Introduction to Data Mining: 21191441
Zhejiang University
Professor Deng Cai

May 27, 2019
Homework 3

Homework 3

Collaborators:
Name: Junlin Yin
Student ID: 3160104340

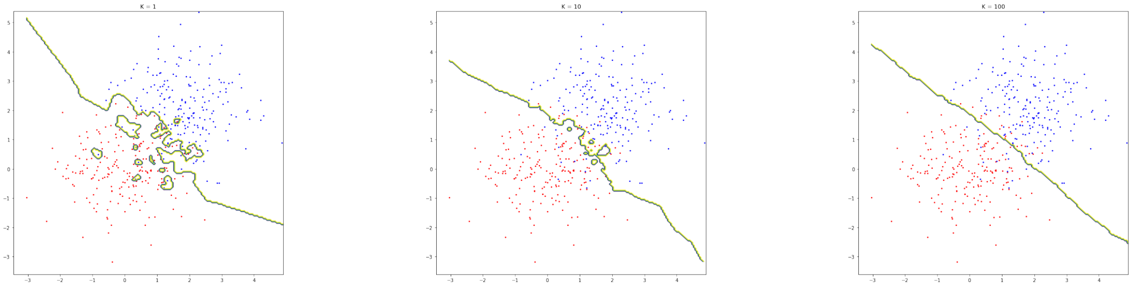
Problem 3-1. Neural Network

(a) Answer: Test accuracy: 92.4%

Problem 3-2. K-Nearest Neighbor

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

Answer:



(a) $K = 1$ (b) $K = 10$ (c) $K = 100$

Figure 1: plot results

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

Answer: We can introduce validation set to choose a proper K.
In addition, when the boundary looks underfitting, we should lower K; when the boundary looks overfitting, we should higher K.

(c) Now let us use KNN algorithm to hack the CAPTCHA of a website that we are all familiar with.

Answer: Note: I use script 'spider.py' to fetch 100 checkcodes from the website and save them to foler 'checkcode/'. Then I use script 'label.py' to label them one by one and save train data to the file 'hack_data.npz'. Finally, I call 'hack.py' to get the 'CheckCode.aspx' and test it. In my test case, the system works well when $K = 10$.

Problem 3-3. Decision Tree and ID3

(a) Answer:

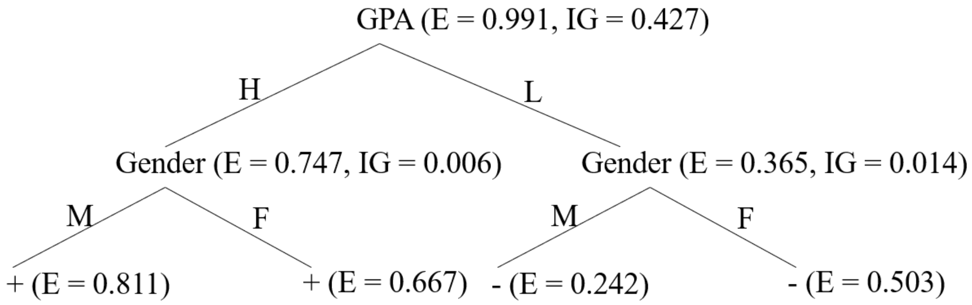
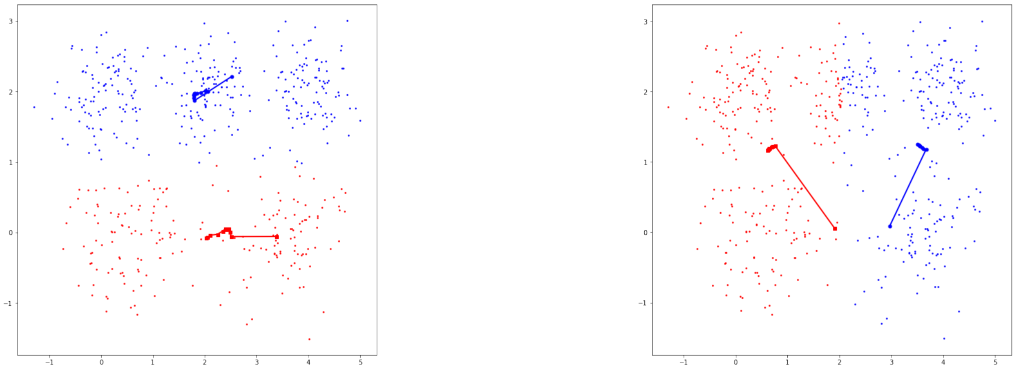

GPA (E = 0.991, IG = 0.427)
H
L
Gender (E = 0.747, IG = 0.006)
Gender (E = 0.365, IG = 0.014)
M
F
M
F
+ (E = 0.811)
+ (E = 0.667)
- (E = 0.242)
- (E = 0.503)

Figure 2: Dicision Tree (E is denoted to entropy and IG is denoted to information gain)

Problem 3-4. K-Means Clustering

(a) Visualize the process of k-means algorithm for the two trials with largest and smallest SD.

Answer:



(a) max SD

(b) min SD

Figure 3: kmean two results

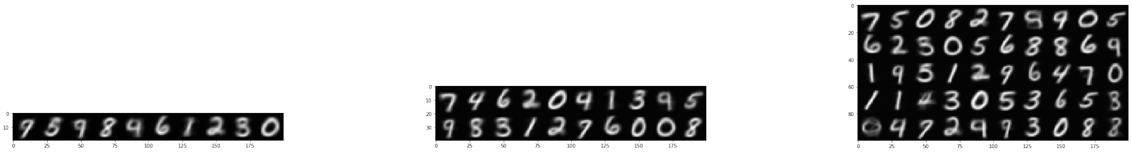
(b) How can we get a stable result using k-means?

Answer:

1. Try a lot of times, and choose the one with lowest TSD value like we did in last question.
2. Use validation set to determine a proper K.

(c) Visualize the centroids using show_digit.py.

Answer:



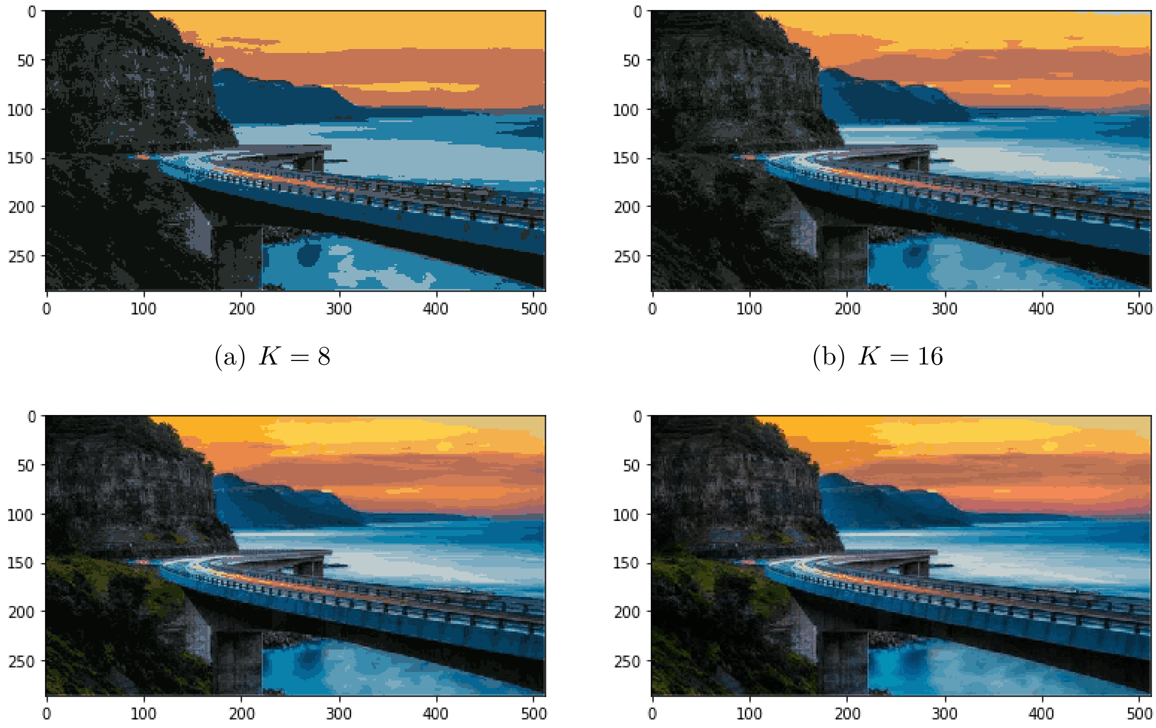
(a) $k = 10$ (b) $k = 20$ (c) $k = 50$

Figure 4: kmeans digit results

(d) Compress images with K set to 8, 16, 32 and 64. What is the compress ratio if we set K to 64?

Answer:

Figure 5: Original Image



(a) $K = 8$

(b) $K = 16$

(c) $K = 32$

(d) $K = 64$

Figure 6: Result Images with different K

In normal case, when K is set to 64, we need 6 bits to represent a pixel instead of 24 bits, so the compress ratio is 25%. You also need to store the centroids

If we consider Huffman encoding, we should know how many pixels belong to each cluster. Besides, we should write a program to construct Huffman-tree and calculate the total bits needed. In order to reach the goal, I store an array 'cnt' to a file named 'cnt.npz'. This array tells us the number of pixels of each cluster. Then I wrote a python script named 'huffman_encode.py' which is zipped along with the code part as well. In this program, I load the 'cnt' array and construct a Huffman-tree, and then calculate:

$$\sum_{i \in leaves} n_i d_i$$

Here n_i means the number of pixels that belong to this leaf node, and d_i means the depth of this leaf node, i.e., the number of bits we need to represent this cluster. The summation equals the number of total bit used to represent the whole image.

According to 'huffman_encode.py', we only need 841663 bits compared with 881664 bits when we don't use Huffman encoding, and the compress ratio reduces to 23.9%.

Excellent! You got 10 points!