Homework Assignment 1

CSE 251A: ML - Learning Algorithms

Due: April 11th, 2023, 9:30am (Pacific Time)

Instructions: Please answer the questions below, attach your code in the document, and insert figures to create **a single PDF file**. You may search information online but you will need to write code/find solutions to answer the questions yourself.

Grade: ____ out of 100 points

1 (10 points) Classification vs. Clustering

In this question, you are provided with several scenarios. You need to identify if the given scenario is better formulated as a *classification* task or a *clustering* task. You should also provide the reason that supports your choice.

1. Scenario 1: Assume there are 100 graded answer sheets for a homework assignment (scores range from 0 to 100). We would like to split them into several groups where each group has similar scores.

   Choice: clustering task

   Reason:

   Because the 100 grades are not labeled. Unsupervised learning should use clustering task

2. Scenario 2: Assume there are 100 graded answer sheets for a homework assignment (scores range from 0 to 100). We would like to split them into several groups where each group represents a letter grade (A, B, C, D) following the criteria: A (90-100), B (75-90), C (60-75), D (0-60).

   Choice: classification task

   Reason:

   Now the 100 grades are labeled, Supervised learning should use classification task

2 (40 points) Basic Calculus

2.1 (20 points) Derivatives with Scalars

1. $f(x) = \frac{1}{2}(ax - b)^2$ where $a, b \in \mathbb{R}$ are constant scalars, derive $\frac{\partial f(x)}{\partial x}$.

$$\frac{\partial f(x)}{\partial x} = a(ax - b)$$
$$= a^2x - ab$$

2. $f(x) = \ln(1 + e^x)$, derive $\frac{\partial f(x)}{\partial x}$.

$$\frac{\partial f(x)}{\partial x} = \frac{e^x}{1 + e^x}$$

2.2 (20 points) Derivatives with Vectors

Several particular vector derivatives are useful for this course. For matrix $\mathbf{A} \in \mathbb{R}^{M \times M}$, column vector $\mathbf{x} \in \mathbb{R}^M$ and $\mathbf{a} \in \mathbb{R}^M$, we have

- $\frac{\partial \mathbf{a}^\top \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^\top \mathbf{a}}{\partial \mathbf{x}} = \mathbf{a}$,
- $\frac{\partial \mathbf{x}^\top \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}^\top)\mathbf{x}$. If $\mathbf{A}$ is symmetric, $\frac{\partial \mathbf{x}^\top \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{A}\mathbf{x}$.
  A special case is, if $\mathbf{A} = \mathbf{I}$ (identity matrix), $\frac{\partial \mathbf{x}^\top \mathbf{x}}{\partial \mathbf{x}} = \frac{\partial \mathbf{x}^\top \mathbf{I}\mathbf{x}}{\partial \mathbf{x}} = 2\mathbf{I}\mathbf{x} = 2\mathbf{x}$.

The above rules adopt a *denominator-layout* notation. For more rules, you can refer to this Wikipedia page. Please apply the above rules and calculate following derivatives:

1. $f(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{a})^\top(\mathbf{x} - \mathbf{a})$ where $\mathbf{a} \in \mathbb{R}^M$ is a constant vector, derive $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$.

$f(x) = \frac{1}{2}(x^Tx - a^Tx - xa + a^Ta)$

$\frac{\partial f(x)}{\partial x} = \frac{1}{2}(2x - a - a)$

$= x - a$

2. $f(\mathbf{x}) = \frac{1}{2}(\mathbf{A}\mathbf{x} - \mathbf{b})^\top(\mathbf{A}\mathbf{x} - \mathbf{b})$ where $\mathbf{A} \in \mathbb{R}^{M \times M}$ is a constant matrix and $\mathbf{b} \in \mathbb{R}^M$ is a constant vector, derive $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$.

   Hint: Note that $(\mathbf{A}^\top\mathbf{A})^\top = \mathbf{A}^\top\mathbf{A}$, thus $\mathbf{A}^\top\mathbf{A}$ is a symmetric matrix.

$f(x) = \frac{1}{2}(x^TA^T - b^T)(Ax - b)$

$\frac{\partial f(x)}{\partial x} = \frac{1}{2}(x^TA^TAx - b^TAx - x^TA^Tb + b^Tb)$

$= \frac{1}{2}(2A^TAx - A^Tb - A^Tb)$

$= A^TAx - A^Tb$

3 (20 points) Metrics

In machine learning, we have many metrics to evaluate the performance of our model. For example, in a binary classification task, there is a dataset $S = \{(\mathbf{x}_i, y_i), i = 1, .., N\}$ where each data point $(\mathbf{x}, y)$ contains a feature vector $\mathbf{x} \in \mathbb{R}^M$ and a ground-truth label $y \in \{0, 1\}$. We have obtained a classifier $f : \mathbb{R}^M \to \{0, 1\}$ to predict the label $\hat{y}$ of feature vector $\mathbf{x}$:

$$\hat{y} = f(\mathbf{x})$$

Assume $N = 200$ and we have the following *confusion matrix* to represent the result of classifier f on dataset S:

	Actual Positives ($y = 1$)	Actual Negatives ($y = 0$)
Predicted Positives ($\hat{y} = 1$)	5	5
Predicted Negatives ($\hat{y} = 0$)	10	180

Please follow the lecture notes to compute the metrics below:

1. Please compute the *accuracy* of the classifier f on dataset S.

$$ACC = \frac{TP+TN}{ALL} = \frac{5+180}{200} = 92.5\%$$

2. Please compute the *precision* of the classifier f on dataset S.

$$pre = \frac{TP}{TP+FP} = \frac{5}{10} = 50\%$$

3. Please compute the *F1 score* of the classifier f on dataset S.

$$\text{recall} = \frac{TP}{TP+FN} = 33.3\%$$

$$F1 = 2 \times \frac{\text{pre} \cdot \text{recall}}{\text{pre} + \text{recall}} = 2 \times \frac{\frac{1}{2} \times \frac{1}{3}}{\frac{1}{2} + \frac{1}{3}} = 2 \times \frac{\frac{1}{6}}{\frac{5}{6}}$$

$$= 40\%$$

4. You may find the accuracy of current model very high. Does it mean the performance of this model is always very good? Why?

   Hint: You may refer to other metrics you have computed.

No, the data is not balanced because there is far more negative than positive data. Even the model just predicts negative, the accuracy is still pretty high. From the calculation, the precision of the model is only 50% and F1 score is 40%, so the performance of the model can still be increased over positive cases.

4 (10 points) Loss functions

1. Prove the Rooted Mean Square Error (RMSE) is always greater than or equal to the Mean Absolute Error (MAE).

   Hint: You may use Cauchy-Schwarz inequality.

$\vec{a} = (y_{p_1} - y_{a_1}, y_{p_2} - y_{a_2} \cdots y)$ $(n \times 1)$ $\vec{b} = (1, \cdots 1)$ $(n \times 1)$

$$ab = \sum_{i}^{n} (y_{pi} - y_{ai})$$

$$\|a\| = \sqrt{\sum_{i}^{n} (y_{pi} - y_{ai})^2} \qquad \|b\| = \sqrt{n}$$

$$a \cdot b \leq \|a\| \cdot \|b\| \qquad \frac{1}{n} \sum_{i}^{n} |y_{pi} - y_{pa}| \leq \sqrt{\frac{\sum (y_{pi} - y_{ai})^2}{n}}$$

$\therefore$ MAE $\leq$ RMSE

2. Another commonly used loss function is called Max Error (ME), which is the maximum absolute difference between two vectors. Suppose we have two vectors $y \in \mathbb{R}^n$ and $\hat{y} \in \mathbb{R}^n$, then $ME = \max_{i=1,\cdots,n} |\hat{y}_i - y_i|$. Prove ME is always greater than or equal to RMSE.

$$RMSE = \sqrt{\frac{\sum_{i}^{n} (y_{pi} - y_{ai})^2}{n}} \leq \sqrt{\frac{(ME)^2 \times n}{n}} \leq ME$$

Q4 Data Visualization

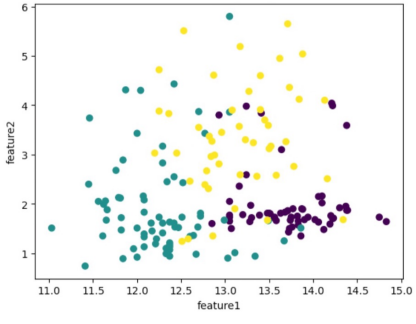
In [7]:
```
import matplotlib.pyplot as plt
from sklearn import datasets
```

In [8]:
```
# Load data
wine = datasets.load_wine()
X = wine.data
Y = wine.target
print(X.shape, Y.shape)
```

```
(178, 13) (178,)
```

In [9]:
```
###### To be filled ######
# Hint: You may use plt.scatter() to draw the plot. You may need to set the 'c' argument
#       in order to have different color for the data points with different classes in Y.
```

In [12]:
```
feature1 = np.array([data[0] for data in X]) # X[:,0]
feature2 = np.array([data[1] for data in X]) #X[:,1]
plt.scatter(feature1, feature2, c=Y)
plt.xlabel('feature1')
plt.ylabel('feature2')
plt.show()
```



Q5 Data Manipulation

In [5]:
```
import numpy as np
```

1. Show the first 2 features of the first 3 data points.

In [6]:
```
two_features_of_three_datapoints = [data[:2] for data in X[:3]]
print(two_features_of_three_datapoints ) # X[:3, :2]
```

```
[array([14.23,  1.71]), array([13.2 ,  1.78]), array([13.16,  2.36])]
```

2. Calculate the mean and the variance of the 1st feature (the 1st column) of array X.

In [13]:
```
###### To be filled ######
# Hint: You may use np.mean() and np.var()
mean = np.mean(feature1 )
mean
```

Out[13]:
```
13.00061797752809
```

In [14]:
```
var = np.var(feature1 )
var
```

Out[14]:
```
0.6553597304633255
```

3. Randomly sample 3 data points (rows) of array X by randomly choosing the row indices. Show the indices and the sampled data points.

In [15]:
```
###### To be filled ######
# Hint: You may use np.random.randint().
indices = np.random.randint(0, len(X), size=3)
print(indices)
```

```
[154  57 118]
```

In [16]:
```
random_data = [X[i] for i in indices] #X[indices]
print(random_data)
```

```
[array([1.258e+01, 1.290e+00, 2.100e+00, 2.000e+01, 1.030e+02, 1.480e+00,
       5.800e-01, 5.300e-01, 1.400e+00, 7.600e+00, 5.800e-01, 1.550e+00,
       6.400e+02]), array([1.329e+01, 1.970e+00, 2.680e+00, 1.680e+01, 1.020e+02, 3.000e+00,
       3.230e+00, 3.100e-01, 1.660e+00, 6.000e+00, 1.070e+00, 2.840e+00,
       1.270e+03]), array([ 12.77,   3.43,   1.98,  16.  ,  80.  ,   1.63,   1.25,   0.43,
         0.83,   3.4 ,   0.7 ,   2.12, 372.  ])]
```

4. Add one more feature (one more column) to the array X after the last feature. The values of the added feature for all data points are constant 1. Show the first data point (first row) of the new array

In [21]:
```
###### To be filled ######
# Hint: You may use np.hstack() and np.ones()

new_col = np.ones((len(X),1))
X = np.hstack((X, new_col))

print(X[0])
```

```
[1.423e+01 1.710e+00 2.430e+00 1.560e+01 1.270e+02 2.800e+00 3.060e+00
 2.800e-01 2.290e+00 5.640e+00 1.040e+00 3.920e+00 1.065e+03 1.000e+00]
```

****Submission Requirement****

Please combine your code, plot and results for Q4, Q5 with your answers for Q1, Q2, Q3 together as a single PDF and submit it through Gradescope.

A easy way is, you can save your completed Jupyter notebook as a PDF (e.g. in Chrome, right click the web page -> Print ... -> Save as PDF) and then merge it with your answers for Q1, Q2, Q3.