Linear Regression and Logistic Regression Fundamentals

Dr. Pruthwik Mishra

16.12.2025

Presented in

Faculty Development Program on “Artificial Intelligence and Data Science: Foundations, Pedagogy, Tools, and Research Trends”

for the Faculty Members of P. P. Savani University, Surat organized by the Department of AI, SVNIT, Surat

Concept of Linear Regression

- Linear regression models the relationship between a dependent variable y and one or more independent variables x.
- This is the simplest regression algorithm to implement.
- The aim is to fit a straight line that best predicts y from x.
- Simple linear regression: $y = w_0 + w_1x + \varepsilon$, where w_0 is the intercept, w_1 is the slope, and ε is the error.

Fundamental Properties

- The regression line minimizes the sum of squared differences between observed and predicted values.
- The regression line always passes through the mean of the X and Y variables.
- The regression constant (intercept) is equal to the y-intercept of the regression line.
- The regression coefficient (slope) represents the average change in the dependent variable (Y) for a unit change in the independent variable (X).

Geometric and Statistical Properties

- Correlation coefficient (r) is the geometric mean of the two regression coefficients for X-on-Y and Y-on-X.
- The sign of the correlation coefficient matches the sign of the regression coefficients.
- The two regression lines intersect at the means of X and Y.
- Regression coefficients are unaffected by a shift of origin, but change of scale affects their values.

Optimality and Interpretation

- The regression line provides the best linear unbiased estimate given standard assumptions.
- The model quantifies relationships and can be interpreted to provide meaningful predictions and insights.

Key Assumptions

- Linearity: Relationship between variables is linear.
- Independence: Errors (residuals) are independent.
- Homoscedasticity: Constant variance of errors for all values of X.
- Normality: Errors are normally distributed.
- No multicollinearity (for multiple regression): Predictors are not too highly correlated.

Ordinary Least Squares (OLS) Estimation

- OLS finds the line that minimizes the sum of squared errors between predictions and actual values.
- The error for each point is: $e_i = y_i - \hat{y}_i = y_i - (w_0 + w_1 x_i)$.
- Total Squared Error: $S = \Sigma [y_i - (w_0 + w_1 x_i)]^2$, summed over all data points.
- Mean Squared Error (MSE) can also be used
  - $MSE = 1/n * \Sigma [y_i - (w_0 + w_1 x_i)]^2$

Mathematical Solution for OLS

- Best-fit coefficients are:
- $w_1 = \Sigma(x_i - \bar{x})(y_i - \bar{y}) \;/\; \Sigma(x_i - \bar{x})^2$
- $w_0 = \bar{y} - w_1\bar{x}$
- where $\bar{y}$ is the mean of y values and $\bar{x}$ is the mean of x values.
- This gives the unique line minimizing the squared error.

Mean Squared Error (MSE)

- MSE measures average squared difference between predicted and actual values.
- MSE = $(1/n) \Sigma [y_i - \hat{y}_i]^2$
- Lower MSE means better fit.
- Linear regression seeks to minimize MSE using OLS.

Geometric Interpretation

- OLS regression line: minimizes squared vertical distances from data points to the line.
- Residuals (errors) are orthogonal (perpendicular) to the regression line.
- Line provides the best linear unbiased estimate for y given x.

Analytical OLS Solution

- OLS seeks to minimize the sum of squared residuals: $S = \Sigma\, [y_i - (w_0 + w_1 x_i)]^2$.
- The normal equations for the coefficients W are solved as $W = (X^t X)^{-1} X^t y$.
- Here, X is the design matrix of predictors, y is the vector of responses.
- Good Explanation of this: https://www.youtube.com/watch?v=g8qF61P741w
- The implementation from scratch is attached in simple_linear_regression.ipynb

Condition for Unique OLS Solution

- The OLS solution $W = (X^tX)^{-1}X^ty$ is unique if and only if the matrix X^tX is invertible.
- This is equivalent to X (the design matrix) having full column rank (its columns are linearly independent).
- No explanatory variable can be a linear combination of others.
- If X^tX is not invertible (i.e., is singular), there are infinitely many solutions.

Geometric Interpretation

- Unique OLS solution exists if the projection of y onto the column space of X is unique.
- This requires the column vectors of X to span a space of the same dimension as the number of predictors.
- Otherwise, the regression hyperplane is not uniquely defined.

Linear Regression Using Gradient Descent

- OLS technique uses a convex loss function ensuring which offers a closed-form solution.
- However, many real world applications have non-convex loss functions for which closed-form solutions do not exist.
- So, we need a technique that can update the parameters in non-convex loss functions.
  - This technique is called the Gradient Descent.
  - It is a generic method for continuous optimization

What is Gradient Descent?

- An iterative optimization algorithm that finds the minimum of a function by moving in the opposite direction of the gradient as our objective is to minimize the loss.
- It converges to optimal parameters
- Gradient: Vector of partial derivatives indicating direction of steepest increase

Core Concepts of Gradient Descent

- $L(w)$ - measures error between predicted and actual values
- Learning Rate (lr/η) - Step size controlling how far to move in each iteration
- Parameters (w) - Weights and biases being optimized. It depends on the number of input features you have
- Update (Delta) Rule:
  - $\mathbf{w_{new} = w_{old} - \eta \, \nabla L(w)}$

Steps of Algorithm

- Step 1: Initialize parameters w (randomly or with zeros or with ones)
- Step 2: Compute loss L(w) for current parameters (is dependent on the type of problem - for regression it is the squared error whereas for classification it is the binary cross-entropy loss)
- Step 3: Calculate gradients $\nabla L(w)$ using backpropagation
- Step 4: Update parameters as $\mathbf{w_{new} = w_{old} - \eta \nabla L(w)}$
- Step 5: Repeat until convergence (loss stops decreasing) or until maximum iterations are reached (the maximum iteration technique can result in divergence)
- Step 6: Convergence Criteria
  - When $||\nabla L(w)|| < \varepsilon$ (epsilon threshold)

Types of Gradient Descent

- Batch Gradient Descent (BGD): Uses entire dataset to compute gradient in each iteration
- Stochastic Gradient Descent (SGD): Uses single random sample to compute gradient per iteration
- Mini-Batch Gradient Descent: Compromise between BGD and SGD using small subsets of data (Batch sizes are usually 16, 32, 64, ..)

Logistic Regression

- What is Logistic Regression?
  - Binary classification algorithm modeling probability of outcomes
  - It uses the sigmoid/logistic function mapping any real number to (0,1)
  - Maps continuous linear combination to a probability in the range of (0, 1)
  - Foundation for deep learning and modern neural networks
- For features X and weights w, the probability of class 1 is:
  - $p(X; w, b) = \sigma(w^T * X + b) = 1/(1 + e^{-(wT * X + b)})$

Loss Function

- Logistic Regression uses binary corss-entropy loss for binary classification tasks.
- $L(w)=(-1/m) * \sum_{i=1}^{m}[y_i \log p_i + (1-y_i)\log(1-p_i)]$ (this is mean cross entropy loss)
- Only Cross Entropy Loss can be computed as:
  - $L(w)=\sum_{i=1}^{m}[y_i \log p_i + (1-y_i)\log(1-p_i)]$
  - Cross Entropy Loss for a single sample:
    - $L(w) = - [y_i \log p_i + (1-y_i)\log(1-p_i)]$
- Where $p_i = \sigma(w^T * x_i + b)$

Gradient Calculation

- $\partial J/\partial w = (1/m) \Sigma_{i=1}^{m} (p_i - y_i)x_i$
- If there only one feature x, then $p_i = \sigma(w_1 * x_i + w_0)$
  - parameter updates:
    - $\partial J/\partial w_1 = \Sigma_{i=1}^{m} (p_i - y_i)x_i$
    - $\partial J/\partial w_0 = \Sigma_{i=1}^{m} (p_i - y_i)$

Update Weights: Gradient Descent

- Apply the gradient descent as explained in the earlier slides.
- $\mathbf{w_{new} = w_{old} - \eta \nabla L(w)}$

Worked Example: 1-D Logistic Regression Update

- Suppose initial weight w_1=0, w_0=0, points: (x=2, y=1), (x=-1, y=0) and η=0.1.
- Compute prediction: $p(2) = 1/(1+e^{-(0*2+0)}) = 0.5$, $p(-1) = 0.5$
- Gradients:
  - For w_1:
    - (0.5 - 1) * 2 = -1.0
    - (0.5 - 0) * (-1) = -0.5
    - Sum of Gradients = -1.0 - 0.5 = -1.5
  - For w_0
    - (0.5 - 1) = -0.5
    - (0.5 - 0) = 0.5
    - Sum of Gradients = -0.5 + 0.5 = 0
- Update
  - $w_1 = 0 - 0.1 * (-1.5) = 0.15$
  - $w_0 = 0 - 0.1 * 0 = 0$
- Repeat next epoch with updated weights.
- The implementation is given in logistic_regression.ipynb

Evaluation Metrics

- Logistic regression is used for classification tasks.
- Evaluation Metrics for Classification Tasks
  - Precision
  - Recall
  - F1-Score
  - Accuracy (May be misleading for imbalanced classes)

References

1. Gradient Descent Visualized - https://www.youtube.com/watch?v=kkfjQ65CpKI
2. Gradient Descent Blog - https://blog.skz.dev/gradient-descent
3. Linear Regressoion - https://www.geeksforgeeks.org/machine-learning/ml-linear-regression/
4. Logistic Regression Loss Function - https://www.geeksforgeeks.org/machine-learning/ml-cost-function-in-logistic-regression/
5. Logistic Regression - https://www.geeksforgeeks.org/machine-learning/understanding-logistic-regression/