Stony Brook University
CSE512 – Machine Learning – Spring 21
Homework 3, Due: 27 Mar 2021 at midnight 23:59

This homework contains 2 questions. The second question requires programming. The maximum number of points is 100.

1 Question 1 – Nearest Neighbor Classifiers (40 points)

1.1 1-NN with asymmetric loss (20 points)

Suppose we want to build a binary classifier, but the cost of false positive (predicting positive for a negative case) is much higher than the cost of false negative (predicting negative for a positive case). One can consider an asymmetric loss function, where the cost of false negative is 1, while the cost of false positive is $\alpha > 1$. For a point x, let $\eta(x)$ be the probability that x is positive.

1.1.1 (3 points)

Show that the optimal Bayes risk for data point x is $r^*(x) = \min\{\eta(x), \alpha(1-\eta(x))\}$.

1.1.2 (4 points)

Let $r(x)$ be the asymptotic risk for data point x, express $r(x)$ in terms of α and $\eta(x)$.

1.1.3 (10 points)

Prove that $r(x) \leq (1+\alpha)r^*(x)(1-r^*(x))$. Hint: use $\alpha > 1$

1.1.4 (3 points)

Let R be the asymptotic risk of the 1-NN classifier and R^* be Bayes risk. Prove that: $R \leq (1+\alpha)R^*(1-R^*)$

1.2 k-NN classifier (20 points)

Consider a k-NN classifier: classify a point as positive if at least (k+1)/2 nearest neighbors are positive.

1.2.1 (3 points)

Consider drawing k points randomly from a Bernoulli distribution with two outcomes: positive or negative, and the probability of the point being positive is η. Let $g(\eta, k)$ be the probability that at least $(k+1)/2$ out of k points are positive. Express the asymptotic risk $r(x)$ for a point x in terms of $\eta(x)$ and the function $g(\cdot, \cdot)$.

1.2.2 (10 points)

Prove that $r(x) = r^*(x) + (1 - 2r^*(x))g(r^*(x), k)$

1.2.3 (3 points)

Using Hoeffding's Inequality (`https://en.wikipedia.org/wiki/Hoeffding_inequality`), prove that:

$$g(r^*(x), k) \leq \exp(-2(0.5 - r^*(x))^2 k) \tag{1}$$

1.2.4 (4 points)

Prove that: $r(x) \leq r^*(x) + \frac{1}{\sqrt{2k}}$. Hint: you should use the above inequality Eq. (1). Note that: from this result, you can see that the Asymptotic risk of k-NN classifier is the Bayes Risk if k goes to infinity.

2 Question 2 – Implementation (60 points)

2.1 Implementation (20 points)

Implement a k-Nearest Neighbor classifier. Write a Python function with the signature:

$$\hat{\mathbf{y}}, Idxs = knn\_classifier(\mathbf{X}_{train}, \mathbf{y}_{train}, \mathbf{X}_{test}, k)$$

where
Inputs:

- $\mathbf{X}_{train}$: a two dimensional Numpy array of size $n \times d$, where n is the number of training data points, and d the dimension of the feature vectors.
- $\mathbf{y}_{train}$: a Numpy vector of length n. $\mathbf{y}[i]$ is a categorical label corresponding to the data point $\mathbf{X}[i, :]$, $\mathbf{y}[i] \in \{0, 1, \ldots, k-1\}$. You can assume that the number of classes k is the maximum entry of $\mathbf{y}$ plus 1.
- $\mathbf{X}_{test}$: a two dimensional Numpy array of size $m \times d$, where m is the number of test data points, and d the dimension of the feature vectors.
- k: the number of nearest neighbors to use for classification

Outputs:

- $\hat{\mathbf{y}}$: a Numpy vector of length m for the predicted labels for the test data
- $Idxs$: a Numpy array of size $m \times k$ for the indexes of the k nearest neighbor data points. Each row corresponds to a test data point.

Do not use/import $sklearn.neighbors.KNeighborsClassifier$
Hint: Use Euclidean distance to find the k nearest neighbors. You can use $scipy.spatial.distance.cdist$.

2.2 Experiment and result reporting (40 points)

In this section, you will run the kNN classifier on a subset of the MNIST dataset. MNIST is a dataset of images of handwritten digits (`http://yann.lecun.com/exdb/mnist/`) that is widely used in the field of machine learning. It includes digits from 0 to 9.

Step 1

The first step is to load your training and test data that are provided in the files $mnist\_train.csv$ and $mnist\_test.csv$ correspondingly. (Tip: To load the data from a csv file, you can use numpy.genfromtxt or csv.reader).

Inspect your data carefully: The first row of each csv file is the csv header. Each following row corresponds to a sample. The first column corresponds to the label of each sample, which is a number from 0 to 9. The rest 784 columns correspond to the features of the samples. Essentially, these 784 features are the pixel values of the corresponding original image of size 28×28.

Load your data to numpy arrays $\mathbf{X}_{train}, \mathbf{y}_{train}, \mathbf{X}_{test}, \mathbf{y}_{test}$, as integers. The feature values are in the range [0, 255]. Normalize the features of both the training and test sets by dividing them by 255, in order to convert them to float values in the range [0, 1] (e.g. $\mathbf{X}_{train} = \mathbf{X}_{train}/255.$).

Step 2

Visualize some of the samples to better understand your dataset. You can use the following code:

```
import numpy as np
import matplotlib.pyplot as plt

fig = plt.figure()
for i in range(9):
    plt.subplot(3, 3, i + 1)
    plt.imshow(np.reshape(X_train[i], (28, 28)))
    plt.title(y_train[i])
    plt.axis("off")
plt.tight_layout()
plt.show()
```

2.2.1 Question 2.2.1 (10 points)

Run the kNN classifier that you implemented in Question 2.1. Run it for different values of $k = 2 * m + 1$, where $m = 0, 1, 2, 4, 8, 16, 32$. Plot the accuracy on the test data against k. Does k affect the performance of the classifier? (You can use sklearn.metrics.accuracy_score.)

2.2.2 Question 2.2.2 (10 points)

Run the kNN classifier again for $k = 3$, but now using smaller subsets for your training set. Keep the first n rows of $\mathbf{X}_{train}, \mathbf{y}_{train}$. Try $n = 100, 200, 400, 600, 800, 1000$. Plot the accuracy on the test data against n. Does the number of training data affect the performance of the classifier?

2.2.3 Question 2.2.3 (10 points)

Now try different distance metrics. Run the kNN classifier again for $k = 3$, using the whole training set. Use "Manhattan" distance and report the accuracy on the test data. Is the result better than using the Euclidean distance?

2.2.4 Question 2.2.4 (10 points)

Run the kNN classifier again for $k = 5$, using the whole training set and Euclidean distance. Display the 5 nearest neighbors of 3 failed cases: Choose 3 test samples that are wrongly classified. For each of them, get their 5 nearest neighbors. Display each sample and its neighbors using plt.imshow, similarly to Step 2.

3 What to submit

3.1 Blackboard submission

You will need to submit both your code and your answers to questions on Blackboard. Put the answer file and your code in a folder named: SUBID_FirstName_LastName (e.g., *10947XXXX_lionel_messi*). Zip this folder and submit the zip file on Blackboard. Your submission must be a zip file, i.e, SUBID_FirstName_LastName.zip.

The answer file should be named: hw3-answers.pdf. You can use Latex if you wish, but it is not compulsory. The first page of the hw3-answers.pdf should be the filled cover page at the end of this homework. The remaining of the answer file should contain:

1. Answers to Question 1
2. Answers to Question 2.2

Your Python code must be named hw3.py. It should contain the requested function for Question 2.1, as well as the code used for Questions 2.2. For automated testing, it should be possible for us to import your

functions using 'from hw3 import ...'. You can submit other python/data files if necessary. Your code hw3.py can also include other functions if necessary.

Make sure you follow the instructions carefully. Your will lose points if:

1. You do not attach the cover page. Your answer file is not named hw3-answers.pdf
2. Your functions do not have the exact signatures as instructed
3. Your functions cannot be imported for automatic testing
4. Your functions crash or fail to run

4 Cheating warnings

Don't cheat. You must do the homework yourself, otherwise you won't learn. You cannot ask and discuss with students from previous years. You cannot look up the solution online.

Cover page for answers.pdf
CSE512 Spring 2021 - Machine Learning - Homework 3

Your Name:

Solar ID:

NetID email address:

Names of people whom you discussed the homework with: