HOMEWORK 3

>>Zhiqi Gao<<
>>zgao93 / 9081156037<<

Instructions: Use this latex file as a template to develop your homework. Submit your homework on time as a single pdf file to Canvas. Late submissions may not be accepted. Please wrap your code and upload to a public GitHub repo, then attach the link below the instructions so that we can access it. You can choose any programming language (i.e. python, R, or MATLAB). Please check Piazza for updates about the homework.
Here is a link to my CS760 GitHub repository:
https://github.com/ZhiqiGao-Leo/CS_760_Machine_Learning/blob/main/HW3/homework3.py

1 Questions (50 pts)

1. (9 pts) Explain whether each scenario is a classification or regression problem. And, provide the number of data points (n) and the number of features (p).

   (a) (3 pts) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in predicting CEO salary with given factors.
   Type: Regression
   Number of data points (n): 500
   Number of features (p): 3 (profit, number of employees, industry)

   (b) (3 pts) We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.
   Type: Classification
   Number of data points (n): 20
   Number of features (p): 13 (price, marketing budget, competition price, and ten other variables)

   (c) (3 pts) We are interesting in predicting the % change in the US dollar in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the dollar, the % change in the US market, the % change in the British market, and the % change in the German market.
   Type: Regression
   Number of data points (n): 52 (number of weeks in 2012)
   Number of features (p): 3 (% change in US market, % change in British market, % change in German market)

2. (6 pts) The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

X_1	X_2	X_3	Y
0	3	0	Red
2	0	0	Red
0	1	3	Red
0	1	2	Green
-1	0	1	Green
1	1	1	Red

   Suppose we wish to use this data set to make a prediction for Y when $X_1 = X_2 = X_3 = 0$ using K-nearest neighbors.

(a) (2 pts) Compute the Euclidean distance between each observation and the test point, $X_1 = X_2 = X_3 = 0$.

The Euclidean distances are calculated as follows:

- Observation 1: $d_1 = \sqrt{(0-0)^2 + (3-0)^2 + (0-0)^2} = 3$
- Observation 2: $d_2 = \sqrt{(2-0)^2 + (0-0)^2 + (0-0)^2} = 2$
- Observation 3: $d_3 = \sqrt{(0-0)^2 + (1-0)^2 + (3-0)^2} = \sqrt{10}$
- Observation 4: $d_4 = \sqrt{(0-0)^2 + (1-0)^2 + (2-0)^2} = \sqrt{5}$
- Observation 5: $d_5 = \sqrt{(-1-0)^2 + (0-0)^2 + (1-0)^2} = \sqrt{2}$
- Observation 6: $d_6 = \sqrt{(1-0)^2 + (1-0)^2 + (1-0)^2} = \sqrt{3}$

(b) (2 pts) What is our prediction with $K = 1$? Why?

Since $K = 1$, we select the observation with the smallest distance $\sqrt{2}$, which is Observation 5 (Green).

Prediction: $Y = $ Green

(c) (2 pts) What is our prediction with $K = 3$? Why?

Since $K = 3$, we consider the three observations with the smallest distances:

i. Observation 5 (Green), distance $\sqrt{2}$
ii. Observation 6 (Red), distance $\sqrt{3}$
iii. Observation 2 (Red), distance 2

Among these three, there are 2 Reds and 1 Green. Hence,

Prediction: $Y = $ Red

3. (12 pts) When the number of features p is large, there tends to be a deterioration in the performance of KNN and other local approaches that perform prediction using only observations that are near the test observation for which a prediction must be made. This phenomenon is known as the curse of dimensionality, and it ties into the fact that non-parametric approaches often perform poorly when p is large.

(a) (2pts) Suppose that we have a set of observations, each with measurements on $p = 1$ feature, X. We assume that X is uniformly (evenly) distributed on [0, 1]. Associated with each observation is a response value. Suppose that we wish to predict a test observation's response using only observations that are within 10% of the range of X closest to that test observation. For instance, in order to predict the response for a test observation with $X = 0.6$, we will use observations in the range [0.55, 0.65]. On average, what fraction of the available observations will we use to make the prediction?

Without considering boundaries, we would expect 10% of observations to be available. However, we need to consider the boundary case.

By integration in boundaries, we can get that on average we have 9.75% of observations available.

$$\int_0^{0.05} 100x + 5\,dx + \int_{0.05}^{0.95} 10\,dx + \int_{0.95}^{1} 105 - 100x\,dx = 9 + 0.375 + 0.375 = 9.75\%$$

(b) (2pts) Now suppose that we have a set of observations, each with measurements on $p = 2$ features, $X1$ and $X2$. We assume that predict a test observation's response using only observations that $(X1, X2)$ are uniformly distributed on [0, 1] × [0, 1]. We wish to are within 10% of the range of $X1$ and within 10% of the range of $X2$ closest to that test observation. For instance, in order to predict the response for a test observation with $X1 = 0.6$ and $X2 = 0.35$, we will use observations in the range [0.55, 0.65] for $X1$ and in the range [0.3, 0.4] for $X2$. On average, what fraction of the available observations will we use to make the prediction?

By using part (a), knowing that the 2 features are independent, we can have we have 9.75% for X1 and 9.75% for X2. Which implies that we can have

$$9.75\% \times 9.75\% = 0.0975 \times 0.0975 = 0.00950625 \text{ or } 0.950625\%$$

observations available.

(c) (2pts) Now suppose that we have a set of observations on $p = 100$ features. Again the observations are uniformly distributed on each feature, and again each feature ranges in value from 0 to 1. We wish to predict a test observation's response using observations within the 10% of each feature's range that is closest to that test observation. What fraction of the available observations will we use to make the prediction?

Similarly, we have

$$9.75\%^{100} = 0.0975^{100} = 7.95 \times 10^{-102} \approx 0$$

observations available when we have 100 features.

(d) (3pts) Using your answers to parts (a)–(c), argue that a drawback of KNN when p is large is that there are very few training observations "near" any given test observation.

As seen in Parts 1 through 3, if we want each feature to be close enough to the given point, the fraction of observations approaches 10^{-p}, $\rightarrow 0$ for large p. In general, the volume of a p-dimensional ball with fixed radius R tends to zero as p grows. Hence, with many predictors, the KNN are likely far from the test observation. Consequently, KNN with large p may not yield accurate predictions.

(e) (3pts) Now suppose that we wish to make a prediction for a test observation by creating a p-dimensional hypercube centered around the test observation that contains, on average, 10% of the training observations. For p =1, 2, and 100, what is the length of each side of the hypercube? Comment what happens to the length of the sides as $\lim_{n\to\infty}$.

A p-dimensional hypercube with a volume V has a side length of $s = V^{1/p}$. In our case, we need to solve for $V = 0.1$. Thus for $p = 1$, $s = 0.1$; for $p = 2$, $s = 0.1^{1/2} \approx 0.31622776601$; and when $p = 100$, $s = 0.1^{1/100} \approx 0.97723722095$.

By our observation, we can conclude that for $\lim p \to \infty$, $s \to 1$, which implies that the majority of the volume of a p-dimensional hypercube is located very close to its boundary.

4. (6 pts) Supoose you trained a classifier for a spam detection system. The prediction result on the test set is summarized in the following table.

		Predicted class	
		Spam	not Spam
Actual class	Spam	8	2
	not Spam	16	974

Calculate

(a) (2 pts) Accuracy

$$\begin{aligned}\text{Accuracy} &= \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Population}}\\ &= \frac{8+974}{1000}\\ &= 0.982\end{aligned}$$

(b) (2 pts) Precision

$$\begin{aligned}\text{Precision} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}\\ &= \frac{8}{8+16}\\ &\approx 0.333\end{aligned}$$

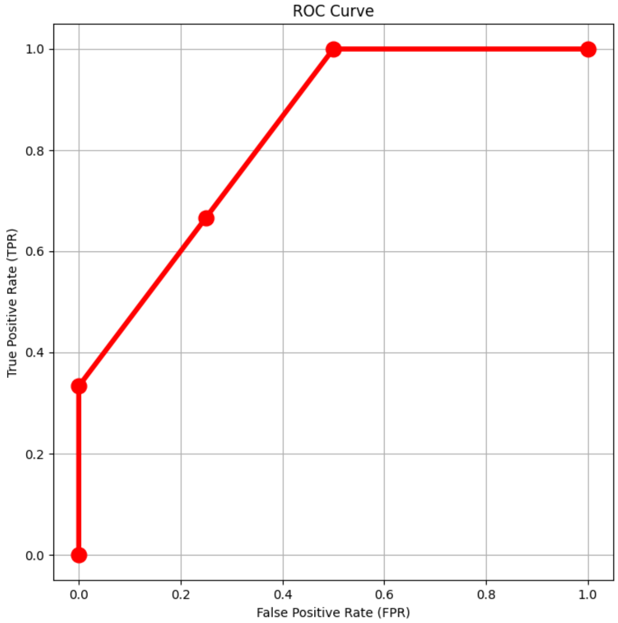
(c) (2 pts) Recall

$$\begin{aligned}\text{Recall} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}\\ &= \frac{8}{8+2}\\ &= \frac{8}{10}\\ &= 0.8\end{aligned}$$

5. (9pts) Again, suppose you trained a classifier for a spam filter. The prediction result on the test set is summarized in the following table. Here, ”+” represents spam, and ”-” means not spam.

Confidence positive	Correct class
0.95	+
0.85	+
0.8	-
0.7	+
0.55	+
0.45	-
0.4	+
0.3	+
0.2	-
0.1	-

(a) (6pts) Draw a ROC curve based on the above table.



(b) (3pts) (Real-world open question) Suppose you want to choose a threshold parameter so that mails with confidence positives above the threshold can be classified as spam. Which value will you choose? Justify your answer based on the ROC curve.

Given the ROC curve, we want to choose a threshold value that is closest to the point (0, 1) while still being above the diagonal line. This is because we want to maximize the true positive rate (TPR) while keeping the false positive rate (FPR) low.

The threshold value that achieves this is approximately 0.25. At this threshold, we achieve a TPR of 0.67 while keeping the FPR at 0.25. This strikes a good balance between minimizing false positives and maximizing true positives.

So, based on the provided ROC curve, we should choose a threshold of approximately 0.25 to classify emails as spam.

6. (8 pts) In this problem, we will walk through a single step of the gradient descent algorithm for logistic regression. As a reminder,

$$\hat{y} = f(x, \theta)$$
$$f(x; \theta) = \sigma(\theta^\top x)$$
$$\text{Cross entropy loss } L(\hat{y}, y) = -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$$
$$\text{The single update step } \theta^{t+1} = \theta^t - \eta \nabla_\theta L(f(x; \theta), y)$$

(a) (4 pts) Compute the first gradient $\nabla_\theta L(f(x; \theta), y)$.

first, we can notice that $\hat{y} = f(x, \theta) = \sigma(\theta^\top x) = \frac{1}{1+e^{-\theta^\top x}}, , 1 - \hat{y} = \frac{e^{-\theta^\top x}}{1+e^{-\theta^\top x}}$. hence we can get

$$\begin{aligned} \frac{\partial y}{\partial \theta} &= \frac{x \cdot e^{-\theta^\top x}}{(1 + e^{-\theta^\top x})^2} = y \cdot (1 - \hat{y}) \cdot x \\ \nabla_\theta L(f(x; \theta), y) &= -y \cdot \frac{1}{\hat{y}} \cdot \frac{\partial y}{\partial \theta} + (1 - y) \cdot \frac{1}{1 - \hat{y}} \cdot \frac{\partial y}{\partial \theta} \\ &= -y(1 - \hat{y})x + (1 - y)\hat{y}x \\ &= x(\hat{y} - y) \end{aligned}$$

(b) (4 pts) Now assume a two dimensional input. After including a bias parameter for the first dimension, we will have $\theta \in \mathbb{R}^3$.

$$\text{Initial parameters : } \theta^0 = [0, 0, 0]$$
$$\text{Learning rate } \eta = 0.1$$
$$\text{data example : } x = [1, 3, 2], y = 1$$

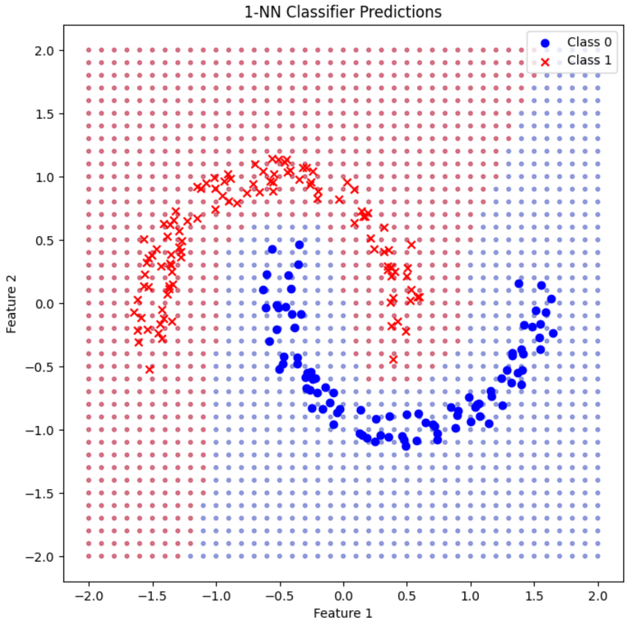
Compute the updated parameter vector θ^1 from the single update step.

first, notice that $\sigma(\theta^\top x) = \frac{1}{1+e^{[0,0,0]^\top \cdot [1,3,2]}} = \frac{1}{1+e^0} = \frac{1}{2}$

$\theta^1 = \theta^0 - \eta \nabla_\theta L(f(x; \theta), y) = [0, 0, 0] - 0.1[1, 3, 2](\frac{1}{2} - 1) = [0, 0, 0] + 0.05[1, 3, 2] = [0.05, 0.15, 0.10]$

2 Programming (50 pts)

1. (10 pts) Use the whole D2z.txt as training set. Use Euclidean distance (i.e. $A = I$). Visualize the predictions of 1NN on a 2D grid $[-2 : 0.1 : 2]^2$. That is, you should produce test points whose first feature goes over $-2, -1.9, -1.8, \ldots, 1.9, 2$, so does the second feature independent of the first feature. You should overlay the training set in the plot, just make sure we can tell which points are training, which are grid.



- Task: spam detection
- The number of rows: 5000
- The number of features: 3000 (Word frequency in each email)
- The label (y) column name: 'Predictor'
- For a single training/test set split, use Email 1-4000 as the training set, Email 4001-5000 as the test set.
- For 5-fold cross validation, split dataset in the following way.
  - Fold 1, test set: Email 1-1000, training set: the rest (Email 1001-5000)
  - Fold 2, test set: Email 1000-2000, training set: the rest
  - Fold 3, test set: Email 2000-3000, training set: the rest
  - Fold 4, test set: Email 3000-4000, training set: the rest
  - Fold 5, test set: Email 4000-5000, training set: the rest

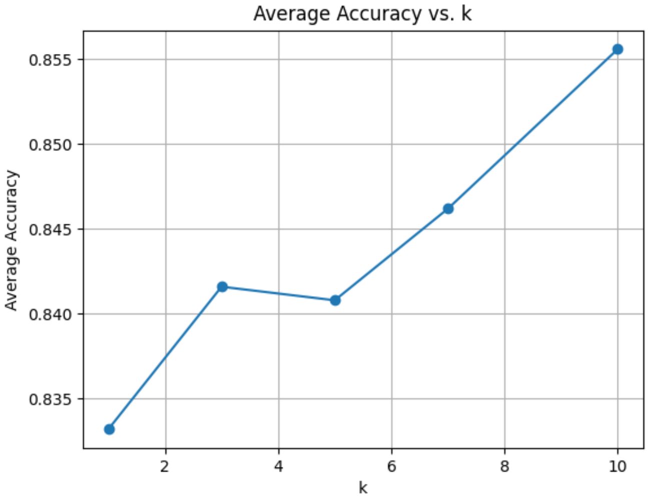
2. (8 pts) Implement 1NN, Run 5-fold cross validation. Report accuracy, precision, and recall in each fold.

   Fold 1: Accuracy: 0.8250 Precision: 0.6545 Recall: 0.8175
   Fold 2: Accuracy: 0.8530 Precision: 0.6857 Recall: 0.8664
   Fold 3: Accuracy: 0.8620 Precision: 0.7212 Recall: 0.8380
   Fold 4: Accuracy: 0.8510 Precision: 0.7164 Recall: 0.8163
   Fold 5: Accuracy: 0.7750 Precision: 0.6057 Recall: 0.7582

3. (12 pts) Implement logistic regression (from scratch). Use gradient descent (refer to question 6 from part 1) to find the optimal parameters. You may need to tune your learning rate to find a good optimum. Run 5-fold cross validation. Report accuracy, precision, and recall in each fold.

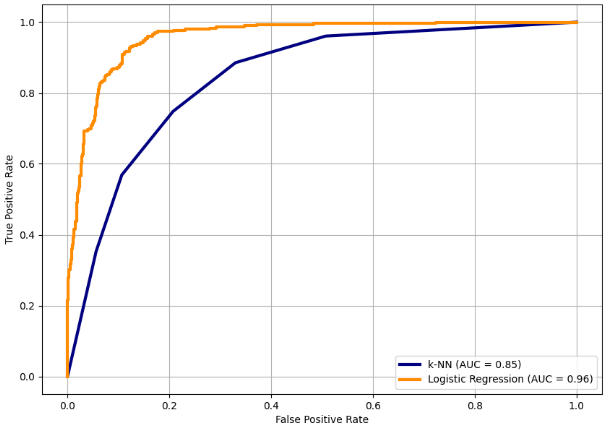
   Fold 1: Accuracy: 0.9260 Precision: 0.9073 Recall: 0.8246
   Fold 2: Accuracy: 0.9250 Precision: 0.8976 Recall: 0.8231
   Fold 3: Accuracy: 0.9130 Precision: 0.9227 Recall: 0.7570
   Fold 4: Accuracy: 0.9030 Precision: 0.8803 Recall: 0.7755
   Fold 5: Accuracy: 0.8880 Precision: 0.8566 Recall: 0.7614

4. (10 pts) Run 5-fold cross validation with kNN varying k (k=1, 3, 5, 7, 10). Plot the average accuracy versus k, and list the average accuracy of each case.



k = 1, Average Accuracy = 0.8332
k = 3, Average Accuracy = 0.8416
k = 5, Average Accuracy = 0.8408
k = 7, Average Accuracy = 0.8462
k = 10, Average Accuracy = 0.8556

5. (10 pts) Use a single training/test setting. Train kNN (k=5) and logistic regression on the training set, and draw ROC curves based on the test set.


True Positive Rate
False Positive Rate
k-NN (AUC = 0.85)
Logistic Regression (AUC = 0.96)