INDR 450/550

Spring 2022

Lecture 17: GAMs and Trees

April 18, 2022

Fikri Karaesmen

Announcements

- Class Exercise at the end of lecture today. If you are participating online, please upload your document under Course Contents/Class Exercises
- Lab 7 material (on lasso and ridge) and a short video are available
- Exam on May 7.
  - Some exercises will be available

- The first seven labs were uploaded. Please follow them.

Predictive Analytics

- Remaining topics (to complete at the latest the week after the spring break)
  - Validation ✓
  - Model selection / regularization ✓
  - Non-linear regressions, ✓
  - Generalized additive models, Tree-based methods (today)

Non-linear regressions: basis functions

- **Basis Functions**: Consider a family of transformations for the predictor X: $b_1(X)$, $b_2(X)$,...,$b_n(X)$.
- Polynomials are an example where we have $b_i(X) = X^i$.
- We then consider the following regression using the basis variables:

$$y_t = \beta_0 + \beta_1 b_1(x_t) + \beta_2 b_2(x_t) + ... + \beta_n b_n(x_t) + \epsilon_t$$

- All tools from ordinary least squares regression are available under such transformations.

Non-linear regressions: knots / cubic splines

- Approximations of functions using knots fall in the class of spline approximations.
- Note that when we use a knot $(t - t_1)^+$, we are allowing a slope change at t_1. This results in a change of slope at t_1 and therefore a non-smooth fit.
- To have a smooth curve, we need continuity at the knot and also the continuity of the first and second derivatives.
- The basis that gives such smooth functions consists of the following knots:

$$\left((t - t_1)^+\right)^3 = \begin{cases} (t - t_1)^3 & \text{if } t > t_1 \\ 0 & \text{otherwise} \end{cases}$$

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Non-linear regressions: knots / cubic splines

- Approximations of functions using knots fall in the class of spline approximations.
- Note that when we use a knot $(t - t_1)^+$, we are allowing a slope change at t_1. This results in a change of slope at t_1 and therefore a non-smooth fit.
- To have a smooth curve, we need continuity at the knot and also the continuity of the first and second derivatives.
- The basis that gives such smooth functions consists of the following knots:

$$\left((t - t_1)^+\right)^3 = \begin{cases} (t - t_1)^3 & \text{if } t > t_1 \\ 0 & \text{otherwise} \end{cases}$$

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Non-linear regressions: knots / cubic splines

To determine the location and the number of knots: try several ones and choose by cross-validation.

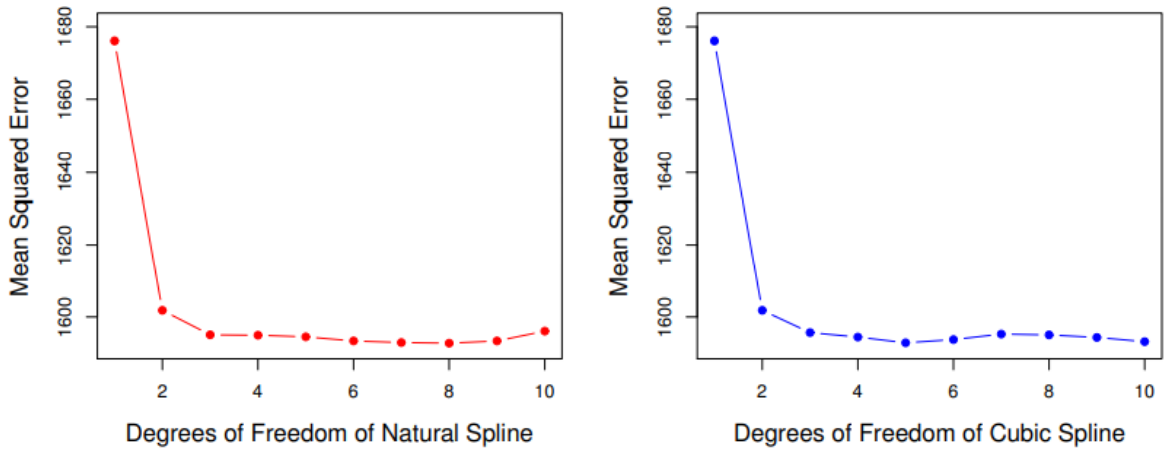


FIGURE 7.6. *Ten-fold cross-validated mean squared errors for selecting the degrees of freedom when fitting splines to the* `Wage` *data. The response is* `wage` *and the predictor* `age`. Left: *A natural cubic spline.* Right: *A cubic spline.*

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Some trials on the Google Share Price Data

- I also a tried a cubic spline with three knots at 25,40 and 60.

```
In [35]: fit1 = sm.GLM(y_train, t_3knots).fit()
         fit1.params

Out[35]: array([2066.14112978,    5.07698694,  -79.74134178,  304.27959612,
                 266.51573955,  877.49354265,  781.23132479])

In [36]: pred1 = fit1.predict()

In [45]: np.sqrt(np.mean(np.square(pred1-y_train)))

Out[45]: 40.530954127347194

In [43]: np.mean(np.abs((pred1-y_train)/y_train))

Out[43]: 0.013367668355775858
```

RMSE (train) = 40.53

Non-linear regressions: GAMS

- Generalized additive models extend the one predictor approach to multiple predictors assuming a model of the form:

$$y_t = \beta_0 + f_1(x_{t2}) + f_2(x_{t2}) + ... + f_p(x_{tp}) + \epsilon_t$$

- We separately explain the effects of each predictor and then sum them up.

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Non-linear regressions: GAMS

- GAMs are flexible tools somewhere between linear regression and non-parametric approaches.
- With standard basis functions (i.e. cubic splines) we can use least squares regression to fit GAMs.
- Since non-linearity is considered, they may be more accurate fits than a linear regression.
- We can control the smoothness of the curve that is fit by the choice of basis
- GAMs do not consider interaction effects between predictors but these can be added manually.
- They extend to logistic regression directly.

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Non-linear regressions: GAMS / Gradient Boosting

- Gradient boosting is a sequential optimization approach that has proved useful in improving predictions. Consider:

$$y_t = \beta_0 + f_1(x_{t2}) + f_2(x_{t2}) + ... + f_p(x_{tp}) + \epsilon_t$$

- Instead of trying to solve the least squares problem with all predictors (and their transformations), we use a greedy sequential optimization approach.
- We fit each of the p predictors separately and individually. We select the one that gives the best MSE and compute the residuals from this best fit.
- We then use the remaining predictors to find the best individual fit that explains the residuals from the first round.
- and continue in this manner fitting the best individual predictor the remaining residuals after each round.
- Note that more of the remaining residual is explained in each round.

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Tree-based Methods

- Tree-based methods segment the predictor space into multiple regions in a smart way. They then make a very simple prediction for each region using the training data.
- The prediction for a given region is simply the average of the observations in the training data that fall into that region.
- Conceptually simple and easy to explain.
- But require additional work to be competitive with earlier methods (ridge, lasso, GAMs etc.)

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Tree-based Methods

Predicting the salary of a baseball player using his experience (years) and Performance (# of hits).

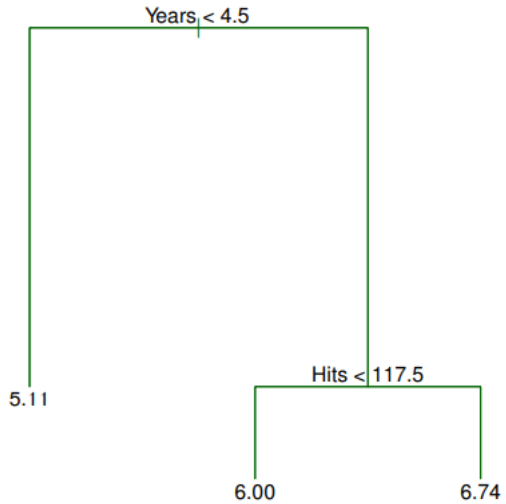


FIGURE 8.1. *For the* `Hitters` *data, a regression tree for predicting the log salary of a baseball player, based on the number of years that he has played in the major leagues and the number of hits that he made in the previous year. At a given internal node, the label (of the form* $X_j < t_k$*) indicates the left-hand branch emanating from that split, and the right-hand branch corresponds to* $X_j \geq t_k$*. For instance, the split at the top of the tree results in two large branches. The left-hand branch corresponds to* `Years<4.5`*, and the right-hand branch corresponds to* `Years>=4.5`*. The tree has two internal nodes and three terminal nodes, or leaves. The number in each leaf is the mean of the response for the observations that fall there.*

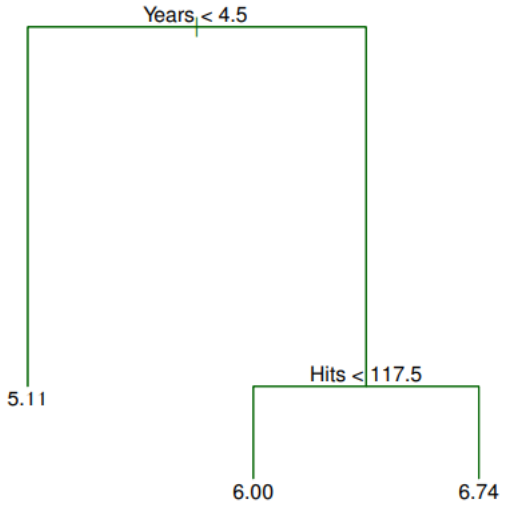
From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

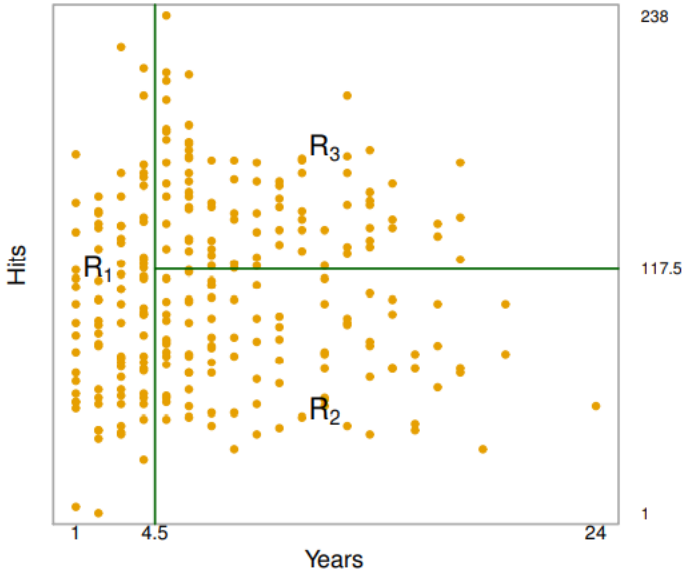
Tree-based Methods

- Here's a summary of the basic approach.
  1. We divide the predictor space (the space of all possible values of the predictors $(X_1, X_2, ..., X_p)$ into J distinct (non-overlapping regions) R_1, R_2,..,R_j.
  2. For any observation that falls in region R_j, we use the same prediction: the average of all observations from that region in the training set.
  3. Note that this is natural for binary variables. If it's a weekend day, we make a prediction based on past weekend days and if it's a week day, we make a prediction based on week days.
- We are making a prediction for $\mathbf{x}_p$ that falls in region R_j. The mean of all training observations in $R_j = 200$, then $\hat{y}(\mathbf{x}_p) = 500$

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Tree-based Methods





From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Tree-based Methods

- The common approach is to divide to predictor space into p dimensional boxes. We then choose J overlapping boxes such that:

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \bar{y}_{R_j})^2$$

- We immediately note that finding the exactly optimal partition (boxes) is computationally prohibitive.
- We take a greedy approach called recursive binary splitting.

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Tree-based Methods: recursive binary splitting

- For each predictor X_j, we find the optimal cutpoint s: to do this we define the half planes:

$$R_1(j,s) = \{X|X_j < s\} \text{ and } R_2(j,s) = \{X|X_j \geq s\}$$

- and we seek the cutpoint s that minimizes

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \bar{y}_{R1})^2 + \sum_{i:x_i \in R_2(j,s)} (y_i - \bar{y}_{R2})^2$$

- We repeat this for all predictors and choose the one that leads to lowest residual sum of squares as the top split of the tree.

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Tree-based Methods: recursive binary splitting

- The top of the tree now splits at the chosen predictor j, and cutoff s'.
- We repeat the same procedure for the second split. We are then dividing the partition from stage 1 further.
- We repeat the splitting until we reach the desired number of regions or few observations remain in each region.
- This it not computationally difficult when the number of predictors is not large.

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Tree-based Methods: pruning

- Note that as usual, overfitting is an issue. We would prefer trees that are small (i.e. use few regions) to avoid overfitting.
- We can stop the split when the incremental decrease in RSS is lower than a threshold but the this would not be the globally optimal way to do it.
- A better strategy first grows a large tree and then prunes it down to obtain a smaller subtree.

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Tree-based Methods: pruning

- We grow a large tree T_0 using recursive binary splitting.
- We consider a penalty parameter α and solve the following minimization problem:

$$\min \sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

  where T denotes the number of terminal nodes of the subtree T.
- This is similar to lasso. If $\alpha = 0$, we use the initial tree T_0 but as α increases some of the terminal nodes will have to be reduced resulting in a loss in residual sum of squares. The optimization finds the right tradeoff between which branches to prune and the loss in residual sum of squares.

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Tree-based Methods: pruning algorithm

Algorithm 8.1 *Building a Regression Tree*

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.

2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees, as a function of α.

3. Use K-fold cross-validation to choose α. That is, divide the training observations into K folds. For each $k = 1, \ldots, K$:

   (a) Repeat Steps 1 and 2 on all but the kth fold of the training data.

   (b) Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of α.

   Average the results for each value of α, and pick α to minimize the average error.

4. Return the subtree from Step 2 that corresponds to the chosen value of α.

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Tree-based Methods: classification

- The tree-based approach extends direclty to classification problems.
- The only difference is that as before MSE or RSS cannot be the error criterion.
- The error function takes into account the classification errors.
  - There are other useful error measurements such as:
  - Gini index
  - Cross-entropy

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani

Tree-based Methods: advantages and disadvantages

- Decision trees for regression are easy to communicate and understand
- Natural for qualitative predictors
- But do not have the same level of predictive accuracy as the previous models
- There are some robustness issues. A small change in data can lead to a big change in the resulting regression tree.
- The variance of the typical tree-based regression estimator is high (much higher than a linear regression based estimator).

From *An Introduction to Statistical Learning*, James, Witten, Hastie, Thibshirani