CS412 - Machine Learning - 2025 Fall
Homework 2

Deadline: *November 9, 2025 @11pm*
Late homeworks will be accepted for 3 days with -10pt late penalty per day

Submission Guidelines

Jupyter Notebook: Must include all code cells and outputs. (The notebook will not be re-run during grading.)

File Naming: `CS412-HW2-YourName.ipynb`

Late Policy: Accepted until Nov. 12 at 11pm. 10 points will be deducted per day - no exceptions!

Notebook versus Quiz (IMPORTANT):

- You will be required to answer questions related to your findings and interpretations in the corresponding Quiz (HW2-Submission) for auto grading. Add code to produce the requested information (e.g. test MSE) in your notebook and answer the corresponding questions on the HW2-Submission quiz.

- Report all numerical answers in 4 significant digits after decimal point and DO NOT ROUND the numbers.

Overview

This assignment is divided into two parts. In the first part, you will build a logistic regression model on a given dataset using the gradient descent algorithm.

In the second part, you will perform polynomial regression on given datasets and observe how the degree of the polynomials affects variance and bias. You will then perform regularization and analyze how regularization affects model fit.

Datasets

The datasets will be generated for you in both parts within the starter notebook. You can access the **starter notebook** via the following link: **LINK TO THE STARTER NOTEBOOK**

Do not change anything in the dataset generation sections of the notebook, so that training results are deterministic.

Part 1 – Logistic Regression

You are given a 2D, 2-class dataset containing 100 data points (50 from each class). The data points are sampled from the given distributions in the given starter notebook and shown in Fig. 1. Your task is to find the parameters w_0, w_1, w_2 for a logistic regression model by running the gradient descent algorithm.

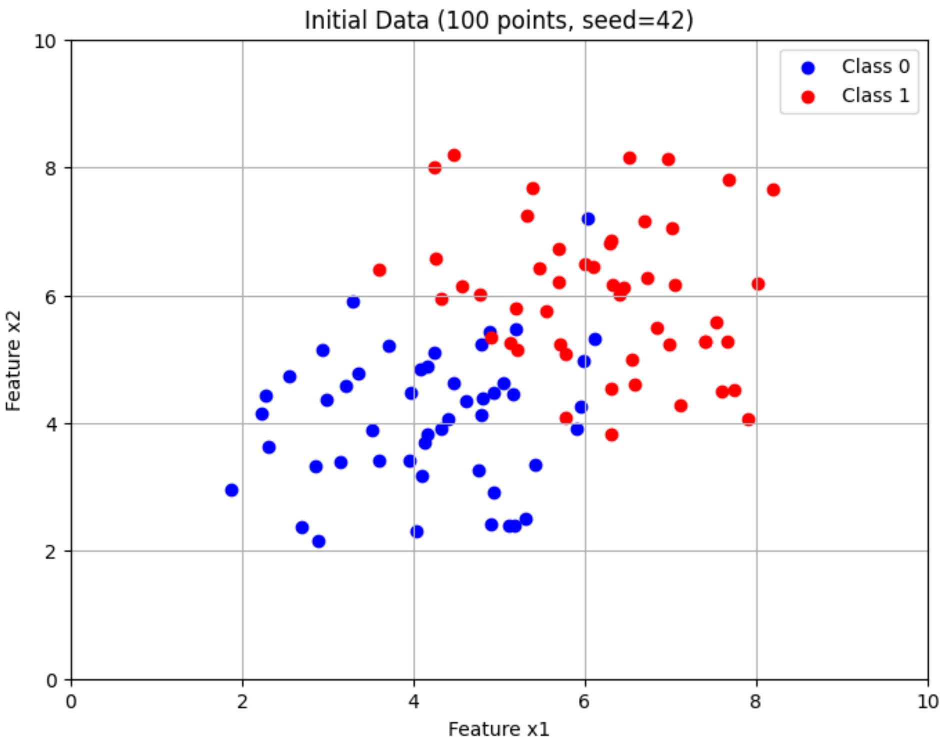

Figure 1: The input dataset.

Initial Setup

- **Dataset:** 100 points (50 in class 0, 50 in class 1), generated by the provided Python script with `seed=42` as in Figure 1.
- **Initial Parameters:** $\mathbf{w}_{t=0} = [w_0, w_1, w_2] = [0.0, 0.0, 0.0]$
- **Step Size (Learning Rate):** $\eta = 0.1$

Model Formulas

1. Prediction ($\hat{y}$): First, calculate the linear score z_i for each data point i. Then, use the sigmoid function σ to "squash" this score into a probability $\hat{y}_i$ between 0 and 1.

$$\hat{y}_i = \sigma(z_i) = \sigma(w_1 x_{i,1} + w_2 x_{i,2} + w_0) = \frac{1}{1 + e^{-z_i}}$$

2. Cost Function (Binary Cross-Entropy): This function J measures the average error of the model across all $m = 100$ data points. Your goal is to minimize this value.

$$J(w_i) = -\frac{1}{m} \sum_{i=1}^{m} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

3. Gradient Formulas: To minimize the cost, you must calculate its gradient with respect to each parameter. Use the predictions $\hat{y}_i$ from all m data points.

$$\frac{\partial J}{\partial w_j} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i) x_{i,j} \quad (\text{for } j = 1, 2)$$

$$\frac{\partial J}{\partial w_0} = \frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i)$$

4. Gradient Descent Update Rule: After calculating the gradients, update the parameters by taking one step "downhill" using the learning rate η.

$$w_j \leftarrow w_j - \eta \frac{\partial J}{\partial w_j}$$

To Do

Experiment with different stopping conditions to strengthen your understanding (normally we would not use a fixed number of iterations, since training will be fast):

1. **Fixed number of iterations:** Starting from the initial parameters, run the algorithm for exactly **10 iterations**. Report the following in the corresponding quiz question (final parameters $[w_0, w_1, w_2]_{t=10}$). Plot the final line and report the error rate on the plot, in your notebook.

2. **Until low error:** Starting from the initial parameters, run the algorithm until the **cost J changes by less than** 10^{-5}. Report the following in the corresponding quiz question (final parameters $[w_0, w_1, w_2]_{t=10}$). Plot the final line fit and report the error rate on the plot, in your notebook.

See the important notes on the first page.

Part 2 – Polynomial Regression with Regularization

The dataset used in the second part consists of two training sets (D1 and D2) and a single test set, all generated assuming an underlying sinusoidal function:

$$y = \sin(1.5x) + \epsilon$$

Zero-mean Gaussian noise is added to each set to better simulate real-world data and to introduce variation between the training sets.

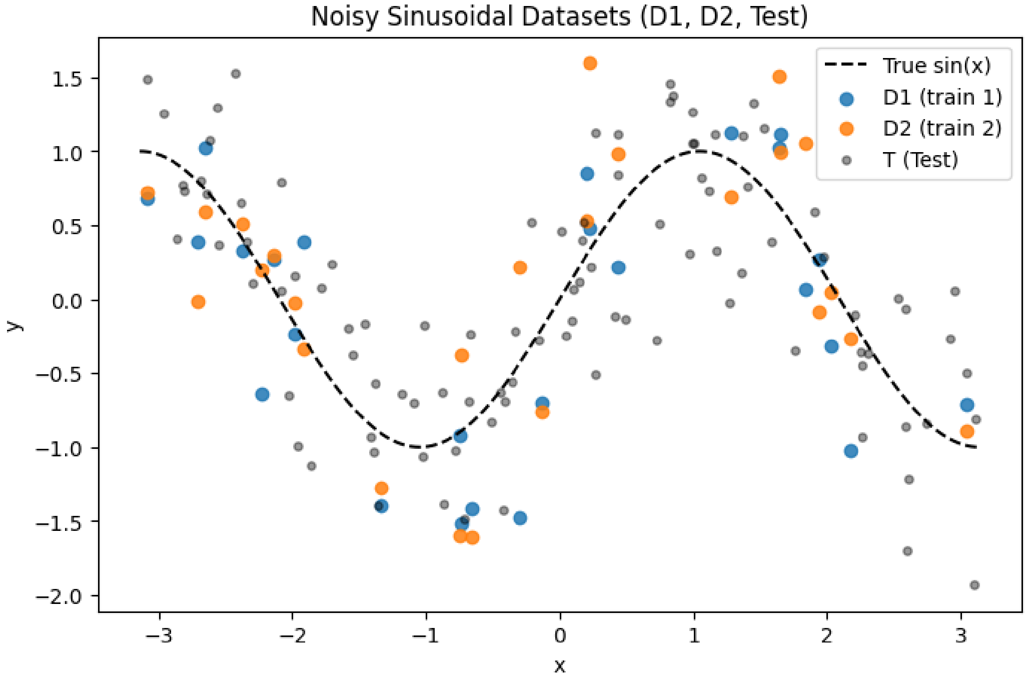

Figure 2: Plots of the datasets, as well as the original sinusoidal function.

This part is subdivided into two sections.

- In the first section, you will implement 3rd and 5th-degree polynomial fits to the training datasets and evaluate your models using the given test set. You will observe how the degree of the polynomial affects model variance and how the fits depend on model complexity.
- In the second section, you will implement Ridge regression to the 5th degree polynomial and only dataset D1, to observe how different regularization coefficients (α) affects model fit.

  Report **Mean Squared Error (MSE)** performance metric as requested.

Model Formulas

- **Polynomial Regression Formulation:** A polynomial regression model of degree d expresses the target variable y as a weighted sum of polynomial terms of the input variable x:

$$f(x) = w_0 + w_1 x + w_2 x^2 + \cdots + w_d x^d = \sum_{i=0}^{d} w_i x^i$$

- **Ridge Regularization:** Ridge regression adds an L_2 penalty proportional to the sum of squared coefficients.

$$\text{Loss}_{\text{Ridge}} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{p} w_j^2$$

To Do

1. **Model Complexity (Bias–Variance Analysis):** Train 3rd-degree and 5th-degree polynomial regression models using the two datasets $D1$ and $D2$ separately. Evaluate each model on the **common test set** using the MSE metric. Plot the trained models together with the original sinusoidal function in your notebook, producing two figures:

   - 3rd-degree model on $D1$ and $D2$ (in the same plot)
   - 5th-degree model on $D1$ and $D2$ (in the same plot)
   - Observe their fit and variance visually.

   Important: Answer corresponding questions on the quiz. See the Notebook vs Quiz item on the first page. Your notebook should have all requested output, but only some of the answers will be asked on the quiz tool.

2. **Regularization with Ridge Regression:**

   Apply **ridge regression** on $D1$ using the 5th-degree polynomial model with the α values:

$$\alpha = [0, 0.5, 1, 1.5, 5]$$

   Train the model for each α value. Evaluate each model on the **common test set** using the MSE metric. Plot the trained models together with the original sinusoidal function in your notebook, producing five figures (one for each α).

   Create a table that reports both **Train MSE** and **Test MSE** for each α, including zero. Include your table in the notebook only.

   Important: Answer corresponding questions on the quiz. See the Notebook vs Quiz item on the first page. Your notebook should have all requested output, but only some of the answers will be asked on the quiz tool.