CS 5727: Homework #1

Due on Wednesday, September 13, 2017

Zhan Zhang

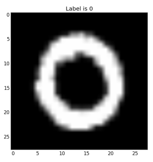
zz524@cornell.edu

Jialiang Wang

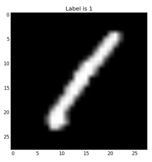
jw2476@cornell.edu

September 13, 2017

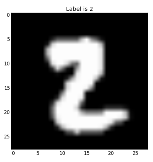
Contents



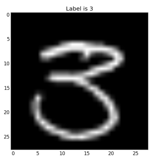
(a) 0



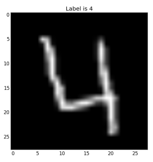
(b) 1



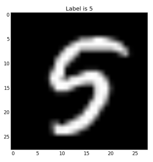
(c) 2



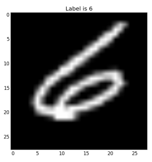
(d) 3



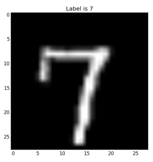
(e) 4



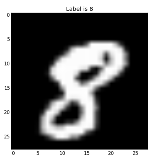
(f) 5



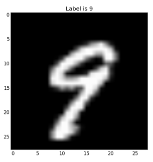
(g) 6



(h) 7



(i) 8



(j) 9

Figure 1: Images of the digits

PROGRAMMING EXERCISES

Problem 1

(b) The function to display the digits is shown in listing 1 and the digits are shown in figure 1.

Listing 1: Python Script

```python
def display_digit(digit_sample):
    with open('train.csv', 'r') as csv_file:
        reader = csv.reader(csv_file)
        next(reader, None)

        count = 0
        for data in reader:
            label = int(data[0])
            pixels = np.array(data[1:], dtype='uint8')

            if(label in digit_sample.keys()):
                count += 1
                continue
            else:
                digit_sample[label] = [pixels, count]
                pixels = pixels.reshape((28, 28))
                plt.title('Label is {label}'.format(label=label))
                plt.imshow(pixels, cmap='gray')
                plt.savefig(str(label) + '.png')
            count += 1
            if len(digit_sample) == 10:
                break
```

(c) The prior probability of the digits are approximately uniform and the histogram (shown in figure 2) is even.

Problem 1 continued on next page. . .

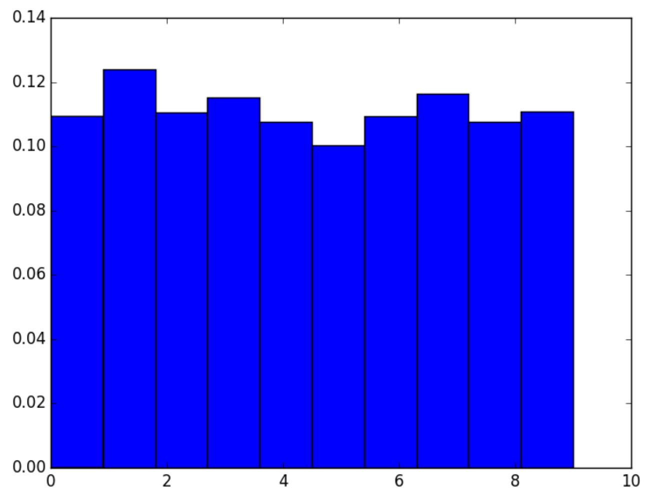
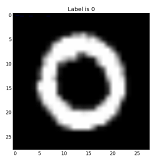
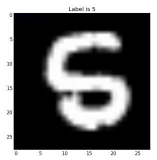


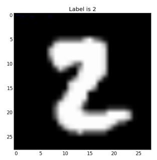
Figure 2: Normalized histogram of digit counts



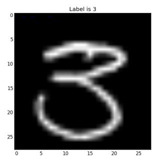
(a) Best Match with 0



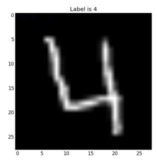
(b) Best Match with 1*



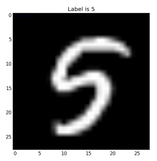
(c) Best Match with 2



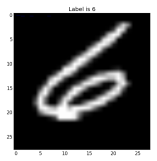
(d) Best Match with 3



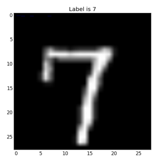
(e) Best Match with 4



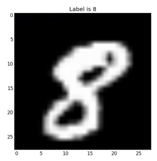
(f) Best Match with 5



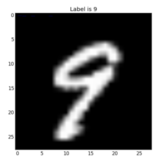
(g) Best Match with 6



(h) Best Match with 7



(i) Best Match with 8



(j) Best Match with 9

Figure 3: Images of the Matches

(d) The best matches are shown in figure 3.
(e) The histogram of the genuine and impostor distances on the same set of axes is shown in figure 4.

(f) The ROC curve is shown in figure 5. The equal error rate is 85%

(h) The average accuracy of 3 fold cross-validation is 0.972095238095.

(i) The confusion matrix is shown below and digit '5' is the most tricky to classify.

$$\begin{array}{cccccccccc}
1.364e{+}03 & 1.000e{+}00 & 1.000e{+}00 & 0.000e{+}00 & 0.000e{+}00 & 1.000e{+}00 & 4.000e{+}00 & 0.000e{+}00 & 0.000e{+}00 & 0.000e{+}00 \\
0.000e{+}00 & 1.560e{+}03 & 2.000e{+}00 & 4.000e{+}00 & 1.000e{+}00 & 1.000e{+}00 & 3.000e{+}00 & 1.000e{+}00 & 1.000e{+}00 & 2.000e{+}00 \\
9.000e{+}00 & 6.000e{+}00 & 1.381e{+}03 & 4.000e{+}00 & 2.000e{+}00 & 0.000e{+}00 & 0.000e{+}00 & 1.300e{+}01 & 9.000e{+}00 & 3.000e{+}00 \\
3.000e{+}00 & 2.000e{+}00 & 6.000e{+}00 & 1.361e{+}03 & 0.000e{+}00 & 9.000e{+}00 & 0.000e{+}00 & 3.000e{+}00 & 1.900e{+}01 & 7.000e{+}00 \\
1.000e{+}00 & 1.100e{+}01 & 0.000e{+}00 & 0.000e{+}00 & 1.308e{+}03 & 0.000e{+}00 & 5.000e{+}00 & 0.000e{+}00 & 0.000e{+}00 & 4.300e{+}01 \\
4.000e{+}00 & 0.000e{+}00 & 1.000e{+}00 & 1.800e{+}01 & 1.000e{+}00 & 1.225e{+}03 & 1.200e{+}01 & 0.000e{+}00 & 8.000e{+}00 & 7.000e{+}00 \\
9.000e{+}00 & 1.000e{+}00 & 0.000e{+}00 & 0.000e{+}00 & 1.000e{+}00 & 4.000e{+}00 & 1.383e{+}03 & 0.000e{+}00 & 1.000e{+}00 & 0.000e{+}00 \\
2.000e{+}00 & 1.100e{+}01 & 6.000e{+}00 & 0.000e{+}00 & 8.000e{+}00 & 0.000e{+}00 & 0.000e{+}00 & 1.406e{+}03 & 1.000e{+}00 & 3.000e{+}01 \\
4.000e{+}00 & 8.000e{+}00 & 5.000e{+}00 & 1.200e{+}01 & 1.000e{+}00 & 8.000e{+}00 & 7.000e{+}00 & 0.000e{+}00 & 1.288e{+}03 & 1.000e{+}01 \\
5.000e{+}00 & 2.000e{+}00 & 0.000e{+}00 & 6.000e{+}00 & 5.000e{+}00 & 2.000e{+}00 & 2.000e{+}00 & 9.000e{+}00 & 6.000e{+}00 & 1.330e{+}03
\end{array} \tag{1}$$

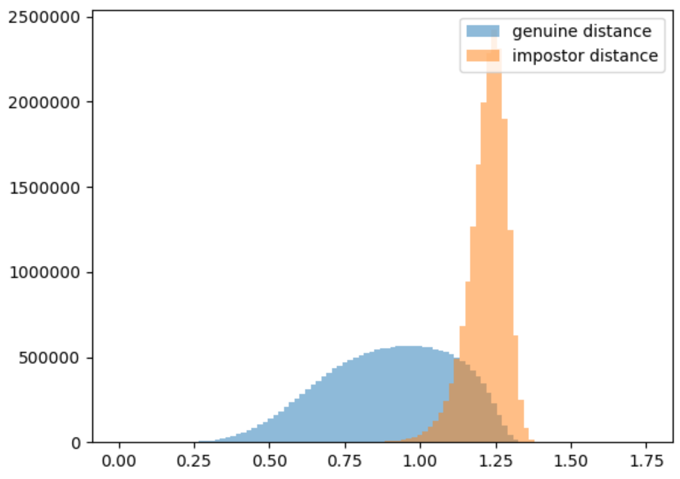


Figure 4: Histograms of the genuine and impostor distances

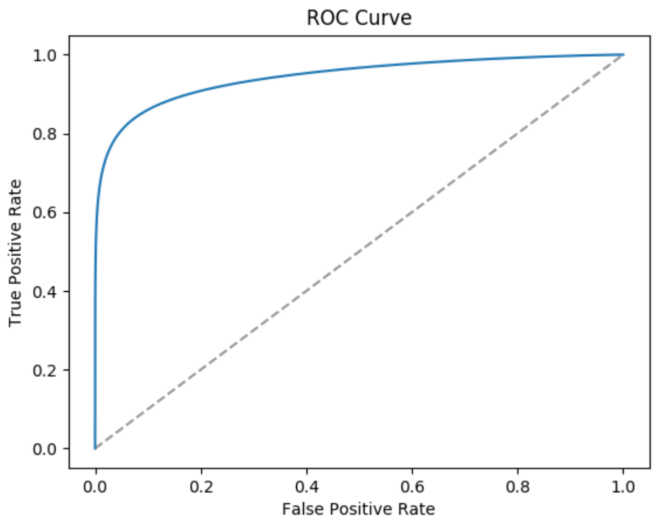
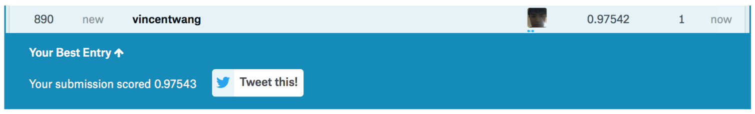


Figure 5: ROC curve

Problem 1 continued on next page. . .

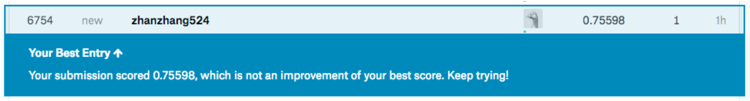


890 new vincentwang 0.97542 1 now

Your Best Entry

Your submission scored 0.97543

Figure 6: Kaggle MNIST Submission Record



6754 new zhanzhang524 0.75598 1 1h

Your Best Entry

Your submission scored 0.75598, which is not an improvement of your best score. Keep trying!

Figure 7: Kaggle Titanic Submission Record

(j) Submitted in Kaggle MNIST competition. Figure 6 shows the submission record.

Problem 2

(b) 'Name','Ticket','Cabin' are ignored for logistic regression.

- "Name" and "Ticket" fields contain non-categorical strings which have little meaning in classification.
- "Cabin" field is significantly lost for a majority of passengers thus removed from the attributes.

(c) The outcome has been submitted to kaggle, shown in figure 7.

WRITTEN EXERCISES

Problem 3

$$\begin{aligned} var[X-Y] &= E[(X-Y)^2] - E[(X-Y)]^2 \\ &= E[X^2+Y^2-2XY] - E[(X-Y)]^2 \\ &= E[X^2]+E[Y^2]-2E[XY]-E[X]^2-E[Y]^2+2E[X]E[Y] \\ &= var[X]+var[Y]-2(E[XY]-E[X]E[Y]) \\ &= var[X]+var[Y]-2cov[X,Y] \end{aligned} \tag{2}$$

Problem 4

(a) Define case "A widget is defective" as A, "A widget is detected as defective" as B:

$$\begin{aligned} P(A|B) &= \frac{P(A)P(B|A)}{P(B)} \\ &= \frac{\frac{1}{100000}\times 0.95}{\frac{1}{100000}*0.95+(1-\frac{1}{100000})\times(1-0.95)} \\ &= 1.8997\times 10^{-4} \end{aligned} \tag{3}$$

(b)

- Probability of a product to be good when detected as defective:

$$\begin{aligned} P(\ A|B) &= \frac{P(\ A)P(B|\ A)}{P(B)} \\ &= \frac{(1-\frac{1}{100000})(1-0.95)}{(1-0.95)*(1-\frac{1}{100000})+0.95*\frac{1}{100000}} \\ &= 0.9998 \end{aligned} \tag{4}$$

Good product to be thrown:

$$1 \times 10^7 \times P(B) \times P(\ A|B) = 499995 \tag{5}$$

- Probability of a product to be defective when detected as good:

$$\begin{aligned} P(A|\ B) &= \frac{P(A)P(\ B|A)}{P(\ B)} \\ &= \frac{\frac{1}{100000}\times(1-0.95)}{(1-\frac{1}{100000})\times 0.95+\frac{1}{100000}\times(1-0.95)} \\ &= 5.2632 \times 10^7 \end{aligned} \tag{6}$$

Defective product to be shipped:

$$1 \times 10^7 \times P(\ B) \times P(A|\ B) = 5 \tag{7}$$

Problem 5

(a) As the classifier is tested within the training set, when $k = n$, all samples are included in the system thus the error rate is 0.5.
When k comes to 1, it would get vote from itself and thus has a error rate of 0.
The value would start to increase when k increases from 1 to n, but may have fluctuation in between.

(b) Firstly, we assume that samples are uniformly removed from the dataset.
The error rate would be approximately 0.5 when k = n/2.
By decreasing k from n/2, the error rate would decrease until its local minimum. It would then go up until k = 1.
At k = 1, the error rate would be approaching 0.5 but not exceeding.
Quick sketch to show the trend of error rate is shown as figure 8.
However, when samples are randomly removed from the dataset, there would be more fluctuation and the trend is hard to tell.

(c) For small fold numbers, like 2 or 3, we are only allowed to use a small part of dataset to train our classifier which would cause inaccuracy.
When more folds are added, more computational resource would be needed resulting in longer training time.
To reach a balanced performance, we would generally choose 5 to 10 folds for cross validation

(d) To enhance the KNN method, weight is introduced to each neighbor based on the distance between the neighbors and the point.
Weight could be invertly proportional to the distance or the square of distances: $W \propto \frac{1}{D}$ or $W \propto \frac{1}{D^2}$
The first weight would improve the model while the second order weight would further reduce the impact

Problem 5 continued on next page...

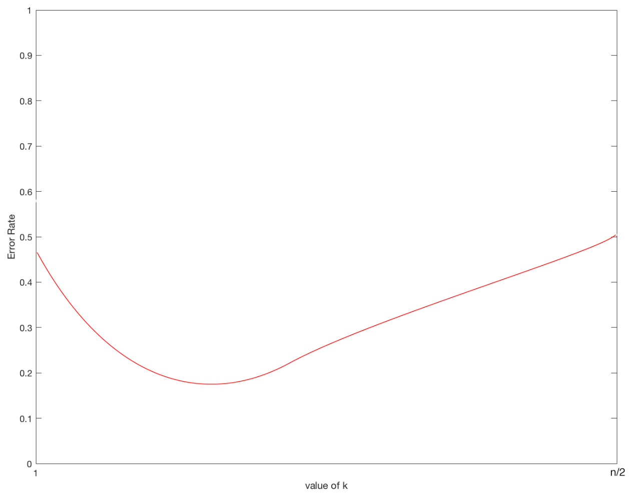


Figure 8: Error Rate as K Changes

from the outliers.

(e) For high dimensional data, effect of the curse of dimensionity appears. The euclidean distances would have little meaning when the distances between vectors are quite similiar.
At the same time, kNN would require exponential computational resource while having a non-satisfactory outcome.