MIT 6.C01/6.C51 Modeling with Machine Learning
Spring 2022
Problem Set 2

Release date. **Friday, February 18 2022, 5:00 PM ET**
Due date. **Friday, March 11 2022, 5:00 PM ET**

Instructions

- This problem set contains four questions, each containing sub-questions.
- Each of these four questions has one sub-question marked as **Graduate version.** This question is mandatory for those who have registered for the graduate version of this class (`6.C51`). It is optional for others. All questions are weighted equally.
- Submissions should be made via Gradescope. You are required to prepare one PDF containing all your answers–both your written answers and code snippets. Please answer each question in a separate page, since Gradescope will require you to select pages pertaining to each sub-question.
- You have been provided `hw2.py`, a Python file containing starter code for Problem 1. Use it to write and test your implementations of the different functions we ask you to provide code for. When submitting your solutions, paste code that the question asks you into the final PDF you will submit.
- Please adhere to the collaboration policy described on our course page on Canvas. Clearly mention the names of your collaborators on the top of your first page in your submission.

Problem 1: Sentiment Analysis (Written & Code)

In this problem, we will consider sentiment analysis as discussed in class. You have been provided two data files consisting of restaurant reviews. The first file, `positive.txt`, includes a list of positive reviews. The second file, `negative.txt`, consists of negative reviews. The data has been processed such that each sentence has been tokenized and lower-cased. We have also provided you with skeleton code in `hw2.py`.

1. Write a python script that builds a vocabulary, V, from both the positive and negative reviews. Restrict the vocabulary to words that appear at least $t = 10$ times, with an additional `<oov>` token to represent out-of-vocabulary words. What is the size of the vocabulary $|V|$? Include a copy of your updated `build` function in your response.

2. Choosing the vocabulary count threshold, t, helps control the number of rare words in the vocabulary, and can prevent a model from learning noisy rare artifacts. In this question, we will explore the trade-off between vocabulary size and the threshold. Consider t in the range $1, 2, 3, \ldots, 15$. For each value of t, rebuild the vocabulary and obtain its size, $|V|$. Include a line plot of the relationship between the size of the vocabulary, $|V|$, and t. What do you observe?

3. In a bag-of-words feature vector with respect to the vocabulary, V, words not belonging to V should be treated as `<oov>`. In the previous question, we saw that the threshold, t, allows us to trim rare words from the dictionary. In this problem, we will consider the effect of increasing t on the bag-of-words representation for a sentence. Write a python script to count how many sentences contain at least one `<oov>` word. Include a line plot of the relationship between the number of sentences that contain at least one `<oov>` word and t. Describe the observed relationship, specifically, does increasing t lead to an increase/decrease in the number of sentences that contain at least one `<oov>` word? Why?

   Hint: Consider using a python dictionary to store your vocabulary.

4. Given the following vocabulary: { *"a", "and", "the", "was", "atmosphere", "scene", "restaurant", "food", "drink", "bad", "game", "service", "not", "dull", "tasteless", "good", "<oov>"*}, what is the bag-of-words representation for the following sentence:

   the food was not bad and the service was relatively good.

   Can you give a version of the sentence above that has negative sentiment such that the new sentence has the same bag-of-words representation as the original sentence? Note: `<oov>` is the out-of-vocabulary token.

5. Describe an approach to extend the vocabulary so that we can now have a bag-of-words representation that will be able to distinguish between the sentence in part (4) and the sentence with negative sentiment that you provided. Provide the key tokens

in the extended bag-of-words representation that will allow a classifier to distinguish these two sentences.

6. **Graduate version.**

   (a) We will now explore trade-offs between expanding a bag-of-words vocabulary, to capture more sophisticated representations, and the size of the vocabulary. Rebuild the vocabulary, for both positive and negative reviews combined, to include both unigrams (single token) and bigrams (two adjacent tokens). For a threshold, $t = 10$, what is the size of this new vocabulary? What is a possible downside of the bigram expansion if the entire new vocabulary is to be used as the feature set for a logistic regression model to differentiate sentences with positive versus negative sentiment? Can you think of a way to address the challenge you came up with?

   (b) We will now explore the downsides of token frequency as a selection criteria for inclusion in a vocabulary. For the positive reviews only, what are the top 5 tokens with the highest frequencies? Similarly for the negative reviews only, what are the top 5 tokens with the highest frequencies? Do you expect that these tokens should capture signal regarding the sentiment for both the positive and negative reviews? Include both unigrams and bigrams in your vocabulary.

   If we consider the set of positive reviews as document 1, and the set of negative reviews as document 2, can you describe an approach for ranking tokens for each document so that the top ranked tokens capture signals about each document's sentiment?

Problem 2: Simpson's Paradox (Written)

A medical study compares the two proposed treatments (A and B) for a novel disease with two subtypes (1 and 2). The following table summarizes the number of successful treatments over the total number of cases for each treatment-subtype pair.

	Treatment A	Treatment B
Subtype 1	8/10	23/30
Subtype 2	20/30	6/10

Table 1: Treatment results for diseases subtypes.

1. What are the respective success rates of the two treatments for the subtype 1 disease? What about the subtype 2 disease? Based on this interpretation, which treatment is more effective?

2. What are the respective success rates of the two treatments for the disease as a whole? Does this interpretation support the conclusion from before? How do you explain it?

3. **Graduate version.**

   (a) How many more people with disease subtype 1 should receive treatment A, at the same success rate, in order for the treatment A to become more promising also when the subtypes are combined?

   (b) One method to correct the Simpson's paradox is through the use of an adjustment formula that conditions on the confounding variables. In our case, the adjustment formula for treatment A is as follows:

   $$P(\text{Success}|A) = P(\text{Success}|1, A)P(1) + P(\text{Success}|2, A)P(2)$$

   such that $P(1)$ and $P(2)$ is the probability of seeing subtype 1 and 2, respectively. Using the proposed formula, what are the respective success rates of the two treatments? Why would conditioning on the confounding variables help in our case?

Problem 3: Recommendation System (Written)

Alice and Bob are developing a recommendation system for an online shopping website. The website has N users $a \in \{1, \ldots, N\}$ and M items $b \in \{1, \ldots, M\}$.

1. They first design a matrix-based approach. They have access to the website's purchase log $D = [(a_1, b_1), (a_2, b_2), \ldots, (a_K, b_K)]$, where each (a_i, b_i) indicates user a_i has purchased item b_i before. Assume there are no duplicates in D.

   Bob constructs the observation matrix $Y \in \mathbb{R}^{N \times M}$, where

   $$Y_{ab} = \begin{cases} 1 & \text{if } (a, b) \in D, \\ 0 & \text{if } (a, b) \notin D, \end{cases}$$

   Please design an objective function $J(X, Y)$ which would estimate the underlying matrix X containing the estimates of each user's recommendation of every item using a squared error loss and regularization. Include a bias term in your formulation, which you can assume to be a known constant bias. Find the closed-form solution of X.

2. Alice realizes that the previous approach cannot provide meaningful results. She decides to do matrix factorization. Specifically, Alice wants to find $U \in \mathbb{R}^{N \times d}$ and $V \in \mathbb{R}^{M \times d}$ such that $X \approx UV^T$. Please design an objective function $J(U, V)$ for Alice. Ensure your objective is regularized. How would you determine the value of d?

3. After taking a deep learning class, Alice and Bob want to try neural networks for recommendation. They represent (a, b) as two one-hot vectors x_a and x_b, where $x_a \in \mathbb{R}^N, x_b \in \mathbb{R}^M$. Please design a neural network model with one hidden layer of size d. The model should take x_a, x_b as input and output a probability y. Specify all the parameters in your model and their dimensions, as well as your choices of non-linear activation. (You may also use the matrices U, V obtained in part 2 above). The answer should be written as a function of x_a, x_b.

4. **Graduate version.**

   (a) Table 2 summarizes the ratings matrix of three good friends Alice (A), Bob (B), Claire (C) and their movie ratings of the movies *Office Space* (OS), *The Departed* (TD), and *Spirited Away* (SA).

	OS	**TD**	**SA**
A	3	2	?
B	?	2.5	1
C	1.5	?	?

Table 2: Movie preferences

   Knowing that the three are good friends, you choose to estimate the missing entries (denoted by ?) in the ratings matrix using rank-1 approximation. What is the approximated ratings matrix? For this question, you can ignore regularizing the decomposed rank-1 vectors which result in the approximated matrix.

   (b) The three friends invite their common friend Dirk (D) to watch movies with them, and add another movie *American Beauty* (AB) in their movie collection. However, since this is the first time Dirk is being invited, and since none of them has watched AB, the ratings corresponding to D and AB are unknown (?). You run the ratings estimation algorithm used in 3.4a. What will the approximated ratings matrix be if we consider the new friend and the movie as well? Justify your answer.

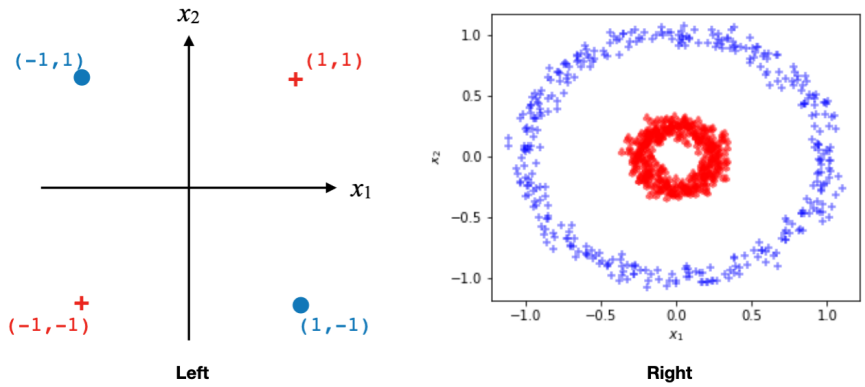
   (c) Having to estimate ratings or the interactions of a new user or item is commonly referred to as the *cold start problem* in collaborative filtering. The following are possible solutions to circumvent this problem. For each solution, describe at least one positive consequence (beyond that it solves the cold start problem) and one negative consequence of that solution choice on the final approximated matrix. Justify your answers mathematically if needed.

      i. For a new user, instead of starting with unknown ratings (?), you initialize the ratings of each item with the average rating of that item across all other users.

ii. Before a user is added to the system, you approximate the ratings matrix X of the existing users by learning low-rank matrices U and V such that $X \approx UV^\intercal$, where $U \in \mathbb{R}^{N \times d}$ and $V \in \mathbb{R}^{M \times d}$. For the new user, you initialize the entries in U by averaging the entries of the other users in U.

Problem 4: Non-linear Prediction (Written)

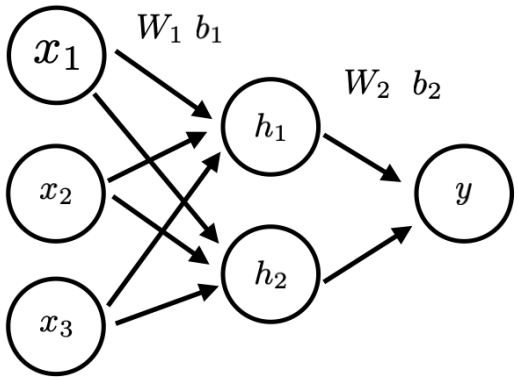
Consider binary classification on the following two datasets (Left and Right), where the blue and red data points belong to two different classes. Both of these tasks cannot be solved by a linear classifier in the (x_1, x_2) feature space as shown.



1. We are interested in transforming these data points, for both datasets, using a feature transformation that would make it so that a linear classifier in the transformed feature space can separate the blue and red classes. For each data, specify a feature transformation on the input dimensions, $\phi(x_1, x_2)$, that will allow each dataset to become linearly separable in the transformed space.

2. In this question, we will hand design a multilayer perceptron (MLP) to solve a slightly modified XOR task. As shown in the 2-layer MLP below, we are given 3 inputs (x_1, x_2, x_3) that are binary valued (i.e. $\{0, 1\}$). The task here is to specify all the parameters, i.e. W_1, b_1, W_2, b_2, so that the MLP only outputs 1 if **exactly two** of the inputs are set to 1, and the other is 0. For the network, we use as non-linearity a threshold indicator function specified as follows:

$$\mathbb{I}(a) = \begin{cases} 1, & \text{if } \ a \geq 0, \\ 0, & \text{if } \ a < 0. \end{cases}$$



The two-layer MLP corresponds to:

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = W_1 \cdot x + b_1$$
$$h = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \mathbb{I}(z)$$
$$y = \mathbb{I}(W_2 \cdot h + b_2)$$

where $W_1 \in \mathbb{R}^{2\times 3}$, $b_1 \in \mathbb{R}^2$, $W_2 \in \mathbb{R}^2$, and $b_2 \in \mathbb{R}$. In this question, we ask that you instantiate (W_1, b_1, W_2, b_2) such that they solve the following problems:

(a) In this part, we will focus on finding W_1 and b_1. In Figure 1, we show a transformation of a series of inputs from the data points in 3D, (x_1, x_2, x_3), to h, (h_1, h_2). In the figures, the red points correspond to inputs where two of the input dimensions are exactly 1, while the blue points do not meet this requirement. Can you find linear separator(s) in the input, 3D, space to separate these classes and what would it look like? What about the linear separator(s) in the (h_1, h_2) space (Right in Figure 1)? What are W_1 and b_1?

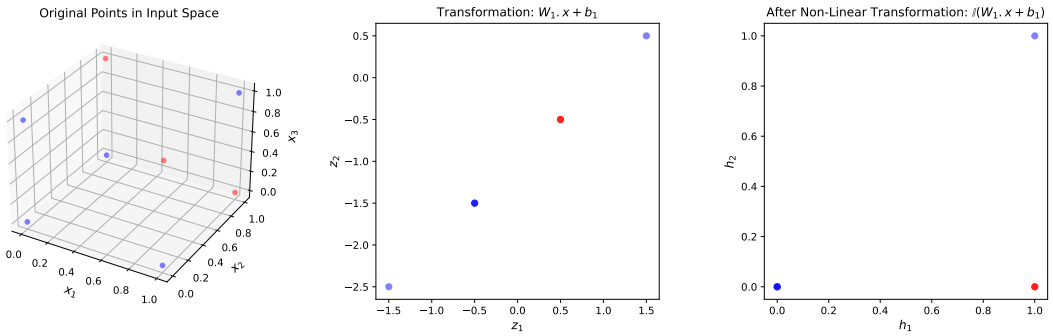


Figure 1: **Left**: Input samples in the (x_1, x_2, x_3) space. **Middle**: After applying the transformation: $W_1 \cdot x + b_1$. Note that we have not applied the indicator non-linearity to these values, and some points on the graph have more than one data point. **Right**: $h : (h_1, h_2)$, which is: $\mathbb{I}(z)$. This corresponds to applying the indicator function to the output of the linear transformation from the middle graph.

(b) In Figure 2, we show the transformation from $h : (h_1, h_2)$ to the output y along with the intermediate representation before the non-linearity is applied. What are W_2, and b_2 that transform h into y?

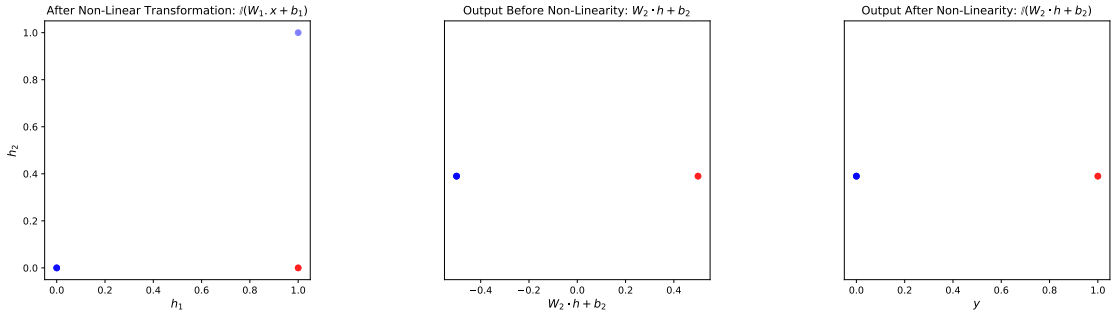


Figure 2: **Left**: $h : (h_1, h_2)$, which is: $\mathbb{I}(W_1 \cdot x + b_1)$. **Middle**: The output of a second linear transformation: $W_2 \cdot h + b_2$. **Right**: The final output of a network across all data points: $\mathbb{I}(W_2 \cdot h + b_2)$.

(c) Given n input-label pairs, $\{x^i, y^i\}_{i=1}^n$, where x^i has 3 dimensions (x_1^i, x_2^i, x_3^i), and the label, $y^i \in \{0, 1\}$, we can define a modified version of the 2-layer network as

follows:

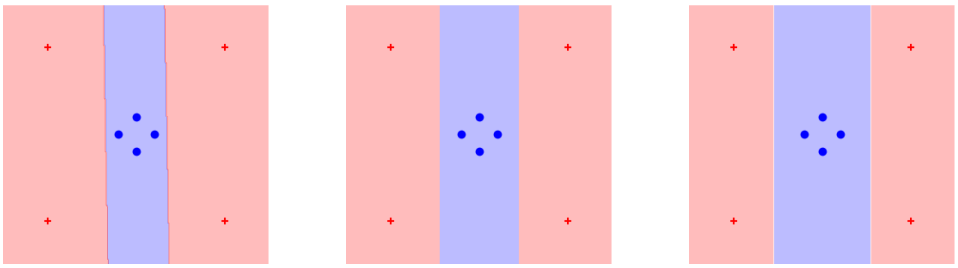
$$
\begin{aligned}
z_h &= W_1 x^i + b_1 \\
h &= \tanh(z_h) \\
z_y &= W_2 h + b_2 \\
y &= \text{sigmoid}(z_y)
\end{aligned}
$$

Further, we can define a loss function between the labels and the output of the MLP as:

$$\ell = \frac{1}{2}(y^i - y)^2,$$

where y^i is the label for input x^i. Derive the backpropagation expression for: $\frac{d\ell}{dW_1}$, the derivative of the loss function with respect to the weight matrix W_1. Note that the dimensions are: $W_1 \in \mathbb{R}^{2\times 3}$, $b_1 \in \mathbb{R}^2$, $W_2 \in \mathbb{R}^2$, and $b_2 \in \mathbb{R}$.

3. **Graduate version.** In this question, we will consider the empirical effects of regularization on the decision boundary of an MLP with a single hidden layer. We trained three different neural network variants to obtain Figures (a)-(c) below. The Figures are the decision boundaries resulting from each training variation. Your task is to decide whether the approach described below could have plausibly produced Figures (a)-(c).



(a) (b) (c)

Increasing vs. Decreasing L2 Regularization. Without changing the neural network architecture, impose an l_2 regularization over all model parameters as follows: $L_{\text{regularized}} = L_{\text{prediction}} + \lambda||w||_2^2$ where $||w||_2$ is the l_2-norm of the parameters of the prediction model and λ is the regularization weight. Does (a) to (c) correspond to increasing values of λ or decreasing values? More specifically, can the changes in the decision boundary from Figures (a) to (c) be attributed to increasing or decreasing regularization. Briefly justify your answer.