COGS 118A, Fall 2018: Introduction to Machine Learning I

Homework Assignment 2

Due: 11:59pm, Sunday, October 14, 2018 (Pacific Time).

Instructions: Answer the questions below, attach your code, and insert figures to create a PDF file; submit your file via Gradescope. You may look up the information on the Internet, but you must write the final homework solutions by yourself.

Late Policy: 5% of the total points will be deducted on the first day past due. Every 10% of the total points will be deducted for every extra day past due.

Grade: ____ out of 100 points

1 (25 points) Decision Boundary

1.1 (5 points)

We are given a classifier that performs classification in $\mathbb{R}^2$ (the space of data points with 2 features (x_1, x_2)) with the following decision rule:

$$h(x_1, x_2) = \begin{cases} 1, & \text{if} \quad x_1 + 2x_2 - 4 \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Draw the decision boundary of the classifier and shade the region where the classifier predicts 1. Make sure you have marked the x_1 and x_2 axes and the intercept points on those axes.

Ans:

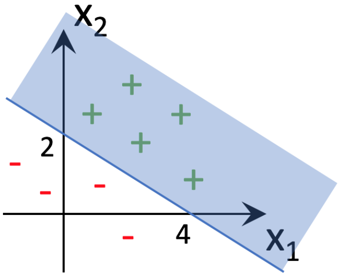


Figure 1: Answer of decision boundary.

1.2 (10 points)

We are given a classifier that performs classification on $\mathbb{R}^2$ (the space of data points with 2 features (x_1, x_2)) with the following decision rule:

$$h(x_1, x_2) = \begin{cases} 1, & \text{if} \quad w_1x_1 + w_2x_2 + b \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Here, the normal vector $\mathbf{w}$ of the hyperplane (decision boundary) is normalized, i.e.:

$$||\mathbf{w}||_2 = \sqrt{w_1^2 + w_2^2} = 1.$$

1. Compute the parameters w_1, w_2 and b for the decision boundary in Figure 2. Please make sure the prediction of the decision boundary you got is consistent with Figure 2. **Hint**: Utilize the intercepts in the figure to find the relation between w_1, w_2 and b. Then, substitute it into the normalization constraint to solve the values for the parameters.

2. Compute the predictive labels of the following two data points: A = (3,2), B = (-1,0).

Ans:

$$\frac{\sqrt{2}}{2}x_1 + \frac{\sqrt{2}}{2}x_2 - \frac{\sqrt{2}}{2} = 0$$

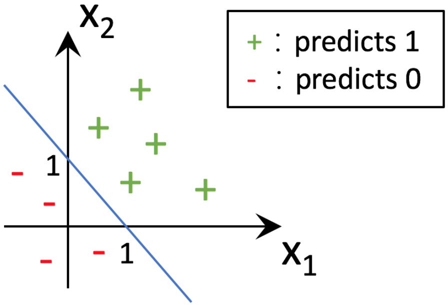


Figure 2: Decision boundary to solve the parameters.

1.3 (10 points)

We are given a classifier that performs classification on $\mathbb{R}^3$ (the space of data points with 3 features (x_1, x_2, x_3)) with the following decision rule:

$$h(x_1, x_2, x_3) = \begin{cases} 1, & \text{if} \quad w_1x_1 + w_2x_2 + w_3x_3 + b \geq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Here, the normal vector $\mathbf{w}$ of the hyperplane (decision boundary) is normalized, i.e.:

$$||\mathbf{w}||_2 = \sqrt{w_1^2 + w_2^2 + w_3^2} = 1.$$

In addition, we set $b \leq 0$ to have an unique equation for the decision boundary.

1. Compute the parameters w_1, w_2, w_3 and b for the decision boundary that passes through three points A = (3,2,4), B = (-1,0,2), C=(4,1,5) in Figure 3.

   Hint: Note that the normal vector is orthogonal to the hyperplane, which means the normal vector is orthogonal to any vector on the hyperplane. One way to compute the normal vector is to find two nonparallel vectors on the hyperplane, and use the orthogonal property plus the normalization constraint to solve the values for $\mathbf{w}$. Or you can set the value of b as any arbitrary value and solve for $\mathbf{w}$, then come back to solve the true value of b.

2. Compute the predictive labels of the following three data points: p = (0,0,0), q = (1,0,5).

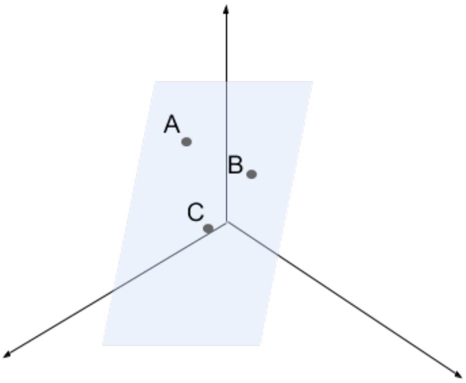


Figure 3: Decision boundary to solve the parameters.

Ans:

$$w_1 = -\frac{2}{\sqrt{14}}, w_2 = \frac{1}{\sqrt{14}}, w_3 = \frac{3}{\sqrt{14}}, b = -\frac{8}{\sqrt{14}}$$

$$f(p) = -\frac{8}{\sqrt{14}} < 0 \rightarrow \text{p is labeled as 0}$$

$$f(q) = \frac{5}{\sqrt{14}} > 0 \rightarrow \text{q is labeled as 1}$$

2 (15 points) Conditional Probability

Oftentimes, the performance of a binary medical diagnostic test is measured as follows:

1. True positive rate (correctly identified) = $P(test + |sick+)$, i.e. the probability that a sick person correctly diagnosed as sick.
2. False positive rate (incorrectly identified) = $P(test + |sick-)$, i.e. the probability that a healthy person incorrectly identified as sick.
3. True negative rate (correctly rejected) = $P(test - |sick-)$, i.e. the probability that a healthy person correctly identified as healthy.
4. False negative rate (incorrectly rejected) = $P(test - |sick+)$, i.e. the probability that a sick person incorrectly identified as healthy.

Here, we look at a particular mammogram tests for breast cancer. The true positive rate is 98%. The true negative rate is 94%. The incident rate of breast cancer among a certain population is 0.06%. Suppose that a person is randomly drawn from the population.

2.1 (5 points)

Given that the person is tested as positive, what is the probability of the person has breast cancer? In other words, what is $P(cancer + |test+)$? (the *cancer+* means *sick+* in slides and previous page)

Ans:

$$\begin{aligned} P(C + |T+) &= \frac{P(C+, T+)}{P(T+)} \\ &= \frac{P(T + |C+)P(C+)}{P(T + |C+)P(C+) + P(T + |C-)P(C-)} \\ &= \frac{0.98 \times 0.0006}{0.98 \times 0.0006 + (1 - 0.94) \times (1 - 0.0006)} \\ &= 0.971\% \end{aligned}$$

2.2 (5 points)

Given that the person is tested as negative, what is the probability of the person does **not** has breast cancer? In other words, what is $P(cancer - |test-)$?

Ans:

$$\begin{aligned} P(C - |T-) &= \frac{P(C-, T-)}{P(T-)} \\ &= \frac{P(T - |C-)P(C-)}{P(T - |C+)P(C+) + P(T - |C-)P(C-)} \\ &= \frac{0.94 \times (1 - 0.0006)}{(1 - 0.98) \times 0.0006 + 0.94 \times (1 - 0.0006)} \\ &= 99.9987\% \qquad (1) \end{aligned}$$

2.3 (5 points)

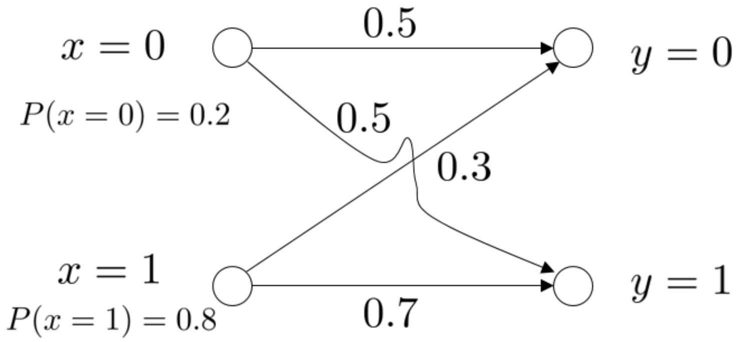
Compute *precision*, *recall*, and $F - value = \dfrac{2 \times precision \times recall}{precision + recall}$.

Ans:

$$\begin{aligned}\text{Precision} &= P(C + | T+) = 0.971\% \\ \text{Recall} &= P(T + | C+) = 98\% \\ \text{F-value} &= \frac{2 * 0.971\% * 98\%}{0.971\% + 98\%} = 0.0192\end{aligned}$$

3 (10 points) Binary Communication System

For the binary communication system shown below, compute the following probabilities:



(a) $P(x = 2)$

(b) $P(y = 0|x = 1)$

(c) $P(y = 0)$

(d) $P(x = 1|y = 0)$

Ans:

$$\begin{aligned}
P(X = 2) &= 0 \\
P(Y = 0|X = 1) &= 0.3 \\
P(Y = 0) &= P(Y = 0|X = 0)P(X = 0) + P(Y = 0|X = 1)P(X = 1) \\
&= 0.5 \cdot 0.2 + 0.3 \cdot 0.8 = 0.34 \\
P(X = 1|Y = 0) &= \frac{P(X = 1, Y = 0)}{P(Y = 0)} \\
&= \frac{P(Y = 0|X = 1)P(X = 1)}{P(Y = 0)} \\
&= \frac{0.3 \cdot 0.8}{0.34} = 0.7058823529
\end{aligned}$$

4 (15 points) Minimizers and Maximizers

4.1 (5 points) Probability

The joint probability mass function of the random variables (x, y) is given by the following table. Compute the following:

	$x = 0$	$x = 1$	$x = 2$
$y = 0$	0.2	0.1	0.1
$y = 1$	0.3	0.2	0.1

1. $(i^*, j^*) = \arg\max_{(i,j)} P(x = i, y = j)$
   Ans:
   $$\arg\max_{(i,j)} P(x = i, y = j) = 0.3 \implies (i^*, j^*) = (0, 1)$$

2. $j^* = \arg\min_j P(y = j|x = 1)$
   Ans:
   $$\arg\min_{j} P(y = j|x = 1) = \frac{0.1}{0.1 + 0.2} \approx 0.33 \implies j^* = 0$$

3. $i^* = \arg\max_i P(x = i)$
   Ans:
   $$\arg\max_{i} P(x = i) = 0.2 + 0.3 = 0.5 \implies i^* = 0$$

4.2 (10 points) Function

(Check Lecture Slide 4, Page 44-48) An unknown estimator is given an estimation problem to find the maximizer of the objective function $G(\theta) \in (0, 2]$:

$$\theta_1 = \arg\max_{\theta} G(\theta). \tag{2}$$

The solution to Eq. 2 by the estimator is $\theta_1 = 67$. Given this information, obtain θ^* such that

$$\theta^* = \arg\min_{\theta}[10 - 3 \times \ln(G(\theta))]. \tag{3}$$

Ans: Since ln and scalar multiplication and subtraction are all monotonic functions

$$\begin{aligned} \arg\max_{\theta} G(\theta) &= \arg\max_{\theta}[3 \times \ln(G(\theta))] \\ &= \arg\min_{\theta}[-3 \times \ln(G(\theta))] \\ &= \arg\min_{\theta}[10 - 3 \times \ln(G(\theta))] \end{aligned}$$

Thus the optimal $\theta^* = \theta_1 = 67$

5 (10 points) Training vs. Testing Errors

In this problem, we are given two trained predictive models on the Iris dataset (Pre-processed, see the Jupyter Notebook). Each data point $\mathbf{x}_i \in \mathbb{R}^4$ has 4 features and its corresponding label $y_i \in \{0, 1\}$, where $i \in \{1, 2, \ldots, 150\}$. To predict on the new data, here we consider two types of model, a regression model and a classification model. The regression model is trained to predict a real-value label, while the classification model adds a threshold on the output of the regression model, converting the real-value label into a binary label.

The regression model is as followed:

$$\hat{y}_i(\mathbf{x_i}) = \mathbf{w}^T \mathbf{x_i} + b$$

The classifier is as followed:

$$h(\mathbf{x_i}) = \begin{cases} 1, & \text{if} \quad \hat{y}_i(\mathbf{x_i}) \geq 1/2 \\ 0, & \text{otherwise.} \end{cases},$$

where $\mathbf{w} = [0.1297, 0.1225, -0.1171, 0.6710]^T, b = -1.1699$. The regression error is defined as $\sqrt{\frac{\sum_i^N (\hat{y}_i - y_i)^2}{N}}$, and the classification error is defined as $\frac{\sum_i \mathbf{1}(h(\mathbf{x_i}) \neq y_i)}{N}$, where N is the number of data points.
The data as well as the split of training and testing set is given in the Jupyter Notebook we provided. Here, we will evaluate the following predictive errors given the trained model. **You should not use the scikit-learn library**.

- Training error of the regression model.
- Testing error of the regression model.
- Training error of the classification model.
- Testing error of the classification model.

Ans:

```python
def reg_and_class_err(x, y):
    w = reg.coef_
    b = reg.intercept_
    reg_diff = 0
    class_diff = 0
    for i in range(len(x)):
        # prediction based on x
        y_hat = np.dot(w, x[i,:]) + b

        # regression error, doing the sum here
        reg_diff += (y_hat - y[i]) ** 2

        # classification error
        y_hat_binary = 1 if y_hat >= 0.5 else 0
        class_diff += (y_hat_binary != y[i])

    # regression error, calculate the mean and square root here
    reg_diff = (reg_diff / len(x)) ** (0.5)
    class_diff /= len(x)
    return reg_diff, class_diff

print('Training regression and classification errors are:')
print(reg_and_class_err(X_train, Y_train))
print('Testing regression and classification errors are:')
print(reg_and_class_err(X_test, Y_test))
```

```
Training regression and classification errors are:
(0.27976412743241214, 0.06)
Testing regression and classification errors are:
(0.33100713441395574, 0.14)
```

6 (25 points) Decision Stump

In this problem, we will perform a binary classification task on the Iris dataset. Again, this dataset has 150 data points, where each data point $\mathbf{x} \in \mathbb{R}^4$ has 4 features and its corresponding label $y \in \{0, 1\}$.

To classify these 2 labels above, we decide to utilize a decision stump. The decision stump works as follows (for simplicity, we restrict our attention to uni-directional decision stumps):

- Given the j-th feature $\mathbf{x}_i(j)$ and a threshold Th_j, for data point i, the classification function is defined by $y = f(\mathbf{x}, j, Th_j)$ as:

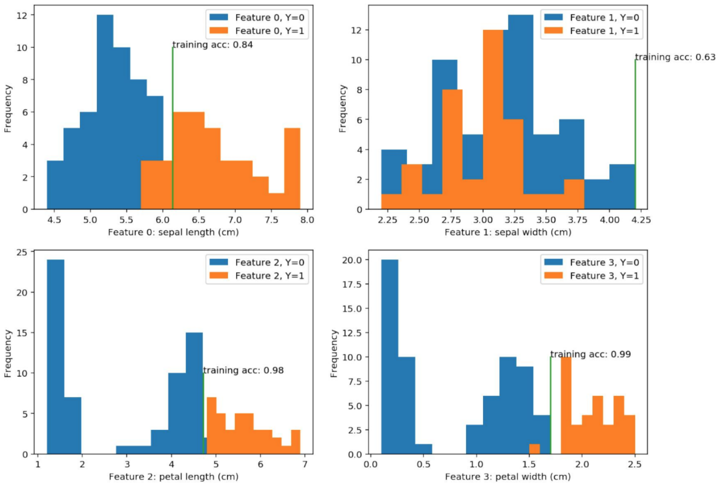
$$f(\mathbf{x_i}, j, Th_j) = \begin{cases} 1 & if\ \mathbf{x}(j) > Th_j \\ 0 & otherwise. \end{cases}$$

Based on the decision stump above, we wish to write an algorithm to find the **best feature** and **best threshold** on training set to create a "best" decision stump, in a sense that such decision stump achieves the **highest accuracy on training set**.

Follow the instructions in the skeleton code and report:

- All 4 histograms in last part of the code.
- The best feature, best threshold, training and test accuracy in last part of the code.

Ans:



- Best feature: 3
- Best threshold: 1.70
- Training accuracy of best feature: 0.99
- Test accuracy of best feature: 0.90

```python
    for i,j in zip(Xj, Y):

        print(list(zip(Xj,Y)))
        # Check against threshold
        if (Xj[i] > thres):
            f = 1
        else:
            f = 0

        # check the prediction
        if(f == Y[i]):
            n_correct = n_correct + 1
        else:
            n_incorrect = n_incorrect + 1
```