SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

50.007 Machine Learning, Spring 2026
Homework 1

Due 27 February 2026, 11.59pm

In this homework, we would like to look at **Linear Classification** including perceptron algorithm and hinge loss, and **Regression** including linear regression and ridge regression.

Question 1: Perceptron Algorithm and Initialization Effects [20 points]

Consider a toy binary classification dataset consisting of four samples in $\mathbb{R}^2$ with labels $y \in \{-1, 1\}$. We use the augmented feature vector $\tilde{\mathbf{x}} = [1, x_1, x_2]^T$ to include the bias term θ_0 directly in the weight vector $\tilde{\theta} = [\theta_0, \theta_1, \theta_2]^T$ as discussed in class.

The Dataset

- $\mathbf{x}_1 = (1, 2)^T, y_1 = 1$
- $\mathbf{x}_2 = (2, 1)^T, y_2 = 1$
- $\mathbf{x}_3 = (0, 0)^T, y_3 = -1$
- $\mathbf{x}_4 = (1, 0)^T, y_4 = -1$

Tasks

Run the Perceptron algorithm on this dataset. You will perform this for two different initializations of the weight vector $\tilde{\theta}^{(0)}$:

1. **Initialization A:** $\tilde{\theta}_A^{(0)} = [1, 1, 1]^T$
2. **Initialization B:** $\tilde{\theta}_B^{(0)} = [0, -1, 1]^T$

For both initializations, process the points in the fixed order $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)$. If a point is misclassified (*i.e.*, $y_i(\tilde{\theta} \cdot \tilde{\mathbf{x}}_i) \leq 0$), update the weights. Repeat passes through the data until the algorithm converges.

Required:

1. Explicitly write out the Perceptron update steps for each initialization, showing how the parameters change after each misclassified example. [10 points]
2. Write down the corresponding decision boundary equations for both final solutions. [4 points]
3. Briefly discuss how different initializations lead to different update iterations and different separating hyperplanes in the Perceptron algorithm, and explain how these relate to the Perceptron's convergence guarantee regarding uniqueness. [6 points]

Question 2: Linear Classification on the Iris Dataset (Code Exercise) [30 points]

In this exercise, you will use the Iris flower dataset, a classic in machine learning. You are tasked with building a binary classifier to distinguish between two species: Iris *Versicolor* and Iris *Virginica* based on their physical measurements. To simplify the problem and allow for 2D visualization, you will use two specific features: *Petal Length* (cm) and *Petal Width* (cm).

Instructions:

- The training and test datasets are provided in the files `iris_train.csv` and `iris_test.csv`, respectively. Each row in the CSV files represents an example.
- Ensure that you extract the two features: *Petal Length* (cm) and *Petal Width* (cm).
- Ensure that the labels are converted: Iris *Versicolor* becomes +1 and Iris *Virginica* becomes -1 for binary classification.

Tasks

(1) **Implement SSGD with Hinge Loss with Offset.** Write a function to perform stochastic sub-gradient descent. [6 points]
(2) **Train and Evaluate. Important: Set `np.random.seed(42)` before training.** Use a learning rate $\eta = 0.05$ and train for $T = 800$ iterations.

- Report final weights $\hat{\theta}_0, \hat{\theta}_1, \hat{\theta}_2$. [3 points]
- Record the training loss and test loss every 80 iterations. Plot these values against the iteration number and discuss the stability of the model. [8 points]
- Explain how the learning rate $\eta = 0.05$ affects the speed of convergence compared to a much smaller rate (*e.g.*, 0.005) [6 points]

(3) **Perceptron Comparison.** If you were to use the Perceptron algorithm from Question 1 on this Iris dataset. Would the algorithm ever terminate? Why or why not? [7 points]

Question 3: Multi-Output Linear Regression and Optimization [20 points]

Up until now, our discussions have been limited to scenarios where the target variable y is a singular scalar quantity. In many applications (as discussed in the page 6 of lecture 4), we wish to predict multiple target variables $\mathbf{y} \in \mathbb{R}^m$ simultaneously from a set of features $\mathbf{x} \in \mathbb{R}^d$. Now, let us derive the analytical solution (*i.e.*, closed-form solution) for regression with multiple target variables and analyze the computational trade-offs between exact and iterative solvers (*i.e.*, closed-form solution *v.s.* SGD)

Suppose we have a dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{y}_i \in \mathbb{R}^m$. We wish to learn a weight matrix $\mathbf{W} \in \mathbb{R}^{d \times m}$ (assuming for simplicity the data is centered and there is no bias term) that minimizes the total squared error across all m outputs. The objective function is defined as:

$$J(\mathbf{W}) = \frac{1}{2} \sum_{i=1}^{n} \|\mathbf{y}_i - \mathbf{W}^T \mathbf{x}_i\|_2^2$$

.

Note that this objective function can be written in terms of the **Frobenius norm**:

$$J(\mathbf{W}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2$$

Hint: $\|\mathbf{A}\|_F^2 = Tr(\mathbf{A}^T\mathbf{A})$.

Tasks

(1) The Multi-Output Closed-Form Solution [10 points]
Derive the gradient $\nabla_{\mathbf{W}} J(\mathbf{W})$ and find the closed-form expression for the optimal $\mathbf{W}^*$. Discuss how the computational cost of this solution scales with the number of outputs m.

(2) Analysis: Closed-Form vs. SGD [10 points]
Suppose you implement Stochastic Gradient Descent to solve this problem.

- Derive the SGD update rule for the matrix $\mathbf{W}$ given a **single** random sample $(\mathbf{x}_i, \mathbf{y}_i)$.
- Explain under what conditions (in terms of n, d, and m) you would prefer the closed-form solution over SGD, and vice versa.

Question 4: Ridge Regression – Optimization and Model Selection (Code Exercise) [30 points]

In this exercise, you will implement Ridge Regression using both exact (closed-form) and iterative (Stochastic Gradient Descent) methods. You will analyze their differences and perform hyperparameter tuning using cross-validation.

You will use the Diabetes dataset, which contains measurements for $n = 442$ diabetes patients. The dataset includes ten baseline variables (age, sex, body mass index, average blood pressure, and six blood serum measurements) and a target variable representing a quantitative measure of disease progression one year after baseline.
Note: We will use the Mean Squared Error (MSE) to evaluate the training and testing performance. MSE is defined as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

where N indicates the number of training/testing samples, y_i is the true target, and $\hat{y}_i = w^T x_i$ is the predicted value.

Tasks

1. Implementation of Exact and Iterative Solvers [10 points]

Implement two versions of Ridge Regression. Assume the input data matrix $X \in \mathbb{R}^{n\times d}$ has been standardized (mean 0, variance 1) and includes a column of ones to account for the bias term.

1. **Closed-Form Solution:** Implement a function that computes the optimal weight vector w^* using the analytical solution: [5 points]

$$w^* = (X^T X + \lambda I)^{-1} X^T y$$

2. **Stochastic Gradient Descent (SGD):** Implement an iterative solver that minimizes the following Ridge objective function: [5 points]

$$J(w) = \frac{1}{2n}\|y - Xw\|_2^2 + \frac{\lambda}{2}\|w\|_2^2$$

For each iteration, sample a random example (x_i, y_i) and perform the following update step:

$$w \leftarrow (1 - \lambda\eta)w + \eta(y_i - w^T x_i)x_i$$

Repeat this update until the maximum number of iterations T_{max} is reached.

2. Experiments & Analysis [10 points]

Ensure `np.random.seed(42)` is set. Fix the hyperparameters as follows: regularization coefficient $\lambda = 1$ and learning rate $\eta = 0.01$.

- Train the SGD model on the provided `diabetes_train.csv` dataset using three different stopping criteria: $T_{max} \in \{10,000, 20,000, 30,000\}$. [3 points]
- Compute the **Training MSE** and **Test MSE** (using `diabetes_test.csv`) for each of the three SGD models. [3 points]
- Compute the **Training MSE** and **Test MSE** using the optimal w^* from the **Closed-Form Solution**. [1 points]

Discussion: Compare the performance (training and testing error) of the iterative SGD solver at different T_{max} levels against the exact closed-form solution. Discuss the trade-offs between these methods: which solver is more practical for this specific task and dataset size, and why? [3 points]

3. Model Selection via 5-Fold Cross-Validation [10 points]

Using only the `diabetes_train.csv` dataset:

1. Implement **5-fold cross-validation** from scratch to find the optimal regularization parameter λ from the candidate set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2\}$. **Ensure the fold splitting is reproducible by setting the random seed.** [5 points]
2. **Visualization:** Plot the MSE for both the **Training set** and the **Validation set** against $\log_{10}(\lambda)$. [2 points]
3. **Analysis:** On your plot, explicitly identify and label the regions of **underfitting** (high bias) and **overfitting** (high variance). Report the optimal λ^* and the corresponding MSE on the held-out test set (`diabetes_test.csv`). [3 points]

Important Instructions for Homework Submission

- **Submission Format:** Please submit your homework as a single **Jupyter Notebook (.ipynb)** file. The notebook should contain your code, the resulting outputs (plots, print statements), and your written answers (markdown cells) for theoretical questions.
- **Reproducibility:** For all coding tasks that involve random initialization or sampling (e.g., Stochastic Gradient Descent, Cross-Validation), you **MUST** set the random seed to **42** at the beginning of your code to ensure reproducibility.

```
import numpy as np
np.random.seed(42)
```

- **Output:** Ensure that you run all cells in your notebook before submission so that the output (plots, loss values, etc.) is visible.