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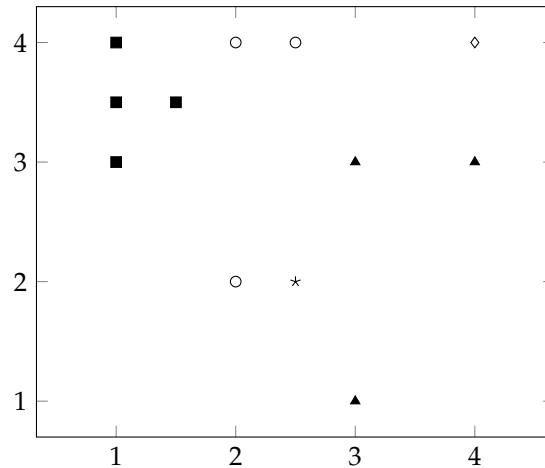
(a) What is the *diamond* classified as for $K = 1$? Explain briefly.   **(2 points)**

Triangle.   **(1 point)**
The closest point to the diamond is clearly the triangle located at $(4, 3)$.   **(1 point)**

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

The smallest value is $K = 3$.   **(1 point)**
The closest 3 points to the star are: the circle at $(2, 2)$, the triangle at $(3, 1)$, and another triangle at $(3, 3)$. Therefore, if $K = 1$, the prediction would have been circle, while if $K = 3$, the prediction would be triangle. **(2 points)**

(c) 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)**

The points that are misclassified during the N runs are the circle at $(2, 2)$ and the triangle at $(3, 1)$, and thus the error rate is $2/10 = 0.2$.
Rubrics:

- List the two points correctly.   **(2 points)**

- No listing of other points.   **(2 points)**

- Report the error rate correctly as the number of listed points (which might not be the correct answer 2) divided by 10.   **(1 points)**

Problem 2   Linear Regression   (24 points)

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

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

where $X \in \mathbb{R}^{N \times D}$ is the data matrix with each row corresponding to the feature of an example, $y \in \mathbb{R}^N$ 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:

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

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

1. Show that the new method is a generalization of the standard L2 regularization by picking a matrix M such that w_*' in Eq. (2) equals w_* in Eq. (1).   (2 points)

   $M = \lambda I$ where I is the D by D identify matrix clearly makes $w^T M w$ equal to $\lambda \|w\|_2^2$.   (2 points)

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

   The gradient is $2X^T(Xw - y) + 2Mw$.   (2 points)
   Setting it to $\mathbf{0}$ and using the fact that M is invertible gives   (2 points)

   $$w_*' = \left(X^T X + M\right)^{-1} X^T y.$$

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

   The Hessian of F at any point is always $2(X^T X + M)$.   (2 points)
   Therefore, applying Newton method to any initial point $w^{(0)}$ gives   (2 points)

   $$w^{(1)} = w^{(0)} - (X^T X + M)^{-1}(X^T(Xw^{(0)} - y) + Mw^{(0)})$$
   $$= w^{(0)} - (X^T X + M)^{-1} \left((X^T X + M)w^{(0)} - X^T y\right)$$
   $$= \left(X^T X + M\right)^{-1} X^T y = w_*'.$$

2.2 (14 points) Assume we have a training set $(x_1, y_1), \ldots, (x_N, y_N) \in \mathbb{R}^D \times \mathbb{R}$, where each outcome y_n is generated by a probabilistic model $w_*^T x_n + \epsilon_n$ 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 x_n is

$$\Pr(y \mid x_n; w_*, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(y - w_*^T x_n)^2}{2\sigma^2}\right).$$

1. Assume σ is fixed and given, find the maximum likelihood estimation for w_*. In other words, first write down the probability of seeing the outcomes $y_1, \ldots, y_N$ given $x_1, \ldots, x_N$ as a function of the value of w_*; then find the value of w_* that maximizes this probability. You can assume $X^T X$ is invertible where X is the data matrix as used in Problem 2.1. (6 points)

The probability of seeing the outcomes $y_1, \ldots, y_N$ given $x_1, \ldots, x_N$ for a linear model w is

$$\mathcal{P}(w) = \prod_{n=1}^{N} \Pr(y_n \mid x_n; w, \sigma) = \prod_{n=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(y_n - w^T x_n)^2}{2\sigma^2}\right).$$

Taking the negative log, this becomes

$$F(w) = N \ln\sqrt{2\pi} + N \ln\sigma + \frac{1}{2\sigma^2} \sum_{n=1}^{N}(y_n - w^T x_n)^2 = N \ln\sqrt{2\pi} + N \ln\sigma + \frac{1}{2\sigma^2}\|Xw - y\|_2^2.$$

Maximizing $\mathcal{P}$ is the same as minimizing F, which is clearly the same as just minimizing $\sum_{n=1}^{N}(y_n - w^T x_n)^2$, the same objective as for least square regression. Therefore the MLE for w_* is exactly the same as the least square solution:

$$w_* = (X^T X)^{-1} X^T y.$$

Rubrics:

- Write down the correct likelihood function. (2 points)
- Any derivation revealing that this is the same as least square regression. (2 points)
- Arrive at the correct final answer. (2 points)

---

2. Now consider σ as a parameter of the probabilistic model too, that is, the model is specified by both w_* and σ. Find the maximum likelihood estimation for w_* and σ. (8 points)

From previous calculation, the MLE for w_* and σ is the minimizer of the function

$$F(w, \sigma) = N \ln \sqrt{2\pi} + N \ln \sigma + \frac{1}{2\sigma^2} \|Xw - y\|_2^2.$$

We first fix σ and minimize over w, which leads to the same MLE for $w*$:

$$w_* = (X^\mathsf{T}X)^{-1}X^\mathsf{T}y.$$

Next we minimize $F(w_*, \sigma)$ as function of σ by setting the derivative w.r.t. σ to be 0:

$$\frac{\partial F(w_*, \sigma)}{\partial \sigma} = \frac{N}{\sigma} - \frac{1}{\sigma^3}\|Xw_* - y\|_2^2 = 0.$$

Solving for σ gives the MLE estimate:

$$\sigma = \frac{1}{\sqrt{N}}\|Xw_* - y\|_2 = \frac{1}{\sqrt{N}}\|X(X^\mathsf{T}X)^{-1}X^\mathsf{T}y - y\|_2.$$

Rubrics:

- Write down the correct likelihood as a function of both w and σ. (2 points)
- Arrive at the correct solution for w_*. (2 points)
- Correct derivative with respect to σ. (2 points)
- Arrive at the correct solution for σ. (2 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 $(x_1, y_1), \ldots, (x_N, y_N) \in \mathbb{R}^D \times \{-1, 1\}$, we define the following new loss function for a linear model $w \in \mathbb{R}^D$:

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

1. For a fixed n, write down the gradient $\nabla F_n(w)$ (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)

---

1 **Input:** A training set $(x_1, y_1), \ldots, (x_N, y_N) \in \mathbb{R}^D \times \{-1, 1\}$, learning rate $\eta > 0$
2 **Initialization:** $w = 0$
3 **Repeat:**
4 $\quad$ randomly pick an example (x_n, y_n)
5 $\quad$ update $w \leftarrow w + 2\eta y_n \max\{0, 1 - y_n w^\mathsf{T} x_n\} x_n$

When $1 - y_n w^\mathsf{T} x_n < 0$, the function is a constant and thus the gradient is $\mathbf{0}$. On the other hand, when $1 - y_n w^\mathsf{T} x_n > 0$, the function is simply $F_n(w) = (1 - y_n w^\mathsf{T} x_n)^2$, and thus by chain rule we have $\nabla F_n(w) = -2y_n(1 - y_n w^\mathsf{T} x_n)x_n$, which is also $\mathbf{0}$ when $y_n w^\mathsf{T} x_n$ approaches 1. Therefore, we have

$$\nabla F_n(w) = -2y_n \max\{0, 1 - y_n w^\mathsf{T} x_n\} x_n.$$

Rubrics:

- 4 points for the derivation of the gradient. Technically we need some discussion for the case $y_n w^\mathsf{T} x_n = 1$ as the solution above shows, but for simplicity it is okay if such discussion is missing as long as the final gradient is correct.

- 2 points for the SGD implementation. Do not deduct more points for wrong gradient from the earlier wrong derivation.

2. Next, consider modifying $F_n(w)$ as

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

(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 w_1 or w_2 in Figure 1 do you think is more likely the resulting decision boundary? Explain briefly.   (2 points)
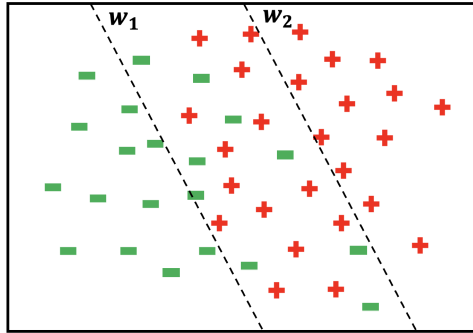

Figure 1: A binary classification task

w_1 is more likely. First note that the dataset is not linearly separable so any linear classifier will make mistakes. Next, due to the asymmetry in weighting, the new loss function does not penalize misclassifying negative points as heavily as it penalizes misclassifying positive points. Therefore, w_1 is preferred over w_2.

(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)

For example, in fraud detection ($+1$ represents a fraud), it is far more important to make sure that an actual fraud is not missed, and thus the loss function should give more weights to positive labels.

Rubrics: Any reasonable example where the one label represents a critical or high-stakes outcome and is far more important than the other label works.

(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)

1 **Input:** A training set $(x_1, y_1), \ldots, (x_N, y_N) \in \mathbb{R}^D \times \{-1, 1\}$, learning rate $\eta > 0$
2 **Initialization:** $w = 0$
3 **Repeat:**
4   randomly pick an example (x_n, y_n)
5   update $w \leftarrow w - \begin{cases} -2\eta \max\{0, 1 - w^\mathsf{T} x_n\} x_n, & \text{if } y_n = 1, \\ 0.2\eta \max\{0, 1 + w^\mathsf{T} x_n\} x_n & \text{else.} \end{cases}$

Based on previous calculation, the gradient of the modified loss is

$$\nabla F_n(w) = \begin{cases} -2 \max\{0, 1 - w^\mathsf{T} x_n\} x_n, & \text{if } y_n = 1, \\ 0.2 \max\{0, 1 + w^\mathsf{T} x_n\} x_n & \text{else.} \end{cases}$$

Rubrics:
- 4 points for the correct gradient.
- 2 points for the SGD implementation. Do not deduct more points for wrong gradient from the earlier wrong derivation.