# Homework 3
# Decision Trees, k-NN, Perceptron, Regression [*]

10-301 / 10-601 Introduction to Machine Learning (Fall 2023)
http://mlcourse.org

OUT: Friday, September 15
DUE: Saturday, September 23
TAs: Haohui, Rakshith, Sahithya, Siva, Monica, Neural the Narwhal

**Summary** It's time to practice what you've learned! In this assignment, you will answer questions on topics we've covered in class so far, including Decision Trees, K-Nearest Neighbors, Perceptron, and Linear Regression. This assignment consists of a written component split into four sections, one for each topic. These questions are designed to test your understanding of the theoretical and mathematical concepts related to each topic. For each topic, you will also apply your understanding of the concept to the related ideas such as overfitting, error rates, and model selection. This homework is designed to help you apply what you've learned and solve a few concrete problems.

## START HERE: Instructions

- **Collaboration Policy**: Please read the collaboration policy here: http://www.cs.cmu.edu/~mgormley/courses/10601/syllabus.html
- **Late Submission Policy: For this homework, you will only have 2 late days instead of the usual 3.** This allows us to provide feedback before the exam. See the late submission policy here: http://www.cs.cmu.edu/~mgormley/courses/10601/syllabus.html
- **Submitting your work:** You will use Gradescope to submit answers to all questions and code. Please follow instructions at the end of this PDF to correctly submit all your code to Gradescope.
    - **Written:** For written problems such as short answer, multiple choice, derivations, proofs, or plots, please use the provided template. Submissions can be handwritten onto the template, but should be labeled and clearly legible. If your writing is not legible, you will not be awarded marks. If your scanned submission misaligns the template, there will be a 2% penalty. Alternatively, submissions can be written in LaTeX. Each derivation/proof should be completed in the boxes provided. If you do not follow the template, your assignment may not be graded correctly by our AI assisted grader.

[*] Compiled on Friday 15th September, 2023 at 19:36

## Instructions for Specific Problem Types

For “Select One” questions, please fill in the appropriate bubble completely:

**Select One:** Who taught this course?

● Matt Gormley

○ Marie Curie

○ Noam Chomsky

If you need to change your answer, you may cross out the previous answer and bubble in the new answer:

**Select One:** Who taught this course?

● Henry Chai

○ Marie Curie

~~●~~ Noam Chomsky

For “Select all that apply” questions, please fill in all appropriate squares completely:

**Select all that apply:** Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton

□ I don’t know

Again, if you need to change your answer, you may cross out the previous answer(s) and bubble in the new answer(s):

**Select all that apply:** Which are scientists?

■ Stephen Hawking

■ Albert Einstein

■ Isaac Newton

~~■~~ I don’t know

For questions where you must fill in a blank, please make sure your final answer is fully included in the given space. You may cross out answers or parts of answers, but the final answer must still be within the given space.

**Fill in the blank:** What is the course number?

10-601

10-~~6~~301

# 1 LaTeX Bonus Point and Template Alignment (1 points)

1. (1 point) **Select one:** Did you use LaTeX for the entire written portion of this homework?

   ● Yes

   ○ No

2. (0 points) **Select one:** I have ensured that my final submission is aligned with the original template given to me in the handout file and that I haven't deleted or resized any items or made any other modifications which will result in a misaligned template. I understand that incorrectly responding yes to this question will result in a penalty equivalent to 2% of the points on this assignment.
   **Note:** Failing to answer this question will not exempt you from the 2% misalignment penalty.

   ● Yes

# 2 Decision Tree (Revisited) (15 points)

1. Consider the following $4 \times 4$ checkerboard pattern. Suppose our goal is to perfectly classify the range shown such that all black regions are labeled as $+1$ and all white regions are labeled as $-1$. Let the horizontal axis denote feature $x_1$ and vertical axis denote feature $x_2$.

   **NOTE:** If a point is on the border or corner of a region, it has the same label as the region that is above it and/or to its right. For example, $(1, 0)$ has label $+1$, $(1, 1)$ has label $-1$, and $(1, 1.5)$ has label $-1$.

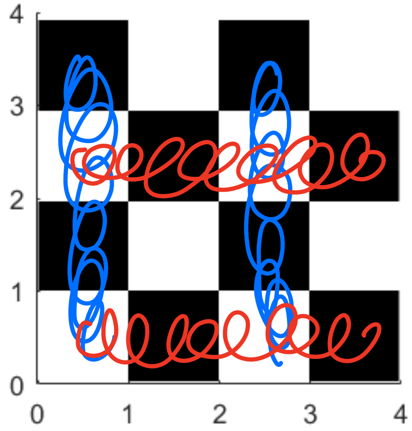


Figure 1: Checkerboard Pattern

(a) (2 points) What is the minimum depth of a binary decision tree that perfectly classifies the colored regions in Figure 1, using *only* features of the form $x_1 < c$ or of the form $x_2 < c$ for different constants $c$?

Your Answer

4

(b) (2 points) What is the minimum depth of a binary decision tree that perfectly classifies the colored regions ($0 < x_1 < 4$) in Figure 1, using features that only inspect either $x_1$ or $x_2$ but not both $x_1$ and $x_2$? Feel free to get creative in your use of the features $x_1$ and $x_2$ for the splits!

Since this is a binary decision tree, we can only use features that split into two branches. Some feature examples: $ceil(x_1)\%2 = 0$, $2 < x_1 < 4$, $x_2 < 1$ or $x_2 > 3$

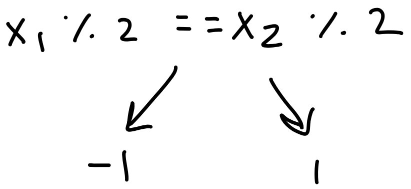
Your Answer

2

(c) (2 points) What is the minimum depth of a binary decision tree that perfectly classifies the colored regions in Figure 1, using ANY features that involve $x_1$, $x_2$, or both?

Your Answer

1



2. Consider the graph below analyzing the size of tree vs. accuracy for a decision tree which has been pruned back to the vertical (red) line. Assume all of our labeled data for this task was randomly divided into a training dataset $D_{train}$, a validation data set $D_{val}$, and a test dataset $D_{test}$. The full tree was trained only on $D_{train}$, then reduced-error pruning was applied using $D_{val}$.

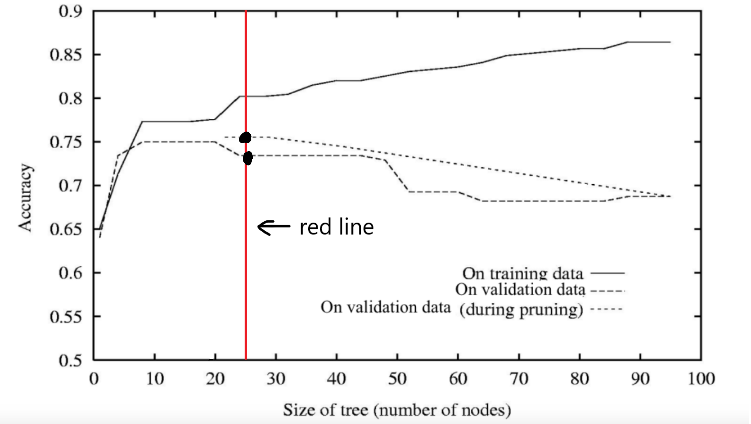


Figure 2: Pruned decision tree. The lowest running dotted curve is the validation performance of the unpruned tree as it grows during training. Finally, we prune it back to the size given by the red line with a reduced validation error (the upper running dotted curve)

(a) (1 point) **Select one:** Refer to Figure 2. Note that $D_{test}$ was not used for training or pruning. When the size of the pruned tree is at **25 nodes**, what is its accuracy on $D_{test}$ likely to be?

○ Slightly higher than the pruned tree's accuracy on $D_{val}$

○ The same as the pruned tree's accuracy on $D_{val}$

● Slightly lower than the pruned tree's accuracy on $D_{val}$

(b) (1 point) **Select one:** Which of the following gives us the best approximation of the true error?

○ Error corresponding to training data $D_{train}$

○ Error corresponding to validation data $D_{val}$

● Error corresponding to test data $D_{test}$

3. (2 points) **Select all that apply:** Which of the following are valid ways to avoid overfitting?

□ Decrease the training set size.

■ Set a threshold for a minimum number of examples required to split at an internal node.

■ Prune the tree so that cross-validation error is minimal.

□ Increase the tree depth.

□ None of the above.

4. Consider a binary classification problem using 1-nearest neighbors with the Euclidean distance metric. We have $N$ 1-dimensional training points $x^{(1)}, x^{(2)}, \ldots x^{(N)}$ and corresponding labels $y^{(1)}, y^{(2)}, \ldots y^{(N)}$, with $x^{(i)} \in \mathbb{R}$ and $y^{(i)} \in \{0, 1\}$.

Assume the points $x^{(1)}, x^{(2)}, \ldots x^{(N)}$ are in ascending order by value. If there are ties during the 1-NN algorithm, we break ties by choosing the label corresponding to the $x^{(i)}$ with lower value.

**NOTE:** For subparts (a) and (b), 2 models "behave exactly the same" when the decision boundaries produced by them are the same, leading to the same performance on all possible test data.

(a) (2 points) **Select one:** Is it possible to build a decision tree that behaves exactly the same as the 1-nearest neighbor classifier? Assume that the decision at each node takes the form of "$x \leq t$" or "$x > t$", where $t \in \mathbb{R}$.

● Yes

○ No

If your answer is yes, please explain how you will construct the decision tree. If your answer is no, explain why it's not possible.

Your answer:

Yes, we would learn a decision tree with real-valued features by turning them into discrete features in the decision tree.

(b) (3 points) **Select all that apply:** Let's add a dimension! Now the training points are 2-dimensional where $\mathbf{x}^{(i)} = \left(x_1^{(i)}, x_2^{(i)}\right) \in \mathbb{R}^2$. For which of the following training sets is it possible to build a decision tree that behaves exactly the same as a 1-nearest neighbor classifier? Assume that (1) the decision at each node takes the form of "$x_j \leq t$" or "$x_j > t$", where $t \in \mathbb{R}$ and $j \in \{1, 2\}$. (2) the decision tree has finite depth.

■ Points: $\{(-5, 5), (5, 5)\}$, Labels: $\{0, 1\}$

■ Points: $\{(0, -9), (0, 10)\}$, Labels: $\{0, 1\}$

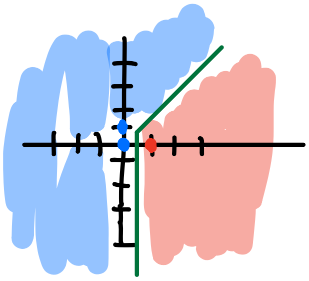
□ Points: $\{(3, 4), (-3, -4)\}$, Labels: $\{0, 1\}$ → no

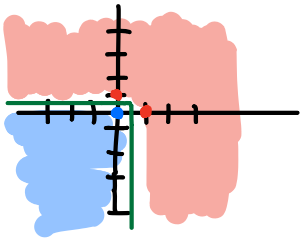
□ Points: $\{(0, 0), (0, 1), (1, 0)\}$, Labels: $\{0, 0, 1\}$ → no

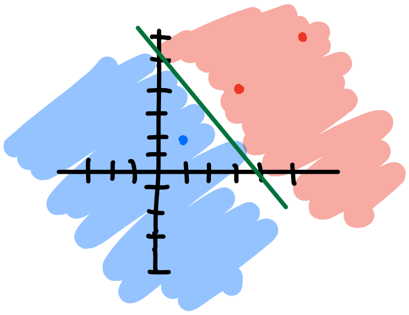
■ Points: $\{(0, 0), (0, 1), (1, 0)\}$, Labels: $\{0, 1, 1\}$

□ Points: $\{(1, 2), (3, 4), (5, 6)\}$, Labels: $\{0, 1, 1\}$ → no

□ None of the above







# 3 $k$-Nearest Neighbors (26 points)

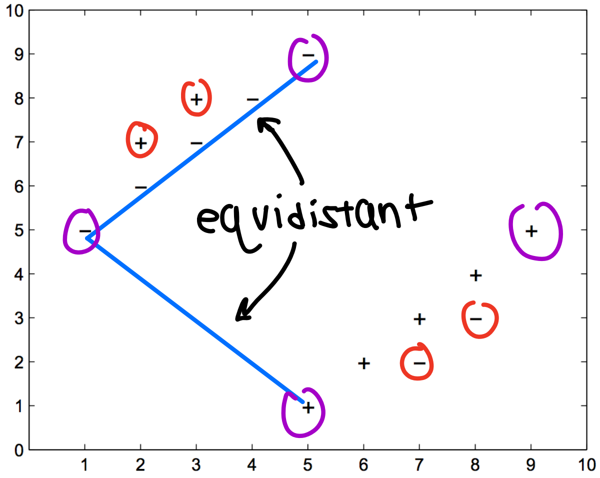


Figure 3: k-NN Dataset

1. Consider a $k$-nearest neighbors ($k$-NN) binary classifier which assigns the class of a test point to be the class of the majority of the $k$-nearest neighbors in the training dataset, according to the Euclidean distance metric. Assume that ties are broken by selecting one of the labels uniformly at random.

   **NOTE:** An example tie scenario can occur when the classes of the 6 nearest neighbors are {+, +, +, -, -, -} i.e. the number of neighbors belonging to each class type is equal. In this case, you can assume the test point's class to be + or - randomly.

   (a) (2 points) Using Figure 3 shown above to train the classifier and choosing $k = 6$, what is the classification error on the training set? **Report your answer either as a fraction or as a decimal with 4 decimal places after the decimal point.**

Your answer:

$\frac{3}{7}$

4 wrong

4 random ties

↳ 50% correct

↳ 2 wrong

$4+2=6 \qquad \frac{6}{14} = \frac{3}{7}$

(b) (2 points) **Select all that apply:** Let's say that we have a new test point (not present in our training data) $\mathbf{x}^{\text{new}} = [3, 9]^T$ that we would like to apply our $k$-NN classifier to, as seen in figure 4.

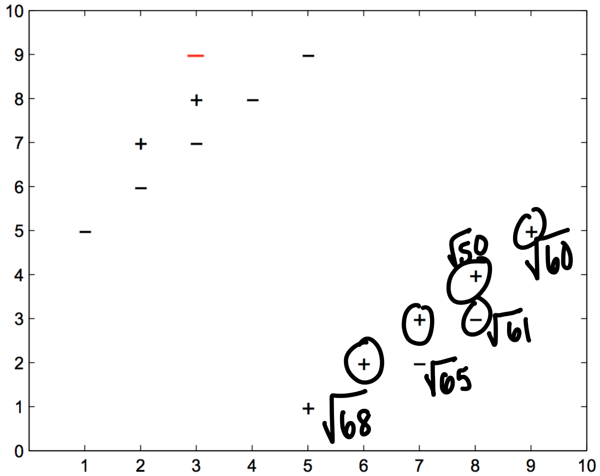


Figure 4: k-NN Dataset with Test Point

For which values of $k$ is this test point always correctly classified by the $k$-NN algorithm?

□ $k = 1$

■ $k = 5$

■ $k = 9$

□ $k = 12$

□ None of the above

2. **Select one:** Assume we have a large labeled dataset that is randomly divided into a training set and a test set, and we would like to classify points in the test set using a $k$-NN classifier.

(a) (1 point) In order to minimize the classification error on this test set, we should always choose the value of $k$ which minimizes the training set error.

○ True

● False

(b) (2 points) **Select one:** Instead of choosing the hyperparameters by merely minimizing the training set error, we instead consider splitting the training-all data set into a training and a validation data set, and choose the hyperparameters that lead to lower validation error. Is choosing hyperparameters based on validation error better than choosing hyper-parameters based on training error?

● Yes, lowering validation error instead of training error is better because lowering training error will not always help generalize our model and may lead to overfitting.

○ Yes, lowering validation error is better for the model because cross-validation guarantees a better test error.

○ No, lowering training error instead of validation error is better because lowering validation error will not help generalize our model and may lead to overfitting.

○ No, lowering training error is better for the model because we have to learn the training set as well as possible to guarantee the best possible test error.

(c) (2 points) **Select one:** Your friend Sally suggests that instead of splitting the original training set into separate training and validation sets, we should instead use the test set as the validation data for choosing hyperparameters. Is this a good idea? Justify your opinion with no more than 3 sentences.

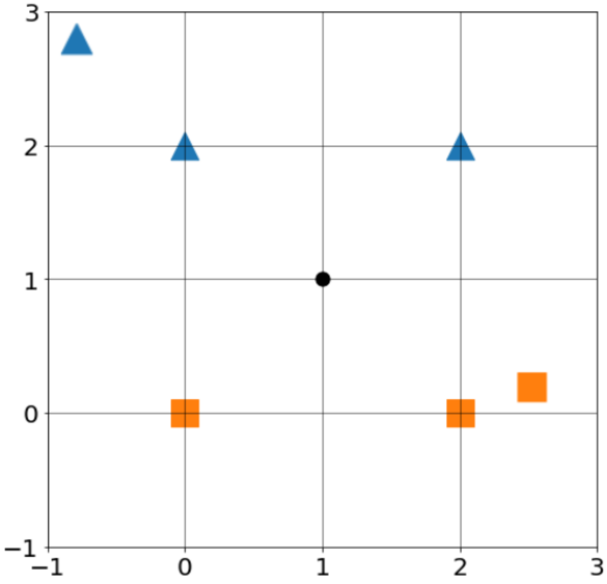
○ Yes

● No

Your answer:

No, we should not use the testing data to inform our model. The test set should not be biased by the data at all.

3. (2 points) **Select all that apply:** Consider a binary $k$-NN classifier where $k = 4$ and the two labels are “triangle” and “square”. Consider classifying a new point $\mathbf{x} = (1, 1)$, where two of the $\mathbf{x}$’s nearest neighbors are labeled “triangle” and two are labeled “square” as shown below.



Which of the following methods can be used to break ties or avoid ties on this dataset?

□ Assign $\mathbf{x}$ the label of its nearest neighbor

■ Flip a coin to randomly assign a label to $\mathbf{x}$ (from the labels of its 4 closest points)

□ Use $k = 3$ instead

■ Use $k = 5$ instead

□ None of the above.

4. (3 points) **Select all that apply:** Which of the following is/are correct statement(s) about $k$-NN models?

■ A larger $k$ tends to give a smoother decision boundary.

■ To reduce the impact of noise or outliers in our data, we should increase the value $k$.

□ If we make $k$ too large, we could end up overfitting the data.

■ We can use cross-validation to help us select the value of $k$.

□ We should never select the $k$ that minimizes the error on the validation dataset.

□ None of the above.

5. Consider the following data concerning the relationship between academic performance and salary after graduation. High school GPA and university GPA are two numerical features and salary is the numerical target. Note that salary is measured in thousands of dollars per year.

| **Student ID** | **High School GPA** | **University GPA** | **Salary** |
|---|---|---|---|
| 1 | 2.5 | 3.8 | 45 |
| 2 | 3.3 | 3.5 | 90 |
| 3 | 4.0 | 4.0 | 142 |
| 4 | 3.0 | 2.0 | 163 |
| 5 | 3.8 | 3.0 | 2600 |
| 6 | 3.3 | 2.8 | 67 |
| 7 | 3.9 | 3.8 | unknown |

(a) (2 points) Among Students 1 to 6, who is the nearest neighbor to Student 7, using Euclidean distance? Answer the Student ID only.

Your answer:

3

(b) (2 points) Now, our task is to predict the salary Student 7 earns after graduation. We apply $k$-NN to this regression problem: the prediction for the numerical target (salary in this example) is equal to the average of salaries for the top $k$ nearest neighbors. If $k = 3$, what is our prediction for Student 7's salary? Be sure to use the same unit of measure (thousands of dollars per year) as the table above.
**Round your answer to the nearest integer.**

Your answer:

944

(c) (2 points) **Select all that apply:** Suppose that the first 6 students shown above are only a subset of your full training data set, which consists of 10,000 students. We apply $k$-NN regression using Euclidean distance to this problem and we define training loss on this full data set to be the mean squared error (MSE) of salary. Now consider the possible consequences of modifying the data in various ways. Which of the following changes **could** have an effect on training loss on the full data set as measured by mean squared error (MSE) of salary?

- ■ Rescaling only "High School GPA" to be a percentage of 4.0
- ■ Rescaling only "University GPA" to be a percentage of 4.0
- □ Rescaling both High School GPA and University GPA by the same percentage/scale
- □ None of the above.

6. An archaeologist discovers a 242 kilobyte 8-inch floppy disk buried beneath the hedges near Wean Hall. The floppy disk contains a few hundred black and white images of 3x3 pixels. You are asked to classify them as either a photo ($y = +$) or artwork ($y = -$) to aid in the analysis.

   You build a k-Nearest Neighbor (k-NN) classifier trained on a training dataset obtained from the web (converted to similarly small black and white images). Suppose you are informed that each image is represented as a $3 \times 3$ matrix $\mathbf{x}$ of binary values and you plan to use Hamming distance to measure the distance between each pair of $3 \times 3$ pixel images as follows:

$$d(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{3} \sum_{j=1}^{3} \mathbb{1}(\mathbf{u}_{i,j} \neq \mathbf{v}_{i,j}) = \text{the number of pixels that differ between } \mathbf{u} \text{ and } \mathbf{v}$$

   While calculating the distance metric, if there is a tie in distance among the points competing for $k$ nearest points, the classifier increases $k$ to include all those tied points in the majority vote. If, in the end, there is a tie in the vote, your classifier returns $\hat{y} =?$. You can try out your k-NN implementation on the images below.

| i | $y$ | $x_{1,1}$ | $x_{1,2}$ | $x_{1,3}$ | $x_{2,1}$ | $x_{2,2}$ | $x_{2,3}$ | $x_{3,1}$ | $x_{3,2}$ | $x_{3,3}$ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | $+$ | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 3 |
| 2 | $+$ | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 4 |
| 3 | $+$ | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 6 |
| 4 | $-$ | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| 5 | $-$ | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 6 |

Table 1: Training Data

| i | $x_{1,1}$ | $x_{1,2}$ | $x_{1,3}$ | $x_{2,1}$ | $x_{2,2}$ | $x_{2,3}$ | $x_{3,1}$ | $x_{3,2}$ | $x_{3,3}$ |
|---|---|---|---|---|---|---|---|---|---|
| 6 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 7 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |

Table 2: Test Data

(a) (1 point) What is the distance between $\mathbf{x}^{(2)}$ and $\mathbf{x}^{(6)}$?

Your answer:

2

(b) (1 point) **Select one:** What would a k-NN classifier with $k = 1$ predict as the label for test point $\mathbf{x}^{(7)}$?

○ $\hat{y} = +$

○ $\hat{y} = -$

● $\hat{y} = ?$

(c) (1 point) **Select one:** What would a k-NN classifier with $k = 3$ predict as the label for test point $\mathbf{x}^{(7)}$?

● $\hat{y} = +$

○ $\hat{y} = -$

○ $\hat{y} = ?$

(d) (1 point) **Select one:** What would a k-NN classifier with $k = 5$ predict as the label for test point $\mathbf{x}^{(7)}$?

● $\hat{y} = +$

○ $\hat{y} = -$

○ $\hat{y} = ?$

(e) (2 points) **Short answer:** Your friend says that you should try using Euclidean distance because it might give better results. Do you agree that switching could lead to lower test error? Why or why not?

Your answer:

NO → The euclidean distance calculations would be similar to using hamming distance. The hamming distance is summing the pixels that are different. Because all the values are zero and one, the euclidean distance would be just the square root of hamming distance. Therefore, the predictions would be the same and so would the test error. Hamming distance is a better metric for comparing the images and because euclidean distance would not lower test error, my friend should not switch to it.

# 4 Perceptron (19 points)

1. (1 point) **Select one:** Consider running the online perceptron algorithm on some sequence of examples $S$ (an example is a data point and its label). Let $S'$ be the same set of examples as $S$, but presented in a different order.

   *True or False:* The online perceptron algorithm is guaranteed to make the same number of mistakes on $S$ as it does on $S'$.

   ○ True

   ● False

2. (3 points) **Select all that apply:** Suppose we have a perceptron whose inputs are 2-dimensional vectors and each feature vector component is either 0 or 1, i.e., $x_i \in \{0, 1\}$. The prediction function is $y = \text{sign}(w_1 x_1 + w_2 x_2 + b)$, and

   $$\text{sign}(z) = \begin{cases} 1, & \text{if } z > 0 \\ 0, & \text{otherwise.} \end{cases}$$

   Which of the following functions can be implemented with the above perceptron? That is, for which of the following functions does there exist a set of parameters $w, b$ that correctly define the function.

   ■ AND function, i.e., the function that evaluates to 1 if and only if all inputs are 1, and 0 otherwise.

   ■ OR function, i.e., the function that evaluates to 1 if and only if at least one of the inputs are 1, and 0 otherwise.

   □ XOR function, i.e., the function that evaluates to 1 if and only if the inputs are not all the same. For example

   $$\text{XOR}(1, 0) = 1, \text{ but } \text{XOR}(1, 1) = 0.$$

   □ None of the above.

3. (2 points) **Select one:** Suppose we have a dataset $\left\{ \left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(N)}, y^{(N)}\right) \right\}$, where $\mathbf{x}^{(i)} \in \mathbb{R}^M$, $y^{(i)} \in \{+1, -1\}$. We would like to apply the perceptron algorithm on this dataset. Assume there is no intercept term. How many parameter values is the perceptron algorithm learning? W

   ○ $N$

   ○ $N \times M$

   ● $M$

4. (2 points) **Select one:** The following table shows a data set and the number of times each point is misclassified during a run of the perceptron algorithm. What is the separating plane $\boldsymbol{\theta}$ found by the algorithm, i.e. $\boldsymbol{\theta} = [b, \theta_1, \theta_2, \theta_3]$? Assume that the initial weights are all zero.

| $x_1$ | $x_2$ | $x_3$ | $y$ | Times Misclassified |
|---|---|---|---|---|
| 2 | 1 | 5 | 1 | 10 |
| 5 | 3 | 3 | 1 | 5 |
| 1 | 6 | 2 | 1 | 8 |
| 7 | 2 | 1 | -1 | 2 |
| 3 | 2 | 6 | -1 | 3 |

25 15 15
8 48 16
14 4 2
9 6 18

→ b=10 W=[20 10 50]
→ b=15 w=[45 25 65]
→ b=23 w=[53 73 81]
→ b=21 w=[39 69 79]
→ b=18 w=[30 63 61]

○ $[18, 25, 14, 34]$

● $[18, 30, 63, 61]$

○ $[16, 56, 18, 47]$

○ $[18, 52, 19, 47]$

5. (2 points) **Select all that apply:** Which of the following is/are correct statement(s) about the mistake bound of the perceptron algorithm?

$\left(\frac{R}{\gamma}\right)^2$

□ If the minimum distance from any data point to the separating hyperplane is increased, without any other change to the data points, the mistake bound will also increase.

■ If the whole dataset is shifted away from origin i.e. translated by adding a constant to each feature, then the mistake bound will also increase. R↑

□ If the pair-wise distance between data points is increased, i.e. the data is scaled by some constant value, then the mistake bound will also increase.

□ The mistake bound is linearly inverse-proportional to the minimum distance of any data point to the separating hyperplane of the data.

□ None of the above.

6. (2 points) **Select one:** Suppose we have data whose examples are of the form $[x_1, x_2]$, where $x_1 - x_2 = 0$. We do not know the label for each element. Suppose the perceptron algorithm starts with $\boldsymbol{\theta} = [3, 5]$; which of the following values will $\boldsymbol{\theta}$ never take on in the process of running the perceptron algorithm on the data?

○ $[-1, 1]$

○ $[4, 6]$

● $[-3, 0]$

○ $[-6, -4]$

$3x_1 + 5x_2 = y$

1 1

$x_1$ $x_2$

$w = [3,5] + [1,1]$

$w = [4,6]$

$x = [-6,-6]$

$w = [4,6] + [-6,-6]$

$w = [-2,0] + [5,5]$

$w = [3,5] + [-4,-4]$

$w = [-1,1] + [-5,-5]$

$w = [-6,-4]$

7. (2 points) **Select all that apply:** Consider the linear decision boundary below and the test dataset shown. Which of the following weight vectors $\boldsymbol{\theta}$ is paired with its corresponding test error on this dataset? (Note: Assume the decision boundary is fixed and does not change while evaluating error.)

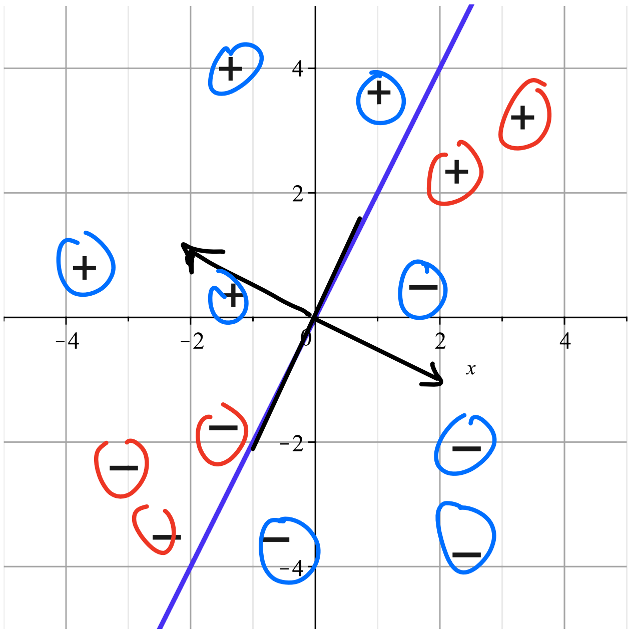
   ■ $\boldsymbol{\theta} = [-2, 1]$, error = 5/13

   ■ $\boldsymbol{\theta} = [2, -1]$, error = 8/13

   □ $\boldsymbol{\theta} = [2, -1]$, error = 5/13

   □ $\boldsymbol{\theta} = [-2, 1]$, error = 8/13

   □ None of the above.



8. The following problem will walk you through an application of the Perceptron Mistake Bound. The following table shows a linearly separable dataset, and your task will be to determine the mistake bound for the dataset.

**NOTE:** The proof of the perceptron mistake bound requires that the optimal linear separator passes through the origin. To make the linear separator pass through the origin, we fold the bias into the weights and prepend a 1 to each training example's input. The original data is on the left, and the result of this prepending is shown on the right. **Be sure to use the modified dataset on the right in your calculations.**

| $x_1$ | $x_2$ | $y$ |
|---|---|---|
| -2 | 2 | 1 |
| -1 | -3 | -1 |
| -2 | -3 | -1 |
| 0 | 1 | 1 |
| 2 | -1 | 1 |

| $\boldsymbol{x_0}$ | $x_1$ | $x_2$ | $y$ |
|---|---|---|---|
| **1** | -2 | 2 | 1 |
| **1** | -1 | -3 | -1 |
| **1** | -2 | -3 | -1 |
| **1** | 0 | 1 | 1 |
| **1** | 2 | -1 | 1 |

8/5 → 1.0243

9/5

12/5

2

8/5

(a) (2 points) Compute the radius $R$ of the "circle" centered at the origin that bounds the data points. **Round to 4 decimal places after the decimal point.**

Radius:

3.7417

(b) (2 points) Assume that the linear separator with the largest margin is given by

$$\boldsymbol{\theta}^{*T}\begin{bmatrix}1\\x_1\\x_2\end{bmatrix} = 0,\text{, where } \boldsymbol{\theta}^{*} = \begin{bmatrix}6\\3\\4\end{bmatrix}$$

$$\frac{|\theta^{*} x|}{||\theta^{*T}||}$$

Now, compute the margin of the dataset.
**Round to 4 decimal places after the decimal point.**

Margin:

1.0243

$$\frac{6\cdot 1 + 3(-2)+4(2)}{\sqrt{6^2+3^2+4^2}}$$

(c) (1 point) Based on the above rounded values, what is the theoretical perceptron mistake bound for this dataset, given this linear separator?
**Round to 4 decimal places after the decimal point.**

Mistake Bound:

13.3439

$$\left(\frac{R}{\gamma}\right)^2$$

# 5 Linear Regression (19 points)

1. Consider the following dataset:

| x | 9.0 | 2.0 | 6.0 | 1.0 | 8.0 |
|---|---|---|---|---|---|
| y | 1.0 | 0.0 | 3.0 | 0.0 | 1.0 |

Let $\boldsymbol{x}$ be the vector of datapoints and $\boldsymbol{y}$ be the label vector. Here, we are fitting the data using gradient descent, and our objective function is $J(w, b) = \frac{1}{N}\sum_{i=1}^{N}(wx_i + b - y_i)^2$ where $N$ is the number of data points, $w$ is the weight, and $b$ is the intercept.

$2(wx_i + b - y_i)\, x_i$

**Note:** Showing your work in these questions is optional, but it is recommended to help us understand where any misconceptions may occur. We may give partial credit for correct work if your answer is incorrect.

(a) (2 points) If we initialize the weight as 3.0 and intercept as 0.0, what is the gradient of the loss function with respect to the weight $w$, calculated over all the data points, in the first step of the gradient descent update?
**Round to 4 decimal places after the decimal point.**

Gradient:

209.2000

$w=3 \quad b=0$

Work

$$\nabla_w J^{(i)}(w,b) = \begin{bmatrix} dJ/dw_1 \\ \vdots \\ dJ/dw_m \end{bmatrix} = 2(w\,x^{(i)} + b - y^{(i)})\,\vec{x}^{(i)}$$

$$\nabla J(w) = \begin{bmatrix} 2(3\cdot 9 + 0 - 1)\cdot 9 \\ 2(3\cdot 2 + 0 - 0)\cdot 2 \\ 2(3\cdot 6 + 0 - 3)\cdot 6 \\ 2(3\cdot 1 + 0 - 0)\cdot 1 \\ 2(3\cdot 8 + 0 - 1)\cdot 8 \end{bmatrix} = \begin{bmatrix} 468 \\ 24 \\ 180 \\ 6 \\ 368 \end{bmatrix} \rightarrow 209.2$$

(b) (2 points) What is the gradient of the loss function with respect to the intercept $b$, calculated over all the data points, in the first step of the gradient descent update?

Gradient:

29.2000

Work

$$\nabla_b J^{(i)}(w,b) = \begin{bmatrix} dJ/db, \\ \vdots \\ dJ/dom \end{bmatrix} = 2(w x^{(i)} + b - y^{(i)})$$

$$\nabla J(b) = \begin{bmatrix} 2(3\cdot 9+0-1) \\ 2(3\cdot 2+0-0) \\ 2(3\cdot 6+0-3) \\ 2(3\cdot 1+0-0) \\ 2(3\cdot 8+0-1) \end{bmatrix} = \begin{bmatrix} 52 \\ 12 \\ 30 \\ 6 \\ 46 \end{bmatrix}$$

(c) (2 points) Let the learning rate be 0.01. Perform one step of gradient descent on the data. Fill in the following blanks with the value of the weight and the value of the intercept after this step. **Round to 4 decimal places after the decimal point.**

Weight:

0.9080

Intercept:

-0.2920

$$\vec{\theta} \leftarrow \vec{\theta} - \gamma \vec{g}$$

$$\vec{w} \leftarrow \vec{w} - \gamma \nabla_w J$$

$$\vec{b} \leftarrow \vec{b} - \gamma \nabla_b J$$

$$w = 3 - 0.01(209.2)$$

$$w = 0.908$$

$$b = 0 - 0.01(29.2)$$

$$b = 0.2920$$

2. Consider a dataset $\mathcal{D}_1 = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)})\}$. Assume the linear regression model that minimizes the mean-squared error on $\mathcal{D}_1$ is $y = w_1 x + b_1$.

   (a) (2 points) **Select one:** Now, suppose we have the dataset $\mathcal{D}_2 = \{(x^{(1)} + \alpha,\ y^{(1)} + \beta), \ldots, (x^{(N)} + \alpha,\ y^{(N)} + \beta)\}$ where $\alpha > 0, \beta > 0$ and $w_1\alpha \neq \beta$. Assume the linear regression model that minimizes the mean-squared error on $\mathcal{D}_2$ is $y = w_2 x + b_2$. Select the correct statement about $w_1, w_2, b_1, b_2$ below. Note that the statement should hold no matter what values $\alpha, \beta$ take on within the specified constraints.

   ○ $w_1 = w_2, b_1 = b_2$

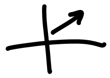
   ○ $w_1 \neq w_2, b_1 = b_2$

   ● $w_1 = w_2, b_1 \neq b_2$

   ○ $w_1 \neq w_2, b_1 \neq b_2$

   $y_1 = w_1 x + b_1$

   $y_2 = W_2 x + b_2$ $x + \alpha$ $y + \beta$



   (b) (2 points) We decide to ask a friend to analyze $\mathcal{D}_1$; however, he makes a mistake by duplicating a subset of the rows in $\mathcal{D}_1$. Explain why the linear regression parameters that minimize mean-squared error on the duplicated data *may* differ from the parameters learned on $\mathcal{D}_1$, i.e. $w_1$ and $b_1$.

   Your answer:

   The duplicated rows could have been of outliers in the data which would bias the regression parameters to try to fit to noise in the dataset. This may slightly shift $w_1$ and/or $b_1$ to provide extra weightage to the repeated point.

3. We wish to learn a linear regression model on the dataset $\mathcal{D} = \{(\boldsymbol{x}^{(1)}, y^{(1)}), \ldots, (\boldsymbol{x}^{(N)}, y^{(N)})\}$ where $\boldsymbol{x} \in \mathbb{R}^k$. For this question, define the *log-cosh loss* as

$$\ell(\hat{y}, y) = \log(\cosh(\hat{y} - y))$$

   In this question, log function has base **e**.

   In particular, for a given point $\boldsymbol{x}^{(i)}$, the log-cosh loss of a model with parameters $\boldsymbol{\theta}$ is

$$J^{(i)}(\boldsymbol{\theta}) = \log\left(\cosh\left(\boldsymbol{\theta}^T \boldsymbol{x}^{(i)} - y^{(i)}\right)\right)$$

   We are interested in minimizing loss over our training data, so we minimize the average log-cosh loss over all points in $\mathcal{D}$.

   (a) (2 points) What is the objective $J(\boldsymbol{\theta})$ in this setting?

Your answer:

$$J(\theta) = \frac{1}{N} \sum_{i=1}^{N} \log(\cosh(\theta^T x^{(i)} - y^{(i)}))$$

(b) (3 points) What is the partial derivative of $J^{(i)}(\boldsymbol{\theta})$ with respect to the $j^{\text{th}}$ parameter, $\theta_j$? It may be helpful to know that $\frac{d}{dx}\cosh(x) = \sinh(x)$.

Your answer:

$J^{(i)}(\boldsymbol{\theta}) = \log\left(\cosh\left(\boldsymbol{\theta}^T \boldsymbol{x}^{(i)} - y^{(i)}\right)\right)$

$$\frac{\partial J^{(i)}(\theta)}{\partial \theta_j} = \frac{1}{\cosh(\theta^T x^{(i)} - y^{(i)})} \cdot \sinh(\theta^T x^{(i)} - y^{(i)}) (x_j^{(i)})$$

$$\frac{\partial J^{(i)}(\theta)}{\partial \theta_j} = \frac{\sinh(\theta^T x^{(i)} - y^{(i)}) (x_j^{(i)})}{\cosh(\theta^T x^{(i)} - y^{(i)})}$$

(c) (2 points) What is the gradient of $J^{(i)}(\boldsymbol{\theta})$ with respect to the entire parameter vector $\boldsymbol{\theta}$?

Your answer:

$$\frac{\partial J^{(i)}(\theta)}{\partial \theta} = \frac{\sinh(\theta^T x^{(i)} - y^{(i)}) (\vec{x}^{(i)})}{\cosh(\theta^T x^{(i)} - y^{(i)})}$$

(d) (2 points) To find the optimal parameter vector $\boldsymbol{\theta}^*$ that minimizes $J(\boldsymbol{\theta})$, we again decide to use gradient descent. Write pseudocode that performs gradient descent for one iteration. Set learning rate $\alpha = 0.1$ and initialize $\boldsymbol{\theta}$ to be the zero vector. You may use `gradient[i]` as a variable that contains your answer to part (c) in your pseudocode. Limit your answer to 10 lines.

Your Answer

$\theta = \text{zeros}(1+M)$ → where $M = \#$ of features

$\alpha = 0.1$

$\theta = \theta - \gamma g$

$g = \text{gradient}[i]$

## 6 Collaboration Questions

After you have completed all other components of this assignment, report your answers to these questions regarding the collaboration policy. Details of the policy can be found here.

1. Did you receive any help whatsoever from anyone in solving this assignment? If so, include full details.
2. Did you give any help whatsoever to anyone in solving this assignment? If so, include full details.
3. Did you find or come across code that implements any part of this assignment? If so, include full details.

Your Answer

1. NO
2. NO
3. NO