Linear Regression: Theory, Formulas, Applications, and Practice

September 9, 2025

1 Introduction

Linear Regression is a fundamental supervised learning algorithm that models the linear relationship between input features and a continuous target. Thanks to its interpretability, efficiency, and closed-form solution, it is widely used as a baseline model and a quick validation tool in practice.

2 Theory and Formulas

2.1 Model Assumption

For a feature vector $\mathbf{x} \in \mathbb{R}^d$, the model predicts

$$\hat{y} = f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b, \quad \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R}. \tag{1}$$

By augmenting features with a constant 1 and absorbing the bias into parameters, we write $\hat{y} = \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}$.

2.2 Matrix Form

Given $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\mathbf{y} \in \mathbb{R}^n$, predictions are $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w} + b\mathbf{1}$. With augmented matrix $\tilde{\mathbf{X}} = [\mathbf{X}\ \mathbf{1}]$ and $\tilde{\mathbf{w}} = [\mathbf{w}; b]$, we have $\hat{\mathbf{y}} = \tilde{\mathbf{X}}\tilde{\mathbf{w}}$.

2.3 Loss (Least Squares)

We minimize the mean squared error (MSE):

$$\mathcal{L}(\tilde{\mathbf{w}}) = \frac{1}{2n} \|\tilde{\mathbf{X}}\tilde{\mathbf{w}} - \mathbf{y}\|_2^2. \tag{2}$$

2.4 Closed-form (Ordinary Least Squares)

If $\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}$ is invertible, the optimum is

$$\tilde{\mathbf{w}}^* = \left(\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}\right)^{-1} \tilde{\mathbf{X}}^\top \mathbf{y}. \tag{3}$$

Numerically, QR or SVD is often preferred for stability.

2.5 Gradient Descent (Optional)

One can also use first-order optimization:

$$\nabla_{\tilde{\mathbf{w}}} \mathcal{L} = \frac{1}{n} \tilde{\mathbf{X}}^\top \left(\tilde{\mathbf{X}}\tilde{\mathbf{w}} - \mathbf{y}\right), \tag{4}$$

$$\tilde{\mathbf{w}} \leftarrow \tilde{\mathbf{w}} - \eta \, \nabla_{\tilde{\mathbf{w}}} \mathcal{L}, \tag{5}$$

with learning rate η.

2.6 Regularization (Optional)

Ridge (L2) regression adds

$$\min_{\tilde{\mathbf{w}}} \frac{1}{2n} \|\tilde{\mathbf{X}}\tilde{\mathbf{w}} - \mathbf{y}\|_2^2 + \frac{\lambda}{2} \|\mathbf{w}\|_2^2, \tag{6}$$

with closed form $\tilde{\mathbf{w}}^* = (\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}} + \lambda \mathbf{I})^{-1} \tilde{\mathbf{X}}^\top \mathbf{y}$.

3 Applications and Tips

- **Numerical forecasting**: Prices, sales, temperature and other continuous targets.
- **Interpretable baseline**: Coefficient magnitude/sign hints at feature impact.
- **Practical tips**: Feature scaling, outlier handling, multicollinearity diagnostics, CV for regularization.

4 Python Practice: Closed-form Fit and Visualization

This example will:

1. Generate 1D linear data with noise;
2. Fit via the augmented-matrix OLS closed form;
3. Plot scatter and fitted line, saved to `figures/linear_regression_fit.png`.

Listing 1: linear$_{r}egression_{c}losed_{f}orm.py$

```python
import os
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(42)

# 1) Generate data: y = 3x + 2 + noise
n = 80
X = np.linspace(-3, 3, n).reshape(-1, 1)
true_w, true_b = 3.0, 2.0
y = true_w * X[:, 0] + true_b + np.random.normal(0, 1.0, size=n)

# 2) Augmented matrix and closed form
X_aug = np.hstack([X, np.ones((n, 1))])  # [x, 1]
theta = np.linalg.pinv(X_aug.T @ X_aug) @ X_aug.T @ y
w_hat, b_hat = theta[0], theta[1]

# 3) Visualization and save
fig, ax = plt.subplots(figsize=(6, 4))
ax.scatter(X[:, 0], y, s=18, alpha=0.7, label='data')
xx = np.linspace(X.min(), X.max(), 200)
yy = w_hat * xx + b_hat
ax.plot(xx, yy, color='crimson', lw=2.0, label=f'fit: y={w_hat:.2f}x+{b_hat:.2f}')
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.legend()
ax.set_title('Linear Regression (Closed-form OLS)')

os.makedirs('figures', exist_ok=True)
out_path = os.path.join('figures', 'linear_regression_fit.png')
plt.tight_layout()
plt.savefig(out_path, dpi=160)
print('saved to', out_path)
```

5 Result

Figure **??** shows the fitted line against noisy samples.

6 Summary

Linear Regression is a simple yet powerful baseline. In practice, pay attention to scaling, outliers, and multicollinearity, and select regularization via cross-validation for robust generalization.

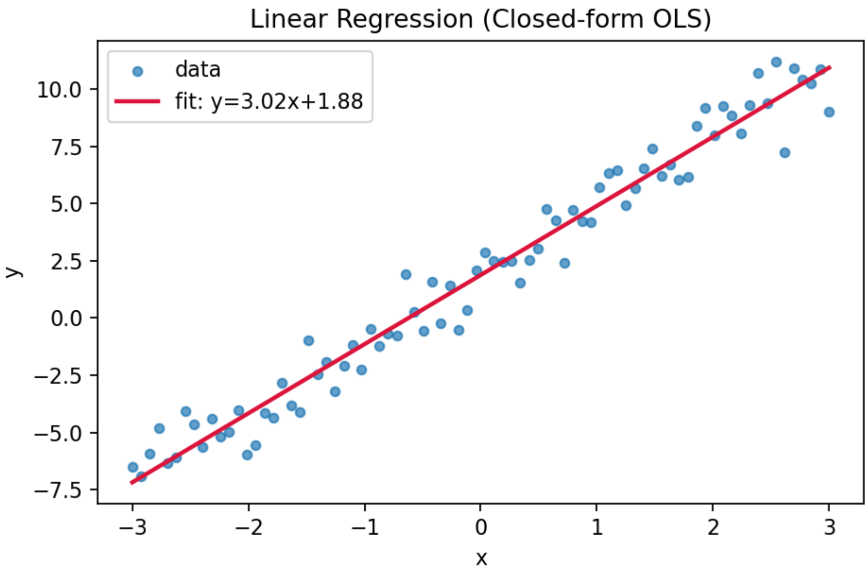

Figure 1: Linear regression fit on synthetic data