Unit 3 Regression mathematical Fundamentals

Kiran Bagale

July 2025

1. Verification of Least Squares Properties with Numerical Example

Let the system be:

$$\mathbf{d} = X\mathbf{w} + \boldsymbol{\varepsilon}$$

with:

$$X = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \boldsymbol{\varepsilon} = \begin{bmatrix} 0.1 \\ -0.2 \\ 0.1 \end{bmatrix}$$

Then:

$$\mathbf{d} = X\mathbf{w} + \boldsymbol{\varepsilon} = \begin{bmatrix} 1 \cdot 1 + 2 \cdot 2 \\ 2 \cdot 1 + 1 \cdot 2 \\ 1 \cdot 1 + 1 \cdot 2 \end{bmatrix} + \begin{bmatrix} 0.1 \\ -0.2 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ 3 \end{bmatrix} + \begin{bmatrix} 0.1 \\ -0.2 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 5.1 \\ 3.8 \\ 3.1 \end{bmatrix}$$

A. Least Squares Estimate

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

Compute:

$$X^T X = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 5 \\ 5 & 6 \end{bmatrix}$$

$$X^T \mathbf{d} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 5.1 \\ 3.8 \\ 3.1 \end{bmatrix} = \begin{bmatrix} 5.1 + 7.6 + 3.1 \\ 10.2 + 3.8 + 3.1 \end{bmatrix} = \begin{bmatrix} 15.8 \\ 17.1 \end{bmatrix}$$

Now:

$$\hat{\mathbf{w}} = \begin{bmatrix} 6 & 5 \\ 5 & 6 \end{bmatrix}^{-1} \begin{bmatrix} 15.8 \\ 17.1 \end{bmatrix}$$

Inverse of symmetric matrix:

$$(X^T X)^{-1} = \frac{1}{6 \cdot 6 - 5 \cdot 5} \begin{bmatrix} 6 & -5 \\ -5 & 6 \end{bmatrix} = \frac{1}{11} \begin{bmatrix} 6 & -5 \\ -5 & 6 \end{bmatrix}$$

Thus:

$$\hat{\mathbf{w}} = \frac{1}{11}\begin{bmatrix} 6 & -5 \\ -5 & 6 \end{bmatrix}\begin{bmatrix} 15.8 \\ 17.1 \end{bmatrix} = \frac{1}{11}\begin{bmatrix} 6 \cdot 15.8 - 5 \cdot 17.1 \\ -5 \cdot 15.8 + 6 \cdot 17.1 \end{bmatrix} = \frac{1}{11}\begin{bmatrix} 94.8 - 85.5 \\ -79 + 102.6 \end{bmatrix} = \frac{1}{11}\begin{bmatrix} 9.3 \\ 23.6 \end{bmatrix} \approx \begin{bmatrix} 0.845 \\ 2.145 \end{bmatrix}$$

B. Un-biasedness: $E[\hat{\mathbf{w}}] = \mathbf{w}$ if $E[\boldsymbol{\varepsilon}] = 0$

We assume $E[\boldsymbol{\varepsilon}] = 0$. The bias is:

$$\text{bias} = \hat{\mathbf{w}} - \mathbf{w} = \begin{bmatrix} 0.845 \\ 2.145 \end{bmatrix} - \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -0.155 \\ 0.145 \end{bmatrix}$$

Small bias is due to non-zero noise. As noise $\to 0$ or with more samples, the bias tends to 0.

Thus, **numerical evidence supports unbiasedness.**

C. Covariance: $\text{Cov}[\hat{\mathbf{w}}] = \sigma^2(X^TX)^{-1}$ for white noise

Let us estimate variance from noise:

$$\hat{\mathbf{d}} = X\hat{\mathbf{w}} = \begin{bmatrix} 0.845 + 4.29 \\ 1.69 + 2.145 \\ 0.845 + 2.145 \end{bmatrix} = \begin{bmatrix} 5.135 \\ 3.835 \\ 2.990 \end{bmatrix}$$

$$\mathbf{e} = \mathbf{d} - \hat{\mathbf{d}} = \begin{bmatrix} 5.1 - 5.135 \\ 3.8 - 3.835 \\ 3.1 - 2.99 \end{bmatrix} = \begin{bmatrix} -0.035 \\ -0.035 \\ 0.11 \end{bmatrix}$$

Estimate noise variance:

$$\hat{\sigma}^2 = \frac{\mathbf{e}^T\mathbf{e}}{N - M} = \frac{(-0.035)^2 + (-0.035)^2 + 0.11^2}{3 - 2} = \frac{0.001225 + 0.001225 + 0.0121}{1} = 0.01455$$

Then:

$$\text{Cov}[\hat{\mathbf{w}}] = 0.01455 \cdot \frac{1}{11}\begin{bmatrix} 6 & -5 \\ -5 & 6 \end{bmatrix} \approx \begin{bmatrix} 0.00795 & -0.00661 \\ -0.00661 & 0.00795 \end{bmatrix}$$

So we verified the covariance formula numerically.

D. Orthogonality: $(\mathbf{d} - X\hat{\mathbf{w}})^T X\hat{\mathbf{w}} = 0$

Let:

$$\mathbf{e} = \mathbf{d} - X\hat{\mathbf{w}} = \begin{bmatrix} -0.035 \\ -0.035 \\ 0.11 \end{bmatrix}, \quad X\hat{\mathbf{w}} = \begin{bmatrix} 5.135 \\ 3.835 \\ 2.99 \end{bmatrix}$$

Check inner product:

$$\mathbf{e}^T X\hat{\mathbf{w}} = (-0.035)(5.135) + (-0.035)(3.835) + (0.11)(2.99) \approx -0.1797 - 0.1342 + 0.3289 \approx 0.015$$

This is approximately zero (within numerical rounding), verifying:

$$(\mathbf{d} - X\hat{\mathbf{w}})^T X\hat{\mathbf{w}} \approx 0$$

Orthogonality is verified numerically.

2. Verification of Identity: $\left(\mathbf{w}^T\mathbf{x}\right)^2 = \mathbf{w}^T\left(\mathbf{x}\mathbf{x}^T\right)\mathbf{w}$

Given:

Let:

$$\mathbf{w} = \begin{bmatrix}2\\3\end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix}4\\5\end{bmatrix}$$

Step 1: Compute $\mathbf{w}^T\mathbf{x}$

$$\mathbf{w}^T\mathbf{x} = \begin{bmatrix}2 & 3\end{bmatrix}\begin{bmatrix}4\\5\end{bmatrix} = 2\cdot 4 + 3\cdot 5 = 8 + 15 = 23$$

$$\Rightarrow (\mathbf{w}^T\mathbf{x})^2 = 23^2 = 529$$

Step 2: Compute $\mathbf{x}\mathbf{x}^T$

$$\mathbf{x}\mathbf{x}^T = \begin{bmatrix}4\\5\end{bmatrix}\begin{bmatrix}4 & 5\end{bmatrix} = \begin{bmatrix}16 & 20\\20 & 25\end{bmatrix}$$

Step 3: Compute $\mathbf{w}^T(\mathbf{x}\mathbf{x}^T)\mathbf{w}$

$$\mathbf{w}^T(\mathbf{x}\mathbf{x}^T)\mathbf{w} = \begin{bmatrix}2 & 3\end{bmatrix}\begin{bmatrix}16 & 20\\20 & 25\end{bmatrix}\begin{bmatrix}2\\3\end{bmatrix}$$

First compute the matrix-vector product:

$$\begin{bmatrix}16 & 20\\20 & 25\end{bmatrix}\begin{bmatrix}2\\3\end{bmatrix} = \begin{bmatrix}16\cdot 2 + 20\cdot 3\\20\cdot 2 + 25\cdot 3\end{bmatrix} = \begin{bmatrix}32 + 60\\40 + 75\end{bmatrix} = \begin{bmatrix}92\\115\end{bmatrix}$$

Then multiply with the transpose of $\mathbf{w}$:

$$\begin{bmatrix}2 & 3\end{bmatrix}\begin{bmatrix}92\\115\end{bmatrix} = 2\cdot 92 + 3\cdot 115 = 184 + 345 = 529$$

Conclusion:

$$(\mathbf{w}^T\mathbf{x})^2 = \mathbf{w}^T(\mathbf{x}\mathbf{x}^T)\mathbf{w} = 529$$

Identity is verified.

3. Differentiation of the Quadratic Form $\mathbf{w}^T\mathbf{R}_{xx}\mathbf{w}$

Let $J(\mathbf{w}) = \mathbf{w}^T\mathbf{R}_{xx}\mathbf{w}$, where $\mathbf{R}_{xx} \in \mathbb{R}^{n\times n}$ is a symmetric matrix.

Objective:

Find the gradient:

$$\frac{d}{d\mathbf{w}}\left(\mathbf{w}^T\mathbf{R}_{xx}\mathbf{w}\right)$$

Matrix Calculus Identity:

For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, the derivative of the quadratic form is:

$$\frac{d}{d\mathbf{w}}\left(\mathbf{w}^T A\mathbf{w}\right) = 2A\mathbf{w}$$

Apply the Identity:

Since $\mathbf{R}_{xx}$ is symmetric:

$$\frac{d}{d\mathbf{w}}\left(\mathbf{w}^T\mathbf{R}_{xx}\mathbf{w}\right) = 2\mathbf{R}_{xx}\mathbf{w}$$

Intuition (Optional):

$$\mathbf{w}^T\mathbf{R}_{xx}\mathbf{w} = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i R_{ij} w_j$$

Differentiating with respect to w_k gives two terms due to symmetry:

$$\frac{\partial}{\partial w_k}\sum_{i,j} w_i R_{ij} w_j = 2\sum_{j} R_{kj} w_j = 2(\mathbf{R}_{xx}\mathbf{w})_k$$

Final Result:

$$\frac{d}{d\mathbf{w}}\left(\mathbf{w}^T\mathbf{R}_{xx}\mathbf{w}\right) = 2\mathbf{R}_{xx}\mathbf{w}$$

4. Time-Averaged Auto-correlation and Cross-correlation Example

Let:

- $\mathbf{x}(n) \in \mathbb{R}^M$: Input vector at time n
- $d(n) \in \mathbb{R}$: Desired signal at time n
- $r_{dx} = E[d(n)\mathbf{x}(n)]$: Cross-correlation vector

- $R_{xx} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$: Auto-correlation matrix

In practice, we often approximate expectations using time averages over N samples.

a. Given Data

Let the signal be observed over $N = 3$ time samples. The input vector $\mathbf{x}(n) \in \mathbb{R}^2$ and desired signal $d(n) \in \mathbb{R}$ are:

$$\mathbf{x}(1) = \begin{bmatrix}1\\2\end{bmatrix}, \quad \mathbf{x}(2) = \begin{bmatrix}2\\1\end{bmatrix}, \quad \mathbf{x}(3) = \begin{bmatrix}1\\1\end{bmatrix}$$

$$d(1) = 3, \quad d(2) = 4, \quad d(3) = 2$$

b. Time-Averaged Cross-Correlation Vector r_{dx}

$$r_{dx} = \frac{1}{N}\sum_{n=1}^{N} d(n)\mathbf{x}(n)$$

$$r_{dx} = \frac{1}{3}\left[3\begin{bmatrix}1\\2\end{bmatrix} + 4\begin{bmatrix}2\\1\end{bmatrix} + 2\begin{bmatrix}1\\1\end{bmatrix}\right] = \frac{1}{3}\left[\begin{bmatrix}3\\6\end{bmatrix} + \begin{bmatrix}8\\4\end{bmatrix} + \begin{bmatrix}2\\2\end{bmatrix}\right] = \frac{1}{3}\begin{bmatrix}13\\12\end{bmatrix} = \begin{bmatrix}\frac{13}{3}\\4\end{bmatrix}$$

c. Time-Averaged Auto-Correlation Matrix R_{xx}

$$R_{xx} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}(n)\mathbf{x}^T(n)$$

$$\begin{aligned}
R_{xx} &= \frac{1}{3}\left[\mathbf{x}(1)\mathbf{x}^T(1) + \mathbf{x}(2)\mathbf{x}^T(2) + \mathbf{x}(3)\mathbf{x}^T(3)\right] \\
&= \frac{1}{3}\left[\begin{bmatrix}1\\2\end{bmatrix}\begin{bmatrix}1 & 2\end{bmatrix} + \begin{bmatrix}2\\1\end{bmatrix}\begin{bmatrix}2 & 1\end{bmatrix} + \begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1 & 1\end{bmatrix}\right] \\
&= \frac{1}{3}\left[\begin{bmatrix}1 & 2\\2 & 4\end{bmatrix} + \begin{bmatrix}4 & 2\\2 & 1\end{bmatrix} + \begin{bmatrix}1 & 1\\1 & 1\end{bmatrix}\right] \\
&= \frac{1}{3}\begin{bmatrix}1+4+1 & 2+2+1\\2+2+1 & 4+1+1\end{bmatrix} = \frac{1}{3}\begin{bmatrix}6 & 5\\5 & 6\end{bmatrix} = \begin{bmatrix}2 & \frac{5}{3}\\\frac{5}{3} & 2\end{bmatrix}
\end{aligned}$$

d. Final Result

- Cross-correlation vector:

$$r_{dx} = \begin{bmatrix}\frac{13}{3}\\4\end{bmatrix}$$

- Auto-correlation matrix:

$$R_{xx} = \begin{bmatrix}2 & \frac{5}{3}\\\frac{5}{3} & 2\end{bmatrix}$$

5.The Wiener-Hopf equation and estimator properties

1 Wiener–Hopf Equation

Given the cost function:

$$J(\mathbf{w}) = E\left[(d - \mathbf{w}^T\mathbf{x})^2\right]$$

The gradient is:

$$\nabla J(\mathbf{w}) = \mathbf{R}_{xx}\mathbf{w} - \mathbf{r}_{dx}$$

Setting the gradient to zero for optimality:

$$\boxed{\mathbf{R}_{xx}\mathbf{w} = \mathbf{r}_{dx}}$$

Numerical Example

Let:

$$\mathbf{x} = \begin{bmatrix}1\\2\end{bmatrix}, \quad d = 4$$

Step 1: Autocorrelation Matrix

$$\mathbf{R}_{xx} = \mathbf{x}\mathbf{x}^T = \begin{bmatrix}1\\2\end{bmatrix}\begin{bmatrix}1 & 2\end{bmatrix} = \begin{bmatrix}1 & 2\\2 & 4\end{bmatrix}$$

Step 2: Cross-correlation Vector

$$\mathbf{r}_{dx} = d \cdot \mathbf{x} = 4 \cdot \begin{bmatrix}1\\2\end{bmatrix} = \begin{bmatrix}4\\8\end{bmatrix}$$

Step 3: Solve the Wiener–Hopf Equation

$$\begin{bmatrix}1 & 2\\2 & 4\end{bmatrix}\begin{bmatrix}w_1\\w_2\end{bmatrix} = \begin{bmatrix}4\\8\end{bmatrix}$$

This leads to:

$$\begin{cases} w_1 + 2w_2 = 4 \\ 2w_1 + 4w_2 = 8 \end{cases}$$

Choose $w_2 = 0$, then $w_1 = 4$, so:

$$\mathbf{w} = \begin{bmatrix}4\\0\end{bmatrix}$$

Verification:

$$\mathbf{R}_{xx}\mathbf{w} = \begin{bmatrix}1 & 2\\2 & 4\end{bmatrix}\begin{bmatrix}4\\0\end{bmatrix} = \begin{bmatrix}4\\8\end{bmatrix} = \mathbf{r}_{dx}$$

Wiener–Hopf equation is numerically verified.

6. Bias-Variance Decomposition Derivation

We start with the average loss function:

$$L_{\text{av}}(f(\mathbf{x}, \mathbf{w}), F(\mathbf{x}, \hat{\mathbf{w}})) = \mathbb{E}_{\mathcal{T}} \left[(\mathbb{E}[d|\mathbf{x}] - F(\mathbf{x}, \mathcal{T}))^2 \right]$$

Let:

- $\mu(\mathbf{x}) = \mathbb{E}[d|\mathbf{x}]$: the true conditional mean.
- $\hat{f}(\mathbf{x}) = F(\mathbf{x}, \mathcal{T})$: the output of the learned model.
- $\bar{f}(\mathbf{x}) = \mathbb{E}_{\mathcal{T}}[\hat{f}(\mathbf{x})]$: the expected prediction over all training sets.

We now expand the expression by adding and subtracting $\bar{f}(\mathbf{x})$:

$$\begin{aligned} L_{\text{av}} &= \mathbb{E}_{\mathcal{T}} \left[(\mu(\mathbf{x}) - \hat{f}(\mathbf{x}))^2 \right] \\ &= \mathbb{E}_{\mathcal{T}} \left[(\mu(\mathbf{x}) - \bar{f}(\mathbf{x}) + \bar{f}(\mathbf{x}) - \hat{f}(\mathbf{x}))^2 \right] \\ &= (\mu(\mathbf{x}) - \bar{f}(\mathbf{x}))^2 + \mathbb{E}_{\mathcal{T}} \left[(\hat{f}(\mathbf{x}) - \bar{f}(\mathbf{x}))^2 \right] \\ &\quad + 2(\mu(\mathbf{x}) - \bar{f}(\mathbf{x}))\mathbb{E}_{\mathcal{T}}[\bar{f}(\mathbf{x}) - \hat{f}(\mathbf{x})] \end{aligned}$$

Since $\mathbb{E}_{\mathcal{T}}[\bar{f}(\mathbf{x}) - \hat{f}(\mathbf{x})] = 0$, the cross term vanishes. Thus,

$$L_{\text{av}} = \underbrace{(\mu(\mathbf{x}) - \bar{f}(\mathbf{x}))^2}_{\text{Bias}^2} + \underbrace{\mathbb{E}_{\mathcal{T}} \left[(\hat{f}(\mathbf{x}) - \bar{f}(\mathbf{x}))^2 \right]}_{\text{Variance}}$$

7: Bias–Variance Trade-off in Constrained Network Architectures

Statement: A network architecture, constrained through the incorporation of prior knowledge, addresses the bias–variance dilemma by reducing variance at the expense of increased bias.

Solution:

The bias–variance trade-off is a fundamental concept in statistical learning theory, which states that a model's prediction error can be decomposed into three components:

$$\text{Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

- **Bias** refers to the error introduced by approximating a real-world problem, which may be complex, by a simpler model.
- **Variance** refers to the model's sensitivity to fluctuations in the training data.
- **Irreducible error** accounts for noise inherent in the data generation process.

Incorporating prior knowledge into a neural network architecture often involves:

- Restricting the number of parameters.
- Imposing structural constraints (e.g., convolutional layers for image data).
- Encoding domain-specific assumptions or symmetries (e.g., translation invariance).

Impact on Bias and Variance:

- *Reduced variance:* By limiting the model's flexibility, we prevent it from fitting the noise in the training data, which leads to a decrease in variance.
- *Increased bias:* However, the same constraints may prevent the model from capturing the true underlying relationship in the data, which increases the bias.

Conclusion: Constraining the network through prior knowledge leads to a more stable and generalizable model that performs better on unseen data. This approach sacrifices some level of model accuracy (bias) to gain robustness (lower variance), thereby addressing the bias–variance dilemma in a principled way.