ENGN2520 Homework 3

Ming Xu (Banner ID: B01532164)

Problem1

(a)

We know that: $p(w|T) \propto p(T|w) * p(w)$, where

$p(w)$ is the priori distribution of $\boldsymbol{w}$ and $w \sim N(w|0, aI)$. In this case:

$$p(w) = \frac{1}{(2\pi)^{\frac{D}{2}}} * \frac{1}{|aI|^{\frac{1}{2}}} * \exp\left\{-\frac{1}{2} w^T (aI)^{-1} w\right\} = \frac{1}{(2\pi)^{\frac{D}{2}}} * \frac{1}{a^{\frac{D}{2}}} * \exp\left\{-\frac{1}{2a} w^T w\right\}$$
$$= \frac{1}{(2\pi a)^{D/2}} * exp\{-\frac{1}{2a} w^T w\}$$

$p(T|w)$ is the likelihood. In this case:

$$p(T|w) = \prod_{i=1}^{N} P(y_i|x_i, w) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma}} exp\{-\frac{(y_i - f_w(x_i))^2}{2\sigma^2}\}$$

Combine the equations above:

$$p(w|T) \propto \frac{1}{(2\pi a)^{D/2}} * exp\{-\frac{1}{2a} w^T w\} * \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma}} exp\{-\frac{(y_i - f_w(x_i))^2}{2\sigma^2}\}$$

Let $L(T, w)$ be the right part the equation above, in this case:

$$L(T, w) = \frac{1}{(2\pi a)^{D/2}} * exp\{-\frac{1}{2a} w^T w\} * \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma}} exp\{-\frac{(y_i - f_w(x_i))^2}{2\sigma^2}\}$$

The MAP estimate of $\boldsymbol{w}$ is the vector maximizing the posterior probability of $\boldsymbol{w}$ given $\boldsymbol{T}$:

$$w_{MAP} = \max_w p(w|T) = \max_w L(T, w)$$

Denote $l(T, w)$ as the log of $L(T, w)$, then:

$$l(T, w) = -\frac{D}{2} * \log(2\pi a) + -\frac{1}{2a} w^T w + \sum_{i=1}^{n} (log \frac{1}{\sqrt{2\pi\sigma}} - \frac{(y_i - f_w(x_i))^2}{2\sigma^2})$$

Ignore the constant times to get $\hat{l}(T, w)$:

$$\hat{l}(T, w) = -\frac{1}{2a} w^T w - \sum_{i=1}^{n} \frac{(y_i - f_w(x_i))^2}{2\sigma^2}$$

Therefore,

$$w_{MAP} = \max_w L(T, w) = \min_w \{\frac{1}{2a} w^T w + \sum_{i=1}^{n} \frac{(y_i - f_w(x_i))^2}{2\sigma^2}\}$$

Let $\lambda = \frac{\sigma^2}{a}$, then w_{MAP} can be written as:

$$w_{MAP} = \min_{w}\{\frac{\lambda}{2}w^T w + \frac{1}{2}\sum_{i=1}^{n}(y_i - f_w(x_i))^2\}$$

(b)

$$\lambda = \frac{\sigma^2}{a}$$

Problem2

Let x_i be the ***ith*** training data from the data set ***T***. The likelihood can be written as:

$$L(u|T) = P(T|u) = \prod_{i=1}^{k} P(x_i|u) = \prod_{i=1}^{k} P(x_i|u)$$

Note that, from a data x_i, there are n features, namely $\{x_i^{(1)}, x_i^{(2)} \dots x_i^{(n)}\}$.

Also, $x_i^{(j)}$ distributed according to a Bernoulli distribution with mean u_j, then:

$$P(x_i|u) = \prod_{j=1}^{n} P\left(x_i^{(j)} \middle| u_j\right) = \prod_{j=1}^{n} u_j^{x_i^{(j)}} * (1-u_j)^{1-x_i^{(j)}}$$

Therefore,

$$L(u|T) = \prod_{i=1}^{k}\prod_{j=1}^{n} u_j^{x_i^{(j)}} * (1-u_j)^{1-x_i^{(j)}}$$

Denote $l(T,u)$ as the log of $L(T,u)$, then:

$$l(T,u) = \sum_{i=1}^{k}\sum_{j=1}^{n} x_i^{(j)} * \log(u_j) + \sum_{i=1}^{k}\sum_{j=1}^{n}(1 - x_i^{(j)}) * \log(1-u_j)$$

Taking the derivative of this expression with respect to u_j, we get:

$$\frac{\partial l}{\partial u_j} = \sum_{i=1}^{k}\sum_{j=1}^{n} x_i^{(j)} * \frac{1}{u_j} - \frac{1 - x_i^{(j)}}{1-u_j} = 0$$

Let $k_j = \sum_{i=1}^{k}\sum_{j=1}^{n} x_i^{(j)}$, which is the total number of ones in the ***jth*** feature of all the data in the training set.

$$\hat{u}_{MLE}^{(j)} = \frac{k_j}{n}$$

Problem3

(a)

The result is very similar the result of Problem2.

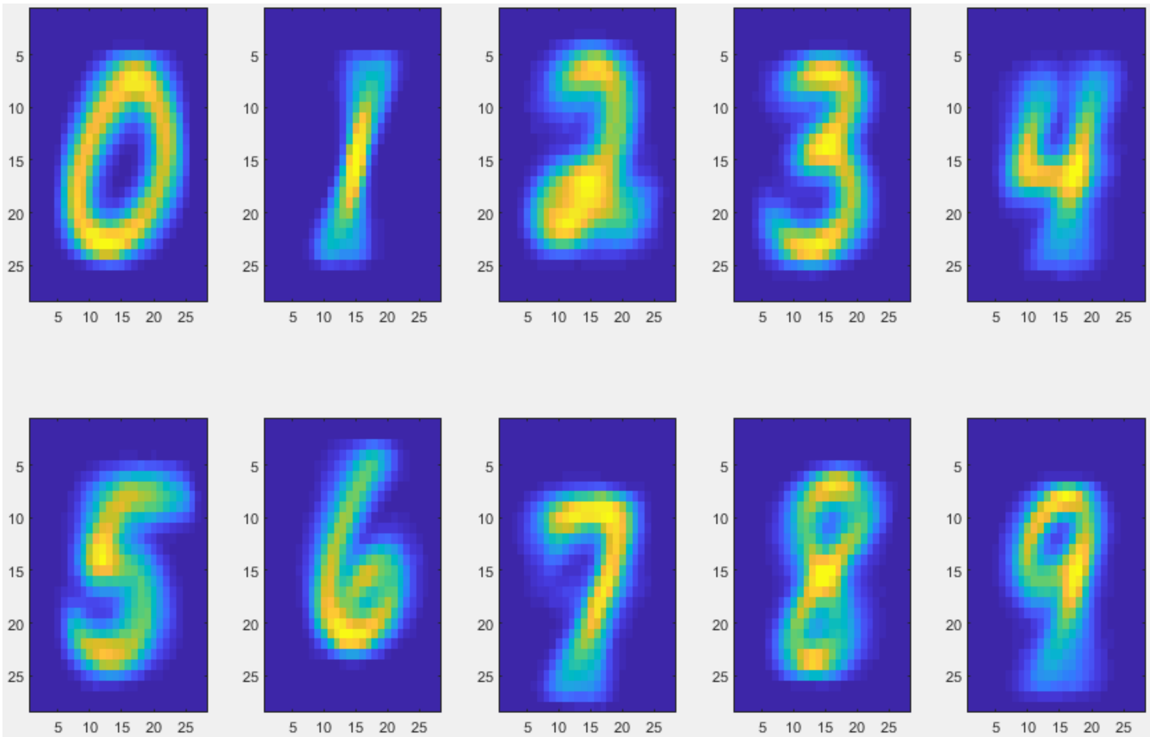
The maximum likelihood estimate for parameters $u_{y,i}$ is:

$$\hat{u}_{MLE}^{(y,i)} = \frac{k_i}{n}$$

Where n is the number of training data of class y and k_i is the total number of ones in the ith pixel of the training set.

(b)

The visualization of the models is shown as below:



(c)

The total number of testing data that is correctly classified is 3942, so the fraction is 78.84%.

The confusion matrix is show as below:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	433	0	4	0	2	33	13	0	14	1
1	0	470	2	3	0	14	4	0	7	0
2	16	12	370	36	8	2	6	11	32	7
3	1	10	4	414	6	23	6	13	10	13
4	4	1	11	0	360	6	13	4	7	94
5	19	1	3	66	19	344	7	5	16	20
6	13	8	30	0	6	33	407	0	3	0
7	8	16	8	4	15	0	2	389	9	49
8	4	16	13	50	13	18	1	4	346	35
9	3	7	4	8	49	7	0	7	6	409

(d) Source code

1. Function to train the classification model: "trainModelForDigit.m"

```
function [model] = trainModelForDigit(trainSet)
    [rowNum, ~] = size(trainSet);
```

```
    model = sum(trainSet)/rowNum;
end
```

2. Function to calculate the probability of the given data and given class:" calculateProbability.m"

```
function [probability] = calculateProbability(data,model)
    result1 = model.^data;
    model = 1-model;
    data = 1-data;
    result2 = model.^data;
    result = result1.*result2;
    probability = prod(result);
end
```

3. Function to classify the testing set by models trained on the training set:" classifyDigit.m"

```
function [classifyResult] = classifyDigit(testData,models)
    %initialization
    [dataCount,~] = size(testData);
    result = zeros(dataCount,10);
    classifyResult = zeros(1,10);

    %loop all testing data
    for row = 1:dataCount
        %loop each class to calculate probability
        for col =  1:10
            result(row,col) = 
calculateProbability(testData(row,:),models(col,:));
        end
        %find the max value and index of the probability
        [~, index] = max(result(row,:));

        %classify
        classifyResult(1,index) = classifyResult(1,index) + 1;
    end
end
```

4. Main function:"main.m"

```
load 'digits';
models = zeros(10,784);

%% Train model for each digit
for i = 1:10
    traindata = sprintf('%s%d','train',i-1);
    models(i,:) = trainModelForDigit(eval(traindata));
end
```

```
%% Draw the visualization of the models
for i = 1:10
    subplot(2,5,i);
    imagesc(reshape(models(i,:),28,28)');
end

%% Classify each testing set
confusion = zeros(10,10);
correct = 0;

for i = 1:10
    testdata = sprintf('%s%d','test',i-1);
    confusion(i,:) = classifyDigit(eval(testdata),models);

    correct = correct + confusion(i,i);
end
```