Machine Learning: 2124095
Zhejiang University — Oct 15, 2019
Professor Deng Cai — Homework 3

Homework 3

Collaborators:
Name: Zhaoyi
Student ID: 21921266

Problem 3-1. Neural Networks

In this problem, we will implement the feedforward and backpropagation process of the neural networks.

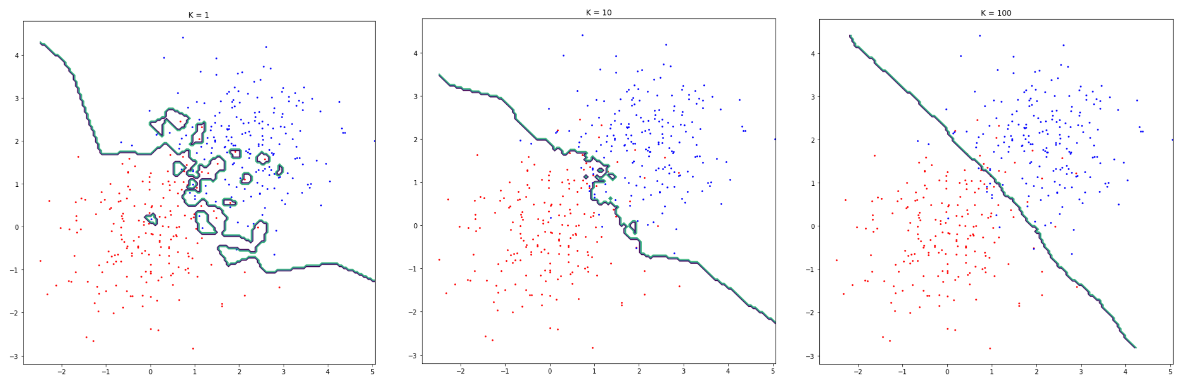
(a) **Answer:** Test accuracy: 94.2%

Problem 3-2. K-Nearest Neighbor

In this problem, we will play with K-Nearest Neighbor (KNN) algorithm and try it on real-world data. Implement KNN algorithm (in *knn.m/knn.py*), then answer the following questions.

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

Answer: see fig 1



(a) k=1 (b) k=10 (c) k=100

Figure 1: Knn decision boundary with different K

(b) We have seen the effects of different choices of K. How can you choose a proper K when dealing with real-world data ?

Answer: Generally speaking, the larger K, the higher bias, the lower variance.
If k is too small, test sample is only influenced by few neighbors. If the neighbors are noise, model will give a wrong answer. In this case, model is very complicated and

prone to be over-fitting.
If k is too large, test sample is also influenced by those points that far away from it. Model discard much useful local information. In this case, model is very simple and prone to be under-fitting.
In practice, we can use cross-validation to choose k values. First choose a set of k (usually small), then use the k having the best performance on validation set.

(c) Finish *hack.m/hack.py* to recognize the CAPTCHA image using KNN algorithm.

Answer: I download and label 45 captchas, 25 of which are used for training and the remaining 20 are used for testing which means size of train set is 100 and size of test set is 80. This data is saved as hack_ data.npz, see the code for details.
The test accuracy is 73/80 = 91.25%

Problem 3-3. Decision Tree and ID3

Consider the scholarship evaluation problem: selecting scholarship recipients based on gender and GPA. Given the following training data:

Answer: see fig 2

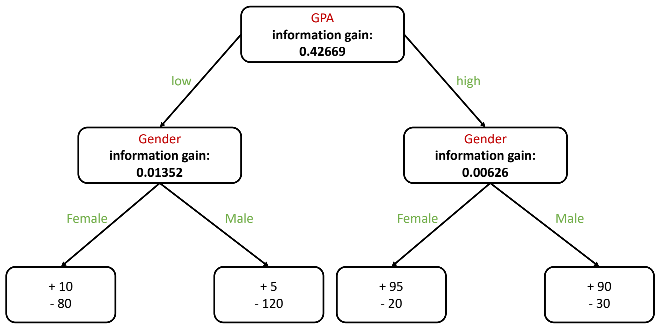

GPA
information gain: 0.42669
low
high
Gender
information gain: 0.01352
Gender
information gain: 0.00626
Female
Male
Female
Male
+ 10
- 80
+ 5
- 120
+ 95
- 20
+ 90
- 30

Figure 2: Decision tree generated by ID3

Problem 3-4. K-Means Clustering

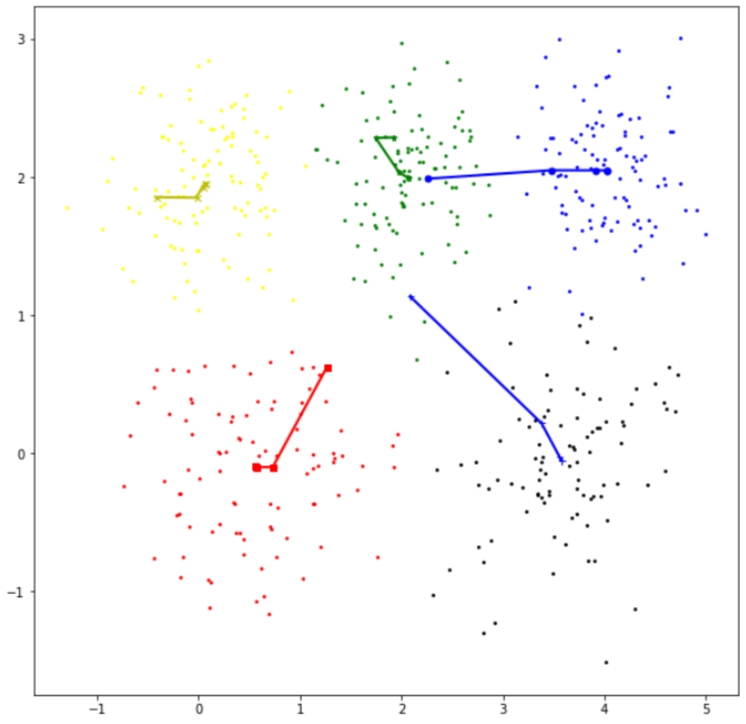
Finally, we will run our first unsupervised algorithm – k-means clustering.

(a) Visualize the process of k-means algorithm for the two trials.

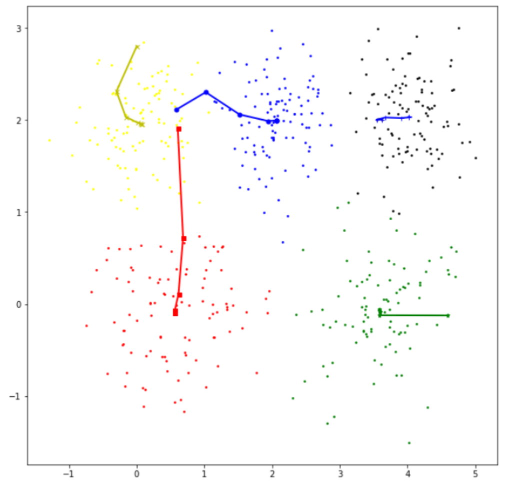
Answer: see fig 3

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

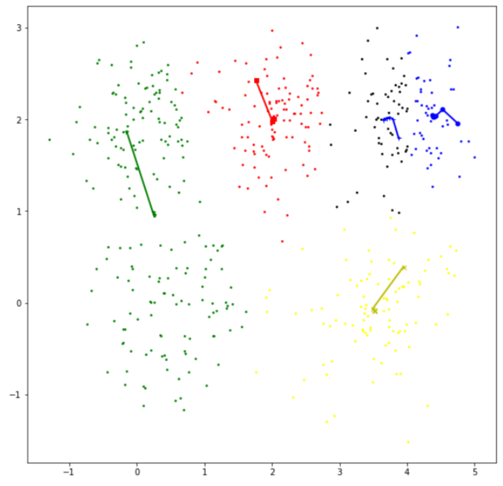
Answer:
A very simple way is choosing the number of clusters by visually inspecting data



(a) result of kmeans (k=5)



(b) result with smallest SD(in 1000 trial)



(c) result with largest SD(in 1000 trial)

Figure 3: Kmeans visualization

points especially when dimension of data and number of sample is not too large.
A more scientific way is Elbow Method. First of all, compute the sum of squared error (SSE) for some values of k (for example 2, 4, 6, 8, etc.). The SSE is defined as the sum of the squared distance between each member of the cluster and its centroid. Mathematically:

$$SSE = \sum_{i=1}^{K} \sum_{x \in c_i} dist\left(x, c_i\right)^2$$

SSE decreases as k gets larger. The idea of the elbow method is to choose the k at which the SSE decreases abruptly.

(c) Visualize the centroids.

Answer:

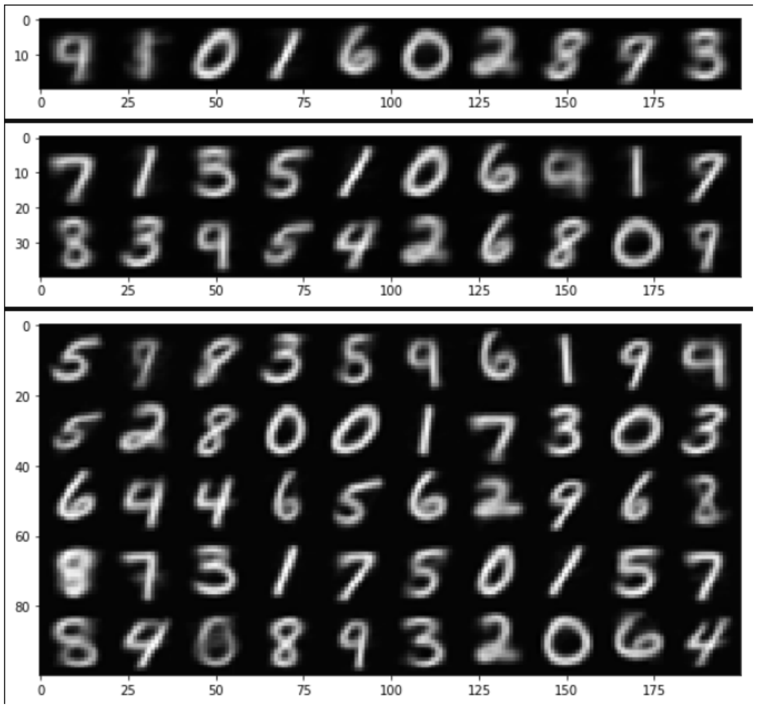
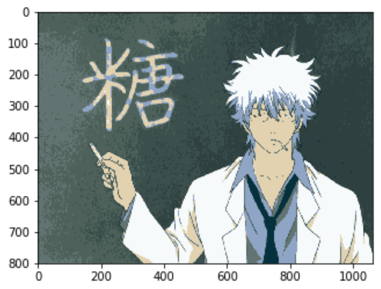


Figure 4: Centroids, 3 images from top to bottom correspond to k=10, k=20, k=30 respectively

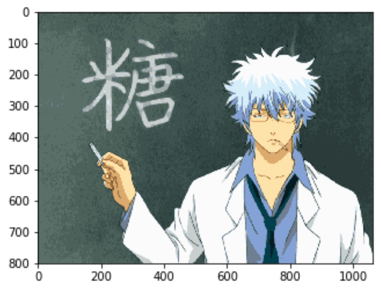
(d) Vector quantization.

Answer: See fig 5
In theory, for all 3-channel 256 color imagesone pixel can be represent with $log_2(64) =$ 6 bits, thus the compress ratio is 6/(3*8)=25%.
For the picture above, I write a function called huffman_ encoding in kmeans/huffman_ encoding.py to compute the compress ratio. With Huffman encoding(without considering the map of the code and value), compress ratio is

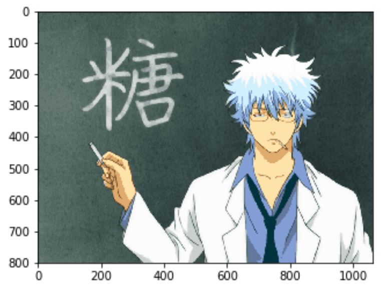
$$4646254/(800 * 1062 * 3) = 22.786\%$$



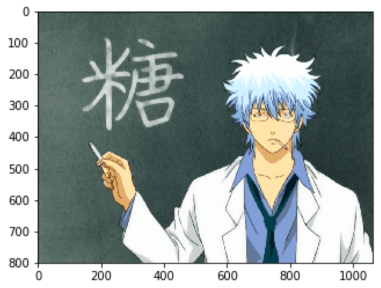
(a) k=8



(b) k=16



(c) k=32



(d) k=64

Figure 5: Images compressed with kmeans.