores) y cuya varianza o se calcula mediante una constante de dispersión y una función v(μ). Esto último significa que, la varianza, skewness y kurtosis, no se modelan directamente en función de las variables predictoras sino de forma indirecta a través de su relación con la media.

Los modelos GAMLSS, introducidos por Rigby y Stasinopoulos en 2005, permiten, además de incorporar distribuciones que no son de la familia exponencial, modelar explícitamente cada uno de sus parámetros en función de las variables predictoras empleando funciones lineales *y* no lineales.

Rigby, R.A. and Stasinopoulos, D.M. (2005), Generalized additive models for location, scale and shape. Journal of the Royal Statistical Society: Series C (Applied Statistics), 54: 507-554.

https://doi.org/10.1111/j.1467-9876.2005.00510.x

39

**Modelos aditivos generalizados para posición**, **escala y** forma **(GAMLSS)**

Los términos empleados dentro del marco de los GAMLSS para referirse a los parámetros de localización, escala y forma son (*μ,* σ, V*, T*).

**Y** ~ *D(μ*, O, *V*, *T*), con **Y** = XTB

donde **Y**

~

*T*

n1 = 91(μ) = x2B+*f1(*x1) + *f2(x2*)+...+fp(xp)

N1 *91*

*N2* = *92*(0) = XTB + f1(x1) + ƒ2*(*x2)+...+ƒp(Xp)

*N3* = *93(v)* = XTB + *ƒ1*(x1) + ƒ2*(*x2)+...+*ƒp*(xp)

*N1* = 94(T) = XTB + f1(x1) + f2*(x2*)+...+fp(xp)

*D(μ, σ*, *v, t*) es la distribución de la variable respuesta (pueden ser menos parámetros), **X** contiene los términos lineales del modelo, *ẞ* son los coeficientes lineales y fi*(xi*) son funciones de suavizado no lineales (smooth) de cada predictor.

40

**GAMLSS**

Los GAMLSS son una forma de superar las limitaciones de los modelos GLM y GAM. Como resultado del modelo, se consigue caracterizar la distribución completa, permitiendo generar intervalos probabilísticos y predicción de cuantiles.

LM, GLM

GAM

GAMLSS

*Imagen de: Schlosser, Lisa & Hothorn, Torsten & Stauffer, Reto* & *Zeileis, Achim. (2018). Distributional Regression Forests for Probabilistic Precipitation Forecasting in Complex Terrain. The Annals of Applied Statistics*. *13. 10.1214*/19*-AOAS1247.*

41

**GAMLSS**

Suponiendo una variable dependiente de una distribución con parámetros

Y1**,**..., *Yn,* dadas las covariables 21, especificación de modelo:

•

•

•

**,**

=

*(01*, ...*, 0L*)' y observaciones ZK у X1, ..., *XQ,* a *gamlss* se pueden describir con la siguiente

У

Κι

*gi*(*01) = m* = *Xißi* + Σ *ZjlYjl*

*j*=1

42

En la Ecuación, *gi*(·) representa una función de enlace monótona conocida, que puede ser diferente para cada parámetro. *X*, representa el subconjunto de todas las variables disponibles *x* = (x1**,**...,*xQ*)' usado para

modelar el parámetro *0*, mientras que *Zjɩɩ* sirve como la matriz de función básica para un efecto no paramétrico de la covariable *z;* en el parámetro *1,* tomado de un subconjunto de variables 21, . . ., Z. El subconjunto

*Zj* específico de covariables z con efectos no paramétricos en el parámetro *0,* tiene una longitud de *KĮ* variables.

43

**Ajuste** de **los modelos GAMLSS**

Los modelos GAMLSS se ajustan mediante una adaptación del algoritmo backfitting, un algoritmo típicamente empleado para el ajuste de modelos aditivos. Por ejemplo, para el modelo aditivo

*g*(μ) = ßo + *f1*(x1) + *ƒ2*(x2)+...+*fp*(*Xp)*

en términos generales, el algoritmo de ajuste por *backfitting* funciona de la siguiente forma:

• Se inicia con un valor para todos los términos *fi* del modelo, por ejemplo poniéndolos todos a cero.

• Se estima el valor del primer término *f1* empleando los datos de entrenamiento, con el objetivo de predecir lo mejor posible la variable respuesta y.

• Se estima el valor del segundo término *f2* empleando residuos *y - f1*(x1).

• Se repite el paso 3, ajustando cada término con los residuos del anterior.

Tras ajustar todos los términos, el valor del primer término se descarta y se reajusta empleando el residuo de todos los demás términos.

Repetir todos los pasos del 3 al 5 hasta que se alcance un criterio de parada (que los valores apenas cambien o que se alcance un número máximo de iteraciones).

44

**Selección de modelos GAMLSS**

Para identificar el mejor modelo GAMLSS es necesario comparar diferentes modelos candidatos, cada uno con una combinación de distribución para la variable respuesta, función link para cada uno de los parámetros, predictores e hiperparámetros. Existen varias estrategias para la selección del mejor modelo de entre los comparados:

•

**Generalized Akaike information criterion (GAIC):** esta métrica emplea el log likelihood del modelo

multiplicado por -2 y añade una penalización por cada parámetro que incluye el modelo. Más detalles sobre el GAIC en el apartado métricas de ajuste.

• **Generalized cross-validation y cross-validation**

El segundo método es más recomendable, aunque supone mayor costo computacional.

45

**GAMLSS**

2

Por ejemplo, al asumir la distribución gaussiana para la variable dependiente y conectar *μ* a efectos paramétricos *\** usando la función de enlace de identidad (*g*(μ) = μ) y el parámetro de varianza *☛2* a una constante, llegamos a una especificación de modelo lineal.

*q*

46

**Datos** de **Wage**

collected by the United States Census Bureau (2011), includes 3000 male individuals living in the Mid-Atlantic region of the United States of America with records of the following variables:

• wage: Worker's raw wage (in $1000)

age: Age of worker

year: Year that wage information was recorded

race: A factor with levels 1. White, 2. Black, 3. Asian and 4. Other

education: A factor with levels 1. < HS Grad, 2. HS Grad, 3. Some College, 4. College Grad and 5. Advanced Degree

• health: A factor with levels 1. <= Good and 2. > Very Good, indicating the health level of worker.

47

**Datos de Autos**

**library** (ISLR)

knitr::kable (head (Wage))

**year age maritl**

**race**

**education region jobclass**

**health health\_ins\_logwage**

**wage**

2.

231655 2006

18

1. Never

Married

1.

White

1. <HS

Grad

Middle 1. Industrial

Atlantic

1.

<=Good

2. No

4.318063 75.04315

2.

2.

1. Never

1.

86582 2004 24

Married

4. College

2.

White Grad

Middle

Atlantic

Information

>=Very 2. No Good

4.255273 70.47602

2.

2.

1.

3. Some

161300 2003 45

Middle 1. Industrial

Married

White

White College

1.

<=Good

1. Yes

4.875061 130.98218

Atlantic

2.

2.

2.

3. 4. College

2.

155159 2003

43

Middle

Married

Asian Grad

Information

Atlantic

>=Very 1. Yes Good

5.041393 154.68529

48

**Datos Wage**

**library**(gamlss)

modelo\_gamlss <- gamlss(

formula = wage sigma. formula =

~

~

family

=

NO, data

bs(age) + race + year + education + health,

bs (age) + race + year + education + health,

=

Wage,)

## GAMLSS-RS iteration 1: Global Deviance = 29292.76

## GAMLSS-RS iteration 2: Global Deviance = 29277.64

## GAMLSS-RS iteration 3: Global Deviance = 29277.57

## GAMLSS-RS iteration 4: Global Deviance = 29277.57

summary (modelo\_gamlss)

## \*\*\*\*\*\*\*\*

\*\*

\*\*

\*\*\*

## Family: c("NO", "Normal")

##

## Call: gamlss (formula = wage

##

##

##

~

bs(age) + race + year + education + sigma.formula = ~bs(age) + race + year + education + health, family = NO, data = Wage)

## Fitting method: RS ()

health,

49

**Datos Wage**

summary (modelo\_gamlss)

## \*\*\*\*\*\*\*\*

\*

## Family: c("NO", "Normal")

##

## Call: gamlss (formula = wage

##

##

##

sigma.formula = ~bs (age)

~

+

family

=

NO, data =

Wage)

\*\*

bs (age) + race + year + education + health,

race + year + education + health,

## Fitting method: RS()

##

##

## Mu link function: identity

## Mu Coefficients:

##

## (Intercept)

Estimate Std. Error t value Pr(>|t|)

-2057.7691 528.9249 -3.890 0.000102 \*\*\*

## bs(age)1

63.8698

6.4926

9.837 < 2e-16 \*\*\*

## bs(age)2

42.4229

5.4234

7.822 7.14e-15 \*\*\*

## bs(age)3

23.9740

6.2488

## race2. Black

-2.7589

1.9174

3.837 0.000127 \*\*\*

-1.439 0.150295

## race3. Asian

-6.4240

2.4237

-2.650 0.008080 \*\*

## race4. Other

-4.8280

4.0419 -1.194 0.232380

50

## \*

##

##

##

##

##

Datos **Wage**

plot(modelo\_gamlss)

Density

0.0

Quantile Residuals

-2

**Against Fitted Values**

60

80

100 120

140 160

Fitted Values

**Density Estimate**

-4

-2

0 2

4

6

8

Quantile. Residuals

Summary of the Quantile Residuals

mean

= 0.002149396

variance = 1.00033

coef. of skewness = 1.407693

coef. of kurtosis = 9.270535

Sample Quantiles

Quantile Residuals

-2

**Against index**

0

500

1000 1500 2000 2500 3000

index

-3

-2

**Normal Q-Q Plot**

-1 0 1 2 3

Theoretical Quantiles

\*\*

51

**Datos Wage**

performance::performance(modelo\_gamlss)

## # Indices of model performance

##

## AIC

##

BIC RMSE | Sigma

## 29329.574 | 29485.740 | 1.000 | 3.131

52

Datos **Wage**

**library**(mgcv)

modelo\_gam2 <- gam( wage summary (modelo\_gam2)

~

s(age) + race + year + education + health,data

##

## Family: gaussian

## Link function: identity

##

## Formula:

## wage ##

~

s(age) + race + year + education + health

## Parametric coefficients:

##

## (Intercept)

## race2. Black

Estimate Std. Error t value Pr(>|t|)

-2375.5642 633.9466 -3.747 0.000182 \*\*\*

-6.0253

2.1768 -2.768 0.005676 \*\*

## race3. Asian

## race4. Other

## year

## education2. HS Grad

## education3. Some College

## education4. College Grad

||

=

Wage,)

-3.1399

2.6706 -1.176 0.239808

-6.7914

5.8206 -1.167 0.243392

1.2251

10.3323

22.5689

36.1930

0.3161 3.876 0.000108 \*\*\*

2.4197 4.270 2.01e-05 \*\*\*

2.5530 8.840 < 2e-16 \*\*\*

2.5543 14.169 < 2e-16 \*\*\*

53

**Datos Wage**

plot(modelo\_gam2)

-40

s(age,4.87)

-20

0 10

20

30

40

50

60

70

80

age

54

**Datos Wage**

performance::performance(modelo\_gam2)

## # Indices of model performance

##

## AIC

##

BIC

R2 | RMSE |

Sigma

## 29850.496 | 29945.831 | 0.299 | 34.843 | 34.954

55