Lecture 3: Prediction

Splines, Additive Models, Stepwise Model Selection

Prof. Alexandra Chouldechova
95-791: Data Mining

March 22, 2016
Course Roadmap

95-791

Predictive Analytics

Descriptive Analytics

Prediction

Classification
Course Roadmap

95-791

- Predictive Analytics
- Descriptive Analytics

- Prediction
  - Classification
• The goal of prediction is to estimate the true, unknown regression function, $f$. 
Recap of last class

- Linear regression imposes **two key restrictions** on the model: We assume the relationship between the response $Y$ and the predictors $X_1, \ldots, X_p$ is:
  1. Linear
  2. Additive

- The truth is **almost never linear**; but often the linearity and additivity assumptions are **good enough**

- When we think **linearity** might not hold, we can try…
  - Polynomials
  - Step functions
  - Splines
  - Local regression
  - Generalized additive models

- When we think the **additivity** assumption doesn't hold, we can incorporate **interaction terms**

- These variants offer increased **flexibility**, while retaining much of the ease and **interpretability** of ordinary linear regression
Recap of last class

Polynomials

lm(wage \sim \text{poly}(age, 4), \text{data} = \text{Wage})

Step Functions

lm(wage \sim \text{cut}(age, 
    \text{breaks} = c(-\text{Inf}, 25, 35, 65, \text{Inf})), 
    \text{data} = \text{Wage})

• We can think of polynomials and step functions as simple forms of feature engineering

• These extensions of linear regression enable us to fit much more flexible models
Prediction topics

- Linear models
- Similarity-based models
- Tree-based models
- Model selection
- Model validation
Prediction topics

Prediction

- Additive models
- Similarity-based models
- Tree-based models
- Model selection
- Model validation
Prediction topics

- Additive models
- Similarity-based models
- Tree-based models
- Model selection
- Model validation
Prediction topics

- Additive models
- Similarity-based models
- Tree-based models
- Model selection
- Model validation
Agenda

• Piecewise polynomial fits

• Splines

• Additive models

• Stepwise model selection (Significance tests, AIC, BIC)
Some motivation

- If the true regression function $f$ is non-linear, ordinary linear regression cannot estimate it consistently.
- No matter how much data you get, the best ordinary linear regression can do is converge to the best linear approximation to $f$. 
• The same problem persists with polynomial regression.

• No matter how much data you get, the best degree-$k$ polynomial regression can do is converge to the best degree-$k$ approximation to $f$. 
Consistency

Consistent estimator

An estimator $\hat{f}(x)$ is consistent if, as our sample size grows, $\hat{f}(x)$ converges to the true regression function $f(x) = \mathbb{E}(Y | X = x)$.

- Unless $f(x)$ is a polynomial of degree $\leq k$, degree-$k$ polynomial regression is not consistent.

Q: Can we get a consistent estimator of a generic regression function $f$?

A: If we're willing to assume $f$ is smooth, then YES.

Use splines!
To understand splines, we first need to understand piecewise polynomials

- We can think of step functions as piecewise constant models
- It's easy to generalize this idea to:
  - Piecewise linear
  - Piecewise quadratic …
  - Piecewise polynomial
Piecewise Polynomial vs. Regression Splines

Piecewise Cubic

Continuous Piecewise Cubic

Cubic Spline

Linear Spline
Piecewise polynomial vs. Regression splines

1 break at \text{Age} = 50

\textbf{Definition: Cubic spline}

A \textit{cubic spline} with knots at $x$-values $\xi_1, \ldots, \xi_K$ is a \textit{continuous piecewise cubic polynomial} with \textit{continuous derivates} and \textit{continuous second derivates} at each knot.
Cubic Splines

- Turns out, **cubic splines** are sufficiently **flexible** to **consistently** estimate smooth regression functions $f$
- You can use higher-degree splines, but **there's no need to**
- To fit a cubic spline, we just need to pick the **knots**
Polynomial regression vs. Cubic splines

- In **polynomial regression**, we had to choose the **degree**.

- For **cubic splines**, we need to choose the **knots**.

**Q:** How **complex** is a cubic spline with $K$ knots?

*Paraphrasing...* A cubic spline with $K$ knots is as complex as a polynomial of degree ____?

- Turns out, there exist functions $b_k(x)^1$ such that a cubic spline with $K$ knots can be modeled as

$$y = \beta_0 + \beta_1 b_1(x) + \beta_2 b_2(x) + \cdots + \beta_{K+3} b_{K+3}(x) + \epsilon$$

- **A:** A cubic spline with $K$ knots is as complex as a polynomial of degree $K + 3$.

---

^1See ISLR pg. 273 for details
Degrees of freedom

- **Degrees of freedom** capture the *complexity* of a regression model.
- A linear regression model with \( p \) independent predictors is said to have \( p \) degrees of freedom \(^2\).
- Take-away from the previous slide:

<table>
<thead>
<tr>
<th>Model</th>
<th># knots</th>
<th>Degrees of freedom</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression, ( p ) predictors</td>
<td></td>
<td>( p )</td>
</tr>
<tr>
<td>Degree-( k ) polynomial regression</td>
<td></td>
<td>( k )</td>
</tr>
<tr>
<td>Cubic spline</td>
<td>( k )</td>
<td>( k + 3 )</td>
</tr>
<tr>
<td>Degree-( d ) Spline</td>
<td>( k )</td>
<td>( k + d )</td>
</tr>
</tbody>
</table>

- In the following slides, we compare *cubic splines* to *polynomial regression*, allowing each method the same *degrees of freedom*.

---

\(^2\)Technically, \( p + 1 \) if you count the intercept. To be consistent with R's `df` arguments, here we *do not* count the intercept.
Polynomial regression vs. Cubic splines

\[
\text{lm(wage} \sim \text{bs(age, df = 3))}
\]
\[
\text{lm(wage} \sim \text{poly(age, degree = 3))}
\]
Polynomial regression vs. Cubic splines

\texttt{\textcolor{blue}{lm(wage} \sim \texttt{bs(age, df = 5))}}

\texttt{\textcolor{blue}{lm(wage} \sim \texttt{poly(age, degree = 5))}}
Polynomial regression vs. Cubic splines

\begin{align*}
\text{lm}(\text{wage} & \sim \text{bs}(\text{age}, \text{df} = 10)) \\
\text{lm}(\text{wage} & \sim \text{poly}(\text{age}, \text{degree} = 10))
\end{align*}
Polynomial regression vs. Cubic splines

```
lm(wage ~ bs(age, df = 15))
```

```
lm(wage ~ poly(age, degree = 15))
```
Natural Cubic Splines

Figure: 7.4 from ISLR. Natural cubic splines are cubic splines that extrapolate linearly beyond the boundary knots. A NCS with $K$ knots uses just $K + 1$ degrees of freedom --- the same as a cubic spline with 2 fewer knots.
Natural Cubic Splines vs Polynomial regression

Figure: 7.7 from ISLR. Natural cubic splines are very nicely behaved at the tails of the data. Polynomial regression shows erratic behaviour. (14 degrees of freedom used for both)
Cubic Splines vs Polynomial regression

- **Polynomial regression** must use a high degree in order to produce flexible fits.
- With **splines**, we can keep the degree fixed, an increase flexibility by adding knots.
- Splines generally tend to be better behaved at the same level of model complexity.
Knot placement: Rules of thumb

- Place **more knots** where $f$ appears to be changing rapidly
- Place **fewer knots** where $f$ appears to be slowly varying

**R's defaults:**
The most common way of specifying splines in **R** is in terms of the spline's **degrees of freedom** (df). **R** then places knots at suitably chosen **quantiles** of the $x$ variable.

<table>
<thead>
<tr>
<th>Model</th>
<th><strong>R</strong> command</th>
<th># internal knots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubic Spline</td>
<td>$\sim$ bs(x, df)</td>
<td>df – 3</td>
</tr>
<tr>
<td>Natural Cubic Spline</td>
<td>$\sim$ ns(x, df)</td>
<td>df – 1</td>
</tr>
<tr>
<td>Degree-$d$ Spline</td>
<td>$\sim$ bs(x, df, degree = d)</td>
<td>df – $d$</td>
</tr>
</tbody>
</table>

- You may also specify the internal knots *manually* for **ns** and **bs** by specifying the **knots =** argument directly. E.g.,
  
  ```r
  lm(wage ~ bs(age, knots = c(25, 40, 60)), data = Wage)
  ```
Q: What is the best way to automatically place knots?

Paraphrasing…If I know how many degrees of freedom I want my spline to have, is there a method for automatically choosing the best locations for the knots?

A: Smoothing splines
Smoothing splines

- The smoothing spline estimator is the solution $\hat{g}$ to the problem:

$$\text{minimize } \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

- This is a penalized regression problem.

- We're saying we want a function that:
  1. Fits the data well; and
  2. isn't too wiggly

- Large $\lambda \Rightarrow \hat{g}$ will have low variability (\& higher bias)
- Small $\lambda \Rightarrow \hat{g}$ will have high variability (\& lower bias)

How is this at all related to splines?

---

3 We'll see more examples of penalized regression next class.
Smoothing splines

\[ \text{minimize } \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt \]

It turns out…

- The solution to \((*)\) is a **natural cubic spline**
- The solution has knots at every unique value of \(x\)
- The **effective degrees of freedom** of the solution is calculable
- \(\lambda \leftrightarrow df\)

**Coding tip:** In R with the `gam` library you can use the syntax `s(x, df)` in your regression formula to fit a smoothing spline with `df` effective degrees of freedom.
Figure: 7.8 from ISLR. We’ll talk about LOOCV (leave-one-out cross-validation) next class.
Another method people like: Local regression

Figure 7.9 from ISLR. Local regression (enabled as lo(x) and loess(x))
Putting everything together: Additive Models

• Recall the **Linear Regression Model**

\[
Y = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \epsilon
\]

• We can now extend this to the far more **flexible Additive Model**

\[
Y = \beta_0 + \sum_{j=1}^{p} f_j(X_j) + \epsilon
\]

• Each \(f_j\) can be **any of the different methods** we just talked about: Linear term \((\beta_j X_j)\), Polynomial, Step Function, Piecewise Polynomial, Degree-\(k\) spline, Natural cubic spline, Smoothing spline, Local linear regression fit, …

• You can mix-and-match different kinds of terms

• The **gam** and **mgcv** packages enable Additive Models in R
Additive Models: Boston housing data

Using the `gam` library, we fit the model

\[
gam(\text{medv} \sim s(lstat, 5) + lo(ptratio) +
   \text{cut(crim, breaks = c(-Inf, 1, 10, 25, Inf))},
   \text{data = Boston})
\]

- This amounts to an additive model

\[
\text{medv} = f_1(lstat) + f_2(ptratio) + f_3(crim) + \epsilon
\]

with terms:

- \(f_1(lstat)\) smoothing spline with 5 df
- \(f_2(ptratio)\) local linear regression
- \(f_3(crim)\) step function with breaks at \(\text{crim} = 1, 10, 25\)
Additive Models: Boston housing data

\[
\hat{f}_1(lstat) \\
\hat{f}_2(ptratio) \\
\hat{f}_3(crim)
\]

\[
\text{gam(medv} \sim \text{s(lstat, 5) + lo(ptratio) + cut(crim, breaks = c(-Inf, 1, 10, 25, Inf)), data = Boston)}
\]
Summary

• **Splines** are a nice way of modeling *smooth* regression functions

• To increase the *flexibility* of a spline, we increase the number of *knots*

• **Natural cubic splines** allow us *retain the model complexity of a cubic spline* while adding *two extra interior knots* at the cost of restricting our model to be *linear* outside the range of the observed data

• **Smoothing splines** enable us to avoid the problem of *knot selection* altogether, and instead specify a single parameter: the desired effective *degrees of freedom* for the fit

• We can put everything together into an **Additive Model**

\[ Y = \beta_0 + \sum_{j=1}^{p} f_j(X_j) + \epsilon \]

where each \( f_j \) can be any of the fits we talked about.
At this stage, we have a lot of questions.

- How do we choose which variables to include in a linear regression?
- How do we choose the degree $k$ in a polynomial regression?
- How do we choose the cuts in a step function regression?
- How do we decide on how many knots to place and where to place them when fitting regression splines?
- How should we choose $\lambda$ or the effective degrees of freedom for a smoothing spline?
- Which variables should we include in an additive model, and what form should we pick for each $f_j$ term?

All of these questions are essentially asking the same thing...
How do we pick the best model?
How do we pick the best statistical model?
How should we think about model selection?

- We want to add variables/flexibility as long as doing so helps capture meaningful trends in the data
- i.e., we want to find the right balance between the model's goodness-of-fit and complexity
One approach

We can think of this as choosing \( \hat{f} \) to:

\[
\text{minimize } \text{Lack-of-fit}(f) + \gamma \cdot \text{Model complexity}(f)
\]

- The **Residual Sum of Squares** (RSS) is our usual way of assessing the lack-of-fit of a model.
- For **additive models**, we saw that we could measure **complexity** using degrees of freedom.

A refined proposal: Choose \( \hat{f} \) to:

\[
\text{minimize } \sum_{i=1}^{n} (y_i - f(x_i))^2 + \gamma \cdot \sum_{j=1}^{p} \text{df}(f_j)
\]

- What should \( \gamma \) be?
Model Selection: AIC and BIC

\[
\text{minimize } \sum_{i=1}^{n} (y_i - f(x_i))^2 + \gamma \cdot \sum_{j=1}^{p} \text{df}(f_j)
\]

- \( \gamma = 2\hat{\sigma}^2 \) is the AIC criterion\(^4\)
- \( \gamma = \log(n)\hat{\sigma}^2 \) is the BIC criterion
- For \( n \geq 8 \), it's clear that the BIC criterion puts a heavier penalty on model complexity compared to AIC

\(^4\hat{\sigma}^2 \) is the estimated variance of the noise/error term: \( \text{var}(\epsilon) = \sigma^2 \)
Example: AIC to select df for smoothing spline (Wage data)

```r
> model0 <- gam(wage ~ 1, data = Wage)
>
> step.gam(model0, scope = list("age"= ~ 1 + age + ns(age, 2) + ns(age, 3) + ns(age, 4) + ns(age, 5) + ns(age, 6) + ns(age, 7) + ns(age, 8))))

Start:  wage ~ 1;  AIC= 30903.75
Step:1  wage ~ age ;  AIC= 30788.67
Step:2  wage ~ ns(age, 2) ;  AIC= 30653.21
Step:3  wage ~ ns(age, 3) ;  AIC= 30641.39
Call:
gam(formula = wage ~ ns(age, 3), data = Wage, trace = FALSE)

Degrees of Freedom: 2999 total; 2996 Residual
Residual Deviance: 4775233
```

`step.gam` is used to consider **Natural cubic splines** with df from 0 to 8. The AIC criterion chooses df = 3.
What a beautiful model!

```
lm(wage ~ ns(age, 3), data = Wage)
```
Acknowledgements

All of the lectures notes for this class feature content borrowed with or without modification from the following sources:

- 36-462/36-662 Lecture notes (Prof. Tibshirani, Prof. G'Sell, Prof. Shalizi)
- 95-791 Lecture notes (Prof. Dubrawski)
- *An Introduction to Statistical Learning, with applications in R* (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani