University of Helsinki, Department of Computer Science

Assignment 2 of Introduction to Machine Learning

Gao Yuan No.014242582

November 11, 2013

1 Problem 1

Answer:
As the logarithmic model is proper, we use logarithmic cost function for evaluation. We

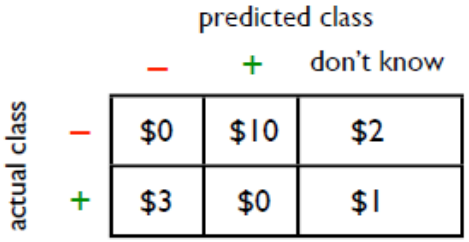

Figure 1.1: Confusion matrix of KNN classifier.

assume the probability of assigning to first class is p then the probability of assigning second class is $1 - p$.
The expected cost would be

$$E\{'-'\} = 0(1-\alpha) + 3\alpha \tag{1.1}$$

$$E\{'+'\} = 10(1-\alpha) + 0\alpha \tag{1.2}$$

$$E\{'don'tknow'\} = 2(1-\alpha) + 1\alpha \tag{1.3}$$

$$E\{'-'\} = E\{'+'\} \tag{1.4}$$
$$0(1-\alpha) + 3\alpha = 10(1-\alpha) + 0\alpha \tag{1.5}$$
$$\alpha_1 = \frac{10}{13} \tag{1.6}$$

$$E\{'-'\} = E\{'don'tknow'\} \tag{1.7}$$
$$0(1-\alpha) + 3\alpha = 2(1-\alpha) + 1\alpha \tag{1.8}$$
$$\alpha_2 = \frac{1}{2} \tag{1.9}$$

$$E\{'+'\} = E\{'don'tknow'\} \tag{1.10}$$
$$10(1-\alpha) + 0\alpha = 2(1-\alpha) + 1\alpha \tag{1.11}$$
$$\alpha_3 = \frac{8}{9} \tag{1.12}$$

The best answer for '-','+' and 'don't know' are that:

- When α is smaller or equal to $\frac{1}{2}$, we predict '-'.
- When α is bigger than $\frac{1}{2}$ and α is smaller or equal to $\frac{8}{9}$, we predict 'don't know'.
- When α is bigger than $\frac{8}{9}$, we predict '+'.

2 Problem 2

Answer:

The expected cost is:

$$E\{C\} = E\{1\} + E\{0\} \tag{2.1}$$
$$= (b-1)^2 a + b^2(1-a) \tag{2.2}$$
$$= b^2 a - 2ba + a + b^2 - b^2 a \tag{2.3}$$
$$= -2ba + a + b^2 \tag{2.4}$$

First order derivative is :

$$\frac{\partial E\{C\}}{\partial b} = \frac{\partial(-2ba + a + b^2)}{\partial b} \tag{2.5}$$
$$= -2a + 2b \tag{2.6}$$

Second order derivative is :

$$\frac{\partial(-2a+2b)}{\partial b} = 2 \tag{2.7}$$

$$\geq 0 \tag{2.8}$$

As a consequence, $E\{C\}$ is minimized, when $-2a+2b=0$ (i.e. when $a=b$).

3 Problem 3

Proximity is typically defined between a pair of objects.

For class $P(y=0)=0.2$, there are $2\times 3=6$ unit areas. For class $P(y=1)=0.7$, there are $4\times 4=16$ unit areas. For class $P(y=2)=0.1$, there are $3\times 3=9$ unit areas. The proportionality of possibility of these three areas are $\frac{0.2}{6}, \frac{0.7}{16}$ and $\frac{0.1}{9}$ respectively.

X_1 is located in the area of $y=0, y=1$ and $y=2$, the normalized possibilities of being in categories are as follows:

$y=0$	$y=1$	$y=2$
0	1	0

X_2 is located in the area of $y=0, y=1$ and $y=2$, the normalized possibilities of being in categories are as follows:

$y=0$	$y=1$	$y=2$
0.3780	0.4961	0.1260.

X_3 is located in the area of $y=1$ and $y=2$, the normalized possibility of being in categories $y=0, y=1$ and $y=2$ are 0, 0.7975 and 0.2025.

$y=0$	$y=1$	$y=2$
0	0.7975	0.2025.

X_4 is located in the area of $y=0$ and $y=2$,, the normalized possibility of being in categories $y=0, y=1$ and $y=2$ are 0.75, 0, 0.25 .

$y=0$	$y=1$	$y=2$
0.75	0	0.25.

X_5 is located in the area of $y=2$, the normalized possibility of being in categories $y=0, y=1$ and $y=2$ are 0, 0, 1.

$y = 0$	$y = 1$	$y = 2$
0	0	1

4 Problem 4

(a) Download the MNIST data from the course web page. In addition to the actual data, the package con-tains some functions for easily loading the data into Matlab/Octave/R and for displaying digits. See the README files for details. Load the rst N=5,000 images using the provided function.

Answer:
done !

```
confusion_matrix =

   241     0     0     2     0     1     1     0     0     0
     0   283     0     0     1     1     0     1     0     0
     2     6   216     5     0     1     1     5     5     0
     0     0     3   221     2    11     2     4     5     5
     0     3     0     0   231     0     0     2     0    19
     5     3     0     6     0   183     5     1     1     3
     3     1     0     0     1     1   240     0     0     1
     0     3     2     0     6     0     0   257     0     7
     4     5     9    10     1     8     0     2   196     5
     2     2     0     2    12     1     2    11     0   219
```

Figure 4.1: Confusion matrix of KNN classifier.

(b) Use the provided functions to plot a random sample of 100 handwritten digits, and show the associated labels. Verify that the labels match the digit images. (This is a sanity check that you have the data is in the right format.)

Answer:
The code past the sanity test.

(c) Divide the data into two parts: A 'training set' consisting of the rst 2,500 images (and associated labels),and a 'test set' containing the remaining 2,500 images (and their associated labels).

Answer:
The following code has been used for dividing training set and test set.

```
first_part= 1:N/2;
second_part = 1:N;
second_part = setdiff(second_part,first_part);
```

(d) For each of the ten classes (digits 0-9), compute a class prototype given by the mean of all the images in the training set that belong to this class. That is, select from the training set all images of class '0' and compute the mean image of these; this should look sort of like a zero. Do this for all ten classes, and plot the resulting images. Do they look like what you would expect?

Answer:
They do look like the same as the average of each digit.

(e) For each of the images in the test set, compute the Euclidean distance of the image to all 10 prototypes, and classify the test image into the class for which the distance to the prototype is the smallest. So, if a test image is closer to the prototype for '3' than it is to the prototypes for any of the other digits, predict its class to be '3'. Compute and display the resulting confusion matrix.
Answer:
The confusion matrix is as follows.

```
confusion_matrix =

   228     0     1     2     0     4     6     2     2     0
     0   281     1     0     0     3     0     0     1     0
     2    20   192     4     5     2     2     2    11     1
     1     9     4   193     2    20     0     5    11     8
     1     4     1     0   202     0     5     2     0    40
     9    16     1    14    11   136     2     6     1    11
     5    12    15     0     9     8   198     0     0     0
     1    14     1     0    15     0     0   232     1    11
     2    13     9    25     4    20     1     1   150    15
     3     4     6     2    38     2     1    14     2   179
```

Figure 4.2: Confusion matrix of edulean classifier.

(f) Classify each of the test images with a nearest neighbor classifer: For each of the test images, compute its Euclidean distance to all (2,500) of the training images, and let the predicted class be the class of the closest training image. Compute and display the resulting confusion matrix.
Answer:
The confusion matrix is as follows.

(g) Compute and compare the error rates of both classi ers (the prototype-based classi er and the nearestneighbor classi er). Which is working better? Based on the confusion matrix, which digits are confusedwith each other? Why do you think this is?
Answer:
1) The error rate of edulean distance classifier is 0.2036 and the error rate of NN is 0.0852.
2) The NN classifier is working better.

```
confusion_matrix =

   241     0     0     2     0     1     1     0     0     0
     0   283     0     0     1     1     0     1     0     0
     2     6   216     5     0     1     1     5     5     0
     0     0     3   221     2    11     2     4     5     5
     0     3     0     0   231     0     0     2     0    19
     5     3     0     6     0   183     5     1     1     3
     3     1     0     0     1     1   240     0     0     1
     0     3     2     0     6     0     0   257     0     7
     4     5     9    10     1     8     0     2   196     5
     2     2     0     2    12     1     2    11     0   219
```

Figure 4.3: Confusion matrix of KNN classifier.

3) 5 and 9 are the most confusing digits.
4) The shapes of these two digits are the similar in many cases.