Model Assessment and Selection

Steve Avsec

Illinois Institute of Technology

February 26, 2024

Overview

Basics

General Linear Models:

$$g(E[Y|\mathbf{X}]) = c_0 + \sum_{j=1^N} c_j X_j$$

where g is a link function (identity, logit, log, etc.)

Basics

General Linear Models:

$$g(E[Y|\mathbf{X}]) = c_0 + \sum_{j=1^N} c_j X_j$$

where g is a link function (identity, logit, log, etc.)

General Additive Models:

$$g(E[Y|\mathbf{X}]) = c_0 + \sum_{j=1^N} f_j(X_j)$$

What are these functions?

Many possibilities:

- Linear terms

$$g(E[Y|\mathbf{X}]) = c_0 + \sum_{j=1^K} c_j X_j + \sum_{j=K+1}^{N} f_j(X_j)$$

What are these functions?

Many possibilities:

- Linear terms

$$g(E[Y|\mathbf{X}]) = c_0 + \sum_{j=1^K} c_j X_j + \sum_{j=K+1}^{N} f_j(X_j)$$

- Splines

What are these functions?

Many possibilities:

- Linear terms

$$g(E[Y|\mathbf{X}]) = c_0 + \sum_{j=1^K} c_j X_j + \sum_{j=K+1}^{N} f_j(X_j)$$

- Splines
- Basis functions (e.g. polynomials, trig polynomials, etc., just least squares).

What are these functions?

Many possibilities:

- Linear terms

$$g(E[Y|\mathbf{X}]) = c_0 + \sum_{j=1^K} c_j X_j + \sum_{j=K+1}^{N} f_j(X_j)$$

- Splines
- Basis functions (e.g. polynomials, trig polynomials, etc., just least squares).
- Nonparametric functions (Kernel estimation).

Backfitting

1. Let $\alpha = \frac{1}{N}\sum_{j=1}^{N} y_j$, $f_j := 0$.

Backfitting

1. Let $\alpha = \frac{1}{N}\sum_{j=1}^{N} y_j$, $f_j := 0$.
2. Iterate $j = 1, \ldots, d$ and do

$$f_j = \mathcal{S}_j\left(y_i - \alpha - \sum_{k \neq j} f_k(x_{i,k})\right)$$

   where S_j is some smoothing operation.
3. Stop when differences between iterations become small.

Backfitting

1. Let $\alpha = \frac{1}{N}\sum_{j=1}^{N} y_j$, $f_j := 0$.
2. Iterate $j = 1, \ldots, d$ and do

$$f_j = \mathcal{S}_j\left(y_i - \alpha - \sum_{k \neq j} f_k(x_{i,k})\right)$$

   where S_j is some smoothing operation.
3. Stop when differences between iterations become small.

Smoothing can be taken to be 1-dimensional on pairs $(x_{i,j}, y_i - \alpha - \sum_{k\neq j} f_k(x_{i,k}))$.

Basics

Estimate the outcome using

$$f(\mathbf{X}) = \sum_{k=1}^{K} c_k I(\mathbf{X} \in R_k)$$

where R_k are regions (usually rectangles).

Basics

Estimate the outcome using

$$f(\mathbf{X}) = \sum_{k=1}^{K} c_k I(\mathbf{X} \in R_k)$$

where R_k are regions (usually rectangles).

Why is this a "tree"? List of conditions:

$$(\text{root}, X_1 \le t_1), (\text{L}, X_2 \le t_2), (\text{R}, X_1 \le t_3) \ldots$$

Basics

Estimate the outcome using

$$f(\mathbf{X}) = \sum_{k=1}^{K} c_k I(\mathbf{X} \in R_k)$$

where R_k are regions (usually rectangles).

Why is this a "tree"? List of conditions:

$$(\text{root}, X_1 \leq t_1), (\text{L}, X_2 \leq t_2), (\text{R}, X_1 \leq t_3) \ldots$$

Each region defined by and-ing conditions from root to leaf.

Fitting (regression)

For a given region, the best estimator $\tilde{c}_k = \text{mean}(y_i | \mathbf{x}_i \in R_k)$.

Fitting (regression)

For a given region, the best estimator $\tilde{c}_k = \text{mean}(y_i | \mathbf{x}_i \in R_k)$.

For a fixed j, let $R_1(j, s) = \{\mathbf{X} | X_j \leq s\}$ and $R_2 = \{\mathbf{X} | X_j > s\}$.
Then consider

$$m_{j,s} = \min_{j,s} \sum_{\mathbf{x}_i \in R_1(j,s)} (y_i - \tilde{c}_1)^2 + \sum_{\mathbf{x}_j \in R_2(j,s)} (y_i - \tilde{c}_2)^2$$

Fitting (regression)

For a given region, the best estimator $\tilde{c}_k = \text{mean}(y_i | \mathbf{x}_i \in R_k)$.

For a fixed j, let $R_1(j,s) = \{\mathbf{X} | X_j \leq s\}$ and $R_2 = \{\mathbf{X} | X_j > s\}$. Then consider

$$m_{j,s} = \min_{j,s} \sum_{\mathbf{x}_i \in R_1(j,s)} (y_i - \tilde{c}_1)^2 + \sum_{\mathbf{x}_j \in R_2(j,s)} (y_i - \tilde{c}_2)^2$$

Optimal s is computationally tractable for each j, so choose optimal j, s pair at each iteration.

Stopping

We need a stopping condition! Let:

1. $N_m = |\{\mathbf{x}_i \in R_m\}|$.
2. $Q_m(T) = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} (y_i - \tilde{c}_m)^2$

Stopping

We need a stopping condition! Let:

1. $N_m = |\{\mathbf{x}_i \in R_m\}|$.
2. $Q_m(T) = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} (y_i - \tilde{c}_m)^2$

Define

$$C_\alpha(T) = \sum_{m=1}^{T} N_m Q_m(T) + \alpha |T|.$$

Stopping

We need a stopping condition! Let:

1. $N_m = |\{\mathbf{x}_i \in R_m\}|$.
2. $Q_m(T) = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} (y_i - \tilde{c}_m)^2$

Define

$$C_\alpha(T) = \sum_{m=1}^{T} N_m Q_m(T) + \alpha |T|.$$

Start with a grand tree T_0 found by stopping when a node reaches a fixed number of points (commonly 5). There is a unique smallest subtree T_α for each α.

Stopping

We need a stopping condition! Let:

1. $N_m = |\{\mathbf{x}_i \in R_m\}|$.
2. $Q_m(T) = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} (y_i - \tilde{c}_m)^2$

Define

$$C_\alpha(T) = \sum_{m=1}^{T} N_m Q_m(T) + \alpha |T|.$$

Start with a grand tree T_0 found by stopping when a node reaches a fixed number of points (commonly 5). There is a unique smallest subtree T_α for each α.

Tune α using k-fold cross validation, bootstrapping, etc.

Things To Look Out For

- Missing values: Imputing the mean of the non-missing values is a bad idea.

Things To Look Out For

- Missing values: Imputing the mean of the non-missing values is a bad idea.
- Instability: Small perturbations to the training data can lead to wild changes in splits.

Things To Look Out For

- Missing values: Imputing the mean of the non-missing values is a bad idea.
- Instability: Small perturbations to the training data can lead to wild changes in splits.
- Interpretability: Very high interpretability since it is easy to see which training samples influence a prediction.

Things To Look Out For

- Missing values: Imputing the mean of the non-missing values is a bad idea.
- Instability: Small perturbations to the training data can lead to wild changes in splits.
- Interpretability: Very high interpretability since it is easy to see which training samples influence a prediction.
- Non-continuity: Indicators are not continuous/differentiable which can be a negative in some contexts. (MARS and HME)

AdaBoost

Consider a classification where $Y \in \{\pm 1\}$.

AdaBoost

Consider a classification where $Y \in \{\pm 1\}$.

Usual error function

$$\overline{\text{err}} = \frac{1}{N}\sum_{i=1}^{N} I(y_i \neq G(\mathbf{x}_i)).$$

(in sample error).

AdaBoost

Consider a classification where $Y \in \{\pm 1\}$.

Usual error function

$$\overline{\text{err}} = \frac{1}{N} \sum_{i=1}^{N} I(y_i \neq G(\mathbf{x}_i)).$$

(in sample error).

Let $G_1, \dots, G_M$ be "weak" learners (ones that are slightly better than random).

AdaBoost

Consider a classification where $Y \in \{\pm 1\}$.

Usual error function

$$\overline{\text{err}} = \frac{1}{N} \sum_{i=1}^{N} I(y_i \neq G(\mathbf{x}_i)).$$

(in sample error).

Let $G_1, \ldots, G_M$ be "weak" learners (ones that are slightly better than random).

Final model:

$$G(\mathbf{x}) = \text{sign} \left(\sum_{m=1}^{M} \alpha_m G_m(\mathbf{x}) \right)$$

Fitting

1. Let $w_i = \frac{1}{N}$ for $i = 1, \ldots, N$.

Fitting

1. Let $w_i = \frac{1}{N}$ for $i = 1, \ldots, N$.
2. Fit a classifier G_m to the (weighted) training data using current weights.

Fitting

1. Let $w_i = \frac{1}{N}$ for $i = 1, \ldots, N$.
2. Fit a classifier G_m to the (weighted) training data using current weights.
3. Compute

$$\mathrm{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(\mathbf{x}_i))}{\sum_{i=1}^{N} w_i}$$

Fitting

1. Let $w_i = \frac{1}{N}$ for $i = 1, \ldots, N$.
2. Fit a classifier G_m to the (weighted) training data using current weights.
3. Compute
$$\text{err}_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(\mathbf{x}_i))}{\sum_{i=1}^N w_i}$$
4. Set
$$\alpha_m = \log\left(\frac{1 - \text{err}_m}{\text{err}_m}\right)$$

Fitting

1. Let $w_i = \frac{1}{N}$ for $i = 1, \ldots, N$.
2. Fit a classifier G_m to the (weighted) training data using current weights.
3. Compute
$$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(\mathbf{x}_i))}{\sum_{i=1}^{N} w_i}$$
4. Set
$$\alpha_m = \log\left(\frac{1 - \text{err}_m}{\text{err}_m}\right)$$
5. Update:
$$w_i = w_i e^{\alpha_m I(y_i \neq G_m(\mathbf{x}_i))}$$

Back to Additive Models

Suppose we want to fit

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m b(\mathbf{x}, \gamma_m)$$

Back to Additive Models

Suppose we want to fit

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m b(\mathbf{x}, \gamma_m)$$

1. Initialize $f_0 := 0$

Back to Additive Models

Suppose we want to fit

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m b(\mathbf{x}, \gamma_m)$$

1. Initialize $f_0 := 0$
2. Compute

$$(c_m, \gamma_m) = \arg\min_{c,\gamma} L(Y, f_{m-1}(\mathbf{X}) + cb(\mathbf{X}, \gamma))$$

Back to Additive Models

Suppose we want to fit

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m b(\mathbf{x}, \gamma_m)$$

1. Initialize $f_0 := 0$
2. Compute

$$(c_m, \gamma_m) = \arg\min_{c,\gamma} L(Y, f_{m-1}(\mathbf{X}) + cb(\mathbf{X}, \gamma))$$

3. Update

$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + c_m b(\mathbf{x}, \gamma_m)$$

Computational Considerations

- Previous example can be intractable depending on the Loss function and basis function involved.

Computational Considerations

- Previous example can be intractable depending on the Loss function and basis function involved.
- However, this can often be reduced to just finding

$$\arg\min_{c,\gamma} L(Y, cb(\mathbf{X}, \gamma))$$

Computational Considerations

- Previous example can be intractable depending on the Loss function and basis function involved.
- However, this can often be reduced to just finding

$$\arg\min_{c,\gamma} L(Y, cb(\mathbf{X}, \gamma))$$

- For instance,

$$L(Y, f(\mathbf{X})) = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2$$

reduces to

$$(y_i - f_{m-1}(\mathbf{x}_i) - cb(\mathbf{x}, \gamma))$$

Computational Considerations

- Previous example can be intractable depending on the Loss function and basis function involved.
- However, this can often be reduced to just finding

$$\arg\min_{c,\gamma} L(Y, cb(\mathbf{X}, \gamma))$$

- For instance,

$$L(Y, f(\mathbf{X})) = \sum_{i=1}^{N} (y_i - f(\mathbf{x}_i))^2$$

reduces to

$$(y_i - f_{m-1}(\mathbf{x}_i) - cb(\mathbf{x}, \gamma))$$

- So in effect just best fit of current residual.

Gradient Boosting

Suppose the loss function L is differentiable and define

$$L(f) = L(Y, f(\mathbf{X}))$$

Gradient Boosting

Suppose the loss function L is differentiable and define

$$L(f) = L(Y, f(\mathbf{X}))$$

Let

$$\mathbf{f} = (f(\mathbf{x_i}))$$

then

$$\hat{(}f) = \arg\min_{\mathbf{f}} L(\mathbf{f})$$

Gradient Boosting

Suppose the loss function L is differentiable and define

$$L(f) = L(Y, f(\mathbf{X}))$$

Let

$$\mathbf{f} = (f(\mathbf{x_i}))$$

then

$$\hat{(}f) = \arg\min_{\mathbf{f}} L(\mathbf{f})$$

Start with an initial guess. At each step, compute

$$g_{i,m} = \left[\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f}\right]_{f=f_m}$$

Gradient Boosting

Suppose the loss function L is differentiable and define

$$L(f) = L(Y, f(\mathbf{X}))$$

Let

$$\mathbf{f} = (f(\mathbf{x_i}))$$

then

$$\hat{(}f) = \arg\min_{\mathbf{f}} L(\mathbf{f})$$

Start with an initial guess. At each step, compute

$$g_{i,m} = \left[\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f}\right]_{f=f_m}$$

Update

$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \rho_m g_{i,m}$$

For Trees

1. Initialize $f_0(\mathbf{x}) = \arg\min_\gamma L(Y, \gamma)$.

For Trees

1. Initialize $f_0(\mathbf{x}) = \arg\min_\gamma L(Y, \gamma)$.
2. Compute
$$r_{i,m} = \left[\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f}\right]_{f=f_{m-1}}$$

For Trees

1. Initialize $f_0(\mathbf{x}) = \arg\min_\gamma L(Y, \gamma)$.
2. Compute
$$r_{i,m} = \left[\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f}\right]_{f=f_{m-1}}$$
3. Fit a regression tree using $\mathbf{r}_m$ as targets.

For Trees

1. Initialize $f_0(\mathbf{x}) = \arg\min_\gamma L(Y, \gamma)$.
2. Compute
$$r_{i,m} = \left[\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f}\right]_{f=f_{m-1}}$$
3. Fit a regression tree using $\mathbf{r}_m$ as targets.
4. Compute
$$\gamma_{j,m} = \arg\min_\gamma \sum_{\mathbf{x}_i \in R_{j,m}} L(y, f_{m-1}(\mathbf{x}) + \gamma)$$

For Trees

1. Initialize $f_0(\mathbf{x}) = \arg\min_\gamma L(Y, \gamma)$.
2. Compute
$$r_{i,m} = \left[\frac{\partial L(y_i, f(\mathbf{x}_i))}{\partial f}\right]_{f=f_{m-1}}$$
3. Fit a regression tree using $\mathbf{r}_m$ as targets.
4. Compute
$$\gamma_{j,m} = \arg\min_\gamma \sum_{\mathbf{x}_i \in R_{j,m}} L(y, f_{m-1}(\mathbf{x}) + \gamma)$$
5. Update
$$f_m(\mathbf{x}) = f_{m-1}(\mathbf{x}) + \sum_{j=1}^{J_m} \gamma_{j,m} I(\mathbf{x} \in R_{j,m})$$