CIS5200: Machine Learning Spring 2025

Homework 1

Release Date: January 30, 2025 *Due Date: February 13, 2025*

- HW1 will count for 10% of the grade. This grade will be split between the written (30 points) and programming (30 points) part.
- All written homework solutions are required to be formatted using LaTeX. Please use the template here. Do not modify the template. This is a good resource to get yourself started, if you are still not comfortable with using LaTeX.
- You will submit your solution for the written part of HW1 as a single PDF file via Gradescope. The deadline is **11:59 PM ET**. Contact TAs on Ed if you face any issues uploading your homeworks.
- Collaboration is permitted and encouraged for this homework, though each student must understand, write, and hand in their own submission. In particular, it is acceptable for students to discuss problems with each other; it is not acceptable for students to look at another student's written answers when writing their own. It is also not acceptable to publicly post your (partial) solution on Ed, but you are encouraged to ask public questions on Ed. If you choose to collaborate, you must indicate on each homework with whom you collaborated.

Please refer to the notes and slides posted on the website if you need to recall the material discussed in the lectures.

1 Written Questions (30 points)

Problem 1: Margin Perceptron (14 points)

In this problem, we will consider a variant of the Perceptron algorithm. As in the lecture, we will assume that $\|x_i\|_2 \leq 1$ for all $i \in \{1, \ldots, m\}$ and the data is linearly separated by w_* with $\|w_*\|_2^2 = 1$ and margin $\gamma = \min_{i \in \{1,\ldots,m\}} |w_*^\top x_i|$.

Algorithm 1: Margin Perceptron

```
Initialize $w_1 = 0 \in \mathbb{R}^d$
for $t = 1, 2, \ldots$ do
    if $\exists i \in \{1, \ldots, m\}$ s.t. $y_i \neq \text{sign}\left(w_t^\top x_i\right)$ or $|w_t^\top x_i| \leq 1$ then update $w_{t+1} = w_t + y_i x_i$
    else output $w_t$
end
```

We will show that Margin Perceptron stops after $3/\gamma^2$ steps and returns a hyperplane w such that

$$\min_{i \in \{1,\ldots,m\}} \frac{|w^\top x_i|}{\|w\|_2} \geq \gamma/3.$$

Note that the margin is the distance of the closest point to the hyperplane, and since $\|w\|_2$ is not necessarily norm 1, this quantity is given by $\min_{i\in\{1,\ldots,m\}} \frac{|w^\top x_i|}{\|w\|_2}$.

1.1 (2 points) Prove the growth lemma, that is, show that after every round t, we have

$$w_*^\top w_{t+1} \geq w_*^\top w_t + \gamma.$$

1.2 (3 points) Prove the control lemma, that is, show that after every round t, we have

$$\|w_{t+1}\|_2^2 \leq \|w_t\|_2^2 + 3.$$

1.3 (3 points) Using the above two parts, show that after T rounds,

$$\gamma T \leq \|w_{T+1}\|_2 \leq \sqrt{3T}.$$

1.4 (1 point) Use 1.3, to conclude that $T \leq 3/\gamma^2$.

1.5 (4 points) Show that the output hyperplane w satisfies

$$\min_i \frac{|w^\top x_i|}{\|w\|_2} \geq \frac{\gamma}{3}.$$

Hint: You will need to use the results in 1.2 and 1.3 plus the stopping condition of the algorithm.

1.6 (1 point) Why is it desirable to learn a predictor that has large margin?

Problem 2: Bayes Optimal Classifier and Squared Loss (10 points)

Let $\eta(x) = \Pr[y = 1|x]$ be the conditional probability of label 1 given input $x \in \mathbb{R}^d$. Consider the squared loss:

$$\ell(h(x), y) = (h(x) - y)^2$$

where now $h(x) \in \mathbb{R}$ can output any real number (not just $\{-1, 1\}$).

2.1 (3 points) Show that for any fixed x, the function $h(x)$ that minimizes the expected squared loss $\mathbb{E}_{y|x}[(h(x) - y)^2]$ is:

$$h^*(x) = \mathop{\mathbb{E}}_{y|x}[y] = 2\eta(x) - 1$$

2.2 (4 points) Now consider a simple generative model where:

$$P(y=1) = \frac{1}{2}$$
$$P(x|y=1) = \mathcal{N}(\mu, I)$$
$$P(x|y=-1) = \mathcal{N}(-\mu, I)$$

where I is the identity matrix. Show that $\eta(x)$ takes the form:

$$\eta(x) = \frac{1}{1+\exp(-2\mu^\top x)}$$

2.3 (3 points) Show that thresholding $h^*(x)$ at 0 for the above model gives a linear decision boundary $w^\top x + b = 0$ for some $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$. Find w and b.

Fact: This decision boundary is the same you would have gotten from the Bayes optimal 0/1 loss classifier.

Problem 3: k-NN Analysis (6 points)

Consider a binary classification problem where we observe data $\{(x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$. Let $N_x \subseteq \{(x_i, y_i)\}_{i=1}^n$ denote the set of k nearest neighbors of a point x, where for any $z \notin N_x$ and $z' \in N_x$, we have $\text{dist}(x, z) \geq \text{dist}(x, z')$. Let $f_k(x)$ be the k-NN classifier that predicts the majority label among the labels in N_x. We use the Euclidean (Minkowski with $p = 2$) distance, where for any two points $x, z \in \mathbb{R}^d$, $\text{dist}(x, z) = \|x - z\|_2 = (\sum_{i=1}^d |[x]_i - [z]_i|^2)^{1/2}$.

3.1 (3 points) Consider two test points x and x' that differ only in one coordinate by ϵ. Show that the difference in their distances to any training point z is at most ϵ. What does this tell us about the robustness of k-NN predictions to small perturbations?

3.2 (3 points) Consider 1-NN classification ($k = 1$) and let Δ be the difference in distances between the nearest and second-nearest training points to a test point $x \in \mathbb{R}^d$. Using the result from 3.1, show that if we perturb each coordinate of x by at most $\epsilon = \frac{\Delta}{2d}$, the nearest training point (and thus the prediction) remains unchanged.

Hint: Use the result from 3.1.

Programming Questions (30 points)

Use the link here to access the Google Colaboratory (Colab) file for this homework. Be sure to make a copy by going to "File", and "Save a copy in Drive". This assignment uses the PennGrader system for students to receive immediate feedback. As noted on the notebook, please be sure to change the student ID from the default '99999999' to your 8-digit PennID.

Instructions for how to submit the programming component of HW 1 to Gradescope are included in the Colab notebook. You may find this PyTorch reference to be helpful - if you get stuck, it may be helpful to review some of the PyTorch documentation and functions.