Machine Learning

Dr.Hajialiasgari

Tehran University
Of
Medical Science

January 12, 2025

TEHRAN UNIVERSITY
OF
MEDICAL SCIENCES

Introduction To Regression

- The term "Regression" was coined by Francis Galton in the 19th century to describe a biological phenomenon. The phenomenon observed was that the height of descendants of tall ancestors tends to regress downward. For Galton, regression had only this biological meaning, but his work was later extended to a broader statistical context by Udny Yule and Karl Pearson.

- A model in machine learning works like a function: it takes input data, processes it based on learned patterns, and produces an output, such as predictions or classifications.

Hypothesis Set

Definition: The set of all possible models or functions $h(x,\theta)$ that the learning algorithm can choose from.

Mathematics:

$$h(x,\theta) = f(x_1, x_2, \ldots, x_n; \theta_1, \theta_2, \ldots, \theta_m)$$

- x: Input features.
- θ: Model parameters (e.g., weights, biases).

Example (Linear Regression):

$$h(x,\theta) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Learning Model Overview

Hypothesis Set: A collection of functions $h(x, \theta)$ that maps input x to output y, parameterized by θ.

$$h(x,\theta) = f(x_1, x_2, \ldots, x_n; \theta_1, \theta_2, \ldots, \theta_m)$$

Linear Hypothesis Space

- **Simplest form:** Linear combination of input features.

$$h_w(x) = w_0 + \sum_{i=1}^{D} w_i x_i$$

- **Linear Hypothesis Examples:**
  - **Single Variable:**

$$h_w(x) = w_0 + w_1 x$$

  - **Multivariate:**

$$h_w(x) = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_D x_D$$

Understanding Cost Functions

- In the hypothesis space, we select a function $h(x; w)$ to approximate the true relationship between input x and output y.
- The objective is to minimize the difference between predicted values $h(x)$ and actual values y.
- This difference is quantified using **cost functions**, which guide us in choosing the optimal hypothesis.

What is a Cost Function?

- A cost function measures how well the hypothesis $h(x; w)$ fits the training data.
- In regression problems, the most common error function is the **Squared Error (SE)**:

$$SE = \left(y^{(i)} - h(x^{(i)}; w)\right)^2$$

- The cost function should measure all predictions. Thus, a possible choice is the **Sum of Squared Errors (SSE)**:

$$J(w) = \sum_{i=1}^{N} \left(y^{(i)} - h(x^{(i)}; w)\right)^2$$

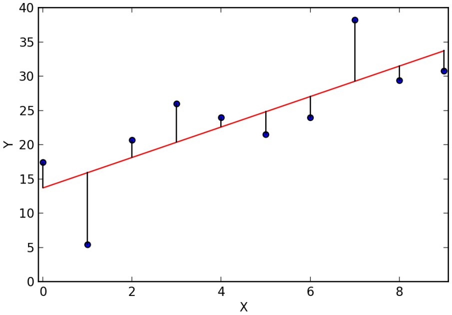
- **Objective:** Minimize the cost function to find the best parameters w.

SSE: Sum of Squared Errors

- **SSE** is widely used due to its simplicity and differentiability.
- Intuitively, it represents the squared distance between predicted and true values.
- Penalizes larger errors more severely than smaller ones (due to the square).
- For linear regression, it can be written as:

$$SSE = \sum_{i=1}^{N} \left(y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)} \right)^2$$

How to measure the error



$$J(w) = \sum_{i=1}^{n} \left(y^{(i)} - h_w(x^{(i)}) \right)^2 = \sum_{i=1}^{n} \left(y^{(i)} - w_0 - w_1 x^{(i)} \right)^2$$

Cost Function Optimization

The goal of linear regression is to minimize the sum of squared errors (SSE) between the predicted values and the actual values.

$$J(w) = \sum_{i=1}^{n} \left(y^{(i)} - w_0 - w_1 x^{(i)} \right)^2 \tag{1}$$

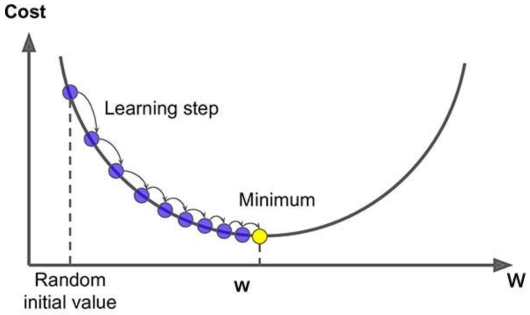
Condition For Optimal Value

To find the optimal values of w_0 and w_1, we need to solve the following partial derivatives and set them to zero:

$$\frac{\partial J(w)}{\partial w_0} = 0 \tag{2}$$

$$\frac{\partial J(w)}{\partial w_1} = 0 \tag{3}$$

Optimization



What is Regression?

Regression is a statistical method used to model the relationship between a dependent variable (output) and one or more independent variables (inputs). The goal is to predict the output based on input features.

Linear Regression is a type of regression where the relationship between variables is modeled as a straight line. The formula is:

$$Y = w_0 + w_1 X \tag{4}$$

Where:

- Y: Predicted value (dependent variable)
- X: Input feature (independent variable)
- w_0: Intercept (bias term)
- w_1: Slope (weight)

Common Example: Predicting House Prices

Problem: A real estate agent wants to predict the price of a house based on its size.

Given Data:

Size (sq ft)	Price (USD)
1000	200,000
1500	300,000
2000	400,000
2500	500,000

Solution: Using linear regression, we fit the line:

$$Y = w_0 + w_1 X \tag{5}$$

Steps:

- Calculate w_1 (slope): $w_1 = \frac{\sum(X-\bar{X})(Y-\bar{Y})}{\sum(X-\bar{X})^2}$
- Calculate w_0 (intercept): $w_0 = \bar{Y} - w_1 \bar{X}$

Medical Example: Predicting Blood Pressure

Problem: A doctor wants to predict a patient's systolic blood pressure based on their age.

Given Data:

Age (years)	Blood Pressure (mmHg)
30	120
40	130
50	140
60	150

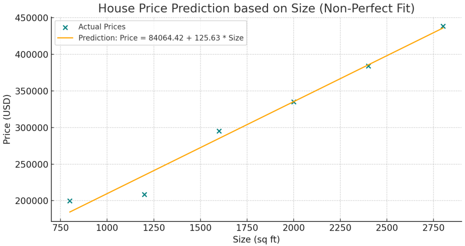
Solution: Using linear regression, we fit the line:

$$Y = w_0 + w_1 X \tag{6}$$

Steps:

- Calculate w_1 (slope): $w_1 = \frac{\sum(X-\bar{X})(Y-\bar{Y})}{\sum(X-\bar{X})^2}$
- Calculate w_0 (intercept): $w_0 = \bar{Y} - w_1\bar{X}$

House Price Prediction Plot



What is Polynomial Regression?

Polynomial Regression is an extension of linear regression where the relationship between the independent variable X and the dependent variable Y is modeled as an n^{th} degree polynomial.

The formula for polynomial regression is:

$$Y = w_0 + w_1 X + w_2 X^2 + \cdots + w_n X^n \tag{7}$$

Where:

- Y: Predicted value (dependent variable)
- X: Input feature (independent variable)
- $w_0, w_1, \ldots, w_n$: Coefficients to be learned

Why use Polynomial Regression?

- To capture non-linear relationships in the data.
- Linear regression fails to fit curves, while polynomial regression can.

Example: Predicting Tumor Growth

Problem: A researcher wants to model the relationship between time (days) and the size of a tumor (in cm).

Given Data:

Time (days)	Tumor Size (cm)
1	2.1
2	4.5
3	8.0
4	14.2
5	23.5
6	35.8

Solution Steps

Using polynomial regression, we fit a curve:

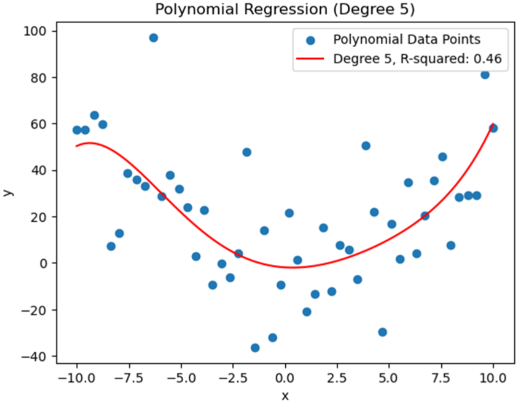
$$Y = w_0 + w_1 X + w_2 X^2 \tag{8}$$

Steps:

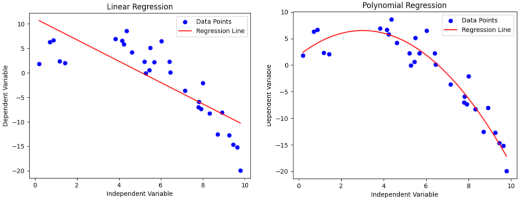
- Calculate coefficients w_0, w_1, w_2 using least squares method.
- Plot the curve and observe the fit.

Result: The model captures the non-linear growth pattern of the tumor.

Polynomial Regression Plot



Why Polynomial?



What is Evaluation?

Evaluation in machine learning is the process of measuring how well a model performs on a given task. It helps to:

- Assess model accuracy.
- Identify overfitting or underfitting.
- Compare different models.

Why Choose the Right Metric?

- Different tasks (regression, classification) require different evaluation methods.
- Incorrect metrics can lead to wrong conclusions.
- Metrics must align with business goals and data characteristics.

Mean Absolute Error (MAE)

Formula:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{9}$$

When to Use:

- When you want to measure the average magnitude of errors.
- Suitable when all errors have the same importance.

Pros: Easy to interpret. **Cons:** Less sensitive to large errors.

Mean Squared Error (MSE)

Formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{10}$$

When to Use:

- When you want to penalize larger errors more.
- Common in regression problems.

Pros: Penalizes large errors more than MAE. **Cons:** Sensitive to outliers.

Root Mean Squared Error (RMSE)

Formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{11}$$

When to Use:

- When you want to interpret the error in the same unit as the target variable.
- Useful for comparing models.

Pros: Easy to interpret. **Cons:** Sensitive to outliers.

R-Squared (R^2)

Formula:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{12}$$

When to Use:

- To measure how well the model explains the variance in the data.
- Useful for determining model fit.

Pros: Provides a normalized score. **Cons:** Can be misleading for non-linear models.

Summary of Evaluation Metrics

Metric	Formula	When to Use
MAE	$\frac{1}{n}\sum \lvert y_i - \hat{y}_i \rvert$	Average magnitude of errors
MSE	$\frac{1}{n}\sum (y_i - \hat{y}_i)^2$	Penalizes larger errors more
RMSE	$\sqrt{\frac{1}{n}\sum (y_i - \hat{y}_i)^2}$	Same unit as target variable
R^2	$1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$	Explains variance in the data

What is Generalization?

Generalization in machine learning refers to a model's ability to perform well on unseen data that it has not encountered during training. It is a measure of how well the model can apply the knowledge learned from the training data to new, real-world scenarios.

Key Concepts:

- **Training Error:** The error a model makes on the training data.
- **Test Error:** The error a model makes on unseen data.
- **Overfitting:** When a model performs well on training data but poorly on test data.
- **Underfitting:** When a model performs poorly on both training and test data.

Why is Generalization Important?

- A well-generalized model can make accurate predictions on new, unseen data.
- It ensures that the model is not just memorizing the training data but learning patterns that apply broadly.
- Poor generalization leads to models that either overfit or underfit the data, resulting in poor performance in real-world applications.

Example: A model trained to predict house prices should perform well not only on the training dataset but also on new properties it hasn't seen before.

Overfitting vs. Underfitting

Overfitting:

- Model captures noise and irrelevant details in the training data.
- High training accuracy but poor test accuracy.
- Solution: Regularization, pruning, cross-validation.

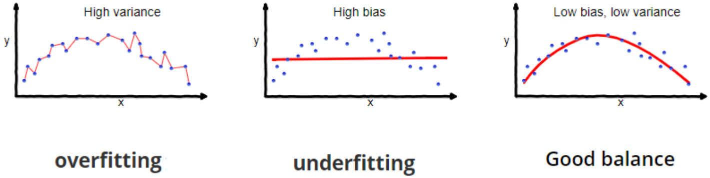
Underfitting:

- Model fails to capture the underlying pattern in the data.
- Both training and test accuracy are low.
- Solution: Use a more complex model, feature engineering.

How to Improve Generalization?

- **Cross-Validation:** Split the dataset into multiple folds and validate the model on different subsets.
- **Regularization:** Add a penalty term to the loss function to prevent the model from becoming too complex.
- **Early Stopping:** Stop training when the performance on a validation set stops improving.
- **Data Augmentation:** Increase the diversity of the training data by creating variations of the existing data.
- **Dropout:** Randomly drop units in a neural network during training to prevent overfitting.

Variance Bias



What is Regularization?

Regularization is a technique used in regression models to reduce overfitting by adding a penalty term to the model's loss function. It prevents the model from becoming too complex and helps improve generalization to unseen data.

Why Use Regularization?

- To prevent overfitting.
- To ensure the model performs well on both training and test data.
- To improve model interpretability by reducing overly complex solutions.

Regularized Linear Regression Formula

The objective in linear regression is to minimize the cost function:

$$J(w) = \sum_{i=1}^{n} \left(y^{(i)} - \left(w_0 + w_1 x^{(i)} \right) \right)^2 \tag{13}$$

In regularized regression, a penalty term is added to the cost function to prevent large coefficient values:

$$J(w) = \sum_{i=1}^{n} \left(y^{(i)} - \left(w_0 + w_1 x^{(i)} \right) \right)^2 + \lambda R(w) \tag{14}$$

where:

- λ is the regularization parameter.
- $R(w)$ is the regularization term.

Types of Regularization

There are two main types of regularization used in regression:

- **L1 Regularization (Lasso)**
- **L2 Regularization (Ridge)**

Both add a penalty term to the cost function, but they do so in different ways.

L1 Regularization (Lasso)

L1 regularization adds the sum of the absolute values of the coefficients to the cost function:

$$J(w) = \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda \sum_{j=1}^{p} |w_j| \tag{15}$$

Effect: Encourages sparsity by shrinking some coefficients to exactly zero, effectively performing feature selection.

L2 Regularization (Ridge)

L2 regularization adds the sum of the squared values of the coefficients to the cost function:

$$J(w) = \sum_{i=1}^{n} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 + \lambda \sum_{j=1}^{p} w_j^2 \tag{16}$$

Effect: Prevents large coefficients by penalizing their magnitude, but does not reduce them to zero.

For more information and code check the related notebook

End of Regression