ADEC 7430:
Big Data Econometrics

Linear and Nonlinear Regression

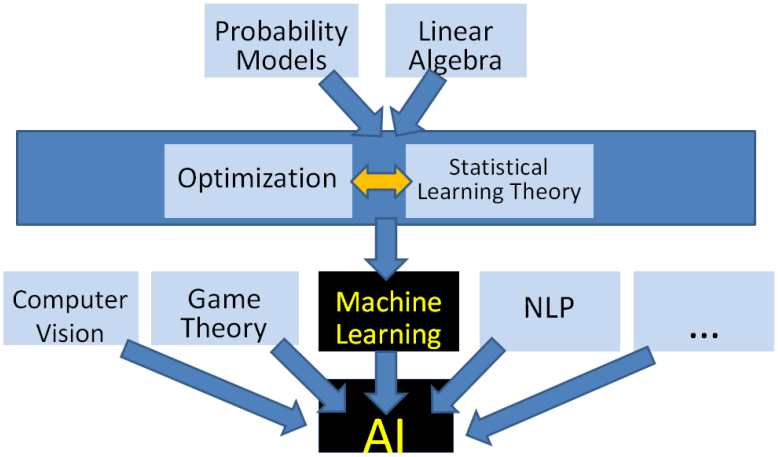
Dr. Nathan Bastian

Woods College of Advancing Studies

Boston College

Assignment

- **Reading:** Ch. 3, Ch. 7
- **Study:** Lecture Slides, Lecture Videos
- **Activity:** Quiz 2, R Lab 2, Discussion #2



References

- *An Introduction to Statistical Learning, with Applications in R* (2013), by G. James, D. Witten, T. Hastie, and R. Tibshirani.
- *The Elements of Statistical Learning* (2009), by T. Hastie, R. Tibshirani, and J. Friedman.
- *Learning from Data: A Short Course* (2012), by Y. Abu-Mostafa, M. Magdon-Ismail, and H. Lin.
- *Machine Learning: A Probabilistic Perspective* (2012), by K. Murphy

Lesson Goals:

- Recognize the basic concepts of expectation, variance, and parameter estimation.
- Recall the basic concepts of statistical decision theory.
- Describe simple and multiple linear regression as a supervised learning algorithm.
- Demonstrate ordinary least squares estimation for linear regression models.
- Explain the basic concepts of *k*-nearest neighbors regression.
- Modify simple OLS linear regression in a flexible way using polynomial regression, step functions, regression splines, smoothing splines, local regression, and generalized additive models.

Overview: Linear Regression

- Linear regression is a simple approach to supervised learning, as it assumes that the dependence of Y on $X_1, X_2, \ldots, X_p$ is linear.
- Most modern machine learning approaches can be seen as generalizations or extensions of linear regression.
- When augmented with kernels or other forms of basis function expansion (which replace X with some non-linear function of the inputs), it can also model non-linear relationships.
- Goal: predict Y from X by $f(X)$

Review: Expectation

- The expectation of a random variable is its “average” value under its distribution.
- The expectation of a random variable X, denoted E[X], is its Lebesgue integral with respect to its distribution.
- If X takes values in some countable numeric set χ, then

$$E(X) = \sum_{x \in \chi} xP(X = x)$$

Review: Expectation (cont.)

- If $X \in \mathbb{R}^m$ has a density p, then $E(X) = \int_{\mathbb{R}^m} xp(x)dx$
- Expectation is linear: $E(aX + b) = aE(X) + b$
- Also, $E(X + Y) = E(X) + E(Y)$
- Expectation is monotone: if $X \geq Y$, then $\mathrm{E}(X) \geq \mathrm{E}(Y)$

Review: Variance

- The variance of a random variable X is:

$$Var(X) = E[(X - E[X])^2] = E[X^2] - (E[X])^2$$

- The variance obeys the following $a, b \in \mathbb{R}$:

$$Var(aX + b) = a^2 Var(X)$$

Review: Frequentist Basics

- The training data $X_1,\ldots,X_n$ is generally assumed to be *independent and identically distributed (iid)*.

- We want to estimate some unknown value ϑ associated with the distribution from which the data was generated.

- In general, our estimate will be a function of the data:

$$\hat{\theta} = f(X_1, X_2, \ldots, X_n)$$

Review: Parameter Estimation

- In practice, we often seek to select a distribution (model) corresponding to our data.
- If the model is parameterized by some set of values, then this problem is that of parameter estimation.
- In general, we typically use maximum likelihood estimation (MLE) to obtain parameter estimates.

Review: Parameter Estimation (cont.)

- Given that the training data are *iid* and come from the probability density function *p*, to use MLE we first specify the joint density function (which is also the *likelihood* function):

$$\mathcal{L}(\boldsymbol{\theta}; X_1, \ldots, X_n) = \prod_{i=1}^{n} p_{\boldsymbol{\theta}}(x_i)$$

- In practice, it is more convenient to work with the logarithm of the likelihood function, called the *log-likelihood*:

$$\ell(\boldsymbol{\theta}) = \ln \mathcal{L}(\boldsymbol{\theta}; X_1, \ldots, X_n) = \sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(x_i)$$

Review: Parameter Estimation (cont.)

- Using the method of MLE: $\widehat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta})$

- Instead of maximizing the log-likelihood, we can equivalently minimize the *negative log-likelihood* (NLL):

$$NLL(\boldsymbol{\theta}) = -\ln \mathcal{L}(\boldsymbol{\theta}; X_1, \ldots, X_n) = -\sum_{i=1}^{n} \ln p_{\boldsymbol{\theta}}(x_i)$$

$$\widehat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}} NLL(\boldsymbol{\theta})$$

- This formulation is sometimes more convenient, since many optimization software packages are designed to find the minima of functions, rather than maxima.

Statistical Decision Theory

- Let $X \in \mathbb{R}^p$ denote a real valued random input vector.
- Let $Y \in \mathbb{R}$ denote a real valued random output variable, with joint distribution Pr(X, Y).
- We seek a function *f*(*X*) for predicting *Y* given values of the input *X*.
- *Loss function L*(*Y, f*(*X*)) → penalizing errors in prediction.

Statistical Decision Theory (cont.)

- *Squared error loss*: $L\big(Y, f(X)\big) = (Y - f(X))^2$

- This leads us to a criterion for choosing *f*,

$$\mathrm{EPE}(f) = E(Y - f(X))^2 = \int [y - f(x)]^2 \Pr(dx, dy)$$

which is the *expected (squared) prediction error*. By conditioning on *X*, we can write EPE as

$$\mathrm{EPE}(f) = E_X E_{Y|X}([Y - f(X)]^2 | X)$$

Statistical Decision Theory (cont.)

- This suffices to minimize EPE as follows:

$$f(x) = \text{argmin}_c E_{Y|X}([Y - c]^2 | X = x)$$

- The solution is $f(x) = E(Y|X = x)$, which is the conditional expectation, also known as the *regression* function.

- The best prediction of *Y* at any point *X* = *x* is the conditional mean, when best is measured by average squared error.

- A linear regression model assumes that the regression function E(*Y* | *X*) is linear in the inputs X_1, X_2,…,X_p.

Linear Regression Model

- Input vector: $X^T = (X_1, X_2, \ldots, X_p)$
- Output Y is real-valued (quantitative response) and ordered
- We want to predict Y from X.
- Before we actually do the prediction, we have to *train* the function $f(X)$.
- By the end of training, we have a function $f(X)$ to map every X into an estimated Y (aka $\hat{Y}$).

Linear Regression Model (cont.)

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$$

- This is a linear combination of the measurements that are used to make predictions, plus a constant.

- No matter the source of the X_j, the model is linear in the parameters.

- β_0 is the intercept and β_j is the slope for the *j*th variable X_j, which is the **average** increase in Y when X_j is increased by one unit and all other X's are held constant.

Assumptions of the Linear Regression Model

1. **Linearity:** The model specifies a linear relationship between the response variable *y* and the predictor variables ***x*** (i.e., linear in the parameters and the disturbance).

2. **Full column rank:** There is no exact linear relationship among any of the predictors.
   - This assumption is necessary for estimation of the parameters of the model (i.e., taking an inverse).

Assumptions of the Linear Regression Model (cont.)

3. **Exogeneity of the independent variables:** The expected value of the disturbance (error term) at observation *i* in the sample is NOT a function of the predictors observed at any observation.
   - $\mathrm{E}[\varepsilon_i|\mathbf{X}] = 0 \; \forall \; i = 1,2,\ldots,n$
   - The predictors will not carry useful information for prediction of the error terms (i.e., no correlation).

Assumptions of the Linear Regression Model (cont.)

4. **Homoscedasticity and nonautocorrelation:** Each error term has the SAME finite variance, $\mathrm{Var}[\varepsilon_i|\mathbf{X}] = \sigma^2 \; \forall \; i = 1,2,\ldots,n$, and is uncorrelated with every other error term, $\mathrm{Cov}\left[\varepsilon_i, \varepsilon_j \middle| \mathbf{X}\right] = 0 \; \forall \; i \neq j$

5. **Data generation:** The data may be any mixture of constants and random variables.

6. **Normal distribution:** The disturbances are normally distributed → $\boldsymbol{\varepsilon}|\mathbf{X} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$.

Ordinary Least Squares Estimation

- Typically we have a set of *training data* $(X_1,Y_1)\ldots(X_n,Y_n)$ from which to estimate the parameters β.

- Each X_i is a vector of feature measurements for the ith case.

- We can apply the method of MLE to the linear regression setting (using the definition of the Gaussian), where the log-likelihood function is given by:

$$\ell(\boldsymbol{\theta}) = \frac{-1}{2\sigma^2} RSS(\boldsymbol{\beta}) - \frac{n}{2} \ln 2\pi\sigma^2$$

OLS Estimation (cont.)

- Note that RSS stands for *residual sum of squares*:

$$RSS(\boldsymbol{\beta}) = \sum_{i=1}^{n}(Y_i - f(X_i))^2 = \sum_{i=1}^{n}(Y_i - \beta_0 - \sum_{j=1}^{p} X_{ij}\beta_j)^2$$

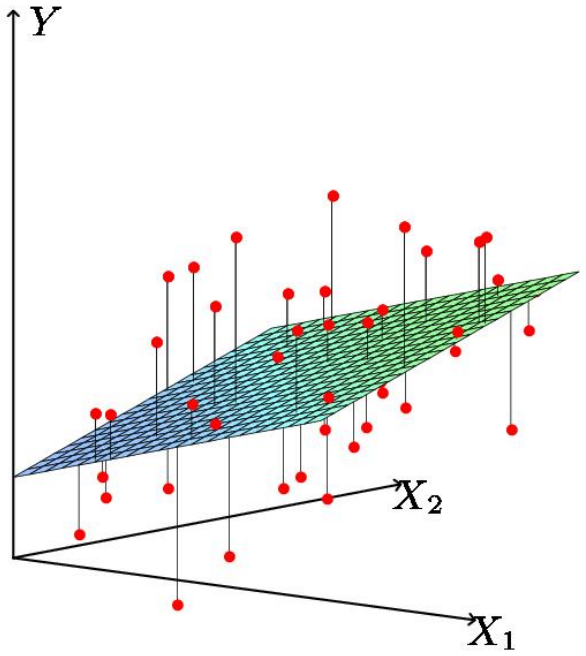
- The RSS is also called the *sum of squared errors* (SSE), where

$$MSE = \frac{SSE}{n} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

- We see that the MLE for $\boldsymbol{\beta}$ is the one that minimizes the RSS. Thus, we estimate the parameters using *ordinary least squares* (OLS), which is identical to the MLE, to choose $\hat{\beta}_0$ through $\hat{\beta}_p$ as to minimize the RSS.

OLS Estimation (cont.)

- We illustrate the geometry of OLS fitting, where we seek the linear function of *X* that minimizes the sum of squared residuals from *Y*.



- The predictor function corresponds to a plane (hyper plane) in the 3D space.
- For accurate prediction, hopefully the data will lie close to this hyper plane, but they won't lie exactly in the hyper plane.

OLS Estimation (cont.)

For the *simple* and *multiple* linear regression model:

- Let $\boldsymbol{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_n \end{bmatrix}$ be the *n*-vector of outputs in the training set.
- Let $\boldsymbol{X} = \begin{bmatrix} 1 & X_{11} & \dots & X_{1p} \\ 1 & X_{21} & \dots & X_{2p} \\ 1 & \dots & \dots & \dots \\ 1 & X_{n1} & \dots & X_{np} \end{bmatrix}$ be the *n* x (*p* + 1) matrix of inputs.

where there are *n* observations and *p* predictors in the training data.

OLS Estimation (cont.)

- Let $\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \dots \\ \beta_p \end{bmatrix}$, so the $RSS(\boldsymbol{\beta}) = (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T \ (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})$

- So, we must solve the following quadratic minimization problem:

$$\widehat{\boldsymbol{\beta}} = \text{argmin}_{\boldsymbol{\beta}} RSS(\boldsymbol{\beta})$$

- This minimization problem has a unique solution, provided that $\boldsymbol{X}$ has full column rank (i.e. the p columns of $\boldsymbol{X}$ are linearly independent), given by solving the normal equations:

$$(\boldsymbol{X}^T\boldsymbol{X})\boldsymbol{\beta} = \boldsymbol{X}^T\boldsymbol{Y}$$

OLS Estimation (cont.)

The unique solution to the normal equations yields the vector $\widehat{\boldsymbol{\beta}}$ (i.e. the OLS estimates of the parameters): $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$

Note for *simple* OLS linear regression (intercept and one predictor):

$$\text{Let } \boldsymbol{X}^T\boldsymbol{X} = \begin{bmatrix} n & \sum X_i \\ \sum X_i & \sum X_i^2 \end{bmatrix}$$

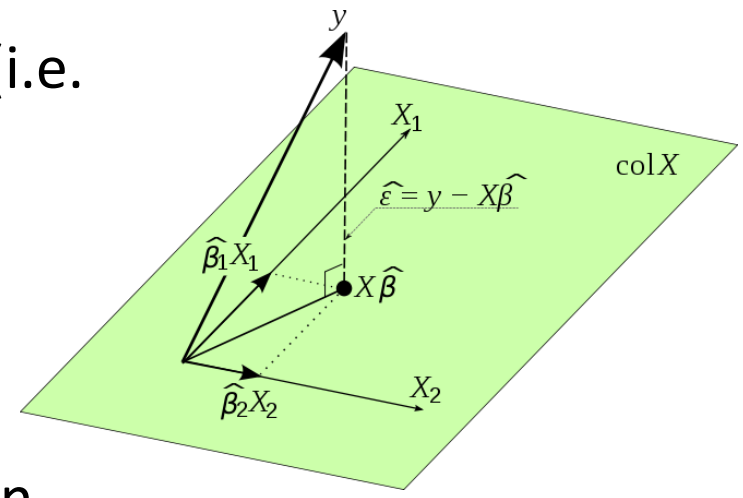
$$\text{Let } \boldsymbol{X}^T\boldsymbol{Y} = \begin{bmatrix} \sum Y_i \\ \sum X_i Y_i \end{bmatrix}$$

$$\text{Let } (\boldsymbol{X}^T\boldsymbol{X})^{-1} = \frac{1}{\det(\boldsymbol{X}^T\boldsymbol{X})} \begin{bmatrix} \sum X_i^2 & -\sum X_i \\ -\sum X_i & n \end{bmatrix}$$

OLS Estimation (cont.)

- The fitted values at the training inputs (i.e. vector of the OLS predictions) are:

$$\widehat{\boldsymbol{Y}} = \boldsymbol{X}\widehat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$



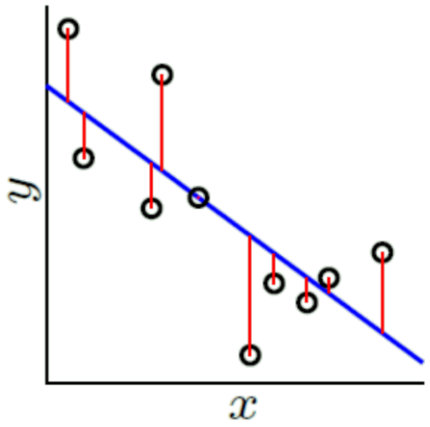
- In geometric representation, this corresponds to an orthogonal projection of ***Y*** onto the column space of ***X***.

- The matrix $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T$ is the projection matrix, which is called the *hat* matrix because it puts a hat on ***Y***.

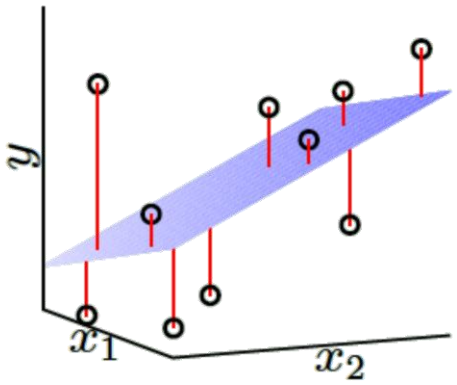
OLS Estimation (cont.)

OLS Linear Regression Algorithm:

1. From the training data set, construct the input matrix $\boldsymbol{X}$ and the output vector $\boldsymbol{Y}$



2. Assuming $\boldsymbol{X}^T\boldsymbol{X}$ is invertible (positive definite and non-singular), compute $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$



3. Return $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$

OLS Estimation (cont.)

If the linear model is true (i.e. if the conditional expectation of *Y* given *X* indeed is a linear function of the X_j's), and *Y* is the sum of that linear function and an independent Gaussian noise, then we have the following properties for OLS estimation:

1. The OLS estimation of *β* is unbiased → $E[\hat{\beta}_j] = \beta_j \; \forall \; j = 0,1, \ldots, p$

2. To draw inferences about *β*, further assume:
   $Y = E(Y|X) + \varepsilon$, where $\varepsilon \sim N(0, \sigma^2)$ and independent of X.

OLS Estimation (cont.)

- The estimation accuracy (variance) of $\widehat{\boldsymbol{\beta}}$ is: $Var(\widehat{\boldsymbol{\beta}}) = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\sigma^2$

- Typically one estimates the variance σ^2 (unbiased) by

$$\hat{\sigma}^2 = \frac{1}{n-p-1}\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2$$

- When σ^2 is higher, the variance of $\widehat{\boldsymbol{\beta}}$ is higher. Also, the variance-covariance matrix tells us the variance for every individual beta and also the covariance for any pair of betas.

- **Gauss-Markov Theorem**: The OLS estimates of the parameters $\boldsymbol{\beta}$ have the smallest variance (i.e. smallest mean squared error) among all linear unbiased estimates.

Population vs. OLS Lines

- Population Regression Line: $Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \varepsilon$

- OLS Regression Line: $\hat{Y}_i = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j X_j$

- We would like to know the population parameters, but we only know the OLS estimates $\hat{\beta}_0$ through $\hat{\beta}_p$.

- Further, we use $\hat{Y}_i$ as an estimate for Y_i.

Accuracy of Coefficient Estimates

- Let's consider a simple linear regression model with $\hat{\beta}_0$ and $\hat{\beta}_1$. How close are $\hat{\beta}_0$ and $\hat{\beta}_1$ to the true values β_0 and β_1, respectively?

- We can answer this by computing the standard errors associated with $\hat{\beta}_0$ and $\hat{\beta}_1$ (see Equation 3.8 in text).

- These SEs can be used to compute confidence intervals (CIs), prediction intervals (PIs), and perform hypothesis tests on the coefficients.

Accuracy of the Model: RSE

- The residual standard error (RSE) is an estimate of the standard deviation of ε.

- In other words, RSE is the average amount that the response will deviate from the true regression line:

$$RSE = \sqrt{\frac{RSS}{n - p - 1}}$$

where *p* is the number of predictors (slopes) in the regression model (not including the intercept).

Accuracy of the Model: R^2

- The proportion of variability in *Y* that can be explained using *X:*

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

where total sum of squares (TSS) measures the total variance in the response *Y*. It is thought of as the amount of variability inherent in the response before the regression is performed.

- Note that RSS measures the amount of variability that is left unexplained after performing the regression.

- Always between 0 (no fit) and 1 (perfect fit).

Two Key Questions

1. Is $\beta_j = 0$ or not?

2. Is at least one of the predictors useful in predicting the response?

Is $\beta_j = 0$ or not?

- H_0: There is no relationship between X_j and Y ($\beta_j = 0$).
- H_a: There is some relationship between X_j and Y ($\beta_j \neq 0$).

- Compute the *t-statistic*: $t = \frac{\hat{\beta}_j - 0}{SE(\hat{\beta}_j)}$

- If *t* is large (and the *p-value* is small, typically < α = 0.05), then we reject H_0 and declare that there is relationship.

- We use the Regression Output table to get the beta coefficients, standard errors, t-statistics and p-values.

Are all regression coefficients 0?

- H_0: all slopes equal 0 ($\beta_1 = \beta_2 = \cdots = \beta_p = 0$).
- H_a: at least one slope $\neq 0$.

- Compute the *F-statistic:* $F = \frac{(TSS-RSS)/p}{RSS/(n-p-1)} \sim F_{p,n-p-1}$

- We use the ANOVA table to get the *F-statistic* and its corresponding p-value.

- If *p-value* < 0.05, reject H_0. Otherwise, all of the slopes equal 0 and none of the predictors are useful in predicting the response.

Deciding on Important Variables

- *Best Subset Selection*: we compute the OLS fit for all possible subsets of predictors and then choose between them based on some criterion that balances training error with model size.

- There are 2^p possible models, so can't examine them all.

- We use an automated approach that searches through a subset of all the models.
  - Forward Selection
  - Backward Selection

Overview: Forward Selection

- We begin with the *null model*, a model containing an intercept but no predictors.
- We fit *p* simple linear regressions and add to the null model that variable resulting in the lowest RSS.
- We add to that model the variable that results in the lowest RSS amongst all two-variable models.
- The algorithm continues until some stopping rule is satisfied (i.e. all remaining variables have a *p-value* greater than some threshold).

Overview: Backward Selection

- We begin with all variables in the model.
- We remove the variable with the largest *p-value* (i.e. least statistically significant).
- The new (p – 1)-variable model is fit, and the variable with the largest *p-value* is removed.
- The algorithm continues until a stopping rule is reached.

Qualitative Predictors

- Some predictors are not quantitative but are *qualitative*, taking a discrete set of values.
- These are known as categorical variables, which we can code as indicator variables (dummy variables).
- Examples: gender, student status, marital status, ethnicity

Qualitative Predictors (cont.)

- When a qualitative predictor has more than two levels, a single dummy variable cannot represent all possible variables.

- Thus, there will always be one few dummy variables than the number of levels in the factor.
  - Factor = Ethnicity
  - Levels = Asian, Caucasian, African American
  - # of Dummy Variables = 3 – 1 = 2

- The level with no dummy variable is the *baseline*.

Qualitative Predictors (cont.)

- Suppose we want to regress the quantitative response variable Y (such as balance) on both a quantitative variable (such as income) and a qualitative variable (such as gender).
- There are two levels of gender: $Gender_i = \begin{cases} 1 \text{ if female} \\ 0 \text{ if male} \end{cases}$
- The regression model (without interaction) is:

$$Balance_i \approx \beta_0 + \beta_1 Income_i + \beta_2 Gender_i$$
$$= \begin{cases} \beta_0 + \beta_1 Income_i + \beta_2 & \text{if female} \\ \beta_0 + \beta_1 Income_i & \text{if male} \end{cases}$$

- β_2 is the average extra balance that females have for a given income level; note that males are *baseline* (coded as 0).

Qualitative Predictors (cont.)

- There are different ways to code categorical variables.
- There are two levels of gender: $Gender_i = \begin{cases} 1 \text{ if female} \\ -1 \text{ if male} \end{cases}$
- The regression model (without interaction) is:

$$Balance_i \approx \beta_0 + \beta_1 Income_i + \beta_2 Gender_i$$
$$= \begin{cases} \beta_0 + \beta_1 Income_i + \beta_2 & \text{if female} \\ \beta_0 + \beta_1 Income_i - \beta_2 & \text{if male} \end{cases}$$

- β_2 is the average amount that females are above the average, for any given income level; note that males are *baseline* again.

Extensions of the Linear Model

- Allow for *interaction effects.* Note that if interaction is included in the model, all of the *main effects* should be include as well (even if not statistically significant).

$$\begin{aligned} \texttt{sales} &= \beta_0 + \beta_1 \times \texttt{TV} + \beta_2 \times \texttt{radio} + \beta_3 \times (\texttt{radio} \times \texttt{TV}) + \epsilon \\ &= \beta_0 + (\beta_1 + \beta_3 \times \texttt{radio}) \times \texttt{TV} + \beta_2 \times \texttt{radio} + \epsilon. \end{aligned}$$

- Accommodate nonlinear relationships using *polynomial regression* or other nonlinear regression methods. For example, you can include transformed versions of the predictors in the model.

$$\texttt{mpg} = \beta_0 + \beta_1 \times \texttt{horsepower} + \beta_2 \times \texttt{horsepower}^2 + \epsilon$$

Potential Problems

- There are several potential problems that may occur when fitting a linear regression model.

1. Non-linearity of the response-predictor relationships.
2. Correlation of residuals.
3. Non-normality and non-constant variance of the residuals.
4. Outliers (refer to Section 3.3.3 in text).
5. High-leverage points (refer to Section 3.3.3 in text).
6. Collinearity (refer to Section 3.3.3 in text).

Non-linearity of the Data

- The linear regression model assumes that there is a straight-line relationship between predictors and the response.
- If the true relationship is non-linear then conclusions are suspect.
- Examine the *residual plots*, as strong patterns (U-shape) in the residuals indicate non-linearity in the data.
- If there are non-linear associations in the data, then use non-linear transformations of the predictors (e.g. log X).

Correlation of Residuals

- An important assumption of the linear regression model is that the residuals are uncorrelated.

- If there is correlation among the residuals (Durbin-Watson test), then the estimated standard errors will tend to *underestimate* the true standard errors – this makes the CIs and PIs narrower than they should be.

- These correlations frequently occur in the context of *time series* data, so consider employing *time series analysis* methods (such ARIMA, etc.).

Non-normality and Non-constant Variance of Residuals

- Another important assumption of the linear regression model is that the residuals are normally distributed and have constant variance across all levels of X.

- If the residuals are not normally distributed (Anderson-Darling test), you can perform a Box-Cox transformation on the response Y.

- If there is heteroscedasticity (Breusch-Pagan, Modified Levene, or Special White's tests), then you can consider transforming the response Y. If this doesn't fix the problem, consider computing *robust standard errors* or conduct *weighted least squares regression*.

Nonparametric Regression

- The nonparametric regression approach is to choose a function f from some smooth family of functions.

- We do need to make some assumptions about f – that it has some degree of smoothness and continuity. However, these restrictions are far less limiting than the parametric way.

- Unlike parametric models, nonparametric models do not have a formulaic way of describing the relationship between the predictors and the response; this often needs to be done graphically.

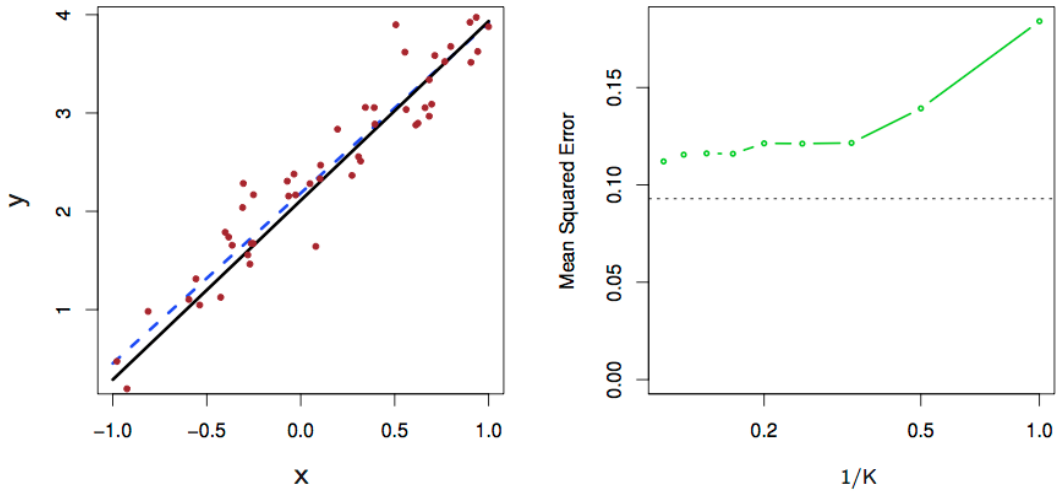
- However, the nonparametric approach is more flexible, assumes far less (and so is less liable to make bad mistakes), and is particularly useful when little past experience is available to know the appropriate form for a parametric model.

KNN Regression

- *K-nearest neighbors (KNN) regression* is a nonparametric, flexible approach for performing regression.
- It is closely related to the KNN classifier discussed last week.
- To predict Y for a given value of X, consider the *k* closest points to X in the training data and take the average of the responses.
- If *k* is small, then KNN is more flexible than linear regression; it will have low bias but high variance.

KNN Regression (cont.)

- The parametric approach (e.g. linear regression) will outperform the nonparametric approach (e.g. KNN regression) if the parametric form that has been selected is close to the true form of *f*.



- Here, linear regression achieves a lower test MSE than does KNN regression, since *f(X)* is in fact linear.

Moving Beyond Linearity

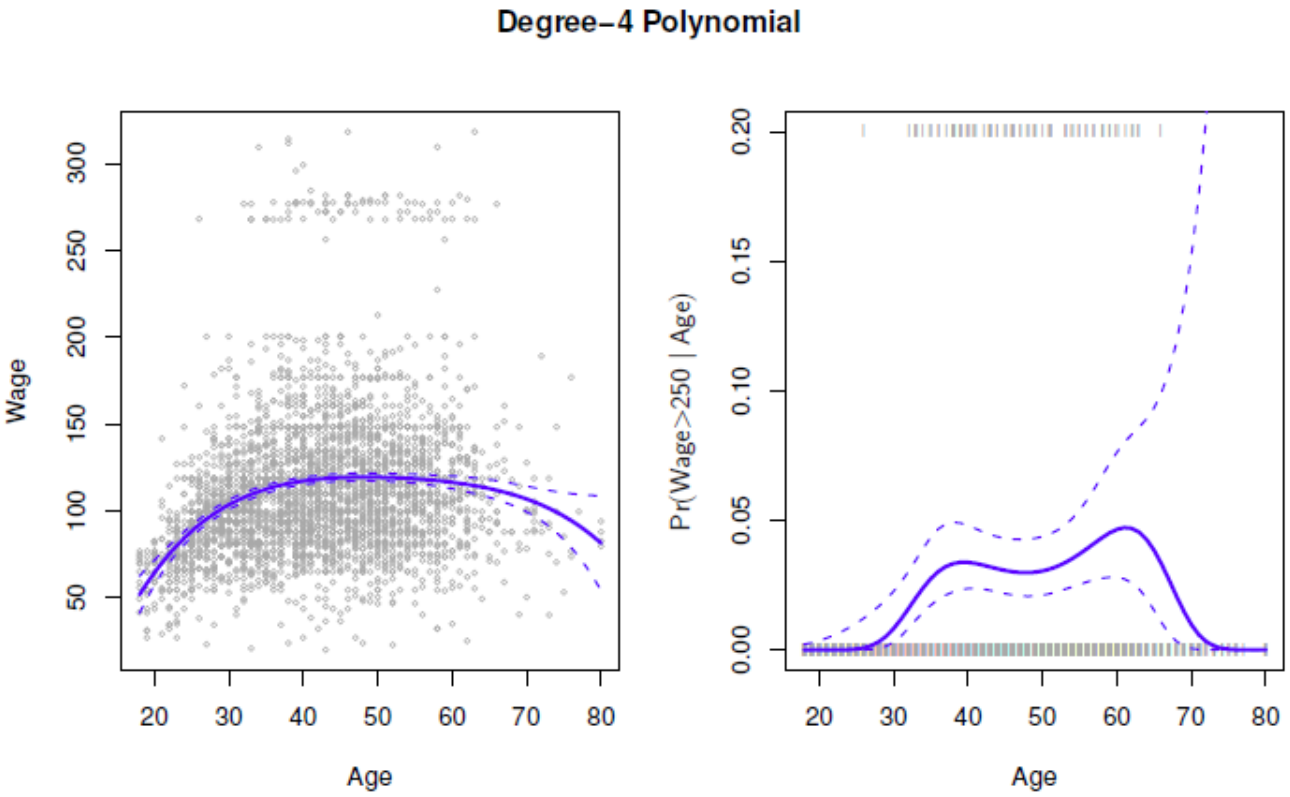
- The linearity assumption is good in many machine learning problems.

- However, there are other nonlinear regression methods that offer a lot of flexibility, without losing the ease and interpretability of linear models:
  - Polynomial regression
  - Step functions
  - Regression splines
  - Smoothing splines
  - Local regression
  - Generalized additive models (GAMs)

Polynomial Regression

- Replace the standard linear model with a polynomial function:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i$$



Polynomial Regression (cont.)

- For large enough degree *d*, a polynomial regression allows us to produce an extremely non-linear curve.

- We do this by creating new variables $X_1 = X$, $X_2 = X^2$, etc. and then treat as multiple OLS linear regression.

- In general, we are not really interested in the coefficients, but instead the fitted function values at any value x_0:

$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0 + \hat{\beta}_2 x_0^2 + \hat{\beta}_3 x_0^3 + \hat{\beta}_4 x_0^4$$

Polynomial Regression (cont.)

- It is unusual to use *d* greater than 3 or 4 because for large values of *d*, the polynomial curve can become overly flexible and can take on very strange shapes.

- Note that we can also use cross-validation to choose *d*.

Step Functions

- Using polynomial functions of the features as predictors in a linear model imposes a *global* structure on the non-linear function of X.
- To avoid imposing such a global structure, we can create transformations of a variable by cutting the variable into distinct regions.
- In particular, we use *step functions* to break the range of *X* into bins, where we fit a different constant in each bin.

Step Functions (cont.)

- This amounts to converting a continuous variable into an *ordered categorical variable*.

- In greater detail, we create cutpoints (or knots) c_1, c_2,….,c_K in the range of X and then construct $K + 1$ new variables:

$$\begin{aligned} C_0(X) &= I(X < c_1), \\ C_1(X) &= I(c_1 \leq X < c_2), \\ C_2(X) &= I(c_2 \leq X < c_3), \\ &\vdots \\ C_{K-1}(X) &= I(c_{K-1} \leq X < c_K), \\ C_K(X) &= I(c_K \leq X), \end{aligned}$$

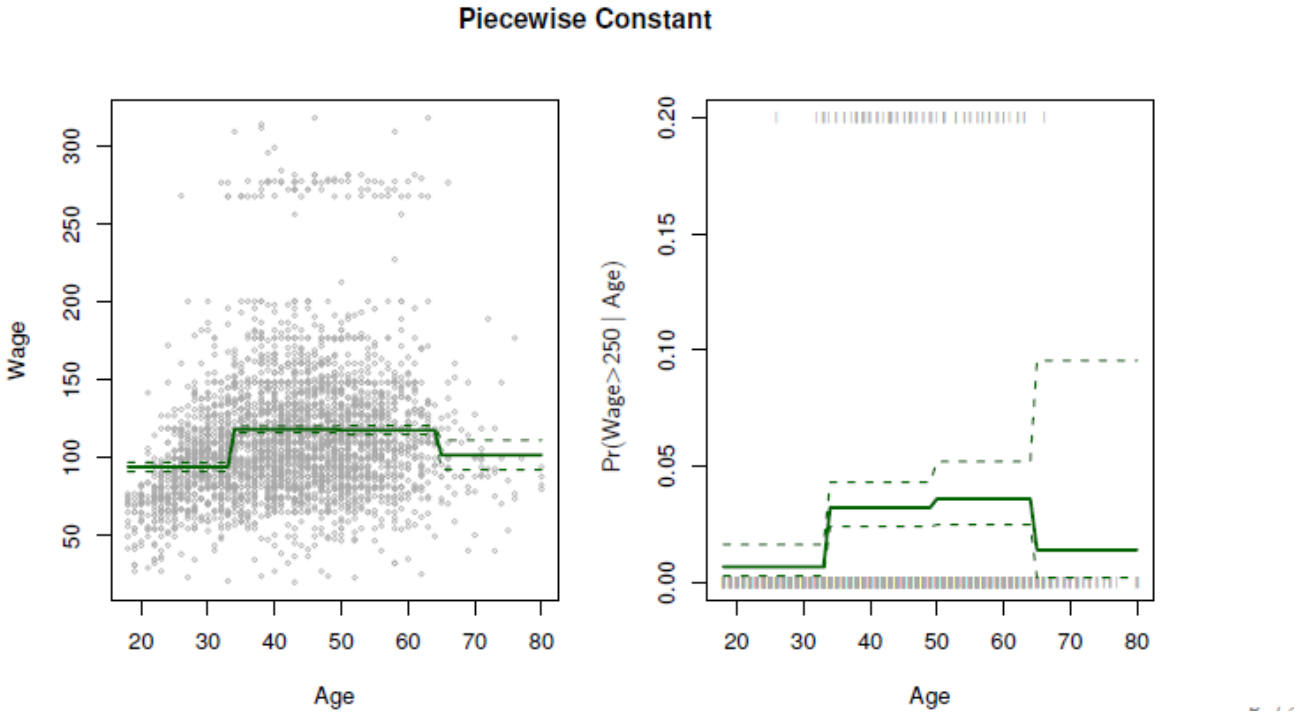
where $I(.)$ is an *indicator function* that returns a 1 if the condition is true and 0 otherwise.

Step Functions (cont.)

- Note that for any value of X, $C_0(X) + C_1(X) + \ldots + C_K(X) = 1$, since X must be exactly in one of the $K + 1$ intervals.
- We then use OLS estimation to fit a linear model using these $K + 1$ new variables:
- For a given value of X, at most one of $C_1, C_2, \ldots, C_K$ can be non-zero.
- β_j represents the average increase in the response for X in $c_j \leq X \leq c_{j+1}$ relative to $X < c_1$.

Step Functions (cont.)

$$C_1(X) = I(X < 35), \quad C_2(X) = I(35 \leq X < 50), \ldots, C_3(X) = I(X \geq 65)$$



- Unless there are natural breakpoints in the predictors, piece-wise constant functions can miss the action.

Basis Functions

- Polynomial and piece-wise constant regression models are special cases of a *basis function* approach.
- The idea is to have at hand a family of functions or transformations that can be applied to a variable X: $b_1(X),\ldots,b_K(X)$
- Instead of fitting a linear model in X, we fit the following model:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + \ldots + \beta_K b_K(x_i) + \epsilon_i$$

- Note that the basis functions $b_1(.),\ldots,b_K(.)$ are fixed and known.

Basis Functions (cont.)

- For polynomial regression, the basis functions are $b_j(x_i) = x_i^j$
- For piece-wise constant functions, the basis functions are
$$b_j(x_i) = I(c_j \leq x_i \leq c_{j+1})$$
- Note that we can use OLS to estimate the unknown regression coefficients.
- Thus, all of the inference tools for linear models (standard errors, F-statistics, etc.) are available in this setting.

Regression Splines

- Regression splines are a flexible class of basis functions that extend upon the polynomial regressions and piece-wise constant regression approaches.
- They involve dividing the range of *X* into *K* distinct regions; within each region, a polynomial function is fit to the data.
- These polynomials are constrained so that they join *smoothly* at the region boundaries (or *knots*).
- Provided that the interval is divided into enough regions, this can produce an extremely flexible it.

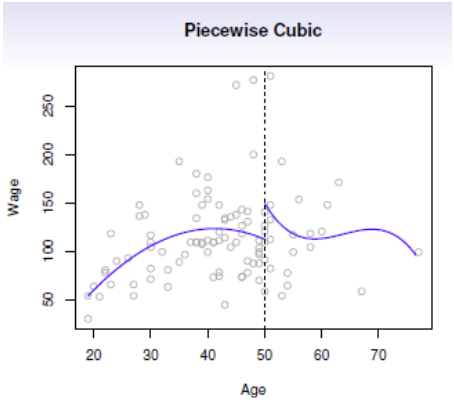
Piecewise Polynomials

- Instead of fitting a high-degree polynomial over the entire range of *X*, *piece-wise polynomial regression* involves fitting separate low-degree polynomials over different regions of *X*.

- Here, the beta coefficients differ in different parts of the range of *X*; the points where the coefficients change are called *knots*.

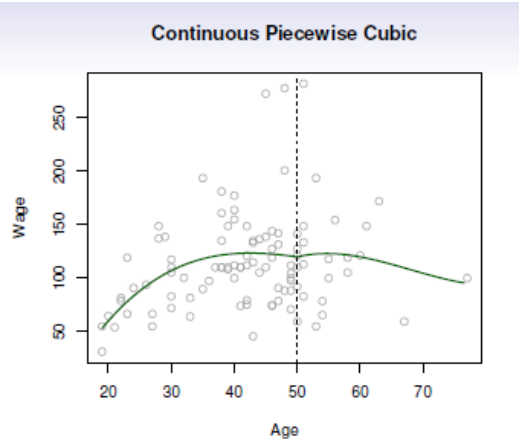
- Example: A piecewise cubic polynomial with a single knot at a point *c* takes the following form:

$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \epsilon_i & \text{if } x_i < c; \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \epsilon_i & \text{if } x_i \geq c. \end{cases}$$



Piecewise Polynomials (cont.)

- Each of the polynomial functions can be fit using OLS applied to simple functions of the original predictor.
- Using more knots leads to a more flexible piecewise polynomial.
- If general, if we place *K* different knots through the range of *X*, then we end up fitting *K* + 1 different polynomials.
- It is better to add *constraints* to the polynomials (e.g., continuity). *Splines* have the maximum amount of continuity.

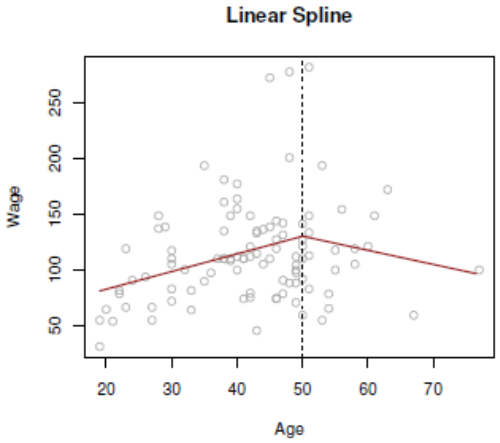


Regression Splines (cont.)

- Each constraint that we impose effectively frees up one degree of freedom, by reducing the complexity of the resulting piecewise polynomial fit.

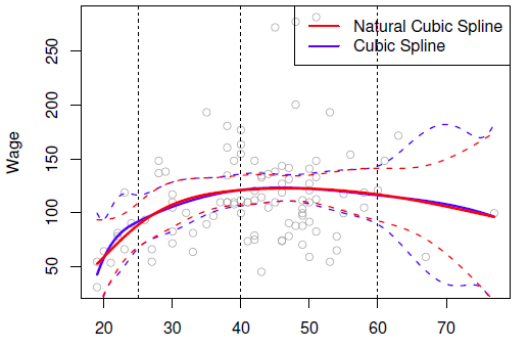
- The general definition of a degree-*d* spline is that it is a piecewise degree-*d* polynomial, with continuity in derivatives up to degree *d* – 1 at each knot.

- Thus, a **linear spline** is obtained by fitting a line in each region of the predictor space defined by the knots, requiring continuity at each knot.



Regression Splines (cont.)

- A *natural spline* is a regression spline with additional boundary constraints.
- The function is required to be linear at the boundary (in the region where *X* is smaller than the smallest knot, or larger than the largest knot).
- This additional constraint means that natural splines generally produce more stable estimates at the boundaries.
- A *natural cubic spline* extrapolates linearly beyond the knots.



Regression Splines (cont.)

Choosing the Location of Knots

- The regression spline is most flexible in regions that contain a lot of knots, because in those regions the polynomial coefficients can change rapidly.

- One option is to place more knots in places where we feel the function might vary most rapidly, and to place fewer knots where it seems more stable.

- In practice, it is common to place knots in a uniform fashion. For example, one strategy is to decide *K*, the number of knots, and then place them at appropriate quantiles of the observed *X*.

Regression Splines (cont.)

Choosing the Number of Knots

- One option is to try out different numbers of knots and see which produces the best looking curve.

- However, a more objective approach is to use cross-validation.

- The procedure is repeated for different number of knots *K*; then the value of *K* giving the smallest RSS is chosen.

- Splines allow us to place more knots, and hence flexibility, over regions where the function *f* seems to be changing rapidly, and fewer knots where *f* appears more stable.

Smoothing Splines

- We create regression splines by specifying a set of knots, producing a sequence of basis functions, and then use OLS to estimate the spline coefficients.

- What we really want is a function *g* that makes RSS small and *smooth*. Thus, consider the following criterion for fitting a smooth function *g*(*x*) to some data (known as a *smoothing spline*):

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

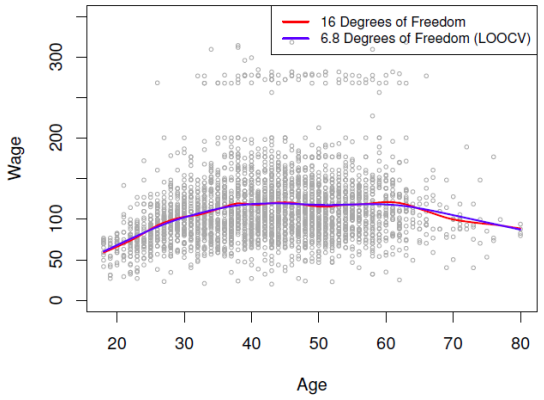
where λ is a nonnegative tuning parameter.

Smoothing Splines (cont.)

- The first term is a loss function (RSS), which tries to make $g(x)$ match the data at each x_i.
- The second term is a *roughness penalty* and controls how wiggly $g(x)$ is; this is modulated by the tuning parameter λ.
- The larger the value of λ, the smoother g will be. The smaller the value of λ, the more wiggly the function.
- As $\lambda \rightarrow \infty$, the function $g(x)$ becomes linear.

Smoothing Splines (cont.)

- It turns out that the solution is a natural cubic spline, with a knot at every unique value of x_i.
- The tuning parameter λ controls the level of roughness (i.e., the effective degrees of freedom).
- Smoothing splines avoid the knot-selection issue, leaving a single λ to be chosen.
- We use LOOCV to find λ!!



Local Regression

- *Local regression* is a different approach for fitting flexible non-linear functions, which involves computing the fit at a target point x_0 using only the nearby training observations.

- It is a *memory-based* procedure because we need all the training data each time we wish to compute a prediction.

- The *span* plays a role like that of the tuning parameter λ in smoothing splines; it controls the flexibility of the non-linear fit.

Local Regression (cont.)

- The smaller the value of the span *s*, the more *local* and wiggly will be our fit.
- A very large value of *s* will lead to a global fit to the data using all of the training observations.
- We can use cross-validation to choose *s* or specify it directly.
- Another choice to be made includes how to define the weighting function *K*, and whether to fit a linear, constant, or quadratic regression.

Local Regression (cont.)

Algorithm 7.1 *Local Regression At* $X = x_0$

1. Gather the fraction $s = k/n$ of training points whose x_i are closest to x_0.

2. Assign a weight $K_{i0} = K(x_i, x_0)$ to each point in this neighborhood, so that the point furthest from x_0 has weight zero, and the closest has the highest weight. All but these k nearest neighbors get weight zero.

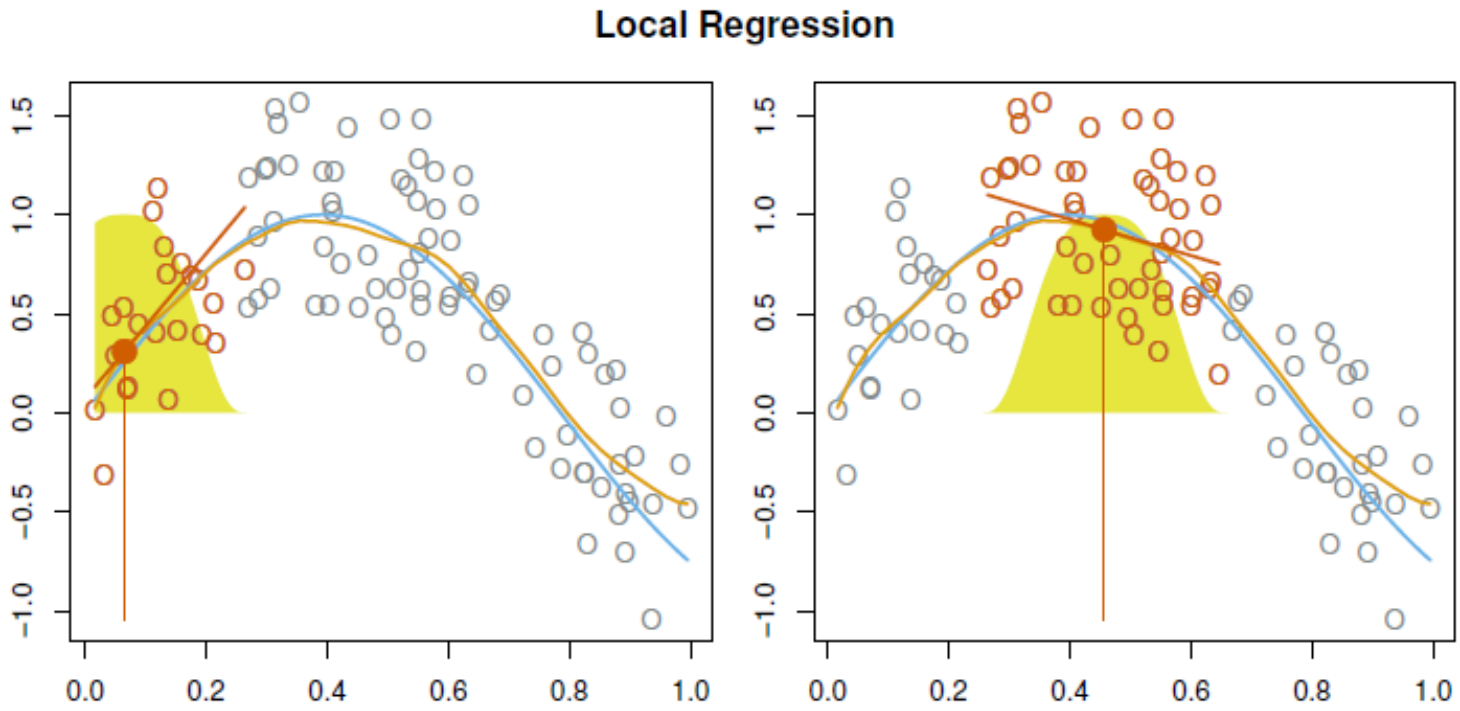
3. Fit a *weighted least squares regression* of the y_i on the x_i using the aforementioned weights, by finding $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize

$$\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2. \tag{7.14}$$

4. The fitted value at x_0 is given by $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$.

Local Regression (cont.)

- With a sliding weight function, we fit separate linear fits over the range of *X* by weighted least squares.



Generalized Additive Models

- *Generalized additive models* (GAMs) allow for flexible nonlinearities in several variables, but retains the additive structure of linear models.

- GAMs can be applied with both quantitative and qualitative responses.

- In particular, we replace each linear component of the multiple linear regression model with a (smooth) non-linear function.

Generalized Additive Models (cont.)

$$
\begin{aligned}
y_i &= \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i \\
&= \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i.
\end{aligned}
$$

- It is called an *additive* model because we calculate a separate f_j for each X_j, and then add together all of their contributions.

- We can use any of the previously discussed methods (smoothing splines, local regression, polynomial regression, etc.) as building blocks for fitting an additive model.

Generalized Additive Models (cont.)

- GAMs allow us to fit a non-linear f_j to each X_j, so that we can automatically model non-linear relationships that standard linear regression will miss.

- This means that we do not need to manually try out many different transformations on each variable individually.

- The non-linear fits can potentially make more accurate predictions for the response Y.

Generalized Additive Models (cont.)

- Because the model is additive, we can still examine the effect of each X_j on Y individually while holding all of the other variables fixed.

- Thus, if we are interested in inference, GAMs provide a useful representation.

- The smoothness of the function f_j for the variable X_j can be summarized via degrees of freedom.

Generalized Additive Models (cont.)

- The main limitation of GAMs is that the model is restricted to be additive.

- With many variables, important interactions can be missed. However, we can manually add interaction terms to the GAM model by including additional predictors of the form X_j x X_k.

- Although we have not yet covered classification problems, note that GAMs can also be used in situations where Y is qualitative.

Summary

- Review of expectation, variance, and parameter estimation.
- Basic concepts of statistical decision theory.
- Simple and multiple linear regression as a supervised learning algorithm.
- Ordinary least squares estimation for linear regression models.
- Basic concepts of *k*-nearest neighbors regression.
- Polynomial regression, step functions, regression splines, smoothing splines, local regression, and generalized additive models.