Introduction to Data Mining: Assignment #3

Junpei Zhou

3140102286

1. Neural Networks

After finishing the code, I use “gradient_check.m” to check the correctness of my computation and get the result shown as below, which prove the correctness of my code.

```
>> gradient_check
2.70e-03, 2.70e-03, 2.70e-03
7.26e-03, 7.26e-03, 7.26e-03
-9.70e-03, -9.70e-03, -9.70e-03
2.82e-03, 2.82e-03, 2.82e-03
-7.55e-03, -7.55e-03, -7.55e-03
-7.95e-03, -7.95e-03, -7.95e-03
4.35e-03, 4.35e-03, 4.35e-03
9.12e-03, 9.12e-03, 9.12e-03
3.85e-03, 3.85e-03, 3.85e-03
4.44e-03, 4.44e-03, 4.44e-03
```

The final result of training my three layer neural network is:

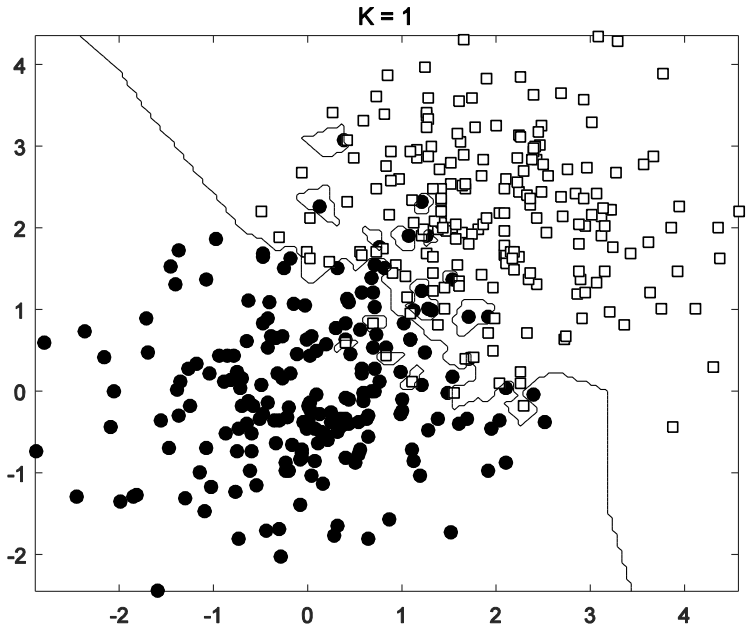
loss: 2.643e-01

accuracy: 0.920000

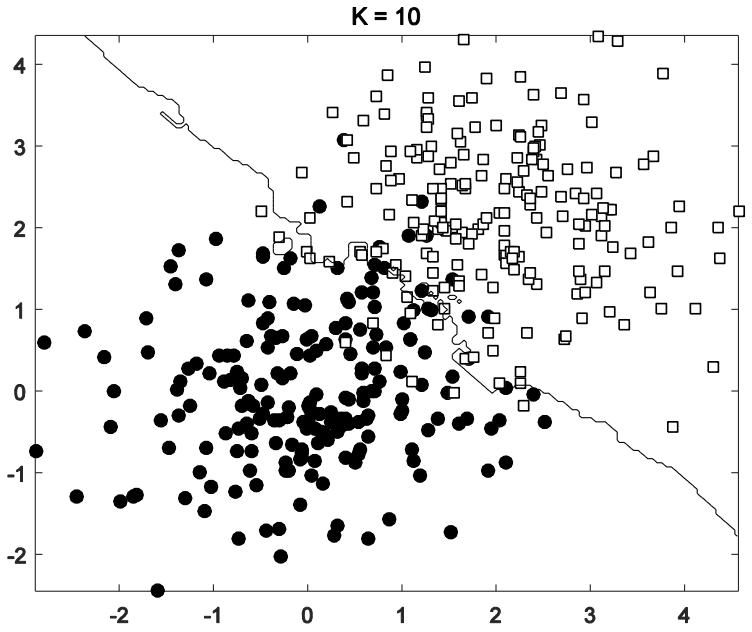
2. K-Nearest Neighbor

a) Try KNN with different K and plot the decision boundary

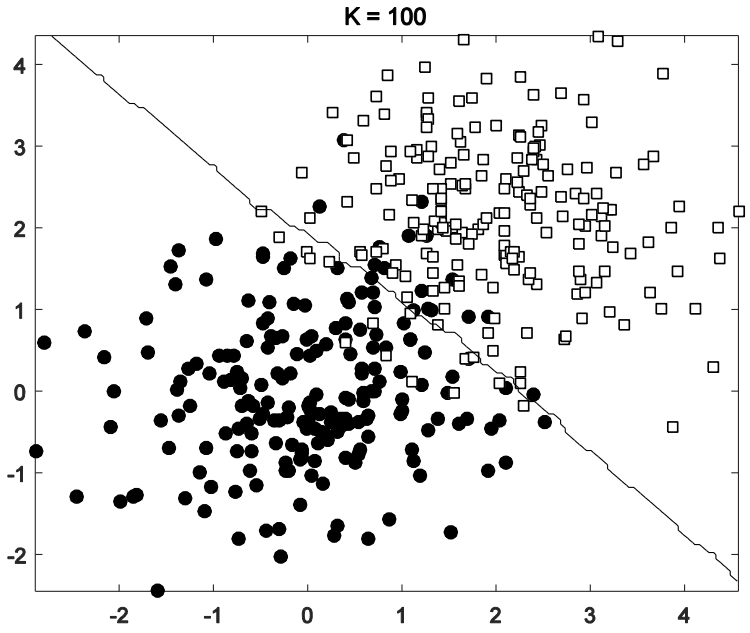
When K=1, the plot is shown as below:



When K=10, the plot is shown as below:



When K=100, the plot is shown as below:

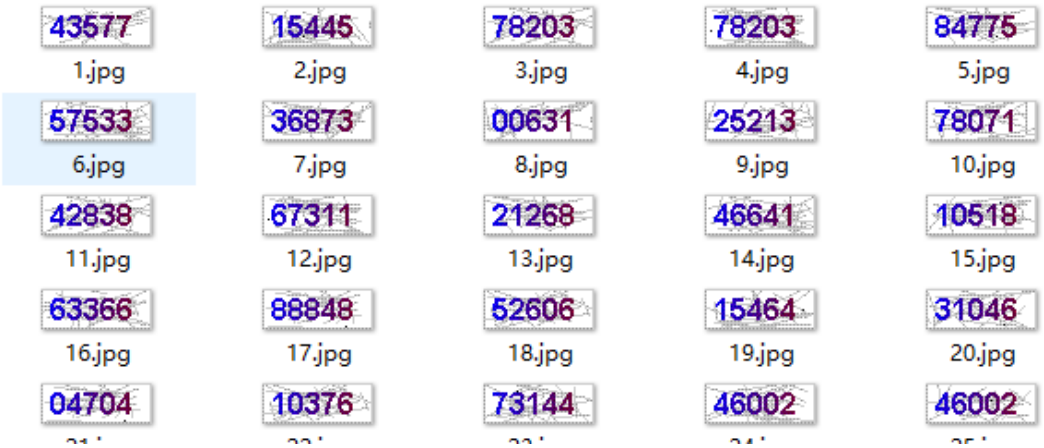


b) How can you choose a proper K when dealing with real-world data

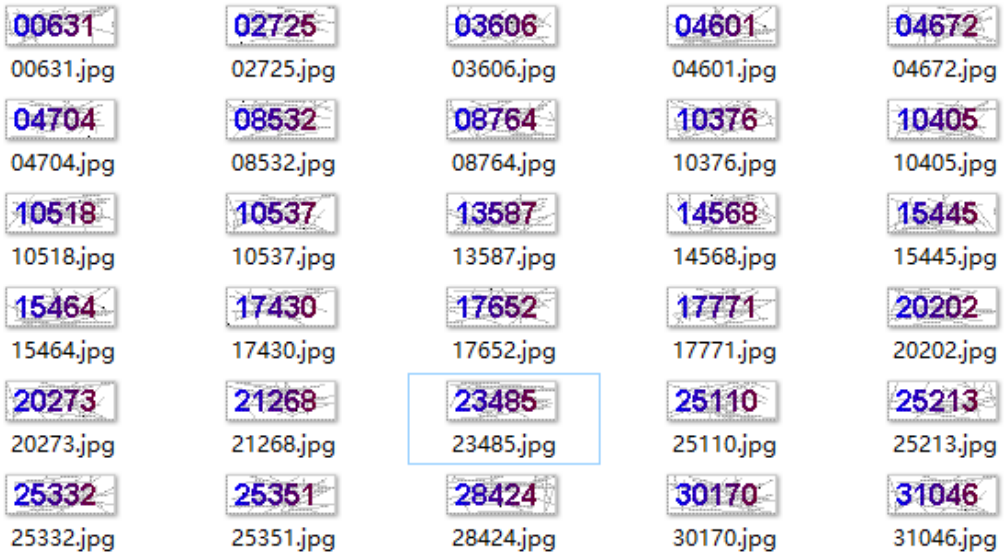
We can split the dataset into the training set and validation set. Then we can test different K on the validation dataset and choose the K which performs best on the validation dataset. What's more, the average silhouette of the data is another useful criterion for assessing the natural number of clusters.

c) Use KNN algorithm to hack the CAPTCHA

首先查看教务网的验证码对应的 URL，发现它是通过访问一个 URL 来产生的：http://jwbinfosys.zju.edu.cn/CheckCode.aspx。因此直接用 python 抓了一百张下来当做训练集，python 代码在“get_capcha.py”中，这些验证码图片存放在同个文件夹中，如下图所示。



然后再对其进行人工标注，把文件名改为数字的标注结果，当然有极少量的几张是重复的，而且重复的都是序号相邻的两张，估计是写爬虫的时候应该每爬一张 sleep 一下。把重复的验证码图片删去，最后剩余 90 张，如下图所示。



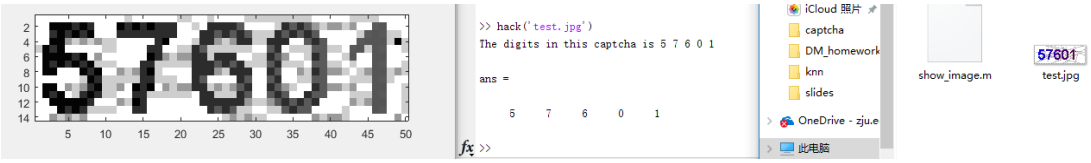
利用 listing = dir('captcha/*.jpg')可以把所有文件名放入到一个结构体里，所有用于生成训练数据的操作都在 get_traindata.m 里面。顺便吐槽一下，在统计的时候发现验证码是没有 9 的，严重怀疑是教务网的程序员写生成验证码程序的时候 9 作为边界没有取到。

然后测试了一些验证码，都是完全正确的，主要是因为教务系统这个验证码实在是太友好了。比如我用和 hack.m 同一目录下的 test.jpg 进行测试，效果如下：

```
>> hack('test.jpg')
The digits in this captcha is 5 7 6 0 1

ans =

     5     7     6     0     1
```



3. Decision Tree and ID3

I calculate the Entropy of each set by the script 'ID3.py', and I choose 2 as the base of the log during the calculation.

The entropy of the original dataset is

$$E_0 = -\frac{4}{9}\log\left(\frac{4}{9}\right) - \frac{5}{9}\log\left(\frac{5}{9}\right) = 0.99108$$

The entropy of the new set divided according to "Gender" is

$$E(Female) = -\frac{105}{205}\log\left(\frac{105}{205}\right) - \frac{100}{205}\log\left(\frac{100}{205}\right) = 0.99957$$

$$E(Male) = -\frac{95}{245}\log\left(\frac{95}{245}\right) - \frac{150}{245}\log\left(\frac{150}{245}\right) = 0.96333$$

$$E(Gender) = \frac{205}{450} * \big(E(Female)\big) + \frac{245}{450} * \big(E(Male)\big) = 0.97984$$

The entropy of the new set divided according to "GPA" is

$$E(High) = -\frac{185}{235}\log\left(\frac{185}{235}\right) - \frac{50}{235}\log\left(\frac{50}{235}\right) = 0.74674$$

$$E(Low) = -\frac{15}{215}\log\left(\frac{15}{215}\right) - \frac{200}{215}\log\left(\frac{200}{215}\right) = 0.36506$$

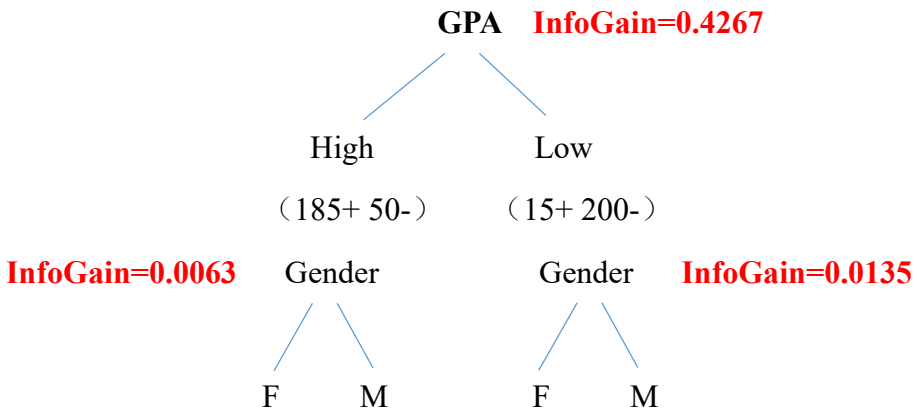
$$E(GPA) = \frac{235}{450} * \big(E(High)\big) + \frac{215}{450} * \big(E(Low)\big) = 0.56438$$

So the info gained by these two features are

$$Gain(S, Gender) = E_0 - E(Gender) = 0.0112$$

$$Gain(S, GPA) = E_0 - E(GPA) = 0.4267$$

So we should choose "GPA" as the feature to split the dataset at first, which means the split criteria of the tree's root is "GPA".



To computer the Info Gained by using Gender to divide the set of "High", the original entropy is

$$E_{high} = -\frac{185}{235}\log\left(\frac{185}{235}\right) - \frac{50}{235}\log\left(\frac{50}{235}\right) = 0.74674$$

The entropy of the new set divided according to "Gender" is

$$E_{high}(Female) = -\frac{95}{115}\log\left(\frac{95}{115}\right) - \frac{20}{115}\log\left(\frac{20}{115}\right) = 0.66658$$

$$E_{high}(Male) = -\frac{90}{120}\log\left(\frac{90}{120}\right) - \frac{30}{120}\log\left(\frac{30}{120}\right) = 0.81128$$

$$E_{high}(Gender) = \frac{115}{235} * \left(E_{high}(Female)\right) + \frac{120}{235} * \left(E_{high}(Male)\right) = 0.74047$$

To computer the Info Gained by using Gender to divide the set of "Low", the original entropy is

$$E_{low} = -\frac{15}{215}\log\left(\frac{15}{215}\right) - \frac{200}{215}\log\left(\frac{200}{215}\right) = 0.36506$$

The entropy of the new set divided according to "Gender" is

$$E_{low}(Female) = -\frac{10}{90}\log\left(\frac{10}{90}\right) - \frac{80}{90}\log\left(\frac{80}{90}\right) = 0.50326$$

$$E_{low}(Male) = -\frac{5}{125}\log\left(\frac{5}{125}\right) - \frac{120}{125}\log\left(\frac{120}{125}\right) = 0.24229$$

$$E_{low}(Gender) = \frac{90}{215} * \left(E_{low}(Female)\right) + \frac{125}{215} * \left(E_{low}(Male)\right) = 0.35153$$

So the Info Gained by using Gender to divide the set of "High" and the set of "Low" are

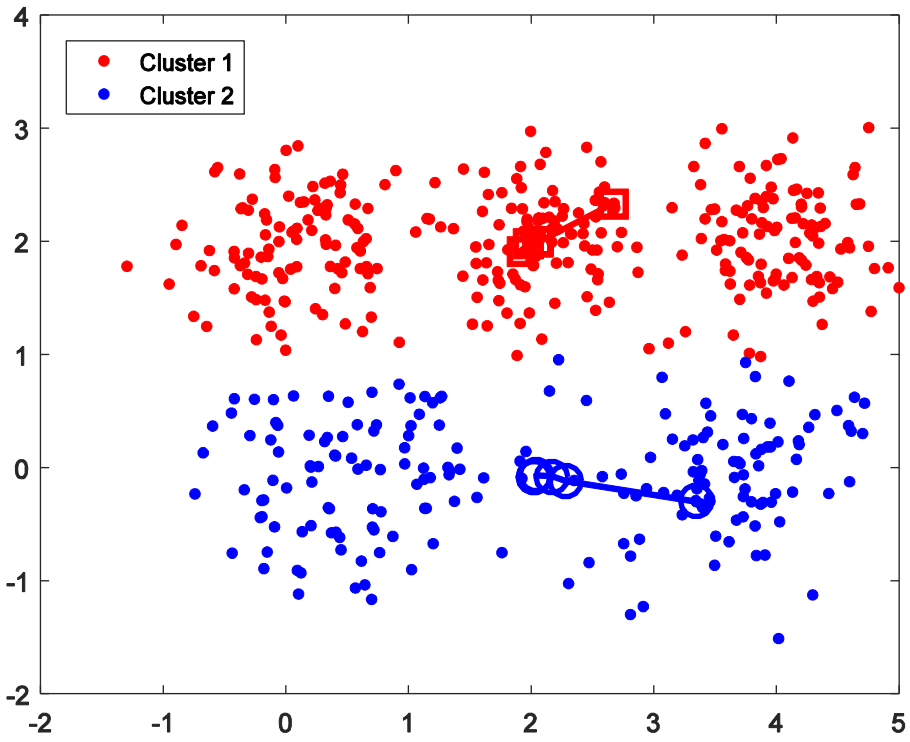
$$Gain\left(S_{High}, Gender\right) = E_{high} - E_{high}(Gender) = 0.0063$$

$$Gain(S_{low}, Gender) = E_{low} - E_{low}(Gender) = 0.0135$$

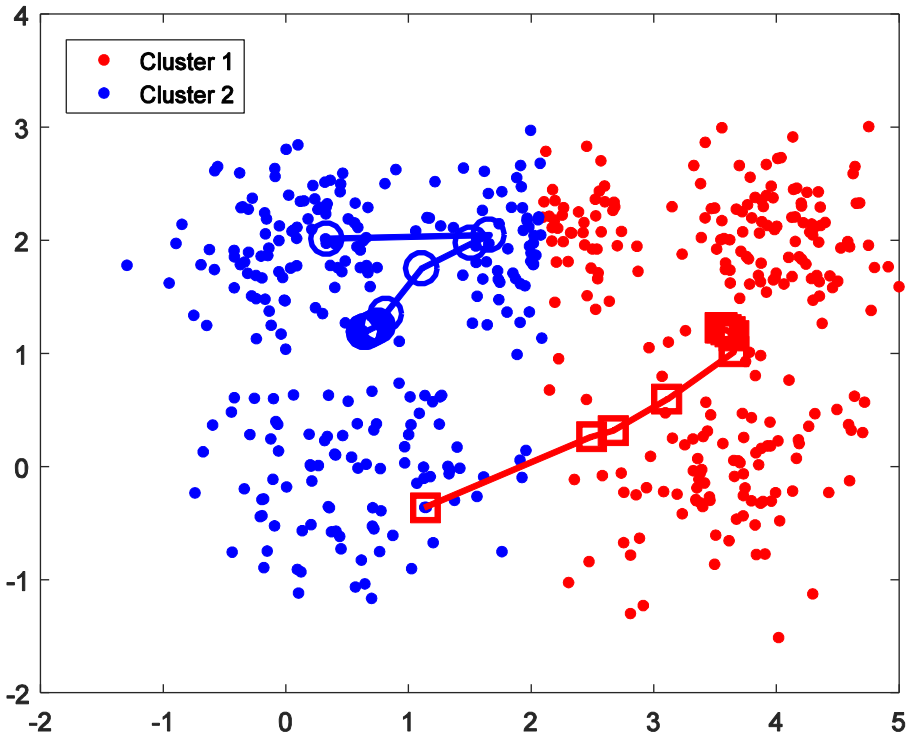
4. K-Means Clustering

a) Run your k-means algorithm on kmeans_data

I implement the code about this problem in a new file named “run.m”. With K=2 and I repeat the experiment for 1000 times, the trial with largest SD is plotted as below



With K=2 and I repeat the experiment for 1000 times, the trial with smallest SD is plotted as below



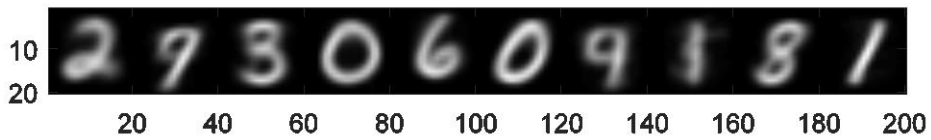
b) How can we get a stable result using k-means

We can repeat the experiment for many times and pick up the result with the lowest TSD. There are also many kinds of techniques about the stability of the k-means like the Alexander Rakhlin and Andrea Caponnetto's Stability of K-Means Clustering[1] and so on.

c) Run your k-means algorithm on the digit dataset

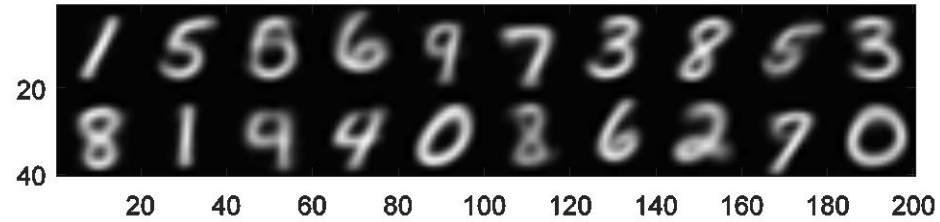
I implement the code about this problem in a new file named "run_digit.m".

When K = 10, the centroid is

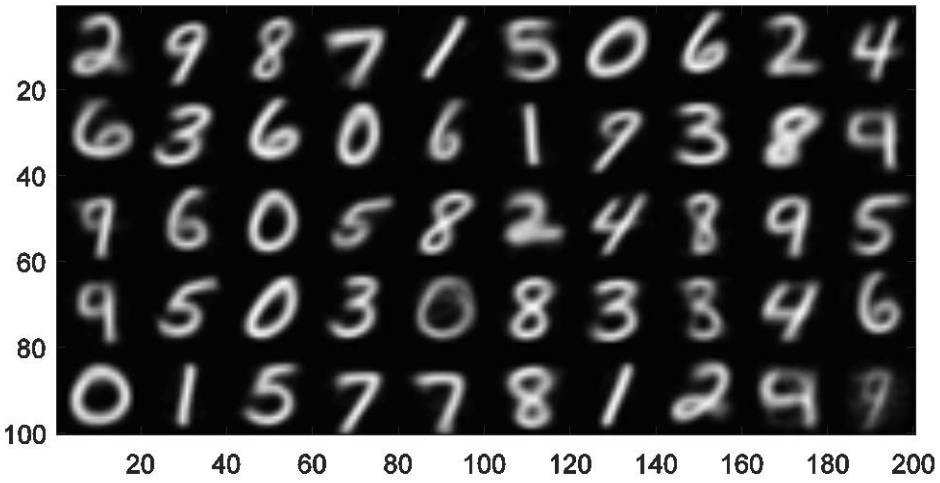


When K = 20, the centroid is

[1] http://www-stat.wharton.upenn.edu/~rakhlin/papers/stability_clustering.pdf



When K = 50, the centroid is



d) Vector quantization

I use my own images and here is the original image

I compress the image with K set to 8, 16, 32, and 64. Here are the results, and it is obvious that the visual quality of K=8 is much lower.

K = 8

K=16

K=32

K=64

Compress ratio

If we set K=64, there will remain 64 clusters, which means there are 64 different colors, so we need $\log_2 64 = 6$ bits to store one pixel. The pixel in original picture has 3*8=24 bits. So the compress ratio is $\frac{6}{24} = \frac{1}{4}$