Introduction to Machine Learning

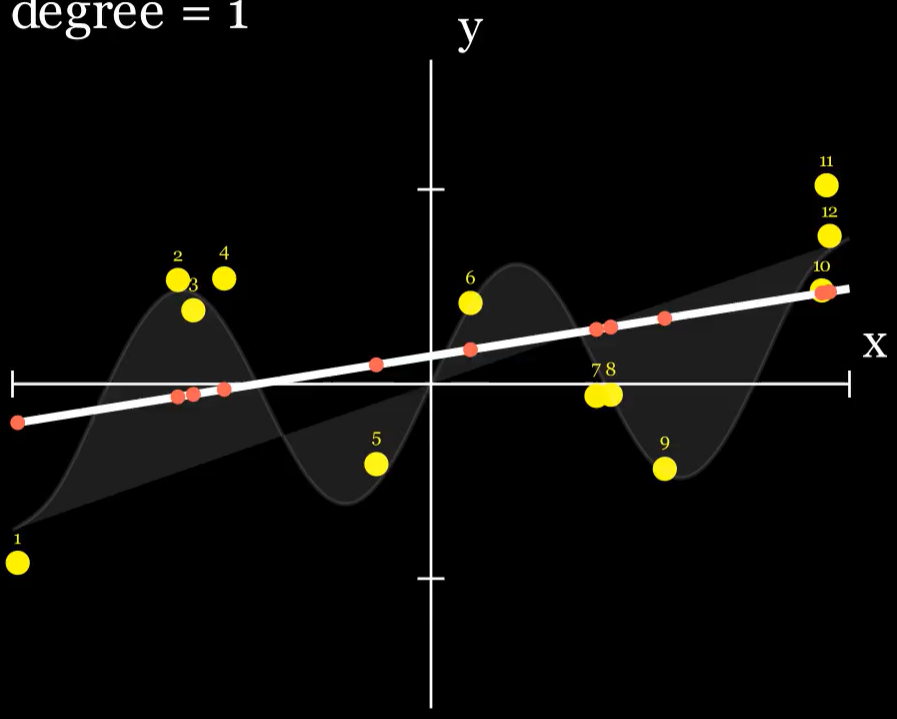
Lecture 9:

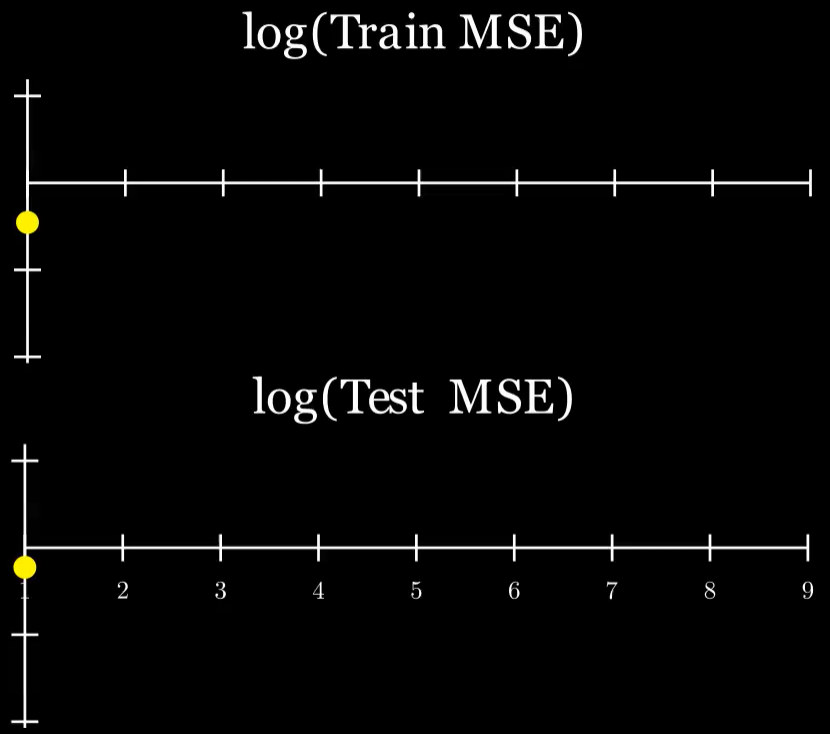
Linear Regression

Prof. Subrahmanya Swamy Peruru
Department of Electrical Engineering
IIT Kanpur

Recap

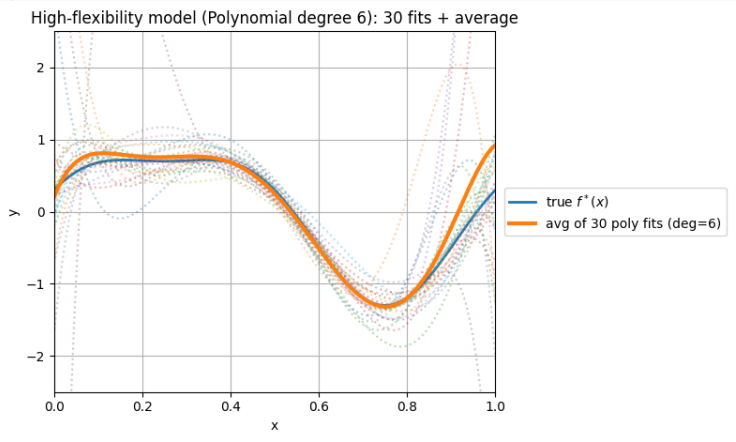
Train MSE: 0.351558
Test MSE: 0.595714

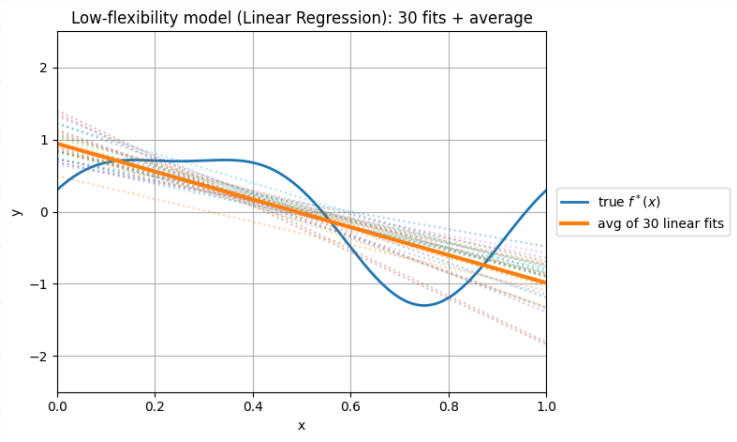




Bias–Variance Tradeoff (Demo)

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





Less flexible models: higher bias, lower variance (**underfitting**)

Highly flexible models: lower bias, higher variance (**overfitting**)

Data Preprocessing

Handling Non-Numeric Features in Data

Two types of categorical data:

Nominal: No natural order

Examples:

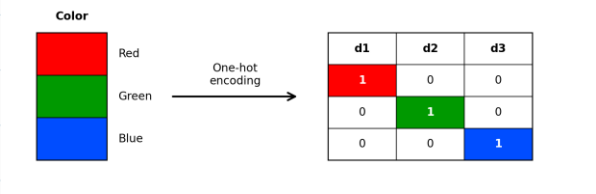
- Gender (Male / Female),
- Color (Red / Blue / Green)

Ordinal: Have a meaningful **order**

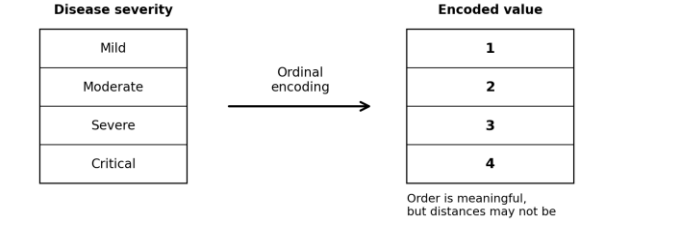
Examples:

- Disease Severity (Mild / Moderate / Severe)
- Course Grades (A / B / C)

One-hot encoding



Ordinal encoding



Feature Scaling

Different **features are measured in very different units.**

Body Temperature: Fahrenheit (~97–102) Annual Income: rupees (~100,000 – 5,000,000)

1. Standardization (Z-score normalization)

$$x_j \leftarrow \frac{x_j - \mu_j}{\sigma_j}$$

Mean: μ_j ,Std Dev: σ_j

Zero mean, unit variance

Commonly used for k-NN, SVMs, linear models

2. Min–Max Scaling

$$x_j \leftarrow \frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)}$$

Scales values to $[0,1]$

Useful when bounded ranges matter

Question: Should scaling be done before or after the train/test split?

Feature Scaling results in meaningful distances

Different **features are measured in very different units.**

- Body Temperature: Fahrenheit (~97–102)
- Annual Income: rupees (~100,000 – 5,000,000)

Without feature scaling, distance-based algorithms (k –NN) can be dominated by large-scale features.

Person	Temperature (°F)	Salary (₹)
A	98	100,000
B	102	100,000
C	98	100,004

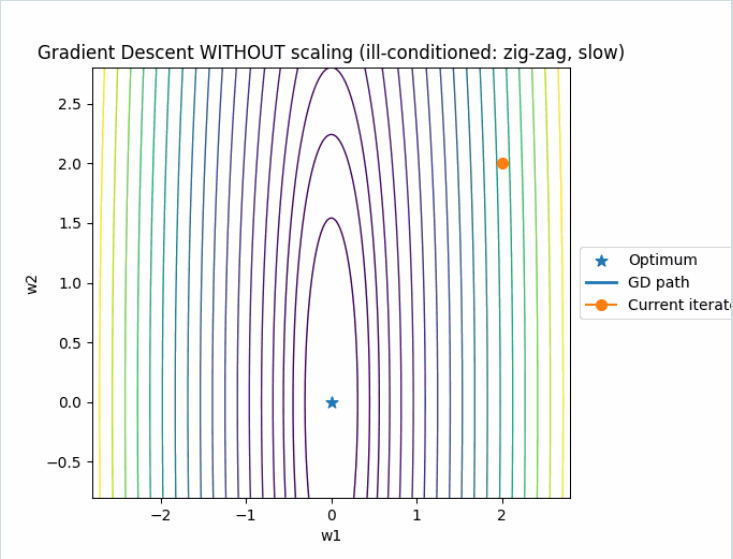
Without Scaling: $d(\mathrm{A},\mathrm{B}) = \mathbf{4},\ d(\mathrm{A},\mathrm{C}) = \mathbf{4}$

With Min-Max Scaling: $d(\mathrm{A},\mathrm{B}) \approx \mathbf{0.8},\ d(\mathrm{A},\mathrm{C}) \approx \mathbf{0.0000008}$

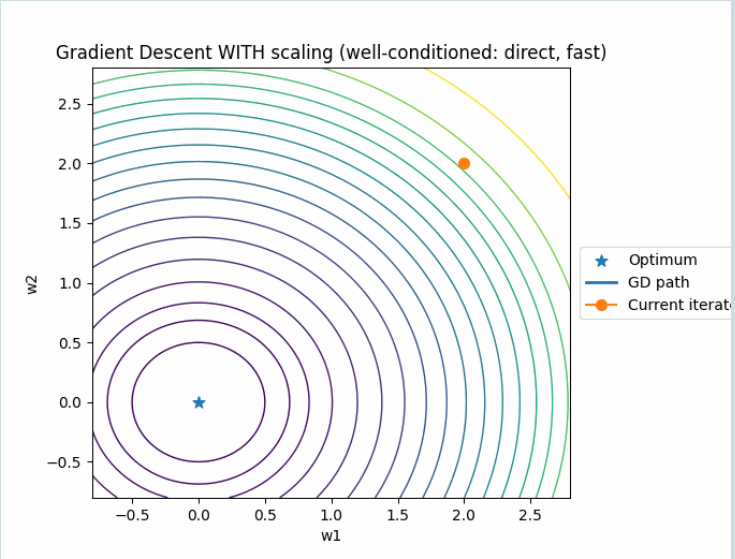
After scaling, a 4-degree temperature change
matters far more than a ₹4 salary change

Feature Scaling Improves Optimization

Without Scaling



With Scaling



$$\underset{f \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} (f(\mathbf{x}_i) - y_i)^2$$

- **Gradient descent** is the backbone of most deep learning algorithms
- **Feature scaling** improves **convergence speed** and **stability**

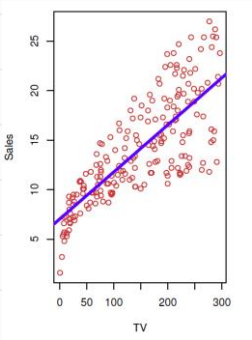
Linear Regression

Linear Regression

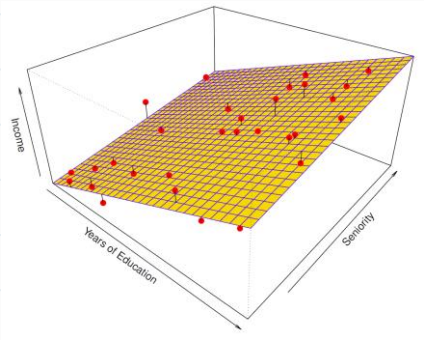
Single feature: $f(\mathbf{x}) = w_1 x_1 + w_2$

Two features: $f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + w_3$

Line



Plane



Generally: $f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots w_{d-1} x_{d-1} + w_d$ **(Hyperplane)**

Task:

Given training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$

Find the unknown parameters $\mathbf{w} = (w_1, w_2, w_3, \ldots, w_d)$ that best fits the data

Figure Source: ISLP Book

Vector Notation

$$f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots w_{d-1} x_{d-1} + w_d$$

$$= \begin{bmatrix} w_1 & w_2 & w_3 & \cdots & w_{d-1} & w_d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{d-1} \\ 1 \end{bmatrix}$$

$$= \mathbf{w}^T \mathbf{x}$$

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{d-1} \\ w_d \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{d-1} \\ 1 \end{bmatrix}$$

A dummy feature

$x_d = 1$

added to simply notation

Task:

Given training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ find the $\mathbf{w}$ that minimizes the loss function

$$\frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}_i) - y_i)^2$$

$$\underset{\mathbf{w}}{\operatorname{argmin}}\, L(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{w}^T\mathbf{x}_i - y_i\right)^2$$

Convexity makes Optimization Easier

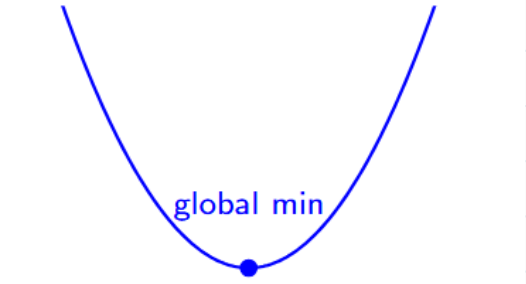
Convex (definition) $g: \mathbb{R}^d \to \mathbb{R}$ is convex if for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $\lambda \in [0,1]$, we have

$$g\left(\lambda \mathbf{x} + (1-\lambda)\mathbf{y}\right) \leq \lambda\, g(\mathbf{x}) + (1-\lambda) g(\mathbf{y})$$

Key fact: $\nabla g(\mathbf{x}^\star) = 0 \Rightarrow x^\star$ is global min (if g is **convex + differentiable**).

Non-convex

$\nabla g(x) = 0$ can be a *local* min/max



Convex

$\nabla g(x) = 0$ means global minimum

Gradient of a function

- The gradient **extends the notion of a derivative** to functions with vector inputs.
- It is obtained by **stacking the partial derivatives** with respect to each component.
- Consider a function $g(\mathbf{w}): \mathbb{R}^d \rightarrow \mathbb{R}$, then gradient $\nabla_{\mathbf{w}}(g(\mathbf{w}))$ is given by

$$\nabla_{\mathbf{w}}(g(\mathbf{w})) = \begin{bmatrix} \frac{\partial \mathrm{g}}{\partial w_1} \\ \frac{\partial g}{\partial w_2} \\ \vdots \\ \frac{\partial g}{\partial w_\mathrm{d}} \end{bmatrix}$$

Example: Gradient of $g(\mathbf{w}) = \mathbf{w}^\top \mathbf{x}$

Consider $g(\mathbf{w}) = \mathbf{w}^\top \mathbf{x}$ where:

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_\mathrm{d} \end{bmatrix}$$

Write the inner product explicitly:

$$\mathbf{w}^\top \mathbf{x} = w_1 x_1 + w_2 x_2 + \cdots + w_\mathrm{d} x_\mathrm{d}$$

Compute partial derivatives:

$$\frac{\partial}{\partial w_1}(\mathbf{w}^\top \mathbf{x}) = x_1, \quad \frac{\partial}{\partial w_2}(\mathbf{w}^\top \mathbf{x}) = x_2, \quad \cdots, \quad \frac{\partial}{\partial w_d}(\mathbf{w}^\top \mathbf{x}) = x_\mathrm{d}$$

Stack partial derivatives (definition of gradient):

$$\nabla_{\mathbf{w}}(\mathbf{w}^\top \mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial w_1}(\mathbf{w}^\top \mathbf{x}) \\ \frac{\partial}{\partial w_2}(\mathbf{w}^\top \mathbf{x}) \\ \vdots \\ \frac{\partial g}{\partial w_\mathrm{d}}(\mathbf{w}^\top \mathbf{x}) \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_\mathrm{d} \end{bmatrix} = \mathbf{x}$$

Few basic algebra and calculus rules

(A1) Inner product symmetry

$$\mathbf{a}^\top \mathbf{b} = \mathbf{b}^\top \mathbf{a}$$

(A2) Linearity of matrix–vector multiplication

$$(A + B)\mathbf{x} = A\mathbf{x} + B\mathbf{x}, \qquad A(\mathbf{x} + \mathbf{y}) = A\mathbf{x} + A\mathbf{y}$$

(A3) Associativity of matrix products

$$A(BC) = (AB)C$$

(C1) Linearity of the gradient

$$\nabla_{\mathbf{w}} \left(\sum_{i=1}^{n} f_i(\mathbf{w}) \right) = \sum_{i=1}^{n} \nabla_{\mathbf{w}} f_i(\mathbf{w})$$

(C2) Chain rule (Example) Consider $z(\mathbf{w})$: $\mathbb{R}^d \to \mathbb{R}$, the gradient of z^2 through chain rule is

$$\nabla_{\mathbf{w}}(z(\mathbf{w})^2) = 2\, z(\mathbf{w}) \times \nabla_{\mathbf{w}} z$$

Least squares: compute the gradient

Objective function

$$L(\mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

Step 1: Linearity of gradient

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = \nabla_{\mathbf{w}} \sum_{i=1}^{n} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2 = \sum_{i=1}^{n} \nabla_{\mathbf{w}} (y_i - \mathbf{w}^\top \mathbf{x}_i)^2$$

Step 2: Chain rule for a square

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = \sum_{i=1}^{n} 2\,(y_i - \mathbf{w}^\top \mathbf{x}_i) \nabla_{\mathbf{w}} (y_i - \mathbf{w}^\top \mathbf{x}_i)$$

Step 3: Use $\nabla_{\mathbf{w}} (y_i - \mathbf{w}^\top \mathbf{x}_i) = -\mathbf{x}_i$

$$\nabla_{\mathbf{w}} L(\mathbf{w}) = \sum_{i=1}^{n} 2\,(y_i - \mathbf{w}^\top \mathbf{x}_i)(-\mathbf{x}_i)$$

Stationary point condition $\nabla_{\mathbf{w}} L(\mathbf{w}) = 0$

Using the gradient from the previous slide:

$$2\sum_{i=1}^{n}(\mathrm{y_i} - \mathbf{w}^\top \mathbf{x}_i)\,(-\mathbf{x}_i) = \mathbf{0}$$

Divide both sides by 2 and rearrange :

$$\sum_{i=1}^{n}(\mathbf{w}^\top \mathbf{x}_i)\,(\mathbf{x}_i) = \sum_{i=1}^{n} y_i\,\mathbf{x}_i$$

Since $\mathbf{w}^\top \mathbf{x}_i$ **is a scalar,** $(\mathbf{w}^\top \mathbf{x}_i)\,\mathbf{x}_i = \mathbf{x}_i(\mathbf{w}^\top \mathbf{x}_i)$

$$\sum_{i=1}^{n}\mathbf{x}_i\,(\mathbf{w}^\top \mathbf{x}_i) = \sum_{i=1}^{n} y_i\,\mathbf{x}_i$$

Since inner product is symmetric $\mathbf{w}^\top \mathbf{x}_i = \mathbf{x}_i^T \boldsymbol{w}$

$$\sum_{i=1}^{n}\mathbf{x}_i\left(\mathbf{x}_i^T\mathbf{w}\right) = \sum_{i=1}^{n} y_i\,\mathbf{x}_i$$

Associativity of matrix multiplication gives $\mathbf{x}_i\left(\mathbf{x}_i^T\mathbf{w}\right) = (\mathbf{x}_i\mathbf{x}_i^T)\mathbf{w}$

$$\sum_{i=1}^{n}(\mathbf{x}_i\mathbf{x}_i^T)\mathbf{w} = \sum_{i=1}^{n} y_i\,\mathbf{x}_i$$

Least Squares Solution

The equation from previous slide:

$$\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{x}_i^T)\mathbf{w} = \sum_{i=1}^{n} y_i \mathbf{x}_i$$

Taking summation inside:

$$\left(\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{x}_i^T)\right)\mathbf{w} = \sum_{i=1}^{n} y_i \mathbf{x}_i$$

If $\left(\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{x}_i^T)\right)$ is invertible:

$$\mathbf{w} = \left(\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{x}_i^T)\right)^{-1} \left(\sum_{i=1}^{n} y_i \mathbf{x}_i\right)$$

Take pseudo inverse if not invertible.

Matrix view of Least Squares

Suppose we have:

- $n = 5$ data points (samples)
- $d = 3$ features

Sample	Feature 1	Feature 2	Feature 3	Target
1	x_{11}	x_{12}	x_{13}	y_1
2	x_{21}	x_{22}	x_{23}	y_2
3	x_{31}	x_{32}	x_{33}	y_3
4	x_{41}	x_{42}	x_{43}	y_4
5	x_{51}	x_{52}	x_{53}	y_5

Each **row** corresponds to one data point; each **column** corresponds to one feature.

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \\ x_{51} & x_{52} & x_{53} \end{bmatrix} \in \mathbb{R}^{5\times 3}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} \in \mathbb{R}^5$$

Let the parameter vector be:

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} \in \mathbb{R}^3.$$

Least Squares Error: Matrix Norm Form

For data point i, the prediction error is:

$$\mathbf{x}_i^\top \mathbf{w} - y_i,$$

Stacking all errors together gives the **residual vector**:

$$\begin{bmatrix} x_{11}w_1 + x_{12}w_2 + x_{13}w_3 - y_1 \\ x_{21}w_1 + x_{22}w_2 + x_{23}w_3 - y_2 \\ x_{31}w_1 + x_{32}w_2 + x_{33}w_3 - y_3 \\ x_{41}w_1 + x_{42}w_2 + x_{43}w_3 - y_4 \\ x_{51}w_1 + x_{52}w_2 + x_{53}w_3 - y_5 \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ x_{41} & x_{42} & x_{43} \\ x_{51} & x_{52} & x_{53} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \mathbf{X}\mathbf{w} - \mathbf{y}$$

The total squared error over all data points is the squared ℓ_2 norm of the residual vector:

$$\sum_{i=1}^{\mathrm{n}} \left(\mathbf{x}_i^\top \mathbf{w} - y_i\right)^2 = \| \mathbf{X}\mathbf{w} - \mathbf{y} \|_2^2$$

Thus, the least squares problem can be written compactly as:

$$\min_{\mathbf{w}} \| \mathbf{X}\mathbf{w} - \mathbf{y} \|_2^2.$$

Projection view

Thus, the least squares problem can be written compactly as:

$$\min_{\mathbf{w}} \ \| \mathbf{X}\mathbf{w} - \mathbf{y} \|_2^2.$$

As discussed in Lecture 6:

This is the orthogonal projection of the target vector onto the column space of the data matrix $\mathbf{X}$ (i.e., the span of the feature columns)

If $\mathbf{X}^\top\mathbf{X}$ is invertible, the solution is:

$$\boxed{\mathbf{w} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}}.$$

Verify that this is the same solution previously derived using summations or vector calculus—now expressed cleanly in matrix form:

$$\mathbf{w} = \left(\sum_{i=1}^{n} (\mathbf{x}_i \mathbf{x}_i^T)\right)^{-1} \left(\sum_{i=1}^{n} y_i \, \mathbf{x}_i\right)$$

Thank You