Machine Learning for OR & FE

Supervised Learning: Regression I

Martin Haugh

Department of Industrial Engineering and Operations Research
Columbia University
Email: martin.b.haugh@gmail.com

Some of the figures in this presentation are taken from "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 (JWHT).

Additional References: Chapter 3 of JWHT and *Learning from Data* by Abu-Mostafa, Lin and Magdon-Ismail.

Outline

Linear Regression
- Applications from ISLR
- Potential Problems with Linear Regression

Linear Regression with Basis Functions

Bias-Variance Decomposition
- A Case Study: Overfitting with Polynomials

Linear Regression

- Linear regression assumes the regression function $\mathsf{E}[Y|\mathbf{X}]$ is linear in the inputs, $X_1, \ldots, X_p$.
- Developed many years ago but still very useful today
  - simple and easy to understand
  - can sometimes outperform more sophisticated models when there is little data available.
- Linear models can also be applied to transformations of the inputs
  - leads to the basis function approach (and kernel regression)
  - which extends the scope of linear models to non-linear models.
- But linear models also have many weaknesses including a tendency to over-fit the data
  - will return to this later when we discuss the bias-variance decomposition and (in a later set of slides) shrinkage methods.

Linear Regression

In a linear regression model the dependent variable Y is a random variable that satisfies

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \epsilon$$

where $\mathbf{X} = (X_1, \ldots, X_p)$ and ϵ is the "error" term.

The linear model therefore implicitly assumes that $\mathbb{E}[Y \mid \mathbf{X}]$ is approximately linear in $\mathbf{X} = (X_1, \ldots, X_p)$.

The input or independent variables, X_i, are numerical inputs

- or possibly transformations, e.g. product, log, square root, $\phi(x)$, of "original" numeric inputs
- the ability to transform provides considerable flexibility.

The X_i's can also be used as "dummy" variables that encode the levels of qualitative inputs

- an input with K levels would require $K - 1$ dummy variables, $X_1, \ldots, X_{K-1}$

Model Fitting: Minimizing the Residual Sum of Squares

We are given training data: $(y_1, \mathbf{x}_1)$, $(y_2, \mathbf{x}_2)$, ..., $(y_N, \mathbf{x}_N)$.

Then obtain $\hat{\beta}$ by minimizing the residual sum of squares or RSS:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 = \min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

where

$$\mathbf{y} := \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \quad \mathbf{X} := \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \dots & x_{Np} \end{bmatrix}$$

Model Fitting: Minimizing the Residual Sum of Squares

This is a simple (convex) quadratic optimization problem so $\hat{\beta}$ satisfies

$$\nabla \|\mathbf{y} - \mathbf{X}\beta\|^2 = -2\mathbf{X}^\top(\mathbf{y} - \mathbf{X}\beta) = \mathbf{0}\big|_{\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}} \quad \Rightarrow \quad \hat{\beta} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$$

Also have

$$\hat{\mathbf{y}} = \underbrace{\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top}_{:=\ \mathbf{H},\ \text{the ``hat'' matrix}} \mathbf{y} \qquad \text{and} \qquad \hat{\boldsymbol{\epsilon}} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I} - \mathbf{H})\mathbf{y}.$$

Have (implicitly) assumed that $(\mathbf{X}^\top\mathbf{X})$ is non-singular.

This is not always the case in which case $\hat{\beta}$ will not be unique (although $\hat{\mathbf{y}}$ still is)

- can resolve by dropping redundant columns from $\mathbf{X}$
  - many software packages do this automatically

But in many modern applications $p >> N$ in which case at least $N - p$ columns would need to be dropped – something we may not want to do!

- hence the need for another solution approach e.g. ridge regression.

For now we will assume $p \leq N$.

Model Fitting: Minimizing the Residual Sum of Squares

The residual-sum-of-squares is defined as

$$\mathsf{RSS} := \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 = \hat{\boldsymbol{\epsilon}}^\top \hat{\boldsymbol{\epsilon}}$$

whereas the the total-sum-of-squares is

$$\mathsf{TSS} := \sum_{i=1}^{N} (y_i - \bar{y})^2.$$

The R^2 statistic is a measure of the linear relationship between **X** and Y:

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

R^2 always lies in the interval $[0, 1]$ with values closer to 1 being "better"

- but whether a given R^2 value is good or not depends on the application
- in physical science applications we looks for values close to 1 (if the model is truly linear); in social sciences an R^2 of .1 might be deemed good!

The Geometry of Least Squares

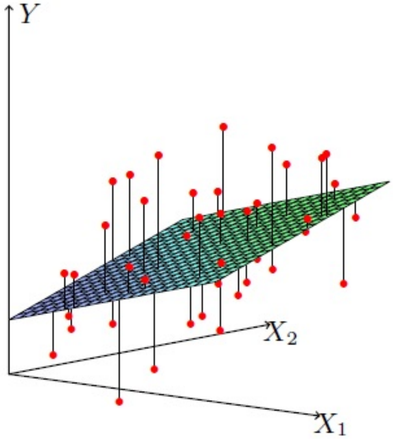


Figure 3.1 from HTF: Linear least squares fitting with $X \in \mathbb{R}^2$. We seek the linear function of X that minimizes the sum of squared residuals from Y.

The Geometry of Least Squares

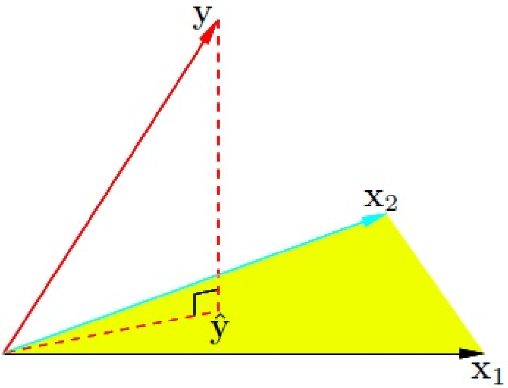


Figure 3.2 from HTF: The N-dimensional geometry of least squares regression with two predictors. The outcome vector $\mathbf{y}$ is orthogonally projected onto the hyperplane spanned by the input vectors $\mathbf{x}_1$ and $\mathbf{x}_2$. The projection $\hat{\mathbf{y}}$ represents the vector of the least squares predictions.

Normal Errors: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$

If $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ then $\hat{\boldsymbol{\beta}}$ satisfies

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} = \boldsymbol{\beta} + (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\beta}, (\mathbf{X}^\top \mathbf{X})^{-1} \sigma^2).$$

Can estimate σ^2 with the sample variance:

$$\hat{\sigma}^2 := \frac{1}{N-p-1} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 = \frac{\text{RSS}}{N-p-1}.$$

e.g. Under the null hypothesis that $\beta_j = 0$, the z-score

$$z_j := \frac{\hat{\beta}_j}{\hat{\sigma}\sqrt{(\mathbf{X}^\top \mathbf{X})^{-1}_{jj})}} \sim t_{N-p-1} \approx \mathcal{N}(0,1) \quad \text{for large } N.$$

- so absolute z scores ≥ 2 ensure significance at the 5% level.

An approximate $(1-2\alpha)$-confidence interval for β_j is given by

$$\left(\hat{\beta}_j - z_{1-\alpha}\hat{\sigma}\sqrt{(\mathbf{X}^\top \mathbf{X})^{-1}_{jj}}, \hat{\beta}_j + z_{1-\alpha}\hat{\sigma}\sqrt{(\mathbf{X}^\top \mathbf{X})^{-1}_{jj}}\right)$$

where $z_{1-\alpha} := \Phi^{-1}(1-\alpha)$.

Hypothesis Testing (Assuming Normal Errors)

To test the null hypothesis

$$H_0 \ : \ \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

versus the alternative

$$H_a \ : \ \text{at least one } \beta_i \text{ is non-zero.}$$

We can compute the F-statistic

$$F = \frac{(\mathsf{TSS} - \mathsf{RSS})/p}{\mathsf{RSS}/(N-p-1)}$$

which has an $F_{p,N-p-1}$ distribution under H_0

- hence large values of F constitute evidence against H_0
- can compute the p-value $= \text{Prob}(F_{p,N-p-1} \geq F)$

Hypothesis Testing (Assuming Normal Errors)

Can also test that a subset of the coefficients equal zero:

$$H_0 \;:\; \beta_{p-q+1} = \beta_{p-q+2} = \cdots = \beta_p = 0$$

versus the alternative that at least one of these coefficients is non-zero.

In this case the F-statistic is

$$F := \frac{(\mathsf{RSS}_0 - \mathsf{RSS})/q}{\mathsf{RSS}/(N-p-1)}$$

where $\mathsf{RSS}_0 = \mathsf{RSS}$ for model that uses all variables except for last q.

Under the null hypothesis that the nested model (with $\beta_{p-q+2} = \cdots = \beta_p = 0$) fits the data sufficiently well we have $F \sim F_{q,n-p-1}$

- which we can use to compute p-values.

Such F-tests are commonly used for model selection in classical statistics

- but they only work when $p << N$
- they are also problematic due to issues associated with multiple testing.

Will prefer to use more general validation set and cross-validation approaches for model selection – to be covered soon.

The Advertising Data Set from ISLR

Figure 2.1. displays the advertising data set from ISLR. It consists of:

- Sales of a particular product in 200 different markets
- Advertising budgets for the product in each of those markets for three different media:
    1. TV
    2. radio
    3. newspaper

The goal is to answer the following questions:

1. Is there a relationship between advertising budget and sales?
2. How strong is the relationship between advertising budget and sales?
3. Which media contribute to sales?
4. How accurately can we estimate the effect of each medium on sales?
5. How accurately can we predict future sales?
6. Is the relationship linear?
7. Is there synergy among the advertising media?

Section 3.4 of ISLR provides answers to these questions

- but need to read earlier sections of chapter 3 first.

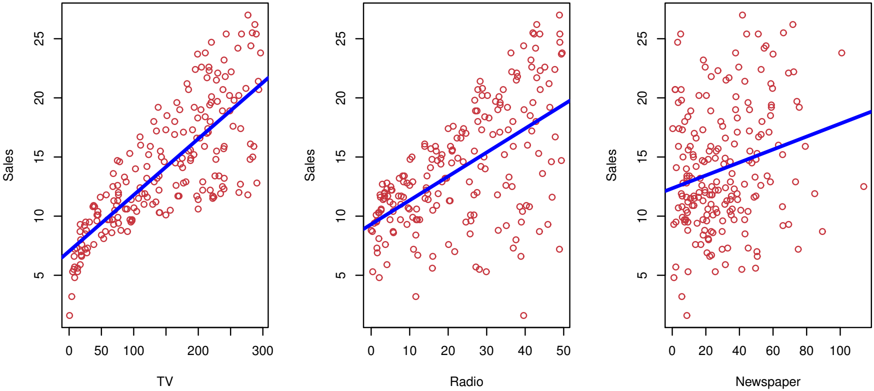


Figure 2.1 from ISLR: The Advertising data set. The plot displays sales, in thousands of units, as a function of TV, radio, and newspaper budgets, in thousands of dollars, for 200 different markets. In each plot we show the simple least squares fit of sales to that variable, as described in Chapter 3. In other words, each blue line represents a simple model that can be used to predict sales using TV, radio, and newspaper, respectively.

The Credit Data-Set from ISLR

The credit data-set from ISLR contains quantitative data on following variables for a number of customers. See Fig. 3.6 for corresponding scatter-plot matrix .

- balance (average credit card debt)
- age (in years).
- cards (number of credit cards)
- education (years of education)
- income (in thousands of dollars)
- limit (credit limit)
- rating (credit rating)

There are also four qualitative variables:

- gender
- student (student status)
- status (marital status)
- ethnicity (Caucasian, African American or Asian)

See Section 3.3 of ISLR for analysis and discussion of this data-set and in particular, how to handle qualitative variables using dummy variables.

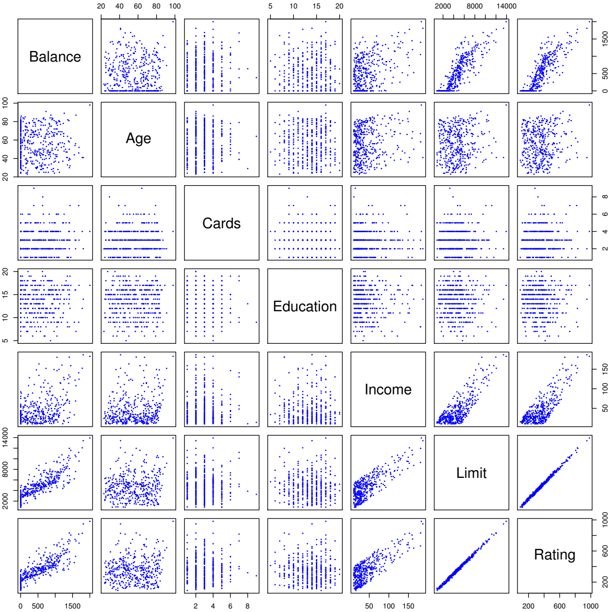


Figure 3.6 from ISLR: The Credit data set contains information about balance, age, cards, education, income, limit, and rating for a number of potential customers.

Potential Problems with Linear Regression

Many problems can arise when fitting a linear model to a particular data-set:

1. Non-linearity of the response-predictor relationships
   - plotting residuals against fitted values are a useful graphical tool for identifying this problem
   - a simple solution is to use non-linear transformations of the predictors.
2. Correlation of error terms
   - a serious problem since estimation of σ^2 and statistical tests all depend on assumption of zero-correlation
   - problem can arise with time-series data – can detect it then by plotting residuals against time.
3. Non-constant variance or heteroscedasticity of error terms
   - another important assumption that can be tested by plotting residuals against fitted values
   - if problem exists consider applying a concave function to Y.
4. Outliers, i.e. points for which y_i is far from the predicted value $\hat{\boldsymbol{\beta}}^\top X_i$
   - could be genuine or a data error
   - may or may not impact fitted model – but regardless will impact $\hat{\sigma}^2$, confidence intervals and p-values, possibly dramatically
   - can identify them by plotting studentized residuals against fitted values – values > 3 in absolute value are suspicious.

Potential Problems with Linear Regression

5. High-leverage points
    - these are points whose presence has a large impact on the fitted model
    - generally correspond to extreme predictor **X**
    - can identify such points via their leverage statistic, $h_i := H_{ii}$; always the case that $h_i \in [1/N,\ 1]$.

6. Collinearity and multi-collinearity
    - collinearity is the problem when two or more predictor variables are highly correlated
    - difficult then to separate out the individual effects and corresponding coefficients tend to have very high variances
    - can assess multi-collinearity by computing the variance inflation factor (VIF) which is the ratio of Var $\left(\hat{\beta}_i\right)$ when fitting the full model divided by Var $\left(\hat{\beta}_i\right)$ if fit on its own
        - smallest possible value is 1; rule of thumb is that values exceeding 5 or 10 indicate collinearity
    - solution is to either drop one of the variables or combine them into a single predictor. e.g. in credit data set could combine limit and rating into a single variable.

See discussion in Section 3.3.3 of ISLR for further discussion.

Linear Regression with Basis Functions

Can also do everything with basis functions

$$Y = \beta_0 + \sum_{i=1}^{M} \beta_i \psi_i(\mathbf{x}) + \epsilon$$

where $\psi_i : \mathbb{R}^p \mapsto \mathbb{R}$ is the i^{th} basis function.

Example: $\psi_i(\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{p/2}} e^{-\frac{1}{2\sigma^2} \|\mathbf{x} - \boldsymbol{\mu}_i\|_2^2}$.

The $\psi_i(\mathbf{x})$'s are often used to encode domain-specific knowledge.

Parameter estimate:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{\Psi}^\top \boldsymbol{\Psi})^{-1} \boldsymbol{\Psi}^\top \mathbf{y}$$

where

$$\boldsymbol{\Psi} = \begin{bmatrix} 1 & \psi_1(\mathbf{x}_1) & \psi_2(\mathbf{x}_1) & \ldots & \psi_M(\mathbf{x}_1) \\ 1 & \psi_1(\mathbf{x}_2) & \psi_2(\mathbf{x}_2) & \ldots & \psi_M(\mathbf{x}_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \psi_1(\mathbf{x}_N) & \psi_2(\mathbf{x}_N) & \ldots & \psi_M(\mathbf{x}_N) \end{bmatrix}$$

Linear Regression with Basis Functions

Ability to use basis functions extends the scope of linear regression to obtain non-linear relationships between Y and $\mathbf{X}$

- this flexibility can be very valuable
- when basis functions are simply powers of the original inputs we call it polynomial regression
- splines can also be implemented via basis functions.

If M gets too large then solving for $\hat{\beta}$ may become intractable

- but kernel methods and so-called "kernel trick" can then come to the rescue
- in which case possible to even take $M = \infty$

Will defer study of kernel methods until we study support vector machines

- but note here they are applicable to many forms of regression including linear and ridge regression.

Can also fit non-linear models using smoothing splines, local regression or GAMs (generalized additive models)

- will not study them in this course but see Chapter 7 of ISLR for details.

Why Minimize the Sum-of-Squares?

Let $\mathbf{X}$ be non-random and suppose we want to estimate $\theta := \mathbf{a}^\top \boldsymbol{\beta}$.

Then least-squares estimate of θ is

$$\hat{\theta} = \mathbf{a}^\top \hat{\boldsymbol{\beta}} = \mathbf{a}^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

– a linear function of the response $\mathbf{y}$.

If the linear model is correct then easy to check that $\mathrm{E}[\hat{\theta}] = \theta$ so $\hat{\theta}$ is unbiased.

Gauss-Markov Theorem: Suppose $\mathbf{c}^\top \mathbf{y}$ is any unbiased estimate of θ. Then

$$\mathrm{Var}\left(\mathbf{a}^\top \hat{\boldsymbol{\beta}}\right) \leq \mathrm{Var}\left(\mathbf{c}^\top \mathbf{y}\right).$$

The Gauss-Markov Theorem says that the least-squares estimator has the smallest variance among all linear unbiased estimators.

Question: Great! But is unbiasedness a good thing?

Mean-Squared Error

To answer this question let $\tilde{\theta}$ be some estimator for θ.

The mean-squared-error (MSE) then satisfies

$$\begin{aligned}\mathsf{MSE}(\tilde{\theta}) &= \mathsf{E}\left[\left(\tilde{\theta}-\theta\right)^2\right] \\ &= \mathsf{Var}(\tilde{\theta}) + \underbrace{\left(\mathsf{E}\left[\tilde{\theta}\right]-\theta\right)^2}_{\mathsf{bias}^2}.\end{aligned}$$

If the goal is to minimize MSE then unbiasedness not necessarily a good thing

- can often trade a small increase in bias^2 for a larger decrease in variance
- will do this later with subset selection methods as well as shrinkage methods
  - an added benefit of some of these methods is improved interpretability.

But first let's study the bias-variance decomposition.

The Bias-Variance Decomposition

Assume the true model is $Y = f(\mathbf{X}) + \epsilon$ where $\mathsf{E}[\epsilon] = 0$ and $\mathsf{Var}(\epsilon) = \sigma_\epsilon^2$. Let $\hat{f}(\mathbf{X})$ be our estimate at a new fixed point, $\mathbf{X} = \mathbf{x}_0$. Then the error at $\mathbf{x}_0$ assuming the training inputs are fixed, i.e. non-random, is:

$$
\begin{aligned}
\mathsf{Err}(\mathbf{x}_0) &= \mathsf{E}\left[\left(Y - \hat{f}(\mathbf{x}_0)\right)^2\right] \\
&= \mathsf{E}\left[\left(f(\mathbf{x}_0) + \epsilon - \hat{f}(\mathbf{x}_0)\right)^2\right] \\
&= \mathsf{E}\left[\epsilon^2\right] + \mathsf{E}\left[\left(f(\mathbf{x}_0) - \hat{f}(\mathbf{x}_0)\right)^2\right] \qquad (1) \\
&= \sigma_\epsilon^2 + \mathsf{E}\left[\left(f(\mathbf{x}_0) - \mathsf{E}[\hat{f}(\mathbf{x}_0)] + \mathsf{E}[\hat{f}(\mathbf{x}_0)] - \hat{f}(\mathbf{x}_0)\right)^2\right] \\
&= \sigma_\epsilon^2 + \left(f(\mathbf{x}_0) - \mathsf{E}[\hat{f}(\mathbf{x}_0)]\right)^2 + \mathsf{E}\left[\left(\hat{f}(\mathbf{x}_0) - \mathsf{E}[\hat{f}(\mathbf{x}_0)]\right)^2\right] \\
&= \sigma_\epsilon^2 + \mathsf{Bias}^2\left(\hat{f}(\mathbf{x}_0)\right) + \mathsf{Var}\left(\hat{f}(\mathbf{x}_0)\right) \\
&= \text{Irreducible Error} + \text{Bias}^2(\mathbf{x}_0) + \text{Variance}(\mathbf{x}_0). \qquad (2)
\end{aligned}
$$

The Bias-Variance Decomposition

The irreducible error is unavoidable and beyond our control.

But we can exercise control over the bias and variance via our choice of $\hat{f}(\mathbf{x}_0)$

- the more complex the model the smaller the bias and the larger the variance.

Example: k-Nearest Neighbor Regression. In this case (2) reduces to

$$\mathrm{Err}(\mathbf{x}_0) = \sigma_\epsilon^2 + \left(f(\mathbf{x}_0) - \frac{1}{k} \sum_{l=1}^{k} f(\mathbf{x}_{(l)}) \right)^2 + \frac{\sigma_\epsilon^2}{k} \tag{3}$$

where $\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(k)}$ are the k nearest neighbors to $\mathbf{x}_0$ and (for simplicity) we've assumed the training inputs are all fixed.

Here k is inversely related to model complexity (why?) and then see:

- bias typically decreases with model complexity
- variance increases with model complexity.

Can repeat this analysis for other models, e.g. linear or ridge regression etc

- goal is to choose the model complexity which corresponds to the optimal bias-variance tradeoff.

The Bias-Variance Decomposition

A more general form of bias-variance decomposition assumes test point $\mathbf{x}$ is selected randomly and also accounts for randomness of training data, $\mathcal{D}$ say.

In this case and starting from (1) we obtain the error in entire learning process:

$$
\begin{aligned}
\mathsf{Err}(\hat{f}) &= \sigma_\epsilon^2 + \mathsf{E}_{\mathbf{x},\mathcal{D}}\left[\left(f(\mathbf{x}) - \hat{f}(\mathbf{x};\mathcal{D})\right)^2\right] \\
&= \sigma_\epsilon^2 + \mathsf{E}_{\mathbf{x}}\left[\mathsf{E}_{\mathcal{D}}\left[\left(f(\mathbf{x}) - \hat{f}(\mathbf{x};\mathcal{D})\right)^2\right]\right] \\
&= \sigma_\epsilon^2 + \mathsf{E}_{\mathbf{x}}\left[\mathsf{E}_{\mathcal{D}}\left[\hat{f}(\mathbf{x};\mathcal{D})^2\right] - 2\mathsf{E}_{\mathcal{D}}\left[\hat{f}(\mathbf{x};\mathcal{D})\right] f(\mathbf{x}) + f(\mathbf{x})^2\right] \\
&= \sigma_\epsilon^2 + \mathsf{E}_{\mathbf{x}}\left[\mathsf{E}_{\mathcal{D}}\left[\hat{f}(\mathbf{x};\mathcal{D})^2\right] - 2\bar{\hat{f}}(\mathbf{x}) f(\mathbf{x}) + f(\mathbf{x})^2\right] \\
&= \sigma_\epsilon^2 + \mathsf{E}_{\mathbf{x}}\Bigg[\underbrace{\mathsf{E}_{\mathcal{D}}\left[\hat{f}(\mathbf{x};\mathcal{D})^2\right] - \bar{\hat{f}}(\mathbf{x})^2}_{\mathsf{E}_{\mathcal{D}}\left[\left(\hat{f}(\mathbf{x};\mathcal{D}) - \bar{\hat{f}}(\mathbf{x})\right)^2\right]} + \underbrace{\bar{\hat{f}}(\mathbf{x})^2 - 2\bar{\hat{f}}(\mathbf{x}) f(\mathbf{x}) + f(\mathbf{x})^2}_{\left(\bar{\hat{f}}(\mathbf{x}) - f(\mathbf{x})\right)^2}\Bigg] \\
&= \text{Irreducible Error} + \mathsf{E}_{\mathbf{x}}\left[\text{Variance}(\mathbf{x}) + \text{Bias}^2(\mathbf{x})\right] \\
&= \text{Irreducible Error} + \text{Variance} + \text{Bias}^2.
\end{aligned}
$$

Example: the Bias-Variance Trade-Off

Consider the following example from Bishop:

1. The "true" model to be estimated is

$$y(x) = \sin(2\pi x) + \epsilon, \quad x \in [0,1], \qquad \epsilon \sim \mathcal{N}(0, c) \tag{4}$$

   - a very nonlinear function of x.

2. We fit a linear regression model with $M = 24$ Gaussian basis functions

$$\psi_j(x) := e^{-\frac{1}{2\sigma^2}(x-\mu_j)^2}$$

   with $\mu_j = \frac{j}{M-1}$ for $j = 0, \ldots, M-1$ and $\sigma = \frac{1}{M-1}$.

3. Including the constant term the parameter vector $\boldsymbol{\beta}$ is $(M+1) \times 1$.

4. We will also include a regularization term so that regression problem solves

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{j=1}^{N} \left(Y_j - \beta_0 - \sum_{i=1}^{M} \beta_i \psi_i(\mathbf{x}_j) \right)^2 + \frac{\lambda}{2} \boldsymbol{\beta}^\top \boldsymbol{\beta} \tag{5}$$

Example: the Bias-Variance Trade-Off

5. A data-set if of the form $\mathcal{D} = \{(y_i, x_i) : i = 1, \ldots, N\}$, with $N = 25$
    - the x_i's are sampled randomly from $[0, 1]$
    - the y_i's are then sampled using (4).
        - so noise comes both from measurement and sampling.
6. We generate $L = 100$ of these data-sets.
7. The model is fit by solving (5) for each of the L data-sets and various values of λ.

Results are displayed in Figure 3.5:

The model bias is clear from graphs in right-hand column.

The variance of individual fits is clear from graphs in left-hand column.

The bias-variance tradeoff is clear and quantified in Figure 3.6.

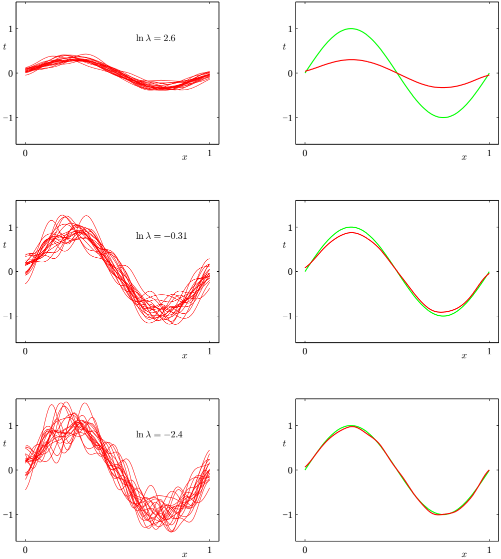


Figure 3.5 from Bishop: Illustration of the dependence of bias and variance on model complexity, governed by a regularization parameter λ, using the sinusoidal data set from Chapter 1. There are $L = 100$ data sets, each having $N = 25$ data points, and there are 24 Gaussian basis functions in the model so that the total number of parameters is $M = 25$ including the bias parameter. The left column shows the result of fitting the model to the data sets for various values of $\ln \lambda$ (for clarity, only 20 of the 100 fits are shown). The right column shows the corresponding average of the 100 fits (red) along with the sinusoidal function from which the data sets were generated (green).

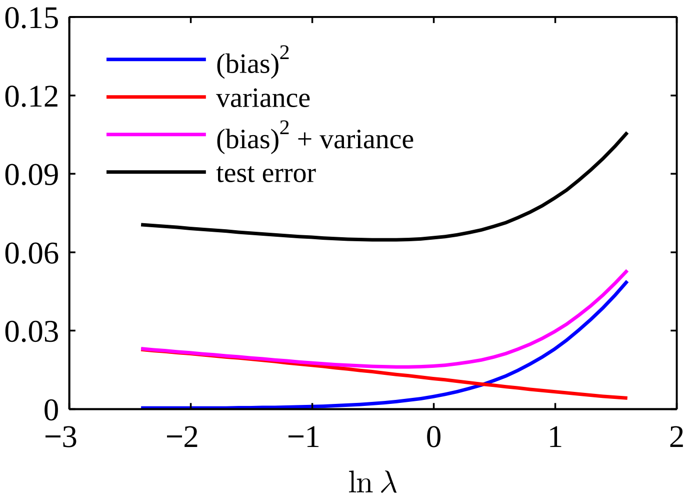


Figure 3.6 from Bishop: Plot of squared bias and variance, together with their sum, corresponding to the results shown in Figure 3.5. Also shown is the average test set error for a test data set size of 1000 points. The minimum value of $(\text{bias})^2$ $+$ variance occurs around $\ln \lambda = -0.31$, which is close to the value that gives the minimum error on the test data.

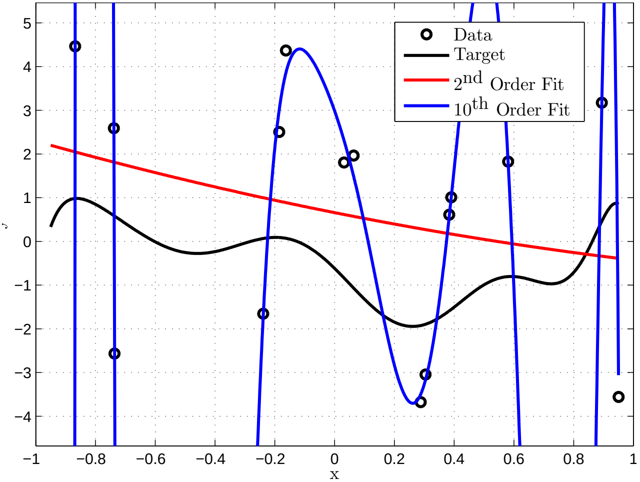
A Case Study: Overfitting with Polynomials

Overfitting is a very serious issue that needs to be handled in supervised learning problems.

To explore overfitting in further detail we will consider two 1-dimensional polynomial regression problems.

Problem 1

- True model is $y = f(x) + \epsilon$ where ϵ is IID noise and $f(x)$ is a 10^{th} order polynomial on $x \in \mathbb{R}$.
- There are $n = 15$ datapoints: $(x_1, y_1), \ldots, (x_n, y_n)$
  - the x_i's were generated $\sim U(-1, 1)$ and then $y_i = f(x_i) + \epsilon_i$ where the ϵ_i's were generated IID $\mathrm{N}(0, 3)$.
- We fit 2^{nd} and 10^{th} order polynomials to this data via simple linear regression, that is we regress Y on $1, X, \ldots, X^J$ where $J = 2$ or $J = 10$.
- The results are displayed in the figure on the next slide.



Fitting a Low-Order Polynomial With Noisy Data: The target curve is the 10^{th} order polynomial $y = f(x)$.

A Case Study: Overfitting with Polynomials

Question: Which regression results in a superior fit *to the data*?

Question: Which regression results in a superior *out-of-sample* or generalization error?

Note that the set of 10^{th} order polynomials *contains* the true target function, $y = f(x)$, whereas the set of 2^{nd} order polynomials does not.

We might therefore expect the 10^{th} order fit to be superior to the 2^{nd} order fit

- but this is not the case!

Question: Why do you think the 2^{nd} order fit does a better job here?

Question: Do you think the 2^{nd} order fit will always be better irrespective of N, the number of data-points?

A Case Study: Overfitting with Polynomials

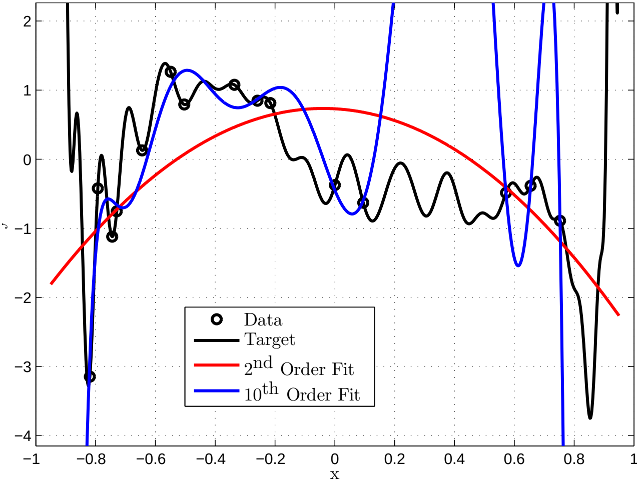
Problem 2

- True model is $y = f(x)$ and $f(x)$ is a 50^{th} order polynomial on $x \in \mathbb{R}$.
- There are $n = 15$ datapoints: $(x_1, y_1), \ldots, (x_n, y_n)$
  - the x_i's were generated $\sim U(-1, 1)$ and then $y_i = f(x_i)$ so the observations are noiseless.
- We fit 2^{nd} and 10^{th} order polynomials to this data via simple linear regression, that is we regress Y on $1, X, \ldots, X^J$ where $J = 2$ or $J = 10$.
- The results are displayed in the figure on the next slide.

Commonly thought that overfitting occurs when the fitted model is too complex relative to the true model

- but this is not the case here: clearly the 10^{th} order regression overfits the data but a 10^{th} order polynomial is considerably less complex than a 50^{th} order polynomial.

What matters is how the model complexity matches the quantity and quality of the data, not the (unknown) target function.



Fitting a High-Order Polynomial With Noiseless Data: The target curve is the 10^{th} order polynomial $y = f(x)$.

Note: This case study is based on the case study in Section 4.1 of "*Learning from Data*" by Abu-Mostafa, Magdon-Ismail and Lin.

Methods for Exploring the Bias-Variance Trade-Off and Controlling Overfitting

It is **vital** then to control over-fitting when performing supervised learning, i.e. regression or classification. There are many approaches:

- Subset selection where we retain only a subset of the independent variables
- Shrinkage methods where coefficients are shrunk towards zero.
- Regularization where we penalize large-magnitude parameters
  - shrinkage often achieved via regularization

Cross-validation often used to select the specific model. Other methods include:

- Bayesian models
  - many shrinkage / regularization methods can be interpreted as Bayesian models where the penalty on large-magnitude parameters becomes a prior distribution on those parameters.
- Methods that explicitly penalize the number of parameters, p, in the model
  - Akaike Information Criterion (AIC) $= -2\ln(\text{likelihood}) + 2(p+1)$
  - Bayesian Information Criterion (BIC): $-2\ln(\text{likelihood}) + (p+1)\ln(N)$

  Choose the model that minimizes the AIC or BIC
  - these methods apply to models fit via MLE.

Will study most of these methods during the course!