

BUAD 5082

Machine Learning II

Generalized Linear Models
(ISLR Chapter 5, ESL Chapter 9)

Agenda

- Brief word on Multivariate Splines
- Introduction to Generalized Additive Models
- Fitting Generalized Additive Models
 - The Backfitting Algorithm
- Examples

Multivariate Splines

- In our discussions of splines and local regression, we have focused on the one-dimensional setting. Each of the approaches have multivariate analogs.

Univariate vs Multivariate Functions

Univariate Normal

$$f(x) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Where:

- μ is mean or expectation of the distribution (and also its median and mode).
- σ is standard deviation
- σ^2 is variance

Bivariate Normal

$$f(x, y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} e^{\left(-\frac{1}{2(1-\rho^2)}\left[\frac{(x-\mu_X)^2}{\sigma_X^2} + \frac{(y-\mu_Y)^2}{\sigma_Y^2} - \frac{2\rho(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y}\right]\right)}$$

where ρ is the correlation between X and Y and where $\sigma_X > 0$ and $\sigma_Y > 0$. In this case,

$$\mu = \begin{pmatrix} \mu_X \\ \mu_Y \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{pmatrix}.$$

Multivariate Splines

- In our discussions of splines and local regression, we have focused on the one-dimensional setting. Each of the approaches have multivariate analogs.
- The theory behind these approaches involve somewhat more advanced mathematics than our scope allows (using tensor products to form the alternative bases representation of our X 's.)
 - I recommend Chapter 5 of Elements of the Statistical Learning book for those interested in pursuing this topic further)
 - The R package called **mgcv** (for Mixed GAM Computation Vehicle) contains extensions to the gam package fit multidimensional splines (e.g. thin plate splines) and other multidimensional smoothers.
 - There are several other packages in R that contain similar functionality

Modeling Methods discussed so far...

1. KNN and KNN.Reg
2. ?
3. ?
4. ?
5. ?
6. ?
7. ?
8. ?
9. ?
10. ?
11. ?
12. ?
13. ?
14. ?
15. ?
16. Local Regression

Modeling Methods discussed so far...

1. KNN and KNN.Reg
2. Simple Linear Regression
- 3.
4. Logistic Regression
- 5.
- 6.
7. Subset Selection (Best, Forward, Backward)
- 8.
9. The Lasso
- 10.
11. Partial Least Squares Regression
12. (a Basis Function method)
13. Step Function Regression (also Basis Function method)
14. (also Basis Function method)
15. Smoothing Splines
- 16.

Modeling Methods discussed so far...

1. KNN and KNN.Reg
2. Simple Linear Regression
3. Multiple Linear Regression
4. Logistic Regression
5. LDA
6. QDA
7. Subset Selection (Best, Forward, Backward)
8. Ridge Regression
9. The Lasso
10. Principal Component Regression
11. Partial Least Squares Regression
12. Polynomial Regression (a Basis Function method)
13. Step Function Regression (also Basis Function method)
14. Regression Splines (also Basis Function method)
15. Smoothing Splines
16. Local Regression

Summary So Far

Dimensionality			
		1 Predictor	1 or More Predictors
Parametric Methods			
Nonparametric Methods			

Summary So Far

Dimensionality		
	1 Predictor	1 or More Predictors
Parametric Methods	<ul style="list-style-type: none">• Simple Linear Regression• Basis Function Methods<ul style="list-style-type: none">• Polynomial Regression• Step Function Regression• Regression Splines	<ul style="list-style-type: none">• Multiple Linear Regression• Logistic Regression• LDA and QDA• Subset Selection• Ridge Regression and Lasso• PCR and PLS Regression
Nonparametric Methods	<ul style="list-style-type: none">• Smoothing Splines• Local Regression	<ul style="list-style-type: none">• KNN and KNN.Reg

Why are Smoothing Splines and Local Regression Nonparametric?

- Recall the way Smoothing splines work – we seek a function (not a set of parameters) that minimize this:

$$\sum_1^n (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

- Recall the way Local Regression works – there is no “equation” connecting the y’s with the x’s. The \hat{y} associated with an x_0 is computed individually, and depends only on the “nearest neighbors” to x_0 .

GAM's Does This...

		Dimensionality	
		1 Predictor	1 or More Predictors
Parametric Methods	<ul style="list-style-type: none">• Simple Linear Regression• Basis Function Methods<ul style="list-style-type: none">• Polynomial Regression• Step Function Regression• Regression Splines		<ul style="list-style-type: none">• Multiple Linear Regression• Logistic Regression• LDA and QDA• Subset Selection• Ridge Regression and Lasso• PCR and PLS Regression
	<ul style="list-style-type: none">• Smoothing Splines• Local Regression		<ul style="list-style-type: none">• KNN and KNN.Reg
Nonparametric Methods			

Introduction to GAMs

- Here we explore an extension to multiple linear regression called GAMs that allows us to flexibly predict Y on the basis of several predictors, X_1, \dots, X_p .
- This extension is achieved by allowing non-linear, nonparametric functions for each of the variables, while maintaining *additivity*.
- While the nonparametric form for the f_j 's makes the model more flexible, the additivity is retained and allows us to interpret the model in much the same way as before.
 - For example, $f_i(X_i)$ represents the effect of X_i on the link function if all other X_j 's are held constant
 - F values, p -values, and standard errors are interpreted as in multiple regression

Introduction to GAMs

- GAMs can be applied with both quantitative and qualitative responses (i.e.: both regression and classification).
- In the regression setting, a generalized additive model has the form

$$y \sim \alpha + f_1(X_1) + f_2(X_2) + \dots + f_p(X_p)$$

where the f_j 's can be unspecified smooth parametric or nonparametric functions.

- In the case of parametric f_j 's that we have discussed so far, we could model each function using an expansion of basis functions (as we have done with PCR, PLS regression, polynomial regression, and regression splines, for example). The resulting model could then be fit by least squares (in R, the `lm()` or `glm()` functions).

Introduction to GAMs

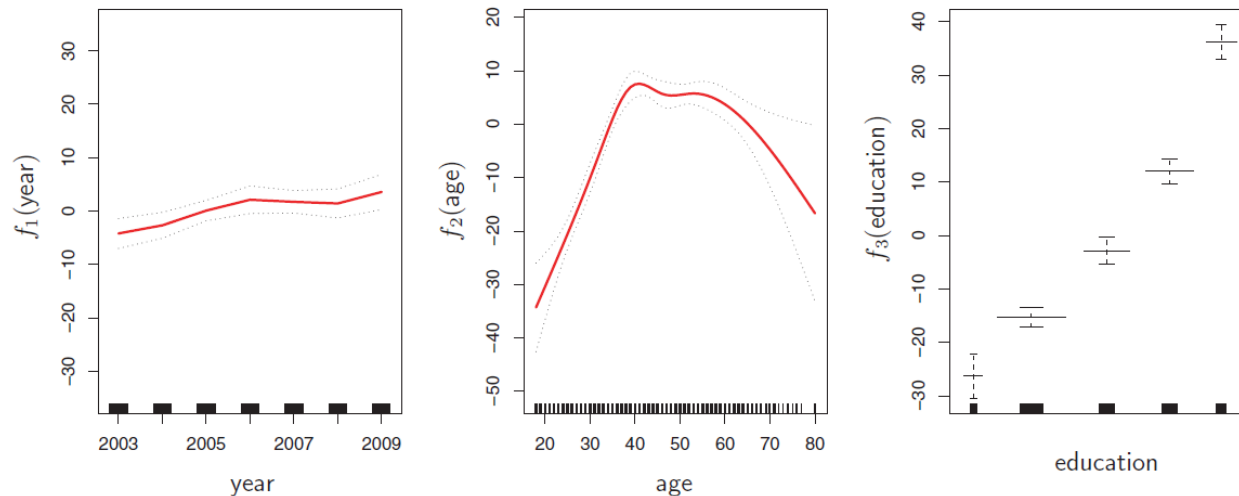
- Example:

- Consider natural splines, and the task of fitting the model:

$$\text{wage} = \beta_0 + f_1(\text{year}) + f_2(\text{age}) + f_3(\text{education}) + \epsilon$$

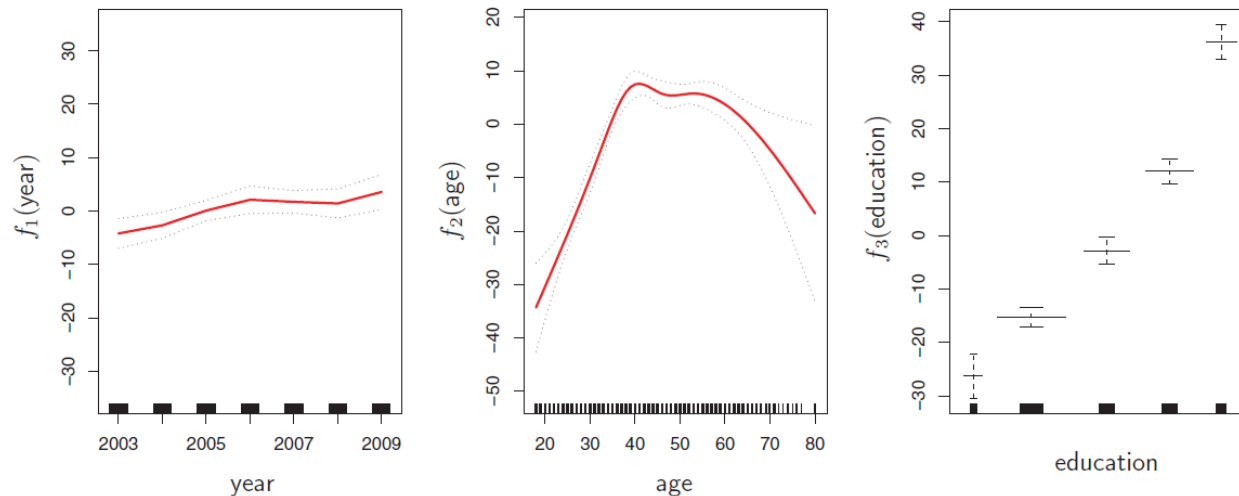
- Here year and age are quantitative variables, and education is a qualitative variable with five levels.
- If we wish to fit the first two functions using natural splines, and the third function using a dummy variable, then this is simply an expansion of basis functions representing the X's so we can solve it using `lm()`.
- See **GAMsTalk.R Section 1** for an example (this is the example in the Chapter 7 lab on GAMs)

Generalized Additive Models (GAMS)



- The figure above shows the results of fitting this additive model of splines and dummy variables using least squares.
 - Hence the entire model is just a big regression onto spline basis variables and dummy variables, all packed into one big regression matrix that we can fit with `lm()`.

Generalized Additive Models (GAMS)



- The left-hand panel indicates that holding age and education fixed, wage tends to increase slightly with year (inflation?).
- The center panel indicates that holding education and year fixed, wage tends to be highest for intermediate values of age, and lowest for the very young and very old.
- The right-hand panel indicates that holding year and age fixed, wage tends to increase with education: the more educated a person is, the higher their salary, on average.
- All of these findings are intuitive.

Introduction to GAMs

- But with GAM's, we can also fit arbitrary nonparametric functions f_j - say smoothing splines, local regression, or (in the mgcv) tensor product based smoothers.
- In these cases, we can no longer use `lm()` or `glm()` since such smoothers cannot be specified as simply a different basis representation of the X 's.
- The GAM's approach provides an algorithm, called the Backfitting Algorithm, for simultaneously estimating all p of the unknown f_j functions.

Introduction to GAMs

- In GAMs, we can also mix in linear and other parametric forms with the nonlinear terms (a necessity when some of the inputs are factors).
- The nonlinear terms are not restricted to main effects either; we can have nonlinear components in two or more variables, or separate curves in X_j for each level of the factor X_k .

Fitting Additive Models

- The building block for fitting additive models is a scatterplot smoother for fitting nonlinear effects in a flexible way.
- For concreteness, here we will use the cubic smoothing spline as our scatterplot smoother.
- The additive model has the form

$$Y = \alpha + \sum_1^p f_j(X_j) + \varepsilon$$

where the error term ε has mean zero.

Fitting Additive Models

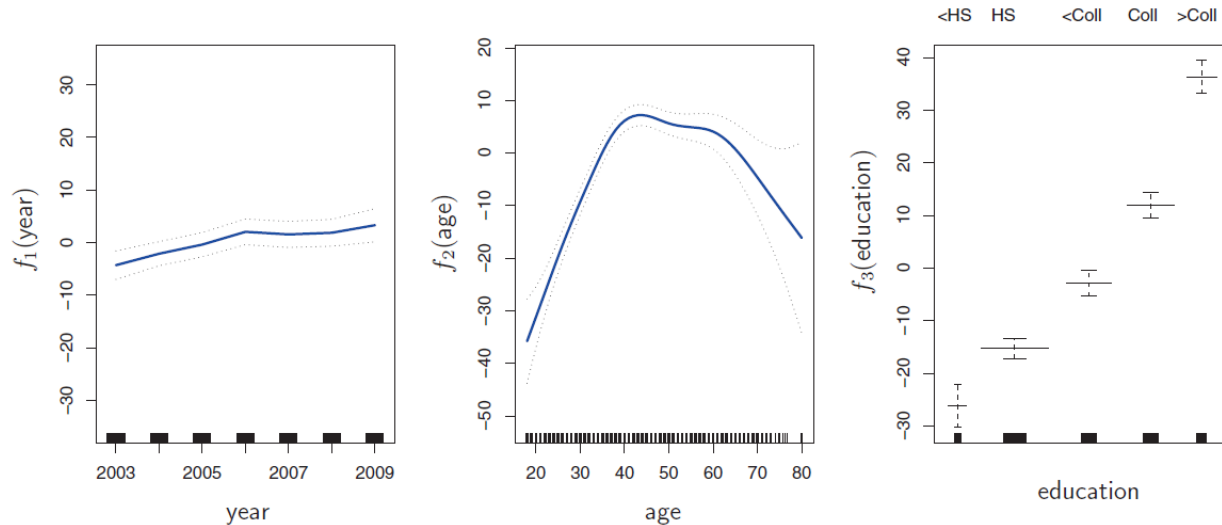
- Given observations x_i, y_i , a criterion like the penalized sum of squares (PRSS) can be specified for this problem:

$$\text{PRSS}(\alpha, f_1, f_2, \dots, f_p) = \sum_{i=1}^N (y_i - \alpha - \sum_{j=1}^p f_j(x_{ij}))^2 + \sum_{j=1}^p \lambda_j \int f_j''(t_j)^2 dt_j$$

where the $\lambda_j \geq 0$ are tuning parameters

- It can be shown that the minimizer of this expression is an additive cubic spline model; each of the functions f_j is a cubic spline in the component X_j , with knots at each of the unique values of x_{ij} , $i = 1, \dots, N$.

Fitting Additive Models



- The figure above shows three plots similar to those shown earlier, but this time f_1 and f_2 are smoothing splines with four and five degrees of freedom, respectively.
- Standard software such as the `gam()` function in R (part of the `gam` package) can be used to fit GAMs using smoothing splines, via an approach known as *backfitting*.
- See [GAMsTalk.R Section 2](#)

Fitting Additive Models

- This method fits a model involving multiple predictors by repeatedly updating the fit for each predictor in turn, holding the others fixed.
- The beauty of this approach is that each time we update a function, we simply apply the fitting method for that variable to a *partial residual*.
 - A partial residual for X_3 , for example, has the form
$$r_i = y_i - f_1(x_{i1}) - f_2(x_{i2}).$$
- If we know f_1 and f_2 , then we can fit f_3 by treating this residual as a response in a non-linear regression on X_3

Fitting Additive Models

Formally, the “Backfitting” algorithm for Additive Models is as follows:

1. Initialize: $\hat{\alpha} = \frac{1}{N} \sum_1^N y_i$, $\hat{f}_j = 0$, *for all* i, j
2. Cycle: $j = 1, 2, \dots, p, \dots, 1, 2, \dots, p, \dots$

$$\hat{f}_j = S_j \left[\left\{ y_i - \hat{\alpha} - \sum_{k \neq j} \hat{f}_k(x_{ij}) \right\}_1^N \right],$$

This is the
partial
residual for f_j

until the functions \hat{f}_j change less than a pre-specified threshold.

- See **GAMsTalk.R Section 3** for a simple example of a typical backfitting algorithm

Summary

- Pros and Cons of GAMs:
 - Pros:
 - GAMs allow us to fit a non-linear, nonparametric f_j to each X_j , so that we can automatically model non-linear relationships that standard linear regression will miss.
 - This means that we do not need to manually try out many different transformations on each variable individually.
 - These nonparametric non-linear fits can potentially make more accurate predictions for the response Y .
 - The smoothness of the functions f_j for the variables X_j can be controlled individually.
 - Because the model is additive, we can still examine the effect of each X_j on Y individually while holding all of the other variables fixed. Hence if we are interested in inference, GAMs provide a useful representation.

Summary

- Pros and Cons of GAMs:
 - Cons:
 - The model is restricted to be additive. With many variables, important interactions can be missed.
 - But we can manually add interaction terms to the GAM model We can also add low-dimensional interaction functions of the form $f_{jk}(X_j, X_k)$ into the model; such terms can be fit using two-dimensional smoothers such as local regression, or two-dimensional splines.

Generalized Additive Models (GAMS)

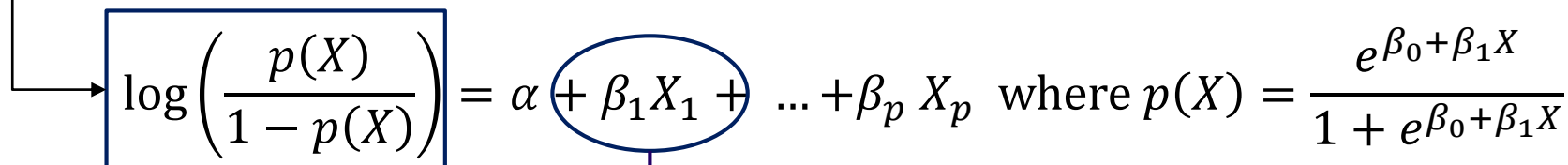
- Last words:
 - For fully general models, we have to look for even more flexible approaches such as random forests and boosting models, which we will cover next.
 - GAMs provide a useful compromise between linear and fully nonparametric models.

**SUPPLEMENTARY NOTES:
GAMS IN THE
CLASSIFICATION SETTING**

GAMs for Classification

- In the classification setting, we used the logistic regression model for binary data (i.e. 2 classes) to relate the conditional probability of “yes” given an x , $\mu(X) = \Pr(Y = 1|X)$, to the predictors via a linear regression model and the logit link function:

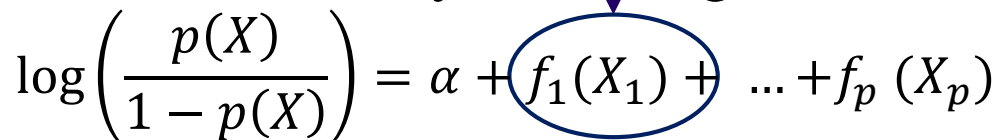
the logit link
function



The diagram shows the equation for the logit link function. A blue box encloses the term $\log\left(\frac{p(X)}{1-p(X)}\right)$. A blue oval encloses the linear term $\alpha + \beta_1 X_1 + \dots + \beta_p X_p$. A blue arrow points from the text 'the logit link function' to the boxed term. Another blue arrow points from the oval to the next equation below.

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \alpha + \beta_1 X_1 + \dots + \beta_p X_p \quad \text{where } p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

- The GAMs logistic regression model replaces each linear term by a more general functional form



The diagram shows the equation for the GAM logistic regression model. A blue oval encloses the term $f_1(X_1)$. A blue arrow points from the oval in the equation above to this oval.

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \alpha + f_1(X_1) + \dots + f_p(X_p)$$

where again each f_j is an unspecified smooth function.

GAMs for Classification

- In general, the expected value of a response Y is related to an additive function of the predictors via a link function g :

$$g[u(X)] = \alpha + f_1(X_1) + f_2(X_2) + \dots + f_p(X_p)$$

- Examples:
 - $g(\mu) = \mu$ is the identity link, used for linear and additive models for Gaussian (normal) response data (i.e.: familiar multiple regression).
 - $g(\mu) = \text{logit}(\mu)$ as above for modeling log odds
 - $g(\mu) = \log(\mu)$ for log-linear models for Poisson data.
 - $g(\mu) = \text{probit}(\mu)$, for modeling binomial probabilities
- Recall the family= parameter in the glm() function (from the glm Help page):
 - `binomial(link = "logit")`
 - `gaussian(link = "identity", the default)`
 - `Gamma(link = "inverse")`
 - `inverse.gaussian(link = "1/mu^2")`
 - `poisson(link = "log")`
 - `quasi(link = "identity", variance = "constant")`
 - `quasibinomial(link = "logit")`
 - `quasipoisson(link = "log")`

GAMs for Classification

- Examples:
 - $g(u) = X\beta + \alpha_k + f(Z)$ - a semiparametric model, where
 - X is a matrix of predictors to be modeled linearly
 - α_k is the effect for the k th level of a qualitative input variable V (i.e.: dummy variables for α),
 - $f(Z)$ is the effect of predictor Z being modeled nonparametrically by say, cubic smoothing spline or some other “scatterplot smoother.”
 - $g(u) = f(X) + g_k(Z)$ - again k indexes the levels of a qualitative input V , and thus creates an interaction term $g(V, Z) = g_k(Z)$ for the effect of V and Z .
 - See the following link for a nice example of how these ideas can be applied to modeling time series using GAMs
 - <http://www.fromthebottomoftheheap.net/2014/05/09/modelling-seasonal-data-with-gam/>
 - $g(u) = f(X) + g(Z, W)$ where g is a nonparametric function in two features – say a two-dimensional spline.