CS6220: DATA MINING TECHNIQUES

Matrix Data: Prediction

Instructor: Yizhou Sun

yzsun@ccs.neu.edu

January 19, 2016

Announcements

- Team formation due next Wednesday
- Homework 1 out by tomorrow

Today's Schedule

- Course Project Introduction
- Linear Regression Model
- Decision Tree

Methods to Learn

	Matrix Data	Text Data	Set Data	Sequence Data	Time Series	Graph & Network	Images
Classification	**Decision Tree**; Naïve Bayes; Logistic Regression SVM; kNN			HMM		Label Propagation	Neural Network
Clustering	K-means; hierarchical clustering; DBSCAN; Mixture Models; kernel k-means*	PLSA				SCAN; Spectral Clustering	
Frequent Pattern Mining			Apriori; FP-growth	GSP; PrefixSpan			
Prediction	**Linear Regression**				Autoregression	Collaborative Filtering	
Similarity Search					DTW	P-PageRank	
Ranking						PageRank	

How to learn these algorithms?

- Three levels
  - When it is applicable?
    - Input, output, strengths, weaknesses, time complexity
  - How it works?
    - Pseudo-code, work flows, major steps
    - Can work out a toy problem by pen and paper
  - Why it works?
    - Intuition, philosophy, objective, derivation, proof

Matrix Data: Prediction

- Matrix Data
- Linear Regression Model
- Model Evaluation and Selection
- Summary

Example

	Sex	Race	Height	Income	Marital Status	Years of Educ.	Liberal-ness
R1001	M	1	70	50	1	12	1.73
R1002	M	2	72	100	2	20	4.53
R1003	F	1	55	250	1	16	2.99
R1004	M	2	65	20	2	16	1.13
R1005	F	1	60	10	3	12	3.81
R1006	M	1	68	30	1	9	4.76
R1007	F	5	66	25	2	21	2.01
R1008	F	4	61	43	1	18	1.27
R1009	M	1	69	67	1	12	3.25

A matrix of $\mathrm{n} \times p$:

- n data objects / points
- p attributes / dimensions

$$\begin{bmatrix} x_{11} & \cdots & x_{1f} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{if} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nf} & \cdots & x_{np} \end{bmatrix}$$

Attribute Type

- Numerical
  - E.g., height, income
- Categorical / discrete
  - E.g., Sex, Race

Categorical Attribute Types

- **Nominal:** categories, states, or "names of things"
  - *Hair_color* = {*auburn, black, blond, brown, grey, red, white*}
  - marital status, occupation, ID numbers, zip codes
- **Binary**
  - Nominal attribute with only 2 states (0 and 1)
  - Symmetric binary: both outcomes equally important
    - e.g., gender
  - Asymmetric binary: outcomes not equally important.
    - e.g., medical test (positive vs. negative)
    - Convention: assign 1 to most important outcome (e.g., HIV positive)
- **Ordinal**
  - Values have a meaningful order (ranking) but magnitude between successive values is not known.
  - *Size* = {*small, medium, large*}, grades, army rankings

Matrix Data: Prediction

- Matrix Data
- Linear Regression Model
- Model Evaluation and Selection
- Summary

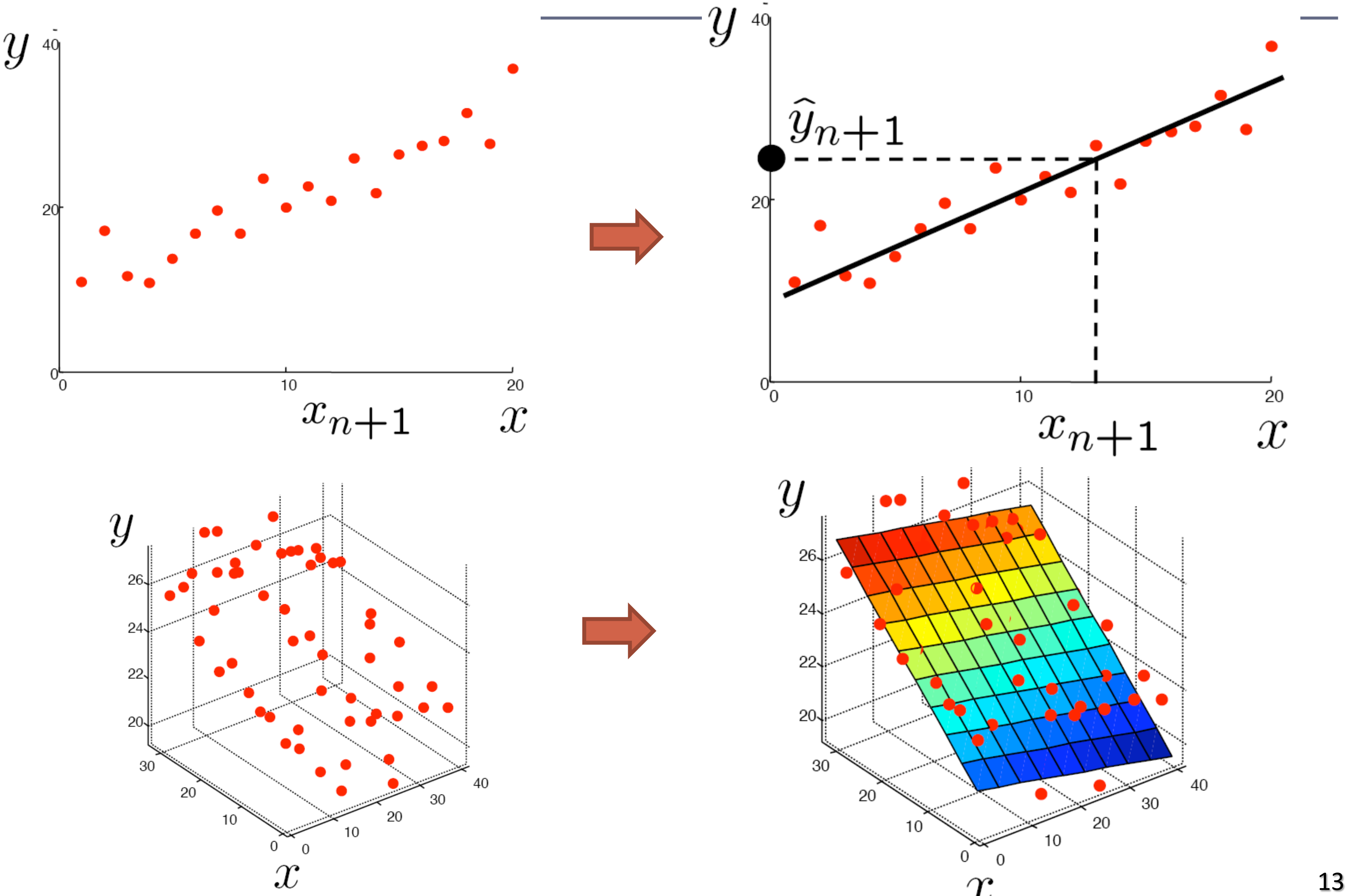
Linear Regression

- Ordinary Least Square Regression
  - Closed form solution
  - Gradient descent
- Linear Regression with Probabilistic Interpretation

The Linear Regression Problem

- Any Attributes to Continuous Value: $\mathbf{x} \Rightarrow y$
  - {age; major ; gender; race} ⇒ GPA
  - {income; credit score; profession} ⇒ loan
  - {college; major ; GPA} ⇒ future income
  - ...

Illustration



Formalization

- Data: n independent data objects
  - $y_i, \mathrm{i} = 1, \dots, n$
  - $\boldsymbol{x}_i = \left(x_{i0}, x_{i1}, x_{i2}, \dots, x_{ip}\right)^{\mathrm{T}}, \mathrm{i} = 1, \dots, n$
    - A constant factor is added to model the bias term, $\mathrm{i.e.}, x_{i0} = 1$
- Model:
  - y: *dependent variable*
  - $\boldsymbol{x}$: *explanatory variables*
  - $\boldsymbol{\beta} = \left(\beta_0, \beta_1, \dots, \beta_p\right)^T$: *weight vector*
  - $y = \boldsymbol{x}^T\boldsymbol{\beta} = \beta_0 + x_1\beta_1 + x_2\beta_2 + \cdots + x_p\beta_p$

A 2-step Process

- Model Construction
  - Use training data to find the best parameter $\boldsymbol{\beta}$, denoted as $\widehat{\boldsymbol{\beta}}$
- Model Usage
  - Model Evaluation
    - Use validation data to select the best model
      - Feature selection
  - Apply the model to the unseen data (test data): $\hat{y} = \boldsymbol{x}^T \widehat{\boldsymbol{\beta}}$

Least Square Estimation

- Cost function (Total Square Error):
  - $J(\boldsymbol{\beta}) = \sum_i \left(\boldsymbol{x}_i^T \boldsymbol{\beta} - y_i\right)^2$
- Matrix form:
  - $J(\boldsymbol{\beta}) = (\mathrm{X}\boldsymbol{\beta} - \boldsymbol{y})^T (X\boldsymbol{\beta} - \boldsymbol{y})$

    $or \left|\left|\mathrm{X}\boldsymbol{\beta} - \boldsymbol{y}\right|\right|^2$

$$\begin{bmatrix} 1, x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ 1, x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ 1, x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix} \qquad \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix}$$

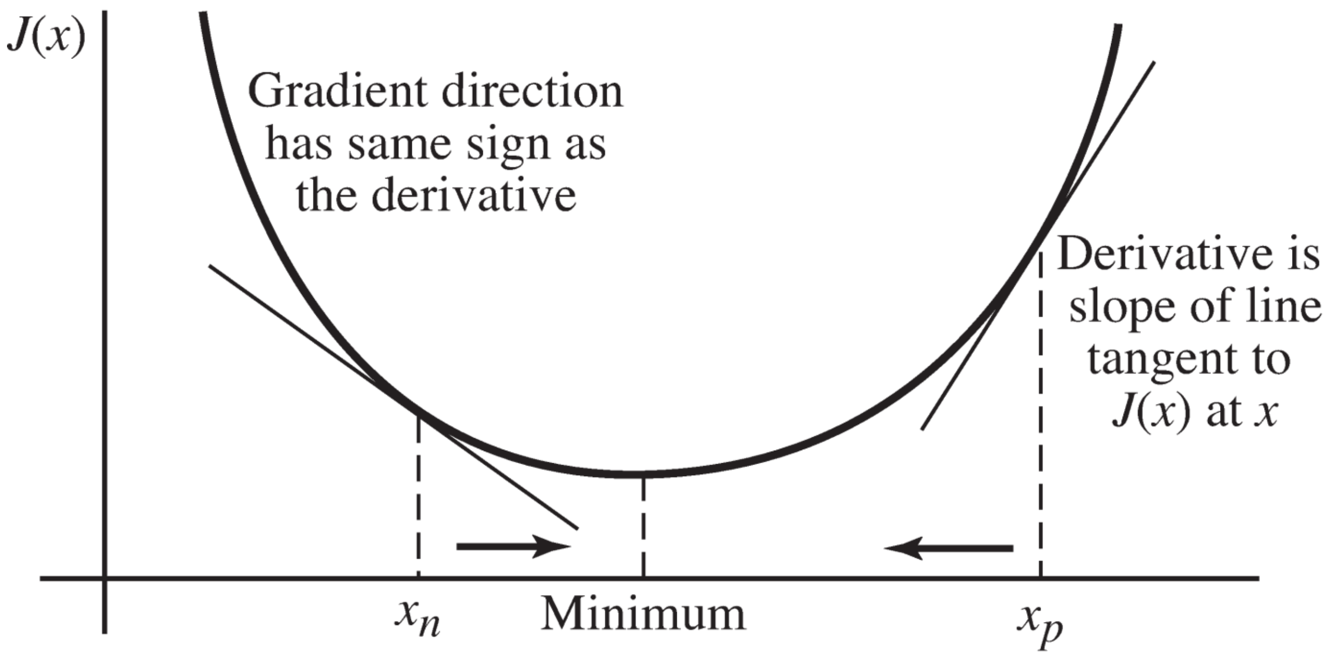
$\boldsymbol{X}$: $\boldsymbol{n} \times (\boldsymbol{p} + \mathbf{1})$ **matrix** **y**: $\boldsymbol{n} \times \mathbf{1}$ **vector**

Ordinary Least Squares (OLS)

- Goal: find $\widehat{\boldsymbol{\beta}}$ that minimizes $J(\boldsymbol{\beta})$
  - $J(\boldsymbol{\beta}) = (\mathrm{X}\boldsymbol{\beta} - y)^T(X\boldsymbol{\beta} - y)$
    $= \boldsymbol{\beta}^T X^T X\boldsymbol{\beta} - y^T X\boldsymbol{\beta} - \boldsymbol{\beta}^T X^T y + y^T y$
- Ordinary least squares
  - Set first derivative of $J(\boldsymbol{\beta})$ as 0
  - $\frac{\partial J}{\partial \boldsymbol{\beta}} = 2\boldsymbol{\beta}^T \mathrm{X}^\mathrm{T}\mathrm{X} - 2y^T X = 0$
  - $\Rightarrow \widehat{\boldsymbol{\beta}} = \left(X^T X\right)^{-1} X^T y$

Gradient Descent

- Minimize the cost function by moving down in the steepest direction



Arrows point in minus gradient direction towards the minimum

Batch Gradient Descent

- Move in the direction of steepest descend

Repeat until converge {

$$\boldsymbol{\beta}^{(t+1)} := \boldsymbol{\beta}^{(\mathrm{t})} - \eta \frac{\partial J}{\partial \boldsymbol{\beta}} |_{\boldsymbol{\beta} = \boldsymbol{\beta}^{(\mathrm{t})}} , \qquad \text{e.g., } \eta = 0.1$$

}

Where $J(\boldsymbol{\beta}) = \sum_i \left(\boldsymbol{x}_i^T \boldsymbol{\beta} - y_i\right)^2 = \sum_i J_i(\boldsymbol{\beta})$ *and*

$$\frac{\partial J}{\partial \boldsymbol{\beta}} = \sum_i \frac{\partial J_i}{\partial \boldsymbol{\beta}} = \sum_i 2\boldsymbol{x}_i \ (\boldsymbol{x}_i^T \boldsymbol{\beta} - y_i)$$

Stochastic Gradient Descent

- When a new observation, *i*, comes in, update weight immediately (extremely useful for large-scale datasets):

Repeat {

for i=1:n {

$$\boldsymbol{\beta}^{(t+1)} := \boldsymbol{\beta}^{(\mathrm{t})} + 2\eta(y_i - \boldsymbol{x}_i^T \boldsymbol{\beta}^{(t)})\boldsymbol{x}_i$$

}

}

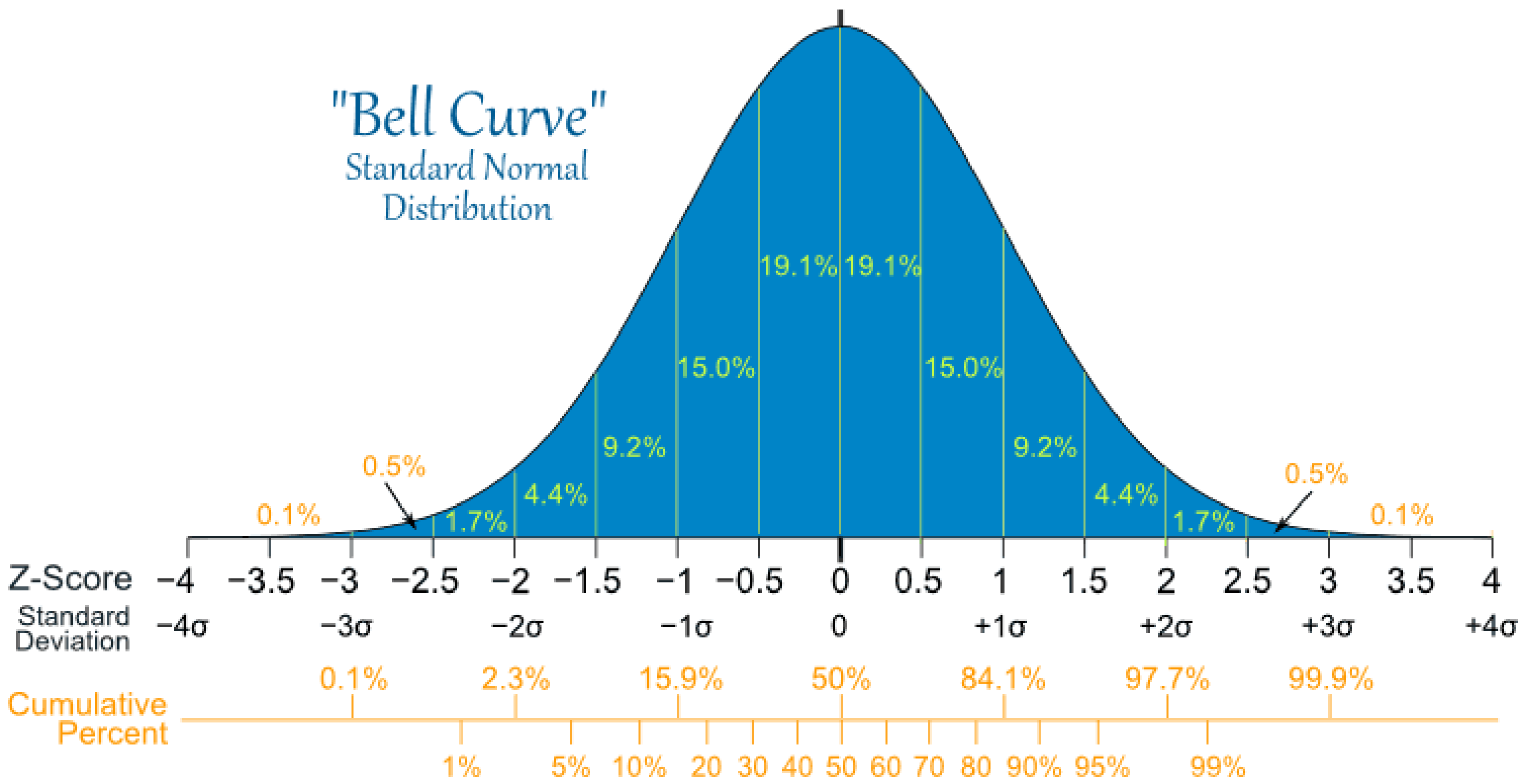
If the prediction for object *i* is smaller than the real value, $\boldsymbol{\beta}$ should move forward to the direction of $\boldsymbol{x}_i$

Other Practical Issues

- What if $X^T X$ is not invertible?
  - Add a small portion of identity matrix, λI, to it (ridge regression*) $\sum_i (y_i - \mathbf{x}_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$
- What if some attributes are categorical?
  - Set dummy variables
    - E.g., $x = 1, if\ sex = F; x = 0, if\ sex = M$
    - Nominal variable with multiple values?
      - Create more dummy variables for one variable
- What if non-linear correlation exists?
  - Transform features, say, x to x^2

Probabilistic Interpretation

- Review of normal distribution
  - $X \sim N(\mu, \sigma^2) \Rightarrow f(X = x) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$



Probabilistic Interpretation

- Model: $y_i = x_i^T \beta + \varepsilon_i$
  - $\varepsilon_i \sim N(0, \sigma^2)$
  - $y_i | x_i, \beta \sim N(x_i^T \beta, \sigma^2)$
    - $E(y_i | x_i) = x_i^T \beta$
- Likelihood:
  - $L(\boldsymbol{\beta}) = \prod_i p(y_i | x_i, \beta)$

$$= \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{(y_i - \boldsymbol{x}_i^T \boldsymbol{\beta})^2}{2\sigma^2}\}$$

- Maximum Likelihood Estimation
  - find $\widehat{\boldsymbol{\beta}}$ that maximizes $\mathrm{L}(\boldsymbol{\beta})$
  - $\arg\max L = \arg\min J$, Equivalent to OLS!

Matrix Data: Prediction

- Matrix Data
- Linear Regression Model
- Model Evaluation and Selection
- Summary

Model Selection Problem

- Basic problem:
  - how to choose between competing linear regression models

- Model too simple:
  - “underfit” the data; poor predictions; high bias; low variance
- Model too complex:
  - “overfit” the data; poor predictions; low bias; high variance
- Model just right:
  - balance bias and variance to get good predictions

Bias and Variance

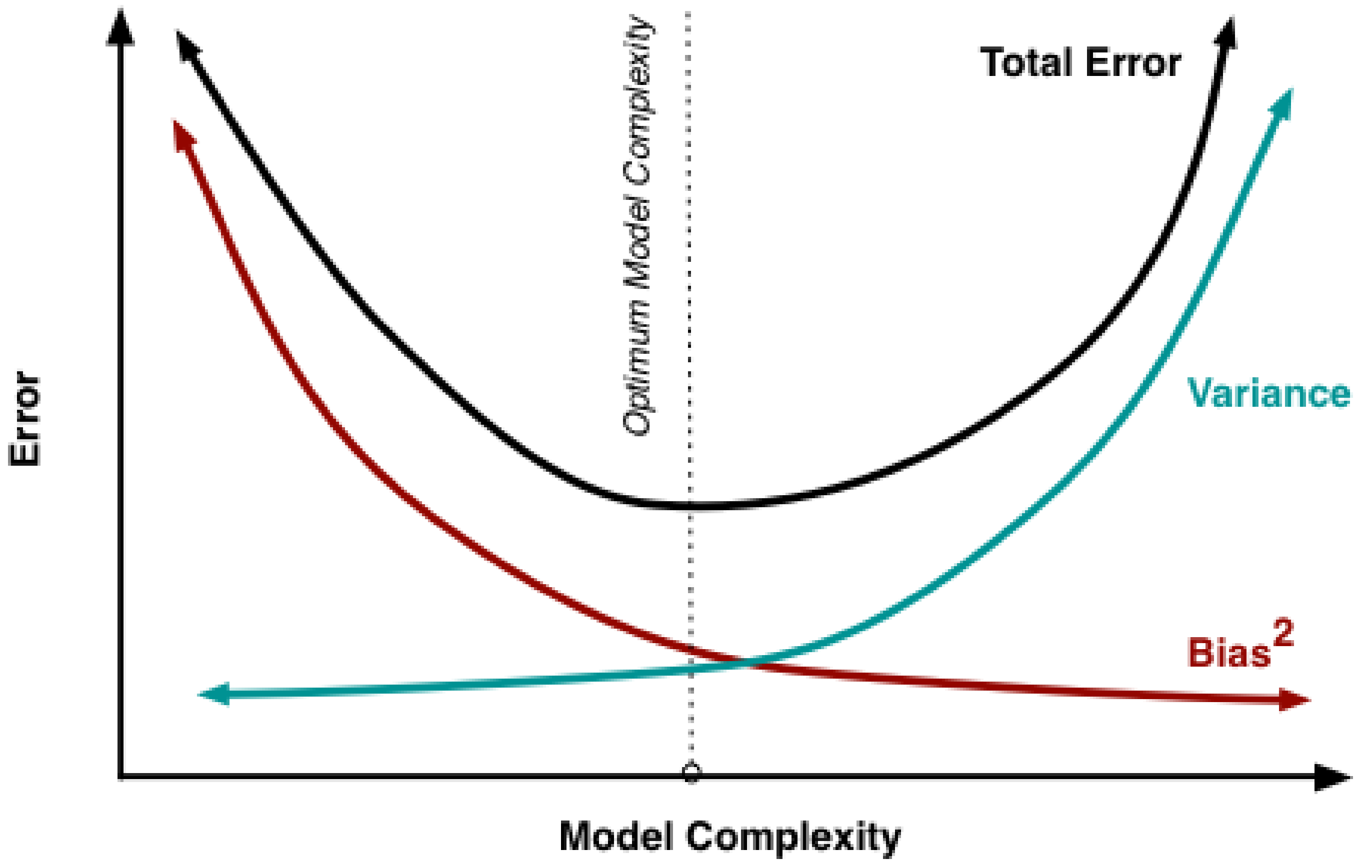
True predictor $f(x)$: $x^T\boldsymbol{\beta}$

Estimated predictor $\hat{f}(x)$: $x^T\widehat{\boldsymbol{\beta}}$

- Bias: $E(\hat{f}(x)) - f(x)$
  - How far away is the expectation of the estimator to the true value? The smaller the better.
- Variance: $Var\left(\hat{f}(x)\right) = E[\left(\hat{f}(x) - E\left(\hat{f}(x)\right)\right)^2]$
  - How variant is the estimator? The smaller the better.
- Reconsider mean square error
  - $J(\widehat{\boldsymbol{\beta}})/n = \sum_i (\boldsymbol{x}_i^T\widehat{\boldsymbol{\beta}} - y_i)^2/n$
  - Can be considered as
    - $E[(\hat{f}(x) - f(x) - \varepsilon)^2] = bias^2 + variance + noise$

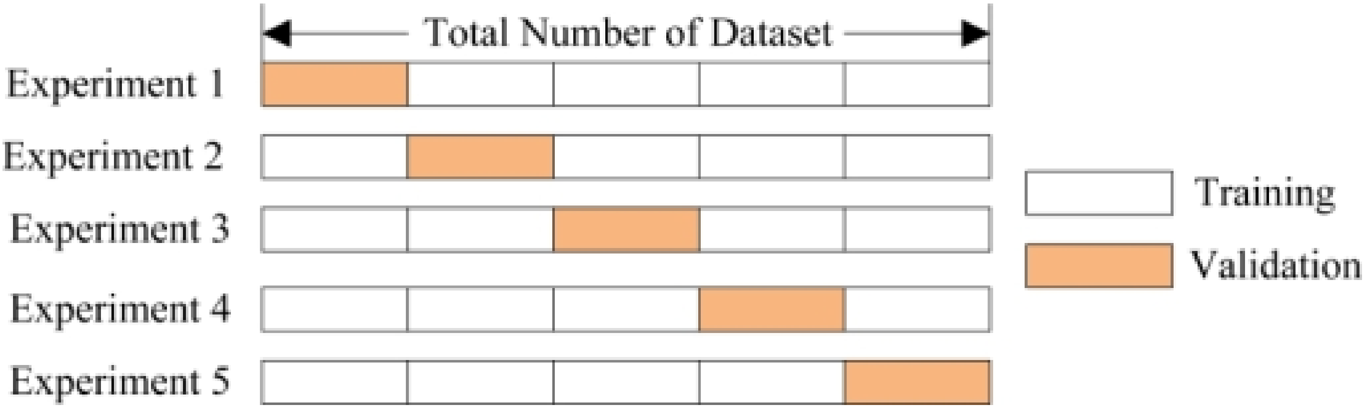
Note $E(\varepsilon) = 0, Var(\varepsilon) = \sigma^2$

Bias-Variance Trade-off



Cross-Validation

- Partition the data into K folds
  - Use K-1 fold as training, and 1 fold as testing
  - Calculate the average accuracy best on K training-testing pairs
    - Accuracy on validation/test dataset!
      - **Mean square error** can again be used: $\sum_i \left(\boldsymbol{x}_i^T \widehat{\boldsymbol{\beta}} - y_i\right)^2 / n$



AIC & BIC*

- AIC and BIC can be used to test the quality of statistical models
  - AIC (Akaike information criterion)
    - $AIC = 2k - 2\ln(\hat{L})$,
    - where k is the number of parameters in the model and $\hat{L}$ is the likelihood under the estimated parameter
  - BIC (Bayesian Information criterion)
    - B$IC = kln(n) - 2\ln(\hat{L})$,
    - Where n is the number of objects

Stepwise Feature Selection

- Avoid brute-force selection
  - 2^p
- Forward selection
  - Starting with the best single feature
  - Always add the feature that improves the performance best
  - Stop if no feature will further improve the performance
- Backward elimination
  - Start with the full model
  - Always remove the feature that results in the best performance enhancement
  - Stop if removing any feature will get worse performance

Matrix Data: Prediction

- Matrix Data
- Linear Regression Model
- Model Evaluation and Selection
- Summary

Summary

- What is matrix data?
  - Attribute types
- Linear regression
  - OLS
  - Probabilistic interpretation
- Model Evaluation and Selection
  - Bias-Variance Trade-off
  - Mean square error
  - Cross-validation, AIC, BIC, step-wise feature selection