Lecture 3:

From Set to Linear Regression

Ho-min Park

homin.park@ghent.ac.kr

powersimmani@gmail.com

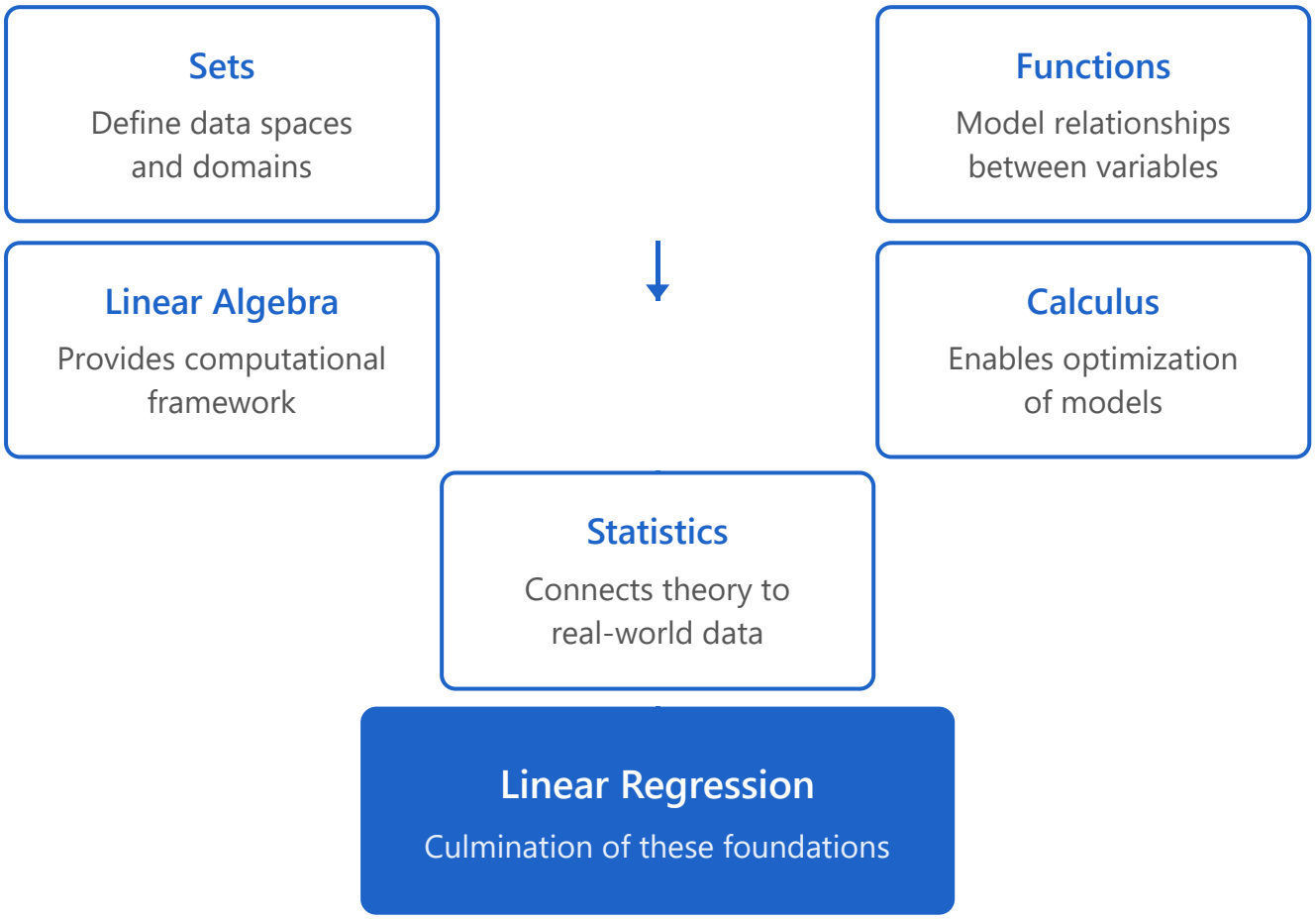
Lecture Contents

Part 1/3:

Mathematical Foundations

1. Course Overview - From Sets to Regression
2. Set Theory Basics and Notation
3. Functions and Mapping Concepts
4. Vector Spaces and Basis
5. Inner Product and Orthogonality
6. Matrix Operations and Properties
7. Inverse Matrices and Determinants
8. Eigenvalues and Eigenvectors
9. Differentiation and Partial Derivatives

Journey from Abstract Mathematics to Practical Machine Learning



Goal: Build intuition for why math matters in ML

Set Theory Basics and Notation

Set: Collection of distinct objects
$X = \{x_1, x_2, ..., x_n\}$

Common sets: $\mathbb{R}$ (real numbers)
$\mathbb{R}^n$ (n-dimensional space)

Membership: $x \in X$ means "x belongs to set X"

Subset: $A \subseteq B$ means all elements of A are in B

Operations: Union ($\cup$), Intersection ($\cap$)
Complement (')

Cartesian product: $X \times Y = \{(x,y) \mid x \in X, y \in Y\}$

Set Operations

$A \cup B$ Union

$A \cap B$ Intersection

ML Application

Data points live in feature space $\mathbb{R}^n$

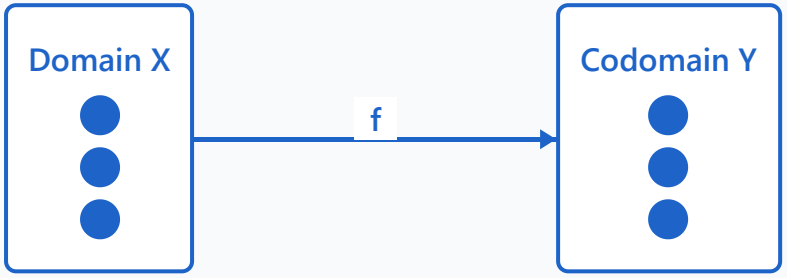
Example: Image data → $\mathbb{R}^{784}$ (28×28 pixels)

Sets define **domains** for regression functions

Sets define domains for our regression functions

Functions and Mapping Concepts

Function: f: X → Y



Each input maps to **exactly one** output

Domain X: Set of all possible inputs

Codomain Y: Set of all possible outputs

Range: Actual outputs achieved by f

One-to-one (Injective)

Different inputs → Different outputs
No two inputs map to the same output

Onto (Surjective)

Every output is reached
Range = Codomain

Bijection

Both injective and surjective
Perfect one-to-one correspondence

ML Application

Regression models are functions
f(x) = y

Function Mapping Examples

Natural Number Division

f: ℕ × ℕ\{0} → ℚ

- f(6, 2) = 3
- f(7, 2) = 3.5

Square Function

f: ℝ → ℝ≥0

- f(-3) = 9
- f(0) = 0

Absolute Value

f: ℝ → ℝ≥0

- f(-5) = 5
- f(0) = 0

• f(10, 4) = 2.5

✓ Well-defined function
Each pair maps to unique rational

• f(4) = 16

✓ Surjective onto non-negative reals
✗ Not injective (±x → same output)

• f(3) = 3

✓ Surjective onto non-negative reals
✗ Not injective (±x → same output)

Vector Spaces and Basis

Vector Space V

Set closed under **addition** and **scalar multiplication**

- $\mathbb{R}^n$ (n-dimensional space)
- Polynomial space
- Function space

Linear Combination

$v = c_1v_1 + c_2v_2 + \ldots + c_nv_n$

Span

All possible linear combinations of vectors

Linear Independence

No vector is combination of others

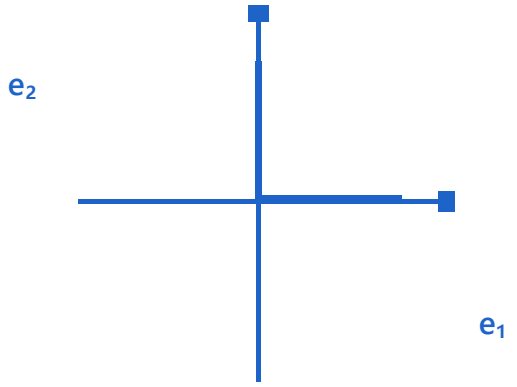
Basis

Minimal spanning set (linearly independent)

Dimension

Number of basis vectors

Standard Basis in $\mathbb{R}^2$



Standard Basis in $\mathbb{R}^n$

$$e_1 = (1, 0, \ldots, 0)$$

$$e_2 = (0, 1, \ldots, 0)$$

Any vector $v = c_1e_1 + c_2e_2 + \ldots + c_ne_n$

Inner Product and Orthogonality

Inner Product (Dot Product)

$$\langle x, y \rangle = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$$

Geometric Interpretation

$$\langle x, y \rangle = \|x\| \|y\| \cos(\theta)$$

Measures alignment between vectors

Norm (Length)

$$\|x\| = \sqrt{(\langle x, x \rangle)} = \sqrt{(x_1^2 + x_2^2 + \ldots + x_n^2)}$$

Orthogonality

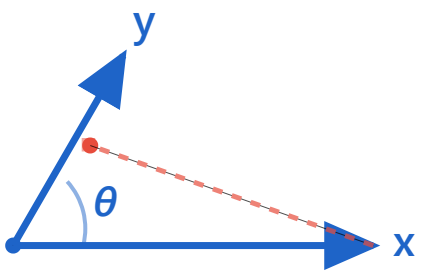
$$x \perp y \text{ when } \langle x, y \rangle = 0$$

Vectors are perpendicular ($\theta = 90°$)

Orthonormal Basis

Basis vectors with unit length, mutually orthogonal

Geometric Interpretation



Inner product ⟨x, y⟩ measures how much x "projects onto" y

Dashed line shows the projection of x onto y

Key for Regression

Projection

$$\text{proj}_y(x) = (\langle x, y\rangle / \langle y, y\rangle) \cdot y$$

Component of x in direction of y

- Projecting data onto subspaces
- Residuals are orthogonal to fitted values
- Minimizing distance = maximizing projection

Matrix Operations and Properties

Matrix $A \in \mathbb{R}^{m \times n}$

Rectangular array of numbers

$$\begin{bmatrix} a_{1\,1} & a_{1\,2} & \cdots & a_{1\,n} \\ a_{2\,1} & a_{2\,2} & \cdots & a_{2\,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Matrix-Vector Multiplication

Ax represents linear transformation

Matrix Multiplication

$(AB)_{ij} = \Sigma_k A_{ik} B_{kj}$

Transpose

A^T swaps rows and columns

Symmetric Matrix

$A = A^T$ (important in regression)

Identity Matrix

$AI = IA = A$

Matrix Multiplication Example

$$\underset{m \times k}{A} \times \underset{k \times n}{B} = \underset{m \times n}{AB}$$

Identity Matrix I

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Key Properties

$$(AB)^T = B^T A^T$$

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

ML Application

Matrices encode systems of linear equations

Meanings of Matrix Multiplication

Geometric Meaning

Linear Transformation

Multiplying matrix A by vector x (Ax) transforms the vector through **rotation, scaling, reflection, or shearing**.

```
2D Rotation Matrix Example:
[cos θ  -sin θ] [x]
[sin θ   cos θ] [y]
→ Rotates point (x,y) by angle θ
```

Composition of Transformations

AB means applying transformation B first, then A. It combines two transformations into one.

Other Meanings

1. System of Linear Equations

Express multiple equations at once in Ax = b form

```
2x + 3y = 5      [2 3][x]   [5]
4x + 1y = 6  →   [4 1][y] = [6]
```

2. Composition of Relations

Connects two relations to create a new one

Student → **A** → Course → **B** → Professor

AB = Student → Professor relation

3. Weighted Summation (Neural Networks)

In neural networks, it multiplies inputs by weights and sums them. Used to compute each neuron's output.

Inverse Matrices and Determinants

Inverse Matrix A^{-1}

$$AA^{-1} = A^{-1}A = I$$

✓ Exists only when A is **square** and **non-singular**

Solving Linear Systems

$$Ax = b$$

↓

$$x = A^{-1}b$$

Determinant det(A)

Scalar measuring "volume scaling"

$\det(A) \neq 0 \iff$ A is invertible

$$\det(AB) = \det(A) \cdot \det(B)$$

$$\det(A^T) = \det(A)$$

2×2 Matrix Example

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\det(A) = ad - bc$$

Invertibility Condition

If **$\det(A) \neq 0$**

then A^{-1} exists

$$\mathbf{A^{-1} = (1/\det(A)) \cdot adj(A)}$$

Geometric Interpretation

- det(A) > 0: Preserves orientation
- det(A) < 0: Reverses orientation
- |det(A)|: Volume scaling factor

ML Application

Normal equations in regression require matrix inverse: **$(X^TX)^{-1}$**

Meanings of Inverse Matrix

Geometric Meaning

Undo Transformation

If A is a transformation, A^{-1} **reverses it back** to the original state.

```
Examples:
A = 90° rotation  →  A⁻¹ = -90° rotation
A = 2× scaling    →  A⁻¹ = 0.5× scaling
A = shear right   →  A⁻¹ = shear left
```

Other Meanings

1. Direct Solution to Linear Systems

$Ax = b \rightarrow x = A^{-1}b$
Computes the solution directly by matrix multiplication.

2. Reverse Relation

If A represents "Parent → Child" relation,
A^{-1} represents "Child → Parent" relation.

3. Inverse Function

For $f(x) = Ax$, the inverse function is $f^{-1}(y) = A^{-1}y$

Meanings of Determinant

Geometric Meaning

Volume/Area Scaling Factor

|det(A)| tells how much the transformation **scales volumes** (or areas in 2D).

```
Example:
det(A) = 3   → Volume becomes 3×
det(A) = 0.5 → Volume becomes 0.5×
```

Orientation

Other Meanings

1. Invertibility Test

$\det(A) \neq 0 \Leftrightarrow A^{-1}$ exists
A quick way to check if inverse is possible.

2. Linear Independence

$\det(A) \neq 0 \Leftrightarrow$ Column vectors are linearly independent
(No column can be written as a combination of others)

- det > 0: Preserves orientation (no flip)
- det < 0: Reverses orientation (mirror flip)

Dimension Collapse

det = 0 means the transformation **collapses dimensions**
(e.g., 3D → 2D plane, 2D → 1D line)

3. Unique Solution Existence

det(A) ≠ 0 ⟺ Ax = b has a unique solution
(Exactly one solution exists for any b)

Eigenvalues and Eigenvectors

Eigenvector v

$$Av = \lambda v$$

Direction unchanged by matrix A

Eigenvalue λ

Scaling factor for the eigenvector

Characteristic Equation

$\det(A - \lambda I) = 0$

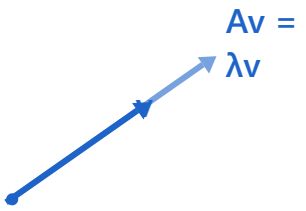
Number of Eigenvalues

n×n matrix has n eigenvalues
(counting multiplicity)

Spectral Theorem

Symmetric matrices have
orthogonal eigenvectors

Geometric Interpretation



Matrix A **stretches** v by factor λ
but keeps the **same direction**

Eigendecomposition

$$A = Q\Lambda Q^T$$

For symmetric matrix A
Q: orthogonal eigenvectors
Λ: diagonal matrix of eigenvalues

ML Applications

- Understanding data variance and covariance structure
- PCA (Principal Component Analysis)
- Regression diagnostics use eigenanalysis

Example: Finding Eigenvalues and Eigenvectors

Given Matrix A

$$A = \begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}$$

Step 1: Find eigenvalues using $\det(A - \lambda I) = 0$

$$\det\begin{bmatrix} 4-\lambda & 1 \\ 2 & 3-\lambda \end{bmatrix} = 0$$

$$(4-\lambda)(3-\lambda) - 2 = 0$$
$$\lambda^2 - 7\lambda + 10 = 0$$
$$(\lambda-5)(\lambda-2) = 0$$

$\lambda_1 = 5, \lambda_2 = 2$

Find Eigenvectors

For $\lambda_1 = 5$: Solve $(A - 5I)v = 0$

$$\begin{bmatrix} -1 & 1 \\ 2 & -2 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$-v_1 + v_2 = 0 \rightarrow v_2 = v_1$$

$v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

For $\lambda_2 = 2$: Solve $(A - 2I)v = 0$

$$\begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$2v_1 + v_2 = 0 \rightarrow v_2 = -2v_1$$

$v_2 = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$

Differentiation and Partial Derivatives

Derivative f'(x)

$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Rate of change at a point

Partial Derivative ∂f/∂x

$$\frac{\partial f}{\partial x_i} = \lim_{h \to 0} \frac{f(x+he_i) - f(x)}{h}$$

Derivative w.r.t. one variable, holding others constant

Chain Rule

$$\frac{d}{dx} f(g(x)) = f'(g(x)) \cdot g'(x)$$

Essential for backpropagation

Product Rule

$$(fg)' = f'g + fg'$$

Quotient Rule

$$\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}$$

Gradient ∇f(x)

$$\nabla f(x) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right]^T$$

Vector of all partial derivatives

Points in direction of steepest ascent

Magnitude = rate of increase

Hessian Matrix H(f)

$$H_{ij} = \frac{\partial^2 f}{\partial x_i \partial x_j}$$

Matrix of second derivatives

Captures curvature of function

Optimization Condition

At minimum/maximum:

$$\nabla f(x^*) = 0$$

ML Applications

- Gradient descent optimization
- Finding regression coefficients
- Minimizing loss functions

- $\partial L/\partial \beta = 0$ solves for optimal β

Example: Computing Derivatives and Gradient

Example 1: Chain Rule

Given: $f(x) = (3x^2 + 2x)^5$

Find: $f'(x)$

Solution: Let $u = 3x^2 + 2x$

$$f(x) = u^5$$

$$f'(x) = 5u^4 \cdot u'$$

$$u' = 6x + 2$$

$$f'(x) = 5(3x^2 + 2x)^4(6x + 2)$$

Example 2: Partial Derivatives

Given: $f(x,y) = x^2y + 3xy^2 + y^3$

Find: $\partial f/\partial x$ and $\partial f/\partial y$

$\partial f/\partial x$: (treat y as constant)

$$\partial f/\partial x = 2xy + 3y^2$$

$\partial f/\partial y$: (treat x as constant)

$$\partial f/\partial y = x^2 + 6xy + 3y^2$$

$$\nabla f = [2xy + 3y^2, x^2 + 6xy + 3y^2]^T$$

Example 3: Gradient Descent Step

Loss function: $L(w) = (w - 3)^2$

Current: $w = 0$, learning rate $\alpha = 0.1$

Step 1: Compute gradient

$$\nabla L(w) = 2(w - 3)$$

$$\nabla L(0) = 2(0 - 3) = -6$$

Step 2: Update parameter

$$w\_new = w - \alpha \nabla L(w)$$

$$w\_new = 0 - 0.1(-6) = 0.6$$

New $w = 0.6$ (closer to minimum at $w = 3$)

Example 4: Hessian Matrix

Given: $f(x,y) = x^2 + 2xy + 3y^2$

Find: Hessian $H(f)$

First derivatives:

$$\partial f/\partial x = 2x + 2y$$

$$\partial f/\partial y = 2x + 6y$$

Second derivatives:

$$\partial^2 f/\partial x^2 = 2, \quad \partial^2 f/\partial x \partial y = 2$$

$$\partial^2 f/\partial y \partial x = 2, \quad \partial^2 f/\partial y^2 = 6$$

H = [2 2]
[2 6]

Part 2/3:

Probability and Statistics Fundamentals

10. Probability Spaces and Random Variables
11. Probability Distributions - Discrete and Continuous
12. Expectation, Variance, and Covariance
13. Conditional Probability and Bayes' Theorem
14. Central Limit Theorem and Law of Large Numbers
15. Parameter Estimation - MLE and MAP
16. Hypothesis Testing and Confidence Intervals
17. Correlation vs Causation

Probability Spaces and Random Variables

Sample Space Ω

Set of all possible outcomes

Event A

Subset of sample space

$$A \subseteq \Omega$$

Probability Measure P

$$P(A) \in [0,1]$$

$$P(\Omega) = 1$$

Random Variable X

Function mapping outcomes to real numbers

Types of Random Variables

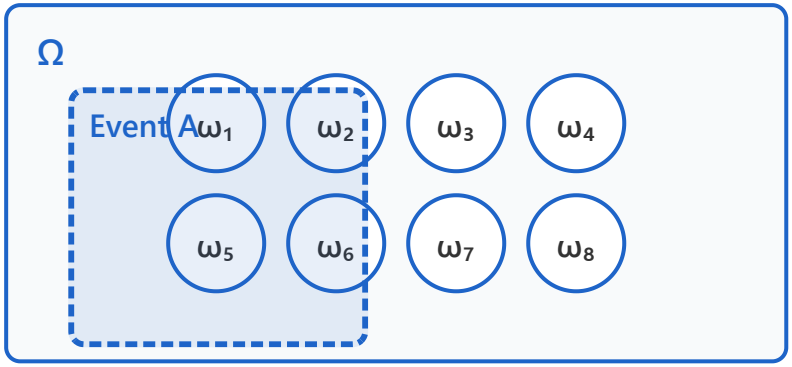
Discrete RV

Countable values (coin flips, dice)

Continuous RV

Uncountable values (heights, temps)

Sample Space & Event Example



Random Variable X: Ω → ℝ

Outcomes		Values
Heads	→	1
Tails	→	0

CDF: F(x) = P(X ≤ x)

Cumulative distribution function

ML Application

Foundation for modeling uncertainty in regression

📌 Why It Matters

1 Mathematical Framework for Uncertainty

The Sample Space defines "what can happen," while the Probability Measure quantifies "how likely each outcome is." This allows us to rigorously analyze uncertain phenomena.

2 Converting to Analyzable Form

Random Variables transform abstract outcomes like "heads" into numbers. This enables mathematical operations (mean, variance, expectation) and forms the basis for statistical inference and ML modeling.

3 Tools for Prediction & Decision-Making

CDFs and probability distributions let us calculate "the probability that X is below a certain value." This is essential for risk assessment, A/B testing, and confidence intervals.

4 Mathematical Foundation of ML/AI

Probability spaces underpin Bayesian inference, probabilistic graphical models, and generative models (VAE, Diffusion). They're essential for expressing and learning model uncertainty.

🔍 Concrete Examples

🎲 Rolling a Die

Sample Space Ω

{1, 2, 3, 4, 5, 6}

Event A: Even number

{2, 4, 6}, P(A) = 1/2

Random Variable X

Face value → E[X] = 3.5

📧 Spam Filtering

Sample Space Ω

{Spam, Not Spam}

Event A

Contains "free" → Spam

Random Variable X

Spam=1, Not Spam=0

📈 Stock Prediction

Sample Space Ω

All possible price changes

Event A

Price rises ≥5% tomorrow

Random Variable X

Daily return (Continuous RV)

Key Applications in Machine Learning

Classification

Estimate $P(Y=k|X)$ to predict class probabilities

Regression

Estimate $E[Y|X]$ and quantify prediction uncertainty

Generative Models

Learn $P(X)$ distribution to generate new samples

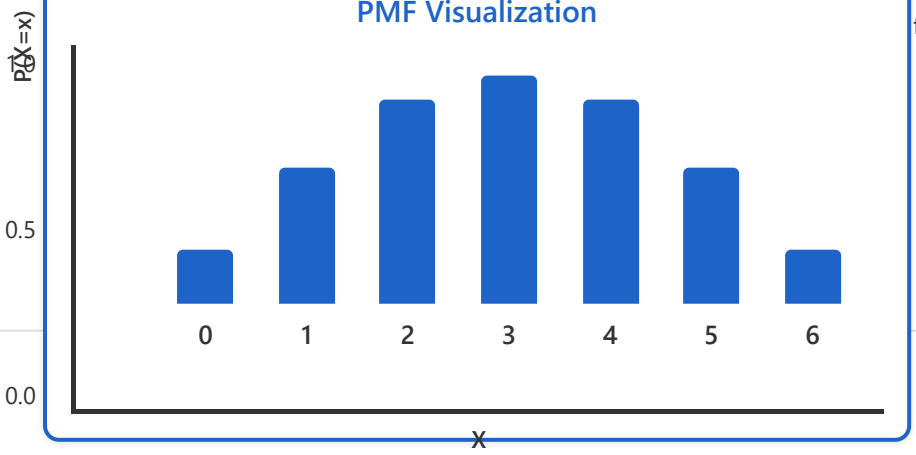
Probability Distributions - Discrete and Continuous

Discrete Distributions
Probability Mass Function (PMF)

P(X = x)
Probability of specific values

Examples
- **Bernoulli:** Coin flip
- **Binomial:** n trials
- **Poisson:** Rare events

PMF Visualization



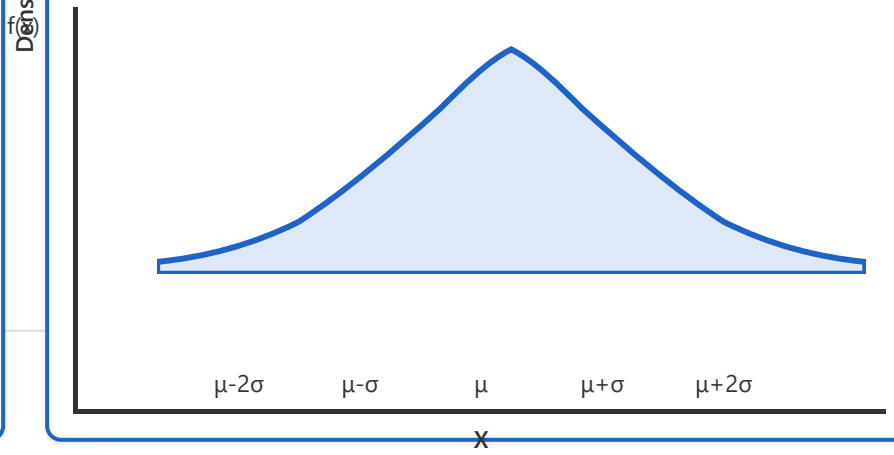
Continuous Distributions
Probability Density Function (PDF)

f(x)
$P(a \le X \le b) = \int_a^b f(x)dx$

Normal Distribution
$$X \sim N(\mu, \sigma^2)$$

Bell-shaped curve

PDF Visualization (Normal)



Standard Normal Distribution

ML Impact

Z ~ N(0, 1) - Used for standardization

Errors in regression often assumed normal

Enables statistical inference

Expectation, Variance, and Covariance

Expectation E[X]

Average value

$$E[X] = \Sigma\ xP(X=x) \text{ or } \int xf(x)dx$$

Properties

$$E[aX + b] = aE[X] + b$$

$$E[X + Y] = E[X] + E[Y]$$

Variance Var(X)

Spread around mean

$$Var(X) = E[(X - \mu)^2]$$

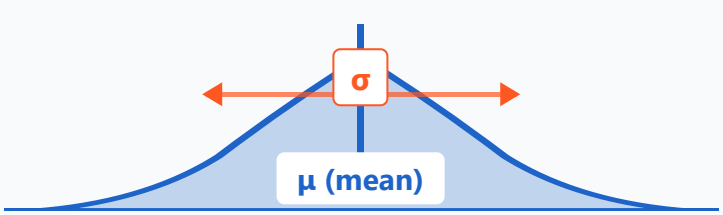
Standard Deviation σ

$$\sigma = \sqrt{Var(X)}$$

Covariance

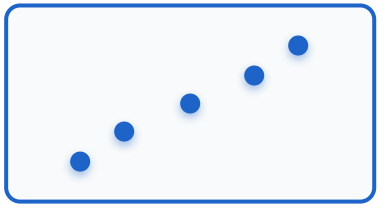
$$Cov(X,Y) = E[(X - \mu_X)(Y - \mu_Y)]$$

Variance: Spread Around Mean

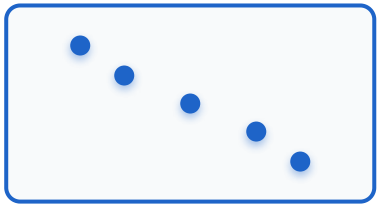


Covariance Sign Interpretation

Positive Cov



Negative Cov



Covariance Matrix

$$\Sigma = [Cov(X_i, X_j)]$$

For multiple variables

Regression Application

Positive: X and Y increase together

Understanding variable relationships

📚 Intuitive Understanding

🎯 Expectation

> "What value can we expect on average?"

Expectation is the **"average over infinite repetitions"**.

🎲 Dice Example

Each face (1-6) has probability 1/6

$$E[X] = 1\times(1/6) + 2\times(1/6) + \ldots + 6\times(1/6) = 3.5$$

You can't roll 3.5 in a single throw,
but roll 1000 times and the average converges to 3.5.

📊 Variance

> "How far are values from the mean?"

Variance measures how spread out data is around the mean.

📝 Test Scores Comparison

Student A: 70, 70, 70, 70 → Mean 70, Variance 0 (consistent)
Student B: 40, 60, 80, 100 → Mean 70, High variance (volatile)

Why use squared differences?
Using (X - μ) alone causes positive/negative values to cancel out.
Squaring makes all deviations positive and penalizes outliers more heavily.

🖇️ Covariance

> "Do two variables move together?"

Covariance measures the directional relationship between two variables.

🗂️ Covariance Matrix

> "How to see all variable relationships at once?"

A table organizing covariances between all pairs of variables.

X vs Mean	Y vs Mean	Product Sign
Above (+)	Above (+)	+
Below (-)	Below (-)	+
Above (+)	Below (-)	-
Below (-)	Above (+)	-

Positive Cov Height↑ Weight↑ — increase together

Negative Cov Study time↑ Gaming↓ — move opposite

Near Zero Variables move independently

$$\Sigma = \begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) \end{bmatrix}$$

Diagonal: *Variance of each variable (relationship with itself)*
Off-diagonal: *Covariance between variables*

ML Application

Used in regression to understand variable relationships and diagnose multicollinearity issues.

Conditional Probability and Bayes' Theorem

Conditional Probability

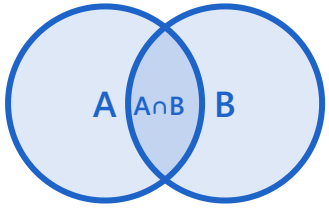
$$P(A|B) = P(A \cap B) / P(B)$$

Probability of A given B has occurred

Multiplication Rule

$$P(A \cap B) = P(A|B)P(B)$$

Visual Representation



Independence: X ⊥ Y

$$P(X|Y) = P(X)$$

Bayes' Theorem

$$P(A|B) = P(B|A)P(A) / P(B)$$

Prior	→	Likelihood	→	Posterior
$P(A)$		$P(B\|A)$		$P(A\|B)$

Applications

- Updating beliefs with new data
- Medical diagnosis and testing
- Spam filtering and classification

ML Foundation

Bayesian regression and inference

🎯 Intuitive Understanding

Conditional Probability

Multiplication Rule

Think of it as "zooming in" on a smaller world. Instead of looking at all possibilities, we only consider the cases where B already happened. P(A|B) asks: "*Within the world where B is true, how often does A also happen?*"

Real-world Example

What's the probability it's raining **given that** the ground is wet? We ignore all dry-ground scenarios and only look at wet-ground cases.

This tells us how to find the probability of two things happening together. It's like asking: "What's the chance of B happening, *and then* A also happening given that B occurred?"

Real-world Example

Probability of drawing two aces in a row = P(first ace) × P(second ace | first was ace)

Bayes' Theorem

Bayes lets you **flip the direction** of conditional probability. If you know P(symptom | disease), Bayes helps you find P(disease | symptom) — which is usually what we actually care about!

The Three Steps

Prior: Your initial belief before seeing evidence
Likelihood: How well the evidence fits your hypothesis
Posterior: Your updated belief after seeing evidence

Independence

Two events are independent when knowing one tells you *nothing* about the other. Learning that B happened doesn't change your belief about A at all — the probability stays exactly the same.

Real-world Example

A coin flip result is independent of previous flips. Getting heads 5 times doesn't change the probability of the next flip being heads (still 50%).

The Big Picture

Bayes' Theorem is fundamentally about **learning from evidence**. You start with a belief (prior), observe new information (likelihood), and update your belief (posterior). This is exactly how machine learning works — models start with initial

parameters and update them as they see more data. It's also how rational thinking works: we should update our beliefs proportionally to the strength of new evidence.

Central Limit Theorem and Law of Large Numbers

Law of Large Numbers

Sample mean converges to population mean

Convergence in probability

Central Limit Theorem

Sum of random variables approaches normal distribution

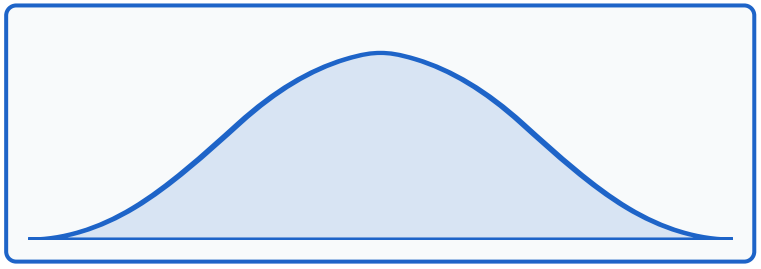
as

CLT: Any Distribution → Normal Distribution

Original Distribution
(any shape)

Take means

Distribution of Means
(normal)



Key Insight

Averages are approximately normal with large

Regression Use

Justifies normal assumption in residuals

Enables

Confidence intervals & hypothesis tests

Real-World Applications

Quality Control

Manufacturing defect rates follow normal distribution with large samples.

Use: *Setting acceptable quality limits in production lines*

A/B Testing

Website conversion rates comparison uses CLT for statistical significance.

Use: *Deciding which website design performs better*

Opinion Polls

Survey results become normally distributed with sufficient sample size.

Use: *Predicting election outcomes with confidence intervals*

Financial Risk

Portfolio returns approximate normal distribution over time.

Use: *Value-at-Risk (VaR) calculations for investments*

Medical Trials

Average treatment effects tested using normal approximation.

Use: *Determining if new drugs are effective*

Machine Learning

Bootstrap confidence intervals rely on CLT for model evaluation.

Use: *Estimating model performance reliability*

Parameter Estimation - MLE and MAP

Parameter Estimation: Inferring θ from observed data

Maximum Likelihood Estimation (MLE)

Objective

$$\text{argmax } L(\theta|\text{data})$$

Likelihood Function

$$L(\theta) = P(\text{data}|\theta) = \prod_i P(x_i|\theta)$$

Log-Likelihood

$$\ell(\theta) = \log L(\theta)$$

Easier to optimize

MLE for Normal Distribution

$$\hat{\mu} = \bar{X}$$

$$\sigma^2 = (1/n)\Sigma(x_i - \bar{X})^2$$

Maximum A Posteriori Estimation (MAP)

Objective

$$\text{argmax } P(\theta|\text{data})$$

Using Bayes' Theorem

$$P(\theta|\text{data}) = P(\text{data}|\theta)P(\theta) / P(\text{data})$$

MAP = MLE + Prior

Incorporates prior knowledge

Key Difference

MAP includes **prior P(θ)**
Allows regularization and
incorporation of domain knowledge

Quick Comparison

Aspect	MLE	MAP
Prior	No prior	Uses prior P(θ)

Regression Use

MLE/MAP estimate regression coefficients

Focus	Data only	Data + knowledge

📊 Practical Examples: MLE vs MAP

Discrete: Bernoulli

🪙 Coin Flip

Flip a coin **5 times**

`4 Heads, 1 Tail`

Estimate P(Heads) = θ

MLE Approach

Likelihood:

$L(\theta) = \theta^4 (1-\theta)^1$

Maximize → derivative = 0:

$\hat{\theta} = k/n = 4/5$

MLE Estimate

$\hat{\theta} = 0.80$

⚠ Overfits to limited data

MAP Approach

Prior belief: fair coin

$P(\theta) = \text{Beta}(\alpha=2, \beta=2)$

Posterior ∝ Likelihood × Prior:

$P(\theta|X) \propto \theta^4 (1-\theta)^1 \times \theta^1(1-\theta)^1$

$= \theta^5(1-\theta)^2 = \text{Beta}(6, 3)$

MAP = Beta mode:

$(\alpha'-1)/(\alpha'+\beta'-2) = (6-1)/(6+3-2) = 5/7$

MAP Estimate

$\hat{\theta} \approx 0.71$

✓ Balanced by prior knowledge

Continuous: Gaussian

🌡 Sensor Reading

Measure temperature **3 times** ($\sigma^2 = 4$ known)

`24°C, 26°C,`

MLE Approach

Likelihood (Gaussian):

$L(\mu) = \prod_i N(x_i|\mu, \sigma^2)$

MLE = Sample mean:

$\hat{\mu} = \bar{X} = (24+26+28)/3$

MLE Estimate

$\hat{\mu} = 26.0°C$

MAP Approach

Prior belief: room temp ~22°C

$P(\mu) = N(\mu_0=22, \tau^2=9)$

Posterior ∝ Likelihood × Prior (both Gaussian):

$P(\mu|X) \propto N(\bar{X}, \sigma^2/n) \times N(\mu_0, \tau^2)$

MAP = Precision-weighted average:

28°C

Estimate true temp μ

⚠ Pure data average

$\hat{\mu} = (n\bar{X}/\sigma^2 + \mu_0/\tau^2) / (n/\sigma^2 + 1/\tau^2)$

$= (3\times26/4 + 22/9) / (3/4 + 1/9)$

$= 21.94 / 0.86$

MAP Estimate

$\hat{\mu} \approx 25.5°C$

✓ Shrinks toward prior (22°C)

Hypothesis Testing and Confidence Intervals

Hypothesis Testing

H_0: Null Hypothesis

Default assumption

$$e.g., \beta = 0$$

H_1: Alternative Hypothesis

Claim to test

$$e.g., \beta \neq 0$$

Test Statistic

Measure computed from data (t-stat, z-stat)

p-value

P(observe data or more extreme | H_0 true)

Decision Process

Set significance level α
(typically 0.05)

↓

Compute test statistic

↓

Confidence Intervals

95% Confidence Interval

$$[\hat{\theta} - 1.96SE, \hat{\theta} + 1.96SE]$$

Calculate p-value

↓

p-value < α ?

Yes: Reject H_0

No: Fail to reject H_0

Interpretation

Range likely containing true parameter

95% of such intervals contain θ

Connection to Testing

If H_0: $\theta = \theta_0$ is not in the CI, then reject H_0 at $\alpha = 0.05$

Real-World Examples: Hypothesis Testing

Case 1: Drug Efficacy

A pharmaceutical company tests whether a new drug reduces blood pressure more than placebo.

H_0: μ_drug = μ_placebo
H_1: μ_drug < μ_placebo

Data:
n = 100 patients
Mean difference = -8.5 mmHg
SE = 2.1 mmHg
t-stat = -4.05

p-value: 0.0001
α level: 0.05

Reject H_0
Drug is effective!

Case 2: A/B Testing

An e-commerce site tests whether a new checkout design increases conversion rate.

H_0: p_new = p_old
H_1: p_new ≠ p_old

Data:
Control: 450/10000 = 4.5%
Treatment: 520/10000 = 5.2%
z-stat = 2.28

p-value: 0.023
α level: 0.05

Reject H_0
New design works!

Case 3: Salary Regression

Testing whether years of education significantly predict salary (β_1 coefficient).

H_0: $\beta_1 = 0$
H_1: $\beta_1 \neq 0$

Data:
n = 500 employees
$\hat{\beta}_1$ = 5,200 ($/year)
SE($\hat{\beta}_1$) = 800
t-stat = 6.5

p-value: < 0.001
95% CI: [3,632, 6,768]

Reject H_0
Education matters!

Correlation vs Causation

Correlation

Definition

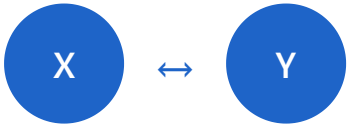
Statistical association between variables

Pearson Correlation

$$\rho = \mathrm{Cov}(X,Y) \;/\; (\sigma_X \sigma_Y)$$

$\rho = -1$	$\rho = 0$	$\rho = 1$
Perfect negative	No linear	Perfect positive

Association only



Causation

Definition

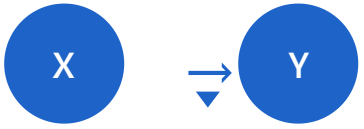
X directly influences Y

Causal Relationship

Requires:

- Temporal precedence
- Mechanism
- Control of confounders

Direct influence



⚠ Key Warning

Correlation does NOT imply causation

Confounding Variable

Classic Example

Z affects both X and Y,
creating spurious correlation

Correlated, but ice cream
doesn't cause drownings

Regression Limitation

Regression shows association, not necessarily causation

💡 Understanding Intuitively

Correlation is "Dancing Together"

Two people dancing to the same rhythm doesn't mean one person is making the other move. They might both be responding to **the same music** (a hidden variable).

"Moving together" ≠ "One causes the other to move"

Causation is "The Domino Effect"

When you push the first domino, the next one falls. Here, **the push** is the direct cause. There's a clear mechanism and temporal order.

"If you do A → B happens" (Direct influence)

🤔 3 Questions to Verify Causation

1

Temporal Order

Did X happen before Y?

2

Mechanism

Can you explain how X affects Y?

3

Third Variable

Is there a hidden factor affecting both?

🌎 Real-World Examples

"Coffee drinkers live longer"

Correlation (Healthy lifestyle?)

"This medication lowers blood pressure"

Causation (Clinical trial verified)

"More ad spending increases sales"

Correlation? (Economic conditions?)

📌 Key Takeaways

📈 Correlation

"Two variables change together" is just an **observation**. It's a starting point for finding patterns, not a conclusion.

🎯 Causation

"X causes Y" is a **proven relationship**. It requires controlled experiments or strong evidence.

Part 3/3:

Linear Regression Model

18. Linear Regression Problem Definition
19. Simple Linear Regression Model
20. Deriving the Least Squares Method
21. Normal Equation Solution
22. Geometric Interpretation
23. Multiple Linear Regression Extension
24. Model Assumptions and Diagnostics
25. Python Implementation and Practice

Linear Regression Problem Definition

Goal

Model relationship between input X and output Y

Core Assumption

Linear relationship

$$Y = f(X) + \varepsilon$$

Training Data

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$

n labeled examples

Objective

Find function f that best approximates the true relationship

Learning Paradigm

Supervised Learning:
Learning from labeled examples

Regression Workflow

1. Collect Data

Training examples (X, Y)

↓

2. Learn Function f

Find best fit to data

↓

3. Predict

Given new x → estimate $\hat{y} = f(x)$

Applications

- Price prediction
- Trend forecasting
- Causal inference

Bridge from theory to practical machine learning

Matrix Representation

Linear Model

For multiple features:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon$$

In matrix form:

$$Y = X\beta + \varepsilon$$

where:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} 1 & x_{1\,1} & x_{1\,2} & \cdots & x_{1\,p} \\ 1 & x_{2\,1} & x_{2\,2} & \cdots & x_{2\,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Y: n×1, X: n×(p+1), β: (p+1)×1, ε: n×1

Least Squares Solution

Objective: Minimize sum of squared errors

$$\min ||Y - X\beta||^2$$

Normal Equation:

$$X^T X\beta = X^T Y$$

Solution:

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

Prediction: $\hat{Y} = X\hat{\beta}$

This requires $X^T X$ to be invertible
(X has full column rank)

Simple Linear Regression Model

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

β_0: Intercept

Value when X = 0

β_1 : Slope

Change in Y per unit change in X

ε: Error Term

Random error, $\varepsilon \sim N(0, \sigma^2)$

Fitted Line

$$\hat{y} = \hat{\beta_0} + \hat{\beta_1} x$$

Estimated parameters

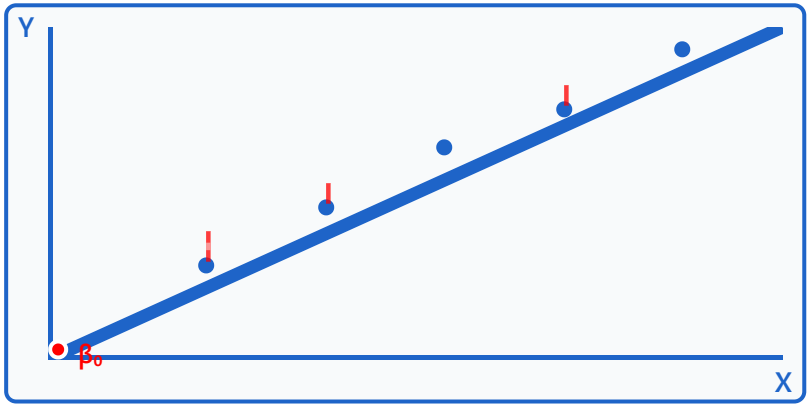
Residual: $e_i = y_i - \hat{y}_i$

Prediction error

Objective

Minimize total prediction error

Visual Representation



Blue line: fitted regression | Dashed red: residuals

Real-World Example

$$\text{Salary} = \beta_0 + \beta_1 \times (\text{Years of Experience}) + \varepsilon$$

📊 2D Interactive Simulator

Explore simple linear regression with an interactive 2D simulator!

Launch 2D Simulator →

🎯 3D Interactive Simulator

Visualize multiple regression in 3D space with interactive controls!

Launch 3D Simulator →

Deriving the Least Squares Method

Loss Function: Sum of Squared Errors

$$L(\beta_0, \beta_1) = \sum_i (y_i - \beta_0 - \beta_1 x_i)^2$$

SSE: Sum of Squared Errors

Why Squares?

- Penalizes large errors more heavily
- Mathematically convenient (differentiable)
- No cancellation of positive/negative errors

Optimization Goal

Find $\hat{\beta}_0, \hat{\beta}_1$ that minimize L

Key Requirement

Unique solution exists when
X has full rank

Statistical Connection

Derivation Steps

1 Define Loss Function

$$L(\beta_0, \beta_1) = \sum (y_i - \beta_0 - \beta_1 x_i)^2$$

↓

2 Take Partial Derivatives

$$\partial L / \partial \beta_0 = 0$$

$$\partial L / \partial \beta_1 = 0$$

↓

3 Solve System of Equations

Normal Equations

↓

✓ Solution

Optimal parameters $\hat{\beta}_0, \hat{\beta}_1$
that minimize prediction error

Least Squares = Maximum Likelihood
under normal errors

Normal Equation Solution

Closed-form solution for linear regression

Matrix Form

$$Y = X\beta + \varepsilon$$

X is the design matrix

↓

Normal Equation

$$X^TX\beta = X^TY$$

↓

General Solution

$$\hat{\beta} = (X^TX)^{-1}X^TY$$

When X^TX is invertible

Simple Regression

$$\hat{\beta_1} = \text{Cov}(X,Y) \;/\; \text{Var}(X)$$

$$\hat{\beta_0} = \bar{Y} - \hat{\beta_1}\,\bar{X}$$

Complexity

Computational cost

$$O(np^2 + p^3)$$

n: samples
p: features

Alternative

For large-scale:
Gradient Descent
(iterative method)

Advantage

Closed-form
Direct computation
No iterations needed

Gradient Descent Process

Parameter update with MSE Loss - Step by Step

Mean Squared Error (MSE) Loss

$$L(\beta) = (1/n)\Sigma(y_i - \hat{y}_i)^2 = (1/n)\Sigma(y_i - \beta_0 - \beta_1 x_i)^2$$

↓

1 Compute Partial Derivatives

$$\partial L/\partial \beta_0 = -(2/n)\Sigma(y_i - \beta_0 - \beta_1 x_i)$$

$$\partial L/\partial \beta_1 = -(2/n)\Sigma x_i(y_i - \beta_0 - \beta_1 x_i)$$

Calculate gradient for each parameter

2 Evaluate at Current Parameters

$$g_0 = \partial L/\partial \beta_0 | \beta^{(t)}, \quad g_1 = \partial L/\partial \beta_1 | \beta^{(t)}$$

Substitute current $\beta_0^{(t)}$ and $\beta_1^{(t)}$ values

3 Update Parameters

$$\beta_0^{(t+1)} = \beta_0^{(t)} - \alpha \cdot g_0$$

$$\beta_1^{(t+1)} = \beta_1^{(t)} - \alpha \cdot g_1$$

α is the learning rate (step size)

4 Check Convergence

$$|L^{(t+1)} - L^{(t)}| < \varepsilon \text{ or } ||\nabla L|| < \varepsilon$$

Repeat steps 1-3 until convergence criterion is met

Learning Rate α

Controls step size

Too large: diverge

Too small: slow

Matrix Form

$$\beta^{(t+1)} = \beta^{(t)} - \alpha \cdot \nabla L$$

$$\nabla L = -(2/n)X^T(Y - X\beta)$$

Convergence

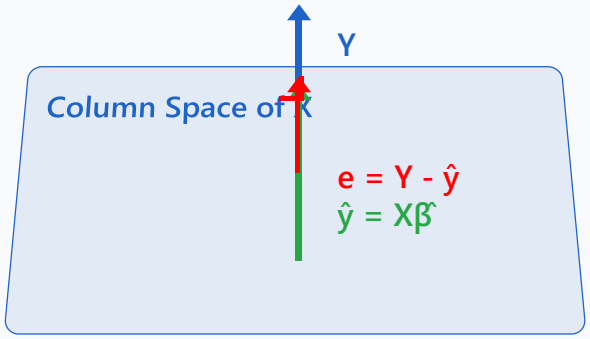
Iterative approach

Works for large-scale

O(np) per iteration

Geometric Interpretation

Projection onto Column Space



Column Space

All possible linear combinations of predictors

Prediction ŷ

Projection of Y onto column space of X

Residual e

Orthogonal to column space (perpendicular)

Projection Matrix

$$P = X(X^TX)^{-1}X^T$$

$$\hat{y} = PY$$

Residual Maker

$$M = I - P$$

$$e = MY$$

Orthogonality

$$\langle e, X_j \rangle = 0$$

for all predictors

$\hat{y}$ is the closest point in column space to Y

Unifies **Linear Algebra** and **Statistics**

Multiple Linear Regression Extension

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon$$

Multiple predictors model

Simple vs Multiple Regression

Aspect	Simple
Predictors	1 variable
Model	$Y = \beta_0 + \beta_1 X$
Aspect	**Multiple**
Predictors	p variables
Model	$Y = X\beta$

Matrix Notation

$$Y = X\beta + \varepsilon$$

$$X \in \mathbb{R}^{n \times (p+1)}$$

Same Solution

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

Normal equation applies

Model Fit Metrics

R^2 (R-squared)

$$R^2 = 1 - SSE/SST$$

Proportion of variance explained

Adjusted R^2

Penalizes model complexity

⚠ Multicollinearity

Correlated predictors
cause instability in estimates

Challenges

- Increased complexity
- More parameters to estimate
- Risk of overfitting
- Interpretation becomes harder

Feature Selection

Choosing relevant predictors
to improve model performance

Coefficient Interpretation

β_j is the effect of X_j
holding all other predictors constant

Understanding Multicollinearity in Detail

📊 Definition

High correlation between two or more predictor variables

$$Corr(X_i, X_j) \approx \pm 1$$

Makes it difficult to isolate individual effects

⚠️ Problems

1. Unstable estimates:
$\hat{\beta}$ coefficients become highly sensitive to small data changes

2. Large standard errors:
Var($\hat{\beta}$) increases dramatically

3. Poor interpretability:
Cannot determine which variable is truly important

$(X^TX)^{-1}$ becomes ill-conditioned

🔍 Detection Methods

1. Correlation Matrix:
Check pairwise correlations

$$|corr| > 0.8 \rightarrow \text{Warning}$$

2. VIF (Variance Inflation Factor):

$$VIF = 1/(1 - R^2_j)$$

R^2_j: R^2 when X_j is regressed on other predictors

VIF > 10 → Serious multicollinearity
VIF > 5 → Moderate concern

💡 Solutions

1. Remove variables:
Drop one of the highly correlated predictors

2. Combine variables:
Create composite index (e.g., PCA)

3. Regularization:
Ridge regression (L2) or Lasso (L1)

$$\text{Ridge: } \min ||Y - X\beta||^2 + \lambda||\beta||^2$$

📝 Example Scenario

Predicting house price with:

- Square footage
- Number of rooms

These are highly correlated!
Larger houses → more rooms

Solution: Use only one,
or create "size index"

🎯 Key Insights

Important: Multicollinearity affects *interpretation* and *stability*, but NOT prediction accuracy

The model can still predict well, but we cannot trust individual coefficient values

Overall model fit (R^2) remains good
Individual t-tests become unreliable

Model Assumptions and Diagnostics

Key Assumptions

1. Linearity
True relationship is linear (or approximately)

2. Independence
Observations are independent

3. Homoscedasticity
Constant error variance across X

4. Normality
Errors ~ $N(0, \sigma^2)$

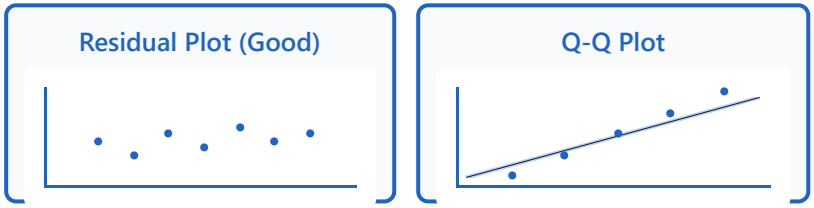
5. No Multicollinearity
Predictors not highly correlated

Diagnostic Tools

- Residual Plots
- Q-Q Plots
- Leverage Plots
- VIF Scores

Visual Diagnostics

Example Diagnostic Plots



Random scatter in residual plot = Good fit
Points on diagonal in Q-Q = Normal errors

Violations & Remedies

- Transformations (log, sqrt)
- Robust regression methods
- Regularization (Ridge, Lasso)

Model Validation

- Train-Test Split
- Cross-Validation
- Bootstrap

Python Implementation and Practice

Key Python Libraries

NumPy
Matrix operations

Scikit-learn
ML convenience functions

Pandas
Data manipulation

Statsmodels
Statistical output

Matplotlib/Seaborn
Visualization

Implementation Examples

```
# NumPy (from scratch)
β = np.linalg.inv(X.T @ X) @ X.T @ y
```

```
# Scikit-learn (easy)
from sklearn.linear_model import LinearRegression
```

Standard Workflow

1. Load Data
↓
2. Explore & Visualize
↓
3. Fit Model
↓
4. Validate (diagnostics)
↓
5. Predict

Best Practices

- Feature scaling
- Handle missing data

```
model = LinearRegression()
model.fit(X, y)
```

```
# Statsmodels (detailed stats)
import statsmodels.api as sm
results = sm.OLS(y, X).fit()
```

Train-test split

Cross-validation

Practice Datasets

Boston Housing

California Housing

Diabetes

Thank you

Ho-min Park

homin.park@ghent.ac.kr

powersimmani@gmail.com