Linear Regression and Machine Learning

Ryan Miller

Grinnell College
Statistics

Introduction

- Up until this point we've exclusively used two algorithms, KNN and decision trees, to allow us to focus on other aspects of machine learning
  - This included training vs. testing, pre-processing pipelines, cross-validation, hyperparameter selection, and performance evaluation
- For the next few weeks we will expand our toolbox of algorithms, focusing on methods that are suitable for "flat" or "tabular" data
  - Today's focus is *linear regression*, which I assume you have some familiarity with already

Linear Regression

Linear regression is a *supervised* method that expresses the outcome as a linear combination of predictors:

$$y_i = f(\mathbf{x}_i) + \text{Noise}$$

where: $f(\mathbf{x}_i) = w_0 + w_1 x_{i1} + w_2 x_{i2} + \ldots + w_p x_{ip}$

- In machine learning, the coefficients, $\{w_1, w_2, \ldots, w_p\}$, are called "weights" and the intercept, w_0, is called the "bias"
- Regression is a **parametric** method as it uses a fixed number of parameters (weights) to define the functional form relating the predictors and outcome
  - KNN and decision trees are *non-parametric*, as their structure is determined by the data, not a pre-defined function

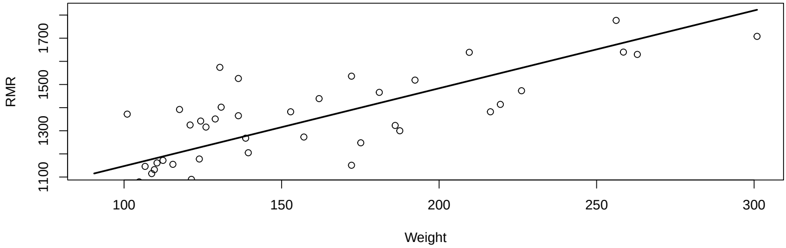
Linear Regression Example

Consider predicting the resting metabolic rate of an individual using their body weight. Here the raw data looks like a single predictor and the outcome:

weight_lbs	rate_kcal
104.79	1079
106.68	1146
108.78	1115
110.46	1161
120.96	1325
128.94	1351

Linear Regression Example (cont.)

We can apply the linear regression algorithm, which estimates the parameters w_0 and w_1 to minimize the error on the training data:



Does this seem like a *high bias* or a *high variance* model? Could we manipulate this tradeoff using the current model?

Grinnell College
Statistics

Feature Expansion

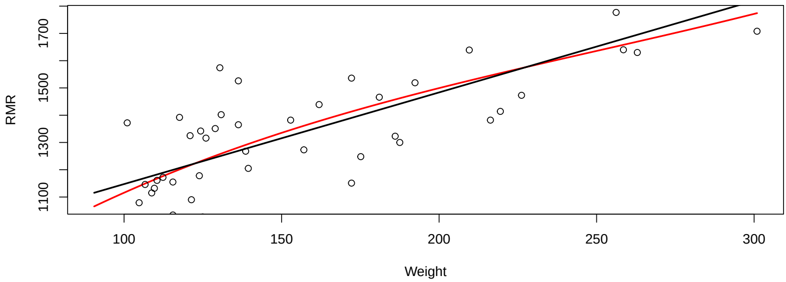
To add flexibility to our linear model, we might *expand* our single predictor using polynomials:

X1	X2	X3	rate_kcal
104.79	10980.94	1150693	1079
106.68	11380.62	1214085	1146
108.78	11833.09	1287203	1115
110.46	12201.41	1347768	1161
120.96	14631.32	1769805	1325
128.94	16625.52	2143695	1351

We're now using 3 columns to represent an individual's body weight: weight, weight squared, and weight cubed.

Polynomials

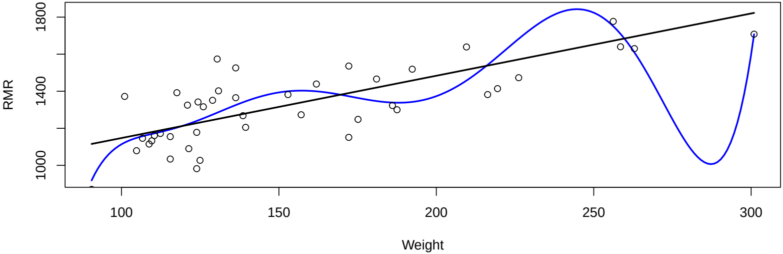
A linear regression model fit to the expanded data is shown in red:



Did feature expansion improve the accuracy of our model on the training data?

Polynomials (cont.)

The second model estimated 3 weight parameters (and 1 bias) from the data, it has greater flexibility to represent small trends.



However, more flexibility is not always better, as the blue line depicts an 8th degree polynomial expansion. See any problems?

Discretization

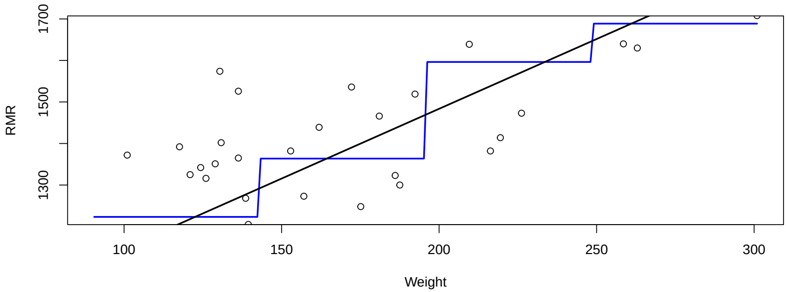
A simple alternative to polynomial expansion is *discretization*:

(90.3,143]	(143,196]	(196,248]	(248,301]	rate_kcal
1	0	0	0	1079
1	0	0	0	1146
1	0	0	0	1115
1	0	0	0	1161
1	0	0	0	1325
1	0	0	0	1351

The idea is to split a numeric predictor in to categories and represent them using one-hot encoding.

Discretization (cont.)

The discretizing body weight into 4 equally spaced bins yields the following model:



What are some strengths/weaknesses of this approach?

Grinnell College
Statistics

Splines

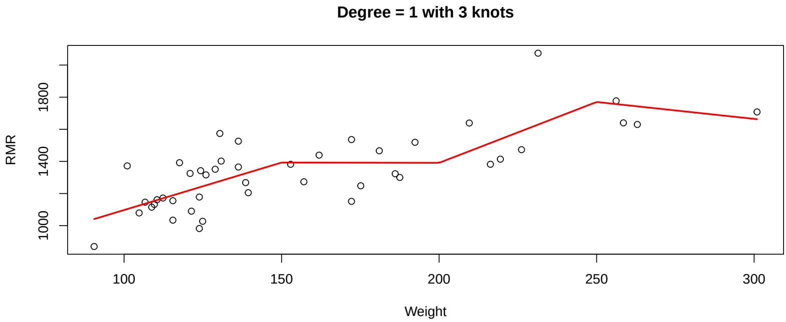
Splines are an alternative without some of the negative aspects of polynomials and discretization:

X1	X2	X3	rate_kcal
0.177	0.013	0.000	1079
0.196	0.016	0.000	1146
0.217	0.021	0.001	1115
0.233	0.024	0.001	1161
0.318	0.054	0.003	1325
0.366	0.082	0.006	1351

Basis splines, or "b-splines", use a basis matrix to represent piecewise polynomials that connect at specified interior knots

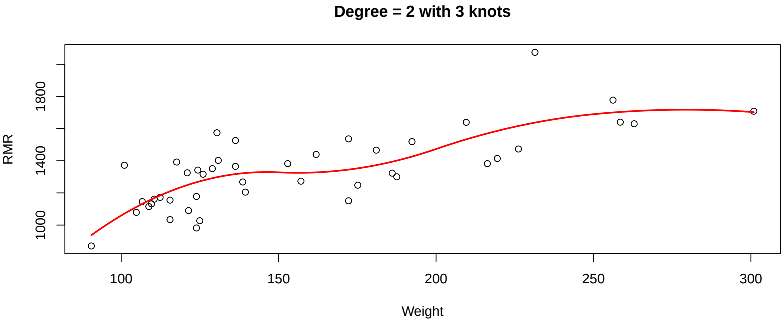
Splines (cont.)

Polynomials with degree $= 1$ are just lines, the model below demonstrates a b-spline with 3 knots and degree $= 1$:



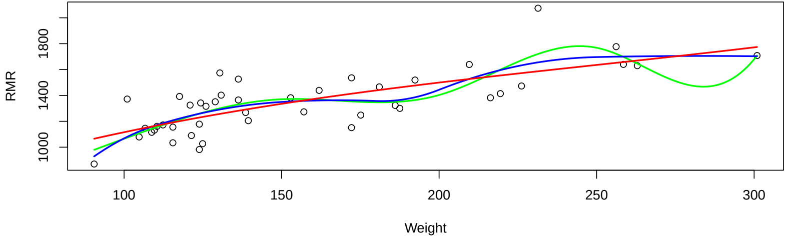
Splines (cont.)

Higher degree splines ensure smoothness by requiring continuity of derivatives up to the order degree minus one (so quadratic splines require continuity of the first derivative, or the slope at the location of the knot):



Splines (cont.)

The polynomial degree and number of knots can be used to manipulate the flexibility of b-splines:



- Red: degree=3 polynomial
- Green: degree=3 b-spline with knots at 150, 200, 250
- Blue: degree=2 b-spline with knots at 110, 150, 180, 200, 250

What to Know for the Next Quiz

- How regression fits into the supervised learning framework
- How concepts like training/testing, the bias-variance tradeoff, and feature expansion apply to regression
- A basic understanding of how certain types of feature expansions (discretization, polynomials, and splines) influence the flexibility of a regression
  - In particular, be familiar with the visual differences in the prediction lines for a model with a single predictor