CSCI 567, Spring 2025
Haipeng Luo

Written Assignment #1

Due: Feb 06, 2025, 11:59 pm, PT

Instructions

Total points: 50

Submission: Solutions must be typewritten or neatly handwritten and submitted through gradescope. You can submit multiple times, but only the last submission counts. It is your responsibility to make sure that you submit the right things, and we will *not* consider any regrading requests regarding mistakes in making submissions.

Recall that you have a total of three "late days" for the entire semester, and you can use at most one late day for each written assignment.

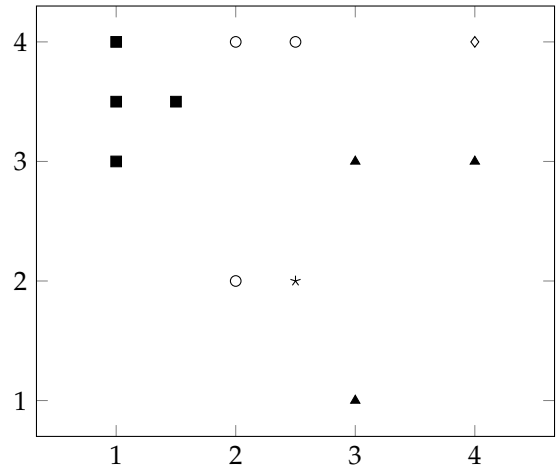
Notes on notation:

- Unless stated otherwise, scalars are denoted by small letter in normal font, vectors are denoted by small letters in bold font, and matrices are denoted by capital letters in bold font.
- $\|.\|$ means L2-norm unless specified otherwise, i.e., $\|.\| = \|.\|_2$.

Academic integrity: Our goal is to maintain an optimal learning environment. You can discuss the written assignments at a high level with others, but you should not look at any other's solutions. Trying to find solutions online or from any other sources (including ChatGPT and other similar tools) is prohibited, will result in zero grade and will be reported. To prevent any future plagiarism, uploading any materials from this course to the Internet is also prohibited, and any violations will be reported. Please be considerate and help us help everyone get the best out of this course.

Problem 1 Nearest Neighbor Classification (10 points)

For the data given below, squares, triangles, and open circles are three different classes of data in the training set and the diamond ($\diamondsuit$) and star (*) are test points with an unknown class. We denote the total number of training points as N (which equals 10) and consider K-nearest-neighbor (KNN) classifier with L2 distance.



1. What is the *diamond* classified as for $K = 1$? Explain briefly. (2 points)

2. What is the smallest odd value of K for KNN to predict triangle for the test point *star*? Explain briefly. (3 points)

3. Suppose one performs leave-one-out validation (that is, N-fold cross validation) to choose the best hyper-parameter K. List all the points that are misclassified during the N runs when testing the hyper-parameter value $K = 1$, and report the averaged error rate for this hyper-parameter. (5 points)

Problem 2 Linear Regression (24 points)

2.1 (10 points) In the class, we discussed L2 regularized least square solution defined as

$$\boldsymbol{w}_* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^D} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{w}\|_2^2 \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{N\times D}$ is the data matrix with each row corresponding to the feature of an example, $\boldsymbol{y} \in \mathbb{R}^N$ is a vector of all the outcomes, $\|\cdot\|_2$ stands for the L2 norm, and λ is the regularization coefficient. In this problem, we consider a different regularization method:

$$\boldsymbol{w}'_* = \arg\min_{\boldsymbol{w}\in\mathbb{R}^D} \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \boldsymbol{w}^T\boldsymbol{M}\boldsymbol{w} \tag{2}$$

where $\boldsymbol{M} \in \mathbb{R}^{D\times D}$ is a positive definite matrix.

1. Show that the new method is a generalization of the standard L2 regularization by picking a matrix $\boldsymbol{M}$ such that $\boldsymbol{w}'_*$ in Eq. (2) equals $\boldsymbol{w}_*$ in Eq. (1). (2 points)

2. Find the closed form of $\boldsymbol{w}'_*$ by writing down the gradient of $F(\boldsymbol{w}) = \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \boldsymbol{w}^T\boldsymbol{M}\boldsymbol{w}$ and setting it to $\mathbf{0}$. (4 points)

3. Recall the Newton method: $\boldsymbol{w}^{(t+1)} \leftarrow \boldsymbol{w}^{(t)} - \boldsymbol{H}_t^{-1}\nabla F(\boldsymbol{w}^{(t)})$ where $H_t = \nabla^2 F(\boldsymbol{w}^{(t)})$. Show that no matter what the initialization $\boldsymbol{w}^{(0)}$ is, Newton method always takes one step only to find the minimizer $\boldsymbol{w}'_*$ of $F(\boldsymbol{w}) = \|\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}\|_2^2 + \boldsymbol{w}^T\boldsymbol{M}\boldsymbol{w}$. (4 points)

2.2 (14 points) Assume we have a training set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \in \mathbb{R}^D \times \mathbb{R}$, where each outcome y_n is generated by a probabilistic model $\boldsymbol{w}_*^{\mathrm{T}} \boldsymbol{x}_n + \epsilon_n$ with ϵ_n being an independent Gaussian noise with zero-mean and variance σ^2 for some $\sigma > 0$. In other words, the probability of seeing any outcome $y \in \mathbb{R}$ given $\boldsymbol{x}_n$ is

$$\Pr(y \mid \boldsymbol{x}_n; \boldsymbol{w}_*, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(y - \boldsymbol{w}_*^{\mathrm{T}} \boldsymbol{x}_n)^2}{2\sigma^2}\right).$$

1. Assume σ is fixed and given, find the maximum likelihood estimation for $\boldsymbol{w}_*$. In other words, first write down the probability of seeing the outcomes $y_1, \ldots, y_N$ given $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ as a function of the value of $\boldsymbol{w}_*$; then find the value of $\boldsymbol{w}_*$ that maximizes this probability. You can assume $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$ is invertible where $\boldsymbol{X}$ is the data matrix as used in Problem 2.1. (6 points)

2. Now consider σ as a parameter of the probabilistic model too, that is, the model is specified by both $\boldsymbol{w}_*$ and σ. Find the maximum likelihood estimation for $\boldsymbol{w}_*$ and σ. (8 points)

Problem 3 Linear Classifiers (16 points)

In Lecture 3 we have seen the hinge loss $\ell(z) = \max\{0, 1 - z\}$, which is non-differentiable at $z = 1$. To avoid this issue, we can consider the square of hinge loss $\ell(z)^2$, which is differentiable everywhere. More specifically, given a binary dataset $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \in \mathbb{R}^D \times \{-1, 1\}$, we define the following new loss function for a linear model $\boldsymbol{w} \in \mathbb{R}^D$:

$$F(\boldsymbol{w}) = \frac{1}{N} \sum_{n=1}^{N} F_n(\boldsymbol{w}), \quad \text{where } F_n(\boldsymbol{w}) = \left(\max\left\{0, 1 - y_n \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_n\right\}\right)^2. \tag{3}$$

1. For a fixed n, write down the gradient $\nabla F_n(\boldsymbol{w})$ (show your derivation), then fill in the missing details in the repeat-loop of the algorithm below which applies SGD to minimize F. (6 points)

Algorithm 1: SGD for minimizing Eq. (3)

Input: A training set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \in \mathbb{R}^D \times \{-1, 1\}$, learning rate $\eta > 0$
Initialization: $\boldsymbol{w} = \boldsymbol{0}$
Repeat:

2. Next, consider modifying $F_n(\boldsymbol{w})$ as

$$F_n(\boldsymbol{w}) = \begin{cases} \left(\max\left\{0, 1 - \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_n\right\}\right)^2, & \text{if } y_n = 1, \\ 0.1 \left(\max\left\{0, 1 + \boldsymbol{w}^{\mathrm{T}} \boldsymbol{x}_n\right\}\right)^2, & \text{else.} \end{cases} \tag{4}$$

   (a) Consider a binary classification dataset of points in two dimensions as shown in Figure 1, where the red, plus signs denote samples with label $+1$, and the green, minus signs denote samples with label -1. When training a linear classifier with the modified loss in Eq. (4), which of $\boldsymbol{w}_1$ or $\boldsymbol{w}_2$ in Figure 1 do you think is more likely the resulting decision boundary? Explain briefly. (2 points)

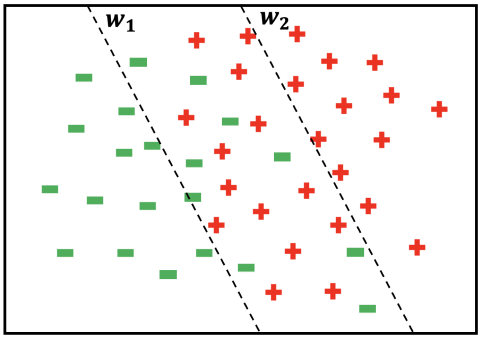


Figure 1: A binary classification task

(b) Based on your answer from the last question, give an example where one would want to modify the loss function in such a way. (2 points)

(c) Similarly to Question 3.1, write down the gradient of this modified loss F_n, then fill in the missing details in the repeat-loop of the algorithm below which applies SGD to minimize F. (6 points)

Algorithm 2: SGD for minimizing modified loss Eq. (4)

Input: A training set $(x_1, y_1), \ldots, (x_N, y_N) \in \mathbb{R}^D \times \{-1, 1\}$, learning rate $\eta > 0$
Initialization: $w = \mathbf{0}$
Repeat: