SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

50.007 Machine Learning, Spring 2025
Homework 2

Due 18 March 2025, 11.59pm

In this homework, we will look at K-means Clustering, K-medoids Clustering, Support Vector Machines, Logistic Regression, and Neural Networks.

1 Code Exercise: K-means *vs.* K-medoids Clustering [30 points]

In this task, you will evaluate and compare the clustering performance of K-means and K-medoids on two synthetic datasets. First, generate a dataset by creating three groups of 40 observations each in 5 dimensions. The cluster centers should be at (0, 0, 0, 0, 0), (5, 5, 5, 5, 5), and (11, 11, 11, 11, 11), respectively, with each group having a diagonal covariance matrix with variance $\sigma^2 = 2$ in every dimension, yielding a total of 120 points. Then, modify this dataset by adding three outlier observations: $(-12, 5, 0, 0, 0)$, $(100, 11, 11, 5, 5)$, and $(-100, 0, 0, 0, -100)$, so that the total number of points becomes 123.

(a) [10 points] Comparing Cost Functions and Cluster Assignments: Implement the K-means clustering using Lloyd's Algorithm, where points are assigned to the nearest centroid and centroids are updated as the mean of the points in each cluster (refer to page 20 of Lecture 5 slides). For K-medoids clustering, implement the solution using Lloyd's Algorithm (a.k.a. Alternate), where points are assigned to the nearest medoid and medoids are updated by minimizing the within-cluster distance (refer to page 9 of Lecture 6 slides).

Use the squared Euclidean distance for K-means and Manhattan distance (L1 distance) for K-medoids. For both datasets (with 120 and 123 points), perform clustering using K-means and K-medoids with $k = 3$. Compare the resulting cluster assignments and compute the total cost function for each method. In your analysis, discuss which method is more affected by the presence of outliers.

(b) [10 points] Impact the Choice of k: Using the modified dataset with 123 points, generate an "elbow graph" for both clustering methods by varying k from 2 to 10. For K-means, plot k versus the total within-cluster sum of squared distances, and for K-medoids, plot k versus the total within-cluster sum of absolute distances. Compare and contrast the two plots. Based on your observations, indicate which value of k you would choose and explain your reasoning. Also, comment on whether the outliers influence the optimal k differently for the two methods.

(c) [10 points] Sensitivity to Initialization: Run K-means clustering 20 times on the modified dataset (n = 123) with $k = 3$, using different random initializations for each run. Compute the variance of the total cost function across these runs and compare this variability with that observed for K-medoids. Discuss which method is generally less sensitive to initialization and explain why. (Hint: Reflect on how each method updates its cluster representative.)

2 Soft-Margin SVM with Kernel Methods [25 points]

Consider a training set $\{(x_i, y_i)\}_{i=1}^n$ with $y_i \in \{-1, 1\}$, and let $K(x, x')$ be a kernel function with an associated feature mapping $\phi(x)$ satisfying

$$K(x, x') = \phi(x) \cdot \phi(x').$$

In class, we discussed the primal and dual problems for the linear SVM with the kernel method. Now, we will focus on the soft-margin SVM with kernel methods. To account for potential misclassifications in the ϕ-coordinates, we introduce slack variables $\xi_i \geq 0$ and a regularization parameter $C > 0$.

(a) [15 points] Primal and Dual Problems: Write down the corresponding primal and dual problems for the soft-margin SVM when the kernel method is applied. First, formulate the primal problem for the soft-margin SVM with the kernel method. Then, using Lagrange multipliers and the KKT conditions, derive the dual problem formulation for the soft-margin SVM. *Note: We expect you to go through the mathematical steps in detail (e.g., write down the Lagrangian function) and show the derivations (e.g., deriving the stationarity condition) step by step.*

(b) [10 points] Optimal parameters and Support Vectors: Show how the optimal parameters $\hat{\theta}$ and $\hat{\theta}_0$ can be expressed as functions of optimal $\hat{\alpha}_i$. *Note that you should derive the expression for $\hat{\theta}_0$ in terms of $\hat{\alpha}_i$.* Next, derive the mathematical definition of support vectors by making the optimal $\hat{\alpha}$ satisfy the KKT conditions.

3 Code Exercise: SVM *vs.* Logistic Regression [20 points]

Download and install the widely used machine learning package `scikit-learn` (along with other standard Python packages such as `numpy`, `pandas`, and `matplotlib`). You will use the Breast Cancer Wisconsin (Diagnostic) dataset for this task. Training data and test data are stored in `train.csv` and `test.csv`, respectively, provided in the attached folder `breast cancer dataset`. There are 455 training samples and 114 testing samples. Each row of the CSV file corresponds to one sample with 30 dimensional features and 1 target label.

(a) [10 points] SVM with Different Kernels: Train your SVM with different kernels on the Breast Cancer dataset:

- Use the linear kernel.
- Use the polynomial kernel with $degree = 10$.
- Use the RBF kernel with $gamma = 0.0001$.

For each kernel, report the corresponding accuracy on the testing data.

(b) [10 points] Logistic Regression and Comparative Analysis: Train a Logistic Regression model on the same dataset using scikit-learn and report its accuracy on the testing data. Then compare its performance with that of the SVM models trained in part (a), focusing primarily on the test accuracy. Based on your experimental results, indicate which method you would choose for this task and provide a clear justification for your decision.

4 Code Exercise: Neural Network Architecture Exploration and Decision Boundary Visualization [25 points]

In this task, you will investigate how different neural network architectures affect a synthetic binary classification problem. Your goal is to understand how the network depth, width, and activation functions impact the training / test performance and the model decision boundary. Use the following baseline configuration: a network with 1 hidden layer of 2 neurons using ReLU activation, a learning rate of 0.01, and 100 training epochs.

First, install the necessary Python libraries (TensorFlow, numpy, matplotlib, and scikit-learn). Generate the dataset using scikit-learn's `make_circles` function with 1000 samples, noise 0.1, factor 0.5, and random state 42 (yielding feature matrix X and label vector y). Then split the data set into training and test sets (using an 80/20 split).

(a) [15 points] Neural Network Experiments: Begin by implementing the baseline model (depth = 1, width = 2, activation = ReLU). Then, run three sets of experiments:

1. **Vary Network Depth:** Increase the number of hidden layers (e.g. 2 or 3 layers) while keeping the baseline width and activation.

2. **Vary Network Width:** Change the number of neurons per hidden layer (e.g. 8, 16, etc.) while using the baseline depth and activation.

3. **Vary Activation Functions:** Test different activation functions (e.g. tanh and sigmoid, in addition to ReLU) while keeping the baseline depth and width.

For each configuration, train your model using a learning rate of 0.01 for 100 epochs, and record both the training loss and the test loss at each epoch. Then briefly discuss how changes in network depth, width, and activation function affect both loss values.

(b) [10 points] Decision Boundary Visualization and Analysis: Discuss any signs of overfitting or underfitting observed in your experiments, and explain which configuration achieved the best performance and why. Additionally, visualize the decision boundary of the optimal configuration by plotting its decision boundary in the input space along with the training and testing data points.