HOMEWORK 3

>>John Hawthorne<<
>>908 495 9692<<

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.

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.

   This is regression since our model must predict outputs on a continuous space. We have n = 500 and p = 4.

   (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.

   This is a classification problem since our model must predict a discrete output on the set {success, failure}. We have n = 20 and p = 13.

   (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.

   This is a regression problem since our model must predict a continuous spectrum output. We have n = 52 and p = 3.

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, $X_1 = X_2 = X_3 = 0$.

   We have distance from our initial point (0,0,0) as d = { 3, 2, $\sqrt{10}$, $\sqrt{5}$, $\sqrt{2}$, $\sqrt{3}$ }

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

   Our prediction is Green, because the nearest neighbor is { -1, 0, 1 }

(c) (2 pts) What is our prediction with $K = 3$? Why?
For K=3, our prediction is Red since 2/3 of the nearest neighbors our 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?
We will use 1/10 or the observations.

(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?
We will use $(1/10)^2$

(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?
We will use $(1/10)^{100}$

(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.
For large p we have an exponentially decreasing size of the observation space to predict on. Thus this would require and exponetially increasing amount of data to maintain consistent availability of nearest neighbors within a set interval. We could try to expand our interval, but this would lead to greater error and more importantly it would still not be sufficient since our near-neighbor sample space decreases exponentially (as a proportion of the total sample space).

(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 on your answer.
For p = 1, we have $d_i = 0.10$. For p = 2, we have $d_i = 10^{-1/2}$. For p = 100, we have $d_i = 10^{-1/100}$. As is consistent with part (d.), our required component interval d_i approaches 1 as p becomes large. That is, because of dimensionality, we require an individual component sample of near to the whole data set, which is computationally intractable.

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 = 982/1000.

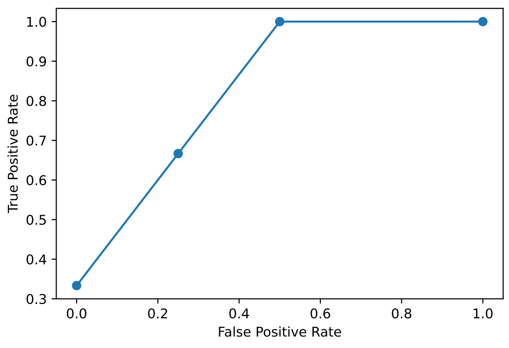
(b) (2 pts) Precision = 8/24

(c) (2 pts) Recall = 8/10

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.



This is the resulting figure.

(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.

I would want the confidence positive threshold for spam to be very high because I think it is worse if an important email isn't seen because it is marked as spam than if a junk email is let through into someones main inbox.

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

$$f(x;\theta) = \sigma(\theta^\top x)$$

$$\text{Cross entropy loss } L(\hat{y}, y) = -[y \log \hat{y} + (1-y)\log(1-\hat{y})]$$

$$\text{The single update step } \theta^{t+1} = \theta^t - \eta \nabla_\theta L(f(x;\theta), y)$$

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

We simply use the chain rule and the fact that:

$$\frac{\partial \sigma}{\partial \theta_i} = \frac{-x_i e^{-\sum_i \theta_i x_i}}{(1 - e^{-\sum_i \theta_i x_i})^2}$$

Skipping some basic steps we have:

$$\frac{\partial L}{\partial \theta_i} = x_i e^{-\sum_i \theta_i x_i} \frac{\sigma}{1-\sigma}(y - \sigma)$$

which simplifies to:

$$\frac{\partial L}{\partial \theta_i} = (\sigma - y)x_i$$

which is:

$$\nabla_\theta L = (\sigma - y)\vec{x}$$

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

$$\text{Initial parameters : } \theta^0 = [0, 0, 0]$$

$$\text{Learning rate } \eta = 0.1$$

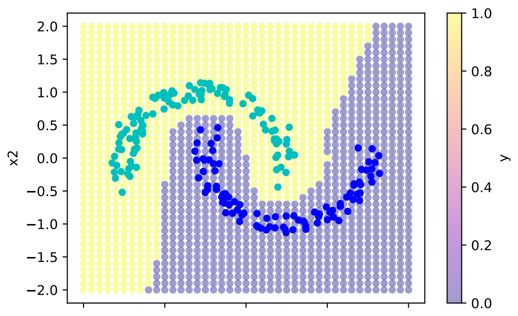
$$\text{data example : } x = [1, 3, 2], y = 1$$

Compute the updated parameter vector θ^1 from the single update step.
We just apply the gradient update to get:

$$\theta^1 = \begin{pmatrix} .05 \\ .10 \\ .15 \end{pmatrix}$$

2 Programming (50 pts)

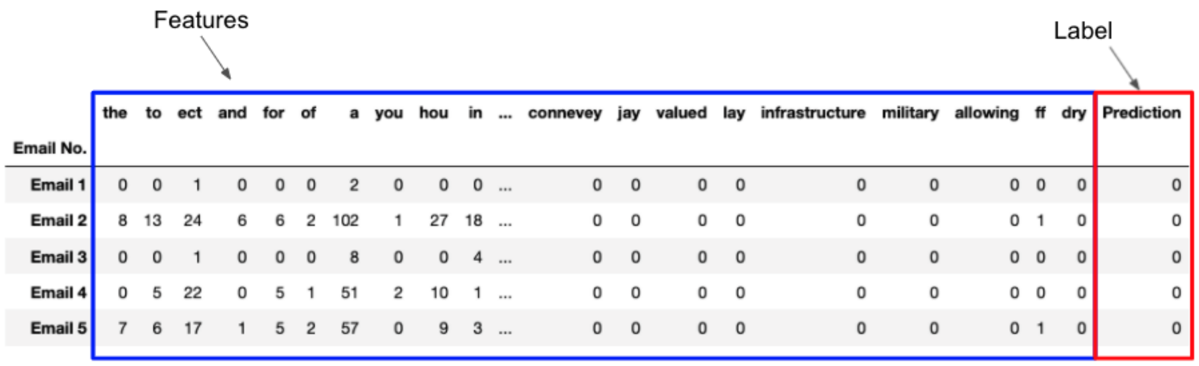
1. (10 pts) Use the whole D2z.txt as training set. Use Euclidean distance (i.e. $A = I$). Visualize the predictions of 1NN on a 2D grid $[-2 : 0.1 : 2]^2$. That is, you should produce test points whose first feature goes over $-2, -1.9, -1.8, \ldots, 1.9, 2$, so does the second feature independent of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



This is the resulting figure above.

Spam filter Now, we will use 'emails.csv' as our dataset. The description is as follows.

- 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


Features
Label

Email No.	the	to	ect	and	for	of	a	you	hou	in	...	connevey	jay	valued	lay	infrastructure	military	allowing	ff	dry	Prediction
Email 1	0	0	1	0	0	0	2	0	0	0	...	0	0	0	0	0	0	0	0	0	0
Email 2	8	13	24	6	6	2	102	1	27	18	...	0	0	0	0	0	0	0	1	0	0
Email 3	0	0	1	0	0	0	8	0	0	4	...	0	0	0	0	0	0	0	0	0	0
Email 4	0	5	22	0	5	1	51	2	10	1	...	0	0	0	0	0	0	0	0	0	0
Email 5	7	6	17	1	5	2	57	0	9	3	...	0	0	0	0	0	0	0	1	0	0

- 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

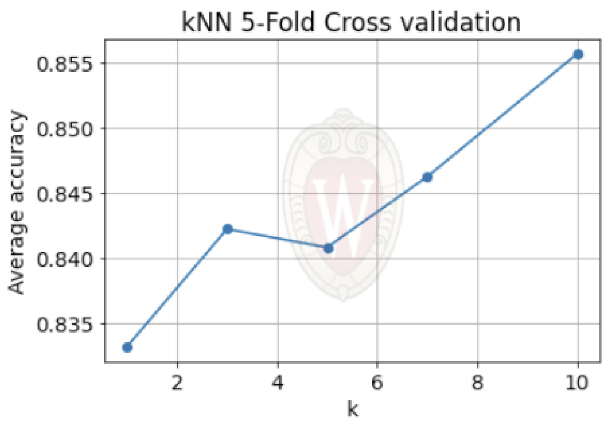
2. (8 pts) Implement 1NN, Run 5-fold cross validation. Report accuracy, precision, and recall in each fold.

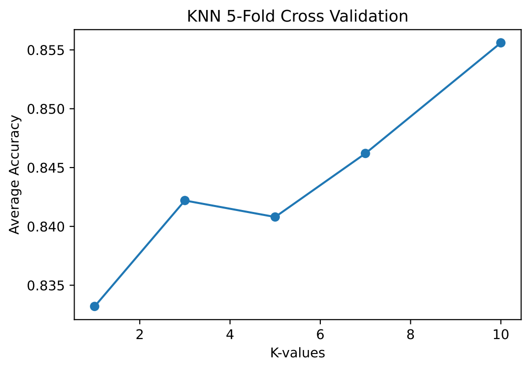
   Fold 1: Accuracy = 0.825, Precision = 0.6544943820224719, Recall = 0.8175438596491228
   Fold 2: Accuracy = 0.853, Precision = 0.6857142857142857, Recall = 0.8664259927797834
   Fold 3: Accuracy = 0.862 Precision = 0.7212121212121212, Recall = 0.8380281690140845
   Fold 4: Accuracy = 0.851, Precision = 0.7164179104477612, Recall = 0.8163265306122449
   Fold 5: Accuracy = 0.775, Precision = 0.6057441253263708, Recall = 0.7581699346405228

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.

   Learning Rate = 0.1, 10 iterations through training data
   Fold 1: Accuracy = 0.915, Precision = 0.9672897196261683 Recall = 0.7263157894736842
   Fold 2: Accuracy = 0.909, Precision = 0.8940677966101694 Recall = 0.7617328519855595
   Fold 3: Accuracy = 0.898, Precision = 0.9136363636363637 Recall = 0.7077464788732394
   Fold 4: Accuracy = 0.837, Precision = 0.9781021897810219 Recall = 0.4557823129251701
   Fold 5: Accuracy = 0.873, Precision = 0.7737003058103975 Recall = 0.826797385620915

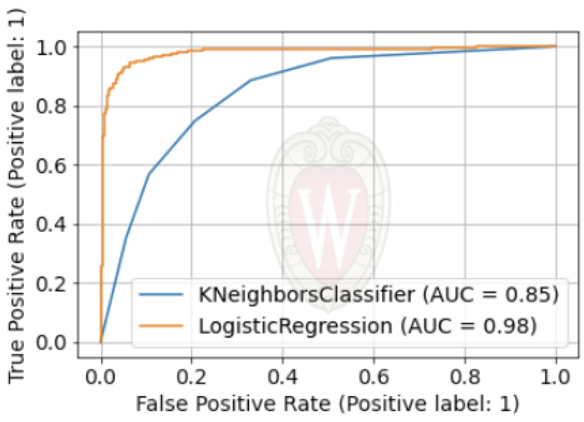
4. (10 pts) Run 5-fold cross validation with kNN varying k (k=1, 3, 5, 7, 10). Plot the average accuracy versus k, and list the average accuracy of each case.
   Expected figure looks like this.

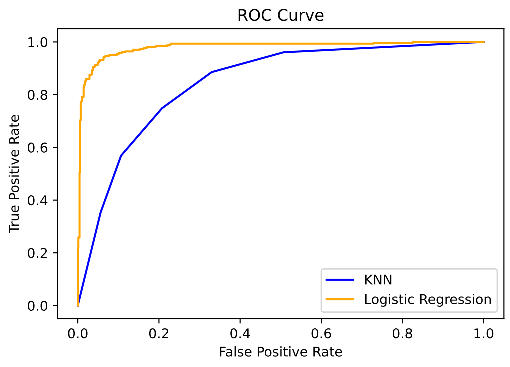




This is the resulting figure above.

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.
   Expected figure looks like this. Note that the logistic regression results may differ.





The resulting figure.