Section. Generative Additive Models

1. Generalized Additive Models

2. Backfitting

➢ Generalized Additive Models

Generalized additive models are a combine of two worlds: **generalized linear models** and **additive models**, and is altogether a very flexible, powerful platform for modeling.

Let $\vec{X} \in \mathbb{R}^d$ be a vector of predictors. Let Y be a label vector.

Let $\mu(\vec{X}) = E(Y|\vec{X})$

1. Linear Models: We model $\mu(\vec{X})$ as a linear function of $\vec{X}$:

$$E(Y|\vec{X}) \equiv \mu(\vec{X}) = \vec{\beta}^T\vec{X} = \beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d$$

2. Generalized Linear Models: model some function of $\mu(\vec{X})$ as a linear function of $\vec{X}$:

$$g\left(\mu(\vec{X})\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d$$

Here $g(u)$ is the **link function.**

Generalized Linear Models:

$$\mu(\vec{X}) = g^{-1}(\vec{\beta}^T \vec{X})$$

Example of link function:

1. $g(\mu) = \mu$ Gaussian model(standard regression)

2. $g(\mu) = \log \frac{\mu}{1-\mu}$ Binomial model(logistic regression)

3. $g(\mu) = \log(\mu)$ Poisson model

4. $\mu = g^{-1}(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\xi} \exp\left(-\frac{u^2}{2}\right) du$ Probit model

non-canonical link function:

3. An **additive model** is given by

$$E\left(Y\middle|\vec{X}\right) \equiv \mu\left(\vec{X}\right) = \beta_0 + f_1(x_1) + \cdots + f_d(x_d)$$

4. Generalized Linear Models for a fixed link function $g(\mu)$ and basis $f_1, \ldots, f_d$ has the form

$$g\left(\mu\left(\vec{X}\right)\right) = \beta_0 + f_1(x_1) + \cdots + f_d(x_d)$$

For example, if $f_j\left(x_j\right) = \beta_j x_j$, then it is the generalized linear regression. More generally, we can choose $f_1(x_1), \ldots, f_d(x_d)$ be the *smoothing spline* function of x_j.

The generalized linear model allows us to account for different types of outcome data y, and the additive model element allows us to consider a transformation of the mean $\mu\left(\vec{X}\right) = E(Y|\vec{X})$ as a nonlinear (but additive) function of $\vec{x}$.

Example:

Assume $y|x \sim N(\mu, \sigma^2)$

- Standard Linear Model

$$\mu = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2$$

- Polynomial Regression (Additive models)

$$\mu = b_0 + b_1 \cdot x_1 + +b_2 \cdot x_2 + b_3 \cdot x_1^2 + b_4 \cdot x_2^2$$

- GLM formulation

$$g(\mu) = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2$$

- GAM formulation

$$g(\mu) = f(X)$$

➢ Additive cubic smoothing spline model

Suppose the training data set is $\mathcal{D} = \left\{\left(\vec{x}^{(i)}, y^{(i)}\right)\right\}_{i-1}^{n}$

Fit a Generalized Additive Model:

$$g\left(\mu(\vec{X})\right) = \beta_0 + f_1(x_1) + \cdots + f_d(x_d)$$

How to specify functions $\boldsymbol{f_1, \ldots, f_d}$?

We focus on the case of smoothing spline functions.

Note that f_j sees only the j^{th} coordinate x_j. Then now f_j sees only the j^{th} coordinate $x_j^{(i)}$ of the training point $\vec{x}^{(i)}$.

Consider **(Sobolev) space**

$$\mathcal{S}[a, b] = \left\{f\colon [a, b] \to \mathbb{R} \mid f'' \text{continuous and } \int_a^b \left(f''(t)\right)^2 dt \text{ is finite}\right\}$$

Among all functions $f_j(x) \in \mathcal{S}[a,b]$ that **minimizes** the penalized residual sum of squares.

$$(\hat{\beta}_0, \widehat{f}_1, \dots, \widehat{f}_d) = \underset{\substack{f_j \in \mathcal{S}[a,b] \\ \beta_0}}{\operatorname{argmin}} \sum_{i=1}^{N} \left(y^{(i)} - \beta_0 - \sum_{j=1}^{d} f_j\left(x_j^{(i)}\right) \right)^2 + \sum_{j=1}^{d} \lambda_j \int_a^b \left(f_j''(t_j)\right)^2 dt_j$$

The knots of f_j can be chosen as any points, but they end up just being at the $x_j^{(i)}$, i.e. the j^{th} coordinates of the data points $\vec{x}^{(i)}$ (as usual).

How to go about solving this?

Solution above is an called an additive cubic smoothing spline model.

Additive cubic smoothing spline model

(1) So our desired regression function $\mu(\vec{X}) = E(Y|\vec{X})$ is estimated as

$$\left(\hat{\mu}(\vec{X})\right) = \widehat{\beta_0} + \hat{f}_1(x_1) + \cdots + \hat{f}_d(x_d)$$

such that for fixed j, each function $\hat{f}_j(x_j)$ is a cubic spline with knots at the unique points $\vec{x}_j^{\,(i)}$

Notice that above we can freeze j and look at different values $x_j^{(i)}$ of x_j based on the training set, i.e., $\left\{x_j^{(i)}\right\}_{i=1}^{N}$

(2) Solution is not unique. It is the same solution if we replace $\hat{f}_j$ by $\hat{f}_j + c_j$ and replace $\widehat{\beta_0}$ by $\widehat{\beta_0} - \sum c_j$.

Restrict to f_j such that $\sum_{i=1}^{N} \widehat{f_j}\left(x_j^{(i)}\right) = 0$

(3) For any given (fixed) j, the optimization problem looks like

$$\underset{f_j}{\operatorname{argmin}} \sum_{i=1}^{N} \left(z_i - f_j\left(x_j^{(i)}\right)\right)^2 + \lambda_j \int_a^b \left(f_j''(t_j)\right)^2 dt_j$$

where $$z^{(i)} = y^{(i)} - \beta_0 - \sum_{k \neq j} \widehat{f_k}\left(x_j^{(i)}\right)$$

iterate repeatedly through the dimensions $j = 1, \ldots, p$

So a standard smoothing spline problem in each separate dimension.

➢ Backfitting (one variable at a time) for Generalized Additive Models

The above separation of variables x_j suggests a Gauss-Seidel method/ implementation i.e., want

$$\hat{\vec{\theta}} = \underset{\vec{\theta} \in \mathbb{R}^m}{\operatorname{argmin}}\, h(\vec{\theta})$$

Algorithm:

1. Initialize $\hat{\vec{\theta}}$

2. **While** not converged, do

   For $i = 1, \dots, m$

   $$\widehat{\theta_i} = \underset{\theta_i}{\operatorname{argmin}}\, h(\hat{\theta}_1, \dots, \hat{\theta}_{i-1}, \theta_i, \hat{\theta}_{i+1}, \dots, \hat{\theta}_m)$$

i.e., freeze all values $\hat{\theta}_j$ for $j \neq i$ and just optimize single θ_i at a time.

- Popular in partial differential equation methods.
- Usually linear convergence, i.e. error e_n after n steps satisfies

$$e_n = c\ e_{n-1}, \text{for } c < 1$$

[As opposed say to quadratic convergence]

Backfitting for Generalized Additive Models

Model: $y = \beta_0 + f_1(x_1) + \cdots + f_d(x_d)$

1. Initialize $\widehat{\beta_0} = \frac{1}{N}\sum_{i=1}^{N} y^{(i)}$

$\hat{f}_j = 0$, for $j = 1, \ldots, d$

2. While some $\hat{f}_j$ still changing for $j = 1, \ldots, d$

Model just j^{th} function $f_j(x_j)$ and freeze the other $f_k(x_k)$, thus write

$$y = \beta_0 + f_1(x_1) + \cdots + f_d(x_d) = \beta_0 + f_j(x_j) + \sum_{k \neq j} f_k(x_k)$$

So model

$$y - \beta_0 - \sum_{k \neq j} f_k(x_k) = f_j(x_j)$$

So we will fit f_j by replacing

$$y^{(i)} \longmapsto y^{(i)} - \beta_0 - \sum_{k \neq j} \widehat{f_k}(x_k) \quad =: z^{(j)} = \textit{adjusted response}$$

and do least squares with these adjusted y values $z^{(j)}$

Form vector $\overrightarrow{z_j} = \left(z^{(1)}, z^{(2)}, \ldots, z^{N}\right)$ *of* adjusted y values.

Recall with least squares get smoothing matrix S_j to get estimated function. Thus write

$$\widehat{\overrightarrow{f_j}} = S_j \, \overrightarrow{z_j}$$

Where

$$\vec{\widehat{f_j}} = \begin{bmatrix} \vec{\widehat{f_j}}\left(x_j^{(1)}\right) \\ \vec{\widehat{f_j}}\left(x_j^{(2)}\right) \\ \vdots \\ \vec{\widehat{f_j}}\left(x_j^{(N)}\right) \end{bmatrix}$$

Also adjust:

$$\vec{\widehat{f_j}} \longmapsto \vec{\widehat{f_j}} - \frac{1}{\mathrm{N}} \sum_{i=1}^{N} \vec{\widehat{f_j}}\left(x_j^{(i)}\right)$$

Comments:

Note that 'smoothing matrix S_j takes adjusted $y^{(i)}$ values and gives estimates $\hat{y}^{(i)}$ for them under current model, here just for the j^{th} coordinate part.

This means we are just fitting the function f_j and freezing the other f_k

But note this easy computation only gives $\hat{f}\left(x_j^{(i)}\right)$ at data points. More complicated if we want a formula for all x.

Note again that brackets $\vec{z}_j$ contains vector $\vec{y}$ with $y^{(i)}$ replaced by

$$y^{(i)} - \beta_0 - \sum_{k \neq j} \widehat{f_k}(x_k)$$

Comments:

This algorithm can be computationally intensive, so may slow down say leave one out cross-validation, which must be repeated many times.

This method uses the assumed above structure as a sum of individual coordinate functions $f_j(x_j)$ to get at curse of dimensionality.

But what if there are interactions among variables?

Then include low order products like $f_i(x_i) \cdot f_j(x_j)$ in the model.

Textbook:

Hastie[HTF]: Sec 9.1