CIS 4190/5190: Applied Machine Learning — Spring 2025

Homework 2

Handed Out: Feb 12 — *Due: Mar 3, 8 p.m.*

- You are encouraged to format your solutions using LaTeX. Handwritten solutions are permitted, but remember that you bear the risk that we may not be able to read your work and grade it properly — we will not accept post hoc explanations for illegible work. You will submit your solution manuscript for written HW 2 as a single PDF file.
- The homework is **due at 8 PM** on the due date. We will be using Gradescope for collecting the homework assignments. Please submit your solution manuscript as a PDF file via Gradescope. Post on Ed Discussion and contact the TAs if you are having technical difficulties in submitting the assignment.
- Make sure to assign pages to each question when submitting homework to Gradescope. The TA may deduct 0.2 points per sub-question if a page is not assigned to a question.

1 Written Questions

Note: You do not need to show work for multiple choice questions. If formatting your answer in LaTeX, use our LaTeX template `hw2_template.tex` (This is a read-only link. You'll need to make a copy before you can edit. Make sure you make only private copies.).

1. **[Logistic Regression] (8 pts)**

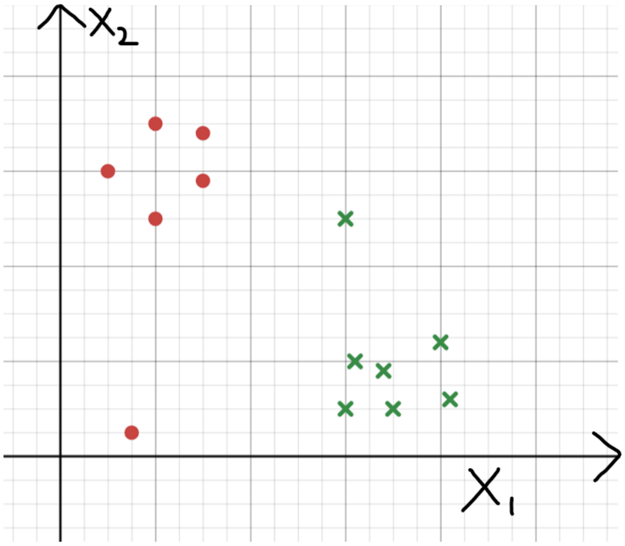

Figure 1: Data for Logistic Regression Question

Let the data distribution, as shown in Figure1, represent the binary classification problem where we fit the model $p(y = 1|x, \theta) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$. As seen in class, we do this by minimizing the negative log loss (same as maximizing the likelihood), as shown below:

$$L(\theta) = -\ell(\theta, D_{\text{train}})$$

where $\ell(\theta, D_{\text{train}})$ represents the log likelihood on the training set.

For the questions below, submit the answer to each question as a separate figure. We just expect an approximation of the figures if you submit hand-drawn solutions, also be careful about the clarity of your submitted figures.

(a) Show a decision boundary that possibly would correspond to $\hat{w}$(final weights) after training the regressor. How many datapoints are wrongly classified on the training data?

(b) For this part, consider that a strong regularization is applied to the θ_0 parameter and we minimize

$$L_0(\theta) = -\ell(\theta, D_{\text{train}}) + \lambda\theta_0^2$$

Since we apply a strong regularization, assume that λ is a very large, so, θ_0 is pulled down all the way to 0, but all other parameters are unregularized. Show a decision boundary that possibly would correspond to $\hat{w}$. How many datapoints are wrongly classified on the training data?[Hint: consider the behavior of simple linear regression $\theta_0 + \theta_1 x_1 + \theta_2 x_2$ when $x_1 = x_2 = 0$]

(c) Now, heavy regularization is performed only on the θ_1 parameter, i.e., we minimize

$$L_1(\theta) = -\ell(\theta, D_{\text{train}}) + \lambda\theta_1^2$$

Show a decision boundary that possibly would correspond to $\hat{w}$. How many datapoints are wrongly classified on the training data?

(d) Finally, heavy regularization is done only on the θ_2 parameter. Show a decision boundary that possibly would correspond to $\hat{\theta}$. How many datapoints are wrongly classified on the training data?

2. [k **Nearest Neighbors] (8 pts)**

Consider properties of k-NN models:

a. (2 pts) Suppose that we are using k-NN with just two training points, which have different (binary) labels. Assuming we are using $k = 1$ and Euclidean distance, what is the decision boundary? Include a drawing with a brief explanation.

b. (2 pts) For binary classification, given infinite data points, can k-NN with $k = 1$ express any decision boundary? If so, describe how the dataset should be structured so that k=NN with k=1 correctly follows any given decision boundary. If no, give an example of a decision boundary that cannot be achieved.

c. (2 pts) Suppose we take $k \to \infty$; what type of function does the resulting model family become?

d. (2 pts) What effect does increasing the number of nearest neighbors k have on the bias-variance tradeoff? Explain your answer. [Hint: Use parts (b) and (c) in your explanation.]

3. **[Decision Trees] (10 pts)**

In class, we discussed early stopping of generating splits and post-pruning. Here, we consider how they interact.

a. (4 pts) Naïvely, one might expect that if we are using post-pruning on a validation set, then there is never any benefit to using early stopping conditions (e.g., maximum depth to split). Explain why this is not the case.

b. (4 pts) Suppose we are training a decision tree with both early-stopping conditions and post-pruning. For each of the following, indicate whether it increases bias or variance: (i) increase the maximum depth of the decision tree, (ii) increase the minimum number of samples needed to split, (iii) disable post-pruning, and (iv) assuming we are using a feature map, add more features to the feature map.

c. (2 pts) For the given boolean functions, please provide the corresponding decision tree representations.

(a) $A \wedge (B \vee C)$

(b) $(A \wedge B) \vee (C \wedge D)$

4. **[Decision Trees] (10 pts)** Consider the following set of training examples for a decision tree classifier: [Hint: a1, a2 are attributes, e.g. High Income? Y/N]

Instance	a1	a2	Classification
1	-	+	F
2	-	-	T
3	+	+	T
4	+	+	T

Recall the following definitions of entropy and information gain, respectively, which are useful for this problem:

$$H(Z) = -\sum_y P(Y = y) \log_2 P(Y = y)$$

$$\mathrm{IG}(Z, j, t) = H(Z) - H(Z[x_j = t])P(x_j = t) - H(Z[x_j \neq t])P(x_j \neq t).$$

a. (1 pts) Given this decision tree, what would be the classification for a new instance with:a1 = +, a2 = -?

b. (2 pts) What is the entropy of this collection of training examples?

c. (4 pts) What is the information gain of the two attributes respectively relative to these training examples?

d. (3 pts) Draw the complete (unpruned) decision tree, showing the class predictions at the leaves. Assuming you are using LaTeX, you may (i) very neatly hand draw the tree, photograph it, and include it as a figure, (ii) draw it using a graphics program or PowerPoint, or (iii) express the tree in a series of if statements, preferably using LaTeX's verbatim environment.

5. [k **Nearest Neighbors**] **(8pts)** Imagine you want to apply k Nearest Neighbors to the binary classification problem of predicting whether a house is worth more than $600, 000 (y = +1) or less than $600, 000 (y = -1). Suppose you have two features: the square footage of the house, and the distance of the house from the nearest school (in miles). Most houses in the dataset are located in residential neighborhoods with schools nearby.

a. (1 pts) In terms of the features above, what is the modeling assumption that k Nearest Neighbors makes about house prices?

b. (2 pts) Given that your two features are square footage and the distance of the house from the nearest school (in miles), why might Euclidean distance be a bad metric of similarity / dissimilarity between houses? What might you do to solve this problem?

c. (1 pts) Suppose you found another potential feature: the type of roof the house has, expressed as a categorical feature via integer IDs (i.e. 0 = shingles, 1 = padded, etc.). Is it a good idea to use this feature alongside the other two features, with Euclidean distance? Explain why or why not.

d. (2 pts) Suppose you have n training examples $x_1, ..., x_n$, each with d features. What is the naive running time complexity of 1 Nearest Neighbors in terms of n and d? (Hint: express time complexity in terms of n and d using Big-O notation)

e. (2pts) Briefly describe the curse of dimensionality. Why can k-NN still perform well on some datasets like the handwritten digits, despite these images being very high dimensional?

6. [**Neural Networks**] **(6 pts)**

a. (1 pts) **True or False**: For any neural network, the validation loss will always decrease monotonically with the number of iterations of gradient descent, provided the step size is sufficiently small.

- True
- False

b. (1 pts) **True or False**: For multi-class classification using neural network, the final softmax layer is used to transform the raw values calculated by the network into probabilities of each class.

- True
- False

c. (1 pts) We do not initialize weights to all zeros in Neural Networks because:

- A: Not true; we can initialize weights to zero
- B: Zero-initialized neural networks produce numerical errors due to finite precision representations
- C: Units within each layer after running gradient descent remain the same

d. (1 pts) For a feed forward neural network, the intuition of using a non-linear activation function is:

- A: To make the network more computationally efficient during training
- B: To make the network more computationally efficient during inferencing
- C: To enable the network to model complex, non-linear relationships in the data
- D: To reduce overfitting by limiting the capacity of the network

e. (1 pts) Consider a fully connected neural network with the following architecture:

Input layer: 100 neurons

Hidden layer 1: 50 neurons

Hidden layer 2: 30 neurons

Output layer: 10 neurons

Assume there are no biases in the network. Calculate the total number of trainable parameters in this network.

- A: 6300
- B: 7300
- C: 6800
- D: 7800

f. (1 pts) Consider a fully connected neural network with the following architecture:

Input layer: 100 neurons

Hidden layer 1: 40 neurons

Hidden layer 2: 30 neurons

Output layer: 10 neurons

Assume biases are included in the network. Calculate the total number of trainable parameters in this network.

- A: 5370
- B: 5580
- C: 6890
- D: 4890

2 Python Programming Questions

A IPython notebook is linked on the class website. It will tell you everything you need to do, and provide starter code. Remember to include the plots and answer the questions in your written homework submission!