Lecture 2: Linear Regression

Luu Minh Sao Khue

From intuition to implementation — simple and multiple linear regression, OLS theory, regularization (Ridge/Lasso), evaluation with R^2, and practical diagnostics.

1. Learning Objectives

After this lecture, you will be able to:

- Explain the intuition behind simple and multiple linear regression.
- Derive the closed-form Ordinary Least Squares (OLS) solution.
- Interpret regression coefficients, residuals, and R^2.
- Understand Ridge, Lasso, and Elastic Net regularization.
- Implement regression in Python and analyze model diagnostics.

2. Intuition

Linear regression models how a continuous target y changes with one or more predictors $x_1, \ldots, x_p$. We start with a straight line (simple regression), then extend to multiple features (multiple regression). Parameters are chosen to minimize squared prediction errors (*least squares*). This connects optimization, geometry (projection onto the column space of X), and statistics (assumptions about noise).

Example (simple): Predict house price from floor area. A line $\hat{y} = \beta_0 + \beta_1 x$ summarizes the trend; the best line is the one with the smallest average squared residual.

When predictors are many or correlated, OLS can overfit or become unstable. **Regularization** (Ridge/Lasso) shrinks coefficients to improve generalization and interpretability.

3. Simple Linear Regression

3.1 Population Model

Assume two real-valued random variables X and Y satisfy

$$Y \approx \beta_0 + \beta_1 X.$$

The best linear approximation (smallest expected squared error) solves

$$\min_{\beta_0, \beta_1} E[(Y - \beta_0 - \beta_1 X)^2].$$

Differentiating and solving gives

$$\beta_1^* = \frac{\mathrm{Cov}(X, Y)}{\mathrm{Var}(X)}, \qquad \beta_0^* = E[Y] - \beta_1^* E[X].$$

3.2 Sample Model

Given samples (x_i, y_i), $i = 1, \ldots, n$, estimate by minimizing the sum of squared errors (SSE):

$$\min_{\beta_0, \beta_1} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2.$$

The solution is

$$\hat{\beta}_1 = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}, \qquad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}.$$

Prediction for new x_{new}: $\hat{y}_{\text{new}} = \hat{\beta}_0 + \hat{\beta}_1 x_{\text{new}}$.

Notation note. In statistics, regression coefficients are denoted by β, while in machine learning literature, parameters are often written as θ or w. Throughout this course we use β for clarity.

4. Multiple Linear Regression

4.1 Model and Matrix Form

With p predictors,

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \varepsilon_i.$$

In matrix form,

$$y = X\beta + \varepsilon,$$

where $y \in \mathbb{R}^n$, $X \in \mathbb{R}^{n \times (p+1)}$ (first column of ones), and $\beta \in \mathbb{R}^{p+1}$.

4.2 Least Squares Solution

We minimize

$$J(\beta) = \|y - X\beta\|^2.$$

Setting $\nabla_\beta J = 0$ yields

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

if $X^T X$ is invertible.

Non-full rank X. If $X^T X$ is singular (e.g. $p > n$ or exact collinearity), the minimum-norm least-squares solution uses the *Moore–Penrose pseudoinverse*:

$$\hat{\beta} = X^+ y.$$

Regularization (e.g. Ridge) always makes the matrix invertible.

4.3 Geometric Interpretation

The fitted values $\hat{y} = X\hat{\beta}$ are the projection of y onto the column space of X; residuals $r = y - \hat{y}$ are orthogonal to all columns of X, i.e. $X^T r = 0$.

5. Ordinary Least Squares (OLS)

5.1 Model Assumptions

$$y = X\beta + \varepsilon, \quad E[\varepsilon] = 0, \quad \mathrm{Var}(\varepsilon) = \sigma^2 I.$$

Errors are independent, mean-zero, and homoscedastic. These ensure OLS is BLUE (Gauss–Markov). Normality of ε is *not* required for unbiasedness but is often assumed for t/F inference.

5.2 Unbiasedness

$$E[\hat{\beta}] = (X^T X)^{-1} X^T E[y] = (X^T X)^{-1} X^T X\beta = \beta.$$

5.3 Gauss–Markov Theorem

Under the above assumptions, OLS is the **Best Linear Unbiased Estimator (BLUE)**:

$$\mathrm{Var}(a^T\hat{\beta}) = \sigma^2 a^T (X^T X)^{-1} a$$

for any constant vector a.

6. Regularized Linear Models: Ridge and Lasso Regression

OLS may overfit when predictors are correlated or $p \gg n$. Regularized models add a penalty on coefficient magnitude to reduce variance.

6.1 Ridge Regression (ℓ_2 Regularization)

$$J(\beta) = \|y - X\beta\|^2 + \lambda\|\beta\|_2^2, \quad \hat{\beta}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y.$$

- Shrinks coefficients continuously toward zero.
- Reduces variance and stabilizes estimates under multicollinearity.
- $\lambda = 0$ gives OLS; $\lambda \to \infty$ shrinks all coefficients to zero.

6.2 Lasso Regression (ℓ_1 Regularization)

$$J(\beta) = \|y - X\beta\|^2 + \lambda\|\beta\|_1 = \|y - X\beta\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$

No closed form; solved by coordinate descent.

- Encourages sparsity—some coefficients exactly zero (automatic feature selection).
- Improves interpretability with many predictors.

6.3 Comparison and Bias–Variance

	Ridge	Lasso
Penalty	ℓ_2	ℓ_1
Closed form	Yes	No
Feature selection	No	Yes
Effect	Shrinkage	Shrinkage + sparsity

Regularization increases bias but can substantially reduce variance, often lowering test error. λ is tuned by cross-validation.

Elastic Net. Combines both penalties:

$$J(\beta) = \|y - X\beta\|^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2.$$

Implementation details. Standardize features before regularization, and never penalize the intercept.

7. Python Implementation: Ridge and Lasso

```
import numpy as np
from sklearn.linear_model import Ridge, Lasso
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score

# Data
X = np.array([[1], [2], [3], [4], [5]])
y = np.array([2.2, 4.1, 5.9, 8.2, 10.1])

# Always scale features for regularization
scaler = StandardScaler().fit(X)
X_scaled = scaler.transform(X)

ridge = Ridge(alpha=1.0).fit(X_scaled, y)
lasso = Lasso(alpha=0.1).fit(X_scaled, y)

print("Ridge coef:", ridge.coef_, "R^2:", r2_score(y, ridge.predict(X_scaled)))
print("Lasso coef:", lasso.coef_, "R^2:", r2_score(y, lasso.predict(X_scaled)))
```

8. Coefficient of Determination and Error Metrics

8.1 R^2 Definition

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}}, \quad \text{RSS} = \sum_i (y_i - \hat{y}_i)^2, \quad \text{TSS} = \sum_i (y_i - \bar{y})^2.$$

8.2 Adjusted R^2

$$R^2_{\text{adj}} = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1},$$

where p is the number of predictors (excluding the intercept).

8.3 Related Metrics

MSE $=$ RSS$/n$, RMSE $= \sqrt{\text{MSE}}$, RSE (residual std. error) $\widehat{\sigma} = \sqrt{\text{RSS}/(n - p - 1)}$.

Uncertainty intervals. A $(1 - \alpha)$ confidence interval (CI) for the mean response at x_0 is narrower than a prediction interval (PI) for a new observation, since PI adds noise variance σ^2.

9. Algorithm: Gradient Descent

When p is large or data are streaming, minimize

$$J(\beta) = \frac{1}{n}\|y - X\beta\|^2$$

iteratively:

$$\beta^{(t+1)} = \beta^{(t)} - \eta \frac{2}{n} X^T (X\beta^{(t)} - y),$$

where η is the learning rate. Converges to the OLS solution if properly tuned.

10. Interpretation and Diagnostics

- **Slope (β_j):** expected change in y per unit increase in x_j (others fixed).
- **Intercept (β_0):** predicted y when all $x_j = 0$.
- **Residuals:** should have zero mean, constant variance, and no visible pattern.
- **Outliers:** distort slope; examine residual plots.

Hat matrix and leverage. $\hat{y} = Hy$ where $H = X(X^T X)^{-1} X^T$. Diagonal h_{ii} measures leverage; large values indicate influential points. Use Cook's distance to assess influence.

Multicollinearity. Highly correlated predictors inflate variance of $\hat{\beta}$. The Variance Inflation Factor (VIF) for feature j is $\text{VIF}_j = 1/(1 - R_j^2)$, where R_j^2 is from regressing x_j on the remaining predictors.

Heteroscedasticity and autocorrelation. Breusch–Pagan and Durbin–Watson tests help detect these violations.

11. Practical Tips

- Standardize or center features; centering makes $\beta_0 = \bar{y}$ and improves conditioning.
- Use Ridge/Lasso for correlated or many predictors.
- Check residuals and leverage before trusting coefficients.
- Use train/validation/test or K-fold cross-validation to tune λ and estimate generalization error.
- Avoid data leakage: fit preprocessing only on training data.
- Do not extrapolate far beyond observed x range.

12. Summary

We studied simple and multiple linear regression, derived OLS from first principles, and interpreted its geometry as a projection. Under Gauss–Markov assumptions, OLS is BLUE; with many or correlated predictors, regularization (Ridge/Lasso) improves stability and can yield sparse, interpretable models. We evaluated fit with R^2 and error metrics, explored gradient descent for large-scale data, and reviewed diagnostics for reliable inference.

- Simple $\rightarrow$ multiple regression: $y = X\beta + \varepsilon$.
- OLS closed form $(X^T X)^{-1} X^T y$; pseudoinverse for singular X.
- Assumptions: linearity, independence, homoscedasticity (normality for inference only).
- Regularization: Ridge (ℓ_2) for shrinkage, Lasso (ℓ_1) for sparsity, Elastic Net combines both.
- Evaluation: RSS/MSE/RMSE, R^2, adjusted R^2, CIs and PIs.
- Diagnostics: residuals, leverage, multicollinearity (VIF).
- Practice: scale features, cross-validate, avoid extrapolation.

13. Exercises

1. Derive the OLS estimator from $J(\beta) = \|y - X\beta\|^2$.
2. Prove that $E[\hat{\beta}] = \beta$.
3. Implement gradient descent for linear regression from scratch.
4. Compare Ridge and Lasso on a dataset using cross-validation.
5. Compute VIF and identify multicollinearity.
6. Interpret R^2_{adj} and RMSE differences between models.