Generalized Additive Models

(An introduction)

R. Bellio & N. Torelli

Spring 2018

University of Udine & University of Trieste

Nonlinear regression

Semiparametric regression: an introductory example

Some theory

Generalized Additive Models (GAMs)

Nonlinear regression

Intro: the limitations of linearity

Models that are built upon the linear effects of predictors, such as **linear models** or **generalized linear models**, play a crucial and non-replaceable role in the applications of statistics.

Linearity is always an approximation, which in many cases it leads to very sensible results.

Yet, there are instances where linearity is too strong a limitation, preventing the development of realistic models. That's why **nonlinear models** are important in statistics.

Classes of nonlinear models

Whereas linear models (including also GLMs) are easy to characterize, nonlinear models may be of several different types.

Some important instances, which do not certainly cover all the possibilities, are:

- Models which are *nonlinear in the predictors*, belonging to the class of **semiparametric regression models**, such as **generalized additive models** (GAMs);
- Models which are *nonlinear in the parameters*, such as **nonlinear regression models**, often based on some biological or physical model;
- **Neural networks** and their extension (such as the models used in *deep learning*), which are again *nonlinear in the parameters*, but have their own peculiarities and would deserve a specific treatment.

Here we focus on the models of the first class, and we introduce them with a simple example.

Semiparametric regression: an introductory example

An actuarial example: Automobile Bodily Injury Claims

Dataframe `AutoBi` in the R package `insuranceData`:

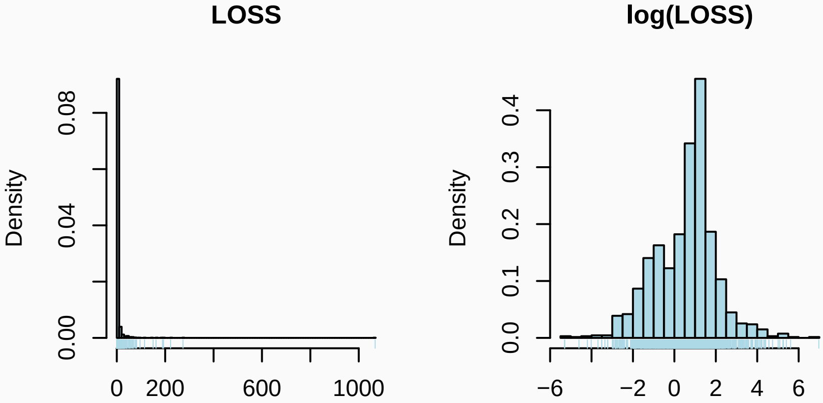
> *The data contains information about the claimant, attorney involvement and the economic loss (LOSS, in thousands), among other variables. We consider here a sample of $n = 1{,}340$ losses.*

Main variables (after some transformations):

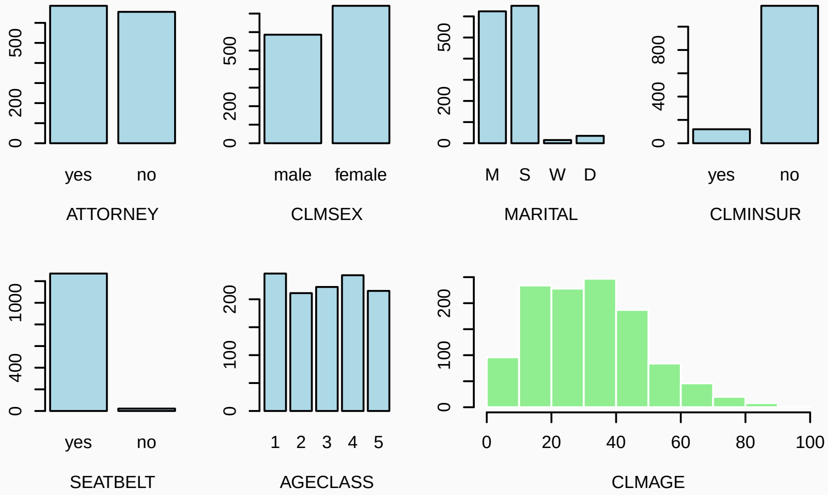
- `ATTORNEY` : The claimant is represented by an attorney (yes/no)
- `CLMSEX` : male/female
- `MARITAL` : married(M)/single (S)/widowed (W)/divorced (D)
- `CLMINSUR` : The driver of the claimant's vehicle uninsured (yes/no)
- `SEATBELT` : The claimant was wearing a seatbelt/child restraint (yes/no)
- `CLMAGE` : Claimant's age
- `AGECLASS` : Claimant's age split into five classes: (-18] / (18,26] / (26,36] / (36,47] / (47+)
- `LOSS` : Claimant's total economic loss (in thousands)

AutoBi: Distribution of `LOSS`

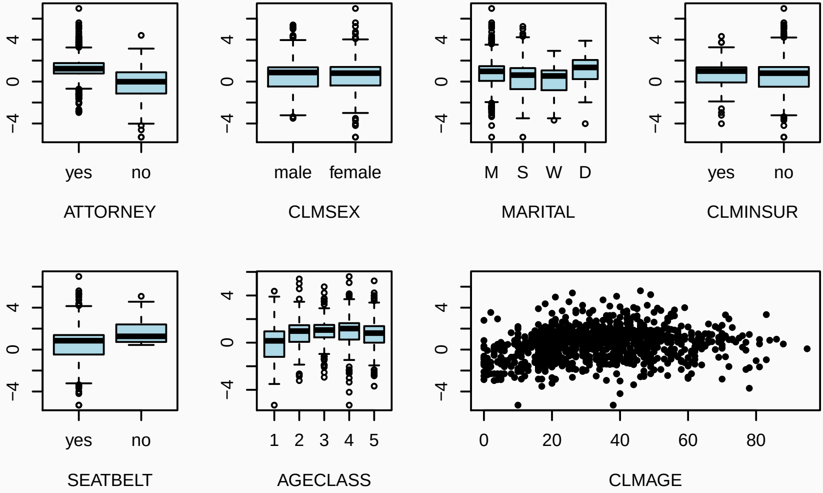
- Severity refers to the amount of a claim.
- It is of interest to build a statistical model for predicting claim amount in future policies based on a sample of claim amounts of past policies.



AutoBi: Information about predictors

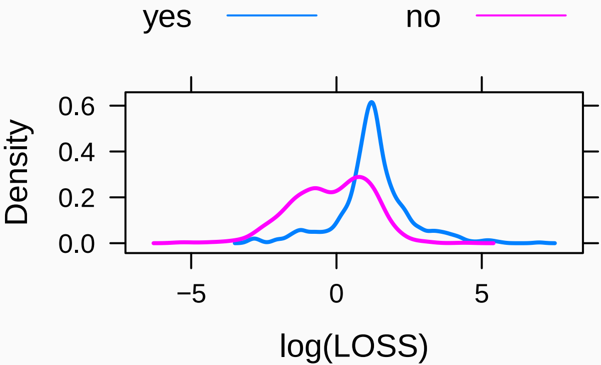


AutoBi: Relation between predictors and `log(LOSS)`



AutoBi: Relation between predictors and `log(LOSS)`

Being represented by an attorney may be important



Also other variables may matter, such as `SEATBELT`, `MARITAL` and `AGECLASS`. On the contrary `CLMSEX`, `CLMINSUR` seem not to have a *marginal* effect. The effect of `CLMAGE` is not clear.

AutoBi: linear models

Based on the AIC, the following model is selected

Table 1: Fitting linear model: log(LOSS) ~ ATTORNEY + CLMAGE + I(CLMAGE^2) + SEATBELT

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.22	0.14	-1.6	0.1
ATTORNEYno	-1.4	0.072	-19	1.6e-67
CLMAGE	0.083	0.0075	11	6.1e-27
I(CLMAGE^2)	-0.00091	9.5e-05	-9.5	1.1e-20
SEATBELTno	0.92	0.27	3.4	0.00059

The model has an R^2 value of around 0.32, and the diagnostic plots (not shown) do not highlight any serious flaw.

More on the effect of age

CLMAGE (in what follows denoted as z) was introduced in the model with three different specifications

1. As a linear term

$$y_i = \beta_0 + \beta z_i + \text{other covariates} + \varepsilon_i$$

2. As a quadratic curve

$$y_i = \beta_0 + \beta_1 z_i + \beta_2 z_i^2 + \text{other covariates} + \varepsilon_i$$

Other specifications could have been considered, such as

$$y_i = \beta_0 + \beta \log z_i + \text{other covariates} + \varepsilon_i$$

$$y_i = \beta_0 + \beta \sqrt{z_i} + \text{other covariates} + \varepsilon_i$$

or any other function $h(z)$;

3. as a discretized version, yet a different number of levels and/or different boundaries for the categories could have been chosen.

Different choices could lead to different results and we can reasonably explore a very restricted number of alternatives.

A different solution: nonlinear (semiparametric) regression

We would like to specify a model for `CLMAGE` which is at the same time

↦ simple, i.e. depending on a restricted number of parameters;

↦ flexible, i.e. capable of modelling a wide range of shapes.

One intuitive (*naive*) idea may be

$$y_i = \beta_0 + \beta_1 z_i + \beta_2 z_i^2 + \ldots + \beta_K z_i^K + \text{other} + \varepsilon_i,$$

since a polynomial function can approximate any shape if K is high enough.

Leaving aside the issue of choosing K, this is a possible solution (which may actually work) but is not very efficient due to the collinearity among the covariates z_i^k, $k = 1, \ldots, K$.

Luckily, we can keep the idea and make things work smoothly (and efficiently) using an *appropriate set of polynomial functions of* z rather than powers as above.

Some theory

Semiparametric regression: basis representation

In semiparametric regression, the specification is

$$y_i = \beta_0 + \sum_{k=1}^{K} b_k B_k(z_i) + \text{other variables} + \varepsilon_i$$

for a fixed set of known functions $B_1, \ldots, B_K$ which is called a **basis**.

In other words, we seek a function describing the relationship between y and z among those which can be written as

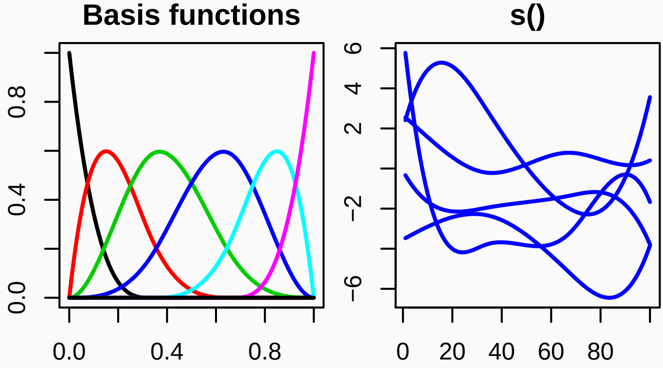
$$s(z) = \sum_{k=1}^{K} b_k B_k(z) \, .$$

For a suitable choice of the $B_k(z)$ we can obtain very different shapes using relatively few $B_k(z)$, i.e. a low value of K, in an efficient manner.

Semiparametric regression: basis representation

We do not discuss the choice of the basis set, since various possibilities exist.

Below we depict a simple example ($K = 6$) based on **B-splines**, which represent one of the most compelling choices.



Semiparametric regression: estimation

Given the specification

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j \, x_{ij} + \sum_{k=1}^{K} b_k \, B_k(z_i) + \varepsilon_i$$

and being the function $B_k(\cdot)$ known, estimation may proceed as usual for linear models.

Namely, we minimize with respect to both $\mathbf{b}$ and β the sum of squares

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} - \sum_{k=1}^{K} b_k B_k(z_i) \right)^2$$

Semiparametric regression: estimation

It may be convenient to define a new matrix

$$\tilde{X} = \begin{bmatrix} 1 & x_{11} & \dots & x_{1p} & B_1(z_1) & \dots & B_K(z_1) \\ \vdots & & \vdots & \vdots & & & \vdots \\ 1 & x_{n1} & \dots & x_{np} & B_1(z_n) & \dots & B_K(z_n) \end{bmatrix}$$

and a new coefficient vector

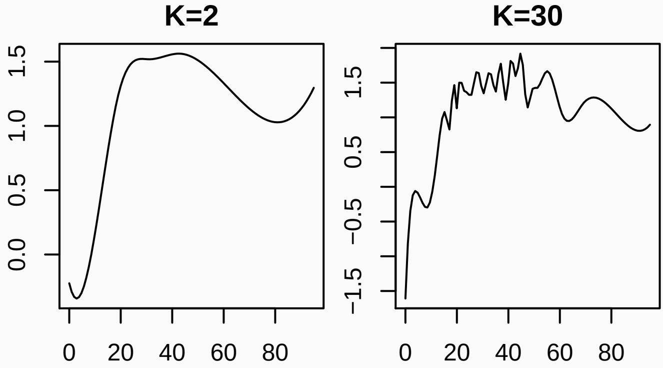
$$\tilde{\beta} = \begin{bmatrix} \beta \\ \mathbf{b} \end{bmatrix} ,$$

then the sum of squares becomes

$$(\mathbf{y} - \tilde{X}\tilde{\beta})^\top (\mathbf{y} - \tilde{X}\tilde{\beta})$$

AutoBi data: estimation

For two models including ATTORNEY, SEATBELT and different sets of B_k(CLMAGE) functions, the estimated $s(\cdot)$ functions are



A larger basis set (higher K) leads to a less smooth estimated function.

Smoothness of regression curve and choice of K

By changing the number of basis functions we estimate curves with different *degrees of smoothness*.

The more basis functions are used

- the better the final curve fits the observations,
- the higher the uncertainty of the estimates (because there are more coefficients to be estimated).

This is a **bias-variance trade off**, where more basis functions means less bias but more variance and viceversa.

Choosing the *optimal balance* is part of the estimation procedure, as discussed in the following.

Quantifying the smoothness of the curve

Although changing the number of basis functions is a strategy to tune the smoothness of the curve, it is not the optimal strategy.

We define a measure of the roughness of the $s(\cdot)$ function as

$$R(s)$$

which is null for a straight line (maximum smoothness) and increases as the curve gets less smooth.

The **roughness penalty** can be expressed as a function of the coefficients $\mathbf{b}$. In particular, there exists a matrix $\mathbf{G}$ such that

$$R(\mathbf{b}) = \mathbf{b}^\top \, \mathbf{G} \, \mathbf{b}$$

The specifics of $\mathbf{G}$ depends on the chosen basis.

Penalized sum of squares

A convenient method to tune the smoothness of the estimated curve is then fix the number of basis functions at a relatively large value, and then penalize the sum of squares according to the roughness penalty: this is the approach of **smoothing splines**.

We then define the penalized sum of squares as

$$(\mathbf{y} - \tilde{X}\,\tilde{\beta})^\top (\mathbf{y} - \tilde{X}\,\tilde{\beta}) + \lambda\,\mathbf{b}^\top\,\mathbf{G}\,\mathbf{b}$$

where $\lambda > 0$ is a *tuning parameter*.

This gives an estimated $s(\cdot)$ function which is smoother with higher values of λ:

- if $\lambda \to \infty$ the fitted curve $s(\cdot)$ is a straight line.
- if $\lambda = 0$ no penalty is considered (the curve will be as wiggly as allowed by the number of basis functions).

This procedure defines a regression curve for each value of λ.

Choice of tuning parameter

The last step is to choose an optimal degree of smoothness of the estimated curve, that is, an *optimal* λ.

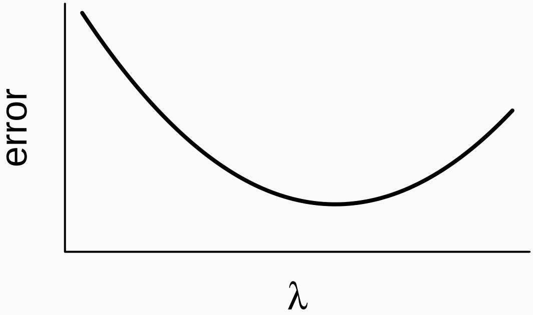
This entails choosing the optimal bias-variance balance as outlined above where a lower λ leads to

- a curve $s(\cdot)$ which better fits sample observations (**smaller bias**)
- a more uncertain estimate (**larger variance**)

The optimal balance can be found looking at the prediction error, that is the error we make when we use the model to perform **prediction on new units**.

Degree of smoothness and predictive accuracy

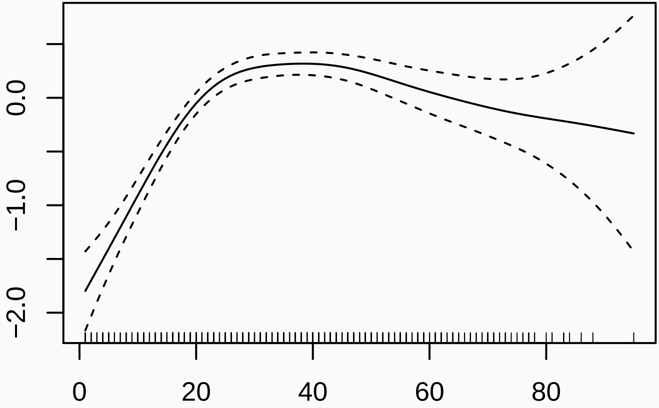
If we measure the prediction error for the different models (as λ varies), we generally get a picture such as



The optimal λ is then the one where the minimum prediction error is attained. As usual with likelihood methods, this can be obtained by **cross validation**, or by less costly alternatives such as the **Generalized Cross Validation (GCV) criterion**, which is similar to the AIC.

AutoBi: optimal λ

Using the gam function in the mgcv package, The following curve for log(LOSS) as a function of CLMAGE is obtained; note that the effect of age is now *conditional* on the other predictors



AutoBi: inference on the other coefficients

Inference for the coefficients is carried out similarly to the the linear case, thus we have a coefficient table

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.3	0.05	25	3.8e-112
ATTORNEYno	-1.4	0.072	-19	6e-69
SEATBELTno	0.9	0.27	3.4	0.00073

The **b** coefficients are usually not included in the table; an overall significance test for the $s(\cdot)$ function may be reported instead

	edf	Ref.df	F	p-value
s(CLMAGE)	4.6	5.6	28	1.9e-29

The `edf` is the *estimated degrees of freedom*: the larger the number, the more wiggly the fitted model, with values around 1 close to a linear effect.

Generalized Additive Models (GAMs)

GAMs: the basic ideas

Generalized Additive Models extend semiparametric regression in two directions:

1. *More than one nonlinear term*: for linear regression with a dependent variable Y and a set of predictor variables $X_1, \ldots, X_p$, the model for the i-th observation is

   $$y_i = \beta_0 + \sum_{j=1}^{p} s_j(x_{ij}) + \text{other variables} + \varepsilon_i \,,$$

   with one smooth term for each predictor, plus possibly other standard (linear) terms. The specification is named **additive** since the various nonlinear terms enter the specification in an additive fashion, with no interaction effects.

(There are ways to introduce interactions, but they require some non-trivial extensions.)

GAMs: the basic ideas

2. *Generalized response*: like for GLMs, binary or count responses are handled by a link function.

The nonlinear terms are introduced in the linear predictor, which now becomes

$$\eta_i = \beta_0 + \sum_{j=1}^{p} s_j(x_{ij}) + \text{other variables}$$

The estimation proceeds by representing the smooth terms using a suitable basis, and then maximizing the **penalized log-likelihood** to jointly estimate the model coefficients and the basis coefficients

$$\ell(\beta, \mathbf{b}) - \lambda\, R(\mathbf{b})\,,$$

where like before $R(\mathbf{b})$ is a measure of roughness. The estimation is often carried out using the **backfitting algorithm**, which updates one set of coefficients at a time, though other possibilities exist.

R lab: an example with binary data

As a simple example, we use the function gamSim from mgcv to generate a binary data set from a model with four additive terms

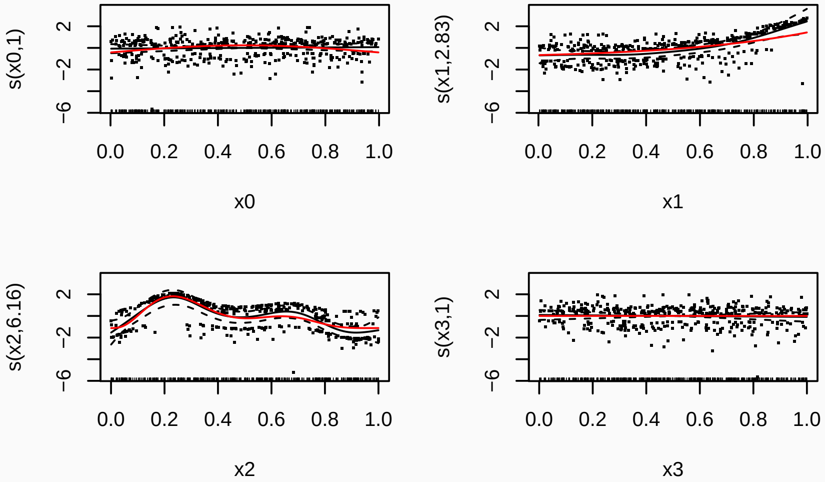
```r
dat <- gamSim(1, n = 400, dist = "binary", scale = .33)
```

```
## Gu & Wahba 4 term additive model
```

```r
lr.fit <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3),
              family = binomial,
              data = dat, method = "REML")
```

Since we know the true model, we can compare it with the results.

Plot model components with truth overlaid in red



R lab: model selection

```r
lr.fit1 <- gam(y ~ s(x0) + s(x1) + s(x2), family = binomial,
               data = dat, method = "REML")
lr.fit2 <- gam(y ~ s(x1) + s(x2), family = binomial,
               data = dat, method = "REML")
AIC(lr.fit, lr.fit1, lr.fit2)
```

```
##               df      AIC
## lr.fit  13.37849 430.8706
## lr.fit1 12.40160 429.0364
## lr.fit2 11.38595 427.1655
```

Winding up

What we have seen is just a glimpse of a very large body of methods.

Indeed, the approach based on smoothing splines with a roughness penalty is just one of several available in statistics. It has the strong advantage of being extendable in several directions, to cover also more complex settings for non-independent data.

Another approach worth mentioning is **local polynomial regression**, which includes **kernel smoothing** as a special case. It covers a broad range of applications as well, and it allows for robust versions, such the method implemented in the `lowess` function for robust scatterplot smoothing.