CS589: Machine Learning - Fall 2018

Homework 5: K-Means

Assigned: December 7^{th} Due: December 14^{th}

Getting Started: In this assignment, you will perform unsupervised learning to compress an image. **Please install Python 3.6 via Anaconda on your personal machine**. Download the homework file HW05.zip via Piazza. Unzipping this folder will create the directory structure shown below,

```
HW05
--- HW05.pdf
--- Data
    |--Scene
--- Submission
    |--Code
```

The data files are in 'Data' directory respectively. You will write your code under the Submission/Code directory. Make sure to put the deliverables (explained below) into the respective directories.

Deliverables: This assignment has two types of deliverables:

- **Report:** The solution report will give your answers to the homework questions (listed below). Feel free to use as many pages as you wish, so long as you assign them correctly on Gradescope. If a question asks for any figures or tables, please include them in your report. You can use any software to create your report, but your report must be submitted in PDF format.
- **Code:** The second deliverable is the code that you wrote to answer the questions, which will involve performing unsupervised learning. Your code must be in Python 3.6 (no iPython notebooks or other formats). You may create any additional source files to structure your code. However, you should aim to write your code so that it is possible to reproduce all of your experimental results exactly by running *python run_me.py* file from the Submissions/Code directory.

Submitting Deliverables: When you complete the assignment, you will upload your report and your code using the Gradescope.com service. Place your final code in Submission/Code. Finally, create a zip file of your submission directory, Submission.zip (NO rar, tar or other formats). Upload this single zip file on Gradescope as your solution to the 'HW05-KMeans-Programming' assignment. Gradescope will run checks to determine if your submission contains the required files in the correct locations. Finally, upload your pdf report to the 'HW05-KMeans-Report' assignment. When you upload your report please make sure to select the correct pages for each question respectively. Failure to select the correct pages will result in point deductions. The submission time for your assignment is considered to be the later of the submission timestamps of your code and report submissions.

Academic Honesty Statement: Copying solutions from external sources (books, internet, etc.) or other students is considered cheating. Sharing your solutions with other students is also considered cheating.

Posting your code to public repositories like GitHub, stackoverflow is also considered cheating. Any detected cheating will result in a grade of -100% on the assignment for all students involved, and potentially a grade of F in the course.

Task:

Unsupervised learning: Contrary to supervised learning (classification, regression), unsupervised learning algorithms learn patterns from unlabeled examples. There are several popular algorithms in unsupervised learning such as principal component analysis (PCA), k-means, independent component analysis (ICA) and density estimation. In this project you will use k-means to compress images.

Algorithms and datasets:

k-means: Is a clustering algorithm that finds centroids of clusters and assigns each sample to one and only one of these clusters according to some criteria. You will use k-means to compress an image of Times Square as shown in Figure 1.

Figure 1: Image to compress using k-means

Questions:

1. (*15 points*) K-means:

(5) a. K-means is a simple unsupervised learning algorithm that splits the data into clusters. There are different ways to determine the "optimal" number of clusters; the elbow rule being a very simple one. Explain it in at most 4 sentences.

(5) b. The pairwise objective gives the total pairwise distance of points within each cluster. Here is the pairwise loss with squared Euclidean distance and K clusters $c_1, c_2 ... c_K$:

$$W(c_1, c_2, ...c_K) = \sum_{k=1}^{K} \sum_{i:c_i=k} \sum_{j:c_j=k} ||x_i - x_j||^2$$

The centroid objective gives the total distance of points from the centroid of each cluster. Here is the centroid loss with squared Euclidean distance and K clusters $c_1, c_2 ... c_K$:

$$F(c_1, c_2, ...c_K) = \sum_{k=1}^{K} \sum_{i:c_i=k} ||x_i - \bar{x}_k||^2$$

where

$$\bar{x}_k = \frac{1}{N_k} \sum_{i:c_i=k} x_i$$

In lecture, Brendan claimed that the centroid loss is equivalent to the pairwise loss. Is this actually true? In other words, does $W(c_1, c_2, ..c_K) = F(c_1, c_2, ..c_K)$ for any K and any assignment of clusters? Prove your result.

(5) c. Explain the relationship between pairwise loss and centroid loss. Under what conditions you would expect them to give similar behavior? Under what conditions would they give different behavior?

2. (*35 points*) **K-means Image Compression:** You are given an RGB image *times_square.jpg* as a $400 \times 400 \times 3$ matrix. Each pixel can be seen as a sample of dimension 3 (3 integers between 0 and 255, one for each component of RGB). For this question you will treat each pixel as a data sample. You are encouraged to use *sklearn's* implementation of *kmeans* for this question.

Figure 2: Example of reconstructed image using 5 clusters

(10) a. Apply k-means using k clusters in the range $\{2, 5, 10, 25, 50, 75, 100, 200\}$ (note that in this case each cluster will represent an RGB color triplet). Replace each pixel in the original image with the centroid of the cluster assigned to that pixel. In your report, make a 3×3 grid plot of the original times square image along with eight reconstructed images corresponding to eight different values of k. Make sure to label the images. An example of the original image and reconstructed image is shown in Figure 2. Helper code is given in 'run_me.py' file to convert the $400 \times 400 \times 3$ matrix to $160,000 \times 3$ matrix and vice versa.

(10) b. Make a plot of the sum of squared errors of each pixel to its respective cluster centroids for different values of k (elbow plot). If the range is too big then make the plot in $\log$ space, but make sure to indicate that in your report.

(2.5) c. In at most three sentences, discuss and comment on your results from (a) and (b). How does k affect the quality of the reconstructed images? How does SSE change with k?

(10) d. Another way to assess image compression is to look at the compression rate. The compression rate is the memory required to store the compressed image divided by the memory required to store the original image. Note that in this case each pixel of the original image uses 24 bits, each centroid is represented by 3 *floats* (each one uses 32 bits), and an integer from 1 to k needs $ceiling(\log_2 k)$ bits (for each pixel in the image you store the index of the centroid assigned). For each value of k report the compression rate using the table below. In python, you can use numpy's nbytes to get the number of bytes consumed by an array, vector, etc.

For each value of k, report the compression rate in a format like the table below.

	Compression Rate
$k = 2$	
$k = 5$	
$k = 10$	
$k = 25$	
$k = 50$	
$k = 75$	
$k = 100$	
$k = 200$	

(2.5) e. In at most two sentences, describe and comment on your reported compression rates. How does k affect the compression rate?