Machine Learning

Topic 4: Linear Regression Models

(contains ideas and a few images from wikipedia and books by Alpaydin, Duda/Hart/Stork, and Bishop. Updated Fall 2015)

Regression Learning Task

There is a set of possible examples $X = \{\mathbf{x_1}, \ldots \mathbf{x_n}\}$

Each example is a **vector** of k **real valued attributes**

$$\mathbf{x}_i = < x_{i1}, \ldots, x_{ik} >$$

There is a target function that maps *X* onto some **real value** *Y*

$$f : X \rightarrow Y$$

The DATA is a set of tuples <example, response value>

$$\{< \mathbf{x}_1, y_1 >, \ldots < \mathbf{x_n}, y_n >\}$$

Find a hypothesis ***h*** such that...

$$\forall \mathbf{x}, h(\mathbf{x}) \approx f(\mathbf{x})$$

Why use a linear regression model?

- Easily understood
- Interpretable
- Well studied by statisticians
  - many variations and diagnostic measures
- Computationally efficient

Linear Regression Model

Assumption: The observed response (dependent) variable, r, is the true function, f(x), with additive Gaussian noise, ε, with a 0 mean.

Observed response $$y = f(\mathbf{x}) + \varepsilon$$

Where $$\varepsilon \sim \mathcal{N}(0, \sigma^2)$$

Assumption: The expected value of the response variable ***y*** is a linear combination of the k independent attributes/features)

The Hypothesis Space

Given the assumptions on the previous slide, our hypothesis space is the set of linear functions (hyperplanes)

$$h(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots w_k x_k$$

(w_0 is the offset from the origin. You always need w_0)

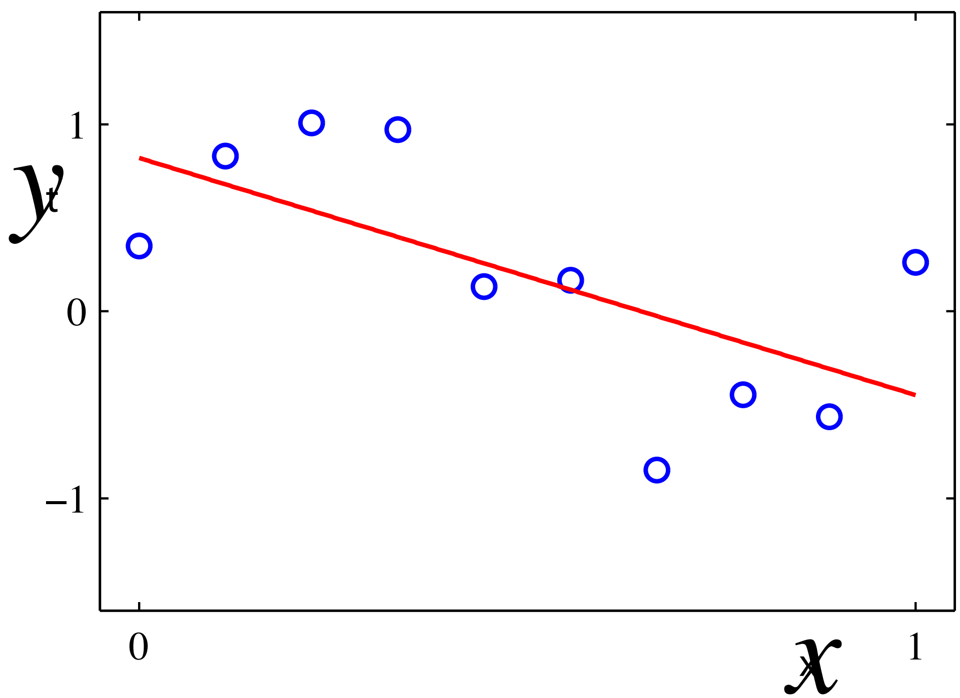
The goal is to learn a k+1 dimensional vector of weights that define a hyperplane minimizing an error criterion.

$$\mathbf{w} = < w_0, w_1, \ldots w_k >$$

Simple Linear Regression

- x has 1 attribute a (predictor variable)
- Hypothesis function is a line:

Example: $$\hat{y} = h(x) = w_0 + w_1 x$$

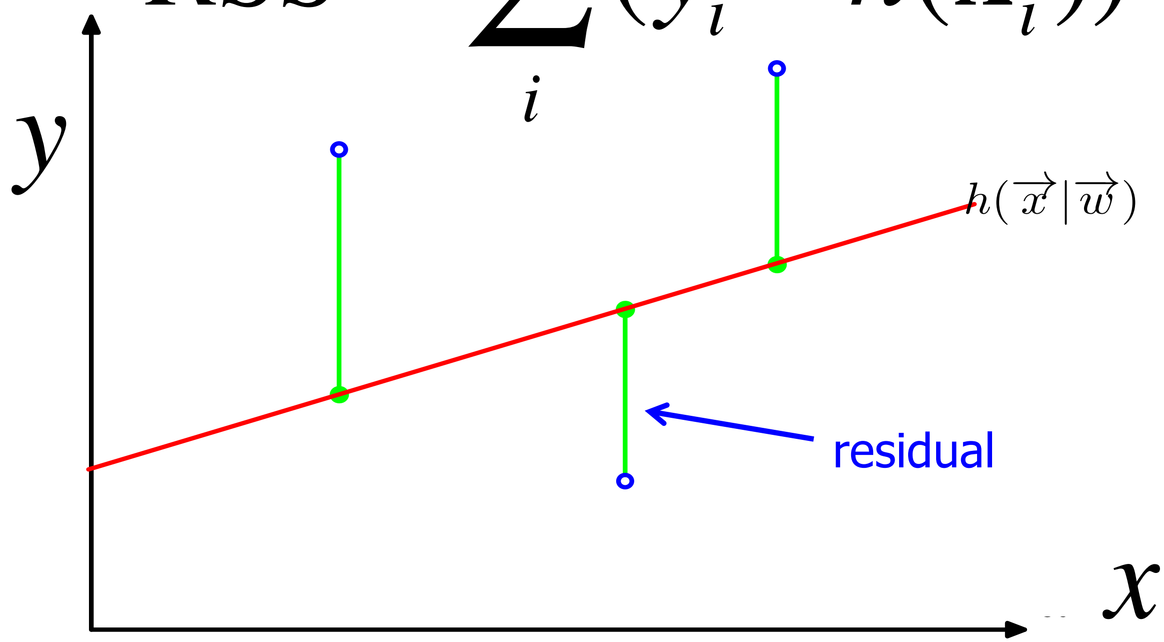


The Error Criterion

Typically estimate parameters by minimizing sum of squared residuals (RSS)…also known as the Sum of Squared Errors (SSE)

number of training examples

$$RSS = \sum_{i}^{n} (y_i - h(\mathbf{x}_i))^2$$

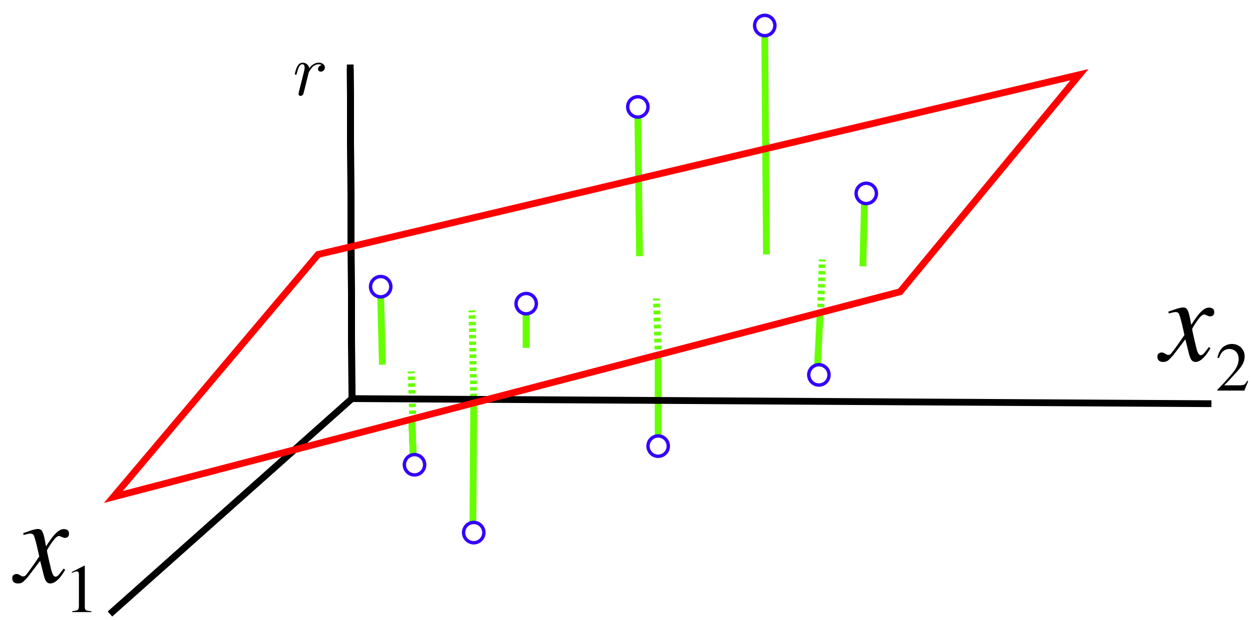


Multiple (Multivariate*) Linear Regression

- Many attributes $x_1, \ldots x_k$
- h(**x**) function is a hyperplane

*NOTE: In statistical literature, multivariate linear regression is regression with multiple outputs, and the case of multiple input variables is simply "multiple linear regression"

$$h(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots w_k x_k$$



Formatting the data

Create a new 0 dimension with 1 and append it to the beginning of every example vector $\mathbf{x}_i$

This placeholder corresponds to the offset w_0

$$\mathbf{x}_i = < 1, x_{i,1}, x_{i,2} \ldots, x_{i,k} >$$

Format the data as a matrix of examples **x** and a vector of response values *y*…

One training example

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & \ldots & x_{1,k} \\ 1 & x_{2,1} & \ldots & x_{2,k} \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_{n,k} & \ldots & x_{n,k} \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{bmatrix}$$

There is a closed-form solution!

Our goal is to find the weights of a function….

$$h(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \ldots w_k x_k$$

…that minimizes the sum of squared residuals:

$$RSS = \sum_{i}^{n} (y_i - h(\mathbf{x}_i))^2$$

It turns out that there is a close-form solution to this problem!

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

Just plug your training data into the above formula and the best hyperplane comes out!

RSS in vector/matrix notation

$$
\begin{aligned}
RSS(\mathbf{w}) &= \sum_{i=1}^{n} (y_i - h(\mathbf{x}_i))^2 \\
&= \sum_{i=1}^{n} (y_i - w_0 - \sum_{j=1}^{k} x_{ij} w_j)^2 \\
&= (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})
\end{aligned}
$$

Deriving the formula to find w

$$RSS(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$\frac{\partial RSS}{\partial \mathbf{w}} = -2\mathbf{X}^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$0 = -2\mathbf{X}^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$0 = \mathbf{X}^T (\mathbf{y} - \mathbf{X}\mathbf{w})$$

$$0 = \mathbf{X}^T \mathbf{y} - \mathbf{X}^T \mathbf{X}\mathbf{w}$$

$$\mathbf{X}^T \mathbf{X}\mathbf{w} = \mathbf{X}^T \mathbf{y}$$

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

What if X is not singular?

- We said there was a closed form solution:

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

- This presupposes matrix $(\mathbf{X}^T \mathbf{X})$ is invertible (non singular) and we can therefore find $(\mathbf{X}^T \mathbf{X})^{-1}$

- If two columns of X are exactly linearly related and thus not independent, then $(\mathbf{X}^T \mathbf{X})$ is NOT invertible

- What then?

Your Friend: Dimensionality Reduction

- We need to make every column of X independent.
- The easy way: add random noise to X.
- The (probably) better way: do dimensionality reduction to get rid of those redundant columns.

Making polynomial regression

You're familiar with linear regression where the input has k dimensions.

$$h(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + \dots w_k x_k$$

We can use this same machinery to make polynomial regression from a one-dimensional input…..

$$h(x) = w_0 + w_1 x + w_2 x^2 + \dots w_k x^k$$

Making polynomial regression

Given a scalar example z. We can make a k+1 dimensional example x

$$\mathbf{x} = \left\langle z^0, z^1, z^2, \ldots, z^k \right\rangle$$

The *i*th element of x is the power z^i

$$h(x) = w_0 + w_1 z + w_2 z^2 + \ldots w_k z^k$$

Making polynomial regression

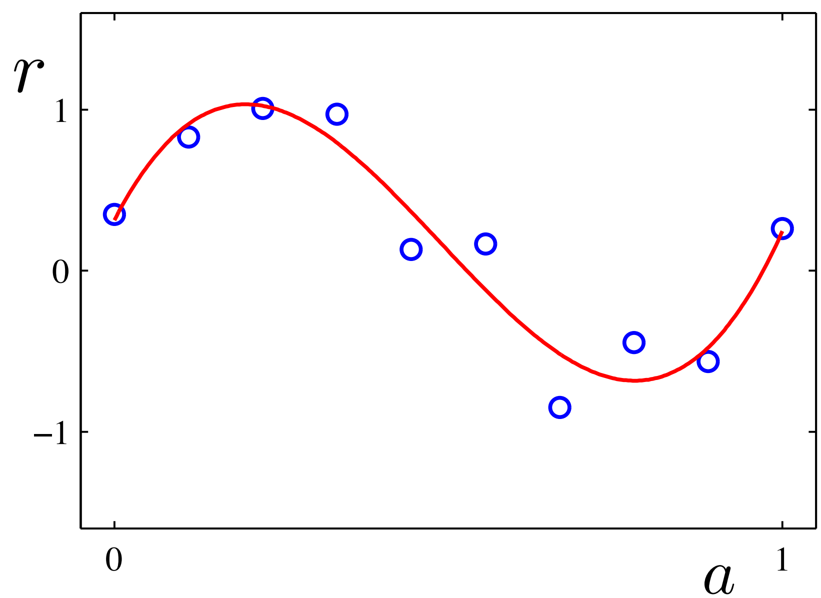
Since $x_k \equiv z^k$ we can interpret the output of the regression as a polynomial function of z

$$h(x) = w_0 + w_1 x_1 + w_2 x_2 + \ldots w_k x_k$$
$$= w_0 + w_1 z + w_2 z^2 + \ldots w_k z^k$$

Polynomial Regression

- Model the relationship between the response variable and the attributes/predictor variables as a k^{th}-order polynomial. While this can model non-linear functions, it is still linear with respect to the coefficients.

$$h(x) = w_0 + w_1 z + w_2 z^2 + w_3 z^3$$



Polynomial Regression

Parameter estimation (analytically minimizing sum of squared residuals):

One training example

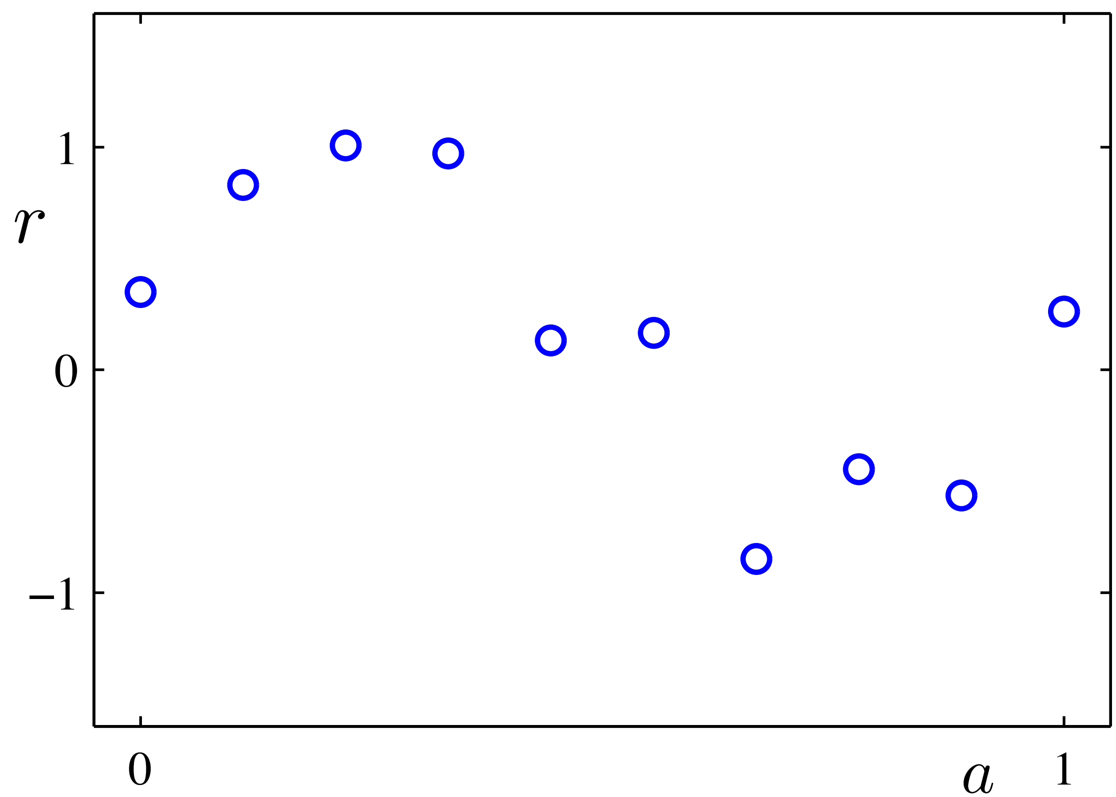
$$\mathbf{X} = \begin{bmatrix} 1 & z_1^1 & \ldots & z_1^k \\ 1 & z_2^1 & \ldots & z_2^k \\ \ldots & \ldots & \ldots & \ldots \\ 1 & z_n^1 & \ldots & z_n^k \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{bmatrix}$$

(Note, there is only 1 attribute z for each training example.
Those superscripts are powers, since we're doing polynomial regression)

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Tuning Model Complexity: Example

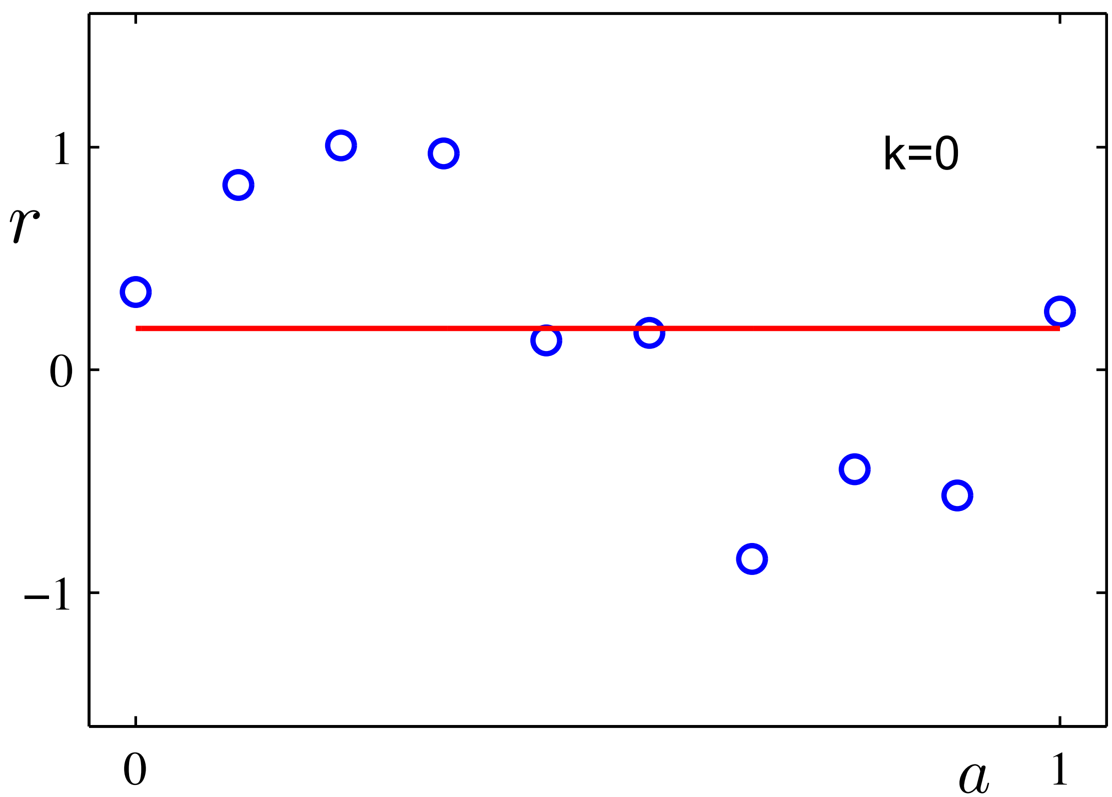
What is your hypothesis for $f(x)$?



Tuning Model Complexity: Example

What is your hypothesis for $f(x)$?

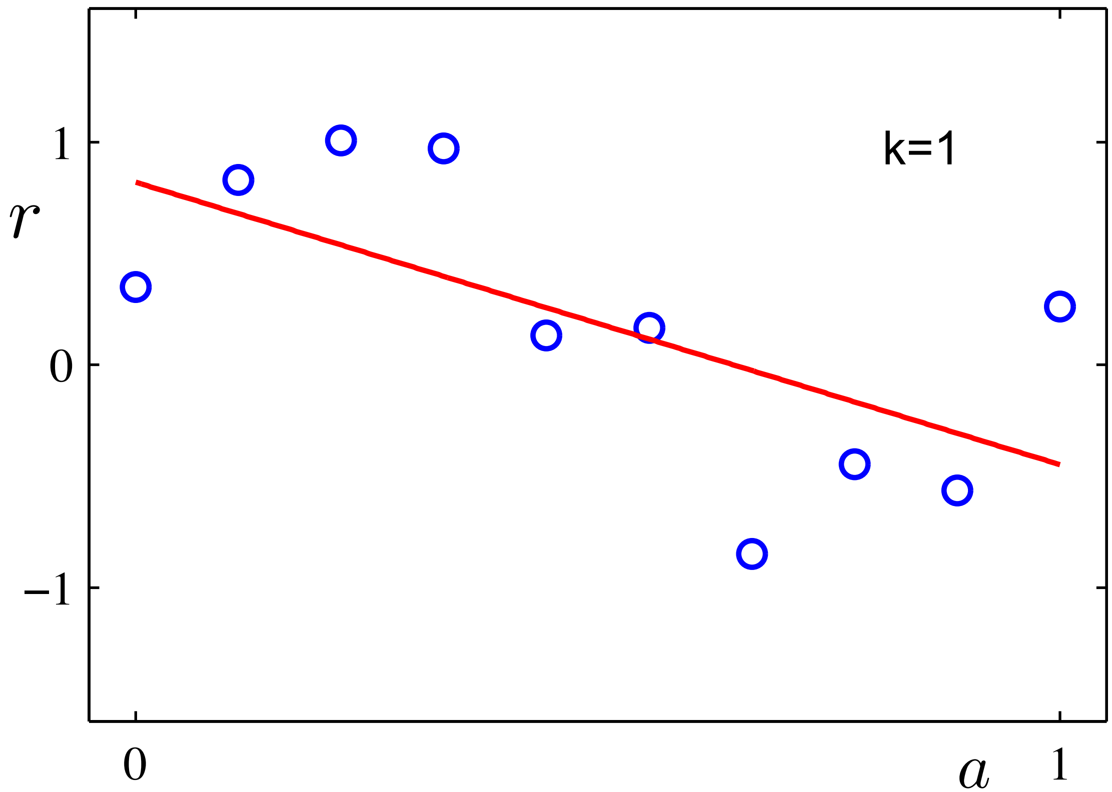
kth order polynomial regression



Tuning Model Complexity: Example

What is your hypothesis for $f(x)$?

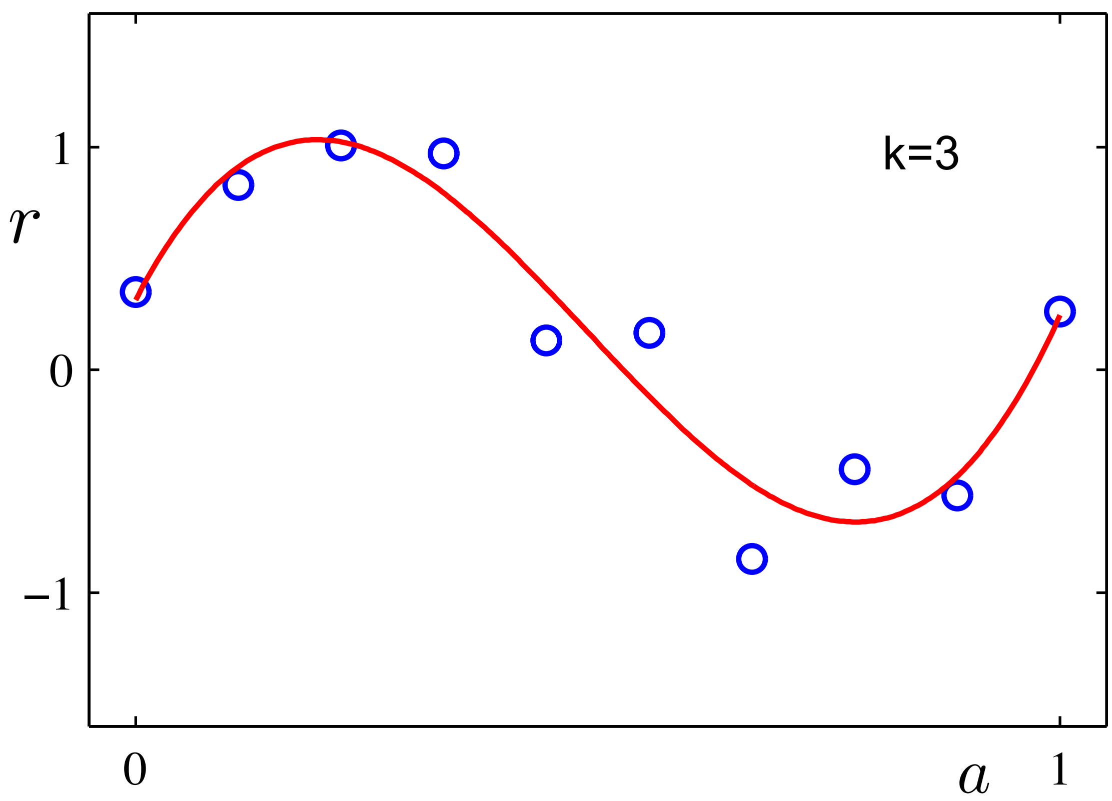
k^{th} order polynomial regression



Tuning Model Complexity: Example

What is your hypothesis for $f(x)$?

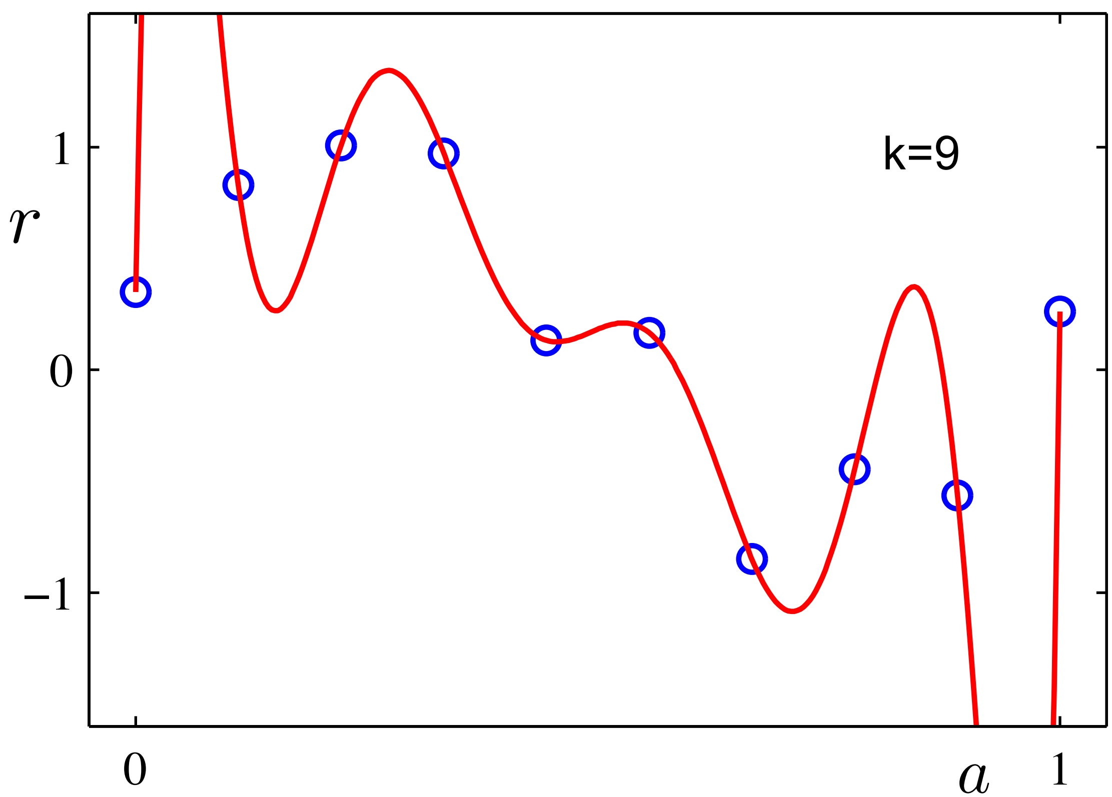
kth order polynomial regression



Tuning Model Complexity: Example

What is your hypothesis for $f(x)$?

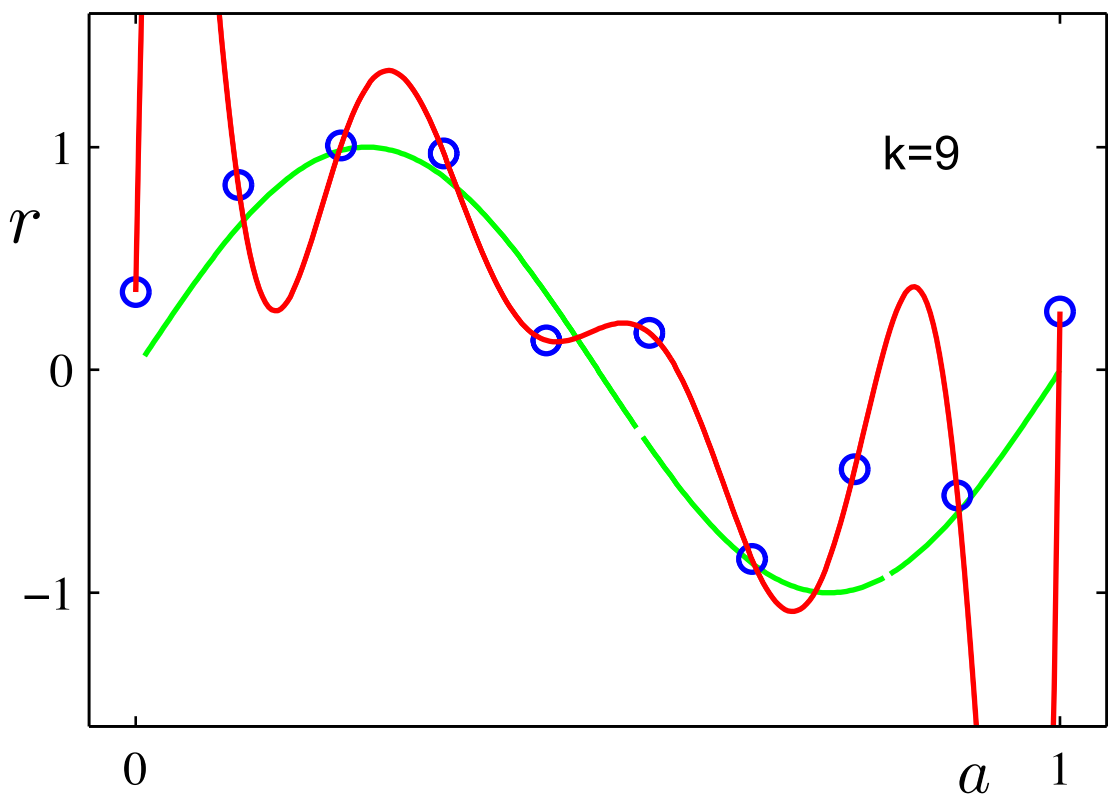
k^{th} order polynomial regression



Tuning Model Complexity: Example

What is your hypothesis for $f(x)$?

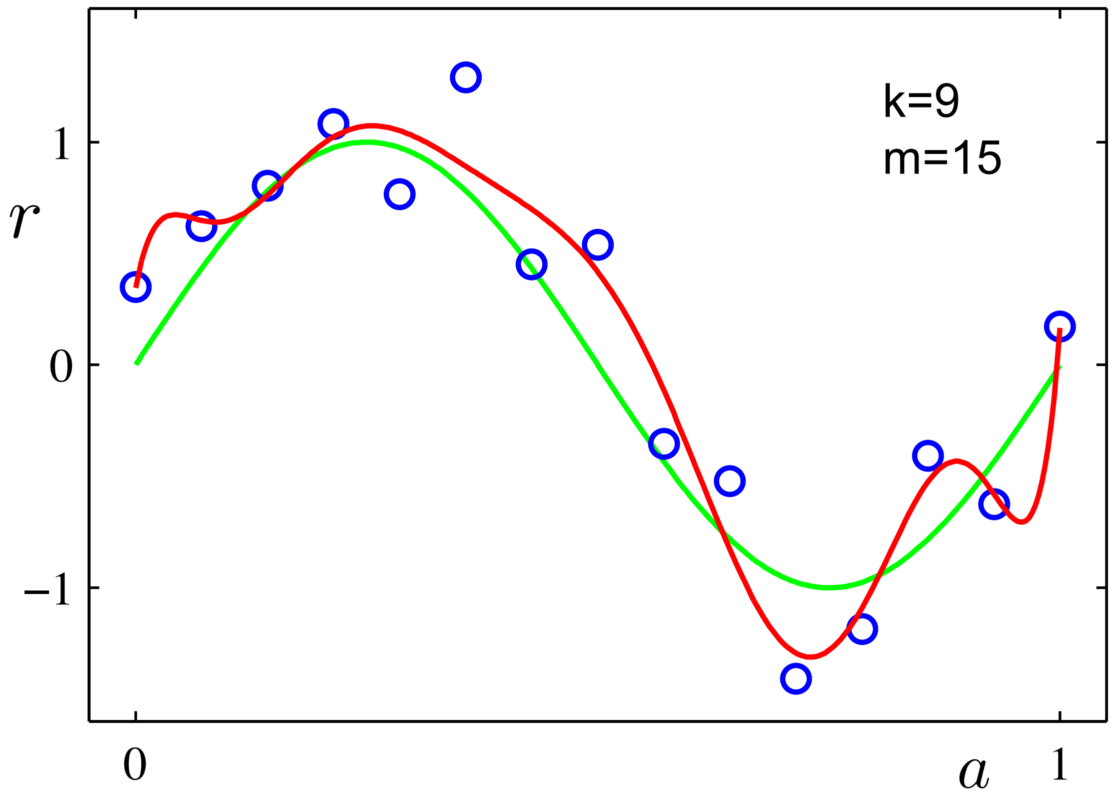
k^th order polynomial regression



Tuning Model Complexity: Example

What happens if we fit to more data?

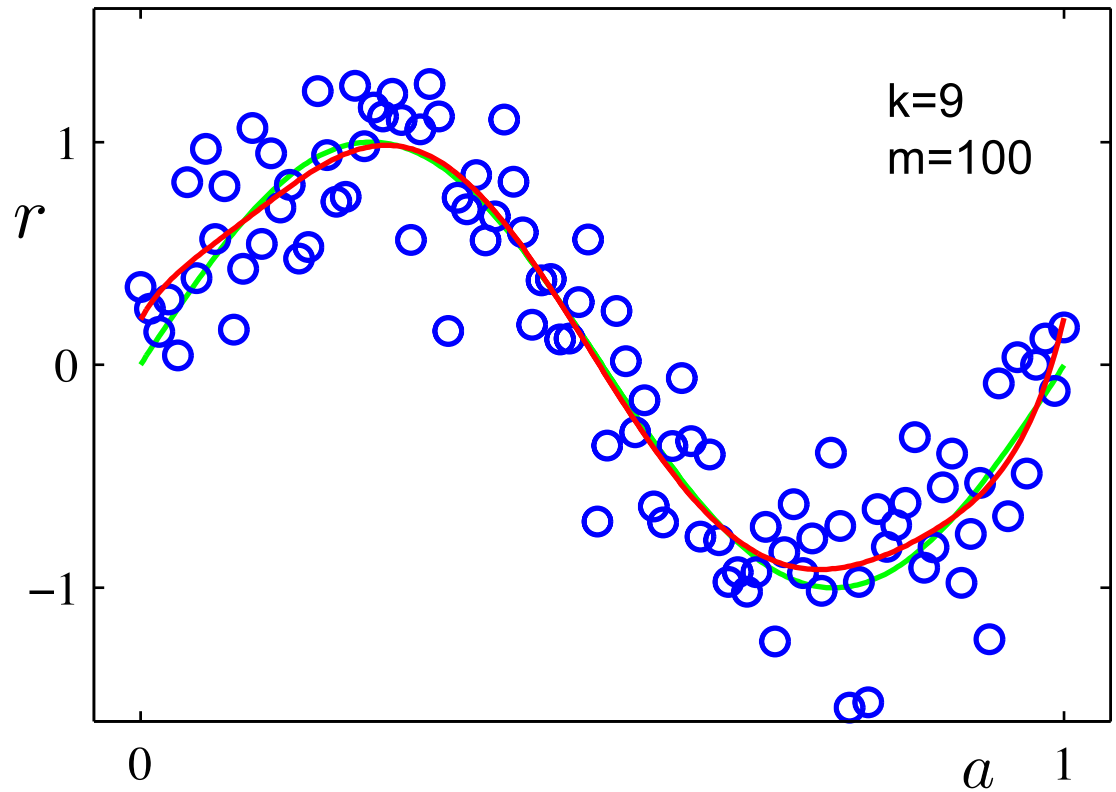
k^{th} order polynomial regression



Tuning Model Complexity: Example

What happens if we fit to more data?

k^{th} order polynomial regression



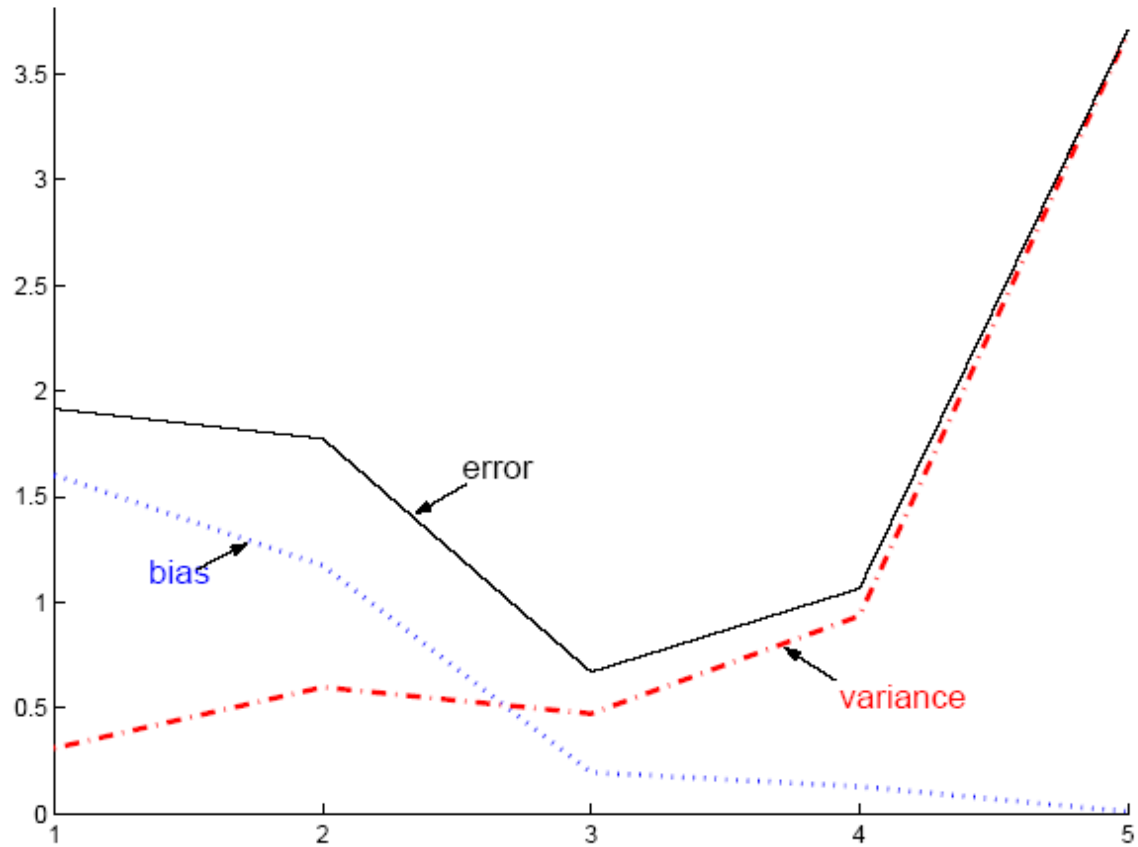
Bias and Variance of an Estimator

- Let X be a sample from a population specified by a true parameter θ
- Let d=d(X) be an estimator for θ

$$\mathbb{E}[(d-\theta)^2] = \mathbb{E}[(d-\mathbb{E}[d])^2] + (\mathbb{E}[d]-\theta)^2$$

mean square error *variance* *bias*^{2}

Bias and Variance



Order of a polynomial fit to some data

As we **increase complexity**, **bias decreases** (a better fit to data) and **variance increases** (fit varies more with data)

Bias and Variance of Hypothesis Fn

- Bias:

  Measures how much *h(x)* is wrong disregarding the effect of varying samples (This the statistical bias of an estimator. This is NOT the same as inductive bias, which is the set of assumptions that your learner is making)

- Variance:

  Measures how much *h(x)* fluctuate around the expected value as the sample varies.

NOTE: These concepts are general machine learning concepts, not specific to linear regression.

Coefficient of Determination

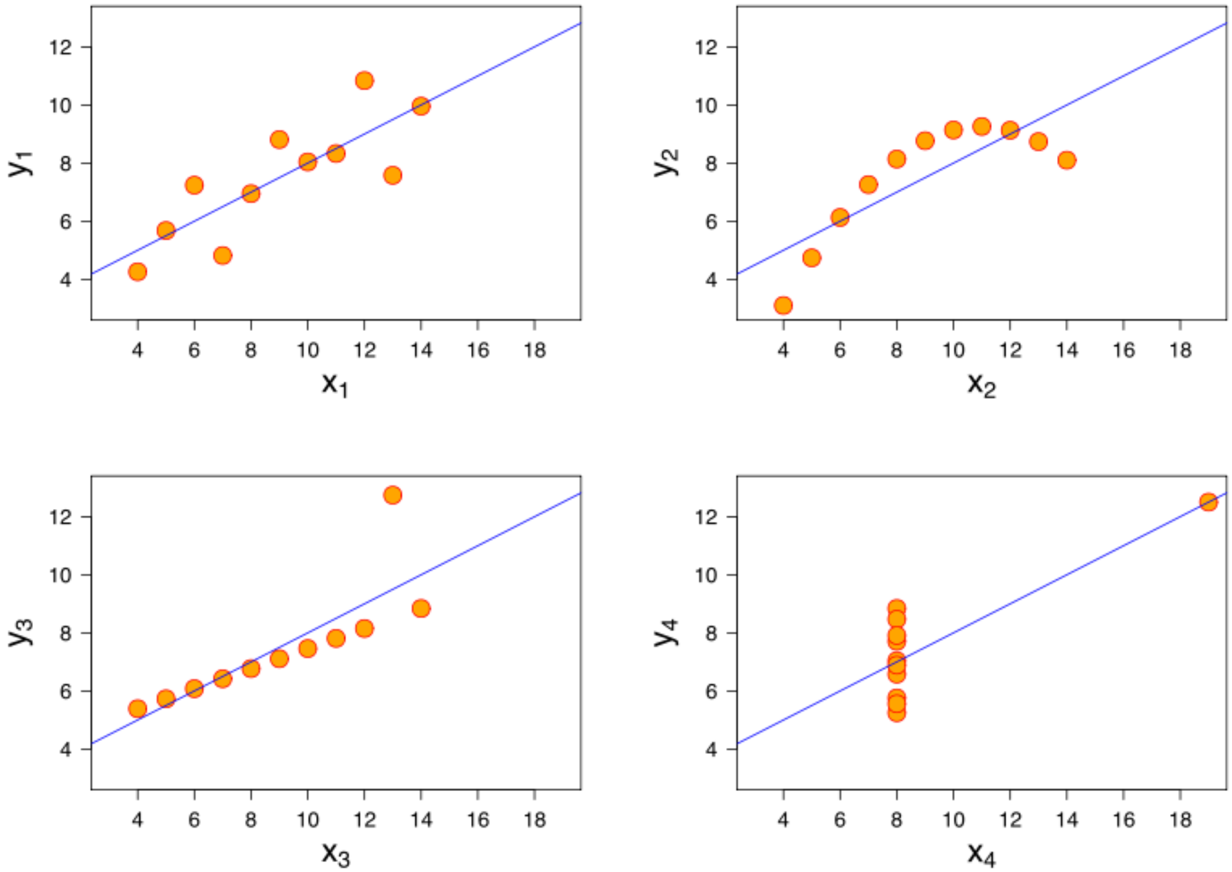
- the **coefficient of determination**, or ***R*²** indicates how well data points fit a line or curve. We'd like R^2 to be close to 1

$$R^2 = 1 - E_{RSE}$$

$$E_{RSS} = \frac{\sum_{i}^{n}(y_i - h(\mathbf{x}_i))^2}{\sum_{i}^{n}(y_i - \overline{y})^2}$$

where $\overline{y}$ is the sample mean

Don’t just rely on numbers, visualize!



For all 4 sets: same mean and variance for x, same mean and variance (almost) for y, and same regression line and correlation between x and y (and therefore same R-squared).

Summary of Linear Regression Models

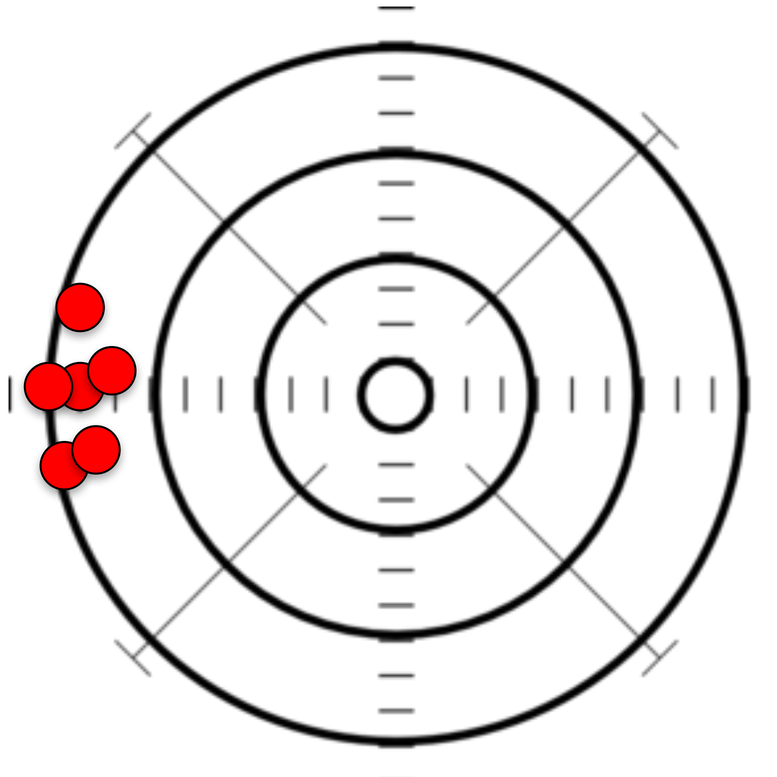
- Easily understood
- Interpretable
- Well studied by statisticians
- Computationally efficient
- Can handle non-linear situations if formulated properly
- Bias/variance tradeoff (occurs in all machine learning)
- Visualize!!
- GLMs

Appendix

(Stuff I couldn't cover in class)

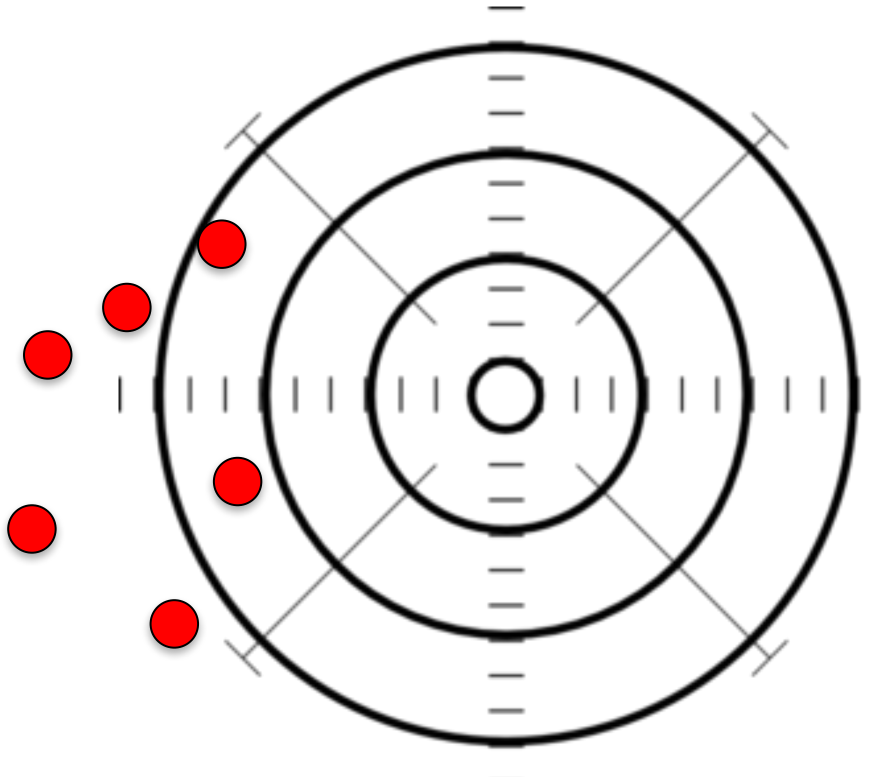
Bias and Variance

high bias, low variance



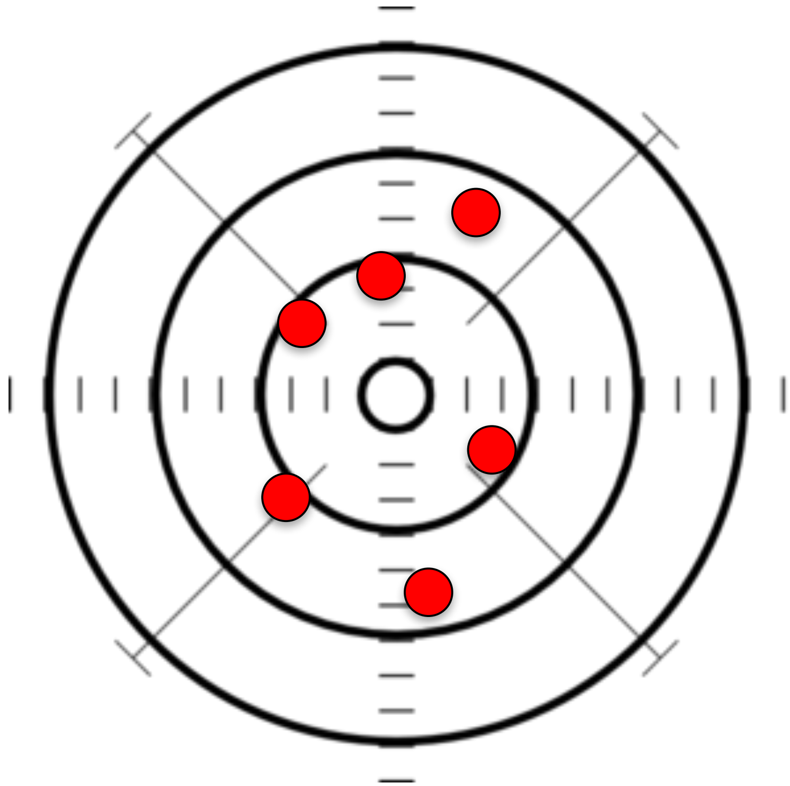
Bias and Variance

high bias, high variance



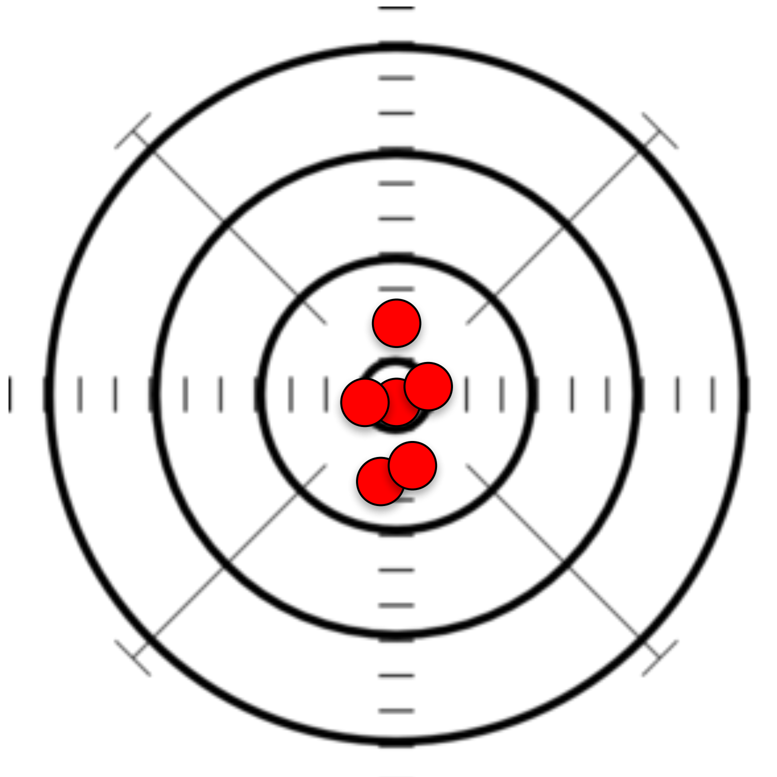
Bias and Variance

low bias, high variance



Bias and Variance

low bias, low variance



Bias and Variance

- Bias:

  Measures how much *h(x)* is wrong disregarding the effect of varying samples

  high bias ➡ underfitting

- Variance:

  Measures how much *h(x)* fluctuate around the expected value as the sample varies.

  high variance ➡ overfitting

There's a trade-off between bias and variance

Ways to Avoid Overfitting

- Simpler model
  - E.g. fewer parameters
- Regularization
  - penalize for complexity in objective function
- Fewer features
- Dimensionality reduction of features (e.g. PCA)
- More data…

Model Selection

- **Cross-validation**: Measure generalization accuracy by testing on data unused during training
- **Regularization**: Penalize complex models
  E'=error on data + λ model complexity
  Akaike's information criterion (AIC), Bayesian information criterion (BIC)
- **Minimum description length (MDL)**: Kolmogorov complexity, shortest description of data
- **Structural risk minimization (SRM)**

Generalized Linear Models

- Models shown have assumed that the response variable follows a Gaussian distribution around the mean

- Can be generalized to response variables that take on *any* exponential family distribution (Generalized Linear Models - GLMs)