Linear Regression and the Bias Variance Tradeoff

Guest Lecturer
Joseph E. Gonzalez

slides available here: http://tinyurl.com/reglecture
Simple Linear Regression

Linear Model: \( Y = mX + b \)

- **Response Variable**
- **Covariate**
- **Slope**
- **Intercept (bias)**
Motivation

• One of the most widely used techniques
• Fundamental to many larger models
  – Generalized Linear Models
  – Collaborative filtering
• Easy to interpret
• Efficient to solve
Multiple Linear Regression
The Regression Model

• For a single data point \((x, y)\):

  \[
  p(x, y) = p(x)p(y|x)
  \]

  Independent Variable (Vector) \(x \in \mathbb{R}^p\)

  Response Variable (Scalar) \(y \in \mathbb{R}\)

  Observe: (Condition)
END DUALITY GAP

BAYSIANS AGAINST DISCRIMINATION

SUPPORT VECTOR MACHINES
The Linear Model

\[ y = \theta^T x + \epsilon + b \]

Vector of Parameters
Vector of Covariates

Scalar Response

Linear Combination of Covariates
\[ \sum_{i=1}^{p} \theta_i x_i \]

Real Value Noise

Noise Model:
\[ \epsilon \sim N(0, \sigma^2) \]

What about bias/intercept term?

Define: \( x_{p+1} = 1 \)

Then redefine \( p := p+1 \) for notational simplicity
Conditional Likelihood \( p(y|x) \)

- Conditioned on \( x \):
  \[
y = \theta^T x + \epsilon \sim N(0, \sigma^2)
\]

- Conditional distribution of \( Y \):
  \[
  Y \sim N(\theta^T x, \sigma^2)
  \]

\[
p(y|x) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left( - \frac{(y - \theta^T x)^2}{2\sigma^2} \right)
\]
Parameters and Random Variables

\[ y \sim N(\theta^T x, \sigma^2) \]

- Conditional distribution of \( y \):
  - Bayesian: parameters as random variables
    \[ p(y|x, \theta, \sigma^2) \]
  - Frequentist: parameters as (unknown) constants
    \[ p_{\theta, \sigma^2}(y|x) \]
So far ...

I’m lonely
Independent and Identically Distributed (iid) Data

• For \( n \) data points:

\[
D = \{(x_1, y_1), \ldots, (x_n, y_n)\} = \{(x_i, y_i)\}_{i=1}^n
\]

Plate Diagram

**Independent Variable (Vector)**

\[ x_i \in \mathbb{R}^p \]

\[ i \in \{1, \ldots, n\} \]

**Response Variable (Scalar)**

\[ y_i \in \mathbb{R} \]
Joint Probability

- For $n$ data points independent and identically distributed (iid):

$$p(D) = \prod_{i=1}^{n} p(x_i, y_i)$$

$$= \prod_{i=1}^{n} p(x_i) p(y_i | x_i)$$
Rewriting with Matrix Notation

• Represent data $\mathcal{D} = \{ (x_i, y_i) \}_{i=1}^n$ as:

\[
    D = \{ (x_i, y_i) \}_{i=1}^n
\]

\[
    X = \begin{pmatrix}
        x_1 \\
        x_2 \\
        \vdots \\
        x_n
    \end{pmatrix} \in \mathbb{R}^{np}
\]

\[
    Y = \begin{pmatrix}
        y_1 \\
        y_2 \\
        \vdots \\
        y_n
    \end{pmatrix} \in \mathbb{R}^{n}
\]

Assume $X$ has rank $p$ (not degenerate)
Rewriting with Matrix Notation

• Rewriting the model using matrix operations:

\[ Y = X\theta + \epsilon \]
Estimating the Model

• Given data how can we estimate $\theta$?

\[ Y = X\theta + \epsilon \]

• Construct maximum likelihood estimator (MLE):
  – Derive the log-likelihood
  – Find $\theta_{\text{MLE}}$ that maximizes log-likelihood
    • Analytically: Take derivative and set $= 0$
    • Iteratively: (Stochastic) gradient descent
Joint Probability

• For $n$ data points:

$$p(D) = \prod_{i=1}^{n} p(x_i, y_i)$$

$$= \prod_{i=1}^{n} p(x_i) p(y_i | x_i)$$

Discriminative Model
Defining the Likelihood

\[ p_\theta(y|x) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{(y - \theta^T x)^2}{2\sigma^2} \right) \]

\[ \mathcal{L}(\theta|D) = \prod_{i=1}^{n} p_\theta(y_i|x_i) \]

\[ = \prod_{i=1}^{n} \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{(y_i - \theta^T x_i)^2}{2\sigma^2} \right) \]

\[ = \frac{1}{\sigma^n (2\pi)^{n/2}} \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \theta^T x_i)^2 \right) \]
Maximizing the Likelihood

- Want to compute:

\[
\hat{\theta}_{\text{MLE}} = \arg \max_{\theta \in \mathbb{R}^p} \mathcal{L}(\theta | D)
\]

- To simplify the calculations we take the log:

\[
\hat{\theta}_{\text{MLE}} = \arg \max_{\theta \in \mathbb{R}^p} \log \mathcal{L}(\theta | D)
\]

which does not affect the maximization because log is a monotone function.
\[ \mathcal{L}(\theta | \mathcal{D}) = \frac{1}{\sigma^n (2\pi)^{n/2}} \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \theta^T x_i)^2 \right) \]

- Take the log:

\[ \log \mathcal{L}(\theta | \mathcal{D}) = - \log(\sigma^n (2\pi)^{n/2}) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \theta^T x_i)^2 \]

- Removing constant terms with respect to \( \theta \):

\[ \log \mathcal{L}(\theta) = - \sum_{i=1}^{n} (y_i - \theta^T x_i)^2 \]

Monotone Function
(Easy to maximize)
\[
\log \mathcal{L}(\theta) = -\sum_{i=1}^{n}(y_i - \theta^T x_i)^2
\]

- Want to compute:

\[
\hat{\theta}_{\text{MLE}} = \arg\max_{\theta \in \mathbb{R}^p} \log \mathcal{L}(\theta|\mathcal{D})
\]

- Plugging in log-likelihood:

\[
\hat{\theta}_{\text{MLE}} = \arg\max_{\theta \in \mathbb{R}^p} -\sum_{i=1}^{n}(y_i - \theta^T x_i)^2
\]
\[ \hat{\theta}_{\text{MLE}} = \arg \max_{\theta \in \mathbb{R}^p} - \sum_{i=1}^{n} (y_i - \theta^T x_i)^2 \]

- Dropping the sign and flipping from maximization to minimization:

\[ \hat{\theta}_{\text{MLE}} = \arg \min_{\theta \in \mathbb{R}^p} \sum_{i=1}^{n} (y_i - \theta^T x_i)^2 \]

**Minimize Sum (Error)²**

- Gaussian Noise Model \(\rightarrow\) Squared Loss
  - Least Squares Regression
Pictorial Interpretation of Squared Error
Maximizing the Likelihood
(Minimizing the Squared Error)

\[ \hat{\theta}_{\text{MLE}} = \arg \min_{\theta \in \mathbb{R}^p} \sum_{i=1}^{n} (y_i - \theta^T x_i)^2 \]

- Take the gradient and set it equal to zero

Convex Function

- \[ -\log \mathcal{L}(\theta) \]

Slope = 0

\[ \hat{\theta}_{\text{MLE}} \]
Minimizing the Squared Error

\[ \hat{\theta}_{\text{MLE}} = \arg \min_{\theta \in \mathbb{R}^p} \sum_{i=1}^{n} (y_i - \theta^T x_i)^2 \]

- Taking the gradient

\[-\nabla_\theta \log \mathcal{L}(\theta) = \nabla_\theta \sum_{i=1}^{n} (y_i - \theta^T x_i)^2\]

Chain Rule \(\rightarrow\)

\[= -2 \sum_{i=1}^{n} (y_i - \theta^T x_i) x_i\]

\[= -2 \sum_{i=1}^{n} y_i x_i + 2 \sum_{i=1}^{n} (\theta^T x_i) x_i\]
• Rewriting the gradient in matrix form:

\[-\nabla_\theta \log \mathcal{L}(\theta) = -2 \sum_{i=1}^{n} y_i x_i + 2 \sum_{i=1}^{n} (\theta^T x_i) x_i\]

\[= -2X^T Y + 2X^T X \theta\]

• To make sure the log-likelihood is convex compute the second derivative (Hessian)

\[-\nabla^2 \log \mathcal{L}(\theta) = 2X^T X\]

• If \(X\) is full rank then \(X^T X\) is positive definite and therefore \(\theta_{MLE}\) is the minimum

  – Address the degenerate cases with regularization
\[-\nabla_\theta \log \mathcal{L}(\theta) = -2X^Ty + 2X^TX\theta = 0\]

• Setting gradient equal to 0 and solve for \(\theta_{\text{MLE}}\):

\[
(X^TX)\hat{\theta}_{\text{MLE}} = X^TY
\]

\[
\hat{\theta}_{\text{MLE}} = (X^TX)^{-1}X^TY
\]
Geometric Interpretation

• View the MLE as finding a projection on col(X)
  – Define the estimator:
    \[ \hat{Y} = X \theta \]
  – Observe that \( \hat{Y} \) is in col(X)
    • linear combination of cols of X
  – Want to \( \hat{Y} \) closest to Y
• Implies \( (Y - \hat{Y}) \) normal to X

\[
X^T(Y - \hat{Y}) = X^T(Y - X \theta) = 0
\]
\[
\Rightarrow X^T X \theta = X^T Y
\]
Connection to Pseudo-Inverse

\[ \hat{\theta}_\text{MLE} = (X^T X)^{-1} X^T Y \]

Moore-Penrose Pseudoinverse

• Generalization of the inverse:
  – Consider the case when X is square and invertible:
  
  \[ X^\dagger = (X^T X)^{-1} X^T = X^{-1} (X^T)^{-1} X^T = X^{-1} \]
  
  – Which implies \( \theta_{\text{MLE}} = X^{-1} Y \) the solution to \( X \theta = Y \) when X is square and invertible
Computing the MLE

\[ \hat{\theta}_{\text{MLE}} = (X^T X)^{-1} X^T Y \]

- **Not** typically solved by inverting \( X^T X \)
- Solved using direct methods:
  - Cholesky factorization:
    - Up to a factor of 2 faster
  - QR factorization:
    - More numerically stable

  \text{or use the built-in solver in your math library.} \quad \text{R: solve(Xt %*% X, Xt %*% y)}

- Solved using various iterative methods:
  - Krylov subspace methods
  - (Stochastic) Gradient Descent

http://www.seas.ucla.edu/~vandenbe/103/lectures/qr.pdf
Cholesky Factorization

\[
\text{solve } \quad (X^T X) \hat{\theta}_{\text{MLE}} = X^T Y
\]

\[
\hat{\theta}_{\text{MLE}} \quad C \quad d
\]

- Compute symm. matrix \( C = X^T X \) \( O(np^2) \)
- Compute vector \( d = X^T Y \) \( O(np) \)
- Cholesky Factorization \( LL^T = C \) \( O(p^3) \)
  - \( L \) is lower triangular
- Forward subs. to solve: \( Lz = d \) \( O(p^2) \)
- Backward subs. to solve: \( L^T \hat{\theta}_{\text{MLE}} = z \) \( O(p^2) \)

Connections to graphical model inference:
http://ssg.mit.edu/~willsky/publ_pdfs/185_pub_MLR.pdf and
http://yaroslavvb.blogspot.com/2011/02/junction-trees-in-numerical-analysis.html with illustrations
Solving Triangular System

\[
\begin{bmatrix}
A_{11} & A_{12} & A_{13} & A_{14} \\
A_{22} & A_{23} & A_{24} & \\
A_{33} & A_{34} & \\
A_{44} & \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
\end{bmatrix}
=
\begin{bmatrix}
b_1 \\
b_2 \\
b_3 \\
b_4 \\
\end{bmatrix}
\]
### Solving Triangular System

<table>
<thead>
<tr>
<th></th>
<th>$A_{11}x_1$</th>
<th>$A_{12}x_2$</th>
<th>$A_{13}x_3$</th>
<th>$A_{14}x_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{22}x_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{33}x_3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$A_{44}x_4$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$x_1 = \frac{b_1 - A_{12}x_2 - A_{13}x_3 - A_{14}x_4}{A_{11}}$

$x_2 = \frac{b_2 - A_{23}x_3 - A_{24}x_4}{A_{22}}$

$x_3 = \frac{(b_3 - A_{34}x_4)}{A_{33}}$

$x_4 = \frac{b_4}{A_{44}}$
Distributed Direct Solution (Map-Reduce)

\[ \hat{\theta}_{\text{MLE}} = (X^T X)^{-1} X^T Y \]

- Distribution computations of sums:
  
  \[ C = X^T X = \sum_{i=1}^{n} x_i x_i^T \quad O(np^2) \]
  
  \[ d = X^T y = \sum_{i=1}^{n} x_i y_i \quad O(np) \]

- Solve system \( C \theta_{\text{MLE}} = d \) on master.
  
  \[ O(p^3) \]
Gradient Descent:
What if $p$ is large? (e.g., $n/2$)

- The cost of $O(np^2) = O(n^3)$ could be prohibitive
- Solution: Iterative Methods
  - Gradient Descent:
    
    For $\tau$ from 0 until convergence
    
    $$\theta^{(\tau+1)} = \theta^{(\tau)} - \rho(\tau) \nabla \log \mathcal{L}(\theta^{(\tau)} | D)$$

Learning rate
Gradient Descent Illustrated:

\[ -\log \mathcal{L}(\theta) \]

Convex Function

\[ \theta^{(0)} \quad \theta^{(2)} \quad \theta^{(3)} \quad \theta^{(1)} \]

Slope = 0

\[ \theta^{(3)} = \hat{\theta}_{\text{MLE}} \]
Gradient Descent:
What if $p$ is large? (e.g., $n/2$)

- The cost of $O(np^2) = O(n^3)$ could be prohibitive
- Solution: Iterative Methods
  - Gradient Descent:

For $\tau$ from 0 until convergence

$$
\begin{align*}
\theta^{(\tau+1)} &= \theta^{(\tau)} - \rho(\tau) \nabla \log \mathcal{L}(\theta^{(\tau)} | D) \\
&= \theta^{(\tau)} + \rho(\tau) \frac{1}{n} \sum_{i=1}^{n} (y_i - \theta^{(\tau)T} x_i)x_i \\
&= O(np)
\end{align*}
$$

- Can we do better?
Stochastic Gradient Descent

• Construct noisy estimate of the gradient:

For $\tau$ from 0 until convergence
1) pick a random $i$
2) $\theta^{(\tau+1)} = \theta^{(\tau)} + \rho(\tau)(y_i - \theta^{(\tau)^T} x_i)x_i$ $O(p)$

• Sensitive to choice of $\rho(\tau)$ typically ($\rho(\tau)=1/\tau$)
• Also known as Least-Mean-Squares (LMS)
• Applies to streaming data $O(p)$ storage
Fitting Non-linear Data

• What if Y has a non-linear response?

• Can we still use a linear model?
Transforming the Feature Space

• Transform features $x_i$

\[ x_i = (X_{i,1}, X_{i,2}, \ldots, X_{i,p}) \]

• By applying non-linear transformation $\phi$: \[ \phi : \mathbb{R}^p \rightarrow \mathbb{R}^k \]

• Example:

\[ \phi(x) = \{1, x, x^2, \ldots, x^k\} \]

– others: splines, radial basis functions, ...
– Expert engineered features (modeling)
Under-fitting

\{1.\}

\{1., x\}

\{1., x, x^2, x^3\}

\{1., x, x^2, x^3, x^4, x^5\}

Over-fitting
Really Over-fitting!

\[ \{1, x, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}, x^{11}, x^{12}, x^{13}, x^{14}\} \]

- Errors on training data are small
- But errors on new points are likely to be large
What if I train on different data?

Low Variability:

High Variability
Bias-Variance Tradeoff

- So far we have minimized the error (loss) with respect to **training data**
  - Low training error does not imply good expected performance: **over-fitting**
- We would like to reason about the **expected loss (Prediction Risk)** over:
  - Training Data: \{ (y_1, x_1), ..., (y_n, x_n) \}
  - Test point: \( (y_*, x_*) \)
- We will decompose the expected loss into:
  \[
  E_{D,(y_*,x_*)} \left[ \left( y_* - f(x_*|D) \right)^2 \right] = \text{Noise} + \text{Bias}^2 + \text{Variance}
  \]
• Define (unobserved) the true model \((h)\):

\[
y_\star = h(x_\star) + \epsilon_\star
\]

• Completed the squares with: \(h(x_\star) = h_\star\)

\[
E_D, (y_\star, x_\star) \left[ (y_\star - f(x_\star|D))^2 \right]
\]

Expected Loss

\[
= \mathbb{E}_{D, (y_\star, x_\star)} \left[ (y_\star - h(x_\star) + h(x_\star) - f(x_\star|D))^2 \right]
\]

\[
(a + b)^2 = a^2 + b^2 + 2ab
\]

\[
= \mathbb{E}_{\epsilon_\star} \left[ (y_\star - h(x_\star))^2 \right] + \mathbb{E}_D \left[ (h(x_\star) - f(x_\star|D))^2 \right]
\]

\[
+ 2\mathbb{E}_D, (y_\star, x_\star) \left[ y_\star h_\star - y_\star f_\star - h_\star h_\star + h_\star f_\star \right]
\]
• Define (unobserved) the true model \((h)\):

\[ y_* = h(x_*) + \epsilon_* \]

• Completed the squares with: \(h(x_*) = h_*\)

\[
\begin{align*}
\mathbb{E}_D,(y_*,x_*) \left[ (y_* - f(x_*|D))^2 \right] \quad &\text{Expected Loss} \\
= \mathbb{E}_D,(y_*,x_*) \left[ (y_* - h(x_*) + h(x_*) - f(x_*|D))^2 \right] \\
= \mathbb{E}_{\epsilon_*} \left[ (y_* - h(x_*))^2 \right] + \mathbb{E}_D \left[ (h(x_*) - f(x_*|D))^2 \right] \\
&+ 2\mathbb{E}_D,(y_*,x_*) \left[ y_* h_* - y_* f_* - h_* h_* + h_* f_* \right] \\
\end{align*}
\]

Substitute defn. \(y_* = h_* + \epsilon_*\)

\[
\begin{align*}
\mathbb{E} \left[ (h_* + \epsilon_*)h_* - (h_* + \epsilon_*) f_* - h_* h_* + h_* f_* \right] = \\
\end{align*}
\]
• Define (unobserved) the true model ($h$):

$$y_* = h(x_*) + \epsilon_*$$

• Completed the squares with: $h(x_*) = h_*$

$$E_D (y_*, x_*) \left[ (y_* - f(x_* | D))^2 \right]$$ 

Expected Loss

$$= E_D (y_*, x_*) \left[ (y_* - h(x_*) + h(x_*) - f(x_* | D))^2 \right]$$

$$= E_{\epsilon_*} \left[ (y_* - h(x_*)]^2 \right] + E_D \left[ (h(x_*) - f(x_* | D))^2 \right]$$

Noise Term
(out of our control)

Model Estimation Error
(we want to minimize this)

• Minimum error is governed by the noise.
• Expanding on the model estimation error:
  \[ E_D \left[ (h(x_*) - f(x_*|D))^2 \right] \]

• Completing the squares with \( E[f(x_*|D)] = \tilde{f}_* \)

\[
E_D \left[ (h(x_*) - f(x_*|D))^2 \right] \\
= E \left[ (h(x_*) - E [f(x_*|D)] + E [f(x_*|D)] - f(x_*|D))^2 \right] \\
= E \left[ (h(x_*) - E [f(x_*|D)])^2 \right] + E \left[ (f(x_*|D) - E [f(x_*|D)])^2 \right] \\
+ 2E \left[ h_*\tilde{f}_* - h_*f_* - \tilde{f}_*f_* + \tilde{f}_*^2 \right] \\
\]

\[
= h_*\tilde{f}_* - h_*E [f_*] - \tilde{f}_*E [f_*] + \tilde{f}_*^2 = \\
h_*\tilde{f}_* - h_*\tilde{f}_* - \tilde{f}_*\tilde{f}_* + \tilde{f}_*^2 = 0
\]
• Expanding on the model estimation error:

\[ \mathbb{E}_D \left[ (h(x_*) - f(x_*|D))^2 \right] \]

• Completing the squares with \( \mathbb{E}[f(x_*|D)] = \bar{f}_* \)

\[
\begin{align*}
\mathbb{E}_D \left[ (h(x_*) - f(x_*|D))^2 \right] &= \mathbb{E} \left[ (h(x_*) - \mathbb{E}[f(x_*|D)])^2 \right] + \mathbb{E} \left[ (f(x_*|D) - \mathbb{E}[f(x_*|D)])^2 \right] \\
&= (h(x_*) - \mathbb{E}[f(x_*|D)])^2
\end{align*}
\]
• Expanding on the model estimation error:
  \[E_D [(h(x_*) - f(x_*|D))^2]\]

• Completing the squares with \(E[f(x_*|D)] = \bar{f}_*\)

\[E_D [(h(x_*) - f(x_*|D))^2] = (h(x_*) - E[f(x_*|D)])^2 + E[(f(x_*|D) - E[f(x_*|D)])^2]\]

  (Bias)\(^2\)  Variance

• Tradeoff between bias and variance:
  – **Simple Models**: High Bias, Low Variance
  – **Complex Models**: Low Bias, High Variance
Summary of Bias Variance Tradeoff

\[ E_D, (y_*, x_*) \left[ (y_* - f(x_*|D))^2 \right] = \]
\[ E_{\epsilon_*} \left[ (y_* - h(x_*))^2 \right] + (h(x_*) - E_D [f(x_*|D)])^2 \]
\[ + E_D \left[ (f(x_*|D) - E_D [f(x_*|D)])^2 \right] \]

• Choice of models balances bias and variance.
  – Over-fitting ➔ Variance is too High
  – Under-fitting ➔ Bias is too High
Bias Variance Plot

Image from http://scott.fortmann-roe.com/docs/BiasVariance.html
Assume a true model is linear: \( h(x_*) = x^T \theta \)

\[
\text{bias} = h(x_*) - \mathbb{E}_D [f(x_*|D)] \\
= x^T \theta - \mathbb{E}_D \left[ x^T \hat{\theta}_{\text{MLE}} \right] \\
= x^T \theta - \mathbb{E}_D \left[ x^T (X^T X)^{-1} X^T Y \right] \\
= x^T \theta - \mathbb{E}_D \left[ x^T (X^T X)^{-1} X^T (X \theta + \epsilon) \right] \\
= x^T \theta - \mathbb{E}_D \left[ x^T (X^T X)^{-1} X^T X \theta + x^T (X^T X)^{-1} X^T \epsilon \right] \\
= x^T \theta - \mathbb{E}_D \left[ x^T \theta + x^T (X^T X)^{-1} X^T \epsilon \right] \\
= x^T \theta - x^T \theta + x^T (X^T X)^{-1} X^T \mathbb{E}_D [\epsilon] \\
= x^T \theta - x^T \theta = 0 \\
\hat{\theta}_{\text{MLE}} \text{ is unbiased!}
\]
Analyze Variance of $f(x_* | D) = x_*^T \hat{\theta}_{\text{MLE}}$

- Assume a true model is linear: $h(x_*) = x_*^T \theta$

\[
\text{Var.} = \mathbb{E} \left[ (f(x_* | D) - \mathbb{E}_D [f(x_* | D)])^2 \right] \\
= \mathbb{E} \left[ (x_*^T \hat{\theta}_{\text{MLE}} - x_*^T \theta)^2 \right] \\
= \mathbb{E} \left[ (x_*^T (X^T X)^{-1} X^T Y - x_*^T \theta)^2 \right] \\
= \mathbb{E} \left[ (x_*^T (X^T X)^{-1} X^T (X \theta + \epsilon) - x_*^T \theta)^2 \right] \\
= \mathbb{E} \left[ (x_*^T \theta + x_*^T (X^T X)^{-1} X^T \epsilon - x_*^T \theta)^2 \right] \\
= \mathbb{E} \left[ (x_*^T (X^T X)^{-1} X^T \epsilon)^2 \right]
\]

- Use property of scalar: $a^2 = a a^T$
Analyze Variance of \( f(x_*|D) = x_*^T \hat{\theta}_{\text{MLE}} \)

- Use property of scalar: \( a^2 = a a^T \)

\[
\text{Var.} = E \left[ (f(x_*|D) - E_D [f(x_*|D)])^2 \right] \\
= E \left[ (x_*^T (X^T X)^{-1} X^T \epsilon)^2 \right] \\
= E \left[ (x_*^T (X^T X)^{-1} X^T \epsilon) (x_*^T (X^T X)^{-1} X^T \epsilon)^T \right] \\
= E \left[ x_*^T (X^T X)^{-1} X^T \epsilon \epsilon^T (x_*^T (X^T X)^{-1} X^T)^T \right] \\
= x_*^T (X^T X)^{-1} X^T E \left[ \epsilon \epsilon^T \right] (x_*^T (X^T X)^{-1} X^T)^T \\
= x_*^T (X^T X)^{-1} X^T \sigma_{\epsilon}^2 I (x_*^T (X^T X)^{-1} X^T)^T \\
= \sigma_{\epsilon}^2 x_*^T (X^T X)^{-1} X^T X (x_*^T (X^T X)^{-1})^T \\
= \sigma_{\epsilon}^2 x_*^T (x_*^T (X^T X)^{-1})^T \\
= \sigma_{\epsilon}^2 x_*^T (X^T X)^{-1} x_*
Consequence of Variance Calculation

\[
\text{Var. } = \mathbb{E} \left[ (f(x^*|D) - \mathbb{E}_D [f(x^*|D)])^2 \right] \\
= \sigma^2 \epsilon x^*_T (X^T X)^{-1} x^*_y
\]

Figure from http://people.stern.nyu.edu/wgreene/MathStat/GreeneChapter4.pdf
Summary

• Least-Square Regression is Unbiased:

$$E_D \left[ x_*^{T} \hat{\theta}_{MLE} \right] = x_*^{T} \theta$$

• Variance depends on:

$$E \left[ \left( f(x_*|D) - E[f(x_*|D)] \right)^2 \right] = \sigma_\epsilon^2 x_*^{T} (X^T X)^{-1} x_*$$

$$\approx \sigma_\epsilon^2 \frac{p}{n}$$

– Number of data-points $n$

– Dimensionality $p$

– Not on observations $Y$
Deriving the final identity

- Assume \( x_i \) and \( x_* \) are N(0,1)

\[
\mathbb{E}_{X,x_*} \text{ [Var.]} = \sigma_\varepsilon^2 \mathbb{E}_{X,x_*} \left[ x_*^T (X^T X)^{-1} x_* \right] \\
= \sigma_\varepsilon^2 \mathbb{E}_{X,x_*} \left[ tr(x_* x_*^T (X^T X)^{-1}) \right] \\
= \sigma_\varepsilon^2 tr(\mathbb{E}_{X,x_*} \left[ x_* x_*^T (X^T X)^{-1} \right]) \\
= \sigma_\varepsilon^2 tr(\mathbb{E}_{x_*} \left[ x_* x_*^T \right] \mathbb{E}_X \left[ (X^T X)^{-1} \right]) \\
= \frac{\sigma_\varepsilon^2}{n} tr(\mathbb{E}_{x_*} \left[ x_* x_*^T \right]) \\
= \frac{\sigma_\varepsilon^2}{n} p
\]
Gauss-Markov Theorem

- The linear model:

\[ f(x_*) = x_*^T \hat{\theta}_{MLE} = x_*^T (X^T X)^{-1} X^T Y \]

has the **minimum variance** among all **unbiased** linear estimators

- Note that this is linear in Y

- **BLUE:** Best Linear Unbiased Estimator
Summary

• Introduced the Least-Square regression model
  – Maximum Likelihood: Gaussian Noise
  – Loss Function: Squared Error
  – Geometric Interpretation: Minimizing Projection

• Derived the normal equations:
  – Walked through process of constructing MLE
  – Discussed efficient computation of the MLE

• Introduced basis functions for non-linearity
  – Demonstrated issues with over-fitting

• Derived the classic bias-variance tradeoff
  – Applied to least-squares model
Additional Reading I found Helpful

- http://people.stern.nyu.edu/wgreene/MathStat/GreeneChapter4.pdf