Lecture 18: GAMs

Reading: Sections 7.7

STATS 202: Data mining and analysis

Jonathan Taylor
November 7, 2017
Slide credits: Sergio Bacallado

Generalized Additive Models (GAMs)

Extension of non-linear models to multiple predictors:

$$\texttt{wage} = \beta_0 + \beta_1 \times \texttt{year} + \beta_2 \times \texttt{age} + \beta_3 \times \texttt{education} + \epsilon$$

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

Generalized Additive Models (GAMs)

Extension of non-linear models to multiple predictors:

$$\texttt{wage} = \beta_0 + \beta_1 \times \texttt{year} + \beta_2 \times \texttt{age} + \beta_3 \times \texttt{education} + \epsilon$$

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

The functions $f_1, \ldots, f_p$ can be polynomials, natural splines, smoothing splines, local regressions...

Fitting a GAM

- If the functions f_1 have a basis representation, we can simply use least squares:
  - Natural cubic splines
  - Polynomials
  - Step functions

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

Fitting a GAM

- Otherwise, we can use **backfitting**:

Fitting a GAM

- Otherwise, we can use **backfitting**:

  1. Keep $f_2, \ldots, f_p$ fixed, and fit f_1 using the partial residuals:

     $$y_i - \beta_0 - f_2(x_{i2}) - \cdots - f_p(x_{ip}),$$

     as the response.

Fitting a GAM

- Otherwise, we can use **backfitting**:

  1. Keep $f_2, \ldots, f_p$ fixed, and fit f_1 using the partial residuals:

     $$y_i - \beta_0 - f_2(x_{i2}) - \cdots - f_p(x_{ip}),$$

     as the response.
  2. Keep $f_1, f_3, \ldots, f_p$ fixed, and fit f_2 using the partial residuals:

     $$y_i - \beta_0 - f_1(x_{i1}) - f_3(x_{i3}) - \cdots - f_p(x_{ip}),$$

     as the response.

Fitting a GAM

- Otherwise, we can use **backfitting**:

  1. Keep $f_2, \dots, f_p$ fixed, and fit f_1 using the partial residuals:

     $$y_i - \beta_0 - f_2(x_{i2}) - \cdots - f_p(x_{ip}),$$

     as the response.

  2. Keep $f_1, f_3, \dots, f_p$ fixed, and fit f_2 using the partial residuals:

     $$y_i - \beta_0 - f_1(x_{i1}) - f_3(x_{i3}) - \cdots - f_p(x_{ip}),$$

     as the response.

  3. ...

Fitting a GAM

- Otherwise, we can use **backfitting**:

1. Keep $f_2, \ldots, f_p$ fixed, and fit f_1 using the partial residuals:

$$y_i - \beta_0 - f_2(x_{i2}) - \cdots - f_p(x_{ip}),$$

   as the response.

2. Keep $f_1, f_3, \ldots, f_p$ fixed, and fit f_2 using the partial residuals:

$$y_i - \beta_0 - f_1(x_{i1}) - f_3(x_{i3}) - \cdots - f_p(x_{ip}),$$

   as the response.

3. ...

4. Iterate

Fitting a GAM

- Otherwise, we can use **backfitting**:

  1. Keep $f_2, \ldots, f_p$ fixed, and fit f_1 using the partial residuals:

     $$y_i - \beta_0 - f_2(x_{i2}) - \cdots - f_p(x_{ip}),$$

     as the response.

  2. Keep $f_1, f_3, \ldots, f_p$ fixed, and fit f_2 using the partial residuals:

     $$y_i - \beta_0 - f_1(x_{i1}) - f_3(x_{i3}) - \cdots - f_p(x_{ip}),$$

     as the response.

  3. ...
  4. Iterate

- This works for smoothing splines and local regression. **For smoothing splines this is a descent method, descending on convex loss . . .**

Backfitting: coordinate descent

- Also works for linear regression...

Backfitting: coordinate descent

- Also works for linear regression...
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).

Backfitting: coordinate descent

- Also works for linear regression...
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

$$\hat{\alpha}(T) = \text{argmin}_\alpha \sum_{i=1}^n (Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j\neq k(T)} X_{ij}\hat{\beta}_j^{(T-1)} - \alpha X_{ik(T)})^2$$

$$= \frac{\sum_{i=1}^n X_{ik(T)}(Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j\neq k(T)} X_{ij}\hat{\beta}_j^{(T-1)})}{\sum_{i=1}^n X_{ik(T)}^2}$$

Backfitting: coordinate descent

- Also works for linear regression...
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

$$\hat{\alpha}(T) = \text{argmin}_\alpha \sum_{i=1}^n (Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j\neq k(T)} X_{ij}\hat{\beta}_j^{(T-1)} - \alpha X_{ik(T)})^2$$

$$= \frac{\sum_{i=1}^n X_{ik(T)}(Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j\neq k(T)} X_{ij}\hat{\beta}_j^{(T-1)})}{\sum_{i=1}^n X_{ik(T)}^2}$$

  3. Set $\hat{\beta}^{(T)} = \hat{\beta}^{(T-1)}$ except $k(T)$ entry which we set to $\hat{\alpha}(T)$.

Backfitting: coordinate descent

- Also works for linear regression...
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

$$\hat{\alpha}(T) = \text{argmin}_{\alpha} \sum_{i=1}^{n} (Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j \neq k(T)} X_{ij} \hat{\beta}_j^{(T-1)} - \alpha X_{ik(T)})^2$$

$$= \frac{\sum_{i=1}^{n} X_{ik(T)} (Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j \neq k(T)} X_{ij} \hat{\beta}_j^{(T-1)})}{\sum_{i=1}^{n} X_{ik(T)}^2}$$

  3. Set $\hat{\beta}^{(T)} = \hat{\beta}^{(T-1)}$ except $k(T)$ entry which we set to $\hat{\alpha}(T)$.
  4. Iterate

Backfitting: coordinate descent and LASSO

- Also works for LASSO...

Backfitting: coordinate descent and LASSO

- Also works for LASSO...
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).

Backfitting: coordinate descent and LASSO

- Also works for LASSO...
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

$$\begin{aligned}\hat{\alpha}_\lambda(T) = \text{argmin}_\alpha \sum_{i=1}^n (r_{ik(T)}^{(T-1)} - \alpha X_{ik(T)})^2 \\ + \lambda \sum_{j:j\neq k(T)} |\hat{\beta}_j^{(T-1)}| + \lambda|\alpha|\end{aligned}$$

with $r_j^{(T-1)}$ the j-th partial residual at iteration T

$$r_j^{(T-1)} = Y - \hat{\beta}_0^{(T-1)} - \sum_{l:l\neq j} X_l \hat{\beta}_l^{(T-1)}.$$

Solution is a simple soft-thresholded version of previous $\hat{\alpha}(T)$ – **Very fast! Used in** `glmnet`

Backfitting: coordinate descent and LASSO

- Also works for LASSO...
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

$$\hat{\alpha}_\lambda(T) = \text{argmin}_\alpha \sum_{i=1}^n (r_{ik(T)}^{(T-1)} - \alpha X_{ik(T)})^2 + \lambda \sum_{j:j\neq k(T)} |\hat{\beta}_j^{(T-1)}| + \lambda|\alpha|$$

with $r_j^{(T-1)}$ the j-th partial residual at iteration T

$$r_j^{(T-1)} = Y - \hat{\beta}_0^{(T-1)} - \sum_{l:l\neq j} X_l \hat{\beta}_l^{(T-1)}.$$

Solution is a simple soft-thresholded version of previous $\hat{\alpha}(T)$ – **Very fast! Used in** `glmnet`

Backfitting: coordinate descent and LASSO

- Also works for LASSO...
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

     $$\begin{aligned}\hat{\alpha}_\lambda(T) = \text{argmin}_\alpha & \sum_{i=1}^n (r_{ik(T)}^{(T-1)} - \alpha X_{ik(T)})^2 \\ & + \lambda \sum_{j:j\neq k(T)} |\hat{\beta}_j^{(T-1)}| + \lambda|\alpha|\end{aligned}$$

     with $r_j^{(T-1)}$ the j-th partial residual at iteration T

     $$r_j^{(T-1)} = Y - \hat{\beta}_0^{(T-1)} - \sum_{l:l\neq j} X_l \hat{\beta}_l^{(T-1)}.$$

     Solution is a simple soft-thresholded version of previous $\hat{\alpha}(T)$ – **Very fast! Used in** `glmnet`
  3. Set $\hat{\beta}^{(T)} = \hat{\beta}^{(T-1)}$ except $k(T)$ entry which we set to $\hat{\alpha}_\lambda(T)$.

Backfitting: coordinate descent and LASSO

- Also works for LASSO...
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

     $$\hat{\alpha}_\lambda(T) = \text{argmin}_\alpha \sum_{i=1}^n (r_{ik(T)}^{(T-1)} - \alpha X_{ik(T)})^2 + \lambda \sum_{j:j \neq k(T)} |\hat{\beta}_j^{(T-1)}| + \lambda|\alpha|$$

     with $r_j^{(T-1)}$ the j-th partial residual at iteration T

     $$r_j^{(T-1)} = Y - \hat{\beta}_0^{(T-1)} - \sum_{l:l \neq j} X_l \hat{\beta}_l^{(T-1)}.$$

     Solution is a simple soft-thresholded version of previous $\hat{\alpha}(T)$ – **Very fast! Used in** `glmnet`
  3. Set $\hat{\beta}^{(T)} = \hat{\beta}^{(T-1)}$ except $k(T)$ entry which we set to $\hat{\alpha}_\lambda(T)$.
  4. Iterate...

Backfitting: GAM

- Let's look at basis functions

Backfitting: GAM

- Let's look at basis functions
    1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).

Backfitting: GAM

- Let's look at basis functions
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

$$\hat{\alpha}_\lambda(T) = \text{argmin}_{\alpha \in \mathbb{R}^{n_{k(T)}}} \sum_{i=1}^{n} (Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j \neq k(T)} \sum_{l=1}^{n_j} f_{lj}(X_{ij}) \hat{\beta}_{lj}^{(T-1)} - \sum_{l=1}^{n_{k(T)}} \alpha_l f_{lk(T)}(X_{ik(T)}))^2$$

Backfitting: GAM

- Let's look at basis functions
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

$$\hat{\alpha}_\lambda(T) = \mathrm{argmin}_{\alpha \in \mathbb{R}^{n_{k(T)}}} \sum_{i=1}^{n} \Big(Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j \neq k(T)} \sum_{l=1}^{n_j} f_{lj}(X_{ij}) \hat{\beta}_{lj}^{(T-1)} - \sum_{l=1}^{n_{k(T)}} \alpha_l f_{lk(T)}(X_{ik(T)})\Big)^2$$

Backfitting: GAM

- Let's look at basis functions
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ and find

$$\hat{\alpha}_\lambda(T) = \text{argmin}_{\alpha \in \mathbb{R}^{n_{k(T)}}} \sum_{i=1}^{n} (Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j \neq k(T)} \sum_{l=1}^{n_j} f_{lj}(X_{ij}) \hat{\beta}_{lj}^{(T-1)} - \sum_{l=1}^{n_{k(T)}} \alpha_l f_{lk(T)}(X_{ik(T)}))^2$$

  3. Set $\hat{\beta}^{(T)} = \hat{\beta}^{(T-1)}$ except $k(T)$ entries which we set to $\hat{\alpha}_\lambda(T)$. **Blockwise coordinate descent!**

Backfitting: GAM

- Let's look at basis functions
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \le k(T) \le p$ and find

$$\hat{\alpha}_\lambda(T) = \text{argmin}_{\alpha \in \mathbb{R}^{n_{k(T)}}} \sum_{i=1}^{n} (Y_i - \hat{\beta}_0^{(T-1)} - \sum_{j:j \neq k(T)} \sum_{l=1}^{n_j} f_{lj}(X_{ij}) \hat{\beta}_{lj}^{(T-1)} - \sum_{l=1}^{n_{k(T)}} \alpha_l f_{lk(T)}(X_{ik(T)}))^2$$

  3. Set $\hat{\beta}^{(T)} = \hat{\beta}^{(T-1)}$ except $k(T)$ entries which we set to $\hat{\alpha}_\lambda(T)$. **Blockwise coordinate descent!**
  4. Iterate...

Properties of GAMs

- GAMs are a step from linear regression toward a fully nonparametric method.

Properties of GAMs

- GAMs are a step from linear regression toward a fully nonparametric method.
- The only constraint is additivity. This can be partially addressed by adding key interaction variables $X_i X_j$ (or tensor product of basis functions – e.g. polynomials of two variables).

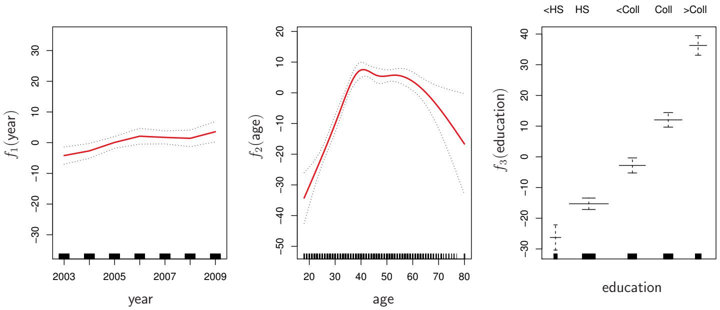
Properties of GAMs

- GAMs are a step from linear regression toward a fully nonparametric method.
- The only constraint is additivity. This can be partially addressed by adding key interaction variables $X_i X_j$ (or tensor product of basis functions – e.g. polynomials of two variables).
- We can report degrees of freedom for many non-linear functions.

Properties of GAMs

- GAMs are a step from linear regression toward a fully nonparametric method.
- The only constraint is additivity. This can be partially addressed by adding key interaction variables $X_i X_j$ (or tensor product of basis functions – e.g. polynomials of two variables).
- We can report degrees of freedom for many non-linear functions.
- As in linear regression, we can examine the significance of each of the variables.

Example: Regression for `Wage`

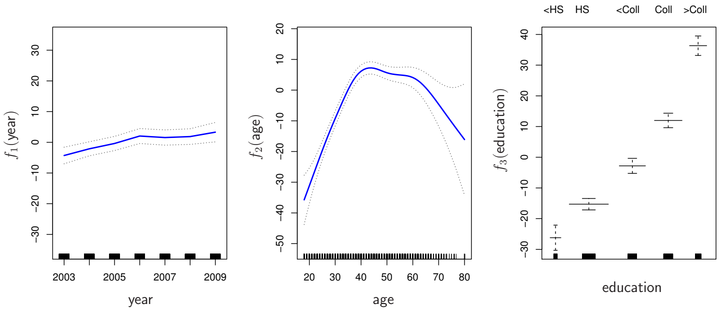


`year`: natural spline with df=4.

`age`: natural spline with df=5.

`education`: factor.

Example: Regression for `Wage`



`year`: smoothing spline with df=4.
`age`: smoothing spline with df=5.
`education`: step function.

GAMs for classification

We can model the log-odds in a classification problem using a GAM:

$$\log \frac{P(Y=1 \mid X)}{P(Y=0 \mid X)} = \beta_0 + f_1(X_1) + \cdots + f_p(X_p).$$

Again fit by backfitting ...

Backfitting: GAM with logistic loss

- Also works for logistic...

Backfitting: GAM with logistic loss

- Also works for logistic...
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).

Backfitting: GAM with logistic loss

- Also works for logistic...
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \le k(T) \le p$ with ℓ logistic loss, find

$$\hat{\alpha}_\lambda(T) = \text{argmin}_{\alpha \in \mathbb{R}^{n_{k(T)}}} \sum_{i=1}^{n} \ell\left(Y_i, \hat{\beta}_0^{(T-1)} + \sum_{j : j \neq k(T)} \sum_{l=1}^{n_j} f_{lj}(X_{ij}) \hat{\beta}_{lj}^{(T-1)} + \sum_{l=1}^{n_{k(T)}} \alpha_l f_{lk(T)}(X_{ik(T)}) \right)$$

Backfitting: GAM with logistic loss

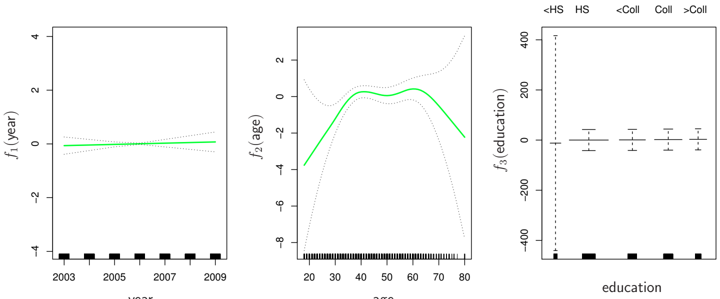
- Also works for logistic...
  1. Initialize $\hat{\beta}^{(0)} = 0$ and, (say).
  2. Given $\hat{\beta}^{(T-1)}$, choose a coordinate $0 \leq k(T) \leq p$ with ℓ logistic loss, find

$$\hat{\alpha}_\lambda(T) = \text{argmin}_{\alpha \in \mathbb{R}^{n_{k(T)}}} \sum_{i=1}^{n} \ell\Bigg(Y_i, \hat{\beta}_0^{(T-1)} + \sum_{j:j\neq k(T)} \sum_{l=1}^{n_j} f_{lj}(X_{ij})\hat{\beta}_{lj}^{(T-1)} + \sum_{l=1}^{n_{k(T)}} \alpha_l f_{lk(T)}(X_{ik(T)})\Bigg)$$

  3. Works for losses that have a *linear predictor*. For GAMs, the linear predictor is

$$\beta_0 + f_1(X_1) + \cdots + f_p(X_p)$$

Example: Classification for `Wage>250`

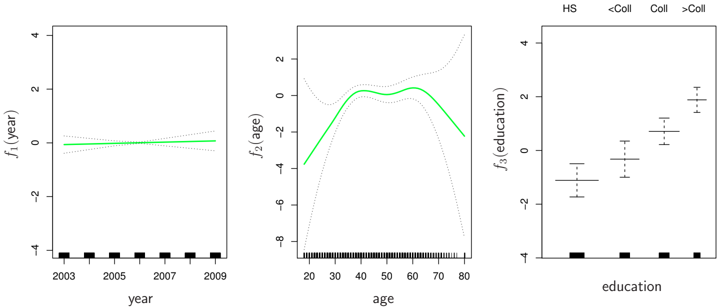


`year`: linear.

`age`: smoothing spline with df=5.

`education`: step function.

Example: Classification for `Wage>250`



`year`: linear.

`age`: smoothing spline with df=5.

`education`: step function.

Exclude samples with `education` < `HS`.