CSCI 145 Problem Set 4

September 17, 2025

Submission Instructions

Please upload *your* work by **11:59pm Monday September 22, 2025.**

- You are encouraged to discuss ideas and work with your classmates. However, you **must acknowledge** your collaborators at the top of each solution on which you collaborated with others and you **must write** your solutions independently.
- Your solutions to theory questions must be written legibly, or typeset in LaTeX or markdown. If you would like to use LaTeX, you can import the source of this document (available from the course webpage) to Overleaf.
- I recommend that you write your solutions to coding questions in a Jupyter notebook using Google Colab.
- You should submit your solutions as a **single PDF** via the assignment on Gradescope.

Grading: The point of the problem set is for *you* to learn. To this end, I hope to disincentivize the use of LLMs by **not** grading your work for correctness. Instead, you will grade your own work by comparing it to my solutions. This self-grade is due the Friday *after* the problem set is due, also on Gradescope.

Problem 1: Spam or Ham

In this problem, we will use Naive Bayes to identify emails as spam or ham (not spam). Consider the *bag-of-words* matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$. Each row corresponds to one of n emails, and each column corresponds to one of d words. For email index $i \in \{1, \ldots, n\}$ and word index $j \in \{1, \ldots, d\}$, the corresponding entry in $\mathbf{X}$ is given by

$$[\mathbf{X}]_{i,j} = \begin{cases} 1 & \text{word } j \text{ appears in email } i \\ 0 & \text{else.} \end{cases} \tag{1}$$

You can find code to load a bag-of-words matrix here.

Part A: Computing Likelihood

Split the bag-of-words matrix $\mathbf{X}$ and the labels $\mathbf{y} \in \{0,1\}^n$ into training and testing sets, using an $80 - 20$ random split.

Compute the likelihoods for the *training data* $\mathbf{p}^{(1)} \in [0,1]^d$ and $\mathbf{p}^{(0)} \in [0,1]^d$ as described in class, *without* using a for loop.

Part B: Computing Log Posteriors

Compute the log of the posteriors for the *testing data* using matrix multiplication between $\mathbf{X}$, $\log(\mathbf{p}^{(1)})$, and $\log(\mathbf{p}^{(0)})$.

Why are we using the *log* likelihoods? What happens to the evidence?

Part C: Accuracy

Use the log posteriors to make predictions for the test set. What is the accuracy?

Problem 2: Logistic Regression and ROC Curve

In this problem, you will gain familiarity working with the Python library `torch`, which is incredibly useful for building and training neural network models.

Part A: Data and 3-Step Recipe Initialization

Load a *binary classification* dataset of your choice (I used the breast cancer dataset from `sklearn`). After reviewing the `torch` quickstart guide, please:

- ☐ Create training and testing dataloaders
- ☐ Initialize a one layer linear model
- ☐ Initialize the binary cross entropy loss function with logits
- ☐ Initialize the Adam optimizer

Hint: ChatGPT is your friend for documentation and examples!

Part B: `torch` Training

Using your set up, train your model for 100 epochs. As you saw in the quickstart quide, training consists of iterating over each *batch* in your training data, and, for each batch, 1) a forward pass where you compute the loss, and 2) a backward pass where you use the optimizer to update your model.

The commands are a little weird at first, but it's worth understanding them well because we'll use them again and again.

Part C: ROC Curve

Pass the test data through the trained model. The outputs are *logits*, which you can convert to probabilities by applying the sigmoid function (like almost everything else, this function is built in to `torch`). What is the accuracy of your model on the test data?

Consider 1000 evenly spaced thresholds between the minimum probability and the maximum probability. For each threshold τ, compute the TPR and FPR if we made predictions based on which probability was above τ. Plot the ROC curve, where the vertical access is TPR and the horizontal access is FPR. What is the area under the curve (AUC)?

Hint: You can compute the AUC using basic calculus i.e., the area of each "rectangle" under the curve.

Part D: ROC Curve Changes?

Suppose you forgot to apply the sigmoid function to your logits. How would the ROC curve and the AUC change?

The *reason* neither changes is because the sigmoid function is a monotonically increasing function: For a particular threshold τ in the logit space, we will recover the same TPR and FPR at $\sigma(\tau)$ in the probability space.